SCRIPTIE: CONTEXT-AWARE AUGMENTED NAVIGATION FOR MOBILE

Auteur	: Wouter van Ewijk
Datum	: 25-05-2011
Versie	: 1.0
Afstudeerbedrijf	: Logica Nederland BV
Divisie	: Working Tomorrow
Practice	: Consulting team IDT
Bedrijfsbegeleiders	: Ingrid van Zaanen & Gideon Bazen
Bedrijfsmentor	: Freek Uijtdewilligen
Onderwijsinstelling	: Hogeschool Utrecht
Opleiding	: Technische Informatica
Afstudeercoördinator	: Marten Wensink
Examinator	: Peter van Rooijen

Locatie: **Amstelveen** **Lichting 12** Datum**: 25 mei 2011**

Versiebeheer

Versie	Omschrijving
v 0.1	Conceptversie, opgeleverd ter controle door bedrijfsbegeleiders
v 0.2	Feedback ontvangen indeling aangepast.
v 0.3	Scriptie uitgebreid.
v 0.4	Opmerkingen Ingrid van Zaanen verwerkt. Aantal hoofdstukken geschrapt, nieuwe indeling scriptie.
v 0.5	Feedback van Gideon Bazen, Peter van Rooijen, Freek Uijtdewilligen en Ingrid van Zaanen verwerkt.

Tabel 1 – versiebeheer

Distributie

Naam	Instantie	Rol / Functie
Gideon Bazen	Logica	Architect
Peter van Rooijen	Hogeschool Utrecht	Docent/ Examinator
Freek Uijtdewilligen	Logica	Mentor
Ingrid van Zaanen	Logica	Project Leader

Tabel 2 – Distributie

Adresgegevens

Examinandus

Naam	: Wouter van Ewijk
Straat / Nummer	: Neuweg 271
Postcode / Plaats	: 1214 GS, Hilversum
Telefoonnummer	: 035-6239382
E-mail	: WoutervE@orange.nl
Onderwijsinstelling	: Hogeschool Utrecht
Afstudeerrichting	: Technische Informatica
Studentnummer	: 1502420

Examinator

Naam	: Peter van Rooijen
E-mailadres	: peter.vanrooijen@hu.nl
Telefoonnummer	: 06 331 75 085

Afstudeercoördinator

Naam	: Marten Wensink
E-mail	: marten.wensink@hu.nl
Telefoonnummer	: +31 (0) 30 238 8591

Bedrijfsgegevens

Naam	: Logica Nederland BV
Afdeling	: Working Tomorrow
Straat / Nummer	: Prof. W.H. Keesomlaan 14
Postcode / Plaats	: 1183 DJ Amstelveen

Bedrijfsbegeleiders

Naam	: Gideon Bazen
Telefoonnummer	: +31 6 134 13 683
E-mail	: gideon.bazen@logica.com
Functie	: Architect

Naam	: Ingrid van Zaanen
Telefoonnummer	: +31 654 28 56 10
E-mail	: ingrid.van.zaanen@logica.com
Functie	: Project Leader

Bedrijfsmentor

Naam	: Freek Uijtdewilligen
Telefoonnummer (M)	: +31 613 40 53 49
E-mail	: freek.uijtdewiligen@logica.com
Functie	: Business Consultant

Practice Manager

Naam	: Marcel Mourits
Telefoonnummer	: +31 20 571 3602
E-mail	: marcel.mourits@logica.com
Functie	: Principal Business Consultant

Voorwoord

Voor u ligt de scriptie over de afstudeeropdracht “Context-aware augmented navigation”. Ik, Wouter van Ewijk, ben een student die Technische Informatica studeert aan Hogeschool Utrecht, te Utrecht. Voor mijn afstuderen heb ik de periode van 1 februari 2011 tot 1 juli 2011 stage gelopen bij het Working Tomorrow programma van Logica te Amstelveen.

Een aantal mensen hebben de realisatie van deze afstudeeropdracht mede mogelijk gemaakt. Ik bedank daarom de volgende mensen: Gerard van Elsen (docent HU FEM en oud-medewerker Logica) en Sander van der Wal (Software Engineer, gestudeerd aan HU) voor het mij attenderen op het Working Tomorrow programma van Logica. Zonder deze personen had ik waarschijnlijk nooit van het bestaan van het Working Tomorrow bij Logica gehoord. Ik bedank Logica voor het bieden van een afstudeerplaats met de bijbehorende faciliteiten.

Verder bedank ik de projectleider van Working Tomorrow, Ingrid van Zaanen, voor alle hulp en ondersteuning die zij mij geboden heeft. Niet alleen als er mensen of materiaal geregeld moest worden, maar ook met andere problemen kon ik altijd bij haar terecht.

Freek Uijtdewilligen bedank ik voor zijn inzet, advies en hulp bij technische problemen. Ook de twee architecten van het Working Tormorrow programma, Gideon Bazen en Tom Oorthuis bedank ik voor hun advies en begeleiding tijdens mijn stageperiode.

Ik bedank Peter van Rooijen voor zijn inzet, adviezen en goede begeleiding.

Tot slot ook dank aan alle collega’s van Working Tomorrow voor de gezellige periode tijdens de afstudeerstage.

Amstelveen, Mei 2011

Wouter van Ewijk

Managementsamenvatting

Om te kunnen afstuderen voor de opleiding Technische Informatica moet er stage gelopen worden bij een bedrijf met een HBO ICT waardige afstudeeropdracht. Logica is zo'n bedrijf en biedt jaarlijks voor zo'n ~200 afstudeerders een stageplaats aan. Vanwege dit grote aantal hebben zij speciaal het programma Working Tomorrow in leven geroepen. Dit werkt aan twee kanten: studenten zijn geholpen met een goede stageplaats en Logica heeft onder andere een goede keuze uit toekomstige werknemers.

Binnen Logica wordt al enige tijd gewerkt aan een augmented reality framework. Dit heeft al tot een paar aansprekende demo's geleid. Logica wil dit concept nu uitbreiden met context-afhankelijkheid. De navigatie moet zich aanpassen aan de situatie.
Er is dus een applicatie ontwikkeld met contextafhankelijke navigatie. Bijvoorbeeld, als een gebruiker langs een bepaalde winkel loopt, kan hij/zij met behulp van "context-aware augmented navigation" een overzicht van kortingen en/of aanbiedingen (die op dat moment gelden) ontvangen van de betreffende winkel. Winkels kunnen d.m.v. een backend applicatie zelf kortingen in het systeem plaatsen of een bestaande aanpassen. Om de opdracht goed te kunnen voltooien, dient er eerst een onderzoek gedaan te worden naar verschillende technieken en methoden, waarvan het resultaat als basis zal dienen voor het ontwerp van de applicatie.

De hoofdvraag die bij het onderzoek naar voren komt is:

"Op welke wijze kan een Android applicatie gerealiseerd worden die gebruik maakt van contextafhankelijke navigatie om kortingen van omliggende winkels te tonen?"

In de hier voor u liggende scriptie kunt u lezen hoe het onderzoek van de hoofdvraag heeft geleid tot een werkend prototype die aan de vereiste specificatie voldoet.

Het eindresultaat is een applicatie die gebruikers in staat stelt bepaalde informatie over winkels te ontvangen. Deze applicatie kan in de toekomst nog verder ontwikkeld worden met extra functionaliteiten. Dit staat tevens in deze scriptie beschreven.

Inhoudsopgave

Voorwoord **4**
Managementsamenvatting **5**
Inleiding **8**
1 Omgeving en organisatie **9**
1.1 Logica 9
1.2 Working Tomorrow 9
1.3 Begeleiding 11
2 Definitie afstudeeropdracht **12**
2.1 Inleiding 12
2.2 Probleemstelling 12
3 Aanpak **13**
3.1.1 Uitvoering (DSDM-fases) 14
3.1.2 Opstartfase 14
3.1.3 Onderzoeksfase 14
3.1.4 Ontwerpfase 14
3.1.5 Realisatiefase 14
3.1.6 Afrondingsfase 15
3.2 Projectgrenzen & randvoorwaarden 15
3.2.1 Projectgrenzen 15
4 Android **16**
4.1 Applicatie basisprincipes 16
4.1.1 Het Android Manifest bestand 16
4.2 Applicatie componenten 17
4.2.1 Activities 17
4.2.2 Services 18
4.2.3 Content providers 20
4.2.4 Broadcast receivers 20
4.3 User Interface 20
4.3.1 View Hierarchy 20
4.3.2 Layout 21
4.3.3 Menu 21
5 Context-awareness **22**
5.1 Het begrip context-awareness 22
5.2 Contextinformatie 22
6 Plaatsbepaling **24**
6.1 Methoden 24
6.2 Digitaal kompas 25
7 Het ontwerp **26**
7.1 Het gebruik van de mobiele applicatie 26
7.2 Use case diagram 30

7.2.1 Use case beschrijvingen 30
7.3 Het data flow diagram 34
7.4 Communicatie 35
7.5 Klassendiagram 36
7.6 Beschrijving klassendiagram 36
7.7 Het database ontwerp 38
7.8 Toelichting database 38
7.8.1 Tbl_Users 38
7.8.2 Tbl_Preferences 38
7.8.3 Tbl_Offer_message 39
7.8.4 Tbl_Employee 39
7.8.5 Tbl_Store 39
7.9 Google Maps implementatie 40
8 Het resultaat 42
9 Evaluatie 43
9.1 Persoonlijke evaluatie 43
9.2 Project evaluatie 44
9.3 Oplossingen & motivatie 46
10 Conclusies & aanbevelingen 48
10.1 conclusie 48
10.2 aanbevelingen 48
Bronnen 50
Bijlagen 52
Verklarende woordenlijst 52

Inleiding

Deze scriptie informeert u over mijn afstudeeropdracht “context-aware augmented navigation”. Het bedrijf waar ik stage loop, Logica, werkt al enige tijd aan een augmented reality framework. Logica wil dit concept uitbreiden met context-afhankelijkheid; de navigatie moet zich aanpassen aan de situatie.
In deze scriptie vindt u een beschrijving van de door mij ontwikkelde contextafhankelijke navigatie die een gebruiker in staat stelt in een winkelstraat een overzicht te ontvangen van de op dat moment geldende kortingen en/of aanbiedingen. Winkels kunnen d.m.v. een backend applicatie zelf kortingen in het systeem plaatsen en bestaande aanpassen. Ik heb onderzoek gedaan naar de verschillende technieken en methoden, waarvan het resultaat als basis dient voor het ontwerp van de applicatie. Ter demonstratie heb ik een proof of concept ontwikkeld.

1 Omgeving en organisatie

1.1 Logica

Logica is een belangrijke internationale speler op het gebied van IT en business services met 39.000 mensen in dienst in 36 landen. Ze bieden oplossingen en diensten aan in het domein van consultancy, design, systeem integratie, bedrijf kritische applicaties en de outsourcing van bedrijfsprocessen. Logica richt zich op vier marktsectoren – Energy Utilities & Telecom, Finance, Public Sector, en Industrie Distribution & Transport. Door deze focus zijn ze in staat om op maat gemaakte oplossingen te bieden voor de uitdagingen waar klanten in hun specifieke markt mee te maken krijgen. In Nederland is deze focus verankerd in vier Industrie Sectoren, afdelingen die zich richten op één van de segmenten.

Logica is gedreven om klanten te helpen om leidende posities te behalen en te behouden in hun individuele markten. De kracht van Logica ligt daarbij op het gebied van industrie- en domeinkennis, sterk bedrijfskundig- en technologisch inzicht en de vele successen met het leveren op tijd en binnen budget.

Het huidige Logica plc. is op 30 december 2002 ontstaan uit het voormalige Logica plc. (60%) en het voormalige CMG plc. (40%). CMG was in Nederland aanzienlijk groter dan Logica, die vooral een focus had in Groot Brittannië. In 2006 zijn daar nog WM-Data (zwaartepunt in Scandinavië) en Unilog (zwaartepunt in Frankrijk) aan toegevoegd, zodat een gespreide dekking in Europa verkregen is. In 2008 is onder leiding van de nieuwe CEO, Andy Green, een programma opgestart om van alle onderdelen binnen het bedrijf een uniforme organisatie neer te zetten - One Logica. Hierbij zijn de waarden gedefinieerd, volgens welke iedere Logicaan zou moeten werken: Committed, Innovative en Open. De missie die hierbij gesteld is: “To be the most trusted innovation partner”.

1.2 Working Tomorrow

De opdracht wordt uitgevoerd binnen het programma Working Tomorrow. Logica heeft dit programma opgestart om studenten de gelegenheid te geven af te studeren op een innovatieve opdracht met goede begeleiding. Innovatie staat centraal. Innovatief wat betreft technologie, concept of methodiek. Zo werken er studenten aan agent technologie, Multi touch technologieën, augmented reality, Smart Grids en slimme meters en mobiele oplossingen. Working Tomorrow is nu op 5 vestigingen van Logica in Nederland aanwezig en biedt jaarlijks plaats aan circa 120 studenten van HBO en WO. Op iedere vestiging is een projectleider van Working Tomorrow die de studenten begeleidt.

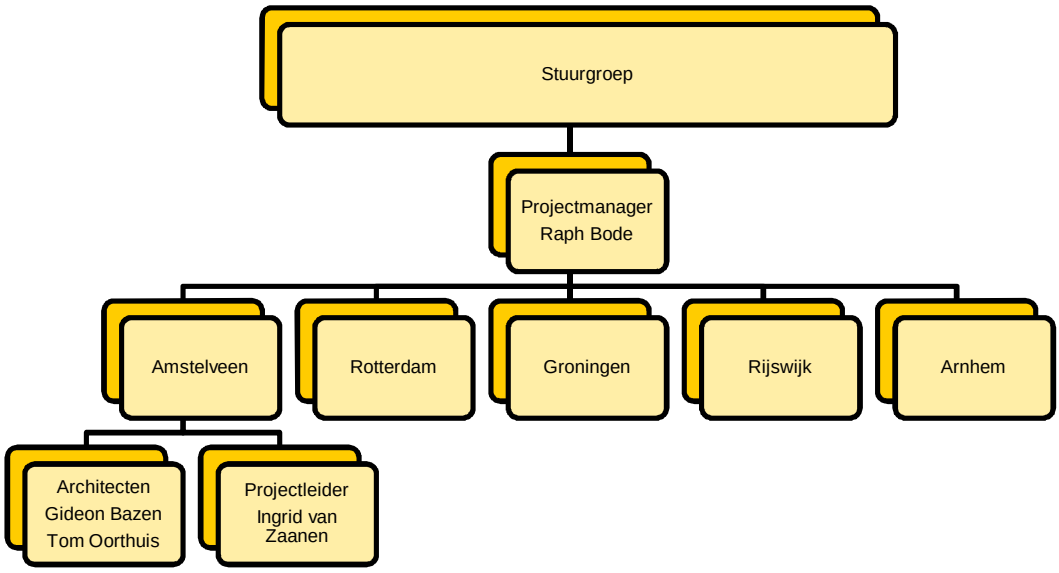
Het Working Tomorrow programma heeft vier hoofddoelen:

1. Een centrale plaats bieden waar studenten uitdagende en innovatieve afstudeerprojecten kunnen uitvoeren.
2. Het werven van toekomstige werknemers.
3. Verhogen van de reputatie van Logica op het gebied van innovatie.
4. Demo’s en resultaten van projecten gebruiken voor het verkrijgen van betaalde opdrachten en vergaren van kennis.

Logica Nederland heeft naast de Industrie Sectoren ook Service Lines (SL). Consulting en Professional Services (C&PS) is er op gericht om de IS zo goed mogelijk te voorzien van goed en gekwalificeerd personeel. C&PS is verder onder te verdelen in Consulting, SI Projects en Professional Skills. Professional Skills is weer verder onderverdeeld in practices, soort van afdelingen die gericht zijn op een bepaalde skillset, bijvoorbeeld; Project Management, Java. Oracle of ISA.

Figuur 1 - Organogram Logica

In het onderstaande figuur is het organogram van Working Tomorrow te zien.



Figuur 2 – Organogram Working Tomorrow

1.3 Begeleiding

Alle studenten die afstuderen bij Working Tomorrow, worden bij een practice geplaatst die past bij de richting van het project. Een student krijgt naast procesmatige begeleiding vanuit Working Tomorrow ook inhoudelijke begeleiding vanuit de practice. Dit project valt onder de practice Bussiness Consulting team IDT. De praticemanager van deze practice is de heer Marcel Mourits. De betrokken WT-begeleiders zijn:

Gideon Bazen; hij is aangesteld voor de algemene begeleiding. Met problemen en vragen kan ik ook bij hem terecht. Tot slot zal hij ook de ingeleverde documenten controleren en beoordelen.

Voor de inhoudelijke begeleiding is er een mentor binnen Logica aangesteld. Deze functie wordt vervuld door de heer Freek Uijtdewilligen die zeer bekwaam is met contextafhankelijke navigatie en augmented reality op mobiele applicaties.

De projectleidersrol wordt vervuld door Ingrid van Zaanen. Verder wordt er een wekelijks werkoverleg georganiseerd waarin twee afstudeerders ter oefening een presentatie houden over het werk dat gedaan is voor de afstudeeropdracht. Bovendien is er ook nog de mogelijkheid om algemene zaken te bespreken en/of problemen op te lossen. Ook is er een maandelijks voortgangsgesprek met de projectleider en/of de architect waarbij het highlight-rapport als basis dient voor het gesprek.

