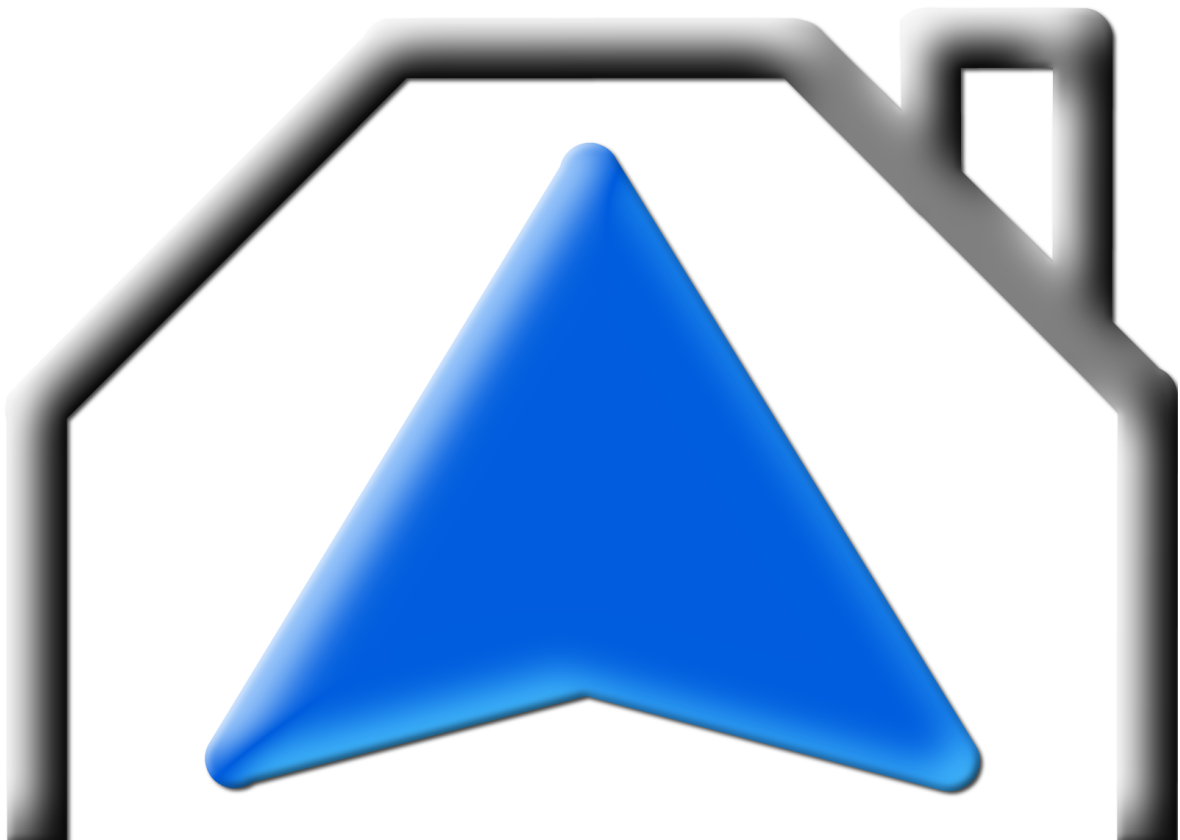


Scriptie

Indoor Navigation



Jochem Kempkens
Christian Litjes
januari 2012

Titelpagina

Gegevens studenten:

Naam + voorletters student
Studentnummer
Opleiding
Afstudeerperiode

Kempkens J.W.
2092401
Technische Informatica
29 augustus 2011 t/m 30 januari 2012

Naam + voorletters student
Studentnummer
Opleiding
Afstudeerperiode

Litjes C.F.
2108352
Technische Informatica
29 augustus 2011 t/m 30 januari 2012

Gegevens Bedrijf:

Naam bedrijf/instelling:
Plaats

TASS Technology Solutions
Eindhoven

Gegevens bedrijfsbegeleider:

Naam
Functie

Tiny Henst
People manager

Gegevens technisch begeleider:

Naam
Functie

Gerben Blom
People manager

Gegevens verslag:

Titel afstudeerverslag
Datum uitgifte afstudeerverslag

Indoor navigation
10 januari 2012

Getekend voor gezien door de bedrijfsbegeleider Tiny Henst:

Datum:

Voorwoord

Dit verslag is geschreven naar aanleiding van de afstudeerstage die plaatsvindt in het vierde en daarbij laatste leerjaar van de opleiding Technische Informatica bij Fontys Hogescholen te Eindhoven. In de periode van augustus 2011 tot en met januari 2012 hebben we 95 dagen gewerkt aan onze afstudeeropdracht bij TASS Technology Solutions te Eindhoven. Tijdens deze periode hebben we onderzoek gedaan naar Indoor Navigation om gebruikers binnen gebouwen naar hun bestemming te leiden.

In dit verslag zullen we verdere toelichting geven over: de aanleiding van de opdracht, onze werkwijze en de gemaakte beslissingen. Ook het onderzoek en het gehele proces van ontwerpen tot en met implementeren worden in dit verslag duidelijk besproken.

We willen graag onze bedrijfsbegeleider Tiny Henst en onze technisch begeleider Gerben Blom bedanken voor hun betrokkenheid, technische ondersteuning en begeleidingsgesprekken. Ook willen we Jos Verhagen, onze docentbegeleider, bedanken voor het bewaken van het afstudeerproces en zijn waardevolle feedback op het afstudeerverslag.

Aangezien de stage uitgevoerd is in duo-vorm willen we tenslotte het voorwoord gebruiken om aan te geven door wie de hoofdstukken die volgen geschreven zijn.

1. Inleiding	Jochem Kempkens
2. Het bedrijf	Jochem Kempkens
3. De opdracht	Christian Litjes
4. De aanpak	Christian Litjes
5. Onderzoek	Jochem Kempkens en Christian Litjes
6. Requirements	Jochem Kempkens
7. Ontwerp	Christian Litjes
8. Implementatie	Jochem Kempkens
9. Conclusie en aanbevelingen	Christian Litjes

Inhoudsopgave

Samenvatting	6
Summary	7
Verklarende woordenlijst	8
1 Inleiding	9
2 Het bedrijf	10
3 De opdracht	11
3.1 Voorafgaand	11
3.2 Opdracht	11
3.3 Afhankelijkheden	12
3.4 Opdrachtswijziging	12
4 Aanpak	13
4.1 Project Management Plan	13
4.2 Requirements	13
4.3 Ontwerp	13
4.4 Implementatie	13
4.5 Scriptie	14
5 Onderzoek	15
5.1 Positioning	15
5.2 Fingerprinting	16
6 Requirements	18
6.1 Eisenpakket	18
6.2 Toelichting	18
7 Ontwerp	21
7.1 Ontwerpmethode	21
7.2 Klassendiagram	21
7.3 Data Flow Diagram	22
7.4 Softwarepakketten	22
7.5 Ontwerpfase van A tot Z	22
8 Implementatie	23
8.1 Ontwikkelomgeving	23
8.2 Versiebeheer	23
8.3 Implementatieproces	23
9 Conclusie en aanbevelingen	26
Evaluaties	27
Literatuurlijst	29

Bijlagen

- A - Project Management Plan**
- B - Onderzoeksdocument Positioning**
- C - Onderzoeksdocument Fingerprinting**
- D - UML-diagram**

Lijst van figuren

1.1	High-end smartphones	9
1.2	Indoor navigatie	9
2.1	TASS hoofdkantoor	10
2.2	Organigram TASS Technology Solutions	10
3.1	Androidtelefoon	11
4.1	V-model	13
4.2	LaTeX	14
5.1	Logo WiFi	15
5.2	Plaatsbepaling via WiFi	16
6.1	Android-logo	18
6.2	Weergave positie	19
6.3	Weergave route	19
7.1	DFD niveau 0	21
7.2	Stuk uit DFD niveau 1	22
8.1	Eclipse	23
8.2	TortoiseSVN	23
8.3	Applicatie-logo	23
8.4	Selecteren van een gebouw	24
8.5	Blackbox-testing	25
8.6	Monkey-testing	25

Samenvatting

TASS Technology Solutions is een dienstverlener op het gebied van technische- en embedded software. Een mensgerichte organisatie die al meer dan 30 jaar voorop loopt in de ontwikkeling van software voor technisch hoogwaardige producten. Eerst als onderdeel van Philips en sinds 2007, met succes als zelfstandige organisatie. TASS richt zich met name op de volgende vijf marktgebieden: Mobility & Automotive, Healthcare & Cure, Mechanics & Control, Consumer Lifestyle en Safety & Security.

TASS had de mogelijkheid om een applicatie te realiseren om binnen gebouwen te kunnen navigeren. Om deze opdracht vorm te geven is er begonnen met een Project Management Plan en een onderzoek om meer over het onderwerp te weten te komen. Dit onderzoek heeft ervoor gezorgd dat er is gekozen om de plaatsbepaling te gaan realiseren met behulp van WiFi-signaalsterktes.

Uit verder onderzoek is gebleken dat er verschillende manieren zijn om met WiFi-signaalsterktes een locatie te bepalen. De meest geschikte methode bleek Fingerprinting te zijn. Bij deze techniek wordt van een gebouw eenmalig een kalibratiekaart gemaakt. Op verschillende punten worden metingen opgeslagen en wanneer een gebruiker later de positie wil bepalen, worden de signalen van dat moment vergeleken met de eerder opgeslagen metingen. Deze methode bleek het meest nauwkeurig en goed te implementeren op de hardware van een telefoon.

Na het doen van onderzoek en het opzetten van de planning zijn de requirements opgesteld. Na overleg met de klant is er gekozen om deze eenvoudig en duidelijk op te schrijven met de belangrijkste eisen op een rij. De belangrijkste eis aan de applicatie was het platform waarop het zou moeten draaien. Hiervoor is Android 2.1 en hoger gekozen.

