



# Amigo iPad

Maarten van Arem

2012

## Titelpagina

### Gegevens student

Naam + voorletters student  
Studentnummer  
Opleiding  
Afstudeerperiode

van Arem M.H.J.  
2108357  
HBO ICT Software Engineering  
05-09-2011 t/m 10-01-2012

### Gegevens bedrijf

Naam bedrijf/instelling  
Plaats

TASS Technology Solutions  
Eindhoven

### Gegevens bedrijfsbegeleider

Naam  
Functie

Josée Zielman  
People Manager

### Gegevens technisch begeleider:

Naam  
Functie

Peter Vink  
People Manager

### Gegevens verslag

Titel afstudeerverslag  
Datum uitgifte afstudeerverslag

Amigo iPad  
09-01-2012

---

Getekend voor gezien door de bedrijfsbegeleider, Josée Zielman:

Datum:

## Voorwoord

Dit verslag is geschreven naar aanleiding van mijn afstudeeropdracht aan de Fontys Hogeschool voor de opleiding HBO-ICT Software Engineering. Binnen het bedrijf TASS Technology Solutions te Eindhoven is er zelfstandig een onderzoek- en ontwikkelopdracht uitgevoerd over de mogelijkheden die de iPad biedt binnen het Amigo netwerk van TASS.

Via deze weg wil ik mijn dank uitbrengen aan iedereen die een bijdrage heeft geleverd aan de totstandkoming van dit verslag. Daarnaast wil ik in het bijzonder mijn dank uitbrengen aan mijn bedrijfsbegeleiders Josée Zielman en Peter Vink en de docentbegeleider Tom Broumels voor de uitstekende begeleiding.

Maarten van Arem  
Januari 2012

## Inhoudsopgave

1. Inleiding.....	7
2. Het Bedrijf .....	8
2.1 Geschiedenis.....	8
2.2 Organisatie .....	9
3. De Opdracht .....	10
3.1 Voorafgaand .....	10
3.1 Amigo .....	10
3.2 Beginsituatie .....	10
3.3 Doel.....	11
3.4 Omschrijving opdracht.....	11
4. Methoden en werkwijze.....	12
4.1 Fasering .....	12
4.2 Iteraties .....	13
5. Oriëntatiefase .....	14
5.1 Amigo .....	14
5.1.1 Amigo Specificatie.....	15
5.1.2 B.A.P.A.O. ....	15
5.2 Probleemstelling .....	15
5.3 Oplossing.....	16
6. Onderzoek .....	17
6.1 Amigo communicatie met iOS .....	17
6.2 Amigo service Discovery.....	18
7. Eindproducten .....	20
7.1 Basic Amigo Communicator.....	20
7.2 Amigo Home Console.....	21
7.3 Bonjour Amigo.....	21
8. Conclusies en aanbevelingen.....	22
Bijlage A: Project Management Plan	
Bijlage B: Software Requirements Plan	
Bijlage C: Software Test Plan	
Bijlage D: Amigo Consortium	
Bijlage E: Onderzoek Amigo Besturing	
Bijlage F: Onderzoek Amigo Services	
Bijlage G: Onderzoek Amigo Discovery	
Bijlage H: Onderzoek Bonjour Amigo	

## Samenvatting

De in dit document beschreven afstudeeropdracht is uitgevoerd bij TASS Technology Solutions te Eindhoven. TASS is een top-3 ICT dienstverlener actief op het gebied van technische software ontwikkeling.

De opdracht bestaat uit het maken van een nieuwe applicatie die de mogelijkheden van het Amigo systeem kan demonstreren. Amigo is een Europees project waarin vijftien bedrijven samenwerken om de mogelijkheden van thuisnetwerken maximaal te benutten om het leven van de mens te vergemakkelijken. De applicatie maakt het mogelijk om via een iPad met het Amigo netwerk te kunnen communiceren en zal als promotiemateriaal dienen voor het Amigo netwerk van TASS.

Het project is gefaseerd uitgevoerd via het Tien Stappen Plan (TSP), waarbij eerst een grondige oriëntatie van de huidige situatie is uitgevoerd. Daarna is er besloten om de onderzoeks- en oplossingsfase in zeven iteraties te verdelen waarbij de eerste twee iteraties onder onderzoek vallen en in de resterende vijf iteraties de daadwerkelijke oplossing gerealiseerd wordt. Elk deelproduct van deze vijf iteraties bestaat uit een planning, ontwerp, realisatie en tests.

Het doel van het project is om een Home Management Console applicatie te realiseren voor de iPad. Deze applicatie kan ontdekken wat voor Amigo apparaten er op het netwerk aangesloten zijn. Verder moet de applicatie kunnen communiceren met deze apparaten om bijvoorbeeld een lamp aan of uit te zetten.

Het communiceren met de verschillende apparaten bleek geen moeilijke klus te zijn na het onderzoek naar de werking van deze apparaten. Het ontdekken van apparaten bleek echter wel moeilijk te realiseren. De beschikbare oplossingen waren voor andere platformen geschreven, voor het iPad platform was er geen oplossing 'out of the box' beschikbaar. Een eigen oplossing realiseren zou er voor zorgen dat de focus van het project volledig zou verschuiven naar het ontdekken van de apparaten terwijl het doel was om een Home Management Console te maken voor de iPad. Daarom is er voor gekozen om het ontdekken van de apparaten door een externe applicatie uit te laten voeren die op een Windows PC draait waarmee de applicatie kan communiceren.

Er kan geconcludeerd worden dat het eindproduct voldoet aan de belangrijkste eisen en meer. Door het onderzoek dat vooraf is uitgevoerd zijn de risico's van het project laag gebleven tijdens de oplossingsfase. Door de extra tijd zijn er extra features in de applicatie verwerkt zoals een plattegrond waar de gebruiker de apparaten per locatie kan zien en aansturen. Daarnaast zijn er ook nog een aantal kleinere features zoals een Twitter-feed, een weerbericht en een nieuws-feed. Zodoende is de Amigo Home Console tot een succesvol en compleet demonstratieproduct gegroeid.

Tijdens het afstuderen is er gebruik gemaakt van het iOS ontwikkelplatform waar TASS nog geringe ervaring mee had. Met de kennis die is opgedaan tijdens deze opdracht kan TASS in de toekomst het iOS platform vaker overwegen om software in te ontwikkelen.

---

## Summary

The in this document described graduation assignment has been conducted at TASS Technology Solutions in Eindhoven. TASS is a top-3 ICT service provider active in the area of embedded software development.

The assignment consists of making a new iPad application that showcases the possibilities of the Amigo system. Amigo is a European project where 15 companies work together to maximize the potential of the possibilities that a home network has to offer, with the goal to improve our daily lives. The application that will be made during this assignment will be used as a demonstration product that will showcase the possibilities that an iPad has to offer for the Amigo network.

The project is carried out in phases with first a thorough orientation of the current situation. The research and solution phase was split up into seven iterations where in the first two iterations the main research for the project was carried out. In the next five iterations the solution for the project was made, whereas in every iteration a plan, design, implementation en tests were made.

The goal of the project is to create an application for the Apple iPad which is able to discover what kind of Amigo devices are connected and can be controlled via WiFi network. The application must be able to communicate with these devices so that it is possible, for example, to turn a lamp in a room on or off.

The research towards communicating with different Amigo devices in the network showed that there were no problems getting it done. Discovering these devices did raise some problems, the available solutions were all written in different languages with no solution made for the iPad platform. To create a whole new solution from scratch was no option since the project's focus would have to be moved towards making Discovery possible on the iPad, where the actual goal of the project was to create a Home Management Console application for the iPad. For this reason the decision was made to create a separate application that would fulfill the task of discovering the devices, the iPad would then connect to this service which provides a list of available Amigo devices.

