

Prime Vision Applicaties naar Android Porten

Afstudeerrapport

Student

Friso Denijs - 10102426

Bedrijf

Prime Vision

Datum

1 September 2015



Titel

Prime Vision Applicaties naar Android Porten

Studie

HBO Bachelor Informatica

Descriptoren

Android, Delhi, Indy, Internet Direct, Mobile App Development

Omschrijving

De student Friso Denijs heeft voor Prime Vision onderzocht of hun huidige frameworks geruikt kunnen worden in een Android app dat gebouwd is met Delphi.

Student

Friso Denijs
10102426

Examinatoren

F.P.M. van Eeden
G.A. Mijnaerends
H.P. Weenink

Hogeschool

De Haagse Hogeschool
Johanna Westerdijkplein 75
2521 EN Den Haag
070 – 445 88 88

Bedrijf

Prime Vision
Olof Palmestraat 16-18
2616 LR Delft
015 – 219 20 90

Voorwoord

Al vanaf het begin van mijn studie heb ik mobiele applicaties willen maken en voor mijn afstuderen heb ik hier dan ook expliciet naar gezocht. Ik had verschillende projecten aangeboden gekregen, maar wat mij het meeste trok naar de opdracht van Prime Vision was dat ik hier ook gelijk een nieuwe taal kon leren, namelijk Delphi's Object Pascal.

Ik wil graag Guido Spaanbroek en Pim Struik bedanken voor de kans om bij Prime Vision te Werken.

Dennis de Bruijn, Jos Boudestein, Theo van der Donk, Ritsaert Hornstra, Emma Roetman, Roosmarijn Schopman en Jan-Willem Vesseur wil ik bedanken voor de hulp die zij mij hebben gegeven met het werken met Delphi, Indy en Prime Vision's eigen frameworks.

Tot slot wil ik Hélène Weenink en Gerard Mijnaresds bedanken voor hun begeleiding tijdens mijn afstuderen.

- Friso Denijs, Den Haag, 1 september 2015

Samenvatting

Prime Vision wil in de toekomst Android applicaties aanbieden aan haar klanten, maar momenteel zijn al haar applicaties gemaakt voor Windows. Ik ben een Informatica student aan De Haagse Hogeschool en heb van Prime Vision de opdracht gekregen om te onderzoeken of hun frameworks van Windows naar Android te porten zijn. Om dat te doen heb ik het al bestaande SMX monitor nagebouwd als Android app, terwijl ik gebruik maakte van die frameworks, om vervolgens een rapport te maken over het bouwen en advies te geven.

Voordat ik de app maakte, maakte ik eerst een proof of concept om aan te tonen dat ik een connectie kon maken tussen een Windows server en een Android app. Hierna was ik aan de eerste iteratie begonnen. Daarin maakte ik de Notifications tab na, welke verantwoordelijk is voor het weergeven van welke gegevens binnenkomen. Die iteratie was al genoeg om aan te tonen dat het mogelijk is om de frameworks te porten en ik kon hierna al het adviesrapport schrijven.

Inhoudsopgave

Voorwoord.....	3
Samenvatting.....	4
Inhoudsopgave.....	5
1. Inleiding.....	6
2. Organisatie.....	7
3. Opdracht.....	9
3.1 Probleemstelling.....	9
3.2 Doelstelling.....	11
3.3 Gewenste Situatie.....	11
4. Methodiek.....	12
5. Proof of Concept.....	13
5.1 Verschil Android Studio en Delphi XE7.....	14
6. Iteratie 1.....	15
6.1 Planning.....	15
6.2 Requirements.....	15
6.3 Ontwerp.....	19
6.3.1 Functioneel Ontwerp.....	19
6.3.2 Technisch Ontwerp.....	22
6.4 Bouw.....	23
6.4.1 Connectie Probleem.....	25
6.5 Test.....	26
6.5.1 Testen Connectie.....	26
6.5.2 Testen CRUD Filter Set.....	27
7. Iteratie 2.....	29
7.1 Planning.....	29
7.2 Requirements.....	29
7.3 Ontwerp.....	32
8. Advies.....	34
9. Evaluatie.....	35
9.1 Product.....	35
9.2 Proces.....	35
9.3 Beroepstaken.....	36
9.3.1 Uitvoeren Analyse Requirements.....	36
9.3.2 ontwerpen systeemdelen.....	36
9.3.3 bouwen applicatie.....	36
9.3.4 uitvoeren en rapporteren testproces.....	36
Bronnen.....	37
Afkortingenlijst.....	38

1. Inleiding

Dit is het afstudeerverslag van Friso Denijs, student Informatica aan De Haagse Hogeschool. Voor mijn afstuderen heb ik bij Prime Vision onderzocht of hun frameworks van Windows naar Android te porten zijn. Porten is het omschakelen van software van één omgeving naar een andere omgeving, zoals van Windows naar Android. Een systeem is portable als de kosten van dat omschakelen lager zijn dan het opnieuw bouwen van de software voor een nieuw systeem. Om dit te doen heb ik een door hun eerder gemaakte Windows applicatie nagemaakt op Android.

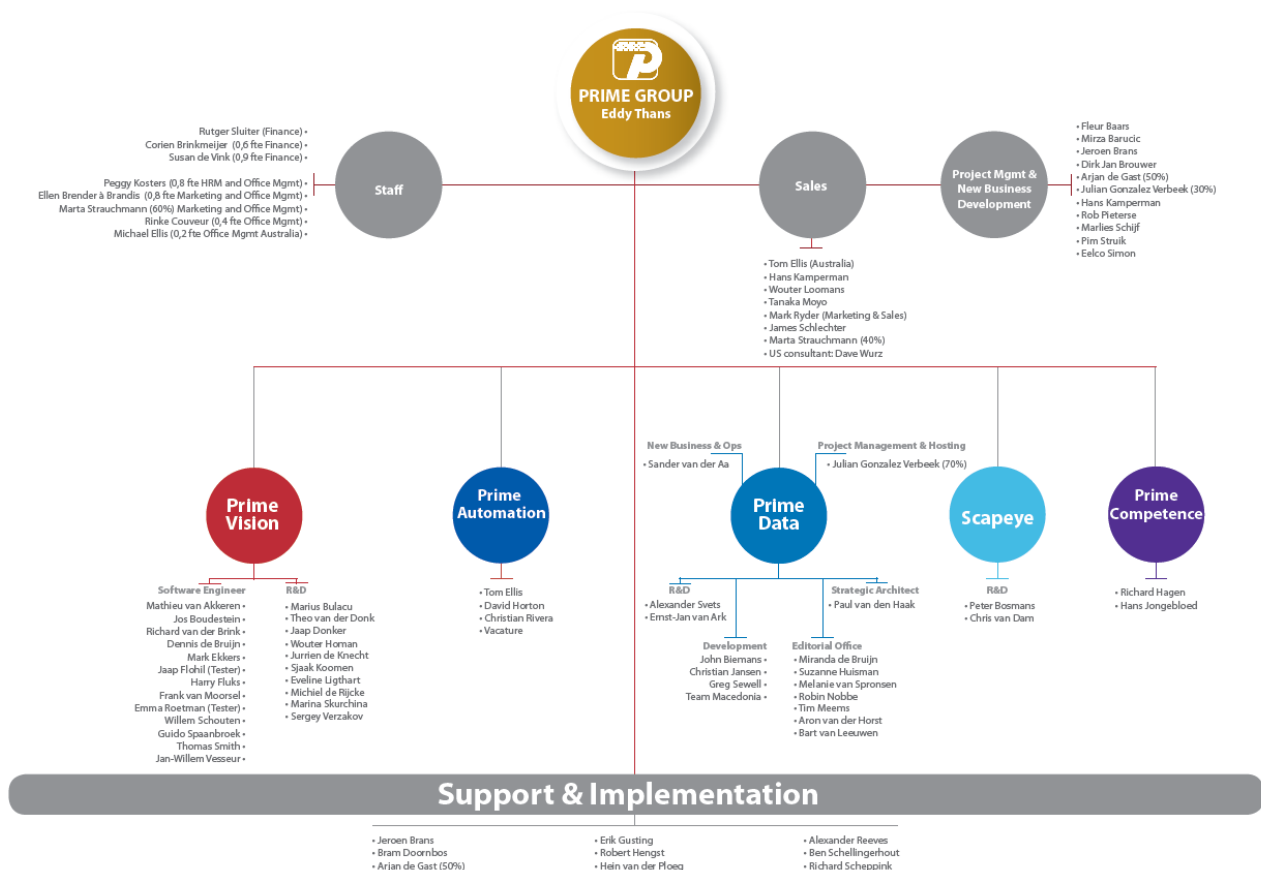
In dit verslag geef ik een beeld van Prime Vision en de opdracht die zij mij hebben gegeven en vertel ik in chronologische volgorde wat ik heb gedaan en waarom ik dat heb gedaan. Tot slot evalueer ik het product, het proces en de beroepstaken die ik wilde halen.

2. Organisatie

In dit hoofdstuk beschrijf ik hoe Prime Vision is ontstaan, hoe Prime Vision is georganiseerd en wat voor werk Prime Vision doet.

Prime Vision was opgericht in 1956 als een R&D departement van PTT om handgeschreven transactie formulieren automatisch te lezen. Toen PTT zich opsplitste in KPN en TNT Post in 1997, werd het een onderdeel van KPN research totdat het in 2000 werd voorbereid om een eigen organisatie te worden. In 2003 werd eigendom van Prime Vision overgedragen aan TNO welke het een individueel commercieel bedrijf van maakte waardoor Prime Vision zich kon uitbreiden naar de wereldmarkt. Het heeft zich uiteindelijk uitgebreid tot Prime Group, waaronder meerdere gespecialiseerde afdelingen zich bevinden.¹

Figuur 1.1 laat de huidige organisatie van Prime Group zien. Het heeft algemene onderdelen voor financiën, verkoop en management en gespecialiseerde onderdelen, waaronder Prime Vision. Prime Vision zelf is verder opgesplitst in R&D en Software Engineering, de laatste was de afdeling waar ik werkte.



Figuur 2.1 Organogram Prime Group

Prime Vision specialiseert zich in het maken van Optical Character Recognition (OCR) en computer vision software, onder andere voor identificatie systemen en geautomatiseerde fabrieksprocessen en postsorteersystemen.²

Toen ik aankwam bij Prime Vision heb ik mijn eigen laptop en werkplek gekregen. Ik had een plek gekregen bij de software engineers die onder andere aan het SMX systeem werken, zodat wanneer ik een vraag had ik snel bij hen terecht kon.

Op de computer die ik had gekregen waren Delphi 2007 en Delphi XE7 geïnstalleerd, 2007 om het oude SMX systeem te kunnen openen en XE7 om de nieuwe app te maken. Verder had ik een eigen SVN branche gekregen voor versiebeheer.

Daarnaast heb ik gemerkt dat wanneer ik met Delphi XE7 de emulator wilde gebruiken ik het elke moest herstarten wanneer ik een nieuwe versie van de app wilde installeren. Hierom heb ik een tablet aangevraagd zodat ik de app direct kon debuggen, zonder dat ik bij elke verandering de emulator moest herstarten.

3. Opdracht

In dit hoofdstuk beschrijf ik wat de situatie was bij Prime Vision voordat ik aan de opdracht begon, wat mijn opdracht was, wat Prime Vision wilde dat ik ging maken.

