
Ontwikkelen management applicatie voor routers

Afstudeerverslag van afstudeeropdracht BlackGATE Telecom



Clemens Wijnekus
11058684
Technische Informatica
Haagse Hogeschool

BlackGATE Telecom
Schenkweg 60
2595 XX
Den Haag

Eerste examiner: Cobie van der Hoek
Tweede examiner: Marinus Maris

2 juni 2016

Referaat

Clemens Wijnekus, "Ontwikkeling management applicatie voor routers".
Procesverslag afstudeerstage Technische Informatica, De Haagse Hogeschool 2016.

Samenvatting

BlackGATE is een bedrijf dat VOIP en netwerkoplossingen biedt voor bedrijven. Hiervoor zijn routers nodig, in dit geval worden MikroTik's en Dell SonicWALL's gebruikt. De opdracht omvat het ontwikkelen van een applicatie waarmee deze routers gebundeld te managen zijn. In de applicatie kunnen de routers worden toegevoegd en vanuit de applicatie kunnen instellingen van de routers gewijzigd worden.

Descriptoren

Web applicatie, management software, PHP, RUP, MySQL, Routers, afstudeerverslag.

Voorwoord

Dit document bevat de procesbeschrijving van de afstudeerstage in opdracht van BlackGATE Telecom. De opdracht omvat het ontwikkelen van een applicatie om meerdere routers gestructureerd te beheren. Hiermee kan het beheer proces makkelijker en minder tijdrovend gemaakt worden.

Binnen BlackGATE wil ik graag de volgende mensen speciaal bedanken voor de ondersteuning tijdens dit project: Bart de Water, Ruben Dam en Stefan van den Eeden. Zij hebben mij altijd voorzien van antwoorden op mijn vragen en geholpen op alle punten waar zij dat kunnen. Daarnaast wil ik graag de opdrachtgever en begeleider van het project Aart van Geemert bedanken. Hij heeft mij uitzonderlijk ondersteund en goed begeleid in de project periode. Verder is ook de dank groot aan het overige personeel van BlackGATE.

Aan de kant van school wil ik graag Cobie van der Hoek en Marinus Maris bedanken voor de begeleiding en alle hulp die zij geboden hebben tijdens het afstudeer traject maar ook in de studietijd daarvoor.

Als laatste wil ik mijn studiegenoten Bart van Wijngaarden en Hein van Ruitenbeek bedanken voor afgelopen studenten tijd.

Den Haag, 2 juni 2016

Clemens Wijnekus

Inhoudsopgave

1	Inleiding	1
2	Organisatie	2
3	Opdracht omschrijving	3
3.1	Probleemstelling	3
3.2	Doelstelling	3
3.3	Resultaat	3
3.4	Project methodiek	4
3.4.1	Globale planning	5
4	Inception fase	6
5	Elaboration fase	8
5.1	Iteratie 1: Requirements en systeem eisen analyse	8
5.1.1	Systeem eisen	10
5.1.2	Communicatie protocollen	10
5.1.3	Onderzoek mogelijkheden	12
5.1.4	Conclusie	16
5.2	Iteratie 2: Architectuur document	17
5.3	Iteratie 3: Architectuur document	19
5.4	Iteratie 4: Ontwerp document	20
6	Construction fase	23
6.1	Iteratie 1: Bouwen op prototype	23
6.1.1	Database	23
6.1.2	Gebruikers systeem	24
6.1.3	Klanten en locaties	24
6.1.4	Connectie router via zelfgeschreven API	25
6.2	Iteratie 2: Uitbreiden applicatie	27
6.2.1	Connectie router via connection wrapper	27
6.2.2	Demo applicatie	30
6.3	Iteratie 3: Verwerken feedback demonstratie en testen applicatie	30
6.3.1	Verwerken feedback demonstratie	30
6.3.2	Testen applicatie	31
6.4	Iteratie 4: Verwerken feedback test rapport	31
7	Transition	32
8	Conclusie	33
9	Evaluatie	34
9.1	Proces	34
9.2	Product	34
9.3	Beroepstaken	35
	Afkortingenlijst	36
	Literatuur	37
	Bijlagen	38

1 Inleiding

BlackGATE is een provider voor VOIP en netwerk diensten voor organisaties. De klanten zijn veelal beheerders van verzamelpanden waar meerdere bedrijven gebruik van maken.

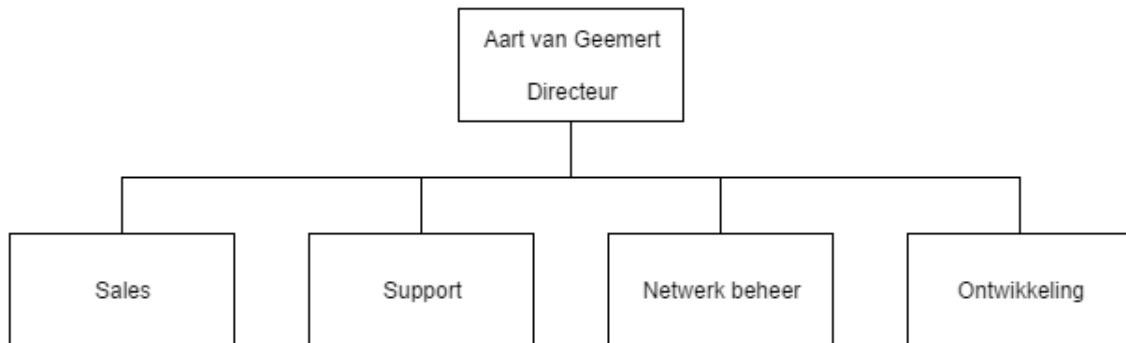
Om deze klanten te voorzien van deze diensten heeft BlackGATE op de locaties van de klanten routers geïnstalleerd. Er worden twee verschillende merken routers, MikroTik en Dell SonicWALL. Deze zijn geconfigureerd om te werken met het netwerk dat BlackGATE zelf beheerd. Het beheer wordt op afstand gedaan vanuit het kantoor van BlackGATE. Op dit moment worden de routers individueel beheerd en ieder afzonderlijk via eigen web interface. Dit beheer proces kost meer tijd dan nodig en vergt meer technische kennis.

Het doel van deze opdracht is om een applicatie te ontwikkelen waarin routers gebundeld kunnen worden en vanuit één punt beheert kunnen worden.

In dit verslag is te lezen hoe het project en de fases, op basis van RUP, daarin zijn verlopen. De organisatie en de specifieke opdrachtomschrijving komen hiervoor nog aan bod.

2 Organisatie

BlackGATE is een kleinschalig bedrijf met elf werknemers, dit is inclusief stagiairs. Het richt zich op het leveren van telecom, in de vorm van VoIP, oplossingen voor bedrijven. Het heeft geen specifieke afdelingen en alle medewerkers zitten samen in één ruimte. Er zijn wel expertises binnen het bedrijf: support, sales, ontwikkeling en netwerkbeheer. In figuur 1 is de onderverdeling gevisualiseerd.



Figuur 1: Organogram BlackGATE

Iedere medewerker binnen BlackGATE valt onder één van deze expertises en de directeur, Aart van Geemert, staat als hoofd bovenaan. Tevens is Aart de opdrachtgever en bedrijfsmentor tijdens dit project. Aart heeft het bedrijf opgericht als Zelfstandige Zonder Personeel. Elk jaar groeit BlackGATE meer en het is ondertussen een volwaardig bedrijf met meerdere mensen in dienst.

Onder klanten worden voornamelijk bedrijfsverzamel panden, zoals “The Office Operators” verstaan. Maar ook zelfstandige bedrijven zoals “Discus Groeneveld”.

De afstudeeropdracht valt onder de expertise “Ontwikkeling”. Onder ontwikkeling vallen de programmeur en grafisch ontwerper. Voor de afstudeeropdracht is er wel communicatie nodig met de support- en netwerkbeheer experts. Dit zijn namelijk de eindgebruikers van de te ontwikkelen applicatie. Deze mensen moeten de routers kunnen beheren in geval van problemen of omdat er wijzigingen doorgevoerd moeten worden om welke reden dan ook. Dit beheer moet gaan via de te ontwikkelen applicatie.

3 Opdracht omschrijving

BlackGATE heeft voor de klanten die zij hebben, meerdere eigen routers op externe locaties in beheer om deze klanten van de services te kunnen voorzien. De routers zijn aanwezig op de locaties van de klanten. Dit zijn twee verschillende soorten routers, Dell SonicWALL's en MikroTik's. BlackGATE is een groeiend bedrijf en het aantal routers in beheer groeit mee met het aantal klanten. Het beheer van deze routers moet in goede banen blijven verlopen. Er is geen gebruiksvriendelijke manier om deze routers te beheren. Ook is dit een technisch onderwerp en is te technisch voor de niet technisch onderlegde support afdeling om deze routers te beheren. Problemen moeten op dit moment worden uitbesteed door support aan het netwerkbeheer in plaats van dat zij het zelf op kunnen lossen.

3.1 Probleemstelling

BlackGATE heeft op dit moment meerdere eigen routers op externe locaties in beheer. De routers moeten nu afzonderlijk en via bijvoorbeeld command line (SSH) of web interface gemanaged worden. Het proces om een wijziging door te voeren op alle routers is daarom vrij inefficiënt. Een command line is totaal niet gebruiksvriendelijk en niet voor de mensen zonder de juiste technische kennis bedoeld.

3.2 Doelstelling

Via de applicatie kunnen meerdere routers op een manier beheerd worden via een gebruiksvriendelijkere weg. Dit zal complexere wijzigingen op meerdere routers gebruiksvriendelijker maken en tijd besparen omdat de routers niet meer individueel gewijzigd te hoeven worden. Met de te ontwikkelen applicatie kan het beheer proces efficiënter gemaakt worden.

3.3 Resultaat

Een applicatie waarin alle routers gebundeld gemanaged kunnen worden. Daarnaast is er een requirements en probleemdomein analyse document, architectuur document, prototype, ontwerp document en test rapport.

3.4 Project methodiek

Het project omvat het ontwikkelen van software. Er moet ook gecommuniceerd worden tussen software en hardware. Met deze communicatie wordt bedoeld dat de te ontwikkelen applicatie moet kunnen communiceren met de routers. Omdat er niets bekend is over de mogelijkheden van de hard- en software is het belangrijk dat er veel aandacht wordt besteed aan het vooronderzoek. Dat heeft als gevolg dat er veel tijd ingepland moet worden voor onderzoek.

Er zijn 3 project methodes die geschikt lijken voor dit project. Er is SCRUM, maar ook RUP en waterval. Dit zijn allen methodes die gebruikt kunnen worden voor software ontwikkeling projecten.

Voor de project methodiek is gekozen RUP (Rational Unified Process) te gebruiken. Er is al ervaring met RUP opgedaan tijdens de stage periode en school projecten. SCRUM is geen optie voor dit project, omdat het project door één persoon wordt uitgevoerd en er weinig teamverband is. SCRUM bevat onder meer SCRUM meetings, deze met één persoon doen is niet nuttig. Tevens kan er niet geïtereerd worden maar moet er aan het einde van elke sprint een nieuw onderdeel worden opgeleverd. De waterval methode is niet geschikt omdat het itereren over bepaalde onderdelen daarbij niet toegepast kan worden. Dit is wel van belang omdat er weinig tot geen kennis beschikbaar is binnen het project, het een geheel nieuwe applicatie oplevert en er met veel stakeholders, de opdrachtgever, support en netwerkbeheer, rekening gehouden moet worden. Bij projecten waarin software wordt ontwikkeld werkt RUP naar ervaring erg goed. Met RUP wordt er namelijk eerste een goed ontwerp gemaakt met een prototype waarover feedback wordt gegeven. Hierna wordt er nog een keer geïtereerd om de ontwerpen en het prototype te verbeteren. Echter zijn niet alle onderdelen van RUP gebruikt. De onderdelen die zijn gebruikt zijn voldoende om de kwaliteit te waarborgen en het project goed af te ronden. Het maken van iteraties is een erg bruikbaar onderdeel voor dit project. Dit kan voor betere feedback verwerking zorgen en een hogere kwaliteit van het eind resultaat.

RUP definieert vier fases in het project: inception, elaboration, construction en transition. Alle vier de fases worden doorlopen in dit project. De inception fase legt de basis voor het project met een vision document en als mijlpaal het Life Cycle Objective. Hier wordt gekeken of het project überhaupt wel haalbaar is. Er wordt een basis use case opgesteld en gekeken naar de afbakening en de risico's van het project.

Vervolgens wordt in de elaboration fase het probleemdomein geanalyseerd en de systeemeisen opgesteld. Dit is voornamelijk onderzoek. Hierna worden het architectuur document opgesteld en een prototype gemaakt. Met het prototype kan worden aangetoond dat het project werkbaar is en bepaalde functies mogelijk zijn of niet. Daar zal met de feedback die gegeven wordt nog een keer over worden geïtereerd. Vervolgens komt er een ontwerp document waarin de definitieve systeemeisen en de ontwerpen voor de applicatie staan. De uiteindelijke mijlpaal van de elaboration fase is de Life Cycle Architecture.

De construction fase omvat het daadwerkelijk ontwikkelen van de applicatie. Er wordt doorontwikkeld op het prototype dat is gebouwd in de elaboration fase. Met een tussen resultaat wordt een demo gegeven en getest met de stakeholders. Hieronder vallen de support- en de netwerkbeheer afdelingen zowel als de opdrachtgever, dit zijn de personen die moeten gaan werken met de applicatie. Nadat de feedback van deze demo verwerkt is, wordt er een test plan opgesteld en getest met de stakeholders. Het testplan is opgesteld uit de systeem eisen van de stakeholders, die al vast liggen in de elaboration fase. Het schrijven van het test plan is verlaat omdat dat beter uit kwam voor de planning zodat er in de elaboration fase gefocust kan worden op het onderzoek en de architectuur. Hieruit volgt een test rapport. Uiteindelijk zal versie 1.0 uitgebracht worden.

Nadat de eerste versie is uitgebracht, treed de transition fase aan. In deze fase zal de applicatie in een live omgeving gebracht worden en de documentatie overgedragen worden aan BlackGATE. Dit is om BlackGATE de basis te geven om de applicatie te onderhouden. Hier wordt de Application Release mijlpaal gehaald.

3.4.1 Globale planning

Inception fase:

Voor de inception fase staat het vision document op de planning. Dit document bevat de basisbeginselen van het project.

Elaboration fase:

In de elaboration fase wordt eerst een architectuur document geschreven. Op basis hiervan wordt een prototype gebouwd. Met dit prototype wordt aangetoond dat de opdracht mogelijk is en op welke manier. Na feedback zal hier overheen worden geïtereerd om het prototype verder uit te breiden. Vervolgens wordt er een ontwerp document gemaakt op basis van de architectuur en de bevindingen.

Construction fase:

De ontwikkeling van de applicatie gebeurt in de construction fase en de mijlpaal die hier bereikt wordt is de Initial Operational Capability. Het prototype wordt verder uitgebreid. Verder komt er een demo aan bod. Deze demo wordt gegeven aan de support- en netwerkbeheer afdelingen en de opdrachtgever. Aan het einde van de fase is er een test rapport. Hierin staan alle bevindingen die de testers op hebben gedaan tijdens het testen van de applicatie.