With these solutions the conclusion can be made that the final product meets all the required elements and more, by conducting the critical pieces of research at the beginning of the project the risk has been drastically reduced. Because of this low risk it was possible to create extra features for the Home Console like a Twitter-feed, a weather report, news-feed and a floor plan where the user can interact with the various devices and see their location. These features have made sure that the Amigo Home console has grown to a successful demonstration product.

For the creation of the application the iOS framework was used, TASS did not have a lot of experience with this framework yet. With the knowledge gained from this assignment TASS will be able to see iOS as a possibility for future projects.

---

---

## Verklarende woordenlijst

<b>Middleware</b>	Een stuk software die de communicatie tussen twee verschillende systemen regelt
<b>XML</b>	Extensible Markup Language, een manier om gegevens gestructureerd weer te geven in tekstvorm
<b>Projectgroep</b>	Alle personen die betrokken zijn bij het project, in dit document betreft het de volgende personen: Maarten van Arem, Josée Zielman en Peter Vink
<b>SDK</b>	Software Development Kit, een verzameling van software die het mogelijk maakt om software te realiseren voor een specifiek platform
<b>TSP</b>	Tien Stappen Plan, deze methode verdeelt het afstudeertraject in 3 fases met in totaal 10 stappen om het afstudeertraject succesvol af te ronden
<b>iOS</b>	Het huidige besturingssysteem voor Apple apparaten, dit besturingssysteem wordt voor iPads, iPhones, Apple TV's en iPods gebruikt
<b>SOAP</b>	Simple Object Access Protocol, een manier van communiceren over het internet tussen twee verschillende systemen
<b>PMP</b>	Project Management Plan, in dit document staan alle organisatorische zaken van het gehele project beschreven
<b>BAPAO</b>	Big Amigo PC that's Always On, dit kan zowel een Windows PC zijn als een Plugtop waar Linux op draait, deze twee computers zijn het hart van het Amigo netwerk, indien deze computers niet aan staan zal het Amigo netwerk niet functioneren
<b>WSDL</b>	Web Services Description Language, een in XML geschreven formaat waarin beschreven wordt wat voor mogelijkheden een service bevat
<b>Tuxdroid</b>	Een robot die lijkt op de pinguin van de Linux mascotte. Deze robot is aan te sturen via SOAP, waardoor je de ogen, mond en armen kunt bewegen

---

## 1. Inleiding

Het aantal tabletgebruikers is de laatste jaren enorm gestegen. Deze sterke groei zorgt er voor dat vele bedrijven interesse tonen om applicaties te ontwikkelen voor deze apparaten. De markt voor tablets is nog enorm in opkomst door de sterke technologische ontwikkelingen binnen de tablet markt. TASS is als dienstverlener op het gebied van technische en embedded software natuurlijk zeer geïnteresseerd in de mogelijkheden die een tablet kan bieden voor zowel nieuwe als bestaande oplossingen.



**Figuur 1.1: High-end Tablets**

Door de innovatieve hardware en technologie is het mogelijk om vernieuwende applicaties te maken waarbij de gebruiker niet meer voor een laptop of pc hoeft te zitten maar vanaf elke willekeurige plek zijn of haar taken kan uitvoeren. Met deze gedachte heeft TASS een opdracht ontwikkeld voor een iOS applicatie waarmee een Home Console Management Systeem gerealiseerd moet worden.

In het volgende hoofdstuk wordt informatie gegeven over het bedrijf TASS Technology Solutions, waar het afstuderen heeft plaatsgevonden. De daarop volgende hoofdstukken geven uitleg over de aanpak van de opdracht, na dit inleidende deel wordt het onderzoek en de uitwerking van de opdracht, inclusief resultaat, besproken. Tot slot wordt er teruggekeken naar het verloop van de opdracht in een conclusie en evaluatie.



## 2. Het Bedrijf

TASS Technology Solutions is een dienstverlener op het gebied van technische en embedded software. TASS heeft vestigingen in Eindhoven, Apeldoorn en Brussel en levert met meer dan 200 medewerkers een grote expertise in de vorm van detachering en projecten. Als top-3 dienstverlener voor technische en embedded software levert TASS een belangrijke bijdrage aan de ontwikkeling van intelligente en innovatieve producten.

### 2.1 Geschiedenis

In september 1978 besloten werknemers van de voormalige automatiseringsdivisie van Philips genaamd 'Information Systems and Automation' (ISA), een nieuwe software afdeling te beginnen onder de naam 'Special Projects Group' (SPG). SPG specialiseerde zich in het ontwikkelen van technische software voor verschillende productdivisies van Philips. Na verschillende naamswijzigingen ging vanaf 1990 deze afdeling door het leven als Philips TASS.

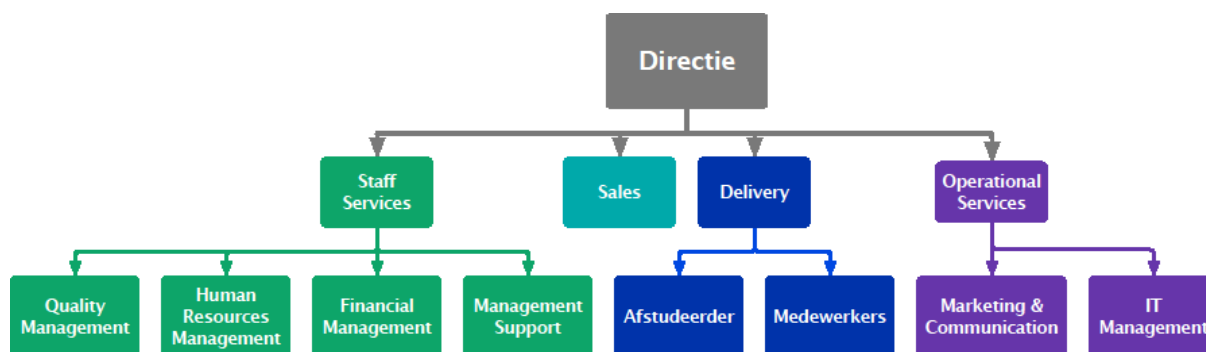


Figuur 2.1: TASS Eindhoven

In 1995 werd Philips TASS een onafhankelijke B.V. en verleende vanaf dat moment diensten op het gebied van technische en embedded software aan externe klanten. Philips TASS groeide in 3 jaar tijd zo hard dat in 1998 TASS International werd opgericht wat bestond uit een Nederlandse divisie en een Belgische divisie.

Philips besloot zich in 2007 meer te gaan richten op de core business van het bedrijf zijn core-business, zodoende werd Philips TASS verkocht en werd het onderdeel van Total Specific Solutions (TSS). Philips TASS werd TASS Technology Solutions maar het duurde echter tot januari 2008 voordat TASS geen onderdeel meer van Philips was en ook een eigen kantoor had op het Larixplein te Eindhoven.

## 2.2 Organisatie



Figuur 3.2: TASS Organogram

De kernactiviteit van TASS bestaat uit het detacheren van medewerkers, hierdoor speelt de afdeling Sales een belangrijke rol. Deze afdeling zorgt ervoor dat de klantenportefeuille gevuld blijft. Wanneer een medewerker akkoord gaat met een aangeboden opdracht wordt hij of zij gekoppeld aan een People Manager, die een begeleidende rol zal vervullen binnen de opdracht. Deze begeleiding bestaat uit toezicht op de opdracht en een periodieke beoordeling van de medewerker, met eventueel een beloning bij goede prestaties. Alle afstudeerders, stagiairs, gedetacheerde werknemers en People Managers zijn onderdeel van Delivery en staan onder leiding van algemeen directeur Edwin Manten.

