NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Afstudeerverslag

Ontwikkeling van een Requirement Capture System voor NAVO software prototype systemen

door
J.J. van Houten

10 juni 2005

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Afstudeerverslag

Ontwikkeling van een Requirement Capture System voor NAVO software prototype systemen

door
J.J. van Houten

examinatoren
P.C.M. Borsboom
A.G.J. Vinkesteijn

bedrijfsmentoren
C.D. Allmon
A.A.M. Schoonen

10 juni 2005

Referaat

Van Houten, J.J., Afstudeerverslag, Ontwikkeling van een Requirement Capture System voor NAVO software prototype systemen, NAVO/NATO, NATO C3 Agency/NC3A, Den Haag, 10 juni 2005.

In het kader van de opleiding Informatievoorzieningen en Informatietechnologie(IVIT) aan de Haagse Hogeschool (HHS) heeft J.J. van Houten een afstudeeropdracht uitgevoerd, genaamd: 'Ontwikkeling van een Requirement Capture System voor NAVO software prototype systemen'. De afstudeer-opdracht is uitgevoerd bij het NATO Consultation, Command & Control Agency (NATO C3 Agency/ NC3A) te Den Haag, gedurende de periode 7 februari 2005 tot en met 10 juni 2005.

Dit verslag dient de lezer inzicht te geven in de werkzaamheden die door J.J. van Houten tijdens het uitvoeren van deze afstudeeropdracht zijn verricht.

Descriptoren:

- Afstudeerverslag
- Afstudeeropdracht
- NAVO/NATO
- NATO C3 Agency/NC3A
- Missile Defence
- LSID
- PlaTo
- Microsoft .NET
- ASP.NET
- VB.NET
- RUP
- UML

Voorwoord

De schrijver van dit verslag is J.J. van Houten, student Informatica en Informatiekunde (I&I)/Informatievoorzieningen en Informatietechnologie (IVIT) aan de Haagse Hogeschool en afstudeerder bij het NATO Consultation, Command & Control Agency (NC3A) te Den Haag.

Dit verslag dient ter verantwoording van de afstudeerperiode. Het verslag is primair vervaardigd voor twee examinatoren van de Haagse Hogeschool en voor een gecommitteerde van buiten de Haagse Hogeschool. Daarnaast is het verslag gericht aan eventuele geïnteresseerden. Er wordt echter wel verondersteld dat de lezer enige kennis heeft met betrekking tot het vakgebied Informatievoorziening en Informatietechnologie.

Hierbij wil ik alle medewerkers van het Command and Control Resource Centre (C2RC) bedanken, zonder wie mijn afstudeerproject niet zo'n leerzame en aangename ervaring zou zijn geweest. In het bijzonder gaat mijn dank uit naar mijn bedrijfsmentoren en tevens opdrachtgevers, dhr. A.A.M. Schoonen en dhr. C.D. Allmon, die mij hebben begeleid en ondersteund gedurende de afstudeer-periode.

10 juni 2005, Den Haag

J.J. van Houten

Inhoudsopgave

Hoofdstuk 1 - Inleiding

Dit verslag is geschreven naar aanleiding van de afstudeeropdracht die J.J. van Houten heeft uitgevoerd bij het NATO Consultation, Command & Control Agency (NATO C3 Agency/NC3A) te Den Haag. Als afstudeeropdracht heeft hij een Requirement Capture System ontwikkeld. Dit is een systeem dat ontwikkelaars van software prototype systemen helpt om systeemeisen te verkrijgen.

Het doel van dit verslag is om inzicht te geven in de omvang en diepgang van de uitgevoerde werkzaamheden binnen het afstudeertraject, de wijze waarop de producten tot stand zijn gekomen, de gemaakte keuzes en overwegingen, de opgetreden problemen, de oplossingen daarvoor en tot slot de leereffecten.

Dit verslag is gericht aan de examinatoren en een lid van de commissie van toezicht. Zij kunnen aan de hand van dit verslag een oordeel geven over de kwaliteit van de uitgevoerde opdracht. Tevens wordt er een exemplaar uitgereikt aan de bedrijfsmentoren en er wordt een exemplaar in de bibliotheek van de Haagse Hogeschool geplaatst. Daarnaast is dit verslag ook bedoeld voor andere geïnteresseerden.

De structuur van dit rapport is als volgt. In Deel I, Bedrijf en opdracht, worden de organisatie en opdracht omschreven. Hoofdstuk 2 beschrijft de organisatie en in hoofdstuk 3 wordt achtergrond-informatie gegeven die relevant is geweest tijdens het uitvoeren van de afstudeeropdracht. De twee software prototype systemen, die door het NATO C3 Agency ontwikkeld worden, worden in hoofdstuk 4 beschreven. Als laatste wordt in dit deel in hoofdstuk 5 de opdracht omschreven en vervolgens in hoofdstuk 6 het Project Plan. Deel II, Werkzaamheden, beschrijft de werkzaamheden die zijn uitgevoerd tijdens het afstudeertraject en is onderverdeeld in de hoofdstukken 7 tot en met 10 waarin de werkzaamheden uit de fasen Start up, Elebaration, Inception en Construction worden beschreven. In Deel III worden de procesgang en de opgeleverd producten geëvalueerd. Een literatuurlijst, verklarende woordenlijst en figurenlijst zijn opgenomen in Deel IV.

Deel I - Bedrijf en opdracht

Hoofdstuk 2 - Beschrijving van de organisatie

In dit hoofdstuk wordt een beschrijving van de organisatie gegeven.

2.1 Noord-Atlantische Verdragsorganisatie (NAVO)

NAVO staat voor Noord-Atlantische Verdragsorganisatie en is internationaal bekend als de North Atlantic Treaty Organisation (NATO). In deze paragraaf zal de NAVO in haar geheel globaal worden beschreven.

2.1.1 Geschiedenis

De NAVO werd in 1949 opgericht nadat de Sovjet Unie de vrije toegang tot de door de westerse Geallieerden bezette zones van Berlijn blokkeerde. Als reactie hierop ondertekenden twaalf staten op 4 april 1949 het Noord-Atlantische verdrag, met als doel de veiligheid van niet-communistische landen van Europa tegen eventuele agressie van de Sovjet-Unie te verzekeren door onderlinge samenwerking en vooral door het waarborgen van Amerikaanse hulp.

Nederland behoorde tot de ondertekenaars, evenals de Verenigde Staten, Groot-Brittannië en Frankrijk. Doel van het verdrag was een collectieve verdediging. Een gewapende aanval op één van de ondertekenaars zou als een aanval op hen allen worden beschouwd.

Sinds de val van de Berlijnse Muur (1989) is de militaire betekenis van de NAVO afgenomen. Een aantal voormalige Oostbloklanden (Tsjechische Republiek, Polen en Hongarije) zijn in 1999 lid geworden, wat een decennium eerder nog volstrekt ondenkbaar zou zijn geweest. De NAVO besloot tijdens de top in Praag (november 2002) om onderhandelingen over de toetreding van Roemenië, Bulgarije, Slowakije, Slovenië, Estland, Letland en Litouwen te beginnen. Deze landen zijn vanaf april 2004 definitief aangesloten bij de alliantie.

2.1.2 Doelstelling

De kern van het verdrag bestaat uit artikel 5, dat stelt dat ingeval van een aanval op één van de staten, dit door de anderen zal worden opgevat als een aanval op allemaal en dat alle landen zullen samenwerken om de aanvaller af te weren. Artikel 5 is tot op heden eenmaal van toepassing verklaard, en wel na de terreuraanvallen op de Verenigde Staten (VS) op 11 september 2001. Tot een feitelijk militair optreden in NAVO-verband kwam het echter niet.

De alliantie wil de laatste jaren niet meer uitsluitend een militaire organisatie zijn. Zij wil ook een politiek verbond van vrije staten vormen. De NAVO wil de veiligheid van Europa en Amerika garanderen door gelijkgezinde, nieuwe democratieën op te nemen. Het lidmaatschap moet deze landen helpen bij hun ontwikkeling en handel en economische groei bevorderen, om zo de stabiliteit in Europa te vergroten. De NAVO heeft hiermee als doel het goede voorbeeld aan de Europese Unie te geven.

2.1.3 Organisatie

De structuur van de NAVO is intergouvernementeel: een bondgenootschap tussen ‘gelijkwaardige regeringen’. Het toporgaan van de NAVO is de North Atlantic Council (NAC). De Raad komt op verschillende niveaus bijeen: de ambassadeurs van de lidstaten van de NAVO, de ministers van Buitenlandse Zaken, staatshoofden en regeringsleiders. Tot 1967 was Parijs de vestigingsplaats van de NAVO. Toen Frankrijk in dat jaar zijn krijgsmacht uit de NAVO terugtrok, werd Brussel het

hoofdkwartier. De militaire leiding van de NAVO-strijdkrachten berust bij de Supreme Allied Commander Europe (SACEUR). Deze functie wordt altijd bekleed door een Amerikaan. Sinds januari 2003 is dat Generaal James L. Jones (Verenigde Staten). De politieke leiding van de NAVO berust bij de secretaris-generaal. Sinds januari 2004 is dat Jaap de Hoop Scheffer.

2.1.4 Lidstaten

De 26 leden van de NAVO zijn België, Bulgarije, Canada, Denemarken, Duitsland, Estland, Frankrijk, Giekenland, Hongarije, IJsland Italië, Letland, Litouwen, Luxemburg, Nederland, Noorwegen, Polen, Portugal, Roemenië, Slovenië, Slowakije, Spanje, de Tsjechische Republiek, Turkije, het Verenigd Koninkrijk en de Verenigde Staten.

Figuur 2.1-1 Logo van de NAVO

2.2 NATO C3 Agency (NC3A)

In deze paragraaf wordt het NATO C3 Agency (NC3A) beschreven. NATO C3 Agency staat voor NATO Constulation, Command & Control Agency en telt ongeveer 580 medewerkers, waaronder zo'n 60 militairen.

2.2.1 Geschiedenis

Het NATO C3 Agency is opgericht op 1 juli 1996 door de samensmelting van de voormalige SHAPE Technical Centre (STC) en de voormalige NATO Communications and Information System Agency (NACISA).

SHAPE Technical Centre (STC)

Het idee voor een technisch centrum om het Supreme Headquarters Allied Powers Europe (SHAPE) te ondersteunen ontstond in 1953, toen de Supreme Allied Commander Europe (SACEUR) zorgen uitte over de zwakte van 'Air Defence' binnen de NAVO. Dit resulteerde in 1955 tot het oprichten van het SHAPE Air Defence Technical Centre (SADTC). In 1963 werd het werk van het centrum uitgebreid tot het ondersteunen van alle militaire activiteiten die behoren tot het Allied Command Europe (ACE). SADTC werd toen veranderd in het Shape Tecnical Centre (STC).

NATO Communications and Information System Agency (NACISA)

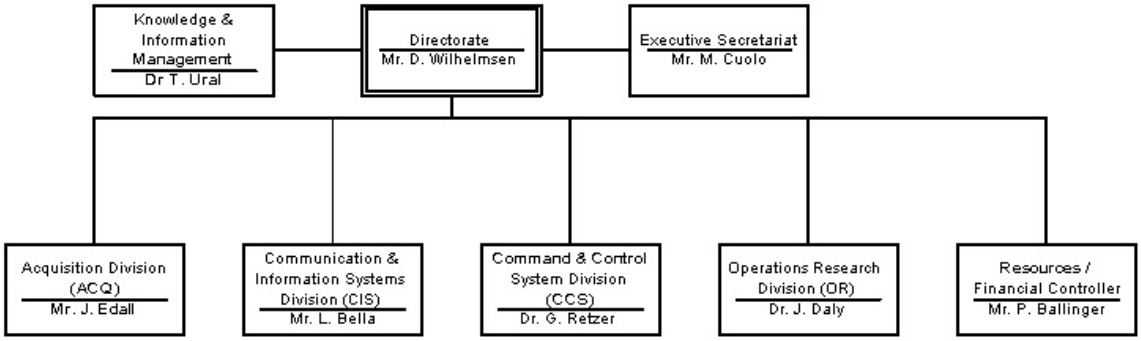
In het begin waren NAVO's strategische communicatiesystemen in ontwikkeling. Ze werden over het algemeen geleverd door nationale PTT (post, telephone, telegraph) organisaties. In de jaren 60 werd er een programma gestart, voor het leveren van een betrouwbaar NATO Integrated Communications System (NICS). In 1971 richtten de NAVO lidstaten het NICS Management Agency (NICSMA) op voor het plannen, ontwerpen en implementeren van de vereiste vermogens van het NICS. In 1986 werd de nieuwe naam het NATO Communications and Information System Agency (NACISA).

2.2.2 Doelstelling

Het NATO C3 Agency ontwikkelt, koopt aan en implementeert 'state of the art' systemen voor de NAVO en het levert onafhankelijk wetenschappelijk advies en ondersteuning aan de NAVO autoriteiten. Het NATO C3 Agency is een geïntegreerd team van ongeveer 580 professionals, bestaande uit burgers en militairen uit NAVO lidstaten die hebben besloten om de NAVO hoge kwaliteit aan expertise te bieden op het gebied van Consultation, Command en Control.

2.2.3 Organisatie

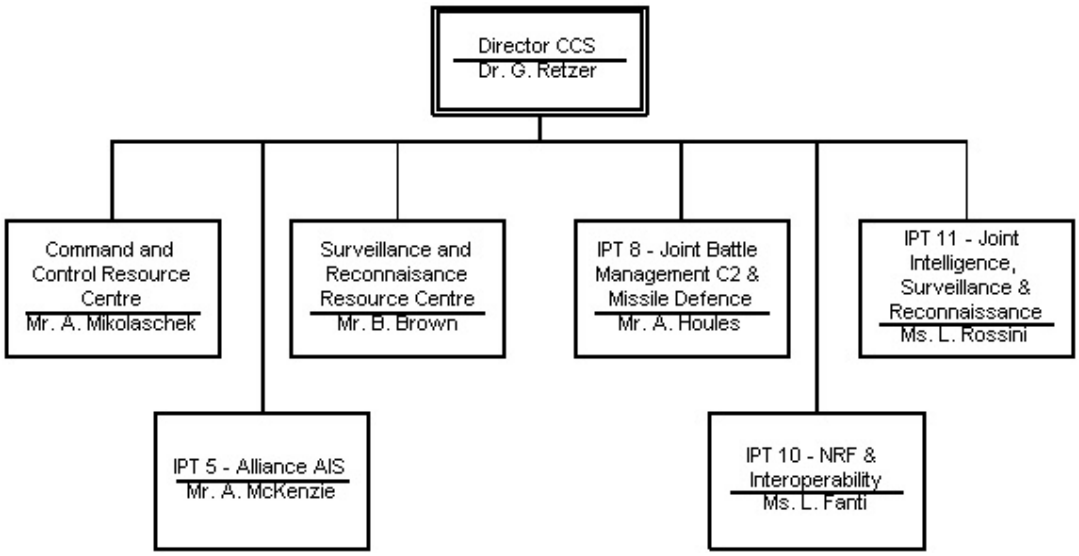
Het NATO C3 Agency is verdeeld in de vier divisies (Acquisition Division, Communication & Information Systems Division, Command & Control Systems Division en Operations Research Division) en is verdeeld over twee vestigingen (Den Haag en Brussel).



Figuur 2.2-1 Organogram NATO C3 Agency

In de bovenstaande figuur wordt een organogram weergegeven waarin de globale organisatorische structuur van het NATO C3 Agency te zien is. De Acquisition Division (ACQ) en de Resources/ Financial Controller zijn gevestigd in Brussel. De overige divisies zijn gevestigd in Den Haag.

De afstudeeropdracht is uitgevoerd voor Integrated Project Team (IPT) 8 - Joint Battle Management C2 & Missile Defence, in het Command and Control Recource Centre (C2RC). C2RC en IPT8 maken deel uit van de NC3A matrixorganisatie en vallen beiden onder de CCS Division. C2RC en IPT8 hebben betrekking op Extended Air Defence (EAD). EAD wordt in het volgende hoofdstuk beschreven. In de onderstaande figuur wordt het organogram van de CCS Division weergeven.



Figuur 2.2-2 Organogram Command & Control System Division

Hoofdstuk 3 - Achtergrondinformatie

In dit hoofdstuk wordt achtergrondinformatie, die relevant is geweest voor het uitvoeren van de afstudeeropdracht, besproken.

3.1 Extended Air Defence (EAD)

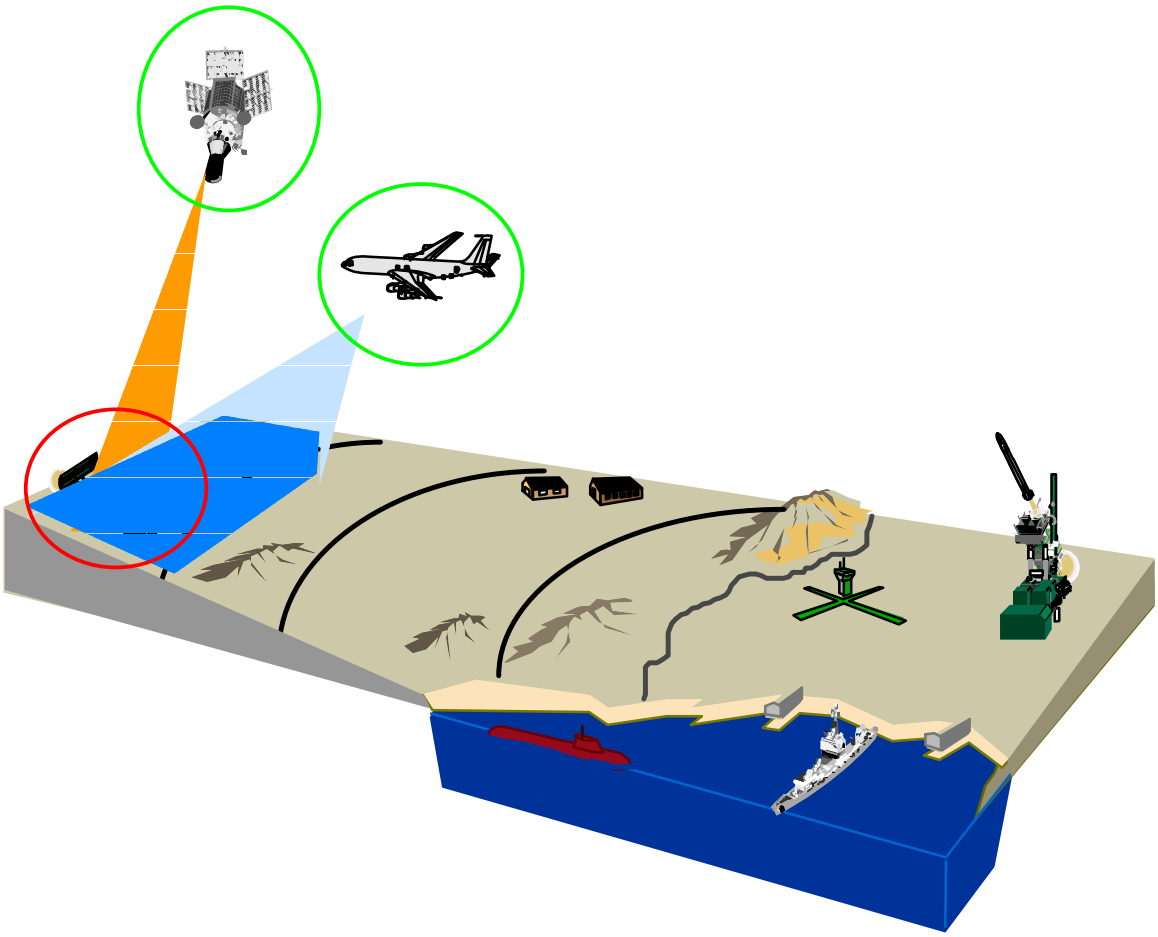
In deze paragraaf wordt het concept Extended Air Defence (EAD) uitgelegd. EAD is een breed concept voor het zo uitgebreid mogelijk verdedigen tegen luchtaanvallen. In de onderstaande figuur wordt het concept vanuit 'helikopter view' afgebeeld.

Figuur 3.1-1 Extended Air Defence

Het EAD concept onderscheidt drie soorten van verdediging tegen raketaanvallen. Dit zijn Conventional Counterforce (CCF), Active Defence (ActD) en Pasive Defence (PD). Dit concept wordt ondersteund door Battle Management, Command and Control, Communications en Intelligence (BMC3I). De drie verschillende functionele gebieden van Missile Defence worden in de volgende paragrafen besproken.

3.1.1 Conventional Counter Force (CCF)

Het doel van Conventional Counter Force (CCF) is het voorkomen van het lanceren van vijandige raketten. Dit gebeurt door middel van het uitschakelen van de vijandige lanceersystemen.



Figuur 3.1-2 Conventional Counter Force

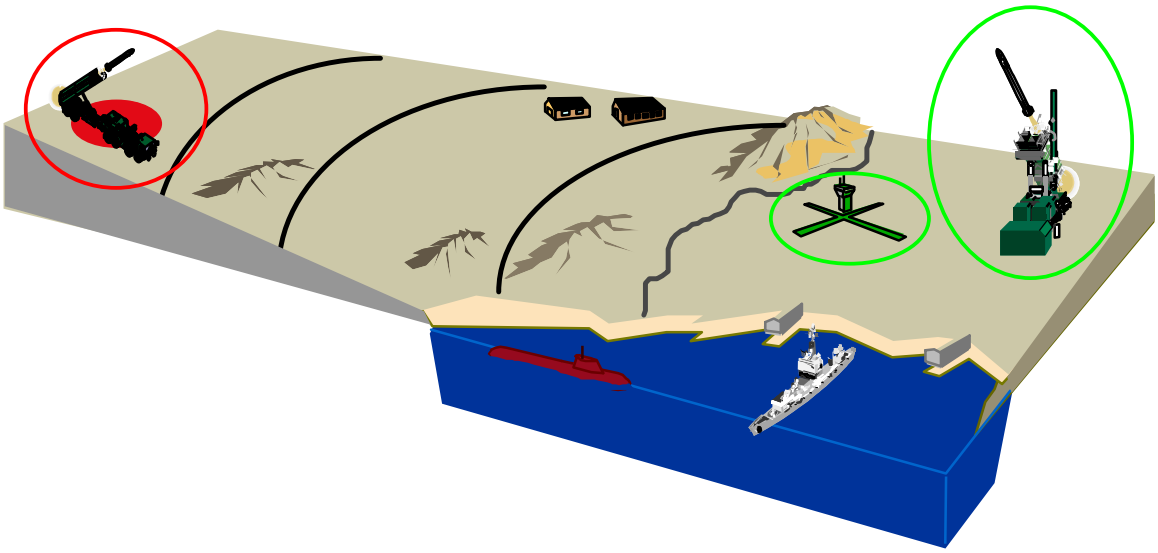
In de bovenstaande figuur is een voorbeeld van Conventional Counter Force weergegeven. De bedoeling is dat het vijandige lanceersysteem of de ondersteunende infrastructuur wordt vernietigd. Dit kan bijvoorbeeld door speciale troepen er op af te sturen. Dit kan zowel van uit de lucht als vanaf de grond gebeuren.

3.1.2 Active Defence (ActD)

Onder Active Defence (ActD) wordt het in de lucht onderscheppen van vijandelijke aanvallen verstaan. Active Defence bestaat uit het verdedigen tegen allerlei raketaanvallen zoals:

- Cruise Missile (CM)
- Tactical Ballistic Missile (TBM)
- Anti-Radiation Missile (ARM)
- Air-to-Surface Missile

Al deze raketten kunnen uitgerust worden zijn met zowel Conventionele ladingen als Weapons of Mass Destruction (WMD)



Figuur 3.1-3 Active Defence

In de bovenstaande figuur is het vijandige object door een rode cirkel omcirkeld en vriendschappelijke objecten door een groene cirkel. Stel dat de vijand een raket afvuurt (links) op een vriendschappelijk doelwit (midden) dan kan deze raket in de lucht worden onderschept door een interceptor die afgevuurd wordt door een wapensysteem (rechts).

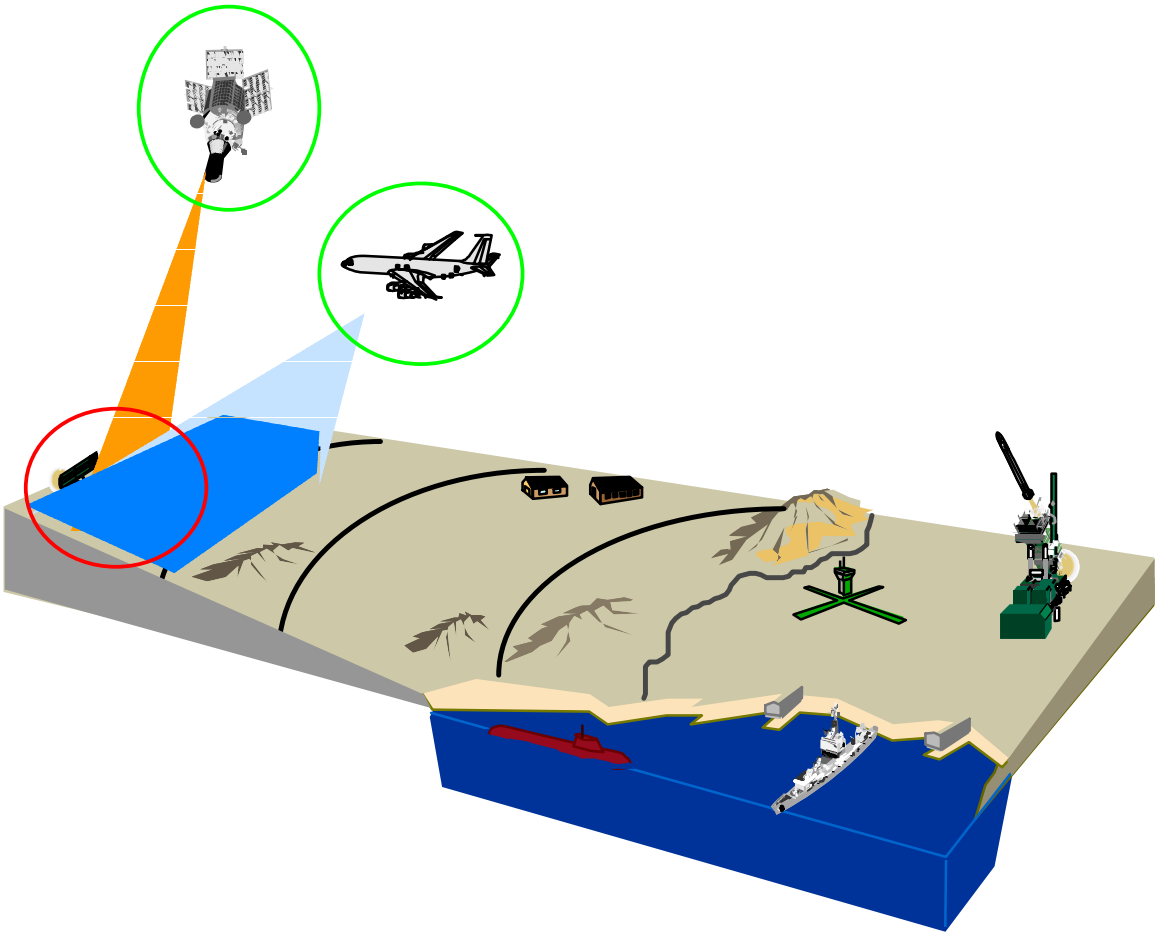
Active Defence kan worden onderverdeeld in de volgende drie categorieën

- Lower layer verdedigingsystemen. Deze onderscheppen een raket op een hoogte tot 35 km.
- Upper layer verdedigingsystemen. Deze onderscheppen een raket op een hoogte boven 35 km.
- Boost phase onderscheppingsystemen. Deze onderscheppen de raket tijdens de opstartfase.

3.1.3 Passive Defence (PD)

Passive Defence (PD) betekent het minimaliseren van de effecten van een vijandige aanval door middel van technieken als:

- Verminderen van het vermogen van de vijand om te richten door middel van camouflage, verspreiding enz.
- Reduceren van de kwetsbaarheid van assets
- Zorgen voor faciliteiten om assets te repareren na een aanval
- Het bestuderen van de effecten van nucleaire aanvallen op bepaalde gebieden



Figuur 3.1-4 Passive Defence

In de bovenstaande figuur is bijvoorbeeld te zien dat door middel van waarnemende vliegtuigen en satellieten vijandelijke objecten gelokaliseerd en geobserveerd kunnen worden. Daardoor kunnen mogelijke aanvallen beter voorspeld worden.

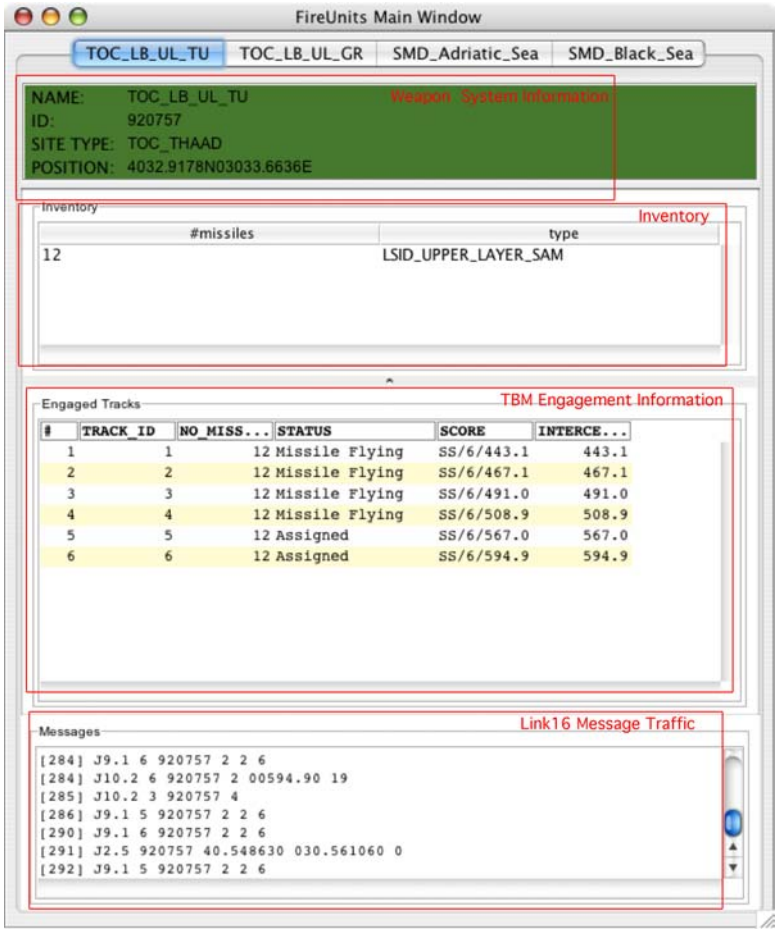
Hoofdstuk 4 - Beschrijving van de software prototype systemen

4.1 Link16 SAM C2 Interoperability Demonstrator (LSID)

De Link16 Surface to Air Missile (SAM) Command and Control (C2) Interoperability Demonstrator is een prototype dat door het NATO C3 Agency wordt ontwikkeld om requirements te verkrijgen die later zullen worden toegevoegd aan en geïntegreerd in het Air Command and Control System (ACCS). Op dit moment bestaat de functionaliteit van LSID uit twee onderdelen. Dit is het leveren van 'Situational Awareness' en het leveren van 'Engagement Coordination'. Deze functionaliteiten worden in deze paragraaf besproken.

4.1.1 Situational Awareness

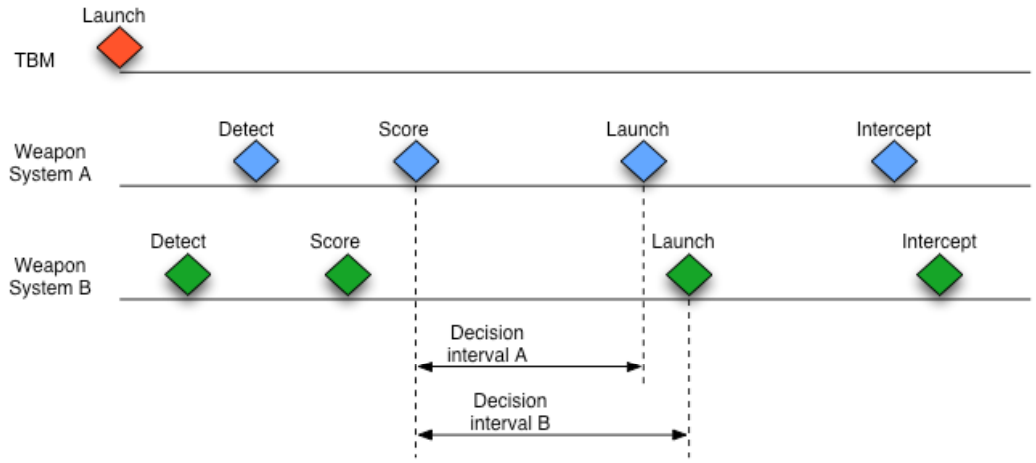
'Situational Awareness' kan worden uitgelegd als het bieden van allerlei gegevens met betrekking tot een gevechtssituatie, op een overzichtelijke en gestructureerde manier, zodat een operator goed beeld krijgt van een bepaalde situatie. Dit kan bijvoorbeeld informatie zijn over wapensystemen en coördinaten. Door middel van de grafische user interface (GUI) zorgt LSID voor 'Situational Awareness'. Deze bestaat uit drie onderdelen te weten het hoofdscherm, een map en een gedetailleerd wapensysteem informatiescherm.



Figuur 4.1-1 Wapensysteeminformatie scherm van LSID

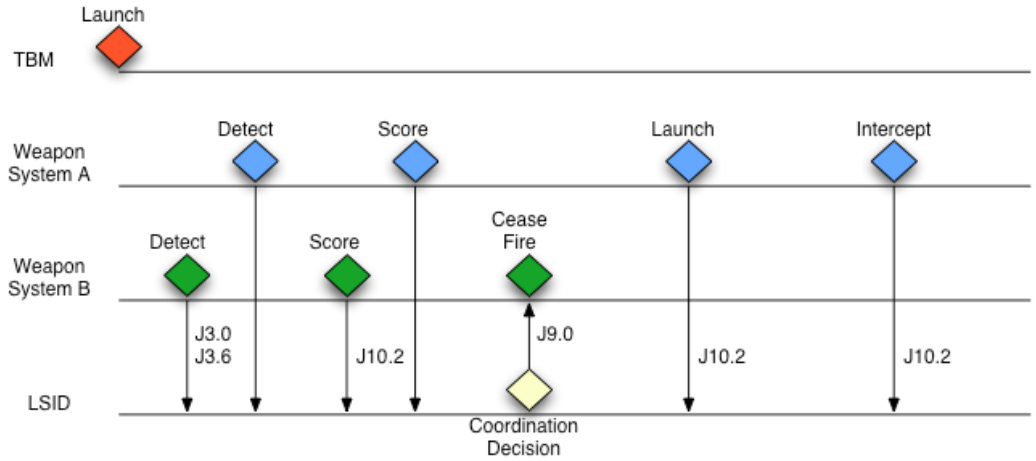
4.1.2 Engagement Coordination

Een belangrijke taak van LSID is het leveren van 'Engagement Coordination'. Als er een raketaanval plaatst vindt, kan het zijn dat er meerdere wapensystemen zijn die de raket detecteren en kunnen onderscheppen. LSID moet er voor zorgdragen dat er zo min mogelijk verspilling van raketten plaats vindt en/of dat het best geschikte wapensysteem voor de tegenaanval geselecteerd wordt. In de volgende figuur wordt een voorbeeld weergegeven waarbij er één Tactical Balistic Missile (TBM) wordt afgevuurd en er twee wapensystemen met overlappend bereik diezelfde TBM gaan onderscheppen.



Figuur 4.1-2 Dubbele onderschepping van een raketaanval

De situatie, zoals afgebeeld in de bovenstaande figuur, resulteert in het afschieten van één of meerdere interceptors door alle twee de wapensystemen. Het doel van LSID is om dit juist te voorkomen. In de onderstaande figuur wordt dezelfde situatie als de bovenstaande figuur weergegeven. Het verschil is dat er gebruik wordt gemaakt van LSID. LSID maakt een beslissing op basis van de informatie die wordt doorgegeven door de wapensystemen. Deze beslissing wordt uitgevoerd op basis van een set gedefinieerde regels.



Figuur 4.1-3 Onderschepping van een raketaanval met behulp van LSID

In dit geval geven de wapensystemen door middel van Link16 berichten door over de gedetecteerde raket. Ook wordt informatie zoals inventaris van het wapensysteem en de locatie doorgegeven. Op basis daarvan bepaald LSID welk wapensysteem wel en welk wapensysteem niet moet vuren om de raket te onderscheppen. LSID stuurt op basis daarvan een ‘Cease Fire’ bericht door aan het wapensysteem dat geen actie moet ondernemen.

4.2 Planning and Tasking Tool (PlaTo)

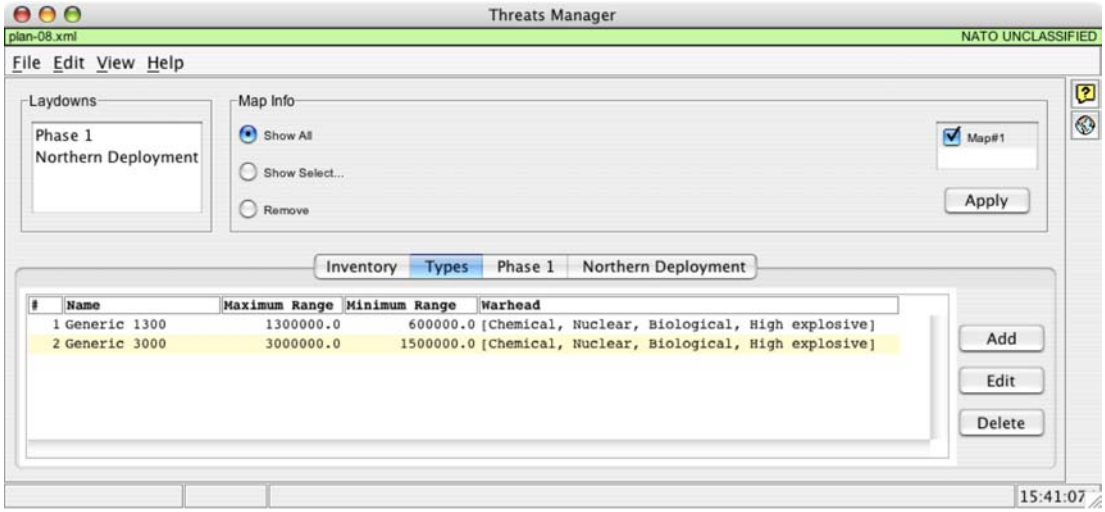
Het prototype de Planning and Tasking Tool (PlaTo) dat door het NATO C3 Agency wordt ontwikkeld, wordt gebruikt om requirements te verkrijgen die in een later stadium worden gebruikt ter uitbreiding van het Strategic Command Automated Information System (Bi-SC AIS) en het Air Command and Control System (ACCS). Het doel van de functionaliteit van PlaTo is om NAVO's vermogen, voor het uitvoeren van geïntegreerde planningen en het gebruik van gezamenlijke strijdkrachten van strategisch niveau tot aan het uitvoerbare niveau, zal moeten verbeteren. PlaTo zal planners in staat moeten stellen gezamenlijke Conventional Counterforce (CCF), Active Defence (ActD) en Passive Defence (PD) operaties tegen raketaanvallen te ondersteunen.

4.2.1 Functionaliteit van PlaTo

PlaTo heeft als doelstelling het gemakkelijker maken voor planners om informatie te verkrijgen die nodig is voor het opzetten van een planning en produceren van opdrachten. Op dit moment kan PlaTo informatie tonen over assets, resources, threats. De informatie die hiervoor nodig is moet ook ingevoerd kunnen worden. Hoewel het prototype bedoeld is om alle drie de functionele gebieden van Missile Defence (MD) te omvatten, bevat het op dit moment voornamelijk functionaliteit voor Active Defence (ActD).

4.2.2 Werking van PlaTo

PlaTo bevat verschillende managers om Assets, Resources en Threats te beheren. Met behulp van deze managers kan allerlei informatie worden ingevoerd en op een gestructureerde manier weer worden verkregen.



Figuur 4.2-1 Threat Manager van PlaTo

Met bijvoorbeeld de Threat Manager kan de inventaris van vijandige wapensystemen worden bijgehouden. Er kunnen ook nieuwe threat typen worden gecreëerd om daarmee vervolgens nieuwe threats te maken en op te slaan in de Threat Inventory.

Hoofdstuk 5 - Omschrijving van de opdracht

In dit hoofdstuk wordt de afstudeeropdracht omschreven op basis van de definitieve opdracht-omschrijving.

5.1 Kader

De opdracht zal worden uitgevoerd voor het Command and Control Resource Centre (C2RC) van het NATO Consultation, Command & Control Agency (NC3A) te Den Haag. NC3A ontwikkelt, koopt aan, implementeert zaken (voornamelijk software prototype systemen) en het levert onafhankelijk wetenschappelijk advies en ondersteuning aan de NAVO autoriteiten. NC3A is een geïntegreerd team van meer dan 580 professionals bestaande uit burgers en militairen uit de verschillende NAVO lidstaten.

NC3A houdt zich onder andere bezig met 'Concept Development Support'. Dit houdt in dat door middel van het ontwikkelen en het gebruik van software prototype systemen de requirements voor operationele systemen verkregen worden. Een extern bedrijf of contractor zal uiteindelijk de operationele systemen ontwikkelen of uitbreiden aan de hand van de door NC3A verkregen requirements.

5.2 Probleemstelling

Te midden van allerlei zaken, ontwikkelt de NC3A twee software prototype systemen, te weten:

- Link16 SAM C2 Interoperability Demonstrator (LSID) voor Theatre Missile Defence (TMD) Engagement Coordination en Situational Awareness.
- Extended Air Defence (EAD) Planning and Talking Tool (PlaTo) voor TMD planning.

De hierboven beschreven software prototype systemen, LSID en PlaTo, zijn gebruikt in verschillende NAVO oefeningen. Ook zijn de systemen geïnstalleerd op een aantal locaties. Het doel van deze installaties en oefeningen is het verkrijgen van zoveel mogelijk requirements en feedback over de software prototype systemen. De uiteindelijke requirements die door dit proces zullen worden verkregen worden in een later stadium geïntegreerd in de volgende twee Command and Control (C2) systemen:

- Bi Strategic Command Automated Information System (Bi-SC AIS)
- Air Command and Control System (ACCS)

Op dit moment kan het verkrijgen van de requirements van de eindgebruikers alleen als het personeel van NC3A tijdens een oefening aanwezig is. Tevens gebeurt dit niet op een gestructureerde manier. De requirements kunnen, door middel van bijvoorbeeld e-mail, worden verstuurd door de gebruikers van de software prototype systemen maar er is geen goede manier om opmerkingen, requirements en feedback op een logische gestandaardiseerde manier te leveren aan NC3A.

De feedback die ontvangen wordt is niet verzameld op een centrale plaats en kan onvolledige of misleidende informatie bevatten. Deze huidige situatie voldoet niet aan de wensen van NC3A.

5.3 Doestelling van de opdracht

Het doel van de opdracht is het ontwikkelen van een systeem dat er voor zorgt dat de gebruikers van de software prototype systemen opmerkingen, requirements en feedback kunnen geven door middel van een efficiënt en gestructureerd proces.

Er zal een proces moeten worden ingericht dat er voor zorgt dat het geven van opmerkingen, doorsturen van requirements en het geven van feedback over de twee software prototype systemen aan NC3A op een duidelijke en gestructureerde manier plaatst vindt. Dit proces moet worden ondersteund door een systeem dat aan de hand van het ingerichte proces zal moeten worden ontwikkeld.

5.4 Op te leveren producten en diensten

- Project Plan
- Lifecycle Objective Milestone:
    - Vision Document
    - Software Architecture Document
    - Conceptual Prototype
    - Business Case
    - Development Case
- Lifecycle Architecture Milestone:
    - refined Vision Document
    - refined Software Architecture Document
    - Executable Prototype
    - refined Business Case
    - refined Development Case
- Inial Operability Capability Milestone:
    - refined Vision Document
    - refined Software Architecture Document
    - Beta Release

Hoofdstuk 6 - Omschrijving van het Project Plan

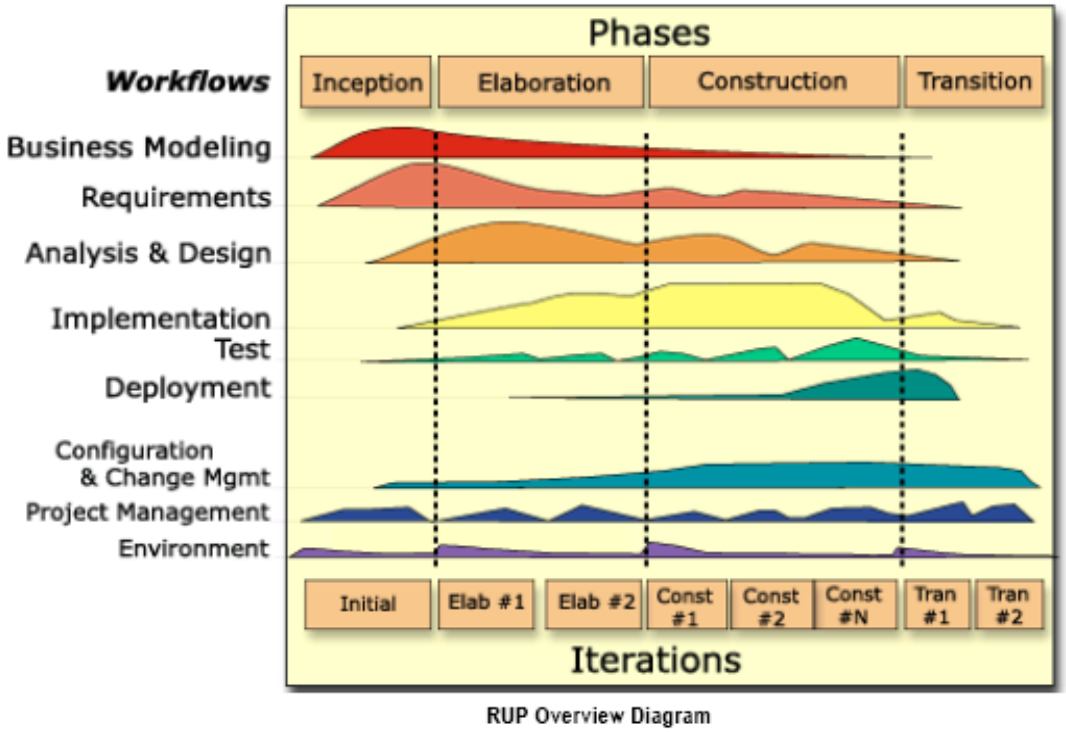
In dit hoofdstuk wordt een beschrijving van het Plan van aanpak besproken dat gebaseerd is op het Project Plan dat als externe bijlage is toegevoegd aan dit afstudeerverslag. Omdat alle rapporten voor de opdrachtgevers in het Engels moesten worden vervaardigd is er voor het Plan van aanpak niet voor de vertaling 'Plan of approach' gekozen. Dit bleek geen goede vertaling voor Plan van aanpak, waardoor gekozen is voor de benaming 'Project Plan'.

6.1 Wijze van rapporteren

In principe is er elke week een meeting geweest met de bedrijfsmentoren/opdrachtgevers en de afstudeerder. Tijdens deze meetings werd de voortgang van het afstudeerproject besproken en indien nodig werd er extra begeleiding gegeven. Ook is deze wekelijkse meeting gebruikt om de opgeleverde producten en tussen resultaten te bespreken.

6.2 Gehanteerde ontwikkelingsmethode

De activiteiten zijn uitgevoerd volgende de Rational Unified Process (RUP) systeemontwikkelmethode. RUP is een software ontwikkelingsaanpak die iteratief, 'architecture-centric' en 'use-case-driven' is. RUP kan worden verdeeld in vier fasen: Inception, Elaboration, Construction en Transition. In elke fase zullen er systeemeisen, analyse, ontwerp, implementatie en testen aanbod komen.



Figuur 6.2-1 Rational Unified Process

Elke fase bevat een of meer iteraties waarbij de nadruk ligt bij het opleveren van de producten die nodig zijn om de doelstellingen van die fase te bereiken. Elke fase heeft een mijlpaal en een set van afgebakende doelstellingen.

6.3 Gebruikte technieken

Om de taken uit te voeren zouden in eerste instantie de volgende technieken worden gebruikt:

- Interview technieken. Deze zijn gebruikt om de systeemeisen en wensen te bepalen.
- Presentatie technieken. Deze zijn gebruikt om presentaties te gegeven en prototypes te demonstreren.
- Programmeertalen zouden in eerste instantie waarschijnlijk Java en HTML bevatten. Als er een web-based applicatie zou worden ontwikkeld zou er gebruik moeten worden gemaakt van HTML. Als er een stand-alone applicatie zou worden ontwikkeld zou er gebruik worden gemaakt van Java omdat dat bij het NATO C3 Agency gebruikt wordt. Uiteindelijk is er een web-based applicatie ontwikkeld met behulp van ASP.NET.
- Unified Modeling Language (UML) ontwerptechnieken. UML werd gebruikt voor analyse en ontwerp.

6.4 Globale planning

De volgende tabel laat een overzicht zien van de oorspronkelijke planning. Deze planning is opgesteld in de eerste week van het afstudeertraject tijdens het opstellen van het Project Plan. Van deze planning is ook een visuele versie gemaakt in de vorm van een Gannt-diagram die is terug te vinden in het Project Plan. De activiteiten zijn onderverdeeld in zes fasen te weten: Start up, Inception, Elaboration, Construction, Transition en End.

Week	Calendar week	Phase	Activity
1	6 (7-2 t/m 11-2)	Start up	Write Project Plan (2 days) Familiarise with the environment and the software prototype systems LSID and PlaTo (6 days) Introduce method and techniques to be used (2 days)
2	7 (14-2 t/m 18-2)	Start up	
3	8 (21-2 t/m 25-2)	Inception	Write Vision Document (3 days) Identify and document critical Use Cases in the Software Architecture Document (2 days) Develop Prototype (3 days) Write Business Case (1 day) Write Development Case (1 day)
4	9 (28-2 t/m 4-3)	Inception	
5	10 (7-3 t/m 11-7)	Elaboration	Refine Vision Document (2 days) Design Use Cases (2 days) Implement Use Cases (2 days) Validate Use Cases (2 days) Baseline the architecture (2 days) Refine Software Architecture Document (1 day) Develop Executable Prototype (5 days) Refine Business Case (1 day) Refine Development Case (1 day)
6	11 (14-3 t/m 18-3)	Elaboration	
7	12 (21-3 t/m 25-3)	Elaboration	
8	13 (28-3 t/m 1-4)	Elaboration	
9	14 (4-4 t/m 8-4)	Construction	Refine Vision Document (2 days) Design Use Cases (5 days) Implement Use Cases (5 days) Validate Use Cases (4 days) Refine Software Architecture Document (1 day) Develop Beta Release of the system (days 5) Refine Business Case (if necessary) Refine Development Case (if necessary)
10	15 (11-4 t/m 15-4)	Construction	
11	16 (18-4 t/m 22-4)	Construction	
12	17 (25-4 t/m 29-4)	Construction	
13	18 (2-5 t/m 6-5)	Construction	
14	19 (9-5 t/m 13-5)	Transition	Refine Vision Document (if necessary) Refine Software Architecture Document (2 days) Develop Final Release of the system (5 days) Write user documentation (2 days)
15	20 (16-5 t/m 20-5)	Transition	
16	21 (23-5 t/m 27-5)	End	Write final internship end report (15 days)
17	22 (30-5 t/m 3-6)	End	
18	23 (6-6 t/m 10-6)	End	

Figuur 6.4-1 Oorspronkelijke globale planning uit het Project Plan

In elke fase van RUP is in de Development Case een aangepaste planning opgesteld. In de aangepaste planning die is opgesteld aan het eind van de Inception fase was de planning al behoorlijk gewijzigd. Allereerst is in overleg met de opdrachtgevers besloten dat de Transition fase niet binnen het afstudeertraject zal worden gedaan. Het werd tijdens deze fase duidelijk dat het niet haalbaar was om binnen 15 weken het gehele traject te doorlopen. Ook zijn er in de aangepaste planning iteraties ingepland hetgeen niet het geval was bij de oorspronkelijke planning. Deze aangepaste planning wordt hieronder weergegeven.

Week	Calendar week	Phase	Activity	
6	11 (14-3 t/m 18-3)	Elaboration (iteration 1)	Refine Vision Document (1 day) Design critical use cases (2 days) Design and implement database(2 days)	
7	12 (21-3 t/m 25-3)	Elaboration (iteration 1)	Built an Executable Prototype (6 days) Test the Executable Prototype (2 days) Refine Business Case (1 day) Refine Development Case (1 day)	
8	13 (28-3 t/m 1-4)	Elaboration (iteration 1)		
9	14 (4-4 t/m 8-4)	Elaboration (iteration 2)	Refine Vision Document (1 day) Redesign critical use cases (2 days) Redesign and implement database (2 days)	
10	15 (11-4 t/m 15-4)	Elaboration (iteration 2)	Rebuilt an Executable Prototype (5 days)	
11	16 (18-4 t/m 22-4)	Elaboration (iteration 2)	Test the Executable Prototype (1 days) Refine Business Case (1 day) Refine Development Case (1 day) Write Lifecycle Architecture Milestone report (2 days)	
12	17 (25-4 t/m 29-4)	Construction (iteration 1)	Refine Vision Document (1 days) Design use cases (4 days)	
13	18 (2-5 t/m 6-5)	Construction (iteration 1)	Build Beta Release (6 days) Test Beta Release (2 days) Refine Business Case (1 day) Refine Development Case (1 day)	
14	19 (9-5 t/m 13-5)	Construction (iteration 1)		
15	20 (16-5 t/m 20-5)	Construction (iteration 2)	Refine Vision Document (1 days) Redesign use cases (4 days)	
16	21 (23-5 t/m 27-5)	Construction (iteration 2) / End	Rebuild Beta Release (5 days)	Write final internship report (15 days)
17	22 (30-5 t/m 3-6)	Construction (iteration 2) / End	Test Beta Release (1 day) Refine Business Case (1 day) Refine Development Case (1 day) Write Initial Operational Capability Milestone report (2 days)	
18	23 (6-6 t/m 10-6)	End		

Figuur 6.4-2 Aangepaste globale planning uit de Inception fase

Uiteindelijk is de bovenstaande planning redelijk gehandhaafd. In deze planning is te zien dat er twee iteraties zijn ingepland in de Construction fase. Uiteindelijk is gebleken dat er nog een derde iteratie nodig is om de doelstellingen van deze fase te bereiken. Deze derde iteratie heeft niet binnen het afstudeertraject plaatsgevonden.

6.5 Beschrijving mijlpaalproducten

Mijlpaalproducten die genoemd zijn in het gedeelte 'Op te leveren producten en diensten' uit de opdrachtomschrijving worden hier kort omschreven.

- Project Plan. Het Project Plan gaf een beschrijving van de manier waarop de opdracht zou worden uitgevoerd.

- Lifycycle Objective Milestone. Het doel van deze mijlpaal was het begrijpen van wat er ontwikkeld moest worden en het identificeren van de kernfunctionaliteit van het systeem. Een ander doel was om inzicht te krijgen in de kosten, de planning en de risico's bij het project en het te volgen proces. Deze mijlpaal bevatte een Conceptueel Prototype.

- Lifecycle Architecture Milestone. Het doel van de architecture milestone was het verkrijgen van een gedetailleerd beeld met betrekking tot de systeemeisen en systeemfunctionaliteit. Een ander doel was het evalueren van een uitvoerbare architectuur. Deze mijlpaal bevatte een Executable Prototype met geïmplementeerde functionaliteit.

- Initial Operational Capability Milestone. Deze mijlpaal zou een Beta Release van het systeem bevatten.

- Product Release Milestone. Deze laatste mijlpaal zou de Final Release van het systeem bevatten.

Omdat in overleg met de opdrachtgevers de Transition fase is komen te vervallen, zal de Product Release Milestone niet worden opgeleverd. De Construction fase is niet geheel afgerond waardoor er een Beta Release is opgeleverd die nog niet definitief is. Het systeem zou in principe kunnen worden voltooid binnen het NATO C3 Agency. Aanpassingen die gewenst zijn aan de opgeleverde Beta Release en de ontbrekende functionaliteit ervan zijn beschreven in de Initial Operational Capability Milestone dat als externe bijlage aan dit afstudeerverslag toegevoegd.

Deel II - Werkzaamheden

Hoofdstuk 7 - Start up fase

In dit hoofdstuk zullen de werkzaamheden die zijn uitgevoerd tijdens de Start up fase worden besproken. Deze fase is aan het afstudeertraject toegevoegd vanwege activiteiten die niet expliciet in het Rational Unified Process (RUP) worden omschreven maar toch vereist waren voor het uitvoeren van de afstudeeropdracht. Het betreft het opstellen van een Project Plan (Plan van aanpak), bekend raken met de omgeving (de organisatie en de software prototype systemen) en het introduceren van de methode en technieken die zouden gaan gebruiken voor het uitvoeren van de opdracht.

7.1 Opstellen van het Project Plan

Het opstellen van het Project Plan is één van de eerste activiteiten die is uitgevoerd. Op basis van de voorlopige opdrachtomschrijving is er begonnen met het opstellen van het Project Plan. In verschillende gesprekken met de opdrachtgevers is het Project Plan besproken en vervolgens weer aangepast. Dit Project Plan is beschreven in Deel I, Bedrijf en Opdracht, en is tevens als externe bijlage toegevoegd aan dit afstudeerverslag.

7.2 Bekend raken met de omgeving

De eerste weken zijn besteedt aan het bekend raken met de organisatie en de software prototype systemen LSID en PlaTo, die worden ontwikkeld binnen het Command and Control Resource Centre (C2RC). Dit is gedaan door middel van het lezen van rapporten, het bekijken van courses en het houden van interviews met verschillende medewerkers van het NATO C3 Agency. Er is besloten hier een verslag van te maken zodat de bedrijfsmentoren dat konden lezen en aan de hand van dit verslag konden bepalen of de afstudeerder inderdaad een goed beeld heeft gekregen van de organisatie en de sofware prototypes systemen LISD en PlaTo. Dit verslag werdt het ‘Familiarization Report’ genoemd. Uiteindelijk is dit verslag met de opdrachtgevers besproken en is er het een en ander aangevuld en aangepast. Gedeeltes uit dit verslag zijn ook voor dit afstudeerverslag gebruikt.

7.3 Introduceren van te gebruiken methode en technieken

Het leek belangrijk dat de opdrachtgevers/bedrijfsmentoren een goed inzicht kregen in de manier waarop de afstudeeropdracht zou worden uitgevoerd. Dit zou eventuele onduidelijkheden of onbegrip van zowel de opdrachtgevers/bedrijfsmentoren als de afstudeerder zelf kunnen voorkomen. Dit is gedaan door een presentatie te geven waarbij het een en ander is toegelicht over het afstuderen aan de Haagse Hogeschool, zodat er een goed beeld ontstaat wat de procedure is en de verwachtingen van het afstudeertraject zijn aan de Haagse Hogeschool. Tijdens deze presentatie zijn ook de methode en technieken die gebruikt gingen worden toegelicht en besproken. Op deze manier konden er naast de te gebruiken methode en technieken die in het Project Plan beknopt zijn uitgelegd, de opdrachtgevers/bedrijfsmentoren en andere betrokken personen enig inzicht krijgen in de methode en technieken die gebruikt zouden gaan worden bij het uitvoeren van de afstudeeropdracht.

Hoofdstuk 8 - Inception fase

In dit hoofdstuk worden de uitgevoerde werkzaamheden in, en de opgeleverde producten uit de Inception fase besproken. De Inception fase is de eerste fase die door het Rational Unified Process (RUP) voorgeschreven wordt. In deze fase zijn het Vision Document, het Software Architecture Document, een Conceptual Prototype, de Business Case en de Developement Case opgeleverd. Uiteindelijk is dit alles verwerkt in een rapport de 'Lifecycle Objective Milestone'. Als externe bijlage is dit rapport toegevoegd aan dit afstudeerverslag. Dit rapport is net als het Conceptual Prototype een mijlpaal uit de Inception fase.

8.1 Opstellen van het Vision Document

Zoals door RUP is voorgeschreven is in de Inception fase begonnen met het opstellen van het Vision Document. Volgens RUP moet in dit document de opdracht en de toekomstige gebruikers worden omschreven en er moet een globaal beeld worden geschetst van de functionaliteit van het te ontwikkelen systeem. Dit document zou aan het einde van de Inception fase stabiel moeten zijn, het was in latere fasen echter wel mogelijk dit document aan te passen wanneer nodig.

8.1.1 Opstellen van het Statement of Work

Het Statement of Work (opdrachtomschrijving) komt in principe overeen met het Statement of Work uit het Project Plan. Om hetzelfde Statement of Work vast te leggen was daardoor overbodig. Toch is dit gedaan om de volgende reden. Het Statement of Work dat in het Project Plan is beschreven is daarna niet meer aangepast. Op deze manier was het voor de opdrachtgevers en de afstudeerder gemakkelijk om altijd terug te kijken naar de oorspronkelijke opdracht. Elke wijziging en aanvulling die plaatst vindt met betrekking tot het Statement of Work heeft hij aangepast en aangevuld in het Vision Document. Aangezien het Vision Doucment gedurende heel het project aangepast kan worden leek het ook daarvoor heel goed geschikt te zijn.

8.1.2 Beschrijven van de target users

In het Vision Document heeft zijn ook de verschillende gebruikers groepen beschreven. Dit wordt in de volgende figuur weergegeven.

2.2.1 Developer user group

The developer user group is part of C2RC, which belongs to the Command and Control System Division (CCSD) at NC3A.

At his moment, the developer user group exists of four staff member and three contractors. Two staff members are working on LSID and three contractors are working on PlaTo. The Senior Scientist is in charge of the development of LSID and PlaTo. LSID and PlaTo are part of the Project TM, which is lead by the Principal Scientist.

2.2.1 End-user group

At this moment the users of PlaTo are limited to the Head Quarters Extended Air Defence Task Force (HQ EADTF) and the Head Quarters Allied Rapid Reaction Corps (HQ ARRC). It is possible that PlaTo is going to be used at the Regional Command (RC) South in Naples (Italia) and the Air Component Command (ACC) South in Izmir (Turkey). There are no users for LSID at this moment.

Figuur 8.1-1 Beschrijving van de target users uit het Vision Document

Het beschrijven van de ‘Developer user group’ was vrij eenvoudig omdat dit de mensen zijn die betrokken zijn bij het ontwikkelen van de software prototype systemen binnen het NATO C3 Agency. Deze informatie was voor dus gemakkelijk te verkrijgen. Het schrijven van de ‘End-user group’ was vrij lastig. Dit zijn de mensen die gebruik maken van de software prototype systemen die door het NATO C3 Agency zijn en worden ontwikkeld. Daarvoor is geen vaste gebruikersgroep en deze kan erg variëren. Dit zijn mensen van buiten het NATO C3 Agency en vaak werkzaam buiten Nederland. Er is geprobeerd om zoveel mogelijk informatie over deze gebruikersgroep te verkrijgen en dat te beschrijven.

8.1.3 Opstellen van de Features

Volgens RUP wordt in het Vison Document een globaal beeld geschetst van de functionaliteit van het te ontwikkelen systeem. Dit kan door middel van geprioriteerde features of zogenaamde key use cases. Deze beschrijven op een zo globaal mogelijk niveau de functionaliteit van het systeem. Er is gekozen om geprioriteerde features op te nemen in het Vision Document omdat op die manier in een vroeg stadium kan worden vast gelegd welke functionaliteit belangrijk is en welke minder belangrijk. Het prioriteren van de features is gedaan op basis van het belang van de feature voor de opdrachtgevers en de geschatte complexiteit. Deze features zijn opgesteld aan de hand van interviews met de opdrachtgevers en andere gebruikers van het te ontwikkelen systeem. Een voorbeeld van een feature wordt weergegeven in de onstaande figuur.

Feature No.5	
Name	Edit status feedback.
Description	Allow NC3A developers to edit the status of feedback.
Value	2.
Complexity	Medium.
Priority	2.

Figure 2.3-5 Feature No.5 Edit status feedback

Figuur 8.1-2 Voorbeeld van een feature uit het Vision Document

8.1.4 Opstellen van de Key use cases

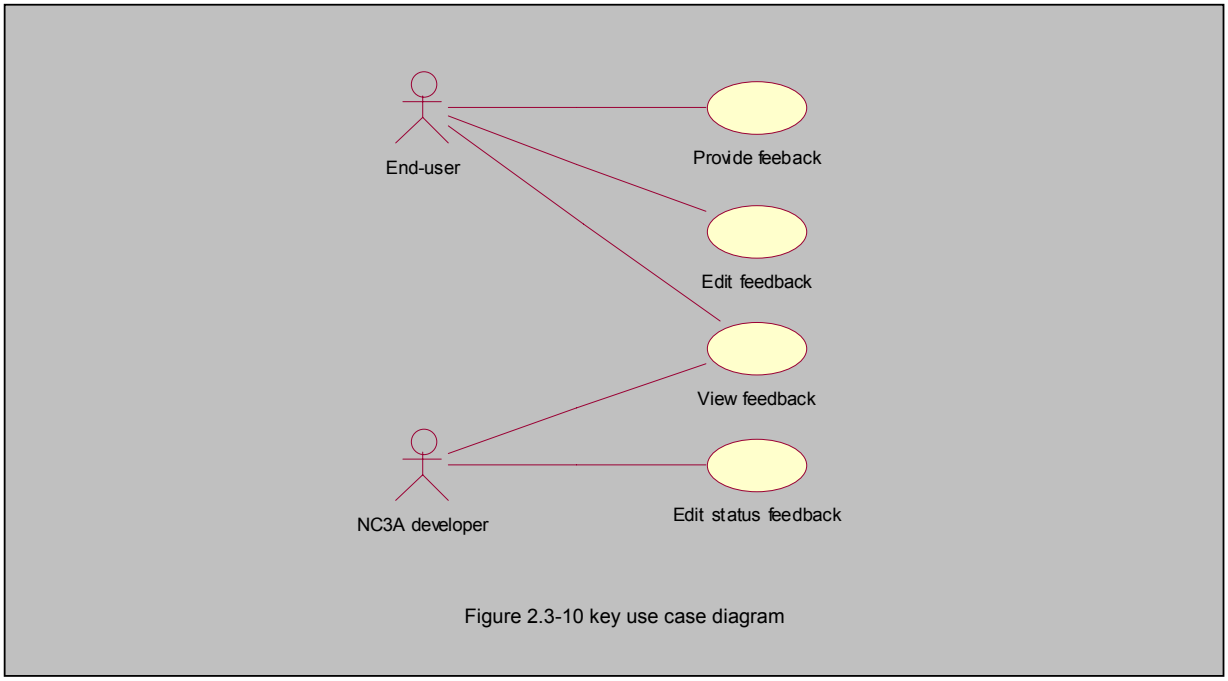
Volgens RUP worden systeemeisen vervangen door use cases. In het Vision Document kunnen key use cases worden opgesteld om een globaal beeld te krijgen van de systeemeisen die het systeem moet gaan bevatten. In principe zijn deze te vergelijken met features alleen zijn deze in principe niet geprioriteerd. In de onderstaande figuur wordt een voorbeeld van een van de key use cases afgebeeld. Deze key uses cases zijn opgesteld op basis van de features.

Key use case No. 4	
Name	Edit status feedback.
Assumption	End-user has logged into the system and has selected feedback.
Actors	NC3A developer.
Description	(1) NC3A developer indicates that he want to edit the status of the selected feedback. (2) The system asks which status he wants to add to the feedback and (3) NC3A developer answers this question. (4) The system confirms the new status of the feedback.
Exceptions.	None.
Result	Status of feedback is changed.

Figure Key 2.3-9 use case No.4 Edit status feedback

Figuur 8.1-3 Voorbeeld van een key use cases uit het Vision Document

Er is gekozen om ook de key use cases op te stellen omdat er aan de hand van deze key use cases en de vastgestelde gebruikersgroepen een use case diagram kon worden opgesteld die inzicht hebben gegeven in de interactie tussen de key use cases en de verschillende gebruikersgroepen. Dit use case diagram wordt hieronder weergegeven.



Figuur 8.1-4 Het key use case diagram uit het Vision Document

8.1.5 Opstellen van de Non-functional requirements

Als laatste zijn ook de niet-functionele systeemeisen opgesteld. Het helder krijgen van de niet-functionele systeemeisen was vrij moeilijk aangezien de opdrachtgevers deze in eerste instantie niet echt hadden. De afstudeerder was namelijk vrij om een database omgeving te kiezen en voor een stand-alone applicatie hadden de opdrachtgevers het liefst een systeem geprogrammeerd in Java. Mocht het systeem een web-based applicatie worden dan was hij daar qua programmeertaal/scripttaal vrij in.

8.2 Opstellen van het Software Architecture Document

In RUP is niet duidelijk beschreven waar en op welke manier het ontwerp gedocumenteerd moet worden. RUP laat de uitvoerder hier namelijk erg vrij in. Er is heeft gekozen voor wat in RUP het Software Architecture Document wordt genoemd. Hierin heeft hij alle ontwerpdocumentatie vastgelegd. De totstandkoming van de ontwerpen die dit document in de Inception fase bevat worden in dit hoofdstuk besproken.

8.2.1 Opstellen van de 'Mile-wide, inch-deep' beschrijving

Door middel van een brainstormsessie zijn er zoveel mogelijk actoren en use cases bedacht en deze zijn heel globaal te beschrijven. Dit wordt in RUP de 'Mile-wide, inch-deep' beschrijving genoemd. Het zou heel goed kunnen dat er later nog meer use cases bij komen, of dat bepaalde use cases worden samengevoegd. Delen uit de 'Mile-wide, inch-deep' omschrijving wordt in de volgende figuren afgebeeld.

Actor No.1	
Name	End-user.
Description	An End-user is a person who is using and testing the software prototype systems developed by NC3A.

Figure 3.1-1 Actor No.1 End-user

Actor No.2	
Name	NC3A developer.
Description	A NC3A developer is a person who works on a software prototype system for NC3A.

Figure 3.1-2 Actor No.2 NC3A developer

Actor No.3	
Name	NC3A administrator
Description	A NC3A administrator is a person who works on software prototype systems NC3A and will be the administrator of the Requirement Capture System that has to be developed.

Figure 3.1-3 Actor No.3 NC3A administrator

Figuur 8.2-1 Actoren uit de 'Mile-wide, inch-deep' beschrijving

Zoals in het Vision Document al is beschreven zijn er grofweg gezien twee gebruikersgroepen. Namelijk de gebruikers van de sofware prototype systemen die ontwikkeld zijn binnen het NATO C3 Agency en de systeemontwikkelaars binnen het NATO C3 Agency. Als derde actor is er een administrator aan toegevoegd. Deze actor zal gebruik gaan maken van de voor de administrator specifieke use cases zoals het toevoegen van nieuwe gebruikers.

Use case No. 24	
Name	Provide feedback.
Assumption	Use has signed in and has selected a software prototype system.
Actors	End-user.
Description	(1) The user tells the system that he/she wants to add feedback. (2) The system asks him/her what the contents of the feedback must be and (3) the user answers that question.
Exceptions	None.
Result	Feedback is provided.

Figure 8.2-2 Use case No.24 Provide feedback

Use case No. 25	
Name	Add user note.
Assumption	User has signed in and has selected a software prototype system and selected feedback.
Actors	End-user.
Description	(1) The user tells the system that he/she wants to add a user note. (2) The system asks him/her what the contents of the user note must be and (3) the user answers that question.
Exceptions	None.
Result	User note is added.

Figure 8.2-2 Use case No.25 Add user note

Figuur 8.2-2 Use cases uit de ‘Mile-wide, inch-deep’ beschrijving

In de bovenstaande figuur worden twee use cases uit de ‘mile-wide, inch-deep’ beschrijving getoond. Dit zijn twee use cases waar alleen de actor ‘End-user’ gebruik van maakt. Op basis van de kritieke use cases is er een beeld verkregen van de functionaliteit die het te ontwikkelen systeem zou moeten gaan bevatten. De systeemeisen worden volgens RUP vervangen door de beschrijvingen van de use cases. Uiteindelijk is er op basis van de gedefinieerde use cases en de betrokken actoren een use case diagram opgesteld. Dit use case diagram gaf enig inzicht in de relaties tussen de verschillende actoren en hun use cases.

8.2.2 Bepalen en uitwerken van de kritieke use cases

Uit deze verzameling use cases zijn ongeveer 20 procent als kritiek beschouwd die verder uitgewerkt zijn omdat dit door RUP wordt voorgeschreven. De kritieke use cases zijn samen met de opdrachtgevers gekozen op basis van welke use cases behoren tot de kernfunctionaliteit van het systeem en welke use cases volgens de opdrachtgevers geïmplementeerd moesten worden.

Critical use case	**No. 1**
Name	Sign in.
Assumption	The system is started up.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system asks the user his/her username and password. (2) The user enters his/her username and password and tells the system he/she wants to sign in. (3) The system signs in the user en shows the available software prototype systems.
Exceptions	[Username unknown.] The system tells the user that the username is unknown. [Username and password combination not valid.] The system tells the user that the username and password combination is not valid.
Result	User has signed in.

Figure 3.2-1 Critical use case No.1 Sign in

Critical use case	**No. 2**
Name	Provide feedback.
Assumption	Use has signed in as End-user and has selected a software prototype system and a module.
Actors	End-user.
Description	(1) The user tells the system that he/she wants to add feedback. (2) The system asks him/her the feedback title, feedback description, module where the feedback is about, version of the software prototype system and the Operational System that has been used. (3) The user answers that question and tells the system that he wants to submit the feedback. (4) The system tells the user that the feedback has been submitted and shows the user the feedback he/she just provided along with the other feedback entries previously been made.
Exceptions	None.
Result	Feedback is provided.

Figure 3.2-2 Critical use case No.2 Provide feedback

Figuur 8.2-3 Voorbeelden van twee kritieke use cases

De use cases die als kritiek zijn beschouwd zijn verder uitgewerkt. De verzameling vormde de basis voor het Conceptual Prototype wat in de Inception fase ontwikkeld is.

8.3 Implementatiemogelijkheden bepalen

Er waren verschillende mogelijkheden om het systeem te implementeren. Het was in dit stadium onduidelijk wat precies de mogelijkheden waren om het systeem binnen de organisatie te implementeren. Het leek belangrijk om dit toch in een vroeg stadium helder te krijgen omdat de manier van implementeren invloed zou kunnen hebben op het gehele ontwikkeltraject. Er is uitgezocht welke mogelijkheden aanwezig zijn voor het implementeren van het systeem. In deze paragraaf worden de verschillende mogelijkheden en de gekozen mogelijkheden besproken

8.3.1 Stand-alone of Web-based

Eén van de belangrijkste keuze is of het systeem een web-based of een stand-alond applicatie zal moeten worden. Het voordeel van een stand-alone applicatie is dat het mogelijk is om de gebruikersinterface precies te ontwikkelen volgens de interface eisen van de gebruikers. Het voordeel van een web-based applicatie is dat de gebruikers niets hoeven te installeren omdat ze in dat geval toegang tot het systeem hebben via de Internet browser. De opdrachtgevers hebben aangegeven dat hun voorkeur ligt bij het ontwikkelen van een web-based systeem vanwege de voordelen die dit met zich meebrengt.

8.3.2 NATO SECRET of NATO UNCLASSIFIED

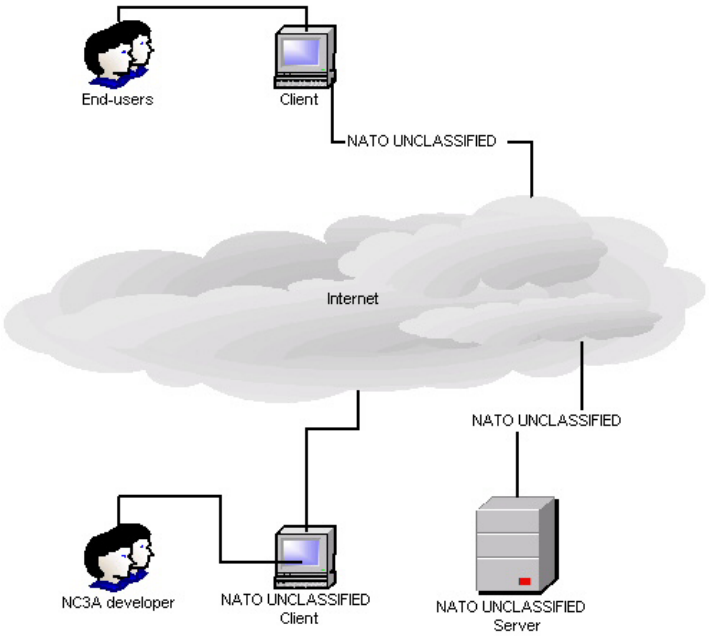
Het systeem zou in principe gehost kunnen worden op het NATO SECRET netwerk of het NATO UNCLASSIFIED netwerk. Wat precies de mogelijkheden en beperkingen waren was op dat moment niet helder. Er is binnen het NATO C3 Agency uitgezocht wat precies de mogelijkheden zijn en wat de daarbij behorende beperkingen voor het hosten van een web-based systeem zijn.

De ‘NC3A Developers’ werken over het algemeen in het Theatre Missile Defence (TMD) lab en in een aantal Class 2 Security Area kamers in het NATO C3 Agency gebouw in Den Haag. Het TMD lab is een Class 1 Security Area. Het netwerk in het TMD lab wordt beschouwd als NATO SECRET maar is niet en zal niet worden verbonden met het NATO SECRET netwerk dat het Crisis Response Operational NATO Open Systems (CRONOS) wordt genoemd. In het TMD lab is een CRONOS terminal verbonden met het CRONOS netwerk en een NATO RESTRICTED computer om de medewerkers informatie op het Internet te laten opzoeken. De Class 2 Security Area kamers hebben één of meer NATO RESTRICTED computers waarmee ze informatie kunnen opzoeken op het Internet.

De ‘End-users’ kunnen zich op verschillende locaties van de NAVO lidstaten bevinden. De ‘End-users’ zouden toegang moeten hebben tot Internet en dus ook tot het NATO UNCLASSIFIED netwerk, bijvoorbeeld de website van het NATO C3 Agency (http://www.nc3a.nato.int). Het is ook mogelijk dat ze toegang hebben tot het NATO SECRET netwerk CRONOS.

NATO UNCLASSIFIED

Het zou mogelijk zijn om het systeem te hosten op een NATO UNCLASSIFIED computer die verder helemaal niet verbonden is met een ander netwerk van de NAVO of het NATO C3 Agency. Dit wordt in de onderstaande figuur globaal weergegeven.



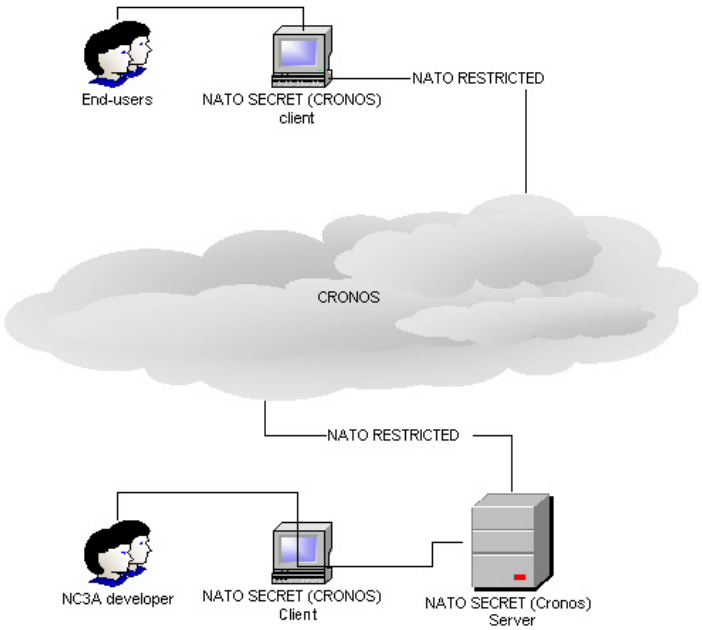
Figuur 8.3-1 NATO UNCLASSIFIED

Het voordeel van deze oplossing is dat alle gebruikers die toegang hebben tot het Internet, ook toegang zouden moeten hebben tot het systeem.

Het nadeel is dat alle gegevens die uitgewisseld worden NATO UNCLASSIFIED moeten zijn. Een ander nadeel kan zijn dat de ‘End-users’ op de testlocatie zelf geen directe toegang hebben tot het Internet omdat de testomgeving waarschijnlijk NATO SECRET zal zijn.

NATO SECRET

De andere mogelijkheid was het hosten van een systeem op het NATO SECRET netwerk CRONOS. Dit is in de onderstaande figuur globaal weergegeven.



Figuur 8.3-2 NATO SECRET

Het voordeel van deze oplossing is dat het mogelijk wordt om CLASSIFIED tot en met NATO SECRET gegevens uit te wisselen. Er hoeft dan ook geen nieuw werkstation te komen omdat er in het TMD lab al een CRONOS terminal beschikbaar is. In verschillende Class 2 Security Area zijn ook CRONOS terminals aanwezig.

Het nadeel is dat de procedure om hier toestemming voor te verkrijgen veel tijd in beslag zal nemen. Het is ook niet zeker dat alle gebruikers toegang tot het NATO SECRET netwerk CRONOS hebben.

In de onderstaande figuur wordt schematisch een vergelijking tussen de twee mogelijkheden weergegeven.

	NATO SECRET	NATO UNCLASSIFIED
Informatie die uitgewisseld kan worden	NATO UNCLASSIFIED NATO RESTRICTED NATO CONFIDENTIAL NATO SECRET	NATO UNCLASSIFIED
Benodigd om toegang tot het systeem te verkrijgen	Toegang tot het NATO SECRET netwerk CRONOS	Toegang tot het Internet
Procedure om toestemming te krijgen	Zal veel tijd inbeslag	Zal minder tijd inbeslag nemen

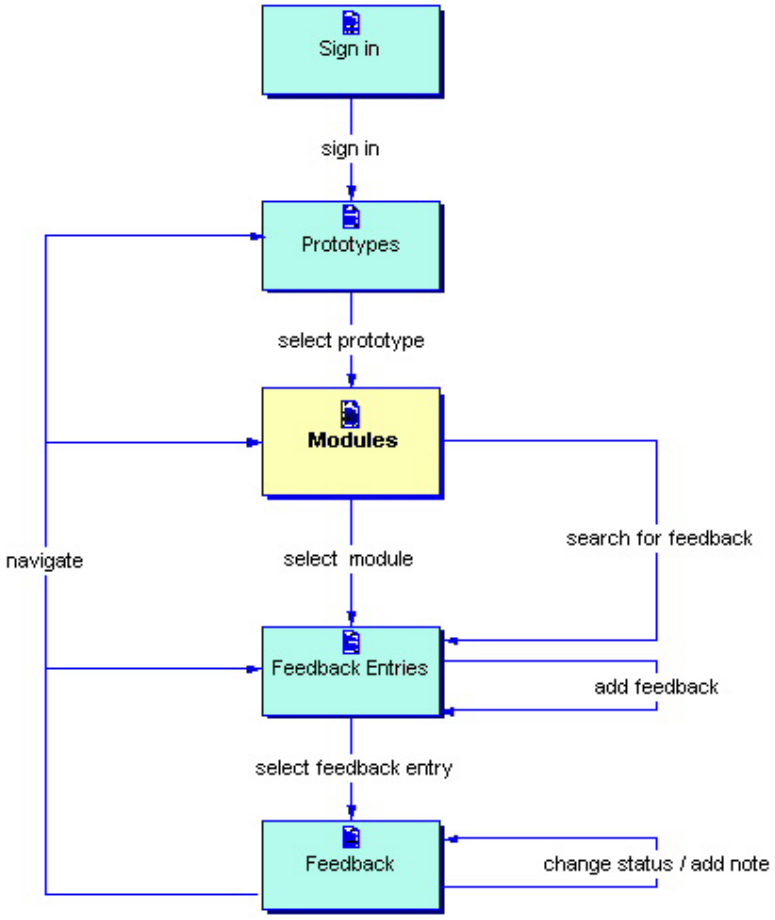
Figuur 8.3-3 Vergelijking tussen NATO SECRET en NATO UNCLASSIFIED

Op basis van de voor en nadelen is er besloten om in ieder geval te beginnen met het ontwikkelen van het systeem, door het systeem op een NATO UNCLASSIFIED server te hosten. De opdrachtgevers hebben aangegeven dat de informatie die zal worden uitgewisseld door middel van het te ontwikkelen systeem waarschijnlijk wel tot NATO UNCLASSIFIED te benoemen is. Het zal waarschijnlijk ook niet mogelijk zijn om binnen de afstudeerperiode toestemming te krijgen voor het hosten van een systeem op het NATO SECRET netwerk CRONOS.

