

AFSTUDEERSCRIPTIE

AUTOMATISCHE REGISTRATIE-TOOL VOOR ZORGMIDDELEN

Door

F.C.D. van der Steen

1576660

Technische Informatica

Hogeschool Utrecht

P.N. de Wit

1573954

Technische Informatica

Hogeschool Utrecht

Eerste examiner

M. Wensink

Hogeschool Utrecht

Referentie: ARTZ-SC-010

Datum: 27-05-2013

Versie: 1.0

Status: Final

© TASS B.V. 2013

Alle rechten voorbehouden. Verveelvuldiging, geheel of gedeeltelijk, is niet toegestaan dan met schriftelijke toestemming van de auteursrechthebbende.

All rights are reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

GEGEVENS STUDENTEN

Naam + voorletters student	van der Steen, F.C.D
Studentnummer	157660
Opleiding	Technische Informatica
Stageperiode	01-02-2013 t/m 01-07-2013

Naam + voorletters student	de Wit, P.N
Studentnummer	1573954
Opleiding	Technische Informatica
Stageperiode	01-02-2013 t/m 01-07-2013

GEGEVENS BEDRIJF

Naam bedrijf/instelling	Tass Technology Solutions
Plaats	Eindhoven
Naam + voorletters bedrijfsbegeleider	Zielman, J.
Functie bedrijfsbegeleider	People Manager

GEGEVENS STUDENTEN

Naam + voorletters docentbegeleider	Wensink, M.
-------------------------------------	-------------

Getekend voor gezien door de bedrijfsbegeleider J. Zielman:

Datum: 27-05-2013

REVISIE HISTORIE

Dit document heeft de volgende versiegeschiedenis.

Versie	Status	Datum	Auteur	Wijzigingen
0.1	Draft	07-02-2013	F.C.D. van der Steen	Initiële versie.
0.2	Draft	22-02-2013	P.N. de Wit	Werkwijze uitgewerkt.
0.3	Draft	03-04-2013	P.N. de Wit F.C.D. van der Steen	Lay-out en inhoud bijgewerkt.
0.4	Draft	12-04-2013	P.N. de Wit F.C.D. van der Steen	Complete review. Onderzoeken toegevoegd.
0.5	Draft	23-04-2013	P.N. de Wit	Review verbeteringen doorvoeren.
0.6	Draft	06-05-2013	P.N. de Wit F.C.D. van der Steen	Review verbeteringen doorvoeren en herstructureren.
0.7	Concept	16-05-2013	P.N. de Wit F.C.D. van der Steen	Review verbeteringen doorvoeren.
0.8	Concept	19-05-2013	P.N. de Wit F.C.D. van der Steen	Persoonlijke evaluatie Patrick. Resultaten. Testen. Conclusie en aanbevelingen.
1.0	Final	27-05-2013	P.N. de Wit F.C.D. van der Steen	Laatste review.

DISTRIBUTIELIJST

Dit document wordt aan de volgende personen gedistribueerd.

Naam	Organisatie	Functie
J. Zielman	Tass Technology Solutions	People Manager
B. Bilos	Tass Technology Solutions	Embedded Software Engineer
M. Wensink	Hogeschool Utrecht	Afstudeercoördinator
F.C.D. van der Steen	Tass Technology Solutions	Afstudeerder
P.N de Wit	Tass Technology Solutions	Afstudeerder
Afstudeercommissie	Hogeschool Utrecht	Afstudeercommissie

WERKVERDELING SCRIPTIE

Dit document is geschreven door beide afstudeerders en heeft de volgende werkverdeling.

Onderdeel	Naam
Voorwoord	P.N. de Wit
Samenvatting	F.C.D. van der Steen & P.N. de Wit
Introductie	F.C.D. van der Steen
Opdrachtbeschrijving	P.N. de Wit
Werkwijze	P.N. de Wit
Onderzoek	F.C.D. van der Steen
Software Ontwerp ARTZ	P.N. de Wit
Testen	F.C.D. van der Steen
Resultaten	F.C.D. van der Steen
Conclusie en aanbevelingen	P.N. de Wit
Evaluatie	F.C.D. van der Steen & P.N. de Wit

VOORWOORD

Dit document is gemaakt door twee afstudeerders aan de Hogeschool Utrecht te Utrecht en is niet alleen bedoeld om een beter inzicht te geven in de opdracht die is uitgevoerd bij Tass Technology Solutions, maar ook als afstudeerscriptie voor de afstudeerders.

De afstudeerders betuigen hun dank aan de medewerkers en andere afstudeerders van Tass Technology Solutions voor de tijd die ze hebben vrijgemaakt om advies te geven over beslissingen.

Extra dank gaat uit naar de afstudeerbegeleider van Tass Technology Solutions; Mevr. J. Zielman, voor de begeleiding tijdens het traject van de opdracht en het geven van feedback op opgeleverde documentatie. Ook gaat dank uit naar de technisch afstudeerbegeleider Dhr. B. Bilos, voor de technische ondersteuning tijdens de uitvoering van de opdracht en de feedback op documentatie.

Als laatste gaat er dank uit naar de afstudeerbegeleider van de Hogeschool Utrecht; Dhr. M. Wensink, voor de begeleiding tijdens het traject van de opdracht en het geven van feedback.

Eindhoven, april 2013

F.C.D. van der Steen & P.N. de Wit
Afdeling Delivery
Tass Technology Solutions

INHOUDSOPGAVE

Samenvatting	1
Summary	2
Afkortingen	4
Definities	6
1 Introductie	8
2 Het afstudeerbedrijf – Tass Technology Solutions	9
2.1 Bedrijfsgeschiedenis	9
2.2 Afstuderen binnen het bedrijf	10
2.3 Bedrijfsorganisatie	10
3 Opdrachtbeschrijving	11
3.1 Probleemanalyse	11
3.2 Doelstelling	11
3.3 Deelopdrachten	12
4 Werkwijze	14
4.1 Scrum	14
4.2 Team en werkverdeling	16
4.3 Kwaliteitswaarborging	16
5 Onderzoek	17
5.1 Bestaande systemen	17
5.2 Draadloze communicatietechnieken	20
5.3 Ontwikkelplatformen	24
5.4 Ontwikkelframework	30
5.5 Ontwerp ARTZ	33
6 Softwareontwerp ARTZ	35
6.1 Backend tier	35
6.2 Frontend tier	38
6.3 Middleware tier	44
6.4 Mobiele app	46
7 Testen	47

8 Resultaten	48
8.1 De backend tier	48
8.2 De middleware tier	48
8.3 De frontend tier	48
8.4 Problemen	49
9 Conclusie	51
10 Aanbevelingen	52
11 Evaluatie	53
12 Referenties	54
13 Bijlagen	58

SAMENVATTING

Tass Technology Solutions is een dienstverlener op het gebied van technische- en embedded software. Tass heeft gemerkt dat zich onder zorgverleners het probleem voordoet dat zorgmiddelen nergens te vinden zijn. De oorzaak van dit probleem is dat de zorgmiddelen vaak met de hand geregistreerd moeten worden en soms wordt vergeten om te doen.

Om dit probleem op te lossen is een automatisch registratiesysteem op basis van draadloze (energiezuinige) technieken ontwikkeld. Het systeem zorgt ervoor dat de zorgaanbieders niet meer afhankelijk zijn van handmatige registratie en verloren zorgmiddelen eenvoudig teruggevonden kunnen worden.

Tijdens de uitvoering van de opdracht is Scrum gebruikt. Bij het opstarten is een globale planning van alle op te leveren producten gemaakt, met de daarbij behorende datum van voltooiing. Er is elke twee weken een sprint gepland.

Om het systeem te kunnen ontwikkelen moest eerst onderzoek worden gedaan. Eerst is onderzoek gedaan naar een mogelijk bruikbare techniek voor het automatisch registreren van zorgmiddelen. Dit onderzoek toont aan dat RFID zou kunnen zorgen voor een oplossing. Door RFID scanners binnen een gebouw op vaste locaties te verspreiden, kunnen zones worden afgebakend. Zodra een zorgmiddel langs een scanner beweegt, merkt de scanner dit op (door een RFID tag aan elk zorgmiddel te bevestigen) waarna de positie van het zorgmiddel, aan de hand van de vorige positie en de locatie van de scanner, kan worden bepaald.

Omdat de RFID hardware duur is en het product gaat worden gebruikt voor commerciële presentaties aan potentiële klanten, is ervoor gekozen om een schaalmodel te maken.

Ook is onderzoek gedaan naar een ontwikkelframework dat bruikbaar is bij het ontwikkelen van een applicatie die in staat is om zorgmiddelen te zoeken. Om de applicatie platformonafhankelijk te maken is ervoor gekozen om een webapplicatie te ontwikkelen. Aangezien Tass intern voor webapplicaties alleen Ruby on Rails gebruikt, is Ruby on Rails gekozen als framework.

Door het gebruik van de applicatie kunnen zorgaanbieders verloren zorgmiddelen in hun zorginstelling terugvinden.

SUMMARY

Tass Technology Solutions is a service provider in the field of technical and embedded software. Tass has noticed that among healthcare providers, the problem arises that healthcare resources are lost within the care institution. The cause of this problem is that the resources are often manually registered, which is often forgotten to do.

To solve this problem, an automatic registration system based on wireless (energy efficient) techniques has been developed. The system ensures that healthcare providers are no longer dependent on manual registration and lost healthcare resources can simply be found through this system.

During the execution of the project Scrum was used. At the startup phase, an overall planning was made of all the products that had to be delivered, including their date of completion. Each two weeks a sprint was planned.

To develop the system research had to be done. First research has been done for finding a possibly usable technique for automatically registering healthcare resources. This research shows that RFID could possibly provide a solution. By spreading RFID scanners throughout a building on fixed locations, zones are created. RFID scanners notice healthcare resource passing by (by using RFID tags that are attached to the healthcare resources) and pinpoint the location by using both the location of the RFID scanner and the previously known location of the healthcare resource.

However, because the RFID hardware is expensive and the product is to be used for commercial presentations to potential customers, the choice was made to make a scale model.

Also research has been done to select a framework usable to develop an application used for searching and displaying the positions of the healthcare resources. Internally Tass is using Ruby on Rails for their own applications. For that reason and the need for platform independency, Ruby on Rails was chosen as the framework for the application.

By using the application, healthcare providers are able to find healthcare resources that are lost within a healthcare institution.

AFKORTINGEN

A	API	A pplication p rogramming i nterface
	ARTZ	A utomatisch r egistratie t ool voor z orgmiddelen
B	BT	B lue t ooth
D	DBMS	D atabase m anagement s ystem
G	GUI	G raphical u ser i nterface
H	HTML	H ypertext m arkup l anguage
	HTTP	H ypertext t ransfer p rotocol
I	IDE	I ntegrated d evelopment e nvironment
	IEC	I nternational e lectrotechnical c ommission
	IR	I nfrar o od
	ISO	I nternational o rganization for s tandardization
J	JS	J avas cr ipt
M	MVC	M odel- v iew- c ontroller
	MVP	M odel- v iew- p resenter

MVVM **M**odel **v**iew **v**iew**m**odel

N

NFC **N**ear **f**ield **c**ommunication

O

OS **O**perating **s**ystem

R

RFID **R**adio **f**requency **i**dentification

RoR **R**uby **o**n **R**ails

RSSI **R**eceived **s**ignal **s**trength **i**ndication

X

XOR **E**xclusive **o**r

DEFINITIES

A	Android	Het mobiele opensource besturingssysteem van Google gebruikt op veel smartphones, tablet en smartcameras.
B	Beacon	Een opzettelijk opvallend apparaat, ontworpen om de aandacht op een specifieke locatie te vestigen.
	Bing	De zoekmachine van Microsoft voor het zoeken naar webpagina's op het internet.
C	Checksum	Een veld berekend uit een blok van digitale data, om fout detectie toe te kunnen passen op de data.
G	Google	De zoekmachine van Google Inc. voor het zoeken naar webpagina's op het internet.
I	iOS	Het mobiele besturingssysteem van Apple gebruikt op de iPhone, iPad, iPod touch en Apple TV.
K	Kamernauwkeurigheid	Mate van nauwkeurigheid voor de plaatsbepaling van objecten. Met kamernauwkeurigheid kan bepaald worden in welke kamer (binnen een gebouw) een object zich bevindt. De positie binnenin de kamer kan niet worden bepaald.
M	Memory leak	Een probleem waar een (computer)programma geheugen allocatie verkeerd beheert. Het kan prestaties van de computer verminderen of in het ergste geval zorgen dat deze niet meer goed functioneert.

P	Prototyping platform	Een ontwikkelboard waarmee prototypen van hardware schakelingen kunnen worden ontwikkeld.
R	Referentiepunt	Locatiepunt met een van te voren bekende locatie
T	Tag	Hardware die aan een zorgmiddel bevestigd is om deze herkenbaar te maken

1 INTRODUCTIE

Zorg in Nederland wordt steeds duurder, dit komt onder andere omdat we steeds langer leven. Dat is positief, maar betekent ook dat we langer gebruik maken van de zorg. Nieuwe en soms kostbare behandelingen en de toename van het aantal chronisch zieken dragen eveneens bij aan de stijgende zorgkosten.

Ook doet het probleem zich voor dat bij veel zorgverleners de registratie van hun zorgmiddelen handmatig wordt uitgevoerd. Hier ontstaat de kans dat dit vergeten wordt en dat zorgmiddelen hierdoor spoorloos raken. Het moeten zoeken of zelfs helemaal moeten vervangen van zorgmiddelen kost onnodig tijd en geld.

De opdracht probeert dit probleem op te lossen door middel van een automatisch plaatsregistratiesysteem. Echter dit vereist onderzoek naar hoe het systeem kan worden gerealiseerd.

Hoofdstuk 2 beschrijft het afstudeerbedrijf waarbij de opdracht is uitgevoerd. In hoofdstuk 3 wordt een overzicht gegeven van de opdracht. Hoofdstuk 4 geeft de werkwijze aan die gedurende de opdracht is gebruikt. Hoofdstuk 5 beschrijft de onderzoeken die zijn uitgevoerd voor de realisatie van de opdracht. In hoofdstuk 6 wordt het software ontwerp beschreven. Hoofdstuk 7 beschrijft de tests die zijn uitgevoerd. In hoofdstuk 8 worden de behaalde resultaten besproken. In hoofdstuk 9 wordt een conclusie gegeven over de opdracht en het resultaat. In hoofdstuk 10 worden de aanbevelingen besproken indien het project doorontwikkeld wordt en in hoofdstuk 11 wordt een evaluatie gegeven terugkijkend op de afstudeerperiode.

Verwijzingen naar bronnen worden aangeduid met [{nummer}] en kunnen worden gevonden in hoofdstuk 12. Verwijzingen naar bijlagen worden aangeduid met [{letter}] en kunnen worden gevonden in hoofdstuk 13.

2 HET AFSSTUDEERBEDRIJF – TASS TECHNOLOGY SOLUTIONS

In dit hoofdstuk wordt het afstudeerbedrijf nader toegelicht. Allereerst wordt in paragraaf 2.1 kort de geschiedenis van het bedrijf beschreven en in welke steden het bedrijf tegenwoordig gevestigd is. Daarna wordt de rol van afstudeerders binnen het bedrijf toegelicht (zie paragraaf 2.2). Als laatste wordt de interne organisatiestructuur van Tass Technology Solutions beschreven (zie paragraaf 2.3).

2.1 BEDRIJFSGESCHIEDENIS

Tass Technology Solutions is in september 1978 begonnen als een software-afdeling onder de naam 'Special Projects Group' (SPG) binnen Philips. 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.

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 zo hard dat in 1998 Tass International werd opgericht dat bestond uit een Nederlandse divisie en een Belgische divisie.



Figuur 1 - Hoofkantoor Eindhoven.

In 2007 besloot Philips zich meer te gaan richten op de core business van het bedrijf, daarom werd Philips Tass verkocht. Philips Tass werd Tass Technology Solutions. Tass ging verder als onderdeel van Total Specific Solutions (TSS) [1]. TSS bundelt de kennis en ervaring van markt-, business- en softwarespecialisten. Samen staan de TSS-bedrijven voor een combinatie van specialistische kennis van markten, bedrijfsprocessen en hoogwaardige software. Ze opereren vanuit afzonderlijke bedrijven, maar hebben een gemeenschappelijke directieraad en een gedeelde visie op de markt. In 2008 vestigde Tass zijn hoofdkantoor op het Larixplein te Eindhoven (zie figuur 1).

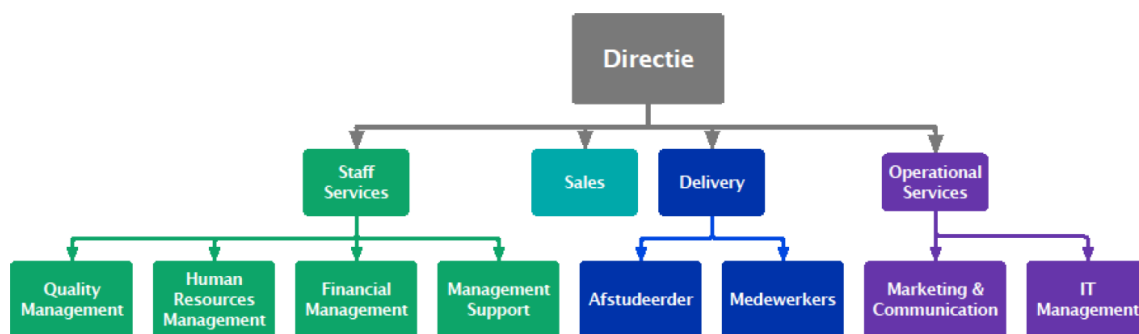
Tass is een dienstverlener op het gebied van technische- en embedded software. Het bedrijf heeft vestigingen in Eindhoven, Apeldoorn en Brussel. Tass levert met meer dan 200 medewerkers een grote expertise in de vorm van detachering en projecten.

