**V.E.D.S. Group**

**Innovatieve oplossing voor toenemende dementie**

**Software voor een indoor lokalisatiesysteem**

**Tom Brekelmans**

**12 juni 2012, Eindhoven**

##### Gegevens student:

Naam: T.W. Brekelmans

Studentnummer: 2120946

Opleiding: HBO-ICT Voltijd

Afstudeerperiode: van 13/02/2012 t/m 29/06/2012

##### Gegevens bedrijf:

Naam: V.E.D.S. Group

Afdeling: Design

Plaats: Eindhoven

Bedrijfsbegeleider: Ir. B. van den Dolder / stagebegeleider

##### Gegevens docentbegeleider:

Naam: Dr. M.J.M. Lamers

##### Gegevens verslag:

Titel afstudeerverslag: Innovatieve oplossing voor toenemende dementie

Datum uitgifte afstudeerverslag : 12/06/2012

# Voorwoord

Deze afstudeerscriptie schrijf ik voor het afstuderen bij de opleiding HBO-ICT van Fontys Hogescholen te Eindhoven. De afstudeerstage is uitgevoerd bij V.E.D.S. Group en omvat het ontwikkelen van een embedded softwaresysteem. Ik wil graag Martijn Lamers, Ben van den Dolder en Dennis Wokke bedanken voor de begeleiding gedurende mijn stage.

Inhoudsopgave

[Samenvatting 5](#_Toc327193799)

[Summary 6](#_Toc327193800)

[Verklarende woordenlijst 7](#_Toc327193801)

[1. Inleiding 8](#_Toc327193802)

[2. Het bedrijf 9](#_Toc327193803)

[3. De opdracht 12](#_Toc327193805)

[4. Onderzoeksfase 14](#_Toc327193809)

[4.1 Initiatiefase 14](#_Toc327193810)

[4.2 Definitiefase 15](#_Toc327193813)

[4.3 Ontwerpfase 15](#_Toc327193814)

[5. Oplossingsfase 17](#_Toc327193819)

[5.1 Realisatiefase 18](#_Toc327193820)

[5.2 Validatiefase 19](#_Toc327193824)

[5.3 Uitbreiding project 20](#_Toc327193825)

[6. Conclusies en aanbevelingen 21](#_Toc327193826)

[Evaluatie 23](#_Toc327193827)

[Literatuurlijst 24](#_Toc327193828)

[Bijlagen 25](#_Toc327193831)

[Bijlage I PID](#_Toc327193832)

[Bijlage II Onderzoeksdocument](#_Toc327193833)

[Bijlage III Ontwerpdocument](#_Toc327193834)

# Samenvatting

V.E.D.S. Group is een ingenieursbedrijf dat gespecialiseerd is in het ontwikkelen van elektronische systemen. Het bedrijf heeft ervoor gekozen om samen met haar partnerbedrijf Delite Group een nieuw systeem te ontwikkelen waarmee personen binnen een gebouw gelokaliseerd kunnen worden. De achterliggende reden hiervoor is dat er door de vergrijzing en de hiermee samenhangende toenemende dementie steeds meer innovatieve maatregelen nodig zijn om ermee om te gaan. Met het systeem kan binnen een zorginstelling de zorg voor dementerenden efficiënter en effectiever worden gedaan.

Het systeem bestaat uit verschillende deelsystemen, waaronder een transponder die gedragen wordt door een persoon, een baken die in elke kamer van het gebouw hangt en een basisstation voor coördinatie waarvan er één aanwezig is binnen het gebouw. De opdracht bestaat uit het realiseren van de software voor het basisstation. Hiervoor moet een onderzoek gedaan worden naar de beste manier van communicatie tussen de verschillende deelsystemen.

De opdracht is uitgevoerd volgens de standaardaanpak van V.E.D.S. Group waarbij verschillende fases na elkaar worden afgerond, namelijk specificatie, ontwerp, realisatie en validatie.

Gedurende de uitvoering van het project bleek tijd over te zijn om het project uit te breiden met het realiseren van de software voor de bakens. Van het basisstation kon hierbij het globale ontwerp wordt overgenomen, evenals een deel van de code.

Verder kwam bij het uitvoeren van het project naar voren dat er weinig controle is op de softwarekwaliteit binnen het bedrijf. Er bleek dat herbruikbaarheid en beheerbaarheid van software moeilijk is doordat er in het ontwerp ervan te weinig rekening mee wordt gehouden.

Om deze reden wordt aanbevolen om bij het ontwerpen van software binnen het bedrijf meer te focussen op het scheiden van de software in verschillende onderdelen die hergebruikt kunnen worden. Hiermee wordt er in de toekomst tijd bespaard doordat programmacode niet opnieuw geschreven hoeft te worden. Verder kan de software gemakkelijker aangepast worden. Ook zouden er code reviews ingevoerd moeten worden, om meer controle te krijgen op de softwarekwaliteit.

Met de gerealiseerde software kan het gehele systeem getest worden en kan er aangetoond worden dat het principe van het systeem werkt.

# Summary

V.E.D.S. Group is an engineering company that specializes in developing electronic systems. Together with its partner Delite Group, the company has chosen to develop a new system with which persons can be located within a building. The motivation for this is that innovative measures are increasingly needed to deal with the growth in the amount of people with dementia. The system can be used in a healthcare facility to improve the efficiency and effectiveness of the care for demented elderly.

The system consists of several subsystems, including a transponder worn by a person, a beacon placed in every room of the building and one base unit used for coordination within the building. In this project the software of the base unit will be realized. To determine the best method of communication between the different subsystems, research will be done.