8.4 Ontwikkelen van het Conceptual Prototype

8.4.1 Opstellen van het Navigatiediagram

Om een globaal beeld te krijgen van de navigatie door het prototype is er een navigatiediagram opgesteld. Dit navigatiediagram is in eerste instantie opgesteld op basis van het beeld wat er op dat moment van het te ontwikkelen systeem was. Op basis van een gesprek met de opdrachtgevers is dit navigatiediagram weer aangepast. In de onderstaande figuur wordt dit navigatiediagram afgebeeld.



Figuur 8.4-1 Navigatiediagram

Het was duidelijk dat dit diagram een grove schets was en uiteindelijk weer aangepast zal worden. Het schema gaf echter wel een beeld van de inhoud van het systeem dat ontwikkeld gaat worden en heeft geholpen bij het ontwikkelen van het Conceptual Prototype.

8.4.2 Eerste Conceptual Prototype

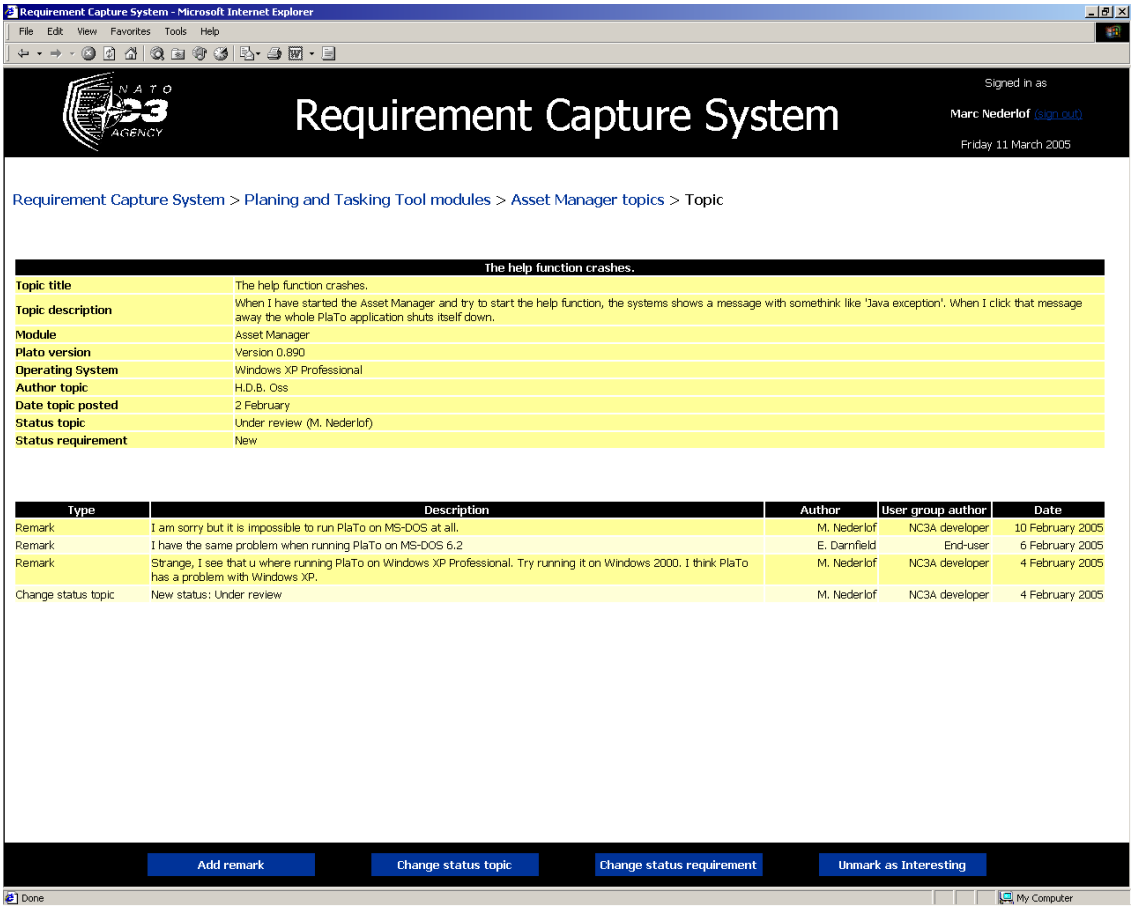
Er is besloten om een conceptueel prototype te maken door middel van HTML webpages. Op die manier kunnen de opdrachtgevers een realistisch beeld krijgen van het te ontwerpen systeem. Aan dit prototype is veel tijd besteed. Het bevat weliswaar geen echte functionaliteit maar deze wordt wel gesimuleerd, zodat er een soort demonstratie kan worden gegeven over hoe het systeem zal gaan werken. Er is heeft besloten om in eerste instantie alleen de functionaliteit van de gebruikersgroep 'End-users' te verwerken.

Tijdens een meeting met de opdrachtgevers en andere toekomstige gebruikers is met behulp van een beamer het eerste Conceptual Prototype gepresenteerd. Door middel van deze presentatie hebben de toekomstige gebruikers een helder beeld kunnen krijgen van het te ontwikkelen systeem en is er ook veel feedback verkregen. Doordat er voor deze meeting verschillende gebruikers zijn uitgenodigd was er ook plaatst voor discussie waardoor onduidelijkheden besproken werden en extra veel feedback is verkregen.

8.4.3 Tweede Conceptual Prototype

Toch bleek het dat er nog heel veel systeemeisen onduidelijk zijn omdat de functionaliteit van de gebruikersgroep 'NC3A developer en Administrator' niet besproken zijn in het prototype. Daarom is er besloten om nog een prototype te ontwikkelen gebaseerd op de use cases voor de gebruikersgroep 'NC3A developer'. Deze activiteit is uiteindelijk beschouwd als de tweede iteratie binnen de Inception fase.

Een onderdeel van het tweede Conceptual Prototype zal nu besproken worden. Het betreft de use case ‘View topic’ waarin de gebruikers alle informatie over een topic kunnen bekijken. Voorafgaand aan dit scherm krijgen de gebruikers een lijst met alle topics te zien. Door op de titel van het topic te klikken komen ze in dit scherm terecht waarop alle informatie over een topic te vinden is. Het scherm wordt hieronder afgebeeld.



Figuur 8.4-2 Screenshot van het tweede Conceptual Prototype

Het scherm is ingedeeld in drie frames te weten een header, het hoofdscherm en een footer. Deze indeling wordt door heel het systeem gehandhaafd.

In de header is een logo van het NATO C3 Agency en de titel van het systeem afgebeeld. Rechts de naam van de ingelogde gebruiker en de huidige datum getoond. Achter de naam van de ingelogde gebruikers bevindt zich een link ‘Sign out’ die gebruikt kan worden om uit te loggen.

Bovenaan in het hoofdscherm bevindt zich een navigatiestructuur. Doormiddel van dit gedeelte kan de gebruiker zien waar hij/zij zich bevindt en tevens kan de gebruiker terug gaan naar een ander niveau door op de link in de navigatiestructuur te klikken. Vervolgens wordt in de eerste kolom alle informatie over het geselecteerde topic weergegeven. Deze informatie bestaat uit:

- Topic title
- Topic description
- Module
- Plato version
- Operating System
- Author topic
- Date topic posted
- Status topic
- Status requirement

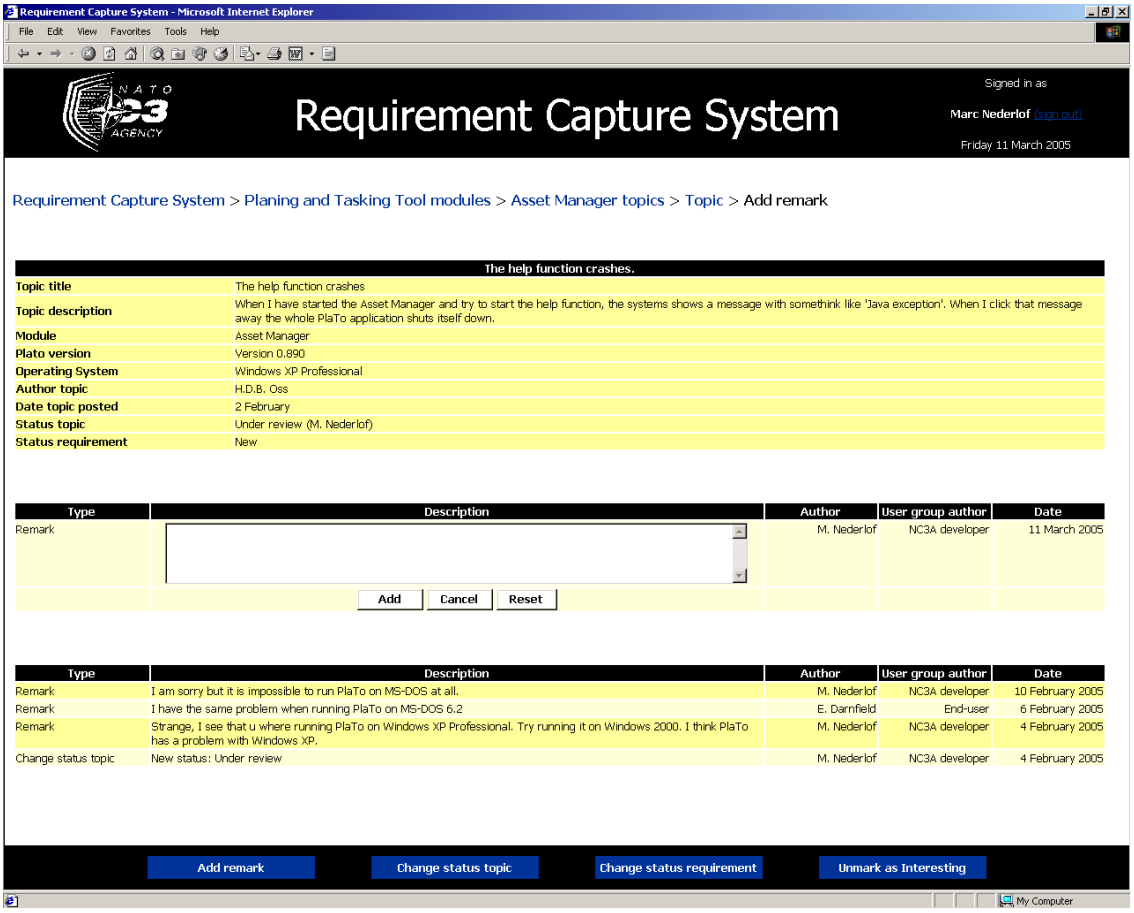
In de tweede kolom worden, gesorteerd op datum en tijd, alle statuswijzigingen en 'Remarks' getoond. De informatie die daarover getoond wordt bestaat uit:

- Type (Dit kan 'Remark' zijn of 'Change topic status')
- Description
- Author
- User group author
- Date

In de footer zijn in dit scherm vier buttons te zien. Deze buttons zullen alleen op dit scherm te zien zijn. Gedurende het gehele systeem zullen er, afhankelijk van welk scherm er in het hoofdscherm getoond wordt, verschillende buttons beschikbaar zijn. Op dit moment worden de volgende buttons weergegeven:

- Add remark. Door middel van deze button kunnen gebruikers een 'Remark' toevoegen aan het topic
- Change status topic. Om de status van het topic te wijzigen kunnen gebruikers op deze button klikken.
- Change status requirement. Om de status van het requirement, wat is gegenereerd naar aanleiding van dit topic, te veranderen kan op deze button geklikt worden.
- Unmark as interesting. Door middel van deze button kan de gebruiker aangeven dat hij/zij dit topic interessant vindt of niet meer interessant vindt.

Hieronder zal de use case ‘Add remark’ besproken worden. Deze use cases vindt plaatst als de gebruiker op de ‘Add remark’ button klikt tijdens het bekijken van een topic (use case ‘View topic’).



Figuur 8.4-3 Screenshot van het tweede Conceptual Prototype

Dit scherm ziet er hetzelfde uit als die van de use case ‘View topic’ behalve dat in dit scherm een nieuwe kolom te zien is die tussen de twee bestaande kolommen is geplaatst. In deze kolom kan de gebruiker een ‘Remark’ toevoegen. Informatie die al bekend is wordt getoond. Dit is:

- Type
- Author
- User group author
- Date

De gebruiker kan bij ‘Description’ de tekst invoeren. Er bevinden zich drie buttons in de nieuwe kolom. Dit zijn:

- Add. Door middel van deze button kan de gebruiker de door hem/haar ingevoerde tekst wegschrijven. Vervolgens zal de gebruiker terug gaan naar de use case ‘View topic’ waarin ook de zojuist toevoegde ‘Remark’ te zien zal zijn.
- Cancel. Als de gebruiker op deze button klikt wordt het toevoegen van een ‘Remark’ geannuleerd
- Reset. Door middel van deze button kan de gebruiker de door hem/haar eventuele ingevoerde tekst in de textbox wissen.

8.5 Opstellen van de Business Case

De Business Case beschrijft volgens RUP onder andere de economische waarde van het project. Dit zijn gedeeltes die eigenlijk direct uit het Project Plan komen. Er is besloten om die gedeeltes in het Project Plan te laten staan als oorspronkelijk en de Business Case te gebruiken als mogelijkheid om deze aan te passen in elke iteratie.

8.6 Opstellen van de Development Case

Volgens RUP wordt de Development Case gebruikt om de uitvoering van het project te beschrijven. Hierin is vastgelegd welke middelen en tools worden gebruikt. Ook is hierin een aangepaste planning meegenomen. In principe komen deze zaken, net zoals de Business Case, direct uit het Project Plan. Er is besloten om die gedeeltes in het Project Plan te laten staan als oorspronkelijk en de Development Case te gebruiken als mogelijkheid om deze aan te passen in elke iteratie.

8.6.1 Aanpassen van de globale planning

In de Development Case van de Inception fase is er een aangepaste planning meegenomen. De globale planning uit de Development Case wordt hieronder weergegeven.

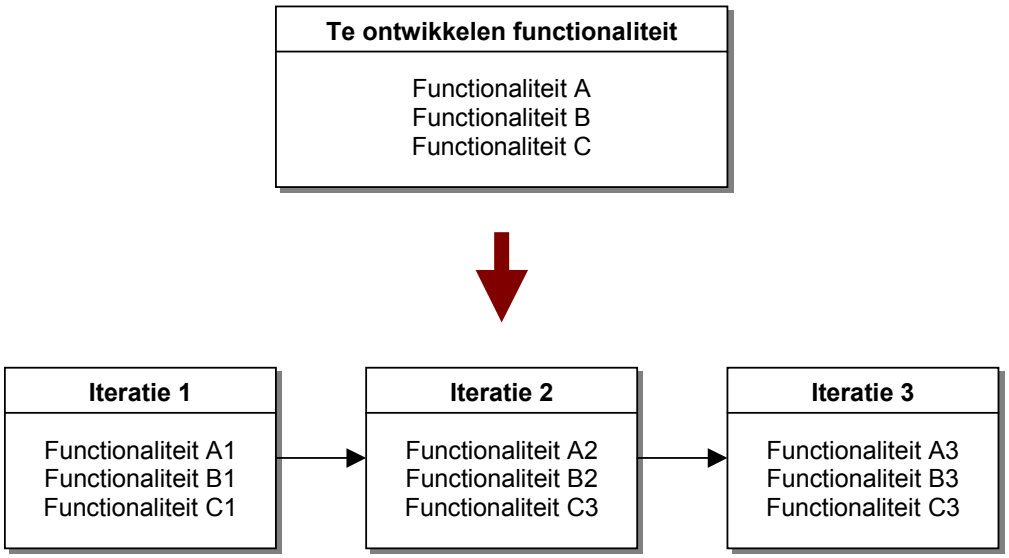
Week	Calendar week	Phase	Activity	
6	11 (14-3 t/m 18-3)	Elaboration (iteration 1)	Refine Vision Document (1 day) Design critical use cases (2 days) Design and implement database(2 days)	
7	12 (21-3 t/m 25-3)	Elaboration (iteration 1)	Built an Executable Prototype (6 days) Test the Executable Prototype (2 days) Refine Business Case (1 day) Refine Development Case (1 day)	
8	13 (28-3 t/m 1-4)	Elaboration (iteration 1)		
9	14 (4-4 t/m 8-4)	Elaboration (iteration 2)	Refine Vision Document (1 day) Redesign critical use cases (2 days) Redesign and implement database (2 days)	
10	15 (11-4 t/m 15-4)	Elaboration (iteration 2)	Rebuilt an Executable Prototype (5 days)	
11	16 (18-4 t/m 22-4)	Elaboration (iteration 2)	Test the Executable Prototype (1 days) Refine Business Case (1 day) Refine Development Case (1 day) Write Lifecycle Architecture Milestone report (2 days)	
12	17 (25-4 t/m 29-4)	Construction (iteration 1)	Refine Vision Document (1 days) Design use cases (4 days)	
13	18 (2-5 t/m 6-5)	Construction (iteration 1)	Build Beta Release (6 days) Test Beta Release (2 days) Refine Business Case (1 day) Refine Development Case (1 day)	
14	19 (9-5 t/m 13-5)	Construction (iteration 1)		
15	20 (16-5 t/m 20-5)	Construction (iteration 2)	Refine Vision Document (1 days) Redesign use cases (4 days)	
16	21 (23-5 t/m 27-5)	Construction (iteration 2) / End	Rebuild Beta Release (5 days)	Write final internship report (15 days)
17	22 (30-5 t/m 3-6)	Construction (iteration 2) / End	Test Beta Release (1 day) Refine Business Case (1 day) Refine Write Initial Operational Capability Milestone report (2 days) Development Case (1 day)	
18	23 (6-6 t/m 10-6)	End		

Figure 7.3-1 Global planning

Figuur 8.6-1 Aangepaste globale planning uit de Development Case

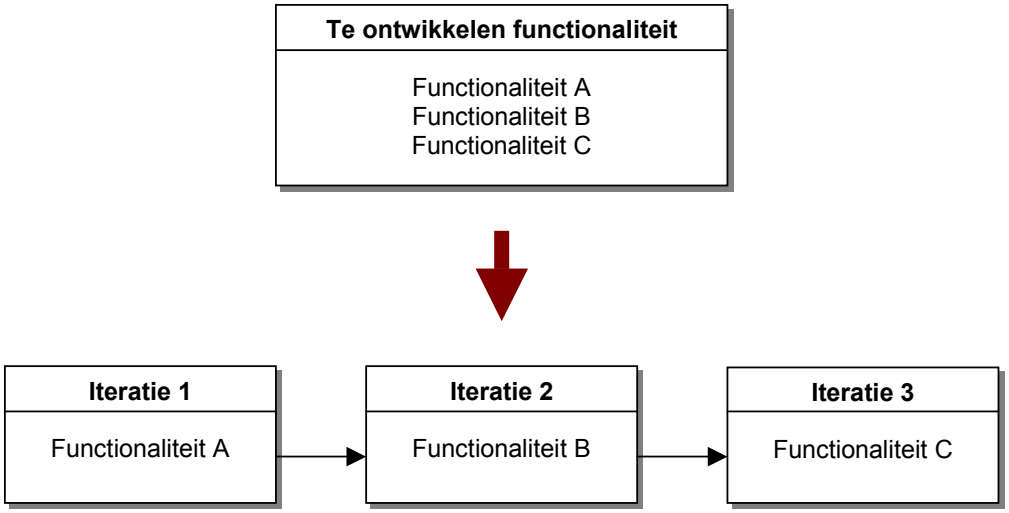
In overleg met de opdrachtgevers en de examinatoren van Haagse Hogeschool zal de Transition fase niet tijdens het afstudeertraject uitgevoerd worden. Deze is daarom uit de planning gehaald. Ook zijn in deze planning verschillende iteraties ingepland. Dit is gebeurd naar aanleiding van een gesprek met de examinatoren aan de Haagse Hogeschool.

Er waren twee mogelijkheden om de verschillende iteraties in te plannen. Per iteratie zou van elke functionaliteit tenminste een gedeelte kunnen worden ontwikkeld waarbij in de volgende iteratie de rest van die functionaliteit verder wordt ontwikkeld. Dit is in de onderstaande figuur schematisch afgebeeld.



Figuur 8.6-2 Iteraties inplannen: strategie 1

Een andere mogelijkheid om de verschillende functionaliteiten te verdelen over de iteraties. Elke functionaliteit wordt in dat geval gedurende een bepaalde iteratie ontwikkeld. Ook hiervan wordt een schematische weergave afgebeeld.



Figuur 8.6-3 Iteraties inplannen: strategie 2

Er is gekozen voor de tweede strategie, het onderverdelen van functionaliteiten in iteraties. Op deze manier konden de functionaliteiten op prioriteit worden ingedeeld in iteraties. De belangrijkste functionaliteiten zijn ingedeeld in de eerste iteratie en de functionaliteiten met een lagere prioriteit in de tweede iteratie. Het prioriteren van de functionaliteit is gedaan aan de hand van gesprekken met de opdrachtgevers. Zij hebben aangegeven per functionaliteit hoe groot het belang hiervan was. Het voordeel van deze strategie was dat na elke iteratie de functionaliteit geëvalueerd kon worden met de opdrachtgevers. Als er voor de eerste strategie zou zijn gekozen zou dat betekenen dat aan het eind van een iteratie de functionaliteit in meer of mindere mate nog niet af is. In dat geval zou er feedback kunnen worden verkregen over zaken die al gepland waren in de volgende iteratie. De iteraties zijn als volgt ingedeeld:

Iteratie 1:

- Kritieke use case nr. 1: Sign in
- Kritieke use case nr. 2: Provide feedback

Iteratie 2:

- Kritieke use case nr. 4: View feedback
- Kritieke use case nr. 5: Change feedback status
- Kritieke use case nr. 6: Add developer note
- Kritieke use case nr. 7: Add user note

In de tweede iteratie zijn er meer functionaliteiten ingedeeld dan in de eerste iteratie. Dit is gedaan omdat er vanuit is gegaan dat er in de eerste iteratie meer tijd kwijt zal zijn in het leren van de programmeertaal/scriptaal waarin het systeemontwikkeld wordt. Als dat eenmaal onder de knie zou zijn dan zou het ontwikkelen van de functionaliteiten in de tweede iteratie minder tijd moeten kosten.

Hoofdstuk 9 - Elaboration fase

In dit hoofdstuk worden de uitgevoerde werkzaamheden tijdens, en de opgeleverde producten uit de Elaboration fase besproken. De Elaboration fase is de tweede fase die in RUP omschreven wordt. In deze fase wordt het Vision Document bijgewerkt, het Software Architecture Document wordt bijgewerkt op basis van nieuwe of bijgewerkte ontwerpen, een Executable Prototype, de eventuele herziene Business Case en de herziene Development Case met een aangepaste planning opgeleverd. Uiteindelijk wordt dit alles verwerkt in een rapport de 'Lifecycle Architecture Milestone'. Ook dit rapport is als externe bijlage toegevoegd aan dit afstudeerverslag. Dit rapport is net als het Executable Prototype een mijlpaal uit de Elaboration fase.

9.1 Bijwerken van het Vision Document

In de Elaboration fase er was ruimte om het Vision Document aan te passen op basis van de resultaten uit de Inception fase of eventuele andere wijzigingen met betrekking tot de afstudeeropdracht. In dit geval waren die er zeker. In deze paragraaf worden de aanpassingen en toevoegingen aan het Vision Document besproken.

9.1.1 Aanpassen van het Statement of Work

Het Statement of Work is aangepast naar aanleiding van de definitieve opdrachtomschrijving in overleg met en met goedkeuring van de examinatoren aan de Haagse Hogeschool en de bedrijfsmentoren/opdrachtgevers van het NATO C3 Agency. In principe kwam dit neer op een letterlijke vertaling van de relevante delen uit de definitieve opdrachtomschrijving.

9.1.2 Aanpassen van de Features

De features van het systeem zijn enigszins veranderd. De feature 'Edit feedback' is verwijderd en er zijn twee nieuwe features bijgekomen te weten 'Generate requirement' en 'Change requirement status'. Deze worden hieronder weergegeven.

Feature No.4	
Name	Generate requirement
Description	Allow Developers to generate a requirement based on feedback provided on the software prototype systems LSID and PlaTo.
Value	1.
Complexity	Medium.
Priority	4.

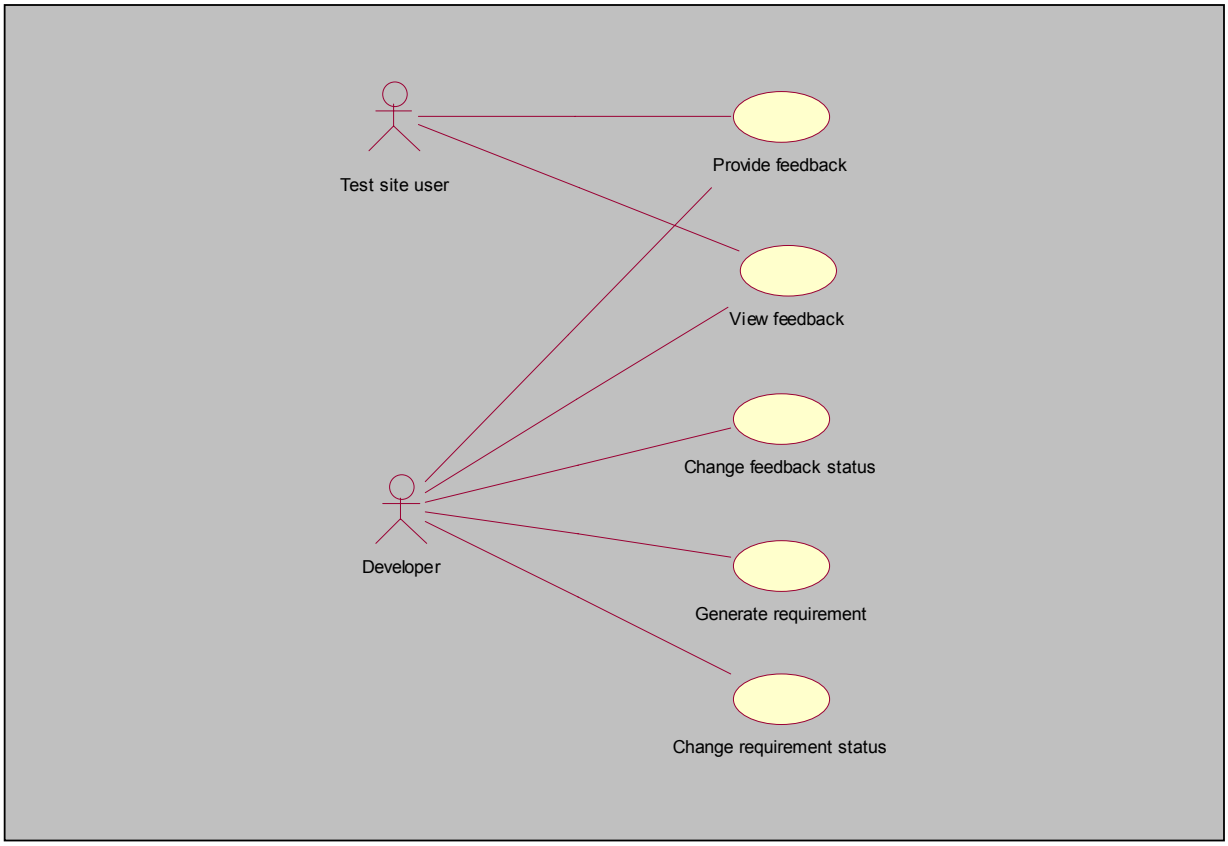
Feature No.5	
Name	Change requirement status
Description	Allow Developers to change the status of a generated requirement.
Value	4.
Complexity	High.
Priority	4.

Figuur 9.1-1 Nieuw features uit het Vision Document

Tijdens het opstellen van de features in de Inception fase zijn deze nieuwe features niet naar bovengekomen. Dit kwam omdat op dat moment de wensen en eisen van de opdrachtgevers nog niet helemaal helder waren. Er werd sowieso rekening gehouden met de mogelijkheid dat wat de features betreft er nog het een en ander zou kunnen worden veranderd. Ook de feature 'Edit feedback' bleek niet nodig te zijn voor het systeem. Deze veranderingen zijn allemaal duidelijk geworden tijdens de evaluaties van de twee Conceptual Prototypes uit de Inception fase. Het zou in principe mogelijk zijn om de features helemaal helder te krijgen in de Inception fase. Dit zou echter veel meer tijd gaan kosten. Dit geld ook voor de key use cases, de 'mile-wide, inch-deep' beschrijving en de beschrijving van de kritieke use cases. De bedoeling van RUP is echter dat de systeemeisen evolueren gedurende het gehele project door middel van prototyping. RUP is echter een iteratieve systeemontwikkeling-methodiek waarin geen watervalmethode wordt toegepast.

9.1.3 Aanpassen van de Key use cases

Net als de features zijn de key use cases op dezelfde manier gewijzigd. De key use case 'Edit feedback' is verwijderd en de use cases 'Generate Requirement' en 'Change requirement status' zijn toegevoegd. De key use cases en het use case diagram zijn eigenlijk een vertaling vanuit de features. Dit resulteerde in het onderstaande key use case diagram.



Figuur 9.1-2 Key use case diagram uit het Vision Document

De opdrachtgevers hebben aangegeven dat de benaming van de actoren anders moet zijn. Met name 'End-user' die veranderd is in 'Test site user'. De term 'End-user' is namelijk voortgekomen uit taal-verwarring omdat de eindgebruikers van de uiteindelijke operationele software niet per definitie

dezelfde mensen zijn die gebruik gaan maken van de prototypen die door het NATO C3 Agency zijn ontwikkeld.

9.1.4 Aanpassen van de Non-functional requirements

Qua non-functional requirements is er het een en ander veranderd. Uit onderzoek en gesprekken met de opdrachtgevers is gebleken dat het wenselijk is dat er een web-based applicatie ontwikkeld wordt die in eerste instantie gehost zal worden op een NATO UNCLASSIEFIED server waardoor het systeem toegankelijk zal worden via het Internet.

Er is binnen het NATO C3 Agency uitgezocht wat de mogelijkheden zijn voor het hosten van een systeem op een NATO UNCLASSIFIED server. Het systeem zou gehost kunnen worden op één van de servers binnen het NATO C3 Agency waarop op dat moment al een aantal portals gehost werden. Als voorwaarde werd wel gesteld dat de webapplicatie gemaakt wordt in Sharepoint Portal Server (SPS) of Windows SharePoint Services (WSS).

9.2 Definiëren van de ontwikkelomgeving

Uiteindelijk is het Executable Prototype dat ontwikkeld is in de Elaboration fase niet ontwikkeld met behulp van SharePoint, maar met behulp van een rich-tekst editor, Visual Studio .NET en MS Access. In deze paragraaf wordt beschreven hoe tot dat besluit is gekomen.

9.2.1 Onderzoek naar SharePoint uitvoeren

Omdat de afstudeerder met SharePoint totaal niet bekend was is er onderzoek gedaan naar wat SharePoint is, hoe het werkt en wat de mogelijkheden zijn. In de organisatie was slechts één boek over SharePoint beschikbaar te weten ‘Microsoft SharePoint 2003 Unleashed’. In het TMD lab heeft één van de medewerkers SharePoint Portal Server, wat standaard bij Windows 2003 server wordt geleverd, zo ingesteld dat de afstudeerder toegang had tot de portal vanaf zijn werkplek. Een aantal medewerkers van andere afdelingen binnen de organisatie hadden ervaring met het maken van portals met behulp van SharePoint. Met een aantal van deze mensen zijn er afspraken gemaakt zodat ook op die manier kon worden uitgezocht hoe SharePoint werkte en wat de mogelijkheden precies waren.

SharePoint is een systeem dat kan worden gebruikt door organisaties om gebruikers, teams en kennis te koppelen, zodat mensen relevante informatie kunnen gebruiken ter ondersteuning van een bedrijfsproces. SharePoint fungeert als een portal. Dit is te vergelijken met een soort intranet waarop het mogelijk is om van een plaats op het web naar een andere plaatst toe te gaan en gebruik te maken van diverse services. De portal stelt mensen in staat om samen te werken en informatie uit te wisselen.

SharePoint biedt een soort basisopzet voor de ontwikkelaar om te starten. Dit pakket bevat bepaalde standaard features die niet meer zelf ontwikkeld behoeven te worden, maar ‘out of the box’ worden meegeleverd dat veel ontwikkeltijd zou kunnen besparen. De feature die interessant zou zijn voor de afstudeeropdracht is de Forum feature. Hiermee kunnen gebruikers berichten met elkaar uitwisselen en discussiëren over uiteenlopende onderwerpen. De web-parts kunnen in SharePoint door de ontwikkelaar gebruikt worden om zijn/haar portal mee in te richten. Deze web-parts zijn gemaakt in ASP.NET. Een ontwikkelaar kan ook zijn/haar eigen web-parts ontwikkelen. Deze kunnen dan op dezelfde wijze als de standaard web-parts in de portal worden geïntegreerd.

Uit onderzoek is gebleken dat de standaard Forum feature van SharePoint niet voldoet aan de wensen van de opdrachtgevers binnen het NATO C3 Agency. Binnen SharePoint is het echter wel mogelijk om een forum min of meer aan te passen naar de behoeften van de organisatie. Toch zal het

waarschijnlijk niet mogelijk zijn om deze standaard feature aan te passen zodat het voldoet aan de wensen en eisen van de opdrachtgevers.

SharePoint is geen pakket waarmee gewoonweg webpagina's gemaakt kunnen worden. Alles binnen SharePoint is geschreven in ASP.NET en wijzingen kunnen niet zomaar zoals bij een HTML pagina worden toegevoegd. Alle standaard functionaliteiten zitten binnen SharePoint weggewerkt in de systeemarchitectuur. Het is echter wel mogelijk om een eigen ASP.NET webapplicatie te maken en deze als webpart te integreren in een SharePoint Portal.

9.2.2 Bekendraken met ASP.NET

Ook met ASP.NET was de afstudeerder onbekend. Daarom is er begonnen met het lezen van een aantal boeken over ASP.NET. Ook is op het Internet het een en ander opgezocht over ASP.NET. Dit heeft is gedaan om een globaal beeld te krijgen en om uit te vinden op welke manier een ASP.NET applicatie ontwikkeld kan worden.

9.2.3 Bepalen van de ontwikkelomgeving voor ASP.NET

Een ASP.NET applicatie kan grofweg op twee manieren worden ontwikkeld. Met behulp van een teksteditor zoals Notepad of met behulp van een meer geavanceerde ontwikkelomgeving zoals Visual Studio .NET 2003. Deze mogelijkheden worden nu kort besproken.

Notepad-'.NET

Het heet niet echt Notepad '.NET' maar net zoals er mensen die deze tool 'Visual' Notepad noemden, is er ook de benaming Notepad '.NET' ontstaan. Omdat Notepad op vrijwel elk Microsoft Windows-platform is te vinden bestaat er waarschijnlijk geen eenvoudiger ontwikkeltool. Een ander voordeel van Notepad is de prijs. Notepad is namelijk gratis en er hoeft niets voor te worden gedownload. Om met ASP.NET en Notepad te ontwikkelen is het alleen vereist dat de SDK van het .NET Framework geïnstalleerd is. Het maken van een ASP.NET pagina werkt op dezelfde manier als het maken van een HTML document met Notepad. Het document moet echter wel worden opgeslagen met de extensie '.aspx'.

Naast Notepad zijn er allerlei third-party teksteditor met mogelijkheden die voorheen alleen beschikbaar waren in de betere ontwikkeltools. Enkele mogelijkheden zijn:

- Kleuring van de syntaxis. Deze voorziening gebruikt kleur om verschillende typen keywords aan te geven in de taal die voor het ontwikkelen wordt gebruikt.
- Automatische aanvulling. Deze voorziening stelt mogelijke aanvullingen van statements voor terwijl er getypt wordt, op basis van wat er al eerder is ingevoerd.

Visual Studio .NET

Voor het ontwikkelen van onder andere ASP.NET Webapplicaties heeft Microsoft Visual Studio .NET ontwikkeld, de nieuwste versie van de ontwikkelsuite Visual Studio. Visual Studio .NET heeft de volgende voordelen:

- Robuust beheer van projectbestanden en verschillende projecten
- Integratie met de broncodebeheeromgeving
- Visuele tools voor het werken met webservices, servercontrols in webformulieren en data-basetools
- Services voor het bundelen en gebruiken van webapplicaties

Er is heeft gekozen om het Executable Prototype te ontwikkelen met behulp van een rich-text editor die de ASP.NET syntax ondersteunt. Dit is besloten omdat de layout al ontwikkeld is tijdens de ontwikkeling van de Conceptual Prototype. Hiervoor zou Visual Studio .NET dus niet meer nodig zijn. Ook betreft het maar één ASP.NET applicatie waardoor de voordelen van Visual Studio .NET zoals het beheer van verschillende projectbestanden en het bundelen van verschillende webapplicaties geen voordelen met zich mee zou brengen.

9.2.4 Programmeertaal kiezen voor ASP.NET

In ASP.NET kan er gebruik worden gemaakt van veel verschillende programmeertalen. De meest gebruikte zijn Visual Basic .NET, C# en in mindere mate Jscript .NET en Visual C++. Voorbeelden van andere talen die gebruikt kunnen worden zijn:

- APL
- Cobol
- Pascal
- Eiffel
- Haskell
- ML
- Oberon
- Perl
- Pyton
- Scheme
- Smalltalk

Omdat de twee ASP.NET boeken die beschikbaar waren voornamelijk Visual Basic .NET hanteerde is er besloten om te kiezen voor Visual Basic .NET als programmeertaal.

9.2.5 Bepalen van de database omgeving

De NATO UNCLASSIFIED server, waar SharePoint Portal Server op draait binnen het NATO C3 Agency, maakt gebruikt van Microsoft SQL Server 2000. Dit was niet geïnstalleerd op de werkplek van de afstudeerder. ASP.NET kan ook gebruik maken van een MS Access database. Dit zou voor de testomgeving in het TMD lab krachtig genoeg moeten zijn. De MS Access database zou eventueel in een later stadium kunnen worden geconverteerd naar een Microsoft SQL Server 2000 database. Aangezien MS Access standaard op elk werkstation was geïnstalleerd is er gekozen om deze als database omgeving te gebruiken.

9.3 Bijwerken van het Software Architecture Document

In deze paragraaf wordt de totstandkoming van nieuwe ontwerpen en aanpassingen aan bestaande ontwerpen uit het Software Architecture Document dat is bijgewerkt tijdens de Elaboration fase beschreven.

9.3.1 Aanpassen van de 'Mile-wide, inch-deep' beschrijving

Op basis van de evaluatieresultaten van de Conceptual Prototypes uit de Inception fase is ook de 'mile-wide, inch-deep' beschrijving aangepast. Er zijn een aantal use cases weggelaten, samengevoegd, opgedeeld en er bij gekomen. Het grootste verschil met de 'Mile-wide, inch-deep' beschrijving is de actoren. In de Inception fase waren dat er twee, te weten 'End-user' en 'NC3A developer'. Tijdens het bijwerken van het Vision Document is al gebleken dat de actor 'End-user' de naam 'Test site user' moet krijgen. Tijdens de Elaboration fase is de actor 'NC3A developer' opgedeeld in verschillende actoren. Elke actor zal verschillende rechten hebben. Tijdens de Inception fase was dit nog niet duidelijk voor de opdrachtgevers en werd er slechts alleen de actor 'NC3A developer' gehanteerd. De actoren zoals gedefinieerd tijdens de Elaboration fase worden in de onderstaande figuur weergegeven.

Actor No.1	
Name	Test site user
Description	A Test site user is a person who is using and testing the software prototype systems developed by NC3A.

Actor No.2	
Name	Analyst
Description	An Analyst is a person who is designing software prototype systems at NC3A.

Actor No.3	
Name	Developer
Description	A Developer is a person who is developing software prototype systems at NC3A.

Actor No.4	
Name	Project manager
Description	A Project manager is a person who is in charge of a project at NC3A.

Actor No.5	
Name	Administrator
Description	An Administrator is a person who is going to be the administrator for the Requirement Capture System that has to be developed.

Figuur 9.3-1 Actoren uit de 'Mile-wide, inch-deep' beschrijving

9.3.2 Bijwerken van de Kritieke use cases

De kritieke use cases zijn aangepast en verder uitgewerkt naar aanleiding van de evaluatie van de twee Conceptual Prototypes uit de Inception fase. De use cases zijn ook geprioriteerd in overleg met de opdrachtgevers. Dit is gedaan om het Executable Prototype dat in deze Elaboration fase ontwikkeld gaat worden te verdelen over iteraties. De onderverdeling van de kritieke use cases onder de iteraties zal gebeuren op basis van prioriteit. De twee nieuwe kritieke use cases 'Generate Requirement' en 'Change Requirement Status' zijn ook omschreven. Deze worden in de onderstaande figuur weergegeven.

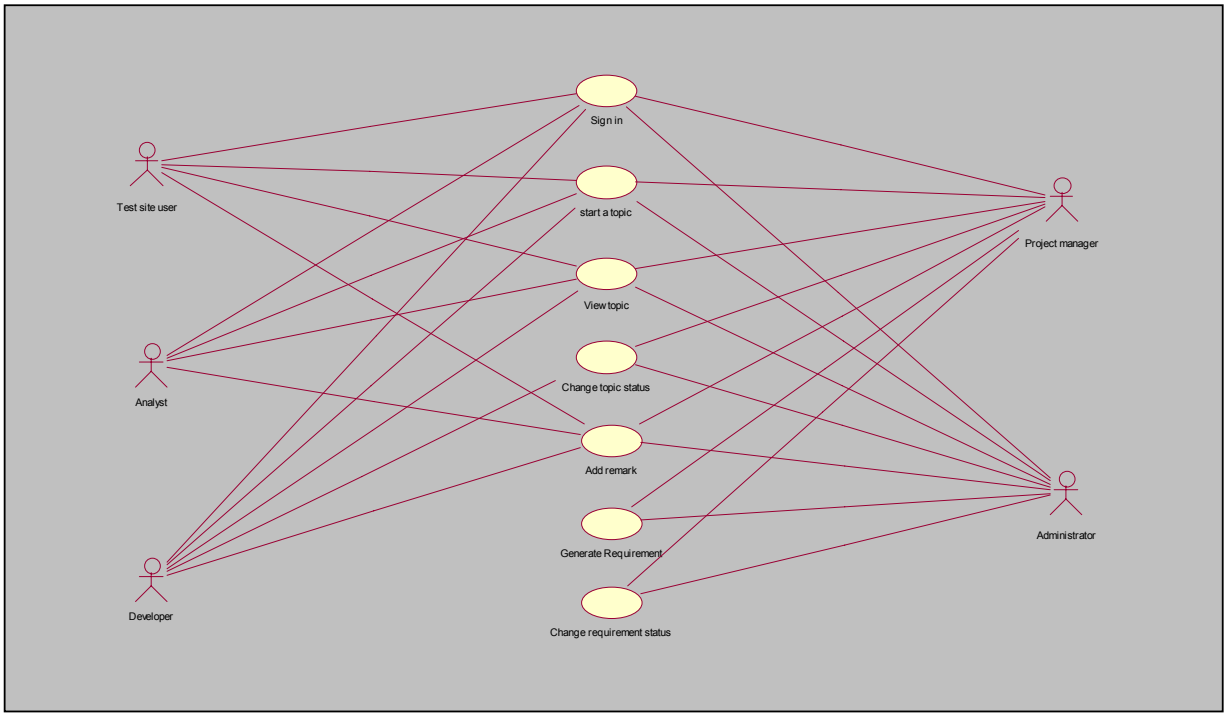
Critical use case	**No. 6**
Name	Generate requirement.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a prototype, a module and a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to generate a requirement from the current topic. (2) The system asks the user to fill in the requirement description and shows information that is already automatically known by the system. This is information contains the author, user group of the author and the date. (3) The user fills in the description of the requirement and tells the system that he/she wants to submit the requirement. (4) The system tells the user that the requirement has been submitted and shows the update topic information.
Exceptions	None.
Result	Requirement is generated.
Priority	3.

Critical use case	**No. 7**
Name	Change requirement status.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a prototype, a module, a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system for which requirement he/she wants to change the status. (2) The system asks the user what status he/she wants to give the requirement and asks for an additional description. (3) The user tells the system what the new status of the requirement must be, fills in an additional description and tells the system that he/she wants to submit the status change. (4) The system shows the user the updated topic information.
Exceptions	[Requirement status is already 'Implemented'.] The system tells the user that the status of the requirement is 'Implemented' and cannot be changed anymore.
Result	Requirement status is changed.
Priority	3.

Figuur 9.3-2 Nieuwe kritieke use cases

Deze nieuwe kritieke use cases komen oorspronkelijk voort uit de bijgewerkte key use cases in de Elaboration fase. De beschrijvingen van de kritieke use case diagrammen zijn besproken met de opdrachtgevers en zijn waar nodig aangepast.

Op basis van de kritieke use cases is er een kritieke use case diagram gemaakt om zo een duidelijk overzicht te krijgen van de verschillende actoren en hun interacties met verschillende kritieke use cases. Dit diagram wordt hieronder weergegeven.



Figuur 9.3-3 Kritieke use case diagram

9.3.3 Ontwerpen van de database

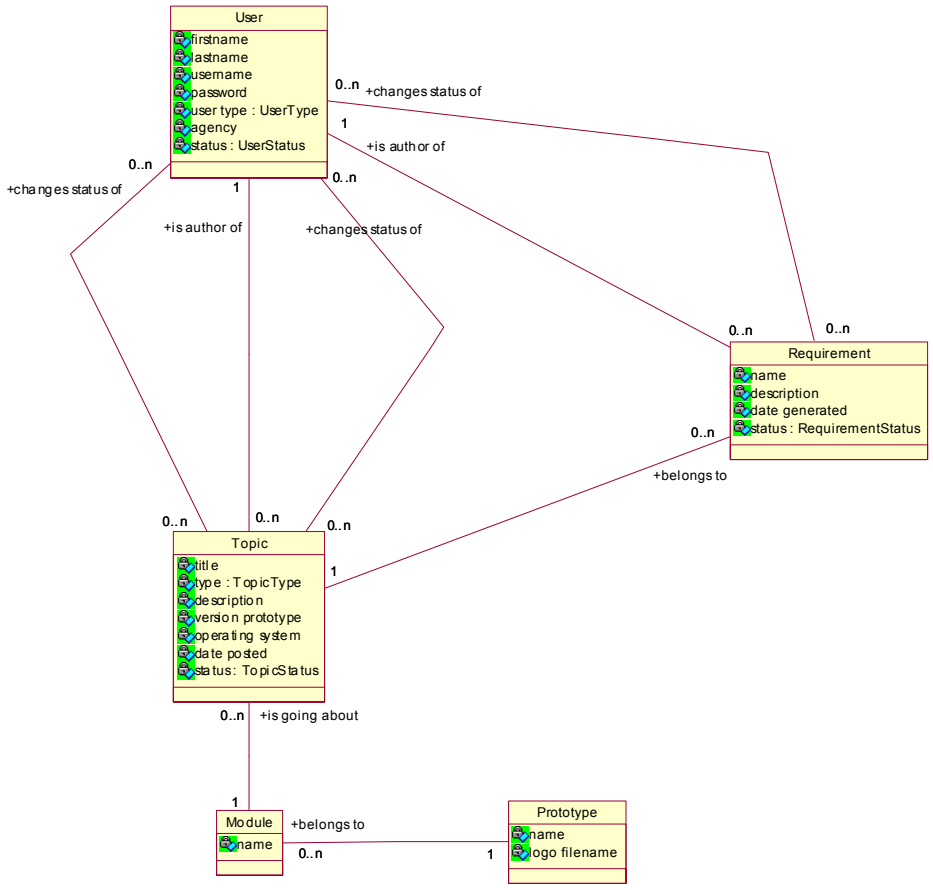
Op basis van evaluatieresultaten van het Conceptual Prototype uit de Inception fase en de bijgewerkte kritieke use cases is er begonnen aan het database ontwerp. De totstandkoming van het database ontwerp tijdens de Elaboration fase wordt in deze paragraaf nader besproken.

Klassendiagram

Er is gekozen om te beginnen met het opstellen van een klassendiagram omdat die als model goed gebruikt kan worden tijdens meetings met de opdrachtgevers. Dit omdat een klassendiagram de werkelijkheid moet weerspiegelen en de opdrachtgevers ook enigszins bekend zijn met klassen-diagrammen. Uiteindelijk zou het klassendiagram dan letterlijk vertaalt kunnen worden naar het database model. Tijdens verschillende brainstormsessies is het uiteindelijke klassendiagram tot stand gekomen. De volgende klassen werden in eerst instantie gedefinieerd:

- Prototype
- Module
- Topic
- User
- Requirement

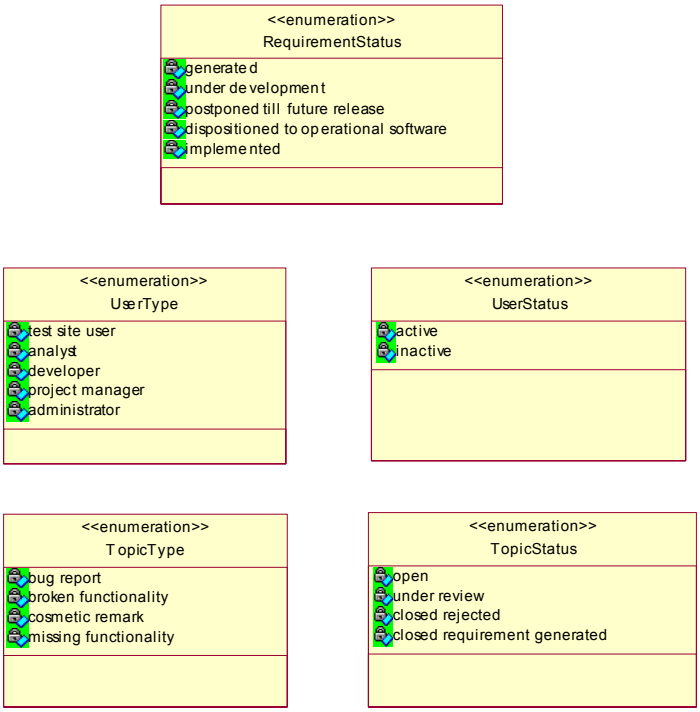
De associaties tussen deze klassen en hun attributen worden in de volgende figuur weergegeven.



Figuur 9.3-4 Associaties tussen de verschillende klassen

Zoals te zien is bevatten een aantal klassen attributen die naar een enumeratie klasse verwijzen. Het betreft de volgende attributen:

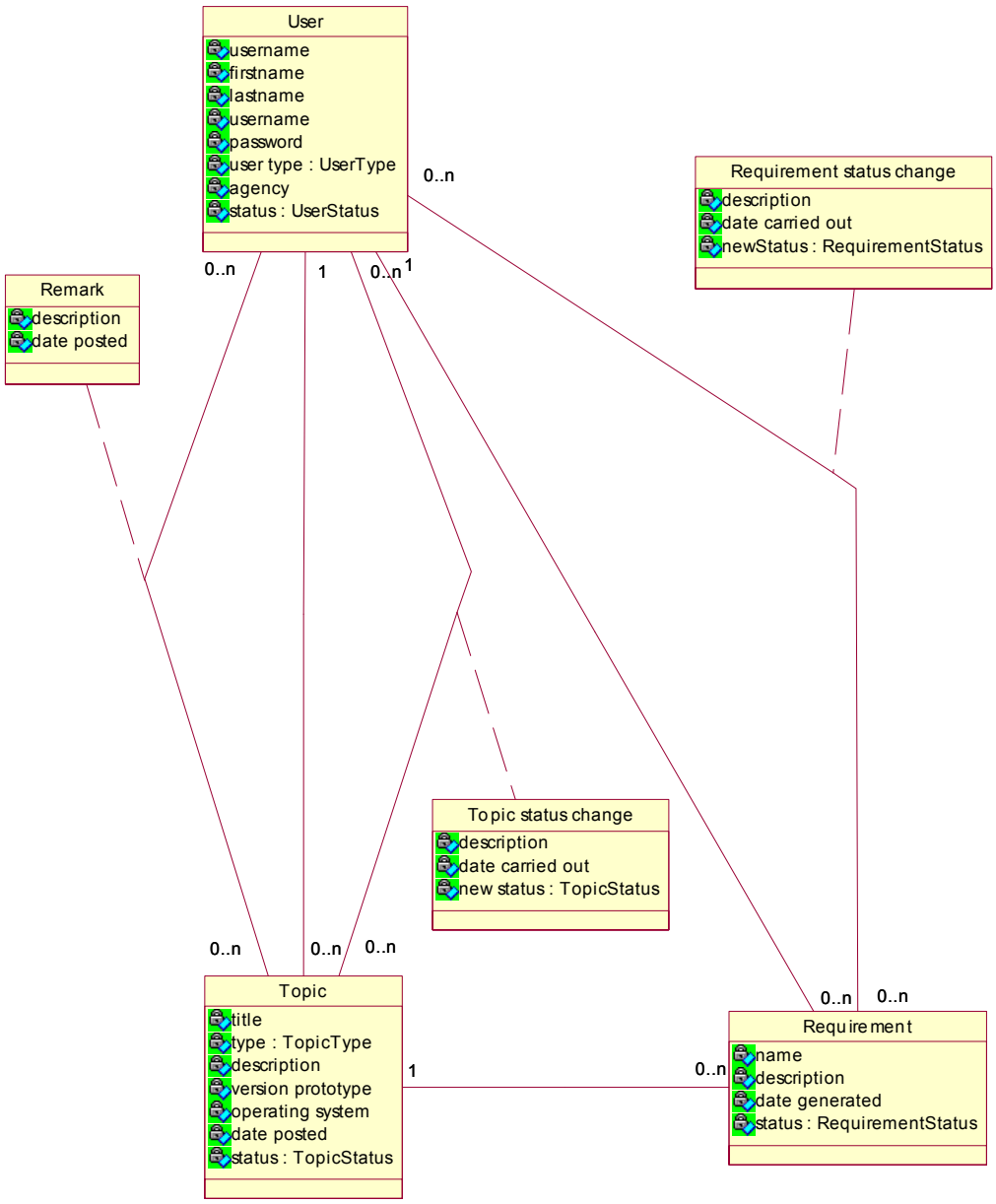
- status (Requirement)
- usertype (User)
- status (User)
- type (Topic)
- status (Topic)



Figuur 9.3-5 Enumeratieklassen

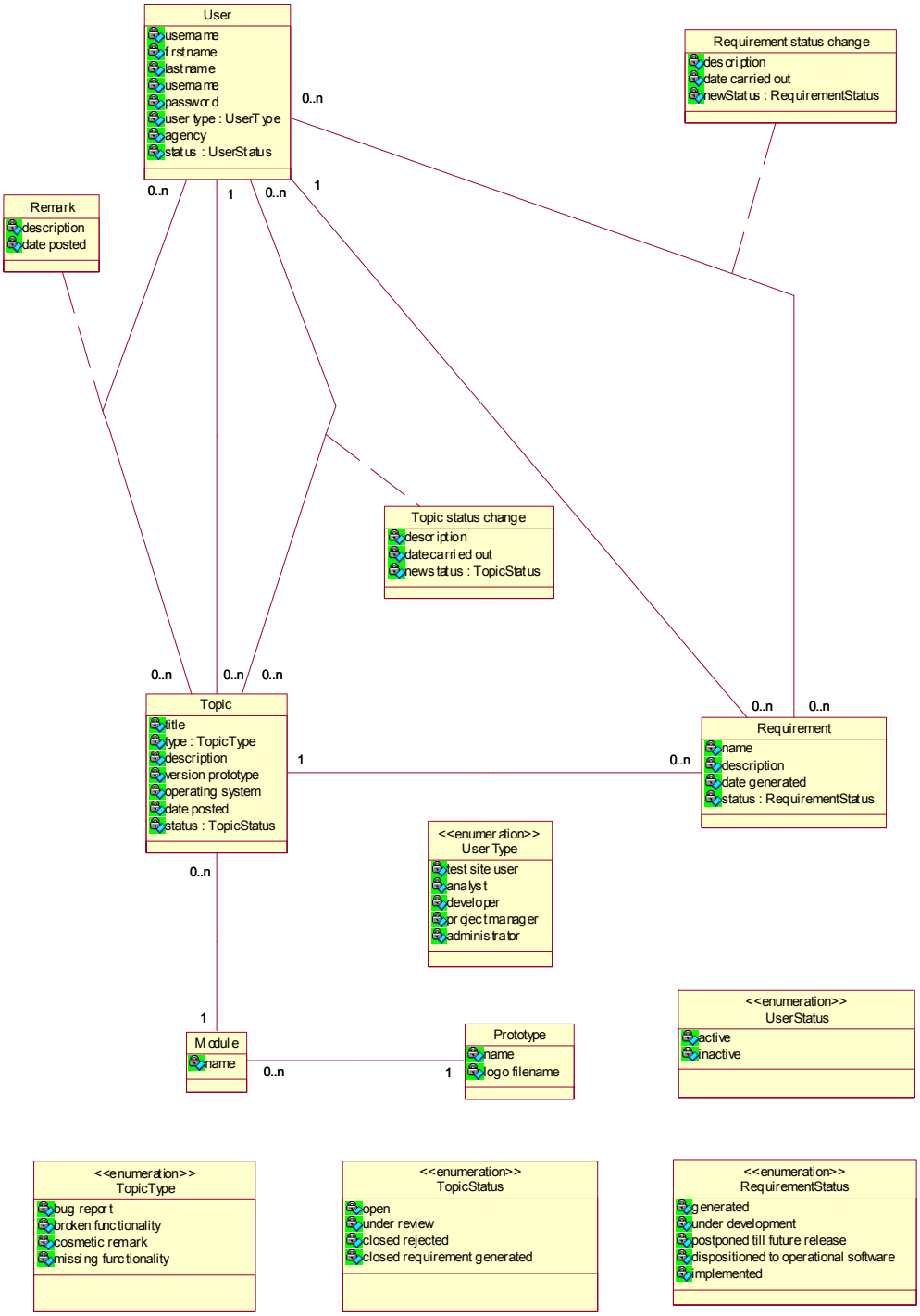
De attributen van de enumeratieklassen RequirementStatus, TopicType, TopicStatus komen puur voort uit input van de opdrachtgevers. Zij hebben aangegeven welke waarden deze moeten bevatten. De attributen uit de enumeratie klasse UserType komen voort uit de verschillende actoren die zijn gedefinieerd. Ook is er voorgesteld om een User de status Active of Inactive te geven. Dit vanwege het feit dat een User niet zomaar verwijderd kan worden als er nog verwijzingen naar zijn. Door een User de status Inactive te geven zou deze op die manier geen toegang meer hebben tot het systeem.

In het klassendiagram bevinden zich drie 'n op n' associaties. Dit zijn de twee associaties tussen de klassen User en Topic ('voegt remark toe' en 'wijzigt status') en de associatie tussen User en Requirement (wijzigt status). In deze gevallen moest er een associatieklasse worden gemaakt om dat er van deze associatie gegevens moeten worden vastgelegd zoals datum en tijd. Het moet namelijk te achterhalen zijn wanneer een Remark is toegevoegd en een 'TopicStatus change' heeft plaatst gevonden. De associatie 'wijzigt status' tussen User en Topic en die tussen User en Requirement verwijzen tevens ook naar enumeratie klassen te weten TopicStatus en RequirementStatus. De associatieklassen 'Remark', 'Topic status change' en 'Requirement status change' en de associaties worden in de onderstaande figuur weergegeven.



Figuur 9.3-6 Het gebruik van associatieklassen

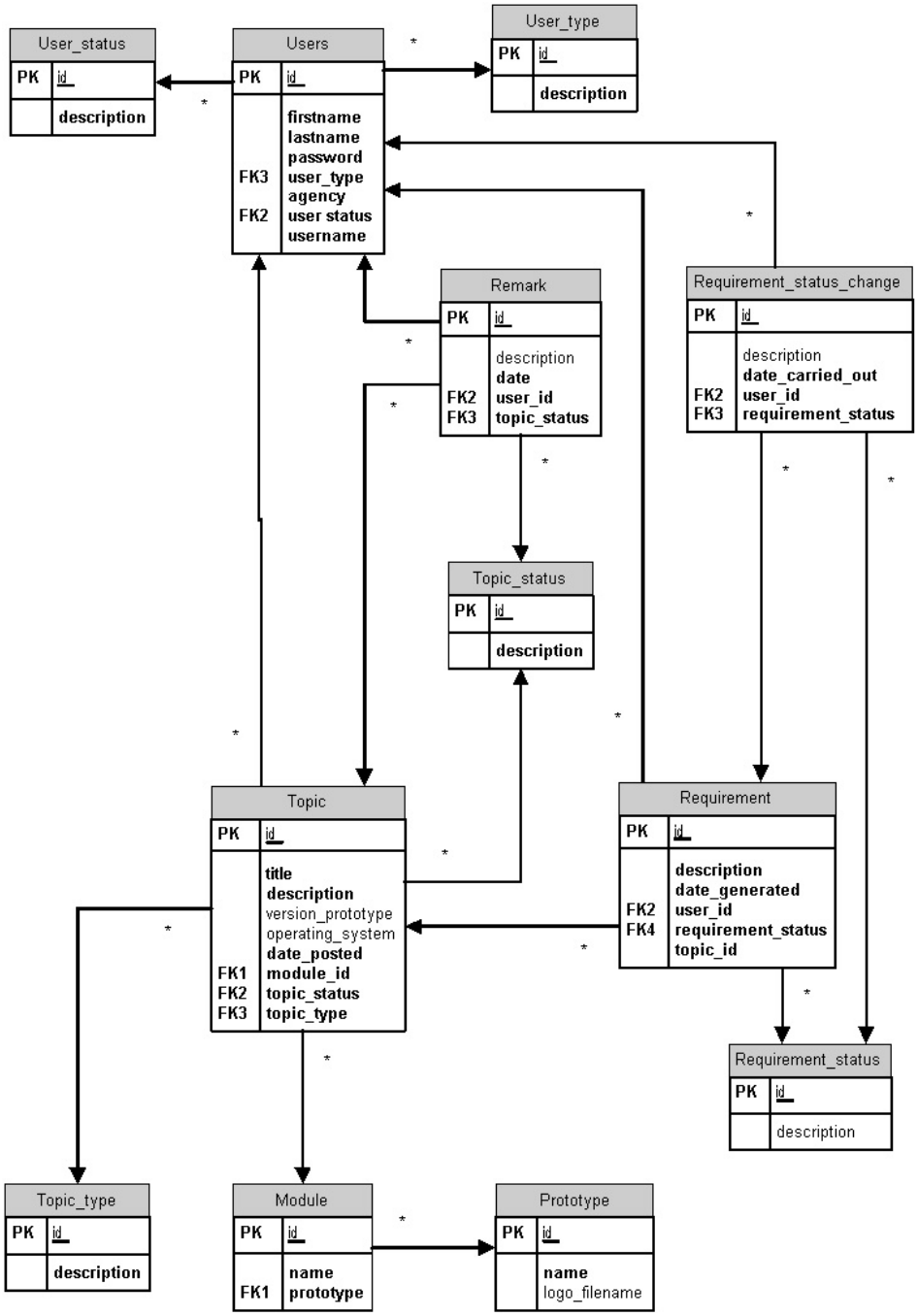
Het eindresultaat is het hieronder afgebeelde klassendiagram.



Figuur 9.3-7 Uiteindelijke klassendiagram

Database model diagram

Op basis van het zojuist besproken klassendiagram is er een database model ontworpen met behulp van MS Visio. Er is gekozen voor het gebruik van het database model diagram uit MS Visio omdat deze de mogelijkheid bied om het model automatisch te vertalen naar een MS Acces database. Dit resulteerde in het onderstaande database model diagram.



Figuur 9.3-8 Database model diagram uit MS Visio

9.4 Ontwikkelen van het Executable Prototype

Op basis van de bijgewerkte kritieke use cases is het Executable Prototype ontwikkeld. In eerste instantie is er tijdens het opstellen van de Development Case in de Inception fase een planning gemaakt waarbij de functionaliteiten verdeeld waren over iteraties. Omdat de kritieke use cases zijn veranderd is de indeling van de iteraties er anders uit komen te zien. De iteraties zijn als volgt ingedeeld.

Eerste iteratie:

- Kritieke use case nr. 1: Sign in
- Kritieke use case nr. 2: Start a topic
- Kritieke use case nr. 3: View topic

Tweede iteratie:

- Kritieke use case nr. 4: Chang topic status
- Kritieke use case nr. 5: Add remark
- Kritieke use case nr. 6: Generate requirement
- Kritieke use case nr. 7: Change requirement status

Volgens de bovenstaande volgorde is het Executable Prototype ontwikkeld behalve de kritieke use cases nr. 4 en nr. 5. Er is besloten om eerst de kritieke use case nr. 5 te ontwikkelen en vervolgens kritieke use case nr. 4. Dit is gedaan omdat de kritieke use case nr. 5 ‘Add remark’ vermoedelijk eenvoudiger was te ontwikkelen, waarna de kritieke use case nr. 4 ‘Change topic status’ vrij gemakkelijk te ontwikkelen is omdat deze, op wat aanvullingen na, bijna hetzelfde is.

9.4.1 Eerste iteratie

Sign in

Het eerste dat is gedaan is het invoeren van testgegevens in de database. Voor de use case ‘Sign in’ was dit het invullen van de tabel ‘Users’. Na het invullen is er uitzocht hoe er door middel van ASP.NET gegevens uit een tabel kunnen worden gehaald en hoe deze kunnen worden vergeleken met door de gebruiker ingevoerde gegevens. Nadat dat gedaan was is er begonnen met het ontwikkelen van een ‘Sign in’ scherm.

Door middel van een ‘if…else’ statement controleert het systeem controleren of de gebruiker toegang heeft tot het systeem. Dit wordt in de onderstaande figuur afgebeeld.

```
If SqlResult1 = ""
        Message.Text = "The username you have entered is unknown,<br>please try again..."
Else
        If SqlResult1 = password.Text
                Adapter.SelectCommand = new OleDbCommand(SqlStatement2, Connect)
                Adapter.SelectCommand.Connection.Open
                        SqlResult2 = Adapter.SelectCommand.ExecuteScalar()
                Adapter.SelectCommand.Connection.Close
                If SQLresult2 = 1
                        Adapter.SelectCommand = new OleDbCommand(SqlStatement3, Connect)
                        Adapter.SelectCommand.Connection.Open
                                SqlResult3 = Adapter.SelectCommand.ExecuteScalar()
                        Adapter.SelectCommand.Connection.Close
                        Session("UserSignedIn") = SqlResult3
                        Response.Redirect("prototypelist.aspx")
                Else
                        Message.Text = "This useraccount is inactive at this moment, please<br>try later or contact the
                        administrator..."
                End if
        Else
                Message.Text = "The username and password combination you<br>entered is not valid, please try again..."
        End if
End if
```

Figuur 9.4-1 ‘if…else’ statement om te bepalen of gebruiker toegang heeft

Als de gebruiker geen toegang krijgt tot het systeem zal hij/zij daar een melding van te zien krijgen. Heeft de gebruiker wel toegang tot het systeem, dan wordt er een sessie object aangemaakt die als waarde de ‘id’ van de gebruiker heeft. Door middel van dat sessie object zal dan gedurende de gehele applicatie te achterhallen zijn welke gebruiker heeft ingelogd.

Start a topic

De volgende use case die zou worden ontwikkeld was ‘Start a topic’. Deze use case zou worden uitgevoerd als de gebruiker is ingelogd, een software prototype systeem heeft geselecteerd, vervolgens een module heeft geselecteerd en aangeeft een nieuw topic te willen starten. Om deze reden zijn de database tabellen ‘Prototype’ en ‘Module’ ook met testgegevens ingevuld. Als de gebruiker succesvol heeft ingelogd is er voor gezorgd dat het systeem een lijst met beschikbare software prototypen laat zien en dat de gebruiker er daar één uit kan kiezen. Als de gebruiker daar op klikt laat het systeem de modules zien die bij het geselecteerde software prototype systeem hoort. Ook daar kan de gebruiker uit kiezen door er op een te klikken. Deze functionaliteiten zijn use cases die pas in een later stadium ontwikkeld zouden worden. Dit is gedaan om het verloop van Executable Prototype compleet te maken zodat de opdrachtgevers aan het eind van de Elaboration fase een zo duidelijk mogelijk beeld zouden krijgen van het systeem. Ook is dit gedaan om vaardigheden, zoals lijsten te tonen en de gebruiker de mogelijkheid gegeven om er één uit te kiezen in ASP.NET, zaken zijn die later ook zeker gebruikt zouden gaan worden. In die zin zou het dus zeker geen verspilde tijd zijn.

Bij de use case ‘Start a topic’ ging het om het wegschrijven van gegevens naar de database. Er is een scherm ontworpen waarin de gebruiker gegevens over het topic dat hij/zij wil gaan plaatsen kan invoeren. Vervolgens heeft hij uitgezocht hoe hij met behulp van ASP.NET die gegevens kan laten wegschrijven naar de database. Het wegschrijven van gegevens naar de database bleek vrij eenvoudig met ASP.NET. Dit zag er als volgt uit.

```
SelectStatement = "Select * From Topic"

Adapter.SelectCommand = new OleDbCommand(SelectStatement, Connect)
TopicCB = New OleDbCommandBuilder(Adapter)
Adapter.Fill(TopicDS,"Topic")

Row = TopicDS.Tables("Topic").NewRow
Row.Item("title") = TopicTitle.Text
Row.Item("topic_type") = TopicType.SelectedItem.Value
Row.Item("description") = TopicDescription.Text
Row.Item("module_id") = Session("SelectedModule")
Row.Item("version_prototype") = Version.Text
Row.Item("operating_system") = OperatingSystem.Text
Row.Item("user_id") = Session("UserSignedIn")
Row.Item("date_posted") = Now
Row.Item("topic_status") = 1
TopicDS.Tables("Topic").Rows.Add(Row)

Adapter.Update(TopicDS, "Topic")
```

Figuur 9.4-2 Invoeren van gegevens in de database

Bepaalde waarden zijn niet door de gebruiker ingevoerd maar zijn op een andere manier verkregen. Dit zijn de datum, de auteur van het topic en de status van het topic. De datum wordt verkregen door middel van de functie ‘Now’ in ASP.NET. Deze geeft de huidige datum en tijd door. De auteur van het topic wordt verkregen door het Sessie Object dat de ‘id’ bevat van de ingelogde gebruiker aan te roepen. De status van een net geplaatst topic is altijd ‘Open’ en is dus een vaste waarde.

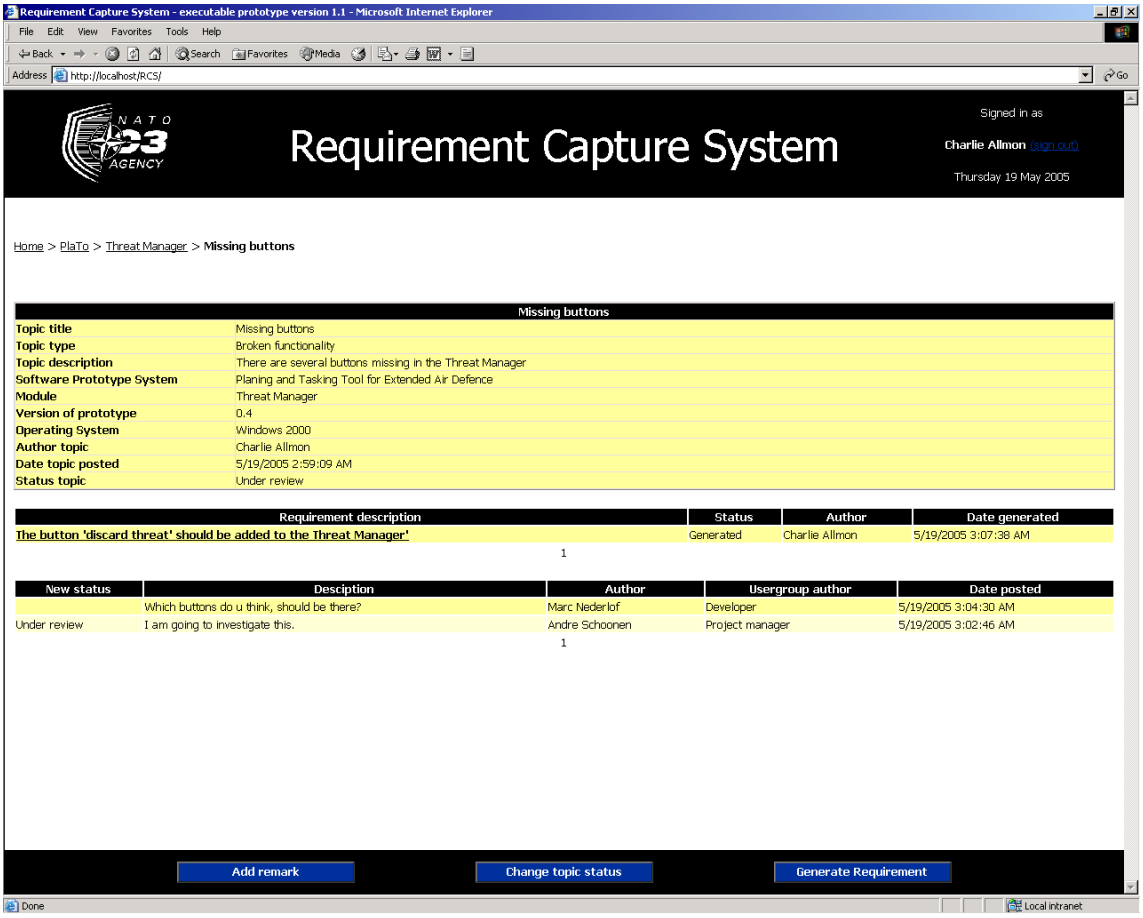
View topic

Bij de use case ‘View topic’ ging het erom simpelweg informatie uit de database te tonen aan de gebruiker. Omdat er tijdens het ontwikkelen van de use ‘Start a topic’ al mee bezig is geweest ging dit vrij makkelijk. ASP.NET maakt gebruik van zogenaamde ‘DataGrids’. Met behulp van ASP.NET kunnen alle gegevens, die opgehaald zijn door middel van een SQL Query, in een DataSet worden geplaatst. Deze DataSet kan vervolgens aan een DataGrid worden gekoppeld. Een voorbeeld van een van de DataGrids die ik heb gebruikt wordt hieronder weergegeven.

```
<asp:DataGrid
        id="RequirementGrid"
        runat="server"
        width="100%" borderwidth="0"
        cellspacing="2"
        autogeneratecolumns="false" />
```

Figuur 9.4-3 Voorbeeld van een DataGrid

Uiteindelijk kwam het scherm waarin alle informatie over een topic wordt weergegeven er als volgt uit te zien.



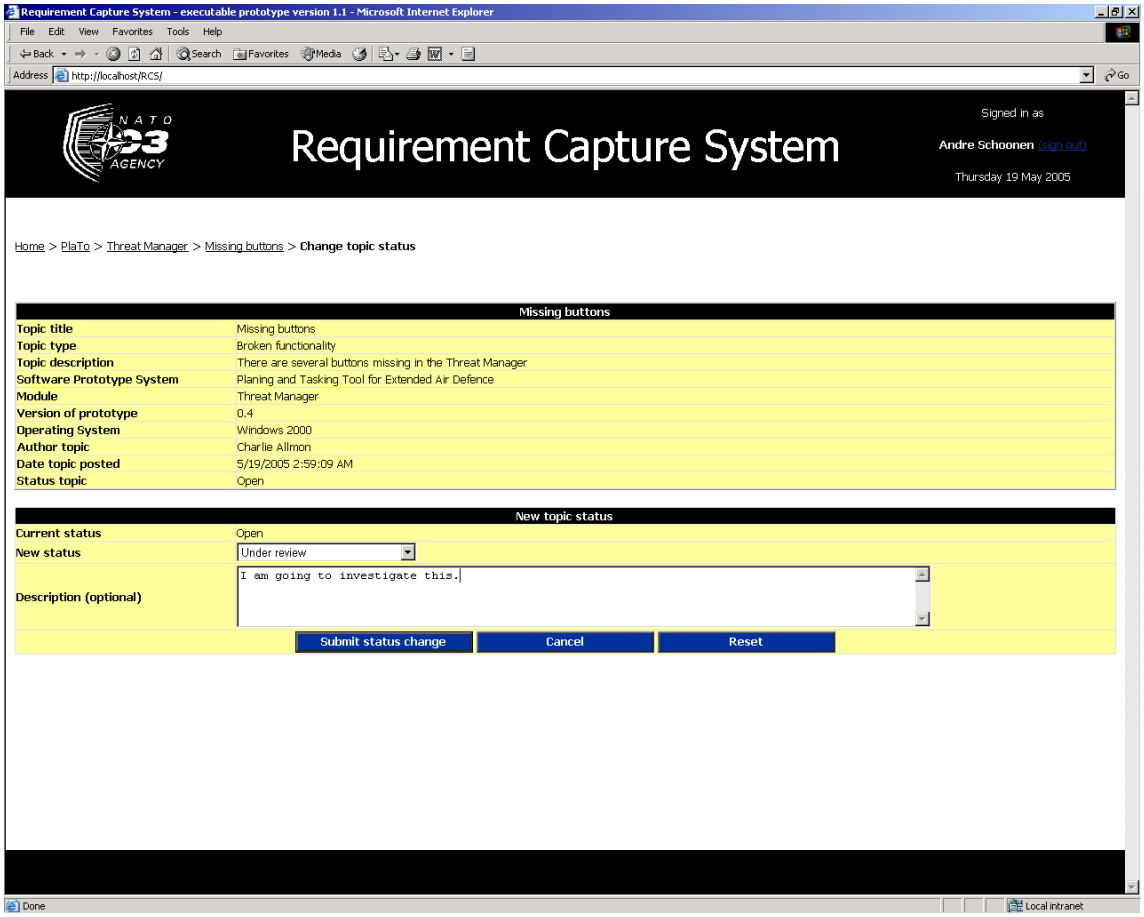
Figuur 9.4-4 Screenhot van het scherm Topic View

In dit scherm zijn drie kolommen te zien. De bovenste kolom bevat gegevens over het topic, de middelste kolom laat alle requirements zien die naar aanleiding van het topic zijn gegenereerd en de onderste kolom laat alle ‘Remarks’ en statuswijzigingen die gemaakt zijn zien. Voor deze kolommen heb ik drie DataGrids gebruikt.

9.4.2 Tweede iteratie

Omdat er in de eerste iteratie van het ontwikkelen van het Executable Prototype is ontdekt hoe gegevens worden opgevraagd, kunnen worden getoond en kunnen worden weggeschreven verliep het ontwikkelen van de use cases in de tweede iteratie een stuk sneller en gemakkelijker. Op het scherm waarin de gebruiker alle informatie over een topic kan bekijken zijn onder aan drie buttons te vinden te weten ‘Add remark’, ‘Change topic status’ en ‘Generate Requirement’. Door middel van die buttons kunnen de gelijknamige use cases worden uitgevoerd. In principe lijken deze use cases vrij veel op elkaar. De gebruiker kan door middel van een nieuw tekstveld een ‘Remark’ toevoegen of een ‘Requirement’ genereren’ en deze weglaten schrijven naar de database.

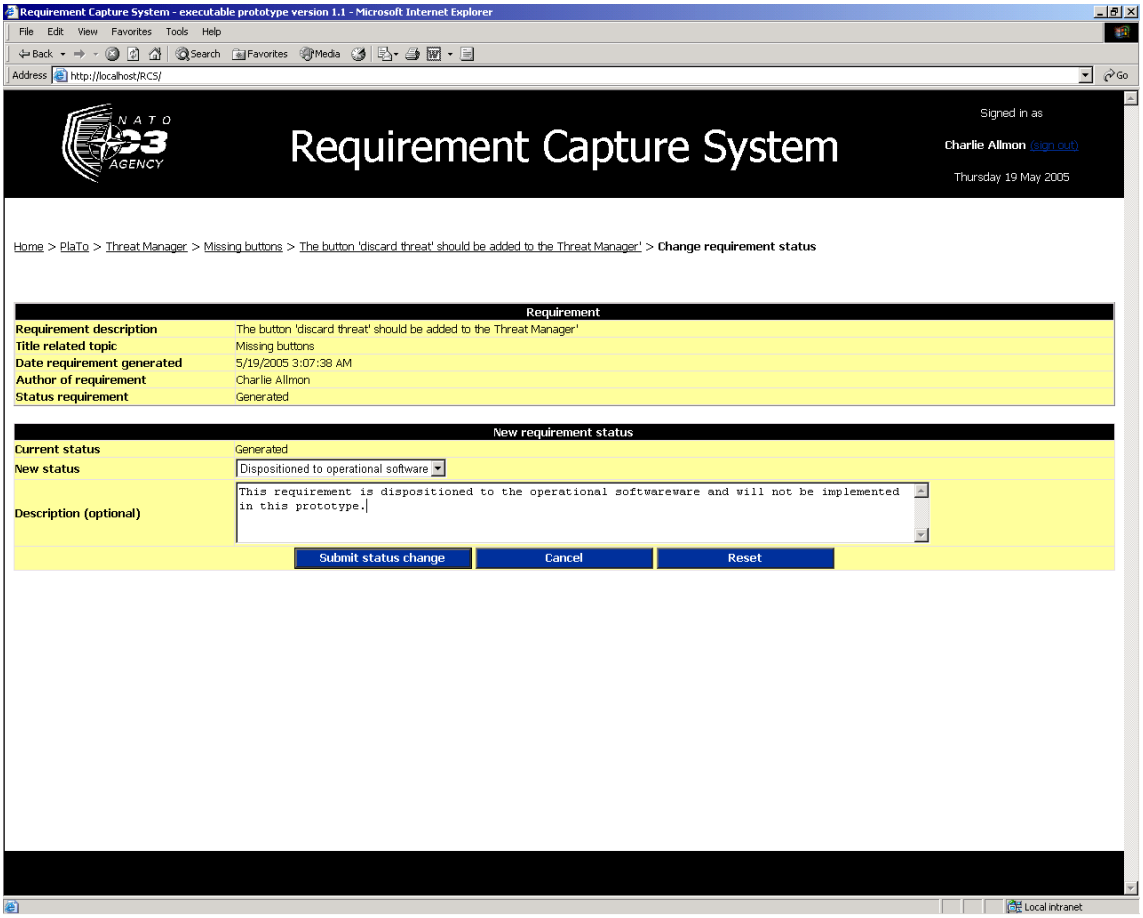
Voor het uitvoeren van een status wijziging van het topic kan de gebruiker kiezen uit een dropdownlist. De gebruiker heeft ook de mogelijkheid wanneer nodig tekst toe te voegen aan de status wijziging. In de onderstaande figuur wordt een voorbeeld gegeven van het uitvoeren van een status wijziging.



Figuur 9.4-5 Status van een topic wijzigen

Voor het wijzigen van de status van een requirement is een nieuw scherm ontwikkeld. De requirements die gegenereerd zijn door de gebruiker worden getoond in een DataGrid in het scherm waarin alle informatie over een topic te vinden is (tweede kolom). De gebruiker kan naar een specifiek requirement gaan door erop te klikken. In overleg met de opdrachtgevers is er voor gekozen om in dit nieuwe scherm een button te maken waardoor de gebruiker de status van dat requirement kan wijzigen. Het nieuwe scherm waar de gebruiker alle informatie over het requirement kan vinden behoort eigenlijk tot de use case ‘View requirement’ die niet was ingepland in deze fase. Toch leek het nodig om deze gedeeltelijk te ontwikkelen voor het verloop van het systeem.

De gebruiker kan op dezelfde de manier als de use case ‘Change topic status’ de status van het requirement veranderen. Dit is in de onderstaande figuur weergegeven.



Figuur 9.4-6 Status van een requirement wijzigen

9.4.3 Evalueren van het Executable Prototype

Met de opdrachtgevers en andere toekomstige gebruikers is het Executable Prototype geëvalueerd. Dit is op dezelfde manier gedaan als waarop de Conceptual Prototypes zijn geëvalueerd. Met behulp van een beamer is het prototype gepresenteerd. Aan de hand van de kritieke use cases is het hele prototype doorlopen en besproken. Op deze manier was er tijdens het doorlopen en bespreken van het Executable Prototype ruimte voor vragen en discussies. Tijdens de evaluatie is veel en vooral nuttige feedback ontvangen.