2.2 AFSTUDEREN BINNEN HET BEDRIJF

Tass Technology Solutions laat afstudeerders opdrachten uitvoeren op ICT-gebied om informatie te winnen voor mogelijke projecten. Ook hoopt men de afstudeerders later te kunnen werven. Als afstudeerder is het noodzakelijk om voor het werk dat uitgevoerd wordt en de keuzen die gemaakt worden verantwoording af te leggen bij de ondersteunende bedrijfsbegeleider en bij de technisch begeleider. Ook moet de docentbegeleider op de hoogte worden gehouden van de status van de opdracht en de gemaakte keuzen.

2.3 BEDRIJFSORGANISATIE

In figuur 2 wordt de organisatiestructuur van Tass Technology Solutions weergegeven.



Figuur 2 – Organigram.

De kernactiviteit van Tass Technology Solutions bestaat uit het detacheren van medewerkers, hierdoor speelt de sales afdeling een belangrijke rol. 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 E. 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 interne zowel 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 OPDRACHTBESCHRIJVING

In dit hoofdstuk wordt de beschrijving van de opdracht gegeven. In paragraaf 3.1 worden het probleem en de aanleiding van de opdracht toegelicht. In paragraaf 3.2 worden de doelstelling van de opdracht beschreven. In paragraaf 3.3 worden de deelopdrachten beschreven die moeten worden uitgevoerd.

3.1 PROBLEEMANALYSE

In de gezondheidszorg worden zorgmiddelen als rollators, anti-decubitus matrassen en vernevelapparaten gebruikt. Doordat bij veel zorgverleners de registratie van hun zorgmiddelen handmatig wordt uitgevoerd, wordt het vaak vergeten om dit te doen. Dit resulteert in het probleem dat zorgmiddelen spoorloos zijn, waardoor onnodig tijd verloren gaat aan het zoeken naar deze zorgmiddelen.

Tass Technology Solutions heeft het idee dat door de ontwikkeling van een geautomatiseerd plaatsregistratiesysteem dit probleem in de zorg kan worden opgelost, waardoor deze onnodig verloren tijd en geld kunnen worden bespaard. Dit geautomatiseerde plaatsregistratiesysteem kan worden onderzocht door de volgende onderzoeksvragen te specificeren en te beantwoorden:

- Wat is een/zijn geschikte techniek(en) voor het registreren van zorgmiddelen?
 - Hoe werken deze technieken?
 - Welk algoritme geeft de gewenste nauwkeurigheid voor locatiebepaling en hoe werkt dit algoritme?
- Welk ontwikkelplatform(en) zijn nodig voor de ontwikkeling van de ARTZ?
- Wat is een geschikte architectuur voor de ARTZ?

3.2 DOELSTELLING

De oplossing voor het eerder beschreven probleem is een geautomatiseerd plaatsregistratiesysteem, maar het is nog niet duidelijk **hoe** dit gerealiseerd kan worden. Het doel van de opdracht is om een systeem en een gebruikerstool te ontwerpen en ontwikkelen. Het systeem registreert automatisch de huidige plaats van de zorgmiddelen en de tool geeft gebruikers de mogelijkheid om zorgmiddelen op te zoeken.

3.3 DEELOPDRACHTEN

Omdat nog niet duidelijk is **hoe** het systeem gerealiseerd gaat worden speelt onderzoek een belangrijke rol. Daarom zijn gedurende de opdracht meerdere onderzoeken uitgevoerd om dit duidelijk te krijgen. Deze onderzoeken hadden een ontwerpende functie en bevatten elk de volgende fasen:

- Enquête over eisen met klant Er is een lijst met vragen opgesteld, die door middel van gesprekken met de klant, door het projectteam is ingevuld op basis van antwoorden van de klant. Tijdens de enquête zijn vragen gesteld over bijvoorbeeld wat de te behalen nauwkeurigheid moest zijn of hoeveel de kosten meespeelde in de beslissingen die worden gemaakt.
- Onderzoek naar oplossingen en/of informatie op het internet. Om oplossingen voor problemen of informatie te zoeken is gebruik gemaakt van zoekmachines (bijvoorbeeld Google of Bing).
- Rapporteren van mogelijkheden en/of gevonden informatie. Het vastleggen van gevonden informatie is belangrijk om de informatie niet verloren te laten gaan. Daarom is de informatie opgeslagen in documenten waardoor deze teruggevonden kan worden.
- Vergadering met de klant over de resultaten en de mogelijke besluiten. Om de klant inspraak te geven in belangrijke keuzen die moesten worden gemaakt, zijn vergaderingen over deze keuzen georganiseerd. Hierdoor was het mogelijk om resultaten van bijvoorbeeld onderzoek aan de klant voor te leggen en samen de meest geschikte oplossing te kiezen.
- Eventueel rapport bijwerken met besluit(en). Zodra een besluit gemaakt werd, zijn de nodige documenten bijgewerkt met hierbij de verklaring voor dit besluit. Door dit goed vast te leggen is het mogelijk om terug te lezen waarom bepaalde keuzen wel of niet zijn gemaakt.

Met deze methoden zijn de volgende onderzoeken uitgevoerd. Deze onderzoeken geven antwoord op de bijbehorende onderzoeksvraag. In tabel 1 is een opsomming te zien van de uit te voeren onderzoeken met de bijbehorende onderzoeksvraag waar antwoord op moet worden gegeven.

Onderzoek	Onderzoekvraag
Bestaande systemen	<i>Welke systemen zijn er op de markt en hoe werken deze systemen?</i>
Draadloze communicatietechnieken	<i>Wat is een/zijn geschikte techniek(en) voor het registreren van zorgmiddelen?</i>
Algoritmen voor locatiebepaling	<i>Welk(e) algoritme(n) ge(eft)(ven) de gewenste nauwkeurigheid voor locatiebepaling en hoe werk(t)(en) d(it)(eze) algoritme(n)?</i>
Onderzoek naar ontwikkelplatformen	<i>Welk(e) ontwikkelplatform(en) zijn nodig voor de ontwikkeling van de ARTZ?</i>
Software ontwerp	<i>Wat is een geschikte architectuur voor de ARTZ?</i>

Tabel 1 - Uit te voeren onderzoeken

Het diagram in figuur 3 toont de initiële planning van bovenstaande onderzoeken.

Onderzoek/planning in weken	Januari				Februari				Maart			
Bestaande systemen												
Ontwikkelplatformen												
Draadloze communicatie												
Locatiebepalingsalgoritmen												
Ontwerp ARTZ												

Figuur 3 - Initiële planning voor onderzoeken.

Het eerste onderzoek op de planning is het onderzoek naar bestaande systemen (zie paragraaf 5.1). Dit onderzoek is nodig voor het creëren van een beter beeld van hoe de ARTZ zou kunnen werken.

De volgende twee onderzoeken zijn gericht op ontwikkelplatformen en draadloze communicatietechnieken (zie paragraaf 5.2 en 5.3). Deze onderzoeken leggen de basis van de ARTZ doordat hierin de mogelijke oplossingen worden onderzocht.

Als laatste is een onderzoek uitgevoerd voor het vinden van een geschikte architectuur voor de ARTZ (zie paragraaf 5.5). Hierin zijn alle resultaten van vorige onderzoeken gebruikt om tot een compleet systeem te komen.

Behalve het doen van onderzoek naar **hoe** de ARTZ kan worden gerealiseerd moet de ARTZ ook ontwikkeld worden. Door middel van de antwoorden op de onderzoeksvragen kunnen alle onderdelen voor de realisatie gedefinieerd worden. De realisatie is beschreven in hoofdstuk 6.

4 WERKWIJZE

In dit hoofdstuk wordt de gebruikte werkwijze voor de uitvoering van de opdracht beschreven. Paragraaf 4.1 beschrijft het software-ontwikkelp proces dat binnen de opdracht is gebruikt. Paragraaf 4.2 beschrijft de werkverdeling tussen de afstudeerders.

4.1 SCRUM

Scrum [2] is een iteratief en incrementeel agile software-ontwikkelp proces. Er zijn sprints gemaakt van twee weken. Binnen deze sprints is dagelijks een daily scrum gehouden. Hierin worden de uitgevoerde en toekomstige activiteiten besproken. Ook is besproken of- en welke problemen zich hebben voorgedaan.

4.1.1 SCRUM ROLLEN

Binnen het Scrum proces zijn de volgende rollen van toepassing:

- **Product owner:** Bij Tass Technology Solutions is de product owner meestal de bedrijfsbegeleider. Hij of zij is verantwoordelijk voor het bijhouden van de backlog (geordende lijst van eisen van de product owner voor een bepaald product) van het project. Echter betekent dit niet dat de afstudeerder geen controle of verantwoordelijkheden heeft voor de backlog. Hij kan nieuwe features voorstellen door deze te bespreken met de product owner.
- **Scrum master:** Per projectteam is een Scrum master aangewezen. De Scrum master begeleidt het team om zijn doelstellingen te behalen. Hij is niet de leider van het team. Hij zorgt ervoor dat het Scrum proces gevolgd wordt. Deze rol kan tijdens het project nog door een ander vervuld worden, wat kan leiden tot een andere (en mogelijk een (deels) betere aanpak).

4.1.2 SCRUM MEETINGS

Binnen het Scrum proces zijn de volgende meetings van toepassing:

- **Voortgangsgesprek:** Aan het einde van elke sprint vindt een voortgangsgesprek plaats met de project- en technisch begeleider. Hierin wordt een demo gegeven van de functionaliteiten die tijdens de bijbehorende sprint zijn toegevoegd. Hierdoor is de product owner in staat om bij te sturen indien het product niet naar wens is doorontwikkeld.
- **Sprint review:** Aan het einde van iedere sprint wordt door het ontwikkelteam een sprint review gehouden. Bij de sprint review worden de voortgang, opgeloste- en onopgeloste problemen, ingebouwde functionaliteiten en opgeleverde documenten besproken.

- **Sprint retrospective:** Aan het einde van iedere sprint wordt door het ontwikkelteam een sprint retrospective gehouden. In de retrospective wordt een korte evaluatie gegeven van de afgelopen sprint van wat goed ging en wat mogelijke verbeteringen zijn.

De sprint review en sprint retrospective zijn gedocumenteerd en naar de docent, project- en technisch begeleider gestuurd.

4.1.3 SCRUMWISE

Scrum kost de nodige tijd, daarom is het van essentieel belang om een tool te gebruiken die ervoor zorgt dat niet alles handmatig gedaan hoeft te worden. Voor het project is de tool Scrumwise gebruikt.



Figuur 4 – Scrumwise.

Met Scrumwise kan de gebruiker een project maken en hiervoor sprints inplannen (van begindatum tot einddatum). Ook kan de gebruiker voor het project een backlog opstellen. Een item op de backlog bestaat uit één of meerdere taken, voor deze taken wordt het aantal te besteden uren ingeschat.

Deze backlog items kunnen aan sprints worden toegewezen, zodat ze op een takenbord komen te staan. De ontwikkelaars kunnen hier taken aan hun naam toewijzen of toegewezen krijgen. Naarmate de taak vordert kan het aantal geschatte uren worden bijgewerkt, waarmee een voltooiingspercentage kan worden berekend. Als de taak af is kan deze naar de 'ready to test' staat worden gezet en zodra deze getest is, naar de 'done' staat.

4.2 TEAM EN WERKVERDELING

Gedurende de afstudeeropdracht zijn delen van de activiteiten samen uitgevoerd en andere delen afzonderlijk van elkaar. In tabel 2 is de onderlinge taakverdeling te zien.

Verantwoordelijken	Taken
Samen	<ul style="list-style-type: none"> • Wensen van de opdrachtgever in kaart brengen. • Ontwerp van de ARTZ. • Inrichten SVN. • Onderzoek bestaande systemen. • Onderzoek draadloze technieken.
F.C.D. van der Steen	<ul style="list-style-type: none"> • Onderzoek hardware. • Ontwikkeling aansturing hardware.
P.N. de Wit	<ul style="list-style-type: none"> • Onderzoek ontwikkelplatformen. • Ontwikkeling frontend.

Tabel 2 – Taakverdeling.

4.3 KWALITEITSWAARBORGING

Om een goede kwaliteit te krijgen en waarborgen zijn de volgende taken uitgevoerd:

Code reviews: Om de kwaliteit van de geschreven code te waarborgen is de code binnen het team, maar ook door de technisch begeleider gereviewd. Wijzigingen en verbeteringen die niet van invloed zijn op de werking van functionaliteiten zijn direct doorgevoerd. De wijzigingen die wel invloed hadden op de werking van functionaliteiten werden binnen het team besproken om die later in te kunnen plannen. Indien de wijziging ook invloed had op de werking van de basisfunctionaliteiten van de ARTZ (zoals het tonen van de kaart, het zoeken van zorgmiddelen of de registratie van tags), werd deze daarnaast ook besproken met de klant.

Documentatie reviews: Documentatie werd per hoofdstuk en paragraaf geschreven. De hoofdstukken en paragrafen werden verdeeld onder de teamleden. Na het afronden van één of meerdere paragrafen werden deze binnen het team gereviewd. Na het afronden of het bereiken van een bepaalde versie van een document, werd het document opgestuurd naar de bedrijfsbegeleider en de technisch begeleider ter review. Deze gaven feedback, waarna fouten en onduidelijkheden werden opgelost.

Ontwerpen en modellen: Ontwerpen en modellen werden met het team gemaakt en besproken. Nadat beide teamleden hun akkoord hadden gegeven zijn doorgestuurd naar de bedrijfsbegeleider en de technisch begeleider ter review. Documenten die ter review zijn opgestuurd werden in een voortgangsvergadering besproken. Verbeteringen werden (indien van toepassing) na de voortgangsvergadering doorgevoerd en opnieuw opgestuurd ter review.

5 ONDERZOEK

In dit hoofdstuk worden de uitgevoerde onderzoeken beschreven. Paragraaf 5.1 beschrijft het onderzoek naar bestaande systemen. Paragraaf 5.2 beschrijft het onderzoek naar draadloze communicatietechnieken. In paragraaf 5.3 wordt het onderzoek naar ontwikkelplatformen beschreven. Paragraaf 5.4 beschrijft het onderzoek naar een ontwikkelframework voor de gebruikerstool en paragraaf 5.5 beschrijft het onderzoek naar hoe de ARTZ ontworpen kan worden.

5.1 BESTAANDE SYSTEMEN

De onderzoeksvraag die in dit onderzoek wordt beantwoord is: *"Welke systemen zijn er op de markt en hoe werken deze systemen?"*. Het beantwoorden van deze vraag zorgt ervoor dat een goed beeld ontstaat van hoe een automatisch plaatsregistratie systeem zou kunnen werken.

Om dit beeld te creëren is gezocht naar wetenschappelijke artikelen. Er is gezocht door middel van (populaire) zoekmachines (bijvoorbeeld Google of Bing). Door op termen te zoeken met betrekking tot het registreren en volgen van zorgmiddelen is informatie verzameld. Een voorbeeld van een gebruikte term tijdens dit zoeken is: *"Tracking assets using wireless technology"*.

De wetenschappelijke artikelen die gevonden zijn door middel van het zoeken, beschrijven verscheidene technieken voor het lokaliseren van objecten (voor de ARTZ zijn dit zorgmiddelen) binnen een gebouw. Ook geven de wetenschappelijke artikelen eigenschappen van deze technieken, met daarbij de voor- en nadelen van de betreffende techniek.

De technieken zijn onderzocht en via de zoekmachines is gezocht naar producten die de desbetreffende techniek toepassen.

9solutions [3] maakt gebruik van Bluetooth (BT) beacons op vaste locaties om hiermee de positie te kunnen bepalen. Een nadeel van BT is dat er niet echt een netwerk opgezet kan worden, omdat het gebruik maakt van het master-slave principe. Elke master kan maximaal maar 8 slaves hebben, wat betekent dat het netwerk niet goed schaalbaar is. Toch kan er zonder één groot netwerk te maken van BT nodes een volgsysteem worden gemaakt, door op bepaalde plekken in het gebouw BT referentiepunten te plaatsen (via Wi-Fi of WLAN met een server verbonden) zonder dat, deze direct met andere BT referentiepunten verbonden zijn. Doordat deze BT referentiepunten in staat zijn om BT tags te ontdekken, kan kamernauwkeurigheid worden behaald. Aangezien de tags ongeveer 3 jaar mee kunnen gaan, de apparatuur niet duur is en ARTZ niet meer nauwkeurigheid vereist dan kamernauwkeurigheid, is dit een goede optie om te gebruiken voor ARTZ.