The standard approach of V.E.D.S. Group is used in which different phases are completed one after another, namely specification, design, implementation, and validation.

Enough time was left to add the realization of the software for the beacons to the project. Most of the design of the base unit could be adopted as well as part of the code.

During the execution of the project it seemed that there is little control on the software quality within the company. While checking old software it appeared that reusability and manageability of software is difficult because it is not taken into account during the design phase.

For this reason it is recommended that software design within the company should have more focus on manageability and reusability. This will save time in the future. Furthermore, code reviews should be introduced in order to have more control on software quality.

With the resulting software, the whole system can be tested and a proof of concept can be made.

# Verklarende woordenlijst

|  |  |
| --- | --- |
| *Term* | *Definitie* |
| Basisstation | Centraal deelsysteem van het lokaliseringssysteem die coördineert en alle data verzamelt. |
| Beacon | Het apparaat dat binnen elke kamer hangt om een ultrasone code uit te zenden naar transponders binnen die kamer. |
| Dementie | Een verzamelnaam voor aandoeningen die gekenmerkt worden door combinaties van meervoudige stoornissen in verstandelijke vermogens (waaronder het geheugen), stemming en gedrag. |
| Ethernet software | De software die verantwoordelijk is voor communicatie met andere systemen via een ethernet kabel. |
| HTTP | Hypertext Transfer Protocol is het protocol dat gebruikt wordt voor communicatie een webclient en een webserver, bijvoorbeeld voor het bezoeken van een website via een browser. |
| IAR Embedded Workbench | Een Integrated Development Environment van IAR Systems die gebruikt wordt voor de ontwikkeling van embedded systemen met de programmeertalen C en C++. |
| Integrated Development Environment | Computersoftware die een softwareontwikkelaar ondersteunt in het ontwikkelen van software. |
| LED | Light Emitting Diode. Een klein controlelichtje dat bijvoorbeeld gebruikt kan worden om aan te geven of een functie van het systeem is geactiveerd of niet. |
| Microcontroller | Processor die wordt gebruikt om elektronische apparaten te besturen. |
| Printed Circuit Board | Een printplaat waarop verschillende hardware componenten geplaatst zijn en met elkaar verbonden worden. Wordt gebruikt in vrijwel alle elektronische systemen. |
| Separation of concerns | Een softwareoplossing opdelen in verschillende onderdelen die zo min mogelijk overlap hebben. |
| Seriële poort | Een communicatieverbinding waarbij bits één voor één worden verstuurd. |
| TCP/IP Framework | Een softwareframework dat de TCP netwerklaag, inclusief alle onderliggende lagen, implementeert en dus gebruikt kan worden voor communicatie met andere systemen. |
| Transponder | Zender en ontvanger dat gedragen wordt door personen voor locatie bepaling. |
| Ultrasoon | Geluid met een frequentie die zo hoog is dat het niet hoorbaar is voor mensen. |
| Zigbee | Een draadloos communicatieprotocol bedoeld voor low-power systemen. |

# Inleiding

Het aantal dementerende Nederlanders zal binnen vier decennia verdubbelen tot een half miljoen. De kosten voor de verzorging van deze mensen neemt in dezelfde mate toe en zal oplopen tot 15 miljard euro per jaar (bron 5.). Om deze reden zijn er steeds meer innovatieve maatregelen nodig om hiermee om te gaan.

V.E.D.S. Group, een bedrijf gespecialiseerd in het ontwerpen van elektronische systemen, speelt hierop in en ontwikkelt een nieuw product waarmee personen binnen een gebouw gelokaliseerd kunnen worden. Dit product kan worden gebruikt binnen een zorginstelling om de zorg voor dementerende ouderen te verbeteren en de werkdruk van de zorgmedewerkers te verlagen. Het product wordt ontwikkeld samen met het partnerbedrijf Delite Group. Het product bestaat uit verschillende deelsystemen en gebruikt als kernprincipe ultrasoon geluid. Er wordt een projectmatige aanpak gebruikt waarin verschillende fases na elkaar worden afgerond. Het doel van het project is om de software van één deelsysteem te realiseren. Hierbij wordt er een ook onderzoek gedaan naar de beste communicatiemogelijkheden tussen de verschillende deelsystemen. De resultaten van het onderzoek wordt als input gebruikt voor het ontwerp van de software.

In hoofdstuk 2 wordt meer informatie gegeven over het bedrijf V.E.D.S. Group. Hoofdstuk 3 geeft alle details over de opdracht. De vervolg hoofdstukken (4 t/m 9) beschrijven per projectfase hoe deze is verlopen, wat er gedaan is en waarom, en welke problemen zijn opgedoken.

# Het bedrijf

Veeren Electronic Design Solutions (V.E.D.S.) Group is een ingenieursbedrijf dat aan productontwikkeling van elektronische systemen doet. Het bedrijf is gespecialiseerd in Printed Circuit Boards (PCB’s). Er worden hiervan alleen kleine series en prototypen geproduceerd. Het bedrijf is opgericht in 1994 en heeft 30 medewerkers. Het management van V.E.D.S. is zelf al sinds 1989 actief in de markt van PCB ontwerp. V.E.D.S. Group heeft een huidige jaaromzet van ongeveer 3 miljoen. Per jaar worden er circa 250 PCB’s ontworpen. V.E.D.S. is sinds 2007 gevestigd in het Bètagebouw op de High Tech Campus in Eindhoven (afgebeeld in figuur 1).