3.1 Probleemstelling

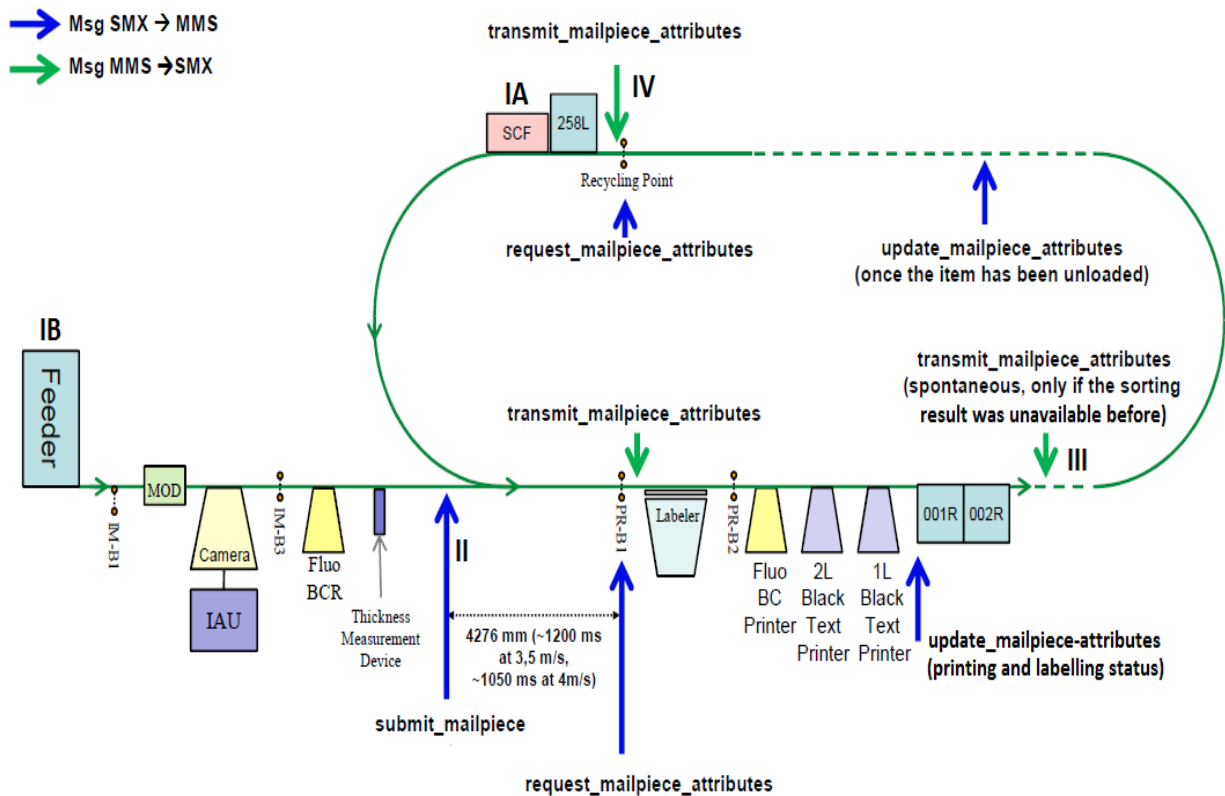
Prime Vision maakt gebruik van haar eigen Delphi frameworks om software te bouwen. Omdat Delphi voormalig alleen gemaakt was om Windows software te maken, zijn de frameworks van Prime Vision ook alleen maar voor Windows gemaakt en getest.

Met de opkomst van mobiele platforms, zoals Android en iOS, heeft Delphi nu ook de mogelijkheid om daar applicaties voor te maken. Prime Vision wil hier op gaan inspelen, maar het was onbekend of haar frameworks geport kunnen worden naar Android. Om die reden had ik de volgende vraag gekregen:

Is het mogelijk om de huidige frameworks van Prime Vision te porten naar Android?

Om die vraag te beantwoorden heb ik de monitor applicatie van het Sorting Machine Extended (SMX) systeem moeten recreëren voor Android. SMX is een onderdeel van het Mail Match System (MMS) dat post sorteert. MMS stuurt attributen van brieven, zoals het adres, naar het SMX systeem welke het vervolgens op basis van die attributen digitaal sorteert. Die sortering wordt terug naar MMS gestuurd welke het vervolgens fysiek sorteert. Figuur 3.1 laat dit proces zien.

Figuur 3.2 laat zien dat de monitor van SMX zeven verschillende tabs bevat met ieder een eigen functie om informatie te geven over wat er gebeurt in SMX. In dit voorbeeld laat het de Notifications tab zien die ik in Iteratie 1 heb geport naar Android.



Figur 3.1 Relatie MMS-SMX

Notifications						
Filter	Active	Time	Type	Code	Application	Description
0.01 All	1					
1.01 All Errors	1	19-2-2015 14:43:18	PVI	NTF_Info_...	smxProcessMonitorPL...	Memory usage for [smxProcessMonitorPLM1]
1.02 All Warnings	1	19-2-2015 14:43:18	PVI	NTF_Info_...	smxProcessMonitorPL...	CPU info for [smxProcessMonitorPLM1]
1.03 All Info	1	19-2-2015 14:43:18	PVI	NTF_Info_...	smxProcessMonitorPL...	Latency for [smxProcessMonitorPLM1] is [0] ms, exceeded is [False]
1.04 All Debug	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI2	Memory usage for [smxServiceGuardMLI2]
1.05 All Progress	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI2	CPU info for [smxServiceGuardMLI2]
1.06 All PVInfo	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI2	Latency for [smxServiceGuardMLI2] is [0] ms, exceeded is [False]
2.01 ProcessMonitor	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI2	Process info for [smxServiceGuardMLI2]
2.02 SamplingCollector	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	Memory usage for [smxServiceGuardMLI1]
2.03 ConfigurationStore	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	CPU info for [smxServiceGuardMLI1]
2.04 MailAttributeStore	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	Latency for [smxServiceGuardMLI1] is [0] ms, exceeded is [False]
2.05 LocalImageStore	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	Process info for [smxServiceGuardMLI1]
2.06 ASNIEDConnector	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	Memory usage for [smxServiceGuardMLI1]
2.07 PoolManager	0	19-2-2015 14:43:17	PVI	NTF_Info_...	smxProcessMonitorPL...	CPU info for [smxProcessMonitorPLM1]
2.08 ServiceGuard	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP1	Latency for [smxServiceGuardGRP1] is [0] ms, exceeded is [False]
2.09 FileSyncer	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP1	Process info for [smxServiceGuardGRP1]
2.10 FileStore	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP1	Memory usage for [smxServiceGuardGRP1]
2.11 ProductManagementCon...	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS1	CPU info for [smxServiceGuardMMS1]
2.12 NVRCconnector	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS1	Latency for [smxServiceGuardMMS1] is [0] ms, exceeded is [False]
2.13 StatisticsCollector	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS1	Process info for [smxServiceGuardMMS1]
2.14 ImageConnector	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS2	Memory usage for [smxServiceGuardMMS2]
2.15 DataConnector	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS2	CPU info for [smxServiceGuardMMS2]
2.16 Controller	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS2	Latency for [smxServiceGuardMMS2] is [0] ms, exceeded is [False]
2.17 MailMatchManager	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP2	Process info for [smxServiceGuardGRP2]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP2	Memory usage for [smxServiceGuardGRP2]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP2	CPU info for [smxServiceGuardGRP2]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Latency for [smxServiceGuardPLM1] is [0] ms, exceeded is [False]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Process info for [smxServiceGuardPLM1]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Memory usage for [smxServiceGuardPLM1]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	CPU info for [smxServiceGuardPLM1]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Latency for [smxServiceGuardPLM1] is [0] ms, exceeded is [False]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Process info for [smxServiceGuardPLM1]

Figur 3.2 SMX Monitor Notifications Tab

3.2 Doelstelling

Prime Vision wil mobiele applicaties aan haar klanten gaan aanbieden, maar met hun huidige frameworks kan dat niet. Door middel van mijn onderzoek moet Prime Vision de kennis verkrijgen voor hoe zij hun frameworks kunnen porten naar Android.

3.3 Gewenste Situatie

Het uiteindelijke doel van mijn opdracht was om een applicatie te maken waarin ik de frameworks gebruikte, voor die applicatie moest er ook documentatie zijn waarin ik beschrijf hoe ik het heb ontworpen, gemaakt en getest. Verder moest er een adviesrapport geschreven worden waarin ik de problemen beschreef die ik was tegengekomen en hoe ik die had opgelost.

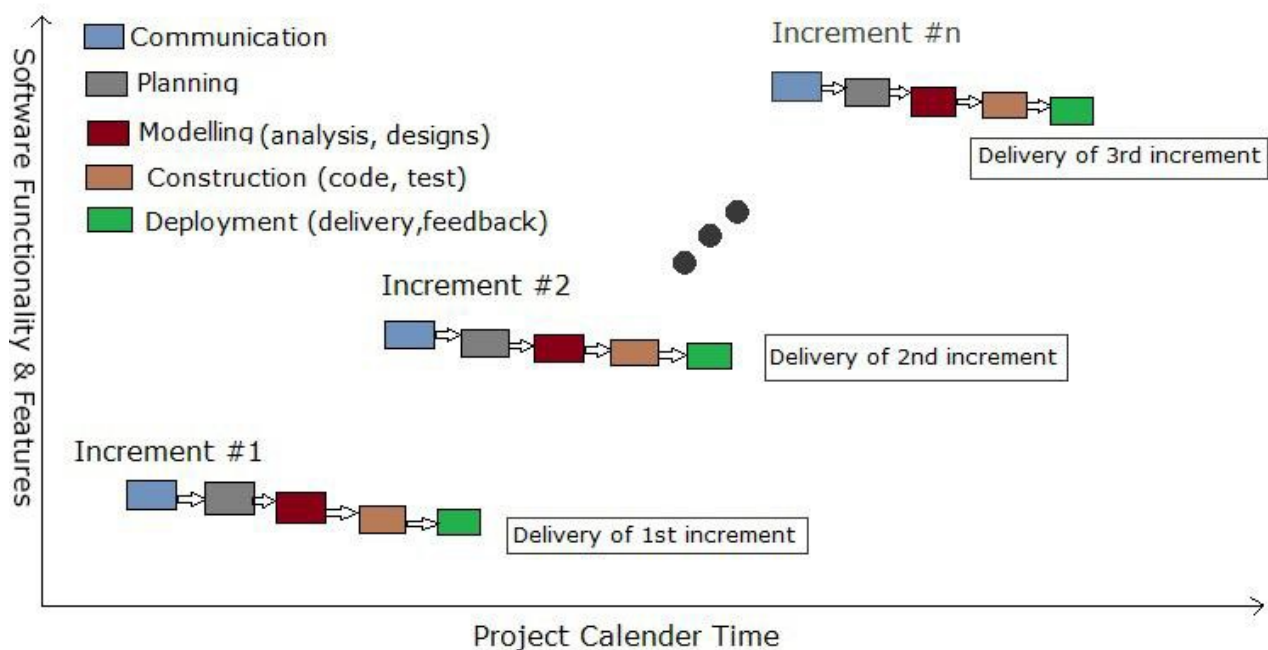
4. Methodiek

In dit hoofdstuk beschrijf ik welke methode ik heb gebruikt om de opdracht uit te voeren en waarom ik voor die methode heb gekozen.

In overleg met mijn werkgever heb ik ervoor gekozen om de applicatie incrementeel op te bouwen. Dat wil zeggen dat ik de applicatie in verschillende delen splits en voor ieder onderdeel de waterval methode toepas om het te maken voordat ik aan het volgende onderdeel begin. Figuur 4.1³ laat hier een voorbeeld van zien.