9.5 Aanpassen van de Development Case

De Development Case is aangepast omdat de totale ontwikkelomgeving nu duidelijk is geworden. Ook is er in de Development Case een aangepaste planning opgenomen. De globale versie van die planning wordt hieronder weergegeven.

Week	Calendar week	Phase	Activity	
12	17 (25-4 t/m 29-4)	Construction (iteration 1)	Refine Vision Document (1 days) Design use cases (4 days)	
13	18 (2-5 t/m 6-5)	Construction (iteration 1)	Build Beta Release (6 days) Test Beta Release (2 days) Refine Business Case (1 day) Refine Development Case (1 day)	
14	19 (9-5 t/m 13-5)	Construction (iteration 1)		
15	20 (16-5 t/m 20-5)	Construction (iteration 2)	Refine Vision Document (1 days) Redesign use cases (4 days)	
16	21 (23-5 t/m 27-5)	Construction (iteration 2) / End	Rebuild Beta Release (5 days)	Write final internship report (15 days)
17	22 (30-5 t/m 3-6)	Construction (iteration 2) / End	Test Beta Release (1 day) Refine Business Case (1 day) Refine Development Case (1 day) Write Initial Operational Capability Milestone report (2 days)	
18	23 (6-6 t/m 10-6)	End		

Figuur 9.5-1 Aangepast globale planning

Tijdens het aanpassen van de planning is in gedachte gehouden dat activiteiten zoals het bijwerken van het Vision Document, het bijwerken van de Business Case en het bijwerken van de Development Case misschien niet plaatst vinden omdat daar geen veranderingen in komen. Toch zijn deze wel meegenomen in de planning zodat er een ruime planning gemaakt kon worden. De iteraties zijn als volgt ingedeeld.

Iteratie 1:

- Sign out.
- View name user signed in.
- Select software prototype system
- Select a module
- Search for topics
- View topics user is interested in
- View topics user is involved in
- Mark topic as interesting
- Unmark topic as interesting

Iteratie 2:

- View requirements generated
- Generate requirement report
- Modify own user profile

Iteratie 3:

- Change user status
- Add user
- Modify user profile
- Modify software prototype system
- Add software prototype system
- Delete software prototype system
- Delete user
- Delete remark
- Delete requirement

Het indelen van deze use cases in iteraties is gedaan op basis van de eerder vastgestelde prioriteit van de use cases. De use cases, die zijn ingedeeld in de eerste iteratie, bevatten functionaliteiten die als basis dienen voor het systeem, terwijl de use cases in de tweede iteratie meer een soort luxe zijn. In de derde iteratie zijn alle administrator specifieke use cases ingedeeld. In overleg met de opdrachtgevers is besloten om de derde iteratie niet uit te voeren tijdens het afstudeertraject. Deze use cases kunnen eventueel handmatig worden uitgevoerd in de database omgeving waardoor ze minder cruciaal zijn voor de Beta Release.

Hoofdstuk 10 - Construction fase

In dit hoofdstuk worden de uitgevoerde werkzaamheden tijdens, en de opgeleverde producten uit de Construction fase besproken. De Construction fase is de derde fase die in RUP omschreven wordt. Tevens is de Construction fase de laatste fase die tijdens dit afstudeertraject zal worden uitgevoerd. In deze fase is het Software Architecture Document bijgewerkt op basis van nieuwe of bijgewerkte ontwerpen en de Beta Release van het systeem is ontwikkeld. Uiteindelijk is dit alles verwerkt in een rapport de 'Initial Operational Capability Milestone'. Dit rapport is ook als externe bijlage toegevoegd aan dit afstudeerverslag. Net als de Beta Release is dit rapport een mijlpaal uit de Construction fase. Omdat dit de laatste fase van het afstudeertraject zal zijn is er geen Development Case met aangepaste planning opgesteld.

10.1 Bijwerken van het Software Architecture Document

Tijdens de Construction fase zijn de niet-kritieke use cases verder uitgewerkt en zijn er een aantal aanvullingen aan het database ontwerp gedaan. Deze aanpassingen en uitbreidingen worden in deze paragraaf beschreven.

10.1.1 Uitwerken van de niet-kritieke use cases

Aan de hand van gesprekken met de opdrachtgevers zijn de niet-kritieke use cases verder uitgewerkt. Nadat deze waren uitgewerkt zijn ze besproken met de opdrachtgevers en waarnodig aangepast. Enkele voorbeelden van deze uitgewerkte niet-kritieke use cases worden hieronder weergegeven.

Use case No. 10	
Name	View requirements generated.
Assumption	User has signed in and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to view the requirements that are generated for the selected software prototype system. (2) The system shows the user all the requirements that are generated for the selected software prototype system.
Exceptions	[No Requirements found.] The system will tell the user that there are no requirements generated for the selected software prototype system.
Result	Generated Requirements are displayed.

Use case No. 11	
Name	Generate requirement report.
Assumption	User has signed in as Project manager or Administrator and has selected a software prototype system.
Actors	Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to generate a Requirement Report for the selected software prototype system. (2) The system will generate a Requirement Report.
Exceptions	[There are no requirements generated for the selected software prototype system.] The system will tell the user that there are no requirements generated for the selected software prototype system and therefore it is not possible to generate a Requirement Report.
Result	Requirement Report is generated.

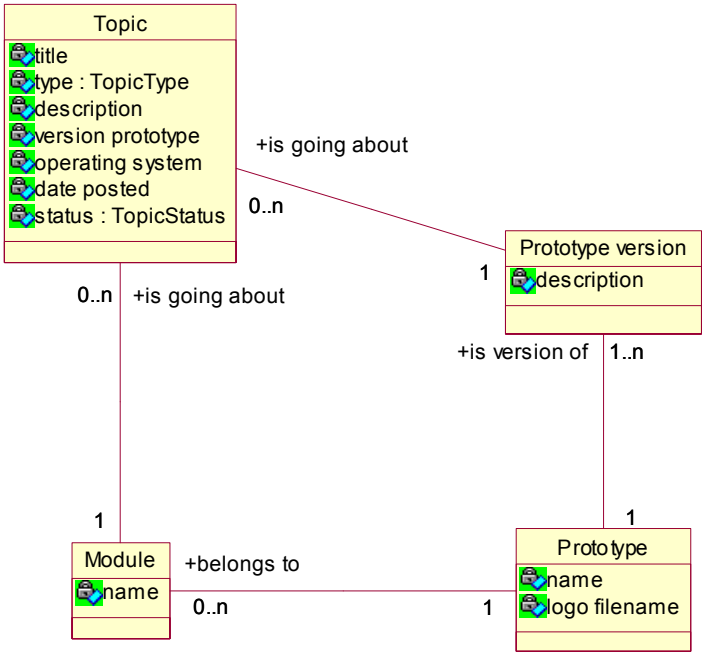
Figuur 10.1-1 Voorbeeld van twee niet-kritieke use cases

10.1.2 Aanpassen van het database ontwerp

Het database ontwerp moest aanpast worden vanwege evaluatieresultaten uit de Elaboration fase en een aantal nieuwe use cases. Er is voor gekozen om terug te gaan naar het klassendiagram en deze aan te passen en vervolgens dit te vertalen naar het database model diagram.

Klassendiagram

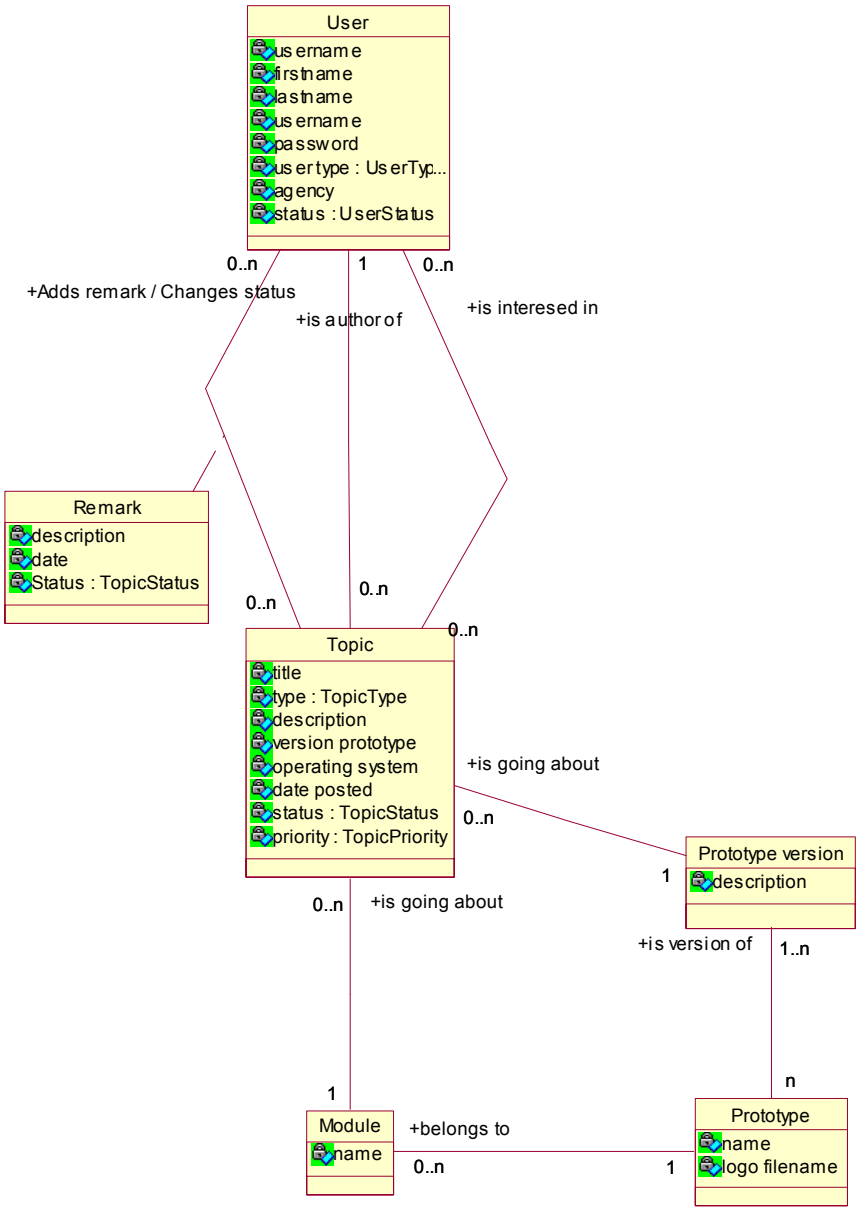
Tijdens het evalueren van de kritieke use case 'Start a topic' is gebleken dat van een prototype meerdere versies bekend moeten zijn. Als de gebruiker de use case 'Start a topic' uitvoert moet hij/zij kunnen aangeven om welke versie van het door hem/haar eerder geselecteerde prototype het gaat. De opdrachtgevers hebben tijdens die evaluatie aangegeven dat ze willen dat de gebruiker dit kan doen door de versie te selecteren uit een dropdownlist. Om dit te realiseren was er een aanpassing nodig aan het database ontwerp. De aanpassing die hiervoor gemaakt is in het klassendiagram wordt hieronder weergegeven. Voor de overzichtelijkheid worden alleen de relevante klassen weergegeven.



Figuur 10.1-2 Aanpassingen in klassendiagram

Er is een nieuwe klasse toegevoegd genaamd 'Prototype version'. Deze klasse heeft slechts één attribuut dat de versie van het prototype omschrijft. Deze klasse heeft een 'n op 1' relatie met de klasse Prototype. Deze zorgt ervoor dat een prototype meerdere versies kan hebben en dat een prototype versie bij een prototype hoort. Op die manier kunnen vanuit de klasse Module de verschillende versies worden opgevraagd. Verder heeft de klasse 'Prototype version' een '1 op n' relatie met de klasse 'Topic'. Op deze manier kan worden vastgelegd over welke versie van het prototype het topic gaat.

De niet-kritieke use cases ‘View topics user is interested in’, ‘Mark topic as interesting’ en ‘Unmark topic as interesting’ moesten de gebruiker de mogelijkheid geven aan te geven welke topics hij/zij interessant vindt en hij/zij moet deze lijst ook kunnen opvragen. Hiervoor is er een ‘n op n’ associatie tussen de klassen ‘Topic’ en ‘User’ toegevoegd. Dit wordt in de onderstaande figuur weergegeven.

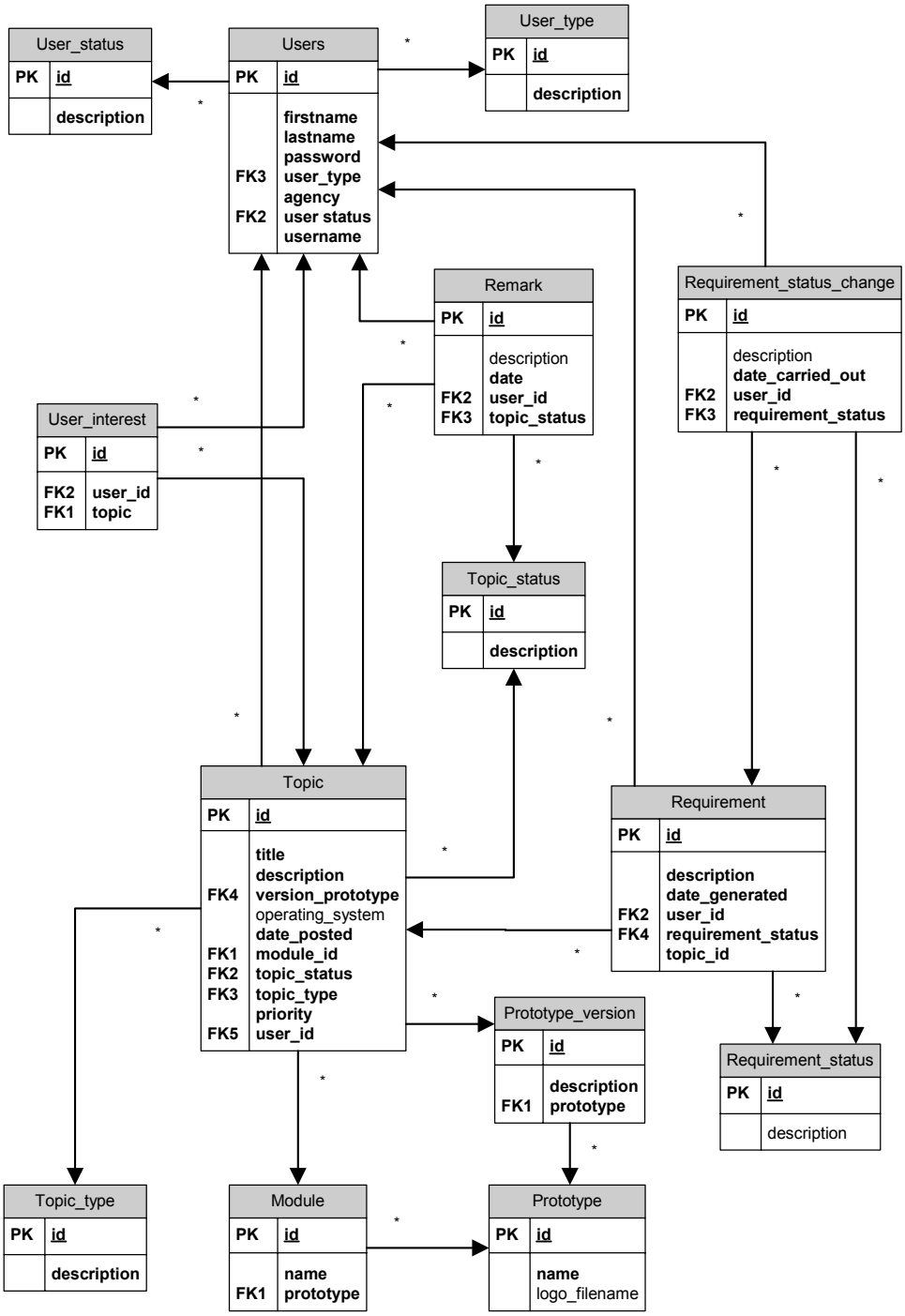


Figuur 10.1-3 Aanpassingen aan het klassendiagram

Aan deze ‘n op n’ associatie is geen associatieklasse gekoppeld omdat er voor deze associatie geen attributen bestaan en ook geen associatie met andere klassen heeft.

Database model diagram

Het aangepaste klassendiagram is vervolgens vertaald naar het database model diagram dat hieronder wordt weergegeven. Met behulp van dit database model diagram heeft hij de MS Access database bijgewerkt.



Figuur 10.1-4 Database model diagram

10.2 Ontwikkelen van de Beta Release

10.2.1 Iteratie 0

Het aanpassen van het systeem naar aanleiding van de evaluatieresultaten uit de Elaboration fase zouden naar verwachting niet al te veel tijd in beslag nemen. Daarom is er besloten deze uit te voeren voordat er aan de oorspronkelijke geplande eerste iteratie van de Construction fase zou worden begonnen en heeft werd 'Iteratie 0' genoemd. Uiteindelijk heeft dit in totaal twee dagen gekost.

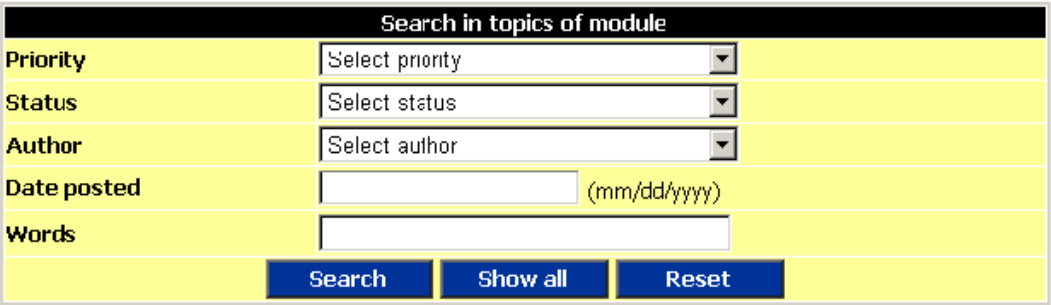
10.2.2 Eerste iteratie

Een aantal van de use cases die waren ingepland waren gedeeltelijk al ontwikkeld tijdens het ontwikkelen van het Executable Prototype. Hierdoor heeft het ontwikkelen van die use cases niet veel tijd gekost. Het betreft de volgende use cases.

- Sign out
- View name user signed in
- Select software prototype system
- Select a module

Search for topics

Het tweede Conceptual Prototypes bevatte de layout van een search engine. Deze search engine zijn als basis gebruikt en besproken met de opdrachtgevers. Op basis van deze gesprekken zijn de layout, voornamelijk de velden waarop gezocht kan worden, veranderd. In de onderstaande figuur is deze search engine te zien. Deze search engine is toevoegd in het scherm waar een overzicht van verschillende topics, die betrekking hebben tot een bepaalde module, wordt weergegeven.



Figuur 10.2-1 Search engine

Het was de bedoeling dat er ook op meerdere zoekcriteria tegelijkertijd gezocht kon worden. Dit zorgde voor een ingewikkeld 'if…else' statement om op die manier het SQL statement op te bouwen.

View topics user is interested in

De bedoeling van deze use case is dat de gebruiker een lijst kan opvragen van de verschillende topics die hij/zij interessant vindt. Dit kan gemakkelijk worden uitgelezen uit de database. Als de gebruiker aan gegeven heeft een topic interessant te vinden, wordt er een nieuwe regel toegevoegd aan de tussentabel 'User_interest'. Deze regel bevat de 'id' van de gebruiker en de 'id' van het topic. Dus om alle topics op te vragen die de gebruiker interessant vindt, hoeft het systeem alleen het sessie object aan te roepen om de 'id' van de ingelogde gebruiker te verkrijgen om vervolgens alle rijen te tonen waarbij deze 'id' voorkomt in de 'User_interest' tabel.

Om te testen of het inderdaad werkt zijn in eerste instantie de tabel 'User_interest' uit database gevuld met testgegevens. Toen dit werkte zijn de use cases 'Mark topic as interesting' en 'Unmark topic as interesting' ontwikkeld. Door middel van een button kan de gebruiker aangeven of hij/zij het topic interessant vindt. Vindt de gebruiker het topic interessant, dan wordt er een regel toegevoegd aan de tabel 'User interest' die de 'id' van de ingelogde gebruiker en de 'id' van het topic die de gebruiker aan het bekijken is. Als de gebruiker een topic niet meer interessant vindt, dan wordt de regel waarbij de 'id' van de ingelogde gebruiker en de 'id' van het topic verwijderd uit de database.

View topics user is involved in

Deze use case moet het mogelijk maken om de gebruiker een lijst te tonen met topics waar hij/zij bij betrokken is. In overleg met de opdrachtgevers is het volgende bepaald. Een gebruiker is betrokken bij een topic als deze:

- de auteur van het topic is
- een remark heeft toegevoegd aan het topic
- de status van het topic heeft gewijzigd
- op basis van het topic een requirement heeft gegenereerd
- de status van een requirement, wat is gegenereerd op basis van het topic, heeft gewijzigd.

Om de juiste informatie uit de database te halen was een vrij ingewikkeld SQL statement nodig. Met behulp van MS Access is dit statement opgesteld. Dit wordt hieronder weergegeven.

```
Dim SQLstatement As String = "SELECT Topic.id, Topic.title, Topic.priority, Topic.date_posted, Topic_type.description,
                    Topic_status.description, Users.firstname & ' ' & Users.lastname AS Author, Max(Remark.date_posted) AS
                    MaxOfdate_posted, Modules.name" & _
          " FROM (((Users INNER JOIN ((Topic_type INNER JOIN (Topic_status INNER JOIN (Modules INNER JOIN Topic ON
                    Modules.id = Topic.module_id) ON Topic_status.id = Topic.topic_status) ON Topic_type.id = Topic.topic_type)
                    INNER JOIN (Remark INNER JOIN Users AS Users_1 ON Remark.user_id = Users_1.id) ON Topic.id =
                    Remark.topic_id) ON Users.id = Topic.user_id) LEFT JOIN Requirement ON Topic.id = Requirement.topic)
                    LEFT JOIN Requirement_status_change ON Requirement.id = Requirement_status_change.requirement)
                    LEFT JOIN Users AS Users_2 ON Requirement_status_change.user_id = Users_2.id" & _
          " WHERE Users_1.id=" & Session("UserSignedIn") & " OR Users_2.id=" & Session("UserSignedIn") & _
          " GROUP BY Topic.id, Topic.title, Topic.priority, Topic.date_posted, Topic_type.description, Topic_status.description,
                    Users.firstname & ' ' & Users.lastname, Modules.name, Modules.prototype" & _
          " HAVING Modules.prototype=" & Session("SelectedPrototype") & _
          " ORDER BY Topic.date_posted DESC"
```

Figuur 10.2-2 SQL statement

10.2.3 Tweede iteratie

View requirements generated

De opdrachtgevers hebben tijdens de evaluatie van het Executable Prototype aangegeven dat het gewenst is om per software prototype systeem een lijst te kunnen verkrijgen waarop alle gegenereerde requirements worden getoond. Deze functie zijn als volgt ontwikkeld. Nadat de gebruiker een software prototype heeft geselecteerd verschijnt er onderaan op het scherm een button 'View requirements'. Als de gebruiker op deze button klikt zal hij naar een nieuw scherm gaan waarop een lijst wordt getoond met alle requirements die voor dat software prototype systeem zijn gegenereerd. Door middel van de resultaten van het juiste SQL statement te stoppen in een 'DataSet', zijn deze 'DataSet' gekoppeld aan een 'DataGrid' dat wordt getoond op het scherm.

Generate requirement report

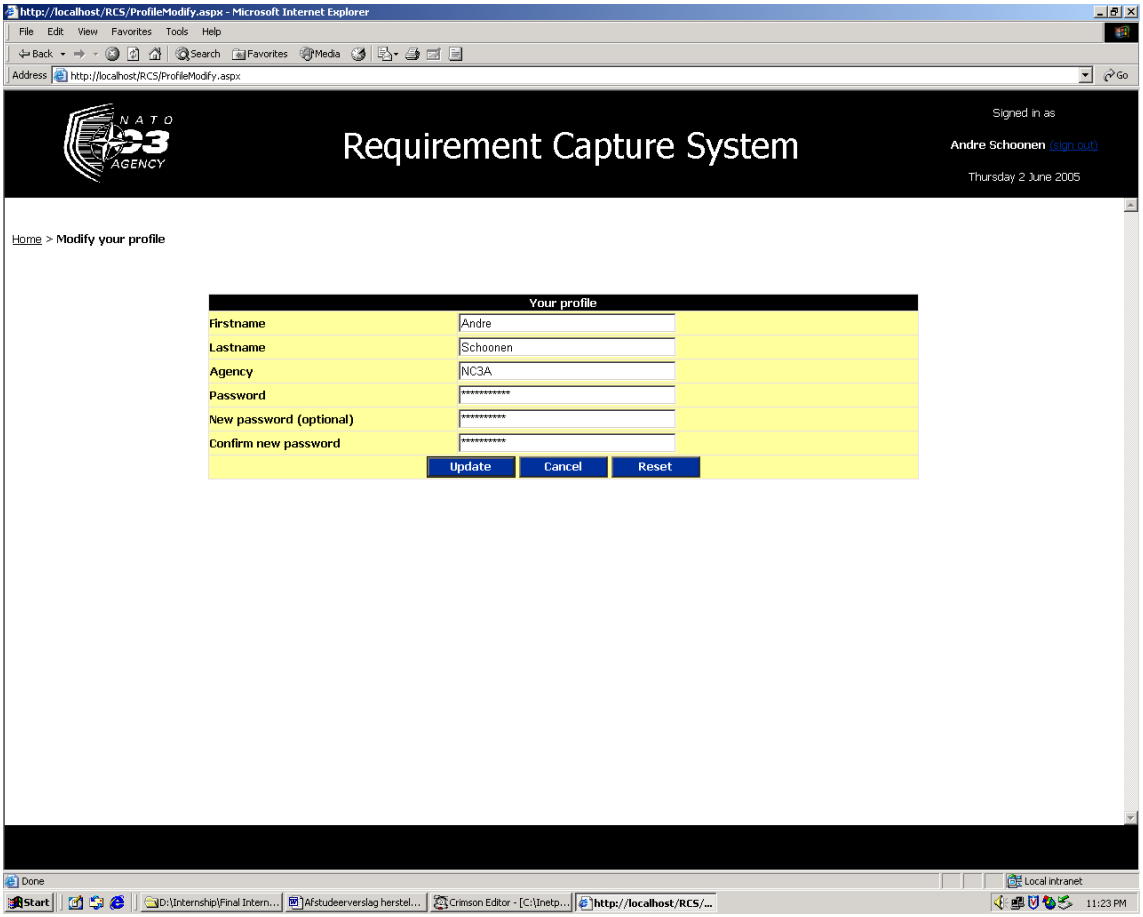
De opdrachtgevers hebben ook aangegeven dat de lijst met requirements per software prototype, moet kunnen worden geëxporteerd naar een Excel bestand. Het uitzoeken van mogelijkheden hiervoor in ASP.NET heeft meer tijd gekost dan dat er eerste instantie werd gedacht. Dit kwam omdat er in de literatuur die hij tot zijn beschikking had niets over deze functionaliteit beschreven werd. Er is uitgebreid op het Internet naar informatie hierover gezocht en daar is de oplossing gevonden. Om bestanden te kunnen exporteren moesten Office Web Components (OWC) geïnstalleerd worden. Toen hij dit geïnstalleerd had kon hij door middel van OWC een script schrijven wat ervoor zorgde dat resultaten van een SQL statement worden opgeslagen in een Excel bestand. Een gedeelte uit de hiervoor benodigde code wordt hieronder afgebeeld.

```
'Add the data to the excel sheet'
    For Each myColumn In myTable.Columns
        ColumnCount += 1
        RowCount = 1
        For Each myRow In myTable.Rows
            RowCount += 1
            objExcel.ActiveSheet.Cells(RowCount, ColumnCount) = CStr(myRow(myColumn))
        Next
    Next
    ColumnCount = 0

'Autofit the columns to make them look pretty'
    For Each myColumn In myTable.Columns
        ColumnCount += 1
        objExcel.ActiveSheet.Columns(ColumnCount).EntireColumn.AutoFitColumns()
    Next

'This saves the excel file to C:\Example.xls on the server'
    objExcel.ActiveSheet.Export(FileAndPath, OWC.SheetExportActionEnum.ssExportActionNone)

'This opens the download dialoge box and allows the user'
'to download the excel sheet from the server'
    Response.Clear()
    Response.ContentType = "application/octet-stream"
    Response.AddHeader("Content-Disposition", "attachment; filename=" & FileName)
    Response.Flush()

'This command actually transfers the file'
    Response.WriteFile(FileAndPath)
```

Figuur 10.2-3 Een Excel bestand genereren met ASP.NET

Modify own user profile

Als laatste is er een scherm ontwikkeld waarin de gebruiker zijn/haar eigen profiel kan wijzigen. Uit veiligheidsoverwegingen zal de gebruiker wel zijn/haar password nogmaals moeten invoeren om dit te bevestigen. Tevens kan de gebruiker in dit scherm zijn/haar password wijzigen. In de onderstaande figuur wordt het scherm afgebeeld.



Figuur 10.2-4 Wijzigen van het eigen profiel

Zoals te zien is in het bovenstaande figuur vindt de wijziging van het profiel samen met het wijzigen van het password plaats. Omdat dit voor sommige gebruikers misschien niet overzichtelijk genoeg is er overwogen om deze wijzigingen apart plaatst te laten vinden. Bijvoorbeeld in verschillende tabellen of zelfs verschillende schermen. Op het moment dat hierover werd getwijfeld was het bovenstaande scherm al ontwikkeld, waardoor er besloten is om het zo te laten. Als het systeem in gebruik wordt genomen zal duidelijk moeten worden of deze aanpassing wel of niet gewenst is.

10.2.4 Evalueren van de Beta Release

Omdat de derde iteratie niet binnen het afstudeertraject plaats zal vinden is er besloten om de eerste twee iteraties met de opdrachtgevers en toekomstige gebruikers te evalueren. Op dezelfde manier als dat het Conceptual Prototype en het Executable Prototype zijn geëvalueerd, heeft hij de Beta Release geëvalueerd. Aangezien het systeem niet verder ontwikkeld zal worden binnen het afstudeertraject had deze evaluatie enigszins een andere doelstelling. Het doel van deze evaluatie was om veranderingen en toevoegingen aan het huidige systeem te bepalen en vast te leggen.

10.3 Nog uit te voeren werkzaamheden

De Beta Release zoals die tot nu tot ontwikkeld was, was nog geen volwaardige Beta Release. De functionaliteiten die ontbraken aan de Beta Release zijn in de derde iteratie ingepland. Dit zijn de volgende use cases:

- Change user status
- Add user
- Modify user profile
- Modify software prototype system
- Add software prototype system
- Delete software prototype system
- Delete user
- Delete remark

Hiervan zijn de laatste drie use cases het minst belangrijk. Dit zijn use cases die de gebruiker de mogelijkheid geeft om informatie te verwijderen uit het systeem, wat hoogstwaarschijnlijk niet zal voorkomen, tenzij iemand per ongeluk verkeerde gegevens heeft ingevoerd. In principe zouden alle use cases die nog overblijven gemakkelijk kunnen worden ingevoerd in, gewijzigd in en verwijderd uit de database door middel van MS Access. Dit is de manier waarop ik testgegeven heb ingevoerd. De moeilijkheid is echter dat het systeem op een NATO UNCLASSIFIED server zal moeten komen te staan die mogelijk niet in het TMD Lab zelf komt te staan. In dat geval is het voor de gebruikers van het NATO C3 Agency lastiger om toegang te krijgen tot de database. Daarom is het ontwikkelen van de hierboven genoemde use cases nodig om het systeem een volwaardige Beta Release te noemen.

Er is tijdens de Construction fase geen Business Case en Development Case opgesteld. Dit is niet gedaan omdat deze voor het afstudeertraject niet interessant waren. De Business Case zou namelijk moeten aangeven of het voortzetten van het traject genoeg voordelen met zich mee zou brengen. Aangezien het afstudeertraject eindigt heeft het opzetten van een Business Case hiervoor geen zin. Als het ontwikkeltraject van het systeem wordt voortgezet op een andere manier, dan zal er uiteraard wel een Business Case moeten worden opgesteld. Om die zelfde rede is ook de Development Case niet opgesteld tijdens de Construction fase. Het zou geen zin hebben om dat moment een planning te maken voor het vervolg omdat op dat moment nog niet bekend is wanneer en of dit wel gaat plaatst vinden. Ook hiervoor geld dat als het traject later zal worden voortgezet dat er wel een Development Case zal moeten worden opgesteld.

Deel III - Evaluatie

Hoofdstuk 11 - Proces en productevaluatie

In dit hoofdstuk worden de opgeleverde producten en het gevolgde proces geëvalueerd.

11.1 Productevaluatie

11.1.1 Project Plan

Het Project Plan, beter bekend als het Plan van aanpak, heeft inzicht gegeven in de initiële opdrachtomschrijving en de geplande aanpak daarvan. Dit Project Plan is tijdens het uitvoeren van het ontwikkeltraject niet verder aangepast omdat daarvoor ruimte was in de op te leveren producten die door RUP werden voorgeschreven zoals het Vision Document, Business Case en de Development Case. Dit is als prettig ervaren omdat er daardoor heel gemakkelijk kon worden teruggekeken op het oorspronkelijke Project Plan.

11.1.2 Lifecycle Objective Milestone

In de Lifecycle Objective Milestone dat aan het einde van de Inception fase is opgeleverd, worden het Vision Document, het Software Architecture Document, het Conceptual Prototype, de Business Case en de Development Case beschreven. Ook zijn in dit document verschillende mogelijkheden van implementatie voor het te ontwikkelen systeem beschreven.

Het Vision Document bevatte gedeeltes uit de opdrachtomschrijving, maar ook een beschrijving van de target users, features en key use cases. Hierdoor heeft dit document geholpen om een globaal beeld te krijgen van het te ontwikkelen systeem. Het Software Architecture Document bevatte in dit stadium beschrijvingen van kritieke use cases en overige use cases. Dit gaf inzicht in de kern van het te ontwikkelen systeem en tevens ook een globaal overzicht van de niet kritieke onderdelen ervan. Ook is in deze fase een beschrijving gegeven van verschillende mogelijkheden om het systeem te implementeren. In deze fase was er echter nog geen oplossing gekozen. Achteraf gezien zouden deze mogelijkheden is deze fase meer onderzocht moeten worden en zou er ook een keuze gemaakt moeten worden. Omdat deze keuze later is gemaakt, heeft dit een grotere impact gehad op het ontwikkeltraject. Als deze keuze eerder gemaakt zou zijn, zou er daardoor beter op geanticipeerd kunnen worden. Het Conceptual Prototype is de mijlpaal uit de Inception fase. Deze zijn in twee iteraties ontwikkeld en hier is veel aandacht en tijd aan besteed. Hierdoor is er meer tijd aan de Inception fase besteedt dan dat er in eerste instantie was gedacht. Achteraf gezien hoefde het Conceptual Prototype niet zo uitgebreid en gedetailleerd te zijn omdat dat in een later stadium ook ontwikkeld zou kunnen worden. Het voordeel van het feit dat het Conceptual Prototype vrij uitgebreid was is dat de opdrachtgevers en andere toekomstige gebruikers daardoor veel en vooral duidelijke feedback konden leveren. De Business Case en de Development Case waren in principe aanpassingen van delen uit het oorspronkelijke Project Plan. De Development Case heeft zijn nut bewezen doordat daarin de aangepaste planning is meegenomen.

11.1.3 Lifecycle Architecture Milestone

In de Lifecycle Architecture Milestone dat aan het einde van de Elaboration fase is opgeleverd, worden het Vision Document, het Software Architecture Document, het Executable Prototype, de Business Case en de Development Case beschreven. Ook is in dit document de gekozen mogelijkheid van implementatie voor het te ontwikkelen systeem uitgebreider beschreven dan in de vorige fase.

In het Vision Document is de definitieve opdrachtomschrijving opgenomen. Ook zijn de features en key use cases aangepast. Op deze manier kon er op een heel globaal niveau een overzicht worden verkregen van de aangepaste functionaliteit van het systeem. Het Software Architecture Document bevatte in dit stadium een bijgewerkte beschrijving van kritieke use cases. De overige use cases zijn in deze fase niet verder gewijzigd. Deze use cases vormden de basis voor het Executable Prototype. Ook bevat het een ontwerp voor de database, wat gebruikt is voor de ontwikkeling van het Executable Prototype. In deze fase is de gekozen oplossing voor het implementeren van het systeem uitgewerkt en beschreven waardoor de opdrachtgever een goed beeld heeft kunnen krijgen van de manier waarop het systeem wordt ontwikkeld. Het Executable Prototype is ontwikkeld op basis van de kritieke use cases. Al de kritieke use cases zijn meegenomen in de ontwikkeling van het Executable Prototype. Tijdens de ontwikkeling van het Executable Prototype is ook veel aandacht besteed aan kleinere details. Dit heeft redelijk wat tijd in beslag genomen, maar heeft uiteindelijk tijd bespaard in de Construction fase. Aan de Business Case zijn in dit stadium geen aanpassingen gedaan waardoor het opnemen van de Business Case achteraf gezien niet echt nut heeft gehad. Aan de Development Case is wel het een en ander aangepast omdat het op dat moment duidelijk was geworden in welke ontwikkelomgeving het systeem ontwikkeld zou worden. Ook is in de Development Case wederom een aangepaste planning meegenomen.

11.1.4 Initial Operational Capability Milestone

In de Initial Operational Capability Milestone dat aan het einde van de Construction fase is opgeleverd, worden het Vision Document, het Software Architecture Document en de Beta Release beschreven. In deze fase zijn de Business Case en de Development Case niet opgesteld omdat de opdracht binnen het afstudeertraject niet verder zal worden uitgevoerd.

Het Software Architecture Document bevatte in deze fase de volledige beschrijvingen van de kritieke en niet-kritieke use cases. Tevens bevat het document wederom een aangepast database ontwerp op basis van de use cases. Op basis van het Software Architecure Document is de Beta Release ontwikkeld. Deze is helaas niet helemaal af omdat de use cases voor de ‘Administator’ actor nog niet zijn ontwikkeld.

11.2 Procesevaluatie

Als systeemontwikkelingmethode is het Rational Unified Process (RUP) gehanteerd. Deze methode heeft voor het uitvoeren van de afstudeeropdracht zowel voor en nadelen met zich meegebracht. Wat zeer goed bevallen is, is het feit dat RUP een zeer iteratieve methode is. Het systeem wat is ontwikkeld is daardoor als het ware geëvolueerd, beginnend als een Conceptual Prototype, tot de uiteindelijk opgeleverde Beta Release. Doordat er als het ware gebruik is gemaakt van prototyping konden de toekomstige gebruikers in een vroeg stadium al een goed beeld krijgen van het uiteindelijke systeem en feedback geven. RUP is een zeer uitgebreide systeemontwikkelingmethode die grote ontwikkelingstrajecten ondersteund. Het is echter ook mogelijk om RUP toe te passen op kleinere projecten. Het is dan wel noodzakelijk dat de juiste elementen uit RUP worden gebruikt en de juiste worden weggelaten. Aangezien de afstudeerder voor het uitvoeren van de afstudeeropdracht geen ervaring had met RUP, was het lastig om van tevoren te bepalen welke elementen uit RUP wel en welke niet gebruikt moesten worden.

De meeste tijd is besteed aan de Inception fase en de Elaboration fase. Dit komt omdat tijdens het uitvoeren van vooral de Inception fase veel zaken nog niet helder waren. Het leren werken met de ontwikkelomgeving heeft in dat opzicht voor vertraging gezorgd bij het uitvoeren van de Elaboration fase. Daarbij komt het feit dat er in de Inception en Elaboration fase veel tijd heeft besteed aan het Conceptual en Executable Prototype waardoor de het ontwikkelen van de Beta Release in de Construction fase veel gemakkelijk is verlopen en zelfs minder tijd in beslag heeft genomen dan van tevoren werd gedacht. Het is wel zo dat de Construction fase niet in zijn geheel is afgerond, maar de nog uit te voeren werkzaamheden zullen waarschijnlijk niet al te veel tijd in beslag gaan nemen. RUP schrijft echter een totaal andere tijdverdeling voor. Volgens RUP zou de Inception fase de minste tijd in beslag moeten nemen en de Construction fase het meeste. Tijdens het uitvoeren van het afstudeertraject is dit precies andersom gegaan. Dit had kunnen worden voorkomen door in het begin niet te veel tijd te besteden aan details maar genoegen nemen met de resultaten die op dat moment bereikt waren en gelijk door te stomen naar de volgende fase. Waarschijnlijk zou het uiteindelijke resultaat min of meer hetzelfde moeten zijn.

Al met al is het een zeer leerzame ervaring en leuke uitdaging geweest om RUP toe te passen op deze afstudeeropdracht. Als er een iteratieve aanpak gewenst is en gebruik wordt gemaakt van UML dan is het zeker aan te raden om RUP te overwegen als systeemontwikkelingmethode.

Deel IV - Bijlagen

Geraadpleegde literatuur

Boeken

Duthie, G. Andrew, 'Microsoft ASP.NET Step by Step'. Washington, USA, Microsoft Corporation, 2002

Evans, T., '10 Minute Guide to HTML 4.0, third edition'. Que Corporation, 1998

Hatfield, B., 'ASP.NET for Dummies'. Foster City, California, Hungry Minds, Inc., 2001

Kleppe, A. en Warmer, J., 'Praktisch UML, 3de editie'. Soest'. Pearson Education Benelux BV., 2004

Kroll, O. en Kruchten, P, 'The Rational Unified Process Made Easy: a practitioner's guide to the RUP'. Boston, Pearson Education, Inc., 2003

Kruchten, P., 'The Rational Unified Process: an introduction, third edition'. Boston, Pearson Education, Inc., 2004

Langfeld, L., Noel, M. en Spence, C., 'Microsoft SharePoint 2003 Unleashed'. USA, Sams Publishing, 2004

Phillips, L.A., 'Using HTML 4, fourth edition'. Que Corporation, 1998

Quatrani, T., 'Visual Modeling with Rational Rose 2004 and UML'. Boston, Pearson Education, Inc., 2003

Tolido R.J.H., 'IAD, Het evolutionair ontwikkelen van informatiesystemen'. Schoonhoven, Academic Service, 1997

Rapporten

Houten, van, J.J., 'Project Plan – Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Houten, van, J.J., 'Lifecycle Objective Milestone, Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Houten, van, J.J., 'Lifecycle Architecture Milestone, Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Houten, van, J.J., 'Initial Operational Capability Milestone, Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Schoonen, A., 'PlaTo Design Document – phase 1'. The Hague, NATO C3 Agency, 2004

Schoonen, A., 'PlaTo User Guide, version 0.890'. The Hague, NATO C3 Agency, 2004

Allmon, C. Bingham, T., Nederlof M. en Schoonen, A, 'LSID Usage in Joint Project Optic Windmill VIII'. The Hague, NATO C3 Agency, 2004

Internet

http://www.asp.net

http://www.aspnl.com

http://www.aspnet101.com

http://www.codeproject.com

http://www.datagridgirl.com

http://www.dotnet-guide.com

http://www.dotnet247.com

http://www.dotnetjunkies.com

http://www.microsoft.com

http://www.nato.int

http://www.nc3a.nato.int

Intranet

http://nc3a-intranet/

Verklarende woordenlijst

ACCS	Air Command and Control System
ACE	Allied Command Europe
ACQ	Acquisition Division
ActD	Active Defence
ARM	Anti-Radiation Missile
ASP.NET	Active Server Pages .NET
Bi-SC AIS	Bi Strategic Command Automatic Information System
C2RC	Command and Control Resource Centre
CCF	Conventional Counter Force
CCS	Command & Control System Division
CM	Cruise Missile
CRONOS	Crises Response Operational NATO Open Systems
EAD	Extended Air Defence
HHS	Haagse Hogeschool
HTML	Hypertext Markup Language
I&I	Informatica & Informatiekunde
IPT	Integrated Project Team
IVIT	Informatievoorzieningen en Informatietechnologie
LSID	Link16 SAM C2 Interoperability Demonstrator
MD	Missile Defence
NAC	North Atlantic Council
NACISA	NATO Communications and Information System Agency
NATO	North Atlantic Treaty Organisation
NAVO	Noord-Atlantische Verdragsorganisatie
NC3A	NATO Consultation, Command & Control Agency
NICS	NATO Integrated Communications System
NICSMA	NICS Management Agency

OWC	Office Web Components
PD	Passive Defence
PlaTo	Planning and Tasking Tool
PTT	post, telephone, telegraph
RUP	Rational Unified Process
SACEUR	Theatre Missile Defence
SADTC	SHAPE Air Defence Technical Centre
SAM	Surface to Air Missile
SDK	Software Development Kit
SHAPE	Supreme Headquarters Allied Powers Europe
SPS	Sharepoint Portal Server
SQL	Structured Query Language
STC	SHAPE Technical Centre
TBM	Tactical Ballistic Missile
UML	Unified Modeling Language
VB.NET	Visual Basic .NET
VS	Verenigde Staten
WMD	Weapons of Mass Destruction
WSS	Windows SharePoint Services

Figurenlijst

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Afstudeerverslag externe bijlage I

Ontwikkeling van een Requirement Capture System voor NAVO software prototype systemen

Externe bijlage I
Project Plan

10 juni 2005

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Project Plan

Development of a Requirement Capture System for NATO software prototype systems

by
J.J. van Houten

8 February 2005

Table of contents

Chapter 1 - Introduction

The purpose of this Project Plan is to give the reader an idea of how the assignment will be carried out. This document is aimed at the mentor, the employer and the trainee. This project plan is divided into three chapters. These are Statement of Work, Project Organisation and Approach.

In chapter two, Statement of Work, the assignment will be well defined. Then a description of how the project will be organised in order to complete the assignment will be given in chapter three. Finally the method and techniques that are going to be used will be explained in chapter four, as well as a brief cost-benefits analysis.

Chapter 2 - Statement of Work

This chapter gives a well-defined description of the assignment.

2.1 General

The intern assignment, described in this program of work will be conducted for the Command and Control Resource Centre (C2RC) of the NATO Consultation, Command & Control Agency (NATO C3 Agency) in The Hague. The NATO C3 Agency (NC3A) develops, procures and implements state of the art Command, Control and Communications (C3) capabilities for NATO and it provides unbiased scientific advice and support to NATO authorities. NC3A is an integrated team of more than 500 professionals, civilian and military, from NATO member nations dedicated to provide to NATO high quality expertise in the domain of Command, Control and Communications.

The Missile Defence (MD) mission area consists of four functional areas. These are:

- Conventional Counter Force (CCF)
- Active Defence (ActD)
- Passive Defence (PD)
- Battle Management, Command, Control, Communications and Intelligence (BMC3I)

In support of these four functional areas, C2RC provides core competences in the following areas:

- Concept Development Support (Prototyping, Exercises)
- Architecture and Architecture Development
- Modelling & Simulation (M&S)
- Integrated Test & Evaluation (IT&E)
- Systems: MD Weapon, Sensors and BMC3 Systems
- MD Decision Support Tools

2.2 Problem description

Amongst other things, NC3A is developing two missile defence related software prototypes:

- Link16 SAM C2 Interoperability Demonstrator (LSID) for Theatre Missile Defence (TMD) Engagement Coordination and Situational Awareness.
- Extended Air Defence (EAD) Planning and Tasking Tool (PlaTo) for TMD planning.

The two software prototypes described above have been used operationally in several NATO and multinational exercises. Furthermore the systems have been installed in a number of locations. The goal for this installation and exercise participation is to capture user requirements and feedback on the prototypes. The resulting prototypes and requirements captured through this process will at a later stage be used to extend the two NATO Command and Control (C2) systems: the Strategic Command Automated Information System (Bi-SC AIS) and the Air Command and Control System (ACCS) with Battle Management and Engagement Coordination functionality.

At the moment, capturing of these user requirements can only be done when NC3A personnel are on site, and the tracking of user requirements is not done in a structured way. Operators might report user requirements back through mechanisms such as email but there is no proper vehicle for them to actually log remarks, requirements and feedback in a logical standardised way and provide them to NC3A.

Feedback received from users is not collected in a central repository and can contain incomplete or misleading information. This current situation does not sufficiently meet NC3A's requirements.

2.3 Goal of the assignment

The goal of this assignment is to develop a system that will allow the operators to log remarks, requirements and feedback through an efficient and structured process.

A process has to be constructed that will facilitate the reporting of user feedback and requirements with respect to the two prototypes to NC3A. Ideally the information provided by the users should be tied into the development process of the named prototypes. The process has to be scalable and should be defined in a clear and structured way. Once the feedback process has been properly worked out and approved, a system has to be implemented to support it.

2.4 Products and services to deliver

- Project Plan (this document)

- Lifecycle Objective Milestone, that includes:
  - Vision Document
  - Software Architecture Document
  - (conceptual) Prototype
  - Business Case
  - Development Case

- Lifecycle Architecture Milestone, that includes:
  - Refined Vision Document
  - Refined Software Architecture Document
  - (executable) Prototype
  - Refined Business Case
  - Refined Development Case

- Initial Operational Capability Milestone, that includes:
  - Refined Vision document
  - Refined Software Architecture Document
  - Refined Business Case
  - Refined Development Case
  - Beta Release of the system

- Product Release Milestone, that includes:
  - Final Vision Document
  - Final Software Architecture Document
  - Final Release of the system
  - User documentation

2.5 Well-defined task

The assignment only involves capturing feedback from the end-users of the software prototype systems LSID and PlaTo. Remarks, requirements and feedback about other prototypes are not relevant to this assignment.

2.6 Limiting conditions

To bring the project to a good end, there have to be some limiting conditions. The most important limiting condition is that there must be sufficient time in the organisation for the trainee to gain information and feedback. The limited duration of the project also requires that the scope of the activity be well defined and managed.

2.7 Risk factors

While conducting the project, there are certain risks involved. It is important to calculate these risks into the planning.

It is possible that licences for required software may not be available. Therefore it is preferable that there are alternatives for the required software.

Required people may not be available and therefore the initial project planning and coordination becomes increasingly important.

Chapter 3 - Project Organisation

This chapter describes how the project will be organised to carry out the assignment.

3.1 Organisation

The next table lists the people who are involved with the project.

Name	Function	Room	E-mail	Phone
J.J. van Houten	Trainee	467	Joost.van.Houten@nc3a.nato.int	070-3743812 (TMD Lab)
A. Schoonen	Mentor and employer	470	Andre.Schoonen@nc3a.nato.int	070-3743773
C. Allmon	Mentor and employer	359A	Charlie.Allmon@nc3a.nato.int	070-3743772
C. Wood	Forum (General)	359A	Clive.Wood@nc3a.nato.int	070-3743774
T. Bingham	Forum (LSID)	359B	Theo.Bingham@nc3a.nato.int	070-3743775
M. Bugru	Forum (LSID)	467	Metin.Bugru@nc3a.nato.int	070-3743812 (TMD Lab)
M. Nederlof	Forum (PlaTo)	467	Marc.Nederlof@nc3a.nato.int	070-3743812 (TMD Lab)
E. van der Koogh	Forum (PlaTo)	467	Erwin.van.der.Koogh@nc3a.nato.int	070-3743812 (TMD Lab)
A. Copner	Forum (PlaTo)	467	Alan.Copner@nc3a.nato.int	070-3743812 (TMD Lab)

3.2 Way of reporting

If possible, every week there will be a meeting with the mentor, employer and the trainee. During this meeting, the progress of the project will be discussed and additional guidance will be provided when required. The weekly meeting will furthermore be used to discuss deliverables and the way ahead.

3.3 Required people and resources

This section will describe the required people, software and hardware.

3.3.1 Required people

The following people are necessary and important for carrying out the assignment:

- Trainee, who will carry out the assignment.
- Employer, who is providing the assignment.
- Mentor, who will guide the trainee with carrying out the assignment.
- Other members of the organisation, who are important as information resources for the trainee and for technical support.

3.3.2 Required software

The following software is required:

- As a solution for the problem is proposed, C2RC will ensure that the required software is available. Potential development tools to be used depending on the type of solution proposed are:
    - IntelliJ IDEA for Java development.
    - MS FrontPage and Adobe GoLive for HTML development.
    - Rational Rose for UML diagram generation.
    - MS SharePoint, for portal development.
- MS Office products for reporting.

3.3.3 Required hardware

The following hardware is required:

- SUN Unix workstation
- Windows based PC

Chapter 4 - Approach

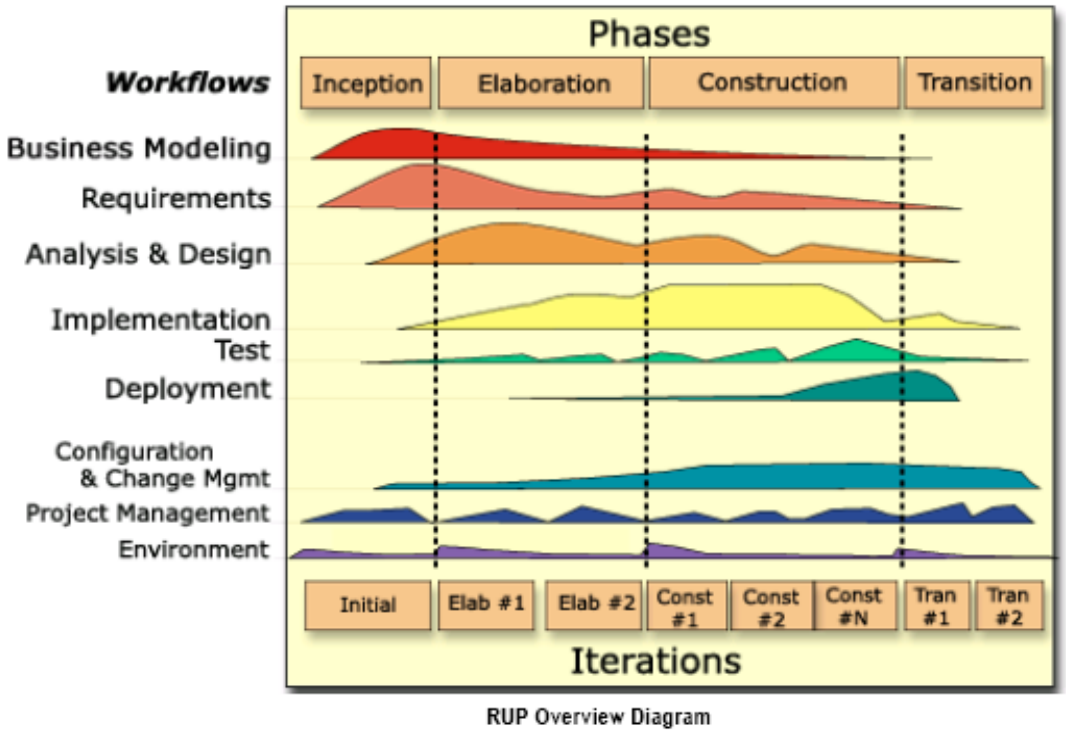
This chapter describes how the project will be carried out.

4.1 Methods and techniques

This section will describe the method and techniques, which are going to be used.

4.1.1 Method

The activities will be carried out according to the Rational Unified Process (RUP) system development method. RUP is a software development approach that is iterative, architecture-centric, and use-case-driven. RUP can be divided into three phases: Inception, Elaboration, Construction and Transition. In each phase, there will be a limited amount of requirements, analysis, design, implementation, and testing.



Process structure of the Rational Unified Process.

Each phase contains one or more iterations that focus on producing the technical deliverables necessary to achieve the objectives of that phase. Each phase has a milestone and a well-defined set of objectives.

4.1.2 Techniques

For conducting the described tasks (section 4.2), the following techniques will be used:

- Interview techniques. These will be uses to determine the requirements and wishes.
- Presentation techniques. These will be used with MS PowerPoint and a beamer for giving presentations and demonstrating prototypes.
- Programming languages will most likely include Java and Hypertext Markup Language (HTML).
- Reporting techniques. All reports will be written with the help of reporting techniques.
- Unified Modeling Language (UML) system development techniques. UML will be used for analysis and design.

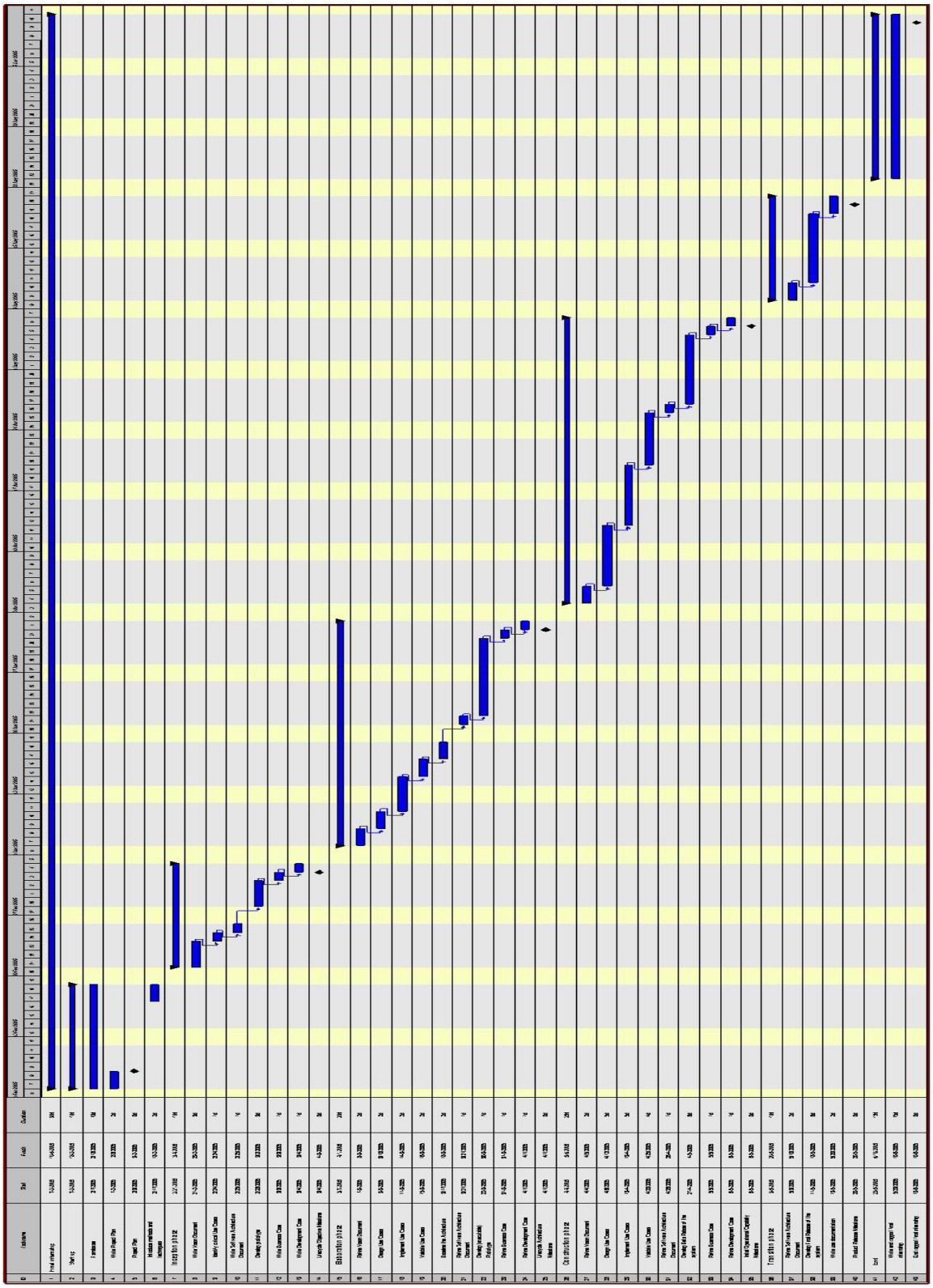
4.2 Global planning

The total project can be divided into six phases. These are:
- Phase 1: Start up
- Phase 2: Inception phase
- Phase 3: Elaboration phase
- Phase 4: Construction phase
- Phase 5: Transition phase
- Phase 6: End

The next table will show an overview of the plan. The activities are divided into the six phases.

Week	Calendar week	Phase	Activity
1	6 (7-2 t/m 11-2)	Start up	Write Project Plan (2 days) Familiarise with the environment and the software prototype systems LSID and PlaTo (6 days) Introduce method and techniques to be used (2 days)
2	7 (14-2 t/m 18-2)	Start up	
3	8 (21-2 t/m 25-2)	Inception	Write Vision Document (3 days) Identify and document critical Use Cases in the Software Architecture Document (2 days) Develop Prototype (3 days) Write Business Case (1 day) Write Development Case (1 day)
4	9 (28-2 t/m 4-3)	Inception	
5	10 (7-3 t/m 11-7)	Elaboration	Refine Vision Document (2 days) Design Use Cases (2 days) Implement Use Cases (2 days) Validate Use Cases (2 days) Baseline the architecture (2 days) Refine Software Architecture Document (1 day) Develop (executable) Prototype (5 days) Refine Business Case (1 day) Refine Development Case (1 day)
6	11 (14-3 t/m 18-3)	Elaboration	
7	12 (21-3 t/m 25-3)	Elaboration	
8	13 (28-3 t/m 1-4)	Elaboration	
9	14 (4-4 t/m 8-4)	Construction	Refine Vision Document (2 days) Design Use Cases (5 days) Implement Use Cases (5 days) Validate Use Cases (4 days) Refine Software Architecture Document (1 day) Develop Beta Release of the system (days 5) Refine Business Case (if necessary) Refine Development Case (if necessary)
10	15 (11-4 t/m 15-4)	Construction	
11	16 (18-4 t/m 22-4)	Construction	
12	17 (25-4 t/m 29-4)	Construction	
13	18 (2-5 t/m 6-5)	Construction	
14	19 (9-5 t/m 13-5)	Transition	Refine Vision Document (if necessary) Refine Software Architecture Document (2 days) Develop Final Release of the system (5 days) Write user documentation (2 days)
15	20 (16-5 t/m 20-5)	Transition	
16	21 (23-5 t/m 27-5)	End	Write final internship end report (15 days)
17	22 (30-5 t/m 3-6)	End	
18	23 (6-6 t/m 10-6)	End	

4.3 Detailed planning



4.4 Description milestone products

In section 2.4 (Products and services to deliver) a number of milestones were mentioned and are briefly described here:

- Project Plan (this document). The Project Plan will describe how the assignment will be carried out.

- Lifecycle Objective Milestone. The goal of this milestone to understand what to build and to identify the key system functionality. Another goal is to understand the costs, schedule and risks associated with the project and the process to follow. This milestone will contain a conceptual prototype.

- Lifecycle Architecture Milestone. The main goal of the architecture milestone is to get a detailed vision about system requirements and functionality and to evaluate an executable architecture. This milestone will include an executable prototype with implemented functionality.

- Initial Operational Capability Milestone. This milestone will include a beta release of the system.

- Product Release Milestone. The last milestone will include the final release of the system.

4.5 Cost-benefit analysis

Carrying out the assignment will incur several costs. These are:
- The costs for the student who will carry out the assignment.
- Facilities that are necessary for the project (computer, software licences).
- Time from persons like the mentor, employer, persons who have to be interviewed and other people who are involved in the project.

Against the costs, which have just been described, the project will also have some benefits:
- The system that will be developed, will eventually improve the development process of the software prototypes LSID and PlaTo.
- The operational community will better accept the added capabilities to Bi-SC AIS and ACCS, since they will be more involved in the development of the requirements.

Glossary

ACCS	Air Command and Control System
ActD	Active Defence
Bi-SC AIS	Strategic Command Automatic Information System
BMC3I	Battle Management, Command, Control, Communications and Intelligence
C2	Command and Control
C2RC	Command and Control Resource Centre
C3	Command, Control and Communications
CCF	Conventional Counter Force
EAD	Extended Air Defence
HTML	Hypertext Markup Language
IT&E	Integrated Test & Evaluation
LSID	Link16 SAM C2 Interoperability Demonstrator
M&S	Modelling and Simulation
MD	Missile Defence
MS	Microsoft
NATO	North Atlantic Treaty Organisation
NC3A	NATO Consultation, Command & Control Agency
PC	Personal Computer
PD	Passive Defence
PlaTo	Planning and Tasking Tool
RUP	Rational Unified Process
SAM	Surface to Air Missile
TMD	Theatre Missile Defence
UML	Unified Modeling Language

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Afstudeerverslag externe bijlage II

Ontwikkeling van een Requirement Capture System voor NAVO software prototype systemen

Externe bijlage II
Lifecycle Objective Milestone

10 juni 2005

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Lifecycle Objective Milestone

Development of a Requirement Capture System for NATO software prototype systems

by
J.J. van Houten

11 March 2005

Foreword

This report is the Lifecycle Objective Milestone of my final internship assignment 'Development of a Requirement Capture System for NATO software prototype systems' at the NATO Consultation, Command & Control Agency (NC3A) in The Hague. This final internship is part of the course 'Informatievoorzieningen en Informatietechnologie' (IVIT) that I currently follow at the Haagse Hogeschool (HHS) in The Hague.

The Lifecycle Objective Milestone with preceding the Project Plan is the second milestone of the project and is part of the Rational Unified Process (RUP) developing method.

Readers that are especially interested in the requirements of the system will be referred to chapter three where the required functionality is being described by use cases. Early conceptual prototypes that have been demonstrated and evaluated are discussed in chapter five. The employer would probably want to read chapter six, Business Case, where carrying out the project is justified.

I hereby want to thank the NC3A personnel for their contributions to this document.

11 March 2005, The Hague

J.J. van Houten

Table of contents

Chapter 1 - Introduction

The development of a new system requires a good understanding of the requirements and wishes of the users. Also the scope of the system has to be well understood. According to the developing method the Rational Unified Process (RUP) this is done in the 'Inception phase'. This phase ends with the Lifecycle Objective Milestone that can be captured in a report.

The purpose of this report is to describe the capturing of the requirements and the conceptual prototypes that have been built in the inception phase of the project. This report will also contain an adapted planning for the next phases of the project.

In chapter two the vision of the project will be described. Chapter three, Software Architecture, will describe the requirements of the system with use cases. Possible solutions for implementing the system will be discussed in chapter four. In chapter five the conceptual prototypes that have been built are discussed. The Business Case that justifies carrying out the project is described in chapter six. Finally the Development Case is described in chapter seven.

Chapter 2 - Vision

The Vision Document is to create a foundation for common understanding of motivation for building the system, as well as a high-level definition of the system to be built. The Statement of Work is somewhat analogous to parts of the Vision Document. The Vision Document should be complete and stable at the end of the Inception phase, but will be refined throughout the project. This chapter will describe the Vision Document.

2.1 Statement of work

2.1.1 General

The intern assignment, described in this program of work will be conducted for the Command and Control Resource Centre (C2RC) of the NATO Consultation, Command & Control Agency (NATO C3 Agency) in The Hague. The NATO C3 Agency (NC3A) develops, procures and implements state of the art Command, Control and Communications (C3) capabilities for NATO and it provides unbiased scientific advice and support to NATO authorities. NC3A is an integrated team of more than 500 professionals, civilian and military, from NATO member nations dedicated to provide to NATO high quality expertise in the domain of Command, Control and Communications.

The Missile Defence (MD) mission area consists of four functional areas. These are:

- Conventional Counter Force (CCF)
- Active Defence (ActD)
- Passive Defence (PD)
- Battle Management, Command, Control, Communications and Intelligence (BMC3I)

In support of these four functional areas, C2RC provides core competences in the following areas:

- Concept Development Support (Prototyping, Exercises)
- Architecture and Architecture Development
- Modelling & Simulation (M&S)
- Integrated Test & Evaluation (IT&E)
- Systems: MD Weapon, Sensors and BMC3 Systems
- MD Decision Support Tools

2.1.2 Problem description

Amongst other things, NC3A is developing two missile defence related software prototypes:

- Link16 SAM C2 Interoperability Demonstrator (LSID) for Theatre Missile Defence (TMD) Engagement Coordination and Situational Awareness.
- Extended Air Defence (EAD) Planning and Tasking Tool (PlaTo) for TMD planning.

The two software prototypes described above have been used operationally in several NATO and multinational exercises. Furthermore the systems have been installed in a number of locations. The goal for this installation and exercise participation is to capture user requirements and feedback on the prototypes. The resulting prototypes and requirements captured through this process will at a later stage be used to extend the two NATO Command and Control (C2) systems: the Strategic Command Automated Information System (Bi-SC AIS) and the Air Command and Control System (ACCS) with Battle Management and Engagement Coordination functionality.

At the moment, capturing of these user requirements can only be done when NC3A personnel are on site, and the tracking of user requirements is not done in a structured way. Operators might report user requirements back through mechanisms such as email but there is no proper vehicle for them to actually log remarks, requirements and feedback in a logical standardised way and provide them to NC3A.

Feedback received from users is not collected in a central repository and can contain incomplete or misleading information. This current situation does not sufficiently meet NC3A's requirements.

2.1.3 Goal of the assignment

The goal of this assignment is to develop a system that will allow the operators to log remarks, requirements and feedback through an efficient and structured process.

A process has to be constructed that will facilitate the reporting of user feedback and requirements with respect to the two prototypes to NC3A. Ideally the information provided bye the users should be tied into the development process of the named prototypes. The process has to be scalable and should be defined in a clear and structured way. Once the feedback process has been properly worked out and approved, a system has to be implemented to support it.

2.2 Target users

This section will describe the different user groups of the system. Roughly there are two user groups. These are the developers of the software prototype systems and the end-users of the software prototype systems.

2.2.1 Developer user group

The developer user group is part of C2RC, which belongs to the Command and Control System Division (CCSD) at NC3A.

At his moment, the developer user group exists of four staff member and three contractors.
Two staff members are working on LSID and three contractors are working on PlaTo. The Senior Scientist is in charge of the development of LSID and PlaTo. LSID and PlaTo are part of the Project TM, which is lead by the Principal Scientist. The next table shows additional information about the developer user group.

Name	Function	Room	E-mail	Phone
A. Schoonen	In charge of development	470	Andre.Schoonen@nc3a.nato.int	070-3743773
C. Allmon	Project Manager TM	359A	Charlie.Allmon@nc3a.nato.int	070-3743772
T. Bingham	Programmer on LSID	359B	Theo.Bingham@nc3a.nato.int	070-3743775
M. Bugru	Programmer on LSID	467	Metin.Bugru@nc3a.nato.int	070-3743812 (TMD Lab)
M. Nederlof	Programmer on PlaTo	467	Marc.Nederlof@nc3a.nato.int	070-3743812 (TMD Lab)
E. van der Koogh	Programmer on PlaTo	467	Erwin.van.der.Koogh@nc3a.nato.int	070-3743812 (TMD Lab)
A. Copner	Programmer on PlaTo	467	Alan.Copner@nc3a.nato.int	070-3743812 (TMD Lab)

Figure 2.2-1 Developer user group

2.2.2 End-user group

At this moment the users of PlaTo are limited to the Head Quarters Extended Air Defence Task Force (HQ EADTF) and the Head Quarters Allied Rapid Reaction Corps (HQ ARRC). It is possible that PlaTo is going to be used at the Regional Command (RC) South in Naples (Italia) and the Air Component Command (ACC) South in Izmir (Turkey). There are no users for LSID at this moment.

2.3 System activities

This section describes the system activities by using features and key use cases.

2.3.1 System features

Feature No.1	
Name	Provide feedback.
Description	Allow End-users to provide feedback on the software prototype systems LSID and PlaTo.
Value	1.
Complexity	Medium.
Priority	1.

Figure 2.3-1 Feature No.1 Provide feedback

Feature No.2	
Name	Edit feedback.
Description	Allow End-users to edit their own provided feedback.
Value	3.
Complexity	High.
Priority	3.

Figure 2.3-2 Feature No.2 Edit feedback

Feature No.3	
Name	View feedback.
Description	Allow End-users view all feedback on the software prototype systems LSID and PlaTo.
Value	4.
Complexity	Low.
Priority	4.

Figure 2.3-3 Feature No.3 View feedback

Feature No.4	
Name	View feedback (NC3A developer).
Description	Allow NC3A developers to view all feedback on the software prototype systems LSID and PlaTo.
Value	1.
Complexity	Low.
Priority	1.

Figure 2.3-4 Feature No.4 View feedback (NC3A developer)

Feature No.5	
Name	Edit status feedback.
Description	Allow NC3A developers to edit the status of feedback.
Value	2.
Complexity	Medium.
Priority	2.

Figure 2.3-5 Feature No.5 Edit status feedback

2.3.2 Key use cases

According to the system features discussed in section 2.3.1 the following key actors can be identified: End-user and NC3A developer. The following key use cases can be identified: Provide feedback, Edit feedback, View feedback and Edit status feedback. The following figures will show these key use cases.

Key use case No. 1	
Name	Provide feedback.
Assumption	End-user has logged into the system.
Actors	End-user.
Description	(1) End-user indicates that he want to enter feedback. (2) The system asks the End-user on which software prototype system he wants to gives feedback. (3) End-user answers that question and (4) the system will ask the End-user for his feedback. End-user enters his feedback and tells the system that he wants to save it. (5) The system will confirm that the feedback has been saved.
Exceptions.	None.
Result	Feedback is added

Figure 2.3-6 Key use case No.1 Provide feedback

Key use case No. 2	
Name	Edit feedback.
Assumption	End-user has logged into the system and has selected feedback he had previously entered.
Actors	End-user
Description	(1) End-user indicates that he want to edit the selected feedback. (2) The system asks End-user to edit the feedback and (3) End-user will edit it. (4) Then End-user says he want to save it. (5) The system will confirm that the feedback has been changed.
Exceptions.	None.
Result	Feedback is changed.

Figure 2.3-7 Key use case No.2 Edit feedback

Key use case No. 3	
Name	View feedback.
Assumption	End-user or NC3A developer has logged into the system.
Actors	End-user, NC3A developer.
Description	(1) End-user or NC3A developer indicated that he wants to view feedback. (2) The system asks on which software prototype system he wants to view feedback and (3) End-user or NC3A developer answers this question. (4) The system will show the requested feedback.
Exceptions.	[No feedback]. The system will tell End-user or NC3A developer that there is no feedback at this moment.
Result	Feedback is retrieved.

Figure 2.3-8 Key use case No.3 View feedback

Key use case No. 4	
Name	Edit status feedback.
Assumption	End-user has logged into the system and has selected feedback.
Actors	NC3A developer.
Description	(1) NC3A developer indicates that he want to edit the status of the selected feedback. (2) The system asks which status he wants to add to the feedback and (3) NC3A developer answers this question. (4) The system confirms the new status of the feedback.
Exceptions.	None.
Result	Status of feedback is changed.

Figure 2.3-9 Key use case No.4 Edit status feedback

2.3.3 Use case diagram

The following figure shows the use case diagram based on the key use cases previously described.

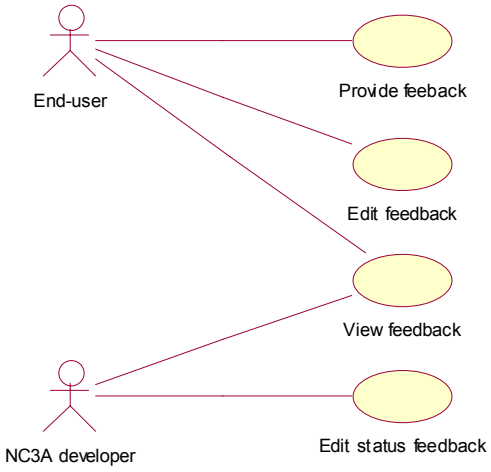


Figure 2.3-10 Use case diagram

2.4 Non-functional requirements

This section briefly describes the non-functional requirements.

- Operation System. The system has to be compatible with Windows 2000 and preferably also with Unix Solaris.

- Database support. There are no specific requirements for choosing the database environment to use. It probably would be MySQL or MS Access.

- Programming environment. When there is going to be a stand-alone application, the programming language would be Java. Developing a web-based application can be done with Active Server Pages (ASP), Active Server Pages .NET (ASPX), Java Server Pages (JSP), PHP Hypertext Preprocessor (PHP), JavaScript or Visual Basic Script (VBScript).

Chapter 3 - Software Architecture

The Software Architecture Document is the overall documentation document that will contain all UML analysis and design models. This document will be refined in every iteration and phase during the project. This chapter describes the Software Architecture Document.

3.1 'Mile-wide, inch-deep' description

There is a need to provide a good understanding of the scope of the system, without going into too much in detail. This is called the 'mile-wide, inch-deep' understanding of the system. In the 'mile-wide, inch-deep' description as many actors and use cases as possible are briefly described. Approximately 20 percent of the use cases are going to be listed as critical.

3.1.1 Actors

Actor No.1	
Name	End-user.
Description	An End-user is a person who is using and testing the software prototype systems developed by NC3A.

Figure 3.1-1 Actor No.1 End-user

Actor No.2	
Name	NC3A developer.
Description	A NC3A developer is a person who works on a software prototype system for NC3A.

Figure 3.1-2 Actor No.2 NC3A developer

Actor No.3	
Name	NC3A administrator
Description	A NC3A administrator is a person who works on software prototype systems NC3A and will be the administrator of the Requirement Capture System that has to be developed.

Figure 3.1-3 Actor No.3 NC3A administrator

3.1.2 Use cases

Use case No. 1	
Name	Sign in.
Assumption	The system is started up.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The user provides his/her username and password and tells the system he/she wants to sign in. (2) The systems signs the user in.
Exceptions	[Username unknown / Username and password combination not correct.] The system tells the user that the sign in procedure failed.
Result	User has signed in.

Figure 3.1-4 Use case No.1 Sign in

Use case No. 2	
Name	Sign out.
Assumption	User has signed in.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The user tells the system that he/she wants so sing out. (2) The system signs out the user.
Exceptions	None.
Result	User has signed out.

Figure 3.1-5 Use case No.2 Sign out

Use case No. 3	
Name	View name singed in user.
Assumption	User has signed in.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows the name of the user that is signed in.
Exceptions	None.
Result	Name user signed in is displayed.

Figure 3.1-6 Use case No.3 View name signed in user

Use case No. 4	
Name	Select software prototype system.
Assumption	User has signed in.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system asks the user on which software prototype system he/she wants to provide feedback and (2) the user answers that question.
Exceptions	[No software prototype systems available.] The system tells the user that there are no software prototype systems available.
Result	A software prototype system is selected.

Figure 3.1-7 Use case No.4 Select software prototype system

Use case No. 5	
Name	Search for feedback.
Assumption	User has signed in and has selected a software prototype system.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system asks for what kind of feedback the user is looking for. (2) The user answers that question and (3) the system shows the feedback.
Exceptions	[No feedback found.] The system tells the user that there is no feedback found.
Result	Feedback is displayed.

Figure 3.1-8 Use case No.5 Search for feedback

Use case No. 6	
Name	View feedback.
Assumption	User has signed in and has selected a software prototype system and searched for feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows the user feedback.
Exceptions	[No feedback present.] The system tells the user there is no feedback.
Result	Feedback is displayed.

Figure 3.1-9 Use case No.6 View feedback

Use case No. 7	
Name	View feedback status.
Assumption	User has signed in and has selected a software prototype system and searched for feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows the user the feedback status.
Exceptions	[No feedback present.] The system tells the user there is no feedback.
Result	Feedback status is displayed.

Figure 3.1-10 Use case No.7 View feedback status

Use case No. 8	
Name	View name author of feedback.
Assumption	User has signed in and has selected a software prototype system and searched for feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows the feedback author.
Exceptions	[No feedback present.] The system tells the user there is no feedback.
Result	Feedback author is displayed.

Figure 3.1-11 Use case No.7 View name author of feedback

Use case No. 9	
Name	View date feedback provided.
Assumption	User has signed in and has selected a software prototype system and searched for feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows the date the feedback was provided.
Exceptions	[No feedback present.] The system tells the user there is no feedback.
Result	Feedback date provided is displayed

Figure 3.1-12 Use case No.9 View date feedback provided

Use case No. 10	
Name	View added user note.
Assumption	User has signed in and has selected a software prototype system and selected feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows additional user note(s) made to the feedback.
Exceptions	[No feedback present.] The system tells the user there is no feedback.
Result	Added user note is displayed.

Figure 3.1-13 Use case No.10 View added user note

Use case No. 11	
Name	View date user note added.
Assumption	User has signed in and has selected a software prototype system and selected feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows the date the additional note was made to the feedback.
Exceptions	[No feedback present.] The system tells the user there is no feedback.
Result	Data user note is displayed.

Figure 3.1-14 Use case No.11 View date user note added

Use case No. 12	
Name	View added developer note.
Assumption	User has signed in and has selected a software prototype system and selected feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows additional developer note(s) made to the feedback.
Exceptions	None.
Result	Added developer note is displayed.

Figure 3.1-15 Use case No.12 View added developer note

Use case No. 13	
Name	View date added developer note added.
Assumption	User has signed in and has selected a software prototype system and selected feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows the date of the additional note that is posted to the feedback.
Exceptions	None.
Result	Date developer note is displayed.

Figure 3.1-16 Use case No.13 View date developer note added

Use case No. 14	
Name	Add prototype.
Assumption	User has signed in.
Actors	NC3A administrator.
Description	(1) The user tells the system he/she wants to add a new software prototype system. (2) The system asks the name and properties of the software prototype system and (3) the user answers that question.
Exceptions	None.
Result	New prototype is added.

Figure 3.1-17 Use case No.14 Add software prototype system

Use case No. 15	
Name	Modify prototype.
Assumption	User has signed.
Actors	NC3A administrator.
Description	(1) The user tells the system that he/she wants to modify the properties of a certain software prototype system. (2) The system asks what changes he/she wants to make and (3) the user answers that question.
Exceptions	None.
Result	Software prototype system is modified.

Figure 3.1-18 Use case No.18 Modify software prototype system

Use case No. 16	
Name	Delete software prototype system.
Assumption	User has signed in.
Actors	NC3A administrator.
Description	(1) The user tells the system that he/she wants to delete a certain software prototype system.
Exception	None.
Result	Software prototype system is deleted.

Figure 3.1-19 Use case No.16 Delete software prototype system

Use case No. 17	
Name	Add user.
Assumption	User has signed in.
Actors	NC3A administrator.
Description	(1) The user tells the system that he/she wants to add a new user. (2) The system asks the user what the properties for the new user are (3) and the user answers that question.
Exceptions	None.
Result	New user is added.

Figure 3.1-20 Use case No.17 Add user

Use case No. 18	
Name	Modify user.
Assumption	User has signed in.
Actors	NC3A administrator.
Description	(1) The user tells the system that he/she wants to modify a certain user. (2) The system asks the user what changes he/she wants to make and (3) the user answers that question.
Exceptions	None.
Result	User is modified.

Figure 3.1-21 Use case No.18 Modify user

Use case No. 19	
Name	Delete user.
Assumption	User has signed in.
Actors	NC3A administrator.
Description	(1) The user tells the system that he/she wants to delete a certain user.
Exceptions	None.
Result	User is deleted.

Figure 3.1-22 Use case No.19 Delete user

Use case No. 20	
Name	Delete user note.
Assumption	User has signed in and has selected a software prototype system and searched for feedback.
Actors	NC3A administrator.
Description	(1) The user tells the system that he/she wants to delete a user note.
Exceptions	None.
Result	User note is deleted.

Figure 3.1-23 Use case No.20 Delete user note

Use case No. 21	
Name	Delete developer note.
Assumption	User has signed in and has selected a software prototype system and searched for feedback.
Actors	NC3A administrator
Description	(1) The user tells the system that he/she wants to delete a user note.
Exceptions	None.
Result	Developer note is deleted.

Figure 3.1-24 Use case No.21 Delete developer note

Use case No. 22	
Name	Change feedback status.
Assumption	User has signed in and has selected a software prototype system and selected feedback.
Actors	NC3A developer.
Description	(1) The user tells the system that he/she wants to change the status of the feedback. (2) The system asks him/her to what status he/she wants to change the feedback and (3) the user answers that question.
Exceptions	None.
Result	Feedback status is changed.

Figure 3.1-25 Use case No.22 Change feedback status

Use case No. 23	
Name	Add developer note.
Assumption	User has signed in and has selected a software prototype system and selected feedback.
Actors	NC3A developer.
Description	(1) The user tells the system that he/she wants to add a developer note. (2) The system asks the user what the contents of the developer note must be and (3) the user answers that question.
Exceptions	None.
Result	Developer note is added.

Figure 3.1-26 Use case No.23 Add developer note

Use case No. 24	
Name	Provide feedback.
Assumption	Use has signed in and has selected a software prototype system.
Actors	End-user.
Description	(1) The user tells the system that he/she wants to add feedback. (2) The system asks him/her what the contents of the feedback must be and (3) the user answers that question.
Exceptions	None.
Result	Feedback is provided.