Vanuit de Hogeschool Utrecht is er een begeleidende docent aangesteld; Peter van Rooijen. Hij begeleidt de afstudeerder tijdens het afstudeertraject en houdt de voortgang van het project in de gaten. Tot slot kunnen documenten worden gecontroleerd door de docent om zo feedback aan de student te kunnen geven.

2 Definitie afstudeeropdracht

2.1 Inleiding

De afstudeeropdracht bij Working Tomorrow van Logica is een combinatie van een vraag- en maakopdracht. Het is de bedoeling dat er een applicatie ontwikkeld wordt waarin “context-aware augmented navigation” is verwerkt.

De afstudeeropdracht heeft meerdere doelen:

- de afstudeerder te testen op kennis en vaardigheden;
- een prototype opleveren die eventueel gebruikt kan worden als basis voor een beginnend product;
- persoonlijk doel; het behalen van een diploma en het werven van een baan bij Logica.

Voor de afstudeeropdracht zullen een aantal termen gebruikt worden om ondersteuning te bieden aan het definiëren van de afstudeeropdracht. Deze termen worden hieronder kort toegelicht:

- augmented reality, is een vakgebied dat zich hoofdzakelijk bezighoudt met het zo realistisch mogelijk toevoegen van computergemaakte beelden aan rechtstreekse, reële beelden;
- contextafhankelijke navigatie (context-aware navigation zie ook hoofdstuk 5), navigatie die zich aanpast aan de situatie;
- Android, is een open source besturingssysteem voor mobiele apparaten;
- Android applicatie, het product dat een onderdeel van het eindresultaat vormt;
- eindgebruiker, persoon die het product ook daadwerkelijk gaat gebruiken.

2.2 Probleemstelling

Binnen Logica wordt al enige tijd gewerkt aan een augmented reality framework. Dit heeft al tot een paar aansprekende demo’s geleid. Logica wil dit concept nu uitbreiden met context-afhankelijkheid. De navigatie moet zich aanpassen aan de situatie.
Er is dus een applicatie ontwikkeld met contextafhankelijke navigatie. Bijvoorbeeld, als een gebruiker langs een bepaalde winkel loopt, kan hij/zij met behulp van “context-aware augmented navigation” een overzicht van kortingen en/of aanbiedingen (die op dat moment gelden) ontvangen van de betreffende winkel. Winkels kunnen d.m.v. een backend applicatie zelf kortingen in het systeem plaatsen of een bestaande aanpassen. Om de opdracht goed te kunnen voltooien, dient er eerst een onderzoek gedaan te worden naar verschillende technieken en methoden, waarvan het resultaat als basis zal dienen voor het ontwerp van de applicatie.

De hoofdvraag die bij het onderzoek naar voren komt is: *"Op welke wijze kan een Android applicatie gerealiseerd worden die gebruik maakt van contextafhankelijke navigatie om kortingen van omliggende winkels te tonen?"*

Om de hoofdvraag juist te kunnen beantwoorden zijn er een aantal deelvragen gedefinieerd. De deelvragen zijn als volgt:

- hoe kan contextafhankelijkheid en navigatie worden toegepast op de Android applicatie;
- hoe is een Android applicatie opgebouwd;
- hoe werkt het digitaal kompas;
- hoe wordt de positie bepaald en welke manier is hiervoor het meest geschikt;
- hoe maak je een GUI voor Android;
- hoe integreer je Google Maps.

3 Aanpak

De aanpak die tijdens de opdracht gevolgd is, is als volgt ingedeeld. Als eerste is de opdracht verder gespecificeerd m.b.v. de opdrachtgever. Vervolgens is er een managementtechniek gekozen. Om de afstudeeropdracht goed en gestructureerd te laten verlopen en tot een succes te maken, is er voor een aanpak gekozen die gebaseerd is op Dynamic Systems Development Method (DSDM), vanwege het feit dat DSDM voor deze afstudeeropdracht veel te uitgebreid is. De onderdelen van DSDM die wel toepasbaar zijn bij dit project, zijn Timeboxing en MoSCoW.

Timeboxing

Timeboxing heeft als doel te zorgen voor een tijdige oplevering en realisatie van een project. Een timebox binnen DSDM heeft dus een begin- en einddatum van een periode waarbij aan het eind een product wordt opgeleverd (deadline). Denk bijvoorbeeld aan een functioneel ontwerp. Samengevat zorgt timeboxing er dus voor:

- dat de focus blijft liggen op de oplevering van een product;
- dat de concentratie ligt op de hoogste prioriteiten;
- dat er een overzicht is van de verwachte tijd dat een bepaald onderdeel kost.

(Bron: Stapleton, J[1])

MoSCoW-methode

Binnen DSDM worden aan vereisten prioriteiten gesteld d.m.v. de MoSCoW-methode:

- ***M**ust have this*
  "Deze eis moet in het eindresultaat terugkomen, zonder deze eis is het product niet bruikbaar." ;
- ***S**hould have this if at all possible*
  "Deze eis is zeer gewenst maar zonder is het product wel bruikbaar.";
- ***C**ould have this if it does not effect anything else*
  "Deze eis mag alleen aan bod komen als er tijd genoeg over is.";
- ***W**ould like to have but would like to have this in the future*
  "Deze eis zal in dit project niet aan bod komen maar kan in de toekomst, bij een vervolg project, interessant zijn."

(Bron: Stapleton, J[1])

Hiermee beschikt het project dus over een tijdmanagement techniek en over een manier om prioriteiten te stellen aan softwarefunctionaliteit. Daarna is er een hoofdvraag (§2.2) opgesteld, die vervolgens is onderverdeeld in deelvragen (§2.2) zodat deze samen een antwoord kunnen geven op de hoofdvraag. Na het beantwoorden van de verschillende vragen is er genoeg kennis vergaard, om het prototype te kunnen bouwen. Zowel het bouwen en documenteren van de opdracht is een iteratief proces. Dit houdt in dat er telkens deelproducten zullen worden opgeleverd. Dit heeft enkele voordelen, namelijk, er is een duidelijk overzicht van de voortgang. Bovendien kan de opdrachtgever en/of begeleider na elke oplevering feedback geven. Wat mogelijk na overleg kan resulteren in een wijziging of aanpassing van het betreffende (deel)product.

Een increment van het iteratieve proces loopt als volgt:

1. Iteratie één, bekijken welke increment gebouwd/ontwikkeld wordt. Hierin worden de functionele eisen bepaald aan de hand van een MoSCoW-lijst. Zodra dit duidelijk is, wordt er een terugkoppeling gemaakt naar de opdrachtgever en/of begeleider.
2. Iteratie twee, de increment van iteratie één wordt verder uitgewerkt, waarna na voltooiing een terugkoppeling aan de opdrachtgever en/of begeleider volgt.
3. Iteratie drie, naar aanleiding van de terugkoppeling wordt de betreffende increment verfijnd.

Na het voltooien van het te maken prototype zal er een demonstrator gemaakt worden in de vorm van een korte film, dit werkt immers tijdsefficiënt tijdens de afstudeerzitting. Tot slot zal er eindpresentatie gemaakt worden.

3.1.1 Uitvoering (DSDM-fases)

De afstudeeropdracht is in een aantal fases opgedeeld, dit is gedaan om activiteiten onder te verdelen en af te bakenen in welke fase welke activiteit dient te worden voltooid. De fases zijn;

Figuur 3 - de verschillende fases van het afstudeerproject

3.1.2 Opstartfase

Deze fase dient ter voorbereiding van de ontwikkeling van de applicatie. In deze fase is dus het plan van aanpak gemaakt.

3.1.3 Onderzoeksfase

Na het afronden van de opstartfase start de onderzoeksfase; inmiddels ligt er een concreet plan van aanpak klaar ter ondersteuning van deze afstudeeropdracht. In de onderzoeksfase is er gezocht naar de antwoorden op de deelvragen die dus gezamenlijk antwoord geven op de hoofdvraag §2.2. De resultaten van het beantwoorden van deze vragen zijn verwerkt in de scriptie.

3.1.4 Ontwerpfase

Nadat de onderzoeksfase was afgerond, is er een begin gemaakt met het ontwerp van de applicatie. Aan het einde van deze fase is er een functioneel ontwerp en een technisch ontwerp opgeleverd.

Het eerste document is een functioneel ontwerp. De zaken die hierin aan bod komen zijn:

- de functionaliteiten die de applicatie zal bevatten;
- hoe zien deze functionaliteiten eruit (GUI) in de applicatie.

Het tweede document is het technisch ontwerp waarbij het functioneel ontwerp als basis dient. De volgende punten zijn behandeld in het technisch ontwerp:

- structuur van de applicatie;
- hoe reageert de applicatie op de gebruiker.

3.1.5 Realisatiefase

In de realisatiefase is er voldoende voorbereid. Het ontwerp is inmiddels af; het enige dat nog restte was het realiseren van de applicatie. Deze fase is iteratief uitgevoerd. Tot slot dienen onderdelen van het prototype qua prioriteit te worden ingedeeld. Hiervoor wordt de MoSCoW-methode gebruikt om te zien welke specificaties minimaal vereist zijn om een werkend prototype op te leveren. Zie ook onderstaande Tabel 3.

Prioriteit	Omschrijving
Must have this	Het weergeven van de locatie.
	Verkrijgen van de positie van de gebruiker en het weergeven hiervan.
	Een (kleine) server applicatie voor bewijs werking (wijzigen/verwijderen/invoeren van kortingsberichten en/of aanbiedingsberichten) server applicatie.
	Kortingsbericht/aanbiedingsbericht.
	Verzenden en ontvangen van locatie en bericht.
	POI (points of interest).
Should have this if it is possible	Looptijd weergeven, hoelang het duurt om op de bestemming te komen.
	Extra informatie tonen over een winkel, bijvoorbeeld openingstijden en telefoonnummer van een winkel.
	Route bepaling.
	Voorkeurskoppeling (gebruiker heeft interesse voor Media Markt, toon ook enkele andere soortgelijke winkels, BCC bijvoorbeeld).

	Instelbare straal van 2 km om positie van de gebruiker heen. (Geofencing)
	Icoon voor het aantonen dat GPS aanstaat.
	Instelbare frequentie van het doorgeven van de positie van de gebruiker.
Could have this if it does not effect anything else	Kompas met kijkrichting, dus naar welke kant is de gebruiker gericht.
	“3D” view via een camera van een mobiel.
	Voorkeur gekoppeld aan tijd/leeftijd/geslacht et cetera.
Would like to have but would like this in the future	Het laten oplichten (highlighten) van een POI terwijl het kompas er op gericht is.
	Toevoeging van augmented reality aan de applicatie.

Tabel 3 - MoSCoW-prioriteitenlijst

3.1.6 Afrondingsfase

In deze fase is het prototype getest door de afstudeerder waarna deze is geaccepteerd door de opdrachtgever of bedrijfsbegeleider. Verder wordt er een demonstrator van de applicatie gemaakt waarmee een demo kan worden getoond. Deze demonstrator zal gerealiseerd worden d.m.v. een korte film. Tot slot wordt er in deze fase ook nog een eindpresentatie gemaakt.

3.2 Projectgrenzen & randvoorwaarden

Om dit project te doen slagen, moeten zowel Hogeschool Utrecht als Logica aan een aantal randvoorwaarden voldoen en dienen er grenzen opgesteld te worden.

3.2.1 Projectgrenzen

Deze stageperiode duurt van 1 februari tot 1 juli en eind mei dient de scriptie klaar te zijn.

Binnen het project valt

- het zoeken naar een geschikte techniek;
- het maken van een zo compleet mogelijke prototype;
- het maken van een scriptie;
- de Android applicatie maken voor de Nederlandse markt;
- het maken van een (simpele) server applicatie om te demonstreren dat er gecommuniceerd wordt.

Buiten het project valt

- het maken van een complete applicatie (het blijft immers een prototype);
- het rekening houden met de eisen van de eindgebruiker van de server applicatie (server dient dus als proof of concept);
- het implementeren van de Android- en server applicatie.

Samengevat kan de applicatie:

- bepalen waar de gebruiker zich bevindt;
- bijhouden en tonen (d.m.v. een kaart) van de posities en kortingen van winkels;
- loopafstand berekenen en deze in minuten weergeven;
- een gebruiker helpen te navigeren naar een betreffende winkel;
- extra informatie tonen over een winkel; bijvoorbeeld openingstijden of het telefoonnummer.

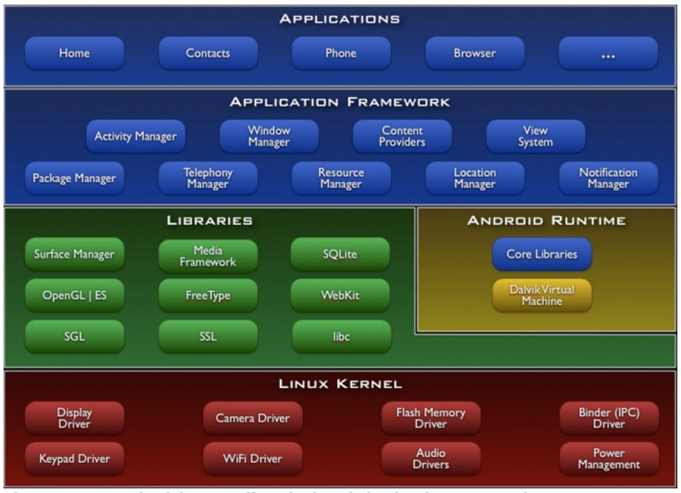
4 Android

Dit hoofdstuk gaat over de beantwoording van twee van de deelvragen die gedefinieerd zijn in hoofdstuk 2 punt 2:

- hoe is een Android applicatie opgebouwd;
- hoe maak je een GUI voor Android;

4.1 Applicatie basisprincipes

Applicaties voor Android zijn geschreven in de programmeertaal Java. Met behulp van Android SDK worden de code en benodigde bronbestanden omgezet naar een Android package. Een apparaat die Android OS draait gebruikt dit bestand vervolgens om de applicatie te installeren. Zodra de applicatie is geïnstalleerd, beschikt het over een eigen beveiliging-sandbox.
Android OS is een multi-user Linux systeem, elke applicatie is dus een aparte gebruiker. Het systeem geeft standaard elke applicatie een uniek Linux user ID. Dit ID is alleen bekend bij het systeem en niet bij de applicatie zelf. Het systeem zet alle permissies voor de bestanden in de applicatie en koppelt deze aan de user ID, zodat de user ID die is toegewezen aan de betreffende applicatie alleen toegang heeft tot die bestanden. Elk proces creëert ook zijn eigen virtuele machine (VM), dit betekent dat de code tijdens het runnen van een applicatie afgezonderd is van de code van andere applicaties. Standaard draait ook elke applicatie zijn eigen Linux proces. Een proces wordt pas door Android gestart als het nodig is. Is het proces niet meer nodig of er moet geheugen vrijgemaakt worden voor andere applicaties dan wordt het proces gesloten.



Figuur 4 - Android's applicatie basisiprincipes van de componenten activities,Services,Content providers,Broadcast receivers,Platform Version en Activating Components

Op deze wijze implementeert Android dus het “principle of least privilege”. Dat betekent dat elke applicatie alleen toegang heeft tot de componenten die moeten worden uitgevoerd om zijn taak te kunnen voltooien. Dit resulteert dus in een veilige omgeving waarbij een applicatie alleen toegang heeft tot de onderdelen van het systeem die het systeem heeft toegewezen. (Bron: Android [2])

4.1.1 Het Android Manifest bestand

Voordat een Android systeem een applicatie component kan starten, moet het systeem weten dat de component bestaat. Om dit te kunnen doen leest het systeem het manifest bestand: AndroidManifest.xml. Deze dient in de root van de project directory geplaatst te worden. Naast het declareren van de applicatie componenten doet het manifest bestand nog een aantal andere zaken, zoals:

- het identificeren van de toegankelijkheid die een applicatie (nodig) heeft zoals toegang tot de contactpersonenlijst;
- het vaststellen van de minimale API Level die een applicatie nodig heeft, dit is dan gebaseerd op de API die gebruikt wordt door de applicatie;

- het vaststellen van de hardware en software functies die gebruikt kunnen worden door de applicatie, zoals een camera;
- het linken van diverse API libraries die een applicatie nodig kan hebben. Dit zijn dus libraries die buiten de Android framework API's vallen.