De afdelingen Operational Services en Staff Services zijn ondersteunende afdelingen, zo zorgt de afdeling Operational Services voor de interne IT voorziening (IT Management) en de zowel interne als externe communicatie (Marketing & Communications).

Onder Staff Services vallen de ondersteunende taken zoals personeelzaken (Human Resources), financiën (Financial Management), Kwaliteitsbewaking (Quality Management) en ondersteuning van het management (Management Support).

### 3. De Opdracht

De opdracht bestaat enerzijds uit een onderzoek en anderzijds uit het maken van nieuwe software die aansluit met het bestaande Amigo netwerk. In dit hoofdstuk wordt eerst behandeld wat Amigo inhoudt. Vervolgens wordt de beginsituatie, de probleemstelling en het doel van de opdracht uitgelegd. Tenslotte volgt de volledige omschrijving van de opdracht.

#### 3.1 Voorafgaand

De keuze om bij TASS af te studeren is een interessante keuze als Software Engineer. TASS is bekend als een ICT bedrijf dat voor het grootste deel mensen in dienst heeft die dicht op de hardware programmeren en dus een Technische Informatica achtergrond hebben. TASS heeft echter veel mobile afstudeeropdrachten in haar portfolio waarbij enkele opdrachten ook geschikt zijn voor software engineers. Mobile development is op dit moment zeer sterk in opkomst, echter zijn veel bedrijven nu nog aan het experimenteren met deze nieuwe systemen. De keuze voor de opdracht was daarom zeer simpel, met het oog op de toekomst.



#### 3.1 Amigo

Vijftien Europese leidende bedrijven in mobiel- en thuisnetwerken, software ontwikkeling, consumentenelektronica en huishoudelijke apparaten werken samen aan Amigo. Dit project benut de mogelijkheden van thuisnetwerken maximaal om het leven van de mens te vergemakkelijken. Een overzicht van het Amigo Consortium is te vinden in Bijlage D: Amigo Consortium.



Samenwerkingen tussen verschillende elektronica apparaten in thuisnetwerken zijn steeds meer in opkomst. Denk bijvoorbeeld aan Apple TV in combinatie met een iPad waarmee video's en foto's naar TV gestreamed kunnen worden en het delen van je muziekcollectie via iTunes. Een complete samenwerking tussen alle verschillende elektronica in een huishouden is echter nog niet gelukt. Redenen hiervoor zijn complexe installatieprocedures en het gebrek aan samenwerken tussen apparaten van verschillende fabrikanten.

Het zoeken naar en het maken van oplossingen voor deze problemen is het doel van het Amigo project. Het project heeft open source, gestandaardiseerde middleware en aantrekkelijke gebruikersservices ontwikkelt om de bruikbaarheid van een thuisnetwerk te verbeteren. Zodoende wordt de installatieprocedure gemakkelijker gemaakt en samenwerken tussen apparaten van verschillende fabrikanten wordt mogelijk.

#### 3.2 Beginsituatie

TASS heeft verschillende demo's op het Larixplein beschikbaar die op Amigo zijn aangesloten. Zo zijn er in het gebouw op verschillende locaties lampen die aan en uit kunnen worden gezet via het internet en ook een deur die bestuurd kan worden via het internet.

Er zijn al verschillende manieren waarop deze apparaten aangestuurd kunnen worden. Zo is er een website en een Android applicatie beschikbaar die kunnen communiceren met de verschillende beschikbare apparaten.

### **3.3 Doel**

Bij TASS wordt er al veel gedaan op het gebied van mobile development, veel van de applicaties die gemaakt worden zijn echter ontwikkeld voor het Android platform. Het iOS platform is nog nieuw voor TASS, waarbij men nog niet veel kennis heeft van de mogelijkheden die het platform kan bieden.

Het Amigo project is een uitstekende case om meer iOS development kennis bij TASS te vergaren. Het heeft de potentie om goed te laten zien wat de mogelijkheden van iOS zijn, hoe het ontwikkelen van een iPad applicatie in zijn gang gaat en hoe iOS samen kan werken met bestaande technieken zoals het Amigo netwerk.

### **3.4 Omschrijving opdracht**

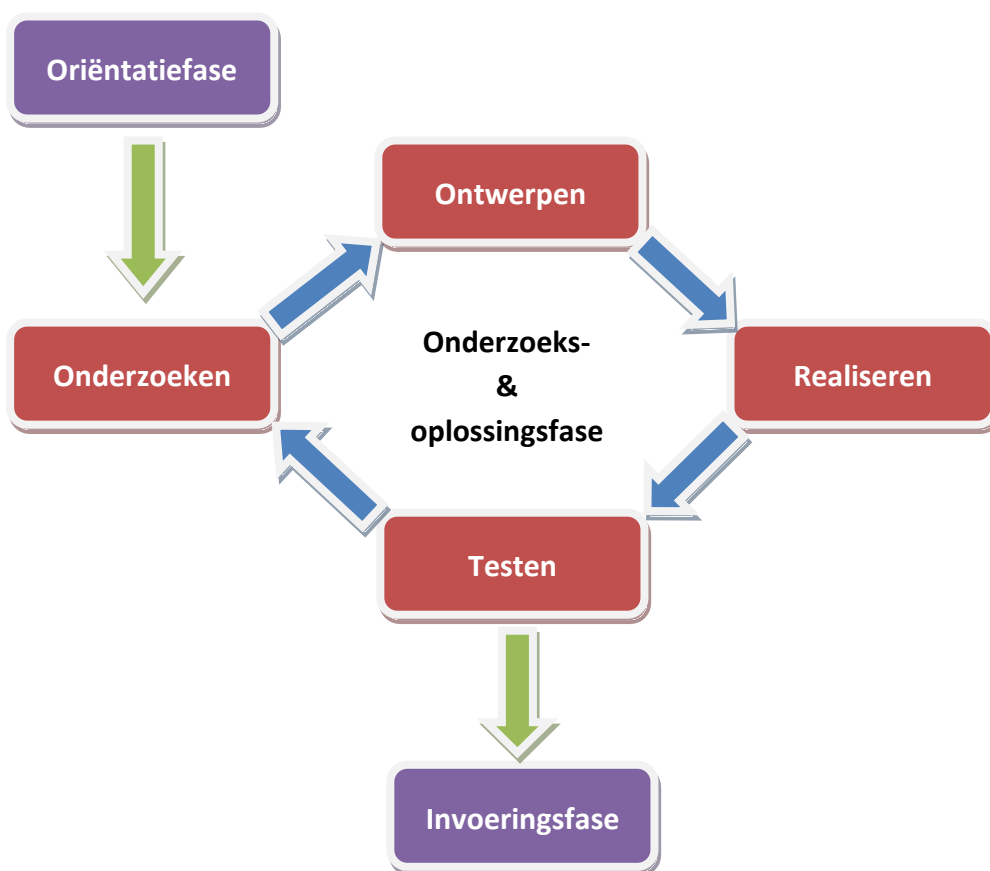
Voordat er een volledige iPad applicatie ontwikkeld kan worden, moet er goed gekeken worden naar de werking van iOS met de technologie van het Amigo netwerk, waar het communiceren en ontdekken van services de grootste rollen spelen.

Het uiteindelijke doel is om een iPad applicatie te ontwikkelen waar de gebruiker met verschillende apparaten op het Amigo netwerk kan communiceren via een gebruiksvriendelijke user interface. Met dit doel in gedachte zal de applicatie aanvoelen als een echte Home Management Console.

## 4. Methoden en werkwijze

Voor de realisatie van dit project is er gebruik gemaakt van het Tien Stappen Plan (TSP) in combinatie met de Scrum methode.