Ik heb hiervoor gekozen omdat ik weinig kennis had over Delphi en het maken van Android applicaties ermee was relatief nieuw, dus ik kon ook geen hulp vragen daarover of uitgebreide informatie vinden online. Door het incrementeel te maken kan ik in de latere incrementen kennis gebruiken die ik in eerder incrementen heb opgedaan. Ook is het mogelijk om in de eerste increment al een antwoord te geven op de vraag of de frameworks van Prime Vision te porten zijn naar Android, omdat dat een werkend onderdeel moet zijn.⁴

Aan het begin van iedere iteratie ben ik begonnen met overleggen met Prime Vision wat er in die iteratie moest gebeuren om op basis daarvan een plan te maken. Daar na heb ik requirements opgesteld om op basis daarvan het ontwerp te maken. Tot slot heb ik de code geschreven, geteste en samen met de documentatie ervoor opgeleverd aan Prime Vision.



Figuur 4.1 Voorbeeld Incrementeel methode

Doordat de SMX monitor ingedeeld was in verschillende tabs heb ik ervoor gekozen om iedere tab als iteratie te zien. Ik was begonnen met de Notifications tab en ik zou in overleg aan het einde van iedere iteratie bepalen aan welke volgende tab ik zou werken.

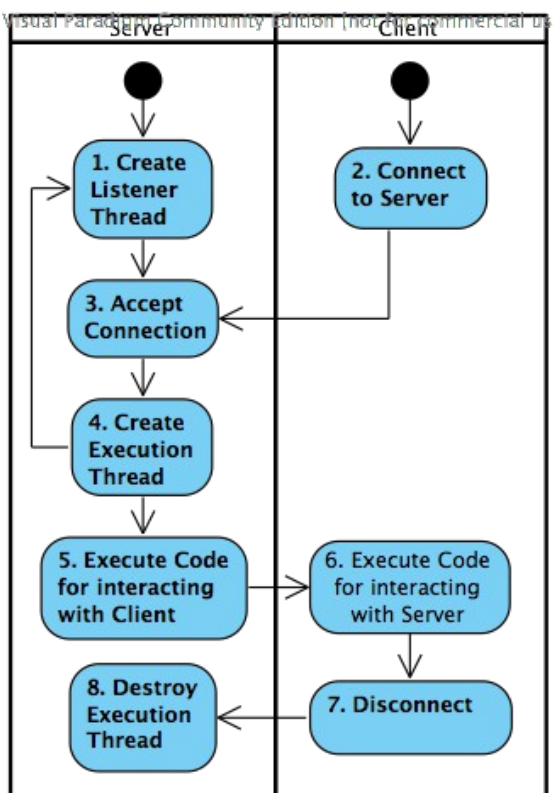
5. Proof of Concept

Toen ik was begonnen met de opdracht bleek al snel dat de framework die Prime Vision om links tussen applicaties te maken niet zou werken op Android. Hierom moest ik een proof of concept maken om aan te tonen dat ik een connectie tussen een Android applicatie en een Windows server kan maken met Delphi.

Ik heb gekozen om Internet Direct (Indy) te gebruiken, omdat deze als enige compatibel is met FMX. Indy heeft de mogelijkheid om verschillende protocollen te gebruiken en ik heb gekozen voor Transmission Control Protocol (TCP) omdat deze een controle heeft of alle bytes correct en compleet aankomen.⁵

Figuur 5.1 laat een versimpelde activity diagram zien hoe de server-client connectie werkt in Indy⁶:

1. Wanneer de server geactiveerd wordt maakt het een listener thread aan, deze wordt gebruikt om te 'luisteren' naar binnenkomende connecties van clients;
2. De client connect met de server;
3. De server 'hoort' de connectie van de client;
4. De server maakt een execution thread voor de connectie met de client en een nieuwe listener thread;
5. De server runt de code voor de interactie met de client;
6. De client runt de code voor de interactie met de client;
7. Uiteindelijk disconnect de client van de server;
8. Tot slot wordt de execution thread van de server vernietigd.



Figuur 5.1 Activiteitsdiagram Indy

5.1 Verschil Android Studio en Delphi XE7

Tijdens het maken van het proof of concept was ik als enige verschil tegengekomen dat Android gebruik maakt van Activities en Fragments voor de Graphic User Interface (GUI) terwijl Delphi maakt gebruik van Firemonkey's (FMX) Forms. Maar, Delphi compilet code automatisch voor het correcte platform, dus de forms van Delphi volgen de lifecycle van Android als ik het voor Android compile.⁷

6. Iteratie 1

Nadat ik had onderzocht hoe ik data kon versturen van een Windows server naar een Android applicatie in het proof of concept, had ik de opdracht gekregen om voor de eerste iteratie de Notifications tab van de SMX monitor na te maken op Android. Deze bevat het andere framework van Prime Vision, het Bolts&Nuts framework, welke verantwoordelijk is voor het serialiseren en deserialiseren van data. De notifications tab zelf laat notificaties zien van welke informatie de monitor ontvangt, bijvoorbeeld dat er gegevens van CPU gebruik van een proces monitor binnen zijn gekomen en op welk tijdstip.

6.1 Planning

Voor de eerste iteratie had ik drie weken uitgerekend, één week voor het onderzoeken van wat er in het systeem moet en de overige twee weken voor het ontwerpen, maken en testen van de tab voor Android, zoals te zien in figuur 6.1.

2015	9-2	10-2	11-2	12-2	13-2	14-2	15-2	16-2	17-2	18-2	19-2	20-2	21-2	22-2	23-2	24-2	25-2	26-2	27-2
Planning maken																			
Onderzoek uitvoeren																			
Requirements vaststellen																			
Tests ontwerpen																			
Architectuur ontwerpen																			
App maken																			
App testen																			
Eventueel app fixen																			
Rapport schrijven																			

Figuur 6.1 Planning Iteratie 1

6.2 Requirements

Omdat de app moest werken zoals de huidige SMX monitor ben ik begonnen met een onderzoek naar wat de Notifications tab doet in het huidige systeem door de volgende vragen te stellen:

- Wat zit er in de Notifications tab?
- Hoe werkt de Notifications tab?

Nadat ik die vragen had beantwoord kon ik de requirements uitwerken, zoals te zien is in tabel 6.1.

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

Tabel 6.1 Requirements Iteratie 1

Figuur 6.2 laat zien dat de Notifications tab uit twee onderdelen bestaat:

- De Filter Set View aan de linkerkant, hierin kan de filter set bewerkt worden, welke gebruikt wordt om de notifications te filteren;
- De Notifications View aan de rechterkant, hierin worden notifications getoond over welke soort informatie de systeemonderdelen van het SMX systeem naar de SMX monitor heeft gestuurd.

Het uiterlijk van de tab heeft geleid tot requirements R.1 tot R.12.

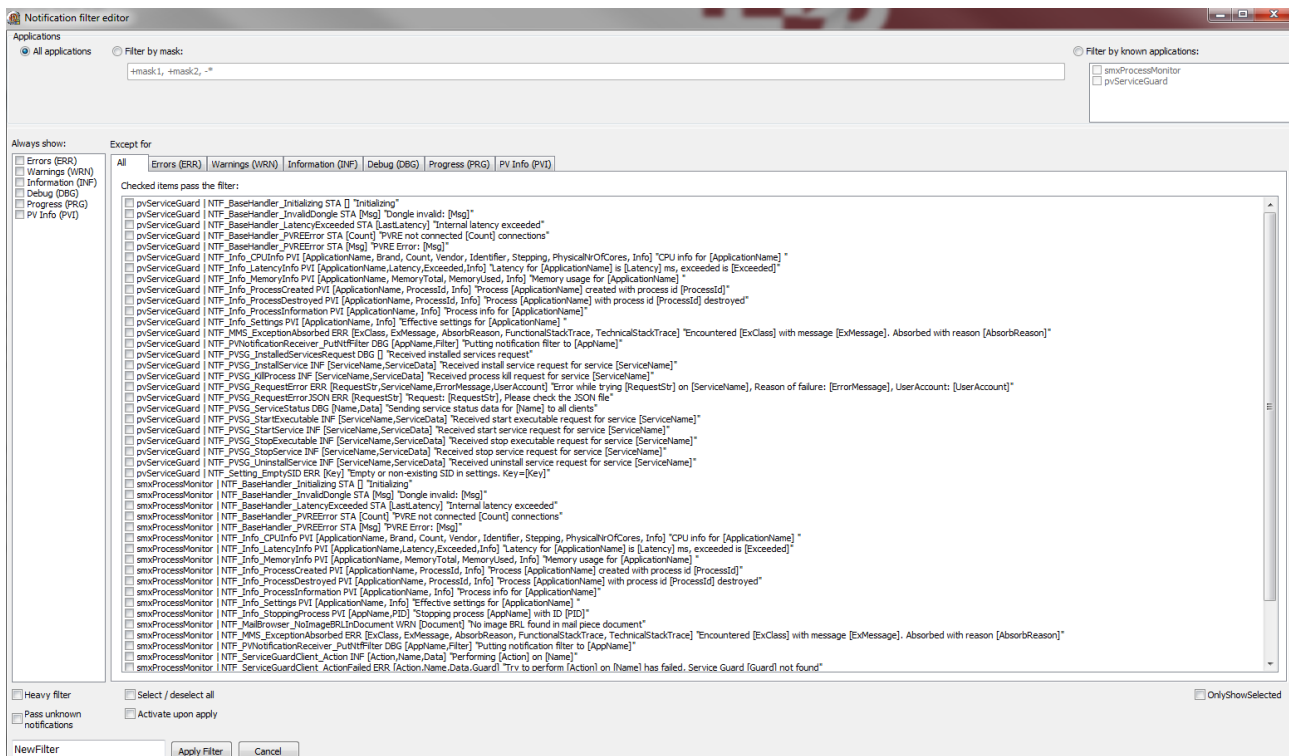
The screenshot displays the 'Notifications' tab in the SMX Monitor application. The interface includes a top navigation bar with icons for Applications, Mail Browser, Image Store Viewer, Sampling, Configuration, Notifications, and Support. The main area is titled 'Notifications' and contains a table of notification details. The table has columns for Filter, Active, Time, Type, Code, Application, and Description. The notifications are listed on the left, and the main table shows details for 'Memory usage for [smxProcessMonitorPLM1]'. The filter editor at the bottom allows users to create new filters, delete selected filters, reload filter sets, and save filter sets.