(Bron: Android [3])

```
<?xml version="1.0" encoding="utf-8"?>
<manifest …>
    <application …>
        <activity android:name="com.example.project.test"
                  android:icon="@drawable/icon.png"
                  android:label="@string/app_name"
                  …>
        </activity>
        …
    </application>
</manifest>
```

Codesegment 1 - een beknopt Android manifest bestand

4.2 Applicatie componenten

Applicatie componenten zijn de bouwstenen van een Android applicatie. Elk component vormt een andere manier om de applicatie te benaderen. De applicatie componenten zijn in vier typen met elk zijn eigen levenscyclus en elk met een eigen doel ingedeeld. Namelijk: "Activities, Services, Content providers en Broadcast receivers". (Bron: Android [4])

4.2.1 Activities

Deze applicatie component biedt gebruikers een venster waarmee zij interacties kunnen uitvoeren. Denk hierbij aan een foto maken of het versturen van een SMS. Een activity krijgt dus een venster waar het zijn gebruikersmenu kan tonen. Een dergelijk venster neemt meestal het hele scherm in beslag. Echter zijn er ook momenten dat het venster veel kleiner is en dat het bovenop andere vensters geplaatst is. Een applicatie is meestal opgebouwd uit meerdere activities die losjes met elkaar verbonden zijn. In de meeste gevallen dient één activity als de hoofdactiviteit. (Bron: Android [5])

4.2.1.1 Activity lifecycle

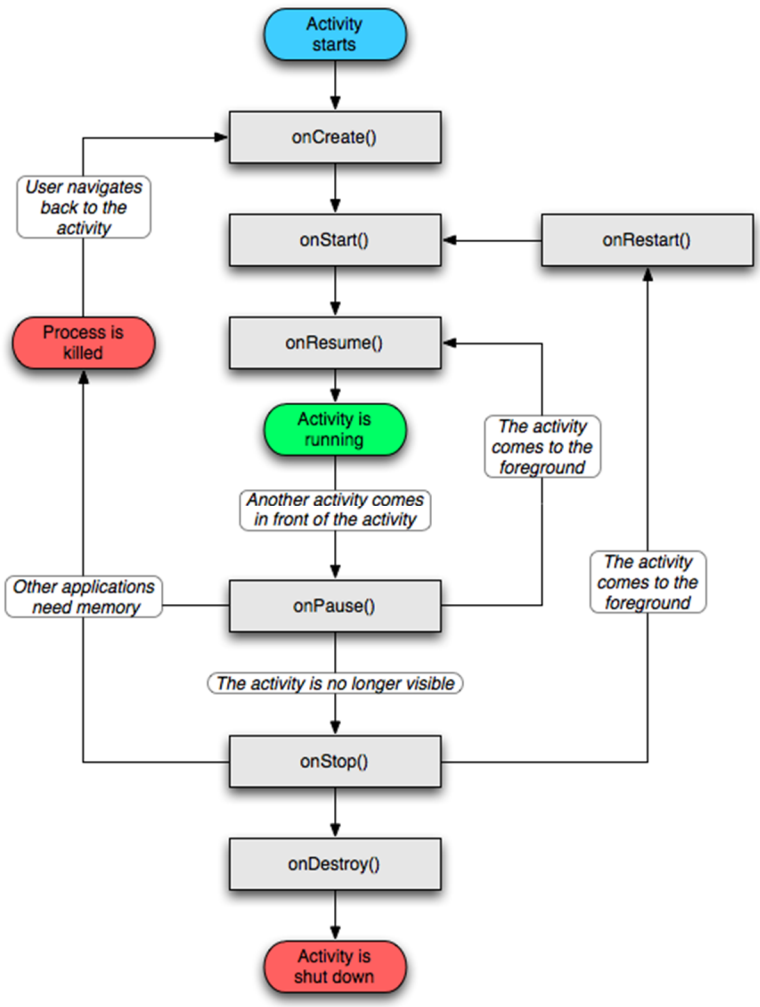
Activities in het systeem worden gemanaged als een activity stack. Als een nieuwe activity gestart wordt, komt deze bovenin de stack terecht. Voorgaande activities komen lager in de stack terecht en komen weer naar de voorgrond als de nieuwe activity beëindigd wordt.

Een activiteit bestaat doorgaans uit drie verschillende toestanden:

Active of running – venster op de voorgrond. Activity bevindt zich bovenin de stack.
Paused – venster niet meer actief maar nog wel zichtbaar.
Stopped – Venster niet meer actief en zichtbaar. Wordt vaak beëindigd door het systeem als er elders meer geheugen nodig is.

Als een activity Paused of stopped (beëindigd): het systeem kan dan de activity compleet beëindigen door het process te stoppen of aan de activity zelf te vragen naar de toestand finished te gaan. Als één van deze twee acties voorkomt en het venster wordt weer aan de gebruiker getoond, dan moet de activity compleet herstart worden om hersteld te kunnen worden naar zijn vorige toestand.

Het onderstaand diagram (Figuur 5) toont de statussen waarin een activity zich kan bevinden. De rechthoeken stellen methodes voor die aangeroepen kunnen worden als de toestand verandert. De ovalen tonen de vier hoofdtoestanden waarin een activity zich kan bevinden.
(Bron: Android [5])



Figuur 5 - levenscyclus Activity (Bron: Android [5])

4.2.2 Services

Een Service is een applicatie component (§4.2) zonder user interface die taken op de achtergrond kan uitvoeren. Een applicatie kan een service starten, deze service blijft vervolgens op de achtergrond draaien ook als de gebruiker wisselt naar een andere applicatie. Het is ook mogelijk dat een component zich kan binden aan een service, deze kunnen dan interacties met elkaar voeren. Hierbij is interprocess communication (IPC) ook mogelijk. Denk hierbij aan bijvoorbeeld netwerk transacties op de achtergrond.

Een service heeft doorgaans twee vormen:

1. Started – een service wordt gestart als een applicatie component zichzelf start door startService() aan te roepen. Als deze service eenmaal gestart is, kan deze onafhankelijk op de achtergrond draaien. Zelfs als de component, die het gestart heeft, beëindigd is. Over het algemeen doet een service een enkele taak en als deze taak voltooid is dan hoort deze service zichzelf te beëindigen.
2. Bound – een service is bound als een applicatie component zichzelf bind via het aanroepen van bindService(). Een service die gebonden is, biedt een client-server interface die het mogelijk maakt dat een component interacties kan uitvoeren met een service. Een service die gebonden is, is alleen actief gedurende de tijd dat een applicatie component eraan gebonden is. Het is mogelijk dat meerdere applicatie componenten zich tegelijkertijd binden aan dezelfde service. Zodra al deze applicatie componenten zich niet meer binden aan de service, zal de service beëindigd worden.

Een combinatie van deze vormen is ook mogelijk, het enige wat er dan gedaan dient te worden is het aanroepen van onStartCommand() (maakt het mogelijk dat een component start) en onBind() (maakt het mogelijk dat er een binding kan plaatsvinden). (Bron: Android [7])

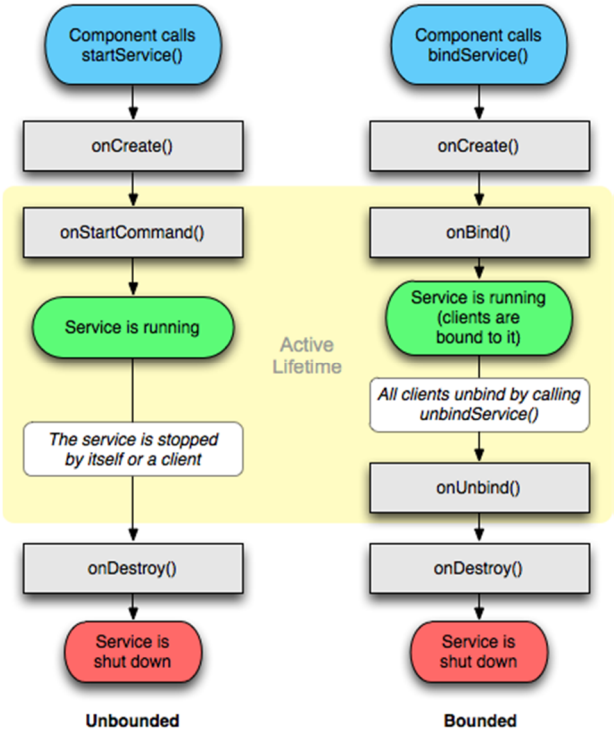
4.2.2.1 Services lifecycle

Een service kan op de achtergrond draaien zonder dat de gebruiker zich hiervan bewust is. Een service lifecycle kan vanaf het moment dat het aangemaakt tot dat het wordt beëindigd twee verschillende paden volgen:

- Started service
  De service wordt gestart als een ander component startService() aanroept. Deze service zal dan voor onbepaalde tijd runnen, het kan dan alleen gestopt worden d.m.v. stopSelf(). Het kan ook gestopt worden met een ander component namelijk: stopService() alleen zodra de service dan stopt wordt deze ook vernietigd.

- Bound service
  De service wordt gestart als een ander component (een client) bindService() aanroept. De client communiceert dan met de service via een IBinder interface (soort remote procedure call mechanisme). De client kan de verbinding verbreken d.m.v. unbindService(). Ook meerdere clients kunnen zich binden aan dezelfde service, als al deze clients zich dan vervolgens unbinden, zal het systeem de service vernietigen (de service hoeft in dit geval niet zichzelf eerst te stoppen).

In het onderstaande figuur (Figuur 6) is de levenscyclus van een service te zien. Het diagram aan de linkerzijde toont de lifecycle van een service die gestart is met startService() (started service) en het diagram aan de rechterzijde toont de lifecycle van een service die gestart is met bindService() (bound service). (Bron: Android [8])



Figuur 6 - de levenscyclus Service (Bron: Android [8])

4.2.3 Content providers

Content providers slaan en halen data op. Zij zorgen er ook voor dat data toegankelijk is voor alle applicaties. Content providers zijn op dit moment de enige manier waarop data tussen applicaties gedeeld kan worden, omdat er geen standaard opslag is waarbij alle Android packages toegang hebben. (Bron: Android [9])

4.2.4 Broadcast receivers

Een broadcoast receiver is een component die reageert op een broadcast van een systeem. Veel van deze berichten komen van het systeem zelf (soms dus ook van applicaties), denk hierbij bijvoorbeeld aan de melding dat de batterijspanning laag is of dat het scherm is uitgeschakeld. Broadcasts receivers tonen geen user interface maar tonen wel mogelijk een 'status bar notification', om de gebruiker op de hoogte te stellen dat er een broadcast event zal plaatsvinden. Zoals Figuur 7, hieronder laat zien.
Een gebruiker kan dan op zo'n icoon klikken om meer informatie getoond te krijgen wat is ontvangen door de broadcast receiver. Tot slot kunnen broadcast receivers ook een activity starten. (Bron: Android [10])

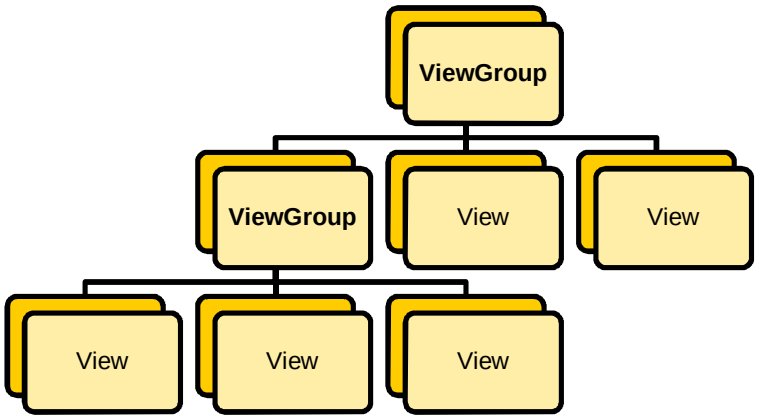
Figuur 7 - status bar met notificatie icoon aan de linkerzijde.

4.3 User Interface

De user interface van een Android applicatie wordt opgebouwd met View en ViewGroup objecten. View objecten zijn de bouwstenen van een user interface op een Android platform. De View klasse dient als basis voor de subklassen genaamd ´Widgets´. Widgets bieden een compleet geïmplementeerd UI object aan, zoals tekstvelden en knoppen (het is ook mogelijk een eigen Widget te maken). Als je kijkt naar de ViewGroup klasse, dient deze juist als basis voor de subklassen ´Layouts´. Een View object is een data cluster waarbij de eigenschappen bijgehouden worden; als layout parameters en content die op een bepaalde plek in het venster geplaatst moet worden. Een view object zorgt zelf voor zijn grootte, indeling, focus, scrollen en interacties in het venster van het gebied waar hij op dat moment actief is. (Bron: Android [11])

4.3.1 View Hierarchy

Bij het Android platform definieert men een Activity UI d.m.v. een hiërarchie bestaande uit View en ViewGroup nodes. Hieronder is te zien hoe zo'n hiërarchische boomstructuur eruit ziet.



Figuur 8 - boomstructuur View objecten

Om er voor te zorgen dat zo'n "boom" weergegeven kan worden op het scherm, moet een activity eerst de methode setContentView() aanroepen, die vervolgens een referentie naar het root node object stuurt. Aan de hand hiervan wordt ook bepaald of het om een child of parent node gaat. (Bron: Android [12])

4.3.2 Layout

De layout in Android wordt gerealiseerd d.m.v. XML omdat men hiermee makkelijk de hiërarchie van de layout kan weergeven. XML biedt immers voor mensen een leesbare taal. Elk element in het XML/bestand behoort tot een View of ViewGroup object. View objecten kunnen gezien worden als de bladeren in een boom en ViewGroup worden gezien als de takken van een boom. Zie ook Figuur 8. De namen die gebruikt worden in het XML-bestand horen overeen te komen met de namen van de klassen van Java. Dit om de structuur tussen de verschillende bestanden te verduidelijken. (Bron: Android [13])

4.3.3 Menu

Een ander onderdeel van de applicatie is het menu. Menu's bieden een interface aan die de functies en instellingen van de applicaties zichtbaar maakt. Menu's worden ook gestructureerd met de zogenoemde View hiërarchie. Deze kan niet zelf gedefinieerd worden. In plaats daarvan wordt de callback methode voor de activity in onCreateOptionsMenu() of de onCreateContextMenu() gedefinieerd. Deze declareert de benodigde items die toegevoegd moeten worden aan het menu. Android maaktdan automatisch de bijbehorende View hiërarchie en tekent hierin de onderdelen van het menu. Een menu handelt zelf zijn events af. Hierdoor is het niet nodig om event listeners toe te voegen zodra een item in het menu is geselecteerd. (Bron: Android [14])

5 Context-awareness

5.1 Het begrip context-awareness

Tegenwoordig komt steeds vaker voor dat informatie- en communicatietechnologie overal, op elk gewenst tijdstip, voor iedereen voorhanden is. Dankzij de huidige technologie en de toekomstige technologie komen steeds meer interessante functionaliteiten binnen handbereik. Zo zijn er mobiele diensten die zich aanpassen aan de steeds veranderde omgeving waarin gebruikers zich bevinden. Er wordt dan gesproken over gepersonaliseerde mobiele diensten die context-aware zijn. “Onder de context van een dienst wordt verstaan: alle omgevingskenmerken die ervoor kunnen zorgen dat de gebruiker een dienst als meer persoonlijk ervaart.” (Bron: Teeuw, W. [15]). Deze informatie kan op twee manieren verkregen worden. Namelijk informatie kan op een automatische wijze (bijvoorbeeld locatie bepaling) worden verkregen of op handmatige wijze worden ingevoerd (bijvoorbeeld persoonsgegevens). “Er wordt betekenis (semantiek) gestopt in de diensten, in de onderliggende netwerken (ambient Internet), devices en players (ubiquitous computing) of in de (web)services (semantic web) die deze diensten ondersteunen.” Zie ook Figuur 9. (Bron: De Heer, J. [15])

5.2 Contextinformatie

Een aantal contextinformatie voorbeelden zijn af te leiden van het Figuur 9 hieronder:

Figuur 9 - Contextinformatie (bron: https://doc.novay.nl/dsweb/Get/Document-24843)

Sociale context
Deze vorm van contextinformatie zegt vooral wat over iemands rol. Dit bepaalt vaak de manier waarop een dienst gebruikt wordt. Het kan namelijk voorkomen dat iemand thuis werkt i.p.v. op het werk. Is de gebruiker bijvoorbeeld een student of een medewerker?

Persoonlijke voorkeuren
Denk hierbij bijvoorbeeld aan de gebruiker: wat is zijn lievelingsgerecht, favoriete televisieprogramma et cetera. Welke voorkeur heeft de dienstenleverancier, de wijze van betaling, hoe het scherm van de dienst wordt ingedeeld et cetera. “De dienst is zich ervan bewust en past zich aan” (Bron: H. van Kranenburg [15]).

Gebruikerslocatie
Een belangrijk onderdeel van vele diensten. Een belangrijk onderdeel vormt plaatsbepaling (zoals ook geldt voor de applicatie waarover deze scriptie gaat). De eerste gedachte die hierbij naar boven komt is: “waar bevindt de gebruiker van een dienst zich?”. De locatie kan gemeten worden in verschillende vormen, (zoals GPS en netwerklocatie) met elk een eigen nauwkeurigheid. Dit geeft verschillende mogelijkheden voor een dienst om zich aan te passen. Bijvoorbeeld waar bevindt zich het dichtstbijzijnde restaurant. Waar bevindt de gebruiker zich (thuis of op kantoor) en welke richting loopt hij op?.

Actualiteiten
Met de contextinformatie actualiteiten wordt bedoeld zaken als actuele weersinformatie of informatie over waar er zich op dat moment files zich voordoen. Weersinvloeden zoals zonneschijn, regen, sneeuw, hagel; zijn allemaal factoren die invloed kunnen hebben op het gebruik van een dienst. Bij regen gaat men liever niet te voet. Met hitte heeft men liever een ijsje (personalisatie).
Ook het geluidsniveau van de omgeving kan invloed hebben, veel lawaai of juist lekker rustig. Of de lichtintensiteit kan een rol spelen (zonnige dag/regenachtige dag).