Figuur 1 - Bètagebouw

Een voorbeeld van een product ontworpen door V.E.D.S. is de betaalterminal, te zien in figuur 2.



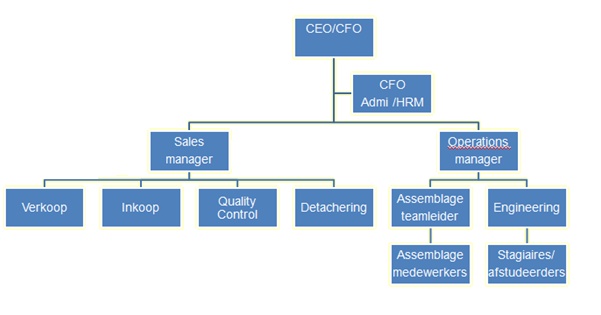
Figuur 2 – Betaalterminal

De missie van V.E.D.S. is om een organisatie neer te zetten die vanuit een heldere klantfocus en een gedegen technische kennis, hun klanten op betrouwbare wijze kwalitatief hoogstaande producten en diensten levert. Het motto van V.E.D.S. is: “Wij kunnen alles voor de klant”. Hiermee worden de waarden benadrukt die voor het bedrijf belangrijk zijn, hoe met de klanten omgegaan dient te worden en hoe intern gewerkt dient te worden.

V.E.D.S. Group werkt door middel van gefaseerd projectmanagement. Dit wil zeggen dat orders projectnamen krijgen, in fases worden gesplitst en verdeeld worden over het personeel. De projecten worden vervolgens op een whiteboard geschreven zodat het personeel kan zien wat de status van elk project is. Aan het begin van elk project wordt een schatting van het aantal benodigde uren gemaakt zodat daarmee een inschatting van de prijs kan worden gemaakt. Tijdens het project moet het personeel dan hun uren bijhouden zodat er achteraf gekeken kan worden hoeveel uren er daadwerkelijk aan een project besteed zijn.

## Organisatie

V.E.D.S. Group heeft een vrij platte organisatie. Het management team bestaat uit vier personen. Verder heeft het bedrijf zijn Operations (Design, Engineering en Productie) en Sales gescheiden met elk een verantwoordelijke manager (zie figuur 3). Onder de sales manager zijn de inkoper, verkoper en de personen voor Quality Control ondergebracht, alsmede de mensen die gedetacheerd zijn. Engineering bestaat uit personeel dat systemen ontwerpt. V.E.D.S. Group onderscheidt Engineering in drie verschillende disciplines: Hardware, Software en Mechanics.



Figuur 3 – V.E.D.S. organisatiestructuur

# De opdracht

# Achtergrond en beginsituatie

Tegenwoordig is er steeds meer druk op de gezondheidszorg. Door de vergrijzing is er meer vraag naar zorg. Het aantal mensen met een chronische beperking neemt toe. Een voorbeeld hiervan is dementie. De toename hiervan gaat zo snel dat in 2050 een verdubbeling van het aantal dementerenden verwacht wordt ten opzichte van nu. Verder is de invulling van de zorgvraag steeds meer gericht op een langere zelfstandigheid van de zorgconsument. Hiervoor zijn de volgende redenen. Arbeidstekort komt vaker voor en de kosten in de zorg stijgen. Nu al wordt verwacht dat de kosten bij een gelijkblijvend zorgniveau op termijn niet meer op te brengen zijn. Daarom worden door de regering nu al flinke bezuinigingen doorgevoerd en nog meer bezuinigingen zijn aangekondigd. Eén van de gevolgen hiervan is dat de zorg steeds efficiënter en effectiever ingezet moet worden.

De vraag naar nieuwe producten die bijdragen aan oplossingen voor mensen met een chronische beperking groeit. Producten die helpen om te gaan met het arbeidstekort in de zorg en voor een kostenreductie zorgen zijn daarnaast ook steeds meer noodzakelijk.

Om deze reden heeft V.E.D.S. Group ervoor gekozen om een nieuw eigen product te ontwikkelen, namelijk een lokaliseringssysteem voor personen binnen een gebouw. Met dit systeem wordt ten alle tijden bekend in welke ruimte een persoon zich bevindt in het gebouw. Verder zijn er mogelijkheden voor functies als persoonlijke alarmen. Dit product zou bijvoorbeeld gebruikt kunnen worden binnen een zorginstelling om de verzorging van dementerende ouderen te verbeteren. De werkdruk van de zorgmedewerkers kan hiermee verlaagd worden.

Het product bestaat uit drie types fysieke systemen, namelijk de transponder, het baken en het basisstation. De personen binnen het gebouw dragen de transponder zodat ze altijd gelokaliseerd kunnen worden. In alle ruimtes van het gebouw hangt een baken die een signaal uitzendt naar de transponders in die ruimte. Het basisstation is het centrale systeem in het gebouw, dat alle gegevens verzamelt en beschikbaar stelt aan een externe online server. Deze server zal vervolgens bijvoorbeeld via smartphones zorgmedewerkers up to date houden over waar de personen zich bevinden.

In eerdere projecten binnen V.E.D.S. is al onderzoek gedaan naar de technieken en mogelijkheden van zo’n systeem. Hieronder valt bijvoorbeeld het gebruik van een ultrasoon signaal dat de bakens uitzenden naar de transponders. Verder is bepaald dat het draadloze communicatieprotocol Zigbee wordt gebruikt voor de communicatie tussen de systemen.