Filter	Active	Time	Type	Code	Application	Description
1.01 All Errors	1	19-2-2015 14:43:18	PVI	NTF_Info_...	smxProcessMonitorPLM1	Memory usage for [smxProcessMonitorPLM1]
1.02 All Warnings	1	19-2-2015 14:43:18	PVI	NTF_Info_...	smxProcessMonitorPLM1	CPU info for [smxProcessMonitorPLM1]
1.03 All Info	1	19-2-2015 14:43:18	PVI	NTF_Info_...	smxProcessMonitorPLM1	Latency for [smxProcessMonitorPLM1] is [0] ms, exceeded is [False]
1.04 All Debug	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI2	Memory usage for [smxServiceGuardMLI2]
1.05 All Progress	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI2	CPU info for [smxServiceGuardMLI2]
1.06 All PVIInfo	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI2	Latency for [smxServiceGuardMLI2] is [0] ms, exceeded is [False]
2.01 ProcessMonitor	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	Process info for [smxServiceGuardMLI1]
2.02 SamplingCollector	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	Memory usage for [smxServiceGuardMLI1]
2.03 ConfigurationStore	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	CPU info for [smxServiceGuardMLI1]
2.04 MailAttributeStore	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	Latency for [smxServiceGuardMLI1] is [0] ms, exceeded is [False]
2.05 LocalImageStore	0	19-2-2015 14:43:18	PVI	NTF_Info_...	smxServiceGuardMLI1	Process info for [smxServiceGuardMLI1]
2.06 ASN1EDConnector	0	19-2-2015 14:43:17	PVI	NTF_Info_...	smxProcessMonitorPLM1	Process info for [smxProcessMonitorPLM1]
2.07 PoolManager	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP1	Memory usage for [smxServiceGuardGRP1]
2.08 ServiceGuard	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP1	CPU info for [smxServiceGuardGRP1]
2.09 FileSyncer	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP1	Latency for [smxServiceGuardGRP1] is [0] ms, exceeded is [False]
2.10 FileStore	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP1	Process info for [smxServiceGuardGRP1]
2.11 ProductManagementCon...	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS1	Memory usage for [smxServiceGuardMMS1]
2.12 NVRConnector	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS1	CPU info for [smxServiceGuardMMS1]
2.13 StatisticsCollector	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS1	Latency for [smxServiceGuardMMS1] is [0] ms, exceeded is [False]
2.14 ImageConnector	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS1	Process info for [smxServiceGuardMMS1]
2.15 DataConnector	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS2	Memory usage for [smxServiceGuardMMS2]
2.16 Controller	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS2	CPU info for [smxServiceGuardMMS2]
2.17 MailMatchManager	0	19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS2	Latency for [smxServiceGuardMMS2] is [0] ms, exceeded is [False]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardMMS2	Process info for [smxServiceGuardMMS2]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP2	Memory usage for [smxServiceGuardGRP2]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP2	CPU info for [smxServiceGuardGRP2]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardGRP2	Latency for [smxServiceGuardGRP2] is [0] ms, exceeded is [False]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Process info for [smxServiceGuardGRP2]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Memory usage for [smxServiceGuardPLM1]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	CPU info for [smxServiceGuardPLM1]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Latency for [smxServiceGuardPLM1] is [0] ms, exceeded is [False]
		19-2-2015 14:43:16	PVI	NTF_Info_...	smxServiceGuardPLM1	Process info for [smxServiceGuardPLM1]

Figuur 6.2 SMX Monitor Notifications Tab

Figuur 6.3 laat de filter editor dialog zien, welke gebruikt wordt om een filter toe te voegen aan de filter set of een al bestaande filter set te bewerken. Er is te zien dat er gefilterd kan worden op type notification, van welk systeemdeel de notification vandaan komt en dat er uitzondering gemaakt kan worden op bepaalde tekst.

Het uiterlijk van de dialog heeft geleid tot requirementts R.13a tot R.13d



Figuur 6.3 SMX Monitor Filter Editor Dialog

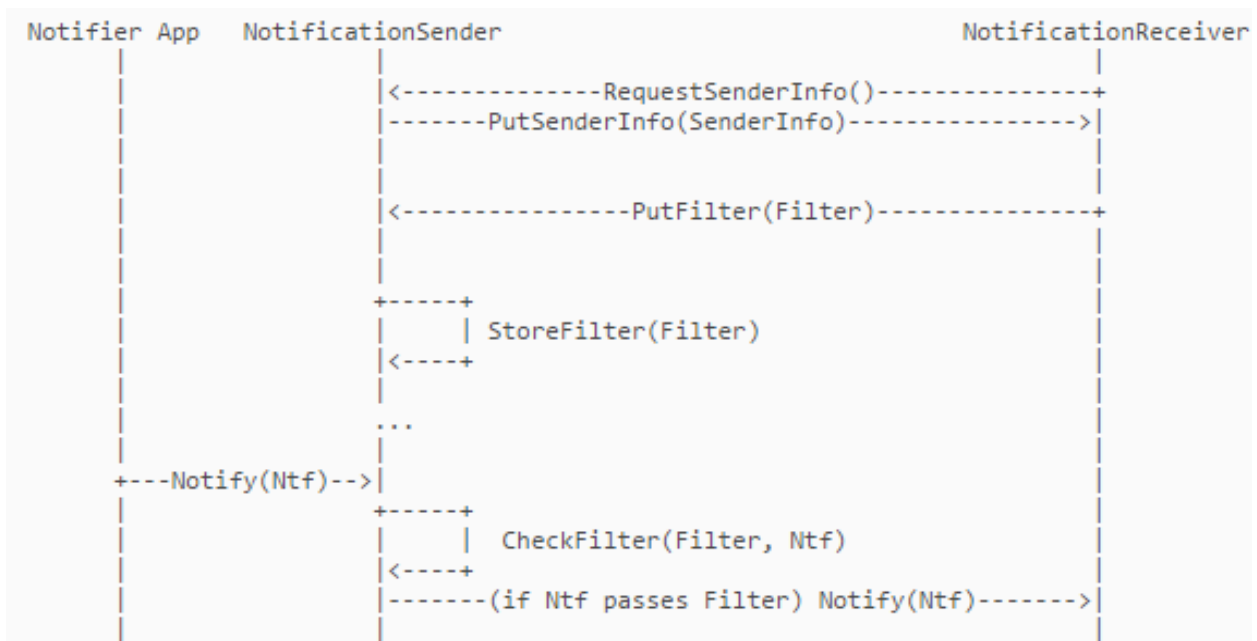
Het notificatie proces bevat 5 belangrijke onderdelen⁸:

- Notifier App, dit is een applicatie die notifications verzendt;
- NotificationSender, dit zit in de applicaties die de notifications verzenden;
- NotificationReceiver, deze zit in het SMX systeem en ontvangt de notifications;
- Notifications, dit is een Bolt list van het Bolts&Nuts framework en bevat de notificatons;
- Filter Set, deze bevat de filters die bepalen welke notifications ontvangen mogen worden.

Figuur 6.4 was gemaakt door Prime Vision nog voordat ik aan de opdracht was begonnen. Het geeft het notification proces weer, vanaf de Notifier App (een onderdeel van een systeemdeel dat notifications aanmaakt) naar de NotificationSender (het onderdeel dat de notifications verstuurt)⁹: naar de NotificationReceiver (het onderdeel van de monitor dat de notifications ontvangt):

1. Eerst vraagt NotificationReceiver informatie over de NotificationSender aan;
2. De NotificationSender stuurt die informatie naar de NotificationReceiver;
3. De NotificationReceiver stuurt haar Filter Set naar de NotificationSender;
4. De NotificationSender slaat de Filter Set op;
5. De Notifier App wil een Notification versturen;
6. De NotificationSender checkt vervolgens of de Filter Set die Notification door laat;
7. Als het Filter de Notification door laat wordt het verstuurd naar de NotificationReceiver.

Requirement R.14 tot R.20 zijn gebaseerd op de functionaliteit van de notifications tab. Later bleek dat het handiger was om de filter set op de server op te slaan, waardoor in overleg mijn bedrijfsbegeleider requirements R.14 tot R.16 zijn komen te vervallen.



Figuur 6.4 Notifications Sequence Diagram

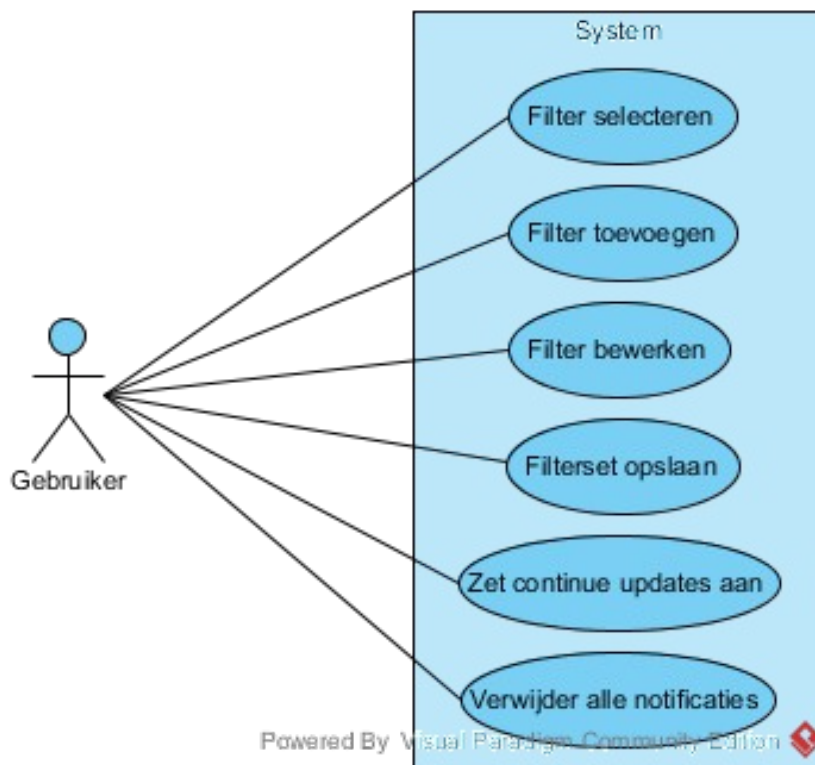
R.21 was gekozen om de filter set en de notifications in een view weer te kunnen geven. R.22 was gekozen om het makkelijker te maken om in later iteraties functionaliteit aan die tabs toe te kunnen voegen. R.23 was gekozen omdat zonder internet verbinding met de server notifications niet ontvangen kunnen worden.

6.3 Ontwerp

Ik heb het ontwerpen van de Notifications in twee stappen gedaan. Eerst heb ik een functioneel ontwerp, gevolgd door het technisch ontwerp. Doordat er geen goede voorbeelden of tutorials voor Indy waren heb ik het technisch ontwerp alleen op abstract niveau kunnen doen, maar ik heb door deze opdracht ervaring op kunnen doen om een volgende keer er dieper op in te gaan.

6.3.1 Functioneel Ontwerp

Op basis van de requirements in tabel 6.1 heb ik de use cases gemaakt. De eerste drie use cases bewerken filters binnen een Filter Set. De vierde use case slaat de Filter Set op. De laatste twee use cases hebben te maken met de notifications, namelijk het binnen halen en het verwijderen van notifications.