Transition fase:

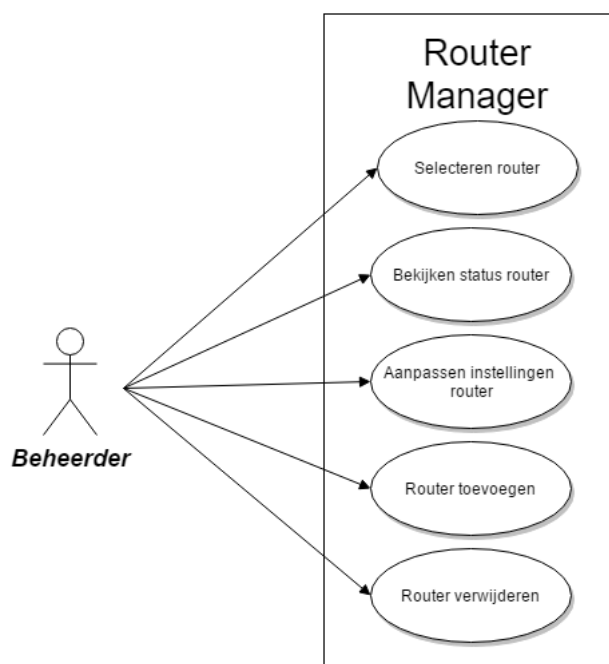
De belangrijkste doelen in deze fase zijn het live brengen van de applicatie en het overdragen van de documentatie. Als alle doelen zijn behaald is de Release mijlpaal gehaald.

4 Inception fase

Inception is de eerste fase van het project. In deze fase wordt een vision document opgesteld om het project vorm te geven. Tevens wordt hier een planning in beschreven. In de inception fase worden inschattingen gemaakt en de basis voor de rest van het project gelegd.

De opdracht wordt gedefinieerd en uitgewerkt om duidelijk te beschrijven wat het probleem, het doel en het verwachte resultaat is.

Vervolgens wordt een use case diagram gemaakt. Het use case diagram, te zien in figuur 2 hieronder, is een globaal diagram dat de basis functionaliteiten van de applicatie specificeert. Het moet mogelijk zijn routers toe te voegen en te verwijderen. Daarnaast de meest belangrijke functionaliteit om de router instellingen te kunnen wijzigen. Er zijn op dit moment nog geen systeem eisen duidelijk, er moet nog gesproken worden met de opdrachtgever, daarom is gekozen voor de term “Aanpassen instellingen router”. Dit komt voort uit de opdrachtomschrijving, hierin wordt beschreven dat de instellingen gewijzigd moeten kunnen worden. Ook moet de status van de router ingezien kunnen worden, oftewel monitoring. Omdat er hier ook nog niet duidelijk is wat er precies gemonitord moet worden, is hier gekozen voor de term “Bekijken status router”. Voordat de monitoring ingezien kan worden van een specifieke router, moet deze router eerst geselecteerd worden omdat de router manager meerdere routers betreft. De volledige opdracht omschrijving is te vinden in bijlage Vision Document hoofdstuk 2. De specifieke acties worden verder uitgewerkt in de elaboration fase. De use case geeft een beeld van wat de applicatie moet kunnen en of deze reëel is.



Figuur 2: Use Case diagram

Voor de afbakening is nog te weinig bekend. Er kan nog geen communicatie protocol of programmeertaal gespecificeerd worden. Het project is echter wel afgebakend op de door BlackGATE beheerde routers. De applicatie werkt alleen met de routers van BlackGATE, hiermee wordt bedoeld dat er niet gegarandeerd kan worden dat deze applicatie met andere routers dan de MikroTik's of Dell SonicWALL's werkt.

De andere afbakening is voor het gebruik van de applicatie. De applicatie is alleen voor gebruik door medewerkers van BlackGATE. Dit betekent dat de applicatie afgeschermd moet worden middels een gebruikers authenticatie systeem.

Met een risicoanalyse wordt van te voren bepaald waar punten liggen waar op gelet moet worden bij de ontwikkeling van de applicatie. Door de risico's nu al te analyseren kan er in de elaboration fase al rekening mee gehouden worden. Ondanks dat het gebruik alleen voor medewerkers van BlackGATE is bedoeld, is er nog steeds een reële kans op hack aanvallen. Dit brengt een risico met zich mee, namelijk dat een ongeautoriseerde gebruiker wijzigingen doorvoert op een router. Eveneens met een gebruikers systeem is het mogelijk om binnen te komen in de applicatie, als dit de enige vorm van beveiliging is. De applicatie zal derhalve zo veel mogelijk beveiligingsmethodieken bevatten. Echter, hiermee kunnen niet alle beveiligings risico's worden afgevangen.

Bijvoorbeeld een SSL (Secure Socket Layer) certificaat, om de verbinding met de applicatie van HTTPS (Hypertext Transfer Protocol Secure) te kunnen voorzien, kan niet worden geforceerd met de applicatie. Een SSL certificaat moet namelijk op de webserver geïnstalleerd worden. Er zullen wel adviezen worden gegeven aan BlackGATE om beveiligingsrisico's, die niet afgevangen kunnen worden met de applicatie, in te dammen. Tevens is er een audit log om in te zien wie wat heeft gewijzigd. Hiermee kan overzichtelijk worden terug gekeken welke wijzigingen er plaats hebben gevonden, door wie en op welke router.

Het is ook mogelijk dat de verbinding tussen de applicatie en de router weg valt terwijl de applicatie bezig is met het aanmaken of wijzigen van instellingen. Het kan zijn dat hierdoor de wijzigingen niet volledig worden doorgevoerd. Dit kan problemen veroorzaken op de router zelf. Door verificatie of deze wijzigingen goed zijn doorgevoerd kan worden gewaarborgd dat het proces goed is afgerond. Mocht dit niet goed gaan dan moeten de wijzigingen terug gedraaid worden.

De volgende risico's zijn hiervoor geformuleerd:

- Wijzigingen worden niet volledig doorgevoerd in de router
- Een ongeautoriseerde gebruiker voert wijzigingen door op een router via de applicatie

De projectorganisatie bevat de betrokken personen, de tussenresultaten van het project en de planning. De tussenresultaten omvatten mijlpalen in het project. Uit de analyse van het probleemdomein en het architectuur document komt een prototype voort. Dit prototype wordt gedemonstreerd aan de opdrachtgever. Met het prototype wordt bewezen dat het mogelijk is de routers te managen met de gekozen opties, zoals programmeertaal en manier van communiceren. Tevens bewijst het prototype dat de specifieke instellingen, conform de eisen van de opdrachtgever, te wijzigen zijn met de te ontwikkelen applicatie. Dit komt verder aan bod in de elaboration fase.

Er is gekozen om de probleemdomein- en systeemeisen analyse in een apart document op te nemen. Dit wordt gedaan omdat het een zelfstandig deel is van de elaboration fase. De analyse richt zich op het probleemdomein en de systeemeisen. Dit komt later terug in het architectuur document en het ontwerp document.

Als laatste is er het test plan en rapport. Deze zijn om de test fase in de construction fase te ondersteunen. De bevindingen hieruit kunnen daarna worden verwerkt in de applicatie.

In de fase planning staat het gehele project en de fases weergegeven in een Gantt kaart. De iteraties per fase en de mijlpalen zijn aangegeven.

5 Elaboration fase

Als eerste wordt er in de elaboration fase een analyse uitgevoerd op het huidige probleemdomein en de systeemeisen van de stakeholders.

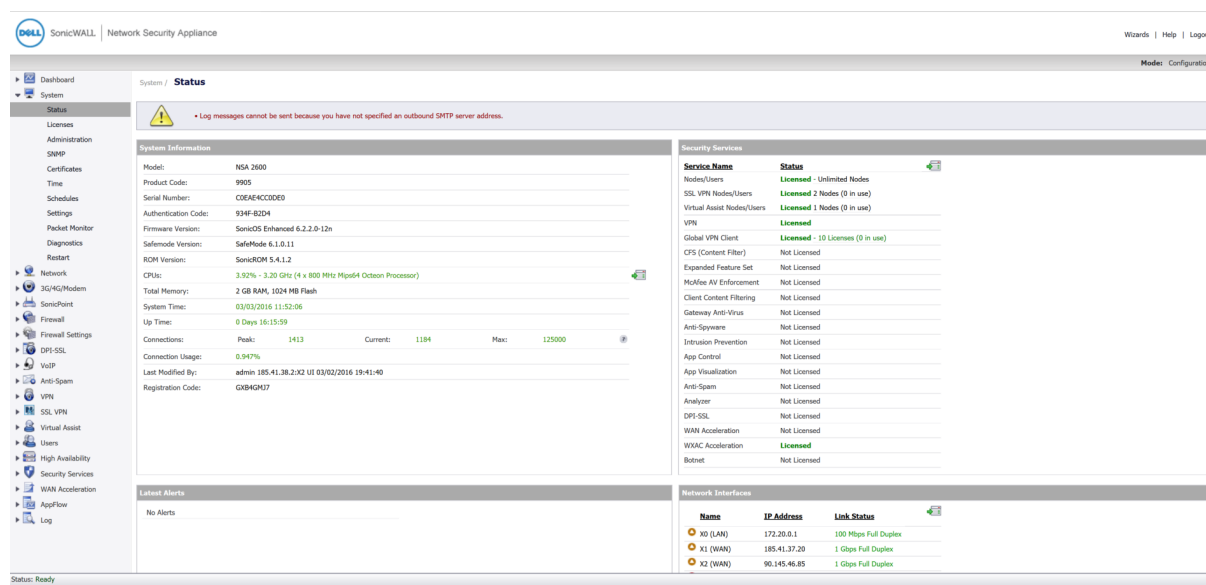
De problemen in de huidige situatie worden uitgebreider geanalyseerd waarna er gekeken wordt naar de mogelijkheden van de hardware. Dit is allemaal vastgelegd in het Probleem domein & requirements analyse document. Deze is bijgevoegd in de bijlage Probleem domein & requirements analyse document. Het onderzoek naar deze mogelijkheden is groter uitpakket dan verwacht. Dit is omdat het onderzoek verschillende onverwachte wendingen heeft genomen.

Met het komende onderzoek moeten de volgende vragen beantwoord worden:

- Wat zijn de mogelijkheden van de routers qua management op afstand?
- Welk communicatie protocol kan gebruikt worden voor het management?
- Welke programmeer taal is het meest geschikt?

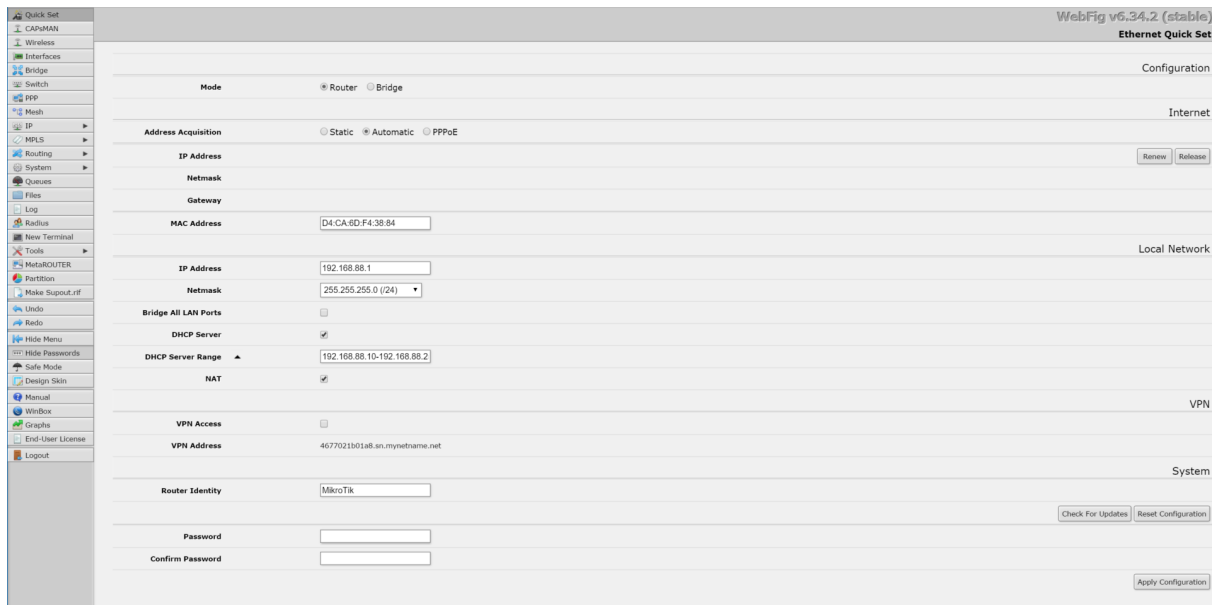
5.1 Iteratie 1: Requirements en systeem eisen analyse

Als eerste wordt de SonicWALL geanalyseerd. Er wordt een lijst met verschillende types van de SonicWALL's die BlackGATE in beheer heeft opgevraagd. Vervolgens is er een SonicWALL NSA 2600 verkregen om mee te testen, dit is een van de type routers die BlackGATE in beheer heeft. De GUI van besturingssysteem zoals te zien is in figuur 3, SonicOS, ziet er gestructureerd uit. De tekst in het figuur is specifiek voor de router en is in dit geval niet belangrijk omdat de focus ligt op het uiterlijk van het beheer paneel. Op dit moment worden alle standaard router functies achterwege gelaten en wordt alleen gekeken naar de mogelijke communicatie protocollen. Alles lijkt goed, SonicOS ondersteunt zowel SSH (Secure Shell), Telnet en SNMP (Simple Network Management Protocol). Dit zijn bekende en standaard communicatie protocollen. Ook ondersteunt het GMS, Global Management System, van Dell. Het GMS is een software pakket ontwikkeld door Dell om SonicWALL's mee te managen. Er wordt niet naar de mogelijkheden van GMS gekeken omdat dit buiten de scope van het project valt en niet binnen het tijdsbestek van het project past. Tevens wordt dit niet ondersteunt door MikroTik wat een reden is om het niet te gebruiken.



Figuur 3: SonicOS op SonicWALL NSA 2600

MikroTik maakt gebruik van RouterOS. Met een test opstelling in combinatie met een MikroTik van het type RouterBOARD 750 is de web interface van RouterOS, genaamd WebFig, bekeken. RouterOS WebFig, te zien in figuur 4, is een eenvoudig beheer paneel. MikroTiks zijn kleinere en goedkopere routers dan de SonicWALL's, ookal beschikken ze over dezelfde mogelijkheden. RouterOS ondersteunt net als de SonicWALL's de protocollen SSH, Telnet en SNMP. Door een tip van één van de netwerkbeheerders en onderzoek in de documentatie is er ook onderzoek naar de API (Application Program Interface) van RouterOS gedaan. Deze API is te gebruiken om data uit de router uit te lezen of de router te configureren door er commando's naar toe te sturen. Uit voorbeelden, te vinden in de documentatie, van programma's in verschillende programmeertalen in combinatie met de API blijkt dat de API op basis van sockets werkt. Socket verbindingen communiceren via TCP (Transmission Control Protocol).



Figuur 4: RouterOS op een MikroTik RouterBOARD 750

De conclusie uit deze analyse is dat de beide routers en besturingssystemen SNMP, SSH en Telnet ondersteunen. MikroTik ondersteunt als extra een API en de Dell SonicWALL ondersteunt als extra GMS.