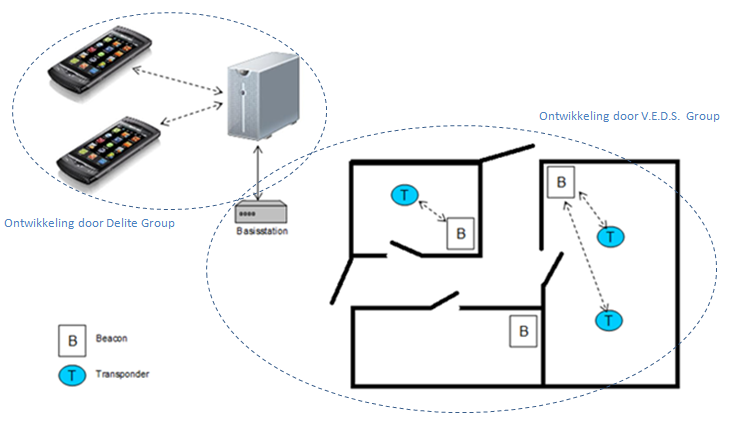
# Doel van het project

Het afstudeerproject is een deelproject van het Sanana-project binnen V.E.D.S. Sanana omvat de realisatie van het lokaliseringssysteem, inclusief alle deelsystemen. Daarnaast wordt er een pilot binnen een zorginstelling en een thuiszorgsituatie gedaan om aan te tonen dat het systeem werkt.

De doelstelling van het huidige deelproject is het ontwikkelen van de software voor het basisstation, het centrale gedeelte, van het systeem. Dit is een onderdeel van het gehele lokaliseringssysteem. Met het gebruik van het lokaliseringssysteem kan de zorg binnen bijvoorbeeld een zorginstelling of een thuiszorgsituatie verbeterd worden. Lokatiegegevens en persoonlijke alarmen kunnen beschikbaar gesteld worden aan zorgmedewerkers. Met deze informatie kunnen zij efficiënter en effectiever werken. Als een patiënt bijvoorbeeld hulp nodig heeft kan hij of zij een alarm activeren. De zorgmedewerkers kunnen dan direct reageren, omdat ze weten waar de patiënt is.

# Opdrachtomschrijving

Op basis van de beginsituatie en het doel van het project is de precieze opdracht bepaald. De opdracht is het specificeren, ontwerpen, realiseren en testen van de software die nodig is voor het basisstation van het systeem. Het basisstation is het centrale communicatiesysteem. Om deze reden dient er onderzocht te worden op welke wijze de communicatie van het lokaliseringssysteem het beste kan verlopen. De transponders en de bakens inclusief de hardware hiervan alsmede de hardware van het basisstation worden binnen V.E.D.S. Group ook ontwikkeld. Er moet samengewerkt worden tussen software-ontwikkelaars onderling, en tussen de ontwikkelaars van de hardware en de software om een goede afstemming te krijgen. De ontwikkeling van de externe online server inclusief de smartphone app’s zal tegelijkertijd ontwikkeld worden door het partnerbedrijf Delite Group. Met dit bedrijf wordt ook samengewerkt om het basisstation af te stemmen met de server die ontwikkeld wordt door deze partner. In figuur 4 is te zien hoe de ontwikkeling van de systemen is verdeeld over de organisaties.



Figuur 4- Takenverdeling op organisatieniveau

# Initiatiefase

### 4.1 Opstart

Om gedurende de stageperiode goed te kunnen werken is het nodig dat je de organisatie goed kent, wat er gedaan wordt en hoe dit gedaan wordt. Hiermee is de stage begonnen. Hieronder valt een presentatie van de bedrijfsleider over het bedrijfsproces met betrekking tot productontwikkeling. Verdere informatie is via gesprekken en documenten opgedaan, onder andere het bestuderen van de bedrijfstemplates voor ontwikkelingsdocumenten. Na bekend te zijn met de organisatie volgde het inlezen in het Sanana-project en het informeren over diverse aspecten van het project bij de betrokkenen, om een beeld te krijgen van de achtergrond, de beginsituatie en het doel. Ook dit is gedaan door middel van gesprekken en het bestuderen van documenten. Vervolgens is verdiept in de opdracht om een preciezer beeld te krijgen. Hieronder viel het achterhalen van het concept en de verdere werking van het te bouwen lokaliseringssysteem.

Gedurende de initiatiefase is er een wijziging in het systeem gekomen die een grote impact gehad heeft op de rest van de stageperiode. In eerste instantie was namelijk het idee dat het basisstation een standaard PC zou worden, waarvan er zich dan één zou bevinden binnen bijvoorbeeld een zorginstelling. Maar er werd gaandeweg besloten om voor het basisstation eigen custom hardware te gaan ontwikkelen, in plaats van een standaard PC te gebruiken. De reden hiervoor was, dat het systeem ook gebruikt moest kunnen worden in een thuiszorgsituatie en daar ook een basisstation aanwezig moest zijn. Hierbij is het niet gewenst om in de thuiszorgsituatie 24 uur per dag een PC aan te hebben staan. Met de custom hardware wordt de software die erop draait embedded, wat een grote impact heeft op de software. Het grootste verschil is dat er geen operating system op draait en dat er minder geheugen beschikbaar is.

### 4.2 Aanpak

Nadat de verdieping van de opdracht had plaatsgevonden, kon de aanpak worden bepaald. De aanpak die gekozen is is de standaardaanpak voor productontwikkeling met meerdere disciplines, namelijk hardware, software en mechanica. Deze binnen V.E.D.S. Group gebruikelijke aanpak is gebaseerd op het V-model. De aanpak bestaat uit verschillende fases die na elkaar worden uitgevoerd, namelijk definitie, ontwerp, realisatie en validatie. Omdat er veel afhankelijkheden met de overige deelsystemen zijn is het belangrijk om deze aanpak te volgen. Er is namelijk veel afstemming nodig in het ontwerp van de verschillende deelsystemen. Pas als dit is afgerond kan de realisatie starten.