Om een goed beeld te krijgen van de software is ervoor gekozen om een design op hoog niveau te maken volgens de UML-standaard. De onderdelen zijn vervolgens uitgewerkt in pseudocode om een goed beeld te krijgen van de implementatie van deze blokken. Door deze manier van ontwerpen was uiteindelijk het implementeren een eenvoudige opdracht. Binnen korte tijd was een werkende applicatie beschikbaar.

Na de afstudeerperiode is er een applicatie beschikbaar die met voldoende WiFi-signalen in staat is om een positie op kamerniveau te bepalen en de gebruiker via de kortste route naar de bestemming te leiden. Het aantal draadloze toegangspunten dat hiervoor nodig is ligt echter vele malen hoger dan normaal nodig is om iedereen van internet te voorzien. De applicatie zal dus niet zonder meer in ieder gebouw werken en de kans bestaat dat er nog veel hardware moet worden aangeschaft om het systeem operationeel te maken. De applicatie op zich werkt dus zoals verwacht alleen het toepassen in de praktijk is niet optimaal door het hoge aantal routers dat nodig is.

Summary

TASS Technology Solutions is a service provider of technical and embedded software. It is build around people and is already 30 years the leading developer of technical and embedded software. TASS has been a part of Philips however TASS became an independent organisation in 2007 and focussed on five segments of the market: Mobility & Automotive, Healthcare & Cure, Mechanics & Control, Consumer Lifestyle and Safety & Security.

TASS would like to have an Indoor Navigation application based on Android. At the start of each project a Project Management Plan should be made and research was started to find out if indoor navigation was possible and what techniques should be used to realise this. After the research it became clear that indoor navigation was possible based on WiFi-signal strengths.

The details of the techniques based on WiFi-signal strengths where unknown. That is why a new research was started. When the research was done there could be concluded, the Fingerprinting-technique was the way to go. It was the easiest and most accurate way to realise indoor navigation on an Android phone. The technique works by calibrating the whole building, so the application takes measurements of signal strength throughout the whole building. These measurements were later on used to match against current measurements to calculate the current location.

When all the research was completed, a work breakdown and requirements specification was made. These where made in deliberation with the client. The client wanted a clear and understandable work breakdown and requirements. All the requirements where really standard for navigation software but it had to work on Android 2.1 and above.

Every developer starts a project with software design which is based on the requirements of the client. There are a couple of standards for designing software. The chosen standard was UML to design the indoor navigation application in combination with pseudocode. The pseudocode was chosen to get a good understanding of the design and check if the design would actually work.

When the project was getting to an, end an indoor navigation application for Android was realized. Indoor navigation was possible with a high density of routers placed throughout the building. This means that it would not work out of the box in most buildings, because the density of routers is to low in standard office buildings. So it should be increased to get indoor navigation possible which results in a costly deployment of this application. Nevertheless the application works in the way it was designed and approved by the client. This meant it is possible to navigate inside a building but with a high density of routers spread out through the building.

Verklarende woordenlijst

Fingerprinting	Een kalibratiemethode om de nauwkeurigheid van de positie te kunnen verhogen.
GPS	Global Positioning System. Manier om te navigeren met behulp van satellieten.
Proof of Concept	Basisimplementatie om aan te tonen dat de bedachte oplossing in de praktijk ook echt werkt.
Indoor Navigatie	Navigeren binnen een gebouw.
Project Management Plan	Een document waarin het project wordt vastgelegd met randvoorwaarden en risico's.
UML	Een ontwerpmethodiek die internationaal geaccepteerd is.
Klassendiagram	Een onderdeel van UML waarin de software in objecten en functies is uitgewerkt.
Data Flow Diagram	Een onderdeel van UML waarin de flow van gegevens en beslissingen van de software zijn uitgewerkt.
Technische software	Software die ontwikkeld is voor technische toepassingen waarin hardware aangestuurd wordt op één niveau hoger dan embedded software.
Embedded software	Software die ontwikkeld is om de aansturing van de hardware mogelijk te maken.
WiFi	Een signaal op de 2.4 en/of 5.0 GHz band die onder andere gebruikt wordt voor draadloos internet.
Android	Een besturingssysteem van Google dat voornamelijk gebruikt wordt voor telefoons en tablets.
Pseudocode	Code die begrijpelijk en programmeertaal-onafhankelijk geschreven is.
IOS	Een besturingssysteem van Apple dat gebruikt wordt voor Apple producten.
Windows Phone	Een besturingssysteem van Microsoft dat voornamelijk gebruikt wordt voor telefoons.
Smartphone	Een telefoon die in staat is om naast bellen en sms'en dezelfde functies te bieden als een computer.
V-model	Een softwareontwikkelmethode.
Hardware	Een elektronisch component of apparaat.
Router	Een zender van WiFi signalen en zorgt voor het routeren van data tussen verschillende apparaten die verbonden zijn met de router.
Software engineer	Ontwikkelaar van software.
Hatley-Pirbhai	Een ontwerptechniek die veel gebruikt bij het ontwerpen van software waarop UML niet toegepast kan worden.
Visio	Tekenprogramma van Microsoft en is onderdeel van het Office-pakket.
GUI	Grafische User Interface is het interactie-deel van een applicatie met de gebruiker.

Hoofdstuk 1

Inleiding

Het aantal smartphonegebruikers is de laatste jaren enorm gestegen. Meer dan de helft van alle mobiele telefoons is tegenwoordig een smartphone op: IOS, Android of Windows Phone. Ook de sterk toenemende technologische ontwikkelingen binnen de mobiele telefoons zorgen voor vele nieuwe toepassingen. TASS is als dienstverlener op het gebied van technische- en embedded software altijd geïnteresseerd in de nieuwste ontwikkelingen op dit gebied.

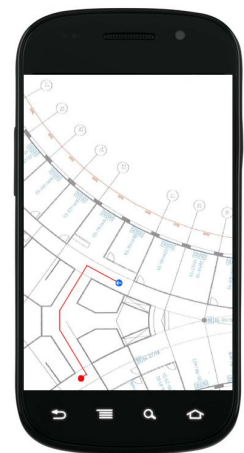


Figuur 1.1: High-end smartphones

Met de nieuwe technologieën en vernieuwende hardware zijn applicaties mogelijk die dat eerst niet waren. Zo heeft TASS de mogelijkheid gezien een opdracht op te stellen om een Android-applicatie te ontwikkelen voor navigatie voor binnen. Navigatie met behulp van een GPSontvanger voor buiten bestaat al vrij lang. Deze GPS-signalen zijn echter vaak niet toereikend voor binnen. Met nieuwe smartphones die beschikken over een uitgebreide range aan sensoren zijn er tegenwoordig ook andere oplossingen mogelijk.

Het doel van de afstudeeropdracht is dan ook het ontwikkelen van een applicatie die in staat is om een gebruiker binnen een gebouw de weg te wijzen. De gebruiker zal hiervoor een gewenste bestemming opgeven en de applicatie zal de gebruiker vervolgens de kortste route naar die bestemming geven. De technieken die hiervoor gebruikt worden zijn te vinden in de verschillende hoofdstukken van deze afstudeerscriptie.

Het hoofdstuk hierna geeft informatie over het bedrijf waar het afstuderen heeft plaatsgevonden. Daarna wordt in detail de opdracht besproken evenals de manier waarop deze aangepakt gaat worden. Na dit inleidende deel wordt de uitwerking beschreven aan de hand van: de onderzoeksmethoden, de requirements, het ontwerp en de implementatie. Tot slot is er een conclusie en een evaluatie van beide afstudeerders.



Figuur 1.2: Indoor navigatie

Hoofdstuk 2

Het bedrijf

TASS technology solutions is een dienstverlener op het gebied van technische- en embedded software. Een mensgerichte organisatie die al meer dan 30 jaar voorop loopt in de ontwikkeling van software voor technisch hoogwaardige producten. Eerst als onderdeel van Philips en sinds 2007, met succes als zelfstandige organisatie.

Dit maakt TASS tot een dynamische en betrouwbare partner, die snel kan schakelen en tijdig weet in te springen op ontwikkelingen in de markt. TASS gaat duurzame relaties aan met medewerker en klant. Relaties gedreven door toewijding en gebaseerd op vertrouwen.

Als top-3 dienstverlener voor technische en embedded software, levert TASS een belangrijke bijdrage aan de ontwikkeling van intelligente en innovatieve producten.

De software professionals van TASS zetten hun kennis, kunde en toewijding om in betere oplossingen, binnen de beschikbare time-to-market. Zo helpt TASS haar klanten, om kwalitatief hoogwaardige producten nog efficiënter en sneller in de markt te zetten.

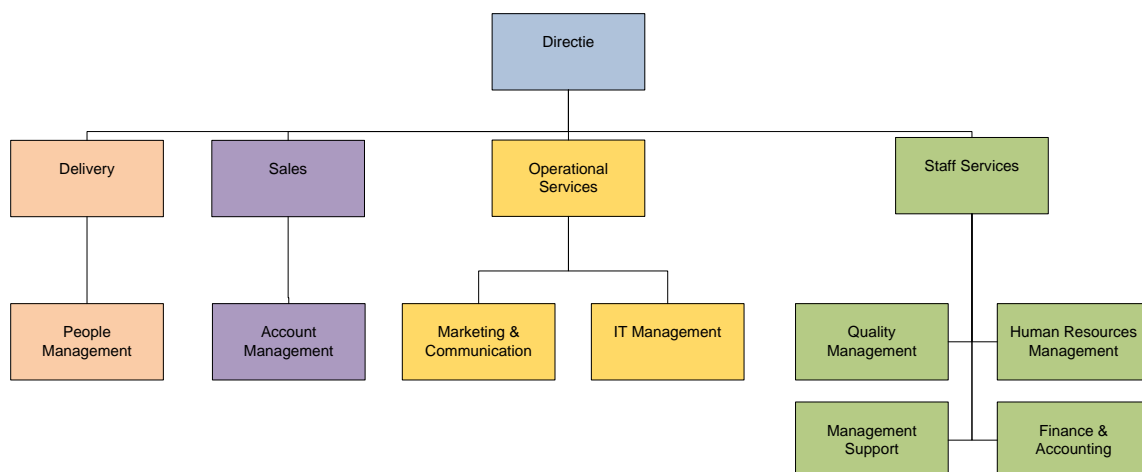


Figuur 2.1: TASS hoofdkantoor

Momenteel heeft TASS een viertal vestigingen. In Nederland: Apeldoorn en Eindhoven. In België: Leuven en Gent.