5.1.1 Systeem eisen

Middels een gesprek met de opdrachtgever worden de systeem eisen duidelijk. Bij dit gesprek is ook Stefan, van netwerkbeheer, betrokken. Hij vertegenwoordigt in dit gesprek ook het support team. Dit zijn de nog onbewerkte systeem eisen die terug gaan komen in het architectuur document. Dit gesprek vindt plaats tijdens het onderzoek. Hieruit zijn de volgende functionele systeem eisen gekomen:

- Iedere gebruiker moet routers toe kunnen voegen
- Iedere gebruiker moet routers kunnen verwijderen
- Iedere gebruiker kan VLAN instellingen wijzigen van een router
- Iedere gebruiker kan de instelling van interfaces van een router wijzigen
- Iedere gebruiker kan NAT policies van een router beheren
- Iedere gebruiker kan DHCP instellingen van een router wijzigen
- Iedere gebruiker kan de ARP tabel van een router inzien
- Er is een zoek functie op MAC adres in ARP tabel van een router
- Er is een zoek functie in de DHCP tabel van een router
- Iedere gebruiker kan “routed subnets” aanmaken op een router
- Er kunnen gebruikers aangemaakt worden in de applicatie
- Er kunnen gebruikers verwijderd worden uit de applicatie
- Er is een dashboard met monitoring van de routers beheerd via de applicatie beschikbaar

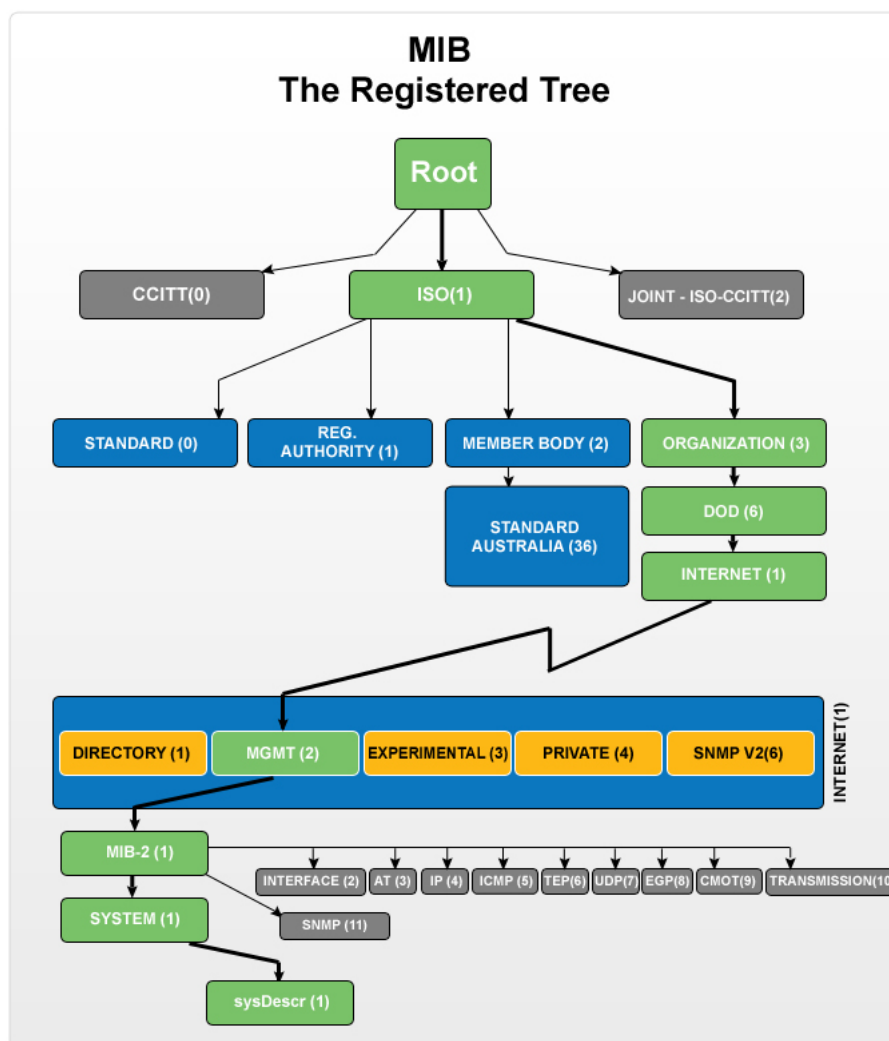
5.1.2 Communicatie protocollen

Er wordt eerst gefocust op de protocollen die beide routers ondersteunen. Een uitgebreide beschrijving voor de protocollen is te vinden in hoofdstuk 4 van bijlage Requirements & probleem domein analyse. De protocollen Telnet en SSH werken beide op basis van TCP. SSH is de opvolger van Telnet, Telnet is zeer verouderd (1969) en ook niet goed beveiligd (“GCIH Practical Assignment TCP Port 23”, 2000). SSH is daarentegen niet verouderd en wel goed beveiligd. Beide zijn protocollen ontwikkeld voor gebruik met de CLI (Command Line Interface). Het feit dat deze twee protocollen werken op TCP zou betekenen dat er met sockets geprogrammeerd kan worden om zo een verbinding op te zetten.

Binnen de programmeertalen is er geen standaard bibliotheek voor deze protocollen waaruit het feit volgt dat de koppeling tussen de programmeertaal en het communicatie protocol zelf geprogrammeerd moet worden of dat er een door derden geschreven bibliotheek gebruikt moet worden. Met het zelf programmeren van de koppeling moet er voldaan worden aan alle eisen die de protocollen stellen. Dit levert veel en onnodig werk op als dit wordt vergeleken met SNMP. SNMP wordt door bijvoorbeeld PHP (Hypertext Preprocessor) wel standaard ondersteund. SNMP is geïntegreerd in de programmeertaal.

Er is gekeken naar de versies van SNMP: v1, v2c en v3. Deze worden alle drie ondersteund door beide routers. Versie 3 is op het moment van schrijven gedefinieerd als standaard door het IETF. SNMP staat voor Simple Network Management Protocol en wordt voornamelijk gebruikt om netwerk apparatuur te monitoren. Versie 1 en 2c bieden weinig beveiliging. Versie 2 is een uitgebreidere versie van versie 1 met dezelfde beveiliging, namelijk community based. Door middel van een enkel woord, de community naam, worden commando's geaccepteerd of geweigerd. De community naam meegestuurd met het commando moet kloppen met de community naam ingesteld op het ontvangend apparaat. SNMPv3 heeft betere beveiligingsopties, er is beveiliging van de verbinding toegevoegd en de authenticatie is verbeterd.

SNMP werkt met OID's (Object Identifiers) en MIB's (Management Information Base). Een OID wordt gebruikt om een eigenschap van het systeem aan te duiden. Deze OID's zijn gestructureerd en deze structuur is opgeslagen in een MIB, te zien in figuur 5. Een voorbeeld van een OID is ".1.3.6.1.2.1.1.1", dit is de standaard OID voor de eigenschap "System description". In een MIB wordt deze aangeduid als ".iso.org.dod.internet.mgmt.mib-2.system.sysDescr".



Figuur 5: MiB Object Identifiers structuur

Met SNMP kunnen GET en SET commando's uitgevoerd worden, hiermee kunnen waarden van instellingen uitgelezen of gewijzigd worden. Vanwege de standaard ondersteuning vanuit de routers en programmeertalen en de beveiligingsopties die reeds voorhanden zijn van SNMPv3 is SNMP de beste optie tot nu toe.

Er wordt ook aandacht besteed aan de API van MikroTik's ondanks dat de SonicWALL's deze niet hebben en dus niet bruikbaar is om de SonicWall's mee te beheren. MikroTik heeft uitgebreide documentatie over alle aspecten van RouterOS en de routers zelf. Uit de documentatie is op te maken dat de API gebruikt kan worden om de router te beheren en zo simpel mogelijk gemaakt is voor gebruik. De commando's zijn bijna identiek aan die van de command line. Er zijn verschillende voorbeelden te vinden van programma's die werken met de API om een router te kunnen beheren. Zo ook voor PHP. De API is ontworpen om met eigen geschreven software de MikroTiks te kunnen beheren.

5.1.3 Onderzoek mogelijkheden

Telnet valt af als communicatie protocol omdat dit niet goed beveiligd is en gericht op de command line. Voor SSH geldt hetzelfde ondanks dat dit wel goed is beveiligd maar wel gericht is op de command line. Daarom is er gekozen voor SNMP, omdat dit protocol niet alleen simpel is maar ook brede ondersteuning heeft vanuit de programmeer talen waaronder PHP. Er is voor PHP gekozen omdat volgens de systeem eisen de applicatie een web applicatie moet worden. Van de web programmeertalen is PHP gekozen omdat daar veruit de meeste ervaring mee is binnen dit project maar ook binnen BlackGATE. PHP is syntactisch makkelijk en erg goed te combineren met javascript, HTML en CSS. Er is ook gekeken naar Ruby, waar ook ervaring mee is binnen BlackGATE maar niet binnen het project. Ruby is compleet nieuw binnen het project en het leren van een nieuwe programmeer taal past niet binnen het tijdsbestek van dit project. De opdrachtgever heeft geen voorkeur uitgesproken voor een programmeer taal. Dit maakt PHP de beste keuze binnen dit project.

Als eerste is getest met de SonicWALL's. Er is een test opstelling gemaakt met de SonicWALL TZ 300 en een simpel PHP script. SNMPv2c wordt geconfigureerd volgens de documentatie van Dell. Het PHP script leest de waarde "System name" uit, wijzigt deze en leest de waarde nogmaals uit door middel van GET en SET commando's. Deze test is een simpel voorbeeld om te bevestigen dat SNMP een optie is om het uiteindelijke doel te bereiken. Uit de eerste test is gebleken dat met versie 2c geen SET commando's uitgevoerd kunnen worden. De waarde van de systeemnaam kan uitgelezen worden, maar het wijzigen genereert een foutmelding. De foutmelding gaf aan dat de tijd om het commando uit te voeren verstreken is en dus niet meer uitgevoerd wordt.

Om te verifiëren dat de fout niet aan PHP ligt, is er een MIB browser bijgehaald genaamd "ManagementEngine MibBrowser". Dit is een gratis programma om de OID's van een apparaat dat SNMP ondersteunt uit te lezen. Middels de MIB browser kunnen wederom alle OID's worden uitgelezen. Zodra er geprobeerd wordt een SET commando uit te voeren om de waarde van een OID te wijzigen, verstrijkt de tijd net als bij het PHP script. Dit bevestigt dat de fout niet aan PHP of het script ligt. De MIB browser geeft aan dat vele OID's "Read only" zijn, oftewel alleen te lezen. Bij enkele OID's wordt aangegeven dat deze "Read-Write" toegankelijk zijn, zoals bijvoorbeeld de systeem naam. Dit zou suggereren dat deze wel gewijzigd kan worden maar dat er een ander onbekend probleem zich voordoet, waarschijnlijk op de router.

Uit het onderzoek naar SNMP is gebleken dat SNMPv2c een inadequate vorm van beveiliging hanteert. Dit zou een reden kunnen zijn waarom de SonicWALL niet reageert op de SNMP SET commando's, uit veiligheidsoverwegingen, wat de SonicWALL wel goed beveiligt maakt. Daarom is vervolgens SNMP versie 3 geconfigureerd om daar dezelfde test mee uit te voeren. De configuratie is verlopen volgens de Dell documentatie. Er is gekozen voor SHA1 voor de authenticatie en AES voor de beveiliging in dit test scenario. Dit omdat deze beveiligingsopties al voor handen zijn en beter zijn dan de andere opties. MD5 is het alternatief voor de encryptie van de authenticatie, dit algoritme is daarentegen helemaal gekraakt ten opzichte van het verzwakte SHA1 ("Analyse: zwakke wachtwoorden op Tweakers.net", 2011) ("MD5 considered harmful today", 2008). AES is gekozen boven DES omdat AES als standaard is aangewezen en (nog) niet gekraakt is in tegenoverstelling tot DES, dat aantoonbaar is gekraakt (Kumar et al., 2006).

Het PHP test script is aangepast om SNMP versie 3 te kunnen gebruiken. Hetzelfde scenario wordt nogmaals uitgevoerd maar ook hier treedt er een foutmelding op. Het is niet dezelfde foutmelding als bij de versie 2 test. Dit keer is de foutmelding “Toegang geweigerd”, zoals te zien is in figuur 6.

```
SonicWALL TZ 300 TEST
Warning: SNMP::set(): Error in packet at 0 object_id: authorizationError (access denied to that object) in C:\xampp\htdocs\public\snmp\index.php on line 24
SonicWALL TZ 300 TEST
```

Figuur 6: SNMP error die de router teruggeeft op het SET commando

De foutmelding suggereert dat de gebruikersnaam en wachtwoord, die gebruikt worden om te authenticeren, fout zijn. Het wel kunnen uitlezen van de systeem naam bewijst hier het tegendeel van. Om er toch zeker van te zijn is het wachtwoord naar een willekeurig woord veranderd in het PHP script en de test nogmaals uitgevoerd. Met deze test werd bevestigd dat het wachtwoord wel degelijk correct is, want met het willekeurige wachtwoord kan helemaal niets worden uitgelezen. Ook hier is weer geprobeerd met de MIB browser hetzelfde te doen. Deze retourneerde echter exact dezelfde foutmelding als te zien is in het bovenstaande figuur.

Ondanks dat de Dell documentatie beschrijft dat het wel mogelijk moet zijn om met SNMP versie 3 waardes aan te passen, is dit tot nu toe niet gelukt. Daarom is contact opgenomen met de technische ondersteuning van Dell. Deze probeert middels email communicatie te helpen met het probleem. Dell support vroeg zelf om een “remote session” om live te zien wat het probleem is en probeert zelf het probleem na te bootsen.

In de tussentijd dat Dell support naar een oplossing voor het probleem zoekt, is er verder onderzoek gedaan naar mogelijkheden. In de documentatie van SonicOS versie 6.2 is er een commando gevonden wat een mogelijke oplossing zou kunnen bieden: “Cli rest-api enable”. Dit commando suggereert dat SonicOS een API zou hebben. Het zou mogelijk kunnen zijn de router te configureren middels deze API. Uit de documentatie voor het commando is helaas niet op te maken wat het commando precies inhoudt. Het commando is uitgevoerd op de router maar dat geeft geen output. De meeste API's bevinden zich op de url “/api/”. Een logisch gevolg is het navigeren naar het IP adres van de router inclusief “/api/”. Dit levert het volgende op zoals te zien in figuur 7.

```
{
  "systime": 1457617914,
  "status": [
    {
      "info": [
        {
          "type": "error",
          "code": "E_HTTP_404",
          "message": "Invalid API call."
        }
      ]
    }
  ]
}
```

Figuur 7: Ongeldige API aanroep

Het feit dat de foutmelding “Invalid API call” wordt teruggegeven, suggereert dat er een API bestaat. Er is verder geen documentatie te vinden van Dell over deze API. Dell support bevestigt tevens dat hier niets over bekend is. De API is een doodlopend spoor omdat er niets mee gedaan kan worden en het vrijwel niet in de documentatie beschreven wordt.

Dell heeft zelf een applicatie ontwikkeld, genaamd Global Management System (GMS), waarmee de SonicWALL's gemanaged kunnen worden. Er is onderzoek gedaan in de documentatie naar hoe GMS communiceert met de SonicWALL's maar daar is helaas niets over te vinden. Dus is er een test opstelling gemaakt met een proef versie van GMS.

De verbinding tussen GMS en de SonicWALL is beveiligd, dus het onderscheppen van packets leverde niets op omdat deze versleuteld zijn met een onbekende encryptie methode. Ook is er gekeken naar de bron code van GMS, door middel van reverse engineering, maar de bron code is “obfuscated” waardoor het moeilijk te lezen is. Er valt alleen uit op te maken dat GMS verzoeken stuurt naar CGI (Common Gateway Interface) bestanden op de router. De parameters die daar bij horen zijn helaas niet te achterhalen. Het nabouwen van de software is bijna onmogelijk en past niet binnen het tijdsbestek en budget van dit project.

In de tussentijd heeft Dell support bevestigd dat SNMP SET ondersteuning is verwijderd uit SonicOS. Vervolgens is de vraag gesteld of er meer documentatie beschikbaar is voor het “cli rest-api enable” commando en de CGI bestanden en of de routers via deze weg gemanaged kunnen worden. Dell support antwoord hier helaas dat GMS de enige manier is om de SonicWALL's te managen. De volledige mail wisseling is te lezen in de bijlage Requirements en probleem domein analyse document, hoofdstuk 5.

Hieruit wordt de conclusie getrokken dat de SonicWALL's niet te combineren zijn met dit project. De MikroTik's ondersteunen dezelfde protocollen voor management als de SonicWALL's afgezien van het GMS en heeft als extra de API. Voor de MikroTik's zal er onderzoek worden gedaan naar de API en zullen dezelfde stappen als bij de SonicWALL's worden ondernomen voor SNMP.