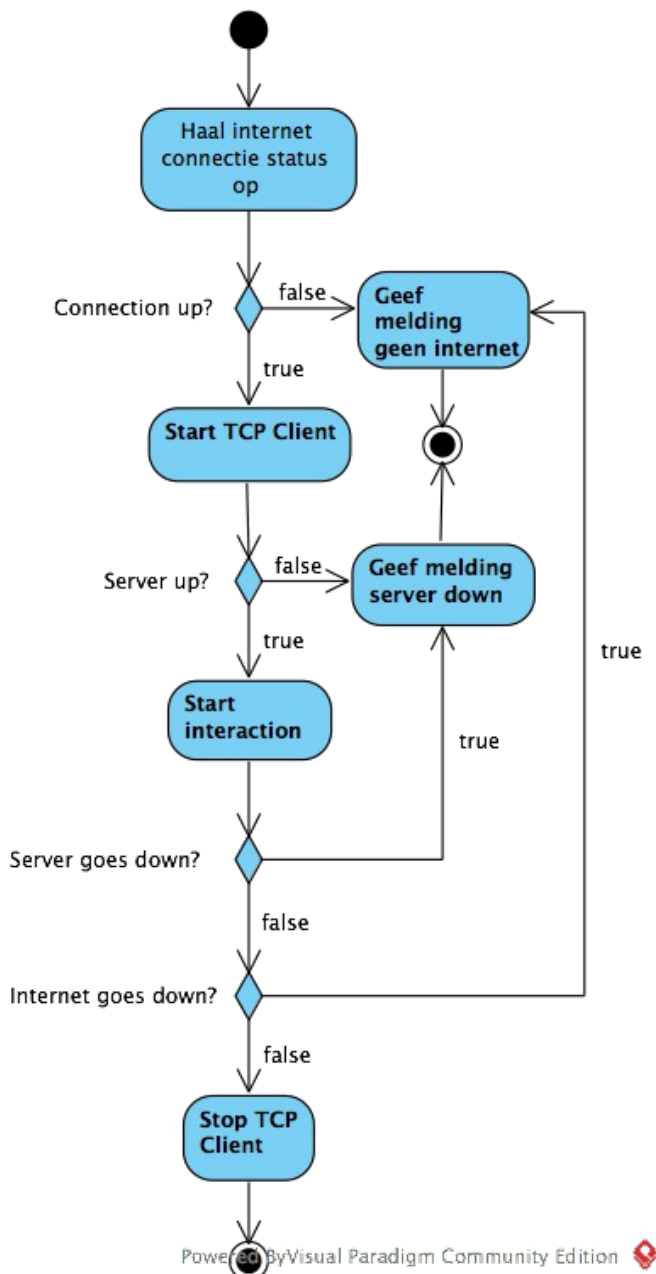


Figuur 6.5 Use Cases Iteraties 1

Het grootste deel van deze iteratie bleef haken op UC.5 doordat het onduidelijk was hoe de Indy client werkt binnen een thread. Meer hierover in paragraaf 6.4.1.

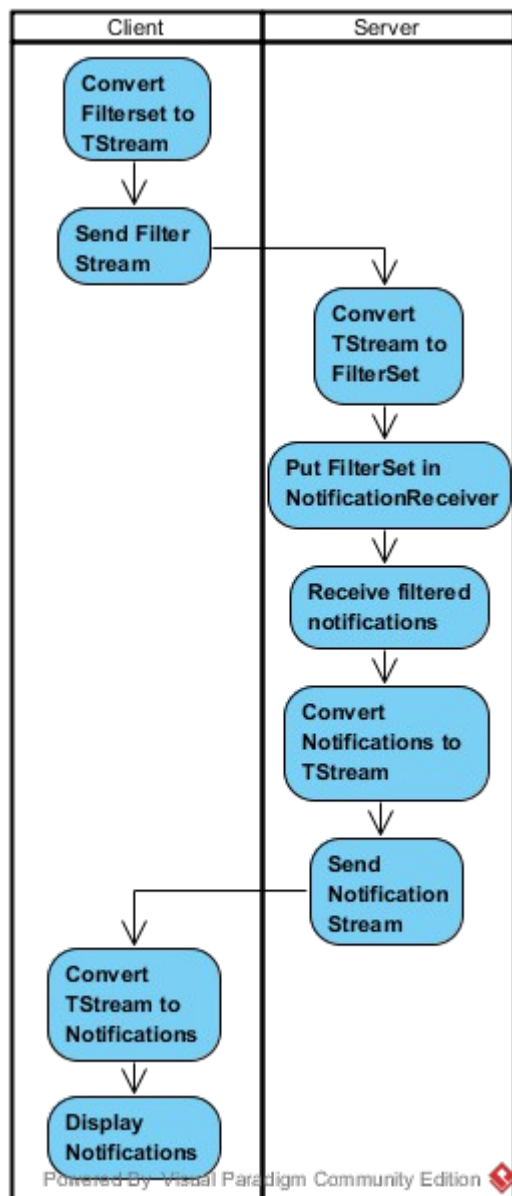
ID	UC.5
Requirements	R.20
Title	Zet continue updates aan
Description	De gebruiker zet continue updates aan
Primary Actor	Gebruiker
Preconditions	De continue updates optie staat uit
Postconditions	De notifications worden continu geupdate
Main Success Scenario	1. De gebruiker selecteert de check box voor continue updates 2. Het systeem begint om de notifications continu te updaten
Extensions	-

Figuur 6.6 laat zien hoe de client de connectie op zou gaan starten. Eerst wordt er gecontroleerd of er een connectie met het internet is, vervolgens wordt de connectie met de server getest. Als beide up zijn wordt de interactie uit figuur 6.7 uitgevoerd, totdat de connectie wegvalt of de client wordt gestopt.



Figuur 6.6 Initieel Activity Diagram Connecting de Client

Figuur 6.7 laat de interactie tussen de client en de server zien. De client converteert een filter set naar een TStream object en verzendt die. Wanneer het op de server aankomt converteert die het terug naar een filter set welke in de NotificationReceiver gestopt wordt. Daarna worden de gefilterde notificaties geconverteerd naar een TStream object en naar de client verzonden, waar ze terug geconverteerd worden naar notifications om ze vervolgens weer te geven. Ik heb ervoor gekozen om het op deze manier te doen zodat de client niet (over)belast zal worden met notifications die zij niet wil ontvangen. Dit leidt er wel toe dat er maar een client per notification receiver mag zijn, omdat de gewenste filtersets anders met elkaar kunnen conflicteren. Een oplossing hiervoor is om wanneer een nieuwe connectie wordt opgezet een nieuwe filterset te maken voor de client met een unieke identifier die gestuurd wordt naar de client. Omdat ik voor dit project maar rekening hoef te houden met 1 client, heb ik in overleg ervoor gekozen om deze oplossing nog niet te implementeren.



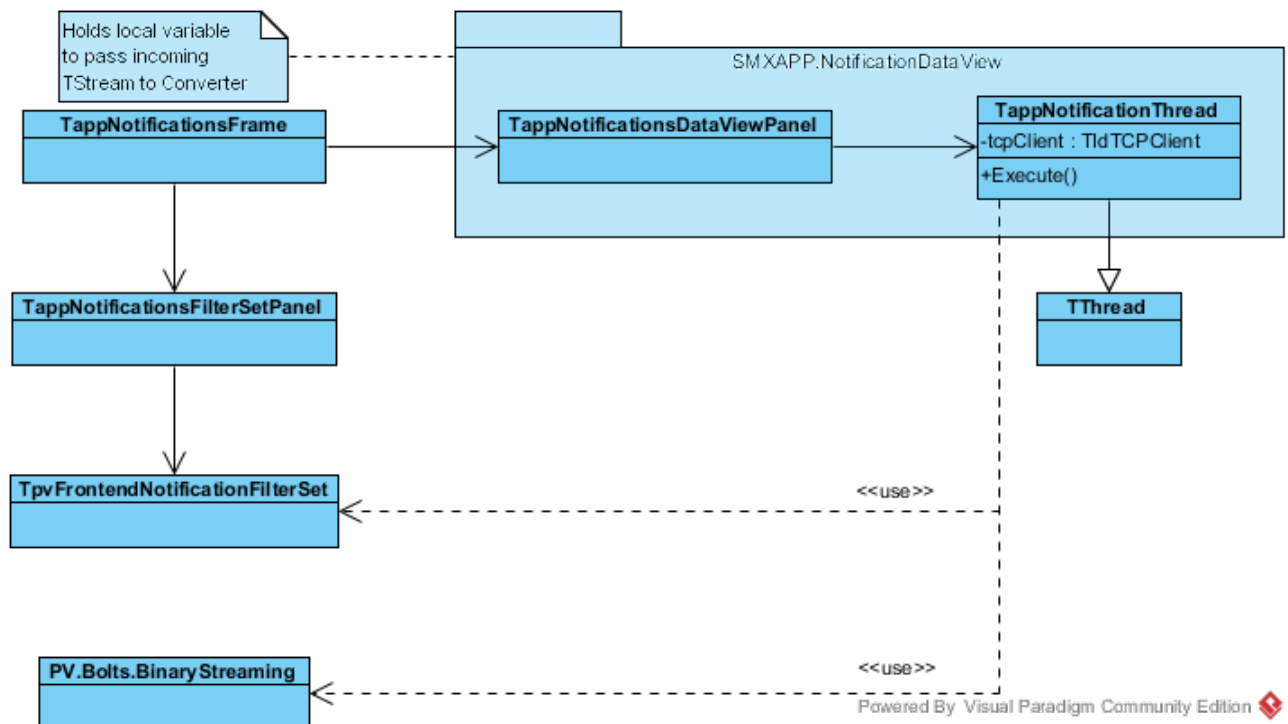
Figuur 6.7 Initieel Activity Diagram Interactie Client-Server

6.3.2 Technisch Ontwerp

Figuur 6.8 geeft het initiële klassendiagram weer voor de client kant.

De Main View voor deze iteratie is TappNotificationsFrame. Hierin bevinden zich de TappNotificationsFilterSetPanel, welke de filter set bevat, en de TappNotificationsDataViewPanel, welke de notifications weergeeft.

Om het TpvFrontendNotificationFilterSet object naar een TStream te kunnen converteren maakt TappNotificationThread gebruik van de methods van PV.Bolts.BinaryStreaming.

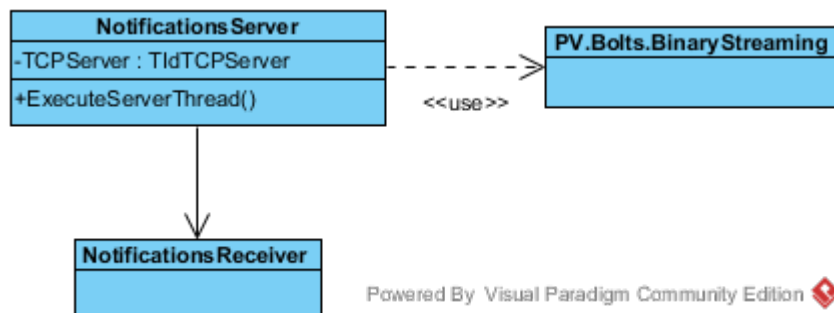


Figuur 6.8 Initieel Class Diagram Client Side

Figuur 6.9 laat het initiële klassendiagram voor de server zien.

De server zit in het NotificationsServer object, de ExecuteServerThread procedure wordt aangeroepen bij het OnExecute event.

De server bevat ook een NotificationsReceiver en gebruikt PV.Bolts.BinaryStreaming om de notifications te converteren naar een TStream en om de filterset die binnenkomt als TStream terug om te zetten naar een TpvFrontendNotificationFilterSet object.