Bij TASS werken ruim 200 software professionals, waarvan het merendeel gedetacheerd is bij één van de vele klanten. De afstudeerlocatie is het gebouw op het Larixplein te Eindhoven geweest.

Hieronder is een organigram te zien van TASS. De positie van de afstudeerders binnen dit organigram is onder het blok People Management.



Figuur 2.2: Organigram TASS Technology Solutions

Hoofdstuk 3

De opdracht

3.1 Voorafgaand

Halverwege het laatste leerjaar moet er gezocht worden naar een afstudeeropdracht. Het bedrijf lag al vast en het moest TASS worden in verband met goede ervaringen via school in het verleden. Omdat TASS gebruik maakt van afstudeeropdrachten in duo's is ervoor gekozen om samen hierop in te schrijven. De keuze om hierop in te schrijven en samen te gaan werken was heel simpel. Na drie jaar prettig samengewerkt te hebben in verschillende projecten was duidelijk wat van elkaar de goede en slechte eigenschappen waren. Door dit duidelijke beeld wordt de samenwerking aan deze afstudeeropdracht niet als risico gezien. Dit maakte de keuze van een opdracht ook heel simpel omdat het beeld van de opdracht voor ons beide hetzelfde was. Het moest een opdracht worden bij TASS op het Android besturingssysteem.

Het zoeken van een opdracht wordt voor de studenten erg makkelijk gemaakt door Fontys. Hiervoor hebben ze het SAS+ systeem in het leven geroepen. In dit systeem staan een groot aantal afstudeeropdrachten waaruit gekozen kan worden. Naast dit systeem mag er ook zelf gezocht worden naar een geschikte opdracht. Op eigen initiatief is er een gesprek ingepland met TASS. Er werden verschillende opdrachten voorgelegd. Deze opdrachten stonden nog niet in het SAS+ systeem van Fontys waardoor er vrij gekozen kon worden. Dit gaf als voordeel dat er niet nog tien andere studenten waren die op dezelfde opdracht aasden. Na een goed gesprek was er de mogelijkheid om een half uur na te denken over de opdrachten. Al snel werd er gebrainstormd over de opdracht; wat houdt het in en welke technieken zijn al bekend. Hierdoor was de keuze snel gemaakt. Er was tenslotte maar één opdracht voor het Android besturingssysteem en dat was al een pré voordat de zoektocht begonnen was.

3.2 Opdracht

De bedoeling van de opdracht was dat er een Android applicatie kwam waarmee de positie van een persoon bepaald kon worden binnen een gebouw. De persoon beschikt over een apparaat met het Android besturingssysteem, dit kan een telefoon of tablet zijn. Er was nog niet gespecificeerd hoe de locatiebepaling gedaan moest worden. Hiervoor moest een onderzoek gestart worden. Maar door het gebruik van een Android apparaat worden de mogelijkheden voor positiebepaling al snel beperkt. Het is natuurlijk niet de bedoeling dat de persoon nog een extra apparaat mee moet nemen. De applicatie is afhankelijk van de sensoren die al aanwezig zijn in een Android apparaat. Gelukkig worden er een aantal sensoren verplicht door het Android besturingssysteem zelf. Dit verschilt helaas wel per versie maar wanneer er gebruik gemaakt wordt van versie 2.1 dan zijn: het kompas, draadloos internet, gps en bluetooth beschikbaar. Met één van deze sensoren of een combinatie hiervan moest de positie bepaald gaan worden. Extra hardware die nodig was voor een eventuele infrastructuur of uitbreiding daarvan mocht later nog aangeschaft worden.



Figuur 3.1: Androidtelefoon

3.3 Afhankelijkheden

Bij de opdracht waren een aantal dingen heel duidelijk: er is nog helemaal niets, het moet op Android draaien, er moet duidelijke documentatie komen en de technieken om het doel te bereiken zijn niet bekend. De aanpak van het project zal volgens het V-model gaan en er wordt gebruik gemaakt van de standaard projectdocumenten van TASS.

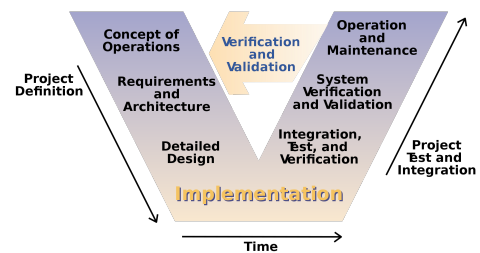
3.4 Opdrachtswijziging

Al snel was duidelijk dat de locatiebepaling niet erg moeilijk was. Binnen een paar werken lag er al een proof of concept. Daarom heeft de opdrachtgever besloten om navigatie naar een bestemming toe te voegen. Dit was gelukkig al voor de ontwerpfase dus hiermee is rekening gehouden.

Hoofdstuk 4

Aanpak

Bij dit project is gebruik gemaakt van het V-model. Dit is een methode om een project op een gestructureerde manier aan te pakken. Hiervoor is gekozen omdat bij het V-model eerst een gedetailleerde planning wordt gemaakt. Hiermee wordt voorkomen dat er dingen op de lange baan geschoven worden waardoor de afstudeeropdracht in gevaar komt.



Figuur 4.1: V-model

4.1 Project Management Plan

Aan het begin van het project is er een Project Management Plan (PMP) opgesteld. In dit document wordt de opdracht omschreven en het project afgebakend. Dit document wordt vervolgens door de klant goedgekeurd. Later kan dit gebruikt worden om aan te tonen dat het opgeleverde project voldoet aan de eisen van de klant zoals deze toen gesteld waren.

4.2 Requirements

Bij het PMP hoort ook een planning maar deze kan pas na de goedkeuring van het PMP opgesteld worden. Na de goedkeuring van de planning en het PMP kan gestart worden aan de specificatie-fase. Tijdens de specificatie-fase worden er twee onderzoeken gestart, Functionele Requirements Specificatie (FRS) en Software Requirements Specificatie (SRS) opgesteld. Er zal onderzocht worden welke technieken er bruikbaar zijn en vervolgens wordt er in detail uitgezocht hoe een techniek werkt.

4.3 Ontwerp

Met duidelijke en heldere specificaties kon er begonnen worden aan de ontwerpfase. Tijdens de ontwerpfase is er een UML-diagram gemaakt, pseudocode geschreven en zijn er tal van proof of concepts gemaakt en hergebruikt uit de onderzoeksfase. De pseudocode werd gebruikt om na te gaan of threading en het gebruik van de Graphical User Interface (GUI) op de manier werkte zoals gedacht was. Na de ontwerpfase kwam de implementatiefase, tijdens deze fase werd het product gerealiseerd en werden pseudo-stukken naar Android-specifieke code omgeschreven.

4.4 Implementatie

Nadat een stuk geïmplementeerd was werd dit stuk code getest. Als het getest was werd de code gebruikt in het hoofdprogramma. Als er nu fouten optreden kan er met zekerheid gezegd worden dat deze in de code hiervoor of erachter zitten. Dit betekent dat of het ontwerp niet goed gevolgd is of dat er een fout in het ontwerp zit waardoor het geheel omgevallen is.

4.5 Scriptie

Aan het einde van dit traject wordt de scriptie geschreven. Dit is gebeurt in LaTeX, dit is een programma waar vaak technische rapporten in worden geschreven. Er is gekozen voor deze methode omdat dit het er professioneler uitziet en er tijdens het schrijven van de scriptie geen rekening gehouden hoeft te worden met de opmaak. Dit gaat in LaTeX veel gemakkelijker als in Microsoft Word omdat bij LaTeX de layout voor het hele document op voorhand wordt vastgelegd. Deze opmaak is leidend en dus niet op een simpele manier toch te veranderen. Ook het genereren van de Inhoudsopgave, Lijst van figuren en tabellen gaat vanzelf. LaTeX werkt deze elke keer bij wanneer het document opnieuw wordt opgeslagen. Er hoeft dus totaal geen rekening meer met de layout gehouden te worden op het moment dat de teksten geschreven worden.



Figuur 4.2: LaTeX

Hoofdstuk 5

Onderzoek

Een belangrijk onderdeel van de afstudeeropdracht is het doen van onderzoek. Onderzoek is nodig om kennis op te doen van de nieuwe materie en te zorgen voor de juiste basis om tijdens het afstuderen diverse beslissingen te kunnen nemen. Om deze onderzoeken een wat hoger niveau mee te geven dan slechts wat zoekwerk op internet is ervoor gekozen door beide afstudeerders om een onderzoeksdocument te maken. Dit onderzoeksdocument bevat de werkwijze waarop het onderzoek is uitgevoerd. Beide onderzoeken zijn als bijlagen bij dit document toegevoegd. In de twee paragrafen die volgen wordt door beide afstudeerders een korte samenvatting gegeven van het onderzoeksrapport en een toelichting op de keuzes die gemaakt zijn en de werkwijze die is toegepast. Uiteraard zijn de onderzoeksresultaten de grondslag geweest voor alle overige documenten en resultaten die in dit document worden beschreven maar deze zullen verder niet specifiek worden genoemd. Voor de complete onderzoeksopzet- en resultaten kan het volledige document in de bijlagen worden geraadpleegd.

