

Prime Vision Applicaties naar Android Porten

Bijlage

Student

Friso Denijs - 10102426

Bedrijf

Prime Vision

Datum

1 September



Inhoudsopgave

Inhoudsopgave.....	2
Bijlage A - Afstudeerplan.....	3
Bijlage B - Plan van Aanpak.....	7
B.1 Achtergronden.....	7
B.2 Projectresultaat.....	8
B.3 Projectactiviteiten.....	8
B.4 Projectgrenzen.....	8
B.5 Tussenresultaten.....	9
B.6 Kwaliteit.....	9
B.7 Planning.....	9
B.8 Risico's.....	9
Bijlage C - Rapport Iteratie 1 - Notifications.....	10
C.1 Inleiding.....	10
C.2 Requirements.....	11
C.3 Functioneel Ontwerp.....	14
C.4 Technisch Ontwerp.....	18
C.5 Test.....	22
C.5.1 Testen Connectie.....	22
C.5.2 Testen CRUD Filter Set.....	23
C.6 Verbeterpunten.....	24
C.A - Sequence Diagrams.....	25
C.B - Test Uitvoering.....	33
C.B.1 Testen Connectie.....	33
C.B.2 Testen CRUD Filter Set.....	34
Bijlage D - Iteratie 2.....	35
D.2 Requirements.....	35
D.3 Functioneel Ontwerp.....	36
Bijlage E - Advies.....	38
E.1 Obstakels en oplossingen.....	38
E.1.1 Android devices.....	38
E.1.2 GUI design.....	38
E.1.3 Delphi 2007.....	41
E.1.4 Delphi XE7.....	41
E.1.5 Code Prime Vision en Third Parties.....	42
E.2 Alternatieve Ontwikkelomgevingen.....	45
E.2.1 Android Studio.....	45
E.2.2 Visual Studio.....	45
E.2.3 XCode.....	46
E.2.4 Xamarin.....	46
E.2.5 Qt.....	47
E.2.6 Appcelerator.....	48
E.3 Alternatieve datastromen.....	49
E.4 Verdere overwegingen.....	49
E.5 Advies.....	50
E.6 Bronnen.....	51

Bijlage A - Afstudeerplan

Informatie afstudeerder en gastbedrijf

Afstudeerblok: 2015-1.1 (start uiterlijk 9 februari 2015)
Startdatum uitvoering afstudeeropdracht: 9 februari 2015
Inleverdatum afstudeerdossier volgens jaarrooster: 5 juni 2015

Studentnummer: 10102426
Achternaam: dhr Denijs
toepassing
Voorletters: M.F.
Roepnaam: Friso
Adres: Regentesselaan 283
Postcode: 2562EB
Woonplaats: Den Haag
Telefoonnummer: nvt
Mobiel nummer: 06487066363
Privé emailadres: friso1990@gmail.com

(*) *weghalen niet van*

Opleiding: Informatica
Locatie: Den Haag
toepassing
Variant: voltijd

(*) *weghalen niet van*

Naam studieloopbaanbegeleider: Anneke Wieman
Naam begeleidend examiner: Hélène Weenink
Naam tweede examiner: Gerard Mijnares

Naam bedrijf: Prime Vision B.V.
Afdeling bedrijf: Software Engineering
Bezoekadres bedrijf: Olof Palmestraat 16-18
Postcode bezoekadres: 2616 LR
Postbusnummer: nvt
Postcode postbusnummer: nvt
Plaats: Delft
Telefoon bedrijf: +31 (0)15 219 20 90

Telefax bedrijf: nvt
Internetsite bedrijf: www.primevision.com

Achternaam opdrachtgever: dhr Spaanbroek
toepassing
Voorletters opdrachtgever: G.R.
Titulatuur opdrachtgever: Ing.
Functie opdrachtgever: Software Engineer
Doorkiesnummer opdrachtgever: +31 641 592 701
Email opdrachtgever: g.spaanbroek@primevision.com

(*) *weghalen niet van*

Achternaam bedrijfsmentor: dhr Spaanbroek
toepassing
Voorletters bedrijfsmentor: G.R.
Titulatuur bedrijfsmentor: Ing.
Functie bedrijfsmentor: Software Engineer
Doorkiesnummer bedrijfsmentor: +31 641 592 701
Email bedrijfsmentor: g.spaanbroek@primevision.com

(*) *weghalen niet van*

opdrachtgever
NB: bedrijfsmentor mag dezelfde zijn als de

Doorkiesnummer afstudeerder: nvt
Functie afstudeerder (deeltijd/duaal): nvt

Titel afstudeeropdracht:

Prime Vision Applicaties naar Android Porten

Opdrachtschrijving

1. Bedrijf

Prime Vision is een BV, opgericht in 1956, dat oplossingen biedt voor automatisering, processing en recognition voor post, warenhuizen en andere logistieke bedrijven. Zij heeft iets meer dan 50 werknemers. De opdracht zal plaats vinden in het team software engineering, deze is verantwoordelijk voor het ontwerpen, maken, testen, leveren en onderhouden van de software voor de klanten van Prime Vision.

2. Probleemstelling

Prime Vision wil mobiele applicaties van haar huidige systemen aan haar klanten gaan aanbieden.

3. Doelstelling van de afstudeeropdracht

Prime Vision wil verkennen hoeveel werk het kost om de huidige systemen om te zetten naar Android met Delphi, identificeren welke bottle necks er zijn en oplossingen voor eventuele problemen aangeboden krijgen.

4. Resultaat

Een door Prime Vision gekozen systeem laten draaien op een Android tablet. Het liefst komt hier een bruikbaar proof of concept uit, maar als dit niet lukt wil Prime Vision weten waarom dat niet is gelukt en hoe het dan wel kan.

Aan het eind van mijn afstudeeropdracht is er een Android applicatie van het gekozen systeem of als dat niet is gelukt een rapport over waarom dat er niet is.

Het gekozen systeem bestaat minstens uit een database connectie, event afhandeling en statistiek generatie.

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

Het systeem wordt ontwikkeld door evolutionary prototyping.

Plan van Aanpak schrijven (2~3 dagen)

Eerste versie schrijven (1 dag)

Eerste versie bespreken (halve dag)

Eventueel eerste versie verbeteren (1 dag)

Eventueel tweede versie bespreken (halve dag)

Requirements vaststellen (3-5 dagen)

Bespreken details opdracht (halve dag)

Details omzetten naar requirements (2 dagen)

Requirements bespreken (halve dag)

Eventueel requirements verbeteren (1 dag)

Eventuele verbeteringen bespreken (halve dag)

Initiele architectuur ontwerpen (5 dagen)

Requirements omzetten naar use cases

Use cases omzetten naar UML

View prototype (15 dagen)

- Android UI design onderzoeken
- View test ontwerpen
- View ontwerpen
- View maken
- View testen
- Eventueel view verbeteren
- View prototype rapport maken

Model prototype (10 dagen)

- Model test ontwerpen
- Model ontwerpen
- Model maken
- Model testen
- Eventueel model verbeteren
- Model prototype rapport maken

Database connection prototype (10 dagen)

- Database connection test ontwerpen
- Database connection ontwerpen
- Database connection maken
- Database connection testen
- Eventueel database connection verbeteren
- Database connection prototype rapport maken

Overige functionaliteit prototype (10 dagen)

- Overige functionaliteit test ontwerpen
- Overige functionaliteit ontwerpen
- Overige functionaliteit maken
- Overige functionaliteit testen
- Eventueel overige functionaliteit verbeteren
- Overige functionaliteit prototype rapport maken

Proof of Concept maken (10 dagen)

- Laatste iteratie prototype opschonen
- Advies opstellen

Afstudeerdossier opbouwen (15 dagen)

6. Op te leveren (tussen)producten

Plan van Aanpak

Design documentatie

Requirements

Literatuuronderzoek

Architectuur

Prototype rapporten

View

Model

Database connection

Overige functionaliteit

Proof of Concept

Adviesrapport

7. Te demonstreren competenties en wijze waarop

De requirements zal ik vaststellen op basis van de al bestaande applicatie, de eisen en wensen van Prime Vision en de werking en beperkingen van Delphi (1.4). Ik schat dit op niveau 3 te doen omdat ik voor een ingewikkeld systeem zelfstandig de requirements moet maken.

Het ontwerpen van de systeem delen (3.2) schat ik op niveau 4 te doen, omdat ik een complexe desktop applicatie moet omzetten naar een gebruikersvriendelijke (moet voldoen aan niet-functionele kwaliteitseisen) mobiele applicatie, die uiteindelijk voor distributie geschikt moet zijn.

Het bouwen van de applicatie (3.3) verwacht ik op niveau 3 te doen, omdat ik een objectgeoriënteerde applicatie moet bouwen waarbij ik waarschijnlijk de geavanceerde functionaliteiten van Delphi voor moet gebruiken. Ik moet ook rekening houden met toekomstige wijzigingen, testbaarheid en hergebruik.

Het uitvoeren van en rapporteren over de tests (3.5) verwacht ik op niveau 3 te doen, omdat ik het hele systeem moet testen op basis van een zelf te maken logisch test ontwerp.

Bijlage B - Plan van Aanpak

B.1 Achtergronden

Prime Vision is een BV, opgericht in 1956, dat oplossingen biedt voor automatisering, processing en recognition voor post, warenhuizen en andere logistieke bedrijven. Zij heeft iets meer dan 50 werknemers.

De opdracht zal plaats vinden in het team software engineering, deze is verantwoordelijk voor het ontwerpen, maken, testen, leveren en onderhouden van de software voor de klanten van Prime Vision.

Prime Vision wil mobiele applicaties van haar huidige systemen aan haar klanten gaan aanbieden.

Maar, voordat zij dit gaat doen wil zij verkennen hoeveel werk het kost om een huidig systeem om te zetten naar Android met Delphi, identificeren welke bottle necks er zijn en oplossingen voor eventuele problemen aangeboden krijgen.

Het gekozen systeem is een onderdeel van het Mail Match System (MMS), genaamd Sorting Machine Extended (SMX). Dit systeem wordt gebruikt om brieven op hoge snelheid te sorteren en de te maken app moet dat proces monitoren. Bij voorkeur komt uit dit project een werkende app, maar als dit (volledig) niet lukt moet gedocumenteerd worden waar de valkuilen lagen en hoe het in de toekomst beter kan.

B.2 Projectresultaat

Aan het eind van het project is er een adviesrapport over of en hoe Android apps met Delphi gemaakt kunnen worden met eventueel een werkende app.

Prime Vision wil mobiele applicaties van haar huidige systemen aan haar klanten gaan aanbieden.

Maar, voordat zij dit gaat doen wil zij verkennen hoeveel werk het kost om een huidig systeem om te zetten naar Android met Delphi, identificeren welke bottle necks er zijn en oplossingen voor eventuele problemen aangeboden krijgen.

Het gekozen systeem is een onderdeel van het Mail Match System (MMS), genaamd Sorting Machine Extended (SMX). Dit systeem wordt gebruikt om brieven op hoge snelheid te sorteren en de te maken app moet dat proces monitoren. Bij voorkeur komt uit dit project een werkende app, maar als dit (volledig) niet lukt moet gedocumenteerd worden waar de valkuilen lagen en hoe het in de toekomst beter kan.

B.3 Projectactiviteiten

Het project is in twee delen opgedeeld omdat er eerst een Proof of Concept gemaakt moet worden om te kijken naar de mogelijkheden en mijn kunnen. Hierna zijn er meerdere iteraties die samen uiteindelijk een product moeten vormen.

Gedurende beide delen zal ik werken aan mijn afstudeerrapport en het uiteindelijk adviesrapport.

De detail planning zal in mijn afstudeerrapport verschijnen.

B.4 Projectgrenzen

Ik zal mij alleen focussen op functionaliteit van het huidige SMX systeem om te zetten naar een mobiele app.

B.5 Tussenresultaten

Gedurende het project zullen de volgende producten worden gemaakt:

- Plan van Aanpak
- Proof of Concept document
- Per iteratie een document
- Afstudeerrapport
- Adviesrapport
- Architectuur Document

B.6 Kwaliteit

Om de kwaliteit te bewaken zal ik mij houden aan de code conventie die opgesteld is door Prime Vision en iedere maandag met mijn begeleider bespreken wat er is gedaan en wat er gedaan zal worden.

B.7 Planning

Belangrijke data:

9 februari - Start project
10 maart - Afstudeerbegeleider op bezoek
3 april - Voortgangsrapport inleveren
10 april - Concept afstudeerrapport inleveren
14 mei - Afstudeerrapport inleveren voor assessment
5 juni - Inleverdatum afstudeerrapport

B.8 Risico's

Het grootste risico is mijn gebrek aan kennis van Delphi. Om dit op te lossen heb ik al wat tutorials gevolgd op <http://www.delphibasics.co.uk/> en <http://docwiki.embarcadero.com/>. Als ik als nog niet uit een probleem kom zal ik het op www.stackoverflow.com of aan een van mijn collega's vragen.

Daarnaast is een risico mijn gebrek aan kennis van het SMX systeem. Om dit op te lossen bestudeer ik de documenten op <https://scm.primevisiononline.com/> en als ik er als nog niet uit kom kan ik het aan een van mijn collega's vragen.