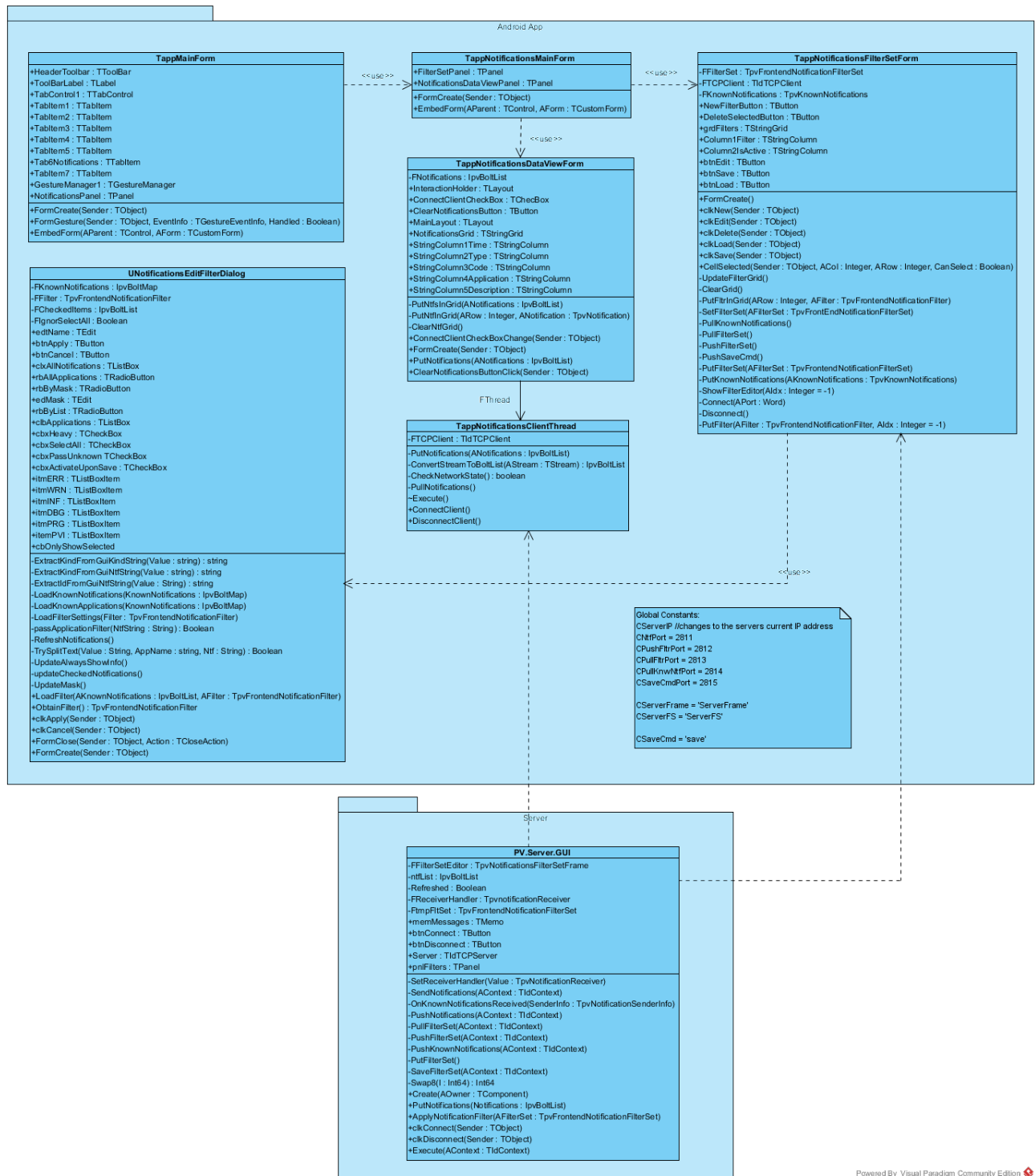


Figuur 6.9 Initieel Class Diagram Server Side

6.4 Bouw

Figuur 6.10 laat zien hoe het de Notifications tab uiteindelijk is gebouwd. Aan de server kant heb ik maar een klasse hoeven maken. Deze hergebruikt de **TpvNotificationReceiver**, **TpvFrontendNotificationFilterSet** en **TpvNotificationsFilterSetFrame** van de monitor. De receiver en filter set werken zoals beschreven in paragraaf 6.2. De filter set frame is de originele GUI voor het bewerken van de filter set. Het bevat alle functionaliteit die nodig is om een filter set te bewerken en ik heb hierom ervoor gekozen om deze te herbruiken in plaats van zelf de functionaliteit opnieuw te schrijven.

TappMainForm is het hoofdscherm van de Android app. Het bevat de notifications tab welke op haar beurt TappNotificationsMainForm bevat. Die form bevat twee panels, een voor TappNotificationsFilterSetForm en de ander voor TappNotificationsDataViewForm. De eerste is om de filter set te bewerken en de tweede laat de notifications zien. TappNotificationsClientThread is de thread om met de server te communiceren. TappNotificationsEditFilterDialog is om filters te bewerken of creëren. IpvBoltList en IpvBoltMap zijn onderdeel van het Bolts&Nuts frameworken en zijn datasets om data op te slaan.



Figuur 6.10 Uiteindelijk Class Diagram

In het document dat ik heb opgeleverd aan Prime Vision, Bijlage C, heb ik de server niet beschreven. Dit heb ik gedaan omdat de server alleen bestaat uit het framework van Prime Vision en enkele pull en push procedures die het zelfde werken als de pull en push procedures van de client.

6.4.1 Connectie Probleem

Tijdens het bouwen kwam ik een knelpunt tegen wat het grootste deel van de tijd van mijn opdracht gekost om op te lossen. Wanneer de notifications van de server naar de app werden gestuurd kwam de eerste set wel aan, maar het leek dat de andere sets niet aankwamen. Eerst heb ik met behulp van collega's en stackoverflow.com gecontroleerd of mijn code klopte. Het leek erop dat mijn code goed was, dus heb ik vervolgens met Wireshark, een programma om verzonden en ontvangen internetpackets te controleren, de verzonden informatie gecontroleerd. Hieruit bleek dat de informatie wel correct werd verzonden. Uiteindelijk heb ik het opgelost door kleine beetjes van de code te veranderen totdat ik het deel van de code in TappNotificationsClientThread dat verantwoordelijk is voor het ontvangen van data had geencapsulate in een eigen procedure. Hierna werkte het. Dit kwam waarschijnlijk omdat ik de gebruikte TMemoryStream, een type die ik gebruikte om de binnenkomend bytes als array op te slaan, moest resetten elke keer nadat ik het had gebruikt, door het te encapsulaten hoef ik dat niet zelf te doen omdat het ieder keer dat het de procedure aanroept een verse TMemoryStream aanmaakt. Code 6.1 laat de geencapsulate code zien.

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

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

Code 6.1 TappNotificationsClientThread.PullNotifications

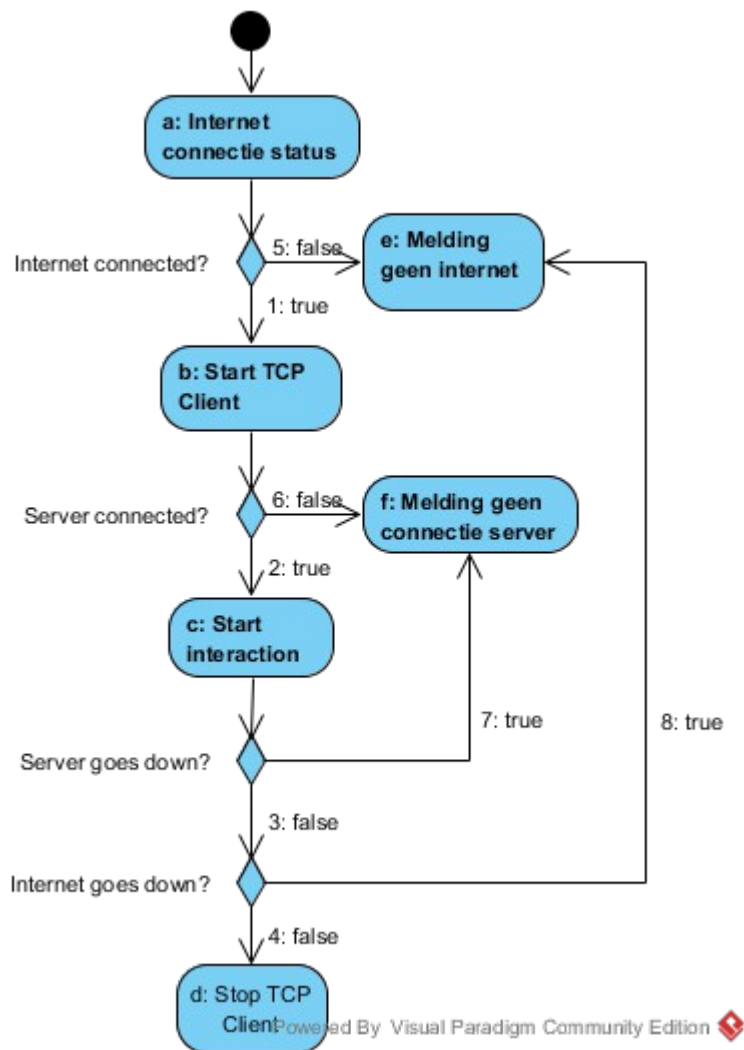
6.5 Test

Ik had in eerste instantie gekozen om gebruik te maken van Testgoal¹⁰ omdat ik hier de meeste ervaring mee heb en het past bij wat ik moest testen, maar ik had een collega die TMap Next¹¹ gebruikte. Omdat ik deze methode nog niet eerder had gebruikt, ik er nieuwsgierig naar was en als ik er vragen over had ik deze aan mijn collega kon stellen heb ik gekozen om van TMap Next gebruik te maken.

6.5.1 Testen Connectie

Voor het testen van de connectie opzetten had ik eerst gekozen voor een algoritme test van Testgoal. Deze heb ik niet teruggevonden in TMap, maar het komt erg overeen met de procescyclustest (PCT). Maar in plaats van dat ik het proces test, test ik op beslispunten, waarbij ik tijdens het testen alle beslispunten minsten een keer moet langsgaan. Het belangrijkste punt hierbij is punt c, dat is waar de notifications binnenkomen. Doordat de notifications ergens worden aangemaakt waar ik niks mee heb gedaan, heb ik ze visueel gecontroleerd door de notifications die in de server binnen waren gekomen te vergelijken met de notifications die in de Android app waren weergegeven.

Zoals te zien in figuur 6.11, komt het overeen met figuur 6.6 van het initiële ontwerp.



Figuur 6.11 Testpaden Client-Server Connection

Tabel 6.2 laat de fysieke test scenarios van deze test zien. Ik heb alle mogelijke paden en uitkomsten getest.

#	Acties	Verwacht Resultaat
1	1. Zet wi-fi aan 2. Start TCP server 3. Check connect checkbox voor Notifications 4. Uncheck checkbox	De notifications komen binnen wanneer de checkbox is checked en stoppen met binnenkomen totdat checkbox is unchecked.
2	1. Zet wi-fi aan 2. Start TCP server 3. Check connect checkbox voor Notifications 4. Zet wi-fi uit	Er wordt een bericht weergegeven dat er geen internet is.
3	1. Zet wi-fi aan 2. Start TCP server 3. Check connect checkbox voor Notifications 4. Stop server	Er wordt een bericht weergegeven dat de connectie met de server weg is.
4	1. Zet wi-fi aan 2. Stop TCP server 3. Check connect checkbox voor Notifications	Er wordt een bericht weergegeven dat er geen connectie met de server opgezet kan worden.
5	1. Zet wi-fi uit 2. Check connect checkbox	Er wordt een bericht weergegeven dat het internet uitstaat.

Tabel 6.2 Fysieke Testen Connectie

Tabel 6.3 laat de resultaten zien van deze test. Scenarios 3 en 4 hadden niet het verwachte resultaat, maar doordat ik erg achterliep op mijn opdracht en ik mijn doel al had gehaald, heb ik in overleg geen tijd gestopt in het verbeteren van de code hiervoor.