5.1 Positioning

Het onderwerp dat centraal staat in het onderzoek “Positioning” is het bepalen van de huidige positie binnen in een gebouw. Een positie bepalen met behulp van GPS is niet echt een uitdaging maar dit verandert snel zodra je een gebouw binnen loopt. GPS-signalen zijn binnen in gebouwen niet betrouwbaar en vallen snel weg als gebruik wordt gemaakt van niet heel dure GPS-ontvangers, zoals die in telefoons. Om binnen toch een goede positie te kunnen bepalen volgen hieronder een aantal technieken die onderzocht zijn met bij iedere techniek een korte omschrijving:

- Beeldherkenning; Door middel van camera’s de positie van personen bijhouden.
- Bluetooth; Door middel van meerdere bluetooth-dongles de dichtstbijzijnde bepalen.
- GPS; Door middel van verschillende satellieten een positie bepalen.
- RFID; Door middel van meerdere RFID-dongles de dichtstbijzijnde bepalen.
- Telefoonmasten; Door middel van een meting de positie bepalen tussen telefoonmasten.
- WiFi; Door middel van signaalsterktes de positie tussen routers bepalen.
- ZigBee; Door middel van meerdere ZigBee-dongles de dichtstbijzijnde bepalen.

WiFi is als beste techniek uitgekozen vanwege: de relatief lage kosten, het feit dat gebruikers alleen een telefoon met WiFi nodig hebben en de relatief hoge nauwkeurigheid.

Met WiFi als meest geschikte techniek zijn er verschillende methoden om een positie te bepalen:

- Tijd naar afstand; De tijd omrekenen die een signaal vanaf de diverse routers nodig heeft om de telefoon te bereiken en zo een positie bepalen.
- Driehoeksmeting; De signaalsterktes vanaf de diverse routers omrekenen naar afstanden en zo een positie bepalen.



Figuur 5.1:
Logo WiFi

- **Fingerprinting;** Een kalibratie uitvoeren en gegevens opslaan van alle signaalsterktes. Wanneer dan een positie bepaald gaat worden, worden de op dat moment aanwezige signaalsterktes vergeleken met de opgeslagen data en zo wordt dan een positie bepaald.

Uit bovenstaande opties is gekozen voor “Fingerprinting” als meest geschikte techniek. Deze techniek is de enige waarbij een nauwkeurigheid op kamerniveau te behalen is die ook nog redelijk betaalbaar is. De andere opties zijn te duur of halen de nauwkeurigheid niet.

De hoofdvraag die in dit onderzoek centraal staat is: “Op welke manier kan het beste in een gebouw een nauwkeurige positie worden bepaald en hoe gaat deze methode in zijn werk?”

Een kort antwoord hierop geeft WiFi als beste techniek omdat deze: redelijk betaalbaar, behoorlijk nauwkeurig en goed te implementeren is. De beste techniek om de positie te bepalen is door middel van een kalibratie, die Fingerprinting wordt genoemd, waarover in de volgende paragraaf meer wordt verteld.

5.2 Fingerprinting

Toen fingerprinting een goede techniek bleek te zijn om een locatie te bepalen was het natuurlijk een logische stap om dit eerst goed te onderzoeken. Op internet waren al tal van rapporten te vinden waarin beschreven staat hoe deze techniek toegepast wordt en wat voor hardware er nodig is. Het fingerprinten is op zich een simpele techniek, dit gebeurt door middel van het in kaart brengen van de signaalsterkte van alle routers in een gebouw. Elke meting bevat de signaalsterktes van alle routers binnen bereik en de locatie in het gebouw waarop de meting plaatsvond, dit wordt een fingerprint genoemd. Deze metingen worden vervolgens opgeslagen in een database. Wanneer deze database gevuld is gebruiken we deze metingen om te bepalen waar een persoon zich bevindt. Dit gebeurt als volgt: er wordt een nieuwe meting gedaan van de signaalsterktes. Deze meting wordt vergeleken met de metingen in de database. Van deze metingen worden de drie meest overeenkomstige gebruikt. Bij deze drie metingen zitten ook de drie locaties waarop de meting hebben plaatsgevonden binnen het gebouw. We middelen nu deze drie locaties. Hieruit ontstaat een geschatte locatie waarop de persoon zich nu bevindt.



Figuur 5.2: Plaatsbepaling via WiFi

Helaas was er nergens een voorbeeldproject te vinden wat zo getest kon worden. Dit betekende dat er een proof of concept gemaakt moest worden aan de hand van de gevonden rapporten. Dit was zo in elkaar gezet. Hiermee was goed aan te tonen dat we twee tot drie meter nauwkeurigheid konden halen. Nu moest de techniek uitgediept worden om eventueel dingen te verbeteren maar bovendien goed te begrijpen hoe de techniek werkt. Voor het geval dat er gaandeweg zich onverwachte dingen voordeden. Natuurlijk gebeurden er onverwachte dingen tijdens het testen van de proof of concept. Hierin kwamen meteen alle nadelen van de techniek naar boven.

- Aantal routers.
- Verplaatsing van routers.
- De gebruiker zelf.

Het aantal routers dat we gebruiken is hoog. Dit komt omdat je overal in het gebouw minimaal drie routers met een hoge signaalsterkte binnen bereik moet hebben. Dit betekent dat je om de tien meter in het gebouw een router neer moet leggen. Wanneer er om de twintig meter routers geplaatst worden zakt de nauwkeurigheid naar twee tot zeven meter wat niet het gewenste kamerniveau is. Daarom worden er om de tien meter routers geplaatst die hier dan ook definitief blijven liggen en deze mogen niet verplaatst worden. Omdat de signalen in kaart van deze routers in kaart gebracht worden is het natuurlijk funest wanneer één van deze routers verplaatst wordt omdat dan de kaart niet meer klopt. Niet alleen het verplaatsen van de routers bleek een groot probleem te zijn maar ook de gebruiker zelf. Aangezien routers radiosignalen uitzenden en mensen radiosignalen tegenhouden, is het moeilijk om een hogere nauwkeurigheid dan drie meter te behalen.

Al met al lijkt deze techniek geen geschikte methode voor locatiebepaling. Alleen al omdat er zoveel routers nodig zijn. Omdat deze opdracht bedoeld is om toegepast te worden in het Deventer ziekenhuis is het wel een goede techniek. Het Deventer ziekenhuis beschikt nu al over een heel hoge dekkinggraad van routers in het gebouw, waardoor deze techniek ideaal is.

Hoofdstuk 6

Requirements

In dit hoofdstuk wordt besproken aan welke eisen de applicatie voldoet. Na het doen van onderzoeken en het verkennen van het onderwerp is er in overleg met de klant een lijst van eisen gekomen die de requirements vormen. In dit hoofdstuk komt aan bod waarom de requirements zijn zoals ze zijn en waarom ze op deze manier zijn opgesteld.

6.1 Eisenpakket

Hieronder volgt een globale lijst van de eisen waaraan de applicatie zal moeten voldoen. Daarna zal bij verschillende onderdelen worden besproken waarom de requirement is zoals deze hier op papier staat.

- Een applicatie die draait op Android, minimaal versie 2.1.
- De applicatie dient per type telefoon gekalibreerd te worden.
- Een kalibratie vereist geen technische kennis.
- De applicatie is in staat zijn positie binnen een gebouw te bepalen zowel horizontaal als verticaal.
- De applicatie kan de locatie van de gebruiker weergeven op een kaart.
- De gebruiker kan een bestemming opgeven.
- De applicatie geeft de gebruiker middels een kaart de snelste looproute.
- De applicatie heeft een nauwkeurigheid op kamerniveau.

Per requirement volgt een toelichting en eventuele extra informatie om te zorgen dat duidelijk is waarom de requirements zijn zoals genoemd en waarom de klant bepaalde wensen heeft.

6.2 Toelichting

De Android-versie van minimaal 2.1 is gekozen om een aantal redenen. Vanaf deze versie zijn er een aantal technieken bijgekomen die het schrijven van een applicatie met de gestelde eisen makkelijker maken. Zo is het gebruik van de sensoren aangepast waardoor ze makkelijker te gebruiken zijn. Daarnaast is de minimale hardware die nodig is voor versie 2.1 tevens gewenst bij de applicatie. Er is behoorlijk wat rekenwerk nodig om tot een mooie gestabiliseerde positie te komen en dus kunnen niet alle oudere telefoons de applicatie vloeiend draaien. Een minimale Android-versie zorgt dat de hardware altijd toereikend is om vloeiend te kunnen navigeren. Deze requirement is tot stand gekomen door onderzoek van de afstudeerders die hebben bepaald dat geen ondersteuning kan worden geboden op lagere versies.



Figuur 6.1:
Android-logo

Kalibratie per type telefoon

De applicatie dient per type telefoon gekalibreerd te worden. Deze requirement is ook afkomstig vanuit onderzoeken van de afstudeerders die verschillende technieken voor plaatsbepaling hebben onderzocht. Uit die onderzoeken is gebleken dat de nauwkeurigste methode een kalibratie vereist. De nauwkeurigheid hangt van veel factoren af waaronder: de kwaliteit en plaatsing van de draadloze antenne in een telefoon. Deze varieert veel tussen verschillende type telefoons en daarom zal per type telefoon een kalibratie nodig zijn. Dit is nodig om te zorgen dat met verschillende merken en type telefoons een goede positie wordt bepaald. Deze requirement kan worden gezien als lastig en tijdrovend. Er is echter geen betere oplossing te bedenken om een goede positie te bepalen en dus zal een groep gebruikers de afweging moeten maken tussen één type telefoon aanschaffen of meermalen een kalibratie uitvoeren om alle typen te ondersteunen.