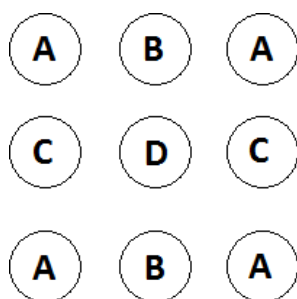
Chainway [4] maakt gebruik van RFID om producten in een supermarkt te kunnen volgen en afrekenen. Een voordeel is dat passieve RFID tags geen batterijen nodig hebben, de tags werken op het magnetisch veld van de kaartlezer. Een groot nadeel is echter dat de kaartlezers extreem duur zijn. Toch is dit één van de technieken die mogelijk is om te gebruiken voor de ARTZ.

IndoorAtlas [5] maakt gebruik van de magnetische polarisatie van de aarde om via een in kaart gebrachte polarisatiemap de locatie van een object te vinden. Om te bepalen waar het object zich bevindt, moet dit vergeleken worden met van tevoren opgezette referentiepunten. Het opzetten van deze referentiepunten duurt erg lang en er bestaat een grote kans dat twee (of meer) punten erg veel op elkaar lijken. Het is dan moeilijk onderscheid te maken, wat kan leiden tot inaccurate resultaten. Doordat het opzetten van het systeem voor elk gebouw erg lang duurt en het eindresultaat onnauwkeurigheden kan opleveren, is dit geen goede oplossing voor de ARTZ.

F-Tech [6] maakt gebruik van ZigBee. Er worden referentienodes geplaatst in ruimten waarmee gecommuniceerd kan worden. Hierdoor kan de locatie van het object worden gevonden. Het nadeel van ZigBee is echter wel dat de communicatie met de server via een centraal punt moet verlopen, wat een bottleneck in de bandbreedte is bij een groot aantal nodes. Het netwerk kan opgesplitst worden in aparte subnetwerken met behulp van ZigBee gateways, maar deze zijn zeer duur. Net als bij BT kan ook bij ZigBee het systeem worden opgesplitst waarbij meerdere netwerken (en dus meerdere coördinatoren) aanwezig zijn binnen een gebouw die allemaal apart op een server zijn aangesloten. Dit maakt het systeem echter ingewikkeld omdat ervoor gezorgd moet worden dat de interferentie tussen de verschillende radio frequency (RF) kanalen minimaal is. Toch is ZigBee een mogelijkheid om te gebruiken in de ARTZ, omdat nodes zeer lang op een batterij mee kunnen gaan en omdat deze nodes goedkoop zijn.

M. I. T. Computer Science and Artificial Intelligence Laboratory [7] maakt gebruik van de Cricket methode om indoor positionering mogelijk te maken. Deze methode is gebaseerd op een combinatie van ultrasoon geluid en RF waarbij de op het plafond verspreide "beacons" tegelijk het ultrasone geluid en het RF-signaal uitzenden en de te positioneren node (de "cricket") deze signalen opvangt. De cricket kan dan de afstand tot de zendende beacon bepalen door het verschil van verspreidingssnelheid van het RF signaal (lichtsnelheid) en het ultrasone signaal (geluidssnelheid) te gebruiken. Een nadeel is dat de crickets intelligent moeten zijn (ze moeten hun eigen positie bepalen). Een ander nadeel is dat dit systeem alleen werkt met "line-of-sight" omdat er van ultrasoon geluid gebruik gemaakt wordt, waardoor de crickets niet worden gezien als ze zich achter muren deuren of andere objecten bevinden.

UbiTrack [8] maakt gebruik van infrarood. Aan het plafond in een gebouw worden zenders opgehangen, met een onderlinge afstand van 90 centimeter tot elkaar om elkaar niet te storen. Zo ontstaat een netwerk van zenders, deze zenders worden onderverdeeld in vier groepen A, B, C of D). De zenders in deze groepen liggen niet naast elkaar, maar worden altijd gescheiden door een zender van een andere groep (zie figuur 5). Hierdoor overlappen de zenders binnen groepen elkaar niet en kan de ontvanger (de tag) zijn positie bepalen door middel van één of meer ID's die wordt/worden meegestuurd vanaf de zender(s). Een nadeel is dat dit systeem alleen werkt met "line-of-sight" omdat infrarood licht niet muren, deuren of andere objecten kan penetreren.



Figuur 5 - Infrarood netwerk set-up

EkaHau [9] maakt gebruik van Wi-Fi. Het Wi-Fi gebaseerde positioneringssysteem dat EkaHau levert maakt gebruik van aanwezige Wi-Fi zenders binnen gebouwen. Het systeem gebruikt de zogenoemde "received signal strength indication" (RSSI) voor de positiebepaling, deze waarde geeft de sterkte weer van het ontvangen signaal. Een nadeel is echter dat RSSI waarden zeer sterk beïnvloed worden door variërende signaal uitzendsterkten, muren, deuren, mensen, alle ijzeren objecten, et cetera. De uitzendsterkte van het signaal varieert mogelijk door energiebesparing (het is uiteraard zonde om een zender voortdurend te laten zenden terwijl er geen apparaten zijn aangesloten).

De beschreven systemen geven antwoord op de vraag: "Welke systemen zijn er op de markt en hoe werken deze systemen?". Door het onderzoek naar de bestaande systemen is een selectie ontstaan van mogelijke technieken om de ARTZ te realiseren. Het onderzoek naar de keuze van de techniek voor de ARTZ wordt beschreven in paragraaf 5.2.

5.2 DRAADLOZE COMMUNICATIETECHNIEKEN

De onderzoeksvraag die in dit onderzoek wordt beantwoord is: "Wat is een/zijn geschikte techniek(en) voor het registreren van zorgmiddelen?". Het beantwoorden van deze onderzoeksvraag zorgt ervoor dat een keuze gemaakt kan worden voor een techniek voor de ARTZ.

Door middel van gesprekken met de klant zijn aan de ARTZ een aantal eisen gesteld, deze zijn; Energiezuinig, kamernauwkeurig, betrouwbaar, redelijke prijs, schaalbaar en uitbreidbaar.

De bruikbare technieken (zie paragraaf 5.1) moeten aan bepaalde criteria voldoen in de onderstaande tabel zijn de criteria voor elk van de eisen gedefinieerd.

Gradatie/Eis	Goed	Matig	Slecht
Energiezuinig	> 2 jaar	> 1 jaar	< 1 jaar
Kamernauwkeurigheid	Elke kamer is apart herkenbaar	Kamers zijn in groepen van 2 herkenbaar	Kamers zijn in groepen van meer dan 2 herkenbaar
Betrouwbaar	> 90% correcte positionering	> 80% correcte positionering	< 80% correcte positionering
Prijs (100 scanners, 1000 tags)	< €15000,-	< €30000,-	> €30000,-
Schaalbaarheid en uitbreidbaarheid	Gaat niet ten koste van de complexiteit van het netwerk	Gaat ten koste van de complexiteit	Schalen is niet mogelijk
Onderhoudskosten (1000 tags, per jaar)	< €400,-	< €1000,-	> €1000,-

Tabel 3 - Criteria communicatietechnieken

RFID: Passieve tags gebruiken geen stroom en actieve tags gaan gemakkelijk jaren mee op een kleine batterij. Door bij elke deuropening een scanner te plaatsen kan lokalisatie op kamernauwkeurigheid behaald worden. Scanners zijn betrouwbaar. De prijs van scanners ligt rond de 200 Euro, wat RFID vrij duur maakt. Er kunnen gemakkelijk scanners worden bijgezet en weggehaald voor een respectievelijk meer en minder nauwkeurige lokalisatie. De complexiteit van RFID is beperkt.

BlueTooth: De tags gebruiken kleine batterijen waarmee ze een aantal jaren kunnen werken. Door in elke kamer een Bluetooth scanner te hangen is kamernauwkeurigheid voor lokalisatie te behalen. Zowel tags als scanners zijn niet duur. Bluetooth is echter niet schaalbaar doordat het aantal apparaten binnen een netwerk zeer gelimiteerd is. Hierdoor moeten netwerken aan elkaar gekoppeld worden waardoor de complexiteit van het systeem toeneemt.

Magnetische polarisatie: Het uitlezen van de sensor is zeer energiezuinig en door in elke kamer een referentiemeting te maken, is kamernauwkeurigheid mogelijk. Referentiemetingen zijn echter niet betrouwbaar, doordat deze metingen veel op elkaar kunnen lijken en omdat het magnetisch veld wordt beïnvloed door ijzeren objecten. De prijs van sensoren ligt laag en de complexiteit van het netwerk is ook zeer beperkt. Verder vereist deze techniek alsnog een netwerk om de sensorinformatie naar een centraal verwerkingspunt te sturen.

ZigBee: Tags kunnen op kleine batterijen gemakkelijk een jaar mee gaan en door in elke kamer een router op te hangen kan kamernauwkeurigheid worden bereikt. Een ZigBee netwerk is betrouwbaar en de prijs van routers en tags ligt laag. Het netwerk is zelfhelend en nieuwe routers kunnen gemakkelijk worden toegevoegd voor uitbreiding. De complexiteit is echter een probleem binnen ZigBee, omdat de data bij een centraal punt moet worden verzameld (coördinator). De doorvoersnelheid van (868MHz of 2.4GHZ) ZigBee is dan te laag om veel tags te beheren. Dit kan worden opgelost door zogenoemde ZigBee gateways toe te voegen aan het netwerk. Deze zijn in staat om verzamelde data door te sturen naar een server. Hiermee wordt de coördinator ontlast. Deze ZigBee gateways zijn echter zeer duur.

Ultrasoon: Een ultrasone ontvanger/zender heeft weinig stroom nodig om te werken en door in elke kamer een scanner te hangen kan kamernauwkeurigheid worden bereikt voor lokalisatie. Scanners zijn zeer betrouwbaar, maar omdat de techniek alleen werkt met "line-of-sight" (de zender en ontvanger moeten elkaar rechtstreeks kunnen 'zien') kan, als deze lijn wordt onderbroken, niet worden gelokaliseerd. De prijs van zenders en ontvangers ligt laag en het netwerk kan gemakkelijk geschaald worden door scanners toe te voegen. Ook de complexiteit van het netwerk is laag.

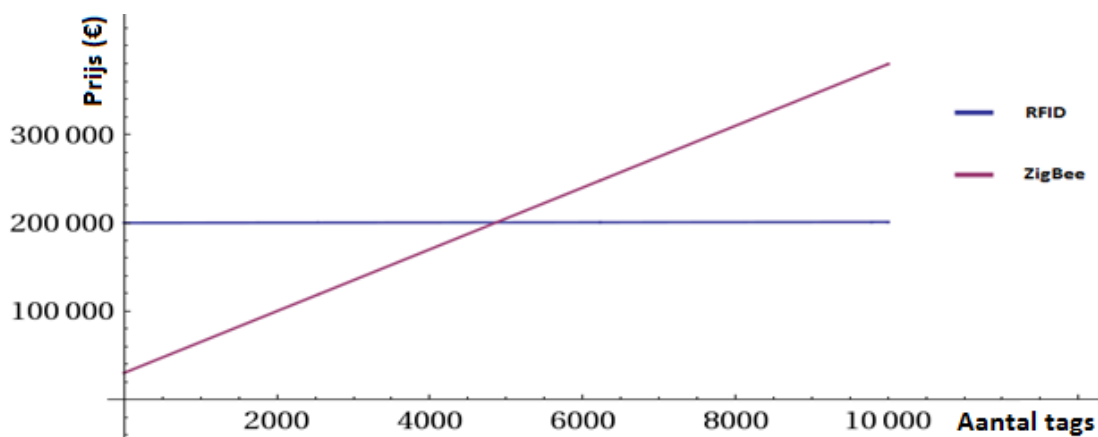
Wi-Fi: Zoals al te merken is op huidige mobiele telefoons, is Wi-Fi zeer energie-onzuinig, wel kan kamernauwkeurigheid worden bereikt door bijvoorbeeld een zender in elke kamer te hangen. Wi-Fi is echter niet echt betrouwbaar door de grote schommelingen in zend/ontvang signaalsterkten. Wi-Fi zenders en ontvangers zijn niet duur en er kunnen extra zenders geplaatst worden voor een grotere dekking van het systeem. Ook de complexiteit van het systeem blijft laag.

De bruikbare technieken zijn beoordeeld op de gestelde eisen (zie tabel 4). De "score" is door middel van kleuren aangegeven (wit geeft aan dat deze afhankelijk is van een keuze voor het communicatienetwerk). De "score" van elk criterium punt uit tabel 3 is door middel van een kleur aangegeven; groen (goed), geel (matig) en rood (slecht). De resultaten zijn in tabel 4 weergegeven.

Techniek/Eis	RFID	BT	Magnetische polarisatie	ZigBee	Ultrasoon	Wi-Fi	IR
Energiezuinig							
Kamernauwkeurigheid							
Betrouwbaar							
Prijs							
Schaalbaarheid en uitbreidbaarheid							
Onderhoudskosten							

Tabel 4 - Beoordeling bruikbare technieken

De twee beste scores zijn die van RFID en ZigBee. Voor het behalen van kamernauwkeurigheid moeten binnen de zorginstelling een aantal referentiepunten worden geplaatst (meestal gelijk aan het aantal kamers). Ook moet op alle zorgmiddelen een tag worden aangebracht, waardoor het systeem ze kan "zien". Er is een grafiek gemaakt (zie figuur 6) waarin de prijs van ZigBee en RFID wordt getoond. De x-as toont het aantal zorgmiddelen dat moet worden gevolgd; de y-as toont de aanschafkosten van de benodigde referentiepunten plus de kosten voor de tags die moeten worden aangebracht op de zorgmiddelen. In de grafiek wordt uitgegaan van 1000 referentiepunten. Zoals te zien is, is RFID in aanschaf vele malen duurder, maar wordt (omdat de tags weinig kosten) ZigBee duurder naarmate meer zorgmiddelen in de ARTZ moeten worden opgenomen. ZigBee wordt duurder bij ongeveer 5000 zorgmiddelen. Bij de gemiddelde zorginstelling, waar het aantal referentiepunten lager ligt, ligt het snijpunt van de prijslijn van RFID met ZigBee vele malen lager (voor 500 referentiepunten is dit rond de 2500 zorgmiddelen).



Figuur 6 – Kosten ZigBee ten opzichte van RFID.

Omdat de tool in de toekomst in een zorginstelling gaat functioneren, is kort onderzoek gedaan naar de grootte van zorginstellingen; Er is gekeken naar twee grote ziekenhuizen, het Universitair Medisch Centrum Utrecht (UMCU) en het Erasmus Medisch Centrum (EMC). Via het zoeken met zoekmachines zijn de volgende feiten gevonden:

EMC [10] [11]:

- Circa 500 patiëntenkamers (locatie centrum).
- Ongeveer 1.100 bedden.
- Ongeveer 12.000 medewerkers.
- Beschikt over circa 400.000m².
- Dagelijks ongeveer 1.500 patiënten voor poliklinische behandeling.

UMCU [12] [13] [14]:

- Circa 350 patiëntenkamers (locatie AZU).
- Ongeveer 1.000 bedden.
- Ongeveer 11.000 medewerkers.
- Bestaat uit 20 gebouwen (gemiddeld 5 tot 8 verdiepingen per gebouw).
- Beschikt in totaal over circa 337.500m².
- Dagelijks ongeveer 1.450 patiënten voor poliklinische behandeling.
- Ongeveer 33.200 opnames per jaar.
- Ongeveer 23.000 operaties per jaar.

De resultaten uit figuur 6 zijn besproken met de klant, waarna is besloten om RFID te gebruiken. Deze keuze is gemaakt omdat de uiteindelijke kosten voor het registreren en lokaliseren minder sterk groeit naar mate het aantal zorgmiddelen stijgt. Vanwege de goedkope tags is de stap om meer dan alleen zorgmiddelen (bijvoorbeeld personeel) te volgen ook stukken kleiner.

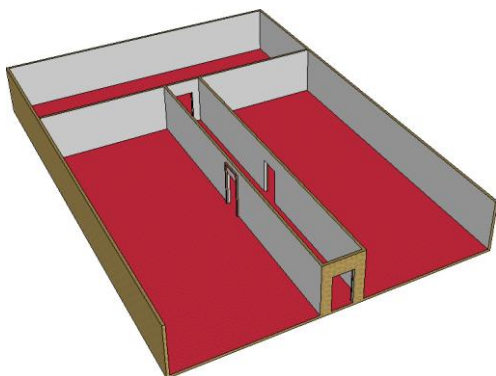
Omdat voor RFID als techniek is gekozen hoeft het onderzoek naar locatiebepaling niet uitgevoerd te worden. Dit omdat bij RFID de locatie van het zorgmiddel gelijk is aan de locatie van waar de RFID scanner zich bevindt.

Dit onderzoek heeft de onderzoeksvraag: “*Wat is een/zijn geschikte techniek(en) voor het registreren van zorgmiddelen?*” beantwoordt. Doordat de keuze van de te gebruiken techniek is gemaakt, kan onderzocht worden welke ontwikkelplatformen (zie paragraaf 5.3) nodig zijn voor de realisatie van de ARTZ.

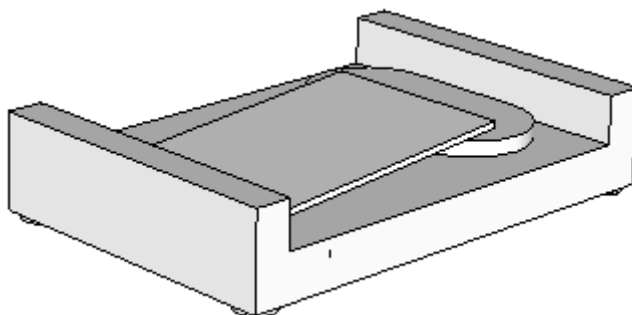
5.3 ONTWIKKELPLATFORMEN

De onderzoeksvraag die in dit onderzoek wordt beantwoord is: "*Welk(e) ontwikkelplatform(en) zijn nodig voor de ontwikkeling van de ARTZ?*". Het beantwoorden van deze onderzoeksvraag zorgt ervoor dat een keuze gemaakt kan worden voor de hardware die gebruikt gaat worden.