Authenticiteit en autorisatie
Dit heeft te maken met rechten, heeft een gebruiker toegang tot bepaalde gegevens of diensten of niet. Omgekeerd komt ook voor, dan kan de gebruiker aangeven wat anderen c.q. een dienst, wel of niet mag. Zoals het weigeren dat de dienst reclame gaat versturen of contact op zal nemen met de gebruiker. Welke contextinformatie wordt wel of niet beschikbaar gesteld aan welke diensten? Welke diensten mogen een gebruiker benaderen?.

Device- en netwerkcapaciteiten
Dit zijn de mogelijkheden van een apparaat dat gebruikt wordt. Heeft het apparaat een beeldscherm, welke resolutie heeft het scherm, is het een kleurenscherm, staat het geluid aan, is het stereo of juist mono. Wat is de batterij-capaciteit, welke bandbreedte kan het apparaat aan enzovoorts.

Tot slot kunnen door het gebruik van deze contextinformatie, eindeloos veel diensten gecreëerd worden, die weer ten goede komen aan de gebruiker. Die krijgt immers zeer gerichte diensten aangeboden.

6 Plaatsbepaling

Dit hoofdstuk gaat over de beantwoording van twee van de deelvragen die gedefinieerd zijn in hoofdstuk 2 punt 2:

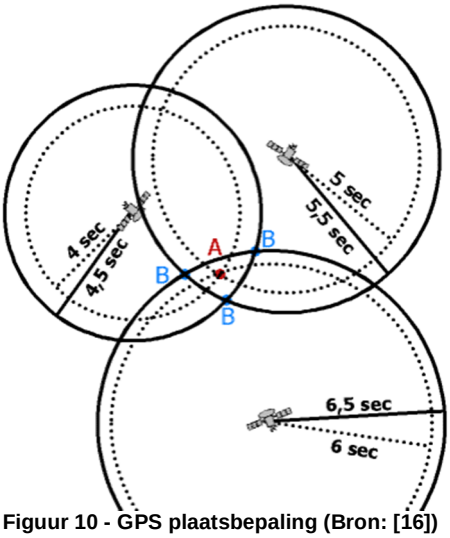
- hoe werkt het digitaal kompas;
- hoe wordt de positie bepaald en welke manier is hiervoor het meest geschikt.

Voor plaatsbepaling zijn een aantal technieken beschikbaar: het kan namelijk via GPS, GSM-masten of aan de hand van een WLAN. De mobiele applicatie zal in eerste instantie gebruik maken van GPS. Als de GPS dienst niet beschikbaar is, zal de applicatie overstappen op de netwerk methode om daarmee de locatie van een gebruiker te bepalen. Er is voor GPS gekozen omdat het in principe overal buiten kan worden gebruikt en dat over het algemeen nauwkeuriger is dan plaatsbepaling via de technieken GSM-masten of WLAN.

6.1 Methoden

GPS

Het Global Positioning System (GPS) kan zeer nauwkeurig werken, dankzij de 24 satellieten die voor de GPS-methode in een baan om de aarde zijn gebracht. Echter voor accurate plaatsbepaling dient er gebruik gemaakt te worden van minstens drie satellieten die zich op dat moment boven de horizon bevinden. De GPS-ontvanger kan vervolgens met behulp van tijd (hoelang het duurt met verzenden/ontvangen van radiosignalen) de precieze afstand bepalen tot deze satellieten (zie Figuur 10). Deze drie cirkels snijden elkaar in principe in één enkel punt zoals Figuur 11 weergeeft. Dit punt duidt dan de plaats aan op aarde waar we ons momenteel bevinden. (Bron: [16])



Figuur 10 - GPS plaatsbepaling (Bron: [16])

Figuur 11 - GPS plaatsbepaling (Bron: [16])

Tot slot heeft GPS een aantal voordelen en nadelen:

Voordelen

- handzaam: een gemiddelde GPS-ontvanger is niet veel groter dan een mobiele telefoon;
- GPS werkt bijna overal;
- eindeloos veel navigatiemogelijkheden, het biedt immers meer informatie dan een kompas;
- de nauwkeurigheid van GPS ligt tussen de tien en twintig meter. (Bron: [16])

Nadelen

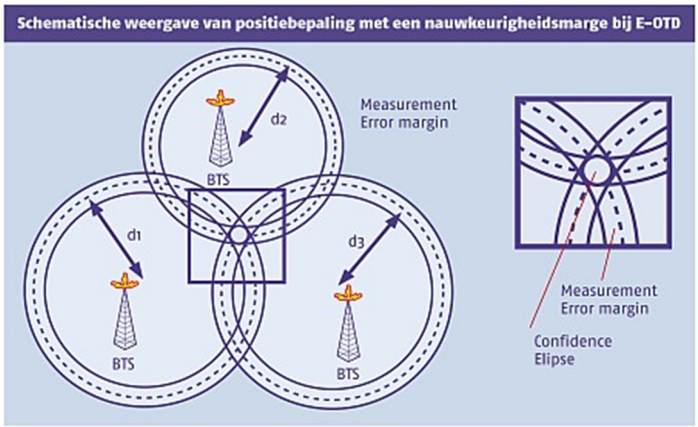
- in bepaalde omstandigheden kan beperkt ontvangst voorkomen (bijvoorbeeld in dichtbebost gebied) Een ontvanger kan dan moeilijker verbinding krijgen met satellieten;
- stroomverbruik van GPS ligt hoog waardoor de accu's van bijvoorbeeld telefoons snel leegraken;
- GPS werkt alleen in de buitenlucht, binnen ligt de nauwkeurigheid erg laag;

GSM-masten
Plaatsbepaling zonder GPS is ook mogelijk. Dit gebeurt aan de hand van het GSM-netwerk van de betreffende mobiele telefoon. Dit kan omdat elke GSM-mast een eigen unieke ID code heeft. De mobiele telefoon ontvangt deze ID's (en de locatie van de mast) en gebruikt net zo als bij GPS de tijd om de afstand te bereken van mobiele telefoon tot GSM-mast, waardoor de locatie van de mobiele telefoon kan worden vastgesteld. (Bron: J. Baaijens, [18])

Enkele voordelen van deze techniek zijn:
- dat het ook binnenshuis werkt;
- het heeft een laag energieverbruik;
- werkt overal zolang er maar voldoende GSM-masten in de buurt aanwezig zijn.

Enkele nadelen van deze techniek zijn:
- minder nauwkeurig dan GPS en de nauwkeurigheid varieert van 500 meter tot 3000 meter;
- niet overal staan GSM-masten;
- afhankelijk van telefoon provider om toegang te krijgen tot het netwerk.

Ondanks dat de nauwkeurigheid van GSM-masten velen malen lager ligt dan GPS, is het een geschikt alternatief voor als GPS niet beschikbaar is. De applicatie zou m.b.v. GSM-masten een schatting kunnen doen waar de gebruiker zich bevindt.

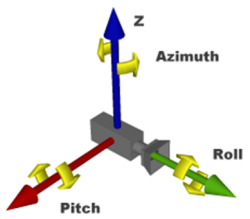


Figuur 12 - GSM-masten plaatsbepaling (bron: van A naar Beter [18])

WLAN
Het Wireless Local Area Network of WLAN kan ook gebruikt worden om de locatie van een mobiele telefoon te bepalen. Er wordt dan een database waarin acces points gegevens staan opgeslagen geraadpleegd. Vervolgens wordt aan de hand van het SSID (**S**ervice **S**et **ID**entifier) van een acces point de locatie opgehaald uit de database. Dit maakt het mogelijk om de locatie van het apparaat te berekenen.

6.2 Digitaal kompas

De Android G1 telefoon maakt gebruik van een digitaal kompas, die het mogelijk maakt om het magnetisch noorden te vinden. Een digitaal kompas biedt in feite dezelfde functionaliteit als een normaal kompas. Alleen zit er geen naald in het digitale kompas maar een magnetometer. Een magnetometer wordt gebruikt om magneetvelden te meten. Tegenwoordig is het mogelijk om een magnetometer in een kleine chip te plaatsen. Deze chips kunnen in meerdere richting het magnetisme meten, een correctie toepassen op het azimuth (dit houdt in dat het noorden gevonden kan worden, ook al wordt het apparaat schuin gehouden). Ouderwetse kompassen werkten niet helemaal juist als deze schuin werden gehouden. (Bron: AWM, [19])



Figuur 13 - Het azimuth, roll en pitch (bron: [23])

7 Het ontwerp

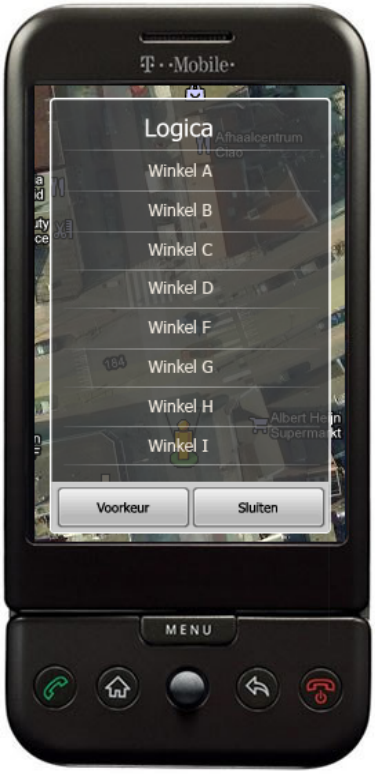
In dit hoofdstuk wordt het ontwerp van het systeem beschreven. Het systeem (een soort real-time reclamefolder) maakt het mogelijk voor een gebruiker die ergens op straat loopt om:

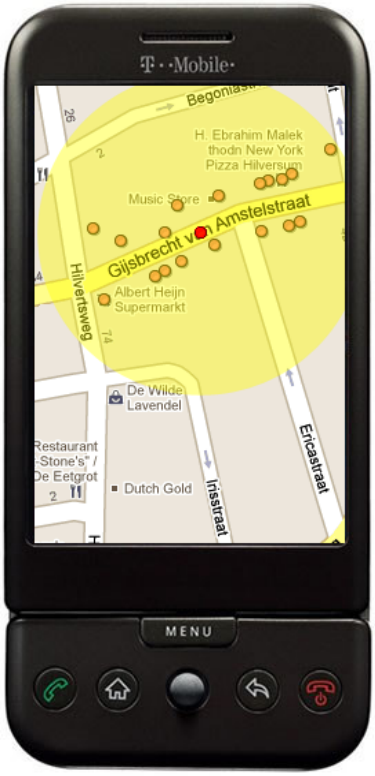
- een kaart te zien met de huidige positie van de gebruiker;
- de posities van winkels die in de buurt liggen aan de hand van de persoonlijke voorkeur van de gebruiker;
- kortingsbericht en / of aanbiedingsbericht te ontvangen;
- extra informatie op te vragen, zoals telefoonnummer, adres, openingstijden, loopafstand en looptijd;
- het weergeven van de te lopen route op de kaart;
- het instellen van de straal waarin gezocht wordt naar winkels die overeenkomen met de persoonlijke voorkeur;
- het kiezen van categorieën (bijvoorbeeld eten en drinken) waar aan de hand van bepaald wordt welke winkels getoond mogen worden.

Om het ontwerp van het systeem weer te geven zijn er aantal documenten geschreven waarvan het functioneel ontwerp en het technische ontwerp (bijlage B en C) de meest belangrijke zijn.

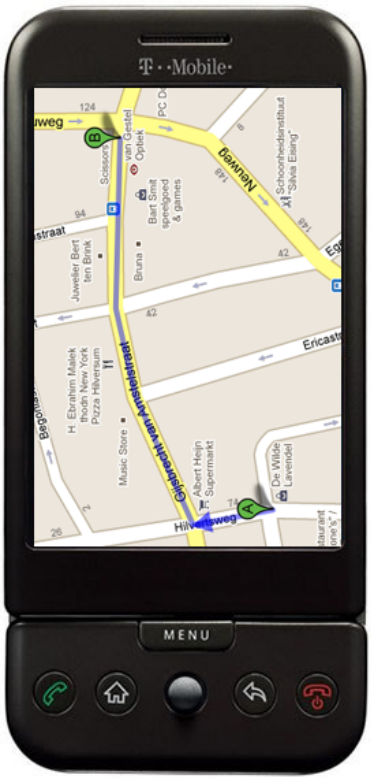
7.1 Het gebruik van de mobiele applicatie

Ter ondersteuning van de opsomming over de werking van de applicatie, is het beknopt uitgeschreven. NB: schermschetsen kunnen afwijken van de werkelijkheid.

Schermschets	Omschrijving
	**[scherm 1]** Een overzicht met winkels die een gebruiker kan ontvangen. Deze lijst is gebaseerd op de voorkeur en positie van de gebruiker, zie ook [scherm 2]. Als de gebruiker op één van deze winkels drukt in de lijst wordt [scherm 5] getoond. Kiest de gebruiker 'Voorkeur' dan is het mogelijk om aan de hand van categorieën op te geven wat voor type winkels gewenst zijn om te tonen. Is er geen voorkeur ingesteld door de gebruiker dan worden standaard alle winkels getoond binnen de straal van 2 km. Kiest de gebruiker 'Sluiten' dan zal de applicatie weer terugkeren naar [scherm 4].

	**[scherm 2]** Deze illustratie dient alleen ter uitleg van [scherm 1]. Het zal niet in de applicatie zelf getoond worden. Hier is de positie van de gebruiker aangeven met een rode stip. Ook wordt de straal (instelbaar) van 2 km weergegeven d.m.v. de gele cirkel. Alle winkels binnen deze cirkel (oranje stippen) worden toegevoegd aan de lijst van [scherm 1] echter moeten deze winkels dan wel overeenkomen met de voorkeur van de gebruiker.
	[scherm 3] Overzicht van categorieën (bijvoorbeeld eten en drinken): hier kan de gebruiker zelf opgeven wat voor type winkels weergegeven mogen worden. Kiest de gebruiker ‘Opslaan’ dan zullen de gekozen categorieën opgeslagen worden op de server. Kiest de gebruiker ‘Sluiten’ dan keert de applicatie terug naar [scherm 4].

	[scherm 4] Een gebruiker loopt met een mobiel in de buurt van of in een winkelstraat. Als de gebruiker kijkt naar het scherm terwijl de applicatie is gestart, wordt de positie en de kijkrichting van de gebruiker getoond. Als er op de telefoon op de menuknop wordt gedrukt, heeft de gebruiker de mogelijkheid die applicatie te beëindigen of de applicatie weer het overzicht met alle winkels te laten tonen. De applicatie keert dan weer terug naar [scherm 1].
	[scherm 5] Op dit venster is het kortingsbericht / aanbiedingsbericht te zien, zoals deze aan de gebruiker getoond wordt. Als de gebruiker kiest voor 'Meer informatie' zal de selectie van de knop oranje oplichten (geldt voor elke knop). Waarna [scherm 6] getoond wordt. Kiest de gebruiker 'Sluiten' dan keert de applicatie terug naar [scherm 4].

	[scherm 6] In dit scherm zal extra informatie over de betreffende winkel getoond worden. Denk hierbij aan het adres, telefoonnummer, loopafstand en looptijd. Kiest de gebruiker voor de optie 'Toon route' dan wordt de route geplot en weergegeven worden in het scherm zie [scherm 6]. Kiest de gebruiker 'Sluiten' dan keert de applicatie terug naar [scherm 4].
	**[scherm 7]** De route wordt getoond op de kaart, als er vervolgens op het menu wordt gedrukt verschijnt de optie om deze weergave te sluiten en keert de applicatie terug naar [scherm 4].

7.2 Use case diagram

Diagram 1 hieronder geeft een overzicht van alle requirements eisen van het systeem d.m.v. een use case diagram.