Kalibreren zonder technische kennis

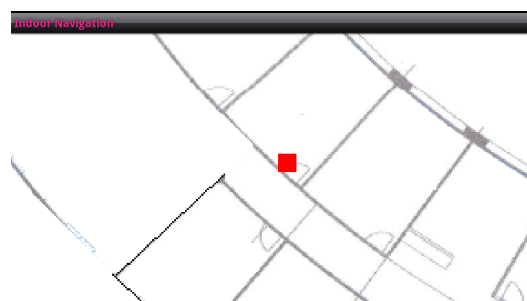
Aangezien een kalibratie opnieuw moet worden gedaan bij nieuwe telefoons en significante veranderingen aan de infrastructuur van het netwerk of het gebouw zelf, is een wens van de klant eenvoudige kalibratie. Zo hoeft de klant niet bij iedere wijziging een technicus te laten komen om opnieuw te kalibreren. De kalibratie wordt zo eenvoudig gehouden dat mensen zonder enige kennis van het systeem toch de kalibratie kunnen uitvoeren. Dit scheelt in kosten voor de klant en maakt wachten op een technicus onnodig.

Positiebepaling zowel horizontaal als verticaal

De belangrijkste functionaliteit van het programma is het bepalen van een positie binnen een gebouw zowel horizontaal als verticaal. De horizontale positie bepaald waar op een verdieping de gebruiker zich bevindt. De verticale positie zorgt ervoor dat er op meerdere verdiepingen een positie kan worden bepaald.

Locatie weergeven op kaart

Naast het bepalen van een positie moet het programma deze ook duidelijk aan een gebruiker kunnen tonen. Hiervoor wordt per verdieping van een gebouw een kaart ingeladen zodat een positie kan worden weergegeven op een plattegrond van een gebouw. Zo is het voor de gebruiker gemakkelijk om de weg te vinden binnen grote gebouwen.

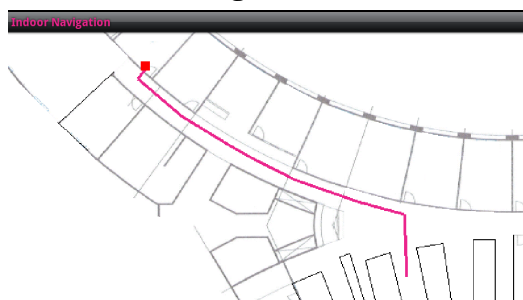


Figuur 6.2: Weergave positie

Bestemming invoeren

Een kaartweergave met een positie is een handig hulpmiddel voor een gebruiker. Nog meer comfort geeft de mogelijkheid om een bestemming op te geven. Zo kan een gebruiker naast de huidige locatie ook de bestemming zien en daar sneller heen gaan.

Kortste route weergegeven



Figuur 6.3: Weergave route

De klant heeft ook aangegeven graag navigatie te willen hebben in de applicatie. De applicatie zal op basis van de bestemming en de huidige locatie van de gebruiker de kortste route bepalen om die afstand af te leggen. Op de kaart zal deze kortste route weergegeven worden als een verbindinglijn tussen het start- en eindpunt. Een gebruiker hoeft zo alleen maar de lijn te volgen om op de plaats van bestemming te komen.

Nauwkeurigheid

Om binnen een gebouw goed te navigeren is een bepaalde nauwkeurigheid nodig. Aangezien WiFi-signalen erg afhankelijk zijn van vele factoren is hier moeilijk een nauwkeurigheid in meters aan te hangen. Een WiFi-sigitaal verandert in open ruimtes namelijk niet genoeg om goed een precieze positie te bepalen. De klant heeft aangegeven dat navigatie binnen een ruimte niet het grootste belang heeft, aangezien dit ook door de gebruiker zelf gezien kan worden. Belangrijker is het navigeren tussen verschillende ruimtes omdat dit niet fysiek te zien is. Daarom is vanuit de klant een nauwkeurigheid op kamerniveau gewenst omdat dit nodig is om soepel te kunnen navigeren tussen de verschillende ruimtes van een gebouw.

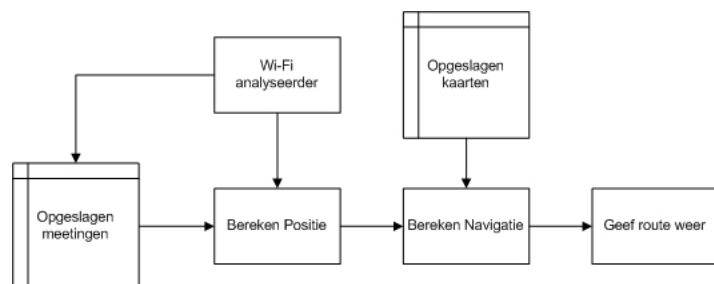
Hoofdstuk 7

Ontwerp

In de ontwerpfase wordt de software beschreven in diagrammen en overzichten. Hiermee is op een abstract niveau duidelijk hoe dat een programma in elkaar zit. In deze fase wordt veel gebruik gemaakt van gestandaardiseerde ontwerpmethodes. De reden hiervoor is natuurlijk erg voor de hand liggend. Op deze manier kan iedere software-engineer het ontwerp lezen. In de ontwerpfase van dit project is er deels gebruik gemaakt van een gestandaardiseerde methode. Normaal maak je het hele ontwerp volgens de UML-standaard, Hatley-Pirbhai of een andere standaard. In deze standaarden staat beschreven hoe een ontwerp voor een programma gemaakt wordt. Hieraan zitten ook regels verbonden met betrekking tot het tekenen van de diagrammen en het gebruik van symbolen. Naast de regels wordt er ook veel aandacht besteed aan de flow binnen het programma. Dit wil zeggen: Is het programma logisch opgezet en gaat de uitwisseling van gegevens op een makkelijke en logische manier?

7.1 Ontwerpmethode

In deze opdracht is er gekozen voor een combinatie van technieken. Er is maar een deel van UML gebruikt in combinatie met pseudocode. De volgende onderdelen van UML zijn gebruikt: het klassendiagram en de Data Flow Diagrammen (DFD) op niveau nul en één. Voor deze techniek is samen met de technisch begeleider gekozen omdat de voorkeur lag bij een



Figuur 7.1: DFD niveau 0

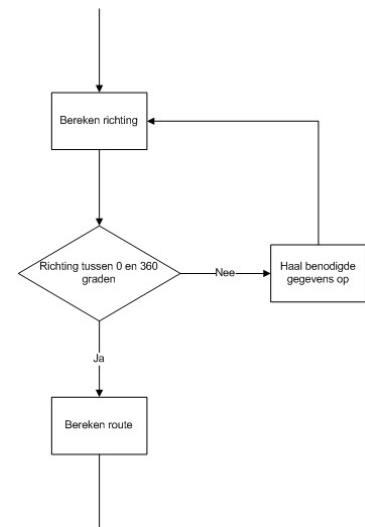
methode die overzichtelijk is en de implementatiefase kort hield. Hiermee hadden we meer grip op het project omdat in de ontwerpfase al duidelijk werd of bepaalde technieken wel of niet mogelijk zijn op het Android-platform. Mochten er toch dingen niet mogelijk zijn dan kon er snel ingegrepen worden waardoor niet de hele ontwerpfase weer opnieuw gedaan moest worden. Dit kan wel gebeuren als er eerst een ontwerp gemaakt werd volgens de UML-standaard en vervolgens gestart werd met de implementatie. Wordt er in het ontwerp één ding over het hoofd gezien dan kan dit resulteren in een herzien van het gehele ontwerp en eventueel in de implementatie. Dit is afhankelijk van wanneer de fout werd opgemerkt, wanneer dit in een vroeg stadium is dan hoeft er maar weinig aangepast te worden. Dit hoeft niet altijd heel erg te zijn maar gezien het feit dat binnen 95 dagen een product opgeleverd moet worden is deze extra zekerheid geen overbodige luxe.

7.2 Klassendiagram

Een klassendiagram beschrijft hoe dat het programma eruit komt te zien. Het programma wordt vervolgens opgedeeld in blokken en elk van deze blokken bevat weer eigen functionaliteiten. Hiermee probeer je een zo logisch mogelijk programma te ontwerpen. Dit is nodig omdat een programma al snel complex kan worden waardoor het overzicht snel zoek is. Voor een collega of een medestudent uit het vierde jaar is het nu makkelijker geworden om hierop verder te bouwen.

7.3 Data Flow Diagram

Een klassendiagram is normaal voldoende maar toch is het handig om ook een Data Flow Diagram (DFD) te hebben. Omdat in een DFD: de datastromen, functionaliteiten en beslissingen van de software staan omschreven. De functionaliteiten van een programma worden weergegeven in blokken. Elk blok staat weer in verbinding met een volgend blok. Vaak zitten er tussen twee blokken beslissingen. Deze worden weergegeven als een ruit. Als voorbeeld nemen we een blok met als functionaliteit; bereken richting (ter illustratie zie Figuur 7.2). Dit blok kan 361 verschillende richtingen berekenen. 361 Richtingen klinkt als een vreemde uitkomst omdat het niet mogelijk is om 361 graden rond je as te draaien maar 360 graden is wel mogelijk. Deze 361^e richting wordt gebruikt om kenbaar te maken of de berekening goed is gegaan. Want dit heeft natuurlijk gevolgen voor de rest van het programma. Zeker als de berekening fout gaat en er geen richting berekend kan worden. Het aangeven hiervan kan op verschillende manieren, de berekening geeft 361 of -1 als uitkomst. Dit zijn allebei mogelijke manieren om duidelijk te maken dat er iets fout is gegaan in de berekening. Deze zo voor de hand liggende fouten worden snel duidelijk in een DFD op niveau één. Omdat in het DFD snel zichtbaar wordt wat voor gevolgen het heeft voor functionaliteiten die hiervan afhankelijk zijn. Hierdoor zijn dit soort problemen al ondervangen voordat er ook maar één regel code geschreven is. Want dit zijn hele typische problemen die pas naar voren komen bij de implementatie- en testfase.