Zoals het onderzoek in paragraaf 5.2 beschrijft is ervoor gekozen om gebruik te maken van RFID als draadloze communicatietechniek. Voor het automatisch registreren van zorgmiddelen zijn RFID scanners nodig. Na overleg met de klant is besloten dat scanners op normale schaal te duur en te groot zijn (RFID scanners zijn al snel een kwart meter breed en hoog) om te gebruiken voor een proof-of-concept. Daarnaast wil de klant het proof-of-concept graag ter demonstratie gebruiken bij potentiële klanten. Daarom is in overleg met de klant besloten om een schaalmodel te maken. Voor het schaalmodel is een ontwerp gemaakt in Sketchup (een tool om 3D modellen te maken). Het schaalmodel bevat 3 kamers en een gang (zie figuur 7).



Figuur 7 - Ontwerp schaalmodel



Figuur 8 - Ontwerp zorgmiddel op schaal

Het schaalmodel voor een zorgmiddel (bed) is te zien in figuur 8. De schaal is in vergelijking met het gebouw klein genoeg om door de deuren heen te kunnen.

5.3.1 RDM880 RFID MODULE

Omdat RFID scanners met het bereik van enkele meters een te groot formaat hebben voor een (draagbaar) schaalmodel, moest er een kleiner alternatief gevonden worden. Dit gaf echter nog teveel mogelijkheden om een definitieve beslissing te kunnen nemen voor een RFID module. Omdat een schaalmodel gemaakt zou worden, was de veelgebruikte creditcard grootte tag te groot en was het van belang dat de tags met de grootte van het model mee schaalden.

Na onderzoek bleken er veel verschillende tags beschikbaar te zijn, waaronder veel kleinere, zoals de laundry tag (zie figuur 9) die zo groot is als een knoop (diameter van 13 millimeter en een dikte van 2 millimeter).



Figuur 9 – Laundry tag

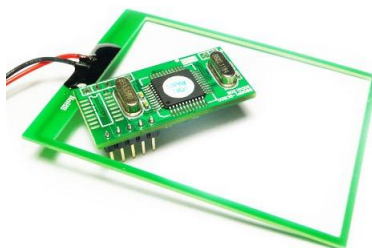
Deze tag werkt alleen op de 13.56MHz frequentieband. RFID scanners zijn over het algemeen, in tegenstelling tot RFID tags, dure apparaten. Omdat het schaalmodel kleine RFID scanners met een kleine leesafstand vereist, zijn de kosten laag te houden. Een andere maatregel die genomen is om kosten te besparen is het bestellen in slechts één of enkele webshops (in verband met verzendkosten).

Voor de webshops heeft Tass Technology Solutions bij de start van het project aangegeven geen bestellingen te willen plaatsen bij onbetrouwbare webshops. Om deze reden is niet alleen besloten om de bestellingen bij Nederlandse (en/of buurlanden) te plaatsen, maar ook voor elke webshop reviews te lezen van andere kopers.

Ook was elk stuk hardware (voor het schaalmodel) nodig voordat begonnen kon worden met het ontwikkelen van de registratie van RFID tags. Doordat bij weinig verschillende webshops bestellingen zijn geplaatst, al deze webshops dichtbij Nederland gevestigd zijn en alle hardware op voorraad was op het moment van bestelling, was de kans dat een levering lang zou duren minimaal.

Om de kans voor een miskoop voor een RFID scanner zo laag mogelijk te houden zijn eerst reviews gelezen van kopers om zeker te weten dat de scanners makkelijk aangesloten en aangestuurd konden worden. Uit reviews bleek dat sommige RFID scanners en de antenne aan dezelfde printplaat bevestigd zijn, waardoor de antenne alleen gewijzigd kon worden door snoertjes in deze printplaat door te snijden en vast te maken aan een andere antenne. Hierdoor ontstond de kans dat bij een fout de gehele RFID scanner onbruikbaar zou worden.

De afbakening van de voorgaande eisen hebben geleid tot het vinden van de RDM880 RFID reader/writer module, die wordt gebruikt voor het proof-of-concept (zie figuur 10).



Figuur 10 – RDM880 RFID reader/writer module.

De module werkt op 13.56MHz en heeft een externe antenne, die rondom de deurpost kan worden gemonteerd. Hij voldoet aan de ISO/IEC 14443 type A standaarden wat overeen komt met de gekozen tags. De RDM880 RFID module geeft zijn output via een TTL Interface. De specificaties van de RDM880 RFID module zijn te zien in tabel 5.

Type	Specificatie
Read/Write chip	MFRC500 / MFRC400 / MFRC531 / MFRC632
Frequentie	13.56MHz
Baud Rate	9600-115200bit/s (configureerbaar, default 9600)
Voltage	5V
Ampère	<70mA
Schrijf/Lees- afstand	30-100mm
Interface	TTL
Afmetingen	39*19*9 mm

Tabel 5 - Specificaties RDM880 RFID module

De module is via een TTL interface aan te spreken door middel van commando's die gespecificeerd staan in de datasheet [15]. De module werkt op basis van een request-response principe. Het formaat van een request is te zien in tabel 6.

STX	STATION_ID	DATA_LENGTH	CMD	DATA[0..N]	BCC	ETX
-----	------------	-------------	-----	------------	-----	-----

Tabel 6 - RDM880 RFID module request

De betekenis van de velden is beschreven in tabel 7.

Veld	Betekenis
STX	"Start of text", geeft de start van een pakket aan.
STATION_ID	Is het adres van de RDM880 RFID module.
DATA_LENGTH	Is de totale lengte van de data die in het pakket zit. Hierin is het CMD veld ook meegerekend.
CMD	Is het commando dat de module moet uitvoeren.
DATA [0..N]	Is een reeks van bytes met een variabele lengte (afhankelijk van het commando). Ook zijn er commando's waar geen data aan meegegeven hoeft te worden.
BCC	<p>Is een checksum. De initiële checksum wordt berekend door een XOR operatie met de STATION_ID, de DATA_LENGTH en het CMD veld uit te voeren.</p> $BCC = STATION_ID \text{ XOR } DATA_LENGTH \text{ XOR } CMD$ <p>De checksum wordt verder berekend door een XOR operatie met de huidige checksum en alle bytes van het DATA veld uit te voeren.</p> $BCC = BCC \text{ XOR } DATA[0..N]$
ETX	"End of text", geeft het einde van een pakket aan.

Tabel 7 - Betekenis RDM880 RFID module request velden

Het formaat van een response is bijna hetzelfde als de request. Echter bij de response is het CMD veld vervangen door het STATUS veld. Het STATUS veld geeft aan of het commando succesvol is uitgevoerd of dat een fout is opgetreden (bijvoorbeeld 0x01 als een niet bestaand commando is geprobeerd uit te voeren).

Als laatste is op internet gezocht naar voorbeeldtoepassingen [16] om direct functionaliteiten te kunnen testen zodra de bestelling (van de RFID scanners) arriveerde.

5.3.2 ARDUINO MEGA

Om de RDM880 RFID module van stroom te voorzien en aan te kunnen sturen, is gekeken naar een prototyping platform met de volgende eisen:

Eis	Criterium
Gebruiksgemak	Handleidingen en voorbeeld code aanwezig om te kunnen starten.
Community	Meer dan 50.000 leden op het forum.
Prijs	Niet meer dan €50,-.
Aansturen meerdere RDM880 RFID module	Minimaal in staat zijn om vier RDM880 RFID module aan te sturen.

Er is gekeken naar een Raspberry PI en Arduino. De prototyping platformen zijn beoordeeld op de gestelde eisen (zie tabel 8). De "score" is door middel van kleuren aangegeven; groen (voldoet wel) en rood (voldoet niet).

Platform/Eis	Raspberry PI	Arduino Uno	Arduino Mega
Gebruiksgemak			
Community			
Prijs			
Aansturen meerdere RDM880 RFID module			

Tabel 8 – Beoordeling prototyping platformen

De beste score is de Arduino Mega (zie figuur 11). Het volledige elektrisch schema van een Arduino Mega is te vinden op de Arduino website [15].



Figuur 11 – Arduino Mega.

De Arduino Mega heeft een 5V uitgang die in staat is om 500mA te leveren. Om meerdere module van stroom te kunnen voorzien moeten ze parallel worden geschakeld, omdat in een parallelschakeling geldt:

$$U_t = U_1 = U_2 = U_n \text{ en } I_t = I_1 + I_2 + I_n$$

Hierdoor krijgen de module de benodigde 5V en omdat ze per module maximaal 70mA van de beschikbare 500mA gebruiken, kunnen er maximaal 7 worden gevoed.

De RDM880 RFID modules hebben input en output via hun TTL interface, deze kunnen worden aangesloten op de seriële poorten van de Arduino Mega:

- Serieel 0: 0 (RX) en 1 (TX)
- Serieel 1: 19 (RX) en 18 (TX)
- Serieel 2: 17 (RX) en 16 (TX)
- Serieel 3: 15 (RX) en 14 (TX)

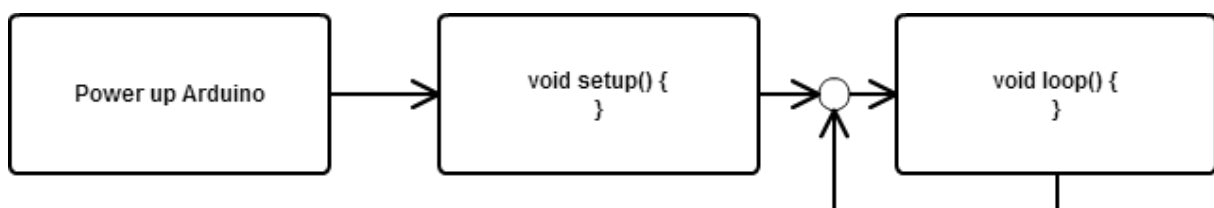
Serieel 0 is echter met de FTDI USB naar serieel chip (seriële communicatie via USB) van de Arduino verbonden. Daarom kan deze niet gebruikt worden (behalve voor communicatie met een PC) indien de Arduino gevoed wordt via USB.

De RDM880 RFID modules kunnen ook op door software gesimuleerde seriële poorten worden aangesloten (bijvoorbeeld GPIO 10 en GPIO 11). Het is mogelijk om meerdere software gesimuleerde seriële poorten te gebruiken, echter is het niet mogelijk er tegelijkertijd naar te luisteren. Dit heeft als gevolg dat binnenkomende data mogelijk verloren gaat.

5.3.2.1 PROGRAMMATUUR ARDUINO

Een Arduino maakt gebruik van wat een "sketch" wordt genoemd. Een sketch is de naam voor de code die wordt geüpload en draait op een Arduino. De flow van de werking van een sketch voor de Arduino is afgebeeld in figuur 12.

De set-up functie wordt aan het begin van het programma eenmalig aangeroepen, hiermee wordt het programma geïnitieerd (bijvoorbeeld variabelen een waarde geven). Nadat het programma geïnitieerd is wordt de loop functie aangeroepen, deze voert code uit in een oneindige for-lus.



Figuur 12 - Arduino flow diagram.

Het is mogelijk om een library te maken in plaats van alle code in de sketch te zetten. Een library bestaat uit C++ code bestanden (.h, .cpp) en kan worden geïmporteerd in de sketch. Hierdoor zijn de in de library ontwikkelde functionaliteiten te gebruiken binnen de sketch. De Arduino community heeft een tutorial gemaakt waarin het schrijven van een eigen library wordt uitgelegd [16].

5.4 ONTWIKKELFRAMEWORK

Dit onderzoek was initieel niet ingepland. De onderzoeksvraag die in dit onderzoek wordt beantwoord is: "Welk ontwikkelframework is het meest geschikt voor de ontwikkeling van de gebruikerstool?". Het beantwoorden van deze onderzoeksvraag zorgt ervoor dat een keuze gemaakt kan worden voor het ontwikkelframework voor de gebruikerstool.

Om de ARTZ op elk mogelijk operating system (OS) beschikbaar te maken is er gekozen om dit in de vorm van een webapplicatie te maken. Voor de ontwikkeling van de ARTZ is de wens van de klant om hiervoor een ontwikkelframework te gebruiken om de ontwikkeling te vergemakkelijken. Er is onderzoek gedaan om een keuze voor een framework te maken. De frameworks moesten aan de volgende criteria voldoen:

Eis	Criterium
Prijs	Is gratis.
Gebruikt het "separation of concerns" [17] principe	Houdt zich aan dit principe voor herbruikbaarheid en splitsing van logica.
Leercurve	Leercurve, voor beginners zonder enige kennis van de programmeertaal (van het framework), ligt laag.
Ervaring	Er is ervaring met het framework (of de programmeertaal waarmee dit framework gebruikt wordt) vanuit het team of het bedrijf.

De frameworks zijn getest door middel van een praktijktest. Bij deze test is een applicatie ontwikkeld door middel van een handleiding voor het framework. Wanneer de prijs van het framework niet aan het bijbehorende criterium voldoet, dan is dit framework direct geen optie meer. Aan het begin is een selectie van de te onderzoeken frameworks gemaakt. De ontwikkelframework die in de selectie zitten zijn gekozen op basis van de kennis en ervaring die binnen het projectteam en bij de klant aanwezig is.

Sencha Ext JS is een framework dat is gebouwd op puur Javascript (JS). Het is bedoeld voor het ontwikkelen van interactieve webapplicaties door gebruik te maken van technieken als Ajax, DHTML en DOM. Na kort onderzoek bleek dat dit geen optie was, omdat elke licentie van dit framework 995 Dollar kost [18] (één licentie per teamlid is vereist).



Symfony is een webapplicatie framework dat is geschreven in PHP. Het volgt het model-view-controller (MVC) patroon en richt zich op het versnellen van de ontwikkeling van een webapplicatie. Tijdens het maken van de testapplicatie [19] is gebleken dat het een goed framework is om mee te gaan werken, omdat het framework niet alleen het MVC-patroon volgt, maar ook omdat het veel handige functies met zich meebrengt zoals het automatisch opbouwen en bijwerken van een databasestructuur. Ook is gebleken dat het ontwikkelen beter kan worden uitgevoerd onder Linux in plaats van Windows. De oorzaak hiervan is dat veel documentatie op het internet is geschreven voor Linux en in korte tijd veel problemen zijn ondervonden met het draaien en onderhouden van de testapplicatie.



Ruby on Rails is een open source webapplicatie framework. Applicaties gemaakt met Ruby on Rails worden ontwikkeld in de programmeertaal Ruby en dwingen het gebruik van het MVC-patroon af. Het streeft naar eenvoud en het ontwikkelen van interactieve applicaties met minder code en minder configuratie dan andere frameworks.



Tijdens het maken van de testapplicatie [20] is Ruby on Rails zeer goed bevallen, ook al was er nog geen kennis van de programmeertaal Ruby. Net als Symfony had ook Ruby on Rails veel handige functionaliteiten voor bijvoorbeeld databaseonderhoud. Toch kostte het ontwikkelen van een applicatie met Ruby on Rails meer tijd, omdat er nog te weinig kennis was van Ruby.

De frameworks zijn beoordeeld op de gestelde eisen (zie tabel 9). De "score" is door middel van kleuren aangegeven; groen (voldoet wel) en rood (voldoet niet).

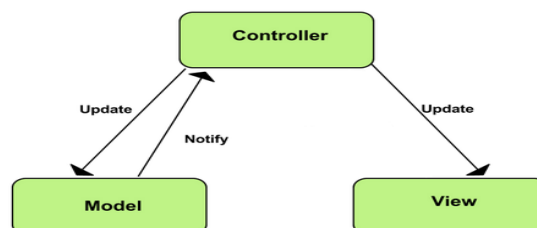
Framework/Eis	Symfony	Sencha Extjs	Ruby on Rails
Prijs	Green	Red	Green
Separation of concerns	Green	Green	Green
Leercurve	Green	Green	Red
Ervaring	Green	Green	Green

Tabel 9 - Beoordeling Frameworks

Na overleg met het bedrijf en de klant is ondanks de betere score van Symfony (welke volgens de beoordeling het beste had kunnen worden gebruikt) gekozen voor Ruby on Rails. Dit omdat het bedrijf al meerdere webapplicaties heeft ontwikkeld met Ruby on Rails en hier ervaring mee heeft. Daarnaast is het bedrijf een framework bovenop Ruby on Rails aan het ontwikkelen. Aangezien de ARTZ in Ruby on Rails wordt gebouwd, kan deze in de toekomst gemakkelijker worden omgezet naar het nieuwe framework.

5.4.1.1 PROGRAMMATUUR RUBY ON RAILS

Ruby on Rails maakt gebruik van het MVC-patroon (zie figuur 13). Het MVC-patroon is een design pattern dat applicaties opdeelt in onderdelen met verschillende verantwoordelijkheden. Er zijn een aantal variaties van het MVC-patroon, waaronder model-view-presenter (MVP) en model view viewmodel (MVVM). Zelfs binnen het MVC-patroon is enige variatie tussen het traditionele MVC-patroon en de moderne interpretatie in verschillende programmeertalen. De volgende uitleg over de modellen, views en controllers is hoe Ruby on Rails het MVC-patroon heeft geïmplementeerd.