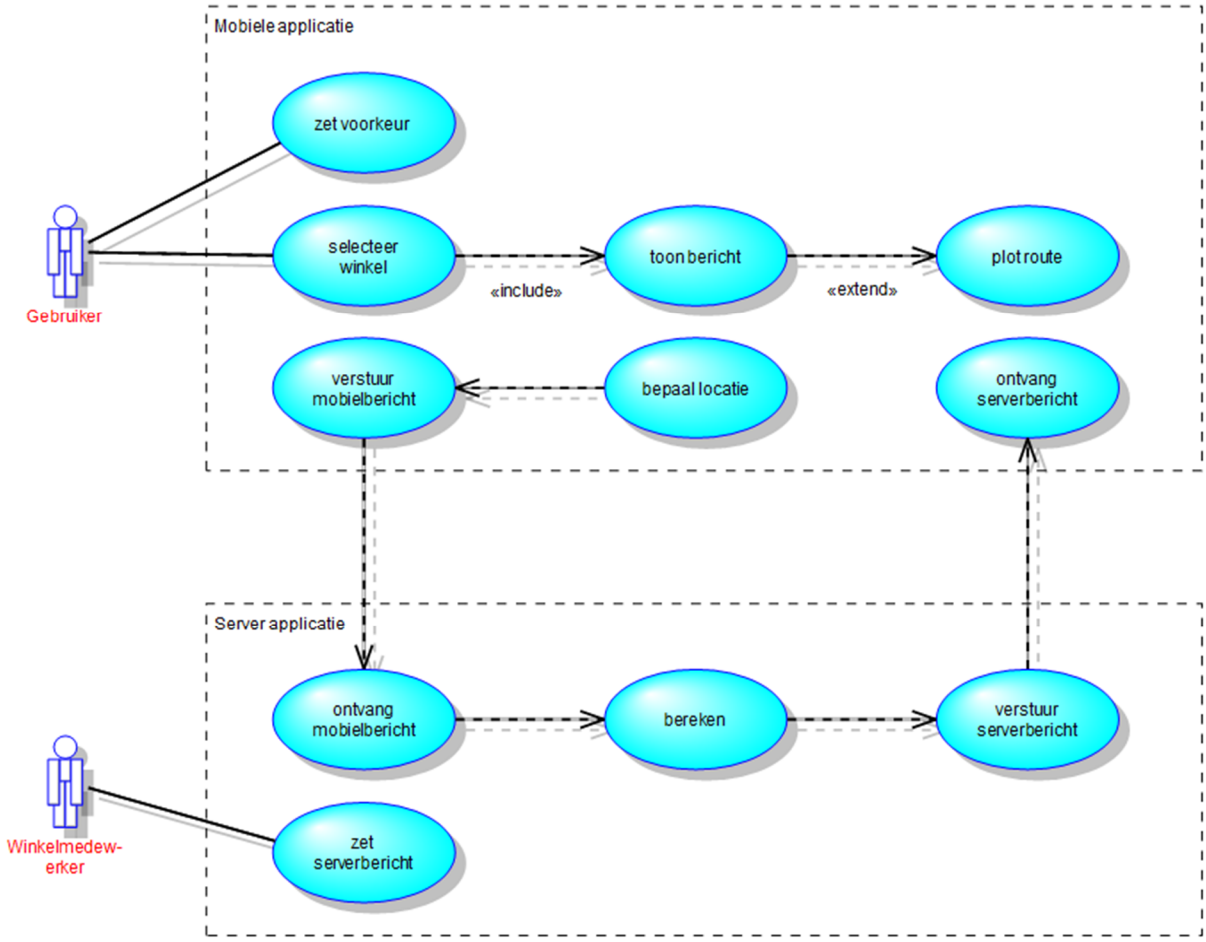


Diagram 1 - use case diagram context-aware navigation

7.2.1 Use case beschrijvingen

Naam	Selecteer winkel
Samenvatting	Een lijst met beschikbare winkels die geselecteerd kunnen worden.
Actor(en)	Gebruiker
Preconditie(s)	De applicatie is opgestart, de locatie van de gebruiker is bepaald en de lijst met een aantal winkels is opgehaald.
Omschrijving	Er wordt een lijst getoond met beschikbare winkels gebaseerd op de voorkeur van de gebruiker. Ook zal de lijst geen winkels bevatten die verder dan 2 km in een straal om de locatie van de gebruiker liggen. 1. De gebruiker kiest een winkel uit de lijst.
Uitzonderingen	Als een winkel geen bericht heeft ingevuld op de server, dan is de betreffende winkel niet zichtbaar voor de gebruiker op de kaart. In het geval dat de lijst leeg is, wordt er een melding aan de gebruiker getoond.
Postconditie(s)	Een winkel is gekozen uit de lijst.

Naam	Zet voorkeur
Samenvatting	Een lijst met categorieën (eten, drinken, elektronica et cetera) die geselecteerd kunnen worden.
Actor(en)	Gebruiker
Preconditie(s)	De applicatie is opgestart. De lijst met categorieën is opgehaald van de server
Omschrijving	Er wordt een lijst getoond met beschikbare categorieën. De gebruiker kan vervolgens kiezen welke categorieën getoond mogen worden. 1. De gebruiker kiest voor opslaan, de lijst met categorieën zal worden opgeslagen op de server. 2. De gebruiker besluit niets op te slaan, alle categorieën zijn dan toegestaan.
Uitzonderingen	N.V.T.
Postconditie(s)	Voorkeur van een gebruiker is ingesteld.

Naam	Toon bericht
Samenvatting	Het kortingsbericht / aanbiedingsbericht van een winkel.
Actor(en)	Gebruiker
Preconditie(s)	De gebruiker heeft een marker geselecteerd en er is een kortingsbericht / aanbiedingsbericht ingesteld door een winkelmedewerker. Een serverbericht is ontvangen.
Omschrijving	1. Het bericht van de winkel wordt opgehaald. 2. Het bericht wordt getoond aan de gebruiker. 3. Gebruiker kan kiezen: ‘Meer informatie’ of ‘Sluiten’ [Gebruiker kiest: ‘Meer informatie’] 1. Extra informatie wordt getoond. 2. Gebruiker kan kiezen: ‘Toon route’ of ‘Sluiten’.
Uitzonderingen	[Bericht kan niet ontvangen worden (bijvoorbeeld geen internet beschikbaar)], toon foutmelding.
Postconditie(s)	Het bericht is weergegeven op het scherm.

Naam	Plot route
Samenvatting	Het grafisch weergeven van de route op de kaart.
Actor(en)	Gebruiker
Preconditie(s)	De gebruiker heeft een marker geselecteerd, ‘Meer informatie’ gekozen en vervolgens op ‘Toon route’ gedrukt.
Omschrijving	Het grafisch weergeven van de route tussen de locatie van de gebruiker en de door de gebruiker geselecteerde marker op de kaart.
Uitzonderingen	[Locatie van de gebruiker kan niet gevonden worden], toon foutmelding.
Postconditie(s)	De route is getoond aan de gebruiker.

Naam	Bepaal locatie
Samenvatting	De locatie van de gebruiker.
Actor(en)	Gebruiker.
Preconditie(s)	Applicatie is gestart.
Omschrijving	De applicatie bepaalt waar de mobiele telefoon zich bevindt ofwel waar de gebruiker op dat moment staat. De gebruiker kan zelf instellen hoe vaak de positie wordt doorgegeven. Verder wordt gekeken wat de kijkrichting is van de gebruiker. Tot slot wordt aangegeven dat de GPS aanstaat.
Uitzonderingen	[Het lukt niet om de locatie te bepalen.], toon foutmelding.
Postconditie(s)	De locatie is bepaald en gecommuniceerd naar de server applicatie.

Naam	Verstuur mobielbericht
Samenvatting	Het verzenden van het bericht naar de server dat afkomstig is van de mobiel. Dit bericht bevat o.a. het ID van de gebruiker en de bijbehorende positie.
Actor(en)	N.V.T.
Preconditie(s)	Applicatie is gestart en locatie van de gebruiker is bepaald.
Omschrijving	De mobiele applicatie heeft bepaald waar de gebruiker zich bevindt en de locatie hiervan en eventuele input (van de gebruiker, marker selectie etc.) in een bericht geplaatst, die vervolgens naar de server applicatie verzonden wordt.
Uitzonderingen	[mobiele applicatie kan geen verbinding maken met de server applicatie], toon foutmelding.
Postconditie(s)	Het bericht dat vanaf de mobiele applicatie wordt verzonden, is ontvangen door de mobiele applicatie.

Naam	Ontvang mobielbericht
Samenvatting	Het ontvangen van het bericht dat afkomstig is van de mobiele applicatie.
Actor(en)	N.V.T.
Preconditie(s)	De mobiele applicatie heeft een bericht verzonden naar de server applicatie.
Omschrijving	De server applicatie ontvangt deze en stuurt de data van het bericht door naar ‘bereken’.
Uitzonderingen	[Het bericht van de mobiele applicatie kan niet ontvangen worden], toon foutmelding.
Postconditie(s)	Het bericht van de mobiele applicatie is ontvangen en doorgestuurd aan ‘bereken’.

Naam	Bereken
Samenvatting	Het berekenen van de verschillende posities van winkels, loopafstand, looptijd aan de hand van de positie van de gebruiker en winkelselectie.
Actor(en)	N.V.T.
Preconditie(s)	Het bericht van de mobiele applicatie is ontvangen.
Omschrijving	Het berekenen van de verschillende posities van winkels, loopafstand, looptijd aan de hand van de positie van de gebruiker. 1. Mobiel bericht wordt ontvangen. 2. De server applicatie voert berekeningen uit. 3. De server applicatie geeft dit door aan ‘Verstuur bericht’.
Uitzonderingen	N.V.T.
Postconditie(s)	De berekeningen zijn klaar en worden doorgegeven aan ‘Verstuur serverbericht’.

Naam	Verstuur serverbericht
Samenvatting	Het versturen van het serverbericht aan de mobiele applicatie.
Actor(en)	N.V.T.
Preconditie(s)	De berekeningen van de server applicatie zijn voltooid. Een medewerker heeft een bericht ingesteld.
Omschrijving	Na het voltooien van de berekeningen wordt er een serverbericht naar de mobiele applicatie verzonden.
Uitzonderingen	[Het versturen van het serverbericht naar de mobiele applicatie mislukt], toon foutmelding
Postconditie(s)	Het serverbericht is verzonden naar de mobiele applicatie.

Naam	Ontvang serverbericht
Samenvatting	Het ontvangen van het serverbericht.
Actor(en)	N.V.T.
Preconditie(s)	De server applicatie heeft een bericht verzonden.
Omschrijving	De mobiele applicatie zal het bericht dat de server applicatie verzonden heeft ontvangen. Waarna de mobiele applicatie in staat is verschillende gegevens te tonen aan de gebruiker.
Uitzonderingen	[Het serverbericht kan niet ontvangen worden], toon foutmelding.
Postconditie(s)	Het serverbericht is ontvangen en de mobiele applicatie heeft gegevens kunnen tonen.

Naam	Zet serverbericht
Samenvatting	Server applicatie waarin het kortingsbericht /aanbiedingsbericht gezet kan worden.
Actor(en)	Medewerker
Preconditie(s)	Een medewerker van een winkel is ingelogd op de server applicatie (backend).
Omschrijving	Het kortingsbericht /aanbiedingsbericht wordt geschreven en opgeslagen in de server applicatie. Het bericht kan ook gewijzigd worden
Uitzonderingen	Winkelmedewerker besluit om geen bericht in te stellen.
Postconditie(s)	Het kortingsbericht /aanbiedingsbericht is ingesteld.

7.3 Het data flow diagram

Om duidelijkheid te verschaffen tussen de relaties van de verschillende componenten (Satellieten, Android G1 telefoon, GeoServer en de database) is hieronder een data flow diagram opgesteld. De componenten wisselen data met elkaar uit, wat het mogelijk maakt dat er kortingsberichten / aanbiedingsberichten (geschreven door een winkelmedewerker) kunnen worden weergegeven aan de eindgebruiker, dat de eindgebruiker zijn locatie kan zien op een kaart en winkels op de kaart worden getoond aan de hand van de persoonlijke voorkeur van de eindgebruiker.

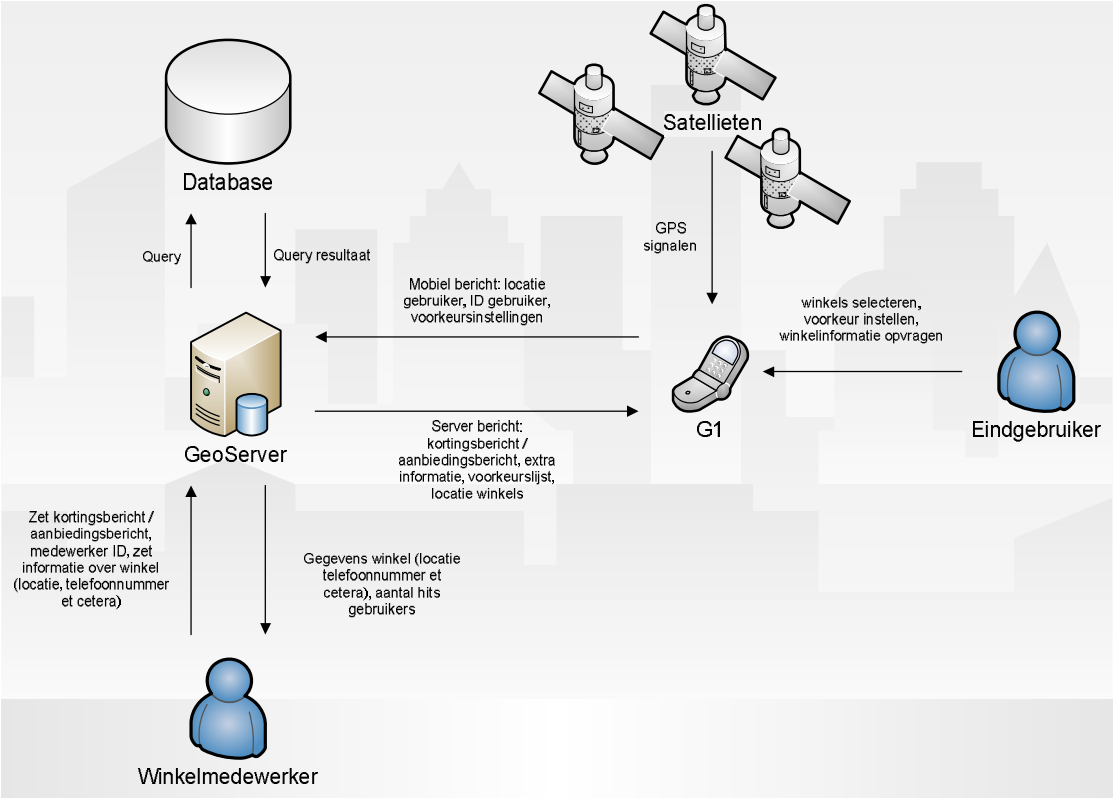


Diagram 2 - Data flow context-aware augmented navigation

Start mobiele applicatie

1. satellieten zenden GPS radiosignalen uit die opgevangen kunnen worden door apparaten zoals de G1;
2. de G1 verzendt data (gebruikers ID, locatie) naar de GeoServer;
3. de Geoserver verzendt de locaties van winkels die in buurt liggen terug, gebaseerd op persoonlijke voorkeur (standaardwaarde: alle winkels in de buurt worden getoond);

Opvragen bericht van winkel

1. de eindgebruiker ziet op de kaart van de applicatie winkels in de buurt en vraagt informatie op van de gekozen winkel;
2. de G1 verzendt data (gebruikers ID, locatie en persoonlijke voorkeur mits gewijzigd) naar de server;
3. de GeoServer ontvangt de data, slaat gegevens zoals de gebruikers ID en persoonlijke voorkeur op in de database;
4. de Geoserver verzendt de data (adres, kortingsbericht / aanbiedingsbericht) van gekozen winkel naar de G1 zodat de eindgebruiker deze data kan bekijken.

Het instellen van een winkelbericht

1. een winkelmedewerker logt in op de backend;
2. een winkelmedewerker wijzigt, verwijderdt of vult gegevens over de winkel in. De wijzigingen worden opgeslagen in de database van GeoServer;

7.4 Communicatie

De communicatie is een zeer belangrijk onderdeel van de applicatie. Zonder de client/server communicatie is de applicatie onbruikbaar! Er wordt gecommuniceerd tussen de server applicatie en de mobiele applicatie. De mobiele applicatie laat aan de server weten waar de gebruiker zich bevindt en wat de voorkeur van de gebruiker is. De server applicatie verstuurt vervolgens de locaties van winkels in de buurt naar de mobiele applicatie. Die deze vervolgens te zien krijgt in een lijst met vooraf opgegeven (door de gebruiker(s)) voorkeuren van welk type winkels, of in sommige gevallen specifieke winkels, getoond mogen worden. De gebruiker zal de voorkeurslijst in kunnen stellen via de mobiele applicatie. Echter is deze lijst niet alleen gebaseerd op de voorkeur van een gebruiker; ook de positie van de gebruiker heeft invloed. De lijst wordt namelijk alleen gevuld met winkels die binnen de straal van 2 km (standaardwaarde) om de positie van de gebruiker liggen. De straal zal instelbaar zijn met een minimale afstand van 1 km. De lijst met categorieën, subcategorieën, winkels en voorkeuren zal op de server worden opgeslagen.
In de toekomst zal dit eenvoudig uitgebreid kunnen worden met winkelsuggesties of met een koppeling aan tijd/leeftijd/geslacht. Als de gebruiker dan een winkel selecteert, zal er een informatiescherm verschijnen met een kortingsbericht en / of een aanbiedingsbericht.

Onder communicatie vallen de volgende zaken:

- verzenden/ontvangen van de positie van de gebruiker;
- het verzenden/ontvangen van het bericht dat afkomstig is van de winkels;
- het verzenden/ontvangen van de posities van winkels;
- het bepalen welke winkels wel of geen voorkeur hebben met een gebruiker.

Het bericht dat afkomstig is van een winkel ziet er als volgt uit als deze door een gebruiker wordt opgevraagd:

Weergave posities.

Kortingsbericht / aanbiedingsbericht van een winkel

Kort bericht met extra informatie. Bijvoorbeeld de looptijd.

Schermschets 1 - verschillende berichten die getoond kunnen worden

7.5 Klassendiagram

In het diagram hieronder is de architectuur van de Android applicatie weergegeven. De applicatie bevindt zich in de package *caan.android* (**C**ontext-**A**ware **A**ugmented **N**avigation).