Figuur 7.2: Stuk uit DFD niveau 1

7.4 Softwarepakketten

Voor het tekenen van deze diagrammen zijn verschillende softwarepakketten beschikbaar. Maar deze pakketten zijn niet per definitie nodig zolang er maar aan de UML-tekenstandaarden gehouden wordt. Daarom is er voor Microsoft Office Visio 2010 gekozen. Hierin zit standaard al de mogelijkheid om UML-diagrammen te tekenen. Echter is dit geen specifiek UML-softwarepakket. Er zijn ook andere gratis programma's waarmee UML-diagrammen gemaakt kunnen worden maar voor deze simpele tekeningen en diagrammen was het niet nodig en voldeed Visio prima.

7.5 Ontwerpfase van A tot Z

Als eerste ontwerp is er een klassendiagram op hoog niveau gemaakt. Hierin staan alle basis bouw-blokken voor de software beschreven. Van deze basisblokken is een DFD op niveau nul gemaakt, waarin beschreven staat hoe dat de software op het hoogste niveau moet gaan werken. Nu het globale beeld gevormd was kon er nagedacht worden over de details. Daarom was er een DFD op niveau één gemaakt, hierin kwamen al snel dingen naar voren die over het hoofd gezien waren. Het ging vaak om terugkoppelingen in het programma zoals in het voorbeeld van een DFD op niveau één wordt omschreven. Nadat deze problemen ondervangen waren kon gestart worden met het maken van het klassendiagram op applicatie-niveau. Dankzij het klassendiagram op hoger niveau en het DFD op niveau één kon er op een snelle en gemakkelijke manier een klassendiagram op applicatie-niveau in elkaar worden gezet. Dit kwam voornamelijk doordat er veel tijd is gaan zitten in het uitdenken van het DFD op niveau één. Met het klassendiagram en het DFD kon er pseudocode geschreven worden om snel na te gaan of het ontwerp ook ging werken. Al snel bleek dat het in pseudocode wel ging werken maar toen werd het tijd om een aantal dingen te testen op Android. Toen bleek dat een aantal dingen toch op een andere manier werkte dan verwacht. Dit zorgde voor een aantal kleine aanpassingen in het ontwerp.

Hoofdstuk 8

Implementatie

8.1 Ontwikkelomgeving

De ontwikkelomgeving die is gebruikt voor het schrijven van de software is Eclipse. Dit is een opensource ontwikkelomgeving met vele features die het programmeren een stuk aangenamer maken. De standaardversie van Eclipse ondersteunt niet het schrijven van Android-code. Hiervoor is de softwareontwikkelkit van Google nodig. Deze toevoeging van Google zorgt dat alle specifieke Android-aspecten kunnen worden gebruikt in het te schrijven programma.



Figuur 8.1: Eclipse

8.2 Versiebeheer



Figuur 8.2: TortoiseSVN

Naast het gemakkelijker schrijven van code heeft Eclipse ook de mogelijkheid voor versiebeheer. Hiervoor kan een plugin worden geïnstalleerd van het versiebeheerprogramma TortoiseSVN. Dit programma zorgt ervoor dat de beide afstudeerders in hetzelfde programma kunnen werken en dat ze altijd de nieuwste versie van de code tot hun beschikking hebben. Naast het gemakkelijk samen coderen kan ook altijd een eerdere versie van de software worden teruggehaald. Iedere versie die via TortoiseSVN wordt geüpload krijgt een nummer en datum mee en kan altijd worden teruggehaald. Dit is handig als er in een later stadium bepaalde functionaliteiten niet meer blijken te werken. Deze kunnen dan met enkele muisklikken worden hersteld.

8.3 Implementatieproces

Na het maken van een ontwerp met behulp van een UML-diagram en het uitwerken van de verschillende klassen in pseudocode bleek het implementeren erg mee te vallen. Doordat alle functies van de verschillende klassen waren vastgelegd in simpel geschreven en duidelijke taal was het nu niet meer moeilijk om dit om te zetten naar goede code. Na enkele dagen was er dan ook al een werkend programma beschikbaar waarvan de meeste functionaliteiten goed werkten. Er waren wel enkele onderdelen waar extra aandacht aan geschonken moest worden en deze zullen hieronder worden toegelicht.



Figuur 8.3: Applicatie-logo

Kaartupdate

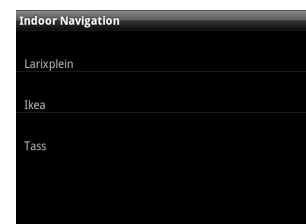
De eerste uitdaging die snel ontdekt werd was een probleem met het updaten van de kaart. Doordat er meerdere malen per seconde een nieuwe kaart op het scherm werd gezet, had de telefoon niet genoeg tijd om het geheugen leeg te gooien. Dit zorgde ervoor dat na enkele minuten navigeren het geheugen van de telefoon vol was en de applicatie afgesloten werd. Dit probleem was met het maken van pseudocode niet aan het licht gekomen omdat het geheugenbeheer op papier moeilijk na te bootsen is. Echter, na wat kleine aanpassingen in het updaten van de kaart en het efficiënter omgaan met het geheugen, is dit probleem goed verholpen.

Verdiepingen

Een ander punt dat bij het implementeren niet helemaal soepel verliep was het navigeren tussen meerdere verdiepingen. Bij het ontwerpen is hier niet veel aandacht aan besteed, omdat de focus altijd heeft gelegen op de positie bepalen en wat navigatiemogelijkheden maar nooit concreet navigatie tussen verdiepingen als prioriteit is gezien. Pas bij het implementeren is hier echt naar gekeken. Op dit moment is navigatie tussen verdiepingen meegenomen bij het implementeren, maar dit is niet uitvoerig getest. Alle aspecten die nodig zijn om dit mogelijk te maken zijn geïmplementeerd. Een volledige test voor de correcte werking zal pas worden uitgevoerd wanneer er tijd over is aan het eind van het afstuderen, maar dit zou geen grote problemen moeten opleveren.

Gebouwselectie

Ook het selecteren van de plattegrond is tijdens het implementeren anders gegaan dan in het proof of concept en het ontwerp is bedacht. In eerste instantie werd gebruik gemaakt van een bestaande applicatie om een bestand te selecteren. Dit is gekozen als een eenvoudige oplossing die snel te configureren was. Echter bleek het niet mogelijk om deze applicatie zo te configureren dat slechts één bestand tegelijk geselecteerd kon worden. Als een gebruiker dus meerdere bestanden tegelijk selecteerde, was het niet mogelijk te achterhalen welk bestand echt bedoeld werd. Dit is de reden geweest om een eigen versie te schrijven zodat het nu altijd alleen maar mogelijk is om één bestand te selecteren. Dit scheelt in de verwarring voor de gebruiker als deze per ongeluk twee bestanden zou aanvinken. Dit is nu niet meer mogelijk en dat betekent dus meer comfort voor de gebruiker.



Figuur 8.4: Selecteren van een gebouw

Multitasken

Naast wat algemene uitdagingen is er ook een specifiek Android-onderdeel dat tijdens het implementeren anders bleek te werken dan van tevoren bedacht. Dit heeft te maken met het updaten van de GUI. Om te voorkomen dat de GUI zou vastlopen tijdens zware berekeningen, is er gebruik gemaakt van een achtergrondtaak. Deze achtergrondtaak zorgt ervoor dat een gebruiker altijd in staat is om het menu te bedienen ook al is de rest van de applicatie druk bezig. Deze achtergrondtaak werkt op een telefoon met Android wat anders dan bij een pc-applicatie. Hier is gedurende het ontwerpen geen rekening mee gehouden omdat het gebruik van pseudocode niet te specifiek moet worden. Dit was tijdens het implementeren echter snel opgelost en heeft geen echte tijdsvertraging opgeleverd in de planning.

Uitvoer van taken

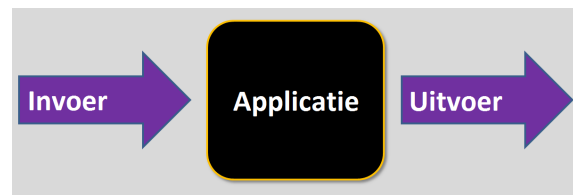
Tot slot is de manier waarop Android verschillende taken uitvoert anders dan gebruikelijk bij andere applicaties. Hier is bij het maken van de diverse proof of concepts rekening mee gehouden maar dit is concreet in het ontwerp niet terug te zien. Dit is gedaan om het ontwerp universeel te houden zodat het makkelijker ook voor andere platformen als Windows Phone of iOS (iPhone) te maken is. Bij het implementeren is dit wel op de Android-manier gedaan omdat anders geen werkende applicatie mogelijk is. De keuze is dus bewust gemaakt om het ontwerp niet al te Android-specifiek te maken om het zo een open ontwerp te houden waarmee andere programmeurs ook uit de voeten kunnen.

Whitebox-testing

Na het implementeren van de diverse onderdelen zijn deze ook per onderdeel getest. Hiervoor is gebruik gemaakt van Whitebox-testing. Bij deze manier van testen worden losse stukken code getest op functionaliteit. Van tevoren moet bekend zijn wat een stuk code doet en wat bij bepaalde invoer de uitvoer zou moeten zijn. Na het schrijven van een stuk code is dit steeds op deze manier getest om te zorgen dat de losse onderdelen werken zoals bedoeld is.