Bijlage C - Rapport Iteratie 1 - Notifications

C.1 Inleiding

In de eerste iteratie van mijn afstudeeropdracht heb ik het Notifications onderdeel van het al bestaande Sorting Machine eXtended (SMX) systeem naar Android geport. Om dit te kunnen doen heb ik ook een server onderdeel moeten maken in het originele systeem, maar omdat het alleen push en pull procedures bevat die overeenkomen met de push en pull procedures van de app heb ik de server niet gedetailleerd beschreven in dit rapport.

Om de app te maken heb ik eerst de requirements opgesteld aan de hand van wat het huidige systeem doet. Vervolgens heb ik die requirements omgezet naar Use Cases. Aan de hand van de requirements en use cases heb ik het technisch ontwerp en de app gemaakt en getest.

Tot slot ben ik tijdens het testen verbeterpunten tegengekomen en deze beschrijf ik in het laatste hoofdstuk.

C.2 Requirements

R.1 tot en met R.18 zijn gebaseerd op het huidige systeem, zoals gebleken uit het onderzoek van 3.4.1 Vooronderzoek. Dit is wat de app moet kunnen.

R.19 heb ik toegevoegd om de gebruiker te dwingen om de app in landscape mode te houden, zodat er meer informatie kan worden weergegeven in een scherm.

R.20 heb ik toegevoegd zodat bij de volgende iteratie al tabs zullen zijn voor de overige functionaliteiten.

R.21 is voor het ontvangen van de notificaties.

Code	Beschrijving
R.1	De main view moet een filter selector bevatten
R.2	De main view moet een notification viewer bevatten
R.3	De filter selector moet filters bevatten
R.4	De filters binnen de filter selector moeten selecteerbaar zijn
R.5	De filter selector moet een button bevatten om een filter toe te voegen
R.6	De filter selector moet een button bevatten om een filter te kopiëren
R.7	De filter selector moet een button bevatten om een filter te bewerken
R.8	De filter selector moet een button bevatten om een filter te verwijderen
R.9	De filter selector moet een button bevatten om een nieuwe filter set te maken
R.10	De filter selector moet een button bevatten om een filter set te loaden
R.11	De filter selector moet een button bevatten om een filter set te save
R.12	De filter selector moet toestaan om filters op en neer te bewegen
R.13	De buttons om een filter toe te voegen of bewerken moeten naar een filter edit dialog openen
R.13a	De filter edit dialog moet een edit box voor de naam van de filter bevatten
R.13b	De filter edit dialog moet een drop down list voor het selecteren van hoe de app om moet gaan met onbekende notificaties bevatten
R.13c	De filter edit dialog moet aangeven of een filter veel berichten door laat
R.13d	De filter edit dialog moet een checklistbox bevatten voor iedere bekende soort notificatie of het filter die door moet laten of niet
R.14	De NotificationReceiver moet alle bekende NotificationSenders in een lijst opslaan
R.15	Wanneer een onbekende notificatie soort wordt ontvangen moet een dialog geopend worden of het aan de interne filter set toegevoegd moet worden
R.16	Wanneer een filter set open is en het geen notificatie soorten bevat die bekend zijn bij de NotificationReceiver moet een dialog geopend worden om te vragen of de gebruiker notificatie soorten toe wil voegen aan de filter set
R.17	De gebruiker moet filtersets op kunnen slaan
R.18	De notification viewer moet binnengekomen notifications weergeven
R.19	De gebruiker moet notifications kunnen verwijderen
R.20	De gebruiker moet kunnen kiezen of de notifications continu geupdate worden of niet
R.21	De layout moet altijd in landscape mode zijn
R.22	De app moet al tabs voor de andere monitor functionaliteiten hebben
R.23	De app moet via het internet verbinding kunnen maken met de monitoring server

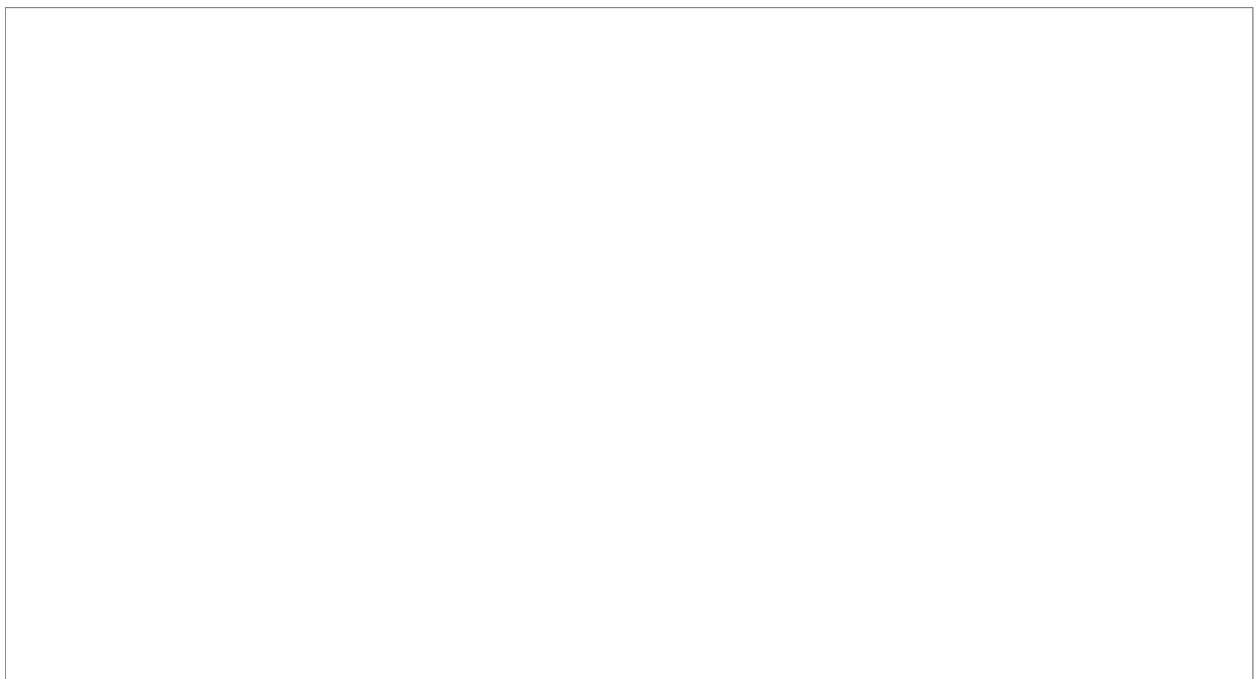
Figuur C.2.1 laat zien hoe de notifications view in de desktop versie er uit ziet. De mobiele versie moet er zo veel mogelijk hetzelfde er uit zien, zodat de gebruiker niet opnieuw hoeft te leren hoe de app in elkaar zit.

Het laat ook zien dat een notificatie bestaat uit:

- Time
- Type
- Code
- Application
- en Description

De filters kunnen gesorteerd worden op naam of of dat ze aan staan of niet.

De notificaties kunnen gesorteerd op tijd, type, code, applicatie en beschrijving. Daarnaast kunnen de notifications verwijderd worden en er kan gekozen worden of ze continu geupdate moeten worden of niet.



Figuur 2.1 View Desktop SMX Notifications

Figuur C.2.2 laat zien hoe de filter edit dialog er uit ziet in de desktop versie. Er is te zien dat er gefilterd kan worden op type, application en er kan uitzondering gemaakt worden voor bepaalde text. Verder kan er gekozen worden om alles te selecteren of deselecteren, of de filter onbekende types notifications door moet laten en of het filtert direct geactiveerd moet worden wanneer er op apply wordt gedrukt. Tot slot kan er een naam voor het filter ingevuld worden.

Applications

☒ All applications
 ☐ Filter by mask:

☐ Filter by known applications:

Always show:

☐ Errors (ERR)
☐ Warnings (WRN)
☐ Information (INF)
☐ Debug (DBG)
☐ Progress (PRG)
☐ PV Info (PVI)

Except for

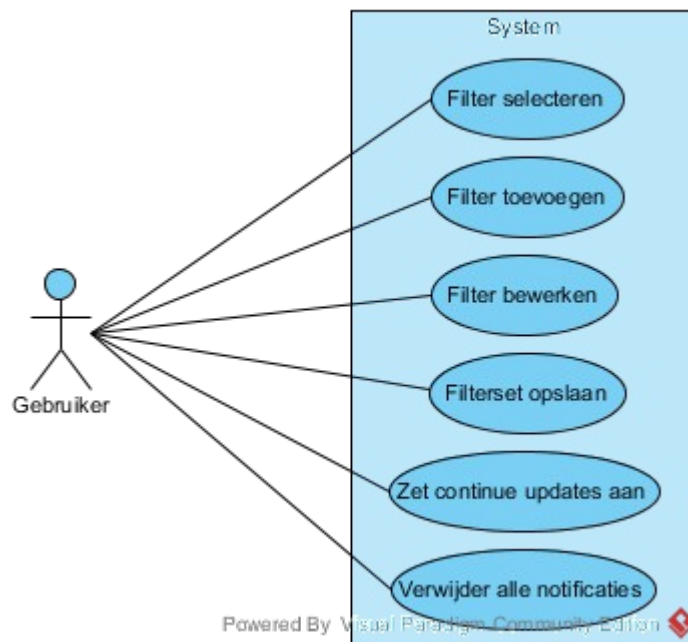
A B C

lblExceptForText

☐ Heavy filter
 ☐ Select / deselect all
 ☐ OnlyShowSelected
 ☐ Pass unknown notifications
 ☐ Activate upon apply

Figuur C.2.2 View Desktop SMX Notifications Filter Edit view

C.3 Functioneel Ontwerp



Figuur C.3.1 Use Cases

ID	UC.1
Requirements	R.4
Title	Filter selectie
Description	Gebruiker selecteert een filter om te gebruiken
Primary Actor	Gebruiker
Preconditions	-
Postconditions	De app laat de notificaties zien die de filter door laat
Main Success Scenario	<ol style="list-style-type: none"> 1. De gebruiker selecteert een filter 2. Het systeem controleert welke types notificaties het filter doorlaat 3. Het systeem toont notificaties die het filter doorlaat
Extensions	<p>2a. Er zijn meerdere filters geselecteerd</p> <ol style="list-style-type: none"> 1. Het systeem controleert welke types notificaties de filters doorlaten 2. Het systeem toont de notificaties die de filters gezamenlijk doorlaten <p>3a. Er zijn geen notificaties binnengekomen die het filter doorlaat</p> <ol style="list-style-type: none"> 1. Het systeem toont een text die zegt dat er geen notificaties binnen zijn gekomen die het filter doorlaat

ID	UC.2
Requirements	R.13
Title	Filter toevoegen
Description	De gebruiker voegt een filter toe
Primary Actor	Gebruiker
Preconditions	-
Postconditions	Er is een filter aan de filterset toegevoegd
Main Success Scenario	<ol style="list-style-type: none"> 1. De gebruiker drukt op de filter toevoegen knop 2. Het systeem opent de edit filter dialog box 3. De gebruiker vult gegevens in 4. De gebruiker drukt op de apply knop 5. Het systeem voegt het filter toe aan de filter set 6. Het systeem sluit de edit filter dialog box
Extensions	4a. De gebruiker drukt op de cancel knop <ol style="list-style-type: none"> 1. Het systeem sluit de edit filter dialog box

ID	UC.3
Requirements	R.13
Title	Filter bewerken
Description	De gebruiker bewerkt een filter
Primary Actor	Gebruiker
Preconditions	-
Postconditions	Een filter is veranderd
Main Success Scenario	<ol style="list-style-type: none"> 1. De gebruiker selecteert een filter 2. De gebruiker drukt op de filter bewerken knop 3. Het systeem opent de edit filter dialog box 4. De gebruiker verandert gegevens 5. De gebruiker drukt op de apply knop 6. Het systeem wijzigt het filter 7. Het systeem sluit de edit filter dialog box
Extensions	5a. De gebruiker drukt op de cancel knop <ol style="list-style-type: none"> 1. Het systeem sluit de edit filter dialog box