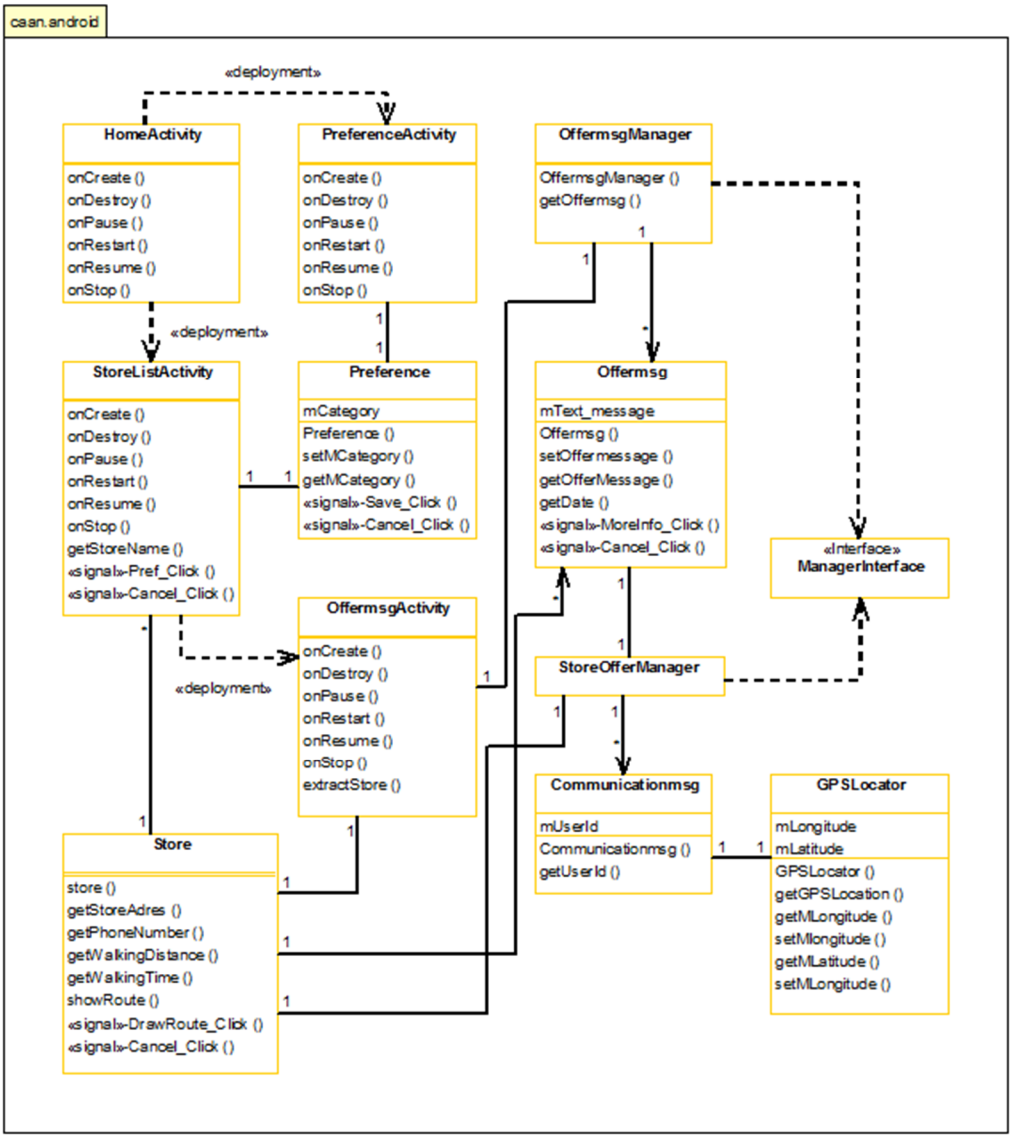


Diagram 3 - Context-aware augmented navigation klassendiagram

7.6 Beschrijving klassendiagram

Klasse	**caan.android::HomeActivity**
Beschrijving	De klasse *HomeActivity* dient als het startscherm van de applicatie. Op dit scherm bevinden zich verschillende User Interface (UI) elementen. Verder handelt het (dankzij het implementeren van listeners) bijvoorbeeld button events af.

Klasse	**caan.android::PreferenceActivity**
Beschrijving	Deze klasse dient als weergave van het scherm voor het proces om een voorkeur te kunnen instellen. Het handelt de input van de gebruiker af zodat deze via de klasse *Preference* in de database op de server opgeslagen kunnen worden.

Klasse	**caan.android::OffermsgManager**
Beschrijving	De klasse *OffermsgManager* zorgt voor een correcte afhandeling van het aanbiedingsbericht /kortingsbericht. Het zorgt ervoor dat het juiste bericht aan de *OffermsgActivity* wordt gegeven.

Klasse	**caan.android::OffermsgActivity**
Beschrijving	Deze klasse wordt gebruikt om het bericht van winkels weer te geven. Dit bericht is een aanbiedingsbericht / kortingsbericht. Aan de hand van dat bericht is het mogelijk meer informatie op te vragen.

Klasse	**caan.android::StoreListActivity**
Beschrijving	De klasse *StoreListActivity* dient als weergave van het scherm voor het proces om een lijst met beschikbare winkels te tonen. Deze lijst is gebaseerd op de positie en voorkeur van een gebruiker.

Klasse	**caan.android::Store**
Beschrijving	De klasse *Store* houdt een aantal zaken bij over de winkels, zoals adres, telefoonnummer en de route (van de gebruikerspositie tot de winkellocatie).

Klasse	**caan.android::StoreOfferManager**
Beschrijving	De klasse *StoreOfferManager* zorgt voor het beheer van de verschillende berichten die langskomen van winkels. Het zorgt ervoor dat een bericht correct is gekoppeld is aan een winkel en/of communicatiebericht en het zorgt ervoor dat een bericht correct wordt afgehandeld.

Klasse	**caan.android::Offermsg**
Beschrijving	De klasse *Offermsg* is bedoeld voor het aanbiedingsbericht / kortingsbericht. Hier wordt het bericht gezet en de datum bijgehouden van wanneer het bericht is aangemaakt.

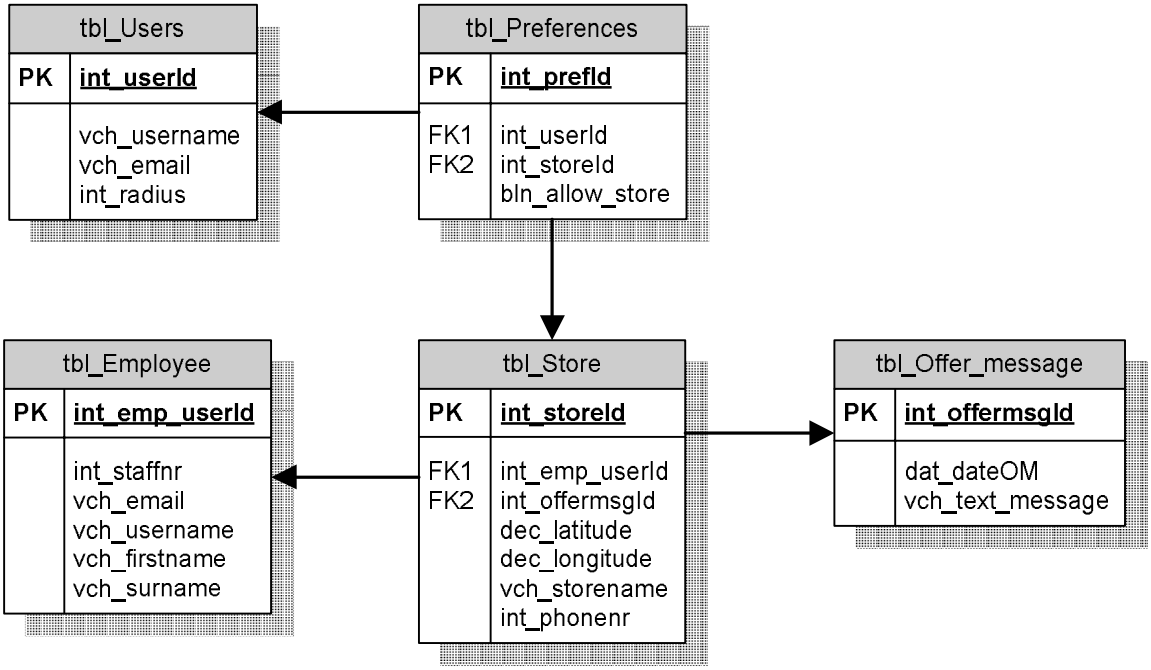
Klasse	**caan.android::Preference**
Beschrijving	Deze klasse zorgt ervoor dat de voorkeur van een gebruiker ingesteld kan worden. Dit gebeurt aan de hand van categorie(ën). De lijst met winkels van de klasse *StoreListActivity* wordt mede door de klasse *Preference* bepaald. Deze lijst met winkels, categorieën, voorkeur zal van de server worden opgehaald en zal op de server worden opgeslagen.

Klasse	**caan.android::Communicationmsg**
Beschrijving	Voor het communiceren tussen de Android applicatie en de server wordt gebruik gemaakt van de klasse *Communicationmsg*. Het bepaalt de ID van de gebruiker en zal de GPS locaties bevatten van de winkels en de gebruiker.

Klasse	**caan.android::GPSLocator**
Beschrijving	Voor het bepalen van de posities wordt de klasse *GPSLocator* gebruikt. Het biedt min of meer een service die meteen GPS locaties kan doorgeven. De GPS sensor geactiveerd, zodra een locatie bekend is in Android wordt de locatie opgeslagen/doorgegeven.

7.7 Het database ontwerp

Hieronder is het ontwerp van de database te zien die op de server aanwezig zal zijn. Er is ook een mogelijkheid om de voorkeuren van de gebruiker op te slaan in een database op de mobiel, echter zal er van die optie geen gebruik gemaakt worden. Er is gekozen om dit op slaan op de server; ten eerste omdat dan de gegevens centraal op één plaats aanwezig zullen zijn en ten tweede omdat dit een relatief snelle mogelijkheid biedt tot extra informatie zoals welke interesses heeft een gebruiker, of welke winkels zijn het meest populair et cetera. Het zou omslachtig zijn dit op te vragen via elke database, die op elke telefoon die gebruikt maakt van de mobiele applicatie aanwezig is.



Figuur 14 - het databasediagram

7.8 Toelichting database

7.8.1 Tbl_Users

Naam	Beschrijving
userId	Uniek nummer van een gebruiker.
username	Gebruikersnaam waarmee geregistreerd is.
email	Het e-mailadres van de gebruiker.
radius	Straal om de positie van een gebruiker heen.

In deze tabel worden de gebruikers opgeslagen, dus de mensen die de applicatie op hun telefoon hebben. Verder wordt hier de gebruikersnaam, het e-mailadres en de straal om de gebruiker heen opgeslagen.

7.8.2 Tbl_Preferences

Naam	Beschrijving
preflid	Uniek nummer van het overzicht met voorkeuren van een gebruiker.
userId	Vreemde sleutel 1, uniek nummer van een gebruiker.
storeId	Vreemde sleutel 2, uniek nummer van een winkel.
allow_store	Bepaald of een winkel wel of niet getoond mag worden aan een gebruiker.

Bij tbl_Preferences wordt de voorkeur van de verschillende gebruikers bijgehouden. Hierin wordt dus vastgezet welke winkels een gebruiker wenst te zien. Standaard (dus als de gebruiker geen voorkeur heeft opgegeven) worden alle winkels getoond op de mobiele applicatie.

7.8.3 Tbl_Offer_message

Naam	Beschrijving
offermsgId	Uniek nummer van het kortingsbericht en / of aanbiedingsbericht van een winkel
dateOM	De datum van het kortingsbericht en / of aanbiedingsbericht.
text_message	Het kortingsbericht en / of aanbiedingsbericht.

In deze tabel staat het bericht van een winkel opgeslagen dat is ingevoerd door een medewerker. Ook wordt bijgehouden op welke datum het bericht is gemaakt.

7.8.4 Tbl_Employee

Naam	Beschrijving
emp_userId	Uniek nummer van een gebruiker.
staffnr	Personeelsnummer.
email	Het e-mailadres van een winkelmedewerker.
username	Gebruikersnaam van een winkelmedewerker.
firstname	Voornaam van een winkelmedewerker.
surname	Achternaam van een winkelmedewerker.

In de tabel tbl_Employee worden de gegevens van medewerkers opgeslagen. Denk hierbij aan hun gebruikersnaam, e-mailadres, volledige naam en eventueel het personeelsnummer.

7.8.5 Tbl_Store

Naam	Beschrijving
storeId	Uniek nummer van een winkel.
emp_userId	Vreemde sleutel 1, uniek nummer van een gebruiker.
offermsgId	Vreemde sleutel 2, uniek nummer van het kortingsbericht en / of aanbiedingsbericht van een winkel
latitude	Breedtegraad van de positie van een winkel.
longitude	Lengtegraad van de positie van een winkel.
storename	Naam van een winkel.
phonenr	Telefoonnummer van een winkel.

De tabel tbl_Store slaat gegevens van de verschillende winkels op. Zoals de positie, winkelnaam en het telefoonnummer. Verder is deze gekoppeld met de tabellen tbl_Offer_message en tbl_Employee.

7.9 Google Maps implementatie

Dit hoofdstuk gaat over de beantwoording van een van de deelvragen die gedefinieerd zijn in hoofdstuk 2 punt 2:

- hoe integreer je Google Maps.

Om de verschillende posities van de gebruiker en winkels visueel weer te geven, is er gebruik gemaakt van Google Maps. Voordat er gebruik gemaakt kan worden van de Google Maps service heeft men een Maps API Key nodig, zodat elke MapView zichzelf kan identificeren met de service. De MapView is een klasse uit de externe library van Maps. Deze klasse zorgt ervoor dat het implementeren van Google Maps eenvoudiger wordt.
Voordat de kaart weergegeven kan worden op de Maps tiles van de MapView controleert de service eerst de API Key. Om er zeker van te zijn:

- dat er een referentie is tussen een certificaat/ontwikkelaar met de service;
- dat de referentie van een certificaat overeenkomt met het certificaat waarmee de applicatie (gebruikmakend van MapView) gesigneerd (signed) is.

Als er niet aan deze twee condities wordt voldaan, zal de service geen Maps tiles aan de MapView bieden, waardoor de kaart niet getoond kan worden.

Het verkrijgen van de API Key is vrij eenvoudig. Als eerste stap moet men het bestand debug.keystore lokaliseren. Deze is aanwezig in de map: USER_HOME/Local Settings/Application Data/android.

De volgende stap is het genereren van het certificaat die een MD5 fingerprint bevat. Dit genereren gebeurt aan de hand van de tool keytool (van Java Development Kit).

C:>keytool -list -alias androiddebugkey -keys
tore .androiddebug.keystore -storepass android -keypass android

Wat de volgende MD5 fingerprint van het certificaat genereerde:
D4:04:E9:57:C8:1B:73:6E:CC:DA:55:C1:4C:22:B4:BC

Deze MD5 fingerprint dient dan te worden ingevuld op:
http://code.google.com/intl/nl/android/maps-api-signup.html wat de unieke API Key zal genereren en terugsturen.