Voor de MikroTik's is er allereerst aandacht besteed aan de API. De API is te gebruiken met veel verschillende programmeer talen. De documentatie van MikroTik bevat voorbeelden om connecties te maken tussen het programma en de router. Er is gekozen om het volgende voorbeeld te gebruiken om mee te experimenteren “API PHP class” (“API PHP class”, z. j.). Er is geëxperimenteerd met de RouterBOARD 750. In figuur 8 is te zien hoe de systeem identiteit zonder problemen wordt uitgelezen.

```
Connection attempt #1 to 192.168.88.1:8728...
<<< [6] /login
>>> [5/5] bytes read.
>>> [5, 39]!done
>>> [37/37] bytes read.
>>> [37, 1]=ret=396bfaa7b34fce8d4e0fc3072e5f26d4
<<< [6] /login
<<< [11] =name=admin
<<< [44] =response=00553e566125f9648c1446d35929ca6ad1
>>> [5/5] bytes read.
>>> [5, 1]!done
Connected...
<<< [20] /system/identity/get
>>> [5/5] bytes read.
>>> [5, 20]!done
>>> [18/18] bytes read.
>>> [18, 1]=ret=name=MikroTik
nameDisconnected...
Disconnected...
```

Figuur 8: RouterOS API test om de identiteit uit te lezen.

Behalve het uitlezen is er ook geprobeerd de systeemnaam te wijzigen via dezelfde API. Door middel van het commando “/system/identity/set” met de parameters “name=MikroTik test” is de systeem naam gewijzigd. Dit is te zien in figuur 9.

```

Connection attempt #1 to 192.168.88.1:8728...
<<<< [6] /login
>>>> [5/5] bytes read.
>>>> [5, 39]!done
>>>> [37/37] bytes read.
>>>> [37, 1]=ret=f6da3a53c0be225038b930c3787214d0
<<<< [6] /login
<<<< [11] =name=admin
<<<< [44] =response=008bb2448910ead78dd878d20ffc9cab54
>>>> [5/5] bytes read.
>>>> [5, 1]!done
Connected...
<<<< [20] /system/identity/set
<<<< [19] =name=MikroTik test
>>>> [5/5] bytes read.
>>>> [5, 1]!done
Disconnected...

```

Figuur 9: RouterOS API test om de identiteit te wijzigen.

Om er zeker van te zijn dat dit niet de enige instelling is die gewijzigd kan worden is er met meerdere instellingen getest, zoals de MTU van een interface.

Ook SNMP wordt getest op de router. SNMP versie 1 en 2c worden overgeslagen omdat SNMPv3 ook wordt ondersteund en minder beveiligingsrisico's met zich mee brengt. Voor dit experiment worden dezelfde stappen ondernomen als bij de SonicWALL's.

De MikroTik wordt geconfigureerd voor SNMPv3 volgens de documentatie. Middels de MIB browser wordt weer gebruikt om alle OID's in te zien. Zonder problemen kunnen alle OID's uitgelezen worden. De documentatie beschrijft dat er maar drie OID's zijn waarvan de waarde gewijzigd kan worden met SNMP:

- De systeem identiteit: 1.3.6.1.2.1.1.5.0
- Herstarten van de router: 1.3.6.1.4.1.14988.1.1.7.1.0
- Het uitvoeren van een script: 1.3.6.1.4.1.14988.1.1.8.1.1.3.X (Waar X het nummer van het script is)

Het eerste experiment is gericht op het wijzigen van de systeem identiteit via de MIB browser. Dit gaat goed, zoals te zien is in figuur 10. De identiteit wordt uitgelezen, gewijzigd met een SET commando en vervolgens weer uitgelezen. Aan de rechterkant in de afbeelding is te zien hoe de systeem naam wijzigt.

Sent GET request to 192.168.88.1: 161	
sysName.0	MikroTik
Sent SET request to 192.168.88.1: 161	
sysName.0	MikroTik Test
Sent GET request to 192.168.88.1: 161	
sysName.0	MikroTik Test

Figuur 10: Het uitlezen en wijzigen van de MikroTik identiteit via de MIB browser.

Vervolgens wordt er een willekeurige instelling gebruikt, die niet in het lijstje voorkomt van wijzigbare OID's, om daar mee te testen. Er wordt geprobeerd de MTU te wijzigen van interface 3. De standaard waarde is 1500 en deze moet veranderd worden naar 1400. In figuur 11 is te zien dat het SNMP SET commando de waarde niet manipuleert. Er wordt ook geen foutmelding terug gegeven door de router. Dat doet vermoeden dat het goed gaat, maar dat gaat het dus niet. De reden is ook niet duidelijk door het ontbreken van een foutmelding.

```

Sent GET request to 192.168.88.1 : 161
ifMtu.3                                     1500
Sent SET request to 192.168.88.1 : 161
ifMtu.3                                     1400
Sent GET request to 192.168.88.1 : 161
ifMtu.3                                     1500

```

Figuur 11: Het uitlezen en wijzigen van de MTU van interface 3 op de MikroTik.

Ook bij MikroTik wordt geverifieerd of dit niet alleen aan één kant ligt. Dus ook hiervoor wordt een PHP script geschreven. Dit voert dezelfde stappen uit als dat via de MIB browser is gedaan, helaas is ook deze manier zonder resultaat. In figuur 12 is wel te zien dat het SNMP commando een “1” retourneert.

```

GET ".iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu.3": 1500
SET ".iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu.3" naar 1400: 1
GET ".iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu.3": 1500

```

Figuur 12: Uitvoer van PHP script.

De “1” die als antwoord wordt gegeven op het SET commando, wat gelijk staat aan “TRUE” in de logica, duid er op dat het commando succesvol is afgerond (“SNMP::set”, z. j.). In de afbeelding is te zien dat de tweede keer uitlezen nog steeds “1500” terug geeft. Dit bewijst dat de instelling niet te manipuleren is met het SNMP SET commando. Omdat er geen foutmeldingen worden terug gegeven in beide situaties wordt er om hulp gevraagd op het forum van MikroTik¹ en op Reddit, zonder resultaat.

In de MikroTik's kunnen zelf gemaakte scripts toegevoegd worden. Een script is een door een beheerder samengestelde set aan commando's die uitgevoerd kunnen worden op de router. Deze kunnen geactiveerd worden via het SNMP OID genoemd in het lijstje voorkomt van wijzigbare OID's. Het gebruik van deze scripts voor het eindproduct levert meerdere problemen op. Er kunnen geen parameters meegegeven worden aan de scripts, dus dat betekent dat er niets variabel is aan het script en er voor elke mogelijke waarde voor elke instelling een script gemaakt moet worden. Voor instellingen zoals bijvoorbeeld de naam van de router is dit al helemaal niet mogelijk, omdat die allerlei waarden kunnen bevatten.

5.1.4 Conclusie

De conclusie uit deze analyse is dat de SonicWALL's helaas niet te managen zijn op afstand vanuit een zelf ontwikkelde applicatie. De MikroTik's daarentegen wel. De opdrachtgever is akkoord gegaan met het laten vervallen van de SonicWALL's uit dit project. Aan de opdrachtgever is geadviseerd om het GMS te gebruiken voor de SonicWALL's, omdat dit de enige optie is. Er is geen onderzoek gedaan naar het GMS en de mogelijkheden hiervan omdat dit buiten de scope van het project valt.

Voor de MikroTik's gaat de API gebruikt worden, dit is de enige manier om de routers vanaf afstand te beheren. De MikroTik's moeten allemaal minimaal RouterOS versie 6.1 hebben, pas vanaf deze versie wordt SSL ondersteund voor de API. Dit is uit veiligheidsoogpunt geadviseerd aan de stakeholders.

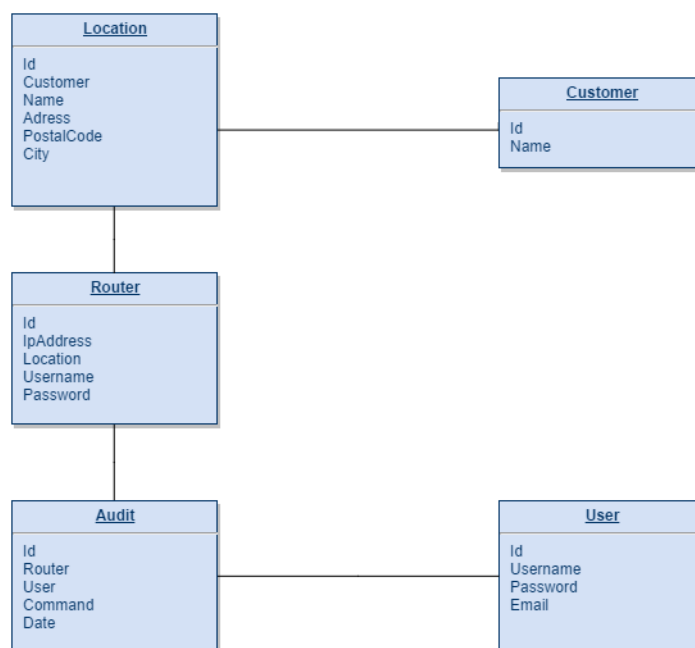
¹<http://forum.mikrotik.com/>

5.2 Iteratie 2: Architectuur document

In het architectuur document wordt de basis gelegd voor de applicatie en het ontwerp. Er wordt gekozen om de applicatie in PHP te programmeren. PHP is de beste optie om te voldoen aan de systeem eisen en binnen de geplande tijd van het project te blijven. Binnen het project en BlackGATE is er veel kennis van PHP. Bij BlackGATE is er ook kennis van Ruby, maar daar is geen kennis van binnen het project en dus geen wijze optie als programmeertaal. Er is binnen het tijdsvak van dit project geen tijd een nieuwe programmeertaal te leren. PHP is een syntactisch simpele en “Object Oriented” programmeertaal en is helemaal gericht op web applicaties. Daardoor werkt het heel goed samen met javascript, html en css. Ook het “deployment proces” is gemakkelijk bij PHP. Tevens is er al geëxperimenteerd met de RouterOS API in combinatie met PHP. Deze experimenten hebben aangetoond dat PHP werkt in combinatie met de API. Dit zijn de redenen waarom er voor PHP gekozen is.

Voor het ontwikkelen van de applicatie is gekozen een framework te gebruiken. Een framework legt een solide basis voor de applicatie en maakt het ontwikkelen makkelijker. Er is binnen het project al veel ervaring met het PHP framework CodeIgniter². CodeIgniter is tevens een uitgebreid en uitbreidbaar framework en erg flexibel in het ontwikkelen van een applicatie. Dit framework is verkozen boven Laravel, omdat er met beide frameworks geëxperimenteerd is, maar Laravel niet voldoet aan de eisen. Laravel is ook een compleet nieuw framework en er is geen kennis over beschikbaar binnen het project of BlackGATE. Laravel stelt tevens veel eisen aan het ontwikkelen, zoals benodigde componenten en het configureren van bepaalde onderdelen. Dat is een groot minpunt. CodeIgniter is gemakkelijk in gebruik te nemen.

De systeemeisen zijn vervolgens nogmaals geformuleerd zoals opgesteld in iteratie 1 van de elaboration fase. Op basis van de huidige eisen wordt een domein model gemaakt. Dit domein model bevat nog niet alle klassen die de applicatie moet gaan bevatten. Het eerste domein model, te zien in figuur 13, is de eerste versie. Op dit moment bevat het alleen de klassen die in de database opgeslagen worden. Er is gekozen om een database te gebruiken om de gegevens op te slaan die behouden moeten blijven, zoals de gebruikersgegevens en de verbindingsvariabelen voor de routers. Hiervoor zal MySQL MariaDB gebruikt worden. Er is al veel ervaring met MySQL binnen dit project en ook binnen BlackGATE. Bij BlackGATE draaien al meerdere applicaties met een MySQL database, dit maakt het daarom ook een gemakkelijkere keuze. Tevens is MySQL een relationele database wat de koppeling tussen bepaalde klassen gemakkelijk maakt, wat erg handig is in dit project. Als laatste zijn de risico's nogmaals beschreven.

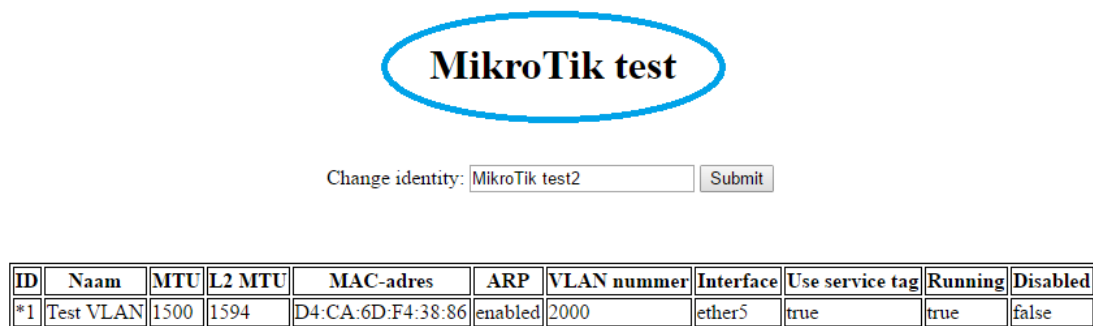


Figuur 13: Domein model versie 1

²<http://www.codeigniter.com/>

De klassen in het domein model voldoen aan de eisen van de opdrachtgever. In de te ontwikkelen applicatie moeten routers bestaan, deze moeten gecategoriseerd worden per locatie per klant. Daarnaast is er een gebruikerssysteem en een audit log. In het audit log worden wijzigingen bijgehouden die een gebruiker heeft gemaakt op een router. De uitgebreide systeemeisen zijn te vinden in bijlage Architectuur Document, hoofdstuk 3.

Er is een prototype gebouwd van de applicatie met de basis functionaliteiten er in. Het prototype kan de VLAN's van de router uitlezen en de identiteit van de router ophalen en wijzigen. In figuur 14 is de GUI van het prototype te zien.



Change identity:

ID	Naam	MTU	L2 MTU	MAC-adres	ARP	VLAN nummer	Interface	Use service tag	Running	Disabled
*1	Test VLAN	1500	1594	D4:CA:6D:F4:38:86	enabled	2000	ether5	true	true	false

Figuur 14: Prototype 1

In de bovenstaande afbeelding staat dikgedrukt de identiteit van de router met blauw omcirkeld, “MikroTik test”. Vervolgens kan in het tekst vak er onder de nieuwe naam ingevoerd worden, de wijziging kan doorgevoerd worden door op “Submit” te drukken. In dit geval wordt de naam dan gewijzigd naar “MikroTik test2”. Onderaan staan in een tabel de VLAN's van de router weergegeven. Er is één VLAN aanwezig in de router op dit moment.

Zodra er op de knop “Submit” wordt gedrukt wordt de identiteit gewijzigd van de router. In figuur 15 is het resultaat daarvan te zien. De identiteit, met rood omcirkeld, is gewijzigd naar “MikroTik test2”.



Change identity:

ID	Naam	MTU	L2 MTU	MAC-adres	ARP	VLAN nummer	Interface	Use service tag	Running	Disabled
*1	Test VLAN	1500	1594	D4:CA:6D:F4:38:86	enabled	2000	ether5	true	true	false

Figuur 15: Prototype 1 identiteit gewijzigd

5.3 Iteratie 3: Architectuur document

Na de demonstratie van het prototype aan de opdrachtgever wordt er nogmaals geïtereerd over het architectuur document en het prototype. In deze iteratie zijn de systeemeisen nog wat meer aangepast om volledig te voldoen aan de eisen van de opdrachtgever binnen de haalbare grenzen van het project. Het domein model is ook verder uitgewerkt. Er zijn nu ook klassen toegevoegd die niet in de database komen te staan. De overweging die hiervoor gemaakt is, is dat de data die uit de router gehaald wordt in een object gestopt kan worden. Hierdoor is de data makkelijker te benaderen en mee te werken. Het domein model staat in bijlage Architectuur document, hoofdstuk 4.