Figure 3.1-27 Use case No.24 Provide feedback

Use case No. 25	
Name	Add user note.
Assumption	User has signed in and has selected a software prototype system and selected feedback.
Actors	End-user.
Description	(1) The user tells the system that he/she wants to add a user note. (2) The system asks him/her what the contents of the user note must be and (3) the user answers that question.
Exceptions	None.
Result	User note is added.

Figure 3.1-28 Use case No.25 Add user note

3.1.3 Use case diagram

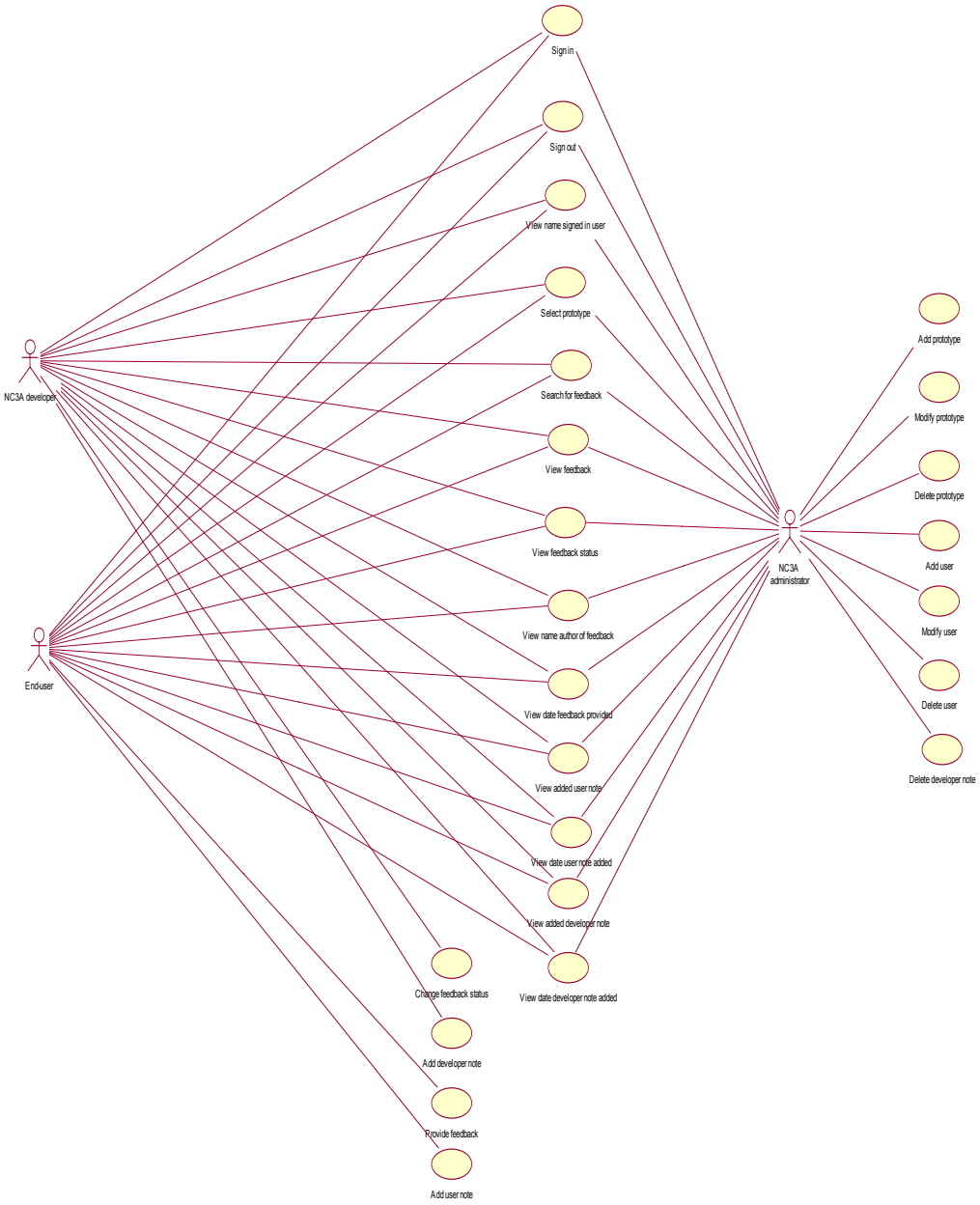


Figure 3.1-29 Use case diagram

3.2 Critical use cases

It is important to decide which use cases are the most essential or architecturally significant for the system to be built. The critical use cases should also describe the core functionality of the system. In this section these critical use cases are described.

3.2.1 Descriptions of the critical use cases

Critical use case	No. 1
Name	Sign in.
Assumption	The system is started up.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system asks the user his/her username and password. (2) The user enters his/her username and password and tells the system he/she wants to sign in. (3) The system signs in the user en shows the available software prototype systems.
Exceptions	[Username unknown.] The system tells the user that the username is unknown. [Username and password combination not valid.] The system tells the user that the username and password combination is not valid.
Result	User has signed in.

Figure 3.2-1 Critical use case No.1 Sign in

Critical use case	No. 2
Name	Provide feedback.
Assumption	Use has signed in as End-user and has selected a software prototype system and a module.
Actors	End-user.
Description	(1) The user tells the system that he/she wants to add feedback. (2) The system asks him/her the feedback title, feedback description, module where the feedback is about, version of the software prototype system and the Operational System that has been used. (3) The user answers that question and tells the system that he wants to submit the feedback. (4) The system tells the user that the feedback has been submitted and shows the user the feedback he/she just provided along with the other feedback entries previously been made.
Exceptions	None.
Result	Feedback is provided.

Figure 3.2-2 Critical use case No.2 Provide feedback

Critical use case	No. 3
Name	View feedback.
Assumption	User has signed in and has selected a software prototype system and a module or searched for feedback.
Actors	End-user, NC3A developer and NC3A administrator.
Description	(1) The system shows the feedback entries that previously have been made and belong to the chosen module or the search query that has been submitted. Feedback entries will include feedback title, module where the feedback is about, status feedback, date feedback posted, author of the feedback, date last note is added and author of last added note. (2) The user tells the system on which feedback entry he/she wants to look at. (3) The system shows the user the feedback title, feedback description, module where the feedback is about, the version of the prototype, the Operation System that has been used, author of the feedback, date feedback posted, status feedback and the developer/user notes that have been made to this feedback. A developer/user note exists of a remark description, author of remark and date remark posted.
Exceptions	[No feedback present.] The system tells the user there is no feedback present or found.
Result	Feedback is displayed.

Figure 3.2-3 Critical use case No.3 View feedback

Critical use case	No. 4
Name	Change feedback status.
Assumption	User has signed in as NC3A developer and has selected a software prototype system, a module and a feedback entry.
Actors	NC3A developer.
Description	(1) The user tells the system that he/she wants to change the status of the feedback. (2) The system asks the user to what status he/she wants to change the feedback. (3) The user answers that question. (4) The system tells the user that the feedback status is changed (also tells the user to what status) and shows the updated feedback entry.
Exceptions	[Feedback status is 'Closed'.] The system tells the user that it is not possible to change the status of the feedback because it is already closed.
Result	Feedback status is changed.

Figure 3.2-4 Critical use case No.4 Change feedback status

Critical use case	No. 5
Name	Add developer note.
Assumption	User has signed in as NC3A developer and has selected a prototype, a module and a feedback entry.
Actors	NC3A developer.
Description	(1) The user tells the system that he/she wants to add a developer note. (2) The system asks the user what should be in the developer note. (3) The user answers that question and tells the system that he wants to submit the note. (4) The system tells the user that the note has been submitted and shows the updated feedback entry.
Exceptions	None.
Result	Developer note is added.

Figure 3.2-5 Critical use case No.5 Add developer note

Critical use case	No. 6
Name	Add user note.
Assumption	User has signed in as End-user and has selected a software prototype system, a module and a feedback entry.
Actors	End-user.
Description	(1) The user tells the system that he/she wants to add a user note. (2) The system asks the user what should be in the user note. (3) The user answers that question and tells the system that he wants to submit the note. (4) The system tells the user that the note has been submitted and shows the updated feedback entry.
Exceptions	None.
Result	User note is added.

Figure 3.2-6 Critical use case No.6 Add user note

Chapter 4 - Possible solutions

There are several possible solutions for the implementation of the system. All possible solutions have to be discussed and one has to be chosen in an early stage because of the influence the chosen solution has on the whole development of the system. In this chapter possible solutions are discussed.

4.1 Location and access of users

The NC3A developers are mainly working in the Theatre Missile Defence (TMD) lab and in some Class 2 Security Area rooms at the NC3A building in The Hague. The TMD lab is a Class 1 Security Area.

The network in the TMD lab is considered NATO Secret but is not and shall not be connected to the NATO Secret network called the Crises Response Operational NATO Open Systems (CRONOS). In the TMD lab there is one CRONOS terminal connected to the CRONOS network and one NATO Restricted computer that allows personnel to check their NATO Restricted and NATO Unclassified email and search the Internet for information.

The Class 2 Security Area rooms have one or more NATO Restricted machines to access unclassified and restricted email and search the Internet for information.

The End-users can be on different locations of the NATO member nations. At this moment the users of PlaTo are limited to HQ EADTF and HQ ARRC. It is possible that PlaTo is going to be used at RC South in Naples (Italia) and ACC South in Izmir (Turkey).

The End-user should have access to the Internet and therefore also to the NATO Unclassified network, for example the website of NC3A (http://www.nc3a.nato.int). It is also possible that they have access to the NATO Secret network CRONOS.

4.2 Stand-alone VS Web-based

One of the main choices that have to be made is whether the system is going to be web-based or stand-alone. The advantage of a stand-alone application is the possibility to develop the user interface exactly according to the non-functional requirements of the users. The advantage of a web-based application is that the users do not have to install the application because they can access the system using a web browser.

4.3 Solution one: unclassified

It is possible to host the system on a NATO unclassified machine that is not connected to any other network of NATO or NC3A at all.

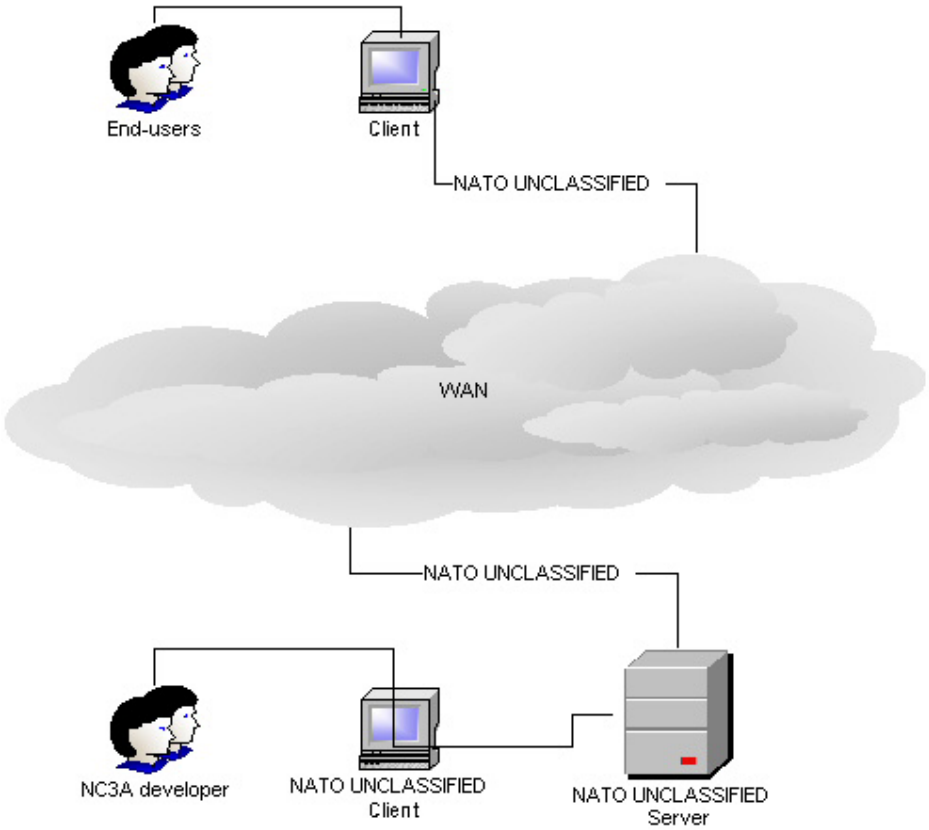


Figure 4.3-1 Solution one: unclassified

The condition of this solution is that all data that would be transferred must be NATO unclassified. This would allow the End-users from different locations outside NC3A to access the system. If possible there could be placed an additional NATO unclassified machine in the TMD lab so the NC3A developers would have access to the system at their working place.

4.4 Solution two: NATO secret

The other solution is to host the system at the NATO secret network CRONOS. The procedure of getting approval for hosting a system on the NATO secret network CRONOS will be time consuming.

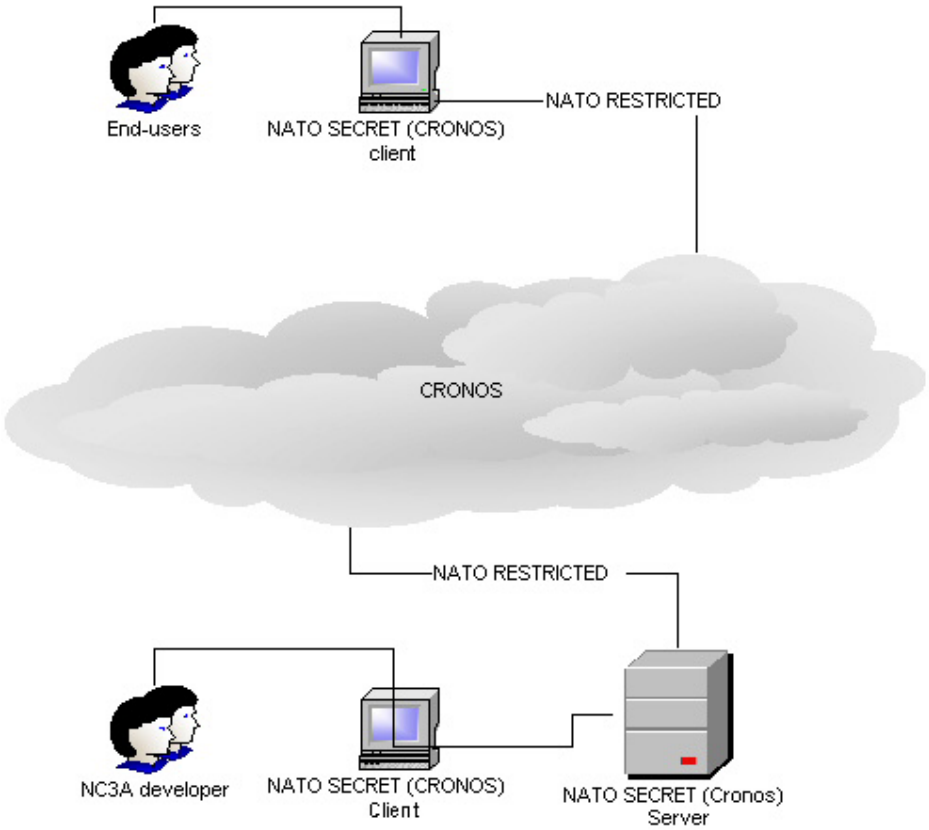


Figure 4.4-1 Solution two: NATO secret

The advantage of this solution is that it is possible to transfer Classified up to NATO Secret data. Also there has not to be installed a new machine in the TMD lab because there is already a CRONOS terminal available. In several Class 2 Security Area there are also CRONOS terminals present.

Chapter 5 - Conceptual prototype

The conceptual prototypes that have been developed in the inception phase are static web pages without implemented functionality. These prototypes are used to gather feedback from the users at NC3A. It should be able to reuse the GUI's that are designed for these prototypes. This chapter will describe the conceptual prototypes that have been demonstrated and evaluated in the inception phase.

5.1 Navigational diagram

Before the actual prototypes where being developed, there was made a navigation diagram to get a rough idea of the navigation trough the system. This diagram in illustrated in the next figure.

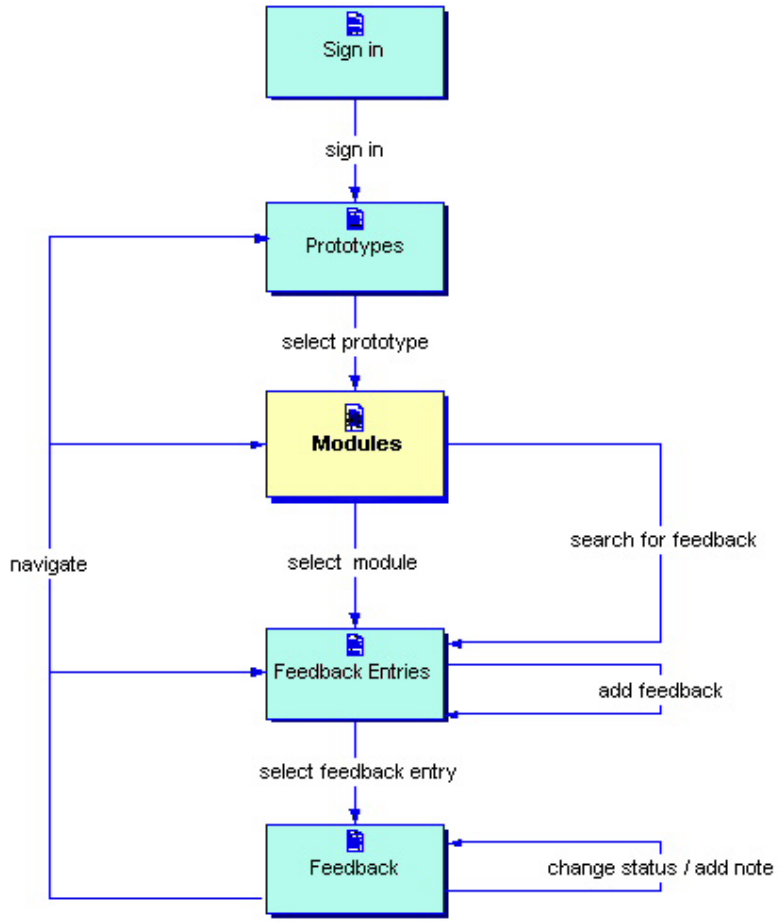


Figure 5.1-1 Navigational diagram

5.2 The End-user view

The first prototype that has been developed is a prototype that is based on the End-user functionality. In this first prototype the End-user functionality is conceptualised. In this section the first prototype will be discussed.

5.2.1 Signing in

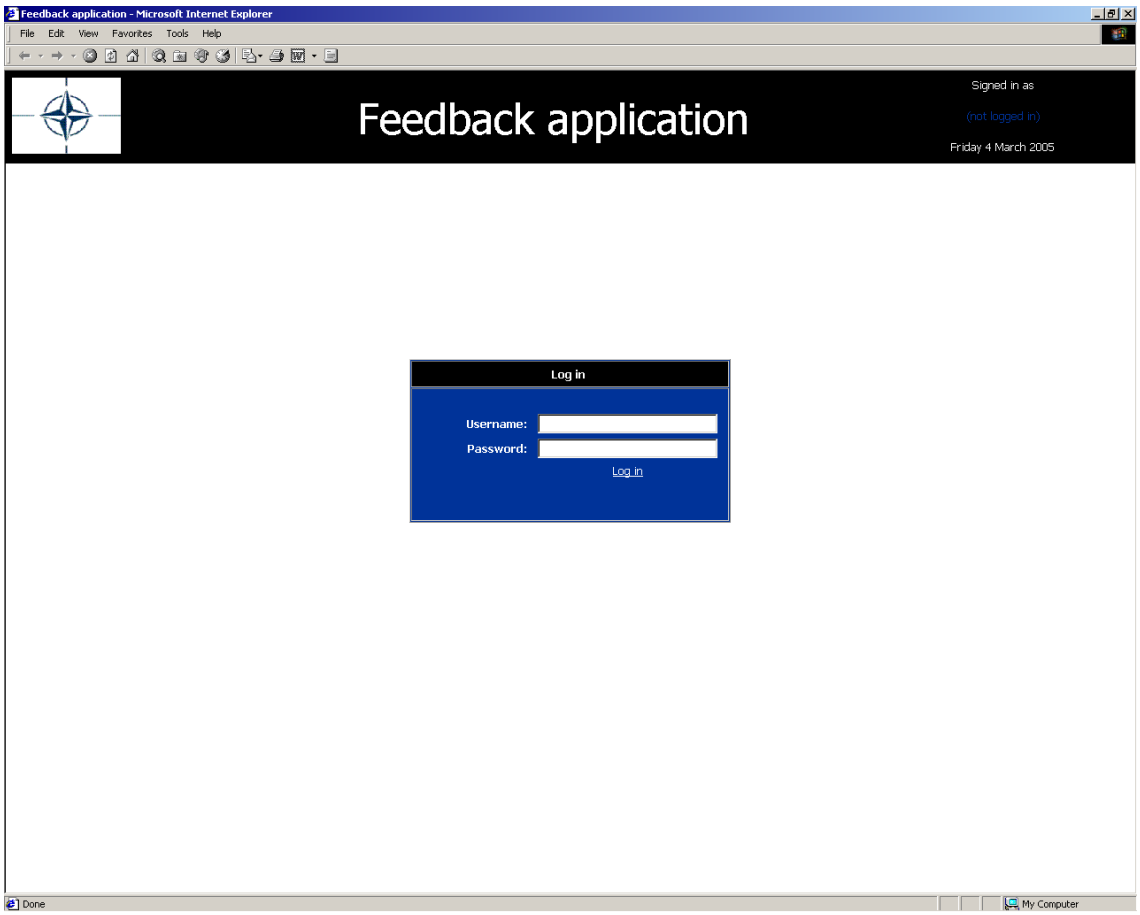


Figure 5.2-1 Sign in screen

There are two frames during the whole application. There is a header frame where a logo, the name of the application and the name of the user who has signed in along with the date of the current day are displayed.

The main frame contains a section where the user can enter his/her username and password and can sign by clicking on ‘Log in’.

5.2.2 Main screen

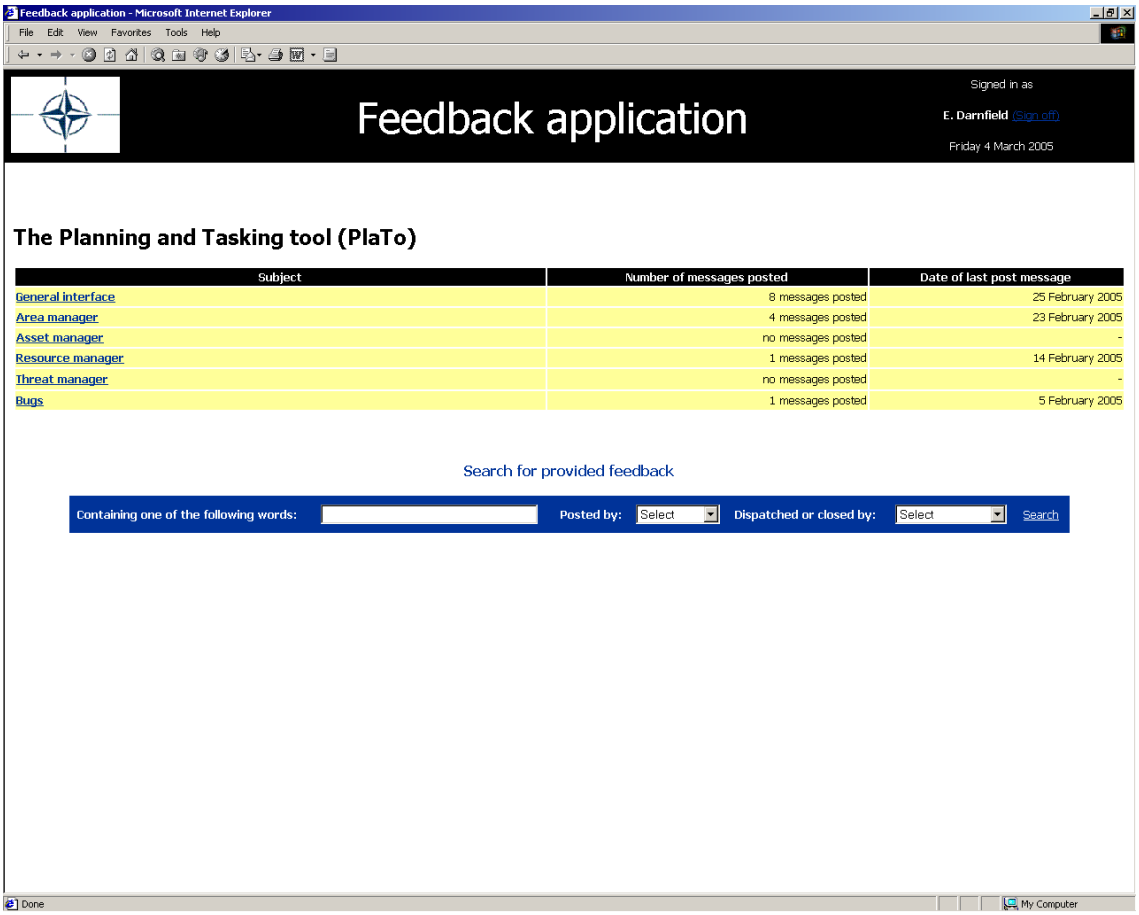


Figure 5.2-2 Main screen

After the user has signed in, the main screen will be displayed. This prototype is only supporting the users of PlaTo. In the main screen the different subjects of feedback are displayed along with the numbers of messages posted in that subject and the date of the last posted message. It is possible to click on a subject.

5.2.3 Viewing feedback

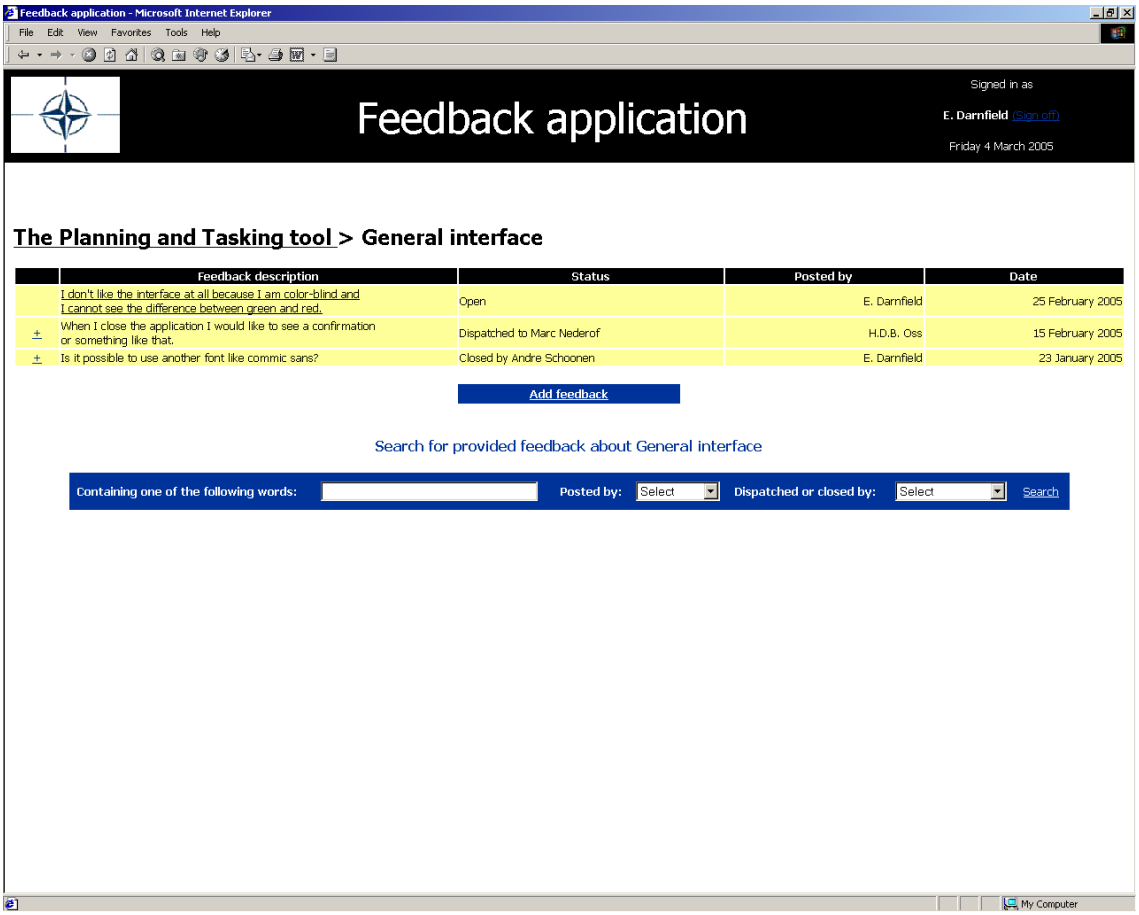


Figure 5.2-3 Feedback screen

When the user has clicked on a subject in the main screen, a list is displayed with feedback previously provided. The list contains a feedback description, status of the feedback, author of feedback and the date the feedback was posted. Some feedback can contain additional notes made by End-users of NC3A developers. When that is the case, it is possible to click on the '+' sign displayed before the feedback description. In this screen it is also possible to click on the 'Add feedback' displayed below the feedback list.

5.2.4 Adding feedback

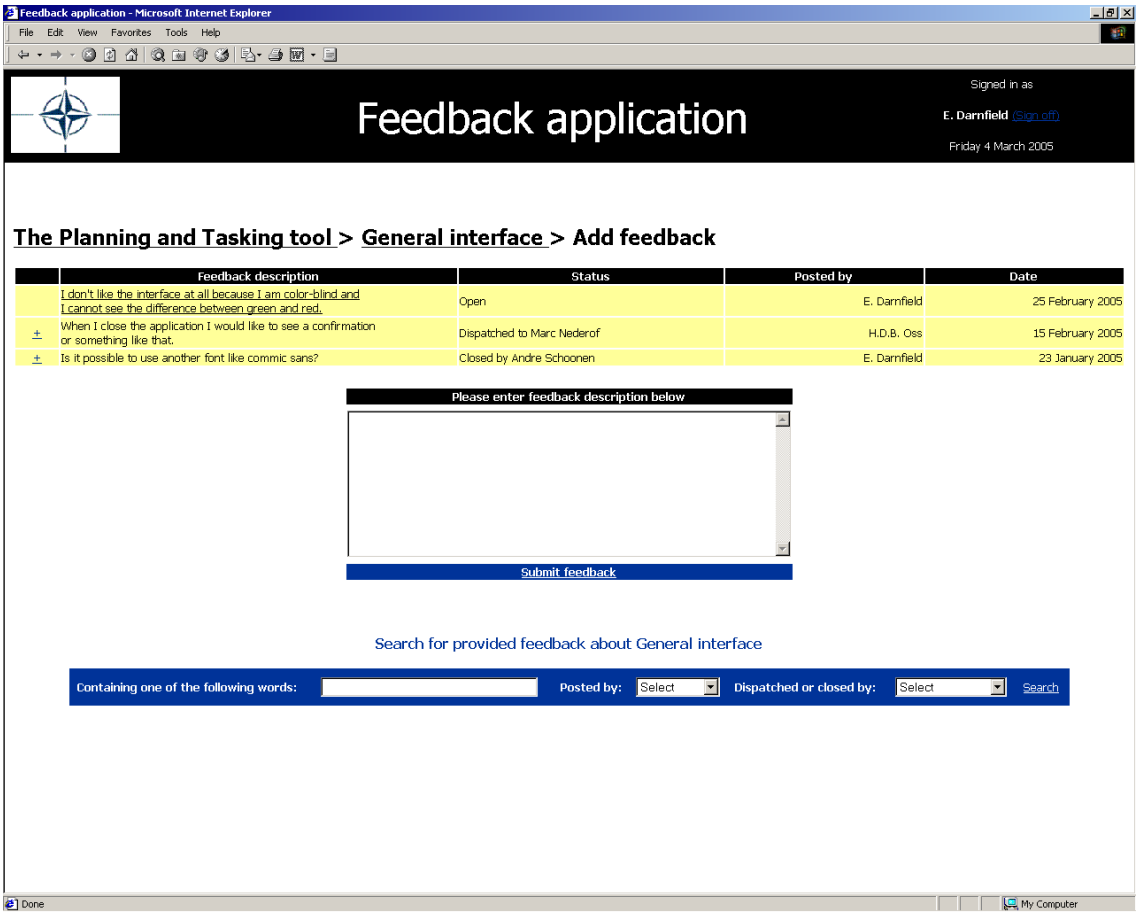


Figure 5.2-4 Add feedback screen

When the user had clicked on the ‘Add feedback’ in de feedback screen, a blank text area appears. In that text area the user can enter a new feedback description and submit is by clicking on ‘Submit feedback’.

5.2.5 Viewing notes

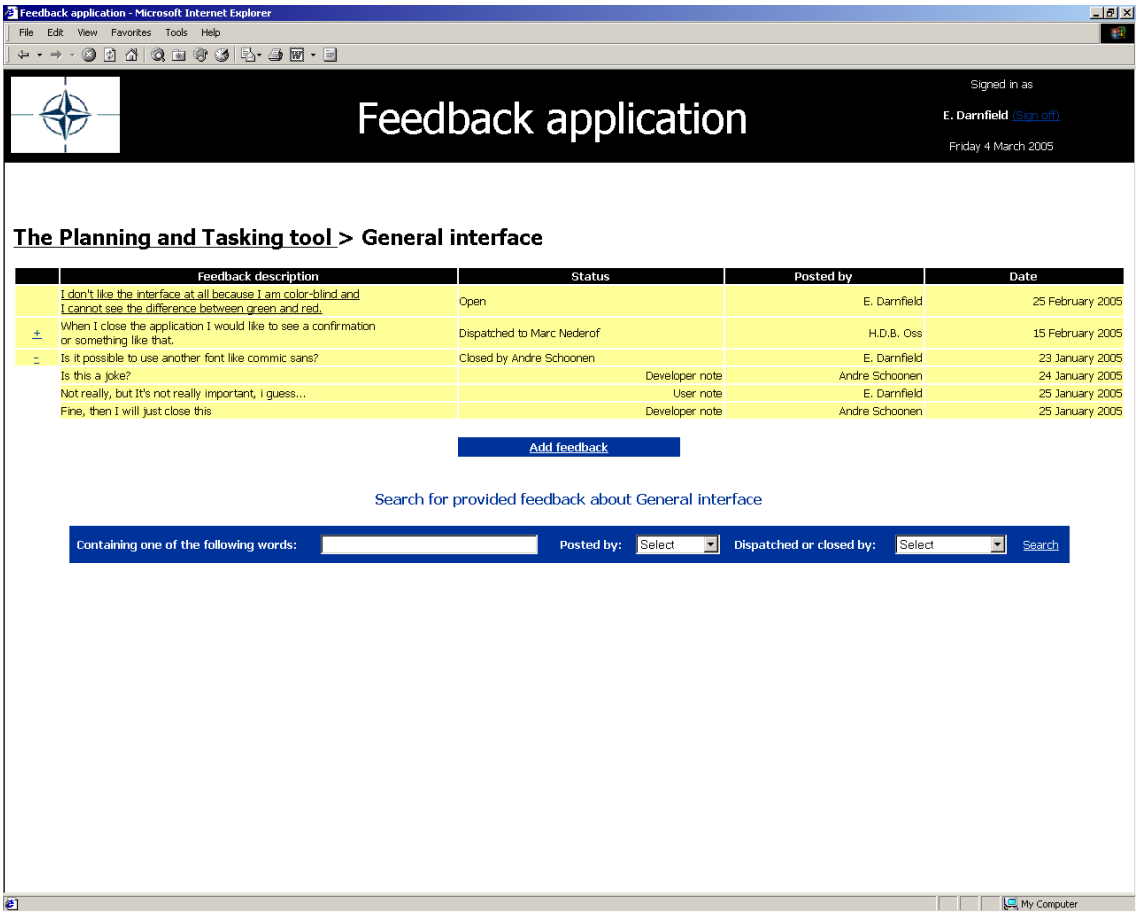


Figure 5.2-5 View note screen

If the user has clicked on a '+' sign in the feedback screen, the additional notes will appear. While viewing these notes, it is possible to click on 'Add note'.

5.2.6 Adding a note

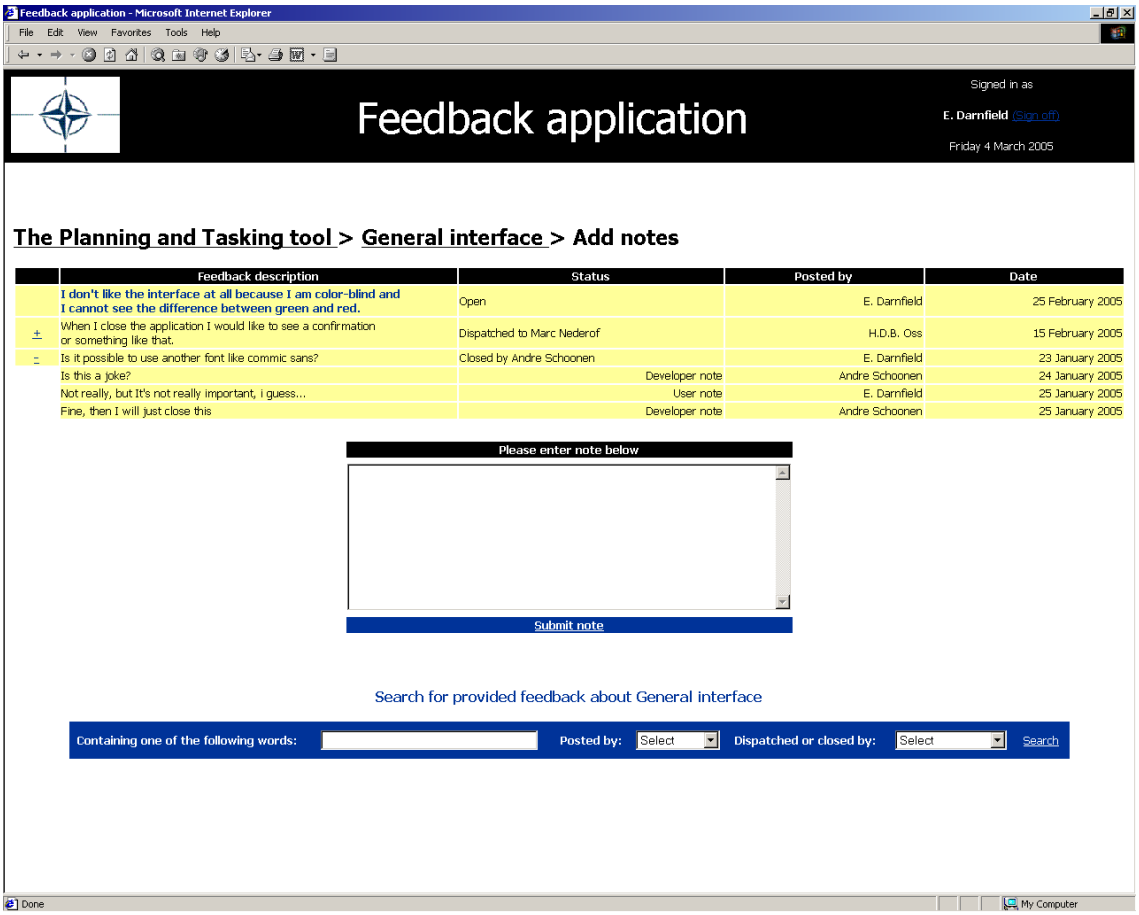


Figure 5.2-6 Add note screen

When the user clicked on ‘Add note’ while viewing additional notes made a feedback entry, a blank text area will appear where the user can enter the note he/she wants to add. The user can submit this note by clicking on ‘Submit note’.

5.2.7 Searching for feedback

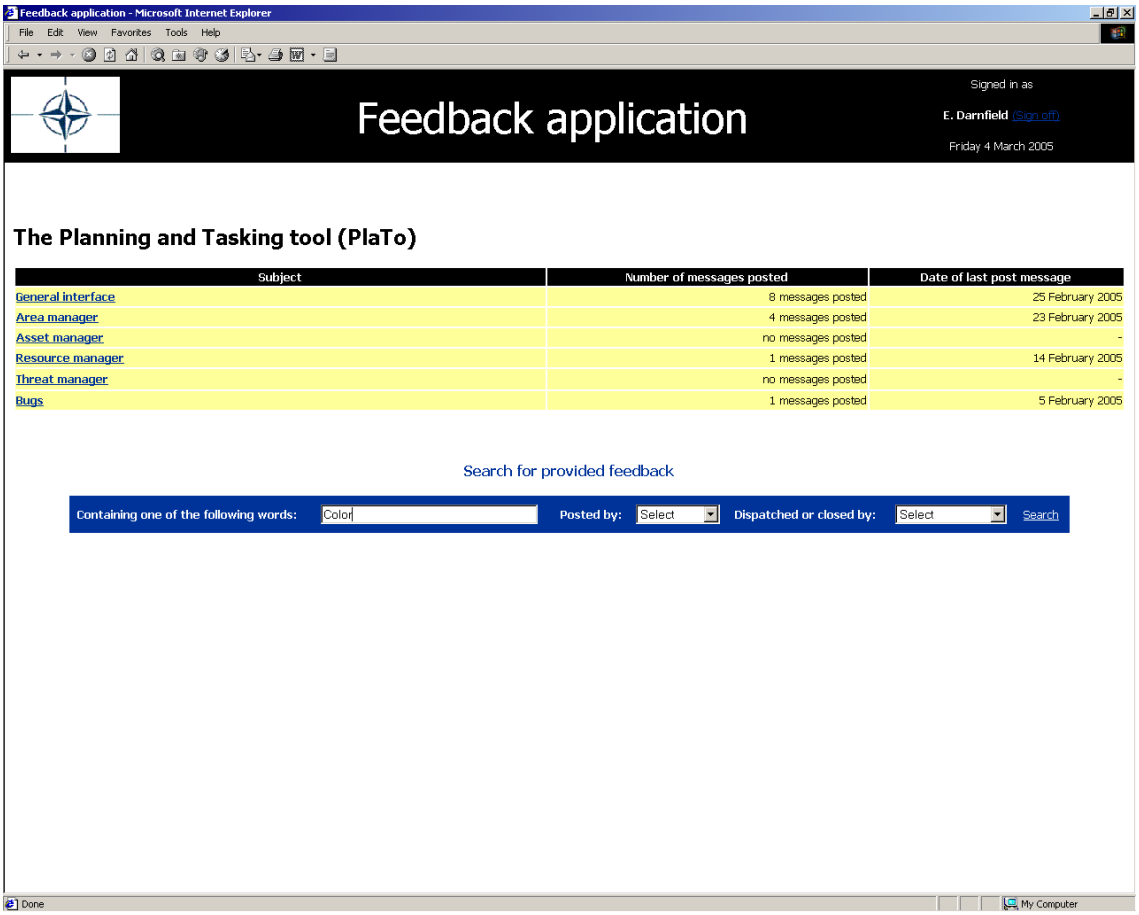


Figure 5.2-7 Search screen

The main screen also contains a search field. It is possible to search for feedback that is previously provided. It is possible to search on words, the author and the NC3A developer to which the feedback is dispatched or who closed it.

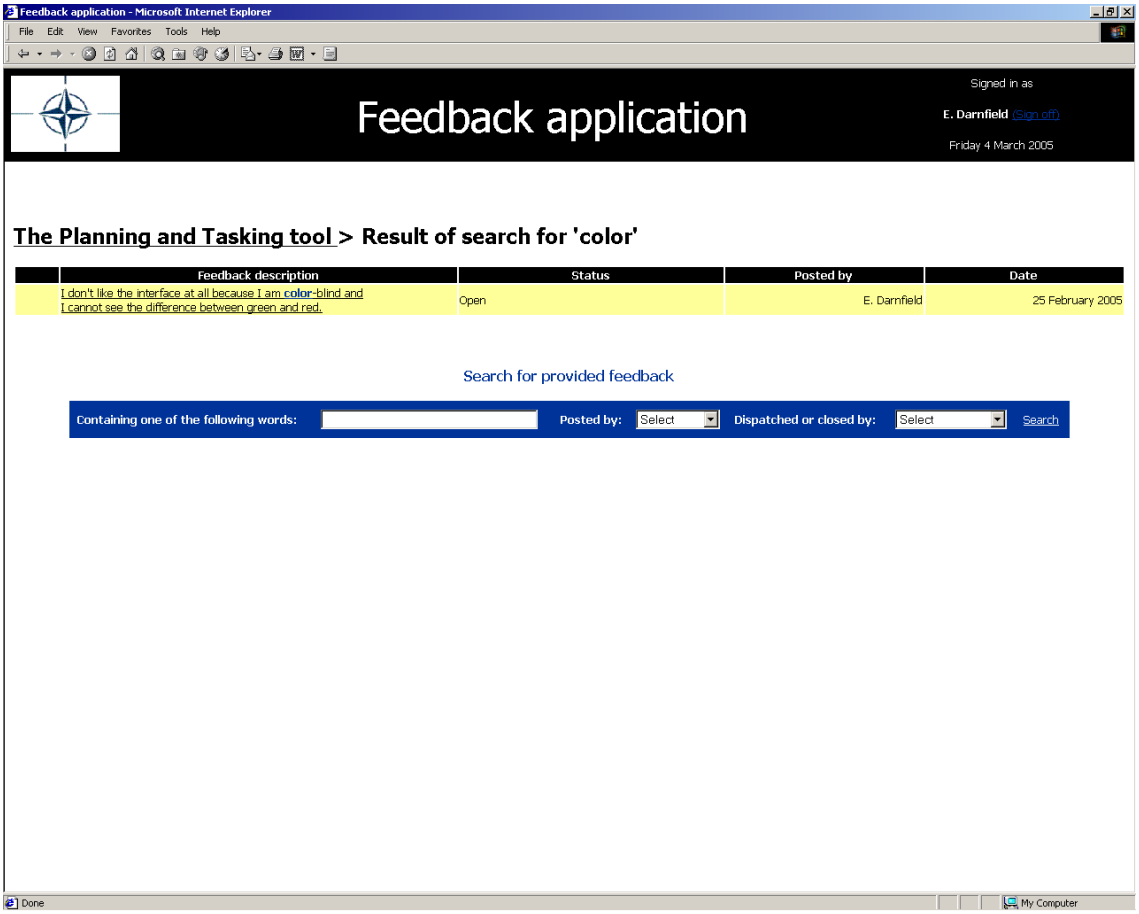


Figure 5.2-8 Search result screen

For example, when the user has searched for feedback that contain the word ‘color’, al the feedback description that contain the word ‘color’ are displayed.

5.3 The NC3A developer view

The second prototype that has been developed is a prototype that is based on the NC3A developer functionality. In this second prototype the NC3A developer functionality is conceptualised. While demonstrating and evaluating the first prototype, a lot of feedback is captured. This feedback will be used in the next phase of this project, the Elaboration phase. Because the second prototype is build after the first one, some of the feedback is used during the building of the second prototype. In this section the second prototype will be discussed.

5.3.1 Signing in

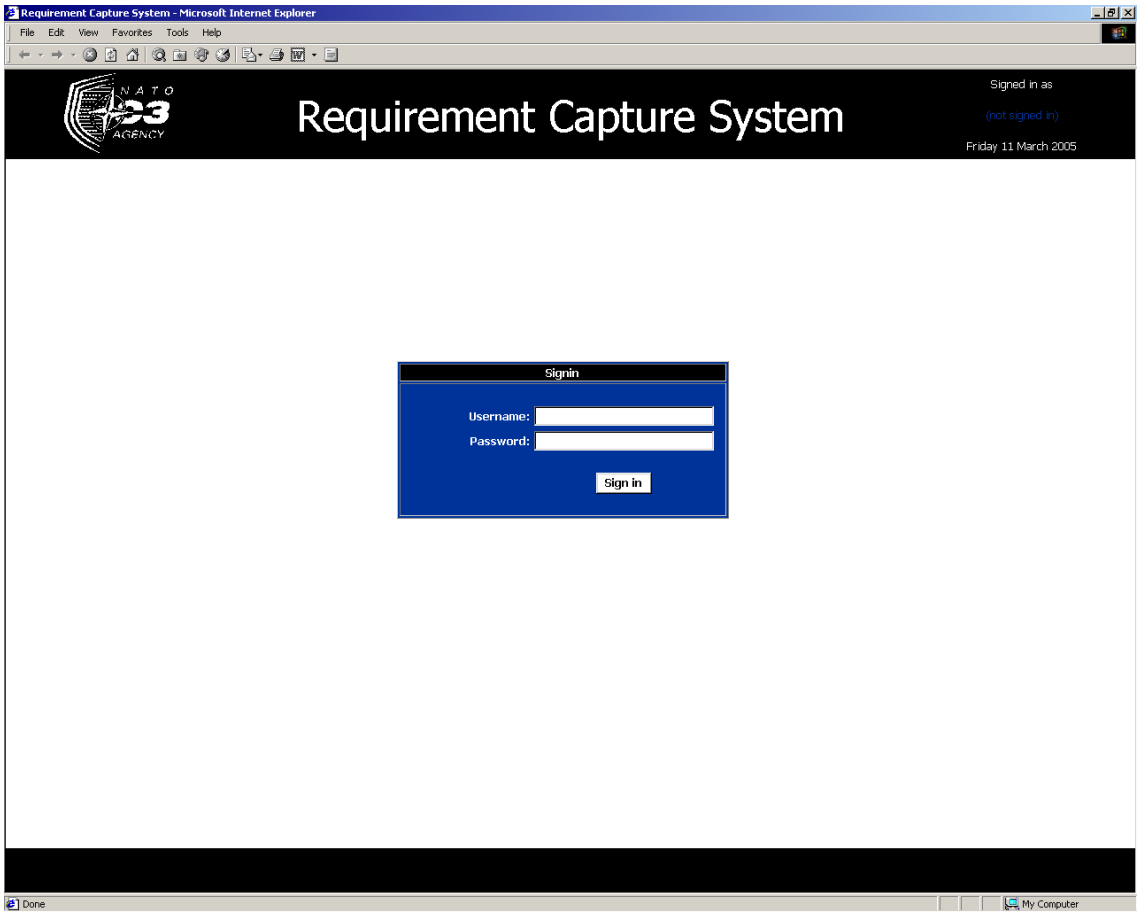


Figure 5.3-1 Sign in screen.

The first difference with the earlier prototype is that there are now three frames instead of two. The new frame is placed at the bottom and will contain buttons for additional functionality when required. The header has slightly changed with a new logo and the name of the application is changed into ‘Requirement Capture System’.

5.3.2 Choosing a prototype

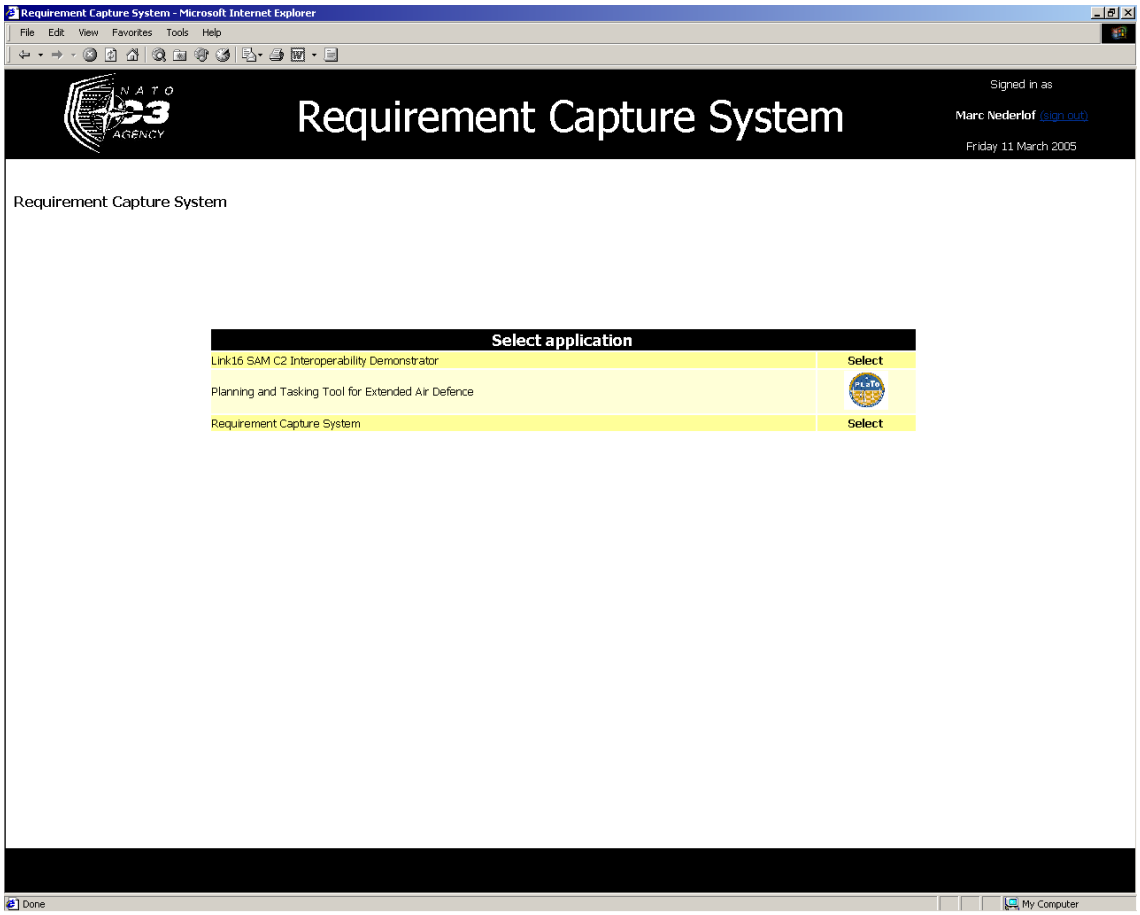


Figure 5.3-2 Prototype screen

When the user has signed in a new screen is displayed. This screen allows the user to choose the software prototype system on which he/she wants to provide or view feedback about.

5.3.3 Modules of a prototype

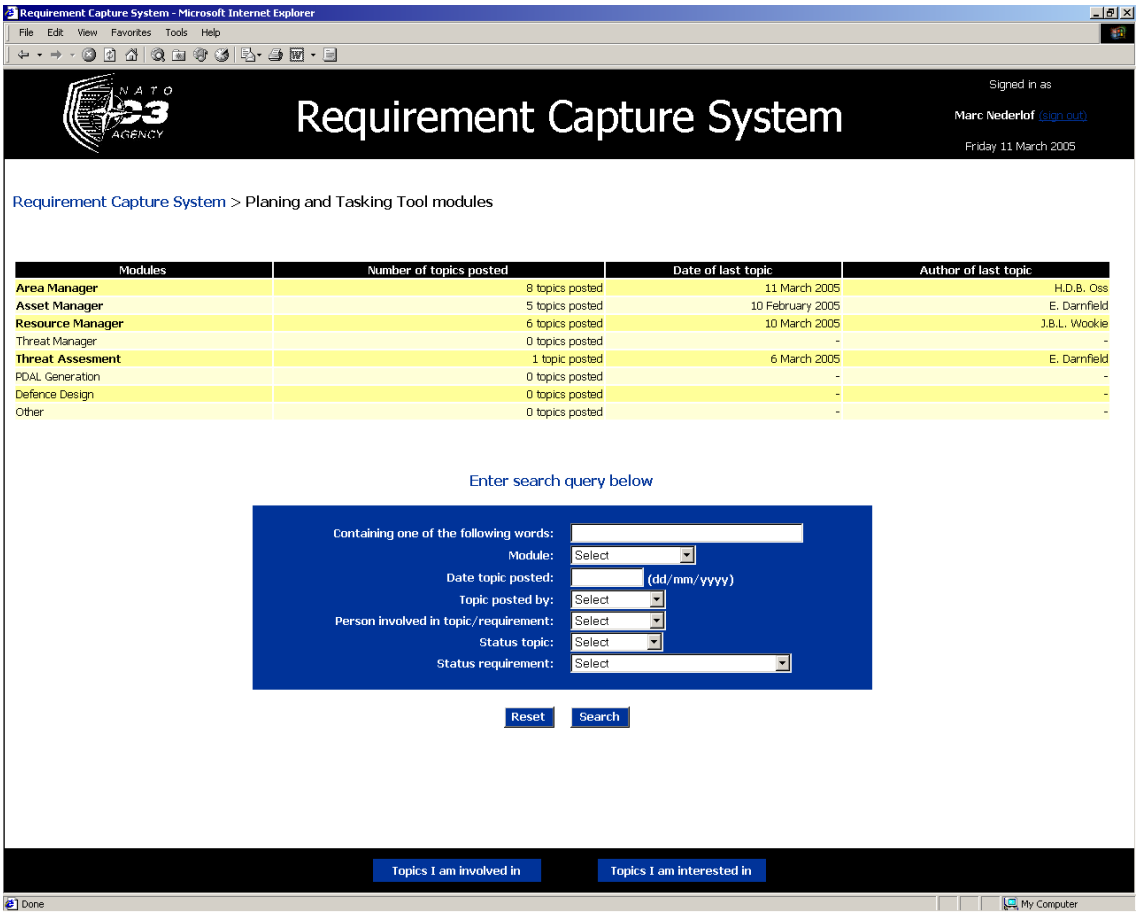
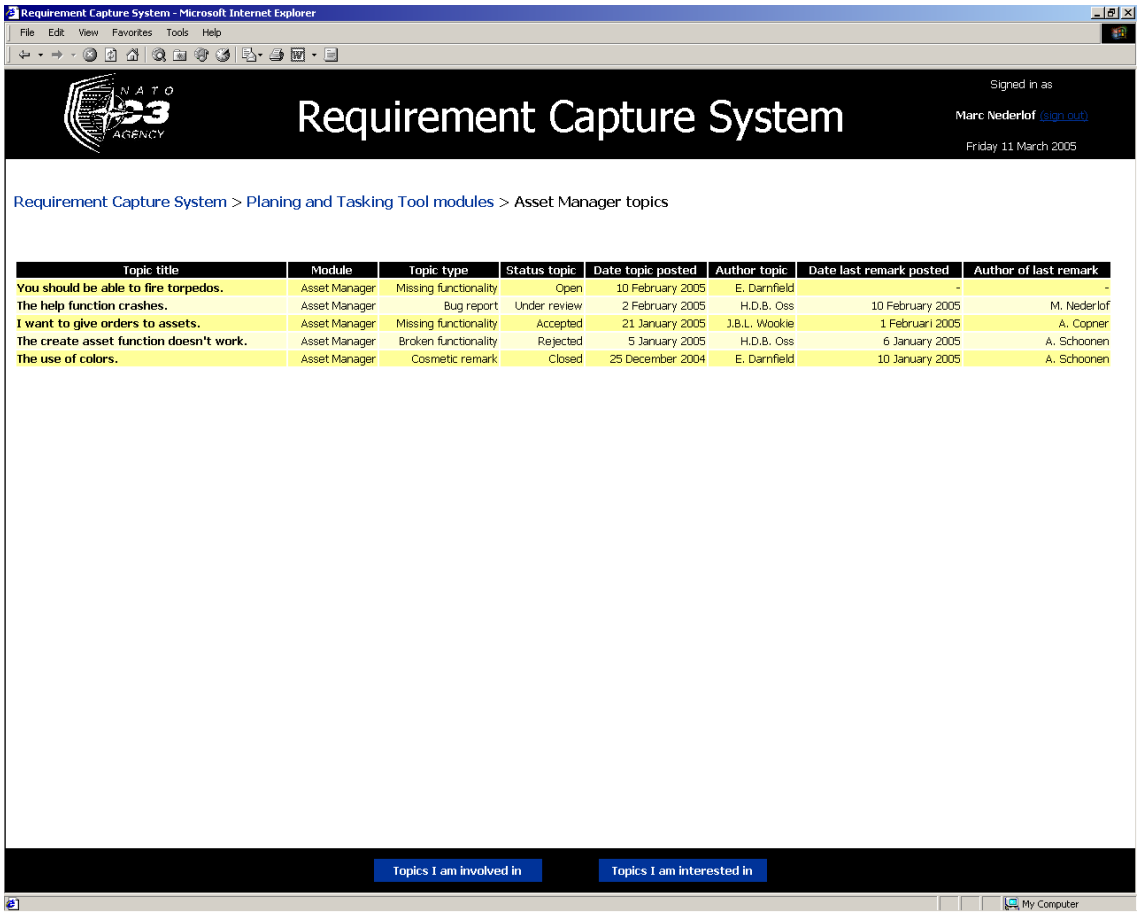


Figure 5.3-3 Module screen

After selecting a prototype the user enter a screen that is somewhat similar to the main screen of the earlier prototype. In this screen the user can choose different modules of the software prototype system. The list of modules contain the module name, the number of topics that have been posted about that module, the date of the last topic that is posted and the name of the author who posted the last topic. The user can now select a module by clicking on it.

The search field has also been altered. There are some new search queries like the module, person who is involved and status requirement.

5.3.4 Topic list



Topic title	Module	Topic type	Status topic	Date topic posted	Author topic	Date last remark posted	Author of last remark
You should be able to fire torpedos.	Asset Manager	Missing functionality	Open	10 February 2005	E. Darnfield	-	-
The help function crashes.	Asset Manager	Bug report	Under review	2 February 2005	H.D.B. Oss	10 February 2005	M. Nederlof
I want to give orders to assets.	Asset Manager	Missing functionality	Accepted	21 January 2005	J.B.L. Wookie	1 Februari 2005	A. Copner
The create asset function doesn't work.	Asset Manager	Broken functionality	Rejected	5 January 2005	H.D.B. Oss	6 January 2005	A. Schoonen
The use of colors.	Asset Manager	Cosmetic remark	Closed	25 December 2004	E. Darnfield	10 January 2005	A. Schoonen

Figure 5.3-4 Topic list screen

When the user has selected a module, the different topics previously created are displayed.
The topic list contains the topic title, the module, the status op the topic, the date the topic was posted, the name of the author who posted the topic, the date when the last remark is posted and the name of the author who posted the last remark. In this screen it is possible to click on the topic title.

5.3.5 Viewing a topic

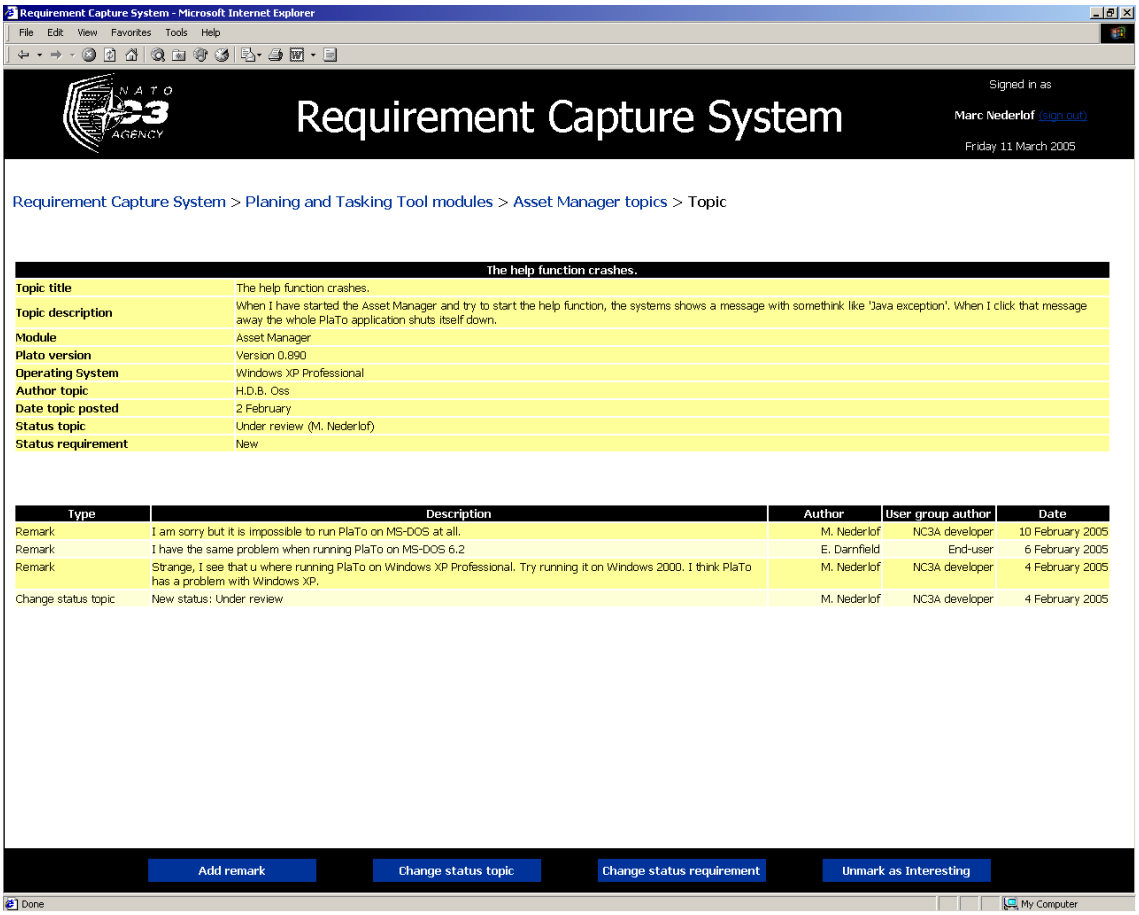


Figure 5.3-5 View topic screen

If the user has clicked on the topic title a new screen will we displayed that shows all the information about the topic including remarks and status changes that have been made.

The information showed about the topic includes a topic title, a topic description, the module, the version of the prototype that it is about, the operation system that is used, the author of the topic, the date the topic was posted, the status op the topic and the status of the requirement. The remarks or status changes contain information as type (can be a remark, a change status topic or change status requirement), a description, the author, the user group to which the author belongs and the date.

On the footer additional buttons are displayed. These are ‘Add remark’, ‘Change status topic’, ‘Change status requirement’ and ‘Mark as interesting’.

5.3.6 Adding a remark

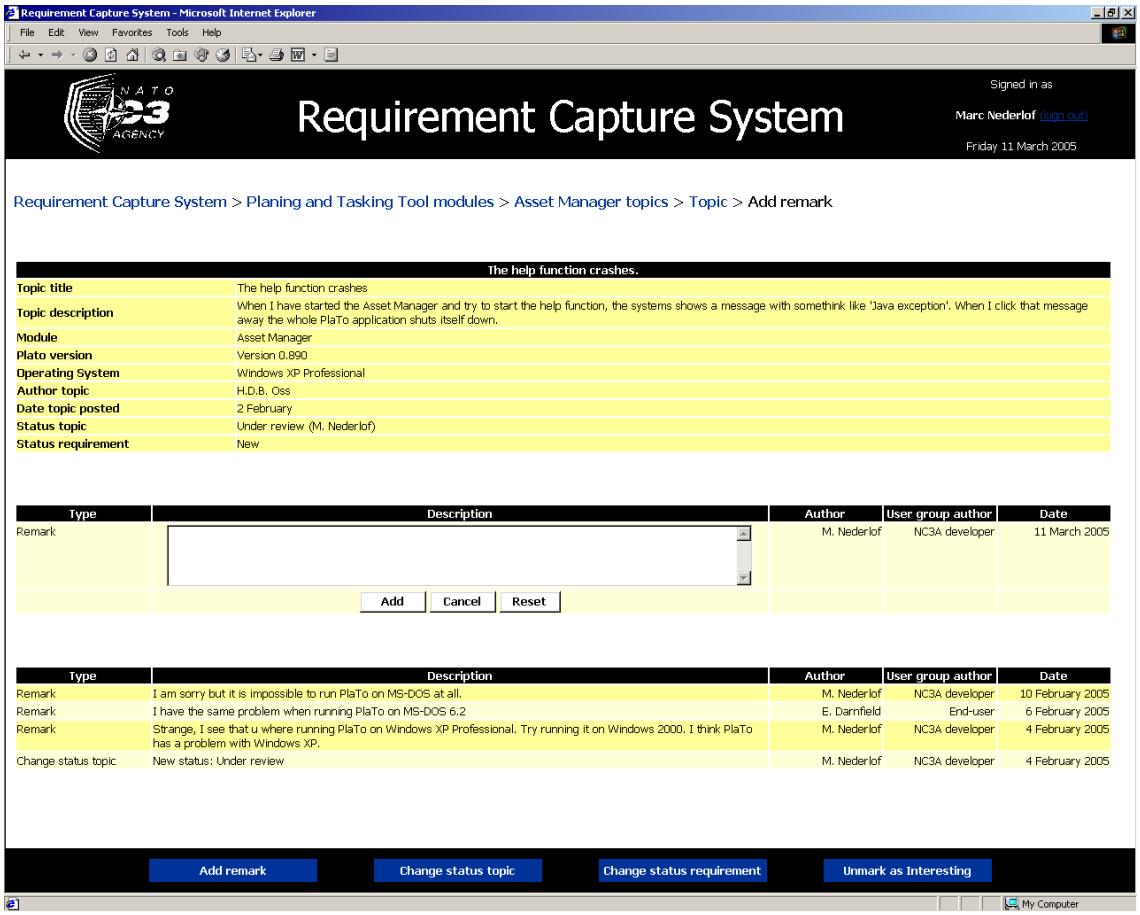


Figure 5.3-6 Add remark screen

When the user clicks on the button ‘Add remark’ there will appear a new text area where the user can enter his remark. Additional information that is already known by system like type, author, user group author and date is showed.

It is possible to clear the text area by clicking on ‘Reset’ and to cancel the remark by clicking on ‘Cancel’. To submit the remark the user can click on ‘Add’.

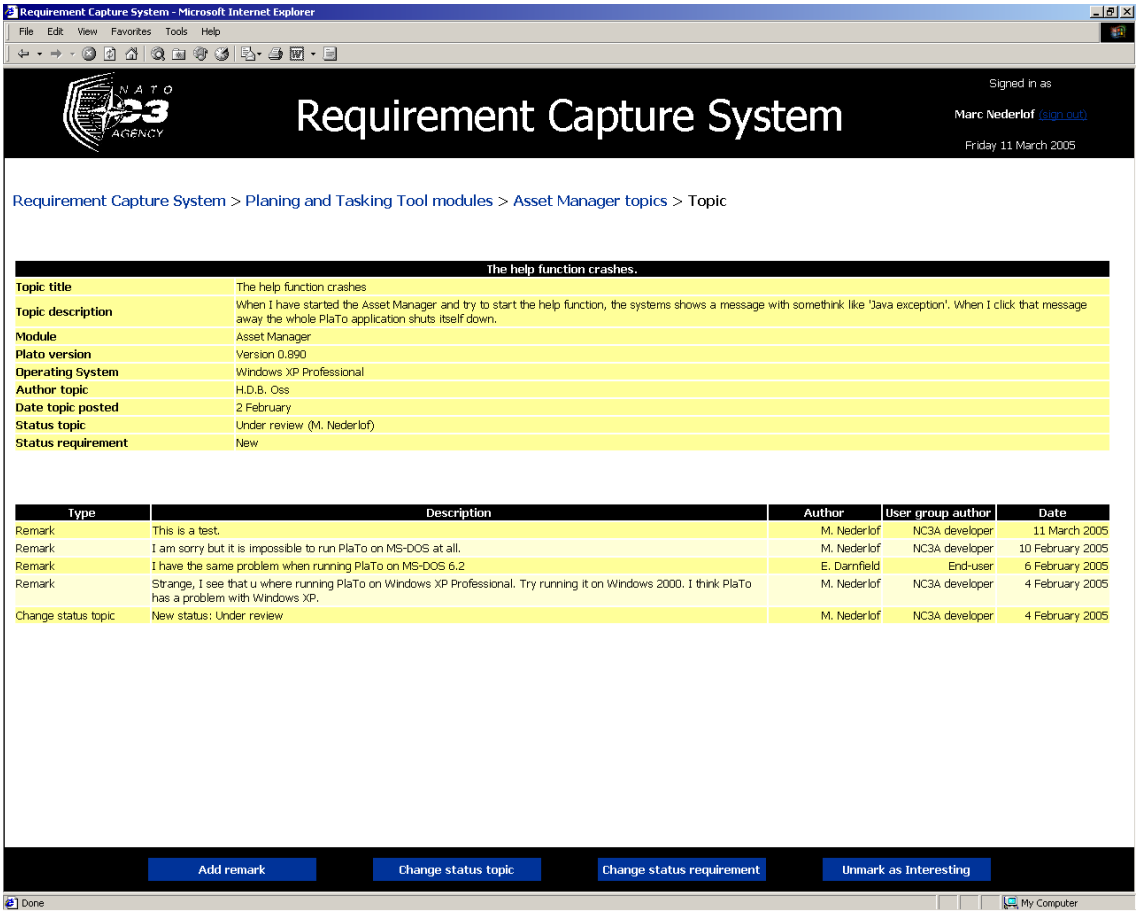


Figure 5.3-7 Remark added screen

When the user has clicked on ‘Add’ to submit a remark, the updated topic system is displayed where the new remarks is also visible.

5.3.7 Changing the status of a topic

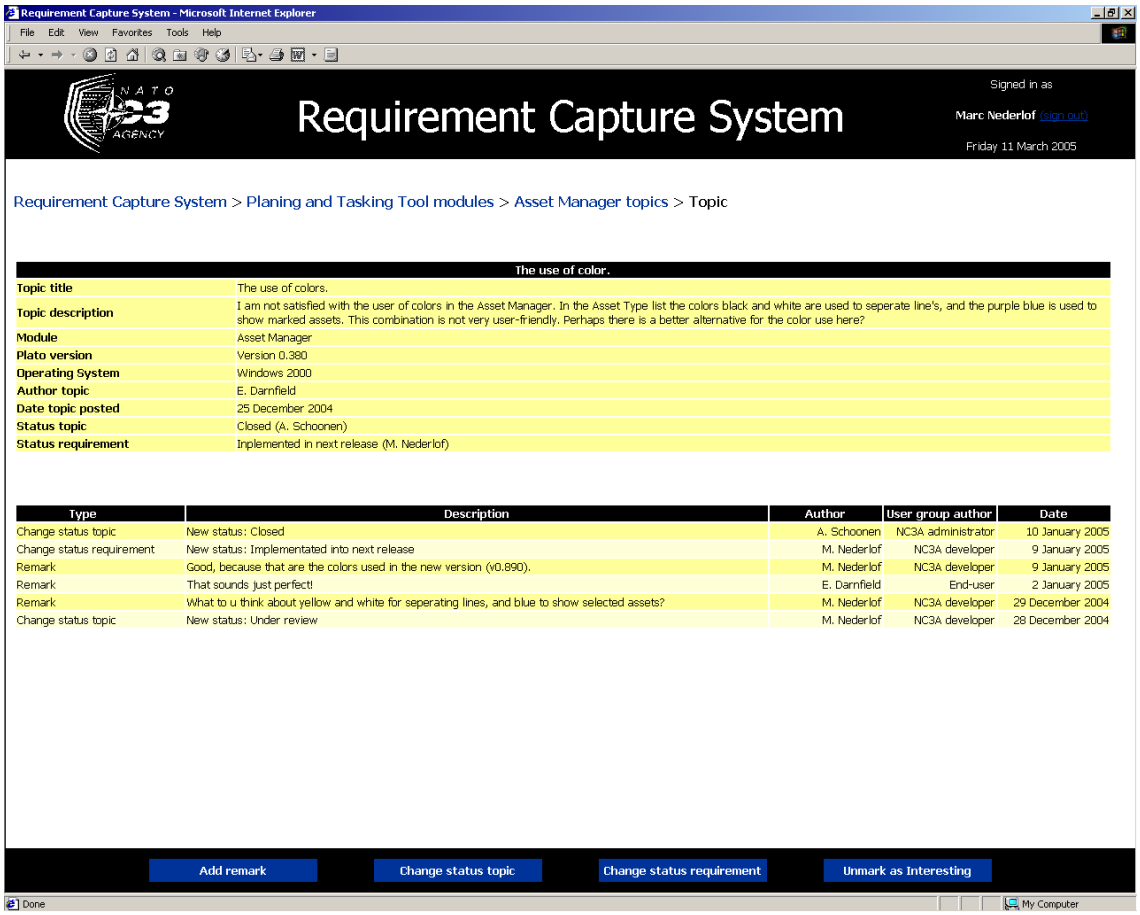


Figure 5.3-8 View topic screen

When viewing a topic in the topic screen, it is also possible to change the status of a topic by clicking on ‘Change status topic’.

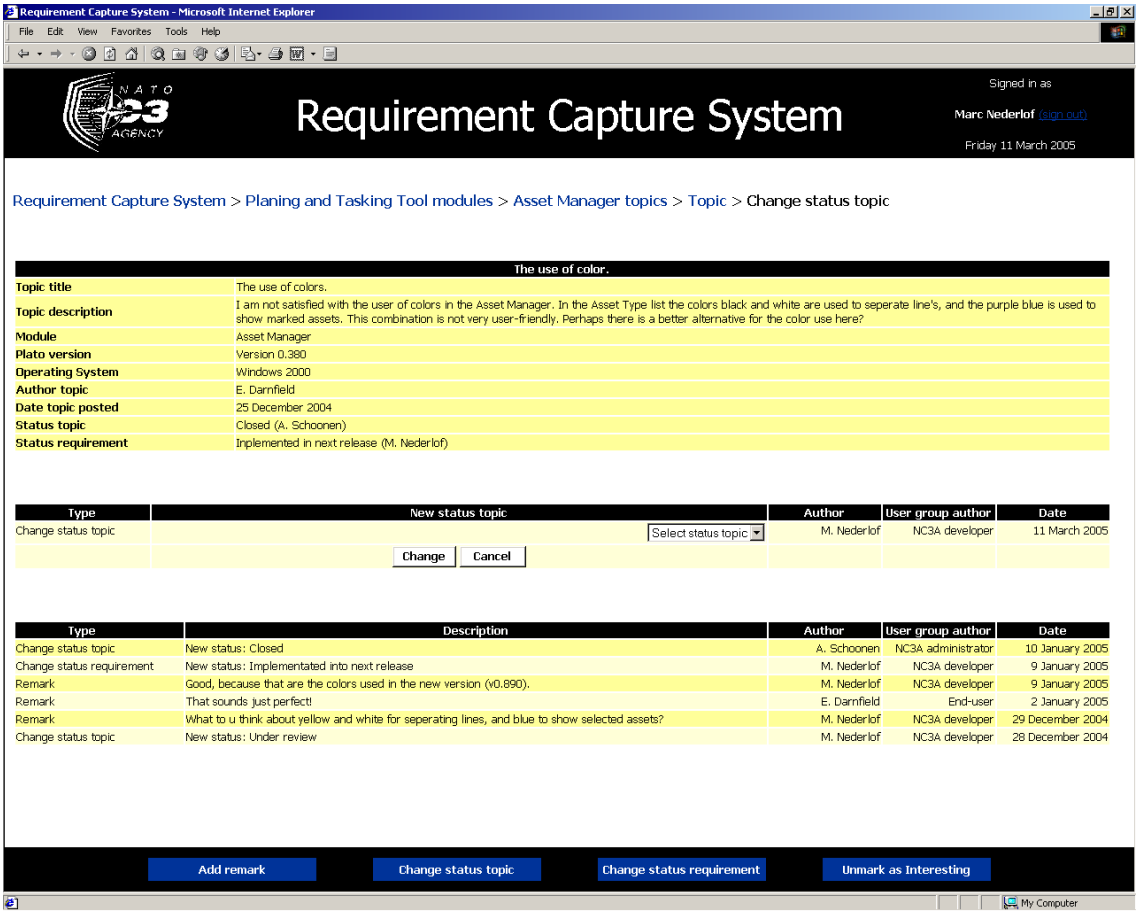


Figure 5.3-9 Change status topic screen

If the user has clicked on ‘Change status topic’ a dropdown list is displayed where the user can select the new status he/she wants to give to the topic. The status can be open, closed, under review, accepted or rejected. Additional information that is already known by the system is also displayed. The user can cancel changing the status by clicking on ‘Cancel’. To submit the change of status the user can click on ‘Change’.

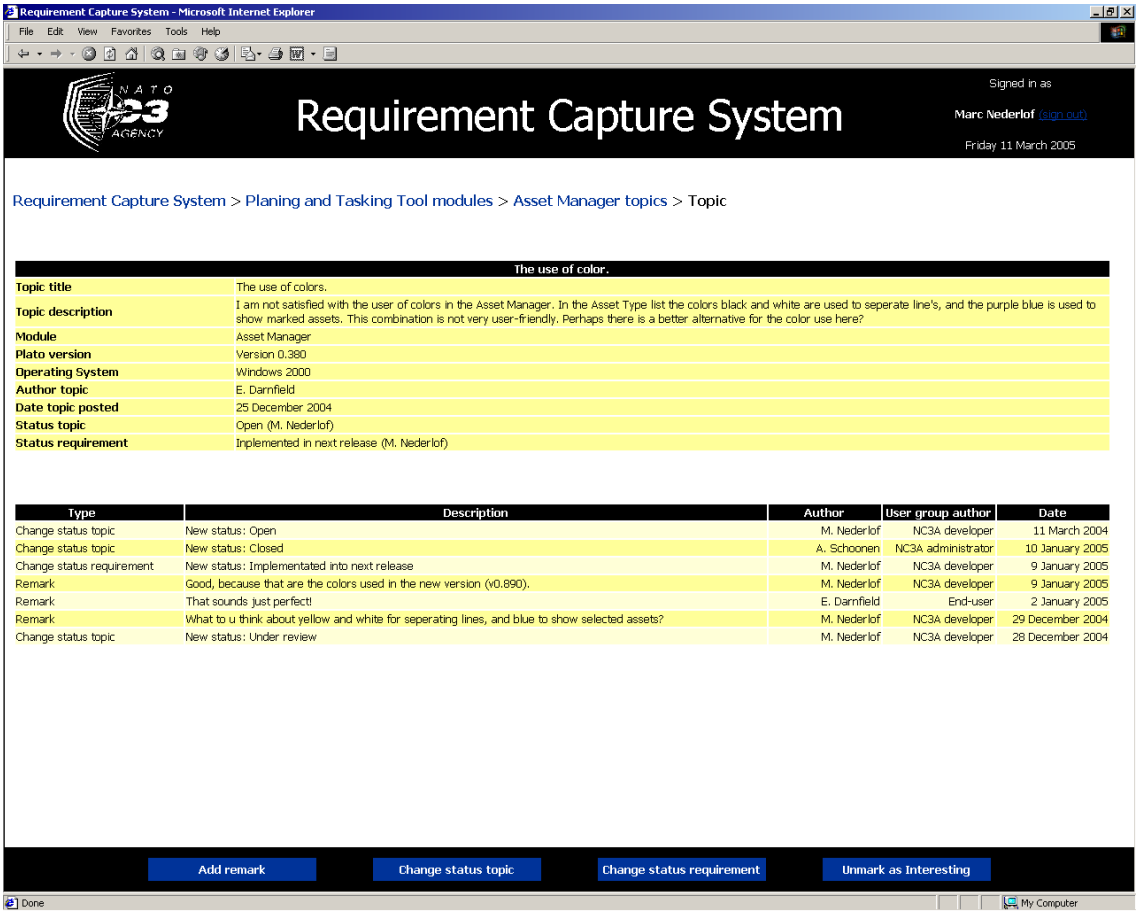


Figure 5.3-10 Status topic changed screen

When the user has submitted the change of status by clicking on ‘Change’, the updated topic screen is displayed where the new status is also visible.