Figuur 13 - Het MVC-model.

Een model is een representatie van de data van de applicatie en de een beschrijving van hoe deze gemanipuleerd kan worden. Bij Ruby on Rails worden modellen primair gebruikt voor de interactie met een database tabel. In de meeste gevallen zal elke database tabel gekoppeld zijn aan één model uit de applicatie. [21]

Een view is een representatie van de user interface. In Ruby on Rails zijn views vaak HTML bestanden waarin embedded Ruby code kan worden uitgevoerd. De embedded Ruby code kan ervoor zorgen dat data wordt gepresenteerd. Een klein voorbeeld van een regel embedded Ruby code is te zien in het volgende code segment: [21]

```
<ul>
  <% 3.times do -%>
    <li>list item</li>
  <% end -%>
</ul>
```

Dit resulteert in de volgende HTML code:

```
<ul>
  <li>list item</li>
  <li>list item</li>
  <li>list item</li>
</ul>
```

De view is verantwoordelijk voor het tonen van de data in een webbrowser of een andere client.

Een controller wordt gezien als de "lijm" tussen de modellen en de views. Bij Ruby on Rails zijn controllers verantwoordelijk voor het verwerken van inkomende requests van een webbrowser. Zo worden modellen geraadpleegd om data op te halen en wordt de data aan de views gegeven om de data te kunnen tonen. [21]

5.5 ONTWERP ARTZ

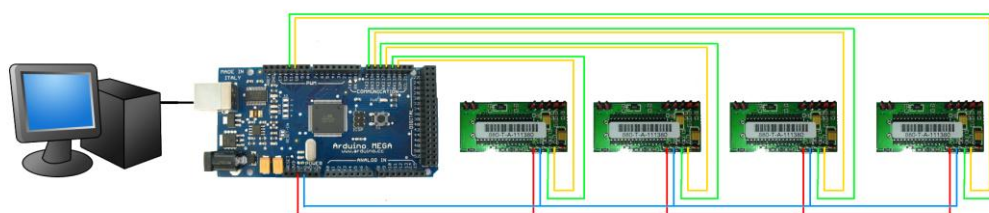
De onderzoeksvraag die in dit onderzoek wordt beantwoord is: *“Wat is een geschikte architectuur voor de ARTZ?”*. Het beantwoorden van deze onderzoeksvraag zorgt ervoor dat een keuze gemaakt wordt voor een architectuur voor de ARTZ.

De wens is het creëren van een flexibele en aanpasbare architectuur. Daarom is het net als het MVC-patroon, dat binnen Ruby on Rails gebruikt wordt, belangrijk dat de architectuur van de ARTZ het *“separation of concerns”* [17] principe toepast. Door dit principe toe te passen kunnen verantwoordelijkheden gescheiden blijven.

Er is gekeken naar de three-tier architectuur [22] die veel in webapplicaties wordt gebruikt. De architectuur beschrijft de splitsing tussen interactie, dataverwerking en logica. Doordat de frontend tier niets van de backend tier hoeft te weten (en andersom) zijn deze onderdelen aan te passen en te vervangen. Het enige gedeelte aangepast moet worden is de middleware tier, omdat deze als de “lijm” functioneert tussen de andere twee tiers. Binnen de ARTZ worden de drie tiers als volgt ingezet:

De frontend tier is de tier waarmee de gebruiker interacteert met behulp van een graphical user interface (GUI). In deze tier draait een webapplicatie op basis van Ruby on Rails. Binnen de webapplicatie is het zoeken en beheren van zorgmiddelen mogelijk indien de gebruiker een geactiveerd gebruikersaccount heeft. Om te zorgen dat deze tier door de middleware tier kan worden benaderd biedt deze via Ruby on Rails een web application programming interface (API). Ook kan door middel van de web-API een andere client gemaakt worden in de vorm van een mobiele app.

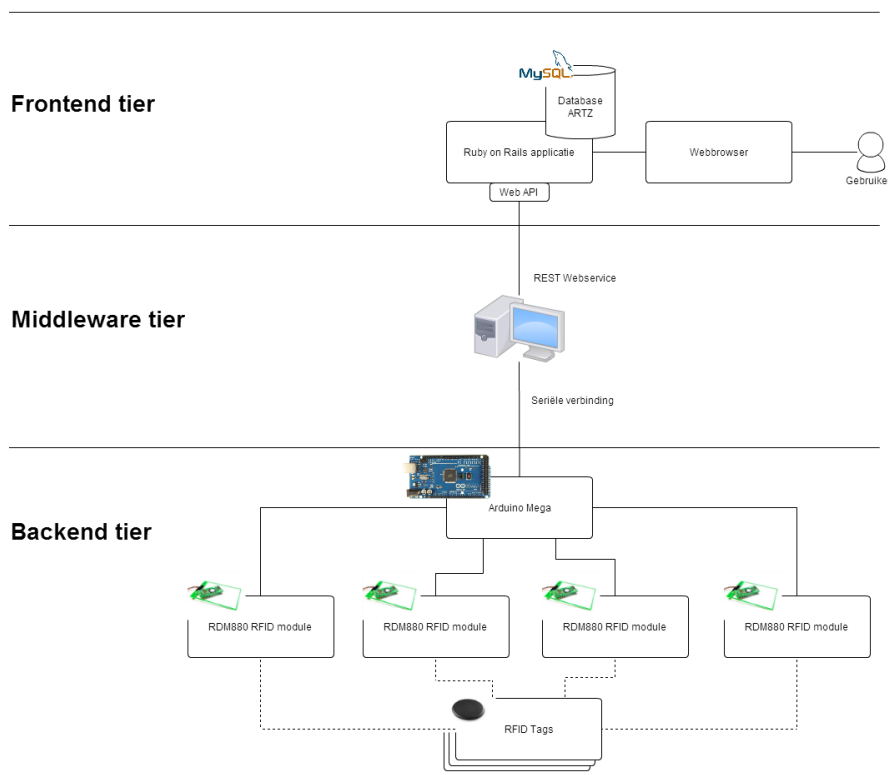
De backend tier is de tier waar de hardware wordt aangestuurd en data wordt verzameld. Zoals in paragraaf 5.3 beschreven is, is de keuze gemaakt voor de RDM880 RFID modulen en de Arduino mega. Er kunnen maximaal zeven modulen worden gevoed. Omdat het schaalmodel drie kamers en één gang bevat, worden er vier RDM880 RFID modulen gebruikt (bij elke deur één module). In de backend tier bevinden zich de vier RDM880 RFID modulen, die zijn aangesloten op serieel 1, 2 en 3 en de laatste op een softwaregesimuleerde seriële poort op GPIO 10 en 11. Wanneer een module een RFID tag registreert wordt dit door middel van de COM over USB interface van de Arduino op een computer verwerkt. Een overzicht van de gehele schakeling is te zien in figuur 14.



Figuur 14 – Overzicht schakeling RDM880 RFID modulen en Arduino Mega.

De middleware tier is de tier die de koppeling legt tussen de frontend tier en backend tier. De middleware tier bestaat uit een Java programma dat de seriële poort uitleest waar de Arduino op aangesloten is, in de middleware tier wordt dit verwerkt en aan de frontend tier doorgegeven via de web-API.

Een overzicht van de totale architectuur is te zien in figuur 15.



Figuur 15 - Overzicht architectuur ARTZ

Een alternatieve architectuur, die veel lijkt op de three-tier architectuur is de "layered" architectuur [23]. Deze architectuur richt zich op het splitsen van functionaliteiten over verschillende lagen binnen één applicatie. Door een goede scheiding tussen verantwoordelijkheden van elke laag aan te brengen wordt de flexibiliteit en onderhoudbaarheid van de applicatie bevorderd.

Een tweede alternatieve architectuur is de "message bus" architectuur [24]. Deze architectuur richt zich op het principe dat software berichten kan ontvangen en verzenden over één gemeenschappelijk communicatiekanaal. De applicaties die gebruik maken van dit communicatiekanaal hoeven geen specifieke informatie te weten over de andere applicaties.

Dit onderzoek heeft de onderzoeksvraag: "Wat is een geschikte architectuur voor de ARTZ?" beantwoordt. Doordat de architectuur van de ARTZ bedacht is kan de daadwerkelijke implementatie worden uitgevoerd (zie hoofdstuk 6).

6 SOFTWAREONTWERP ARTZ

In dit hoofdstuk wordt het ontwerp van de software beschreven. Het ontwerp wordt per tier, die in paragraaf 5.5 is toegelicht, beschreven. Eerst wordt de backend tier beschreven, gevolgd door de frontend tier en de middleware tier.

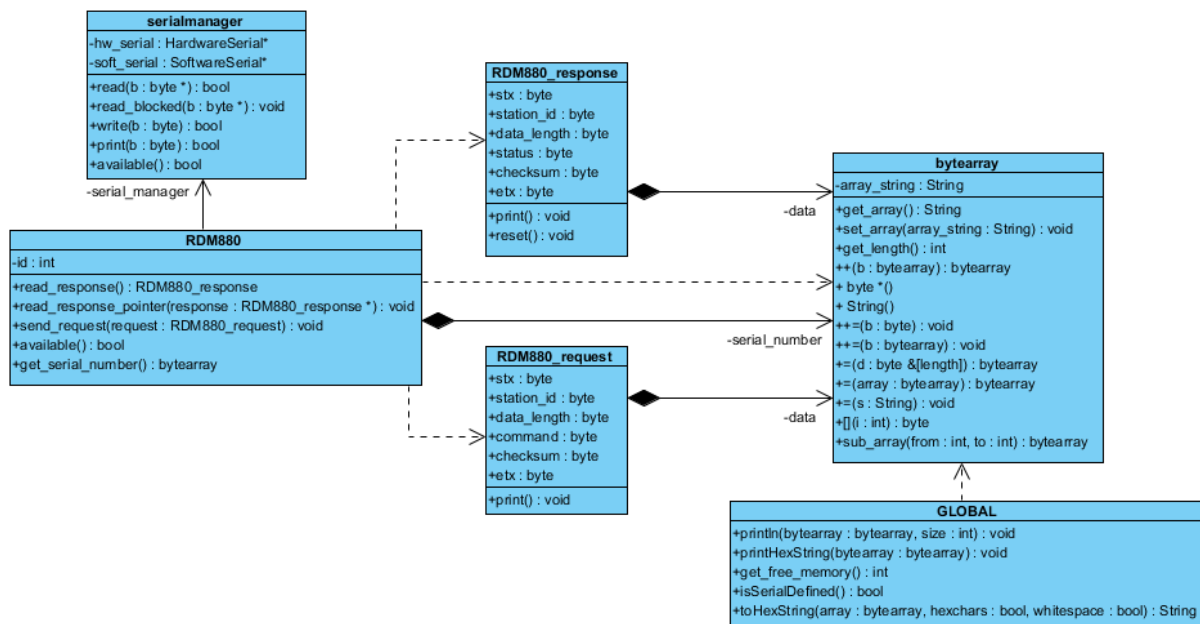
6.1 BACKEND TIER

De backend tier bestaat, zoals beschreven in paragraaf 5.5, uit een Arduino Mega met daarbij vier RDM880 RFID modulen. Voor de RDM880 RFID modulen was het niet nodig om code te schrijven. De Arduino is in staat om commando's via zijn seriële poorten naar de modulen te sturen, die dan intern verwerkt worden.

Om de logica van de software gescheiden te houden van specifieke software voor de RDM880 RFID modulen, is ervoor gekozen om alle specifieke RDM880 RFID module software in een library te plaatsen.

6.1.1 LIBRARY

Het klassendiagram van de library is te zien in figuur 16.



Figuur 16 - Klassendiagram Arduino library

Bytearray is een wrapper klasse rondom de Arduino string klasse. De bytearray klasse doet zich voor als een bytearray (byte [] in C/C++). Het is mogelijk de grootte van de array dynamisch te veranderen. Er zijn veel operatoren geïmplementeerd om het gebruik zo gemakkelijk mogelijk te maken.

GLOBAL is geen klasse, maar bevat hulpfuncties die door het hele programma kunnen worden gebruikt. Één van deze hulpfuncties is een functie die het beschikbare geheugen van de Arduino opvraagt. Hieruit kan worden geconcludeerd dat er een zogeheten "*memory leak*" in de software zit. Als dit optreedt bij de Arduino zal het programma niet langer werken, daarom is het nodig om hier op te controleren (zie subparagraaf 6.1.2).

RDM880_request is een dataobject voor het formaat dat de RDM880 RFID module verwacht (zie subparagraaf 5.3.1). De RDM880_request heeft een compositie met de bytearray klasse. Deze compositie is gemaakt omdat zodra een RDM880_request object vernietigd wordt, dit er ook voor zorgt dat het bytearray object wordt vernietigd. Dit is nodig om zogeheten "*memory leaks*" tegen te gaan.

RDM880_response is een dataobject voor het formaat dat de RDM880 RFID module teruggeeft op basis van een commando (zie subparagraaf 5.3.1). De RDM880_response heeft een compositie met de bytearray klasse. Deze compositie is gemaakt omdat zodra een RDM880_response object vernietigd wordt, dit er ook voor zorgt dat het bytearray object wordt vernietigd. Dit is nodig om zogeheten "*memory leaks*" tegen te gaan.

RDM880 is een object dat een fysieke RDM880 RFID module representeert. Het bevat onder andere het serienummer van de betreffende module en weet hoe hij zijn bijbehorende module kan aanspreken. De RDM880 heeft een compositie met de bytearray klasse. Deze compositie is gemaakt omdat zodra een RDM880 object vernietigd wordt, dit er ook voor zorgt dat het bytearray object wordt vernietigd. Dit is nodig om zogeheten "*memory leaks*" tegen te gaan.

6.1.2 SKETCH

De sketch heeft geen klassendiagram, maar gebruikt de library klassen (zie subparagraaf 6.1.1). De flow die de sketch volgt is in vier stappen weer te geven:

1. Initialiseer vier RDM880 objecten, hier wordt de verbinding met de bijbehorende seriële poort gemaakt en wordt het unieke serienummer van de module opgehaald. Het volgende fragment laat zien hoe de objecten worden geïnitieerd.

```
reader1 = new RDM880(1, 9600);
reader2 = new RDM880(2, 9600);
reader3 = new RDM880(3, 9600);
reader4 = new RDM880(10, 11, 9600);
```

2. Geef de vier RDM880 objecten het commando om te zoeken naar een RFID tag. Het volgende fragment laat zien hoe het commando wordt uitgevoerd.

```
byte GET_SERIAL_NUMBER[] = { MIFARE_GET_SERIAL_NUMBER, 0x26, 0x01 };
RDM880_request request(GET_SERIAL_NUMBER);
reader1->send_request(request);
reader2->send_request(request);
reader3->send_request(request);
reader4->send_request(request);
```

3. Wacht op een event dat optreedt bij een reactie van de RDM880 RFID module. Als data op de seriële poort aanwezig is, wordt binnen de sketch een functie (serialEvent, serialEvent1, serialEvent2 of serialEvent3) uitgevoerd voor de bijbehorende seriële poort.

```
void serialEvent()
{
}
```

4. Verwerk de reactie van de RDM880 RFID module en geef opnieuw het commando om te zoeken naar een RFID tag. Als de status van de response "Ok" was, stuur dan de informatie door over de USB poort. Hierdoor kan de middleware tier beslissen wat met de informatie moet gebeuren.

```
reader1->read_response (&response);
if(response.status == 0x00)
{
    send_to_pc(reader1->get_serial_number());
}
reader1->send_request(request);
```

In de sketch wordt ook gecontroleerd op het geheugengebruik van de Arduino. Door dit te doen is vast te stellen of mogelijke memory leaks aanwezig zijn in het programma. De controle is toegankelijk door gebruik te maken van de `get_free_memory()` functie. In de sketch ziet de controle er als volgt uit:

```
if(get_free_memory() < MEMORY_LEAK)
{
  Serial.println("WARNING POSSIBLE MEMORY LEAK");
}
```

Hierdoor wordt geprint in de console wanneer er ongebruikelijk veel geheugen in gebruik is.

6.2 FRONTEND TIER

De frontend tier bestaat, zoals beschreven in paragraaf 5.5, uit een Ruby on Rails webapplicatie. De folderstructuur van de applicatie ziet er als volgt uit (voor het overzicht zijn de onbelangrijke folders buiten beschouwing gelaten):

- Gui (webroot)
 - app
 - assets
 - images
 - javascripts
 - stylesheets
 - controllers
 - api
 - v1
 - mailers
 - models
 - views
 - config
 - db
 - migrate
 - public
 - uploads

Alle programmacode voor de werking van de functionaliteiten van de applicatie is in een aparte folder in de webroot geplaatst (*app*).

RoR verplicht de programmeur om gebruik te maken van het zogenoemde MVC-model (zie subparagraaf 5.4.1.1). Om deze reden zitten er binnen de *app* folder onder anderen drie folders, namelijk de *models* folder, de *controllers* folder en de *views* folder. Daarnaast bevinden zich hier ook twee andere folders, de *assets* folder en de *mailers* folder. De *assets* folder bevat javascripts, een voor gebruikers niet aanpasbare folder (*images*) met afbeeldingen en stylesheets voor de opmaak van alle webpagina's binnen de webapplicatie. De *mailers* folder bevat programmacode voor het versturen van e-mails zodra bepaalde acties binnen de webapplicatie zijn uitgevoerd.