Het resultaat van de initiatiefase is het Project Initiation Document (PID). Onder andere de aanpak en globale planning wordt daarin beschreven. Het PID is te vinden in bijlage 1.

# Definitiefase

Na de initiatiefase konden de requirements van het systeem bepaald worden. Binnen de stuurgroep (inclusief de opdrachtgever) van het Sanana-project werden de keuzes gemaakt over de requirements. De teamleider (verantwoordelijk voor de technische realisatie) heeft een algemeen systeem requirements document opgesteld. Deze requirements hebben betrekking op het gehele systeem, dus inclusief alle deelsystemen. Uit dit document konden de software requirements worden afgeleid, die relevant zijn voor het basisstation. Omdat de requirements betrekking hadden op alle disciplines van het systeem (hardware, software, mechanica) en op alle deelsystemen, waren sommige softwarerequirements nog relatief vaag. Al deze vaagheden zijn weggewerkt door hierover vragen te stellen en te overleggen met de teamleider.

Het resultaat van de definitiefase was het requirements document. Zie hiervoor bijlage II.

# Ontwerpfase

Nadat de software requirements bekend waren kon er met het ontwerp worden gestart. Dit was de belangrijkste fase van het project, omdat er een grote samenhang was met andere projectonderdelen.

De ontwerpfase begon met het verkrijgen van een globaal idee over welke informatie er tussen de systemen uitgewisseld moest worden aan de hand van de requirements. Dit is gezamenlijk met de ontwikkelaars van de andere systemen gedaan. Deze informatie was nodig voor het uitvoeren van het onderzoek om te bepalen op wat voor manier de communicatie het beste kan verlopen.

### 6.1 Onderzoek

Vervolgens kon het onderzoek naar de communicatiemogelijkheden voor het basisstation worden gedaan. De belangrijkste input voor het onderzoek zijn de requirements. De voornaamste onderzoekspunten zijn geformuleerd, waarna een vooronderzoek en het onderzoek zelf zijn uitgevoerd. Een beschrijving van het onderzoek is te vinden in het onderzoeksdocument in bijlage III. Het onderzoek was nauwelijks afhankelijk van andere projectonderdelen, de informatiebronnen waren bijvoorbeeld op internet te vinden. Hierdoor was de kans op problemen kleiner. Er zijn hierbij dan ook geen problemen opgedoken.

### 6.2 Ontwerp

Het resultaat van het onderzoek kon gebruikt worden als input voor het ontwerp van het basisstation. Verdere input voor het ontwerp waren de requirements, maar ook het ontwerp van de hardware van het basisstation. Dit vormde een probleem, omdat er een lange tijd geen persoon binnen het Sanana-project was aangewezen die verantwoordelijk was voor de hardware van het basisstation. Het project was hiervan afhankelijk, omdat er bij het hardware ontwerp bijvoorbeeld bepaald wordt wat voor microcontroller er gebruikt wordt. Dit is heel bepalend voor de diverse aspecten die aan bod komen bij het bouwen van de software, omdat de programmeertaal, het te gebruiken software framework en de Integrated Development Environment (IDE) hiervan afhankelijk zijn.

Gedurende de tijd dat er nog geen hardware ontwerp was, is er tijd besteed aan het beheersen van de programmeertaal C. Dit is gedaan omdat deze programmeertaal vrijwel altijd wordt gebruikt voor het programmeren van microcontrollers. Via informatie van het internet en het uitproberen van programma’s is het relatief eenvoudig gelukt om de programmeertaal te beheersen, ook vanwege ervaring met andere programmeertalen waaronder de vergelijkbare C++. Omdat de requirements al wel bekend waren, is vervolgens, zonder dat het te gebruiken software framework bekend was, een software opbouw bepaald. Hierbij is bijvoorbeeld rekening gehouden met verschillende serialisatiemethoden door de software voor de serialisatie te scheiden in een apart herbruikbaar onderdeel. Verder is er rekening gehouden met het gemakkelijk kunnen toevoegen en aanpassen van verschillende type berichten. Het bepalen van de software opbouw kon gedaan worden omdat de nog onbekende onderliggende softwarelaag wel minimaal de functionaliteit heeft om berichten te kunnen communiceren met andere systemen. Dit was het uitgangspunt bij het bepalen van de software opbouw.

Nadat het hardware ontwerp voor het basisstation bekend was, en dus bekend was welke microcontrollers er werden gebruikt, kon er worden verdiept in de te gebruiken IDE’s en de software frameworks. Op dat moment werd ook bekend dat het basisstation met twee microcontrollers werkt in plaats van één, namelijk één voor elke netwerk interface. Twee afzonderlijke software-componenten bleken nodig te zijn, die beide met een andere IDE ontwikkeld konden worden. Eén component is ethernet software gebruikmakend van een TCP/IP framework van Microchip. Het andere component is Zigbee software gebruikmakend van Z-Stack (een Zigbee software framework) van Texas Instruments. De ethernet software wordt ontwikkeld in een Microchip IDE en de Zigbee software in IAR Embedded Workbench.