De project fasering is opgedeeld in 3 fasen: de oriëntatie, onderzoeks- & oplossingsfase en de invoeringsfase. Omdat het project in 7 iteraties opgedeeld is, komt tijdens elke iteratie de onderzoeks- en oplossingsfase weer terug. Het voordeel van deze werkwijze is dat er aan het einde van elke iteratie een werkend eindproduct is. Waar ook indien het nodig is op teruggevallen kan worden.



Figuur 4.1: Project Fasering

### 4.1 Fasering

De eerste fase van het project is de definitiefase, in deze fase is het PMP (Project Management Plan) opgezet in overleg met TASS. Daarnaast is er ook de projectrapportage waarin alle stappen van het uit te voeren project beschreven staan. Ook is het Software Requirements plan opgesteld in deze fase.

Deze documenten bevatten allemaal geen concrete definitie van het eindproduct. De basiseisen die aan het product gesteld worden staan wel vast, maar de precieze uitwerking en werking van het eindproduct zullen per iteratie behandeld worden. De eerste iteraties zijn vooraf ingevuld omdat in deze iteraties de basisfunctionaliteiten van de te maken oplossing geïmplementeerd worden.

In de onderzoeksfase zijn de mogelijkheden onderzocht die het Amigo netwerk bied in combinatie met iOS. Er is gekeken naar hoe Discovery mogelijk gemaakt kan worden op iOS en hoe er gecommuniceerd kan worden met de beschikbare Amigo apparaten in het netwerk. Deze resultaten zijn verwerkt in verschillende onderzoeksdocumenten waar elk deelonderzoek uitgelicht wordt.

De oplossingsfase komt in elke iteratie terug. Er wordt eerst een ontwerp gemaakt voor de te maken functionaliteiten, vervolgens wordt er voor elke functionaliteit een beschrijving geschreven, zodat het duidelijk is wat de functionaliteit inhoudt. Het ontwerp wordt vervolgens bij de projectgroep voorgelegd en wordt zo nodig aangepast. Indien er geen wijzigingen nodig zijn is het ontwerp af voor de iteratie en worden de beloofde functionaliteiten gerealiseerd. Aan het einde van de oplossingsfase wordt de oplossing in zijn geheel bekeken en getest op fouten en performance. Indien er tijd voor is worden er verbeteringen uitgevoerd. Indien er geen tijd is, worden deze taken doorgeschoven naar de volgende iteratie.

Tijdens de invoeringsfase wordt het eindproduct klaargemaakt voor gebruik en beheer. Zo wordt er documentatie geschreven over hoe het programma gebruikt kan worden en hoe het programma werkt. Het ontwerp wordt volledig bijgewerkt zodat toekomstige stagiairs of medewerkers sneller bekend raken met het project en zo sneller aan de slag kunnen.

## 4.2 Iteraties

Het project is opgedeeld in zeven verschillende iteraties waarbij de eerste twee iteraties zijn ingevuld met onderzoeksproducten en de resterende vijf iteraties met deelproducten voor de daadwerkelijke applicatie. Deze vijf iteraties bestaan elk uit verschillende stappen. Aan het begin van de iteratie wordt er naar het huidige deelproduct gekeken, in combinatie met het software requirements document wordt er gekeken naar wat er nog ontbreekt in de gemaakte oplossing en wat er in de beschikbare tijd gedaan kan worden in de iteratie. Deze functionaliteiten worden vervolgens besproken in het project team tijdens de wekelijkse vergadering en indien nodig aangepast.

Vervolgens wordt, indien vereist, een deelonderzoek uitgevoerd om te beslissen hoe een bepaalde functionaliteit kan worden verwerkt in het programma (bijvoorbeeld communicatie met een service via iOS).

Daarna worden de beloofde functionaliteiten gerealiseerd en getest. Tijdens de test wordt er gekeken of de verschillende functionaliteiten voldoen aan de beschrijvingen in het gemaakte design document. Ook wordt gekeken naar de prestaties van de gemaakte oplossing (werkt de applicatie snel genoeg en waar is nog ruimte voor verbetering). Deze gegevens worden vervolgens teruggekoppeld met de projectgroep en gaan mee naar de volgende iteratie.

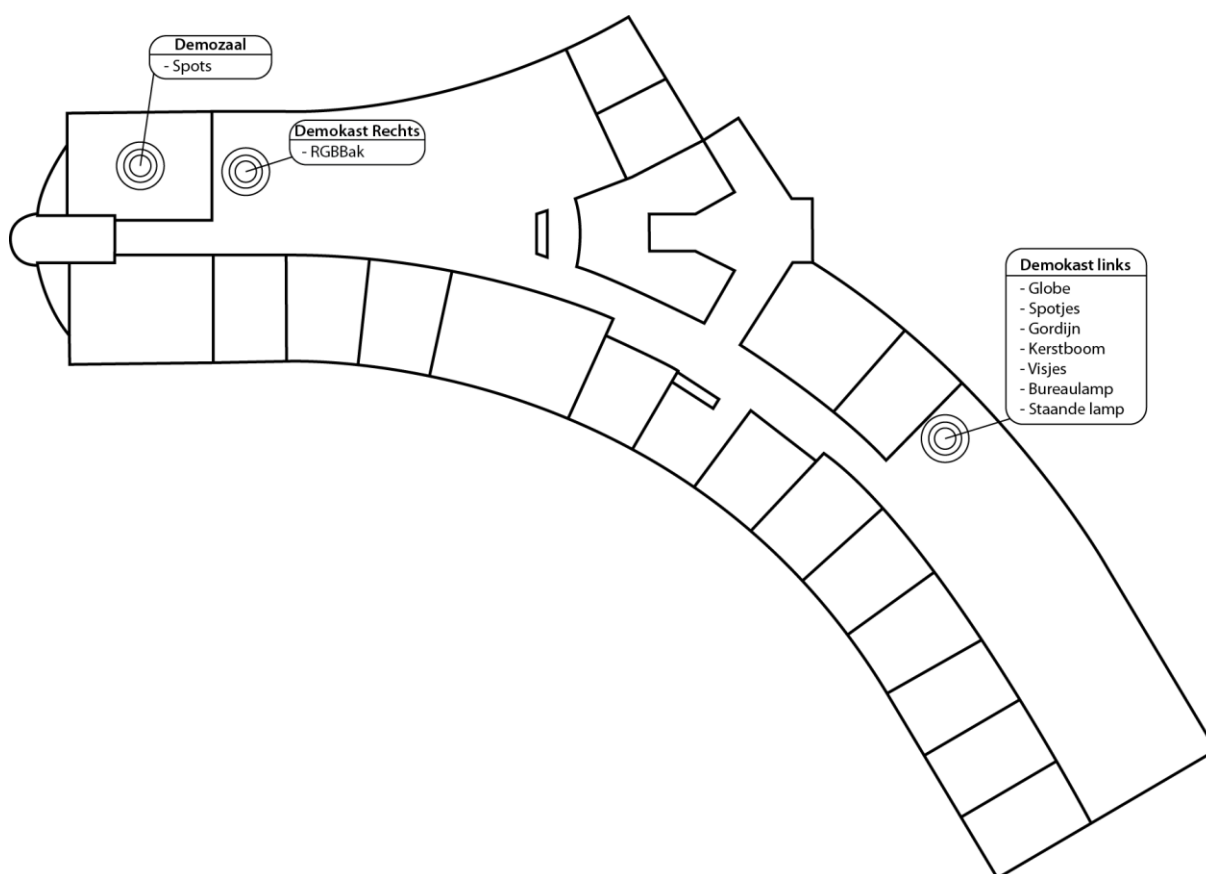
## 5. Oriëntatiefase

De oriëntatiefase is de eerste fase binnen het project. Dit hoofdstuk beschrijft de oriëntatie van de huidige situatie, de te maken oplossing voor de probleemstelling en de afbakening van het gehele project.