De *controllers* folder bevat, naast de controllers voor het verwerken van- en het reageren op acties van gebruikers, ook speciale folders voor de ingebouwde API met ondersteuning voor meerdere API versies.

De *views* folder bevat alle programmacode voor het tonen van de informatie die wordt geleverd vanuit de controllers en kan worden opgehaald uit de modellen. Alle *partials* folders bevatten blokken programmacode voor het tonen van delen van de webapplicatie. Er wordt gebruik gemaakt van deze *partials*, omdat binnen de webapplicatie vaak iets bijgewerkt moet worden na bijvoorbeeld een actie van een gebruiker. Door alleen de *partial* bij te werken en niet de hele pagina, wordt de applicatie interactiever, sneller en is minder dataverkeer vereist waardoor de webserver minder zwaar wordt belast. Als laatste is er de *models* folder, die alle modellen van de applicatie bevat.

De *config* folder bevat instellingen met betrekking tot de configuratie van de webserver, waaronder welk database systeem de ARTZ gebruikt en de routes die benodigd zijn voor het koppelen van URL's aan acties binnen controllers en hierbij dus het navigeren van gebruikers.

De *db* folder bevat informatie voor het invoegen van test data in de database(zie paragraaf 6.2.1) en een folder (*migrate*) met informatie over het opzetten van de database.

De *public* folder is een folder die voor alle gebruikers aanpasbaar is via de webapplicatie. Hier staan bijvoorbeeld de afbeeldingen die gebruikers als profielafbeelding gebruiken.

6.2.1 DATABASE

De ARTZ maakt gebruik van het Ruby on Rails framework. Dit framework biedt de mogelijkheid voor gebruikers om gebruik te maken van een communicatielaag tussen de database en de webapplicatie. Deze communicatielaag maakt gebruik van het active record patroon [25]. Door het active record patroon te gebruiken wordt een tabel (of view) gebonden aan een klasse, waardoor een instantie van deze klasse (een object) gekoppeld is aan een rij uit de bijbehorende tabel. Elk object haalt zijn informatie uit de database en zodra het object wordt aangepast, dan wordt de database ook bijgewerkt.

Als voorbeeld kan een tabel genaamd *items* worden gebruikt met de kolommen *name* (tekenreeks) en *price* (numerieke waarde). In het active record patroon is deze tabel geïmplementeerd in de klasse genaamd *Item*. Een nieuwe rij in de *items* tabel kan als volgt worden toegevoegd, pseudo-code:

```
item = new Item()
item.name = "Hammer"
item.price = 12.34
item.save
```

De SQL code voor het bovenstaande stukje pseudo-code zou er dan als volgt uitzien:

```
INSERT INTO items(name,price) VALUES("Hammer", 12.34)
```

Ook kan met behulp van het active record patroon informatie uit de database worden opgehaald, dit kan op onder andere de volgende manier:

```
item = Item.find_first_item("name", "Hammer")
```

Dit haalt de informatie van de rij op met de naam: "Hammer" en verwerkt deze in een nieuw Item object. Het SQL-commando zou hiervoor kunnen zijn:

```
SELECT * FROM items WHERE name = "Hammer" LIMIT 1;
```

Doordat Ruby on Rails gebruik maakt van het active record patroon en dit patroon ervoor zorgt dat de gebruiker werkt met modellen in plaats van met tabellen, is er geen database diagram gemaakt. In plaats van een database diagram is een model diagram gemaakt, dit beschrijft de associaties tussen modellen. De 6 verschillende associatie types [26] die Ruby on Rails ondersteunt zijn:

belongs_to maakt een één-op-één relatie met een ander model, zodat elke instantie van een model dat gebruik maakt van deze associatie, gelinkt is met een instantie van een ander model.

has_one maakt ook een één-op-één relatie met een ander model, maar geeft weer dat elke instantie van een model dat gebruik maakt van deze associatie, een instantie van een ander model bevat of bezit.

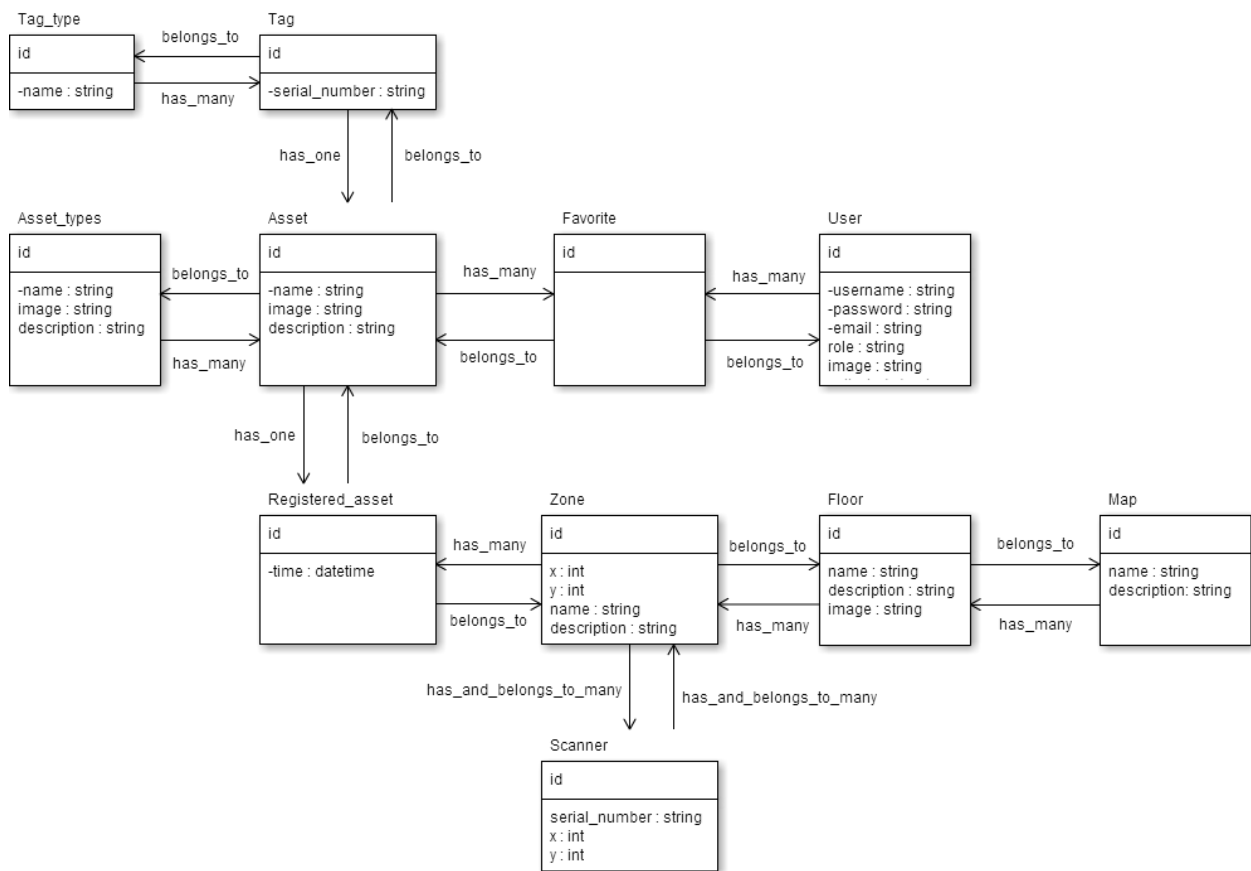
has_many maakt een één-op-veel relatie met een ander model, zodat elke instantie van een model dat gebruik maakt van deze associatie, met nul of meer instanties van een ander model gelinkt is.

has_many :through is een associatie die vaak wordt gebruikt om een veel-op-veel relatie met een ander model te maken. Deze associatie geeft aan dat het model dat gebruik maakt van deze associatie, gelinkt kan worden met nul of meer instanties van een ander model *through* (via) een derde model.

has_one :through maakt een één-op-één relatie met een ander model. Deze associatie geeft aan dat een instantie van een model dat gebruik maakt van deze associatie, gelinkt is met een instantie van een ander model *through* (via) een derde model.

has_and_belongs_to_many is een associatie die een directe veel-op-veel relatie met een ander model maakt, zonder gebruik te maken van een derde (tussenliggend) model.

Het model diagram dat voor de ARTZ is ontworpen, is in figuur 17 te zien.



Figuur 17 - Model diagram

In figuur 17 is onder andere te zien dat een tag (*Tag* model) is verbonden aan een zorgmiddel (*Asset* model). Ook is te zien dat er een apart *Registered_asset* model is gemaakt, deze is gemaakt om een tag een laatst bekende positie (zie link met *Zone* model) te geven. Deze is apart van het *Asset* model, om het ophalen van alle ooit 'geregistreerde' zorgmiddelen te versimpelen.

Daarnaast toont het figuur ook dat plattegronden van verschillende gebouwen en/of vleugels in één database kunnen zitten (zie het *Map* model). Deze gebouwen of vleugels kunnen bestaan uit meerdere verdiepingen (zie het *Floor* model) en deze verdiepingen bestaan uit meerdere zones (zie het *Zone* model). Als laatste is een scanner (*Scanner* model) gekoppeld aan een of meerdere zones.

6.2.2 WEB APPLICATION PROGRAMMING INTERFACE

Om de mogelijkheden van Ruby on Rails te tonen en om de platformafhankelijkheid van de webapplicatie aan te tonen, is er een basisversie van een web-API gemaakt voor de ARTZ.

De web-API bestaat uit vier controllers:

1. **TokensController**, deze controller is verantwoordelijk voor het authenticatieproces. Zodra er ingelogd wordt via de API zal een unieke token worden gegenereerd. De token is nodig om functionaliteiten van de andere controllers te kunnen gebruiken.
2. **FavoritesController**, deze controller is verantwoordelijk voor het ophalen van favorieten van de ingelogde gebruiker, het aanmaken van een nieuwe favorieten en het verwijderen ervan.
3. **UsersController**, deze controller is verantwoordelijk voor het ophalen van de profiel informatie van de ingelogde gebruiker.
4. **AssetRegistrationsController**, deze controller is verantwoordelijk voor het afhandelen van een registratie van een tag.

Deze vier controllers kunnen worden gebruikt door middel van het hypertext transfer protocol (HTTP). Dit is het protocol dat wordt gebruikt tussen webcliënt en webserver. HTTP functioneert als request-response protocol in het cliënt-server model. De cliënt stuurt een HTTP-request waarop de server een HTTP-response teruggeeft. De server kan middelen leveren (als hypertext markup language (HTML) bestanden of andere inhoud) of kan andere functies uitvoeren namens de cliënt.

HTTP-request is een aanvraag van de cliënt aan de server. Deze aanvraag bestaat uit een aantal onderdelen; een requestmethode, een URL en header(s). Als requestmethode zijn een aantal mogelijkheden beschikbaar; POST, GET, PUT, DELETE, HEAD, TRACK, OPTIONS en CONNECT [27]. Al deze requestmethoden hebben een functie, de GET methode is bijvoorbeeld puur voor het ophalen van data over een bepaald item en de DELETE methode voor het verwijderen bepaald item.

HTTP-response is een antwoord op een HTTP-request. Deze bestaat uit een resultaatcode, header(s) en een body (de opgevraagde inhoud). De resultaatcode bestaat uit minimaal drie cijfers en is opgedeeld in groepen op basis van het eerste cijfer [28]:

- 1xx: Een informatieve boodschap van de webserver.
- 2xx: De gevraagde actie succesvol is afgehandeld.
- 3xx: De cliënt door moet worden gestuurd naar een andere locatie.
- 4xx: Er is een fout veroorzaakt door de cliënt.
- 5xx: Er is een fout veroorzaakt door de server.

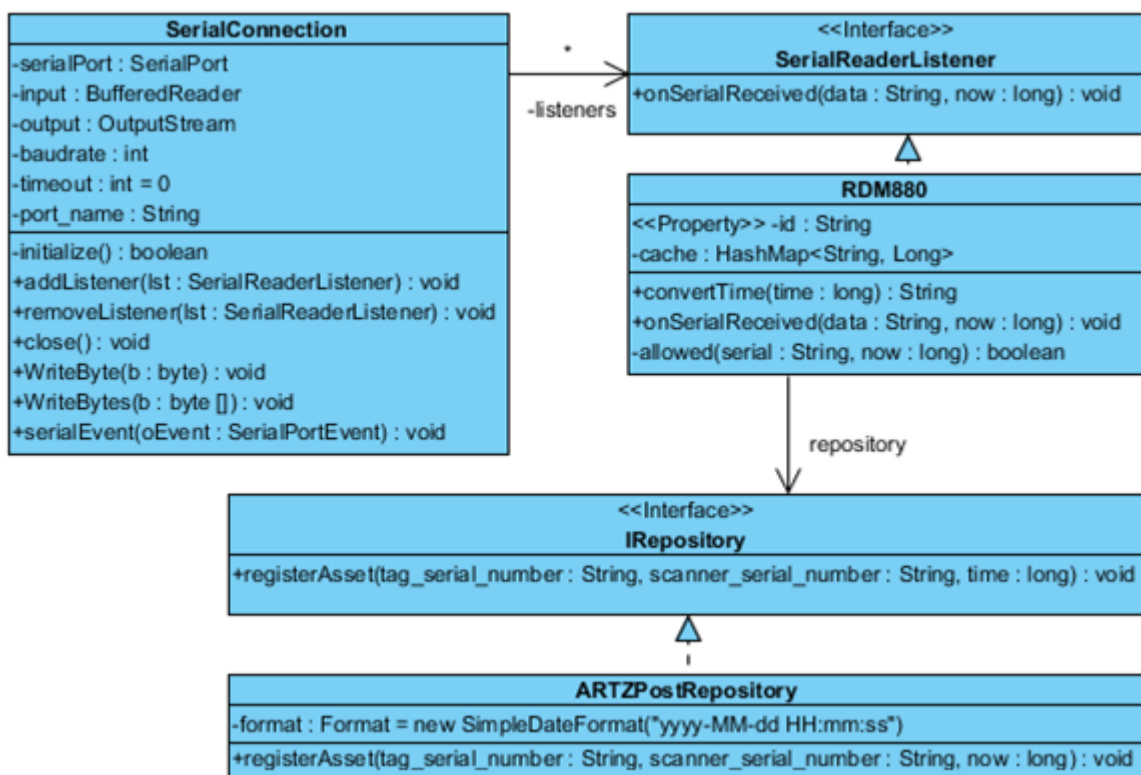
De meest voorkomende resultaatcodes zijn:

- 200 OK - Het opgevraagde document is succesvol opgevraagd.
- 304 Not Modified - De versie in de cache is hetzelfde als de opgevraagde pagina.
- 403 Forbidden - Het opgevraagde document mag niet bekeken worden.
- 404 Not Found - Het opgevraagde document kan niet worden gevonden.
- 405 Not Allowed - De gebruiker is niet gemachtigd het gevraagde document te bekijken.
- 500 Internal Server Error - De opgevraagde actie kon niet worden uitgevoerd.
- 503 Service Temporarily Unavailable - De webserver is tijdelijk in onderhoud.

Voor de ARTZ zijn een aantal HTTP-request methoden geïmplementeerd, dit zijn POST, GET, PUT en DELETE. Deze vier HTTP-request methoden zijn echter niet voor alle web-API controllers geïmplementeerd. Dit komt omdat deze vaak overbodig zijn, de web-API hoeft bijvoorbeeld (nog) niet het verwijderen van gebruikers te ondersteunen (de UsersController bevat dus geen DELETE functionaliteit).

6.3 MIDDLEWARE TIER

De middleware tier bestaat, zoals beschreven in paragraaf 5.5, uit een Java applicatie die de seriële poort uitleest. Voor het uitlezen van de seriële poort wordt gebruik gemaakt van dezelfde library als Arduino gebruikt in zijn IDE die eveneens in Java geschreven is. Een alternatief voor deze library zou kunnen worden gevonden door middel van het zoeken met een zoekmachine. Echter de keuze voor de library die door Arduino wordt gebruikt is gemaakt, omdat deze goed ondersteund is en ook ondersteund blijft. Het klassendiagram van de middleware tier is te zien in figuur 18.



Figuur 18 - Klassendiagram middleware tier

SerialConnection is een klasse die gebruik maakt van de library van Arduino om een verbinding met een seriële poort te openen. Zodra data over de seriële poort binnenkomt wordt de methode `serialEvent` aangeroepen. Hierin worden de listeners die zich bij het object hebben geregistreerd (door middel van de methode `addListener`) gewaarschuwd.

SerialReaderListener is een interface voor het abonneren op events van een **SerialConnection** object. Deze interface is gemaakt zodat de klasse die zich abonneert op events van de seriële poort eenvoudig kan worden veranderd.

RDM880 is een klasse die de `SerialReaderListener` interface implementeert. Door deze interface te implementeren kan hij zichzelf als listener aanmelden bij een `SerialConnection` object. Zodra dit is gedaan wordt hij gewaarschuwd als er een event is geweest. De interface kan als volgt worden gebruikt:

```
IRepository repository = new ARTZPostRepository();  
repository.registerAsset(serial_number, id, now);
```

Door de methode `registerAsset` van de `IRepository` interface uit te voeren blijft de specifieke implementatie verborgen voor de `RDM880` klasse.