Op deze veranderingen en prototype 1 is prototype 2 gebaseerd. Prototype 2 bevat meer functies.

asd

ID	Name	Default name	Type	MTU	Actual MTU	L2 MTU	Max L2 MTU	MAC address	Fast path	Last Link Down	Last Link Up	Link down	Rx bytes	Tx bytes	Rx packets	Tx packets	Rx drop	Tx drop	Rx error	Tx error	Running	Disabled
*1	ether1-gateway	ether1	ether	1500	1500	1600	4076	D4:CA:6D:F4:38:84	true			0	0	0	0	0	0	0	0	0	false	false
*2	ether2-master-local	ether2	ether	1500	1500	1598	2028	D4:CA:6D:F4:38:85	true	apr01/2016 09:35:06	apr01/2016 09:35:20	6	45774267	121871741	395231	119005	0	0	0	0	true	false
*3	ether3-slave-local	ether3	ether	1500	1500	1598	2028	D4:CA:6D:F4:38:86	true			0	0	0	0	0	0	0	0	0	false	false
*4	ether4-slave-local	ether4	ether	1500	1500	1598	2028	D4:CA:6D:F4:38:87	true			0	0	0	0	0	0	0	0	0	false	false
*5	ether5-slave-local	ether5	ether	1500	1500	1598	2028	D4:CA:6D:F4:38:88	true			0	0	0	0	0	0	0	0	0	false	false

ID	Name	Type	MTU	Actual MTU	L2 MTU	MAC address	Fast path	Link down	Rx bytes	Tx bytes	Rx packets	Tx packets	Rx drop	Tx drop	Rx error	Tx error	Running	Disabled
----	------	------	-----	------------	--------	-------------	-----------	-----------	----------	----------	------------	------------	---------	---------	----------	----------	---------	----------

[VLAN toe voegen](#)

DHCP

DHCP pool: 192.168.88.0/24
Gateway: 192.168.88.1 DNS server: 192.168.88.1
Lease count: 0

ARP table

ID	Address	MAC address	Interface	Published	Invalid	DHCP	Dynamic	Complete	Disabled
*2	192.168.88.254	0C:EB:E8:2A:93:32	ether2-master-local	false	false	false	true	true	false

Figuur 16: Prototype 2

In het bovenstaande figuur, figuur 16, is het overzichtsscherm te zien. Boven aan de pagina staat de identiteit van de router, “asd”. Vervolgens is er een tabel met alle ethernet interfaces van de router te zien. Hier staan verschillende instellingen maar ook monitoring gegevens in, zoals de hoeveelheid pakketten er zijn verzonden of ontvangen. De tabel daaronder is voor de VLAN’s. De router heeft op dit moment geen VLAN geconfigureerd en er is dus niets weer te geven. Onder aan de tabel is een link te zien waarop geklikt kan worden om een VLAN toe te voegen. Dit komt later aan bod. Verder worden in het hoofdscherm nog de DHCP gegevens weergegeven en de ARP tabel.

Name

MTU

ARP

VLAN ID

Interface:

Figuur 17: Prototype 2 VLAN toevoegen

Figuur 17 geeft het scherm weer waar een VLAN toegevoegd kan worden. Alle gegevens voor het VLAN kunnen daar ingevuld worden en er kan aangegeven worden aan welke interface het VLAN gekoppeld moet worden

In de onderstaande afbeelding, figuur 18 is het hoofdscherm nogmaals te zien, nu inclusief het toegevoegde VLAN. Aan de rechterkant van de tabel staat op de VLAN rij een link “Remove”. Door op deze link te klikken kan het VLAN weer verwijderd worden.

asd

ID	Name	Default name	Type	MTU	Actual MTU	L2 MTU	Max L2 MTU	MAC address	Fast path	Last Link Down	Last Link Up	Link downs	Rx bytes	Tx bytes	Rx packets	Tx packets	Rx drop	Tx drop	Rx error	Tx error	Running	Disabled
#1	ether1-gateway	ether1	ether	1500	1500	1600	4076	D4-CA:6D-F4:38:84	true			0	0	0	0	0	0	0	0	0	false	false
#2	ether2-master-local	ether2	ether	1500	1500	1598	2028	D4-CA:6D-F4:38:85	true	apr/01/2016 09:35:07	apr/01/2016 09:35:21	6	45809708	121886895	395546	119062	0	0	0	0	true	false
#3	ether3-slave-local	ether3	ether	1500	1500	1598	2028	D4-CA:6D-F4:38:86	true			0	0	0	0	0	0	0	0	0	false	false
#4	ether4-slave-local	ether4	ether	1500	1500	1598	2028	D4-CA:6D-F4:38:87	true			0	0	0	0	0	0	0	0	0	false	false
#5	ether5-slave-local	ether5	ether	1500	1500	1598	2028	D4-CA:6D-F4:38:88	true			0	0	0	0	0	0	0	0	0	false	false

ID	Name	Type	MTU	Actual MTU	L2 MTU	MAC address	Fast path	Link downs	Rx bytes	Tx bytes	Rx packets	Tx packets	Rx drop	Tx drop	Rx error	Tx error	Running	Disabled	
#A	Test	vlan	1500	1500	1594	D4-CA:6D-F4:38:86	true	0	0	0	0	0	0	0	0	0	false	false	Remove

VLAN toe voegen

DHCP

DHCP pool: 192.168.88.0/24
Gateway: 192.168.88.1 DNS server: 192.168.88.1

Lease count: 0

DHCP

DHCP pool: 192.168.88.0/24
 Gateway: 192.168.88.1 DNS server: 192.168.88.1
 Lease count: 0

ARP table

ID	Address	MAC address	Interface	Published	Invalid	DHCP	Dynamic	Complete	Disabled
#2	192.168.88.254	0C-EB-E8-2A-93-32	ether2-master-local	false	false	false	true	true	false

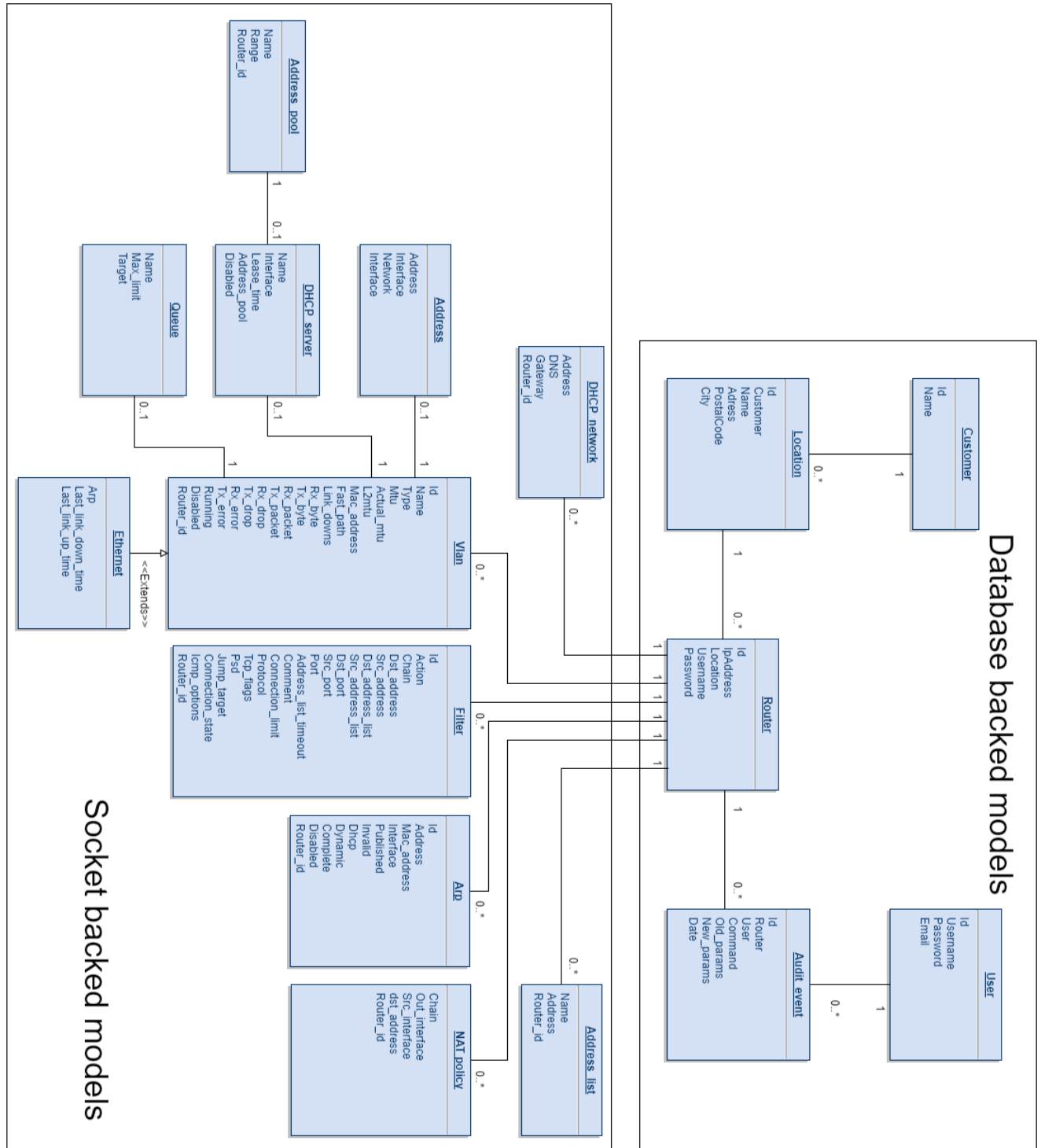
Figuur 18: Prototype 2 VLAN toegevoegd

5.4 Iteratie 4: Ontwerp document

In deze iteratie wordt gewerkt aan het ontwerp document. Dit omvat de definitieve systeemeisen, use cases, mockups en het klassendiagram. De systeemeisen zijn verfijnd om beter te verwoorden wat er precies bedoeld wordt. Voor de systeemeisen en het use case diagram is veelal gekozen voor het woord “beheren”. Bijvoorbeeld, het beheren van de VLAN’s op een router. Met beheren worden toevoegen, wijzigen, verwijderen en bekijken bedoeld. Het use case diagram en de uitgewerkte use case beschrijvingen zijn te lezen in bijlage Ontwerp document, hoofdstuk 3.

Daarnaast zijn er mockups gemaakt voor de Graphical User Interface (GUI). De schermen om instellingen toe te voegen of te wijzigen zijn samengevat in één mockup om het document overzichtelijk te houden. De mockups zijn terug te vinden in bijlage Ontwerp document, hoofdstuk 4.

Als laatste is het definitieve klassendiagram uitgewerkt, te zien in figuur 19.



Figuur 19: Klassendiagram

Het klassendiagram is opgesplitst in twee delen. De “Socket backed” en de “Database backed” modellen. Dit is gedaan omdat de socket based modellen niet worden opgeslagen in de database maar altijd live uit de router opgehaald worden wanneer nodig. De database backed models worden wel opgeslagen in de database. Alle klassen zijn toch in het diagram opgenomen op zo goed mogelijk de structuur van de applicatie te laten zien. De meeste socket based models hebben een connectie met een database backed model, daarom kan het klassendiagram niet worden opgesplitst. Een uitgebreidere beschrijving hiervan is te lezen in bijlage Ontwerp document, hoofdstuk 5.

Er is gekozen om deze splitsing te maken omdat de “socket-backed” modellen actuele data bevatten. Deze data moet elke keer opnieuw uit de router gehaald worden om de huidige situatie van de data weer te geven. Als deze data opgeslagen wordt kan de situatie ontstaan dat het niet de informatie bevat om een probleem, dat net is ontstaan, te kunnen analyseren. De “database-backed” modellen worden wel opgeslagen omdat deze statische data bevatten of omdat dit data is die bewaard moet blijven. Zoals bijvoorbeeld de router gegevens waarmee verbonden wordt of de audit log van alle wijzigingen die gedaan zijn.

6 Construction fase

De construction fase omvat het ontwikkelen van de applicatie. In iteraties met als mijlpalen een demonstratie en een test rapport komt de applicatie tot het eindresultaat. Deze fase is, net als de elaboration fase, opgedeeld in 4 iteraties.

6.1 Iteratie 1: Bouwen op prototype

Volgens de mockups die gemaakt zijn is er te houden aan een bepaalde vormgeving van de applicatie. De ontwerpen zijn vrij generiek en er hoeft niet veel tijd besteed te worden aan de GUI. Daarom wordt er als eerste een design op internet gezocht. Hiervoor is de open source template “AdminLTE”³ gebruikt.

Dit design is gebaseerd op Bootstrap 3. Bootstrap is een framework voor het designen van websites en zorgt voor responsive content. Responsive wil zeggen dat het design zich aanpast aan de grootte van het scherm, hierdoor is het design ook makkelijk op een smartphone te bekijken. AdminLTE is ook een design dat vergelijkbaar is met de mockups. Het gebruik van het design is gratis. Dit is geen specifieke systeem eis van de opdrachtgever om het design te gebruiken, het bespaart tijd bij de ontwikkeling van de applicatie doordat het design gemakkelijker te maken is.

Daarna wordt de PHP klasse toegevoegd uit de MikroTik documentatie⁴. Dit is dezelfde die is gebruikt voor het prototype (“API PHP class”, z. j.). Zodra deze bestanden zijn toegevoegd is de basis voor de applicatie gelegd en wordt het gehele project in GitLab gezet met behulp van Git. Git (Chacon & Straub, 2014) is een versiebeheer systeem dat veel wordt gebruikt binnen de softwareontwikkeling. BlackGATE heeft een eigen account op GitLab waaraan het project toegevoegd kan worden.

Versiebeheer is, uit ervaring, een belangrijk onderdeel van software ontwikkeling. Het staat het individueel programmeren van onderdelen toe maar laat ook de historie zien van de ontwikkeling. Er kunnen zogeheten “branches” van de hoofdlijn aangemaakt worden waarop ontwikkeld kan worden zonder het werkende product aan te tasten op de hoofdlijn. Hierdoor kunnen nieuwe functies eerst uitgebreid getest worden zonder de hoofdlijn, eventuele andere programmeurs of een nieuwe release in de weg te zitten.

6.1.1 Database

CodeIgniter heeft een ingebouwd migratiesysteem om de database mee te beheren. Via dit systeem kunnen tabellen in de database aangemaakt of verwijderd worden. Door al deze migraties een nummer te geven houdt CodeIgniter zelf bij welke versie het meest recent is en welke versie er op dat moment draait. Op deze manier kunnen wijzigingen makkelijk bijgehouden worden en voor iedereen op een simpele manier doorgevoerd worden in de database. Dit houdt het consistent en makkelijk zodat niet iedereen apart een query hoeft te doen op de database om alles up to date te houden.

Alle “database-backed” modellen die in het klassendiagram staan zijn in migraties verwerkt en deze zijn doorgevoerd in de database. De tabel voor de gebruikers wijkt af van het klassen diagram omdat voor het gebruikers systeem een open source bibliotheek speciaal voor CodeIgniter wordt gebruikt. Hierover meer in hoofdstuk 6.1.2.

Verder worden alle tabellen direct overgenomen uit het klassendiagram en de juiste relaties tussen de tabellen worden aangemaakt. Er zijn in dit geval geen koppeltabellen nodig voor de relaties. Tevens worden voor deze tabellen overeenkomende klassen aangemaakt in de bron code. Dit gaat volgens de naamgeving van CodeIgniter, voor bijvoorbeeld de router tabel wordt de klasse “Router_model” aangemaakt. Deze klassen hebben niet expliciet alle properties gedefinieerd staan, deze worden automatisch gegenereerd door CodeIgniter zodra er data uit de database wordt gehaald en in het model wordt gestopt.