### 5.1 Amigo

Om beter inzicht te krijgen in de opdracht is er eerst gekeken naar de huidige situatie. De applicatie die gebouwd moet worden moet foutloos met de omgeving kunnen communiceren. Dit betekent dat de huidige apparaten aangestuurd moeten worden, maar dat ook nieuwe apparaten die voldoen aan de Amigo specificaties meteen aanstuurbaar zijn via de applicaties.

De huidige Amigo opstelling bij TASS is in kaart gebracht in Figuur 5.1. In deze figuur is te zien welke apparaten voldoen aan de Amigo specificatie en extra apparaten die wel via het netwerk aangestuurd kunnen worden maar over een eigen specificatie beschikken.



Figuur 5.1 Plattegrond Amigo apparaten per locatie

### 5.1.1 Amigo Specificatie

Op de Tuxdroid en Doorcontroller na hebben alle apparaten een zelfde manier van communiceren. Deze manier van communicatie tussen deze systemen heet SOAP (Simple Object Access Protocol). Zodoende heeft elk apparaat een eigen SOAP service waardoor bijvoorbeeld een applicatie met dezelfde netwerkverbinding op kan vragen wat een apparaat voor mogelijkheden biedt en hoe de applicatie moet communiceren met het apparaat. De aansturing van de Tuxdroid en Doorcontroller is daarom niet vereist in de opdracht.



Figuur 5.2 De Tuxdroid

### 5.1.2 BAPAO

De BAPAO is een PC of plugtop waarop verschillende services draaien die het mogelijk maken om alle beschikbare apparaten aan te sturen. BAPAO is een afkorting voor "Big Amigo PC that's Always On", dit betekent dus dat deze PC altijd aanstaat zodat de services altijd beschikbaar zijn. Deze services worden automatisch gestart bij het opstarten van de BAPAO. Als de BAPAO uitgezet wordt dan is het gehele Amigo netwerk niet meer beschikbaar, totdat de PC weer aangezet wordt.

## 5.2 Probleemstelling

Wanneer men het over Amigo heeft, heeft men het over gemak van bediening. De mogelijkheid bestaat om vanuit de luie stoel het hele huis te besturen, of het nu het openen van een deur is of het aanzetten van een lamp, het is allemaal mogelijk. Een gebruiksvriendelijke oplossing om vanuit elke willekeurige plek apparaten in het huis aan te sturen is echter nog niet beschikbaar. Men moet ofwel voor de tv ofwel voor de computer zitten.

Op dit moment is er een mobiele Android applicatie die dat mogelijk maakt, echter is deze applicatie verre van af. Het toont een lijst met alle beschikbare apparaten waaruit je er één kan kiezen en aansturen. Het oogt erg technisch en niet af. Wel laat het zien dat het mogelijk is om via een smartphone of tablet apparaten aan te sturen.

Er is binnen TASS vraag naar een iOS applicatie waarmee de gebruiker via een gebruiksvriendelijke user interface gemakkelijk en snel de verschillende beschikbare apparaten aan kan sturen. Voor Android en .NET zijn al verschillende onderdelen standaard beschikbaar, zoals het ontdekken van services binnen een netwerk en communiceren met deze services via het SOAP protocol. Het iOS platform beschikt niet over deze onderdelen. Daarom is het erg belangrijk dat onderzocht wordt of iOS dezelfde functionaliteiten kan verzorgen als Android en .NET.



### 5.3 Oplossing

Het project is in iteraties verdeeld, omdat niet zeker is of alle functionaliteiten gerealiseerd kunnen worden en hoe lang er aan elke functionaliteit gewerkt wordt. Met deze gedachten is er een planning gemaakt met in totaal zeven iteraties. Elke iteratie wordt bij de start pas ingedeeld. Reden hiervoor is om er voor te zorgen dat er geen beloftes aan het begin van het gehele traject gemaakt worden, die uiteindelijk niet waargemaakt kunnen worden door mogelijk tijdsgebrek.

Aan het begin van elke iteratie wordt er gekeken naar wat er in de vorige iteratie gerealiseerd is. Vervolgens worden de te maken functionaliteiten opgesteld in een design document, waarbij tegelijkertijd een schema wordt gemaakt met per functionaliteit een prioriteit (High, Medium en Low) en tijdsduur. Op deze manier wordt de iteratie volledig ingevuld. Vervolgens wordt de planning met Josée en Peter besproken en zodra er een akkoord is voor de iteratie kan de realisatie van de beloofde functionaliteiten beginnen. Aan het einde van dit traject wordt de applicatie getest, of de gemaakte onderdelen naar wens werken en wordt de applicatie ook op stabiliteit getest. Indien nodig worden de functionaliteiten die door tijdsgebrek niet zijn gerealiseerd, meegenomen naar de volgende iteratie.

## 6. Onderzoek

Het belangrijkste onderdeel van de afstudeeropdracht is het uitvoeren van een onderzoek. Kennis opdoen van de beschikbare technieken is zeer belangrijk om de juiste basis te kunnen creëren om zo goed mogelijke beslissingen te kunnen nemen. In de volgende paragrafen wordt een korte samenvatting gegeven van de verschillende deelonderzoeken die zijn uitgevoerd. De volledige onderzoeksdocumenten zijn als bijlagen bij dit document toegevoegd. De onderzoeksresultaten hebben de basis gevormd voor alle beslissingen en resultaten die in dit document geschreven zijn.

### 6.1 Amigo communicatie met iOS

Om te kunnen communiceren met een Amigo apparaat moet er gebruik gemaakt worden van SOAP. Voor deze, door de informatica industrie veel gebruikte vorm van communicatie, verleent de iOS SDK echter geen standaard ondersteuning. SOAP gebruikt XML-type berichten voor de communicatie waar de iOS SDK wel ondersteuning voor biedt. Het zou echter teveel tijd kosten om zelf het wiel opnieuw uit te vinden en een eigen SOAP communicatie component te realiseren. Daarom is er voor gekozen om SOAP niet in detail te doorlichten, maar te zoeken naar al bestaande oplossingen.

De eigenschappen van een SOAP service worden beschreven in een WSDL (Web Services Description Language) bestand. Vooraf is gekeken hoe de WSDL van elk apparaat eruitziet. Amigo is zo opgezet dat elk apparaat over dezelfde functionaliteiten beschikt. De belangrijkste functies zijn hieronder uitgelicht met een korte beschrijving:

- **getDescriptions:** Geeft een bericht terug waarin onder andere beschreven wordt wat het apparaat zijn naam, locatie, type en service adres is. Verder wordt er ook een lijst van opties gegeven die via de set functie beschikbaar zijn.
- **getStatus:** Deze functie geeft de status terug van de device, zo wordt voor een groep spots per rij spots aangegeven op welke dimstand deze staan.
- **getPowerconsumption:** Deze functie geeft een getal terug wat aangeeft hoeveel Watt het apparaat op het huidige moment verbruikt.
- **set:** Via deze functie kan het apparaat aangestuurd worden. Deze functie vereist een 'code' (het apparaat dat aangestuurd moet worden, bijvoorbeeld 'A8') en een 'function' (wat moet er met dat apparaat gedaan worden, bijvoorbeeld 'On'). Met deze invoer kan het apparaat bijvoorbeeld aan- of uitgezet worden. De invoer voor de set functie is te verkrijgen uit de lijst van opties via de getDescriptions functie. De service antwoordt vervolgens met een berekend verbruik van het apparaat in Watt.