Nadat beheersing is verkregen over beide IDE’s en het verdiepen in de software frameworks, werd het software ontwerp opgesteld. Herbruikbaarheid van softwareonderdelen kreeg de focus. Het ontwerp was grotendeels gebaseerd op het al eerder bedachte ontwerp, met uitzondering van een klein aantal punten. Bijvoorbeeld, het bleek dat het Microchip software framework geen ondersteuning biedt voor het HTTP protocol. Hiervoor moest dus een zelf te bouwen softwarelaag worden toegevoegd in het ontwerp.

Binnen de template voor het ontwerpdocument van het bedrijf bleek een belangrijk onderdeel te ontbreken. Er wordt vereist dat de interface van ieder softwareonderdeel wordt beschreven, maar er ontbreekt een uitleg over de werking van de software. Het doel van het ontwerp is juist om vast te leggen hoe het systeem gaat werken. Met alleen een opsomming van functies wordt dit niet bereikt. Bij verschillende scenario’s zou duidelijk moeten zijn wanneer welk softwareonderdeel gebruikt wordt en hoe dit verloopt. Doordat dit niet gedaan wordt is het lastig om de werking van de software te achterhalen. Ook kunnen er nog fouten in het ontwerp zitten die er niet uit gehaald worden. Doordat er geen scenario’s worden beschreven kan het voorkomen dat er bijvoorbeeld functies zijn die ontbreken. Als voorbeeld van de te behalen voordelen zijn er bij het softwareontwerp diagrammen toegevoegd die de werking van de software beschrijven.

### 6.3 Overlapping ontwerp met andere deelsystemen

De transponders, bakens en het basisstation gebruiken allemaal dezelfde microcontroller voor de Zigbee netwerk interface. Het software framework Z-Stack wordt dus ook door allen gebruikt. Op alle deelsystemen moet de software via Zigbee berichten kunnen sturen en ontvangen. Aan de hand hiervan moeten andere applicatietaken worden uitgevoerd. Er is dus een grote overlap in de software van de verschillende deelsystemen. Alle software bevat een onderdeel voor Zigbee communicatie met verder een aantal arbitraire softwaretaken. Om deze reden kan hetzelfde globale softwareontwerp worden gebruikt op ieder deelsysteem. Op dit moment is dat niet gedaan. Hier zou meer gebruik van gemaakt moeten worden om de software zo begrijpelijk mogelijk te houden.

### 6.4 Systeeminterfaces

Vervolgens ontbrak er nog één belangrijk onderdeel, namelijk de vastgelegde communicatie tussen de systemen. Allereerst werd intern binnen de organisatie met andere ontwikkelaars overlegd over de exact benodigde berichten die verstuurd moeten kunnen worden tussen het basisstation, de transponders en de beacons. Dit werd uiteindelijk vastgelegd aan de hand van de requirements en het resultaat van het onderzoek.

Met het partnerbedrijf Delite Group is op éénzelfde manier overeenstemming bereikt over de systeeminterfaces van het basisstation en de externe server. Wekelijks werd meerdere malen samengewerkt door middel van gesprekken en e-mail. Vervolgens is op een succesvolle manier overeenstemming bereikt en is het resultaat vastgelegd.

Het resultaat van de ontwerpfase is het Software Design Specification (SDS) document. Voor verdere technische details wordt hier naar verwezen. De SDS is te vinden in bijlage IV.

# Realisatiefase

Nadat het ontwerp vast lag kon gestart worden met het daadwerkelijk programmeren van het systeem. Allereerst werden volgens het ontwerp de verschillende softwareonderdelen gecreëerd. Vervolgens is er in elk onderdeel verdiept en een verdere invulling gegeven. Tijdens de realisatie is op een aantal details na het ontwerp aangehouden.

### 7.1 Seriële communicatie tussen de microcontrollers

Tussen de twee microcontrollers wordt gecommuniceerd. Dit wordt gedaan via een seriële poort. Bij het programmeren hiervan had zich een probleem voorgedaan. De onderliggende softwarelaag in het Z-Stack framework dat gebruikt kan worden voor seriële communicatie bleek niet te werken. Dit is een cruciaal onderdeel; zonder communicatie tussen de twee microcontrollers werkt het gehele basisstation niet.

Het probleem werd opgelost door voor de seriële communicatie geen Z-Stack te gebruiken, maar deze laag zelf te programmeren. Allereerst werd geïnventariseerd of dit al eens eerder was gedaan binnen het bedrijf. Dit bleek het geval te zijn en voorgaande software zou gebruikt kunnen worden. Na het onderzoeken van deze software bleek het ontwerp zodanig te zijn dat het separation of concerns principe niet was toegepast. Hierdoor kon de software niet worden hergebruikt, omdat de code voor seriële communicatie en de code voor andere softwareonderdelen door elkaar stond. Uiteindelijk is de software opnieuw geprogrammeerd. Dit kon gedaan worden door middel van de documentatie van de microcontroller waarin beschrijvingen staan over seriële communicatie en welke uitgangen van de microcontroller hiervoor gebruikt kunnen worden.

### 7.2 Testen

Tijdens het programmeren van de software kon getest worden door middel van beschikbare ontwikkelhardware. Op deze hardware kan gedebugged worden en bevindt zich bijvoorbeeld een display waarop tekst getoond kan worden evenals een aantal LED’s (zie figuur 5).



Figuur 5 - Ontwikkelhardware Zigbee

Bij de ontwikkelhardware voor de ethernet software (Microchip) deed zich een probleem voor. Het stap voor stap debuggen van de software bleek niet te werken. De oplossing voor dit probleem was om de seriële poort te verbinden met een laptop, zodat er tekst naar de laptop verstuurd kon worden om hierop informatie te tonen. Hoewel dit nog steeds een beperkte manier van debuggen is (stap voor stap de coderegels uitvoeren was nog steeds onmogelijk), was het voldoende om de software te kunnen programmeren.