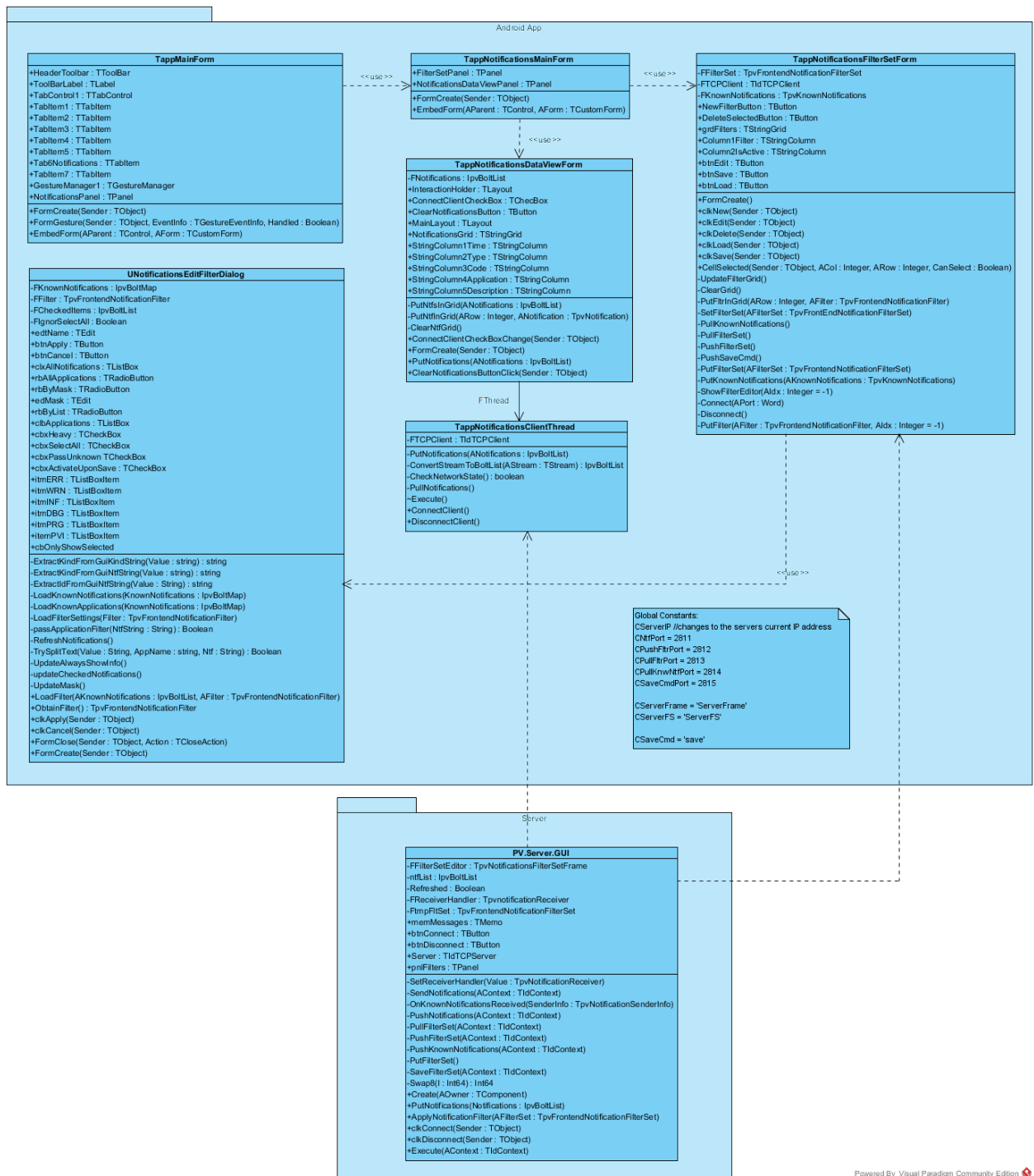
ID	UC.4
Requirements	R.17

Title	Filterset opslaan
Description	De gebruiker slaat de filterset op
Primary Actor	Gebruiker
Preconditions	Er zitten filters in de filterset
Postconditions	De filterset is opgeslagen
Main Success Scenario	<ol style="list-style-type: none"> 1. De gebruiker drukt op de sla filterset op knop 2. Het systeem slaat de filterset op
Extensions	-

ID	UC.5
Requirements	R.20
Title	Zet continue updates aan
Description	De gebruiker zet continue updates aan
Primary Actor	Gebruiker
Preconditions	De continue updates optie staat uit
Postconditions	De notifications worden continu geupdate
Main Success Scenario	<ol style="list-style-type: none"> 1. De gebruiker selecteert de check box voor continue updates 2. Het systeem begint om de notifications continu te updaten
Extensions	-

ID	UC.6
Requirements	R.19
Title	Verwijder alle notificaties
Description	De gebruiker verwijdert alle notificaties
Primary Actor	Gebruiker
Preconditions	Er bestaan notificaties
Postconditions	Er bestaan geen notificaties meer
Main Success Scenario	<ol style="list-style-type: none"> 1. De gebruiker drukt op de verwijder knop 2. Het systeem verwijdert alle notificaties
Extensions	-

C.4 Technisch Ontwerp



Figuur C.4.1 Klassendiagram

Figuur C.4.1 laat het klassendiagram van de android app zien. Het bestaat uit twee delen: de app zelf en de server die de filter set bevat en de notifications verstuurt naar de app.

De app geeft informatie weer door middel van forms. TappMainForm is de form die de andere forms bevat. Onder de Notifications Tab (nummer 1 in figuur C.4.2) zit de NotificationsPanel waarin de TappNotificationsMainForm zit. Deze bevat de FilterSetPanel

en de NotificationsDataViewPanel waarin respectievelijk TappNotificationsFilterSetForm (nummer 2) en TappNotificationsDataViewForm (nummer 3) zitten.

TappNotificationsFilterSetForm is verantwoordelijk voor het ophalen, bewerken en opslaan van de filter set die gebruikt zal worden door de server. In grdFilters (nummer 4) worden de filter namen weergegeven en of ze momenteel actief zijn of niet. Wanneer op een cell in de Actice column wordt gedrukt, wordt dat filter omgezet naar van actief naar inactief of andersom en wordt de verandering naar de server gestuurd.

Wanneer op btnLoad (nummer 9) geklikt wordt wordt de filter set opgehaald van de server en wanneer op btnSave (nummer 6) wordt geklikt wordt de filter set opgeslagen op de server.

Als btnNewFilter (nummer 7) of btnEdit (nummer 5) geklikt worden wordt TappNotificationsEditFilterDialog geopend, zoals te zien is in figuur 4.3. In het geval van btnEdit moet een filter geselecteerd zijn en wordt de informatie van die filter geladen in de dialog.

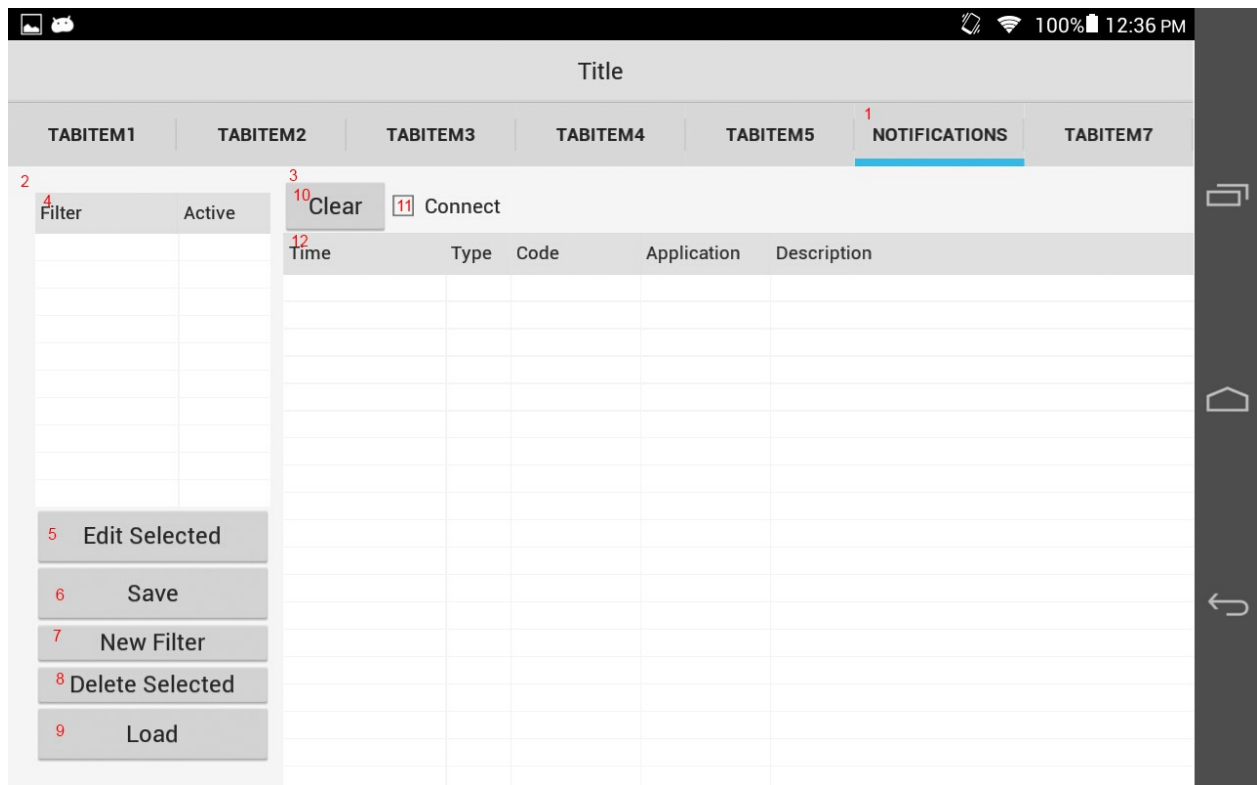
Wanneer op btnDelete wordt geklikt, wordt het geselecteerde filter verwijderd.

TappNotificationsDataViewForm is verantwoordelijk voor het weergeven van de binnenkomende notifications. Wanneer cbxConnectClient (nummer 11) wordt gechecked, wordt er een connectie met de server opgezet welke continu Notifications binnenhaalt. Deze Notifications worden vervolgens weergegeven in grdNotifications (nummer 12). Wanneer op btnClearNotifications wordt gedrukt, worden de Notifications verwijderd.

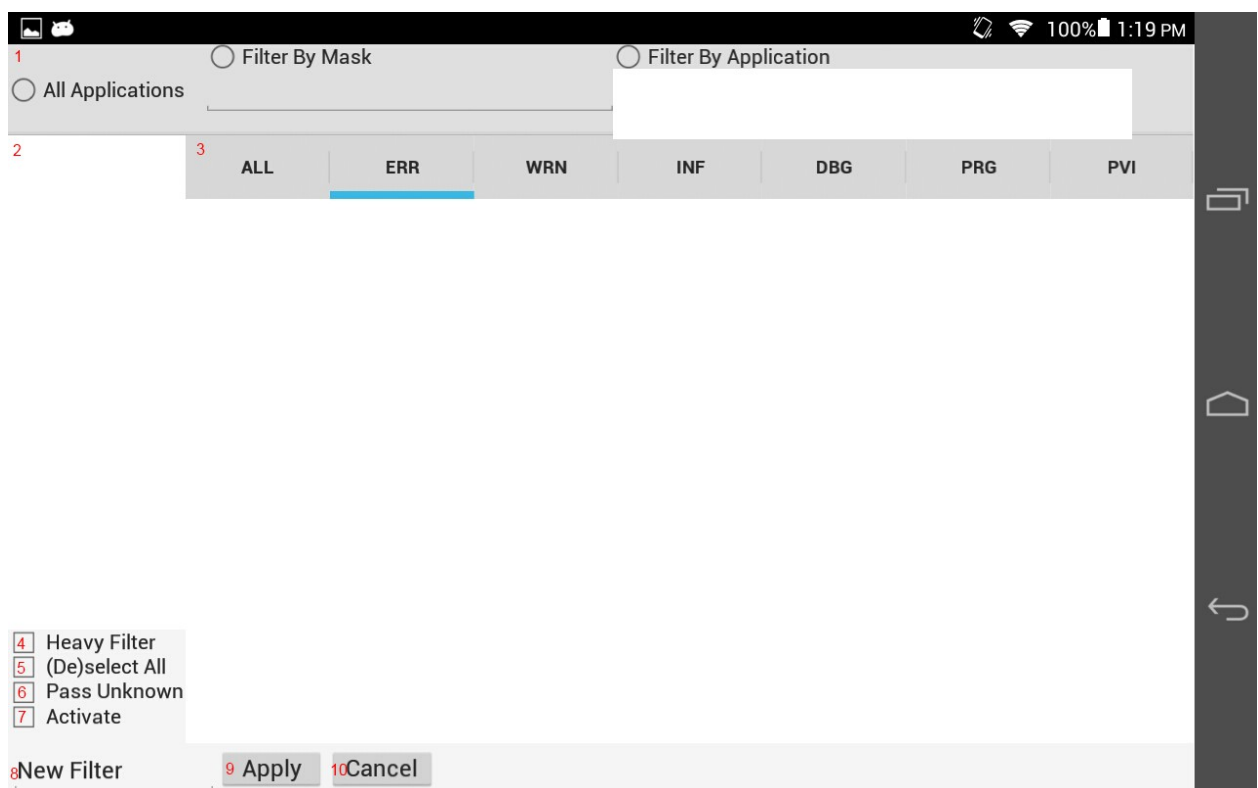
Figuur C.4.3 laat TappNotificationsEditFilterDialog zien. Hierin kan een al bestaande filter set worden bewerkt of een nieuwe worden aangemaakt. Als eerste kan de gebruiker kiezen om te filteren op een bepaalde applicatie (nummer 1), vervolgens kan gekozen worden welk type notifications doorgelaten mogen worden (nummer 2) en tot slot welke specifieke Notifications wel of niet doorgelaten mogen worden (nummer 3).

Als de filter veel vraagt van het systeem wordt moet het aangegeven worden als 'heavy' (nummer 4). Wanneer (de)select all (nummer 5) wordt geselecteerd moeten alle opties van de specifieke Notifications worden geselecteerd of ongeselecteerd. Wanneer pass unknown (nummer 6) wordt geselecteerd moet het filter onbekende notifications doorlaten. Wanneer Activate (nummer 7) wordt geselecteerd, moet het filter direct geactiveerd worden wanneer het aan de filter set wordt toegevoegd.

Wanneer op Apply (nummer 9) wordt gedrukt, wordt het filter toegevoegd aan de filter set met de naam die is opgegeven (nummer 8). Wanneer op Cancel (nummer 10) wordt gedrukt gebeurt er niets.



Figuur C.4.2 Scherm in app



Figuur C.4.3 Edit Filter Dialog

In tabel C.4.1 worden de sequence diagrammen uit bijlage C.A kort beschreven. Voor een gedetailleerdere beschrijving, zie bijlage C.A.