#	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 6.3 Pass (P) of Fail (F) van connectie test

3.5.2 Testen CRUD Filter Set

Voor de filter set heb ik een gegevenscyclustest (GCT) uitgevoerd. De filter set is een entiteit waarin filters zijn opgeslagen en worden aangemaakt of bewerkt met behulp van known notifications en tot slot worden opgeslagen in een xml bestand. Ik heb de CRUD-matrix van tabel 6.4 gemaakt aan de hand van de methodes die interacteren met de filterset en de andere benodigde entiteiten.

#	Method	Filter	Filter Set	Known Notifications	Filter Set XML
1	clkLoad		C, U		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 6.4 CRUD-matrix Filter Set

Tabel 6.5 laat de fysieke test scenarios zien van deze test. Voor elke methode is er een test of die doet wat er verwacht wordt.

#	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 6.5 Fysieke Testen CRUD Filter Set

Tabel 6.6 laat de resultaten zien van de tests. Alle methodes deden wat ze moesten doen, maar wanneer ik een nieuw filter toevoegde moest ik TappNotificationsEditFilterDialog een tweede keer openen om de verandering zichtbaar te maken in de Android app.

#	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 6.6 Pass (P) of Fail (F) van filter set CRUD test

7. Iteratie 2

Omdat ik na iteratie 1 de probleemstelling al had beantwoord moest ik de rest van mijn tijd focussen op het maken van de overige tabs, maar doordat iteratie 1 zo erg was uitgelopen had ik iteratie 2 maar voor een deel kunnen maken. Voor deze iteratie werkte ik aan de Applications tab welke een overzicht geeft van de applicaties en de status van die applicaties

7.1 Planning

Tabel 7.1 laat de planning zien voor het onderzoek en het ontwerp voor de tweede iteratie. Doordat iteratie 1 was uitgelopen ben ik met de tweede iteratie veel later begonnen dan ik origineel had gepland. Deze achterstand betekende ook dat ik minder tijd had voor het maken en testen van de app. Tijdens het 60% gesprek had ik dan ook afgesproken om mij te focussen op het ontwerp en indien ik nog tijd over had in ieder geval de musts van de requirements te maken.

	23-4	24-4	25-4	26-4	27-4	28-4	29-4	30-4	1-5	2-5	3-5	4-5	5-5	6-5	7-5
Onderzoek Applications															
Ontwerp Applications															
Testontwerp Applications															

Tabel 7.1 Planning Voorbereiding Iteratie 2

Tabel 7.2 laat de planning zien voor het bouwen en testen van deze iteratie zien. Helaas had ik het bericht gekregen dat ik mijn afstudeerrapport moest herschrijven en had geen tijd meer om dit deel van de planning uit te voeren.

	7-5	8-5	9-5	10-5	11-5	12-5	13-5	14-5	15-5	16-5	17-5	18-5	19-5	20-5	21-5	22-5	23-5	24-5	25-5	26-5	27-5	28-5	29-5
Bouw Applicaties																							
Test Applicaties																							

Tabel 7.2 Planning Uitvoering Iteratie 2

7.2 Requirements

Tabel 7.3 laat de requirements voor deze iteratie zien. Het meeste van de functionaliteit komt van informatie ontvangen van een server, wat overeenkomt met wat ik heb gedaan in de eerste iteratie. Het grootste probleem is om een treeview te implementeren die werkt als de VirtualTreeView in het SMX systeem, dus heb ik hier een must van gemaakt. Daarnaast heb ik de functionaliteit van die VirtualTreeView als shoulds neergezet. De rest is informatie versturen en verzenden, dus deze zijn coulds.

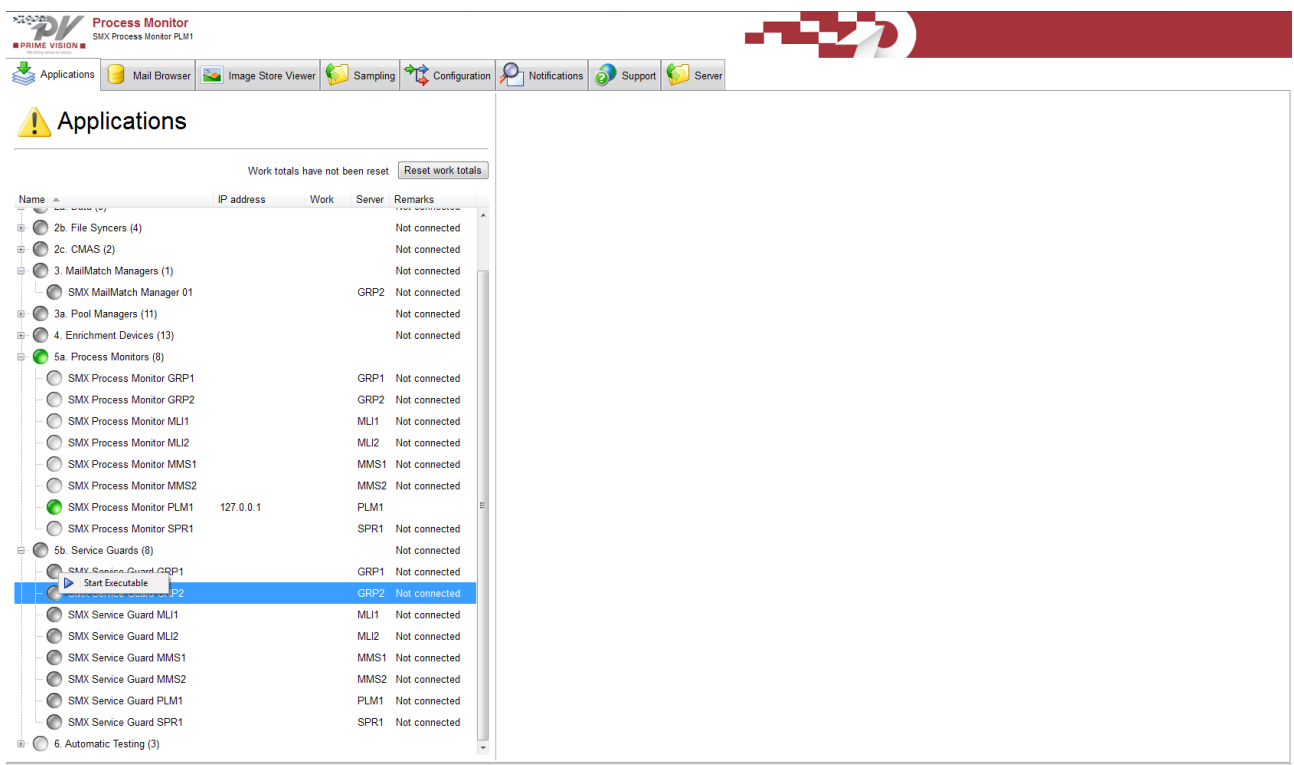
Zoals ik in de planning had beschreven bleek dat ik mijn afstudeerrapport moest herschrijven en hierdoor had ik geen tijd meer om de requirements te realiseren.

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

Tabel 7.3 Requirements Iteratie 2

Ik heb het maken van de requirement op dezelfde manier aangepakt als in iteratie 1.

Voor iteratie 2 werkte ik aan de Applications Tab. Deze geeft een overzicht van de applicaties en de status van die applicaties. Als een applicatie meerdere statussen heeft worden die afwisselend weergegeven. Ook kunnen vanaf deze tab applicaties opgestart worden, zoals te zien is in figuur 7.1.

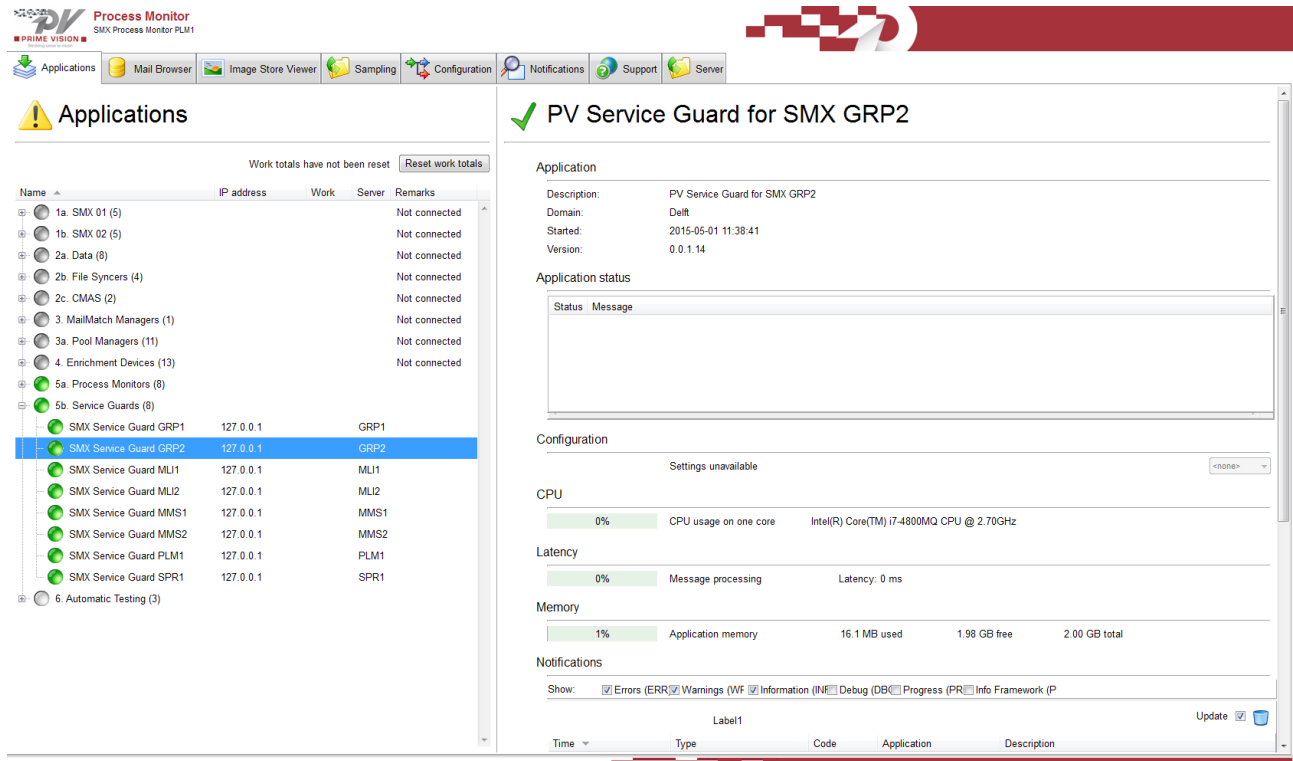


Figuur 7.1 Applications Tab Treeview

Wanneer een applicatie is opgestart en geselecteerd worden de details van de gestarte applicatie in de panel naast de lijst weergegeven, zoals te zien in figuur 7.2:

- description
- domain
- start time
- version

- status
- cpu verbruik
- latency
- memory
- notifications



Figuur 7.2 Applications Tab met Details View

Ik heb voorgesteld om de TreeView en de status afwisselend weer te geven te prioriteren, omdat de rest simpele GUI functionaliteit is en informatie versturen op dezelfde manier als in iteratie 1.

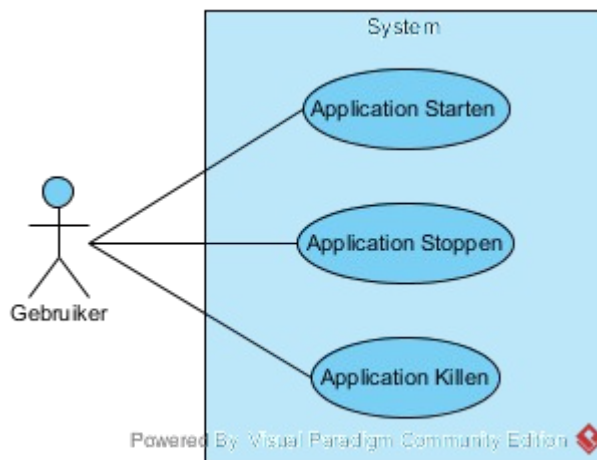
In het huidige systeem worden de applicaties uit een json file gehaald om in de treeview gezet te worden.