Na het aanmaken van de applicatie heeft het Android manifest bestand (§4.1.1) nog een aantal tags nodig:

```
<uses-library android:name="com.google.android.Maps" />
```

Codesegment 2 – tag van de externe library in het Android manifest bestand

De bovenstaande tag dient om het gebruik van de externe library mogelijk te maken. Om maps (over internet/netwerk et cetera) te benaderen, moet er toestemming opgegeven worden. Dit gebeurt aan de hand van de volgende tags:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission  android:name="android.permission.ACCESS_NETWORK_STATE"
/>
```

Codesegment 3 – Permission tags in het Android manifest bestand

Om de kaart ook daadwerkelijk te tonen in de applicatie moet er een klasse geschreven worden en de main.xml moet worden uitgebreid met de volgende code:

```
<com.google.android.maps.MapView
       android:id="@+id/mapView"
       android:layout_width="fill_parent"
       android:layout_height="fill_parent"
       android:enabled="true"
       android:clickable="true"
       android:apiKey="@string/apiKey"
/>
```

Codesegment 4 – Weergeven van de kaart

Na het uitbreiden van de main.xml moet er een klasse gemaakt worden die ‘extends MapActivity’ bevat. Hiermee wijkt de klasse af van een normale Activity klasse. De kaart zal nu weergegeven worden in de applicatie.

8 Het resultaat

Het overzicht van de functionaliteiten van de applicatie is in Tabel 3 - MoSCoW-prioriteitenlijst te zien. De gemaakte applicatie (prototype) bevat niet alle genoemde functionaliteiten, omdat door de beperkte tijd de functionaliteiten met de prioriteiten ‘could have’, should have’, ‘would have’ eruit zijn gelaten. Wel moet gemeld worden dat de functionaliteit van een straal instellen (‘should have’) wel gemaakt is. Dankzij de prioriteiten indeling van MoSCoW was het mogelijk de functionaliteiten te laten vallen zonder dat de applicatie niet meer naar behoren zou werken, maar ze zijn nog wel nuttig voor het systeem natuurlijk.

De essentiële onderdelen vallen in de prioriteit ‘Must have’; deze zijn dan ook allemaal geïmplementeerd. Hierdoor voldoet het prototype die aan de essentiële eisen.

Voor een aantal onderdelen van de applicatie waren eerst testprogramma’s gemaakt om te leren hoe het werkt en wat wel of niet mogelijk was. Zo was er eerst een ListView (de lijst met bijvoorbeeld de kiesbare categorieën), die de elementen in een animatie toont. Deze heb ik toen gedemonstreerd aan Peter van Rooijen en WT-studenten. Het merendeel hiervan vond de animatie storend. Daarom is besloten deze uit de applicatie te verwijderen.

Het uiterlijk van de applicatie is ook meerdere keren gewijzigd. Met het uiterlijk bedoel ik dan zaken als de markers op de kaart en tussenschermen van de applicatie. Zo werden de markers eerst getoond als gele ballonnetjes. Later kregen de markers hetzelfde uiterlijk als het icoontje dat de huidige positie weergeeft op de kaart, echter zijn deze dan wel Logica geel gekleurd.

Aan de markers is ook nog een custom overlay toegevoegd. Hier door is het mogelijk om een ballon met de vorm van een stripfiguur praatwolkje te tonen. Als de gebruiker een marker van een winkel aantikt, zal het praatwolkje getoond worden met de winkelnaam en zal het beeld centreren op de locatie van de winkel. Waarna het kortingsbericht getoond wordt.

De volgende afbeeldingen worden in de applicatie gebruikt:

Afbeelding	Toelichting
	Launcher icon, hierin is een kompas te zien met het WT-logo erin verwerkt. Deze icon wordt getoond in het menu van de Android telefoon als de applicatie nog niet gestart is.
	Deze icon wordt gebruikt om de huidige positie van de gebruiker aan te geven, ook knippert het blauw.
	Deze icon wordt gebruikt als marker om de posities van de winkels te tonen.
	Het praatwolkje (zonder inhoud) wat te zien is nadat een marker van een winkel is aangetikt.
	Het praatwolkje (focused), wordt tijdelijk Logica geel als het wordt aangetikt.
	De afbeelding die aangetikt kan worden om een praatwolkje te sluiten.

Verder is de gebruiker instaat een straal in te stellen waarin de winkel getoond mogen worden en kan de gebruiker zijn voorkeur instellen. De applicatie is geaccepteerd door de opdrachtgever zoals het op dit moment eruit ziet, na het inleveren van de scriptie worden nog wel wat onderdelen aangepast aan de applicatie. Maar de opdrachtgever gaat ervan uit dat dit niet ten nadelen van de applicatie zou zijn en geeft daarom zijn goedkeuring.

9 Evaluatie

9.1 Persoonlijke evaluatie

Mijn afstudeerstage is begonnen op 1 februari 2011. De eerste dag was er nog geen laptop beschikbaar. Ik ben de tweede dag gelijk begonnen met een indeling te maken van de scriptie aan de hand van de zaken die al bekend waren, deze stonden immers al beschreven in de leidraad. Ook had ik de allereerste vrijdag al een afspraak met Freek Uijtdewilligen. Hierbij is tijdens het overleg besloten dat Freek ook de rol van opdrachtgever op zich zou nemen. De opdracht die ik van Logica had ontvangen was nog erg breed en wat er precies gemaakt moest worden, was nog onduidelijk. Ik heb daarom toen de opdracht verder gespecificeerd en waar nodig de (nieuwe) ideeën besproken, voornamelijk met Freek, maar heb het ook besproken met mijn docent Peter van Rooijen. Het plan van aanpak (PVA) had ik na vier weken af en laten ondertekenen door Peter van Rooijen, Ingrid van Zaanen (project leader) en mijzelf. Logica had gesteld dat je voor een goed PVA zeker vier tot vijf weken nodig zou hebben, ik heb het gelukkig binnen dat tijdframe gered. De planning vond ik wel een moeilijk punt omdat het soms lastig in te schatten was hoeveel tijd bepaalde onderdelen zouden kosten.

Nadat het PVA was afgerond, ben ik begonnen met onderzoeken naar Android. Hoe Android ongeveer werkte en wat het allemaal zou kunnen. Tijdens dit onderzoeken is de opdracht nog verder gespecificeerd, ik had moeite met kiezen tussen pure context-aware of context-aware met de toevoeging van augmented reality. Dit kwam omdat mij augmented reality ontwikkelen ook zeer interessant en leuk leek. Ik ben toen door Ingrid van Zaanen voor de keuze gezet om te kiezen tussen context-aware en augmented reality. Uiteindelijk heb ik ervoor gekozen om context-aware aan te houden, omdat dit ook in de allereerste globale opdrachtomschrijving van Logica stond. Dit is wel een punt waar ik aan zou moeten werken. Want als ik te lang zou blijven twijfelen (wat gelukkig niet gebeurd is) zou ik in tijdnood kunnen komen, terwijl dat helemaal niet nodig hoeft te zijn.

Het maken van het functioneel en technisch ontwerp verliep iets minder vlot dan normaal het geval is. Dit kwam eigenlijk door privé omstandigheden. Normaal gesproken kan ik werk en privé goed gescheiden houden. Maar dit keer had ik er toch wel last van tijdens het werken. Uiteindelijk ben ik weer zo goed als op schema gekomen gelukkig, waardoor ik mijn planning niet al teveel hoefde aan te passen.

Het schrijven van de scriptie was een arbeidsintensief proces, omdat ik tijdens het schrijven van de scriptie tegelijkertijd de applicatie aan het bouwen was. Ik moest soms met het schrijven wachten totdat ik weer een onderdeel ontwikkeld had en kon soms niet verder met ontwikkelen, omdat ik eerst feiten goed moest omschrijven. Ik moest mijn tijd zo efficiënt mogelijk verdelen tussen ontwikkelen en schrijven.

Bij Logica wordt er ook altijd een bedrijfsuitje georganiseerd, alle studenten konden kiezen uit drie ideeën. Er was gekozen voor bowlen en uit eten. Dat hebben we toen ook gedaan, ik heb het daarbij erg naar mijn zin gehad.

Binnen Logica is er ook nog het Spark Innovation Center, hier staan demo’s opgesteld van allerlei toepassingen die ontwikkeld zijn door Logica. Zo staat er bijvoorbeeld een irisscanner en een Microsoft Surface tafel. Persoonlijk vind ik dit een leuke en interessante ruimte, ben er ook meerdere keren geweest en heb ook een rondleiding daar gehad. Ik zou het leuk vinden als mijn demo applicatie ook daar eindigt.

Verder heb ik aan veel activiteiten meegedaan binnen Logica; zo ben ik bij de high tea geweest met Andy Green (hoge baas binnen Logica). Ik heb het evenement van Insp!re bezocht, ik heb daarbij mensen van verschillende afdelingen gesproken en gevraagd wat zij precies deden en wat ik eventueel bij zo'n afdeling zou kunnen doen. Bij Logica was er dit jaar ook voor het eerst een mobile app night georganiseerd o.a. door Raph Bode. Hierbij kwamen mensen die een affiniteit met mobiele applicaties hebben bijeen om erover te discussiëren en verschillende presentaties bij te wonen. Ikzelf heb hierbij een elevator pitch presentatie moeten geven, dit was nog redelijk lastig omdat je dan heel weinig tijd krijgt om een uitgebreide opdracht te vertellen. Ik kreeg maximaal 90 seconden. Bovendien was er nog een grote opkomst bij de mobile app night; er werden zo'n 20 tot 25 mensen verwacht maar uiteindelijk zaten er 84 mensen in de zaal! Iets wat ik minder prettig vond toen ik daar stond. Het laatste evenement wat ik bezocht heb, is die van "Logica career drink" daarbij waren aantal masterclasses aanwezig. Tijdens de career drink werden speeddate sessies georganiseerd waar het mogelijk was in gesprek te gaan met iemand die de masterclasses gaf of net had afgerond. Ben toen veel te weten gekomen over de verschillende classes en heb toen uiteindelijk mijn persoonlijke gegevens achtergelaten voor de masterclass certified Java-developer.

Ook op de afdeling zelf van Working Tomorrow (WT) heb ik het erg naar mijn zin. Ik zit een grote, leuke en vooral gezellige groep. Ik heb van WT ook een aantal mensen geholpen met hun PVA, scriptie en technische problemen. Ook heb ik contact met drie WT afstudeerders van de locatie Arnhem die heb ik leren kennen tijdens de introductiedag van Logica. Tot slot heb ik nog contact met een oud klasgenootje (Richard Joseph) die op dit moment werkzaam is bij Logica in Nieuwegein.

Verder heb ik altijd goed contact gehad met mijn begeleiders (Peter, Ingrid, Freek, Gideon) en ben ik waar nodig goed op weg geholpen. Ik heb duidelijke afspraken kunnen maken en heb de ruimte gekregen iets van de afspraken af te wijken als dat nodig was.

Tot slot vond ik het een zeer leuke en uitdagende stage. Ik vind de begeleiding zeer goed. Zeer geschikt examen programma. Veel kan en mag. Ik ben tevreden en trots op het eindresultaat en ik hoop dat mijn begeleiders dezelfde mening zijn toegedaan.

9.2 Project evaluatie

Voordat ik daadwerkelijk aan het project begon had ik vooraf al een beslissing genomen over welke management techniek ik zou gaan gebruiken. Ik wist immers al in grote lijnen wat het project inhield. Ik heb gekozen voor DSDM omdat ik daar het meest bekend mee ben. Dit betekent niet dat ik geen andere management technieken heb overwogen, ik heb bijvoorbeeld ook naar RUP gekeken.

De volgende stap die ik nam voor het project was om zo snel mogelijk mijn opdrachtgever te spreken. Ik heb toen vragen gesteld over wat voor systeem hij in gedachten had. Toen bleek dat hij nog geen vastomlijnd idee had, heb ik een aantal suggesties gedaan en heb ik met de opdrachtgever zitten brainstormen over het systeem. Vlak daarna heb ik de architect van Logica WT gesproken om te vragen of hij nog ideeën of suggesties had. Hier was helaas niets concreets uitgekomen. Pas tijdens het schrijven van het plan van aanpak kon ik een goede basis applicatie bedenken. In overleg met de opdrachtgever werd duidelijk welke applicatie er ontwikkeld moest worden. Het plan van aanpak is als eerste product opgeleverd na ongeveer vier weken.

Vervolgens ben ik met het onderzoek begonnen om basiskennis op te doen over Android. Nieuw opgedane kennis heeft ervoor gezorgd dat ik tijdens het onderzoek de applicatie met de opdrachtgever nog verder ben gaan specificeren.

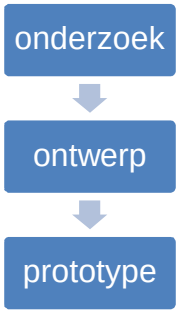
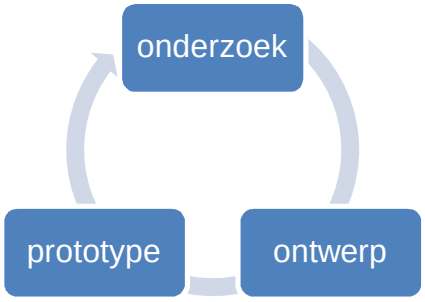
Nadat ik voldoende basiskennis had opgedaan, ben ik begonnen met het functioneel ontwerp, waarbij ik tussendoor de opdrachtgever weer heb gesproken over de specificatie van de

applicatie. Na de goedkeuring hiervan door de opdrachtgever ben ik begonnen met het technisch ontwerp, waarin de applicatie verder is uitgeschreven op technisch vlak. Tijdens het technisch ontwerp zijn prototypen gebouwd om bepaalde bevindingen te testen. Op dit moment in het project was er wat achterstand (ongeveer een week) ontstaan en is de planning daarop aangepast.

Na de afrondingen van het ontwerp ben ik begonnen met de bouw van de applicatie. In het begin verliep dat wat stroef door enkele bugs die ik tegenkwam, §9.3 zal hier dieper op ingaan. Maar later ging het vrij snel, waardoor ik de verloren tijd bijna heb kunnen inlopen. Tijdens de bouw is duidelijk geworden dan dat de gekozen techniek voor de server en database (GeoServer, PostgreSQL met PostGIS) teveel tijd zou kosten om te realiseren binnen de deadline. Na overleg met de opdrachtgever ben ik overgestapt op een server die MySQL en PHP gebruikt en hebben wij besloten om alleen de must haves op te leveren. Het project was weer haalbaar geworden.

Het project verliep over het algemeen vrij soepel, ook moest de planning soms wat aangepast worden. Alle documenten zijn tussendoor opgeleverd aan de opdrachtgever (zelfde persoon als mentor) en aan de project leader en de architect van het Working Tomorrow programma. Sommige documenten zoals het PVA en de scriptie zijn ook opgeleverd aan de docent begeleider. Hierbij is elke keer de ontvangen feedback in de documenten verwerkt.

Fasering
Volgens de planning was het de bedoeling om eerst een onderzoek te doen, daarna een ontwerp te maken en tot slot een prototype te realiseren. In de praktijk bleek dit niet geheel juist te werken, want nieuwe vindingen (zoals bijvoorbeeld hoe bestaande Widgets aangepast kunnen worden) konden tijdens het ontwerpen en ontwikkelen van de applicatie niet direct verwerkt worden in het onderzoek en de nieuwe vindingen van het onderzoek konden niet direct verwerkt worden in het ontwerp en / of in het prototype.
Hierdoor is dus een wat andere werkwijze aangehouden, waarbij ook begonnen is met het onderzoek, daarna het ontwerp en tot slot een prototype. Als er vervolgens uit een test (prototype) bleek dat een bepaald onderdeel gedeeltelijk of niet mogelijk was in de praktijk, werd het ontwerp aangepast. Dit resulteerde in volgende werkwijze:

Van	Naar
	

Tabel 4 - Overgang cyclus van theorie naar praktijk

Hierdoor kon wel gerichter onderzoek gedaan worden en de bevindingen konden daardoor eerder verwerkt worden.

9.3 Oplossingen & motivatie

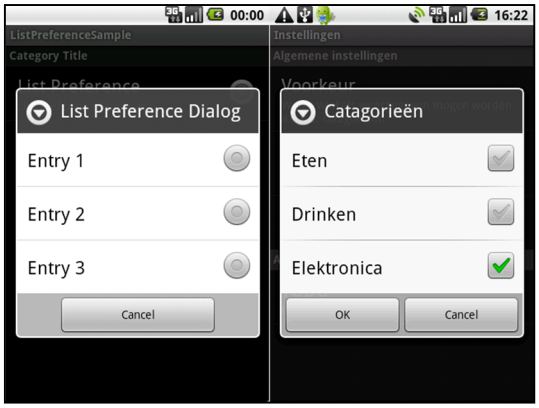
Tijdens deze afstudeerstage hebben zich maar een paar problemen voorgedaan tijdens het ontwikkelen. Zo bleek er een bug in Android 1.5 te zitten, waarbij als men een stijl instelde voor de MapActivity, de kaart zwart werd weergegeven. Gelukkig bleek in de praktijk dat een stijl instellen voor de MapActivity niet nodig is, omdat Google Maps zijn eigen MapView heeft met een eigen stijl.

Het grootste probleem was de laptop die ik in het begin had ontvangen van Logica. Deze had veel beveiligingsinstellingen die een aantal belangrijke zaken zoals de repository van Google SDK blokkeerde, waardoor ik niet in staat was de applicatie verder te ontwikkelen. Als oplossing heb ik een tweede laptop van Logica ontvangen, waarop ik een eigen installatie mocht plaatsen. Hierdoor was ontwikkeling wel mogelijk.

Het kiezen van een geschikte server bleek complex, omdat er ontzettend veel keuze is qua servers die ook geografische data ondersteunen. Er is toen besloten het GEO ICT-team van Logica te benaderen. Daarbij is besproken wat de server allemaal moest doen en kunnen. Zij hebben toen aangeraden om als server GeoServer te gebruiken in combinatie met PostGIS/PostgreSQL (zie bijlage A), dit was iets wat goed aansloot bij de applicatie volgens hen. In de praktijk bleek GeoServer erg omvattend te zijn. Omdat hierdoor tijdnood leek te ontstaan, is besloten na overleg met Freek Uijtdewilligen (opdrachtgever & technische begeleider) om over te stappen op een standaard HTTP-server. Hier is de ontwikkelaar beter bekend mee en dit heeft dus wat tijdwinst opgeleverd om de besproken functionaliteiten alsnog te halen.

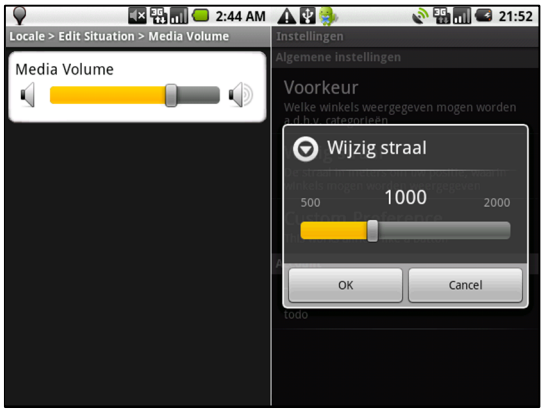