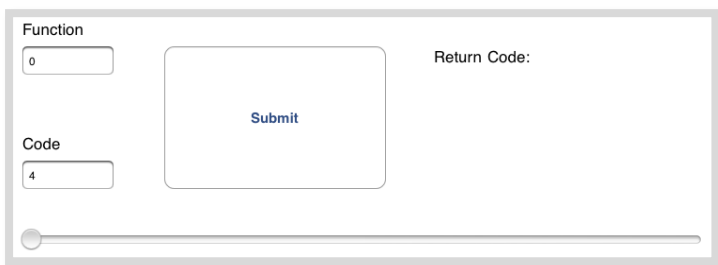
Na het onderzoeken van de Amigo apparaten is er gekeken naar bestaande oplossingen om de SOAP communicatie voor elkaar te krijgen, hier bleken maar 2 oplossingen beschikbaar te zijn:

- wsdl2objc (laatste wijziging: eind 2009)
- sudzc (laatste wijziging: eind 2010).

Sudzc maakt het mogelijk om via een webpagina een WSDL in te voeren waarmee code gegenereerd wordt om te communiceren met de service. Wsdl2objc is een desktop OSX applicatie met dezelfde functionaliteit maar genereert andere code.



Vervolgens zijn beide oplossingen getest door middel van een simpele applicatie te realiseren waarmee het doel was om één enkele Amigo service aan te sturen. Deze applicatie zou de spots in de demoruimte aan moeten sturen via zowel handmatige input als via een slider, aangezien deze spots geen aan of uit functionaliteit hebben maar alleen gedimd kunnen worden. Al snel bleek dat de



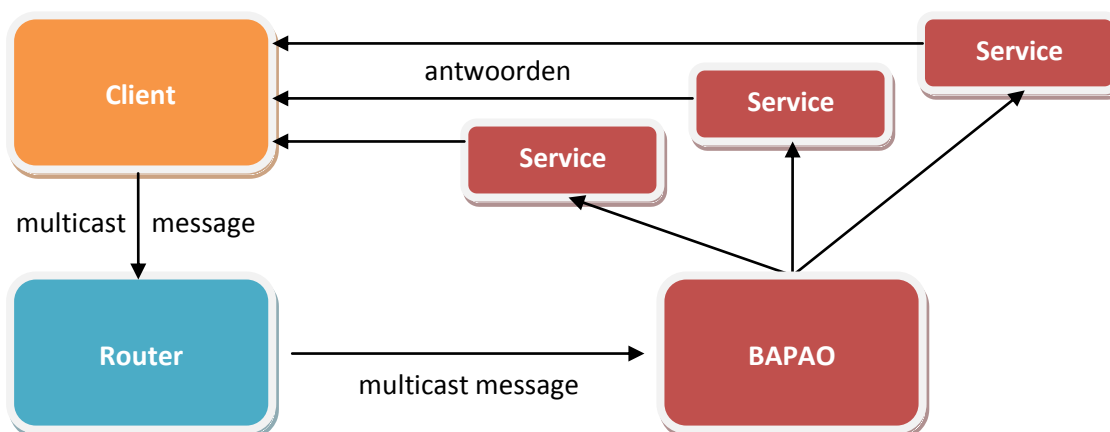
**Figuur 6.1 Amigo test applicatie**

wSDL2objc gegenereerde code niet werkte met een Amigo service, de code was ook erg onoverzichtelijk dus aanpassen zou enige tijd in beslag nemen. De code die via Sudzc.com gemaakt werd was wel goed leesbaar en werkte meteen 'out of the box', met deze kennis was de keuze snel

gemaakt en werd er gebruik gemaakt van de code via sudzc om een basis te leggen voor het project.

## 6.2 Amigo service Discovery

Elk Amigo netwerk is uniek, het heeft zijn eigen apparaten die elk een eigen service hebben met elk hun eigen mogelijkheden. Door deze samenstelling is het niet van tevoren mogelijk om te weten wat voor services beschikbaar zijn en op welk adres deze services te benaderen zijn. Om dit probleem op te lossen is er een methode ontwikkeld genaamd 'Discovery'.



**Figuur 6.2 Service Discovery**

Discovery is op papier een simpele maar effectieve manier om te ontdekken wat voor services beschikbaar zijn in een lokaal netwerk. Zoals in figuur 6.2 te zien is stuurt een applicatie een bericht naar een vast adres (multicast-adres) wat in elk netwerk hetzelfde is. Vervolgens wordt dit pakket naar elk op het netwerk aangesloten apparaat verzonden. Indien een apparaat services bevat die naar dit bericht luisteren, wordt er een bericht teruggestuurd waarin het adres van de service in vermeldt staat. Om er voor te zorgen dat de applicatie een up-to-date lijst heeft van alle verschillende apparaten zijn er een aantal mogelijkheden onderzocht, in de volgende paragrafen volgt een lijst van deze mogelijkheden met elk een korte uitleg.

- **DISCO** Deze verouderde techniek vereist dat de op de server waar een service of meerdere services actief op zijn een bestand geplaatst wordt. Wanneer dit bestand door een andere applicatie wordt opgevraagd krijgt de applicatie een in XML geformatteerde lijst terug waarin alle actieve services staan. Deze techniek bestaat echter sinds 2002 en is sterk verouderd zonder updates. Daarnaast zou deze vorm van Discovery op twee verschillende machines ingesteld moeten worden. Ook alle bestaande services moeten aangepast worden.
- **.NET WS-Discovery** Voor .NET heeft Microsoft zelf een oplossing die services kan ontdekken, deze oplossing bestaat uit een bericht die naar het multicast-adres wordt gestuurd, waarop vervolgens antwoorden van services naar toe komen waardoor een lijst opgesteld wordt met alle services die hebben geantwoord.
- **Java WS-Discovery** De oplossing voor Java zit een stuk complexer in elkaar. Uiteindelijk gebeurt er hetzelfde als bij de .NET versie, maar het vereist meer configuratie voordat Discovery mogelijk is. Tijdens het testen van deze mogelijkheid is het niet gelukt om het voorbeeld werkend te krijgen, met deze reden is deze optie daarbij afgevallen.
- **Interoperabilityservice** Deze Amigo service gebruikt de .NET service om een XML lijst op een website beschikbaar te stellen waarin alle beschikbare services uit op te vragen zijn. Deze service is echter onstabiel, de lijst met beschikbare services blijft beschikbaar te allen tijde maar wordt na een onbepaalde tijd (dit kunnen uren maar ook dagen zijn) niet meer bijgehouden omdat de applicatie om onbekende reden gestopt is met werken.
- **Bonjour** Deze door Apple ontwikkelde Discovery-techniek is in zowel .NET als Java als Objective-C goed ondersteund en is, in werking, vergelijkbaar met de Discovery oplossing van Microsoft. Met een Bonjour service communiceren via iOS is zeer gemakkelijk. Er zijn geen grote hoeveelheden code nodig om zowel een client als een service applicatie te realiseren. De beschikbare libraries bieden een snelle realisatie van een oplossing.
- **Objective-C Discovery** De bovengenoemde oplossingen maken de Discovery-techniek voor de ontwikkelaar gemakkelijk. De low-level code die nodig is om Discovery voor elkaar te krijgen is geen zorg voor de ontwikkelaar. Het is wel mogelijk om in C in combinatie met Objective-C zelf de low-level code te schrijven om Discovery voor elkaar te krijgen, echter er is hier geen ervaring mee en er is weinig documentatie beschikbaar over de daadwerkelijke werking van de Discovery methode die Microsoft ontwikkeld heeft.