IRepository is een interface voor de communicatie naar de frontend tier. Deze interface is gemaakt om ervoor te zorgen dat zodra er op een andere manier met de frontend tier gecommuniceerd moet worden, het mogelijk is een nieuwe klasse te maken die deze interface implementeert. Hierdoor is de klasse verplicht om de `registerAsset` methode een eigen implementatie te geven.

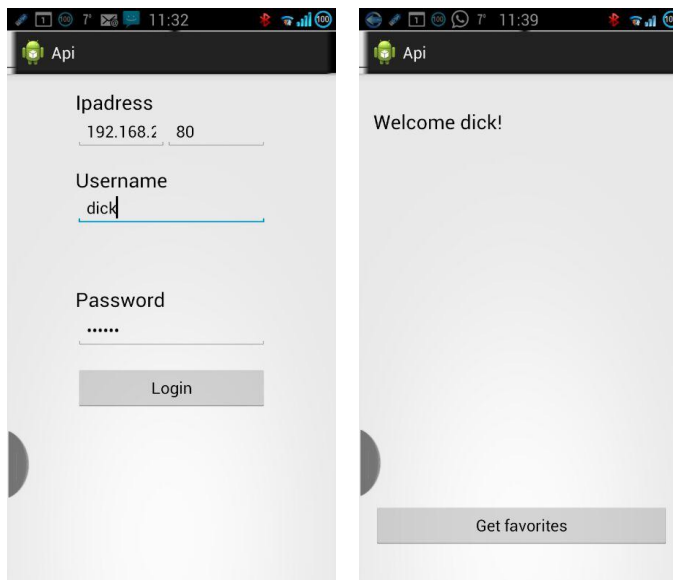
ARTZPostRepository is een klasse die de `IRepository` interface implementeert. De klasse implementeert de `registerAsset()` methode die de `IRepository` interface afdwingt door middel van een HTTP post. Het volgende code fragment laat de implementatie van de `registerAsset()` methode zien:

```
// Constructing the HTTP post for the ARTZ API  
HttpPost http_post;  
http_post = new HttpPost("http://" + ARTZ_IP + ":" + ARTZ_PORT +  
"/api/v1/asset_registrations.json");  
  
// Adding HTTP post headers  
http_post.addHeader("Content-Type", "application/x-www-form-  
urlencoded");  
http_post.addHeader("Accept", "application/json");  
  
// Set parameters needed on the ARTZ API side  
http_post.setEntity(  
new StringEntity("tag_serial_number=" + tag_serial_number +  
"&scanner_serial_number=" + scanner_serial_number, HTTP.UTF_8));  
  
// Open client connection and execute the HTTP post  
HttpClient client = new DefaultHttpClient();  
HttpResponse resp = client.execute(http_post);
```

6.4 MOBIELE APP

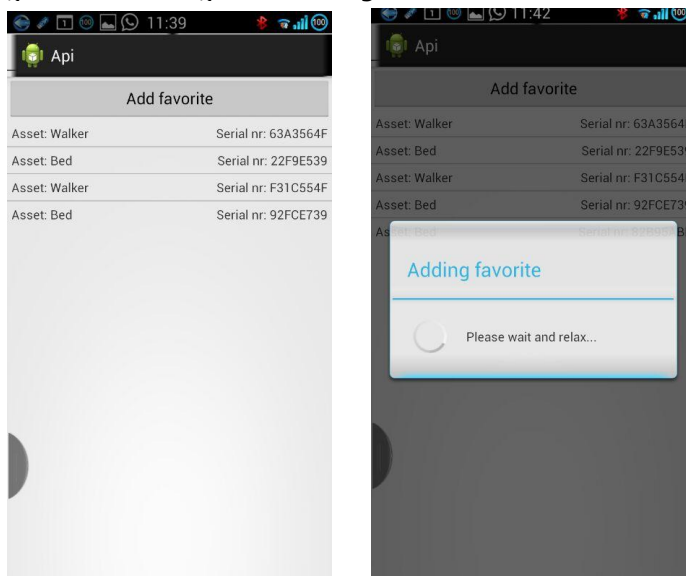
Aan de hand van de functionaliteiten van de web-API van de ARTZ (zie paragraaf 6.2.2) is een Android applicatie ontwikkeld om de werking van deze functionaliteiten te kunnen demonstreren.

De Android app vereist de gebruiker om eerst in te loggen voordat de app verder kan worden gebruikt (zie figuur 20). Hiervoor kunnen de gebruikers gegevens gebruikt worden die ook bij de webapplicatie van de ARTZ worden gebruikt. Zodra de gebruiker is ingelogd wordt het welkomstschermbij de app getoond (zie figuur 19) en de optie om favorieten op te halen met behulp van een knop. Zodra de knop wordt ingedrukt krijgt de gebruiker een nieuw scherm te zien waar een lijst van al zijn huidige favorieten is te zien (zie



Figuur 20 – Inlog scherm Figuur 19 – Welkomstschermbij de app

figuur 21). Daarnaast krijgt de gebruiker de optie om met behulp van NFC in de mobiele telefoon (indien dit ondersteund is) een nieuw favoriet toe te voegen door de telefoon dicht bij de RFID tag te houden (zie figuur 22).



Figuur 21 - Favorieten scherm Figuur 22 - Favoriet toevoegen

7 TESTEN

De ARTZ bestaat uit meerdere hoofdonderdelen die zijn getest. Deze onderdelen zijn:

- Registratie zorgmiddelen.
- Lokalisatie zorgmiddelen.
- Gebruikerstool.

De opstelling voor het testen is zoals het schaalmodel eruit ziet (zie paragraaf 5.3). Voor het testen van elk van de onderdelen is een bijbehorend stappenplan opgezet. Dit stappenplan is opgesteld om ervoor te kunnen zorgen dat zodra er een nieuwe versie van de ARTZ wordt opgeleverd, er verzekerd kan worden dat "oude" functionaliteiten nog werken.

Onderdeel	Stappenplan
Registratie zorgmiddelen	<ol style="list-style-type: none"> 1. Koppel een tag aan een zorgmiddel in de database van de server. 2. Beweeg dit zorgmiddel door de testomgeving en controleer of het systeem het zorgmiddel heeft "gezien". Herhaal dit meerdere malen. 3. Controleer of de informatie van het gedetecteerde zorgmiddel bij de server is aangekomen.
Lokalisatie zorgmiddelen	<ol style="list-style-type: none"> 1. Beweeg een zorgmiddel naar een locatie binnen de testomgeving en controleer of de locatie opgeslagen op de server overeenkomt met de daadwerkelijke locatie.
Gebruikerstool	<ol style="list-style-type: none"> 1. Probeer een nieuw account aan te maken. 2. Probeer met het nog niet geactiveerde account in te loggen. 3. Probeer dit account te activeren in de adminomgeving en probeer nogmaals in te loggen. 4. Probeer nieuwe tags aan te maken. 5. Probeer nieuwe zorgmiddelen aan te maken en koppel deze aan een tag. 6. Probeer een nieuw zorgmiddeltype aan te maken. 7. Probeer een nieuwe map aan te maken. 8. Probeer nieuwe verdiepingen aan de map toe te voegen. 9. Probeer het zoeken van alle zorgmiddelen. 10. Probeer het zoeken van alle zorgmiddelen die als favoriet genoteerd zijn. 11. Probeer het zoeken op zorgmiddeltype. 12. Probeer het opvragen van de gedetailleerde zoekresultaten.

De resultaten van de tests zijn in de bijlagen D, E en F opgenomen.

8 RESULTATEN

Het project bestond uit drie grote onderdelen, de apparatuur en programmatuur voor de registratie van tags, het verwerken van de data afkomstig van de scanners en het tonen van de data aan de gebruiker respectievelijk aangegeven door de backend tier, de middleware tier en de frontend tier.

8.1 DE BACKEND TIER

De backend tier, bevat alle RDM880 RFID modulen en de Arduino Mega voor het aansturen ervan. De Arduino Mega voedt de modulen en stuurt deze aan door middel van commando's (die zijn gespecificeerd in de datasheet [29]). Zodra een tag dichtbij de antenne van een module komt stuurt de module de informatie via TTL naar de Arduino. Deze zal het bericht ontvangen en naar de middleware tier doorsturen door middel van de seriële poort over USB. De backend tier werkt naar behoren en het is mogelijk om deze eventueel compleet te vervangen. Dit is mogelijk door de flexibele architectuur die is gecreëerd.

8.2 DE MIDDLEWARE TIER

De middleware tier bestaat uit een singleboardcomputer of een andere PC met de mogelijkheid tot het uitvoeren van Java code, het ontvangen van seriële data over een seriële poort en het verbinden met het internet (WLAN of WiFi). De PC krijgt data vanaf de backend tier over de seriële poort. Deze data wordt omgezet in data die de frontend tier begrijpt en wordt gestuurd via de web-API van de frontend tier.

8.3 DE FRONTEND TIER

De frontend tier bestaat uit de onderdelen waar een gebruiker direct gebruik van maakt. Er draait de Ruby on Rails applicatie, die gebruikers in staat stelt om zorgmiddelen te zoeken en (indien de gebruiker gemachtigd is) tags, zorgmiddelen en gebruikeraccounts te beheren. Via Ruby on Rails wordt ook een web-API aangeboden. Via deze web-API is een Android app gemaakt waarmee gebruikers kunnen inloggen en via NFC hun favoriete zorgmiddelen kunnen beheren. De frontend tier werkt naar behoren en het is bijvoorbeeld mogelijk om een ander DBMS te gebruiken of te baseren op een ander framework (bijvoorbeeld welke het afstudeerbedrijf ontwikkelt).

8.4 PROBLEMEN

Tijdens het project zijn een aantal problemen opgetreden.

Fout in RDM880 RFID module specificaties

Het eerste probleem was dat er een fout in de datasheet van de RDM880 RFID module zat. De datasheet van de module geeft het volgende over het commando voor het ophalen van het serienummer van een tag [29]:

MF_GET_SNR (0x25)

Request:

Data Field

DATA[0]:	Request mode
0x26 -	Request Idle
0x52 -	Request All

DATA[1]:	00 - do not need to execute the halt command
----------	----------------------------------------------

	01 - need to execute the halt command
--	---------------------------------------

Response:

Data Field

STATUS:	0x00 - OK
DATA[0]:	FLAG
0x00 -	Only one card is in the readable area
0x01 -	At least two cards are in the readable area

DATA[1-4]:	Card Serial Number
------------	--------------------

Description:

The High Level Command integrates the low level commands (Request, AntiColl1, Select) and get the SNR of selected card.

Example:

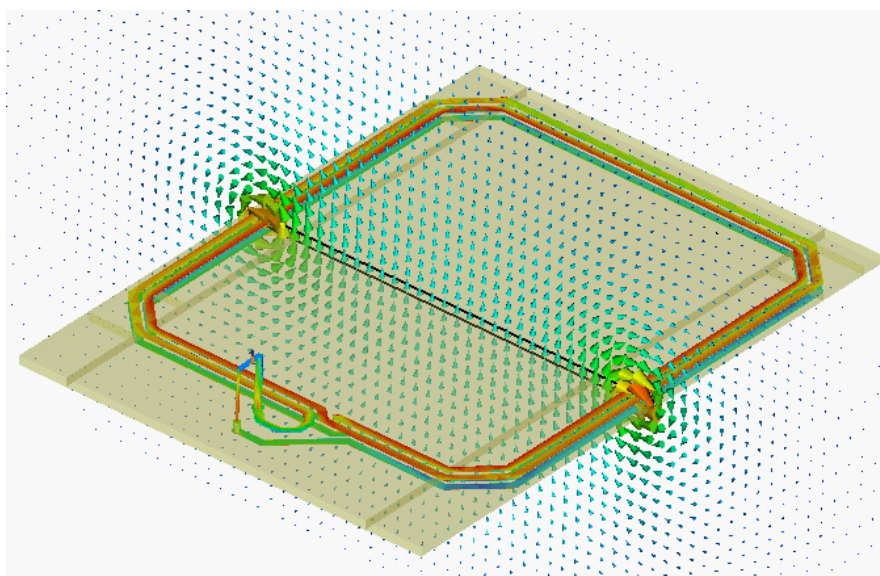
Request Data:	AA 00 03 25 26 00 00 BB
Response Data:	AA 02 06 00 00 16 0F F4 7F 96 BB

Zoals te zien is heeft de request van dit commando (MF_GET_SNR of 0x25 in hexadecimale waarde) twee datavelden. Het eerste veld zorgt ervoor dat alleen het serienummer van een tag wordt opgevraagd als deze niet in de zogeheten "halt-mode" staat. Het tweede dataveld (de vlag) wordt gebruikt om de tag (indien de vlag gezet is) in de "halt-mode" te zetten nadat er een reactie is geweest.

De fout in de datasheet zit in één (of allebei de) dataveld(en). Zodra het eerste dataveld wordt gevuld met 0x26 en het tweede dataveld (de vlag) wordt gezet, zou volgens de datasheet elke tag maar één keer moeten reageren. Bij de eerstvolgende poging voor het uitlezen zou de tag in de "halt-mode" staan, waardoor de RDM880 RFID module deze niet meer uitleest (doordat het eerste dataveld ervoor zorgt dat alleen tags die niet in "halt-mode" staan moeten reageren). Dit is echter niet het geval, de tags lijken er geheel niet op te reageren, het in- en uitschakelen van deze functionaliteit geeft totaal geen veranderingen in het resultaat (de tag blijft reageren met zijn serienummer). Aangezien er op geen andere manier opgevraagd kan worden of een tag in "halt-mode" staat, kan één van of beide datavelden niet of incorrect werken. In hoofdstuk 10 wordt de oplossing voor dit probleem beschreven.

Tag oriëntatie voor registratie

De techniek die is gebruikt voor de ARTZ is passieve RFID. Omdat bij deze techniek geen batterij is verwerkt in de tag, kan deze niet uit zichzelf data versturen. Passive RFID vereist dat de RFID scanners de tags opladen door middel van een magnetisch veld, waardoor ze even genoeg energie hebben om een reactie te versturen. In figuur 23 is met kegeltjes aangegeven hoe de richting van het magnetisch veld staat ten opzichte van de antenne (de vierkantvormige wikkelingen met de afgeronde hoeken in figuur 23) van de scanner (de punt van de kegels geven de richting van het magnetisch veld aan en de grootte geeft de sterkte aan).



Figuur 23 - Magnetisch veld RFID antenne

Om een tag genoeg op te laden zodat deze een reactie kan versturen, moet de antenne van tag haaks op het magnetisch veld staan. Het tweede probleem dat zich voordoet is dat zodra een tag evenwijdig (aan het magnetisch veld) door het magnetisch veld beweegt, deze zichzelf niet genoeg kan opladen om te reageren. In hoofdstuk 10 wordt de oplossing voor dit probleem beschreven.

9 CONCLUSIE

Het doel van de opdracht was om een automatisch plaatsregistratie systeem te ontwerpen en te ontwikkelen. Voor het ontwerpen was onderzoek een belangrijk gedeelte. In paragraaf 3.3 zijn de onderzoeksvragen opgesomd. De antwoorden zullen kort worden toegelicht.

Door informatie over bestaande systemen te verzamelen was het mogelijk om een selectie van draadloze communicatietechnieken te maken die konden worden gebruikt voor de ARTZ. Door een beoordeling te geven aan de technieken kon de onderzoeksvraag: *"Wat is een/zijn geschikte techniek(en) voor het registreren van zorgmiddelen?"* worden beantwoord met RFID is een geschikte techniek. Omdat de techniek bekend was kon onderzocht worden welk algoritme nodig was om kamernauwkeurigheid te behalen. Dit was bij RFID niet nodig omdat de locatie van het zorgmiddel gelijk is aan de locatie van de scanner die de tag ziet.

Om de ARTZ te realiseren was ook RFID hardware nodig. Omdat de klant aangaf dat het om mobiliteitsredenen en kosten handiger was een schaalmodel op te leveren, is in plaats van grote RFID scanners, naar kleine RFID modulen gezocht. Door te zoeken kon de onderzoeksvraag: *"Welk(e) ontwikkelplatform(en) zijn nodig voor de ontwikkeling van de ARTZ?"* worden beantwoord met de RDM880 RFID module en de Arduino Mega.

De architectuur van de ARTZ moest flexibel zijn, mochten bijvoorbeeld de RFID modulen vervangen worden door andere RFID modulen. Door te onderzoeken wat een goede architectuur was om alle onderdelen van het totale systeem van elkaar los te koppelen kon de onderzoeksvraag: *"Wat is een geschikte architectuur voor de ARTZ?"* worden beantwoord met een afgeleide van de three-tier architectuur, waarin de backend tier voor data verzameling is (in plaats van data opslag), de middleware tier voor de logica en de frontend tier voor de dataopslag en de user interface (in plaats van alleen user interface).

Als laatste moest een interface ontwikkeld worden, waarmee gebruikers de ARTZ kunnen raadplegen. Deze interface moest platformonafhankelijk zijn zodat gebruikers deze met vele soorten apparaten kunnen gebruiken. Hier begon een onderzoek dat initieel niet was ingepland. De onderzoeksvraag die in dit onderzoek werd beantwoord was: *"Welk ontwikkelframework is het meest geschikt voor de ontwikkeling van de gebruikerstool?"*. Deze onderzoeksvraag is beantwoord met het Ruby on Rails framework, met daarin een web-API zodat deze toegankelijk is voor bijvoorbeeld native applicaties voor mobiele apparaten.