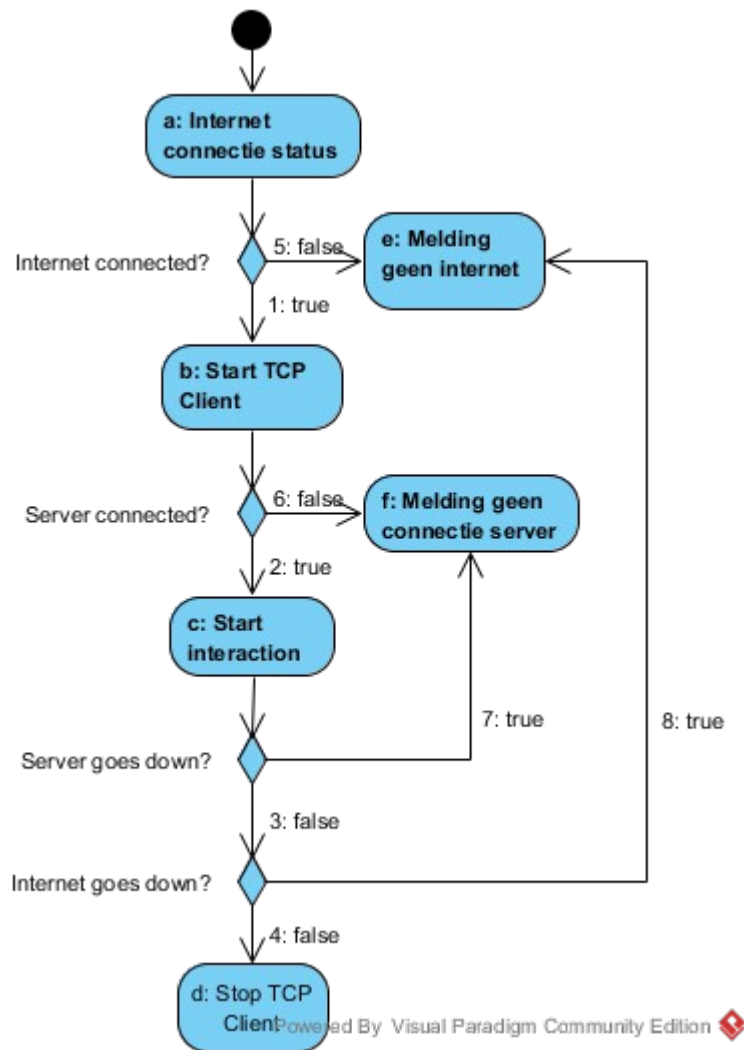
Figuur	Beschrijving
A.1	Laat UC.01, filter selectie, zien. Hierin wordt een filter geactiveerd of gedeactiveerd en de gewijzigde filter set naar de server verstuurd.
A.2	Laat UC.02, filter toevoegen, en UC.03, filter bewerken, zien. Hierin wordt de edit filter dialog geopend om een nieuw filter te maken of een bestaande filter te bewerken en de gewijzigde filter set naar de server verstuurd.
A.3	Laat UC.04, filter set opslaan, zien. Hierin wordt een command word verstuurd die zegt dat de server de huidige filter set moet opslaan in haar xml bestand.
A.4	Laat UC.05, zet continue updates aan, zien. Hierin wordt een thread gestart die continu Notifications van de server ophaalt en weergeeft in de app.
A.5	Laat UC.06, verwijder alle notificaties, zien. Hierin worden de notifications verwijderd van de app.
A.6	Laat zien hoe een connectie wordt opgezet met de server.
A.7	Laat zien hoe de connectie met de server wordt stopgezet.
A.8	Laat zien hoe een TStringGrid wordt leeg gemaakt.
A.9	Laat zien hoe de filter set in de grid wordt gezet.
A.10	Laat zien hoe een specifiek filter in een row van de grid wordt gezet.

Tabel C.4.1 Beschrijving sequence diagrammen

C.5 Test

C.5.1 Testen Connectie

Om de connectie te testen heb ik een testpad opgesteld, zoals te zien is in figuur C.5.1. Deze testpaden leiden tot de scenario's in tabel C.5.1. Voor de fysieke testgevallen en het resultaat, zie bijlage C.B.1.



Figuur C.5.1 Testpaden Client-Server Connection

#	Scenario	Testpad
1	Succesvol de client opstarten en stoppen	1-2-3-4
2	Internet gaat neer voor het stoppen van de client	1-2-3-8
3	Server gaat neer voor het stoppen van de client	1-2-7

4	Server is neer voor de link kan worden opgezet	1-6
5	Internet is neer voor de client kan worden opgestart	5

Tabel C.5.1 Scenarios Client-Server Connection

C.5.2 Testen CRUD Filter Set

De filter set die gebruikt wordt is afhankelijk van drie andere soort entiteiten. Het bestaat uit filters, wanneer een filter wordt gemaakt, bewerkt of verwijderd moet de filter set geupdate worden. De filter set wordt geload uit een xml bestand waarin het ook weer wordt opgeslagen. Om een filter te kunnen maken moet er gebruikt gemaakt worden van bekende Notifications, dus die worden telkens gelezen van de server kant wanneer een nieuwe filter wordt aangemaakt of een al bestaande filter wordt gewijzigd.

Voor het testen wordt gebruik gemaakt van de methoden in tabel C.5.2, in de gegeven volgorde. In die tabel staat ook aangegeven wat voor interactie de methodes hebben met de verschillende entiteiten.

Zie bijlage C.B.2 voor het fysieke testontwerp en het resultaat.

#	Method	Filter	Filter Set	Known Notifications	Filter Set XML
1	clkLoad		C		R
2	clkNew	C	U	R	
3	clkEdit	U	R, U	R	
4	clkDelete	D	U		
5	CellSelected	U	R, U		
6	clkSave		R		U

Tabel C.5.2 CRUD-matrix Filter Set

C.6 Verbeterpunten

Tijdens het testen ben ik enkele dingen tegengekomen die niet in de requirements stonden, maar die wel in het originele systeem werkten, specifiek in de edit filter dialog.

In dat dialog moet een masker mee kunnen worden gegeven om op te filteren, maar omdat die een type, `TpvTextMatchList` van `PV.Util.TextMatchList`, nodig heeft die ik nog niet heb omgezet naar Android had ik die er uit gecommentar'd.

De (de)select checkbox werkt ook niet. Deze zou alle specifieke filters moeten selecteren wanneer het aangevinkt wordt en ze deselecteren wanneer het uit gevinkt wordt.

Ik heb alleen in de all tab van de dialog de specifieke filters gezet, maar ze moeten ook nog gescheiden worden in types en in de bijbehorende tabbs van die types gezet worden.

De buttons van de filter set view en de edit filter dialog zitten soms niet op de plaats waar ik ze had neergezet in de editor. Ik weet niet waar dit door komt.

C.A - Sequence Diagrams

In deze paragraaf wordt de code van de use cases uit hoofdstuk 3 beschreven. Hierin zitten sommige methodes die niet direct te maken hebben met de use cases of welke de diagrammen onoverzichtelijk zouden maken als deze ook volledig uitgewerkt zouden worden in dezelfde diagram, deze worden beschreven in de volgende paragraaf.

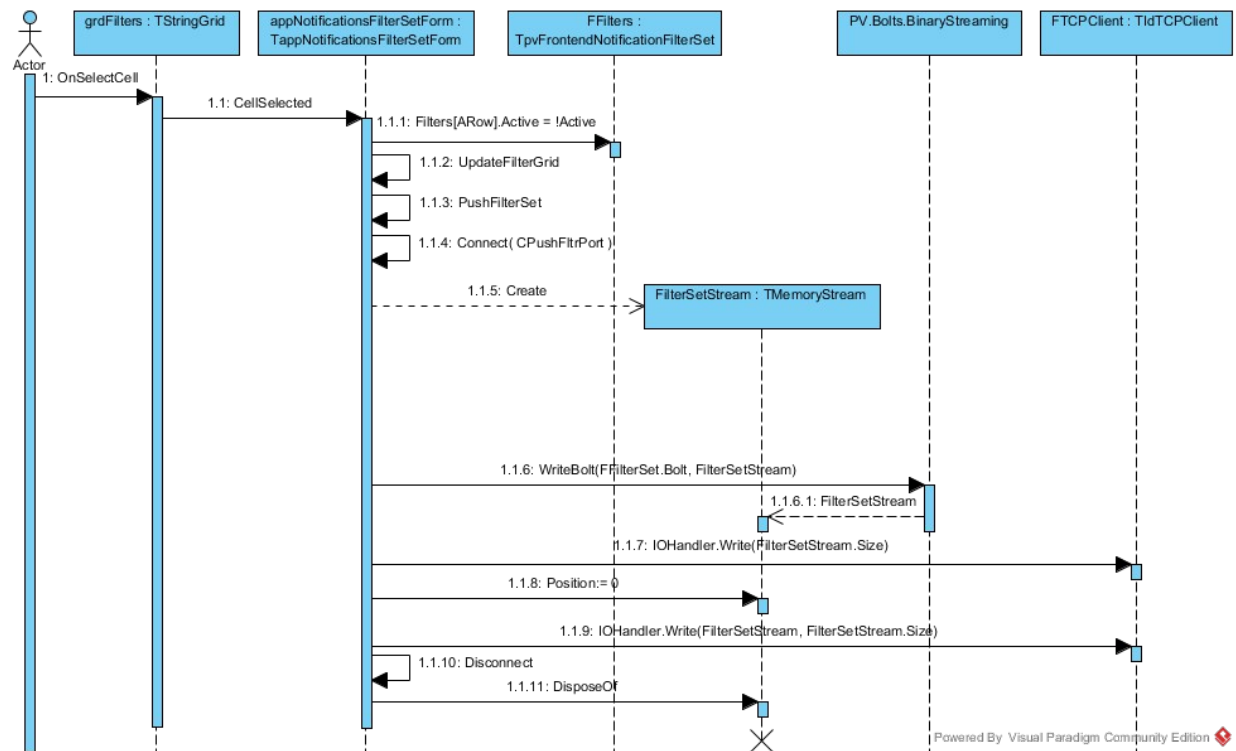
Omdat de server voornamelijk bestaat uit push en pull procedures die overeenkomt met de push en pull procedures van de app heb ik deze niet meegenomen in deze paragraaf. Ik beschrijf de push en pull procedures van de app ook ieder maar een keer, omdat zij er hetzelfde uitzien. Het enige verschil is wat wordt omgezet naar een stream.

Figuur C.A.1 laat het sequentie diagram voor UC.1, Filter Selectie, zien. Wanneer een cell in `grdFilters` wordt geselecteerd wordt het `OnSelectCell` event geactiveerd, deze roept de `CellSelected` procedure van `TappNotificationsFilterSetForm` aan.

In `CellSelected` wordt de status van de filter die verbonden is aan die row omgezet, als het geactiveerd is wordt het gedeactiveerd of als het gedeactiveerd is wordt het geactiveerd. Hierna wordt de grid geupdate zodat de verandering zichtbaar is. Tot slot wordt de `PushFilterSet` procedure aangeroepen.

Deze zet de connectie op met de server. Vervolgens wordt de filter set omgezet naar een stream die verzonden zal worden naar de server.

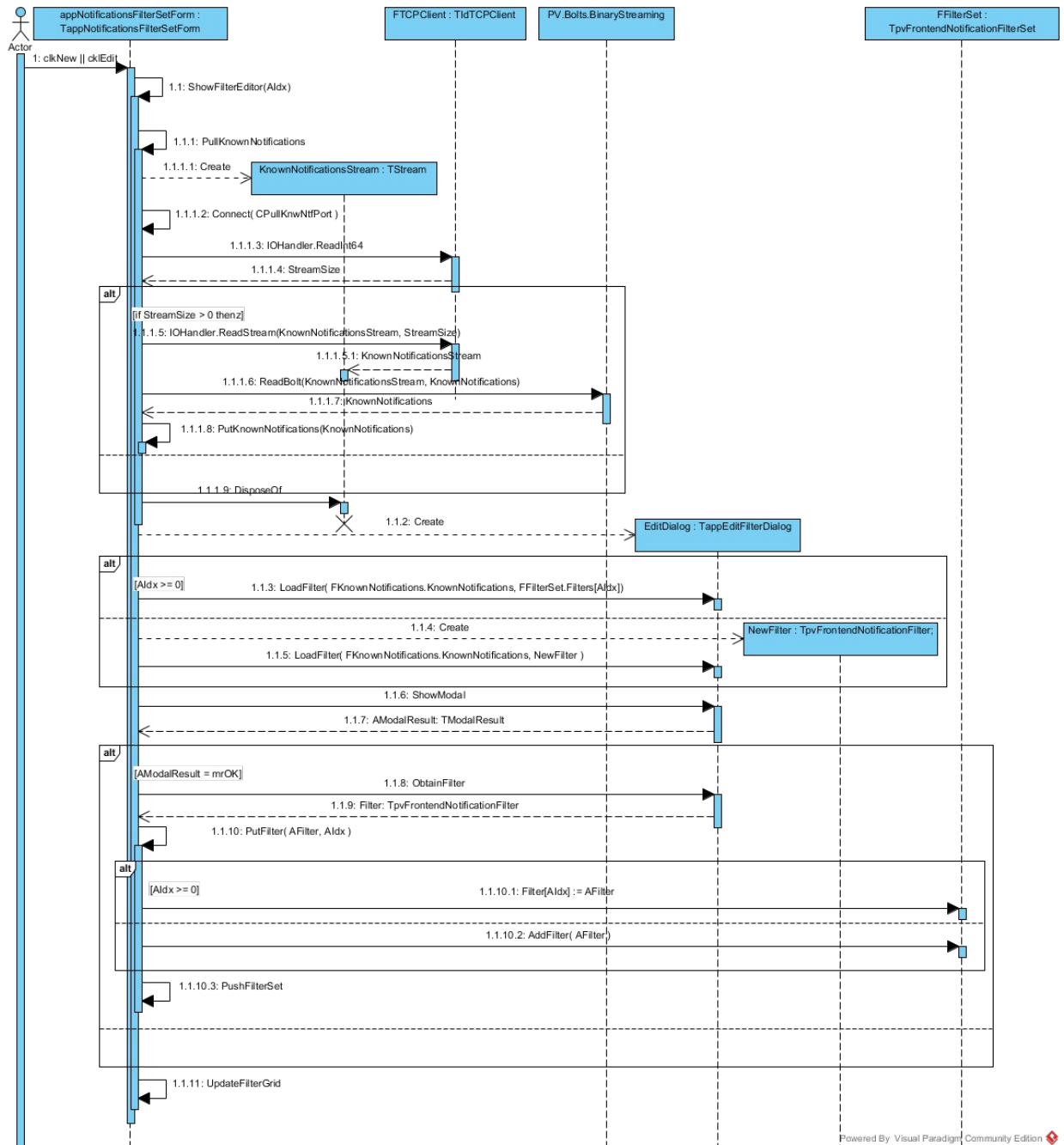
Om de stream te kunnen ontvangen, moet de server eerst de grootte van de stream ontvangen. Wanneer die grootte is verstuurd moet de position van de stream naar 0 worden gezet. Daarna kan de stream zelf worden gestuurd. Deze wordt op de server terug omgezet naar een Filter Set. Tot slot wordt de link met de server verbroken en het geheugen van de stream vrijgegeven.