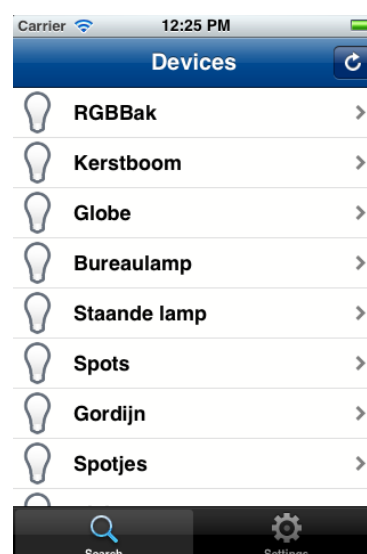
Voor iOS is geen standaard oplossing beschikbaar, het zou te veel tijd kosten om zelf de low-level code te schrijven die nodig is om een betrouwbare oplossing te kunnen maken. Daarom is er voor gekozen om een service te schrijven die zelf Bonjour gebruikt om zichzelf vindbaar te maken in het netwerk. Deze service gebruikt vervolgens de Discovery techniek van Microsoft in .Net om een lijst aan te kunnen bieden waarin alle beschikbare services in staan. In hoofdstuk 7.3: Bonjour Amigo is meer informatie te vinden over de service en de werking.

## 7. Eindproducten

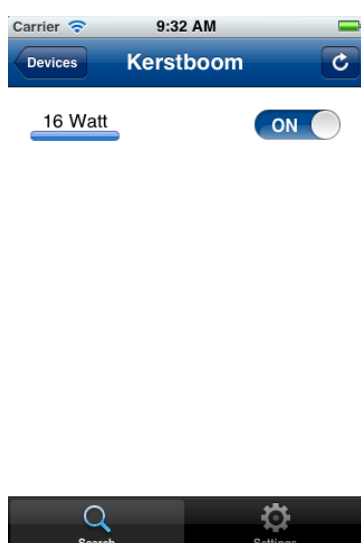
De ontwikkeling van de Amigo Home Console is in vier stappen verdeeld. De eerste iteratie bevatte alleen de basisfunctionaleiten (het tonen en besturen van alle verschillende beschikbare Amigo apparaten), vandaar ook de naam: Basic Amigo Communicator. In de daarop volgende iteratie werd het ontwerp voor de Amigo Home Console gemaakt en de eerste versie gerealiseerd. In de derde iteratie werd de applicatie sterk verbeterd door middel van user tests te verwerken en een plattegrond waarop de gebruiker kan zien waar beschikbare apparaten staan. De laatste iteratie zorgde voor de connectie met de Bonjour Amigo service voor een betrouwbare lijst van apparaten en werd de ontwikkeling van de applicatie afgerond.

### 7.1 Basic Amigo Communicator

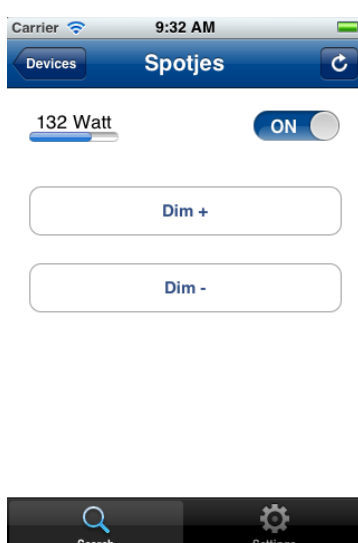
De Basic Amigo Communicator is gemaakt met het doel om aan te tonen dat iOS in staat is om meerdere Amigo apparaten in een netwerk te tonen in een lijst met hun status (aan of uit). Vervolgens kan de gebruiker een apparaat uit de lijst kiezen en deze aansturen met de gegeven besturing. De besturing is voor elk van de drie beschikbare type apparaten verschillend om de gebruiker niet te verwarren met extra keuzes. Een slider apparaat kan alleen via een slider tool aangestuurd worden. Een switch apparaat kan alleen aan of uit gezet worden en een dim apparaat kan zowel aan, uit, omhoog als omlaag gedimd worden. Naast een iPad versie is de Basic Amigo Communicator ook op de iPhone beschikbaar om aan te tonen dat de applicatie ook gemakkelijk voor iPhone gemaakt kan worden.



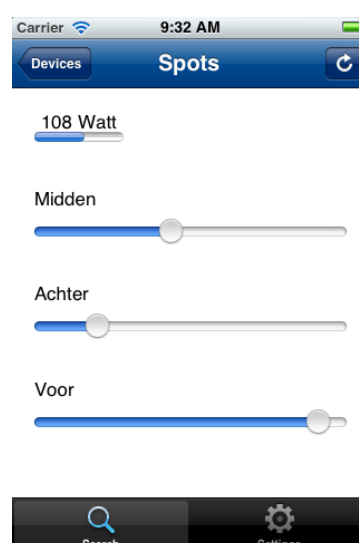
Figuur 7.1 Basic Amigo Communicator



Figuur 7.2 Switch controls



Figuur 7.3 Dim controls



Figuur 7.4 Slider controls

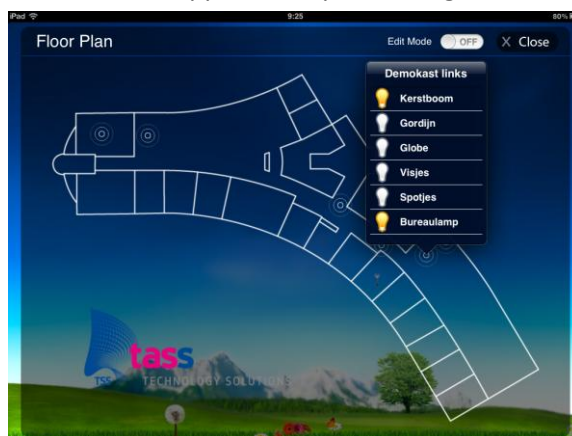
## 7.2 Amigo Home Console



Figuur 7.2 De Amigo Home Console

opgedaan worden in de vorm van een zelf in te stellen Twitter-feed, weerbericht op locatie en nieuws-feed. Indien gewenst kan de gebruiker er voor kiezen om de iPad van de muur af te halen en mee te nemen om overal in het huis de apparaten aan sturen.

De hoofdfunctie van de applicatie blijft echter het aansturen van apparaten op het Amigo netwerk, daarom is er in het hoofdscherm meer ruimte gereserveerd voor de Amigo specifieke functionaliteiten en zijn de extra informatie-velden kleiner. Verder is er ook een plattegrond-optie gemaakt, waarbij het mogelijk is om te zien waar de Amigo apparaten in een gebouw staan. De gebruiker kan een locatie selecteren waarbij een lijst wordt getoond met de beschikbare apparaten in de betreffende ruimte. Vervolgens kan een apparaat geselecteerd en aangestuurd worden met de gegeven besturing.



Figuur 7.3 Plattegrond functionaliteit

## 7.3 Bonjour Amigo

Uit onderzoeken is duidelijk geworden dat Discovery via de iOS SDK te veel tijd in beslag zou nemen om stabiel te krijgen. Met deze kennis is er besloten om een Bonjour service te realiseren in .NET die zelf de Discovery uitvoert. De resultaten komen in een tabel van 2 kolommen te staan waar het adres en type van elke ontdekte service te vinden is. Deze tabel kan vervolgens door een client applicatie opgevraagd worden en kan er geluisterd worden of de tabel wordt geupdate zodat de applicatie daar niet constant om hoeft te vragen. Door Bonjour te gebruiken kan via de iOS SDK gemakkelijk en snel een verbinding gemaakt worden zonder een adres te moeten weten. Het enige wat de client applicatie moet weten is de naam van de service waarmee een verbinding gemaakt moet worden, waardoor er meer tijd is om de service te perfectioneren en de connectie van de service in de applicatie goed te verzorgen.