5.3.8 Changing the status of a requirement

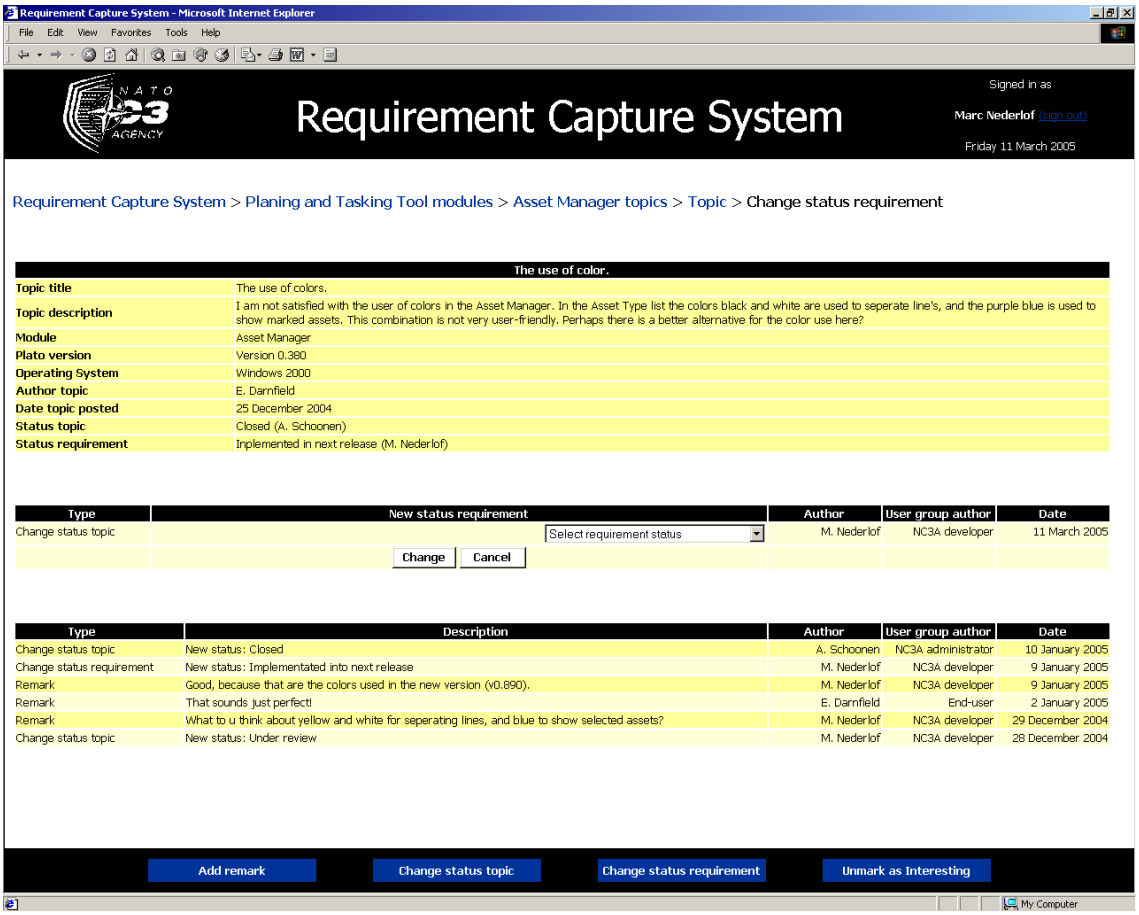


Figure 5.3-11 Change status requirement screen

By clicking on ‘Change status requirement’ in the topic screen, the user can change the status of the requirement. This function works the same as ‘change status topic’. The status of a requirement can be new, accepted but not added to the development cycle, rejected, added to the development cycle or implemented in the next release. The user can submit the status change by clicking on ‘Change’.

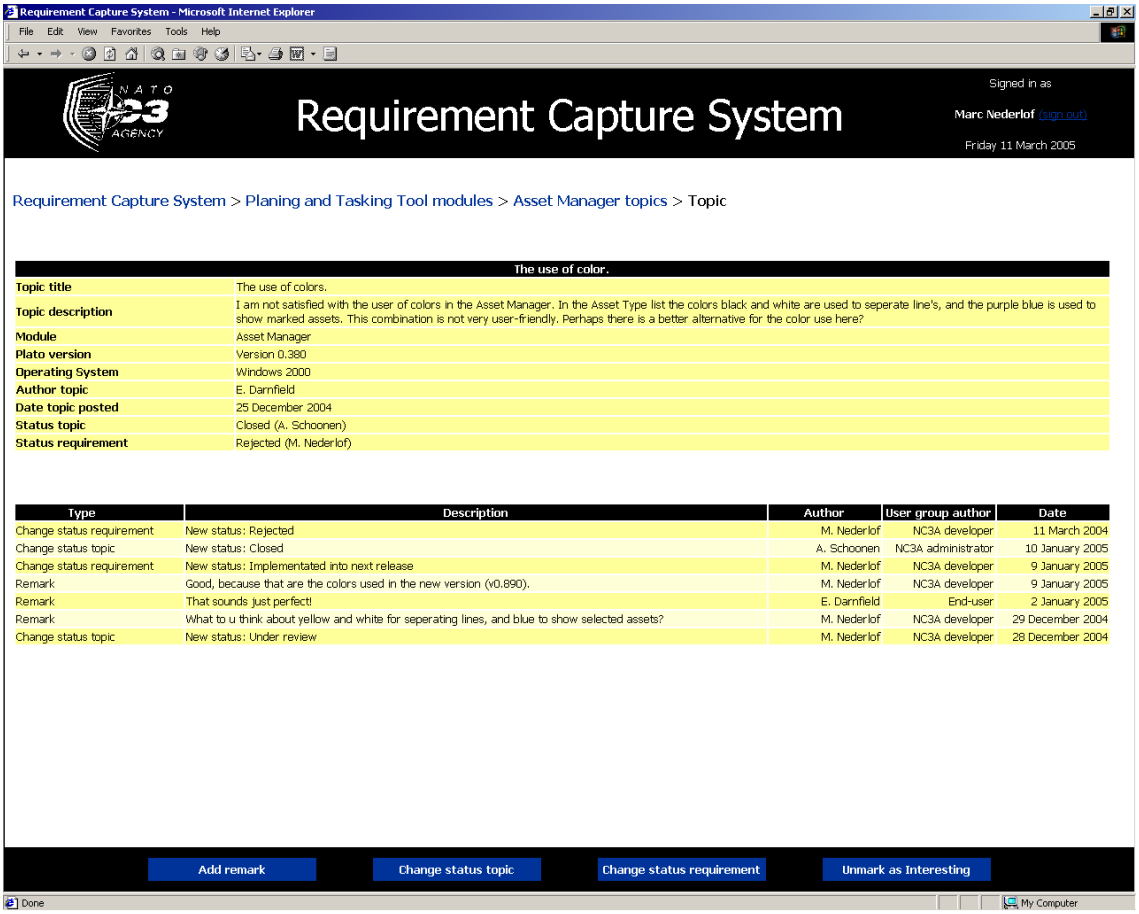
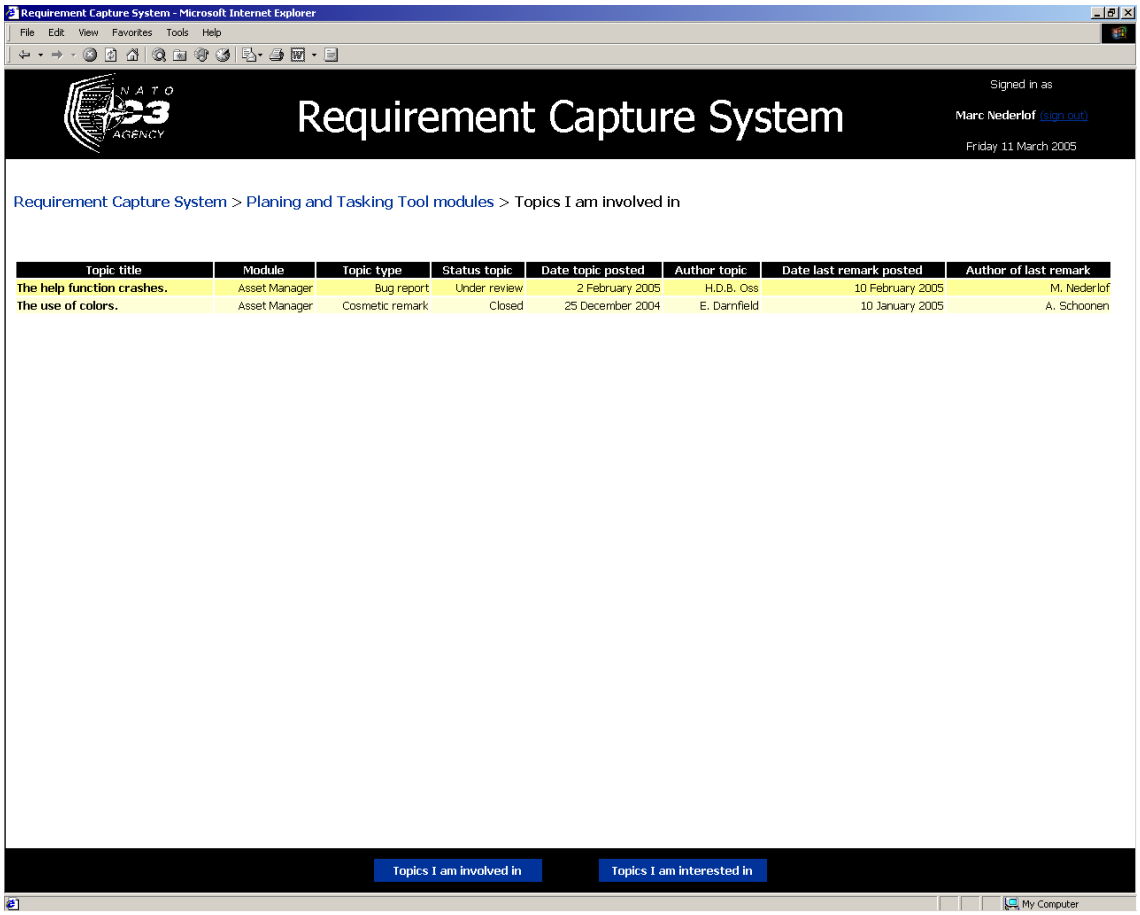


Figure 5.3-12 Status requirement changed screen

When the user has submitted the change of requirement by clicking on ‘Change’, the updated topic screen is displayed and the new status is visible.

5.3.9 Viewing topics involved in

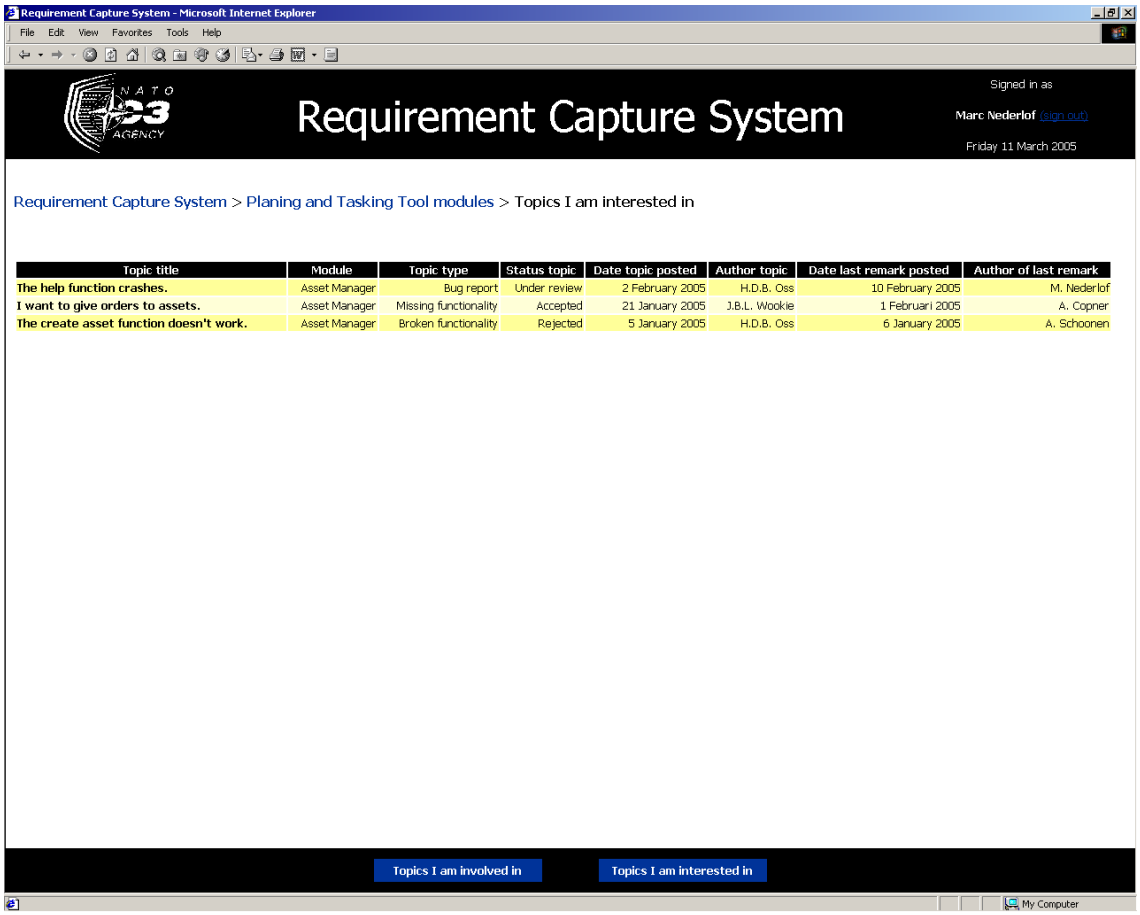


Topic title	Module	Topic type	Status topic	Date topic posted	Author topic	Date last remark posted	Author of last remark
The help function crashes.	Asset Manager	Bug report	Under review	2 February 2005	H.D.B. Oss	10 February 2005	M. Nederlof
The use of colors.	Asset Manager	Cosmetic remark	Closed	25 December 2004	E. Darnfield	10 January 2005	A. Schoonen

Figure 5.3-13 Topic list screen

During the evaluation of the earlier prototype it became clear that there was a need for some additional functionality. In the main screen there are two additional buttons places on the footer named ‘Topics I am involved in’ and Topics I am interested in’. When the user clicks on the button ‘Topics I am involved in’ he/she will get a list of all the topics he/she is involved in. A user is involved in a topic if he/she is the author of the topic, if he/she provided remarks to a topic and when he/she makes a status change to a topic of requirement.

5.3.10 Viewing topics of interest



Topic title	Module	Topic type	Status topic	Date topic posted	Author topic	Date last remark posted	Author of last remark
The help function crashes.	Asset Manager	Bug report	Under review	2 February 2005	H.D.B. Oss	10 February 2005	M. Nederlof
I want to give orders to assets.	Asset Manager	Missing functionality	Accepted	21 January 2005	J.B.L. Wookie	1 Februari 2005	A. Copner
The create asset function doesn't work.	Asset Manager	Broken functionality	Rejected	5 January 2005	H.D.B. Oss	6 January 2005	A. Schoonen

Figure 5.3-14 Topic list screen

When the user clicks on the ‘Topics I am interested in’ button he/she will get a list of topics that are of interest to the user.

5.3.11 Marking a topic as interesting

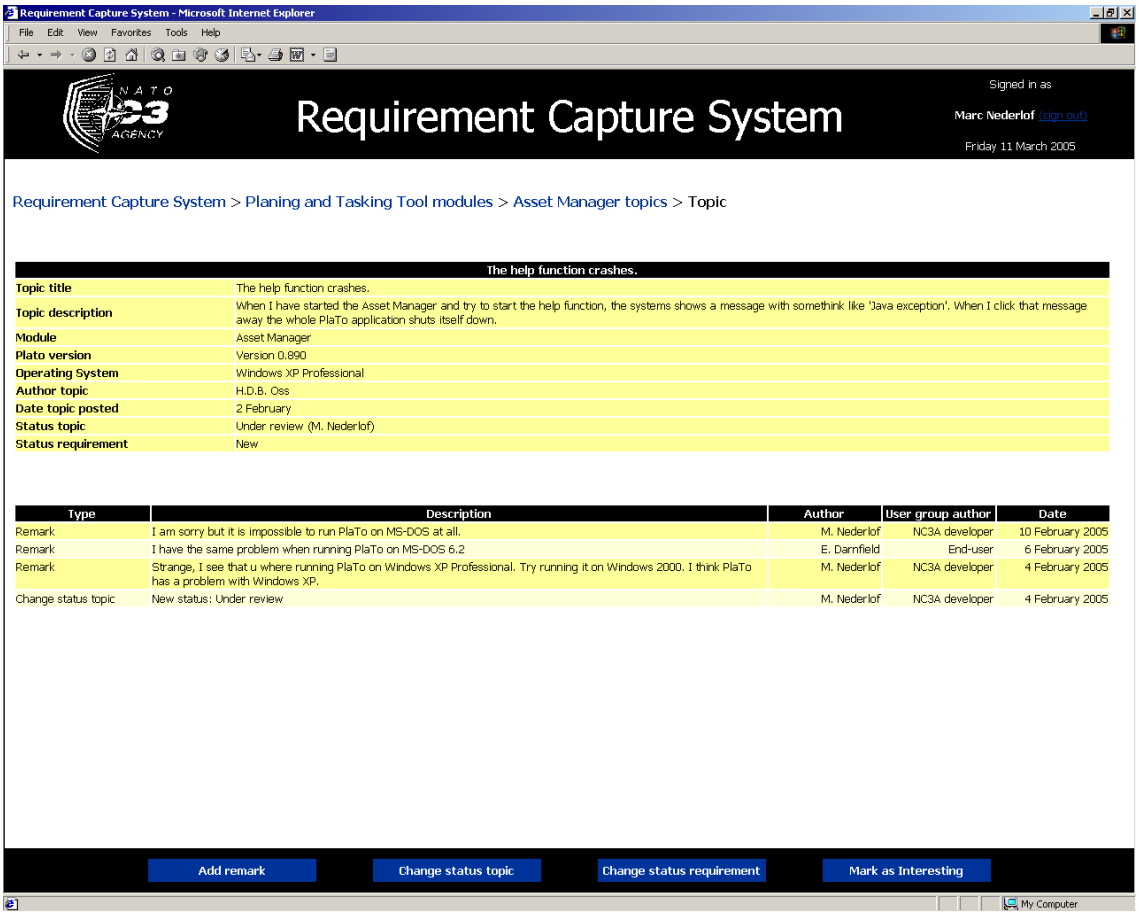


Figure 5.3-15 Topic not marked as interesting screen

When viewing a topic in the topic screen, the user can mark and unmark that topic as interesting by clicking on the right button (Mark as interesting) displayed on the footer. In the figure shown above the topic is at this moment not yet marked as interesting.

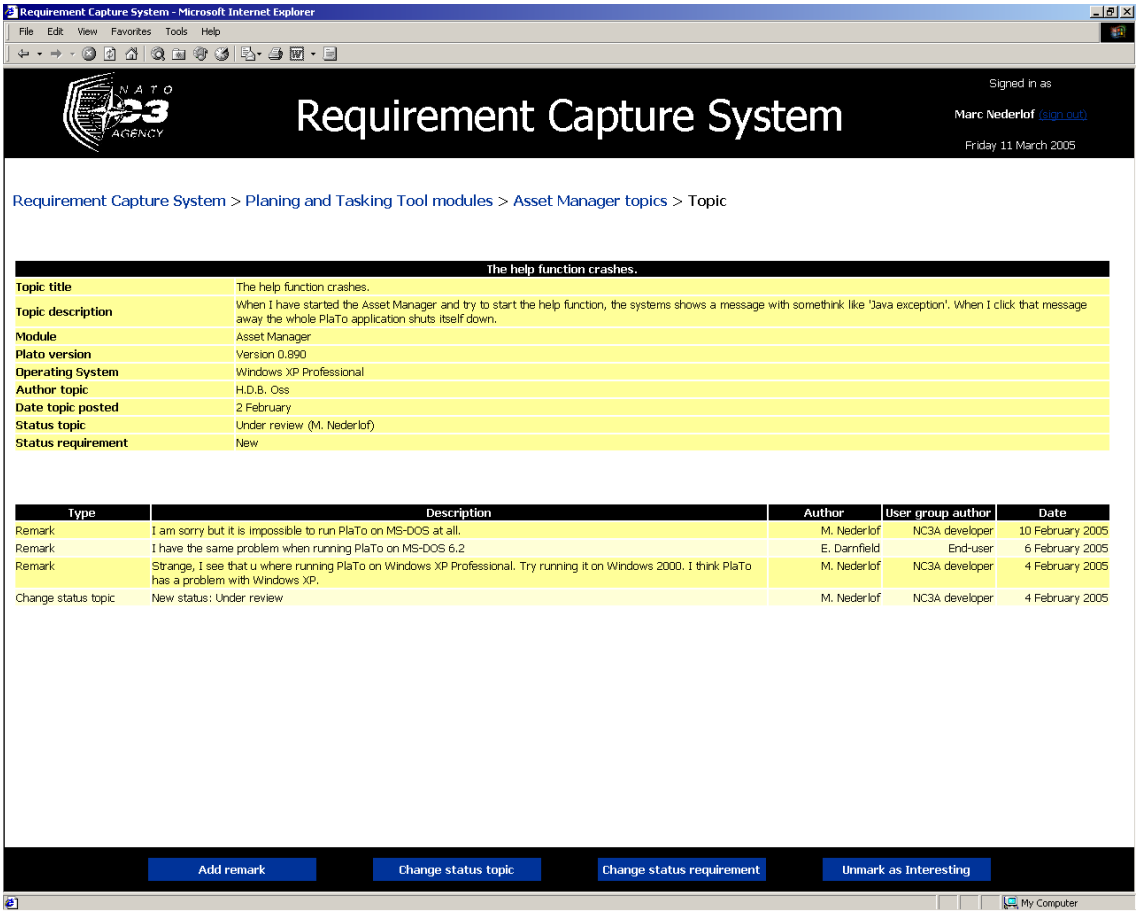


Figure 5.3-16 Topic marked as interesting screen

If the user has clicked on the button ‘Mark as interesting’ in the topic screen the topic will be marked as interesting and the button will change in ‘Unmark as Interesting’. By clicking that button the user can unmark a topic as interesting.

Chapter 6 - Business Case

The Business Case is to describe the economic value of the project. In someway the Business Case will justify carrying out the project. The Business Case should be complete and stable at the end of the Inception phase, but will be refined throughout the project. This chapter will describe the Business Case.

6.1 Costs of the project

The project will incur several costs. There are:

- The student. There will be a compensation for the student who will carry out the project. This will be a fixed amount each day.
- Time from personnel. In order to carry out the project, there has to be a mentor who will guide the student through the organisation and the assignment. Other personnel also have to be available for the student to gain necessary information.
- Facilities. The student will need a working place and the required hardware like a standard Windows PC.
- Software licences. There have to be software licences for the required software. For example: MS Windows, MS Office, MS Visio, MS Project, Rational Rose and IntelliJ IDEA.

6.2 Benefits of the project

The project will also have some benefits. There are:

- Improved development process. The system that will be developed, will allow end users to give feedback on the prototype systems whenever they want. This should cause the NC3A developers to receive feedback more frequently and easier. In theory, the development process of the software prototypes will improve.
- Better acceptation. Because the end users are more involved with the development of the prototypes, the requirements captured with the prototypes should be better accepted. Eventually this should lead to a better acceptation by the operational community of the added requirements and capabilities to Strategic Command Automated Information System (Bi-SC AIS) and ACCS.

6.3 Major risks

While conducting the project, there are certain risks involved like:

- Software licences. One of the risks is software licence for the required software not being available. In that case there should be looked for alternative software as a solution.

- Non-available personnel. The NC3A personal have a very stressed schedule and they often have to go on travel for duty. If the mentor is not available there has to be another point of contact for the student.

- Rejection of the system. The major risk is the rejection of the system by the end users. If they reject the system and will not use it for any reason, all the benefits of the project will be lost.

Chapter 7 - Development Case

In the Development Case is described which products should be produced and when they should be produced. It also describes which tools can be used for support. The Development Case should be refined in each phase of the project. This chapter will describe the Development Case.

7.1 Products

This section will discuss the products that are going to be produced during the development process.

- Vision Document. The Vision Document will describe the statement of work, the different user groups, the key feature the system has to offer and some global non-functional requirements. The Vision Document should be stable at the end of the inception phase but can be refined at later stages.

- Software Architecture Document. The Software Architecture Document will be the overall documentation document, which will contain use case design models, use case implementation models and use case test models. Therefore this document should be evolving till the end of the project.

- The system. In the first phase the system can be just a conceptual prototype. This prototype will then evolve to an executable prototype with functionality in the elaboration phase. In the construction phase there should be a beta release of the system and finally in the transition phase the prototype should have been evolved to the final release of the system.

- Business Case. The Business Case is to describe the economic value of the project. In someway the Business Case will justify carrying out the project. The Business Case should be complete and stable at the end of the Inception phase, but will be refined throughout the project.

- Development Case. In the Development Case is described which products should be produced and when they should be produces. It also describes how the products should be captured. This document should be refined in every phase.

- Major milestones: Each phase ends with a major milestone. Those milestones will contain all documentation, designs and screenshots.

7.2 Tools

This section will describe the tools that can be used for the project. It is possible that not all the tools mentioned in this chapter are going to be used.

- MS Project: MS Project can be used for making Gantt-diagrams.
- MS Visio. MS Visio can be used for developing UML-diagrams, conceptual web sites or web site maps for prototyping, Windows user interfaces for prototyping and database design. MS Visio can also be used for Gantt-diagrams.
- Rational Rose. Rational Rose can be used for producing UML-diagrams, and generating code based on the UML designs.
- MS FrontPage. An executable prototype of a web-based application can be done with MS FrontPage.
- MS Sharepoint: MS Sharepoint can be used for the development of a web-based application or a portal.
- IntelliJ IDEA: IntelliJ IDEA can be used for the development of a Java stand-alone application.

7.3 Planning

In this section the planning for the next phases or the project will be described.

7.3.1 Global planning

Week	Calendar week	Phase	Activity	
6	11 (14-3 t/m 18-3)	Elaboration (iteration 1)	Refine Vision Document (1 day) Design critical use cases (2 days) Design and implement database(2 days)	
7	12 (21-3 t/m 25-3)	Elaboration (iteration 1)	Built an executable prototype (6 days) Test the executable prototype (2 days) Refine Business Case (1 day) Refine Development Case (1 day)	
8	13 (28-3 t/m 1-4)	Elaboration (iteration 1)		
9	14 (4-4 t/m 8-4)	Elaboration (iteration 2)	Refine Vision Document (1 day) Redesign critical use cases (2 days) Redesign and implement database (2 days)	
10	15 (11-4 t/m 15-4)	Elaboration (iteration 2)	Rebuilt an executable prototype (5 days)	
11	16 (18-4 t/m 22-4)	Elaboration (iteration 2)	Test the executable prototype (1 days) Refine Business Case (1 day) Refine Development Case (1 day) Write Lifecycle Architecture Milestone report (2 days)	
12	17 (25-4 t/m 29-4)	Construction (iteration 1)	Refine Vision Document (1 days) Design use cases (4 days)	
13	18 (2-5 t/m 6-5)	Construction (iteration 1)	Build beta release (6 days) Test beta release (2 days) Refine Business Case (1 day) Refine Development Case (1 day)	
14	19 (9-5 t/m 13-5)	Construction (iteration 1)		
15	20 (16-5 t/m 20-5)	Construction (iteration 2)	Refine Vision Document (1 days) Redesign use cases (4 days)	
16	21 (23-5 t/m 27-5)	Construction (iteration 2) / End	Rebuild beta release (5 days)	Write final internship report (15 days)
17	22 (30-5 t/m 3-6)	Construction (iteration 2) / End	Test beta release (1 day) Refine Business Case (1 day) Refine Development Case (1 day) Write Initial Operational Capability Milestone report (2 days)	
18	23 (6-6 t/m 10-6)	End		

Figure 7.3-1 Global planning

7.3.2 Detailed planning

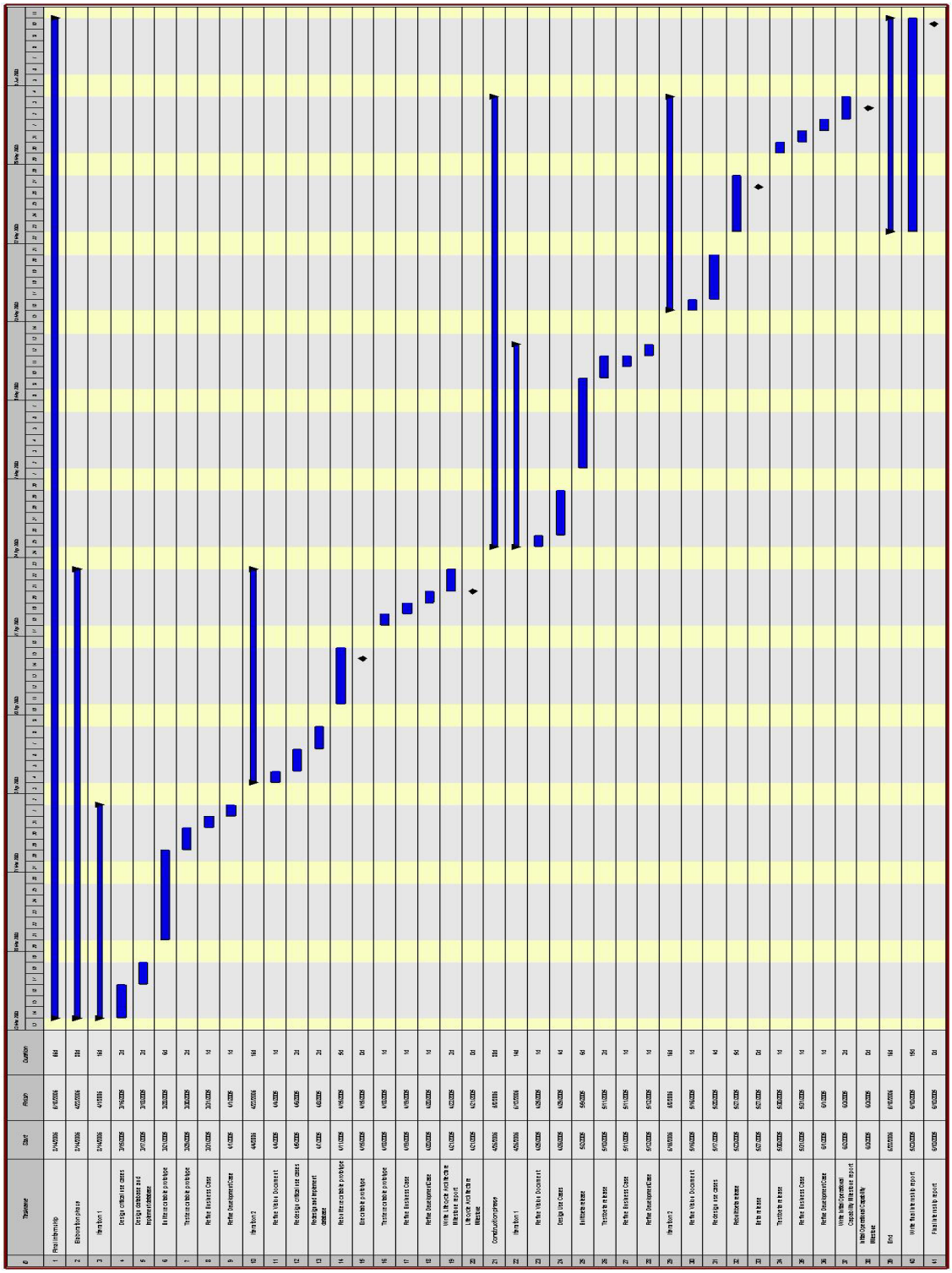


Figure 7.3-2 Detailed planning

Chapter 8 - References

Books

Evans, T., '10 Minute Guide to HTML 4.0, third edition'. Que Corporation, 1998

Kleppe, A. And Warmer, J., 'Praktisch UML, 3de editie'. Soest, Pearson Education Benelux BV., 2004

Kroll, O. and Kruchten, P, 'The Rational Unified Process Made Easy: a practitioner's guide to the RUP'. Boston, Pearson Education, Inc., 2003

Kruchten, P., 'The Rational Unified Process: an introduction, third edition'. Boston, Pearson Education, Inc., 2004

Phillips, L.A., 'Using HTML 4, fourth edition'. Que Corporation, 1998

Quatrani, T., 'Visual Modeling with Rational Rose 2004 and UML'. Boston, Pearson Education, Inc., 2003

Reports

Houten, van, J.J., 'Project Plan – Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Schoonen, A., 'PlaTo Design Document – phase 1'. The Hague, NATO C3 Agency, 2004

Schoonen, A., 'PlaTo User Guide, version 0.890'. The Hague, NATO C3 Agency, 2004

Internet

http://www.nato.int

http://www.nc3a.nato.int

Intranet

http://nc3a-intranet/

Glossary

ACC	Air Component Command
ACCS	Air Command and Control System
ActD	Active Defence
ASP	Active Server Pages
ASPX	Active Server Pages .NET
Bi-SC AIS	Strategic Command Automatic Information System
BMC3I	Battle Management, Command, Control, Communications and Intelligence
C2RC	Command and Control Resource Centre
C2	Command and Control
C3	Command, Control and Communications
CCF	Conventional Counter Force
CCSD	Command and Control System Division
CRONOS	Crises Response Operational NATO Open Systems
EAD	Extended Air Defence
GUI	Graphical User Interface
HHS	Haagse Hogeschool
HQ ARRC	Head Quarters Allied Rapid Reaction Corps
HQ EADTF	Head Quarters Extended Air Defence Task Force
IT&E	Integrated Test & Evaluation
IVIT	Informatievoorzieningen en Informatietechnologie
JSP	Java Server Pages
LSID	Link16 SAM C2 Interoperability Demonstrator
M&S	Modelling and Simulation
MD	Missile Defence
MS	Microsoft
MySQL	My Structured Query Language
NATO	North Atlantic Treaty Organisation

NC3A	NATO Consultation, Command & Control Agency
PC	Personal Computer
PHP	PHP Hypertext Preprocessor
PD	Passive Defence
PlaTo	Planning and Tasking Tool
RUP	Rational Unified Process
SAM	Surface to Air Missile
SQL	Structured Query Language
TMD	Theatre Missile Defence
UML	Unified Modeling Language
VBScript	Visual Basic Script

Table of figures

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Afstudeerverslag externe bijlage III

Ontwikkeling van een Requirement Capture System voor NAVO software prototype systemen

Externe bijlage III
Lifecycle Architecture Milestone

10 juni 2005

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Lifecycle Architecture Milestone

Development of a Requirement Capture System for NATO software prototype systems

by
J.J. van Houten

22 April 2005

Foreword

This report is the Lifecycle Architecture Milestone of my final internship assignment 'Development of a Requirement Capture System for NATO software prototype systems' at the NATO Consultation, Command & Control Agency (NC3A) in The Hague. This final internship is part of the course 'Informatievoorzieningen en Informatietechnologie' (IVIT) that I am following at the Haagse Hogeschool (HHS) in The Hague.

The Lifecycle Architecture Milestone is next to the Project Plan and the Lifecycle Objective Milestone the third milestone of the project and is part of the Rational Unified Process (RUP) developing method.

Readers that are especially interested in the requirements of the system will be referred to chapter three where the required functionality is being described by use cases. This chapter also describes the database design. The executable prototype that is based on the critical use cases is discussed in chapter five.

Again I want to thank the NC3A personnel for their contributions to this document.

22 April 2005, The Hague

J.J van Houten

Table of contents

Chapter 1 - Introduction

The goal of the elaboration phase is to define and baseline the architecture of the system in order to provide a stable basis for the system. The requirements captured in the inception phase will be refined and an executable prototype will be developed. This prototype will be based on the most significant requirements. The elaboration phase is the second phase of the developing method the Rational Unified Process (RUP). This phase ends with the Lifecycle Architecture Milestone that can be captured in a report.

The purpose of this report is to describe the requirements, the database design and the executable prototype in the elaboration phase of the project. This report will also contain an adapted planning for the next phases of the project.

In chapter two the vision of the project will be described. Chapter three, Software Architecture, will describe the requirements and the design of the database. The chosen solution for implementing the system and possible solutions for developing the system is discussed in chapter four. In chapter five the executable prototypes that has been built to capture requirements is discussed. The Business Case justifies carrying out the project is described in chapter six. Finally the Development Case in described in chapter seven.

Chapter 2 - Vision

The Vision Document is to create a foundation for common understanding of motivation for building the system, as well as a high-level definition of the system to be built. The Statement of Work is somewhat analogous to parts of the Vision Document. The Vision Document should be complete and stable at the end of the Inception phase, but will be refined throughout the project. This chapter will describe the Vision Document.

2.1 Statement of work

2.1.1 General

The intern assignment, described in this program of work will be conducted for the Command and Control Resource Centre (C2RC) of the NATO Consultation, Command & Control Agency (NATO C3 Agency) in The Hague. The NATO C3 Agency (NC3A) develops, procures and implements state of the art Command, Control and Communications (C3) capabilities for NATO and it provides unbiased scientific advice and support to NATO authorities. NC3A is an integrated team of more than 500 professionals, civilian and military, from NATO member nations dedicated to provide to NATO high quality expertise in the domain of Command, Control and Communications.

The Missile Defence (MD) mission area consists of four functional areas. These are:

- Conventional Counter Force (CCF)
- Active Defence (ActD)
- Passive Defence (PD)
- Battle Management, Command, Control, Communications and Intelligence (BMC3I)

In support of these four functional areas, C2RC provides core competences in the following areas:

- Concept Development Support (Prototyping, Exercises)
- Architecture and Architecture Development
- Modelling & Simulation (M&S)
- Integrated Test & Evaluation (IT&E)
- Systems: MD Weapon, Sensors and BMC3 Systems
- MD Decision Support Tools

2.1.2 Problem description

Amongst other things, NC3A is developing two missile defence related software prototypes:

- Link16 SAM C2 Interoperability Demonstrator (LSID) for Theatre Missile Defence (TMD) Engagement Coordination and Situational Awareness.
- Extended Air Defence (EAD) Planning and Tasking Tool (PlaTo) for TMD planning.

The two software prototypes described above have been used operationally in several NATO and multinational exercises. Furthermore the systems have been installed in a number of locations. The goal for this installation and exercise participation is to capture user requirements and feedback on the prototypes. The resulting prototypes and requirements captured through this process will at a later stage be used to extend the two NATO Command and Control (C2) systems: the Strategic Command Automated Information System (Bi-SC AIS) and the Air Command and Control System (ACCS) with Battle Management and Engagement Coordination functionality.

At the moment, capturing of these user requirements can only be done when NC3A personnel are on site, and the tracking of user requirements is not done in a structured way. Operators might report user requirements back through mechanisms such as email but there is no proper vehicle for them to actually log remarks, requirements and feedback in a logical standardised way and provide them to NC3A.

Feedback received from users is not collected in a central repository and can contain incomplete or misleading information. This current situation does not sufficiently meet NC3A's requirements.

2.1.3 Goal of the assignment

The goal of this assignment is to develop a system that will allow the operators to log remarks, requirements and feedback through an efficient and structured process.

A process has to be constructed that will facilitate the reporting of user feedback and requirements with respect to the two prototypes to NC3A. Ideally the information provided bye the users should be tied into the development process of the named prototypes. The process has to be scalable and should be defined in a clear and structured way. Once the feedback process has been properly worked out and approved, a system has to be implemented to support it.

2.2 Target users

This section will describe the different user groups of the system. Roughly there are two user groups. These are the developers of the software prototype systems and the users of the software prototype systems.

2.2.1 Developer user group

The developer user group is part of C2RC, which belongs to the Command and Control System Division (CCSD) at NC3A.

At his moment, the developer user group exists of four staff member and three contractors.
Two staff members are working on LSID and three contractors are working on PlaTo. The Senior Scientist is in charge of the development of LSID and PlaTo. LSID and PlaTo are part of the Project TM, which is lead by the Principal Scientist. The next table shows additional information about the developer user group.

Name	Function	Room	E-mail	Phone
A. Schoonen	In charge of development	470	Andre.Schoonen@nc3a.nato.int	070-3743773
C. Allmon	Project Manager TM	359A	Charlie.Allmon@nc3a.nato.int	070-3743772
T. Bingham	Programmer on LSID	359B	Theo.Bingham@nc3a.nato.int	070-3743775
M. Bugru	Programmer on LSID	467	Metin.Bugru@nc3a.nato.int	070-3743812 (TMD Lab)
M. Nederlof	Programmer on PlaTo	467	Marc.Nederlof@nc3a.nato.int	070-3743812 (TMD Lab)
E. van der Koogh	Programmer on PlaTo	467	Erwin.van.der.Koogh@nc3a.nato.int	070-3743812 (TMD Lab)
A. Copner	Programmer on PlaTo	467	Alan.Copner@nc3a.nato.int	070-3743812 (TMD Lab)

Figure 2.2-1 Developer user group

2.2.2 Test site user group

At this moment the users of PlaTo are limited to the Head Quarters Extended Air Defence Task Force (HQ EADTF) and the Head Quarters Allied Rapid Reaction Corps (HQ ARRC). It is possible that PlaTo is going to be used at the Regional Command (RC) South in Naples (Italia) and the Air Component Command (ACC) South in Izmir (Turkey). There are no users for LSID at this moment.

2.3 System activities

This section describes the system activities by using features and key use cases.

2.3.1 System features

Feature No.1	
Name	Provide feedback.
Description	Allow Test site users and Developers to provide feedback on the software prototype systems LSID and PlaTo.
Value	1.
Complexity	Low.
Priority	1.

Figure 2.3-1 Feature No. 1 Provide feedback

Feature No.2	
Name	View feedback
Description	Allow Test site users and Developers to view all feedback provided on the software prototype systems LSID and PlaTo.
Value	2.
Complexity	Low.
Priority	2.

Figure 2.3-2 Feature No.2 View feedback

Feature No.3	
Name	Change feedback status
Description	Allow Developers to change the status of provided feedback on the software prototype systems LSID and PlaTo.
Value	3.
Complexity	High.
Priority	3.

Figure 2.3-3 Feature No.3 Change feedback status

Feature No.4	
Name	Generate requirement
Description	Allow Developers to generate a requirement based on feedback provided on the software prototype systems LSID and PlaTo.
Value	1.
Complexity	Medium.
Priority	4.

Figure 2.3-4 Feature No.4 Generate requirement

Feature No.5	
Name	Change requirement status
Description	Allow Developers to change the status of a generated requirement.
Value	4.
Complexity	High.
Priority	4.

Figure 2.3-5 Feature No.5 Change requirement status

2.3.2 Key use cases

According to the system features discussed in section 2.3.1 the following key actors can be identified: Test site user and Developer. The following key use cases can be identified: Provide feedback, View feedback, Change feedback status, Generate requirement and Change requirement status. The following figures will show these key use cases.

Key use case No. 1	
Name	Provide feedback.
Assumption	The user has signed in.
Actors	Test site user, Developer.
Description	(1) The user tells the system that he/she want to enter feedback. (2) The system asks the user on which prototype he wants to gives feedback. (3) The user answers that question and (4) the system will ask the user for his feedback. The user enters his/her feedback and tells the system that he/she wants to submit it. (5) The system will confirm that the feedback has been submitted.
Exceptions.	None.
Result	Feedback is added.

Figure 2.3-6 Key use case No.1 Provide feedback

Key use case No. 2	
Name	View feedback.
Assumption	The user has singed in.
Actors	Test site user, Developer.
Description	(1) The user tells the system that he/she wants to view feedback. (2) The system shows the user the requested feedback.
Exceptions.	[No feedback found.] The system will tell the user that there is no feedback found.
Result	Feedback is displayed.

Figure 2.3-7 Key use case No.2 View feedback

Key use case No. 3	
Name	Change feedback status.
Assumption	The user has signed in as Developer and has selected feedback.
Actors	Developer.
Description	(1) The user tells the system that he/she wants to change the status of the feedback. (2) The system asks to user to what status he/she wants to change the status of the feedback and (3) the user answers that question.
Exceptions.	None.
Result	Feedback status is changed.

Figure 2.3-8 Key use case No.3 Change feedback status

Key use case No. 4	
Name	Generate requirement.
Assumption	The user has signed in as Developer and has selected feedback.
Actors	Developer
Description	(1) The user tells the system that he/she wants to generate a requirement based on the selected feedback. (2) The system asks the user for the requirement and (3) the user provides the requirement.
Exceptions.	None.
Result	Requirement is generated.

Figure 2.3-9 Key use case No.4 Generate requirement

Key use case No. 4	
Name	Change requirement status.
Assumption	The user has signed in as developer and has selected a requirement.
Actors	Developer.
Description	(1) The user tells the system that he/she wants to change the status of the selected requirement. (2) The system asks the user to what status he/she wants to change the requirement and (3) the user answers that question.
Exceptions.	None.
Result	Requirement status is changed.

Figure 2.3-10 Key use case No.4 Change requirement status

2.3.3 Use case diagram

The following figure shows the use case diagram based on the key use cases previously described.

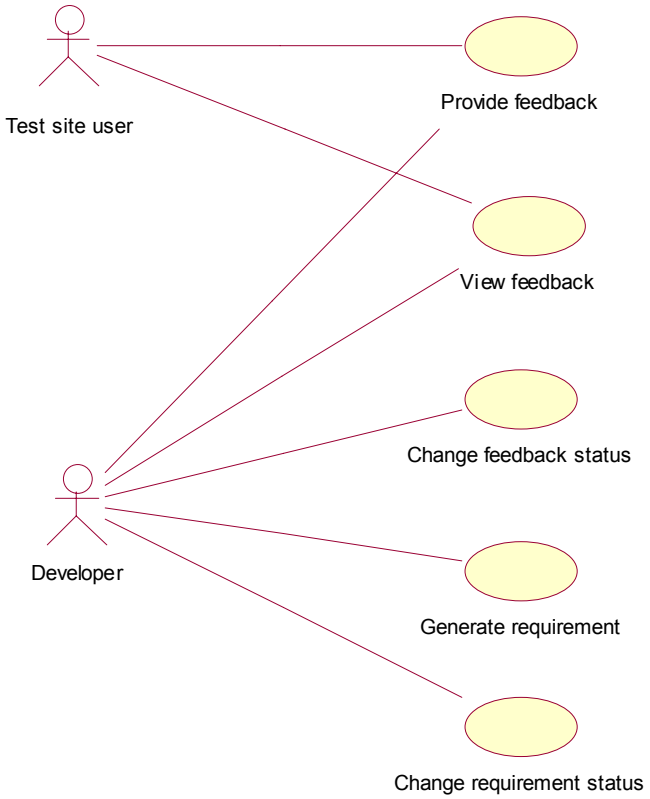


Figure 2.3-11 Use case diagram

2.4 Non-functional requirements

This section briefly describes the non-functional requirements.

- Operation System. The system has to be compatible with Windows 2000 and preferably also with Unix Solaris.

- Database support. There are no specific requirements for choosing the database environment to use. It probably would be SQL Sever 2000 or MS Access.

- Programming environment. It is now clear that there has to be a web-based application. Preferably the web application has to be developed with SharePoint Portal Server (SPS) or Windows SharePoint Services (WSS) that produces ASP.NET pages (ASPX). It is also possible to build a web application using ASP.NET with for example Visual Studio .NET as programming environment. ASP.NET is a Windows based and unfortunately is not compatible with Unix Solaris.

Chapter 3 - Software Architecture

The Software Architecture Document is the overall documentation document that will contain all UML analysis and design models. This document will be refined in every iteration and phase during the project. This chapter describes the Software Architecture Document.

3.1 'Mile-wide, inch-deep' description

There is a need to provide a good understanding of the scope of the system, without going into too much in detail. This is called the 'mile-wide, inch-deep' understanding of the system. In the 'mile-wide, inch-deep' description as many actors and use cases as possible are briefly described. Approximately 20 percent of the use cases are going to be listed as critical. Based on the 'mile-wide, inch-deep' description and the conceptual prototypes from the inception phase the 'mile-wide, inch-deep' description has been refined.

3.1.1 Actors

Actor No.1	
Name	Test site user
Description	A Test site user is a person who is using and testing the software prototype systems developed by NC3A.

Figure 3.1-1 Actor No.1 Test site user

Actor No.2	
Name	Analyst
Description	An Analyst is a person who is designing software prototype systems at NC3A.

Figure 3.1-2 Actor No.2 Analyst

Actor No.3	
Name	Developer
Description	A Developer is a person who is developing software prototype systems at NC3A.

Figure 3.1-3 Actor No.3 Developer

Actor No.4	
Name	Project manager
Description	A Project manager is a person who is in charge of a project at NC3A.

Figure 3.1-4 Actor No.4 Project manager

Actor No.5	
Name	Administrator
Description	An Administrator is a person who is going to be the administrator for the Requirement Capture System that has to be developed.

Figure 3.1-5 Actor No.5 Adminstrator

3.1.2 Use cases

Use case No. 1	(Critical use case No.1)
Name	Sign in.
Assumption	The system is started up.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system asks the user his/her username and password. (2) The user enters his/her username and password and tells the system he/she wants to sign in. (3) The system signs in the user en shows the available software prototype systems.
Exceptions	[Username unknown.] The system tells the user that the username is unknown. [Username and password combination not valid.] The system tells the user that the username and password combination is not valid. [User is inactive] The system tells the user that his/her user profile is inactive at this moment.
Result	User has signed in.

Figure 3.1-6 Use case No.1 Sign in

Use case No. 2	
Name	Sign out.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants so sign out. (2) The system signs out the user.
Exceptions	None.
Result	User has signed out.

Figure 3.1-7 Use case No.2 Sign out

Use case No. 3	
Name	View name user signed in.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system shows the name of the user that is signed in.
Exceptions	None.
Result	Name user signed in is displayed.

Figure 3.1-8 Use case No.3 View name user signed in

Use case No. 4	
Name	Modify own user profile.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to modify his/her own user profile. (2) The system asks the user what changes he/she wants to make and (3) the user answers that question. (4) The system will carry out the modifications to the user profile.
Exceptions	None.
Result	User profile is modified.

Figure 3.1-9 Use case No.4 Modify own user profile

Use case No. 5	
Name	Select software prototype system.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system asks the user to select a software prototype system and (2) the user selects one.
Exceptions	[No software prototype systems available.] The system tells the user that there are no software prototype systems available.
Result	A software prototype system is selected.

Figure 3.1-10 Use case No.5 Select software prototype system

Use case No. 6	
Name	View requirements generated.
Assumption	User has signed in and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to view the requirements that are generated for the selected software prototype system. (2) The system shows the user all the requirements that are generated for the selected software prototype system.
Exceptions	[No Requirements found.] The system will tell the user that there are no requirements generated for the selected software prototype system.
Result	Generated Requirements are displayed.

Figure 3.1-11 Use case No.6 View requirements generated

Use case No. 7	
Name	Generate requirement report.
Assumption	User has signed in as Project manager or Administrator and has selected a software prototype system.
Actors	Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to generate a Requirement Report for the selected software prototype system. (2) The system will generate a Requirement Report.
Exceptions	[There are no requirements generated for the selected software prototype system.] The system will tell the user that there are no requirements generated for the selected software prototype system and therefore it is not possible to generate a Requirement Report.
Result	Requirement Report is generated.

Figure 3.1-12 Use case No.7 Generated requirement report

Use case No. 8	
Name	Select a module.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system asks the user to select a module and (2) the user selects one.
Exceptions	[No modules available.] The system will tell the user that there are no modules for the selected software prototype system
Result	A module is selected.

Figure 3.1-13 Use case No.8 Select a module

Use case No. 9	
Name	Search for topics.
Assumption	User has signed in and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system asks the user to fill in a search query. (2) The user fills in a search query and submits it. (3) The system will display all found search results.
Exceptions	[There are no search results.] The system tells the user that there are no search results matching the search query.
Result	Topic list is displayed.

Figure 3.1-14 Use case No.9 Search for topics

Use case No. 10	
Name	View topics user is interested in.
Assumption	User has signed and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to see a list of topics that are of his interest. (2) The system shows a list of topics with topics that are of the users interest.
Exceptions	[No topics found.] The system will tell the user that there are no topics that are marked as interesting by the user.
Result	A topic list is displayed.

Figure 3.1-15 Use case No.10 View topics user is interested in.

Use case No. 11	
Name	View topics user is involved in.
Assumption	User has signed and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to see a list of topics that he is involved in. (2) The system shows a list of topics with topics which the user is involved in.
Exceptions	[No topics found.] The system will tell the user that he/she is not involved in any topic at this moment.
Result	A topic list is displayed.

Figure 3.1-16 Use case No.11 View topics user is involved in

Use case No. 12	(Critical use case No.2)
Name	Start a topic.
Assumption	User has signed in and has selected a software prototype system and a module.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to start a topic. (2) The system asks him/her the topic title, topic description, topic type, version of the software prototype system and the operating system that has been used. (3) The user answers those questions and tells the system that he/she wants to submit the topic. (4) The system shows the user the topic he/she has just provided.
Exceptions	None.
Result	Topic is started.

Figure 3.1-17 Use case No.12 Start a topic

Use case No. 13	(Critical use case No.3)
Name	View topic.
Assumption	User has signed in and has selected a software prototype system and a module or searched for feedback.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system shows the topic entries that previously have been submitted and belong to the chosen module or the search query that has been submitted or belong to the user topics interested or involved. A topic entry contains the topic title, the module where the topic is about, the topic type, the status of the topic, the author of the topic, the date of the last posted remark on this topic, the author of the last posted remark on this topic, the number of requirements generated and the date of last requirement generated. (2) The user tells the system on which topic entry he/she wants to look at. (3) The system shows the topic title, topic description, the topic type, the module where the topic is about, the version of the software prototype system that is used, the operating system that is used, the author of the topic, the status of the topic. If present, (4) the system will display the remarks and/or status changes that are added/made to this topic. The system will display the type (this can be a remark of a topic status change), a description, the author, the user group of the author and the date the remark is added or the status changes is made. If present, (5) the system will display the requirements that are generated from this topic. The system will display the requirement description, the requirement status, the author of the requirement and the date the requirement was generated.
Exceptions	[No topics present or found.] The system will tell the user there are no topics present or found. [There are no requirements generated.] The system will tell the user there are no requirements generated.
Result	Topic is displayed.

Figure 3.1-18 Use case No.13 View topic

Use case No. 14	
Name	Mark topic as interesting.
Assumption	User has signed and has selected a prototype, a module and a topic that is marked as uninteresting.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to mark the current topic as interesting. (2) The system marks the current topic as interesting.
Exceptions	None.
Result	Topic is marked as interesting.

Figure 3.1-19 Use case No.14 Mark topic as interesting

Use case No. 15	
Name	Unmark a topic as interesting.
Assumption	User has signed and has selected a prototype, a module and a topic that is marked as interesting.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to unmark the current topic as interesting. (2) The system unmarks the current topic as interesting.
Exceptions	None.
Result	Topic is unmarked as interesting.

Figure 3.1-20 Use case No.15 Unmark a topic as interesting

Use case No. 16	(Critical use case No.4)
Name	Change topic status.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a software prototype system, a module and a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to change the status of the topic. (2) The system asks the user what status he/she wants to give the topic and asks for an additional description. (3) The user tells the system what the new status of the topic must be, fills in an additional description and tells the system that he/she wants to submit the status change. (4) The system shows the user the updated topic information.
Exceptions	[Topic status is already 'Closed'.] The system tells the user that the status of the topic is 'Closed' and cannot be changed anymore.
Result	Topic status is changed.

Figure 3.1-21 Use case No.16 Change topic status

Use case No. 17	(Critical use case No.5)
Name	Add remark.
Assumption	User has signed in as Test site user, Analyst, Developer, Project manager or Administrator and has selected a prototype, a module and a topic.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to add a remark to current topic. (2) The system asks the user for a description of the remark and shows information that is already automatically know by the system. This information contains the type (that will be remark), the author, the user group of the author and that date. (3) The user fills in a description and tells the system that he wants to submit the remark. (4) The system tells the user that the remark has been submitted and shows the updated topic information.
Exceptions	None.
Result	Remark is added to topic.

Figure 3.1-22 Use case No.17 Add remark

Use case No.18	(Critical use case No.6)
Name	Generate requirement.
Assumption	User has signed in as Project manager or Administrator and has selected a prototype, a module and a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to generate a requirement from the current topic. (2) The system asks the user to fill in the requirement description and shows information that is already automatically known by the system. This is information contains the author, user group of the author and the date. (3) The user fills in the description of the requirement and tells the system that he/she wants to submit the requirement. (4) The system tells the user that the requirement has been submitted and shows the update topic information.
Exceptions	None.
Result	Requirement is generated.
Priority	3.

Figure 3.1-23 Use case No.18 Generate requirement

Use case No.19	(Critical use case No.7)
Name	Change requirement status.
Assumption	User has signed in as Project manager or Administrator and has selected a prototype, a module, a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system for which requirement he/she wants to change the status. (2) The system asks the user what status he/she wants to give the requirement and asks for an additional description. (3) The user tells the system what the new status of the requirement must be, fills in an additional description and tells the system that he/she wants to submit the status change. (4) The system shows the user the updated topic information.
Exceptions	[Requirement status is already 'Implemented'.] The system tells the user that the status of the requirement is 'Implemented' and cannot be changed anymore.
Result	Requirement status is changed.
Priority	3.

Figure 3.1-24 Use case No.19 Change requirement status

Use case No. 20	
Name	Add software prototype system.
Assumption	User has signed in as Administrator.
Actors	Administrator.
Description	(1) The user tells the system he/she wants to add a new software prototype system. (2) The system asks the name and properties of the software prototype system and (3) the user answers that question.
Exceptions	None.
Result	New prototype is added.

Figure 3.1-25 Use case No.20 Add software prototype system

Use case No. 21	
Name	Modify software prototype system.
Assumption	User has signed as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to modify the properties of a certain software prototype system. (2) The system asks what changes he/she wants to make and (3) the user answers that question.
Exceptions	None.
Result	Software prototype system is modified.

Figure 3.1-26 Use case No.21 Modify software prototype system

Use case No. 22	
Name	Delete software prototype system.
Assumption	User has signed in User has signed as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete a certain software prototype system and (2) the system deletes the software prototype system.
Exception	[There are topics related to the software prototype system.]
Result	Software prototype system is deleted.

Figure 3.1-27 Use case No.22 Delete software prototype system

Use case No. 23	
Name	Add user.
Assumption	User has signed in as Administrator
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to add a new user. (2) The system asks the user what the properties for the new user are (3) and the user answers that question. (4) The system will add a new user.
Exceptions	None.
Result	New user is added.

Figure 3.1-28 Use case No.23 Add user

Use case No. 24	
Name	Modify user profile.
Assumption	User has signed in as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to modify a certain user. (2) The system asks the user what changes he/she wants to make and (3) the user answers that question. (4) The system will carry out the modifications to the user profile.
Exceptions	None.
Result	User profile is modified.

Figure 3.1-29 Use case No.24 Modify user profile

Use case No. 25	
Name	Delete user.
Assumption	User has signed in as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete a certain user. (2) The system will delete the user.
Exceptions	[There are still topics related to the user.]
Result	User status has changed.

Figure 3.1-30 Use case No.25 Delete user

Use case No. 26	
Name	Change user status.
Assumption	User has signed in as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to change the status of a certain user. (2) The system asks the user to what status and (3) the user answers that question with ‘active’ or ‘inactive’.
Exceptions	[The status of the user is the same as the chosen status.] The system tells the user that the user status is already ‘active’ or ‘inactive’.
Result	User status has changed.

Figure 3.1-31 Use case No.26 Change user status

Use case No. 27	
Name	Delete topic.
Assumption	User has signed in as Administrator and has selected a software prototype system and a topic
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete the selected topic.
Exceptions	None.
Result	Topic is deleted.

Figure 3.1-32 Use case No.27 Delete topic

Use case No. 28	
Name	Delete remark.
Assumption	User has signed in as Administrator and has selected a software prototype system, a topic and a remark.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete the selected remark.
Exceptions	None.
Result	Remark is deleted.

Figure 3.1-33 Use case No.28 Delete remark

Use case No. 29	
Name	Delete requirement.
Assumption	User has signed in as Administrator and has selected a software prototype system and requirement.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete the selected requirement
Exceptions	None.
Result	Requirement is deleted.

Figure 3.1-34 Use case No.29 Delete requirement

3.1.3 Use case diagram

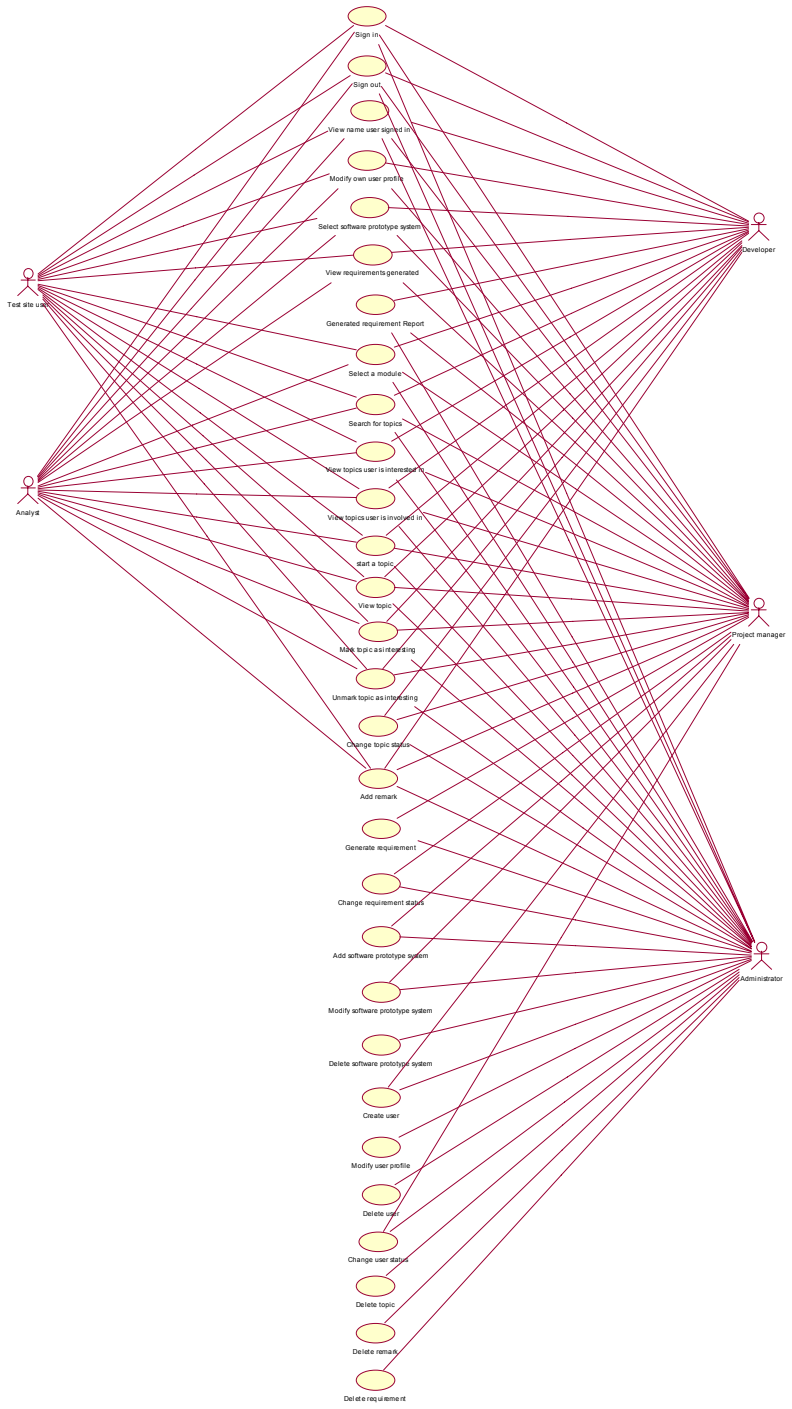


Figure 3.1-35 Use case diagram

3.2 Critical use cases

It is important to decide which use cases are the most essential or architecturally significant for the system to be built. The critical use cases should also describe the core functionality of the system. In this section the refined critical use cases are described.

3.2.1 Descriptions of the critical use cases

Critical use case	No. 1
Name	Sign in.
Assumption	The system is started up.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system asks the user his/her username and password. (2) The user enters his/her username and password and tells the system he/she wants to sign in. (3) The system signs in the user en shows the available software prototype systems.
Exceptions	[Username unknown.] The system tells the user that the username is unknown. [Username and password combination not valid.] The system tells the user that the username and password combination is not valid.
Result	User has signed in.
Priority	1.

Figure 3.2-1 Critical use case No.1 Sign in

Critical use case	No. 2
Name	Start a topic.
Assumption	User has signed in and has selected a software prototype system and a module.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to start a topic. (2) The system asks him/her the topic title, topic description, topic type, version of the software prototype system and the operating system that has been used. (3) The user answers those questions and tells the system that he/she wants to submit the topic. (4) The system shows the user the topic he/she has just provided.
Exceptions	None.
Result	Topic is started.
Priority	1.

Figure 3.2-2 Critical use case No.2 Start a topic

Critical use case	No. 3
Name	View topic.
Assumption	User has signed in and has selected a software prototype system and a module or searched for feedback.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system shows the topic entries that previously have been submitted and belong to the chosen module or the search query that has been submitted. A topic entry contains the topic title, the module where the topic is about, the topic type, the status of the topic, the author of the topic, the date of the last posted remark on this topic, the author of the last posted remark on this topic, the number of requirements generated and the date of last requirement generated. (2) The user tells the system on which topic entry he/she wants to look at. (3) The system shows the topic title, topic description, the topic type, the module where the topic is about, the version of the software prototype system that is used, the operating system that is used, the author of the topic, the status of the topic. If present, (4) the system will display the remarks and/or status changes that are added/made to this topic. The system will display the type (this can be a remark of a topic status change), a description, the author, the user group of the author and the date the remark is added or the status change is made. If present, (5) the system will display the requirements that are generated from this topic. The system will display the requirement description, the requirement status, the author of the requirement and the date the requirement was generated.
Exceptions	[No topics present or found.] The system will tell the user there are no topics present or found. [There are no requirement generated.] The system will tell the user there are no requirements generated.
Result	Topic is displayed.
Priority	1.

Figure 3.2-3 Critical use case No.3 View topic

Critical use case	No. 4
Name	Change topic status.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a software prototype system, a module and a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to change the status of the topic. (2) The system asks the user what status he/she wants to give the topic and asks for an additional description. (3) The user tells the system what the new status of the topic must be, fills in an additional description and tells the system that he/she wants to submit the status change. (4) The system shows the user the updated topic information.
Exceptions	[Topic status is already ‘Closed’.] The system tells the user that the status of the topic is ‘Closed’ and cannot be changed anymore.
Result	Topic status is changed.
Priority	2.

Figure 3.2-4 Critical use case No.4 Change topic status

Critical use case	No. 5
Name	Add remark.
Assumption	User has signed in as Test site user, Analyst, Developer, Project manager or Administrator and has selected a prototype, a module and a topic.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to add a remark to current topic. (2) The system asks the user for a description of the remark and shows information that is already automatically know by the system. This information contains the type (that will be remark), the author, the user group of the author and that date. (3) The user fills in a description and tells the system that he wants to submit the remark. (4) The system tells the user that the remark has been submitted and shows the updated topic information.
Exceptions	None.
Result	Remark is added to topic.
Priority	2.

Figure 3.2-5 Critical use case No.5 Add remark

Critical use case	No. 6
Name	Generate requirement.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a prototype, a module and a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to generate a requirement from the current topic. (2) The system asks the user to fill in the requirement description and shows information that is already automatically known by the system. This is information contains the author, user group of the author and the date. (3) The user fills in the description of the requirement and tells the system that he/she wants to submit the requirement. (4) The system tells the user that the requirement has been submitted and shows the update topic information.
Exceptions	None.
Result	Requirement is generated.
Priority	3.

Figure 3.2-6 Critical use case No.6 Generate requirement

Critical use case	No. 7
Name	Change requirement status.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a prototype, a module, and a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system for which requirement he/she wants to change the status. (2) The system asks the user what status he/she wants to give the requirement and asks for an additional description. (3) The user tells the system what the new status of the requirement must be, fills in an additional description and tells the system that he/she wants to submit the status change. (4) The system shows the user the updated topic information.
Exceptions	[Requirement status is already 'Implemented'.] The system tells the user that the status of the requirement is 'Implemented' and cannot be changed anymore.
Result	Requirement status is changed.
Priority	3.

Figure 3.2-7 Critical use case No.7 Change requirement status

3.2.2 Critical use case diagram

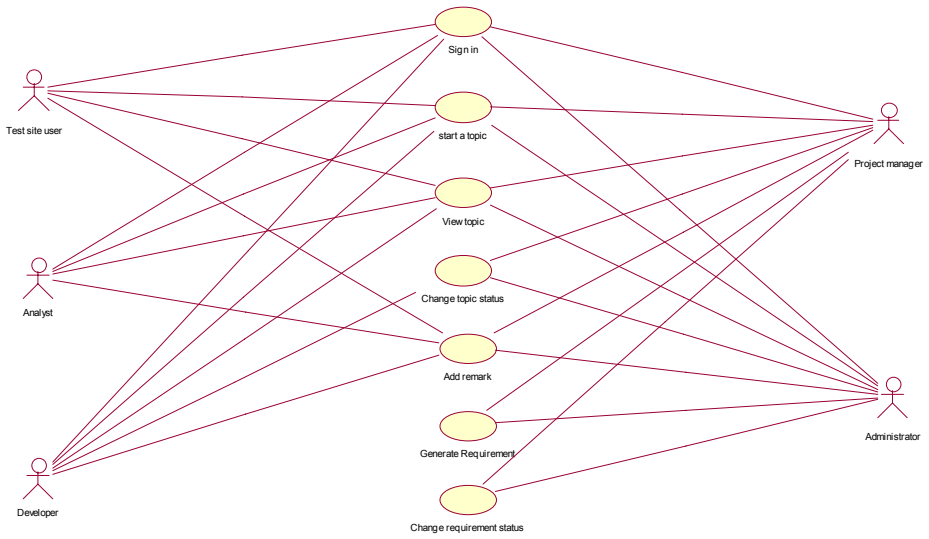


Figure 3.2-8 Critical use case diagram

3.3 Database design

For the development of an executable prototype it is necessary to design and implement a database. This starts with a class diagram that represents the real world. This class diagram will then be translated to a database model diagram. This chapter will describe the design of the database.

3.3.1 Class diagram

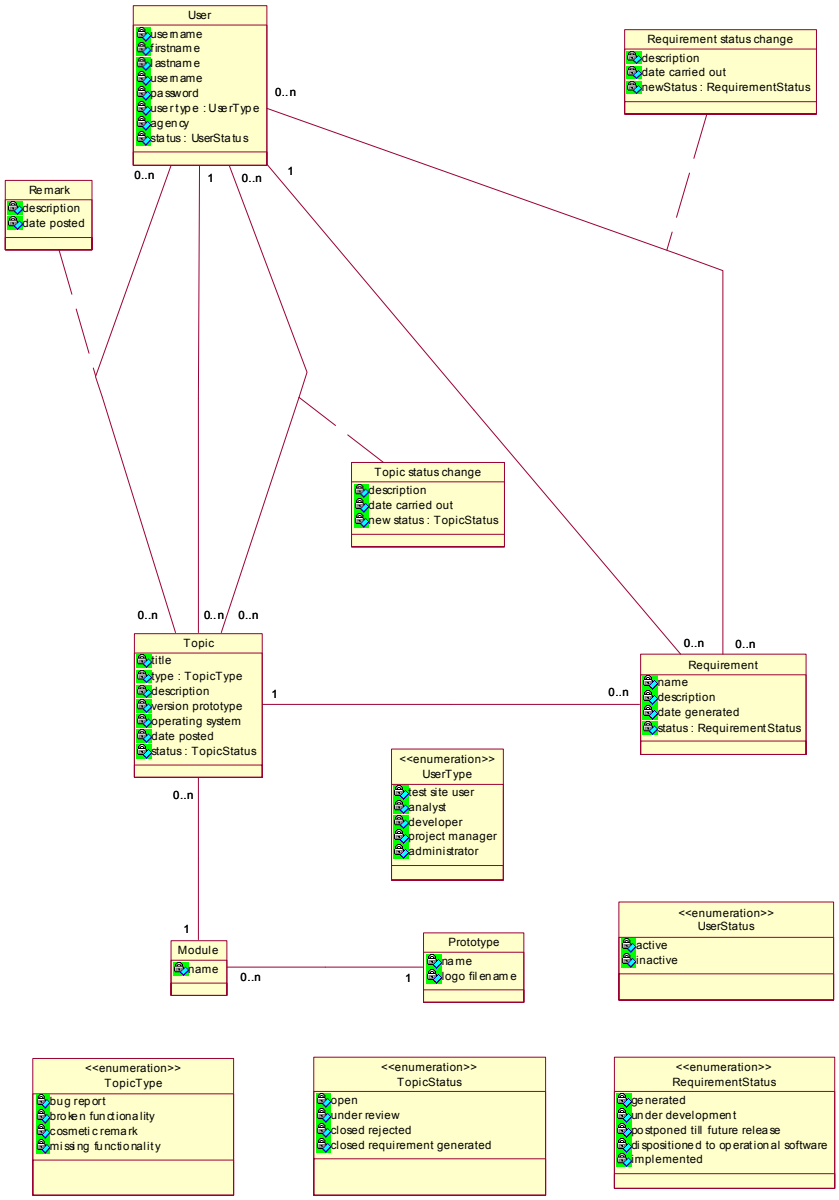


Figure 3.3-1 Class diagram

3.3.2 Database model diagram

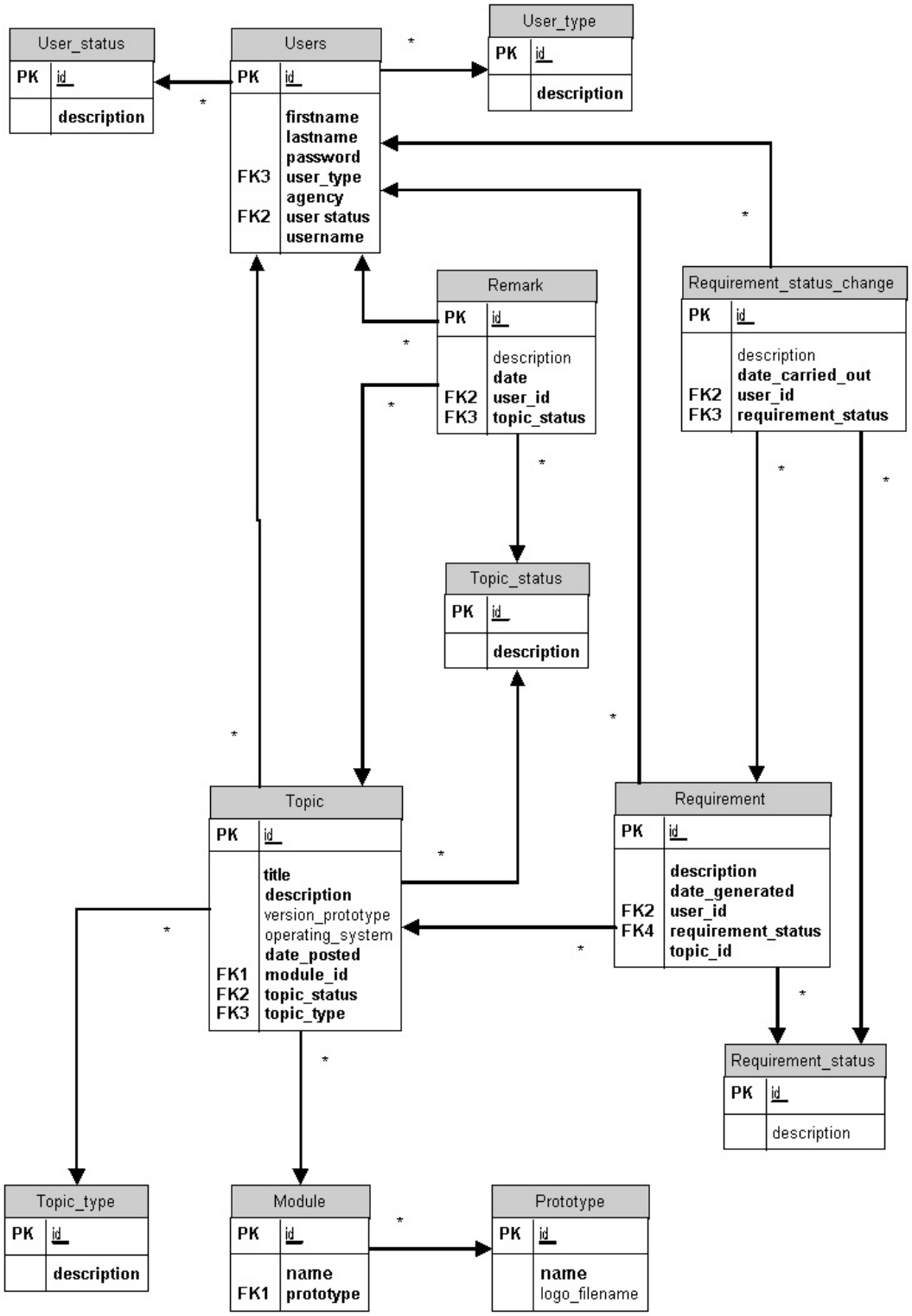


Figure 3.3-2 Database model diagram

3.4 Statechart diagrams

Certain classes from the class diagram described in section 3.3.1 can have multiple states. To get a clear vision of what states and what transitions those classes can have, statechart diagrams are used.

3.4.1 Class User

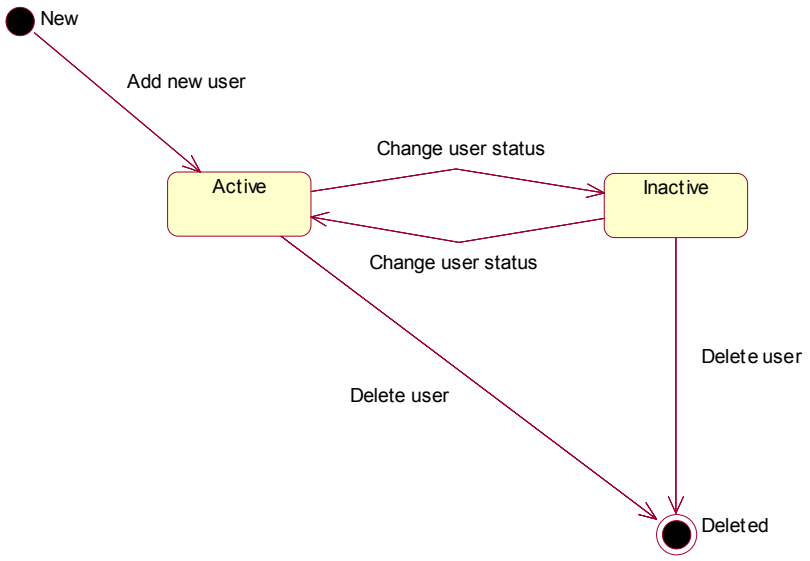


Figure 3.4-1 Statechart diagram Class User

3.4.2 Class Topic

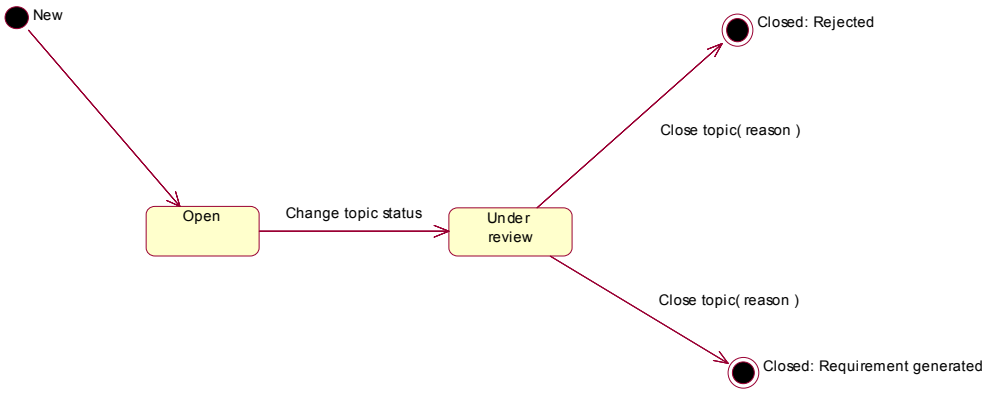


Figure 3.4-2 Statechart diagram Class Topic

3.4.3 Class Requirement

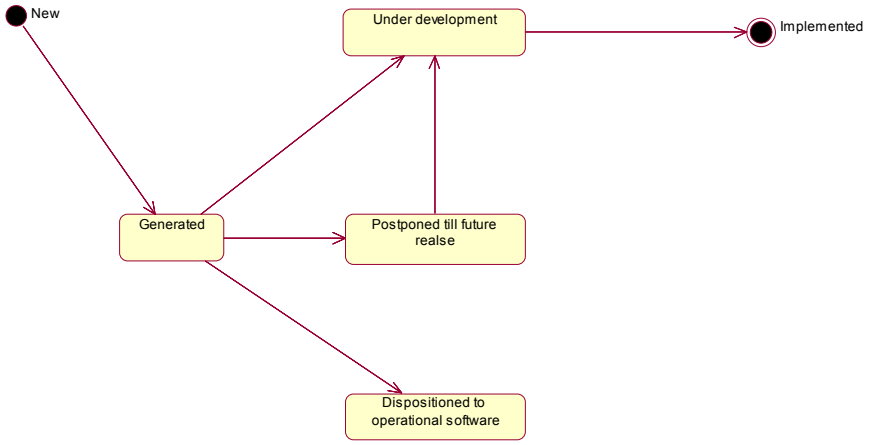


Figure 3.4-3 Statechart diagram Class Requirement

Chapter 4 - Chosen solution

In the Lifecycle Objective Milestone possible solutions for implementing the system where discussed. This chapter will discuss the chosen solution. This chapter will also describe how the system can be developed.

4.1 Web-based

The system is going to be a web-based application. The advantage of a web-based application is the fact that everyone with the access to Internet will have access to the system. This prevents required software being installed on the computers of the users of the system.

4.2 NATO UNCLASSIFIED

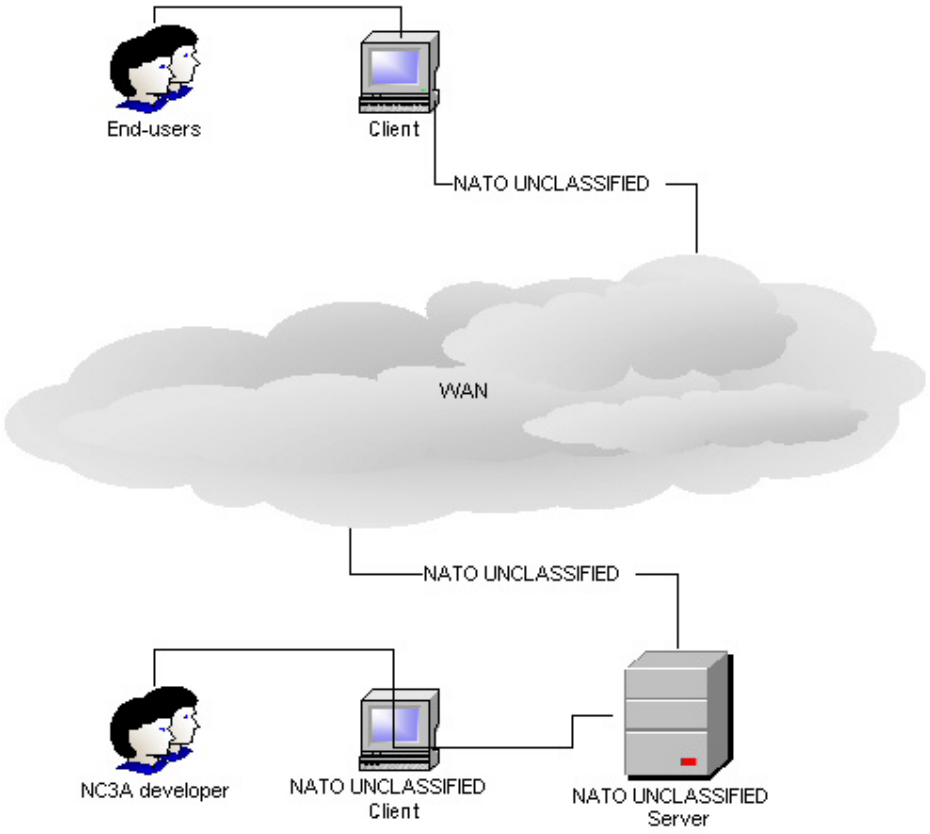


Figure 4.2-1 Chosen solution: unclassified

The system will be hosted on an NATO UNCLASSIFIED server. The consequence of this decision is that all the information that will be transferred must be NATO UNCLASSIFIED releasable for Internet. The advantage is that there is no procedure to follow for getting an approval for hosting a system on the Crises Response Operational NATO Open Systems (CRONOS). The restriction for hosting a system on the NATO UNCLASSIED network is that the system has to be built in SharePoint.

4.2.1 SharePoint

SharePoint is a system that is released by Microsoft at the end of 2003. It can be used by organisations to couple users, teams and knowledge in order for people to use relevant information to support the main process of the organisation.

SharePoint functions as portal. This is similar to an intranet where it is possible to go from one place to another place and the ability to user different services.

The portal allows people to exchange information. The organisation can aggravate the information in the portal to certain users or groups based on what they for fill, team.

SharePoint offers a basis for the developer to start. It contains certain standard features that do not have to be developed anymore. This can save a lot of time. Several standard features that are called web parts are:

- Document libraries. Here documents can be placed that can be requested by other users
- Forum. With this users can exchange messages and discuss about different subjects

To manage the portal the developer can use these web parts. These web parts are developed in ASP.NET. A developer can also develop his/her own web part. This can then in the same way as the standard web parts be integrated into the portal.