Figuur C.A.1 UC.01 Filter Selectie Client Side Sequence Diagram

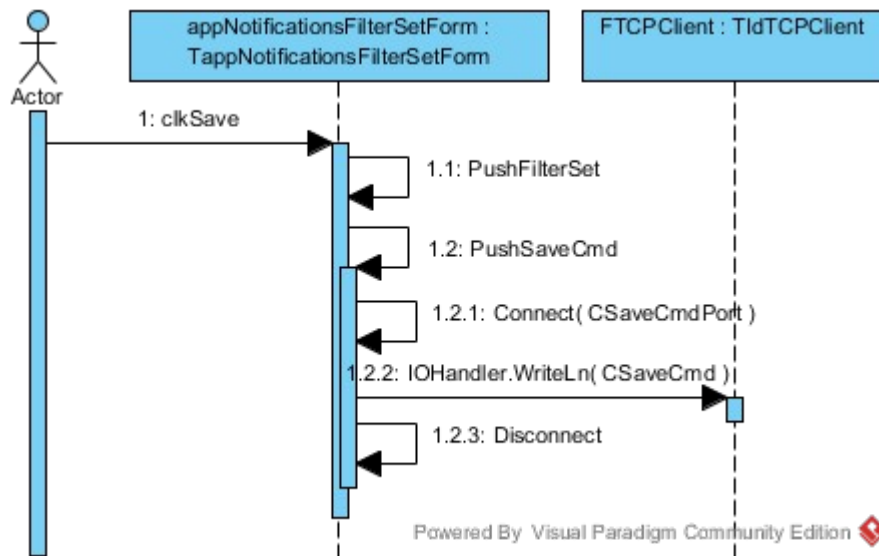
Figuur C.A.2 laat het sequentie diagram voor UC.03, filter toevoegen, en UC.04, filter bewerken zien. Deze werken bijna hetzelfde, maar als een filter wordt toegevoegd met `clkNew` is `Aldx` gelijk aan `-1` en als een filter wordt bewerkt met `clkEdit` is `Aldx` gelijk aan de plaats van de geselecteerde filter in de filter set.

Om een nieuwe filter te maken of een bestaande te bewerken moeten de bekende notifications opgehaald worden met de procedure `PullKnownNotifications`. Deze zet een connectie op met de server. De server werkt in dit geval zoals de client in UC.01. Het zet de `KnownNotification` om naar een stream en verstuurt eerst de grootte van de stream, voordat de stream zelf wordt verstuurd. De app ontvangt ze ook in die volgorde en zet vervolgens de stream terug om naar `KnownNotifications`. Hierna wordt een dialog geopend waarin de `KnownNotifications` en een nieuw filter of het geselecteerde filter worden gestopt. Met deze dialog kan de filter bewerkt worden en als er op ok wordt gestopt wordt de filter in de filter set geplaatst. Tot slot wordt de filter set verstuurd naar de server gestuurd.



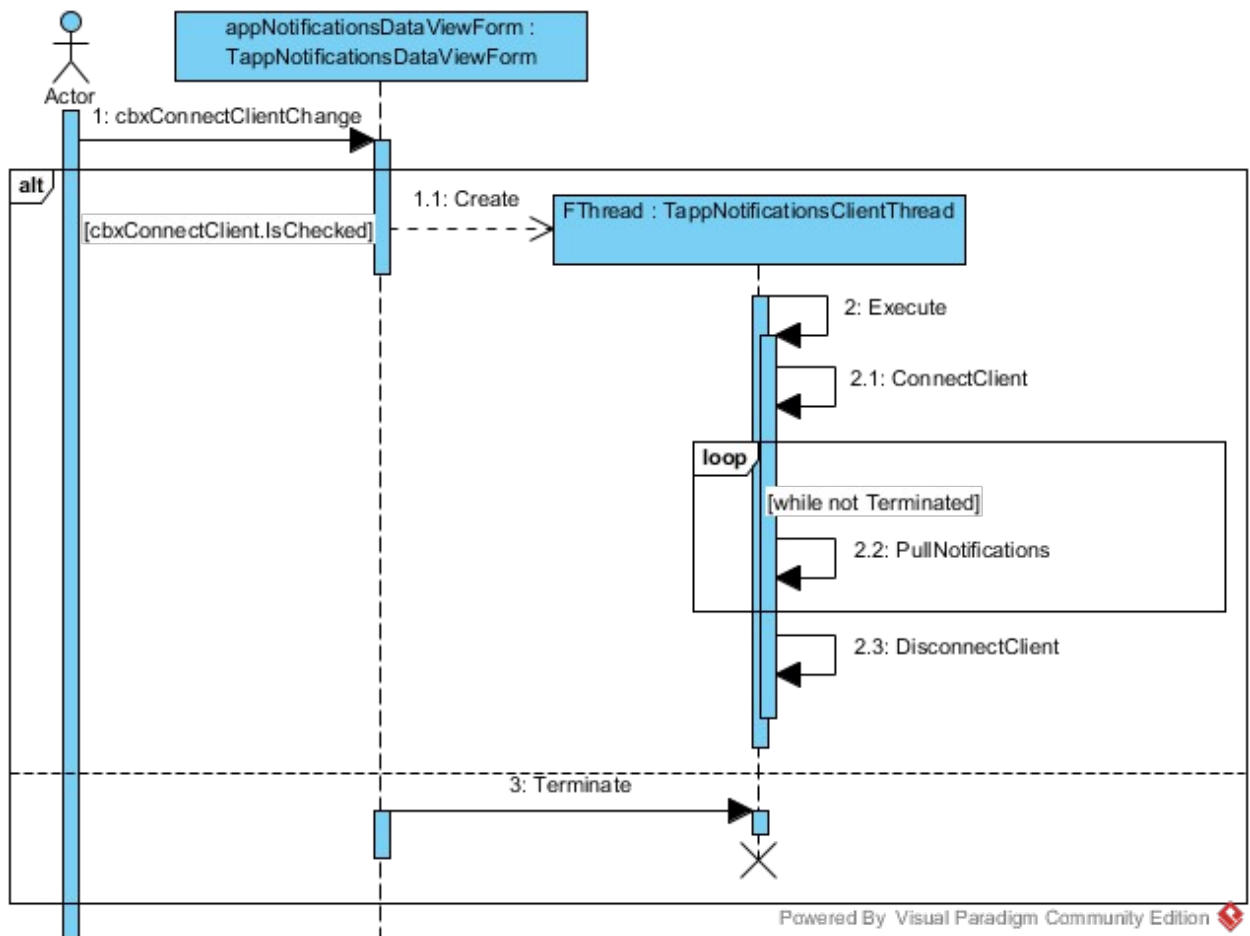
Figuur C.A.2 UC.02 Filter Toevoegen en UC.03 Filter Bewerken Sequence Diagram

Figuur C.A.3 laat UC.04, filterset opslaan, zien. Hierin wordt eerst de filter set verstuurd naar de server, op dezelfde manier als in UC.01. Vervolgens wordt opnieuw een connectie gemaakt met de server, maar dan op een andere port. Over deze connectie wordt de save command naar de server gestuurd, welke vervolgens de filter set in een xml bestand opslaat. Tot slot wordt de connectie verbroken.



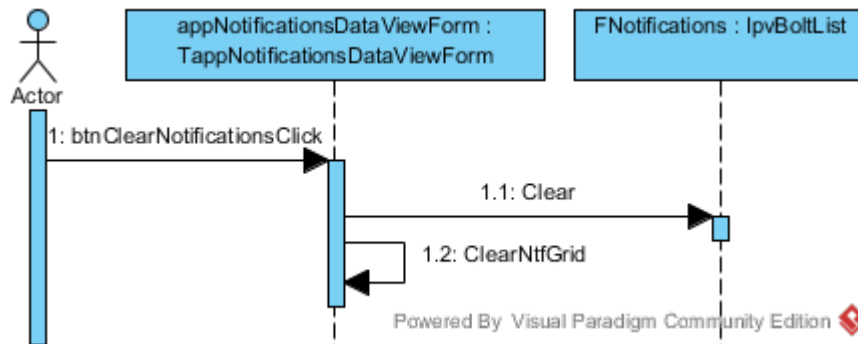
Figuur C.A.3 UC.04 Filterset opslaan Sequence Diagram

PullNotifications werkt als PullKnownNotifications van TappNotificationsFilterSetForm. Figuur C.A.4 laat UC.05, zet continue updates aan, zien. Hierin wordt een checkbox geklikt. Als het aangevinkt wordt maakt de app een nieuwe thread aan. Die thread zet een connectie op met de server en haalt voortdurend nieuw Notifications op. Als de checkbox wordt uitgevinkt wordt de thread geterminate en de connectie met de server stop gezet.



Figuur C.A.4 UC.05 Zet Continue Updates Aan Sequence Diagram

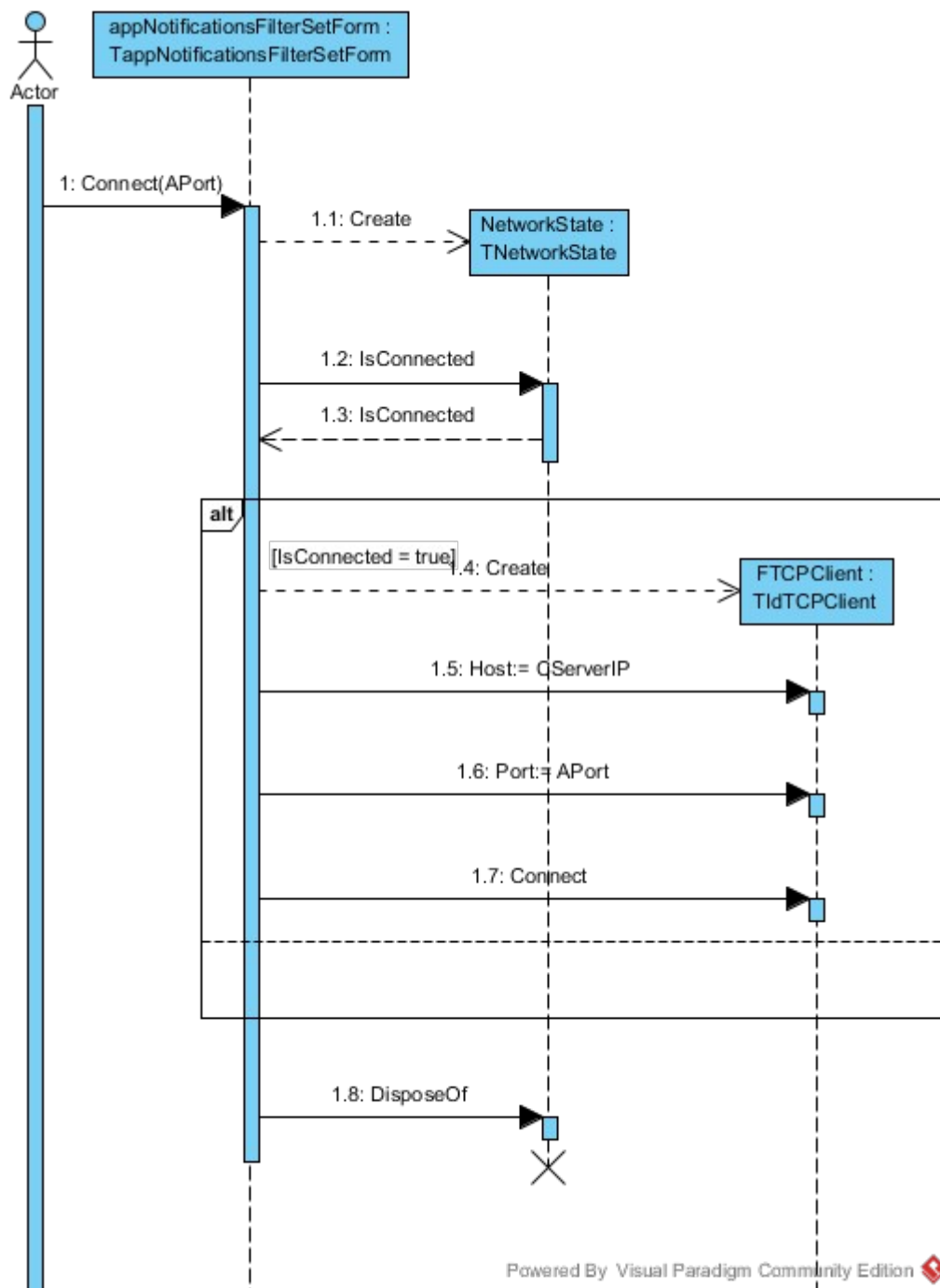
Figuur C.A.5 laat UC.06, verwijder alle notificaties, zien. Hierin worden de lijst waar de Notifications in zitten en de TStringGrid waarin ze zijn weergegeven leeg gemaakt.