Een ander probleem was dat in één van de eerste opdrachtomschrijvingen augmented reality nog genoemd was om toe te voegen aan de applicatie. Maar vanuit Logica was het de bedoeling ‘iets’ puur met context-awareness te doen. Vanwege die reden en vanwege het feit dat de applicatie zonder de toevoeging van augmented reality al behoorlijk tijdrovend is, is er uiteindelijk besloten om af te stappen van augmented reality. Als er tijd over zou zijn, zou augmented reality altijd nog kunnen worden toegevoegd. Dit is uiteindelijk niet gebeurd.

Als laatste was het probleem nog het vinden voor een alternatief voor de server van GeoServer. Hiervoor heb ik een bekende benaderd en die heeft mij vervolgens toegang gegeven tot zijn server en database. Een mooie bijkomstigheid is nu dat de demo ook echt daadwerkelijk over internet gaat van en naar een server, i.p.v. dat dit gebeurt via een localhost op een laptop.
Ook onderdelen voor de applicatie zelf lagen niet altijd voor de hand. Zo is er het ombouwen van standaard Widgets, bijvoorbeeld de lijst met categorieën die de gebruiker kan instellen en opslaan. Hiervoor is de Widget (ListPreference) van Android zelf gebruikt. Het probleem hierbij is dat een ListPreference geen checkboxes bevat maar alleen een cancel knop heeft en zodra een item van de lijst wordt aangetikt (door een gebruiker) hij dit opslaat en vervolgens de lijst sluit. Om dit op te lossen is er een Java-klasse gemaakt waarin een custom dialog gemaakt wordt, die ervoor zorgt dat er checkboxes zijn en dat meerdere items geselecteerd kunnen zijn. Bovendien bevat deze dialog naast een Cancel knop ook nog een Ok knop die dient voor het bewaren van de ingestelde keuze.



Figuur 15 - Android Widget List preference van standaard (links) naar custom (rechts)

Een andere standaard Widget die is aangepast, is die van een Slider. Dit wordt in Android een Seekbar genoemd. Dit wordt eigenlijk gebruikt om het volume van de telefoon te regelen. Voor de applicatie heb ik de Seekbar gebruikt om de straal om de positie van de gebruiker in te stellen. Om deze Seekbar aan te passen zodat het aan de wensen van het ontwerp voldoet, is er een Java-klasse met ondersteuning van een XML-bestand gemaakt. Die zorgt ervoor dat er een custom dialog getoond wordt, waarbij ook meteen zichtbaar is wat de minimale, maximale en huidige waarde is van de Seekbar. Ook hier zijn er knoppen aanwezig voor het opslaan en annuleren van de instelling.



Figuur 16 - Android Widget Volume Seekbar van standaard (links) naar custom (rechts)

10 Conclusies & aanbevelingen

10.1 conclusie

Tijdens dit afstudeerproject stond één vraag centraal, deze luidde als volgt:

"Op welke wijze kan een Android applicatie gerealiseerd worden die gebruik maakt van contextafhankelijke navigatie om kortingen van omliggende winkels te tonen?"

Door het ontwikkelen van het prototype en het beantwoorden van de deelvragen was ik in staat de Android applicatie te bouwen. De theoretische bevindingen bleken in de praktijk uitvoerbaar te zijn, wat geresulteerd heeft in het prototype: Contexct-aware augmented navigation.

Met de realisatie van het prototype heb ik de centrale vraag beantwoord. Tijdens de realisatie heb ik keuzes moeten maken uit de MoSCoW-prioriteitenlijst. De must haves heb ik gebouwd.
Ik vond dit een erg leerzaam afstudeerproject waarin ik veel heb geleerd over Contexct-awareness en Android. Het lijkt mij interessant om in de toekomst bij meer Android projecten betrokken te zijn.

10.2 aanbevelingen

Er zijn bij dit prototype van context-awareness nog veel verschillende mogelijkheden die nadere uitwerking vereisen. Een aantal van deze mogelijkheden worden hieronder kort uitgelegd.

1. **Toegankelijk maken voor groot publiek**
   Bij nader onderzoek is het advies om te kijken naar eventuele commerciële doeleinden. Om dit haalbaar te laten blijven, moet de applicatie wel aan het volgende voldoen:
   - lage kosten; “Een praktische toepassing die overal en altijd binnen handbereik moet zijn, voor een zo groot mogelijk publiek, moet betaalbaar blijven voor de consument”.
   - robuust; “Het moet gebruikt kunnen worden door de doorsnee consument, die doorgaans niet veel van elektronica weet. Hiervoor is soft- en hardware nodig die daar speciaal voor is aangepast”.
   - zelfstandigheid & samenwerking; “De gebruiker verwacht een enerzijds autonome werking van het systeem, maar wil tegelijkertijd een vlotte samenwerking met andere toepassingen”.
   - tracking-support; “Realtime tracking is waarschijnlijk de belangrijkste vereiste van een context-awareness / augmented reality –systeem”.
   - innovatie; “belangrijk is ook om tijd en moeite te blijven steken in het continu ontwerpen van nieuwe systemen of prototypes”.
   - structuur; “gebruikers vragen praktische voordelen die een goede creatie-structuur eisen”.

(Bron: Wagner, D.[22])

2. **Toevoeging van augmented reality**
   Een ander alternatief is kijken naar de mogelijkheid om augmented reality toe te voegen. Vrijwel elke mobiele telefoon beschikt tegenwoordig over een camera. Men zou dan bijvoorbeeld de posities van winkels kunnen zien via computerbeelden die over de reële beelden zijn geplaatst. Hierdoor is ook meteen duidelijk in welke richting bepaalde winkels aanwezig zijn, zonder dat eerst georiënteerd moet worden op de digitale kaart. De kortingsberichten en / of aanbiedingsberichten zouden dan ook gewoon eenvoudig toegevoegd kunnen worden aan de marker die op dat moment aan een gebruiker getoond wordt.

3. **Het uitbreiden van de voorkeuren**
   Op dit moment maakt de Android applicatie alleen gebruik van voorkeur gebaseerd op categorieën zoals eten, drinken, elektronica et cetera. Dit zou uitgebreid kunnen worden met bijvoorbeeld:
   - geslacht;
   - leeftijd;
   - burgerlijke staat;
   - wel of geen kinderen;
   - et cetera.

   Kijkend naar geslacht is wellicht handig om berichten over bijvoorbeeld maandverband niet aan de man te tonen, maar juist alleen aan de vrouw. Leeftijd kan ook een impact hebben; denk hier bijvoorbeeld aan een winkel met wandelstokken. Iemand van een hogere leeftijd zou eerder naar zo'n soort winkel gaan dan een jonger iemand.

4. **Winkel suggestie**
   De huidige Android applicatie maakt nog niet gebruik van (echte) suggestie. Hiermee wordt bedoeld dat de applicatie alleen winkels toont gebaseerd op de ingevoerde voorkeur van een gebruiker. Echter zou dit concept uitgebreid kunnen worden met de mogelijkheid dat de applicatie zelf gaat "bedenken" in welke winkel de gebruiker wellicht geïnteresseerd zou kunnen zijn. Dit zal dan aan de hand van vorige bekeken berichten en /of bezochten winkels moeten worden bepaald. Dus een gebruiker reageert bijvoorbeeld alleen maar op berichten van een kledingzaak. De applicatie kan dan bedenken om minder berichten te tonen van de andere onderwerpen en suggesties doen van andere kledingzaken.

5. **Analyse gegevens**
   Hier wordt bedoeld dat winkelmedewerkers via de back-end van het systeem, informatie op kunnen vragen over hoe vaak hun kortingsbericht en / of aanbiedingsbericht is bekeken. Hoeveel gebruikers in de buurt of in hun winkel komen of kan aantonen waar bepaalde groepen gebruikers (bijvoorbeeld: leeftijd, geslacht) in geïnteresseerd zijn.

6. **Markers**
   Op dit moment maakt de applicatie alleen nog gebruik van een standaard marker (het gele balletje) om winkels op de kaart te tonen. Echter een mooie toevoeging zou zijn (en een vergroting van de gebruiksvriendelijkheid) om deze markers te vervangen met symbolen van de categorieën of te vervangen met het logo van de betreffende winkel. Hierdoor kan de gebruiker in één opslag kunnen zien om welke winkel en om wat voor winkel het gaat.

Tot slot zijn er nog genoeg toepassingen te bedenken voor deze applicatie. Maar mijn advies is vooral om eerst naar de zes bovengenoemde punten te kijken. Dit zijn punten die een goede basis kunnen vormen voor eventuele commerciële toepassingen.

Bronnen

[1] Stapleton, J, DSDM – De methode in de praktijk

[2] Google Android: http://developer.android.com/guide/topics/fundamentals.html

[3] Google Android: http://developer.android.com/guide/topics/fundamentals.html#Manifest

[4] Google Android: http://developer.android.com/guide/topics/fundamentals.html#Components

[5] Google Android: http://developer.android.com/reference/android/app/Activity.html

[6] Google Android: http://developer.android.com/guide/topics/fundamentals/activities.html#Lifecycle

[7] Google Android: http://developer.android.com/guide/topics/fundamentals/services.html

[8] Google Android: http://developer.android.com/guide/topics/fundamentals/services.html#Lifecycle

[9] Google Android: http://developer.android.com/guide/topics/providers/content-providers.html

[10] Google Android: http://developer.android.com/reference/android/content/BroadcastReceiver.html

[11] Google Android: http://developer.android.com/guide/topics/ui/index.html

[12] Google Android: http://developer.android.com/guide/topics/ui/index.html#ViewHierarchy

[13] Google Android: http://developer.android.com/guide/topics/ui/index.html#Layout

[14] Google Android: http://developer.android.com/guide/topics/ui/index.html#Menus

[15] TU Delft: Johan de Heer, Herma van Kranenburg, Celeste Ponsioen, Wouter Teeuw - Personalisatie en context-awareness van mobiele diensten: https://doc.novay.nl/dsweb/Get/Document-24843

[16] Colleen H. Yinger – Operation and Application of the Global Positioning System: http://www.aero.org/publications/crosslink/summer2002/02.html

[17] Mahaffey J. – TTFF Comparisons: http://www.pocketgpsworld.com/ttffcomparisons.php

[18] Baaijens (2005) J. – Van A naar Beter: http://www.computable.nl/artikel/ict_topics/ictbranche/1360656/2379258/van-a-naar-beter.html

[19] AWM (2010) – Hoe werkt een digitaal kompas: http://www.goeievraag.nl/vraag/digitaal-kompas.29772

[20] Android: http://developer.android.com/resources/tutorials/views/hello-mapview.html

[21] Uijtdewilligen, F. (2010) – A framework for context-awareness applications using augmented reality: A train station navigation proof-of-concept on Google Android (Logica intranet)

[22] Wagner D. (2007) – Handheld Augmented Reality: http://studierstube.icg.tugraz.at/thesis/Wagner_PhDthesis_final.pdf

[23] SpacesEyes3D – Camera mangement: http://spaceyes3d.com/plugin/doc/group___camera.html

[24] Dey A. en Abowd, G. – Towards a Better Understanding of Context and Context-Awareness: Logica intranet

[25] Joseph, R. (2010) – VirtualCloth: Logica intranet

[26] Van der Wal, S. (2010) – Augmented maintenance: garbage control: Logica intranet

Bijlagen

Bijlage	Toelichting	Appendix
Ontwikkelomgeving	Onderzoek naar de hardware en software kant van Android en hoe dit toegepast kan worden in het project.	A
Functioneel Ontwerp	Functioneel ontwerp van het prototype.	B
Technisch Ontwerp	Technisch ontwerp van het prototype.	C
Afstudeervoorstel	Afstudeervoorstel die geschreven is om toestemming te krijgen om de stage te beginnen.	D
Plan van aanpak	Document die te uit te voeren opdracht en de aanpak daarvan beschrijft. Verder verplicht onderdeel om goedkeuring te krijgen voor de stage.	E

Tabel 5 - Appendix overzicht

Verklarende woordenlijst

Term	Toelichting
augmented reality	Het zo realistisch mogelijk toevoegen van computergemaakte beelden aan rechtstreekse, reële beelden.
GUI	Graphical User Interface.
Android	Open source platform voor mobiele telefoons.
mobile application	Een softwarematig programma voor op een mobiele telefoon.
backend	Achterkant van een systeem als het ware, in het geval van deze applicatie de achterkant waar winkels hun kortingen kunnen invoeren.
Widget	Widgets (Engels, letterlijk: dingetje) is de algemene naam voor de elementen waaruit een grafische interface kan worden opgebouwd (vensters, menu's, knoppen, keuzerondje).
Android SDK	Een Software Development Kit (SDK) is een verzameling van hulpmiddelen die nuttig kunnen zijn bij het ontwikkelen van een applicatie voor een bepaald besturingssysteem of type hardware.
Android Development Tools plugin	Dit is een plugin voor Eclipse Deze plugin voegt extensies toe aan Eclipse.
Geoserver	Geoserver is een open-source server die geschreven is in Java. Het stelt gebruikers in staat om geospatial (relateert naar de relatieve positie van objecten op het aardoppervlak) data te laten delen en / of bewerken.
PostgreSQL	PostgreSQL is een object-relationeel database systeem en een open-source project.
PostGIS	PostGIS is een open-source software programma, een soort plugin voor PostgreSQL die het mogelijk maakt om geografische objecten toe te voegen aan de PostgreSQL relationele database.

Tabel 6 – Verklarende woordenlijst

Appendix A

Ontwikkelomgeving

Appendix A: Ontwikkelplatform

Ontwikkelomgeving

De eerste stap voor het ontwikkelen van de Android applicatie is het opzetten van een geschikte ontwikkelomgeving. Dit gebeurt aan de hand van de onderdelen, namelijk: Integrated Development Enviroment (IDE), Android (SDK), Android Development Tools (ADT) en een emulator. Uiteindelijk wordt de applicatie ook getest op een echte mobiele telefoon van het type smartphone met Android OS 1.5.

Integrated Development Enviroment

Ter ondersteuning voor het ontwikkelen op Google Android is er een developer site opgericht (http://developer.android.com/index.html, deze site biedt goede ondersteuning en uitleg over hoe er geprogrammeerd kan worden op Android. Deze site en zijn vele gebruikers raden aan om Eclipse te gebruiken als editor. Een belangrijke reden hiervoor is dat Eclipse dankzij de ADT-plugin goede ondersteuning kan bieden aan het ontwikkelen van een applicatie voor Android.

Android SDK

Een Software Development Kit (SDK) is een verzameling van hulpmiddelen die nuttig kunnen zijn bij het ontwikkelen van een applicatie voor een bepaald besturingssysteem of type hardware. Ook voor Android is er zo'n SDK gemaakt voor ontwikkelaars; Android SDK.

Android Development Tools plugin

Dit is een plugin voor Eclipse. Deze plugin voegt extensies toe aan Eclipse. Dit versoepelt en vergemakkelijkt het ontwikkelen voor Android applicaties. Zo is er een wizard aanwezig om een geschikt Android project aan te maken. Door deze wizard worden alle minimaal benodigde bestanden voor het project automatisch toegevoegd. Deze plugin geeft ook de mogelijkheid om te kunnen debuggen. Dit debuggen kan dan aan de hand van de emulator (ook onderdeel van de ADT-plugin) of via een echte Android telefoon.

Debuggen op de telefoon zelf

Het is mogelijk om de applicatie te debuggen op een telefoon waarop Android draait. Zo'n telefoon dient dan wel zo geconfigureerd te worden. Als de telefoon juist geconfigureerd is, kan de telefoon worden aangesloten aan de computer via een USB-verbinding.

Server: 'Geoserver'

Voor de server wordt gebruikt gemaakt van Geoserver op aanraden van het Logica GEO ICT team. Geoserver is een open-source server die geschreven is in Java. Het stelt gebruikers in staat om geospatial (relateert naar de relatieve positie van objecten op het aardoppervlak) data te laten delen en / of bewerken. Het is ontworpen om diverse systemen en organisaties de mogelijkheid te geven om te kunnen samenwerken. (interoperabiliteit). Het publiceert de gegevens van elke grote geospatial gegevensbron met behulp van open standaarden. Geoserver is een handige methode om bestaande gegevens te verbinden met Virtual Globes. Denk hierbij bijvoorbeeld aan Google Earth of web-based kaarten zoals Google Maps. Tot slot zal Geoserver gebruikt worden in combinatie met PostgreSQL en PostGIS database. Voor meer informatie over de database zie §7.7.

PostgreSQL

PostgreSQL is een object-relationeel database systeem en een open-source project. Het biedt de standaardfunctionaliteit van een database, plus extra functionaliteiten die normaal gesproken alleen voorkomen in database management systemen. Het is gebouwd door vrijwilligers overal ter wereld via het internet. Het wordt dus niet beheerd door een bedrijf.

PostGIS
PostGIS is een open-source software programma, een soort plugin voor PostgreSQL die het mogelijk maakt om geografische objecten toe te voegen aan de PostgreSQL relationele database.

Appendix B

Functioneel ontwerp

Appendix C

Technisch ontwerp

Appendix D

Afstudeervoorstel

Appendix E

Plan van aanpak

LOGICA PLC / PROF. W.H. KEESOMLAAN 14 / 1183 DJ / AMSTELVEEN / TEL: +31 (0)20 503 3018
www.logica.com

Locatie: Amstelveen / lichting 12