Blackbox-testing

Na het compleet implementeren van alle losse onderdelen en deze getest te hebben op werking was het tijd voor een globale test van het gehele systeem. Voor dit testen is gebruik gemaakt van Blackbox-testing. Bij deze manier van testen wordt alleen gekeken of een totaalsysteem reageert zoals te verwachten is. Losse onderdelen worden



Figuur 8.5: Blackbox-testing

hier niet getest zolang het totaalsysteem maar doet wat er van verwacht wordt. Wanneer alleen gebruik wordt gemaakt van deze manier van testen zou dit zeer zwak zijn. Als echter de combinatie wordt gemaakt met Whitebox-testing dan wordt deze methode een stuk sterker. Het totaalsysteem blijkt na testen niet te crashen en globaal werken alle functies. Echter blijkt wel dat de nauwkeurigheid zoals die bij de onderzoeken gedacht is te halen in de praktijk soms tegen te vallen. Hiervoor zal in een later stadium nog een uitgebreide test worden gedaan in een omgeving met voldoende WiFi-signalen op verschillende verdiepingen maar de resultaten hiervan zijn bij dit schrijven niet beschikbaar en zullen dus niet in dit verslag worden opgenomen. Qua software kan geconcludeerd worden dat deze werkt zoals van tevoren bepaald. Alleen de nauwkeurigheid laat in sommige gevallen nog wat te wensen over en deze zou zeker nog verbeterd kunnen worden als uitvoerig is gekeken naar de oorzaak hiervan.

Processorgebruik

Iets wat opviel tijdens het uitvoeren van de diverse testen was het processorgebruik. Het onderdeel dat verantwoordelijk is voor het berekenen van de route is behoorlijk intensief. Deze vraagt de nodige processortijd. De telefoons waarop getest is hebben hier nooit moeite mee gehad en hebben nooit de processor maximaal gebruikt. Als echter gebruik wordt gemaakt van oudere telefoons, of als er grote gebouwen worden ingeladen, dan kan dit een kritieke factor worden. Wanneer een groot gebouw wordt ingeladen zal de routeberekening vele malen intensiever worden. Dit kan betekenen dat oudere telefoons de applicatie niet meer vloeiend kunnen gebruiken. De routeberekening kan zeker nog geoptimaliseerd worden, maar daar heeft niet de focus op gelegen, dus dat is iets voor in een later stadium.

Monkey-testing

De laatste test die nog wel is uitgevoerd en hier besproken kan worden is de Monkey-test. Deze test is ingebouwd in de Android-omgeving en kan na een kleine configuratie automatisch uitgevoerd worden. Deze test gaat willekeurig vele diverse acties uitvoeren om te kijken of een applicatie crashbestendig is. Hierbij worden in verschillende willekeurige volgorde alle mogelijke toetsencombinaties op een telefoon uitgeprobeerd. Ook wordt het scherm op vele verschillende punten geactiveerd alsof een gebruiker blind op het scherm aan het drukken is. Ook het volume van de telefoon wordt aangepast om te kijken of dat geen effect heeft op de applicatie. Samengevat doet deze test precies zoals de naam aangeeft; zonder gezond verstand alle onlogische combinaties uitproberen om de applicatie te laten crashen. Deze test is meerdere malen uitgevoerd en na de eerste paar keer wat kleine foutjes te hebben verbeterd, heeft de test uiteindelijk geen crashes meer opgeleverd. Deze test bewijst dus dat een gebruiker nooit dergelijke combinaties kan maken die de applicatie doen crashen en dit is voor ontwikkelaars een belangrijk doel.



Figuur 8.6: Monkey-testing

Hoofdstuk 9

Conclusie en aanbevelingen

In dit project is er gebruik gemaakt van het V-model, dit was een fijne manier van projectmatig werken. Problemen konden tijdig worden opgemerkt omdat er veel grip was op het project. Het realiseren van het einddoel was op deze manier zeker haalbaar. Dit was nodig omdat tijdens het afstuderen de nadruk ligt op het realiseren van een product. Om dit te realiseren moet goed in de gaten worden gehouden of er niet doorgedraafd wordt op een bepaald aspect van het project.

Als ontwikkelingsplatform is er door de klant gekozen voor Android. Het Android-platform was een goede keuze van de klant omdat hierover veel gratis documentatie te vinden is en er behoefde geen extra hardware aangeschaft te worden. Omdat de ontwikkelingsomgeving gewoon op de standaard Windows pc draait en de ontwikkelaars zelf over Android telefoons beschikte.

Er is gekozen voor de fingerprinting-techniek omdat dit de nauwkeurigste en goedkoopste techniek was. Dit komt omdat een Android apparaat vanaf versie 2.1 beschikt over een WiFi ontvanger en de benodigde infrastructuur op een goedkope manier uitgebreid kan worden. Waardoor het product binnen TASS zelf getest kon worden en het dus niet nodig was om het product constant extern te testen.

Voor het gebruik van pseudocode in combinatie met UML is gekozen omdat met deze methode makkelijk en snel nagegaan kan worden of het ontwerp ook daadwerkelijk klopt. Hiermee wordt voorkomen dat deze fouten pas tijdens de implementatie- of testfasen naar voren komen.

Het product is gerealiseerd en getest in een gecontroleerde omgeving. In deze gecontroleerde omgeving was de dekking erg hoog, namelijk één router om de tien meter. Hiermee was een nauwkeurigheid van twee tot drie meter haalbaar en kon er redelijk stabiel genavigeerd worden. Helaas ging de nauwkeurigheid enorm omlaag wanneer de dekking gehalveerd wordt. Theoretisch leek er weinig verschil te zijn maar doordat de WiFi-signalen dermate veel last hebben van storingen was er maar een nauwkeurigheid van twee tot zeven meter haalbaar. Hierdoor is het product niet geschikt voor gebouwen waar momenteel een lage dekking is omdat hier de infrastructuur dermate vergroot moet worden dat de toepassing te duur wordt. Voor ziekenhuizen zoals het Deventer Ziekenhuis en het Maxima Medisch Centrum is het wel toepasbaar omdat hier de dekking momenteel al voldoet aan één router per tien meter.

Als er verder gewerkt wordt aan dit project kan er aandacht besteed worden aan het aantal metingen per seconde waarmee het signaal gestabiliseerd kan worden. Voor het verspreiden van de metingen en de mogelijkheid om te navigeren naar personen zal de logica geheel naar een server verplaatst kunnen worden.

Wanneer dit product toegepast zal worden in een kantoorgebouw met een lage dekking is het verstandig om naar een andere techniek te zoeken. Deze techniek is niet geschikt voor een gebouw met een lage dekking wat hoge uitbreidingskosten tot gevolg heeft.

Evaluaties

Evaluatie Jochem

De afstudeerperiode bij TASS Technology Solutions heb ik als zeer prettig ervaren. Ik heb de kans gekregen om twintig weken te werken aan een opdracht die ik samen met mijn afstudeerpartner Christian Litjes heb mogen uitvoeren. Tijdens een intakegesprek een aantal maanden voor de start van het afstuderen heb ik samen met Christian gekozen voor de opdracht zoals deze in dit document besproken is. Zowel Christian als ikzelf waren op zoek naar een opdracht met Android en deze opdracht heeft daar goed invulling aan gegeven.

Het maken van een applicatie voor Android met als doel navigatie voor binnen is naast een uitdaging in software ook een technische uitdaging geweest. Het werken met WiFi-signaalsterktes is iets wat ik nog nooit gedaan had en dit zorgde meteen voor een verbreding van mijn kennis op dit gebied.

Aangezien het niet direct duidelijk was met welke techniek we een positie zouden gaan bepalen, ben ik daar eerst onderzoek naar gaan doen. Toen eenmaal WiFi als techniek was gekozen, zijn de requirements en het ontwerp gemaakt. Na deze fase is begonnen met implementeren en dit heeft uiteindelijk een werkende applicatie opgeleverd. De nauwkeurigheid en de toepasbaarheid onder normale omstandigheden zijn helaas niet wat we zouden willen, maar we hebben aan kunnen tonen dat navigatie via WiFi-signaalsterktes mogelijk is, mits er een heel hoge dekkingsgraad van WiFi-routers is.

Het samenwerken tijdens de afstudeeropdracht met Christian heb ik als zeer positief ervaren. Ondanks dat we beiden een behoorlijk verschillend karakter hebben, is het gelukt optimaal gebruik te maken van de kennis die we beiden hebben. Naast de extra kennis die aanwezig is bij een tweede persoon is ook het samenwerken op zich een goede extra toevoeging geweest. Met zijn tweeën houd je elkaar scherp en wijs je elkaar op fouten die je alleen pas later zou ontdekken. Ook het maken van afspraken om individueel te kunnen werken hebben bijgedragen aan mijn ervaring met samenwerken. Als ik toekomstige afstudeerders zou mogen adviseren dan zou ik zeker aanraden om dit met twee personen te doen, omdat je een mooier product kunt maken en je naast technische kennis ook een hoop persoonlijke vaardigheden extra aanleert.

De begeleiding die vanuit TASS aan afstudeerders en stagiaires wordt gegeven is zeer goed. Wij hebben een proces- en technisch begeleider toegewezen gekregen, bij wie we terecht konden met eventuele problemen. Naast deze begeleiders zijn ook de andere collega's die bij TASS werken zeker niet te beroerd om ergens mee te helpen als je dit vraagt. Naast de vaste medewerkers van TASS heb je als voordeel dat er altijd veel stagiaires en afstudeerders aanwezig zijn. Als studenten onder elkaar was iedereen bereid te helpen of mee te denken wanneer een probleem zich voordeed.

Tenslotte, maar zeker niet onbelangrijk, is er nog de sfeer. Deze is echt uitstekend te noemen. Mede door het feit dat je met veel studenten bij elkaar zit, voel je jezelf vanaf dag één thuis. Bij TASS worden studenten als volwaardige medewerkers behandeld. Zo ben ik bij alle activiteiten, georganiseerd door TASS, uitgenodigd (en zo vaak mogelijk aanwezig geweest) en stond er met kerst ook een kerstpakket voor me klaar.