Figuur C.A.5 UC.06 Verwijder Alle Notificaties Sequence Diagram

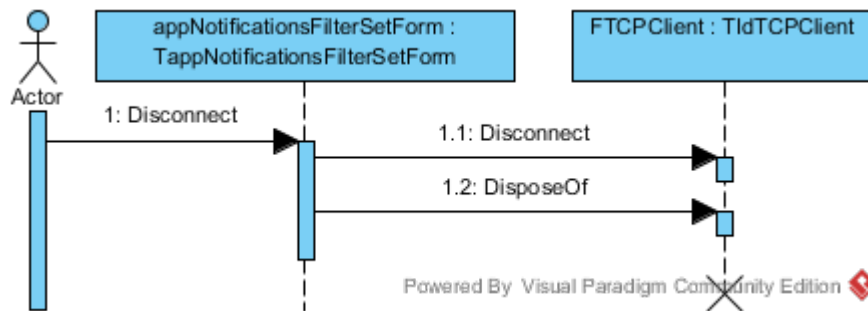
Figuur C.A.6 laat zien hoe een connectie wordt opgezet. Eerst wordt een NetworkState aangemaakt welke gebruikt wordt om de internet verbinding te controleren. Als er een internetverbinding is wordt een nieuwe TIdTCPClient aangemaakt. Deze krijgt de server IP adres en de te gebruiken port mee, voordat het de opdracht krijgt om een connectie te maken.

Wanneer de connectie is opgezet wordt de NetworkState vrijgegeven.



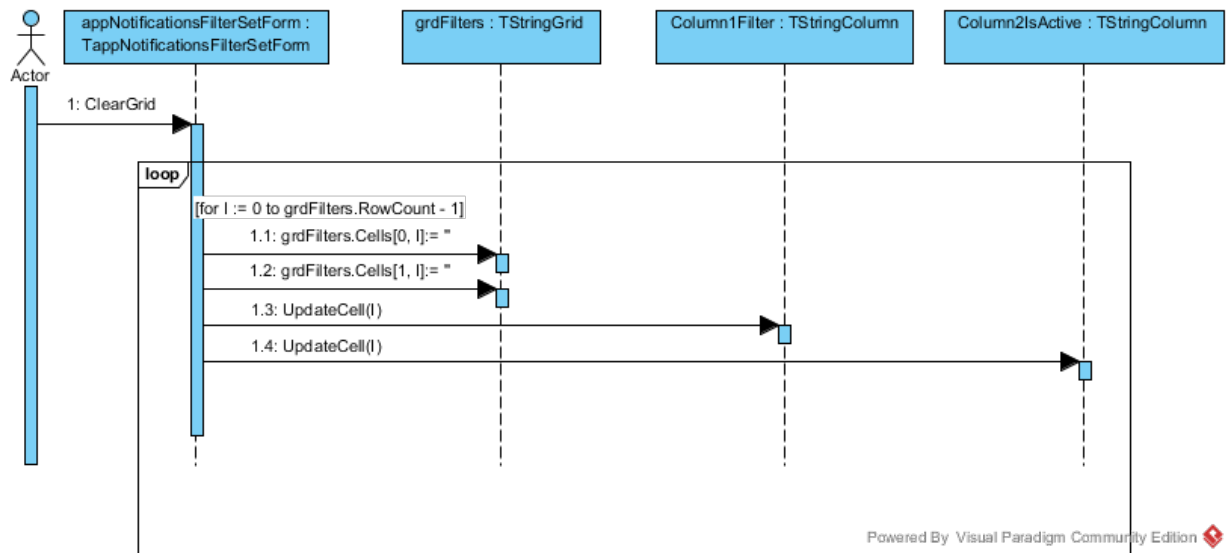
Figuur C.A.6 Connecting TCP Client

Figuur C.A.7 laat zien hoe de connectie met de server verbroken wordt. Eerst wordt de Disconnect procedure van TIdTCPClient aangeroepen om het vervolgens vrij te geven.



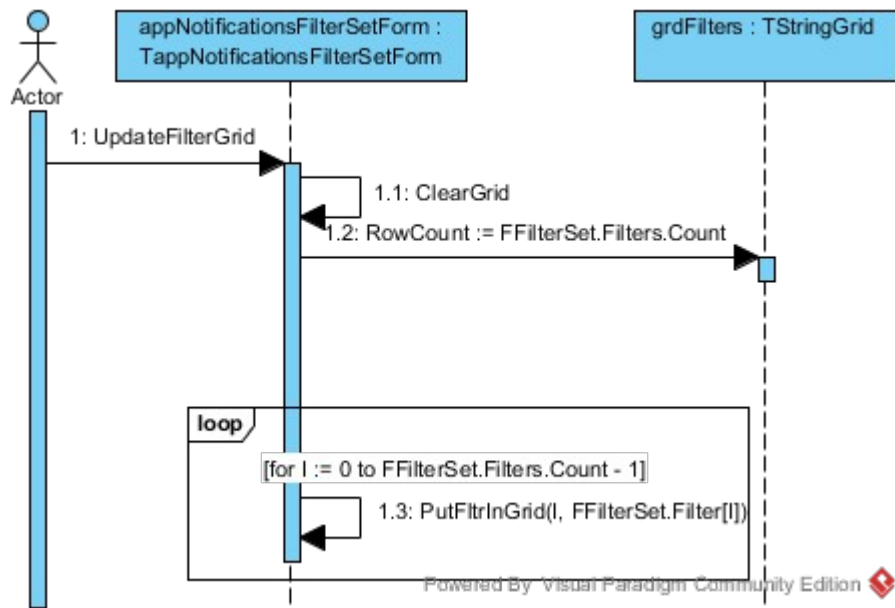
Figuur C.A.7 Disconnecting TCP Client

Figuur C.A.8 laat zien hoe een TStringGrid wordt leeggemaakt. Eerst worden de strings van alle cellen leeggemaakt. Vervolgens worden de cellen via de betreffende TStringColumn geupdate.

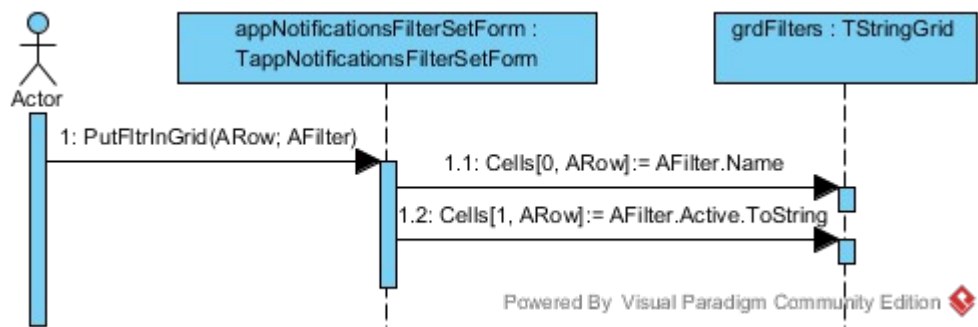


Figuur C.A.8 Clear Grid

Figuur C.A.9 en C.A.10 laten zien hoe een TStringGrid gevuld wordt, in dit geval grdFilters van TappNotificationsFilterSetForm. Eerst wordt het leeggemaakt, vervolgens krijgt het een rowcount die gelijk is aan de hoeveelheid entiteiten die er in gestopt moeten worden. Vervolgens wordt per row informatie in iedere column van de grid gestopt die bij een enkele entiteit hoort.



Figuur C.A.9 Update Filter Grid



Figuur C.A.10 Put filter in grid

C.B - Test Uitvoering

C.B.1 Testen Connectie

#	Acties	Verwacht Resultaat
1	<ol style="list-style-type: none">1. Zet wi-fi aan2. Start TCP server3. Check connect checkbox voor Notifications4. Uncheck checkbox	De notifications komen binnen wanneer de checkbox is checked en stoppen met binnenkomen totdat checkbox is unchecked.
2	<ol style="list-style-type: none">1. Zet wi-fi aan2. Start TCP server3. Check connect checkbox voor Notifications4. Zet wi-fi uit	Er wordt een bericht weergegeven dat er geen internet is.
3	<ol style="list-style-type: none">1. Zet wi-fi aan2. Start TCP server3. Check connect checkbox voor Notifications4. Stop server	Er wordt een bericht weergegeven dat de connectie met de server weg is.
4	<ol style="list-style-type: none">1. Zet wi-fi aan2. Stop TCP server3. Check connect checkbox voor Notifications	Er wordt een bericht weergegeven dat er geen connectie met de server opgezet kan worden.
5	<ol style="list-style-type: none">1. Zet wi-fi uit2. Check connect checkbox	Er wordt een bericht weergegeven dat het internet uitstaat.

Tabel C.B.1 Fysieke Testen Connectie

#	P/F	Notes
1	P	
2	P	Melding blijft zichzelf herhalen
3	F	
4	F	Ook geen exception, maar app loopt wel vast
5	P	

Tabel C.B.2 Pass (P) of Fail (F) van connectie test

C.B.2 Testen CRUD Filter Set

#	Actie	Verwacht Resultaat
1	Laad Filter Set	De filter set aan de client side komt overeen met de filter set aan de server side
2	Voeg nieuw filter toe	De nieuwe filter zit in de filter set aan de client side
3	Bewerk het filter	De nieuwe filter is gewijzigd aan de client side
4	Verwijder een ander filter	De filter set aan de client side bevat de verwijderde filter niet meer
5	Activeer een filter, start notifications connectie	Een filter is geactiveerd en de notifications die die filter doorlaat komen binnen aan de client side
6	Sla filter set op	Het filter set is opgeslagen in het xml bestand aan de server side

Tabel C.B.3 Fysieke Testen CRUD Filter Set

#	P/F	Notes
1	P	
2	P	Zichtbaar aan de server side, maar niet aan de client side totdat ik de edit dialog een tweede keer heb geopend.
3	P	
4	P	
5	P	
6	P	

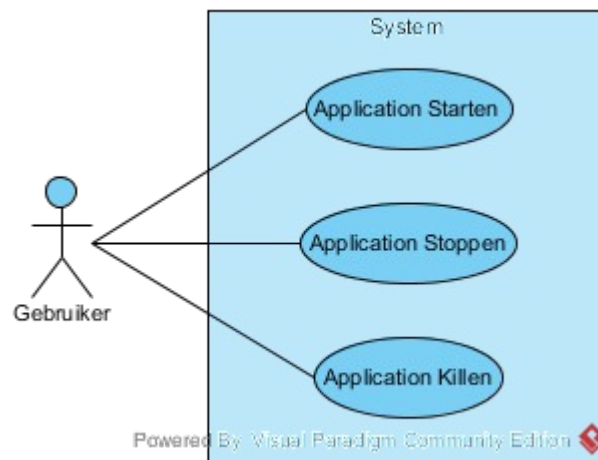
Tabel C.B.4 Pass (P) of Fail (F) van filter set CRUD test

Bijlage D - Iteratie 2

D.2 Requirements

Code	Beschrijving	MoSCoW
R.24	De app moet een treeview van de applications bevatten	Must
R.25	De status van de applications moet weergegeven worden in een remarks kolom	Should
R.26	Als een application meerdere statussen bevat moeten deze afwisselend weergegeven worden	Should
R.27	De gebruiker moet een application kunnen starten	Could
R.28	De gebruiker moet een application kunnen stoppen	Could
R.29	De gebruiker moet een application kunnen killen	Could
R.30	De app moet een detail view van een geselecteerde application bevatten	Would