³<https://almsaeedstudio.com/preview>

⁴http://wiki.mikrotik.com/wiki/Main_Page

6.1.2 Gebruikers systeem

Voor het authenticatie systeem is er gekozen om een bestaande bibliotheek te gebruiken genaamd “Community Auth”. Deze keuze is gemaakt vanwege de navolgende overwegingen. Om zelf een gebruikers systeem te programmeren en te implementeren is er veel om rekening mee te houden. Zowel de juiste authenticatie als het goed afschermen van de beveiligde pagina’s. Om te voorkomen dat er te veel tijd gaat zitten in het schrijven en het uitgebreid testen van het gebruikers systeem alleen al, is er voor gekozen om “Community Auth” te gebruiken. Bij het zelf schrijven moet er rekening gehouden worden met alle bijkomende aspecten, zoals bijvoorbeeld sessie validatie of het laten zien van de inlog pagina zodra er een beveiligde pagina wordt bereikt door een niet geautoriseerde gebruiker.

“Community auth” bestaat al enkele jaren voor CodeIgniter en is sinds een jaar ook voor CodeIgniter versie 3 beschikbaar. Het wordt op het moment van schrijven nog actief onderhouden, de laatste wijziging is 24 april 2016.

Er zijn meerdere van dit soort bibliotheken beschikbaar zoals bijvoorbeeld “Ion auth”. Dit systeem is echter te uitgebreid en lastiger te integreren dan “Community auth”. “Community auth” voldoet aan de eisen daarom is, in combinatie met de bovengenoemde redenen, de keuze op dit systeem gevallen.

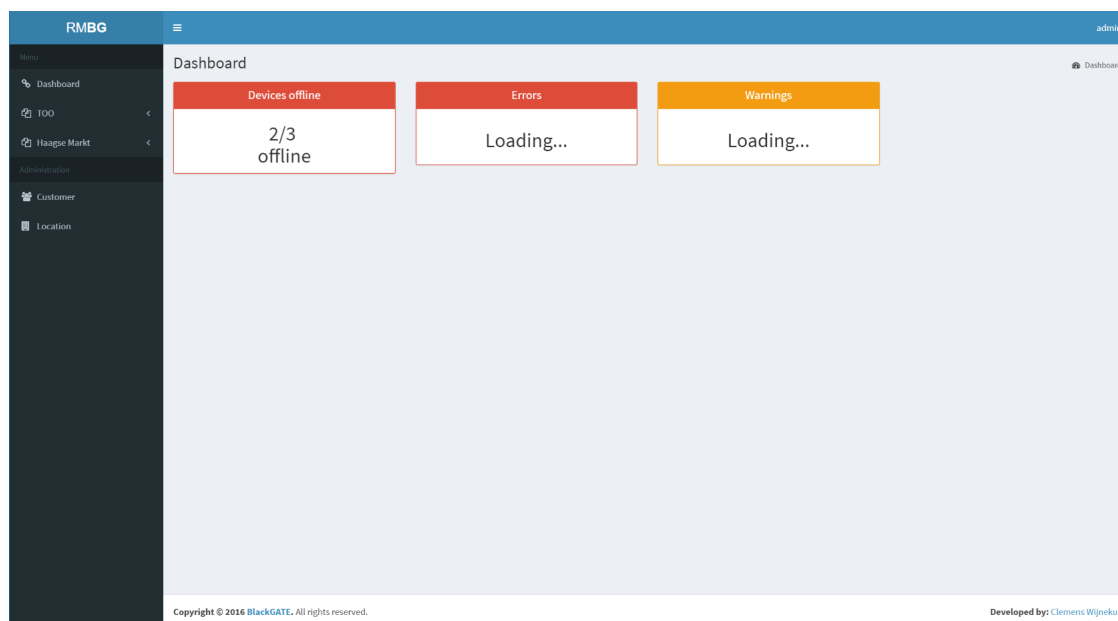
Het gebruikers systeem is makkelijk te integreren. Het werkt uitstekend en alles wordt netjes afgeschermd wat afgeschermd moet worden. De functie om de gebruiker te onthouden als deze de pagina weer opnieuw opent nadat deze gesloten is, is uitgeschakeld, uit veiligheidsoogpunt. Zolang de gebruiker actief blijft, verloopt de sessie niet. Maar mocht de gebruiker inactief worden dan zal deze op nieuw in moeten loggen. Dit in verband met bijvoorbeeld diefstal van hardware.

6.1.3 Klanten en locaties

De opdrachtgever heeft gespecificeerd de routers per locatie per klant georganiseerd te willen hebben. Om dit mogelijk te maken moeten klanten en locaties toegevoegd kunnen worden. Er worden voor zowel de klanten als locaties functies aangemaakt om deze te kunnen toevoegen, bewerken, inzien en verwijderen. Dit worden CRUD (Create, Read, Update, Delete) functies genoemd. Er moet namelijk een locatie bestaan voordat er een router aan toegevoegd kan worden. En voordat er locaties toegevoegd kunnen worden, moeten er klanten toegevoegd kunnen worden. Zodra de klanten en locaties kunnen worden toegevoegd in de applicatie wordt ook de router klasse geïmplementeerd.

6.1.4 Connectie router via zelfgeschreven API

Zodra er routers toegevoegd kunnen worden aan de database worden er de basis functies van het prototype in geïntegreerd. Op de hoofdpagina wordt weergegeven hoeveel van het totale aantal routers online zijn, het dashboard is te zien in figuur 20.

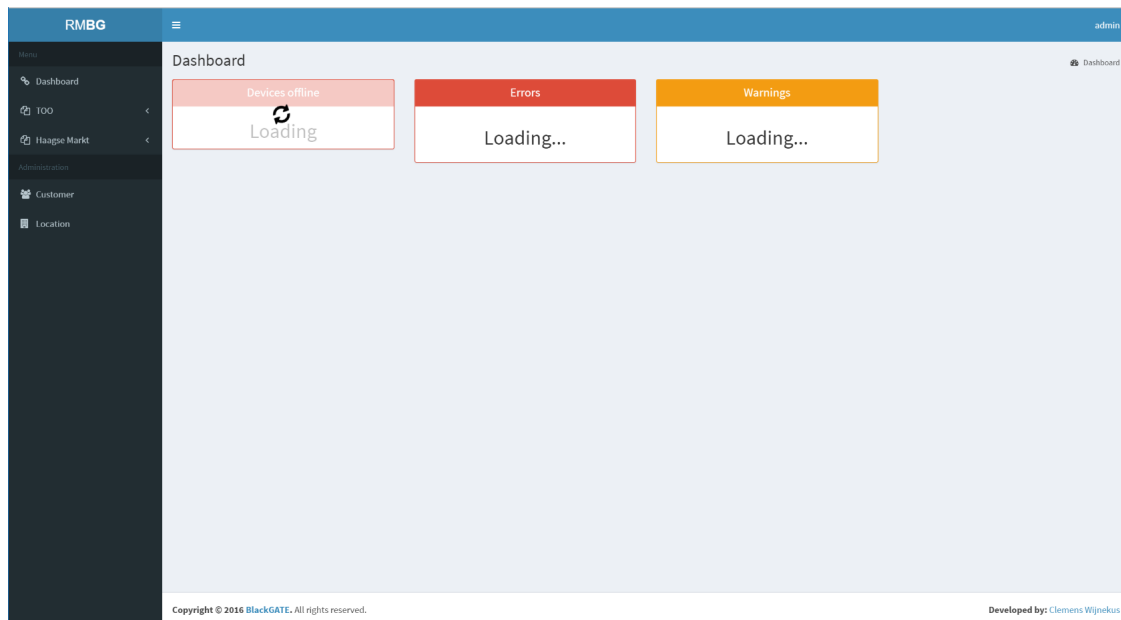


Figuur 20: Monitoring dashboard

Via de MikroTik connector klasse wordt er een verbindingspoging gedaan met elke router die voorkomt in de database. Hier vormt zich een probleem. Er zijn twee willekeurige niet bestaande routers en de bestaande router uit de test opstelling, waar dus verbinding mee is, toegevoegd aan de database. Zodra de hoofdpagina geladen wordt, probeert de applicatie of deze routers bereikbaar zijn. Twee van de routers hebben incorrecte gegevens, hiermee kan dus geen connectie gemaakt worden. De applicatie probeert echter na het mislukken van de verbindingspoging nog twee maal om te verbinden. Elk met 2 seconden pauze tussen de pogingen. Hierdoor duurt het erg lang voordat de pagina geladen wordt. Dit wordt een nog groter probleem als er meer routers worden toegevoegd en er meerdere daarvan niet bereikbaar zijn. Bij experimenteren is bovendien gebleken dat de webserver na dertig seconden van uitvoeren een foutmelding geeft dat de maximum tijd voor het uitvoeren van het script overschreden is en wordt het verzoek afgekapd.

Een oplossing hiervoor is het verhogen van de maximale executie tijd voor PHP. Deze kan zo hoog gezet worden als vereist is. Dit is geen aanrader omdat er met een hele hoge limiet heel veel scripts tegelijk kunnen blijven draaien en dat kan zorgen voor geheugen problemen op de webserver.

Om de gebruiker niet te hinderen tijdens het ophalen van de statistieken, is er een experiment uitgevoerd met AJAX (asynchronous JavaScript and XML). Asynchrone processen kunnen los van het hoofdproces van de applicatie verzoeken uitvoeren zonder het hoofdproces te onderbreken. Hierdoor kan de pagina geladen worden en wordt de data apart opgehaald en pas later op de pagina weergegeven, zoals te zien in figuur 21. Dit houdt de applicatie snel en gebruiksvriendelijk.



Figuur 21: Asynchroon ophalen van data

Met het asynchroon ophalen of wegschrijven van data ontstaat het idee om zelf een API te ontwikkelen in de applicatie. Via deze API zou data opgevraagd of weggeschreven kunnen worden van een of meerdere routers. Door middel van experimenten met een zelf ontwikkelde API en asynchrone verzoeken wordt er gekeken naar de mogelijkheden hiervan. De conclusie hieruit is dat het ophalen van monitoring statistieken via AJAX een geschikte oplossing is, daarentegen is het in de huidige situatie niet geschikt voor het implementeren van AJAX verzoeken. De basis van de applicatie werkt op PHP en CodeIgniter, om hier AJAX in te integreren is lastig. Normaal worden alle invoervelden voor instellingen geverifieerd en daarna verder behandeld door CodeIgniter. Met AJAX verzoeken gaat dit lastig omdat er niet vanuit het PHP script zomaar Javascript aangeroepen kan worden. Tevens is het uitwisselen van variabelen en de waarden daarvan lastig tussen PHP en Javascript. Er is bij het ontwerpen geen rekening gehouden met asynchrone verzoeken. Om de hele applicatie op dit moment nog om te bouwen, zodat alles met Javascript afgehandeld kan worden, past niet meer in het tijdsbestek dit project. Dit zou een eventuele verbetering zijn voor een volgende versie.

Daarnaast wordt er nog een probleem geconstateerd met de asynchrone verzoeken. Zodra er een verzoek wordt gedaan naar de API en de API is data aan het verwerken loopt de applicatie vast. Zolang de API bezig is kan er geen andere pagina geladen worden. Het probleem lijkt alsof de webserver niet meerdere verzoeken tegelijk af kan handelen. Er is daarom getest door met meerdere gebruikers de applicatie te benaderen, dit gaat zonder problemen. Vervolgens is er diepgaander onderzoek gedaan naar de oorzaak van dit probleem. De eerste oplossing die zich aandient is het aanmaken van meerdere PHP processen. Dit kan gedaan worden door verschillende modules te installeren op de webserver en instellingen aan te passen. Dit lijkt echter een omslachtig en moeilijk proces.

Na verder onderzoek dient zich een andere mogelijke oplossing aan, de sessie van PHP sluiten. Een PHP sessie is de sessie die de gebruiker heeft met de webserver waarin de applicatie meerdere sessie variabelen in opslaat. Dit kan bijvoorbeeld de sessie identiteit bevatten, waaraan de applicatie kan herkennen dat de gebruiker is ingelogd. CodeIgniter start automatisch de sessie op zodra er een PHP script wordt aangeroepen. Dit is nodig voor het gebruikers systeem en zorgt er voor dat niet in elk PHP bestand de sessie geïnitieerd wordt.

Doordat de sessie wordt opgestart wordt door dat bepaalde PHP script het sessiebestand voor schrijven vergrendeld, zodat alleen dat PHP script er wijzigingen op kan uitvoeren. Dit is een ingebouwde veiligheidsmaatregel van CodeIgniter. Omdat het PHP script van de API het sessie bestand vergrendelt, kan een ander PHP script niet bij het sessie bestand. Hierdoor wacht het andere PHP script totdat het sessiebestand weer vrij is en blijft dus laden totdat het eerste PHP script klaar is.

CodeIgniter biedt de mogelijkheid om sessie bij te houden in de database, in plaats van het sessiebestand. Dit zou het vergrendel probleem op moeten lossen. Met de juiste configuratie worden de sessies opgeslagen in de database. Echter lost dit het probleem niet op. Uit de documentatie van CodeIgniter blijkt dat de sessie via een PHP functie gesloten moet worden. Door middel van deze functie wordt de schrijfvergrendeling opgeheven maar kan er nog wel uit het sessie bestand gelezen worden. Hiermee wordt het probleem opgelost en kunnen meerdere verzoeken tegelijk afgehandeld worden.

Deze experimenten duren te lang voor het tijdsbestek van het project. Mede daarom is besloten geen API te maken maar zowel front als backend in één applicatie te houden. Omdat er geen rekening is gehouden met asynchrone verzoeken en Javascript in de ontwerp fase, zal bijna de gehele applicatie herschreven moeten worden om dit wel te kunnen ondersteunen omdat alles nu op PHP draait. Dit zal veel tijd kosten en is op dit punt in het project niet meer mogelijk te implementeren binnen de tijdsperiode voor dit project.