De bovenstaande zaken hebben ervoor gezorgd dat ik tevreden terugkijk op mijn afstudeerperiode. Ik heb aan een leuke, uitdagende opdracht kunnen werken en heb op professioneel niveau software ontwikkeld. Ik heb de begeleiding gekregen die nodig was; niet te veel maar zeker niet te weinig. Naast de zakelijke aspecten heb ik gewoon een leuke tijd gehad en ben ik blij dat ik mijn afstudeerperiode bij TASS heb mogen doorbrengen.

Evaluatie Christian

De afstudeeropdracht bij TASS Technology Solutions is mij goed bevallen. Hier heb ik gedurende twintig weken met veel plezier aan mijn afstudeeropdracht gewerkt. Deze opdracht heb ik samen uitgevoerd met Jochem Kempkens. Deze samenwerking is goed verlopen wat opdracht ten goede kwam.

De afstudeerplek bij TASS had ik al een jaar van tevoren geregeld. Ik ben erg blij dat ik dit op tijd geregeld had omdat ik me hierover niet meer druk hoefde te maken. Ik wist al vroeg dat ik bij TASS wilde gaan afstuderen en afstuderen in duo's is hier een pre. Hierdoor kon ik al erg snel opzoek naar iemand om mee af te studeren. Ik ben vervolgens bij Jochem uitgekomen omdat wij samen hetzelfde beeld van een opdracht hadden. Ook eerdere samenwerkingsverbanden hebben mij een goed beeld van Jochem gegeven. Vervolgens hebben we bij een intakegesprek de opdracht bepaald: Indoor Positioning op Android. Doordat de opdracht al vroeg bekend was ben ik mezelf al wat gaan verdiepen in Android. Hierdoor had ik een kleine voorsprong op Jochem maar dit was geen probleem en erg handig zelfs. Ik kon Jochem bij problemen met Android meteen helpen.

Aan het begin van het project heb ik samen met Jochem het Project Management Plan opgesteld. Dit ging erg gemakkelijk omdat we dit al vaker hadden gedaan in projecten op school. Op school werd dit document het Project Initiation Document genoemd. Het maken van de planning was niet zo makkelijk, omdat dit de eerste keer was dat we voor twintig weken een gedetailleerde planning moesten maken. Doordat we veel tijd in het plannen hadden gestoken resulteerde dit in een goede planning waardoor we niet achterop schema kwamen te lopen.

Het opstellen van de specificaties ging wat moeizamer. Dit kwam voornamelijk omdat de klant een SRS wilde zien in de plaats van een FRS die we gemaakt hadden. We hebben vervolgens samen met de klant een SRS opgesteld. Het opstellen hiervan ging snel omdat in essentie hier hetzelfde in staat als het FRS. Na het SRS en FRS zijn we begonnen aan het ontwerp. Deze fase verliep goed, Jochem en ik hadden van tijd tot tijd wel discussies over design keuzes, maar hierdoor is het ontwerp ook goed doordacht. Om snel na te gaan of het ontwerp klopte zijn we met pseudocode aan de slag gegaan. Dit was compleet nieuw voor mij en leverde in het begin wat problemen op. Toen ik dit eenmaal onder de knie had was het een fijne en simpele manier om snel na te gaan of het ontwerp echt ging werken. Dit had als voordeel dat we goed wisten of het project haalbaar was. Omdat het ontwerp zeker ging werken was de implementatiefase erg kort wat ik erg prettig vond.

De begeleiding en sfeer bij TASS zijn erg goed. Er waren 2 begeleiders binnen TASS aangesteld om ons te begeleiden op proces- en technisch vlak. Uiteindelijk heeft de procesbegeleider wat meer de technische begeleiding op zich genomen. Dit vond ik niet erg omdat hij ook technisch goed onderlegd is.

De sfeer binnen TASS is heel goed en je voelt je al snel thuis. Je zit samen in een ruimte met vijftien studenten en de rest van het personeel is informeel te benaderen. Hierdoor voel je geen druk als je informatie nodig hebt van een collega of gewoon een praatje wil maken.

Al dit tezamen heeft ervoor gezorgd dat het een leuke afstudeerplek en afstudeeropdracht was. De samenwerking met Jochem verliep goed, waardoor we samen een goed eindresultaat hebben opgeleverd. Namelijk de Indoor Navigation applicatie voor Android. Ik kan een duostage ten zeerste aanraden, je houdt elkaar scherp en kunt heel technisch op onderdelen in gaan. Hierdoor vind je snel de beste oplossing maar moet je natuurlijk wel open staan voor elkaars ideeën.

Als ik een ding moet noemen wat ik het leukste vond was het wel het onderzoeken van de nieuwe technieken in combinatie met het bouwen van kleine prototypes. Dit gaf extra stimulans om verder te werken en wetende dat het eindresultaat ook haalbaar was gaf een goed gevoel. Deze afstudeeropdracht is iets waar ik zelf erg trots op ben en zal hier later met veel plezier aan terug denken.

Literatuurlijst

Boeken

Zo doe je een onderzoek

Publicatiedatum: 2009

Auteur(s): Roel Grit & Mark Julsing

Beginning Android 3

Publicatiedatum: 2011

Auteur(s): Mark Murphy

Pro Android 3

Publicatiedatum: 2011

Auteur(s): Satya Komatineni & Dave MacLean & Sayed Hashimi

Internet

Best Practices Location-Aware WLAN Design Considerations

<http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/wifich5.html#wp1051879>

Onderwerp: Positiebepaling door middel van WiFi signaalsterktes

Global Positioning System

http://www.dmoz.org/Science/Earth_Sciences/Geomatics/Global_Positioning_System/

Onderwerp: Global Positioning System

ZigBee Alliance

<http://www.zigbee.org/About/AboutAlliance/TheAlliance.aspx>

Onderwerp: ZigBee

The Official Bluetooth SIG Member Website

https://www.bluetooth.org/About/bluetooth_sig.htm

Onderwerp: Bluetooth

The Computer Vision

<http://www.thecomputervision.com/>

Onderwerp: Computer Vision

WiFi Alliance

<http://www.wi-fi.org/>

Onderwerp: WiFi

RADAR

<http://research.microsoft.com/en-us/projects/radar/>

Onderwerp: Het eerste systeem voor positiebepaling met WiFi

Radio Frequency IDentification

<http://www.rfidkenniscentrum.nl/>

Onderwerp: Alle mogelijke informatie met betrekking tot RFID

XBee Explorer Regulated

<http://www.sparkfun.com/products/9132>

Onderwerp: Prijsinformatie

Evaluating Wi-Fi Location Estimation Technique for Indoor Navigation

<http://www.sparkfun.com/products/9219>

Onderwerp: Prijsinformatie

Cell Phone Triangulation Accuracy Is All Over The Map

<http://searchengineland.com/cell-phone-triangulation-accuracy-is-all-over-the-map-14790>

Onderwerp: Positiebepaling door middel van GSM signaalsterktes

Scripties en rapporten

Indoor Navigation System for Handheld Devices

Publicatiedatum: 22 oktober 2009

Auteur(s): Manh Hung V. Le & Dimitris Saragas & Nathan Webb

Indoor Positioning with a WLAN Access Point List on a Mobile Device

Publicatiedatum: 31 oktober 2006

Auteur(s): Marion Hermersdorf

NAVSTAR GPS User Equipment Introduction

Publicatiedatum: september 1996

Auteur(s): United States Coast Guard

Positiebepaling met WiFi-signalen

Publicatiedatum: 13 december 2006

Auteur(s): Cedric Cuypers & Chris Adriaensen & Daniel Balog & Guy Van Den Broeck & Jef Maerien & Philippe Dellaert & Wouter Van Rans

Survey of Wireless Indoor Positioning Techniques and Systems

Publicatiedatum: 6 november 2007

Auteur(s): Hui Liu & Houshang Darabi & Pat Banerjee & Jing Liu

The Horus WLAN Location Determination System

Publicatiedatum: 14 april 2006

Auteur(s): Moustafa Youssef & Ashok Agrawala

WiFi fingerprint Red Team Final Report

Publicatiedatum: 26 mei 2011

Auteur(s): Hisham El Shaer & Chengzhou Jin & Xiaoqian Zhu & Yunlong Huang & Qiuchan Luo & Dinsefa Mustefa

Presentaties

Evaluating Wi-Fi Location Estimation Technique for Indoor Navigation

Publicatiedatum: 2008

Auteur(s): Roshmi Bhaumik

Positiebepaling met WiFi-signalen

Publicatiedatum: 22 mei 2007

Auteur(s): Cedric Cuypers & Chris Adriaensen & Daniel Balog & Guy Van Den Broeck & Jef Maerien & Philippe Dellaert & Wouter Van Rans

Bijlagen

A - Project Management Plan

Het volledige Project Management Plan zoals dit in de beginfase van de afstudeeropdracht is opgesteld volgens de richtlijnen van TASS.

B - Onderzoeksdocument Positioning

Het volledige onderzoeksdocument over Positioning, gemaakt door Jochem Kempkens, waarin wordt uitgelegd hoe de keuze is gemaakt om de positie te gaan bepalen met behulp van WiFi-signaalsterktes.

C - Onderzoeksdocument Fingerprinting

Het volledige onderzoeksdocument over Fingerprinting, gemaakt door Christian Litjes, waarin wordt uitgelegd hoe de keuze is gemaakt om de positie te gaan bepalen door middel van de Fingerprinting-techniek.

D - UML-diagram

Het totaaloverzicht van de software op een hoog niveau, gemaakt volgens de UML-standaard.

Bijlage A

Project Management Plan

Bijlage B

Onderzoeksdocument Positioning

Bijlage C

Onderzoeksdocument Fingerprinting

Bijlage D

UML-diagram