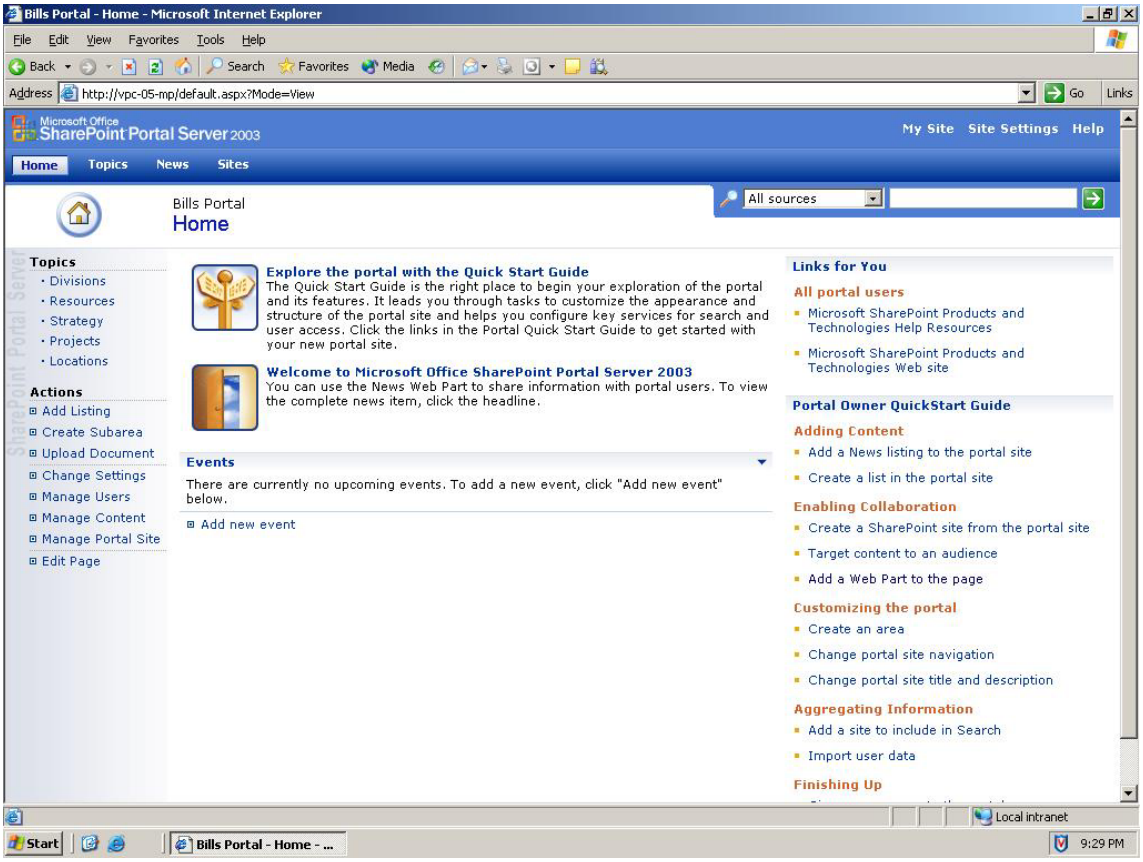


Figure 4.2-2 MS SharePoint Portal Server

It is possible to adapt this Forum according to the needs of the organisation. Unfortunate the standard Forum feature does not fits the needs of the system that have to be developed

SharePoint is not a packet that can be used to developed web applications. Everything within SharePoint is written in ASP.NET en changes cannot be added to a page like it is possible with HTML. All standard functionalities are implemented within SharePoint.

It is possible to develop an ASP.NET web application and integrate this into a SharePoint Portal. ASP.NET is the successor of ASP that stands for Active Server Pages and is part of the Microsoft .NET framework. In the next sections those will be described.

4.2.2 Microsoft .NET

The .NET framework is a 'from the scratch' developed platform to allow developers to built their applications more efficient and flexible. The two main components of the .NET framework are the Common Language Runtime (CLR) and the 'class library'. Another component is the development of server side applications.

Common Language Runtime (CLR)

The Command Language Runtime (CLR) forms the basic of the .NET framework. The CLR is responsible for a safe and efficient execution of applications. The CLR manages the memory, checks if the application has sufficient rights to carry out certain actions and checks whether there is nothing done what could me the system instable. The CLR works on top of the operating system so the applications do not communicate directly with the operating system. In that way the applications are being shielded from the complexity of operating systems.

Applications that belong to the .NET framework are divided into four categories:
- Windows Forms applications
- Windows Services
- ASP.NET applications
- ASP.NET Web Services

All these applications are built with one of the programming languages that the .NET framework supports. These are all language within the Common Language Specification. A schematic representation of the .NET framework is displayed in the following figure.

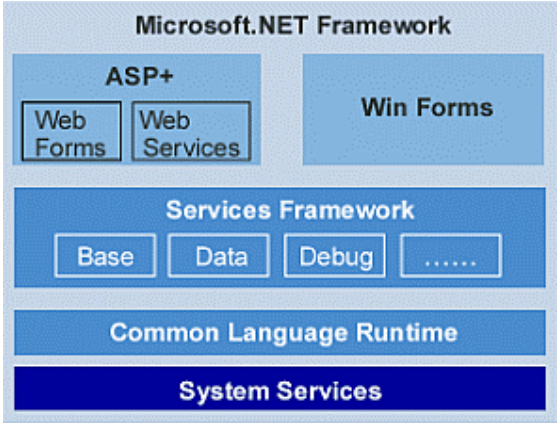


Figure 4.2-3 Microsoft .NET framework

Beneath this structure are the system services (operating system). On top of the system services is the CLR. The CLR loads and executes code that is written and aimed at the CLR. The CLR also offers integrated security.

Class libraries

The class library is an object-oriented collection of useful functions that can be used for the development of applications. The library offers functions that are normally offered by the operating system. This way the applications are being shielded for the complexity of the operating system. All applications that work on the .NET framework use these functions instead of the offered functions by the operating system.

4.2.3 ASP.NET

ASP.NET is the successor of the classical Active Server Pages (ASP) that is a development platform for developing web applications.

An ASP.NET page is a web page that ends on the extension .aspx. The ASP.NET page contains normal HTML code with scripts. The normal HTML code will be executed and interpreted on the client side while ASP.NET compiles that code on the server side. The result is being transmitted to the browser.

There are different ways to develop ASP.NET web applications. Several possibilities will now be discussed.

Notepad-'.NET

It is not really called Notepad '.NET' but as many people call this tool 'Visual' Notepad, it is also called Notepad '.NET. Because Notepad is installed on almost every Microsoft Windows platform there is probably not an easier development tool to find. Another advantage is the price of Notepad. Notepad is free en nothing has to be downloaded. To develop ASP.NET with Notepad the only thing required is to install the Software Development Kit (SDK) of the .NET framework. Building an ASP.NET page works the same as building a HTML document with notepad. The only thing is that the document have to be saved with the extension .aspx.

Besides Notepad there are several third-party text editors with possibilities that where previously only available with better development tools. Several possibilities are:

- Colouring of the syntax. This feature uses colour display different types of keywords in the language the development tool is used for.
- Automatic additions. This feature suggests possible additions of statement while typing, based on text previously provided by the user.

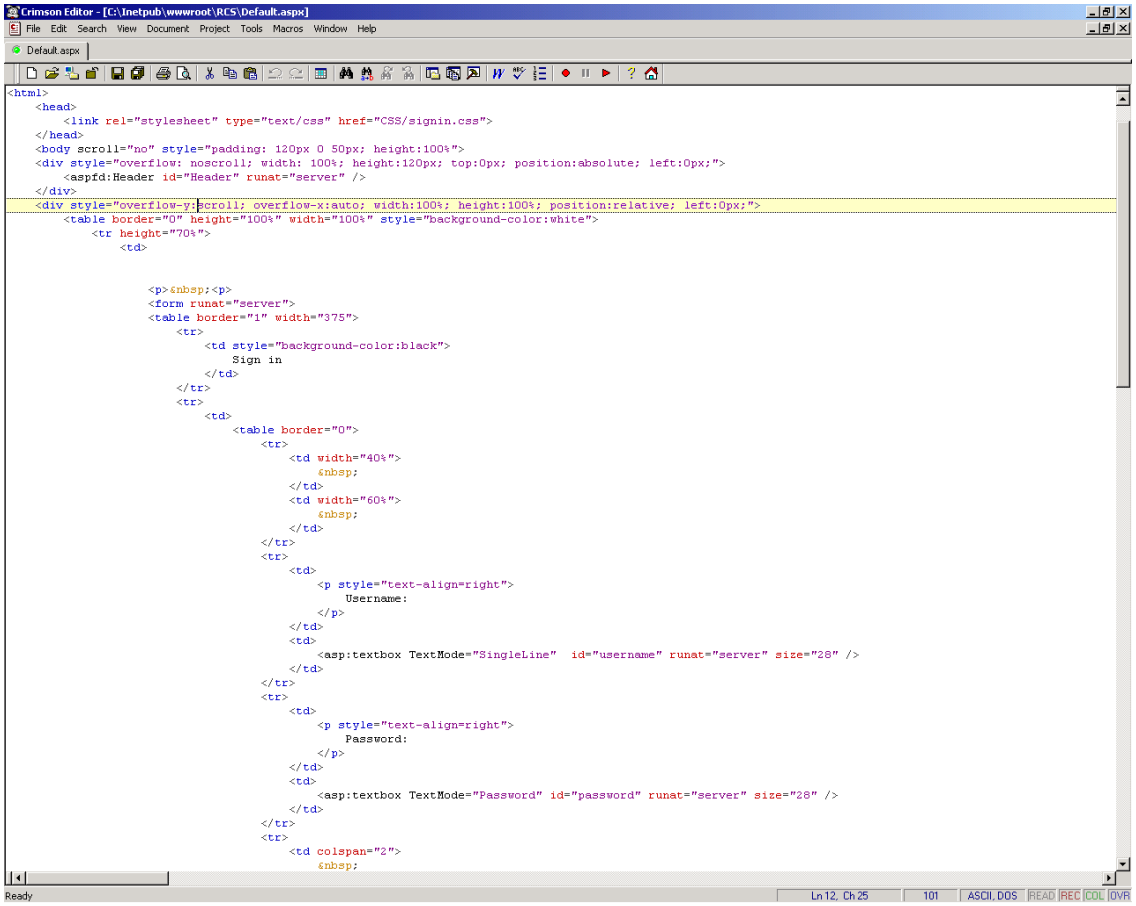


Figure 4.2-4 Rich-text editor: Crimson editor

Visual Studio .NET

For the development of for example ASP.NET web applications Microsoft has developed Visual Studio .NET, de newest version of this development suite Visual Studio. Visual Studio .NET has the following advantages:

- Robust management of project files and different projects.
- Integration with source code management environment
- Visual tools for working with web services, server controls in web forms and database tools
- Services for bundling and using web applications

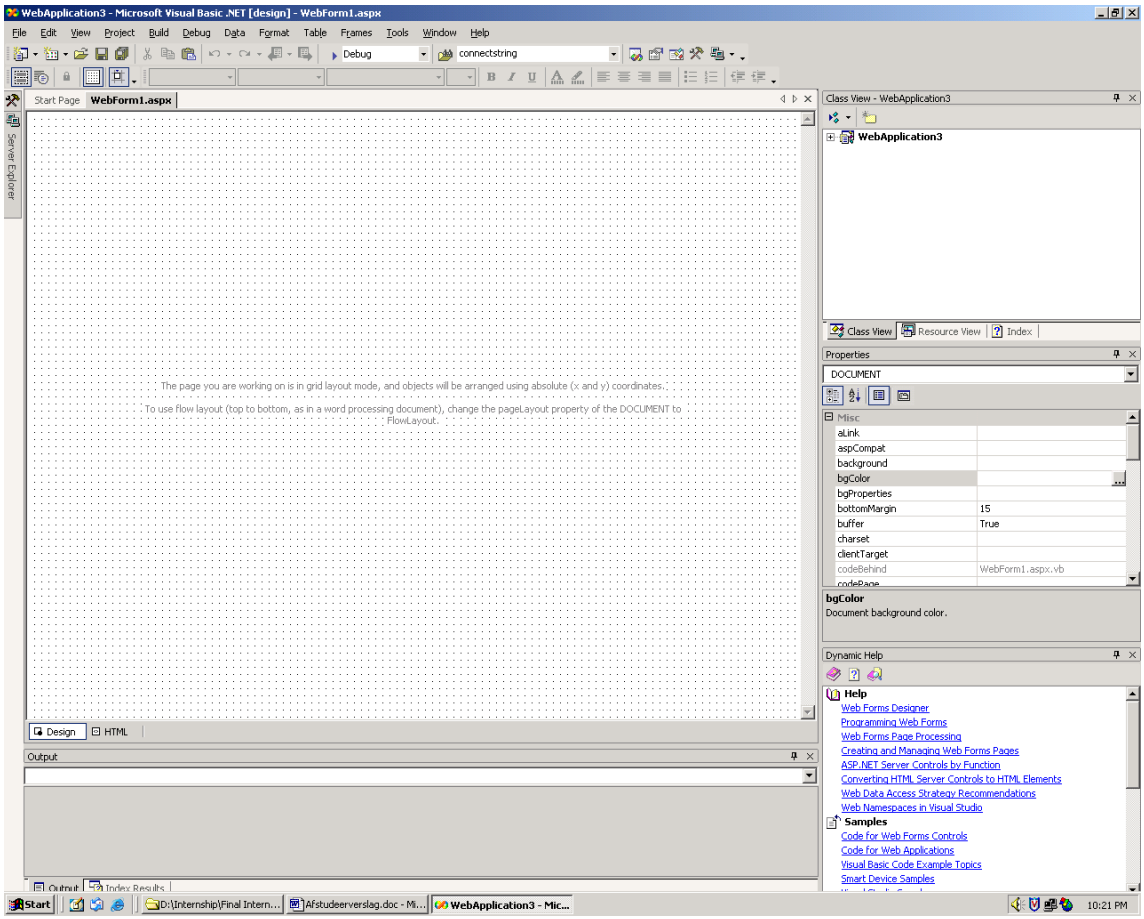


Figure 4.2-5 Visual Studio .NET

Chapter 5 - Executable prototype

The executable prototype is developed in ASP.NET in a rich-text editor and uses a MS Access database. The prototype is based on the critical use cases. Also some non-critical use cases are partially implemented in this prototype. This chapter will describe the executable prototype that has been built along with the evaluation results.

5.1 Critical use cases

This section will discuss the executable prototype based on the critical use cases.

5.1.1 Sign in

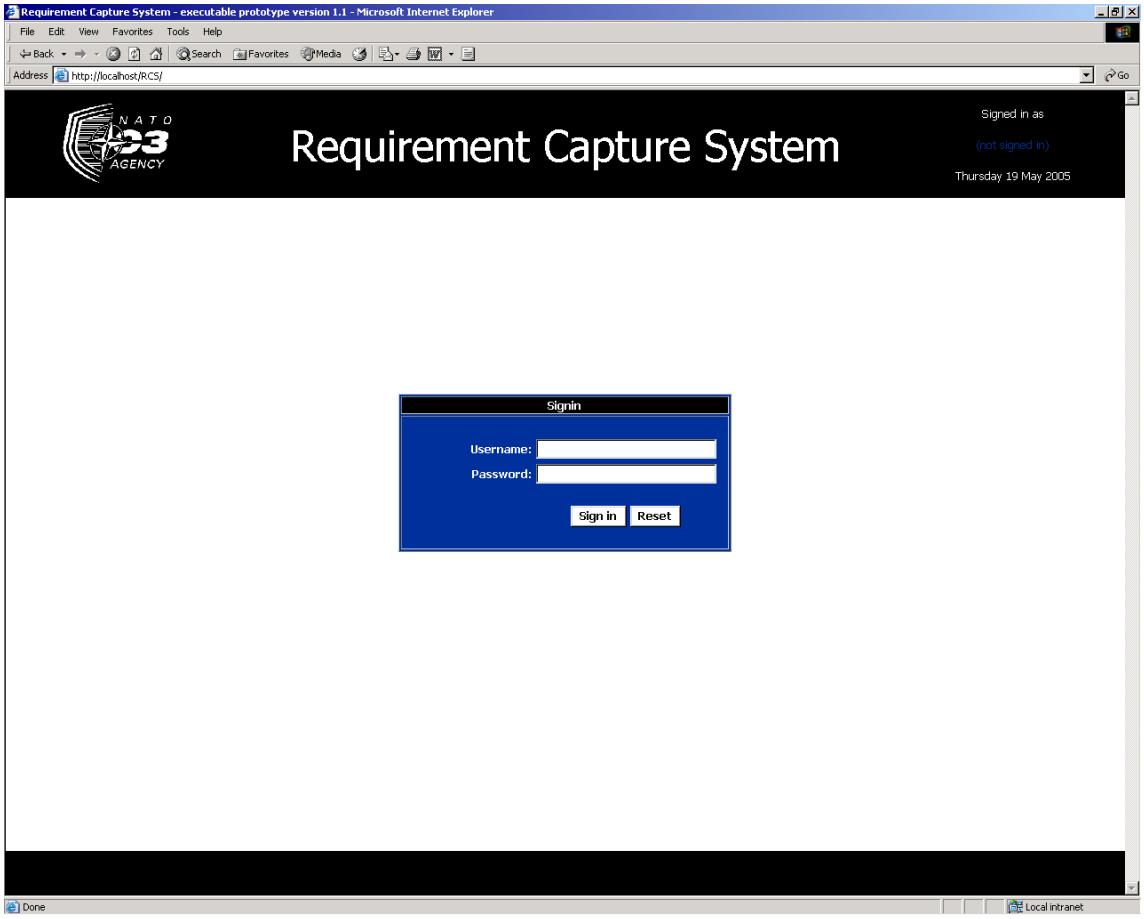


Figure 5.1-1 SignIn screen

In the ‘SignIn’ screen the use can enter his/her username and password. By clicking on the ‘Sign in’ button the use can submit username and password. When the use clicks on the button ‘Reset’ the text fields ‘Username’ and ‘Password’ will reset.

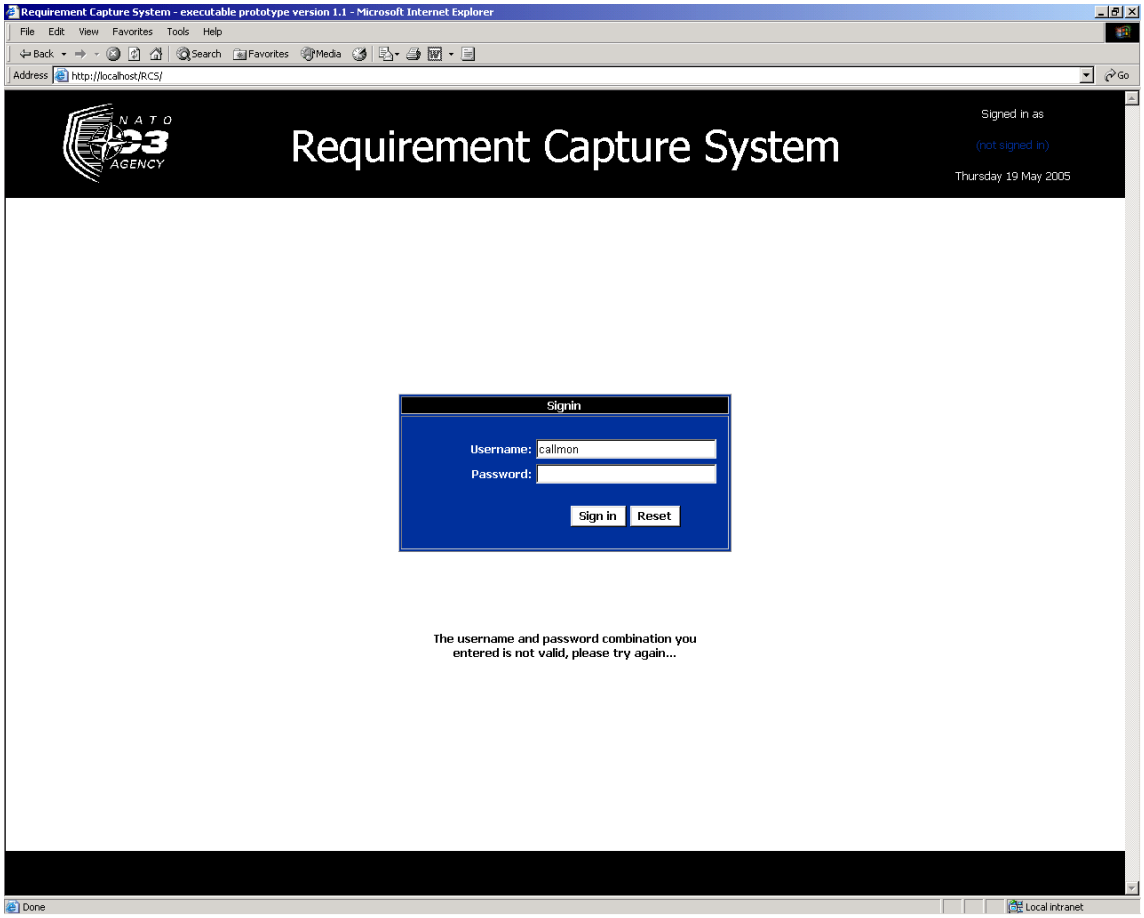


Figure 5.1-2 SignIn screen

When the user submits an unknown username there will be an error message displaying ‘the username you entered is unknown’. If the user account is set to inactive the error message will display ‘You username is currently inactive’. If the username and password combination is not valid the error message will display ‘The username and password combination you entered is not valid, please try again’.

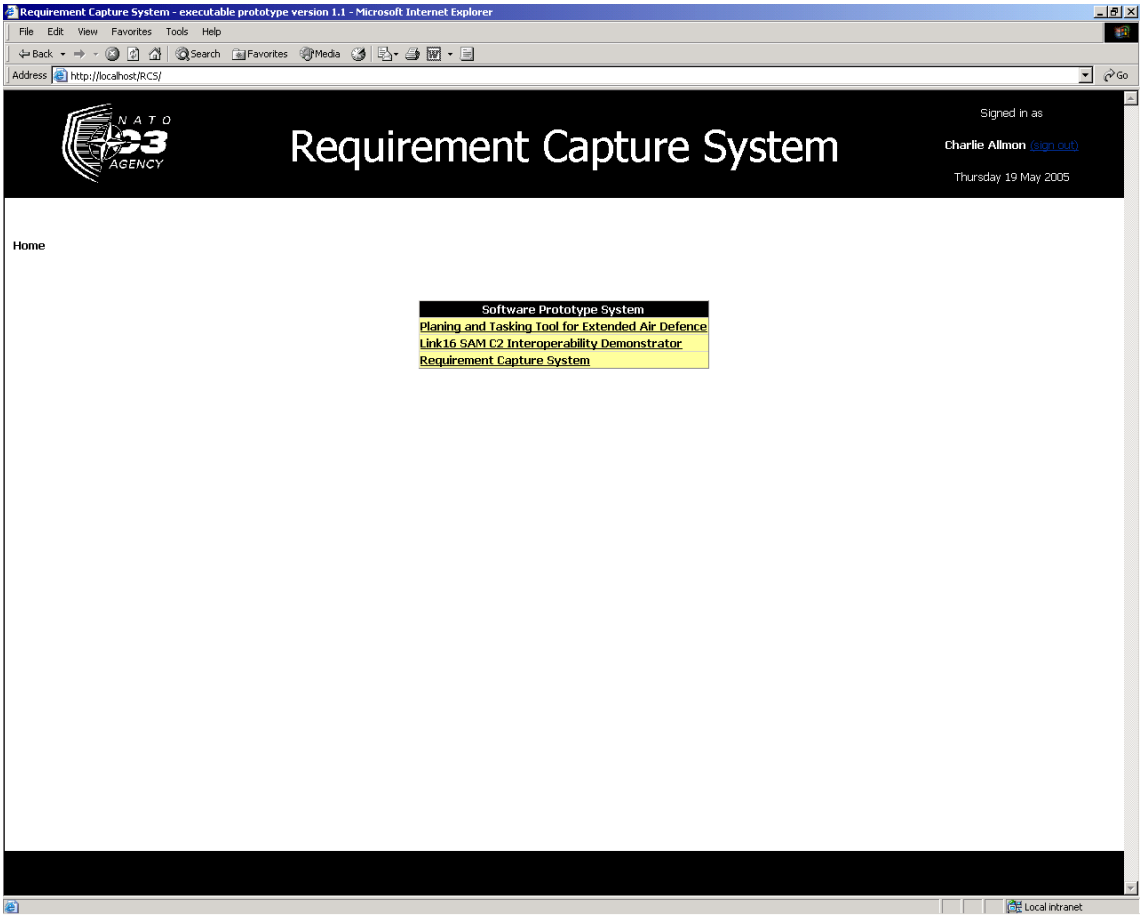


Figure 5.1-3 PrototypeList screen

When the user entered a valid username and password combination the user will be signed in. The name of the user who is signed in will be displayed at the header in the right section.

Evaluation results

- If the user cannot sign in he/she should get the same message at all times. This is because of security reasons. It should not be possible for the user to retrieve usernames from the system.
- For the 'SignIn' page there must be make use of the Secure Socket Layer (SSL) protocol so that people on the server cannot retrieve the information that is submitted.

5.1.2 Start a topic

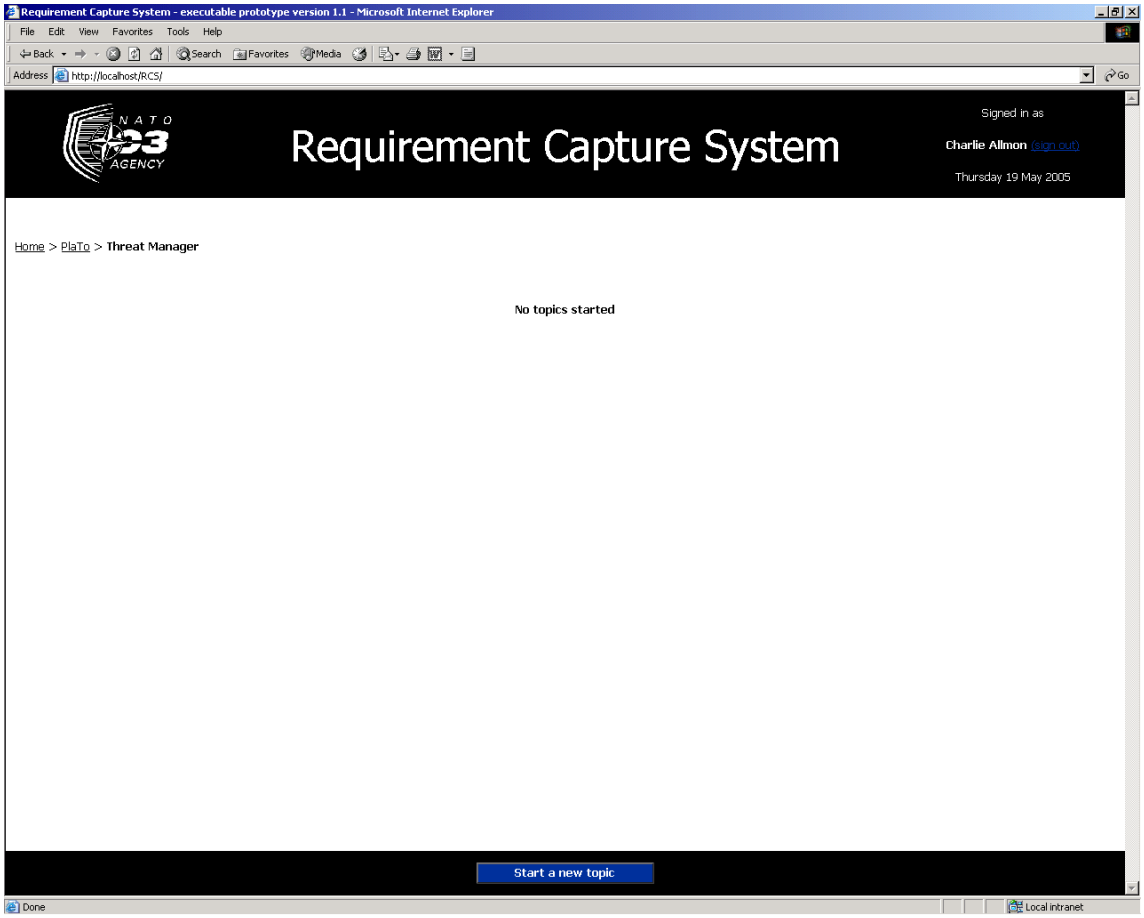


Figure 5.1-4 TopicList screen

After selecting a software prototype system and a module (non-critical use cases) the system will display the ‘TopicList’ screen. In this screen the user can start a new topic by clicking on the ‘Start a new topic’ button on the footer.

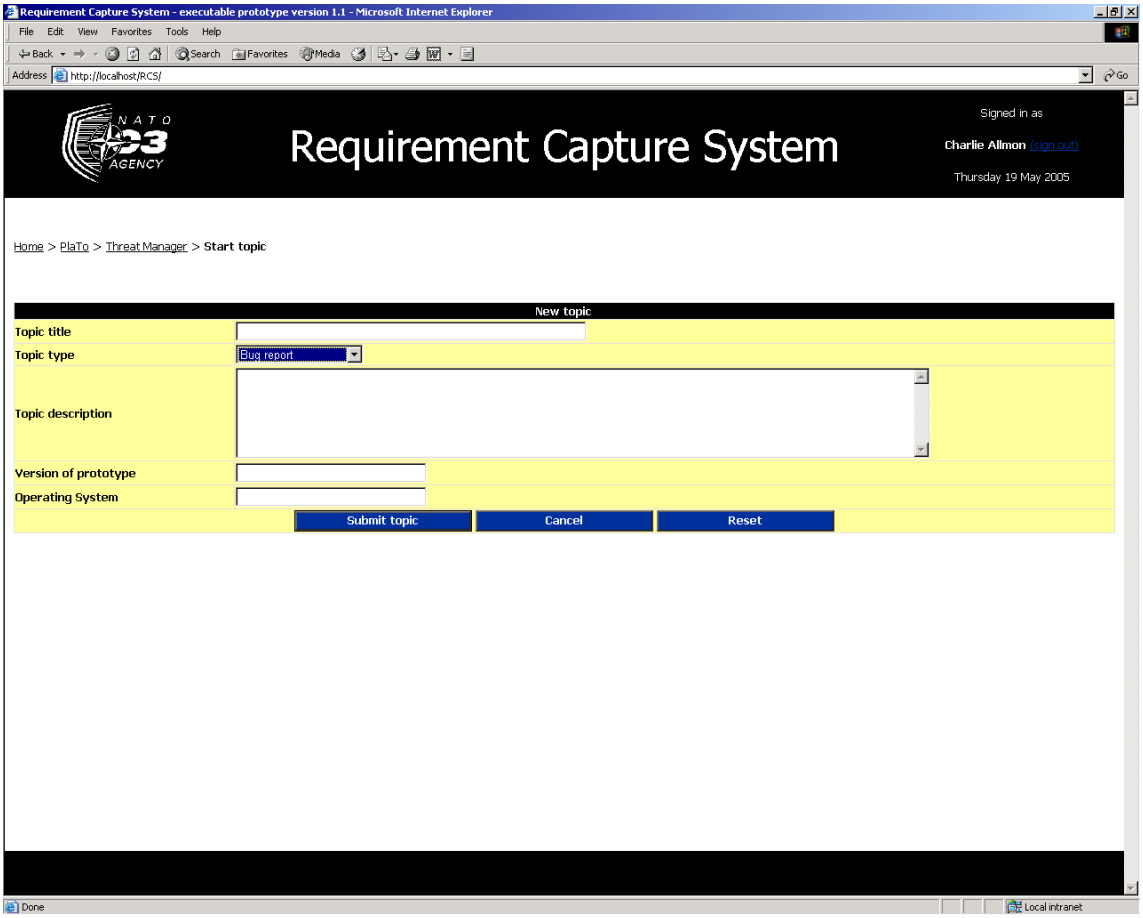


Figure 5.1-5 Start a topic screen

After clicking on the 'Start a new topic' button the system will display a new form where the user can fill in the topic. This is called the 'StartTopic' screen. The fields are:

- Topic title (text field)
- Topic type (dynamic dropdown list)
- Topic description (text field)
- Version of prototype (text field)
- Operating System (text field)

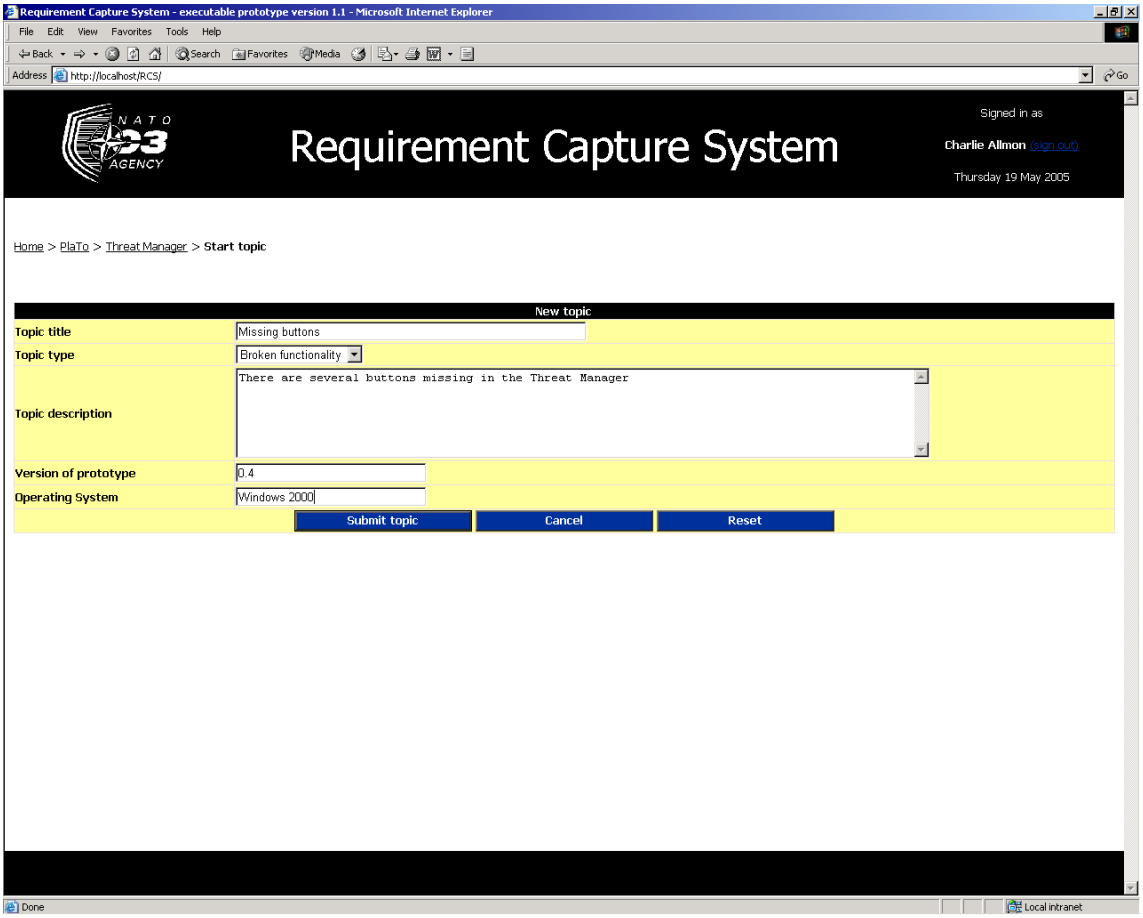


Figure 5.1-6 TopicStart screen

By clicking on the 'Reset' button all the fields will be reset and by clicking on the 'Cancel' button the system will display the previous screen. The user can submit the topic by clicking on the 'Submit topic' button.

Evaluation results

- There must be a validation that makes sure that every field is filled in before submitting the topic. Only the field 'Operating System' is optional and may be left blank.
- The dropdown lists must have a standard value that cannot be submitted so that the user is forced to select a value.
- The 'Version of prototype' must become a dropdown list instead of a text field. This must become a dynamic dropdown list so that it retrieves its values from the database.
- There must come an additional field named 'Priority' where the user can give a priority to the topic that he/she wants to submit. This must become a dropdown list with the values High, Low and Medium.

5.1.3 View topic

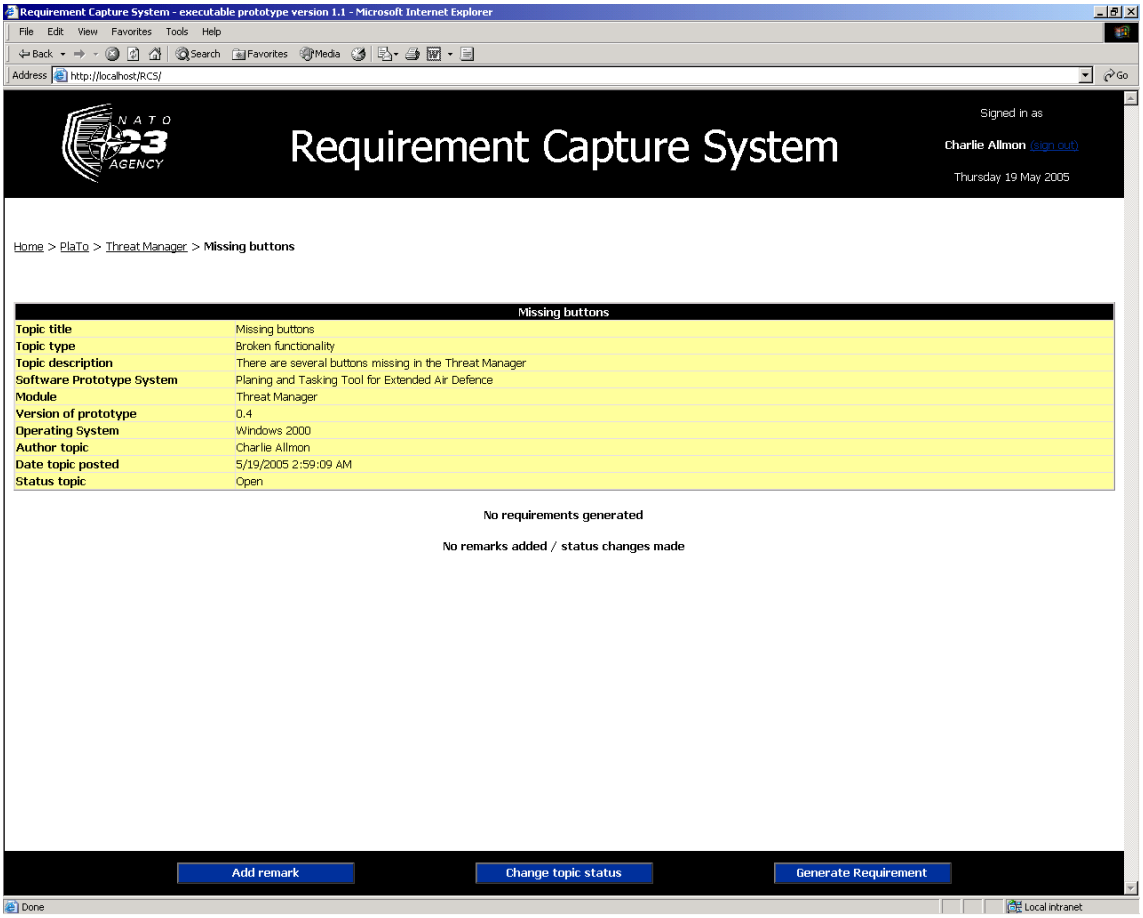


Figure 5.1-7 TopicView screen

When a topic is submitted the system will display 'TopicView' screen that displays all information about topic. This also includes the requirements that have been generated from this topic and remarks and status changes on the topic. This information displayed on the topic:

- Topic title
- Topic type
- Topic description
- Software Prototype System
- Module
- Version of prototype
- Operating System
- Author topic
- Date topic posted
- Status topic

In the footer are now three new buttons:

- Add remark
- Change topic status
- Generate Requirement

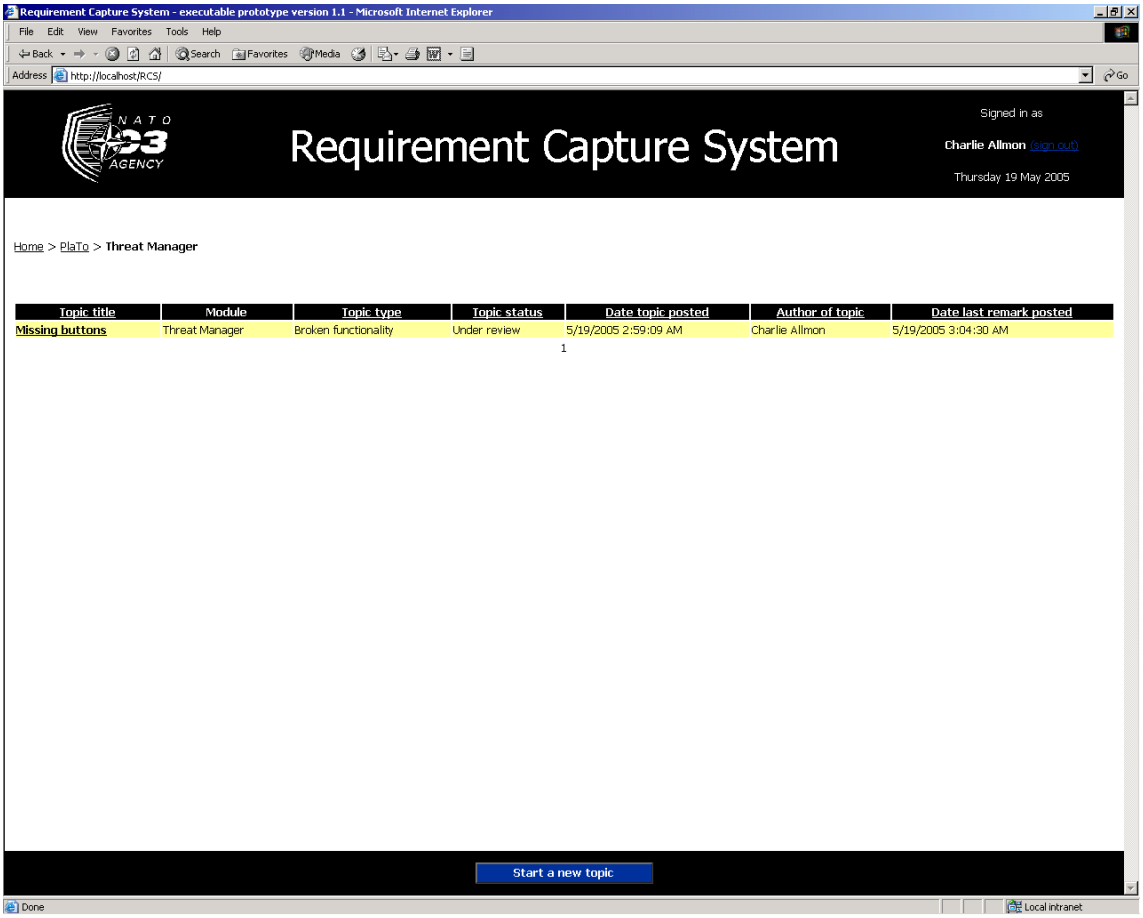


Figure 5.1-8 TopicList screen

When the user goes back to the 'TopicList' screen the topic he/she just added is also briefly displayed. The information displayed on a topic in this screen is:

- Topic title
- Module
- Topic type
- Topic status
- Date topic posted
- Author of topic
- Date last remark posted

By clicking on the header the user can sort the topics. There is also pager that allows the user to page true the topic list.

Evaluation results

- The word 'topic' should not be used that much. For example it should become 'Author' instead of 'Author of topic'. This involved all use cases / screens.

5.1.4 Change topic status

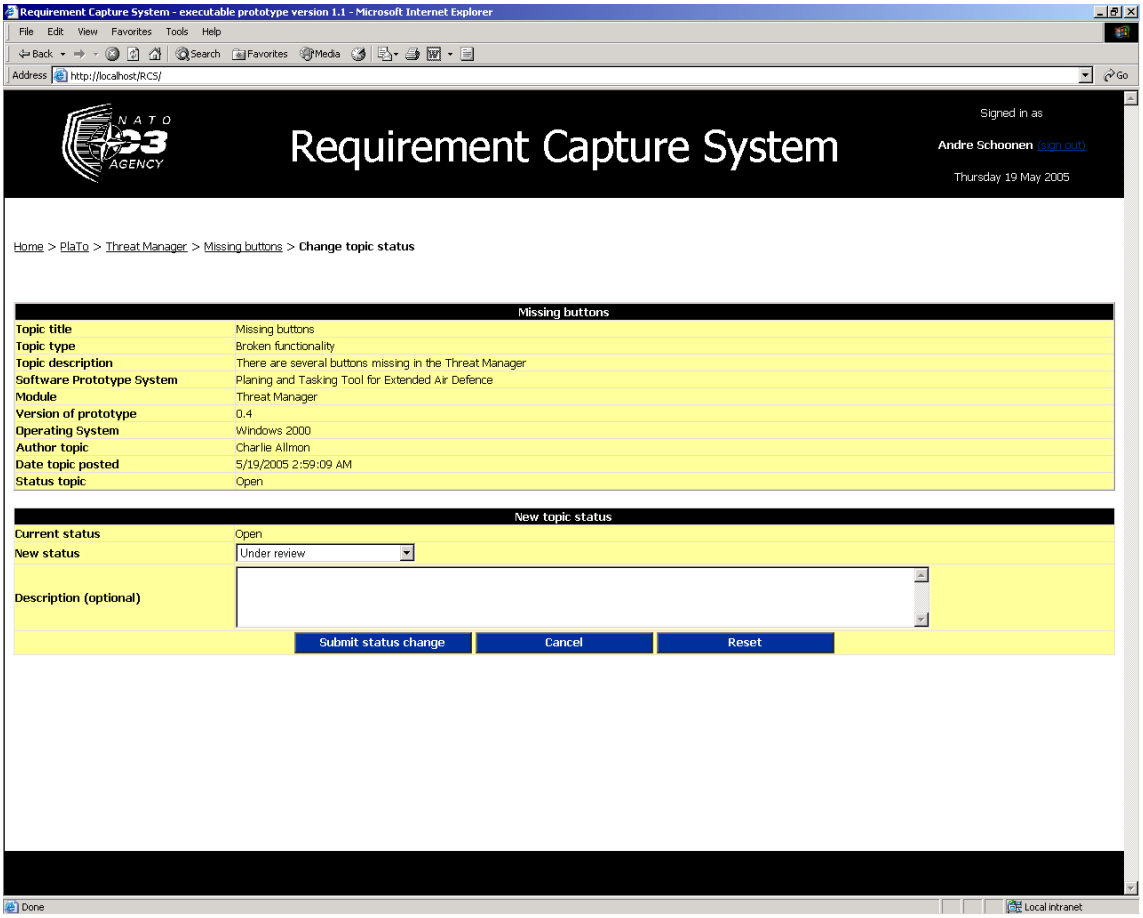


Figure 5.1-9 TopicChangeStatus screen

When the user clicks on the ‘Change topic status’ button in the ‘TopicView’ screen, an additional form is displayed. In this form the user can submit the new status and provide an additional description. The fields in this form are:

- New status (dynamic dropdown list)
- Description (optional) (text field)

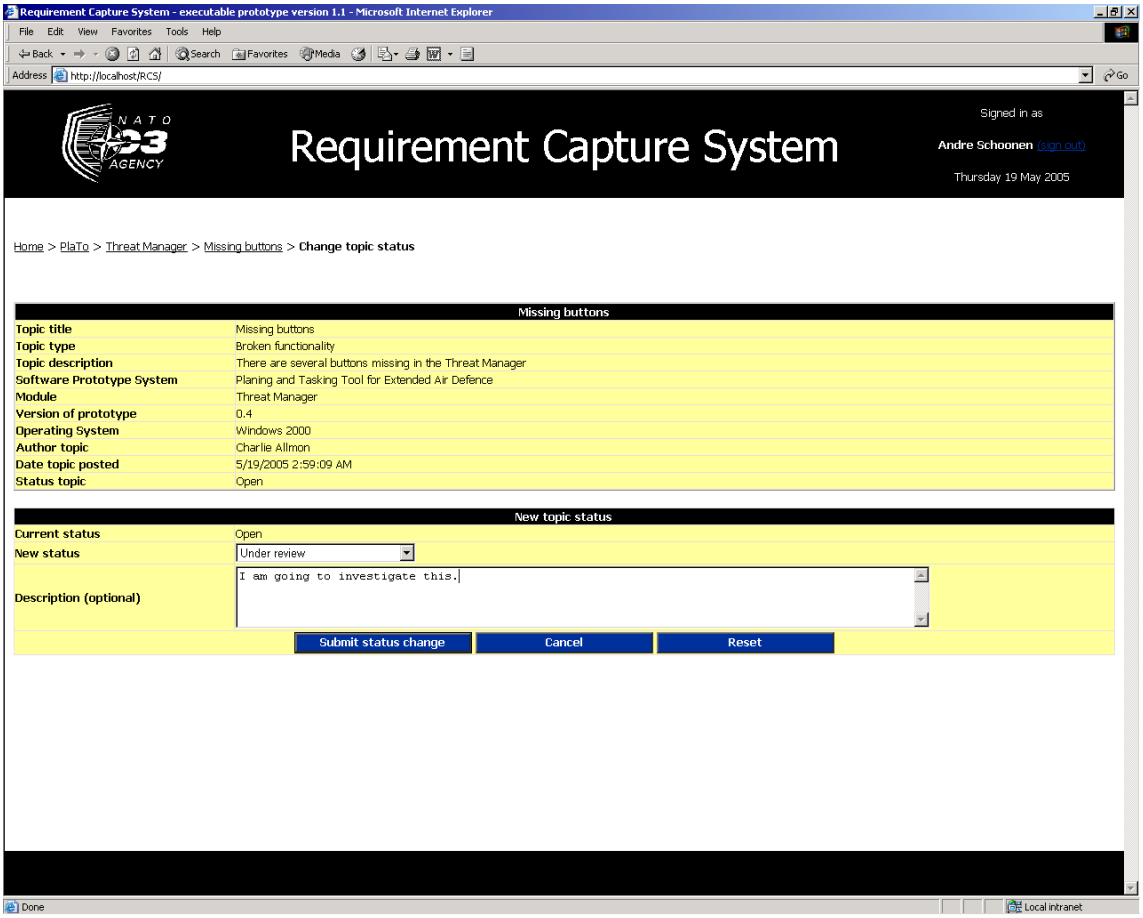


Figure 5.1-10 TopicChangeStatus screen

By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the status change by clicking on the ‘Submit status change’ button.

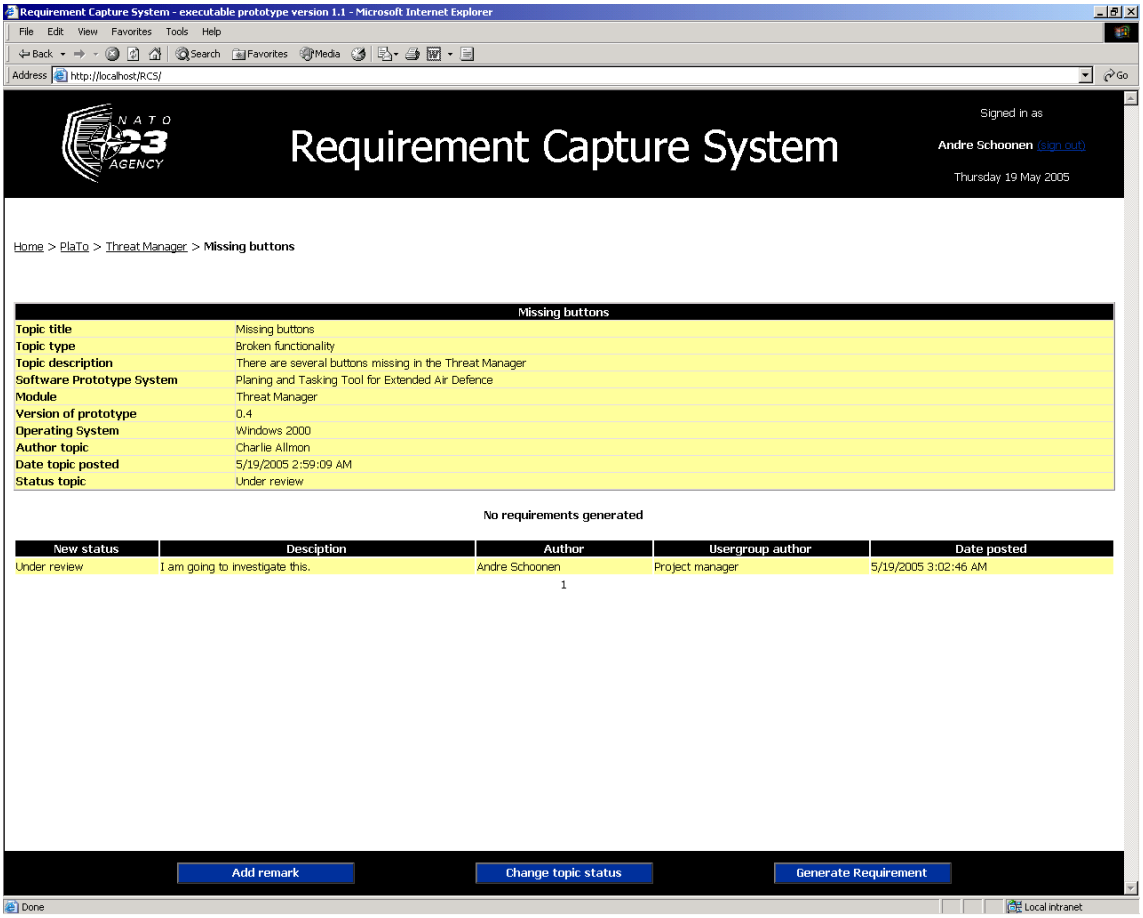


Figure 5.1-11 TopicView screen

When the user submits a topic status change the system will display the ‘TopicView’ screen.
The status change submitted is also displayed in this screen. There is also pager that allows the user to page true the status changes and remarks.

Evaluation results

- If the status of a topic is ‘Closed: rejected’ or ‘Closed: requirement generated’ it should not be possible to change the status of that topic anymore.
- If the users has submitted a new topic automatically there has to be made a new status change with the status ‘Open’. This also involves section 5.1.2 ‘Start a topic’.
- The pager has to be removed.

5.1.5 Add remark

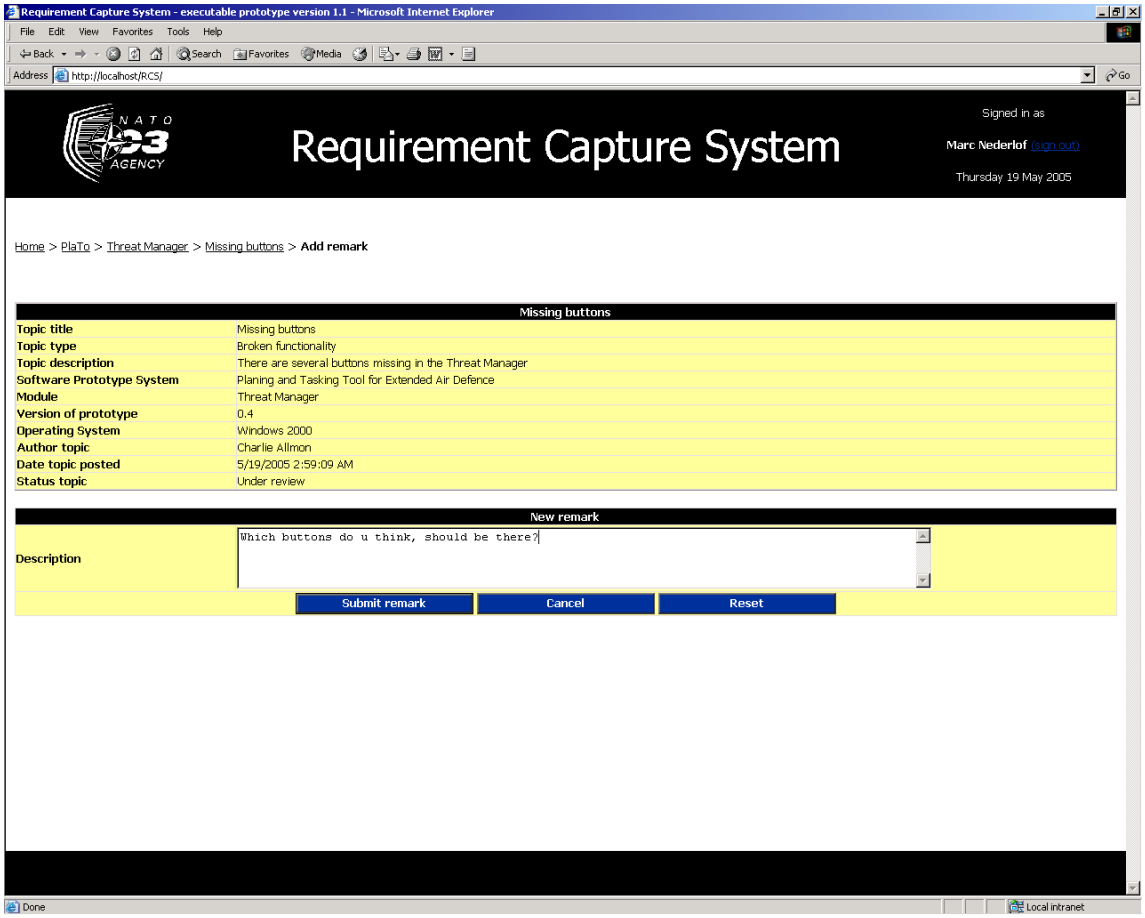


Figure 5.1-12 RemarkAdd screen

When the user clicks on the ‘Add remark’ button in the ‘TopicView’ screen, an additional form is displayed. In this form the user can submit a remark. The only field in this form is ‘Description’ which is a text field. By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the remark by clicking on the ‘Submit remark’ button.

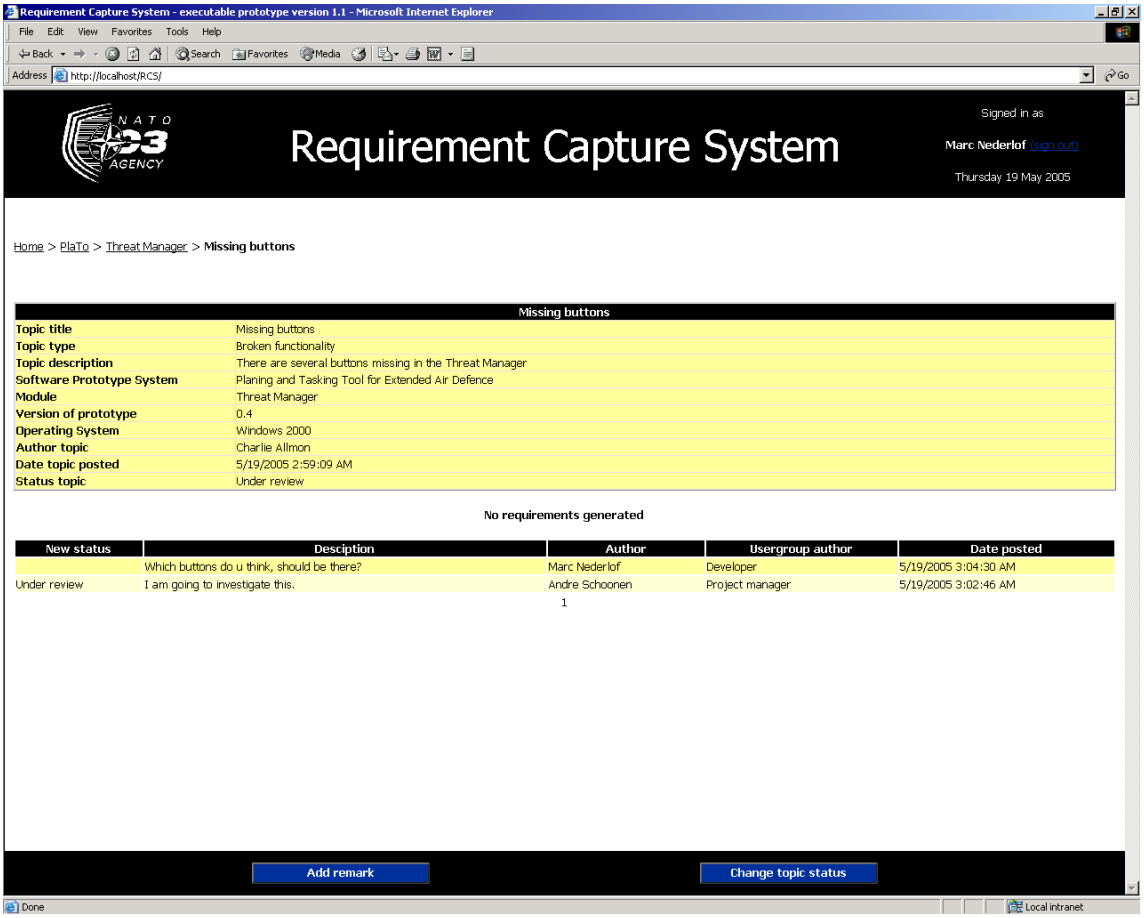


Figure 5.1-13 TopicView screen

When the user submits a remark the system will display the ‘TopicView’ screen. The remark submitted is also displayed in this screen. There is also pager that allows the user to page true the status changes and remarks.

Evaluation results

- When the user has just submitted a remark and clicks on the ‘Back’ button on the browsers he/she will see the previous screen with the text in the text field he/she just provided. If the user submits that remark again, the remark will be submitted twice. There has to come a solution for this problem.

5.1.6 Generate requirement

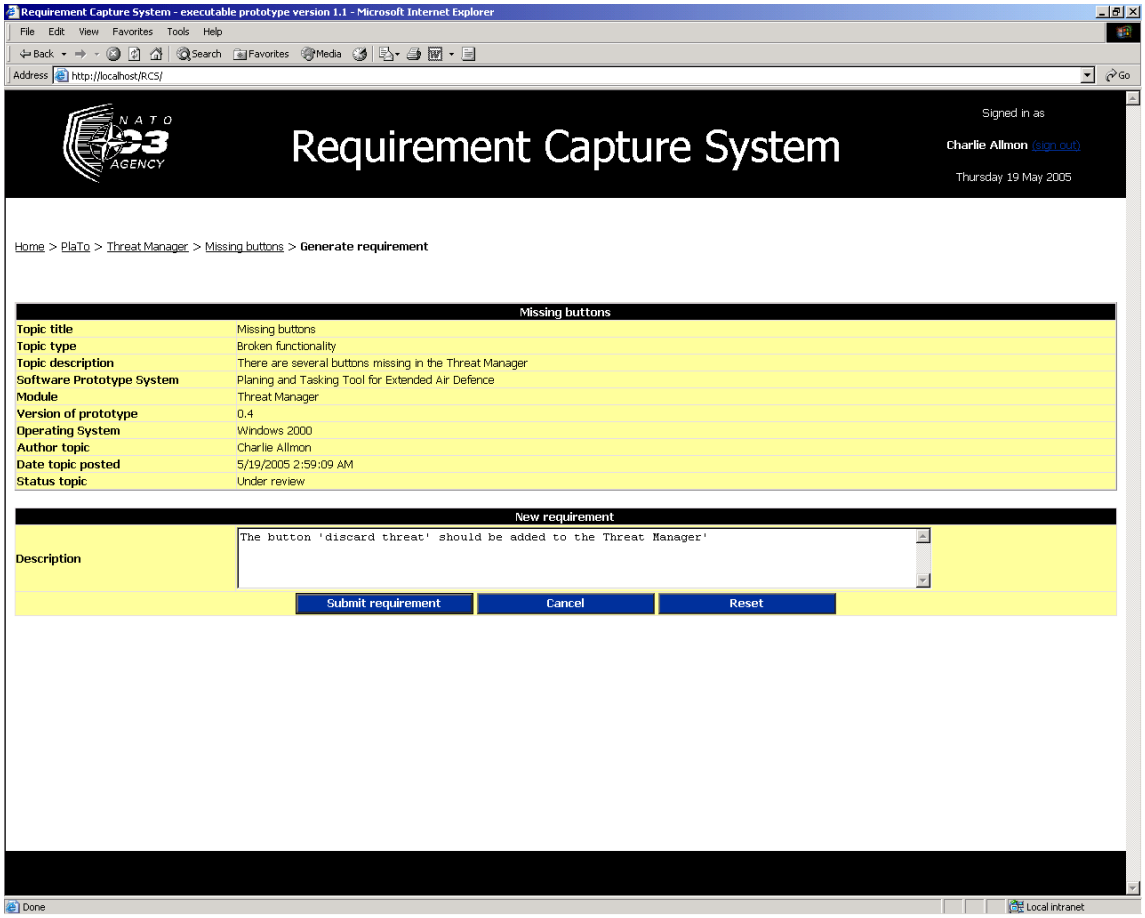


Figure 5.1-14 RequirementGenerate screen

When the user clicks on the ‘Generate Requirement’ button in the ‘TopicView’ screen, an additional form is displayed. In this form the user can submit a requirement. The only field in this form is ‘Description’ which is a text field. By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the requirement by clicking on the ‘Generate Requirement’ button.

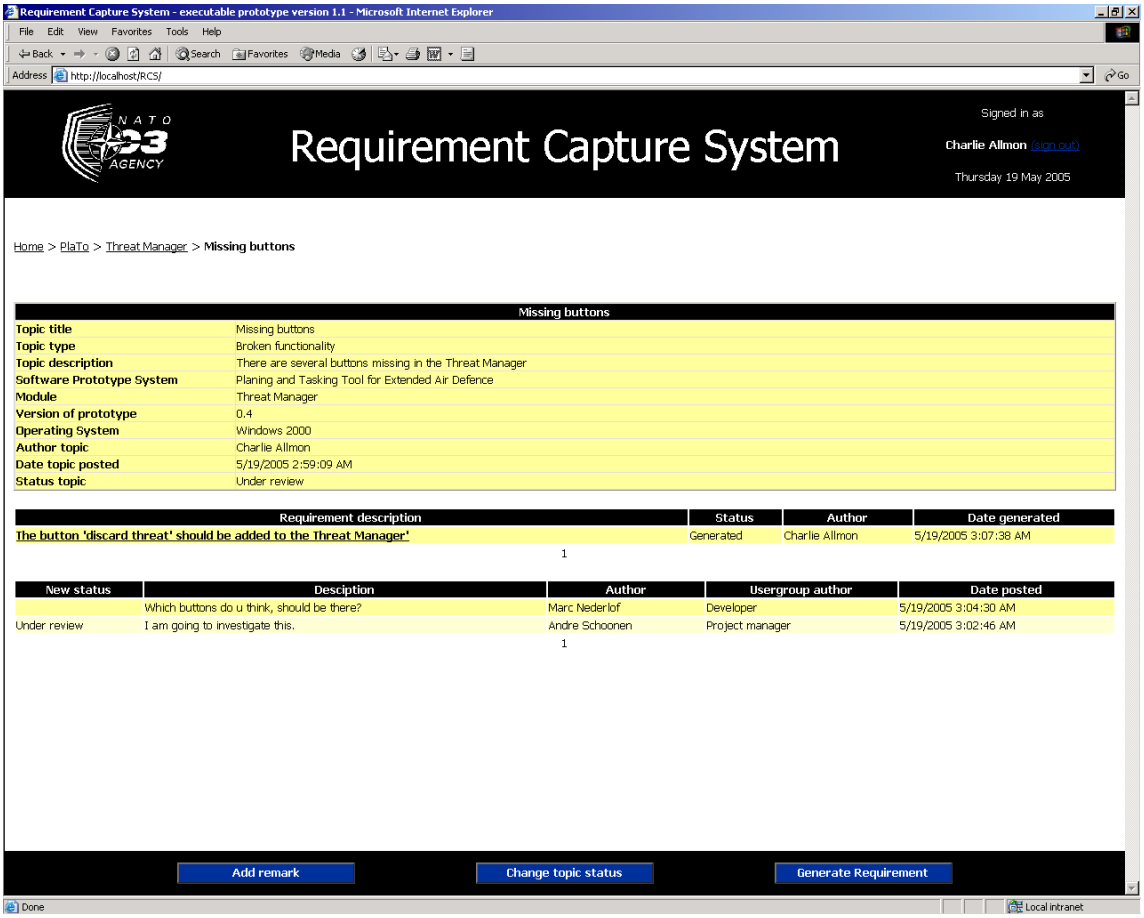


Figure 5.1-15 TopicView screen

When the user submits a requirement the system will display the ‘TopicView’ screen.The requirement generated is also displayed in this screen. There is also pager that allows the user to page true the requirements. To view all information about the requirement the user can click on the Requirement description.

Evaluation results

- Also the pager for the requirements has to be removed.

5.1.7 Change requirement status

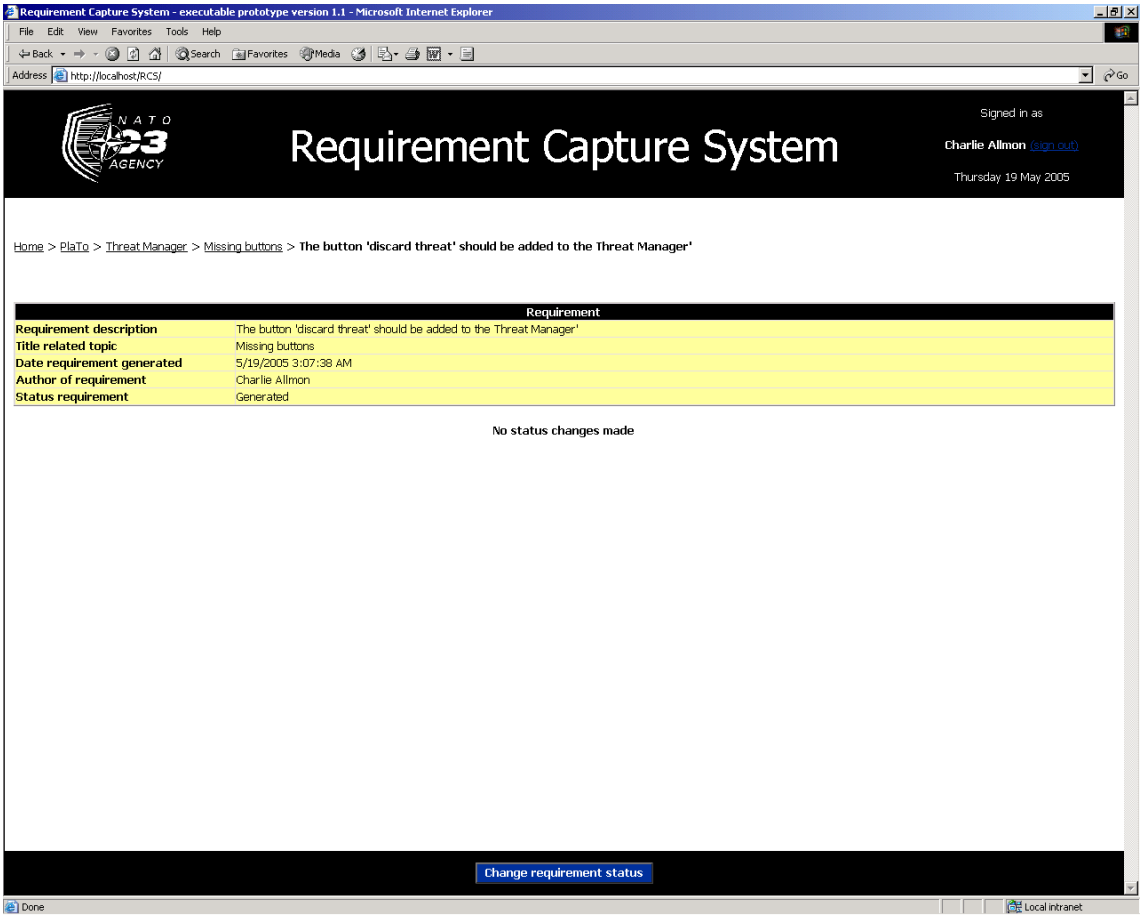


Figure 5.1-16 RequirementView screen

When the user clicks on the Requirement description the ‘RequirementView’ screen will be displayed. The information displayed on a requirement in this screen is:

- Requirement description
- The related topic
- Date requirement generated
- Author of requirement
- Status requirement

By clicking on the ‘Change requirement status’ button the user can change the status of the requirement.

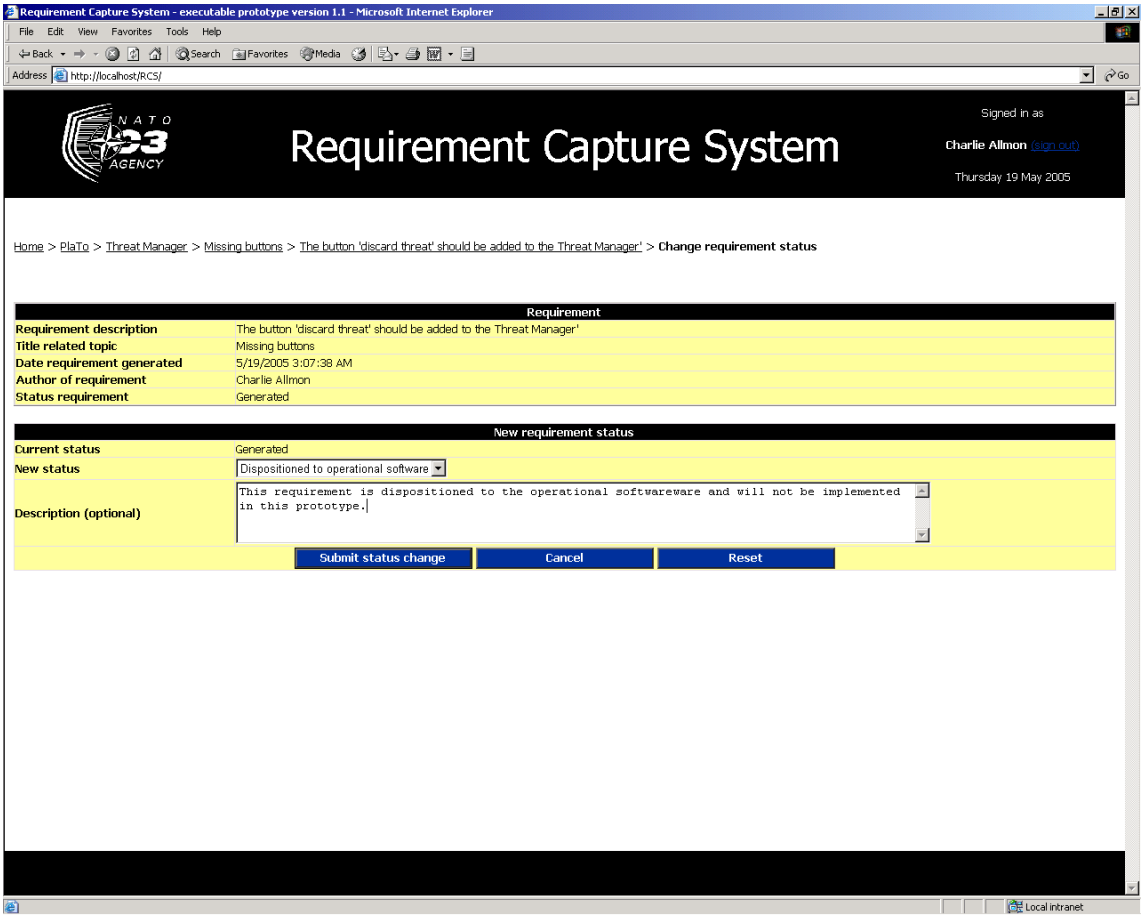


Figure 5.1-17 RequirementChangeStatus screen

When the user clicks on the ‘Change requirement status’ button in the ‘RequirementView’ screen, an additional form is displayed. In this form the user can submit the new status and provide an additional description. The fields in this form are:

- New status (dynamic dropdown list)
- Description (optional) (text field)

By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the status change by clicking on the ‘Submit status change’ button.

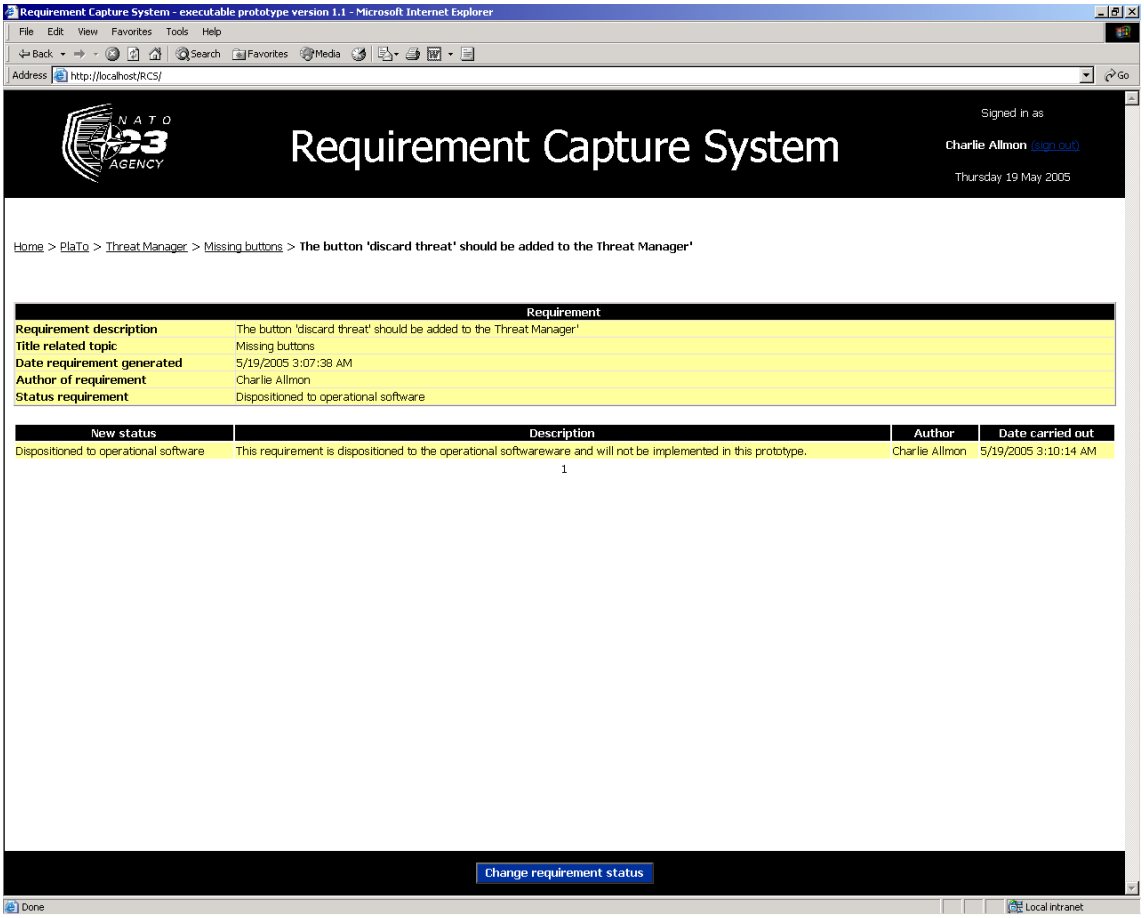


Figure 5.1-18 RequirementView screen

When the user submits a requirement status change the system will display the ‘RequirementView’ screen. The status change submitted is also displayed in this screen. There is also pager that allows the user to page true the status changes.

Evaluation results

- If the users has generated a requirement automatically there has to be made a new status change with the status ‘Generated’. This also involves section 5.1.6 ‘Generate requirement’.
- The pager has to be removed.
- If the status of a requirement is ‘Implemented’ or ‘Dispositioned to operational software’ it should not be able to change the status of the requirement anymore.

5.2 Non-critical use cases

This section will discuss the non-critical use cases that have been partially implemented in the executable prototype.

5.2.1 Sign out

If the is signed in there will be a link named ‘sign out’ in the header in the right section. If the user clicks on this link, the system will sign out the user and display the ‘SignIn’ screen.

5.2.2 View name user signed in

If the is signed in there his/her name will be displayed in the header in the right section.

5.2.3 Select software prototype system

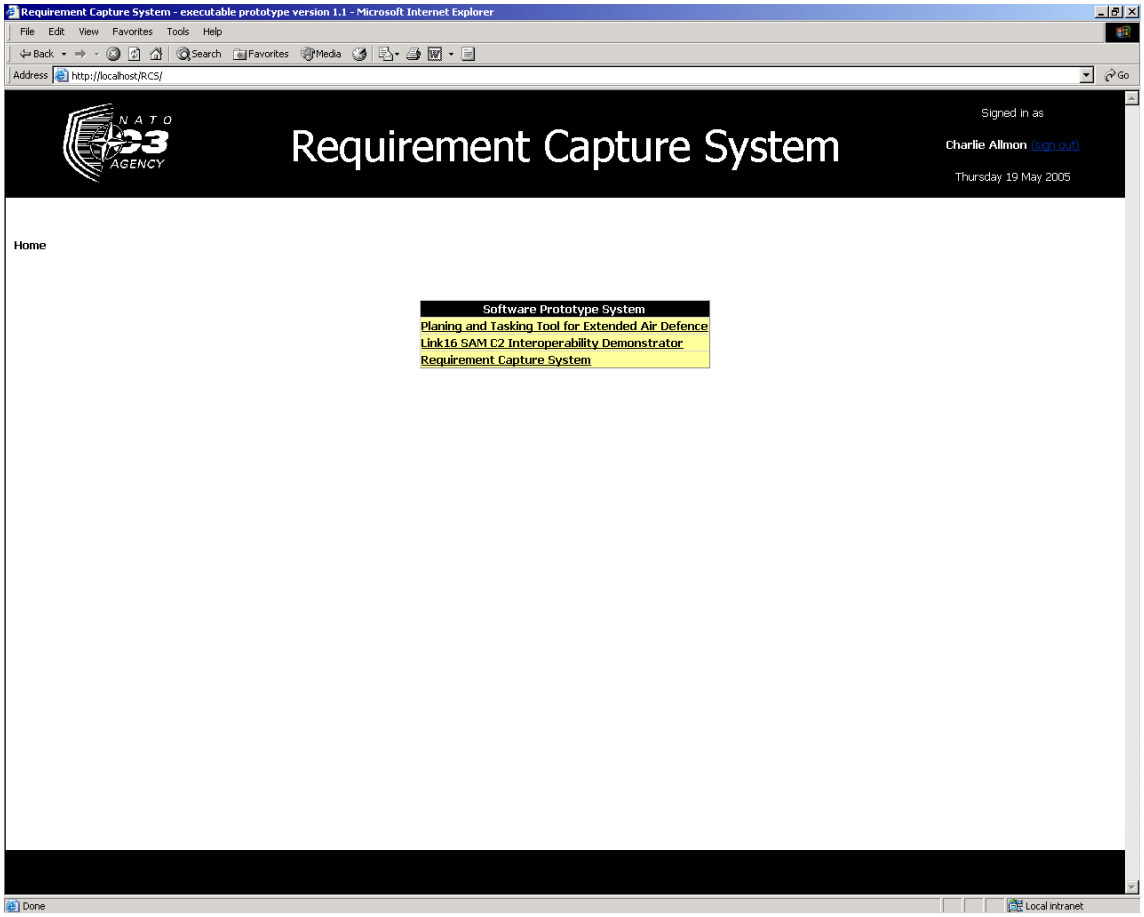


Figure 5.2-1 PrototypeList screen

When the use signs in the ‘PrototypeList’ screen is displayed. In this screen the user can select a software prototype system by clicking on the name of it.