6.2 Iteratie 2: Uitbreiden applicatie

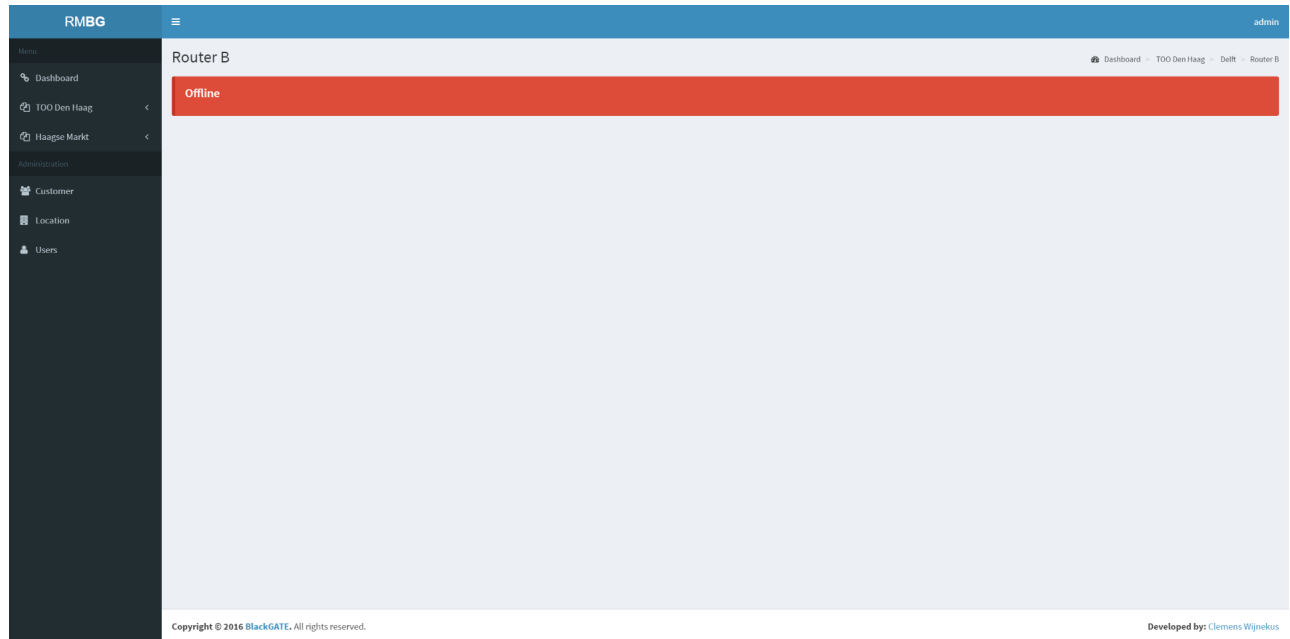
6.2.1 Connectie router via connection wrapper

Omdat de API geen optie is wordt een wrapper geschreven voor de PHP klasse die gebruikt wordt voor de connectie. Een wrapper is een klasse die een schil vormt om een andere klasse om het gebruik makkelijker te maken. Zonder deze wrapper ontstaat er onnodig veel dubbele code. Zo zou in elke functie van controllers die te maken hebben met instellingen van een router de volgende stappen uitgevoerd moeten worden:

- Ophalen router gegevens uit database
- Verbinding maken met router via connector klasse
- Uitvoeren commando op router
- Uitlezen van resultaat en waar nodig in een model stoppen

In meerdere functies moet bijvoorbeeld alle VLAN data uit de router gehaald worden. Hiervoor is er een wrapper geschreven, deze bevat bijvoorbeeld een functie “getAllVlans”, die alle bovengenoemde stappen omvat. Op deze manier is de functie om alle VLAN's op te halen herbruikbaar op verschillende plekken in de bron code. Op meerdere plekken is dit ook nodig, bijvoorbeeld bij het laten zien van alle interfaces maar ook bij het toevoegen van een VLAN.

Voordat het dashboard van een router geladen wordt, wordt er eerst geprobeerd verbinding te maken met de router. Indien de router onbereikbaar is, hoeven er ook geen pogingen gedaan te worden om data op te halen. Dit kost onnodig veel tijd omdat de applicatie opnieuw gaat proberen te verbinden voor de verzoeken om data. Als de router niet te bereiken is, wordt op het dashboard weergegeven dat de router offline is en uiteraard geen data getoond. Dit is te zien in figuur 22



Figuur 22: Dashboard van offline router

In deze iteratie wordt er gewerkt aan de functies voor VLAN's, ethernet interfaces, DHCP en NAT instellingen. De lijst van systeemeisen wordt van boven naar beneden afgewerkt. Voor elk van de bovengenoemde elementen zijn de basis CRUD operaties gemaakt zodat deze beheerd kunnen worden.

Om te bepalen welke configuratie opties ingesteld moeten kunnen worden, is een voorbeeld genomen aan de initiële configuratie die standaard ingeladen wordt voordat de router in gebruik wordt genomen. Enkele voorbeelden uit deze configuratie zijn:

```
/interface vlan
    add interface=ether5 l2mtu=1594 name=vlan1 vlan-id=1
/ip dhcp-server
    add address-pool=dhcp disabled=no interface=ether2
    lease-time=3d name=dhcp1
/ip firewall nat
    add chain=srcnat out-interface=pppoe-out1
    src-address=185.41.37.52/25
```

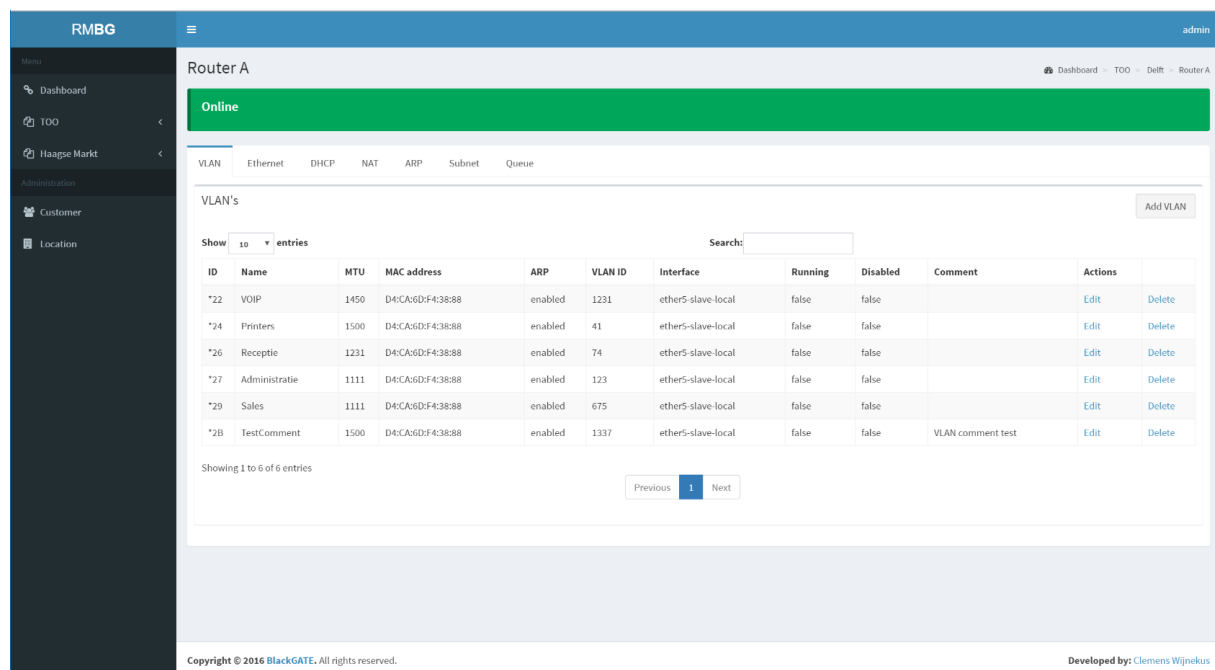
Uit deze configuratie regels valt op te maken dat voor bijvoorbeeld een VLAN de opties “interface”, “l2mtu”, “name” en “VLAN ID” geconfigureerd moeten worden. Deze zijn daarom meegenomen in het maken van de functionaliteiten.

Het vergt wat meer aandacht om de commando's op de juiste manier naar de router te sturen. Voor het zoeken op basis van parameters in een router moet voor de eerste parameter een vraagteken toegevoegd worden. Dit is slecht gedocumenteerd en vereist wat meer experimenten dan verwacht. Tevens is de foutafhandeling belangrijk. Als een commando niet correct is of niet uitgevoerd kan worden geeft de router geen fout code terug maar een “!trap”. Deze “!trap” bevat een bericht met de fout die opgetreden is. Dit moet op de juiste manier afgehandeld worden en de foutmelding moet aan de gebruiker worden getoond. Nadat de CRUD functies voor een onderdeel zijn gemaakt, is het duidelijker hoe de commando's syntactisch in elkaar zitten. Dit maakt het gemakkelijker om voor de volgende onderdelen functionaliteiten toe te voegen.

Er is voor de “socket-backed models” besloten af te wijken van het ontwerp. De eerdere ontwerpen uit het ontwerp document zijn te optimistisch en reflecteren niet goed de mogelijkheden. Dit is gebleken uit het feit dat de output van de command line van de MikroTik, waar de originele ontwerpen op zijn gebaseerd, anders is dan de output van de API. Er wordt vanuit gegaan, refererend aan de documentatie die beschrijft dat de API heel dicht op de command line zit, dat de API dezelfde output geeft als de command line bij dezelfde commando's. De modellen wijken niet significant af maar hebben niet altijd alle properties die in het klassendiagram gedefinieerd staan. Via PHP kunnen properties namelijk dynamisch worden gegenereerd. Op basis van een lijst van namen, kunnen properties aangemaakt worden. Dit is tevens de manier waarop CodeIgniter database tabellen omzet in modellen.

Het opstellen van de juiste API verzoeken en de interpretatie van de output doet de planning van deze iteratie met een week uitlopen. Er is nog niet duidelijk op dit moment waar de inhaalslag voor de planning gaat plaatsvinden. Deze kan in iteratie 3 of 4 van de construction fase plaatsvinden.

Er is bij het dashboard van een router afgeweken van de originele mockup, te vinden in bijlage Ontwerp Document hoofdstuk 4.9, omdat dit een overzichtelijker beeld geeft. De gebruikte template biedt namelijk de mogelijkheid om tabbladen te gebruiken. Het router dashboard zit er nu als volgt uit zoals in figuur 23. Tevens biedt de template een module om tabellen met sorteer en zoek functies te maken. Het zoekveld is echter gericht op elke kolom, dit kan soms verwarring opleveren. Als er bijvoorbeeld wordt gezocht op een MAC adres beginnend met D4 en er bestaat een VLAN met het MAC adres beginnend met D4 maar ook een VLAN met D4 in de naam, dan worden deze beiden getoond.



The screenshot shows a web interface for a router management application. The top navigation bar includes 'RMBG' and 'admin'. A sidebar on the left lists menu items: 'Dashboard', 'TOO', 'Haagse Markt', 'Administration', 'Customer', and 'Location'. The main content area is titled 'Router A' and shows a status 'Online'. Below this, there are tabs for 'VLAN', 'Ethernet', 'DHCP', 'NAT', 'ARP', 'Subnet', and 'Queue'. The 'VLAN' tab is active, displaying a table of VLANs. The table has columns for ID, Name, MTU, MAC address, ARP, VLAN ID, Interface, Running, Disabled, Comment, and Actions. There are 6 entries shown, with a search bar and pagination controls at the bottom of the table.

ID	Name	MTU	MAC address	ARP	VLAN ID	Interface	Running	Disabled	Comment	Actions
*22	VOIP	1450	D4:CA:6D:F4:38:88	enabled	1231	ether5-slave-local	false	false		Edit Delete
*24	Printers	1500	D4:CA:6D:F4:38:88	enabled	41	ether5-slave-local	false	false		Edit Delete
*26	Receptie	1231	D4:CA:6D:F4:38:88	enabled	74	ether5-slave-local	false	false		Edit Delete
*27	Administratie	1111	D4:CA:6D:F4:38:88	enabled	123	ether5-slave-local	false	false		Edit Delete
*29	Sales	1111	D4:CA:6D:F4:38:88	enabled	675	ether5-slave-local	false	false		Edit Delete
*2B	TestComment	1500	D4:CA:6D:F4:38:88	enabled	1337	ether5-slave-local	false	false	VLAN comment test	Edit Delete

Showing 1 to 6 of 6 entries

Copyright © 2016 BlackGATE. All rights reserved. Developed by: Clemens Wijnekus

Figuur 23: Router dashboard

6.2.2 Demo applicatie

Middels een demonstratie van de applicatie worden de voortgang en functionaliteit van de applicatie aan de opdrachtgever en overige stakeholders getoond. Zij zijn erg content met de voortgang en de werking van de applicatie. Er is enkele feedback meerdere punten:

- De locaties zijn niet gescheiden per klant
- Sommige configuratie opties zijn overbodig
- Sommige configuratie opties missen
- Er is geen back up optie
- Er is geen optie om meerdere instellingen in één stap te configureren

De eerste drie opties zijn kleine wijzigingen die zo gemaakt zijn. Er zijn verschillende opties die een standaard waarde moeten hebben, zoals bijvoorbeeld de configuratie opties “L2mtu”. Deze hoeft niet door de gebruiker ingevuld te worden. De “L2mtu” staat wel in het configuratie bestand, echter krijgt deze optie standaard de waarde “1594” van de router als deze niet wordt meegestuurd. Bij bijvoorbeeld de NAT instellingen, worden de poorten gemist. Zowel source als destination. Deze configuratie opties moeten toegevoegd worden. Het splitsen van de locaties per klant is een kwestie van de programma logica iets aanpassen zodat ze worden weergegeven per klant.

De overige 2 aanmerkingen zijn eisen die niet zijn gesteld in de systeemeisen. Het laatste punt doelt op de functie om op één pagina alle configuratie opties voor meerdere instellingen te kunnen invoeren. De stakeholders willen graag één pagina waarop bijvoorbeeld een VLAN, DHCP pool, DHCP server, DHCP netwerk en NAT regel aangemaakt kunnen worden. Er is daarom in overleg met de opdrachtgever besloten om deze functies op dit moment niet te implementeren. Tenzij er tijd over is.

6.3 Iteratie 3: Verwerken feedback demonstratie en testen applicatie

Doordat de vorige iteratie uitloopt moet er deze iteratie geprobeerd worden een inhaalslag te maken. Er wordt in deze iteratie gefocust op de CRUD functionaliteiten voor de ARP tabel, routed subnets, queue's en gebruikersbeheer. Tevens wordt de feedback van de demonstratie verwerkt. De locaties moeten per klant worden opgesplitst in het locatie overzicht scherm. Daarna zal er getest worden met de opdrachtgever en overige stakeholders. De uitkomst hiervan is te lezen in bijlage Test rapport.

6.3.1 Verwerken feedback demonstratie

Als eerste worden de klanten opgesplitst per locatie in het locatie overzicht scherm. Hiervoor worden de locaties anders opgehaald uit de database. In de situatie voor de demo worden alle locaties in één keer uit de database gehaald ongeacht de klant. Om de locaties makkelijk op te splitsen worden nu alle klanten met bijbehorende locaties uit de database gehaald. Vervolgens worden de configuratie opties aangepast naar hoe de stakeholders deze willen hebben. De specifieke instellingen zijn uit overleg met de stakeholders duidelijk gekregen.

Hierna worden de nieuwe functies geïmplementeerd om de ARP tabel, routed subnets, queue's en gebruikers te kunnen beheren. Hiermee kan de grootste inhaalslag qua tijd gemaakt worden. Dit omdat het makkelijker gaat vanwege reeds vergaarde kennis die is opgedaan in de vorige iteratie. Er is niet uit de configuratie, genoemd in Iteratie 2, op te maken hoe routed subnets moeten worden geconfigureerd. Door middel van overleg met de netwerkbeheerders is duidelijk geworden dat dit een samenstelling is van een VLAN, een NAT regel en een IP adres toewijzen aan een interface. Als gevolg hiervan kunnen routed subnets niet als 1 geheel worden ingezien, omdat het aparte instellingen zijn. Voor het beheren van routed subnets hoeft dus alleen de functionaliteit toegevoegd te worden om IP adressen toe te voegen aan of te verwijderen van interfaces.

6.3.2 Testen applicatie

Met behulp van het test plan is de applicatie getest. Er zijn verschillende scenario's uitgezet voor de tester om te testen. Deze scenario's volgen de flow van de applicatie waardoor alles op de juiste volgorde wordt getest. De scenario's zijn bepaald aan de hand van de systeemeisen, om te testen of de ontwikkelde applicatie hieraan voldoet.

De tester is een stakeholder van het project. De opdrachtgever is namelijk niet aanwezig tijdens de test fase. Het test plan wordt omgezet naar het test rapport, hierin worden de bevindingen van de tester opgenomen. Naast elke stap uit een test scenario kan de actuele uitvoer genoteerd worden, vervolgens kan de tester er commentaar bij vermelden indien nodig. Het test rapport en de bevindingen zijn te vinden in bijlage Test rapport.