D.3 Functioneel Ontwerp



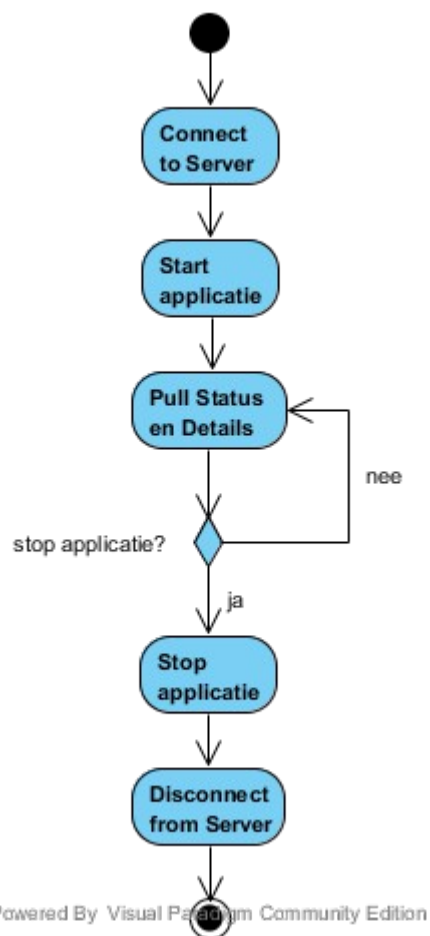
Figuur 3.1 Use Case Diagram

ID	UC.7
Requirements	R.27
Title	Application Starten
Description	De gebruiker start een application op de server
Primary Actor	Gebruiker
Preconditions	-
Postconditions	De application is opgestart
Main Success Scenario	
Extensions	

ID	UC.8
Requirements	R.28
Title	Application Stoppen
Description	De gebruiker stopt een application op de server
Primary Actor	Gebruiker
Preconditions	De application draait
Postconditions	De application is gestopt
Main Success Scenario	

Extensions	
-------------------	--

ID	UC.9
Requirements	R.29
Title	Application Killen
Description	De gebruiker stopt een application op de server, ongeacht of het bezig is
Primary Actor	Gebruiker
Preconditions	De application draait
Postconditions	De application is gestopt
Main Success Scenario	
Extensions	



Figuur 3.1 Activity Diagram Iteratie 2

Bijlage E - Advies

E.1 Obstakels en oplossingen

E.1.1 Android devices

Niet alle Android devices zijn bruikbaar om te debuggen. om toch op android te kunnen debuggen kan de emulator gebruikt worden en in ieder geval ook de Huawei MediaPad M1 8.0 tablet. Als debuggen niet hoeft kan de .apk file direct geïnstalleerd worden op de device.

Voor het het maken van de network state check heb ik de android library gebruikt, hiervoor heb ik ook onderscheid moeten maken in wanneer dat gebruikt wordt, door de volgende constructie:

```
{ $ifdef android }  
    //Android specifieke code  
{ $endif }
```

(docwiki.embarcadero.com/RADStudio/XE7/en/Conditional_compilation_%28Delphi%29 11-3-2015)

E.1.2 GUI design

De GUI design komt niet altijd overeen met de uiteindelijke weergave op de device. Figuur E.1.1 laat het ontwerp zien voor de notifications view. Deze bestond uit een layout waarin een connect checkbox en een clear button zitten, daaronder is een TStringGrid waar de informatie van de strings uiteindelijk in geschreven worden.

Figuur E.1.2 laat zien dat de layout waar de button en checkbox in zitten te dun was en daarom was de onderkant van beide deels weggesneden. Ook was de grid niet zichtbaar en ik weet niet hoe dat kwam.

Toen ik eerst de layout wilde fixen heb ik het wat dikker gemaakt in het ontwerp. Het result daarvan is te zien in figuur E.1.3, hierin is te zien dat de grid ineens zichtbaar was en dat de checkbox en button niet meer zichtbaar waren. Ik weet niet hoe het kwam dat de grid wel zichtbaar werd, maar ik vermoedde dat de button en checkbox niet meer zichtbaar waren omdat de layout langer werd en de button en checkbox buiten het scherm werden gegenereerd.

Dat werd bevestigd toen ik met de align property in de object controller de button en checkbox naar links verplaatste, zoals te zien in figuur E.1.4.

Vanaf toen heb ik besloten om voornamelijk de volgende align properties te gebruiken, zodat het altijd zichtbaar zal zijn:

- Client vult de parent van de control. Als er andere controls aan zijkanten gepind zijn, vult deze control de rest van de parent.
- MostLeft zet de control aan de uiterst linkerkant van de parent en vergroot de hoogte om de parent te vullen.
- MostTop zet de control aan de uiterste bovenkant van de parent en vergroot de breedte om de parent te vullen.
- Left zet de control aan de linkerkant, naast controls die daar al gepind zijn, en vult de hoogte van de parent.
- Top zet de control aan de bovenkant, onder controls die daar al gepind zijn, en vult de breedte van de parent.

(docwiki.embarcadero.com/Libraries/XE7/en/FMX.Types.TAlignLayout 4-3-2015)

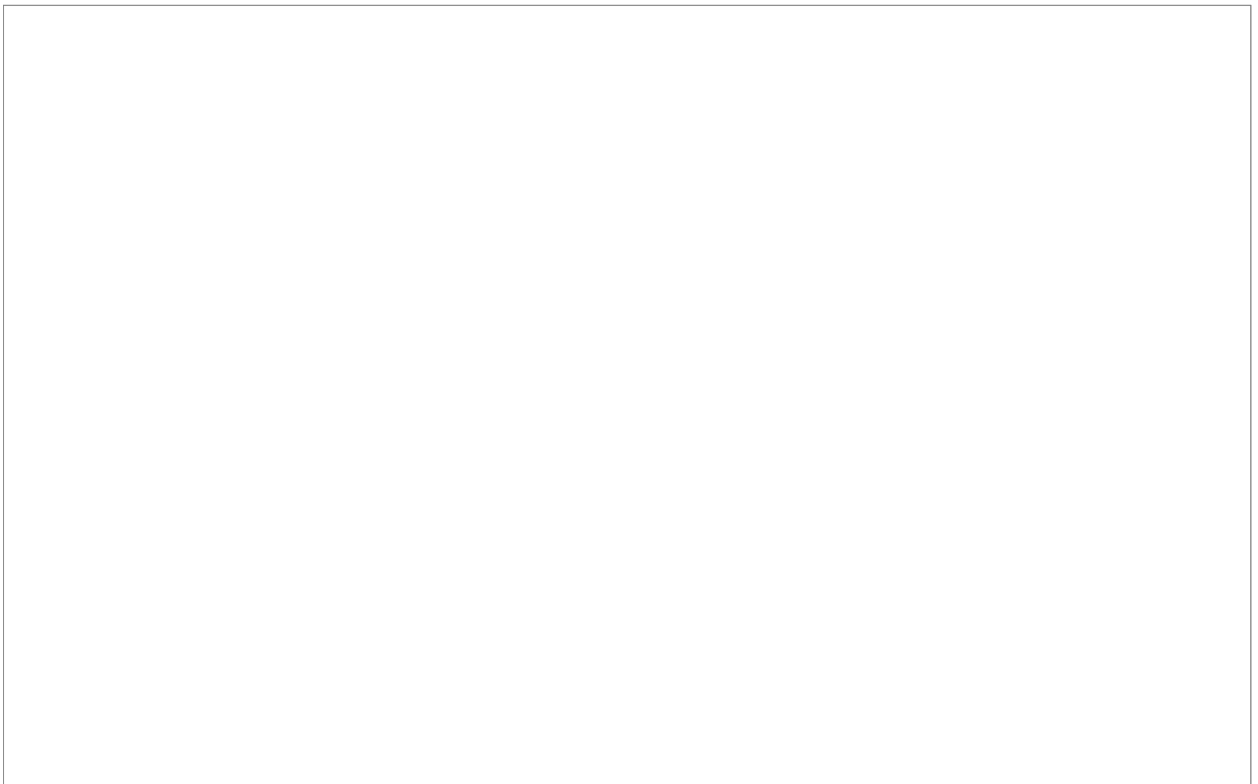
Wanneer een panel aan de onder- of rechterkant een andere panel heeft, gebruikte ik soms ook Bottom en Right alignments, omdat het dat niet kan “ontsnappen” buiten het scherm.

[illegible]

Figuur E.1.1 Initiele Notifications View Design



Figuur E.1.2 Eerste poging Notifications View



Figuur E.1.3 Tweede poging Notifications View



Figuur E.1.4 Derde poging Notifications View

E.1.3 Delphi 2007

Ik kon geen anonymous procedure gebruiken in class procedure
TThread.Synchronize(AThread: TThread; AMethod: TThreadMethod) en ook geen procedure
die een parameter gebruikte, dus heb ik een tmp attribuut in de type aangemaakt welke
wordt gebruikt in een procedure die geen gebruik maakt van parameters.

E.1.4 Delphi XE7

In Delphi XE7 ben ik wat bugs tegengekomen.

Soms kon een unit niet gevonden worden tijdens het compilen of builden, wanneer dit
gebeurde opende ik het bestand, savede het en probeerde ik het opnieuw of sloot en
heropende ik het project.

Soms lukte het linken met Android Debug Bridge niet, wanneer dat gebeurde herstartte ik
RAD XE7 of uninstallede ik de oude versie van de app.

Soms wanneer er al een versie van de app geïnstalleerd was op de tablet en ik probeerde
opnieuw te debuggen kreeg ik soms problemen doordat de nieuwe versie de saved instance
state van de oude versie probeerde te gebruiken, om dit op te lossen uninstallede ik de oude
versie van de app voordat ik de nieuwe versie installeerde.

Regelmatig kreeg ik error meldingen die verdwenen wanneer ik compilede.

E.1.5 Code Prime Vision en Third Parties

De huidige code van Prime vision is niet altijd compatibel met XE7 of Android, hierom heb ik in het begin branch `{$IFDEF VER280}` gebruikt om aan te geven wat met XE7 gecompileerd moet worden. Om het backwards compatibel te houden heb ik `{$ELSE}` gebruikt voor de oude code. Later heb ik verder onderscheid gemaakt in wanneer iets gecompileerd moet worden.

Van de third party code heb ik voornamelijk gebruik geprobeerd te maken van ContinUIT. Wat ik hiervan wilde gebruiken compileerde niet voor Android en daarom heb ik alternatieven gezocht binnen Delphi zelf. Bijvoorbeeld, voor het maken van de NotificationDataView gebruikte de originele applicatie TDataSetList van ContinUIT, terwijl ik in de plaats daarvan een TStringGrid gebruikte om dezelfde informatie weer te geven.

AnsiString en andere 1-based strings kunnen niet gebruikt worden in een Android applicatie, maar er is een patch die dit oplost op andy.jgknet.de/blog/2013/10/the-return-of-the-byte-strings/ (11-3-2015).

Als ik de patch niet had gebruikt, dan zou ik in alle units die 1-based strings gebruiken aanpassingen hebben moeten uitvoeren. En om het backwards compatibel te houden zou ik een constructie zoals hieronder staat hebben moeten gebruiken:

```
{$ifdef nextgen}
    //0-based strings
{$else}
    //1-based strings
{$endif}
```

Sinds het originele MMS-SMX systeem is gebouwd heeft Delphi een eigen FRefCount toegevoegd voor ARC. Dit heeft conflicten opgeleverd met de FRefCount property die gemaakt is door het software team van MMS. Om dit op te lossen heb ik geprobeerd een constructie te gebruiken die de PV FRefCount alleen gebruikt als AUTOREFCOUNT niet defined is, bijvoorbeeld:

```
{$ifndef autorefcount}
    //FRefCount van Prime Vision
{$endif}
```

Sommige van de oude code gebruikte nog `{$ifend}` directives. Om daar geen waarschuwingen over te krijgen tijdens het compileren heb ik in project options moeten specificeren om dat toe te laten.

([docwiki.embarcadero.com/RADStudio/XE7/en/Legacy_IFEND_\(Delphi\)](http://docwiki.embarcadero.com/RADStudio/XE7/en/Legacy_IFEND_(Delphi)) 11-3-2015)

Ik was tijdelijk vastgelopen op het gebruiken van Indy. Ik converteerde de IpvBoltList waar de notifications in zitten naar een TMemoryStream welke ik via een TIdTCPServer probeerde te versturen. Tijdens het debuggen bleek dat de stream elke keer een nieuwe lijst aan notificaties bevatte, welke ik met WireShark controleerde en zag dat ook elke keer nieuwe werd verstuurt. In de client zag ik dat de grootte van de binnenkomende stream overeenkwam met de grootte van de stream die verstuurt was, maar alleen de eerste lijst van notifications die verstuurt was werd telkens uitgelezen.

Uiteindelijk bleek dat ik de TStream.position elke keer weer op 0 moest zetten om de nieuwe streams te kunnen lezen. Maar ik heb er uiteindelijk voor gekozen om de stream te encapsuleren in een eigen procedure, zoals in Code E.1.1, om de execute procedure van de client thread leesbaar te houden.

```

procedure TappNotificationsClientThread.PullNotifications;
var
  Notifications: IpvBoltList;
  NotificationsStream: TStream;
  StreamSize: Int64;
begin
  StreamSize:= 0;
  NotificationsStream := TMemoryStream.Create;
  Notifications := GBoltValueEnv.NewList;

  try
    StreamSize:= TCPCClient.IOHandler.ReadInt64;
    if StreamSize > 0 then
      begin
        TCPCClient.IOHandler.ReadStream(TStream(NotificationsStream), StreamSize);
        Notifications:= ConvertStreamToBoltList(NotificationsStream);
        Synchronize(procedure begin PutNotifications(Notifications); end);
      end;
    finally
      NotificationsStream.Free;
      Notifications.Clear;
    end;
  end;
end;

```

Code E.1.1 PullNotifications Procedure

Omdat het huidige NotificationReceiver framework niet werkt op Android heb ik ervoor gekozen om die op de server te laten en alleen de FilterSet naar de Android app te versturen. Wanneer ik een verandering in de FilterSet aanbreng, moet die verstuurd worden naar de server, maar hierbij liep ik tegen het probleem aan dat de order van de bytes van de Int64, welke ik nodig heb om de grootte van de binnenkomende stream aan te geven, omgedraaid wordt. Ik heb dit uiteindelijk opgelost door assembly code toe te voegen die ik had gevonden op codeverge.com/embarcadero.delphi.basm/fastest-best-way-to-reverse-byte-order/1096017 (20-4-2015).