## 8. Conclusies en aanbevelingen

De conclusies en aanbevelingen hebben zowel betrekking op het proces als op het product zelf. Om tot het gewenste eindresultaat te komen, wat aan het begin van het project nog niet duidelijk was is voorafgaand wel een duidelijke projectaanpak vastgesteld. De keuze om het project iteratief aan te pakken heeft goed uitgepakt, elke iteratie stond in het teken van een specifieke hoofdfeature met daarbij een aantal kleinere features. Zodoende groeide de uiteindelijke applicatie met één grote functionaliteit en wat kleinere onderdelen zodat het deelproduct aanvoelt als een eindproduct.

Door de iteratieve aanpak is het ook mogelijk geweest om alle eisen aan de applicatie ook daadwerkelijk te realiseren. Aan het einde van een iteratie moeten alle beloofde functionaliteiten af zijn, zodoende word je gedwongen te werken naar een eindresultaat en niet constant bij een feature te blijven hangen om deze tot in detail te perfectioneren.

Tijdens het uitvoeren van het onderzoek bleek al snel dat het ontdekken van apparaten (Discovery) een lastige klus zou worden waarvoor werd gewaarschuwd bij de aanvang van het project. Door het gerichte vooronderzoek is er veel tijd bespaard om hier niet in het begin te veel tijd in te steken. Uiteindelijk is er besloten een externe service te maken die de Discovery uitvoert, wat de meest betrouwbare oplossing was. In de toekomst kunnen andere applicaties ook van deze service gebruikmaken omdat de gebruikte techniek een open standaard is.

Onstabieleit is een begrip dat te vaak naar voren kwam tijdens de duur van het project, hiermee worden niet de gerealiseerde producten bedoeld maar de huidige situatie. Van tijd tot tijd reageert het netwerk sloom en het gebeurde ook vaak dat apparaten niet reageerden of dat niet kritieke services te veel geheugen in beslag namen. Dit geeft aan dat het ongeveer vijf jaar oude Amigo netwerk aan een revisie toe is. De huidig draaiende services zouden kritisch bekeken moeten worden en waar nodig herschreven of op zijn minst goed getest worden. Waar nodig zouden reparaties uitgevoerd moeten worden.

Mede door de goede begeleiding en beschikbare informatie over het Amigo netwerk is het project uiteindelijk tot een succesvol einde gekomen. Er is veel kennis op gedaan op zowel procesmatig als technisch vlak. Hierdoor zijn de eindproducten succesvol gerealiseerd en kan de afstudeerperiode als succesvol worden bestempeld.



## Evaluatie

Samen met Josée Zielman en Peter Vink hebben we in het begin besproken wat de Amigo iPad applicatie ongeveer moest gaan doen. In het begin wist ik zelf nog niets van Amigo af, maar het klonk zeer interessant en innovatief. De opdracht bleef tijdens het gehele project erg vrij, ik kon de Home Management Console opdracht goed invullen met zelf bedachte onderdelen. De focus lag, vooral in de eerste helft van het traject, op het werkend krijgen van een stabiele applicatie die de main feature (het aansturen van beschikbare apparaten op het netwerk) kon uitvoeren. Dat ik erg vrij werd gelaten bij de invulling van mijn opdracht vond ik erg prettig. Zodoende kon ik (zolang Josée en Peter het er mee eens waren) mijn eigen beeld dat ik had bij een Home Console Management applicatie goed uitwerken.

Op zowel technisch als procesmatig vlak heb ik veel geleerd. Ter voorbereiding op mijn stage heb ik een applicatie voor de iPhone ontwikkeld voor een klant via mijn eigen bedrijf. Ik heb hier bewust voor gekozen om er voor te zorgen dat de taal (Objective-C) en tools (Xcode) geen obstakel zouden vormen voor mijn werkzaamheden. Maar ook omdat ik uit ervaring weet dat er een groot gat zit qua kennis tussen het maken van simpele demo applicaties tegenover een compleet product maken. Vaak kom je problemen tegen die je niet vooraf had kunnen bedenken. Procesmatig heb ik veel gehad aan mijn opleiding, want aan het einde van de studie wordt er vanuit gegaan dat je weet hoe je bijvoorbeeld een requirements document en PID moet opstellen. Omdat dit zo vaak is herhaald in verschillende vakken en minors was dit geen probleem. Verder was ik ook erg blij met de Agile manier van werken bij TASS, het kwam op hetzelfde neer met wat ik op dit moment voor projecten binnen mijn eigen bedrijf gebruik. Aan het begin van een iteratie wordt er een design en productlog opgezet met prioriteiten en tijdsduur per functionaliteit. Dit werkt erg handig naar de klant (Josée en Peter) toe omdat het voor hun duidelijk is wat je gaat maken, waar de risico's liggen en wat zeker is.

Het Tien Stappen Plan (TSP) heeft ook procesmatig geholpen bij dit project, nadat mijn onderzoek in de beginfase van het project was afgerond had het Amigo netwerk geen geheimen meer voor mij. Ik kon me volledig richten op het ontwikkelen van een goede applicatie die later als demoproduct gebruikt kan worden bij TASS. De iteratieve natuur van het project was goed te verwoorden naar het TSP, de Onderzoeks- en Oplossingsfase. In elke iteratie moest er een werkplanning en organisatie van de iteratie gemaakt worden. Vervolgens werd er, wanneer dit nodig was, een onderzoek uitgevoerd voor bepaalde functionaliteit om een goede keuze te kunnen maken. Aan het eind stond er een product dat weer gegroeid was in functionaliteit en meteen in gebruik kon worden genomen.

Terugkijkend op de afstudeerperiode kan ik niets anders zeggen dan dat ik het zeer naar mijn zin heb gehad. Het gehele project is vlekkeloos verlopen en de gewenste eisen aan de oplossing zijn allemaal gerealiseerd. Ik ben er dan ook erg blij mee dat ik TASS gekozen heb als plek om af te studeren. Ik had zelf geen betere plek kunnen wensen.

---



## Literatuurlijst

### **Amigo project**

Dr. Maddy D. Janse

<http://www.Hitech-projects.com/euprojects/amigo/index.htm>

### **Amigo tutorial, Using Amigo Security Services**

Todor Dimitrov, Edwin Naroska, Jörg Schmalenstroer & Volker Leutnant

[https://gforge.inria.fr/frs/download.php/3255/Web\\_Services-Tutorial.pdf](https://gforge.inria.fr/frs/download.php/3255/Web_Services-Tutorial.pdf)

### **Connected Homes, Amigo Interoperability**

Tim Wijnen & Ruud Ghielen

### **Web Services Discovery**

Wikipedia

[http://en.wikipedia.org/wiki/Web\\_Services\\_Discovery](http://en.wikipedia.org/wiki/Web_Services_Discovery)

### **SOAP**

Wikipedia

<http://en.wikipedia.org/wiki/SOAP>

### **wsdl2objc**

Google Code

<http://code.google.com/p/wsdl2objc/>

### **Sudzc**

Google Code

<http://code.google.com/p/sudzc/>

### **Bonjour**

Apple

<http://www.apple.com/support/bonjour/>

---





## Bijlagen

---

## **Bijlage A: Project Management Plan**

## **Bijlage B: Software Requirements Plan**

## Bijlage C: Software Test Plan

---

## **Bijlage D: Amigo Consortium**



## **Bijlage E: Onderzoek Amigo Besturing**

## **Bijlage F: Onderzoek Amigo Services**

## **Bijlage G: Onderzoek Amigo Discovery**

## **Bijlage H: Onderzoek Bonjour Amigo**