Vervolgens is gekeken hoeveel tijd het scheelt ten opzichte van de oude situatie. Er zijn meerdere scenario's getest hiervoor. Het uitvoeren van één wijziging op een router in de oude situatie is 60 seconden. In de nieuwe situatie is die tijd gehalveerd en kost het nog 30 seconden. Dat betekent dat er, in verhouding, bij tien routers al vijf minuten wordt bespaard. ($10 \times 30s = 300s \div 60 = 5min$) Door het bundelen van de routers in de ontwikkelde applicatie en de gebruiksvriendelijkere interface is het opzoeken van IP adressen, wachtwoorden of instellingen minder tijdrovend. Daarnaast, als er eenmaal in de ontwikkelde applicatie is ingelogd, hoeft niet meer apart ingelogd te worden op een andere router om daar een wijziging in aan te brengen. Dit scheelt ook tijd omdat het opzoeken van het IP adres en hele inlog proces niet meer hoeft worden doorlopen.

6.4 Iteratie 4: Verwerken feedback test rapport

Uit het testen van de applicatie zijn een aantal bugs en fouten naar voren gekomen. Deze fouten variëren van het niet weergeven van een foutmelding tot het niet werken van een functie. Tevens zijn er wat aanpassingen die gemaakt moeten worden om het systeem beter te laten functioneren. Er kan op dit moment een router toegevoegd worden aan een locatie met een IP adres die al in gebruik is door een andere router voor die locatie. Dit mag niet mogelijk zijn om te voorkomen dat er routers meerdere keren worden toegevoegd aan een locatie. Hiervoor moet een aanpassing gemaakt worden in de database en in de bron code.

Zodra alle fouten zijn opgelost worden alle wijzigingen in versie beheer gezet en wordt de hoofdlijn helemaal bijgewerkt met alle wijzigingen. Om het ontwikkelen en uitbrengen van nieuwe versies in de toekomst makkelijk te maken, wordt er op dit punt een nieuwe branch "Release" gemaakt. Dit is de versie die live staat op de omgeving. Hierdoor kunnen wijzigingen worden doorgevoerd op de hoofdlijn en hoeft niet voor elke wijziging een nieuwe versie uitgebracht te worden. Zodra er meerdere wijzigingen op de hoofdlijn zitten kunnen deze worden verwerkt in de "Release" branch om vervolgens in één keer de applicatie bij te werken met meerdere wijzigingen. Vanaf dit punt krijgt de applicatie het label "Versie 1.0.0".

7 Transition

De transition fase omvat de overdracht van de documentatie en het live brengen van de applicatie. Er is binnen BlackGATE een webserver opgezet waarop de ontwikkelde applicatie kan draaien. Zodra versie 1.0.0 op de webserver staat moeten er enkele configuratie opties, zoals de verbindingsparameters voor de database, aangepast worden in de bron code zodat de applicatie op de nieuwe omgeving kan werken. Daarnaast moet de database ingericht worden. Om deze reden is eerder het migratie systeem van het framework ingericht. Middels het migratie systeem kan de database gemakkelijk en consistent met de ontwikkelomgeving worden ingericht. Zodra dit gebeurt is kan getest worden of de applicatie goed is live gebracht.

Het testen of de applicatie functioneel is op de live omgeving brengt een aantal problemen aan het licht. De omgeving die is gebruikt voor het ontwikkelen van de applicatie is een Windows omgeving. De omgeving waarop de applicatie draait is een Linux omgeving. Dit geeft enkele problemen. Linux is iets strenger dan Windows. Het bestandensysteem van Linux is hoofdletter gevoelig, wat enkele problemen oplevert bij het inladen van bestanden door de applicatie. Daarom moeten wat bestanden hernoemd worden zodat Linux er ook mee kan werken, dit levert geen problemen op op de Windows ontwikkel omgeving.

De hoofdletter gevoeligheid is niet het enige probleem. De versie van het Linux systeem, CentOS 6.8, ondersteunt PHP standaard tot versie “5.3”. Dit levert problemen op bij het gebruiken van de applicatie omdat deze versie bepaalde functies niet ondersteunt ten opzichte van versie “5.5” die is gebruikt op de ontwikkel omgeving. Versie “5.5” wordt standaard meegeleverd met het webserver pakket “XAMPP” voor Windows. Er is via een bron gemaakt door derden versie “5.5” geïnstalleerd. Deze bron, oftewel “repository”, maakt latere versies van PHP die niet standaard ondersteunt zijn beschikbaar zodat deze wel gebruikt kunnen worden op verschillende versies van CentOS.

Zodra de applicatie live is, wordt er nog advies uitgebracht aan de opdrachtgever om een SSL certificaat te installeren om een HTTPS verbinding mogelijk te maken. Dit wordt geadviseerd omdat binnen BlackGATE de methode voor handen is en daarom de veiligheid alleen maar kan verbeteren. Vervolgens wordt de benodigde documentatie overhandigd aan de opdrachtgever inclusief de gegevens van het versie beheer systeem.

8 Conclusie

De nieuw ontwikkelde applicatie is een verbetering in het management proces van de MikroTik's. De routers kunnen in de nieuwe applicatie gebundeld en gemanaged worden. Bij het wijzigen of toevoegen van instellingen zijn alleen de benodigde configuratie opties te zien wat het gebruiksvriendelijker en minder complex maakt. Daarnaast zijn de routers gebundeld per locatie per klant, dit maakt het opzoeken van een router ook eenvoudiger. In de oude situatie had een medewerker via de IP registratie van BlackGATE eerst het IP adres op moeten zoeken voordat er naar het configuratie paneel van de router genavigeerd kon worden. Uit het testen is ook gebleken dat er tijd wordt bespaard. De opdrachtgever heeft aangegeven de applicatie nu ook in de praktijk te willen gaan gebruiken.

Uit het onderzoek is ook gebleken dat de SonicWALL's niet op afstand gemanaged kunnen worden, via een andere applicatie dan het GMS. Er is geen advies te geven over het GMS omdat er in dit onderzoek geen onderzoek naar de mogelijkheden van het systeem zijn gedaan. Er is echter wel geadviseerd om naar GMS te kijken als BlackGATE toch een management tool voor de SonicWALL's wil.

Daarnaast wordt een afzonderlijk doel wat niet binnen de scope van dit project valt bereikt, het wordt mogelijk gemaakt voor werknemers zonder de juiste technische kennis in pacht om zelf de routers te beheren. Onder deze technische kennis valt kennis over de command line tot de geavanceerdere instellingen van een router.

9 Evaluatie

In dit hoofdstuk wordt de afstudeeropdracht geëvalueerd. Dit wordt gedaan op de volgende drie punten: het proces van het project, de opgeleverde producten en de beroepstaken.

9.1 Proces

Voor dit project is de methode RUP gebruikt. RUP heeft heel erg geholpen met de planning en de opzet van het project. In het fase plan is al rekening gehouden met de moeilijkheidsgraad van verschillende onderdelen van het project. Zo is er in de elaboration fase meer tijd ingeruimd voor de probleemdomein en systeemeisen analyse maar ook het nogmaals itereren over het architectuur document. De analyse verliep stroever dan verwacht maar leverde geen problemen op in de verdere project planning. In de construction fase is de planning in iteratie 2 iets uitgelopen, dit omdat er moeilijkheden waren met het opstellen van de juiste API verzoeken en het interpreteren van de output. Dit is echter weer ingelopen in iteratie 3 van dezelfde fase. Middels de planning en de bijbehorende mijlpalen is het schrijven van de documentatie structureel gebeurd. Dit was erg fijn in dit project. Hierdoor kon ook het verslag in delen worden opgebouwd. Ook de ingeplande contact momenten met de opdrachtgever en stakeholders hebben geholpen dit project tot een goed einde te brengen.

Er is voor volgende projecten rekening te houden met het aantal zijwegen die ingeslagen worden en hoe diep daarop ingegaan wordt. Zoals met het onderzoek naar AJAX, hoofdstuk 6.1.4. Dit is een zijweg waarop mogelijkheden zijn voor de applicatie, deze zijn echter voor volgende versies beter bestemd omdat het implementeren hiervan niet meer in het tijdsbestek paste op dat moment. Hier moet beter op gelet worden zodat hier niet onnodig veel tijd aan verloren gaat omdat dit niet gebruikt gaat worden in het project. Daarnaast moet er meer inzicht gegeven worden in de voortgang van het project aan de opdrachtgever. Afgezien van de contact momenten over de systeemeisen en de demonstratie moet er meer inzicht gegeven worden in de algemene voortgang van het proces.

9.2 Product

Vision document

Het vision document is een uitgebreide versie van het afstudeerplan. Tevens is in dit document de planning gemaakt voor de fases die gedaan worden. Het maken van de planning heeft veel geholpen bij het nadenken over wat er gedaan moest worden en hoe lang dat ging duren.

Probleemdomein en systeemeisen analyse

In het probleemdomein en systeemeisen analyse document is het uitgebreide onderzoek beschreven wat heeft plaatsgevonden in de eerste iteratie van de elaboration fase. Het is officieel een onderdeel van het architectuur document, maar omdat het een erg groot verslag werd is dit opgenomen in een apart document om het geheel overzichtelijk te houden. In deze analyse zijn ook de mogelijkheden van de hardware onderzocht.

Architectuur document

Er waren twee iteraties ingepland voor het opstellen van het architectuur document. Hierin is de voorgaande analyse van het probleemdomein en de systeemeisen kort samengevat en de basis gelegd voor het project met een prototype. Het architectuur document legt de basis voor het ontwerp document.

Ontwerp document

Het ontwerp document bevat de uiteindelijke systeemeisen en het ontwerp voor de te ontwikkelen applicatie. Onder het ontwerp vallen alle use cases, het use case diagram, mockups en het klassendiagram.

Prototype

Met het prototype wordt aangetoond dat de onderzochte mogelijkheden ook daadwerkelijk in praktijk te brengen zijn. Er is een eerste versie van het prototype ontwikkeld in de tweede iteratie van de elaboration fase om aan te tonen dat de mogelijkheid er is. In de daarop volgende iteratie is er nogmaals over het architectuur document en het prototype geïtereerd om te demonstreren dat de mogelijkheden ook te combineren zijn met de eisen van de opdrachtgever.

Test rapport

Het test rapport is een ingevulde versie van het test plan. In het test rapport staan alle bevindingen van de tester. Aan het einde van het test rapport is door de tester aangegeven of de geteste functies voldoen aan de eisen.

Ontwikkelde applicatie

Het uiteindelijke eindproduct van het project is de ontwikkelde applicatie. Door middel van de demonstratie en het testen is een eindproduct afgeleverd waar de stakeholders content mee zijn. Het versie beheer systeem heeft geholpen bij de ontwikkeling. Helaas kunnen de SonicWALL's niet gemanaged worden met de applicatie. Ondanks dit is het een verbetering voor het management proces van de MikroTik's.

9.3 Beroepstaken

A1 Analyseren van het probleemdomein:

Het onderzoeken en beschrijven van het gegeven probleemdomein (van de opdrachtgever) in termen van het TI domein, en het vaststellen van de gewenste veranderingen.

Het inwerken in het domein van de opdrachtgever, maakt onderdeel van deze taak uit.

In de eerste iteratie van de elaboration fase is er uitgebreid onderzoek gedaan naar het huidige probleem domein. Er is gekeken naar de huidige situatie maar ook naar de mogelijkheden. Zie hiervoor hoofdstuk 5.1 en de bijbehorende bijlage Probleemdomein en Systeemeisen analyse document.

C10 Ontwerpen van een systeemarchitectuur:

Het maken van een ontwerp waarmee het gehele systeem beschreven wordt in termen van samenwerkende onderdelen, zoals subsystemen, componenten en/of klassen. Het kiezen van standaarden voor computersystemen zodat alle componenten van de technische infrastructuur optimaal op elkaar aansluiten en met elkaar samenwerken.

Bij de verantwoording van keuzes tussen gelijksoortige standaards dienen zaken als: open en proprietary standaards, populariteit, complexiteit en levensduur tegen elkaar te worden afgewogen.

Bij het ontwerpen van de systeem architectuur is er rekening gehouden met de mogelijkheden van de hardware en de systeemeisen. Daarnaast is er een prototype gebouwd om te kunnen bewijzen dat de gekozen mogelijkheden werken. Er is beschreven hoe de MikroTik's, de API van RouterOS en de applicatie samenwerken. Hierover is meer te lezen in de hoofdstukken 5.2 en 5.3 en in het bijbehorende Architectuur document te vinden in bijlage Architectuur document.

D17 Testen van software systemen:

Het toetsen of het software systeem voldoet aan de gestelde eisen door het opstellen van een testplan en het toepassen van methoden en technieken voor het testen.

Middels het test plan en het daaruit voortgekomen test rapport is de applicatie getest op alle eisen van de stakeholders. Er zijn meerdere scenario's beschreven, zowel foutieve als correcte. Hierbij is beschreven welke handelingen uitgevoerd moeten worden en wat de verwachte uitvoer van de handeling is. De stakeholder kan vervolgens hierbij noteren wat de uitvoer is en hier eventueel commentaar bij noteren. Hiermee is gekeken of de applicatie voldoet aan de eisen. Het test rapport is bijgevoegd in bijlage Test rapport.

G1 Praktische aspecten hanteren in (internationale) projecten:

Voorbeelden: opstellen opdrachtschrijving en projectplan/plan van aanpak. Inrichten projectorganisatie, meewerken aan projecten en daarbij je eigen werkzaamheden plannen en begroten; werken met methoden, technieken en hulpmiddelen voor projectplanning en het begroten en evalueren van projecten, een risicoanalyse uitvoeren bij de aanvang van kleine en middelgrote projecten, e.d.

Het organiseren van het project is gedaan in de inception fase van het project. Er is maar één project lid. Dit is een factor geweest bij het kiezen van de project methodiek en het maken van de planning. Al deze aspecten zijn terug te vinden in hoofdstuk 4 met bijbehorende bijlage Vision document.

Afkortingenlijst

AJAX	Asynchronous Javascript And XML
API	Application Program Interface
CGI	Common Gateway Interface
CLI	Command Line Interface
CRUD	Create, Read, Update, Delete
GMS	Global Management System
GUI	Graphical User Interface
HTTP(S)	Hypertext Transfer Protocol (Secure)
MiB	Management information Base
OID	Object Identifiers
PHP	Hypertext Preprocessor
RUP	Rational Unified Process
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
VoIP	Voice over IP

Literatuur

- Analyse: zwakke wachtwoorden op Tweakers.net. (2011). *tweakers.net*. Op 22 april 2016 verkregen van <http://tweakers.net/plan/589/analyse-zwakke-wachtwoorden-op-tweakers-punt-net.html>
- API PHP class. (z. j.). *MikroTik*. Op 24 februari 2016 verkregen van http://wiki.mikrotik.com/wiki/API_PHP_class
- Chacon, S. & Straub, B. (2014). *Pro git*. New York City, New York: Apress.
- GCIH Practical Assignment TCP Port 23. (2000). *SANS Institute*. Op 1 mei 2016 verkregen van <https://www.giac.org/paper/gcih/328/tcp-port-23/103233>
- Kumar, S., Paar, C., Pelzl, J., Pfeiffer, G., Rupp, A. & Schimmler, M. (2006). How to break des for bc 8,980. *SHARCS '06-Special-purpose Hardware for Attacking Cryptographic Systems*.
- MD5 considered harmful today. (2008). *Technische Universiteit Eindhoven*. Op 22 april 2016 verkregen van <http://www.win.tue.nl/hashclash/rogue-ca/>
- SNMP::set. (z. j.). *The PHP Group*. Op 25 februari 2016 verkregen van <http://php.net/manual/en/snmp.set.php>

Bijlagen

Bijlage I:	Afstudeerplan
Bijlage II:	Formulier concept bespreking
Bijlage III:	Formulier tussentijds assessment
Bijlage IV:	Vision document
Bijlage V:	Requirements en probleemdomein analyse
Bijlage VI:	Architectuur document
Bijlage VII:	Ontwerp document
Bijlage VIII:	Test rapport

De bijlagennummers komen overeen met de tabblad nummers in het bijlagen document.