Hierna had ik het probleem dat de BoltHeader 1 char te lang was. Dit kwam doordat de procedure WriteS(const Stream : TStream; const S : String) van PV.Bolts.BinaryStreaming op Android deze toevoegde. De Integer daarin kreeg de length van de array van Bytes toegekend, maar het is nu aangepast en het krijgt de length -1 toegekend.

Om meerdere informatiestromen te versturen over een port werd te onoverzichtelijk, dus heb ik iedere stroom een eigen port gegeven. Om dat te doen moet ik bij het creëren van de server meerdere bindings maken:

```

with Server.Bindings.Add do
begin
  IP := CIP;
  Port := CNTfPort;
end;

```

Vervolgens wordt in de server's OnExecute event gecontroleerd met welke port het is verbonden om de bijbehorende procedure uit te voeren, zoals te zien is in Code 1.2.

```

procedure TpvServerFrame.Execute(AContext: TIdContext);
begin
  case AContext.Binding.Port of
    CNTfPort: begin
      PushNotifications(AContext);
    end;
    CPullFiltrPort: begin
      PullFilterSet(AContext);
    end;
    CPushFiltrPort: begin
      PushFilterSet(AContext);
    end;
    CPushKwNtfPort: begin
      PushKnownNotifications(AContext);
    end;
    CSaveCmdPort: begin
      SaveFilterSet(AContext);
    end;
    CPushAppPort: begin
      PushApplications(AContext);
    end;
  end;
end;

```

Code E.1.2 Server's OnExecute methode

Voor de volledige code, zie mijn SVN branch: https://scm.primevisiononline.com/svn/postnl-ncs/MailMatch/views/PostNL-MMS/branches/fb_friso_androidpm

E.2 Alternatieve Ontwikkelomgevingen

Voor de alternatieve ontwikkelomgevingen heb ik gekeken naar welke IDEs gemaakt zijn door de Apple, Google en Windows voor hun respectievelijke platforms.

Ook heb ik gekeken naar verschillende cross-platform IDE's. Tijdens mijn studie heb ik mijn medestudenten vaak horen praten over Qt en Xamarin, waardoor ik denk dat deze zeer populair zijn, dus heb ik deze ook meegenomen.

Om de vergelijking uit te voeren heb ik gekeken naar de prijs, support, community en ontwikkeltijd.

De prijs, support en community heb ik gevonden op de websites van de IDE's. De ontwikkeltijd is afhankelijk van benodigde kennis en hoeveelheid code die nodig is. Hierom heb ik gevraagd aan de software engineers welke andere programmeertalen zij kennen, naast Object Pascal. Hieruit bleek dat 6 programmeurs Java en JavaScript kennen en 3 programmeurs C++ kennen. Verder kennen enkele programmeurs ook C, C#, Objective C of Swift. Daarnaast heb ik ook gekeken naar de lijst op het intranet van, met de melding dat op de lijst ook voormalige werknemers stonden en dat het ook eventueel niet volledig ingevuld is. In die lijst stond dat 10 programmeurs Java kunnen, 8 kunnen JavaScript, 7 kunnen C++, 5 kunnen C#, 4 kunnen C, PHP en Python en 3 kunnen HTML.

Qt en Xamarin gebruiken respectievelijk C++ en C#, dus heb ik verder alleen nog gekeken naar Platforms die JavaScript gebruiken. Ik heb Phonegap en Appcelerator als hoog aanbevolen gevonden, maar omdat de IDE van Phonegap nog in de beta is heb ik me gefocused op Appcelerator.

(<http://www.developereconomics.com/pros-cons-top-5-cross-platform-tools/> 24/03/2015)

E.2.1 Android Studio

Android Studio is de officiële IDE voor Android ontwikkeling en gebruikt Java als programmeertaal. Het is gratis om te gebruiken en Android Packages kunnen zelf geïnstalleerd worden op devices. Maar, om op de Play store een app uit te kunnen brengen moet er eenmalig een \$25 betaald worden.

(developer.android.com/tools/studio/index.html 24/03/2015)

Support is ook gratis en community gebaseerd op developer.android.com/support.html, maar dat betekent ook dat er geen garantie is dat de gebruiker een oplossing aangeboden krijgt. Daarnaast is er een blog op android-developers.blogspot.nl waar regelmatig geschreven wordt wat men kan doen met nieuwe Android versies en updates.

E.2.2 Visual Studio

Visual Studio is de officiële IDE voor Windows Phone ontwikkeling en gebruikt C#, C++ of JavaScript. Het kost €75 om een bedrijf te registreren om apps uit te kunnen geven.

(appdev.microsoft.com/StorePortals/en-US/Account/Signup/SelectAccountType 24/03/2015)

Het lijkt erop dat de support van Microsoft voornamelijk over het opzetten van Visual Studio gaat, maar dat technische vragen op fora gesteld moeten worden.

(<https://dev.windows.com/en-us/community/support?prid=WSDevPortal&sttop=Getting+started> 24/03/2015)

E.2.3 XCode

XCode is de officiële IDE voor iOS ontwikkeling en gebruikt Objective C als programmeertaal. Het kost \$99 per jaar voor een bedrijf om te gebruiken en voor het bedrijf dat intern iOS apps wilt gebruiken kost het \$299 per jaar.

(developer.apple.com/programs/which-program 24/03/2015)

Bij technische problemen kan de programmeur een Technical Support Incident (TSI) naar Apple sturen voor ondersteuning. Per jaar krijgt een gebruiker 2 TSIs, maar kan ook meer kopen. Een set van twee extra TSIs kost \$99 en een set van vijf extra \$249.

(developer.apple.com/support/technical/submit 24/03/2015)

E.2.4 Xamarin

Xamarin is een ontwikkelomgeving waarmee apps voor Android, iOS, Mac en Windows gemaakt kunnen worden, geschreven in C#.

Er zijn verschillende versies van Xamarin met verschillende prijzen. Er moet een jaarlijks abonnement voor worden betaald. Als een jaarlijks abonnement wordt stopgezet is de licentie nog steeds geldig maar kan het niet meer worden geupdate en kan er ook geen gebruik meer worden gemaakt van de support. Voor iedere programmeur en doelplatform moet een nieuw abonnement aangemaakt worden.

De starter editie is gratis, maar heeft een limiet op de grootte van de app en kan geen third party libraries gebruiken.

De Indie editie kost \$25 per maand of \$299 per jaar. Het heeft geen limiet op de app grootte, kan third party libraries gebruiken en geeft toegang tot Xamarin.Forms. Xamarin.Forms is een GUI framework waarmee je met één form een ontwerp voor Android, iOS en Windows Phone kan maken. Studenten kunnen een gratis versie aanvragen.

De Business editie kost \$999 per jaar. Hiermee kan Xamarin code met Visual Studio geïntegreerd worden, business features gebruikt worden en kan de gebruiker email support gebruiken. Voor een extra prijs kan de gebruiker assistentie krijgen van Xamarin voor code troubleshooting.

De Enterprise editie kost \$1899 per jaar. Hiermee is er een gegarandeerde responstijd van één werkdag, de gebruiker krijgt toegang tot hotfixes en hulp bij het opzetten van een project.

(store.xamarin.com 24/03/2015)

Naast een IDE biedt Xamarin ook een Test Cloud aan. Hiermee kan de tester in C# or Ruby tests schrijven die met de app interacteren als een gebruiker.

Er zijn hier vier edities van.

De Basic editie kost \$1000 per maand. Hiermee kunnen de testers maximaal 2 apps testen voor maximaal 200 uur per maand. Eén tester krijgt ook toegang tot de Test Cloud courses van Xamarin University. Tot slot hebben de testers toegang tot email support.

De Professional editie kost \$5000 per maand. Hiermee kunnen de testers maximaal 4 apps testen voor maximaal 1000 uur per maand. De testers krijgen een manager toegewezen om te assisteren bij het testen en een prioriteit op de test server.

De Business editie kost \$8000 per maand. Hiermee kunnen de testers maximaal 10 apps testen voor maximaal 1600 uur per maand. Maximaal drie testers krijgen toegang tot de Test Cloud courses van Xamarin University. Ook krijgen de testers maximaal 4 uur per maand een consultaat.

De Enterprise editie kost \$12000 per maand. Hiermee kunnen de testers maximaal 20 apps testen voor maximaal 2400 uur per maand. Ook krijgen de testers maximaal 8 uur per maand een consultaat.

(xamarin.com/test-cloud 24/03/2015)

Xamarin University geeft tutorials en lessen over het maken en publiceren van apps voor \$1995 per jaar. (xamarin.com/university 24/03/2015)

Daarnaast is er een blog op planet.xamarin.com over verschillende onderwerpen wat betreft app development met Xamarin. Ook is er een forum voor gebruikers op forums.xamarin.com.

E.2.5 Qt

Qt is een ontwikkelomgeving waarmee apps voor Android, iOS, Mac en Windows gemaakt kunnen worden, geschreven in C++ of QML, een script taal gebaseerd op CSS en JavaScript.

Voor Qt zijn verschillende licenses waarvoor een maandelijks abonnement gekocht voor moet worden.

De Community license is gratis, maar valt onder Open Source regelgeving. Deze geeft alleen basis functionaliteit.

De Indie Mobile license kost \$25 per maand. Deze geeft functionaliteit voor het verkopen van de apps en in-app purchases.

De Professional license kost \$174 per maand. Deze geeft toegang tot de Qt cloud service, functionaliteit voor data visualisatie en meer tools voor het ontwerpen van de app. Vanaf deze license krijgt de gebruiker support van Qt, namelijk toegang tot online documentatie, toegang tot de support portal en hogere prioriteit bij het aangeven van bugs.

De Enterprise license wordt op maat gemaakt. Deze geeft de mogelijkheid om embedded devices te maken.

(www.qt.io/download & www.qt.io/support 24/03/2015)

Voor alle gebruikers is er een blog op planet.qt.io en een forum op forum.qt.io.

E.2.6 Appcelerator

Appcelerator is een ontwikkelomgeving waarmee apps voor Android, iOS, Mac en Windows gemaakt kunnen worden, geschreven in XML, CSS en JavaScript. Het is gratis om te gebruiken.

(www.appcelerator.com/titanium 24/03/2015)

Er zijn drie soorten support packages. De Basic package staat 2 gebruikers toe om gedurende werktijden support aan te vragen per mail, met een responstijd van 48 uur. De Gold package staat 4 gebruikers toe om gedurende werkdagen support aan te vragen per mail of telefoon, met een responstijd van 4-8 uur. De Platinum package staat 8 gebruikers toe om gedurende werkdagen support aan te vragen per mail of telefoon, met een responstijd van 2 uur.

(www.appcelerator.com/help/support-services 24/03/2015)

Gebruikers kunnen contact opnemen met andere developers via developer.appcelerator.com/devlink. Verder is er voor alle gebruikers een blog op www.appcelerator.com/blog en een forum op developer.appcelerator.com/questions.

E.3 Alternatieve datastromen

Het is theoretisch mogelijk om de notifications naar JSON of XML om te zetten, en dat weer om te zetten naar een stream om te verzenden, omdat ik dat ook moet encoding en ik daar problemen mee heb gehad heb ik dat nog niet geprobeerd, maar ik verwacht dat ik daar ook problemen mee zal krijgen.

E.4 Verdere overwegingen

De talen van de alternatieve ontwikkelomgevingen uit hoofdstuk 2 zijn niet compatibel met Delphi, maar wel met hun eigen taal en C++ libraries. Dus, als er in een andere taal geschreven gaat worden en Prime Vision wil haar eigen library gebruiken, moet deze geconverteerd worden naar C++ of naar de taal van de ontwikkelomgeving.

E.5 Advies

Als Prime Vision met Delphi wil blijven werken raad ik aan om het Bolts&Nuts framework te converteren naar XE7. Hierbij moet rekening gehouden daarmee met dat Android 0-based strings gebruikt en dat Delphi tegenwoordig een eigen ARC heeft.

Als Prime Vision de mobiele app in een andere taal wil schrijven met het framework van PV, dan zullen in ieder geval de benodigde delen van dat framework geconverteerd moeten worden naar C++ of de gewenste taal. Hierbij raad ik aan om van Xamarin gebruik te maken als PV wenst om op meerdere platforms hun apps te kunnen deployen, omdat zij de meeste ondersteuning lijken te hebben. Maar als PV alleen voor Android wil bouwen raad ik aan om gebruik te maken van Android Studio. Het is gratis, de meeste programmeurs van de software engineering afdeling kennen Java en het gebruikt Android's originele API waar de meeste community support voor te vinden is.

E.6 Bronnen

andy.jgknet.de
codeverge.com
dev.windows.com
developer.android.com
developer.apple.com
docwiki.embarcadero.com
www.appcelerator.com
www.developereconomics.com
www.qt.io
xamarin.com