5.2.4 Select a module

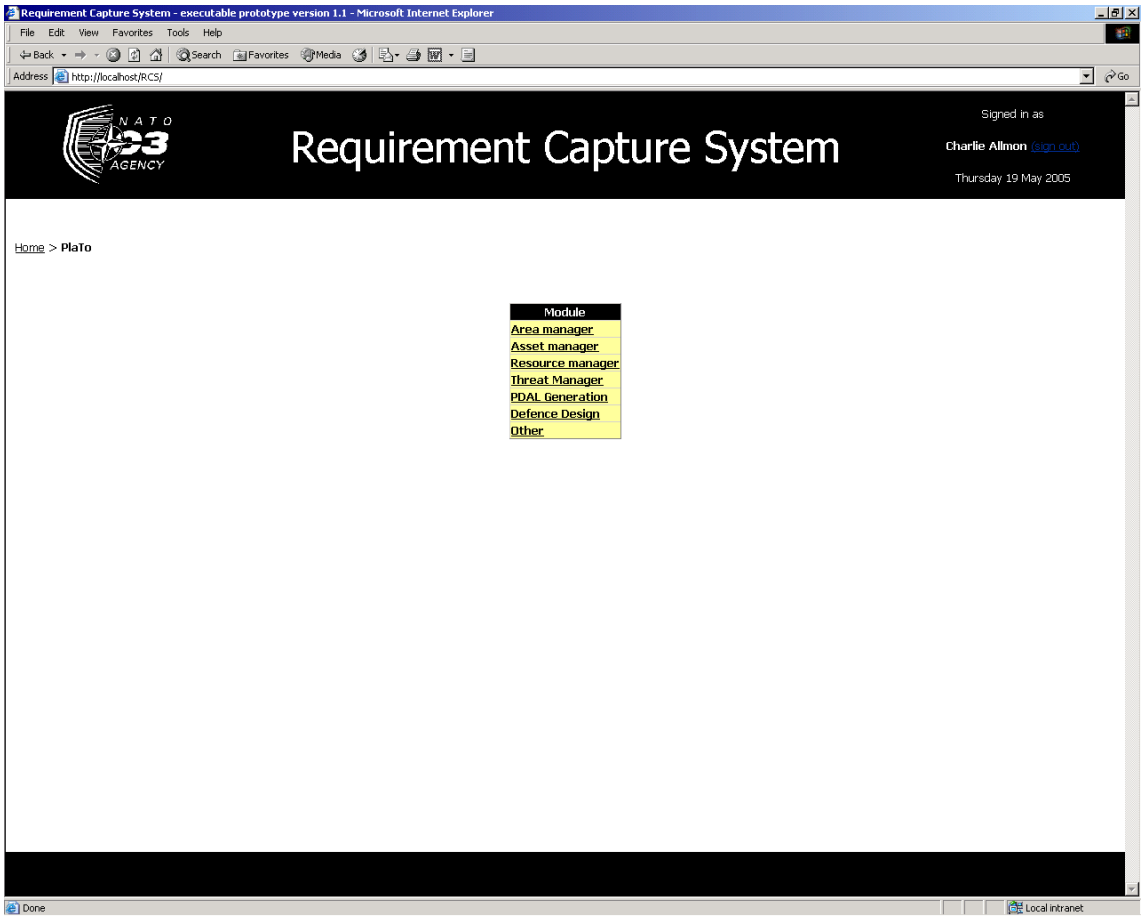


Figure 5.2-2 ModuleList screen

When the user selects a software prototype system the ‘ModuleList’ screen is displayed. In this screen the user can select a software module by clicking on the name of it.

Chapter 6 - Business Case

The Business Case is to describe the economic value of the project. In someway the Business Case will justify carrying out the project. The Business Case should be complete and stable at the end of the Inception phase, but will be refined throughout the project. This chapter will describe the Business Case.

6.1 Costs of the project

The project will incur several costs. There are:

- The student. There will be a compensation for the student who will carry out the project. This will be a fixed amount each day.
- Time from personnel. In order to carry out the project, there has to be a mentor who will guide the student through the organisation and the assignment. Other personnel also have to be available for the student to gain necessary information.
- Facilities. The student will need a working place and the required hardware like a standard Windows PC.
- Software licences. There have to be software licences for the required software. For example: MS Windows, MS Office, MS Visio, MS Project, SharePoint Portal Server, Visual Studio .NET and Rational Rose.

6.2 Benefits of the project

The project will also have some benefits. There are:

- Improved development process. The system that will be developed, will allow end users to give feedback on the prototype systems whenever they want. This should cause the NC3A developers to receive feedback more frequently and easier. In theory, the development process of the software prototypes will improve.
- Better acceptation. Because the end users are more involved with the development of the prototypes, the requirements captured with the prototypes should be better accepted. Eventually this should lead to a better acceptation by the operational community of the added requirements and capabilities to Strategic Command Automated Information System (Bi-SC AIS) and ACCS.

6.3 Major risks

While conducting the project, there are certain risks involved like:

- Software licences. One of the risks is software licence for the required software not being available. In that case there should be looked for alternative software as a solution.

- Non-available personnel. The NC3A personal have a very stressed schedule and they often have to go on travel for duty. If the mentor is not available there has to be another point of contact for the student.

- Rejection of the system. The major risk is the rejection of the system by the end users. If they reject the system and will not use it for any reason, all the benefits of the project will be lost.

Chapter 7 - Development Case

In the Development Case is described which products should be produced and when they should be produced. It also describes which tools can be used for support. The Development Case should be refined in each phase of the project. This chapter will describe the Development Case.

7.1 Products

This section will discuss the products that are going to be produced during the development process.

- Vision Document. The Vision Document will describe the statement of work, the different user groups, the key feature the system has to offer and some global non-functional requirements. The Vision Document should be stable at the end of the inception phase but can be refined at later stages.

- Software Architecture Document. The Software Architecture Document will be the overall documentation document, which will contain use case design models, use case implementation models and use case test models. Therefore this document should be evolving till the end of the project.

- The system. In the first phase the system was just a conceptual prototype consisting of HTML web pages. This prototype has evolved to an executable prototype with functionality in this phase (elaboration). In the construction phase there should be a beta release of the system and finally in the transition phase the prototype should have been evolved to the final release of the system.

- Business Case. The Business Case is to describe the economic value of the project. In someway the Business Case will justify carrying out the project. The Business Case should be complete and stable at the end of the Inception phase, but will be refined throughout the project.

- Development Case. In the Development Case is described which products should be produced and when they should be produces. It also describes how the products should be captured. This document should be refined in every phase.

- Major milestones: Each phase ends with a major milestone. Those milestones will contain all documentation, designs and screenshots.

7.2 Tools

This section will describe the tools that can be used for the project. It is possible that not all the tools mentioned in this chapter are going to be used.

- MS Project: MS Project can be used for making Gantt-diagrams.
- MS Visio. MS Visio can be used for developing UML-diagrams, conceptual web sites or web site maps for prototyping, Windows user interfaces for prototyping and database design. MS Visio can also be used for Gantt-diagrams.
- Rational Rose. Rational Rose can be used for producing UML-diagrams, and generating code based on the UML designs.
- SharePoint Portal Server (SPS) / Windows SharePoint Services (WSS): SharePoint Portal Server / Windows SharePoint Services will be used for the development of the final portal in which the system will be integrated as web part.
- Crimson Editor. Crimson Editor can be used for the development of an ASP.NET web application.
- Visual Studio .NET. Visual Studio .NET can also be used for the development of an ASP.NET web application.

7.3 Planning

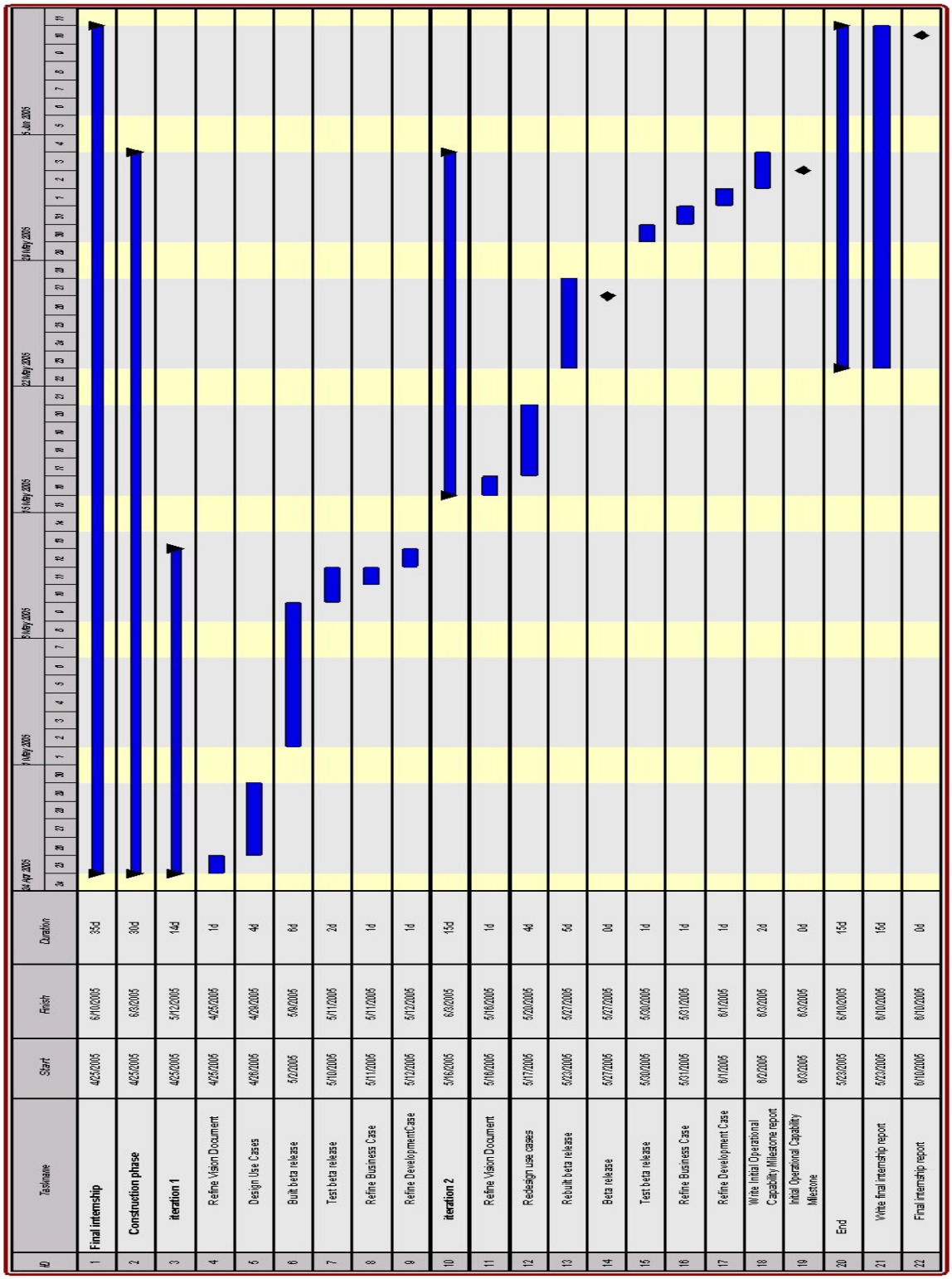
In this section the planning for the next phases or the project will be described.

7.3.1 Global planning

Week	Calendar week	Phase	Activity	
12	17 (25-4 t/m 29-4)	Construction (iteration 1)	Refine Vision Document (1 days) Design use cases (4 days)	
13	18 (2-5 t/m 6-5)	Construction (iteration 1)	Build beta release (6 days) Test beta release (2 days) Refine Business Case (1 day) Refine Development Case (1 day)	
14	19 (9-5 t/m 13-5)	Construction (iteration 1)		
15	20 (16-5 t/m 20-5)	Construction (iteration 2)	Refine Vision Document (1 days) Redesign use cases (4 days)	
16	21 (23-5 t/m 27-5)	Construction (iteration 2) / End	Rebuild beta release (5 days)	Write final internship report (15 days)
17	22 (30-5 t/m 3-6)	Construction (iteration 2) / End	Test beta release (1 day) Refine Business Case (1 day) Refine Development Case (1 day) Write Initial Operational Capability Milestone report (2 days)	
18	23 (6-6 t/m 10-6)	End		

Figure 7.3-1 Global planning

7.3.2 Detailed planning



ID	Taskname	Start	Finish	Duration
1	**Final internship**	4/25/2005	6/10/2005	35d
2	**Construction phase**	4/25/2005	6/3/2005	30d
3	**iteration 1**	4/25/2005	5/12/2005	14d
4	Refine Vision Document	4/25/2005	4/25/2005	1d
5	Design Use Cases	4/26/2005	4/29/2005	4d
6	Built beta release	5/2/2005	5/9/2005	6d
7	Test beta release	5/10/2005	5/11/2005	2d
8	Refine Business Case	5/11/2005	5/11/2005	1d
9	Refine DevelopmentCase	5/12/2005	5/12/2005	1d
10	**iteration 2**	5/16/2005	6/3/2005	15d
11	Refine Vision Document	5/16/2005	5/16/2005	1d
12	Redesign use cases	5/17/2005	5/20/2005	4d
13	Rebuilt beta release	5/23/2005	5/27/2005	5d
14	Beta release	5/27/2005	5/27/2005	0d
15	Test beta release	5/30/2005	5/30/2005	1d
16	Refine Business Case	5/31/2005	5/31/2005	1d
17	Refine Development Case	6/1/2005	6/1/2005	1d
18	Write Initial Operational Capability Milestone report	6/2/2005	6/3/2005	2d
19	Initial Operational Capability Milestone	6/3/2005	6/3/2005	0d
20	End	5/23/2005	6/10/2005	15d
21	Write final internship report	5/23/2005	6/10/2005	15d
22	Final internship report	6/10/2005	6/10/2005	0d

Figure 7.3-2 Detailed planning

References

Boeken

Duthie, G. Andrew, 'Microsoft ASP.NET Step by Step'. Washington, USA, Microsoft Corporation, 2002

Evans, T., '10 Minute Guide to HTML 4.0, third edition'. Que Corporation, 1998

Hatfield, B., 'ASP.NET for Dummies'. Foster City, California, Hungry Minds, Inc., 2001

Kleppe, A. en Warmer, J., 'Praktisch UML, 3de editie'. Soest'. Pearson Education Benelux BV., 2004

Kroll, O. en Kruchten, P, 'The Rational Unified Process Made Easy: a practitioner's guide to the RUP'. Boston, Pearson Education, Inc., 2003

Kruchten, P., 'The Rational Unified Process: an introduction, third edition'. Boston, Pearson Education, Inc., 2004

Langfeld, L., Noel, M. en Spence, C., 'Microsoft SharePoint 2003 Unleashed'. USA, Sams Publishing, 2004

Phillips, L.A., 'Using HTML 4, fourth edition'. Que Corporation, 1998

Quatrani, T., 'Visual Modeling with Rational Rose 2004 and UML'. Boston, Pearson Education, Inc., 2003

Rapporten

Houten, van, J.J., 'Project Plan – Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Houten, van, J.J., 'Lifecycle Objective Milestone, Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Houten, van, J.J., 'Lifecycle Architecture Milestone, Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Schoonen, A., 'PlaTo Design Document – phase 1'. The Hague, NATO C3 Agency, 2004

Schoonen, A., 'PlaTo User Guide, version 0.890'. The Hague, NATO C3 Agency, 2004

Internet

http://www.asp.net

http://www.aspnl.com

http://www.aspnet101.com

http://www.codeproject.com

http://www.datagridgirl.com

http://www.dotnet-guide.com

http://www.dotnet247.com

http://www.dotnetjunkies.com

http://www.microsoft.com

http://www.nato.int

http://www.nc3a.nato.int

Intranet

http://nc3a-intranet/

Glossary

ACC	Air Component Command
ACCS	Air Command and Control System
ActD	Active Defence
ASP	Active Server Pages
ASPX	ASP.NET
Bi-SC AIS	Strategic Command Automatic Information System
BMC3I	Battle Management, Command, Control, Communications and Intelligence
C2	Command and Control
C2RC	Command and Control Resource Centre
CCF	Conventional Counter Force
CCSD	Command and Control System Division
CLR	Common Language Runtime
CRONOS	Crises Response Operational NATO Open Systems
EAD	Extended Air Defence
HHS	Haagse Hogeschool
HQ ARRC	Head Quarters Allied Rapid Reaction Corps
HQ EADTF	Head Quarters Extended Air Defence Task Force
HTML	Hypertext Markup Language
IT&E	Integrated Test & Evaluation
IVIT	Informatievoorzieningen en Informatietechnologie
LSID	Link16 SAM C2 Interoperability Demonstrator
M&S	Modelling and Simulation
MD	Missile Defence
MS	Microsoft
NATO	North Atlantic Treaty Organisation
NC3A	NATO Consultation, Command & Control Agency
PD	Passive Defence

PlaTo	Planning and Tasking Tool
RC	Regional Command
RUP	Rational Unified Process
SDK	Software Developement Kit
SPS	SharePoint Portal Server
SQL	Structured Query Language
SSL	Secure Socket Layer
TMD	Theatre Missile Defence
WSS	Windows SharePoint Services

Table of figures

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Afstudeerverslag externe bijlage IV

Ontwikkeling van een Requirement Capture System voor NAVO software prototype systemen

Externe bijlage IV
Initial Operational Capability Milestone

10 juni 2005

NATO C3 Agency

www.nc3a.nato.int

Haagse Hogeschool

www.hhs.nl

Initial Operational Capability Milestone

Development of a Requirement Capture System for NATO software prototype systems

by
J.J. van Houten

27 May 2005

Foreword

This report is the Initial Operational Capability Milestone of my final internship assignment 'Development of a Requirement Capture System for NATO software prototype systems' at the NATO Consultation, Command & Control Agency (NC3A) in The Hague. This final internship is part of the course 'Informatievoorzieningen en Informatietechnologie' (IVIT) that I am following at the Haagse Hogeschool (HHS) in The Hague.

The Initial Operational Capability Milestone is next to the Project Plan, the Lifecycle Objective Milestone and the Lifecycle Architecture Milestone the fourth milestone of the project and is part of the Rational Unified Process (RUP) developing method.

Readers that are especially interested in the requirements of the system will be referred to chapter three where the required functionality is being described by critical and non-critical use cases. This chapter also describes the refined database design. The beta release that is based on the critical and non-critical use cases is discussed in chapter four.

Again I want to thank the NC3A personnel for their contributions to this document.

27 May 2005, The Hague

J.J van Houten

Table of contents

Chapter 1 - Introduction

The goal of the construction phase is to develop an operational version of the system that can be deployed in the user community. The requirements captured in the inception phase and refined in the elaboration phase will be refined and a beta release will be developed. This prototype will be based on the critical use cases and the non-critical use cases. The construction phase is the third phase of the developing method the Rational Unified Process (RUP). This phase ends with the Initial Operational Capability Milestone that can be captured in a report.

The purpose of this report is to describe the requirements, the database design and the beta release in the construction phase of the project.

In chapter two the vision of the project will be described. Chapter three, Software Architecture, will describe the requirements and the design of the database. Finally in chapter four the beta release that has been built to capture requirements is discussed.

Chapter 2 - Vision

The Vision Document is to create a foundation for common understanding of motivation for building the system, as well as a high-level definition of the system to be built. The Statement of Work is somewhat analogous to parts of the Vision Document. This Vision Document is similar to that one from the Elaboration phase that is documented in the Lifecycle Architecture Milestone.

2.1 Statement of work

2.1.1 General

The intern assignment, described in this program of work will be conducted for the Command and Control Resource Centre (C2RC) of the NATO Consultation, Command & Control Agency (NATO C3 Agency) in The Hague. The NATO C3 Agency (NC3A) develops, procures and implements state of the art Command, Control and Communications (C3) capabilities for NATO and it provides unbiased scientific advice and support to NATO authorities. NC3A is an integrated team of more than 500 professionals, civilian and military, from NATO member nations dedicated to provide to NATO high quality expertise in the domain of Command, Control and Communications.

The Missile Defence (MD) mission area consists of four functional areas. These are:

- Conventional Counter Force (CCF)
- Active Defence (ActD)
- Passive Defence (PD)
- Battle Management, Command, Control, Communications and Intelligence (BMC3I)

In support of these four functional areas, C2RC provides core competences in the following areas:

- Concept Development Support (Prototyping, Exercises)
- Architecture and Architecture Development
- Modelling & Simulation (M&S)
- Integrated Test & Evaluation (IT&E)
- Systems: MD Weapon, Sensors and BMC3 Systems
- MD Decision Support Tools

2.1.2 Problem description

Amongst other things, NC3A is developing two missile defence related software prototypes:

- Link16 SAM C2 Interoperability Demonstrator (LSID) for Theatre Missile Defence (TMD) Engagement Coordination and Situational Awareness.
- Extended Air Defence (EAD) Planning and Tasking Tool (PlaTo) for TMD planning.

The two software prototypes described above have been used operationally in several NATO and multinational exercises. Furthermore the systems have been installed in a number of locations. The goal for this installation and exercise participation is to capture user requirements and feedback on the prototypes. The resulting prototypes and requirements captured through this process will at a later stage be used to extend the two NATO Command and Control (C2) systems: the Strategic Command Automated Information System (Bi-SC AIS) and the Air Command and Control System (ACCS) with Battle Management and Engagement Coordination functionality.

At the moment, capturing of these user requirements can only be done when NC3A personnel are on site, and the tracking of user requirements is not done in a structured way. Operators might report user requirements back through mechanisms such as email but there is no proper vehicle for them to actually log remarks, requirements and feedback in a logical standardised way and provide them to NC3A.

Feedback received from users is not collected in a central repository and can contain incomplete or misleading information. This current situation does not sufficiently meet NC3A's requirements.

2.1.3 Goal of the assignment

The goal of this assignment is to develop a system that will allow the operators to log remarks, requirements and feedback through an efficient and structured process.

A process has to be constructed that will facilitate the reporting of user feedback and requirements with respect to the two prototypes to NC3A. Ideally the information provided bye the users should be tied into the development process of the named prototypes. The process has to be scalable and should be defined in a clear and structured way. Once the feedback process has been properly worked out and approved, a system has to be implemented to support it.

2.2 Target users

This section will describe the different user groups of the system. Roughly there are two user groups. These are the developers of the software prototype systems and the users of the software prototype systems.

2.2.1 Developer user group

The developer user group is part of C2RC, which belongs to the Command and Control System Division (CCSD) at NC3A.

At his moment, the developer user group exists of four staff member and three contractors.
Two staff members are working on LSID and three contractors are working on PlaTo. The Senior Scientist is in charge of the development of LSID and PlaTo. LSID and PlaTo are part of the Project TM, which is lead by the Principal Scientist. The next table shows additional information about the developer user group.

Name	Function	Room	E-mail	Phone
A. Schoonen	In charge of development	470	Andre.Schoonen@nc3a.nato.int	070-3743773
C. Allmon	Project Manager TM	359A	Charlie.Allmon@nc3a.nato.int	070-3743772
T. Bingham	Programmer on LSID	359B	Theo.Bingham@nc3a.nato.int	070-3743775
M. Bugru	Programmer on LSID	467	Metin.Bugru@nc3a.nato.int	070-3743812 (TMD Lab)
M. Nederlof	Programmer on PlaTo	467	Marc.Nederlof@nc3a.nato.int	070-3743812 (TMD Lab)
E. van der Koogh	Programmer on PlaTo	467	Erwin.van.der.Koogh@nc3a.nato.int	070-3743812 (TMD Lab)
A. Copner	Programmer on PlaTo	467	Alan.Copner@nc3a.nato.int	070-3743812 (TMD Lab)

Figure 2.2-1 Developer user group

2.2.2 Test site user group

At this moment the users of PlaTo are limited to the Head Quarters Extended Air Defence Task Force (HQ EADTF) and the Head Quarters Allied Rapid Reaction Corps (HQ ARRC). It is possible that PlaTo is going to be used at the Regional Command (RC) South in Naples (Italia) and the Air Component Command (ACC) South in Izmir (Turkey). There are no users for LSID at this moment.

2.3 System activities

This section describes the system activities by using features and key use cases.

2.3.1 System features

Feature No.1	
Name	Provide feedback.
Description	Allow Test site users and Developers to provide feedback on the software prototype systems LSID and PlaTo.
Value	1.
Complexity	Low.
Priority	1.

Figure 2.3-1 Feature No. 1 Provide feedback

Feature No.2	
Name	View feedback
Description	Allow Test site users and Developers to view all feedback provided on the software prototype systems LSID and PlaTo.
Value	2.
Complexity	Low.
Priority	2.

Figure 2.3-2 Feature No.2 View feedback

Feature No.3	
Name	Change feedback status
Description	Allow Developers to change the status of provided feedback on the software prototype systems LSID and PlaTo.
Value	3.
Complexity	High.
Priority	3.

Figure 2.3-3 Feature No.3 Change feedback status

Feature No.4	
Name	Generate requirement
Description	Allow Developers to generate a requirement based on feedback provided on the software prototype systems LSID and PlaTo.
Value	1.
Complexity	Medium.
Priority	4.

Figure 2.3-4 Feature No.4 Generate requirement

Feature No.5	
Name	Change requirement status
Description	Allow Developers to change the status of a generated requirement.
Value	4.
Complexity	High.
Priority	4.

Figure 2.3-5 Feature No.5 Change requirement status

2.3.2 Key use cases

According to the system features discussed in section 2.3.1 the following key actors can be identified: Test site user and Developer. The following key use cases can be identified: Provide feedback, View feedback, Change feedback status, Generate requirement and Change requirement status. The following figures will show these key use cases.

Key use case No. 1	
Name	Provide feedback.
Assumption	The user has signed in.
Actors	Test site user, Developer.
Description	(1) The user tells the system that he/she want to enter feedback. (2) The system asks the user on which prototype he wants to gives feedback. (3) The user answers that question and (4) the system will ask the user for his feedback. The user enters his/her feedback and tells the system that he/she wants to submit it. (5) The system will confirm that the feedback has been submitted.
Exceptions.	None.
Result	Feedback is added.

Figure 2.3-6 Key use case No.1 Provide feedback

Key use case No. 2	
Name	View feedback.
Assumption	The user has singed in.
Actors	Test site user, Developer.
Description	(1) The user tells the system that he/she wants to view feedback. (2) The system shows the user the requested feedback.
Exceptions.	[No feedback found.] The system will tell the user that there is no feedback found.
Result	Feedback is displayed.

Figure 2.3-7 Key use case No.2 View feedback

Key use case No. 3	
Name	Change feedback status.
Assumption	The user has signed in as Developer and has selected feedback.
Actors	Developer.
Description	(1) The user tells the system that he/she wants to change the status of the feedback. (2) The system asks to user to what status he/she wants to change the status of the feedback and (3) the user answers that question.
Exceptions.	None.
Result	Feedback status is changed.

Figure 2.3-8 Key use case No.3 Change feedback status

Key use case No. 4	
Name	Generate requirement.
Assumption	The user has signed in as Developer and has selected feedback.
Actors	Developer
Description	(1) The user tells the system that he/she wants to generate a requirement based on the selected feedback. (2) The system asks the user for the requirement and (3) the user provides the requirement.
Exceptions.	None.
Result	Requirement is generated.

Figure 2.3-9 Key use case No.4 Generate requirement

Key use case No. 4	
Name	Change requirement status.
Assumption	The user has signed in as developer and has selected a requirement.
Actors	Developer.
Description	(1) The user tells the system that he/she wants to change the status of the selected requirement. (2) The system asks the user to what status he/she wants to change the requirement and (3) the user answers that question.
Exceptions.	None.
Result	Requirement status is changed.

Figure 2.3-10 Key use case No.4 Change requirement status

2.3.3 Use case diagram

The following figure shows the use case diagram based on the key use cases previously described.

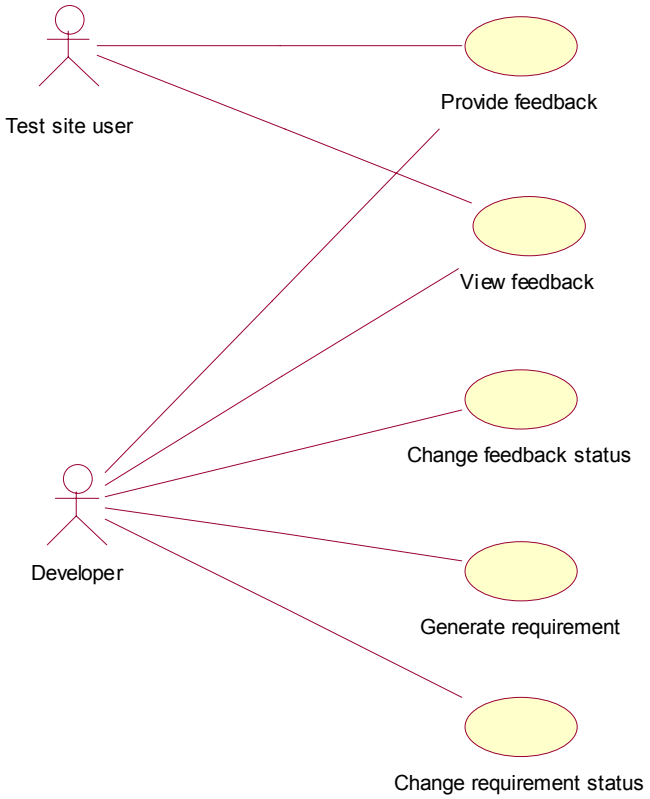


Figure 2.3-11 Use case diagram

2.4 Non-functional requirements

This section briefly describes the non-functional requirements.

- Operation System. The system has to be compatible with Windows 2000 and preferably also with Unix Solaris.

- Database support. There are no specific requirements for choosing the database environment to use. It probably would be SQL Sever 2000 or MS Access.

- Programming environment. It is now clear that there has to be a web-based application. Preferably the web application has to be developed with SharePoint Portal Server (SPS) or Windows SharePoint Services (WSS) that produces ASP.NET pages (ASPX). It is also possible to build a web application using ASP.NET with for example Visual Studio .NET as programming environment. ASP.NET is a Windows based and unfortunately is not compatible with Unix Solaris.

Chapter 3 - Software Architecture

The Software Architecture Document is the overall documentation document that will contain all UML analysis and design models. This document will be refined in every iteration and phase during the project. This chapter describes the Software Architecture Document.

3.1 Use cases

3.1.1 Actors

Actor No.1	
Name	Test site user
Description	A Test site user is a person who is using and testing the software prototype systems developed by NC3A.

Figure 3.1-1 Actor No.1 Test site user

Actor No.2	
Name	Analyst
Description	An Analyst is a person who is designing software prototype systems at NC3A.

Figure 3.1-2 Actor No.2 Analyst

Actor No.3	
Name	Developer
Description	A Developer is a person who is developing software prototype systems at NC3A.

Figure 3.1-3 Actor No.3 Developer

Actor No.4	
Name	Project manager
Description	A Project manager is a person who is in charge of a project at NC3A.

Figure 3.1-4 Actor No.4 Project manager

Actor No.5	
Name	Administrator
Description	An Administrator is a person who is going to be the administrator for the Requirement Capture System that has to be developed.

Figure 3.1-5 Actor No.5 Adminstrator

3.1.2 Critical use cases

Critical use case	No. 1
Name	Sign in.
Assumption	The system is started up.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system asks the user his/her username and password. (2) The user enters his/her username and password and tells the system he/she wants to sign in. (3) The system signs in the user en shows the available software prototype systems.
Exceptions	[Username unknown.] The system tells the user that the username is unknown. [Username and password combination not valid.] The system tells the user that the username and password combination is not valid.
Result	User has signed in.
Priority	1.

Figure 3.1-6 Critical use case No.1 Sign in

Critical use case	No. 2
Name	Start a topic.
Assumption	User has signed in and has selected a software prototype system and a module.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to start a topic. (2) The system asks him/her the topic title, topic description, topic type, version of the software prototype system and the operating system that has been used. (3) The user answers those questions and tells the system that he/she wants to submit the topic. (4) The system shows the user the topic he/she has just provided.
Exceptions	None.
Result	Topic is started.
Priority	1.

Figure 3.1-7 Critical use case No.2 Start a topic

Critical use case	No. 3
Name	View topic.
Assumption	User has signed in and has selected a software prototype system and a module or searched for feedback.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system shows the topic entries that previously have been submitted and belong to the chosen module or the search query that has been submitted. A topic entry contains the topic title, the module where the topic is about, the topic type, the status of the topic, the author of the topic, the date of the last posted remark on this topic, the author of the last posted remark on this topic, the number of requirements generated and the date of last requirement generated. (2) The user tells the system on which topic entry he/she wants to look at. (3) The system shows the topic title, topic description, the topic type, the module where the topic is about, the version of the software prototype system that is used, the operating system that is used, the author of the topic, the status of the topic. If present, (4) the system will display the remarks and/or status changes that are added/made to this topic. The system will display the type (this can be a remark of a topic status change), a description, the author, the user group of the author and the date the remark is added or the status change is made. If present, (5) the system will display the requirements that are generated from this topic. The system will display the requirement description, the requirement status, the author of the requirement and the date the requirement was generated.
Exceptions	[No topics present or found.] The system will tell the user there are no topics present or found. [There are no requirement generated.] The system will tell the user there are no requirements generated.
Result	Topic is displayed.
Priority	1.

Figure 3.1-8 Critical use case No.3 View topic

Critical use case	No. 4
Name	Change topic status.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a software prototype system, a module and a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to change the status of the topic. (2) The system asks the user what status he/she wants to give the topic and asks for an additional description. (3) The user tells the system what the new status of the topic must be, fills in an additional description and tells the system that he/she wants to submit the status change. (4) The system shows the user the updated topic information.
Exceptions	[Topic status is already 'Closed'.] The system tells the user that the status of the topic is 'Closed' and cannot be changed anymore.
Result	Topic status is changed.
Priority	2.

Figure 3.1-9 Critical use case No.4 Change topic status

Critical use case	No. 5
Name	Add remark.
Assumption	User has signed in as Test site user, Analyst, Developer, Project manager or Administrator and has selected a prototype, a module and a topic.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to add a remark to current topic. (2) The system asks the user for a description of the remark and shows information that is already automatically know by the system. This information contains the type (that will be remark), the author, the user group of the author and that date. (3) The user fills in a description and tells the system that he wants to submit the remark. (4) The system tells the user that the remark has been submitted and shows the updated topic information.
Exceptions	None.
Result	Remark is added to topic.
Priority	2.

Figure 3.1-10 Critical use case No.5 Add remark

Critical use case	No. 6
Name	Generate requirement.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a prototype, a module and a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to generate a requirement from the current topic. (2) The system asks the user to fill in the requirement description and shows information that is already automatically known by the system. This is information contains the author, user group of the author and the date. (3) The user fills in the description of the requirement and tells the system that he/she wants to submit the requirement. (4) The system tells the user that the requirement has been submitted and shows the update topic information.
Exceptions	None.
Result	Requirement is generated.
Priority	3.

Figure 3.1-11 Critical use case No.6 Generate requirement

Critical use case	No. 7
Name	Change requirement status.
Assumption	User has signed in as Developer, Project manager or Administrator and has selected a prototype, a module, a topic.
Actors	Developer, Project manager, Administrator.
Description	(1) The user tells the system for which requirement he/she wants to change the status. (2) The system asks the user what status he/she wants to give the requirement and asks for an additional description. (3) The user tells the system what the new status of the requirement must be, fills in an additional description and tells the system that he/she wants to submit the status change. (4) The system shows the user the updated topic information.
Exceptions	[Requirement status is already ‘Implemented’.] The system tells the user that the status of the requirement is ‘Implemented’ and cannot be changed anymore.
Result	Requirement status is changed.
Priority	3.

Figure 3.1-12 Critical use case No.7 Change requirement status

3.1.3 Non-critical use cases

Use case No. 1	
Name	Sign out.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants so sign out. (2) The system signs out the user.
Exceptions	None.
Result	User has signed out.

Figure 3.1-13 Non-critical use case No.1 Sign out

Use case No. 2	
Name	View name user signed in.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system shows the name of the user that is signed in.
Exceptions	None.
Result	Name user signed in is displayed.

Figure 3.1-14 Non-critical use case No.2 View name user signed in

Use case No. 3	
Name	Select software prototype system.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system shows the name of all software prototype systems, the number of requirements that are generated for the software prototype system and the date of the last requirement generated. (2) The user select can sort the list of software prototype systems by name, number of requirements generated and date last requirement generated. (3) The user selects a software prototype system by clicking on the name.
Exceptions	[No software prototype systems available.] The system tells the user that there are no software prototype systems available.
Result	A software prototype system is selected.

Figure 3.1-15 Non-critical use case No.3 Select software prototype system

Use case No. 4	
Name	Select a module.
Assumption	User has signed in and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system shows the name of modules that belong to the selected software prototype system, the number of topics that are posted on the software prototype system, the date of the last topic posted, the number of requirements that are generated for the module and the date of the last requirement generated for the module. (2) The user select can sort the list of modules by name, number of topics posted, date last topic posted, number of requirements generated and date last requirement generated. (3) The user selects a module by clicking on the name.
Exceptions	[No modules available.] The system will tell the user that there are no modules for the selected software prototype system.
Result	A module is selected.

Figure 3.1-16 Non-critical use case No.4 Select a module

Use case No. 5	
Name	Search for topics.
Assumption	User has signed in and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The system asks the user to fill in a search query. (2) The user fills in a search query and submits it. (3) The system will display all found search results.
Exceptions	[There are no search results.] The system tells the user that there are no search results matching the search query.
Result	Topic list is displayed.

Figure 3.1-17 Non-critical use case No.5 Search for topics

Use case No. 6	
Name	View topics user is interested in.
Assumption	User has signed and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to see a list of topics that are of his interest. (2) The system shows a list of topics with topics that are of the users interest.
Exceptions	[No topics found.] The system will tell the user that there are no topics that are marked as interesting by the user.
Result	A topic list is displayed.

Figure 3.1-18 Non-critical use case No.6 View topics user is interested in

Use case No. 7	
Name	View topics user is involved in.
Assumption	User has signed and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to see a list of topics that he is involved in. (2) The system shows a list of topics with topics which the user is involved in.
Exceptions	[No topics found.] The system will tell the user that he/she is not involved in any topic at this moment.
Result	A topic list is displayed.

Figure 3.1-19 Non-critical use case No.7 View topics user is involved in

Use case No. 8	
Name	Mark topic as interesting.
Assumption	User has signed and has selected a prototype, a module and a topic that is marked as uninteresting.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to mark the current topic as interesting. (2) The system marks the current topic as interesting.
Exceptions	None.
Result	Topic is marked as interesting.

Figure 3.1-20 Non-critical use case No.8 Mark topic as interesting

Use case No. 9	
Name	Unmark a topic as interesting.
Assumption	User has signed and has selected a prototype, a module and a topic that is marked as interesting.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to unmark the current topic as interesting. (2) The system unmarks the current topic as interesting.
Exceptions	None.
Result	Topic is unmarked as interesting.

Figure 3.1-21 Non-critical use case No.9 Unmark a topic as interesting

Use case No. 10	
Name	View requirements generated.
Assumption	User has signed in and has selected a software prototype system.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to view the requirements that are generated for the selected software prototype system. (2) The system shows the user all the requirements that are generated for the selected software prototype system.
Exceptions	[No Requirements found.] The system will tell the user that there are no requirements generated for the selected software prototype system.
Result	Generated Requirements are displayed.

Figure 3.1-22 Non-critical use case No.10 View requirements generated

Use case No. 11	
Name	Generate requirement report.
Assumption	User has signed in as Project manager or Administrator and has selected a software prototype system.
Actors	Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to generate a Requirement Report for the selected software prototype system. (2) The system will generate a Requirement Report.
Exceptions	[There are no requirements generated for the selected software prototype system.] The system will tell the user that there are no requirements generated for the selected software prototype system and therefore it is not possible to generate a Requirement Report.
Result	Requirement Report is generated.

Figure 3.1-23 Non-critical use case No.11 Generate requirement report

Use case No. 12	
Name	Modify own user profile.
Assumption	User has signed in.
Actors	Test site user, Analyst, Developer, Project manager, Administrator.
Description	(1) The user tells the system that he/she wants to modify his/her own user profile. (2) The system asks the user what changes he/she wants to make and (3) the user answers that question. (4) The system will carry out the modifications to the user profile.
Exceptions	None.
Result	User profile is modified.

Figure 3.1-24 Non-critical use case No.12 Modify own user profile

3.1.4 Other non-critical use cases

This section describes the non-critical use cases that are not yet implemented in the Beta Release.

Use case No. 13	
Name	Change user status.
Assumption	User has signed in as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to change the status of a certain user. (2) The system asks the user to what status and (3) the user answers that question with 'active' or 'inactive'.
Exceptions	[The status of the user is the same as the chosen status.] The system tells the user that the user status is already 'active' or 'inactive'.
Result	User status has changed.

Figure 3.1-25 Non-critical use case No.13 Change user status

Use case No. 14	
Name	Add user.
Assumption	User has signed in as Administrator
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to add a new user. (2) The system asks the user what the properties for the new user are (3) and the user answers that question. (4) The system will add a new user.
Exceptions	None.
Result	New user is added.

Figure 3.1-26 Non-critical use case No.14 Add user

Use case No. 15	
Name	Modify user profile.
Assumption	User has signed in as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to modify a certain user. (2) The system asks the user what changes he/she wants to make and (3) the user answers that question. (4) The system will carry out the modifications to the user profile.
Exceptions	None.
Result	User profile is modified.

Figure 3.1-27 Non-critical use case No.15 Modify user profile

Use case No. 16	
Name	Modify software prototype system.
Assumption	User has signed as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to modify the properties of a certain software prototype system. (2) The system asks what changes he/she wants to make and (3) the user answers that question.
Exceptions	None.
Result	Software prototype system is modified.

Figure 3.1-28 Non-critical use case No.16 Modify software prototype system

Use case No. 17	
Name	Add software prototype system.
Assumption	User has signed in as Administrator.
Actors	Administrator.
Description	(1) The user tells the system he/she wants to add a new software prototype system. (2) The system asks the name and properties of the software prototype system and (3) the user answers that question.
Exceptions	None.
Result	New prototype is added.

Figure 3.1-29 Non-critical use case No.17 Add software prototype system

Use case No. 18	
Name	Delete software prototype system.
Assumption	User has signed in User has signed as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete a certain software prototype system and (2) the system deletes the software prototype system.
Exception	[There are topics related to the software prototype system.]
Result	Software prototype system is deleted.

Figure 3.1-30 Non-critical use case No.18 Delete software prototype system

Use case No. 19	
Name	Delete user.
Assumption	User has signed in as Administrator.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete a certain user. (2) The system will delete the user.
Exceptions	[There are still topics related to the user.]
Result	User status has changed.

Figure 3.1-31 Non-critical use case No.19 Delete user

Use case No. 20	
Name	Delete remark.
Assumption	User has signed in as Administrator and has selected a software prototype system, a topic and a remark.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete the selected remark.
Exceptions	None.
Result	Remark is deleted.

Figure 3.1-32 Non-critical use case No.20 Delete remark

Use case No. 21	
Name	Delete requirement.
Assumption	User has signed in as Administrator and has selected a software prototype system and requirement.
Actors	Administrator.
Description	(1) The user tells the system that he/she wants to delete the selected requirement
Exceptions	None.
Result	Requirement is deleted.

Figure 3.1-33 Non-critical use case No.21 Delete requirement

3.2 Database design

For the development of the beta release it is necessary to redesign and implement a database. This starts with redesigning the class diagram that represents the real world. This class diagram will then be translated to a database model diagram. This chapter will describe the design of the database.

3.2.1 Class diagram

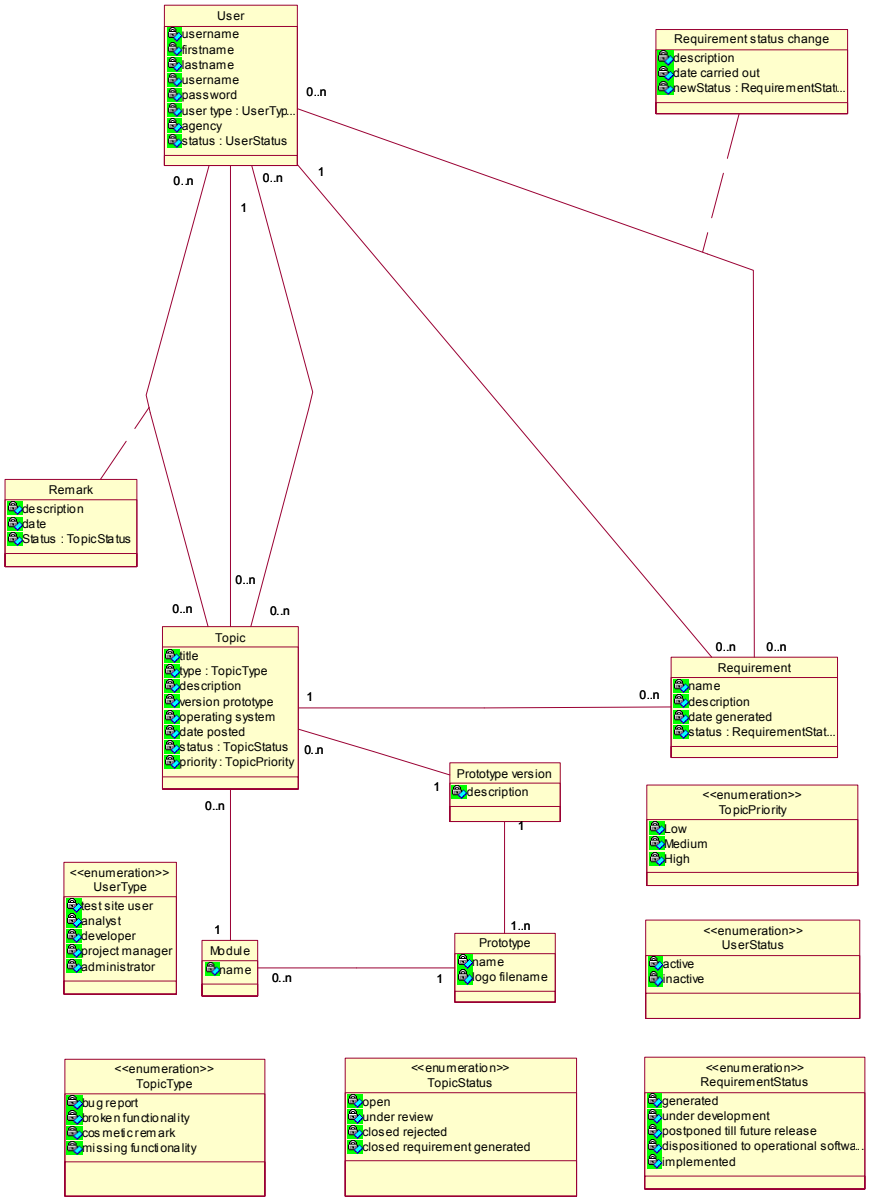


Figure 3.2-1 Class diagram

3.2.2 Database model diagram

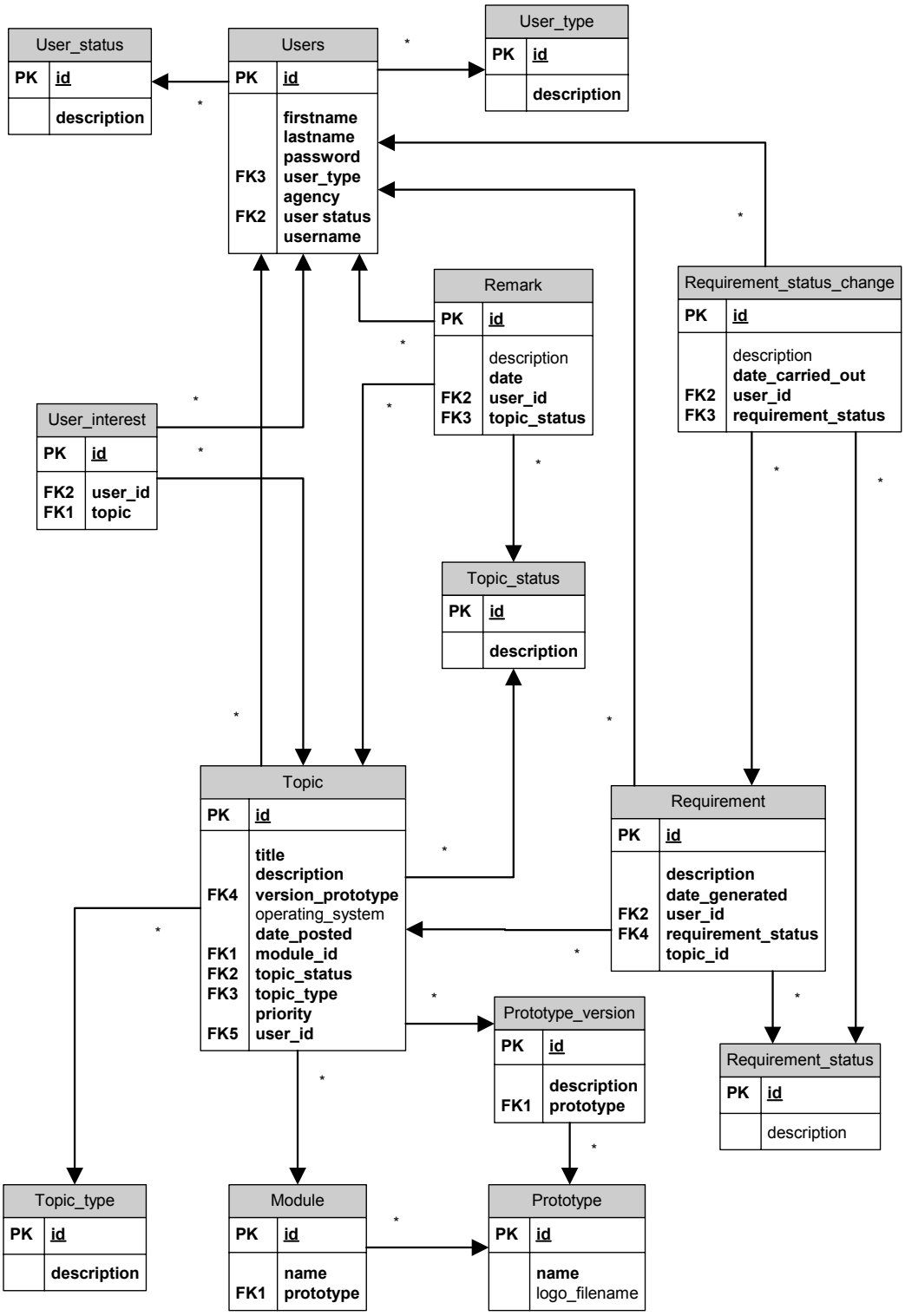


Figure 3.2-2 Database model diagram

3.3 Statechart diagrams

Certain classes from the class diagram described in section 3.2.1 can have multiple states. To get a clear vision of what states and what transitions those classes can have, statechart diagrams are used.

3.3.1 Class User

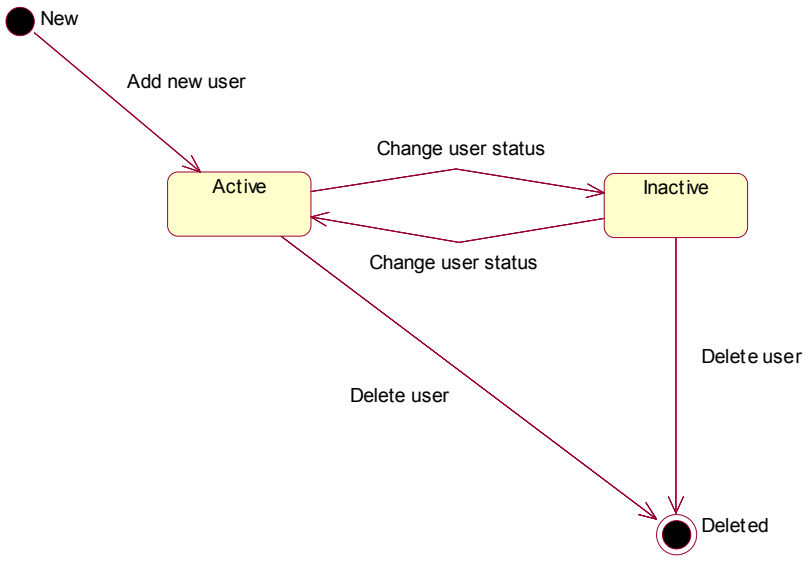


Figure 3.3-1 Statechart diagram Class User

3.3.2 Class Topic

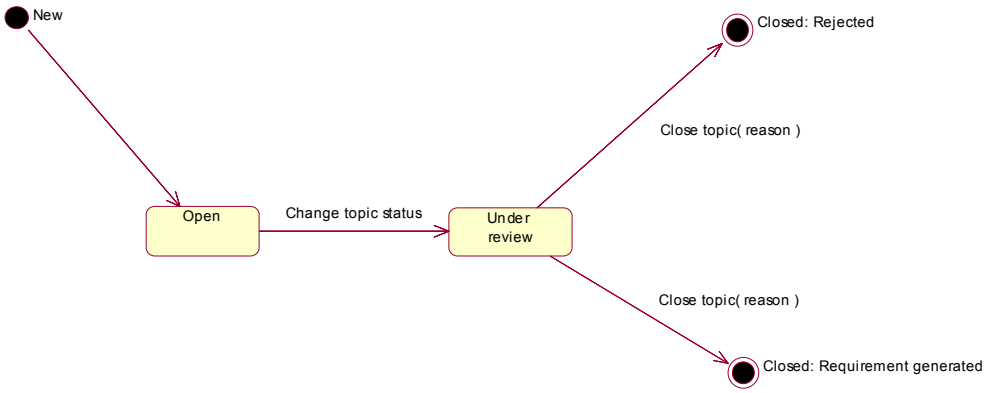


Figure 3.3-2 Statechart diagram Class Topic

3.3.3 Class Requirement

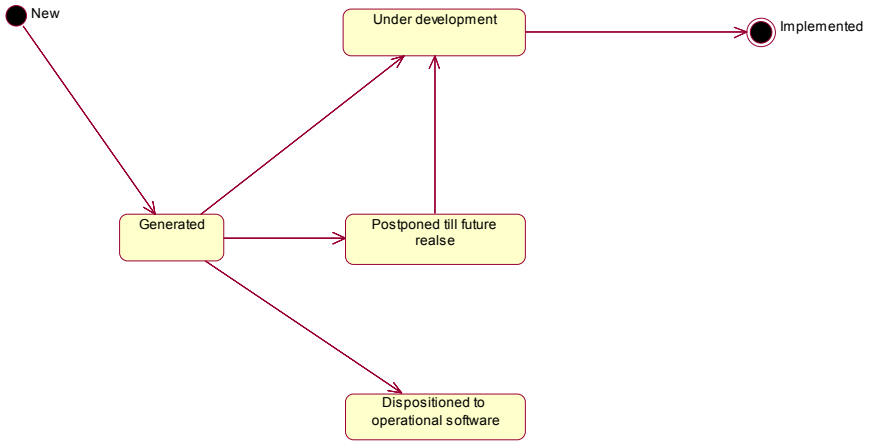


Figure 3.3-3 Statechart diagram Class Requirement

Chapter 4 - Beta Release

The Beta Release is built based on the Executable Prototype that is developed during the Elaboration phase. Most of the non-critical use cases are implemented in this Beta. This chapter will describe the Beta Release that has been built along with the evaluation results.

4.1 Critical use cases

This section will discuss the critical use cases that have been implemented in the Beta Release.

4.1.1 Sign in

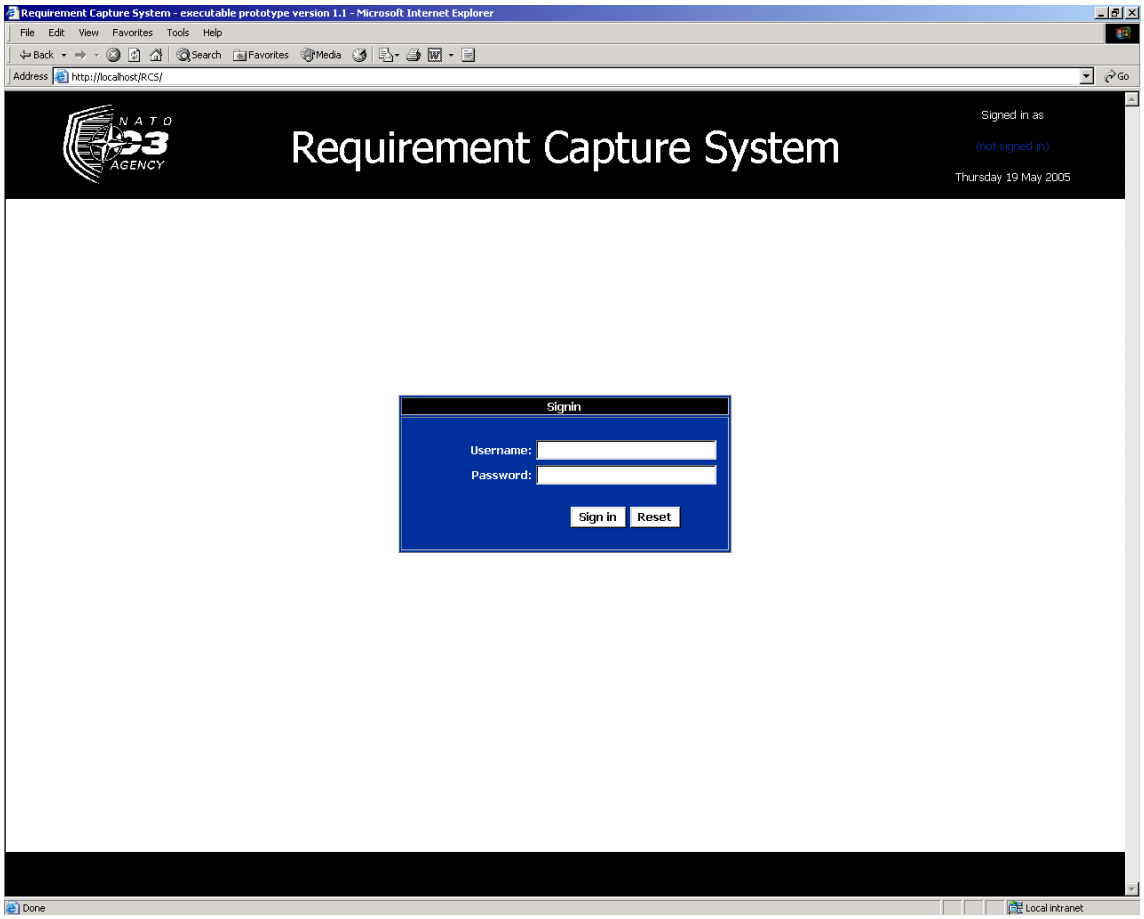


Figure 4.1-1 SignIn screen

In the ‘SignIn’ screen the use can enter his/her username and password. By clicking on the ‘Sign in’ button the use can submit username and password. When the use clicks on the button ‘Reset’ the text fields ‘Username’ and ‘Password’ will reset.

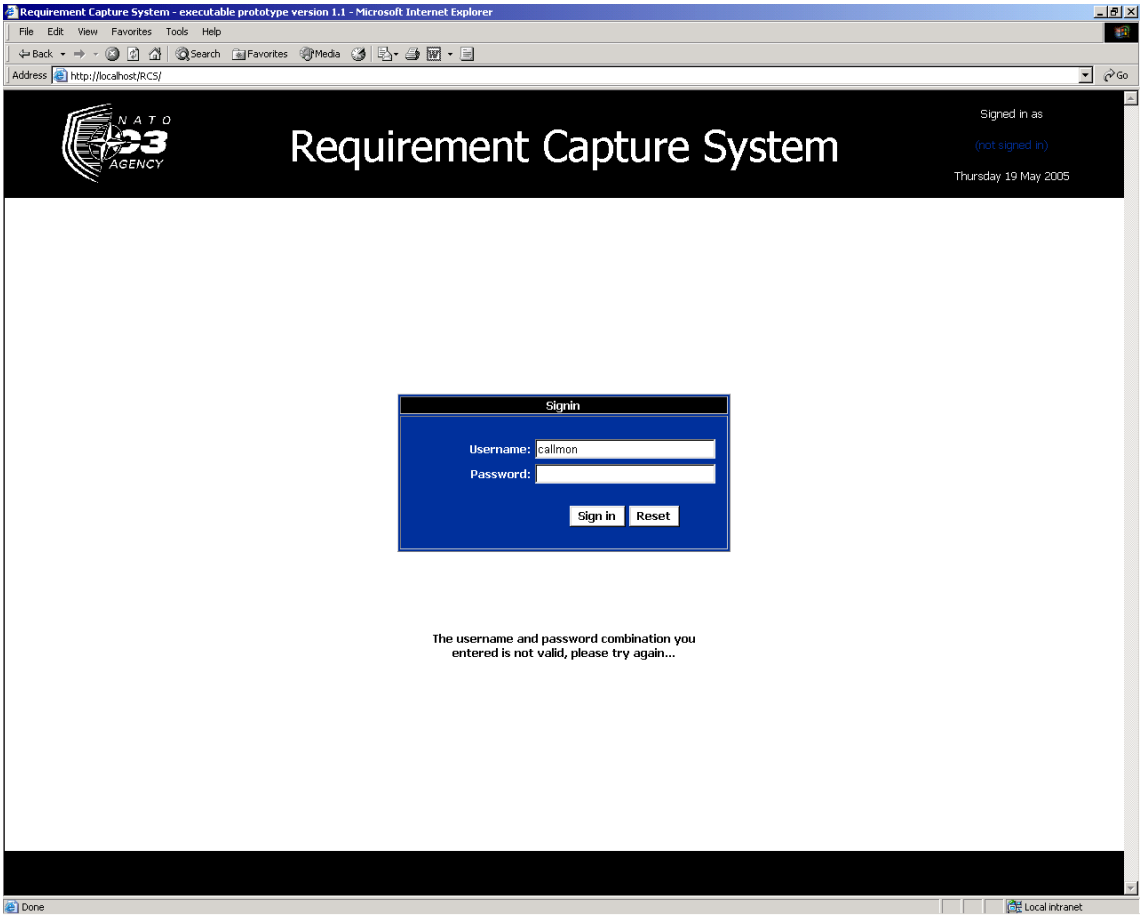


Figure 4.1-2 SignIn screen

If the user submits an unknown username, the user account is set to inactive or the username or password combination is not valid there will be an error message displaying ‘The username and password combination you entered is not valid, please try again’.

4.1.2 Start a topic

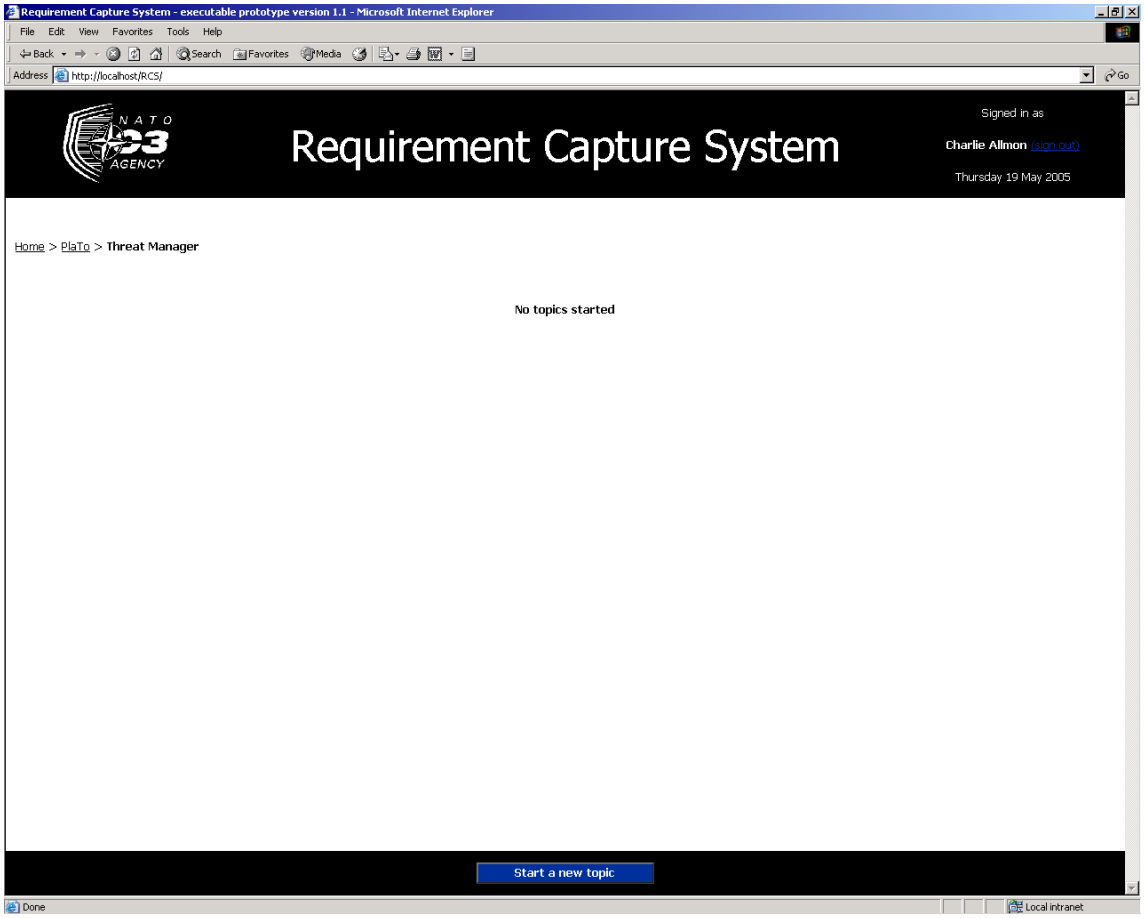


Figure 4.1-3 TopicList screen

After selecting a software prototype system and a module (non-critical use cases) the system will display the 'TopicList' screen. In this screen the user can start a new topic by clicking on the 'Start a new topic' button on the footer.

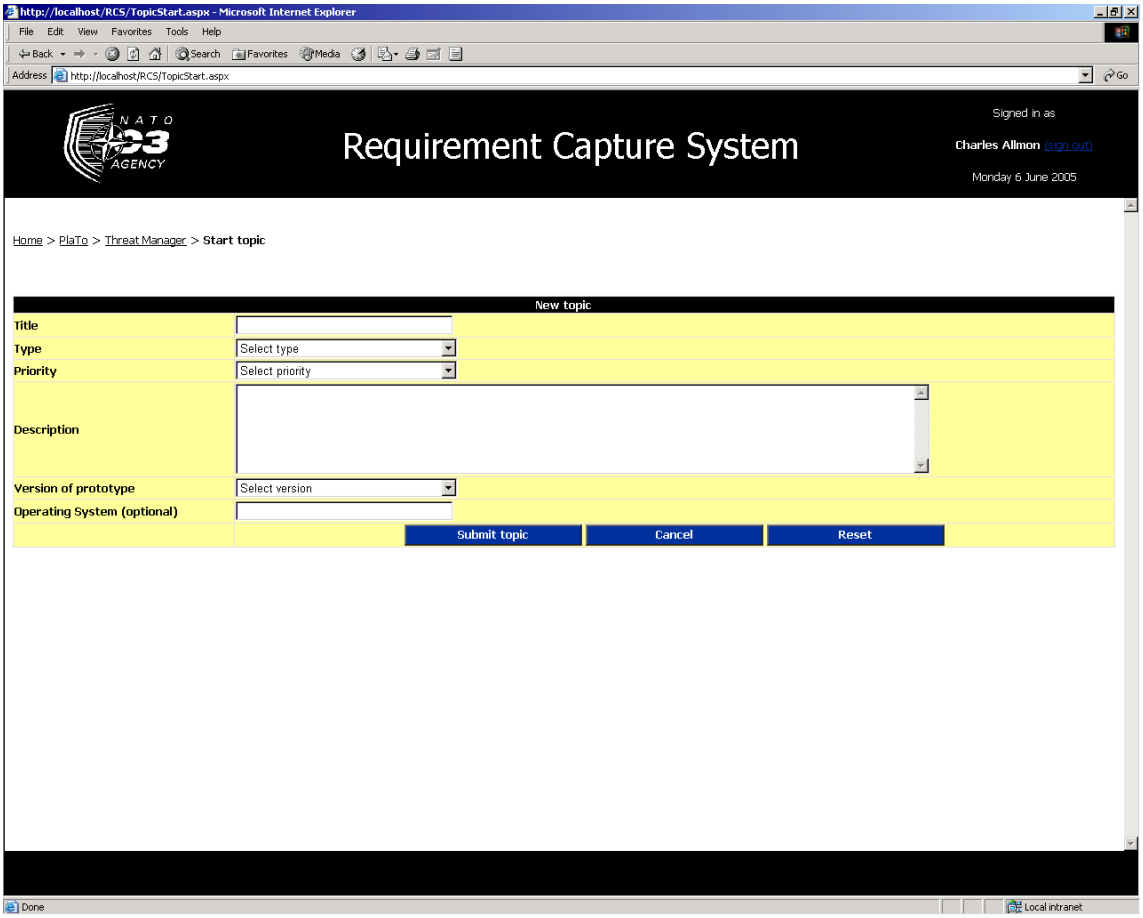


Figure 4.1-4 Start a topic screen

After clicking on the 'Start a new topic' button the system will display a new form where the user can fill in the topic. This is called the 'StartTopic' screen. The fields are:

- Topic title (text field)
- Topic type (dynamic dropdown list)
- Priority (dropdown list)
- Topic description (text field)
- Version of prototype (dynamic dropdown list)
- Operating System (text field)

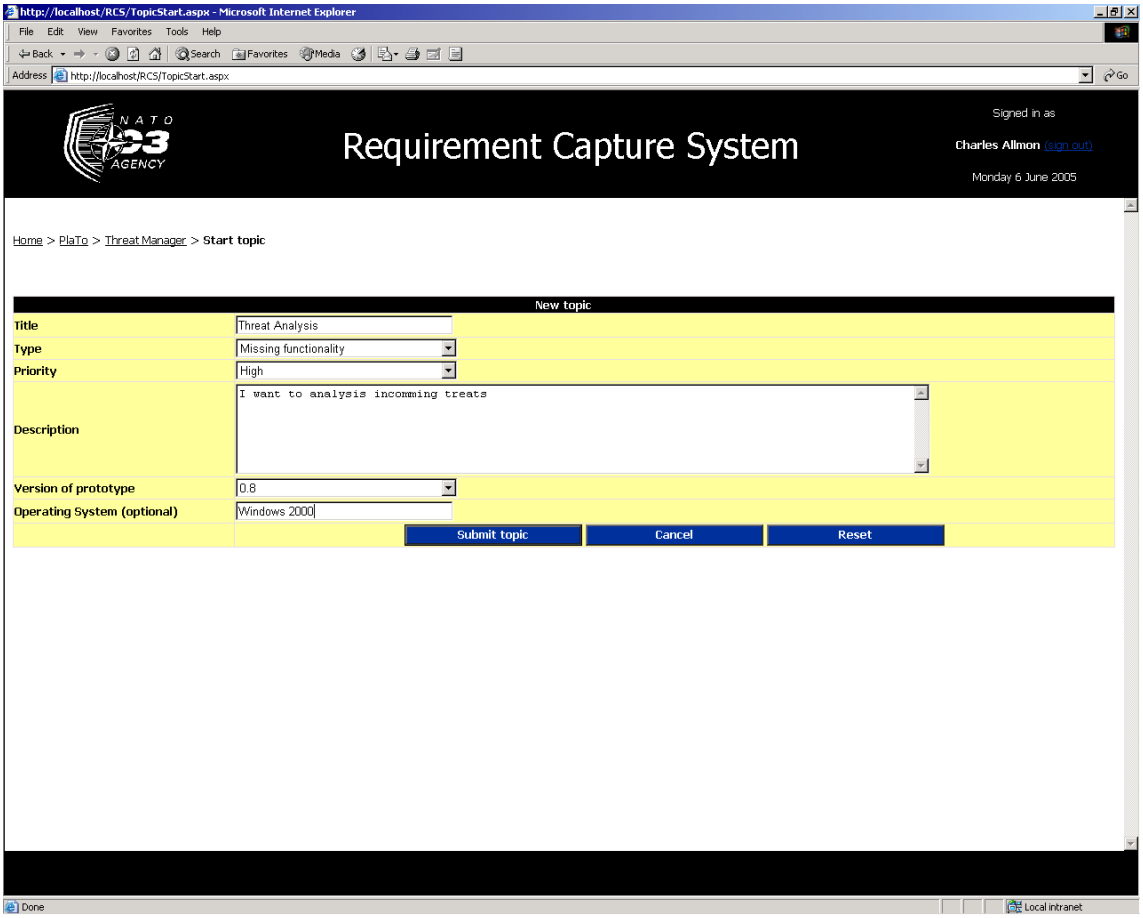


Figure 4.1-5 TopicStart screen

By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the topic by clicking on the ‘Submit topic’ button.

4.1.3 View topic

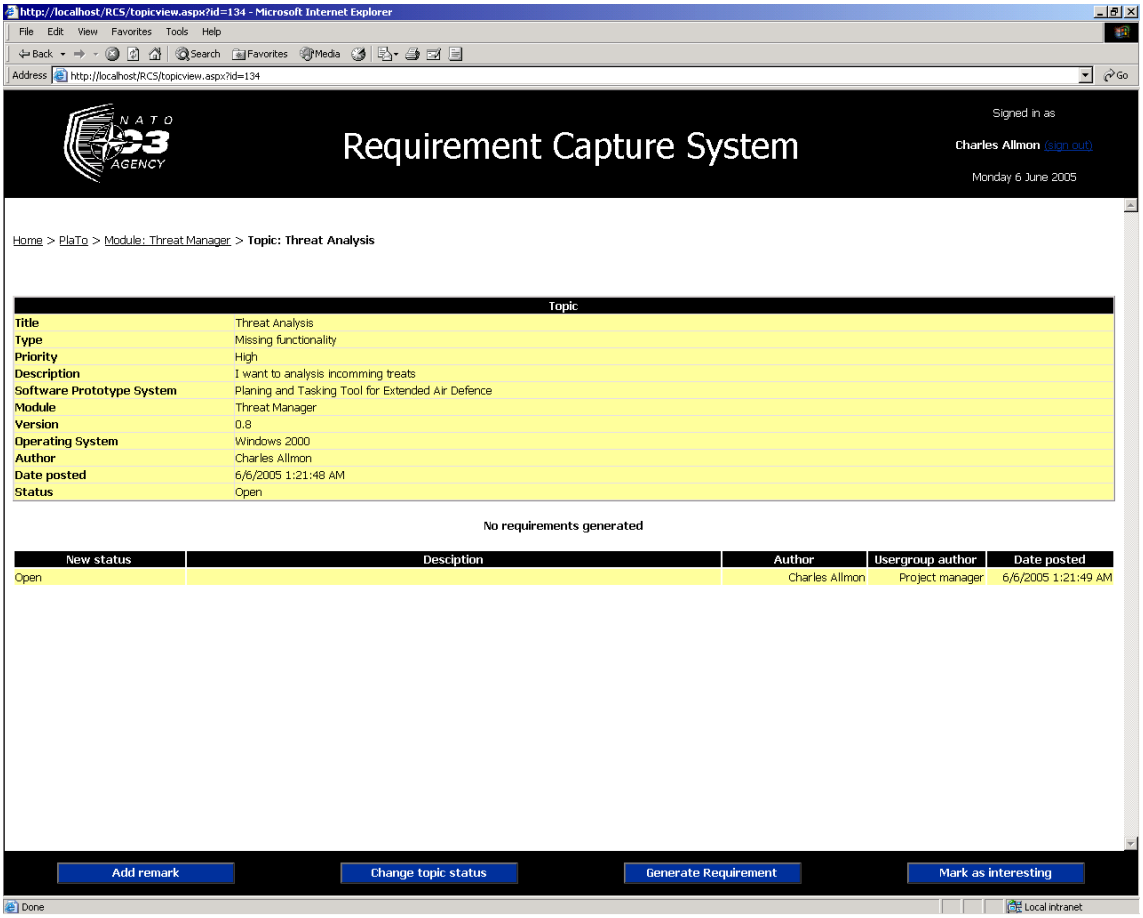


Figure 4.1-6 TopicView screen

When a topic is submitted the system will display 'TopicView' screen that displays all information about topic. This also includes the requirements that have been generated from this topic and remarks and status changes on the topic. This information displayed on the topic:

- Title
- Type
- Priority
- Description
- Software Prototype System
- Module
- Version
- Operating System
- Author
- Date posted
- Status

In the footer are now four new buttons:

- Add remark
- Change topic status
- Generate Requirement
- Mark as interesting

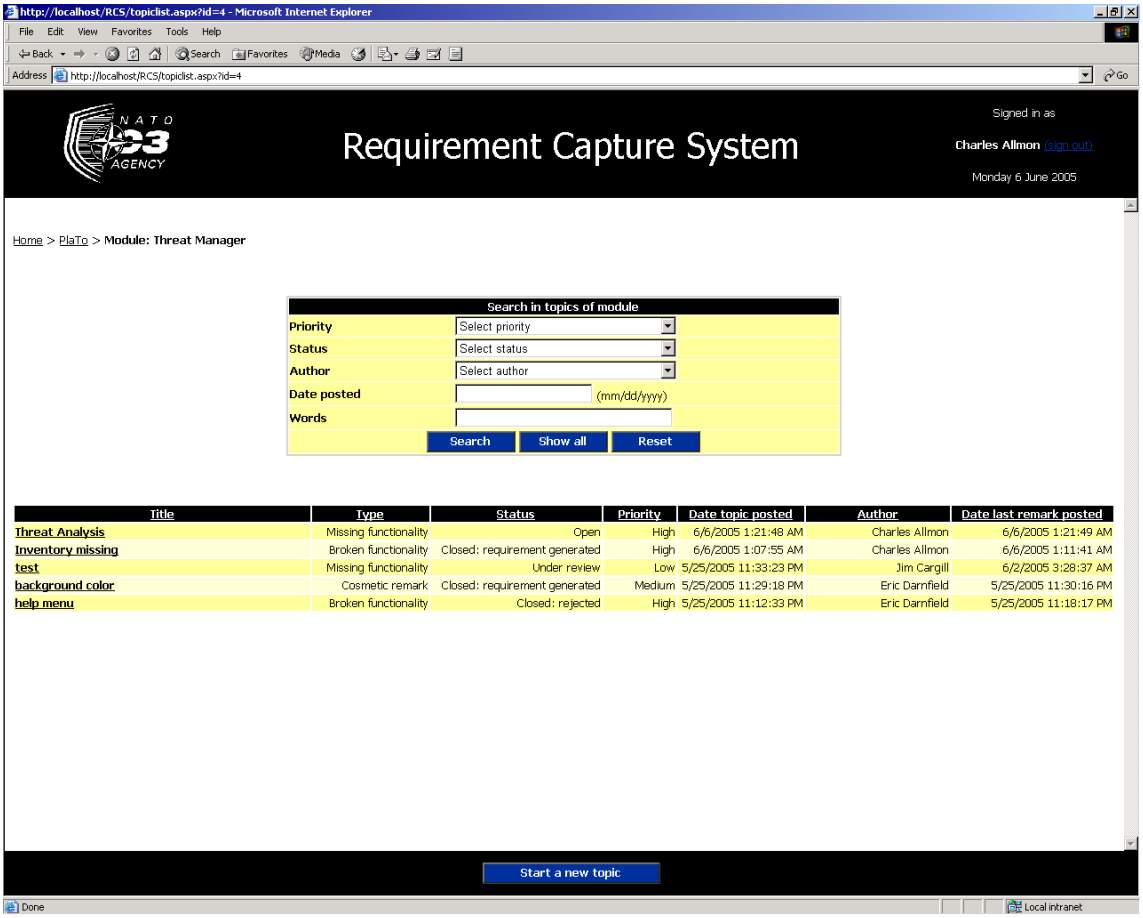


Figure 4.1-7 TopicList screen

When the user goes back to the 'TopicList' screen the topic he/she just added is also briefly displayed. The information displayed on a topic in this screen is:

- Title
- Type
- Status
- Priority
- Date topic posted
- Author
- Date last remark posted

By clicking on the header the user can sort the topics.

4.1.4 Change topic status

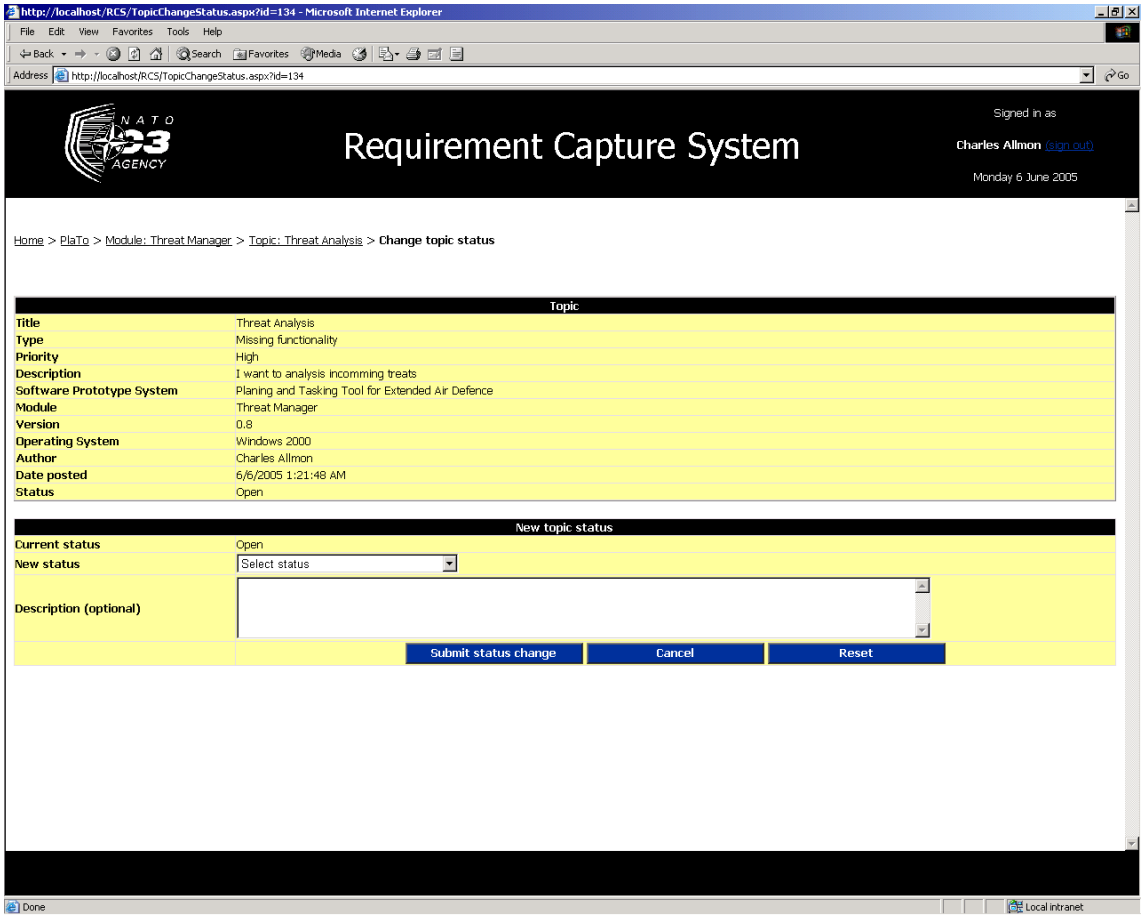


Figure 4.1-8 TopicChangeStatus screen

When the user clicks on the ‘Change topic status’ button in the ‘TopicView’ screen, an additional form is displayed. In this form the user can submit the new status and provide an additional description. The fields in this form are:

- New status (dynamic dropdown list)
- Description (text field)

If the status of a topic is ‘Closed: rejected’ or ‘Closed: requirement generated’ it should not be possible to change the status of that topic anymore.

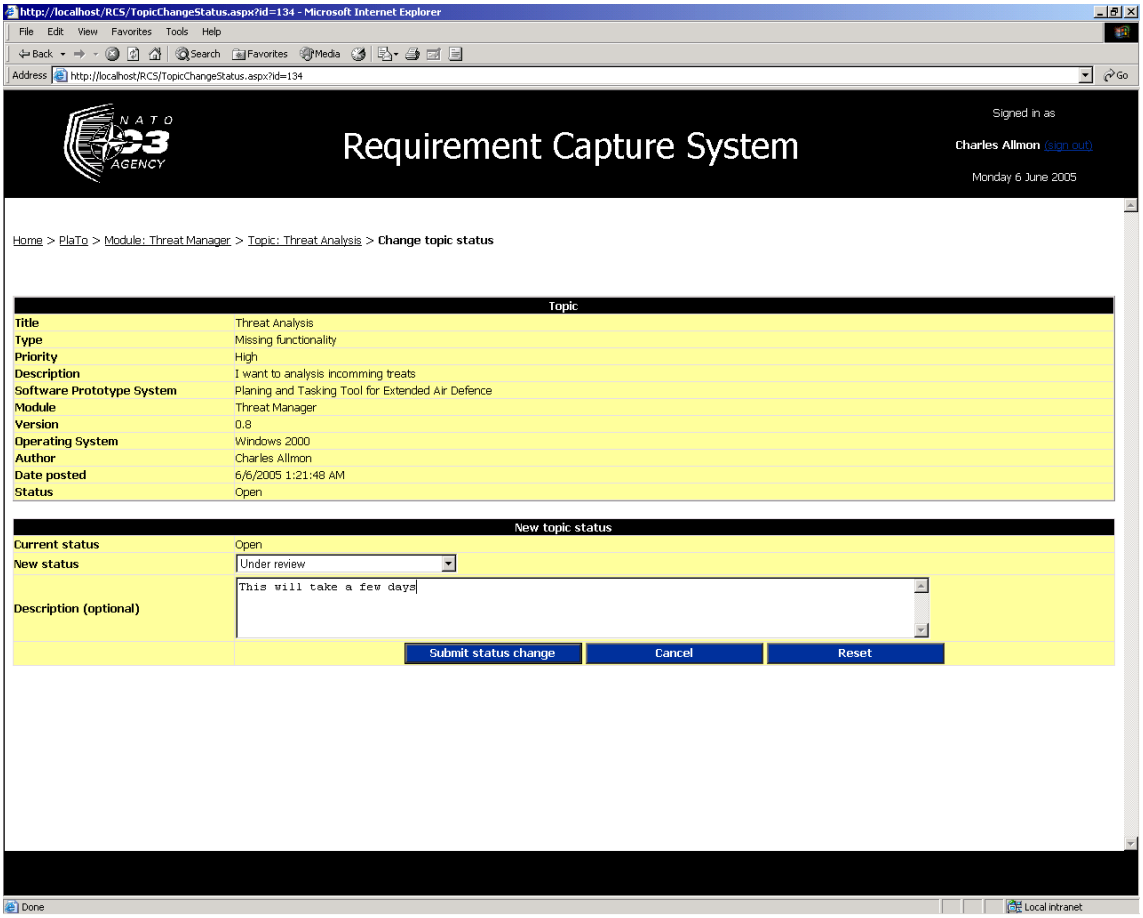


Figure 4.1-9 TopicChangeStatus screen

By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the status change by clicking on the ‘Submit status change’ button.