Om een applicatie op te kunnen starten moeten eerst de service guards opgestart zijn.¹²

Momenteel wordt de VirtualTreeView van Soft Gems gebruikt. Deze is helaas alleen gemaakt voor VCL en kan ik dus niet voor mijn FMX Android applicatie gebruiken.¹³

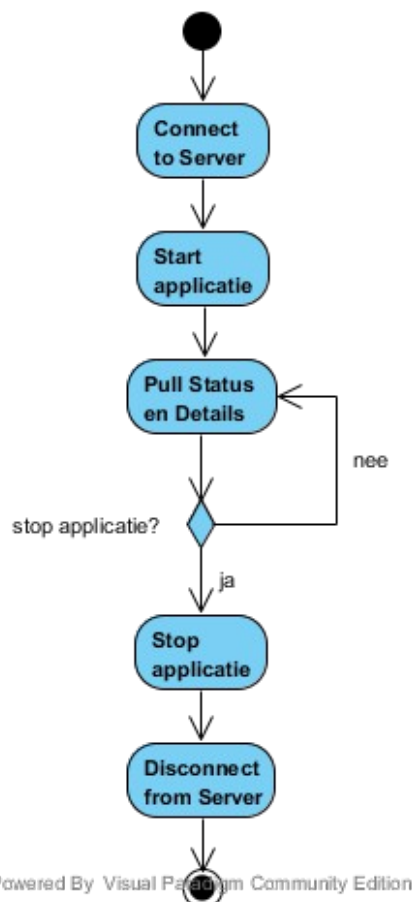
7.3 Ontwerp

De gebruiker interacteert voornamelijk met het systeem door applications te starten en te stoppen, dus ik heb deze als use cases neergezet, zoals te zien is in figuur 7.3.



Figuur 7.3 Use Case Diagram

Figuur 7.4 laat de activiteiten zien die zouden hebben gevonden in de Applications tab. Het belangrijkste hierin was dat wanneer een applicatie eenmaal was opgestart deze continu status updates verzendt, welke gepulld zouden worden door de Android app, totdat het wordt stopgezet.



Figuur 7.4 Activity Diagram Iteratie 2

Ik had al een idee om een TGrid te gebruiken in plaats van een treeview. Ik zou dan mijn eigen functionaliteit hebben gegeven aan de eerste kolom zodat wanneer de cell van een parent node geselecteerd wordt deze opende en de childe nodes weer zou geven.¹⁴

8. Advies

Zoals hoofdstuk 3 beschreef was de vraag of het mogelijk was om de frameworks van Prime Vision naar Android te porten. Na één iteratie bleek al dat dit mogelijk was en kon ik dus al mijn adviesrapport schrijven.

Nathans (2010) beschrijft verschillende manieren om advies te geven, maar uiteindelijk heb ik voor het informeren. Deze pastte het beste bij de opdracht omdat Prime Vision wilde weten of het mogelijk is om huidige frameworks te porten naar Android. Wat zij met die informatie doen is verder aan hun.¹⁵

Om het advies te kunnen geven heb ik alle problemen die ik ben tegengekomen tijdens het bouwen van de applicatie genoteerd met de oplossingen die ik heb gevonden. Ik heb die problemen gesorteerd op soort, zoals te zien is in bijlage E.1:

- Android device, zoals ik heb beschreven in paragraaf 5.1;
- GUI Design, mijn ontwerp kwam niet altijd overeen met hoe het weergegeven werd;
- Delphi 2007, hiermee heb ik de server gebouwd;
- Delphi XE7, hiermee heb ik de Android app gebouwd;
- en de Code van Prime Vision en third parties.

Tijdens een vergadering van software engineers vertelde ik wat ik had gedaan tot dan toe. Hieruit bleek dat er ook interesse was in alternatieven voor Delphi en heb ik dus ook daarnaar nog gezocht, zoals te zien is in Bijlage E.2.¹⁶

9. Evaluatie

Het doel van de opdracht was om te onderzoeken of het mogelijk was om Delphi frameworks van Prime Vision naar Android te porten. Het is mij gelukt om dit aan te tonen en dit hoofdstuk evalueer ik wat ik heb opgeleverd, hoe ik heb gewerkt en of ik aan mijn beroepstaken heb voldaan.

9.1 Product

Het is mij gelukt om al in de eerste iteratie van de application aan te tonen dat het Bolts&Nuts framework te porten is naar Android en hiermee heb ik het doel van mijn opdracht bereikt. Om dit aan te tonen aan mijn werkgever heb ik code, ontwerp- en bouwrapport en een advies gegeven aan hen.

De code is volgens de standaarden van Delphi en Prime Vision geschreven, zodat het het duidelijk leesbaar was voor Prime Vision.

Het ontwerp- en bouwrapport bevat informatie over de code. Waarom ik het op die manier heb geschreven, hoe ik het heb geschreven en de test resultaten.

Het advies rapport beschrijft de problemen die ik tegen was gekomen en hoe ik ze heb opgelost. Die oplossingen gebruik als basis voor het positieve advies dat de frameworks van Prime Vision geport kunnen worden naar Android.

9.2 Proces

Tijdens het project ben ik twee grote obstakels tegengekomen, die ik naar eigen mening professioneel heb aangepakt. De eerste was dat tijdens de eerste iteratie het mij een hele tijd niet lukte om een verbinding te maken tussen de server en de app. Het tweede obstakel was dat ik mijn rapport moest herschrijven 3 weken voor de inleverdatum. Maar, ondanks die obstakels heb ik het gewenste product behaald en veel ervaring opgedaan met Delphi en Indy.

Zoals ik in de vorige paragraaf schreef vind ik dat ik dat ik de tegenslagen professioneel heb aangepakt. Toen het mij niet lukte om in eerste instantie de verbinding te maken heb ik aan mensen met meer ervaring gevraagd of zij een oplossing wisten. Toen bleek dat dat niet het geval was heb ik zelfstandig een oplossing gezocht door telkens een kleine alteratie te maken totdat het werkte.

Toen bleek dat ik mijn afstudeerrapport moest herschrijven heb ik eerst gecontroleerd bij Prime Vision of ik voldeed aan de opdracht zodat ik mij volledig kon focussen op het herschrijven van het rapport.

Dus, alhoewel ik niet meerdere iteraties heb kunnen doen zoals ik had gepland, heb ik

9.3 Beroepstaken

Ik heb alle ingeplande beroepstaken op niveau uitgevoerd. In deze paragraaf beredeneer ik waarom dat zo is.

9.3.1 Uitvoeren Analyse Requirements

Ik heb de requirements analyse op **niveau 3** uitgevoerd. De requirements zijn gebaseerd op het vooronderzoek dat ik iedere iteratie heb gedaan en ik heb in de eerste iteratie bewust gekozen om van alles een must te maken en in de tweede iteratie heb ik een bewuste overweging gemaakt wat een hogere prioriteit had.

9.3.2 ontwerpen systeemdelen

Zoals aangegeven in mijn afstudeerplan heb ik het ontwerpen van systeemdelen op **niveau 4** gedaan. Alhoewel ik bij aanvang van iteratie 1 geen technisch ontwerp kon maken doordat er geen goede voorbeelden of uitleg voor Indy waren voor wat ik nodig had, heb ik het functioneel ontwerp wel goed gemaakt. En, nadat ik iteratie 1 had afgemaakt, heb ik genoeg ervaring opgedaan om voortaan ook een technisch ontwerp te maken als ik Indy weer gebruik.

9.3.3 bouwen applicatie

Ik had aangegeven in mijn afstudeerplan dat het bouwen van de applicatie op niveau 3 zou gebeuren, maar het is uiteindelijk **niveau 4** geworden. Het was complex omdat het aansluit op bestaande software, de server is uiteindelijk onderdeel geworden van het bestaande SMX systeem, en het maakt gebruik van het Framework dat Prime Vision al heeft gemaakt. Ook maakt het gebruik van een versiebeheertool, namelijk TortoiseSVN en Prime Vision's eigen repository.

Omdat mijn app als voorbeeld moet dienen voor toekomstige werkzaamheden heb ik rekening gehouden met hergebruik en wijzigbaarheid. Dit heb ik gedaan door zo veel mogelijk functionaliteit te encapsulaten in eigen procedures, zoals de push en de pull procedures.

9.3.4 uitvoeren en rapporteren testproces

Mijn logische tests zijn gebaseerd op technieken van TMap Next en de fysieke testgevallen waren op de logische testgevallen gebaseerd. Ik had ze ook nog voor de zekerheid laten verifiëren door en collega die gespecialiseerd was in testen. Ook zijn mijn tests herhaalbaar. Dit leidt mij tot de conclusie dat ik deze beroepstaak op **niveau 3** heb gedaan.

Bronnen

1. <http://www.primevision.com/history> Prime Vision. Our History. bezocht op 8-2-2015
2. <http://www.primevision.com/solutions-you> Prime Vision. Solutions for you. bezocht op 8-2-2015
3. http://upload.wikimedia.org/wikipedia/commons/e/e4/Incremental_Model.jpg bezocht op 10-2-2015
4. Selecting a Development Approach. (2008). Centers for Medical and Medicaid Services.
5. www.indyproject.org/index.en.aspx Indy. bezocht op 10-02-2015
6. Hower, C.Z. (Ed.). (2006). Internet Direct (Indy): An Open Source suite of Internet Components. Indy Project.
7. <http://blogs.embarcadero.com/pawelglowacki/2013/09/30/40067> Glowacki, P. Mobile app lifecycle events handling in Delphi XE5. bezocht op 9-8-2015
8. J.W. Vesseur, Persoonlijke Communicatie, 11-2-2015
9. http://scm.primevisiononline.com/projects/postnl-ncs/wiki/Notifications_filtering Prime Vision. Notifications filterint. bezocht op 10-2-2015
10. de Grood, D. (2008). TestGoal: Leerboek resultaatgedreven software testen (1e dr. ed.). Den Haag: Academic Service.
11. Koomen, T., Van der Aalst, L., Broekman, B., & Vroon, M. (2006). TMap Next: Voor resultaatgericht testen. 's-Hertogenbosch: Tutein Nolthenius.
12. J.W. Vesseur, Persoonlijke Communicatie, 4-5-2015
13. <http://www.soft-gems.net/index.php/controls/virtual-treeview> Lischke, M. Virtual Treeview component. bezocht op 4-5-2015
14. <http://docwiki.embarcadero.com/Libraries/XE7/en/FMX.Grid.TGrid> Embarcadero. FMX.Grid.TGrid. bezocht op 5-5-2015
15. Nathans, H. (2010). Adviseren als Tweede Beroep. Deventer: Kluwer.
16. Vergadering, 23-03-2015

Afkortingenlijst

FMX	Firemonkey
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
ICS	Internet Component Suite
Indy	Internet Direct
MMS	Mail Match System
OCR	Optical Character Recognition
RUP	Rational Unified Process
SMX	Sorting Machine eXtended
TCP	Transmission Control Protocol
VCL	Visual Component Library
UDP	User Datagram Protocol