De ontwikkeling van de transponder en de beacon werd ook gedaan door middel van ontwikkelhardware. De Zigbee communicatie tussen het basisstation, de transponders en de beacons kon hiermee dus getest worden. Dit bestond voornamelijk uit het uitproberen van verschillende scenario’s en het tonen van tekst op de display’s.

### 7.3 Programmacode

Tijdens het programmeren werd een code richtlijn aangehouden die gebruikelijk is binnen het bedrijf. Opvallend was dat er weinig controle zit op de kwaliteit van de software vanuit het bedrijf. Dit geldt ook voor het aanhouden van de code richtlijn. Dit viel op omdat bij oude software bijvoorbeeld niet volgens de code richtlijnen was geprogrammeerd.

Het resultaat van de realisatiefase is de programmacode. Voor een voorbeeld hiervan, zie bijlage V.

# Validatiefase

Om te controleren of het systeem aan de eisen voldoet wordt volgens een testplan het systeem getest. In dit testplan wordt beschreven welke tests uitgevoerd kunnen worden om aan te tonen dat aan alle requirements wordt voldaan.

De testplannen binnen het bedrijf bestaan op verschillende niveau’s, namelijk een algemeen testplan voor het gehele systeem, een testplan voor de software van een deelsysteem en een testplan voor de hardware van een deelsysteem. Voor dit project is alleen het testplan voor de software van een deelsysteem relevant.

De validatiefase is op het moment van het schrijven van het verslag nog niet afgerond. Het testen van het basisstation zal voornamelijk bestaan uit het controleren of er bij verschillende scenario’s alle berichten op een juiste manier worden afgehandelt. Bij verschillende mogelijke input’s moet dus gecontroleerd worden of de output correct is.

# Uitbreiding project

Omdat het ontwerpen en bouwen van de software zonder veel problemen relatief gemakkelijk verliep bleek er tijd over te zijn binnen dit project. Dit in combinatie met een ontbrekende programmeur verantwoordelijk voor de software van de beacon betekende dat het specificeren, ontwerpen, bouwen en testen van deze software toegevoegd werd aan het oorspronkelijke project.

Er is gestart met het specificeren van de requirements van de beacon. Dit is op éénzelfde manier gedaan als bij het basisstation, de requirements zijn afgeleid uit de algemene multidisciplinaire systeemrequirements en verder uitgewerkt.

Voor de beacon was in voorgaande projecten al hardware en software gebouwd. Vanwege gedeeltelijk andere requirements voor de nieuwe beacon is een nieuw ontwerp opgesteld. In bijlage VI wordt dit ontwerp beschreven. De oude software is wel bestudeerd om zo veel mogelijk onderdelen te hergebruiken.

Vanwege de overlap in de functionaliteit van de beacon en het basisstation (het zijn namelijk beide Zigbee systemen), kon het globale softwareontwerp van het basisstation worden hergebruikt voor de beacon.

Bij het programmeren van de beacon kon het onderdeel dat verantwoordelijk is voor het uitzenden van een ultrasoon signaal van de oude software worden hergebruikt. Dit onderdeel is in een eerder project al gebouwd om te onderzoeken of het zendprincipe werkt. Dit onderdeel is verder niet veranderd binnen de requirements van het Sanana project. Bij het bestuderen van de oude software bleek het separation of concerns principe goed toegepast te zijn, maar kwam een ander probleem naar voren.

Bij het ontwerpen van het softwareonderdeel is niet gelet op de manier waarop het softwareonderdeel gebruikt kan worden. De programmeur die de softwarelaag later zou gebruiken zou zo min mogelijk keuzes moeten kunnen maken door zoveel mogelijk af te schermen. Dit bevordert de productiviteit. Onderdelen (bijvoorbeeld functies en variabelen) die alleen relevant zijn voor de interne werking van de softwarelaag zouden niet blootgesteld moeten worden. Hier is niet op gelet waardoor de tijd die het kost om uit te zoeken hoe de softwarelaag gebruikt moet worden evenals de kans op het maken van fouten groter is. Het softwareonderdeel is uiteindelijk hergebruikt en aangepast zodat op het hiervoor benoemd probleem wel is gelet.

Verder is de overige functionaliteit geprogrammeerd. Vanwege de focus op herbruikbaarheid bij het ontwerp van het basisstation is zoveel mogelijk functionaliteit van het basisstation bij het programmeren overgenomen.

# Conclusies en aanbevelingen

De software voor het lokaliseringssysteem is gerealiseerd. Hiermee worden de volgende resultaten behaald:

* De hardware van de bakens en de basisstations die door V.E.D.S. Group geproduceerd wordt kan getest worden. Hiermee kunnen fouten in de hardware gevonden worden die vervolgens gerepareerd moeten worden.
* Het systeem kan in zijn geheel getest worden. Hierbij zijn de overige deelsystemen nodig. Daarmee kan aangetoond worden dat het principe van het systeem werkt en kan er in de toekomst verder worden ontwikkeld tot een verkoopbaar product.
* De gerealiseerde software kan hergebruikt worden bij het ontwikkelen van nieuwe software of het aanpassen van bestaande software binnen het bedrijf.

Met betrekking tot het softwareontwikkelproces zijn er verbeteringen mogelijk. Hiervoor worden hieronder een aantal aanbevelingen gedaan.