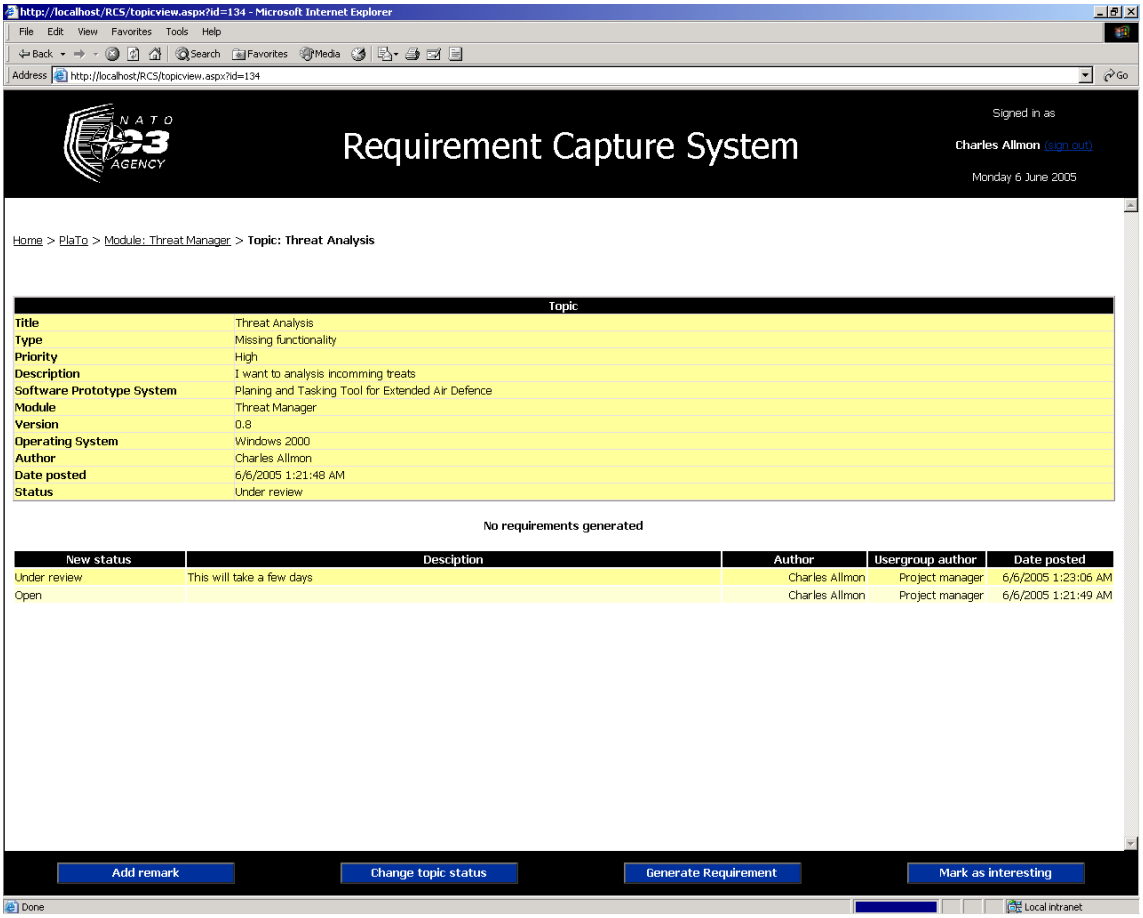


Figure 4.1-10 TopicView screen

When the user submits a topic status change the system will display the ‘TopicView’ screen. The status change submitted is also displayed in this screen.

4.1.5 Add remark

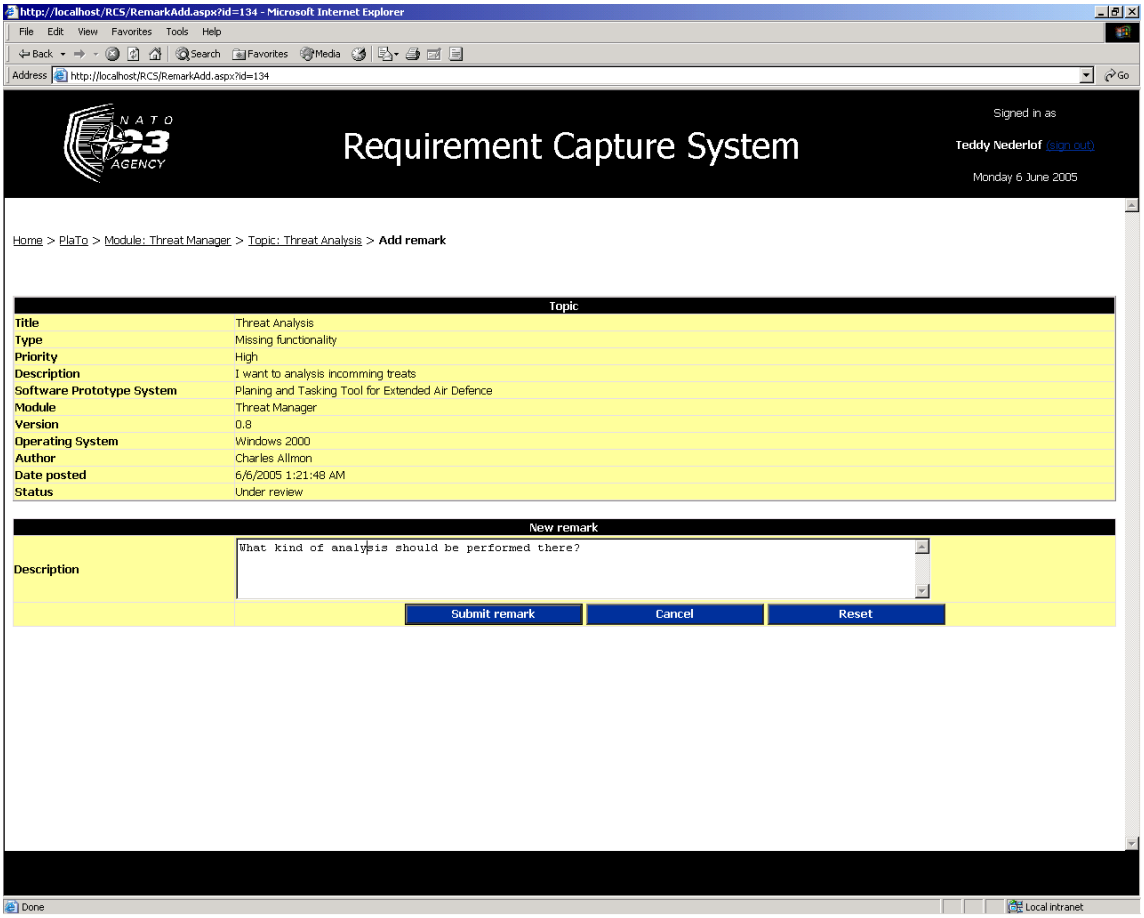


Figure 4.1-11 RemarkAdd screen

When the user clicks on the ‘Add remark’ button in the ‘TopicView’ screen, an additional form is displayed. In this form the user can submit a remark. The only field in this form is ‘Description’ which is a text field. By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the remark by clicking on the ‘Submit remark’ button.

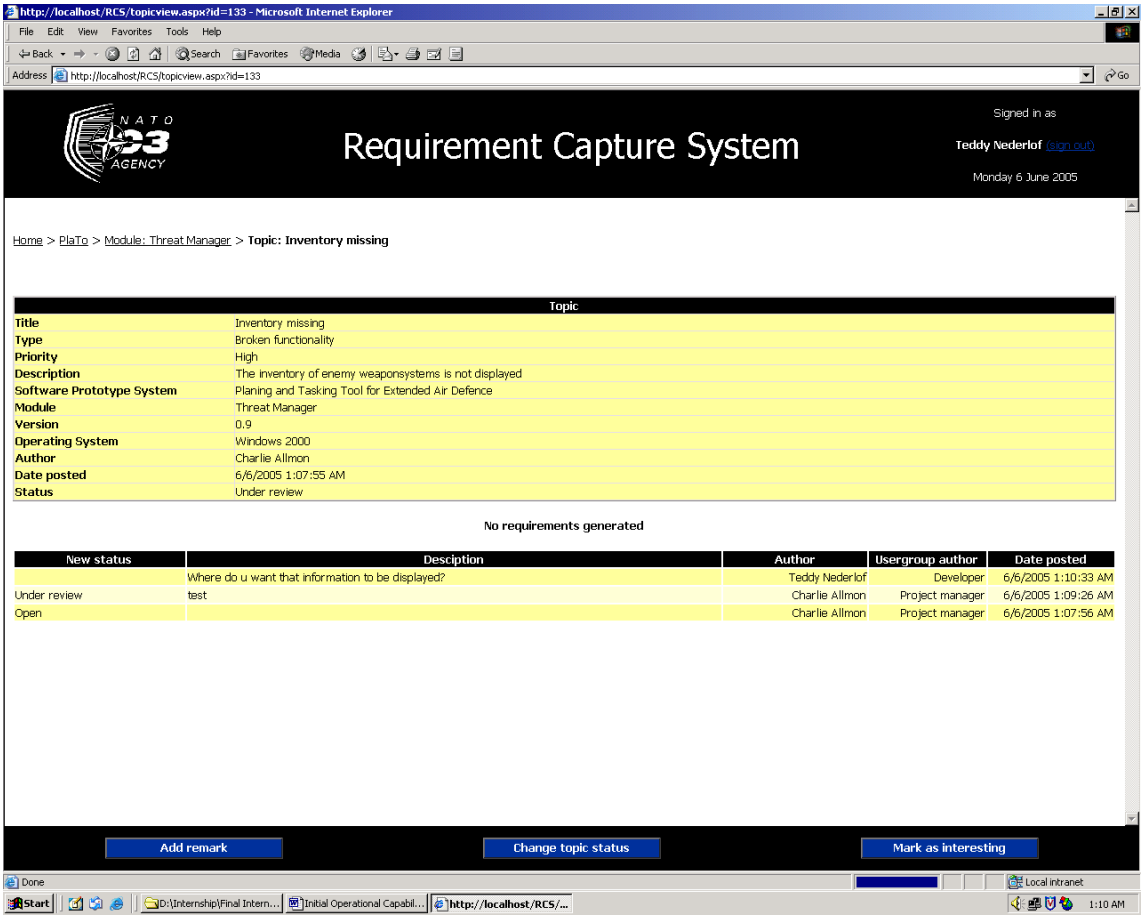


Figure 4.1-12 TopicView screen

When the user submits a remark the system will display the ‘TopicView’ screen. The remark submitted is also displayed in this screen. There is also pager that allows the user to page true the status changes and remarks.

4.1.6 Generate requirement

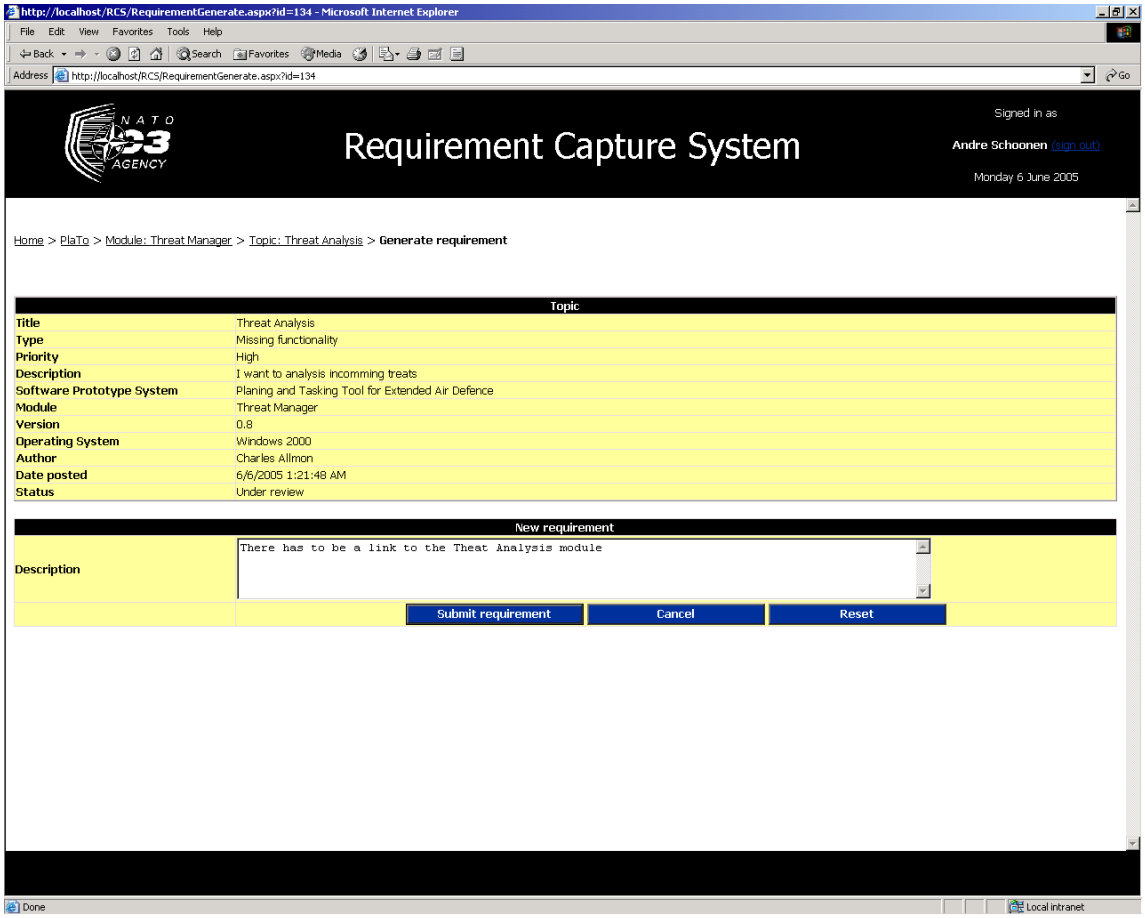


Figure 4.1-13 RequirementGenerate screen

When the user clicks on the ‘Generate Requirement’ button in the ‘TopicView’ screen, an additional form is displayed. In this form the user can submit a requirement. The only field in this form is ‘Description’ which is a text field. By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the requirement by clicking on the ‘Generate Requirement’ button.

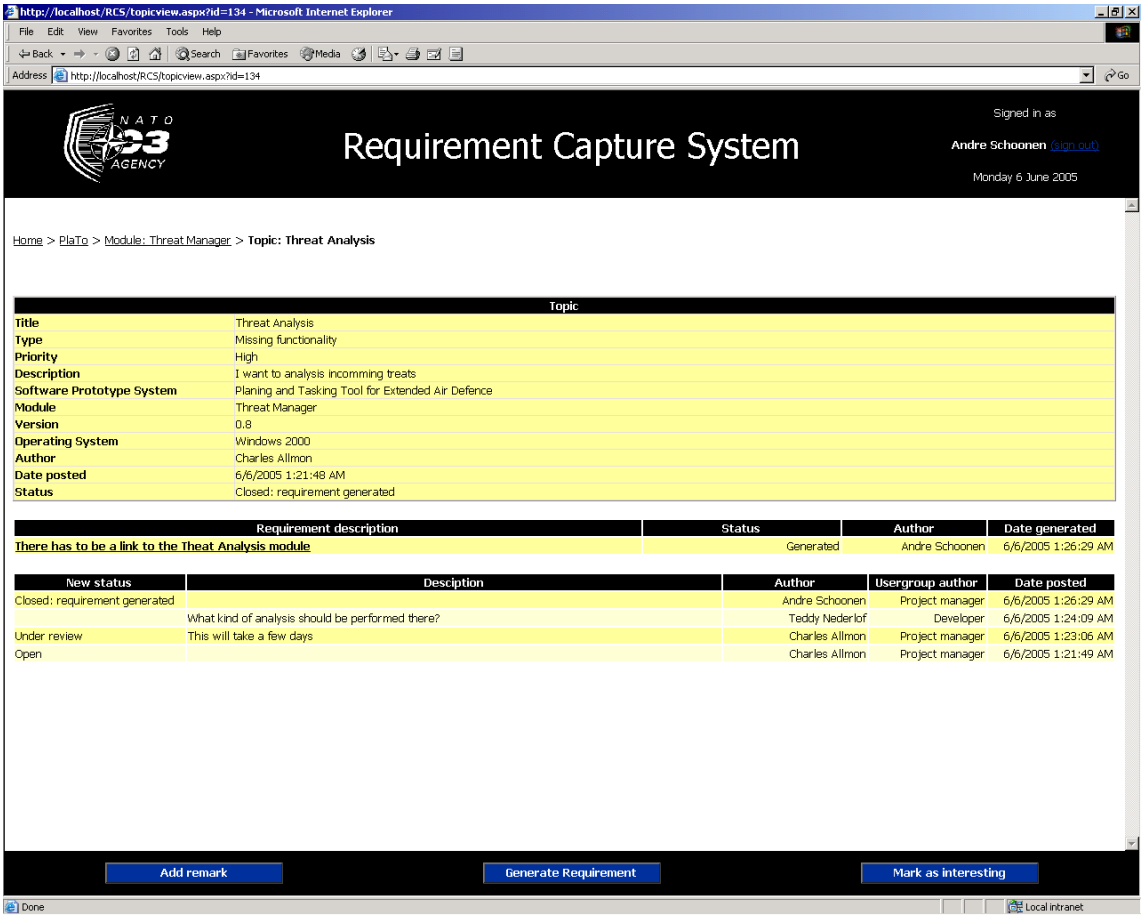


Figure 4.1-14 TopicView screen

When the user submits a requirement the system will display the ‘TopicView’ screen.The requirement generated is also displayed in this screen. To view all information about the requirement the user can click on the Requirement description.

4.1.7 Change requirement status

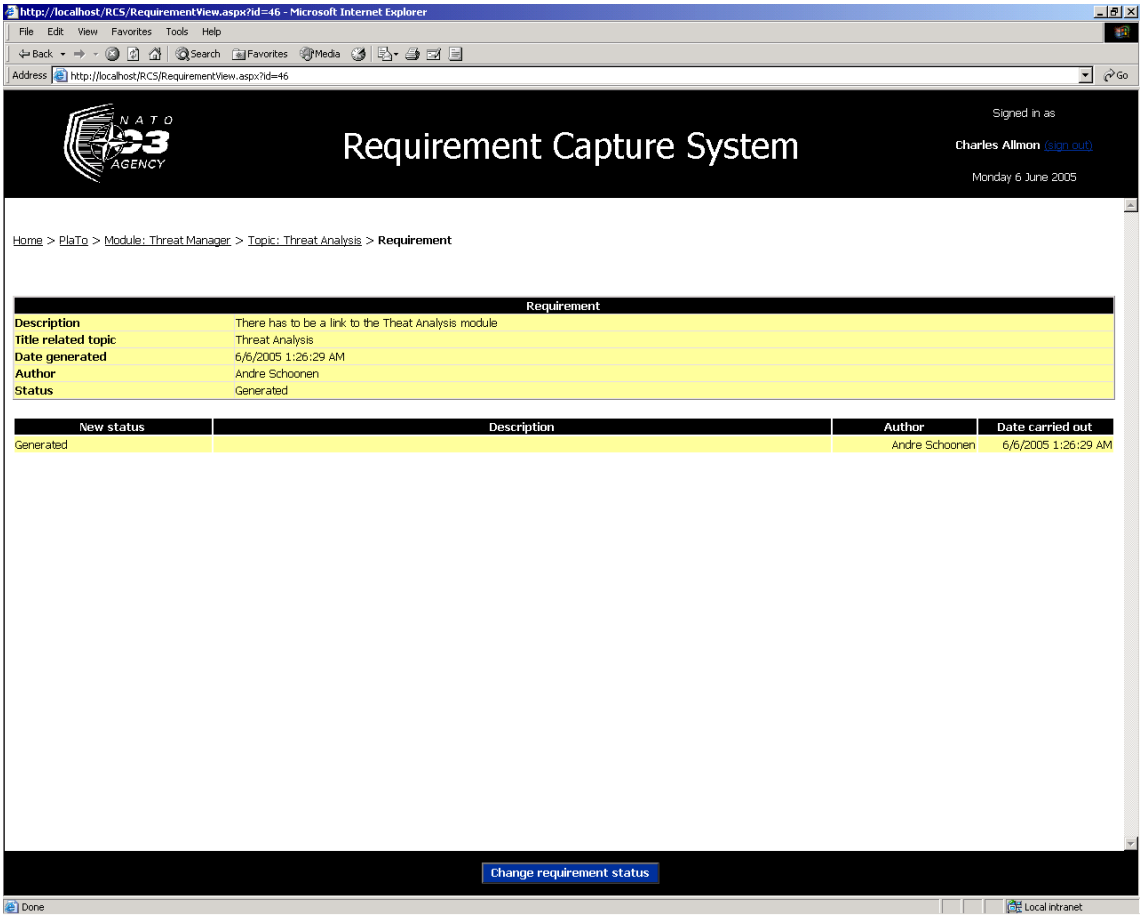


Figure 4.1-15 RequirementView screen

When the user clicks on the Requirement description the ‘RequirementView’ screen will be displayed. The information displayed on a requirement in this screen is:

- Description
- Title related topic
- Date generated
- Author
- Status

By clicking on the ‘Change requirement status’ button the user can change the status of the requirement. If the status of a requirement is ‘Implemented’ or ‘Dispositioned to operational software’ it should not be able to change the status of the requirement anymore.

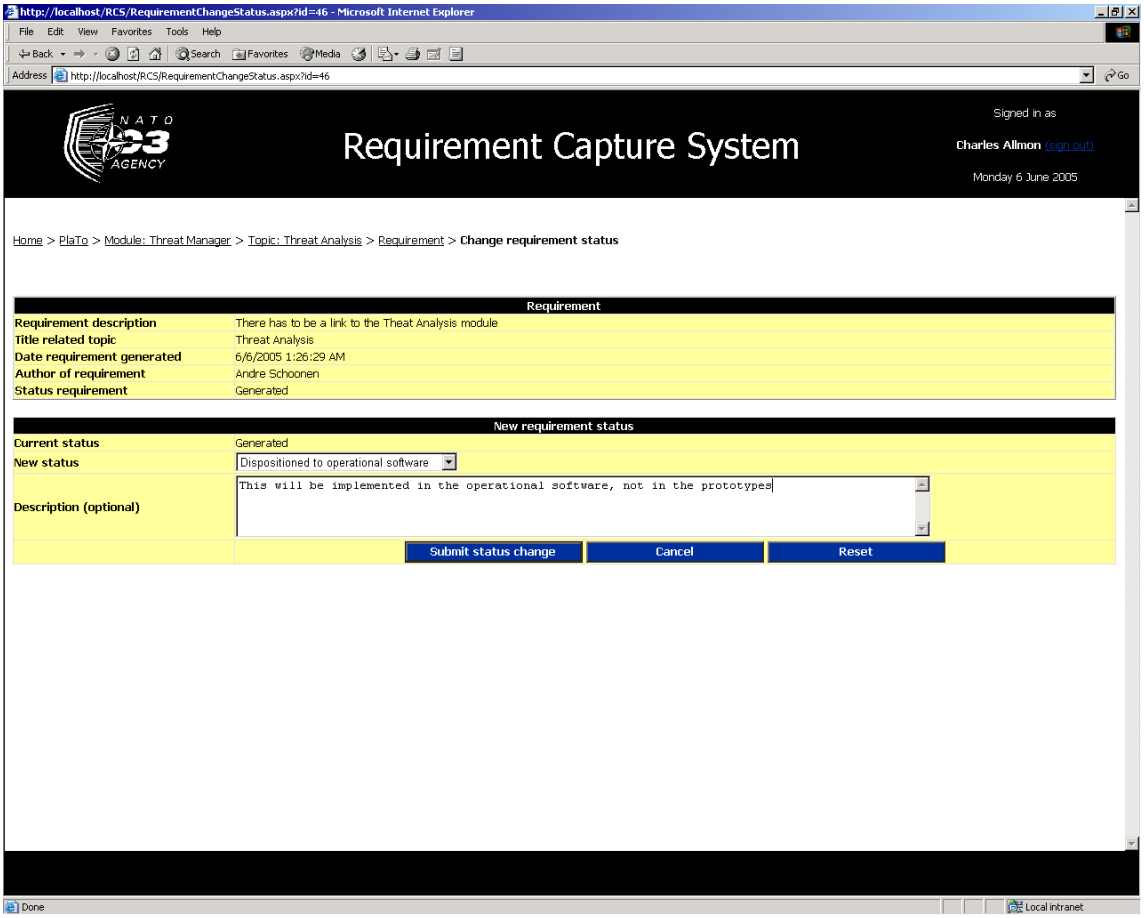


Figure 4.1-16 RequirementChangeStatus screen

When the user clicks on the ‘Change requirement status’ button in the ‘RequirementView’ screen, an additional form is displayed. In this form the user can submit the new status and provide an additional description. The fields in this form are:

- New status (dynamic dropdown list)
- Description (text field)

By clicking on the ‘Reset’ button all the fields will be reset and by clicking on the ‘Cancel’ button the system will display the previous screen. The user can submit the status change by clicking on the ‘Submit status change’ button.

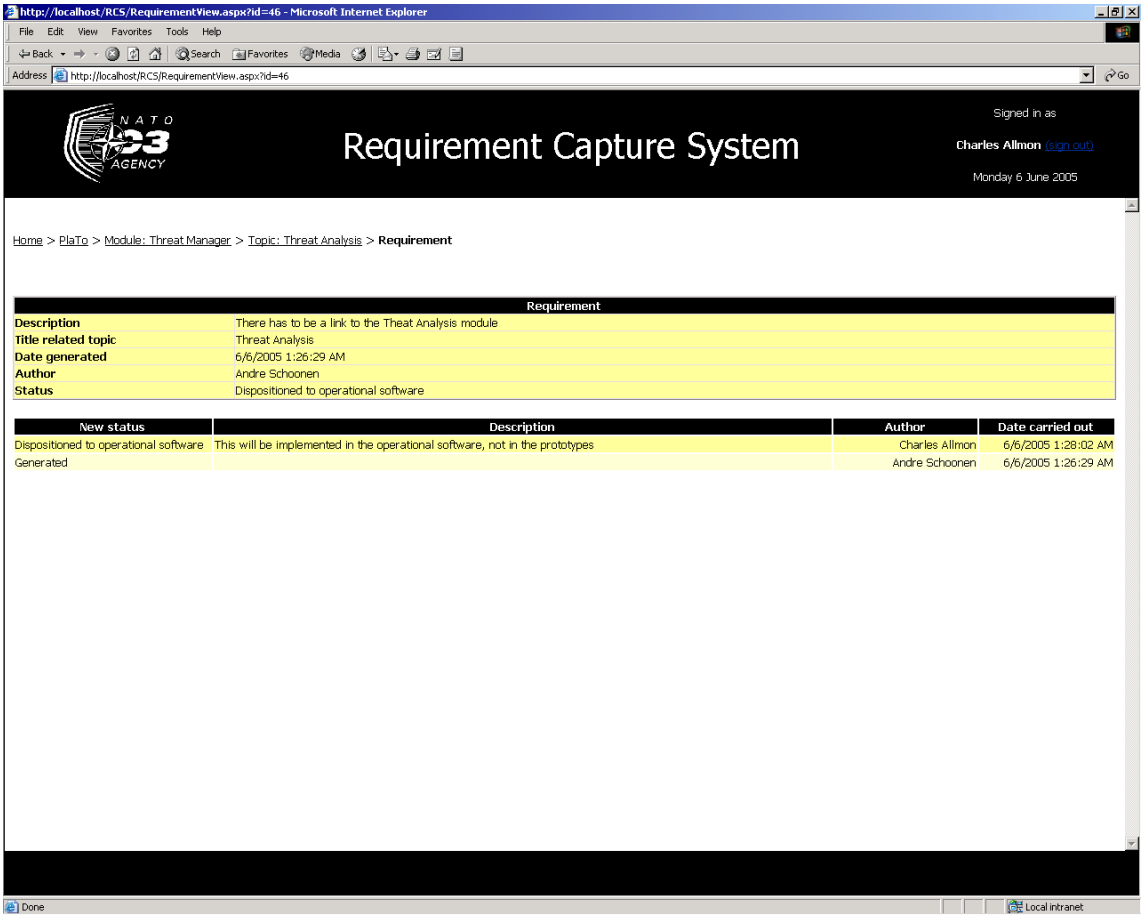


Figure 4.1-17 RequirementView screen

When the user submits a requirement status change the system will display the ‘RequirementView’ screen. The status change submitted is also displayed in this screen. There is also pager that allows the user to page true the status changes.

4.2 Non-critical use cases

This section will discuss the non-critical use cases that have been implemented in the Beta Release.

4.2.1 Sign out

If the user is signed in there will be a link named ‘sign out’ in the header in the right section. If the user clicks on this link, the system will sign out the user and display the ‘SignIn’ screen.

4.2.2 View name user signed in

If the user is signed in there his/her name will be displayed in the header in the right section.

4.2.3 Select software prototype system

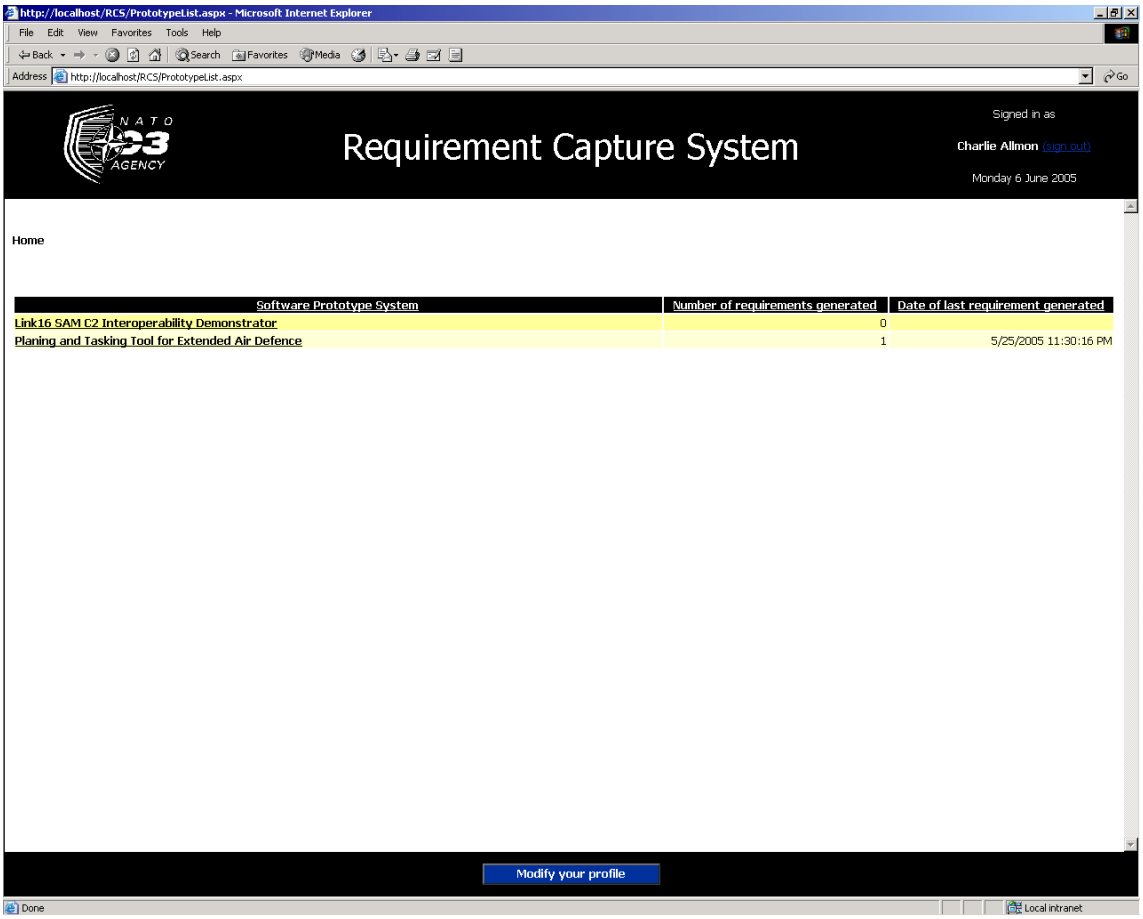


Figure 4.2-1 PrototypeList screen

When the use signs in the ‘PrototypeList’ screen is displayed.

The information displayed on a software prototype system in this screen is:

- Software Prototype System (full name)
- Number of requirements generated
- Date last requirement generated

In this screen the user can select a software prototype system by clicking on the name of it.

4.2.4 Select a module

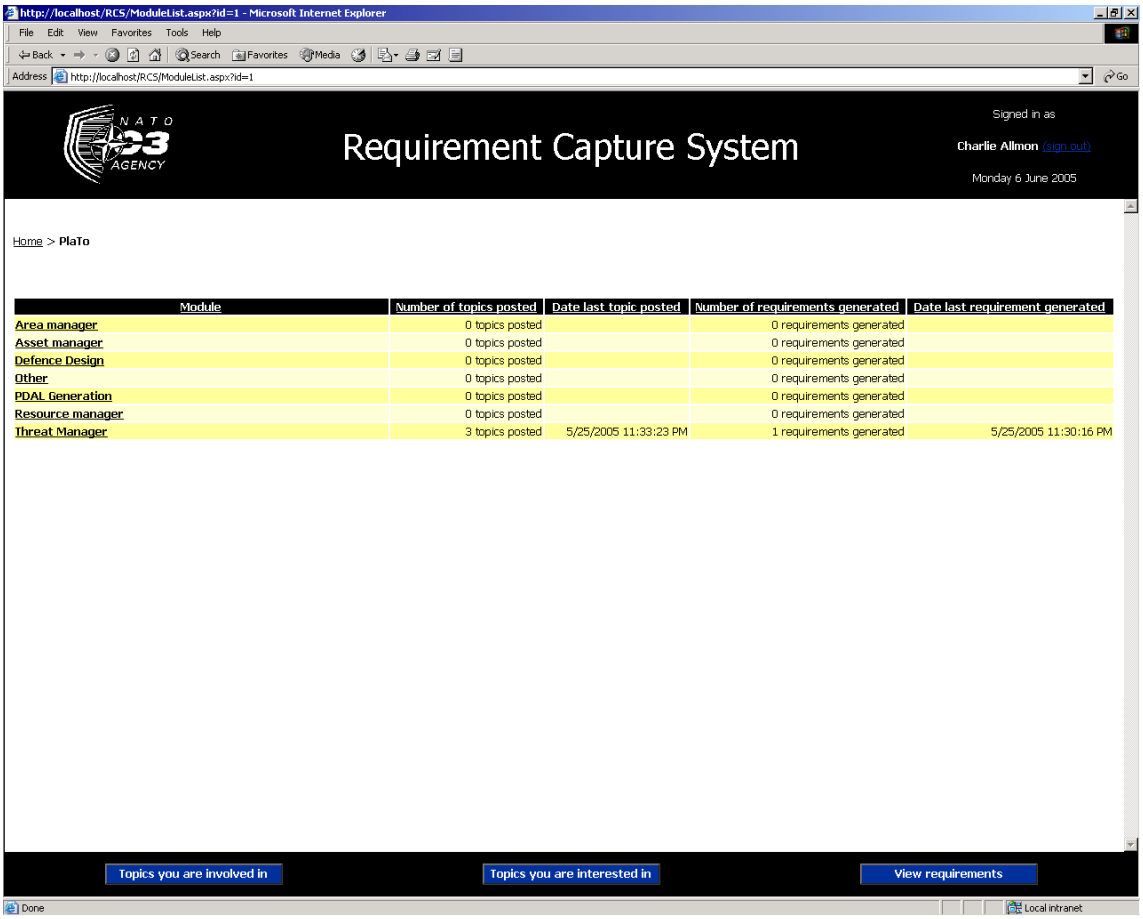


Figure 4.2-2 ModuleList screen

When the user selects a software prototype system the 'ModuleList' screen is displayed.
The information displayed on a software prototype system in this screen is:

- Module (name)
- Number of topics posted
- Date last topic posted
- Number of requirements generated
- Date last requirement generated

In this screen the user can select a software module by clicking on the name of it.

Evaluation results

- In this screen there must also be a possibility to search for topics in the selected software prototype system.

4.2.5 Search for topics

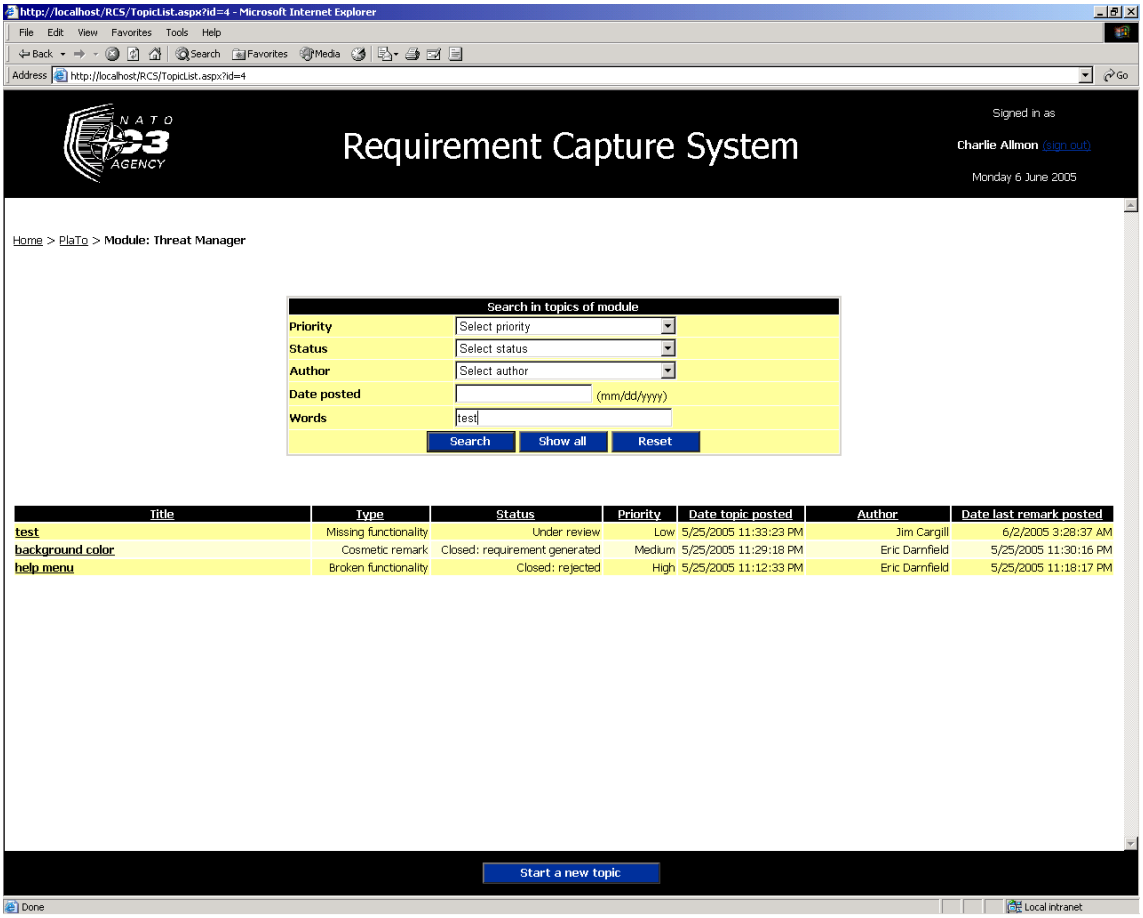


Figure 4.2-3 TopicList screen

When he user has selected a module the system will display ‘TopicView’ screen that displays all information about topic. In this screen it is possible to search for a topic. There can be searched on:

- Priority
- Status
- Author
- Date posted
- Words

-

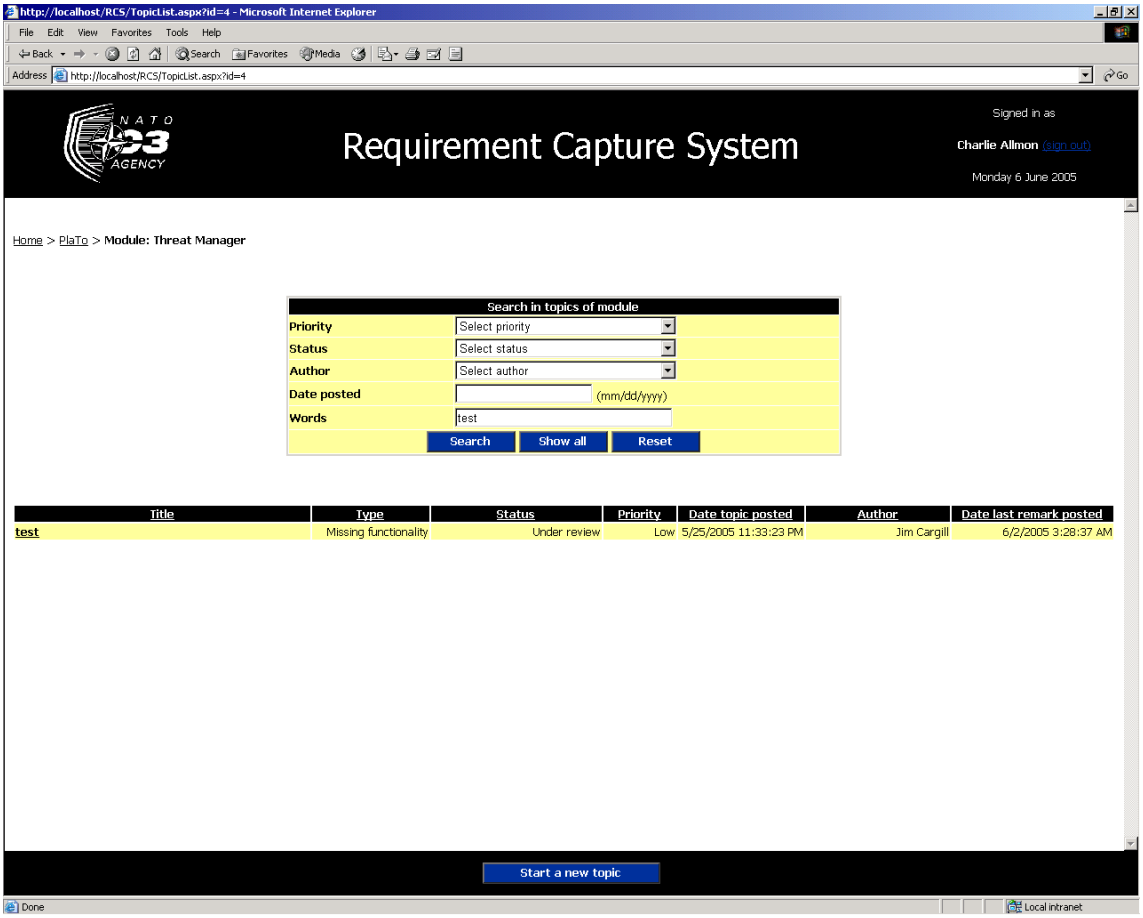


Figure 4.2-4 TopicList screen

By clicking on the ‘ Show all’ button, all topics in the selected module will be displayed. The search fields can be cleared by clicking on the ‘Reset’ button. To submit the search query the user can click on the ‘Search’ button.

Evaluation results

- It must be possible to search on topic type.
- When searching on ‘Status’ there must also be a possibility to search for al topics that do not have the status ‘Closed: Requirement generated’ and ‘Closed: Rejected’.
- When searching on ‘Date posted’ it must be possible to search on a certain range between two different dates.

4.2.6 View topics user is interested in

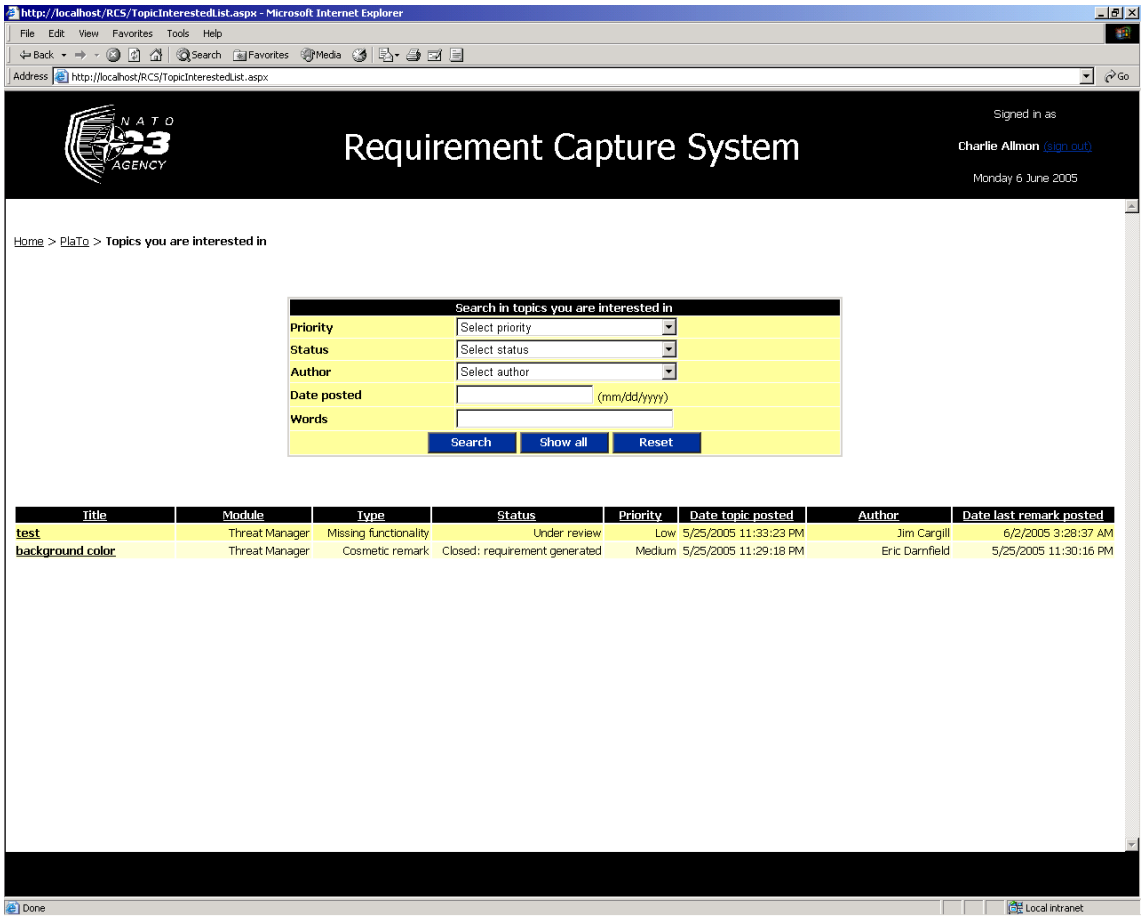


Figure 4.2-5 TopicInterestedList screen

In the ‘ModuleList’ screen the user can click on the button ‘View topics you are interested in’. If the user clicks that button, a list of all the topics he/she is interested in will be displayed. This screen is almost similar to the ‘TopicList’ screen.

Evaluation results

- When viewing the normal ‘TopicList’ screen it must be possible to whether the user marked a topic as interesting or not. This could be done by displaying an icon at with that topic.

4.2.7 View topics user is involved in

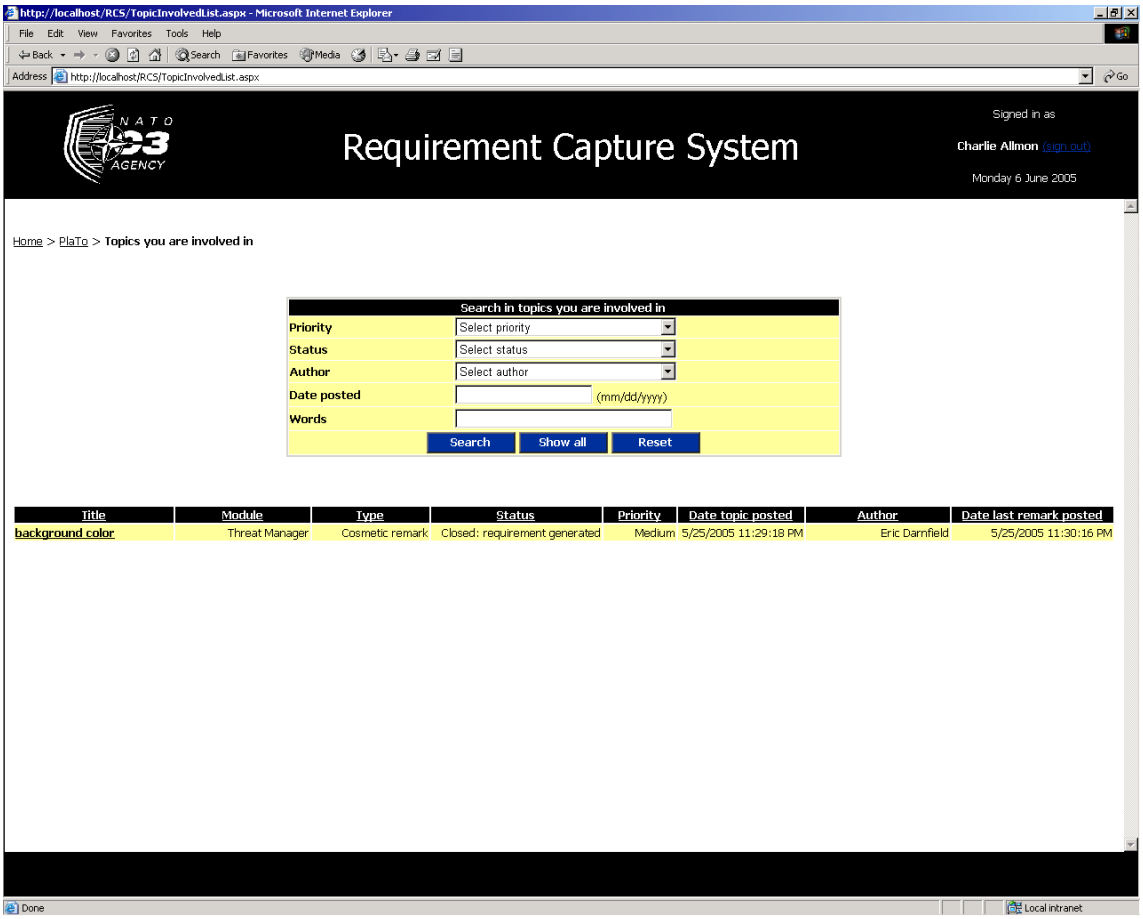


Figure 4.2-6 TopicInvolvedList screen

In the ‘ModuleList’ screen the user can click on the button ‘View topics you are involved in’. If the user clicks that button, a list of all the topics he/she is involved in will be displayed. This screen is almost similar to the ‘TopicList’ screen.

Evaluation results

When viewing the normal ‘TopicList’ screen it must be possible to whether the user is involved in a topic or not. This could be done by displaying an icon at with that topic.

4.2.8 Mark topic as interesting

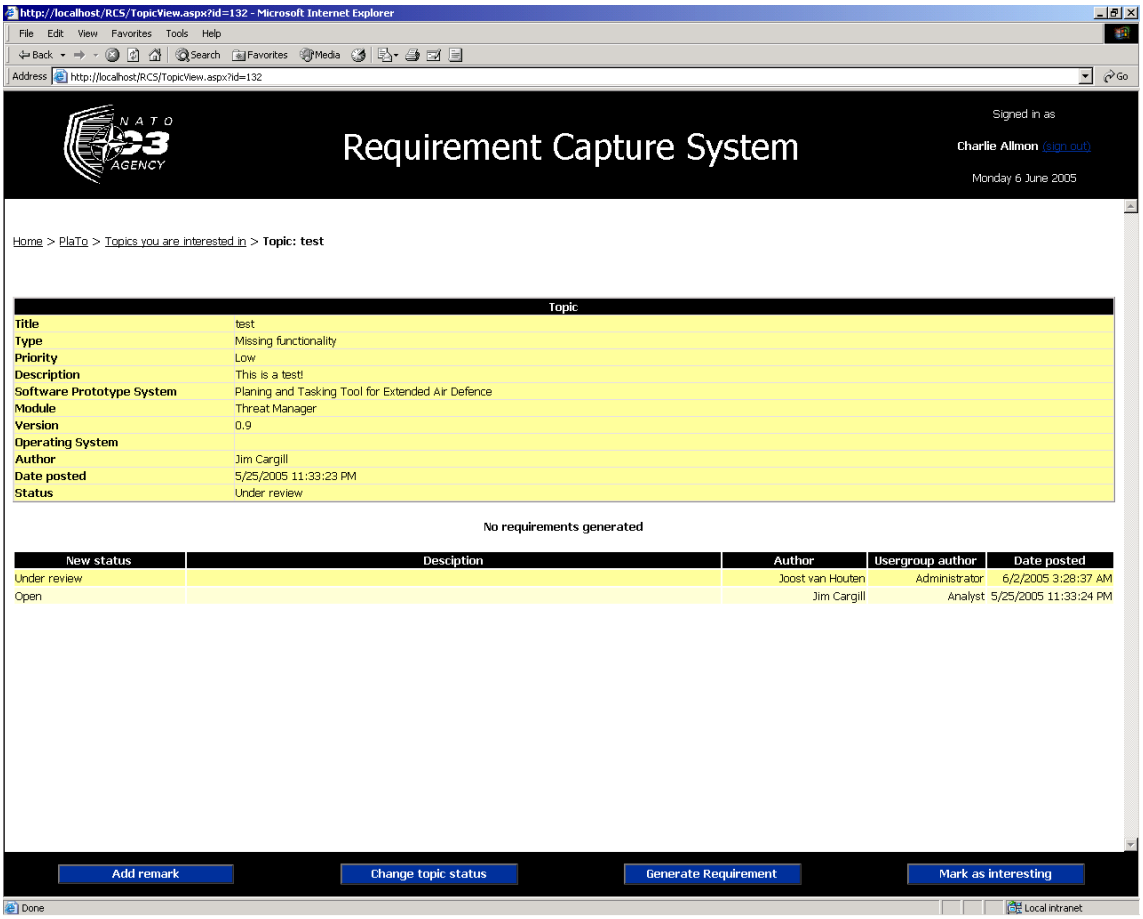


Figure 4.2-7 TopicView screen

While viewing a topic in the 'TopicView' Screen, the user can mark that topic as interesting by clicking on the 'Mark as interesting' button. When the user clicks this button the current topic will be marked as interesting and the text of the button will change to 'Unmark as interesting'.

4.2.9 Unmark a topic as interesting

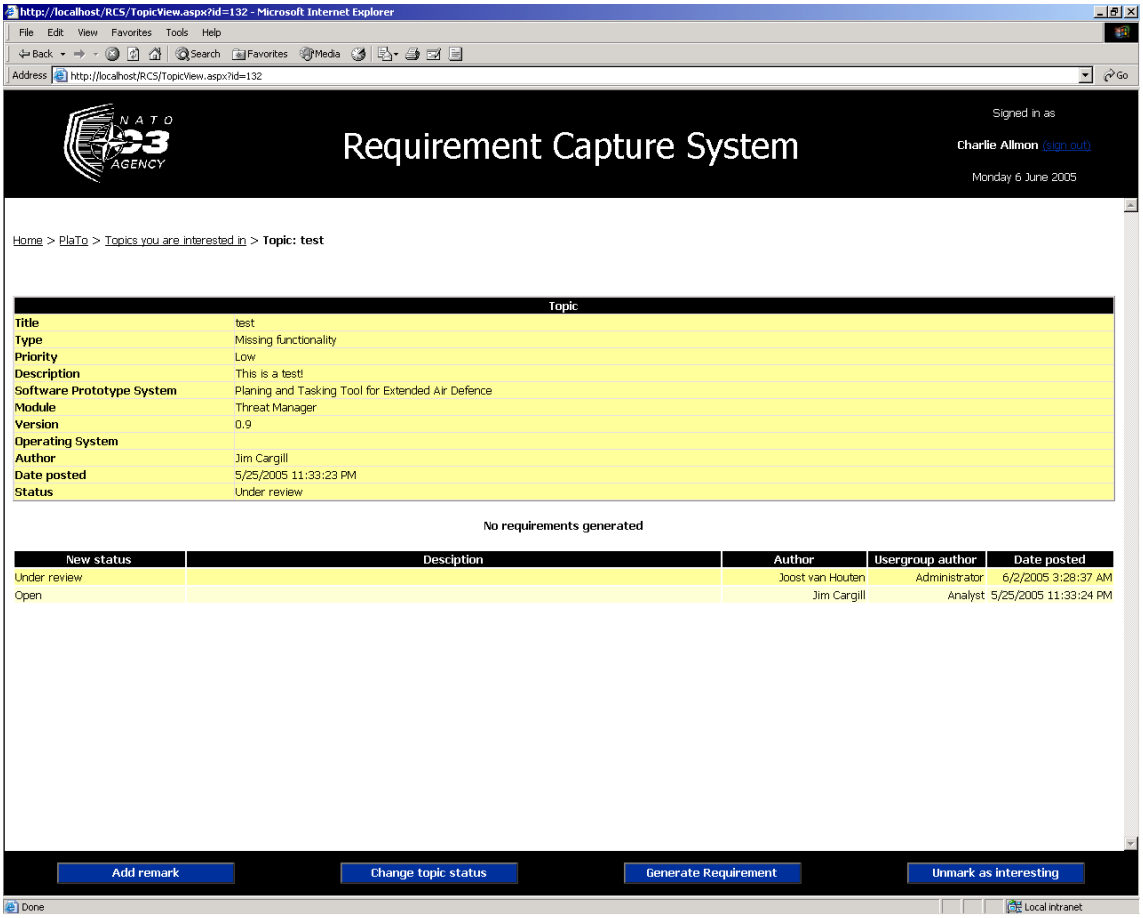


Figure 4.2-8 TopicView screen

By clicking on the ‘Unmark as interesting’ button in the ‘TopicView’ screen the current topic will no longer be mark as interesting and the text of the button will change to ‘Mark as interesting’.

4.2.10 View requirements generated

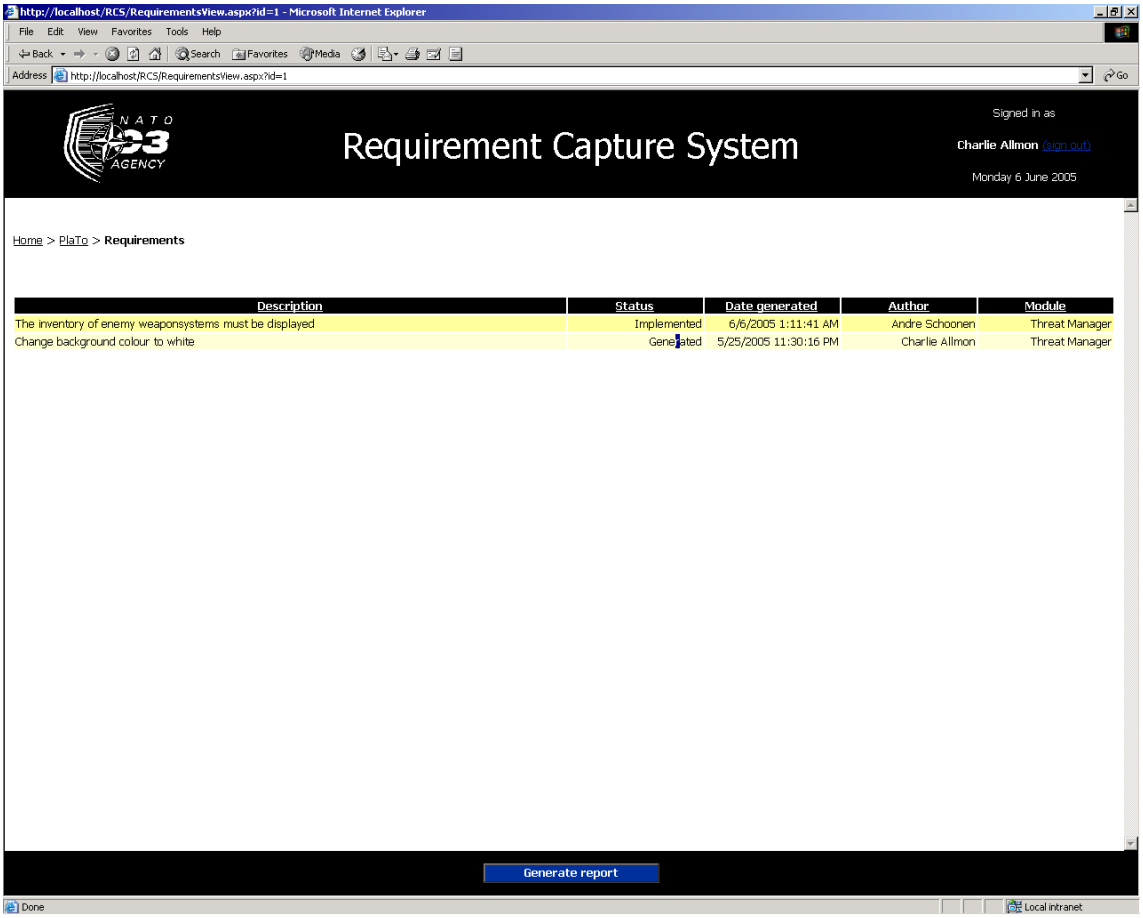


Figure 4.2-9 RequirementsView screen

If the user clicks on the ‘View requirements’ button in the ‘ModuleList screen’, the system will display the ‘RequirementsView’ screen. This screen displays a list of all requirements that are generated for the selected software prototype system. The information display on a requirement is:

- Description
- Status
- Date generated
- Author
- Module

In the footer a button with the name ‘Generate report’ is displayed.

Evaluation results

In this screen it must be possible to click on a requirement and then go to the ‘RequirementView’ screen to view that requirement.

4.2.11 Generate requirement report

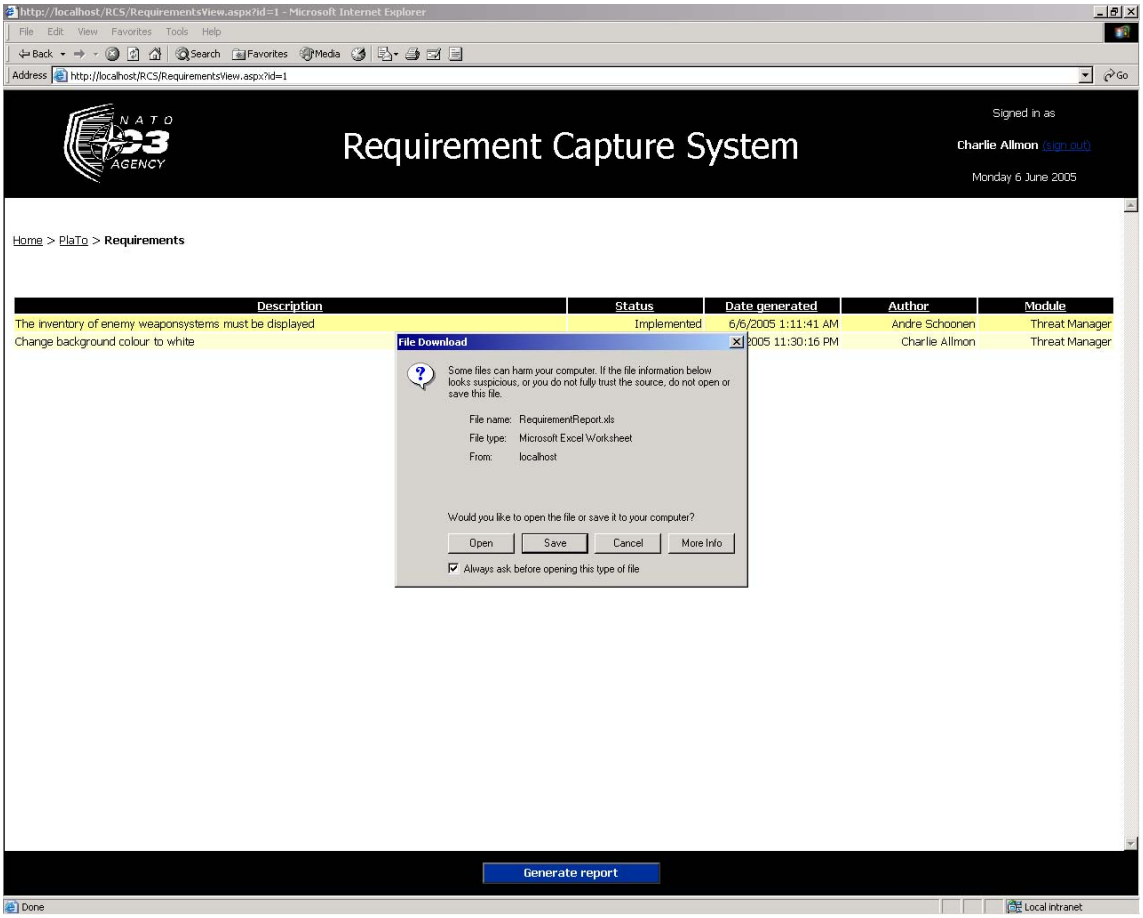
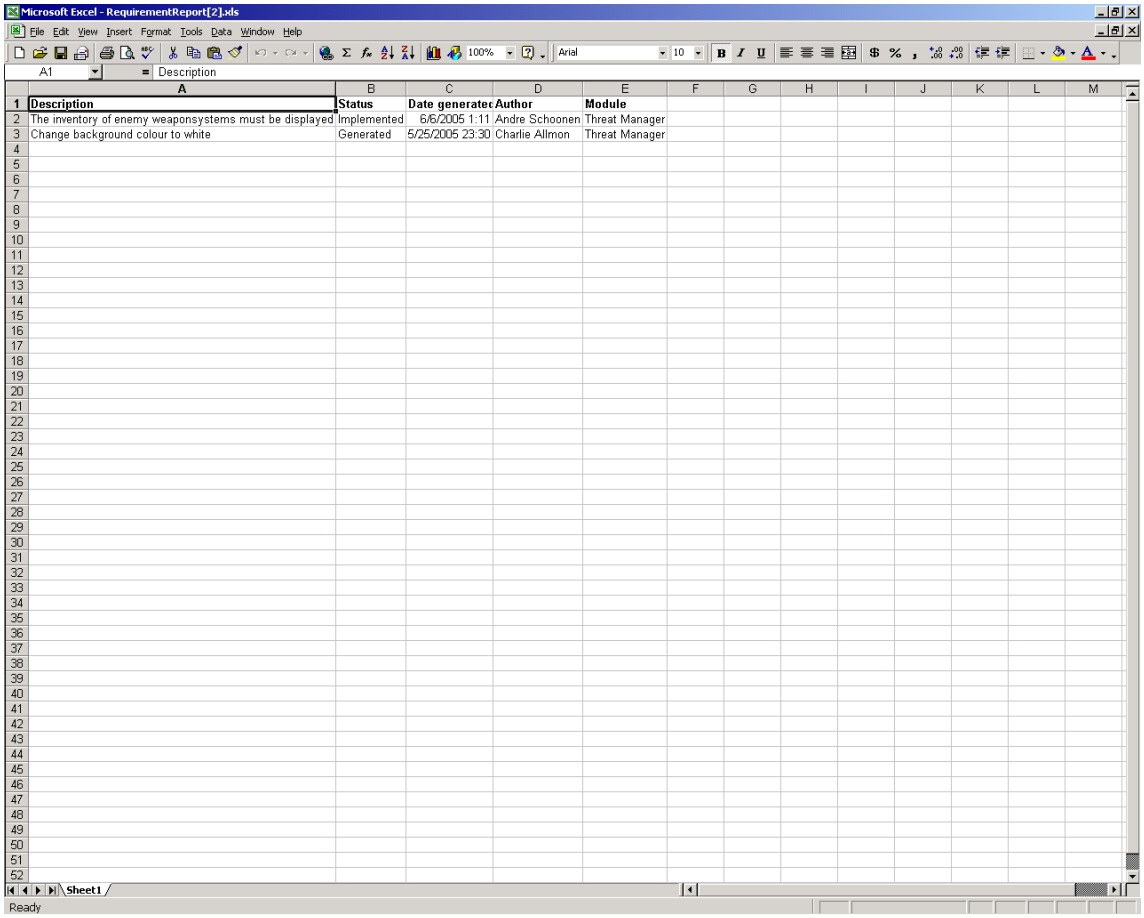


Figure 4.2-10 Pop-up window from the RequirementsView screen

If the user clicks on the ‘Generate report’ button, a pop-up window will appear that allows the user to export the list of requirement to an Excel document.



Description	Status	Date generatec	Author	Module
The inventory of enemy weaponsystems must be displayed	Implemented	6/6/2005 1:11	Andre Schoonen	Threat Manager
Change background colour to white	Generated	5/25/2005 23:30	Charlie Allmon	Threat Manager

Figure 4.2-11 Generated Excel document

The user can save and open the Excel document. The information that is displayed on a requirement in the Excel document is:

- Description
- Status
- Date generated
- Author
- Module

Evaluation results

In the Excel document it must be clear to which software prototype system the requirements belong.

4.2.12 Modify own user profile

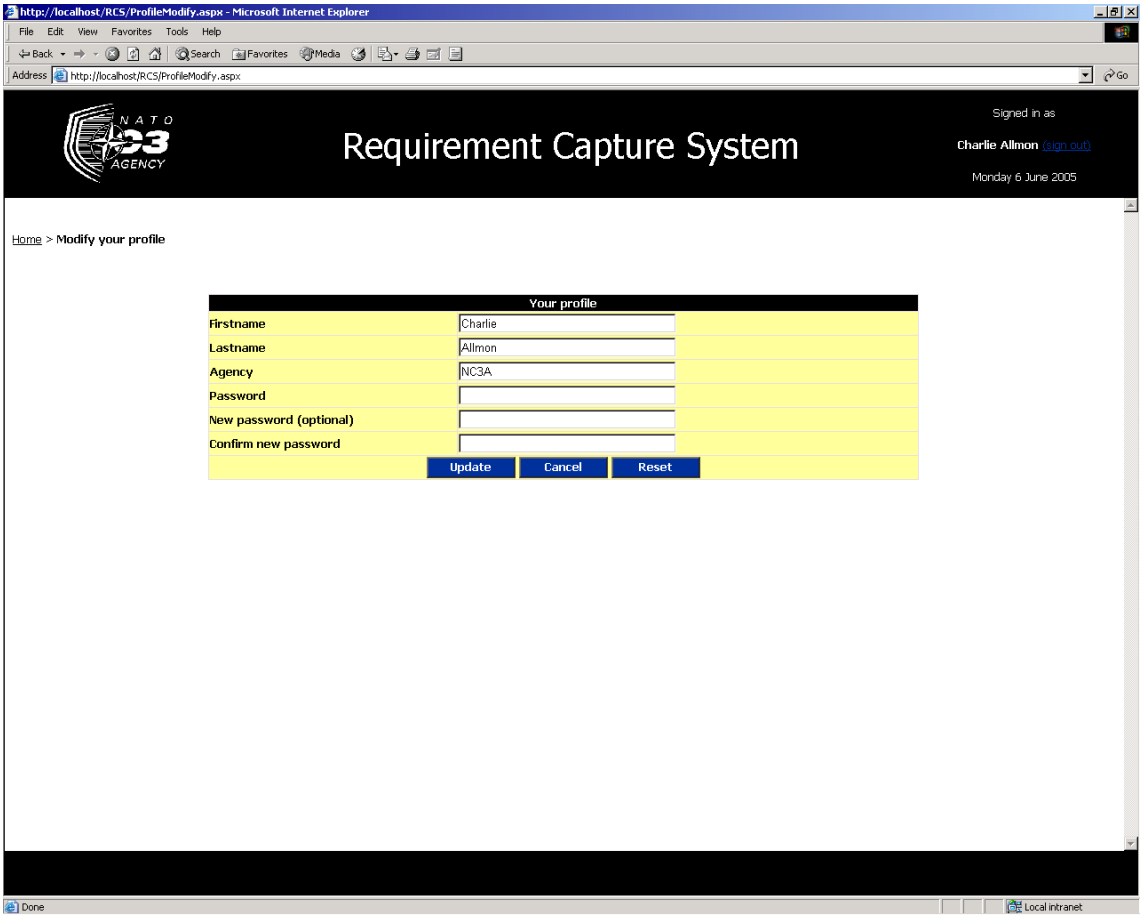


Figure 4.2-12 ProfileModify screen

The user can modify his/her own profile by clicking on the 'Modify own profile' button in the 'PrototypeList' screen. If the user clicks that button the system will display the 'ProfileModify screen'. In this screen the user can fill or modify in his/her profile. The fields that can be filled in or changed are:

- Firstname
- Lastname
- Agency
- Password
- New password
- Confirm new password

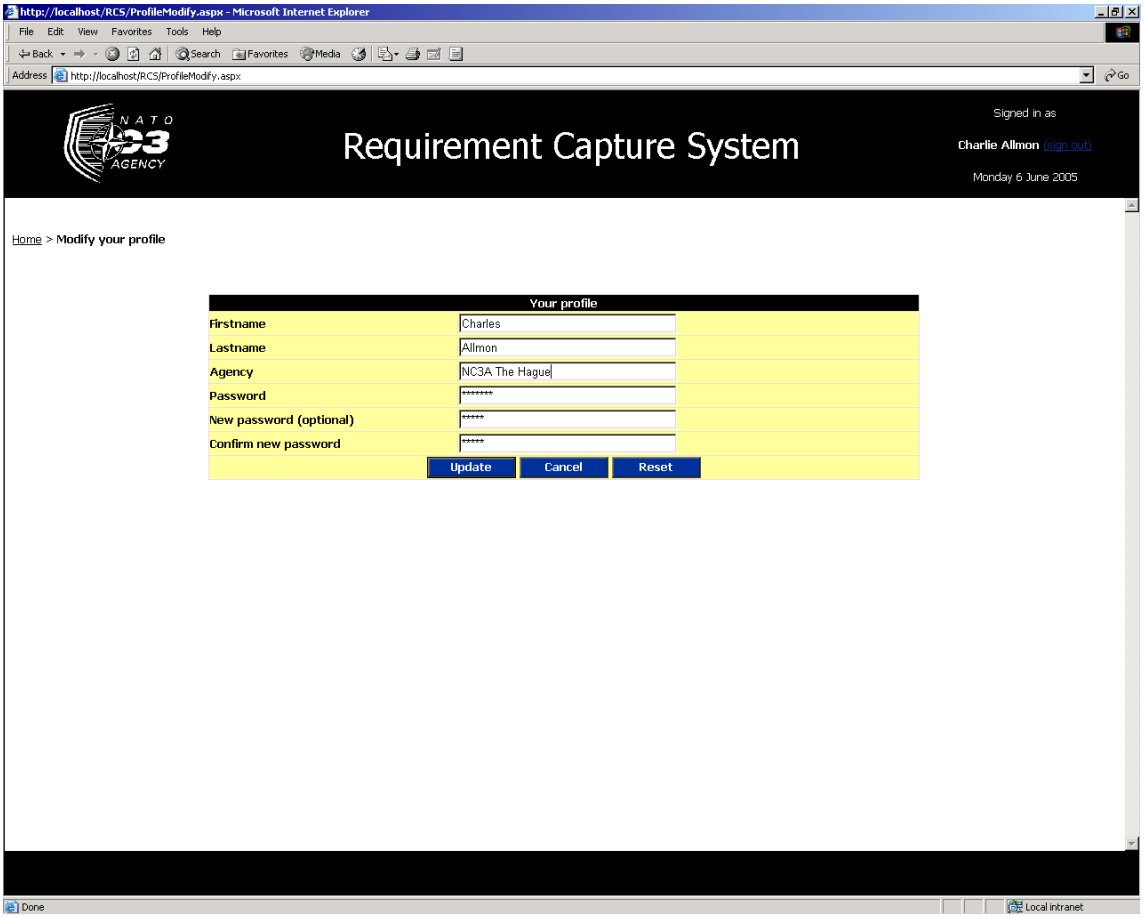


Figure 4.2-13 ProfileModify screen

All fields that are displayed are text fields. When the user clicks the ‘Cancel’ button the system will display the previous screen. The fields can be reset to their initially values by clicking the ‘Reset’ button. To submit the changes in the profile or the new password, the user can click the ‘Update’ button. If the user wants to submit a new password, he/she has to provide her/his current password and must confirm the new password.

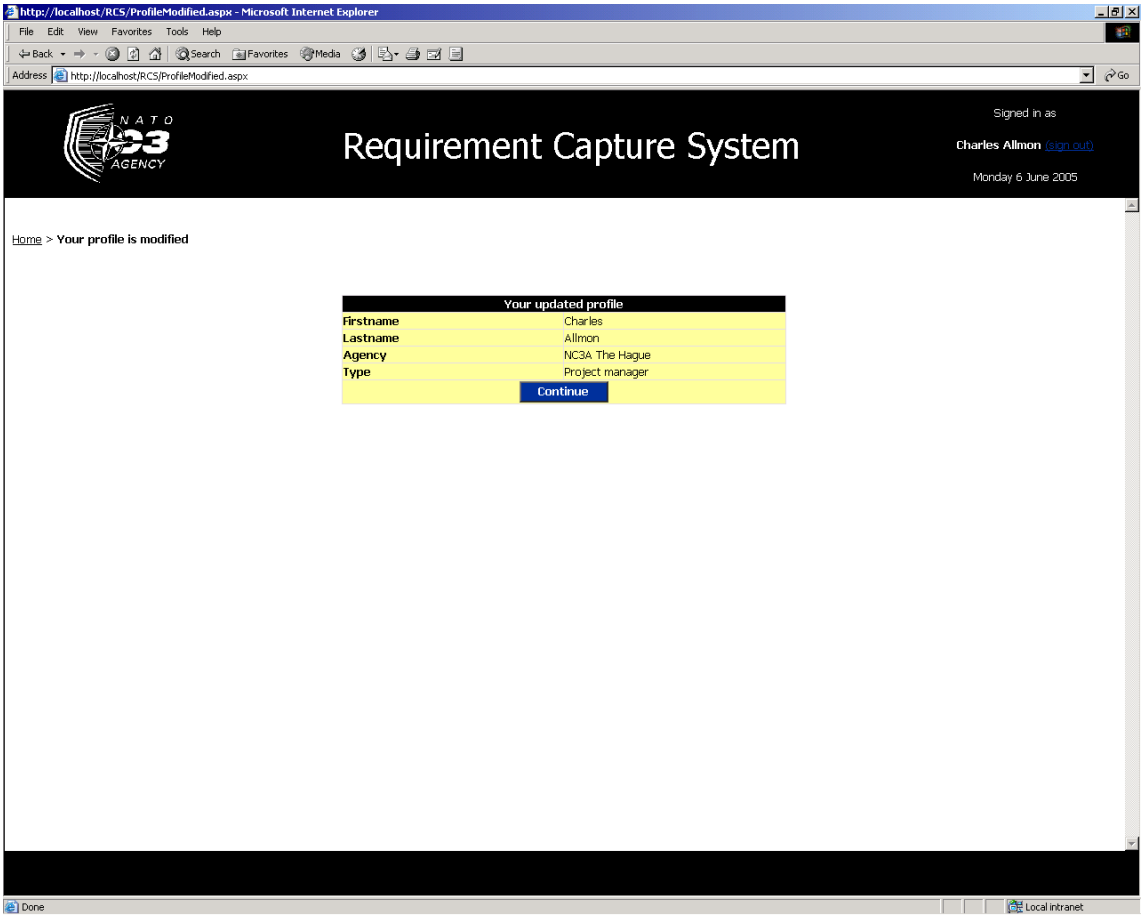


Figure 4.2-14 ProfileModified screen

When the user has submitted the changes to his/her profile, the system will display the 'ProfileModified' screen where the updated profile is displayed. If the user clicks on the 'Continue' button, the system will go back to the 'PrototypeList' screen.

Evaluation results

In the 'ProfileModified' screen the username must be displayed to prevent confusion about the username.

References

Boeken

Duthie, G. Andrew, 'Microsoft ASP.NET Step by Step'. Washington, USA, Microsoft Corporation, 2002

Evans, T., '10 Minute Guide to HTML 4.0, third edition'. Que Corporation, 1998

Hatfield, B., 'ASP.NET for Dummies'. Foster City, California, Hungry Minds, Inc., 2001

Kleppe, A. en Warmer, J., 'Praktisch UML, 3de editie'. Soest'. Pearson Education Benelux BV., 2004

Kroll, O. en Kruchten, P, 'The Rational Unified Process Made Easy: a practitioner's guide to the RUP'. Boston, Pearson Education, Inc., 2003

Kruchten, P., 'The Rational Unified Process: an introduction, third edition'. Boston, Pearson Education, Inc., 2004

Langfeld, L., Noel, M. en Spence, C., 'Microsoft SharePoint 2003 Unleashed'. USA, Sams Publishing, 2004

Phillips, L.A., 'Using HTML 4, fourth edition'. Que Corporation, 1998

Quatrani, T., 'Visual Modeling with Rational Rose 2004 and UML'. Boston, Pearson Education, Inc., 2003

Rapporten

Houten, van, J.J., 'Project Plan – Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Houten, van, J.J., 'Lifecycle Objective Milestone, Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Houten, van, J.J., 'Lifecycle Architecture Milestone, Development of a Requirement Capture System for NATO software prototype systems'. The Hague, 2005

Schoonen, A., 'PlaTo Design Document – phase 1'. The Hague, NATO C3 Agency, 2004

Schoonen, A., 'PlaTo User Guide, version 0.890'. The Hague, NATO C3 Agency, 2004

Internet

http://www.asp.net

http://www.aspnl.com

http://www.aspnet101.com

http://www.codeproject.com

http://www.datagridgirl.com

http://www.dotnet-guide.com

http://www.dotnet247.com

http://www.dotnetjunkies.com

http://www.microsoft.com

http://www.nato.int

http://www.nc3a.nato.int

Intranet

http://nc3a-intranet/

Glossary

ACC	Air Component Command
ACCS	Air Command and Control System
ActD	Active Defence
ASPX	ASP.NET
ASP.NET	Active Server Pages .NET
Bi-SC AIS	Strategic Command Automatic Information System
BMC3I	Battle Management, Command, Control, Communications and Intelligence
C2	Command and Control
C2RC	Command and Control Resource Centre
C3	Command, Control and Communications
CCF	Conventional Counter Force
CCSD	Command and Control System Division
EAD	Extended Air Defence
HHS	Haagse Hogeschool
HQ ARRC	Head Quarters Allied Rapid Reaction Corps
HQ EADTF	Head Quarters Extended Air Defence Task Force
IT&E	Integrated Test & Evaluation
IVIT	Informatievoorzieningen en Informatietechnologie
LSID	Link16 SAM C2 Interoperability Demonstrator
M&S	Modelling and Simulation
MD	Missile Defence
MS	Microsoft
NATO	North Atlantic Treaty Organisation
NC3A	NATO Consultation, Command & Control Agency
PD	Passive Defence
PlaTo	Planning and Tasking Tool
RC	Regional Command

RUP	Rational Unified Process
SAM	Surface to Air Missile
SPS	SharePoint Portal Server
SQL	Structured Query Language
TMD	Theatre Missile Defence
UML	Unified Modeling Language
WSS	Windows SharePoint Services

Table of figures