#### Separation of Concerns

Bij het ontwerpen van software wordt niet voldoende het separation of concerns principe toegepast; software wordt onvoldoende opgedeeld in zo min mogelijk overlappende verschillende onderdelen. Dit hindert de herbruikbaarheid van software binnen het bedrijf. Verder neemt de begrijpelijkheid af en is het aanpassen van software in de toekomst lastig. Separation of concerns zou als uitgangspunt genomen moeten worden bij het softwareontwerp. Hiermee zou software veel vaker geschikt zijn voor hergebruik en kan er een stap in het softwareontwikkelproces toegevoegd worden waarin gecontroleerd wordt welke software hergebruikt kan worden.

#### Het gebruik van een softwarelaag

Bij het ontwerpen en programmeren van een softwareonderdeel, nadat de gehele software al is opgedeeld in verschillende onderdelen, wordt onvoldoende nagedacht over hoe het losse softwareonderdeel gebruikt kan worden door een bovenliggende softwarelaag.

Bij het ontwerpen en programmeren van het softwareonderdeel zou de focus moeten liggen op hoe het onderdeel gebruikt kan worden. De programmeur die het softwareonderdeel gebruikt zou zo min mogelijk keuzes moeten hebben. Dit vermindert de kans op fouten en hierdoor neemt de tijd die nodig is om uit te zoeken hoe het gebruikt kan worden af.

#### Sequentie diagrammen

In het ontwerpdocument ontbreekt een vorm van sequentiediagrammen die uitleg geven over hoe de software werkt. Hierdoor is het lastig om te achterhalen hoe de software werkt.

Een vorm van sequentiediagrammen zou toegevoegd moeten worden aan het ontwerp. Hiermee verbetert de begrijpbaarheid van de software doordat er met het ontwerpdocument beter uitgelegd wordt hoe de software werkt. Verder kunnen er fouten uit het ontwerp gehaald worden. Bij het beschrijven van scenario’s kan bijvoorbeeld blijken dat er functies ontbreken, of dat er juist functies zijn die overbodig zijn (nadat alle scenario’s zijn beschreven).

#### Code reviews

Er is weinig controle op de kwaliteit van de code. Een oplossing hiervoor is om codereviews in te voeren waarmee software ontwikkelaars elkaars code kunnen verbeteren. Hiermee worden bijvoorbeeld inefficiënte oplossingen gefilterd. Verder kan er op die manier ook voor gezorgd worden dat alle code op éénzelfde manier is geprogrammeerd door te controleren of de programmeerrichtlijn is nageleefd. Codereviews zou zorgen voor meer controle op de softwarekwaliteit.

# Evaluatie

Het was een leuke opdracht omdat er een tastbaar systeem wordt gebouwd in een groot project met veel samenwerking. In vergelijking met mijn vorige stage was er veel meer interactie, waardoor het gewoon leuker en tegelijkertijd leerzamer was.

Het belangrijkste dat ik geleerd heb is dat goede samenwerking belangrijk is. Gedurende de stage was ik vrijwel altijd aan het samenwerken met andere personen van andere disciplines en/of een ander bedrijf. Het is belangrijk om zoveel mogelijk te communiceren zodat er geen onduidelijkheden ontstaan, en om duidelijke afspraken te maken en doelen te stellen.

Verder is het belangrijk om goed prioriteiten te stellen, vooral binnen een groter project. Het belangrijkste is om te starten met de onderdelen waarin afhankelijkheden zitten. Deze onderdelen moeten gewoon altijd zo snel mogelijk worden afgerond. Bij afhankelijkheden horen namelijk risico’s. Als je eerst je eigen werk doet en dan pas het werk doet waarin een afhankelijkheid zit en de persoon waar je afhankelijk van bent is bijvoorbeeld ziek, dan kun je niet verder.

Ik heb verder veel geleerd over hoe het er aan toe gaat binnen een groter bedrijf, in tegenstelling tot mijn vorige stage. Alles is heel erg commercieel, gericht op maximale winst. Ik heb gezien dat het af en toe ook helemaal mis gaat in projecten. Binnen dit project werd er binnen de stuurgroep lang gediscussieëerd over bepaalde keuzes waardoor er een flinke vertraging was voor de definitieve requirements van het systeem. Gelukkig had dit voor mij vrijwel geen impact, alle relevante requirements voor mijn onderdelen waren namelijk al wel bekend.

De wijziging die ervoor zorgde dat de software embedded zou worden, heeft veel impact gehad op de software. Voor mij was dit wel balen, ik had liever met een hoger level programmeertaal en met een operating system gewerkt. Achteraf gezien was het wel heel erg leerzaam om de embedded kant te zien. Verder maakte het natuurlijk niet heel veel uit voor de aanpak en de structuur van de stage.

# Literatuurlijst

### Documentatie

1. Oude documentatie van het bedrijf over het systeem en het product.
2. User guide over de CC2530 microcontroller en Z-Stack van Texas Instruments.
3. User guide over de PIC microcontroller en de TCP/IP Stack van Microchip.

### Websites

1. Website voor het leren van de programmeertaal C

<http://www.cprogramming.com/tutorial.html>

1. NU.nl, actueel nieuws

<http://www.nu.nl/binnenland/2337242/aantal-dementerende-nederlanders-verdubbelt.html>

# Bijlagen

## Bijlage I PID

## Bijlage II Requirementsdocument

## Bijlage III Onderzoeksdocument

## Bijlage IV Ontwerpdocument Basisstation

## Bijlage V Programmacode Basisstation (voorbeeld)

## Bijlage VI Ontwerpdocument Baken