Door deze onderdelen te combineren is een werkend schaalmodel gemaakt van de ARTZ. In de toekomst kan de hardware worden vervangen door hardware die geschikt is voor de ARTZ op ware grootte. De gebruikerstool (frontend tier) is compleet onafhankelijk van de hardware (backend tier) en hoeft dus niet aangepast te worden zodra wijzigingen worden doorgevoerd op de backend tier. Ook kan de frontend tier vervangen worden zonder dat wijzigingen in de backend tier moeten worden gemaakt. De architectuur is dus flexibel bewezen.

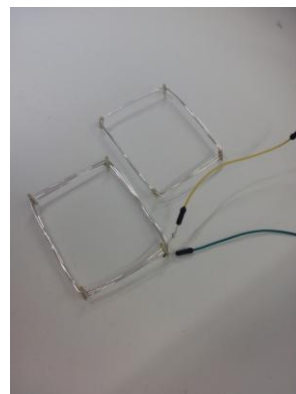
Uit de resultaten is te concluderen dat het project is geslaagd.

10 AANBEVELINGEN

Er zijn een aantal aanbevelingen indien de ARTZ doorontwikkeld gaat worden:

1. **Ware grootte**, de eerste aanbeveling is de backend tier aanpassen voor de ARTZ op ware grootte. Het schaalmodel heeft bewezen dat de ARTZ werkt, maar nog niet bruikbaar in zorginstellingen is. Voordat het systeem bruikbaar is in een zorginstelling moeten bijvoorbeeld de RFID scanners verder kunnen lezen dan de 10 centimeter die de RDM880 RFID module haalt.
2. **RFID modulen**, de RDM880 RFID modulen voldoen niet geheel aan de specificaties uit de datasheet (zie paragraaf 8.4), hierdoor is het niet mogelijk om tags als "gezien" te markeren en zo meerdere tags te gelijk te herkennen. Met andere RFID modulen is dit probleem te verhelpen
3. **RFID antennes**, de derde aanbeveling is het vinden en/of aanpassen van de RFID antennes om het werkgebied van deze antennes te veranderen. Dit werkgebied is, zoals beschreven in paragraaf 8.4 een probleem, omdat gebruik gemaakt wordt van passieve tags. Passieve tags moeten worden opgeladen door het magnetisch veld van de scanner. Zodra de tag incorrect georiënteerd is, heeft deze een grote kans om niet genoeg te worden opgeladen om een reactie te kunnen sturen naar de scanner voor zijn aanwezigheid.
4. **De web-API**, de web-API van de Ruby on Rails applicatie kan worden uitgebreid om meer functionaliteiten te bieden. Bijvoorbeeld het toevoegen van een nieuwe tag zou in de toekomst met de API kunnen worden ondersteund.
5. **Mobiele app**, tijdens het project is een start gemaakt met een native Android app om te laten zien dat het systeem ook dit kan ondersteunen als cliënt. De app kan worden uitgebreid om het bijvoorbeeld voor beheerders mogelijk te maken tags toe te voegen, om zorgmiddelen te zoeken via de app en zelfs te navigeren naar een geselecteerd zorgmiddel. Om deze mobiele app te kunnen uitbreiden moet mogelijk ook de web-API worden uitgebreid.
6. **Uitbreiden webapplicatie**, als laatste kan de webapplicatie natuurlijk altijd verder uitgebreid worden, met bijvoorbeeld navigatie.

Voor aanbeveling 3 is een poging gedaan om zelf antennes te maken (zie figuur 24). Deze poging heeft echter geen werkend resultaat opgeleverd. De RDM880 RFID modulen bleken direct te stoppen met scannen zodra de aangepaste antenne werd aangesloten. Dit komt waarschijnlijk omdat er nog mogelijk componenten verstopt zitten in de oorspronkelijke antennes om het signaal te modificeren.



Figuur 24 - Zelfgemaakte antennes

11 EVALUATIE

Tot aan het opleveren van dit document heeft het project 15 weken geduurd en is de laatste fase van de ontwikkeling bereikt. Terugkijkend op de afgelopen periode is het erg goed gegaan. Er zijn geen noemenswaardige tegenslagen geweest. Een belangrijke les die geleerd is tijdens de opleiding is dat het van cruciaal belang is om helder te krijgen wat de klant precies wil, omdat dit soms nog niet duidelijk is.

Tijdens gesprekken met de klant zijn aan de hand van tevoren bedachte vragen, de wensen van de klant in kaart gebracht. Zodra de wensen duidelijk waren konden hierbij de onderzoeken bedacht worden die nodig waren.

De wensen en onderzoeken konden verwerkt worden in het plan van aanpak, dat ook een globale planning van de afstudeerperiode bevat. Het schrijven van het plan van aanpak is verdeeld in verschillende taken om zo maximale efficiëntie te bereiken. Als er problemen waren bij bepaalde gedeelten is assistentie gevraagd.

Er is binnen het team een werkverdeling gemaakt voor de onderzoeken, applicatiefunctionaliteiten en documentatie hoofdstukken. De tussentijdse resultaten werden met elkaar besproken en verbeterd indien nodig, hierna werd dit samengevoegd en besproken met de begeleiders. Bij deze werkverdeling is rekening gehouden met de specialiteiten van elkaar, maar ook met elkaars interesses. Bij zwakkere punten was er sprake van assistentie van de ander.

De samenwerking is dus erg goed verlopen en de taakverdeling is goed bevallen.

12 REFERENTIES

- [1] Total Specific Solutions, „TSS,” [Online]. Available: <http://www.totalspecificsolutions.nl/>. [Geopend 07 02 2013].
- [2] Scrum.org, „Scrum,” [Online]. Available: <http://www.scrum.org/Resources/What-is-Scrum>. [Geopend 07 02 2013].
- [3] 9solutions, „9solutions,” [Online]. Available: <http://www.9solutions.com/en/>. [Geopend 26 02 2013].
- [4] chainway, „Chainway,” [Online]. Available: <http://www.chainway.net/download/solution/super-market.pdf>. [Geopend 26 02 2013].
- [5] i. atlas, „Indooratlas,” [Online]. Available: <http://web.indooratlas.com/web/WhitePaper.pdf>. [Geopend 26 02 2013].
- [6] F-Tech, „F-Tech,” [Online]. Available: <http://www.f-tech.com.tw/modules/publisher/item.php?itemid=24>. [Geopend 26 02 2013].
- [7] M. I. T. C. S. a. A. I. Laboratory. [Online]. Available: <http://cricket.csail.mit.edu/#people>.
- [8] S. Jung en W. Woo, „ubiTrack: Infrared-based User Tracking System for Indoor Environments,” 2004.
- [9] Ekahau, [Online]. Available: <http://www.ekahau.com/solutions/healthcare/ekahau-asset-tracking-and-management-solution.html>.
- [10] Kiesbeter.nl, „Kiesbeter,” [Online]. Available: <http://www.kiesbeter.nl/zorg-en-kwaliteit/ziekenhuizen/detail/erasmus-mc-centrumlocatie/#algemeen>. [Geopend 25 03 2013].

-
- [11] E. M. Centrum, „Erasmusmc,” [Online]. Available: <http://www.erasmusmc.nl/5663/162999/kengetallen>. [Geopend 25 03 2013].
 - [12] Kiesbeter, „Kiesbeter,” [Online]. Available: <http://www.kiesbeter.nl/zorg-en-kwaliteit/ziekenhuizen/detail/universitair-medisch-centrum-utrecht-wilhelmina-kinderziekenhuis/#algemeen>. [Geopend 25 03 2013].
 - [13] Kiesbeter, „Kiesbeter,” [Online]. Available: <http://www.kiesbeter.nl/zorg-en-kwaliteit/ziekenhuizen/detail/universitair-medisch-centrum-utrecht-locatie-azu/#algemeen>. [Geopend 25 03 2013].
 - [14] U. M. C. Utrecht, „UMCUtrecht,” [Online]. Available: <http://www.umcutrecht.nl/overumcutrecht/Kengetallen-2011.htm>. [Geopend 25 03 2013].
 - [15] Neophob, „Neophob,” [Online]. Available: http://neophob.com/files/rfid/PROTOCOL-821-880%20_2_.pdf. [Geopend 05 23 2013].
 - [16] Seedstudio, „Seedstudio,” [Online]. Available: http://www.seedstudio.com/wiki/index.php?title=13.56Mhz_RFID_module_-_IOS/IEC_14443_type_a. [Geopend 27 05 2013].
 - [17] Arduino, „Arduino,” [Online]. Available: http://arduino.cc/en/uploads/Main/arduino-mega2560_R3-schematic.pdf. [Geopend 27 03 2013].
 - [18] Arduino, „Arduino,” [Online]. Available: <http://arduino.cc/en/Hacking/LibraryTutorial>. [Geopend 27 03 2013].
 - [19] A. Craftsman, „Aspiring Craftsman,” [Online]. Available: <http://aspiringcraftsman.com/2008/01/03/art-of-separation-of-concerns/>. [Geopend 22 05 2013].
 - [20] Sencha, „Sencha,” [Online]. Available: <https://www.sencha.com/store/sencha-complete/>. [Geopend 27 03 2013].

-
- [21] Symfony, „Symfony,” [Online]. Available: http://symfony.com/doc/current/quick_tour/the_big_picture.html. [Geopend 27 03 2013].
 - [22] Railstutorials, „Railstutorial,” [Online]. Available: <http://ruby.railstutorial.org/ruby-on-rails-tutorial-book>. [Geopend 27 03 2013].
 - [23] R. o. Rails, „Rubyonrails,” [Online]. Available: http://guides.rubyonrails.org/getting_started.html#the-mvc-architecture. [Geopend 27 03 2013].
 - [24] „Simcrest,” [Online]. Available: <http://blog.simcrest.com/what-is-3-tier-architecture-and-why-do-you-need-it/>. [Geopend 03 05 2013].
 - [25] Microsoft, „Microsoft,” [Online]. Available: <http://msdn.microsoft.com/en-us/library/ee658117.aspx#LayeredStyle>. [Geopend 05 22 2013].
 - [26] Microsoft, „Microsoft,” [Online]. Available: <http://msdn.microsoft.com/en-us/library/ee658117.aspx#MessageBusStyle>. [Geopend 22 05 2013].
 - [27] M. Fowler, Patterns of Enterprise Application Architecture, Pearson education inc., 2003.
 - [28] R. o. Rails, „Rails Associations,” [Online]. Available: http://guides.rubyonrails.org/association_basics.html. [Geopend 22 05 2013].
 - [29] Rfc-Editor, „Rfc-editor,” [Online]. Available: <http://www.rfc-editor.org/rfc/rfc2616.txt>. [Geopend 24 05 2013].
 - [30] w3, „W3,” [Online]. Available: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>. [Geopend 24 05 2013].
 - [31] P. K.S.J, J. Kahn en B. Bose, „Smart Dust: Wireless Networks of Millimeter-Scale Sensor Nodes,” http://robotics.eecs.berkeley.edu/~pister/publications/1998/smartdust_comm_memo.pdf.

-
- [32] S. Weis, „RFID (Radio Frequency Identification): Principles and Applications,” <http://www.eecs.harvard.edu/cs199r/readings/rfid-article.pdf>.
- [33] R. Mautz, „Indoor Positioning Technologies”.
- [34] M. Steehouder, Leren communiceren, Noordhof uitgevers bv., 2006.
- [35] The agile42 team, „Agile42,” [Online]. Available: <http://www.agile42.com/en/agile-coaching-company/agile-scrum-tools/scrum-nutshell/>. [Geopend 07 02 2013].
- [36] Tass Technology Solutions, „Tass,” [Online]. Available: www.tass.nl. [Geopend 05 02 2013].
- [37] Trac, „Trac,” [Online]. Available: <http://trac.edgewall.org>. [Geopend 05 02 2013].
- [38] Wikipedia, „Wikipedia,” [Online]. Available: https://en.wikipedia.org/wiki/Exclusive_or. [Geopend 17 05 2013].

13 BIJLAGEN

- [A] Plan van aanpak
- [B] Persoonlijke evaluatie Patrick
- [C] Persoonlijke evaluatie Dick
- [D] Systeemtest registratie zorgmiddelen
- [E] Systeemtest lokalisatie zorgmiddelen
- [F] Systeemtest gebruikerstool

A. PLAN VAN AANPAK

PLAN VAN AANPAK

AUTOMATISCH REGISTRATIE TOOL VOOR ZORGMIDDELEN

Referentie:	ARTZ-PA-012
Datum:	11-04-2013
Versie:	1.21
Status:	Final
Auteurs:	F.C.D. van der Steen P.N. de Wit

© TASS B.V. 2013

Alle rechten voorbehouden. Verveelvuldiging, geheel of gedeeltelijk, is niet toegestaan dan met schriftelijke toestemming van de auteursrechthebbende.

All rights are reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Dit document is gepubliceerd door:

TASS B.V.
Eindhoven, Nederland

Commentaar en suggesties kunnen worden gestuurd naar:

TASS B.V.
Postbus 80060
5600 KA EINDHOVEN
Nederland
tel: +31 (0)40 2503200
fax: +31 (0)40 2503201

1	Introductie	4
1.1	Afkortingen en definities	4
1.2	Distributielijst	4
1.3	Documentgeschiedenis	5
2	Projectbeschrijving	6
2.1	Aanleiding	6
2.2	Probleemstelling	6
2.3	Doel & scope	7
2.4	Deelproducten	8
2.5	Organisatie	8
2.5.1	Organisatieschema	9
2.5.2	Stuurgroep	9
2.5.3	De klant	9
2.5.4	De docentbegeleider	10
2.5.5	Het projectteam	10
2.5.6	Verantwoordelijkheden	10
2.5.7	Communicatie en externe interfaces	11
2.6	Milestones	11
2.7	Werkproducten/releases	12
2.8	Afstudeerdocumentatie	12
2.9	Budget	13
2.10	Metrics	13
2.11	Rapportage	14
2.12	Sprint setup	14
3	Randvoorwaarden	17
3.1	Apparatuur, infrastructuur en tools	17
3.2	Risico's	17
4	Software testplan	18
4.1	Scope	18
4.2	Testopstelling	18
4.3	Aanpak	19
5	Configuratiemanagement plan	20
6	Kwaliteitswaarborging	21

6.1	Proces	21
6.2	Reviews	21
7	Acceptatie ARTZ	22
7.1	Acceptatie door het projectteam	22
7.2	Acceptatie door de klant	22

1 INTRODUCTIE

Dit plan beschrijft hoe het project is georganiseerd, wat de globale planning is en welke werkproducten wanneer worden opgeleverd, op welke aannames en randvoorwaarden het plan is gebaseerd, hoe de kwaliteit wordt gewaarborgd.

1.1 AFKORTINGEN EN DEFINITIES

ARTZ **A**utomatisch **r**egistratie **t**ool voor **z**orgmiddelen

1.2 DISTRIBUTIELIJST

Dit document wordt gedistribueerd over de volgende personen:

Naam	Organisatie	Functie
J. Zielman	TASS Technology Solutions	People manager
B. Bilos	TASS Technology Solutions	Technisch begeleider
M. Wensink	Hogeschool Utrecht	Docentbegeleider
F.C.D. van der Steen	TASS Technology Solutions	Afstudeerder
P.N de Wit	TASS Technology Solutions	Afstudeerder

1.3 DOCUMENTGESCHIEDENIS

Dit document heeft de volgende versie geschiedenis:

Versie	Status	Datum	Auteur	Wijzigingen
0.1	Draft	05-02-2013	F.C.D. van der Steen P.N. de Wit	Initiële versie
0.2	Draft	06-02-2013	F.C.D. van der Steen P.N. de Wit	Initiële versie
0.3	Draft	07-02-2013	P.N. de Wit	Sprint setup
0.4	Concept	08-02-2013	P.N. de Wit F.C.D. van der Steen	Concept versie
0.5	Concept	11-02-2013	P.N. de Wit	Error! Reference source not found. toegevoegd + Metrics tabel herzien
0.6	Concept	11-02-2013	F.C.D. van der Steen P.N. de Wit	Bijlage B + Layout
1.0	Final	19-02-2013	F.C.D. van der Steen P.N. de Wit	Wijzigingen review
1.1	Final	15-03-2013	F.C.D. van der Steen P.N. de Wit	Wijzigingen review docentbegeleider
1.2	Final	21-03-2013	F.C.D. van der Steen P.N. de Wit	Wijzigingen review docentbegeleider en technisch begeleider
1.21	Final	11-04-2013	P.N. de Wit	Style updates

2 PROJECTBESCHRIJVING

Dit hoofdstuk beschrijft wat het project inhoudt en welke onderdelen binnen of buiten de scope van het project vallen.

2.1 AANLEIDING

In de gezondheidszorg worden zorgmiddelen als rollators, anti decubitus matrassen en vernevelapparaten gebruikt. Doordat bij veel zorgverleners de registratie van hun zorgmiddelen handmatig gaat, wordt dit vaak vergeten te doen. Dit resulteert in het probleem dat zorgmiddelen spoorloos zijn, waardoor onnodig tijd verloren gaat om naar deze zorgmiddelen te zoeken.

TASS heeft het idee dat door de ontwikkeling van een geautomatiseerd plaatsregistratiesysteem dit probleem in de zorg kan worden opgelost, waardoor deze onnodig verloren tijd en geld kan worden bespaard.

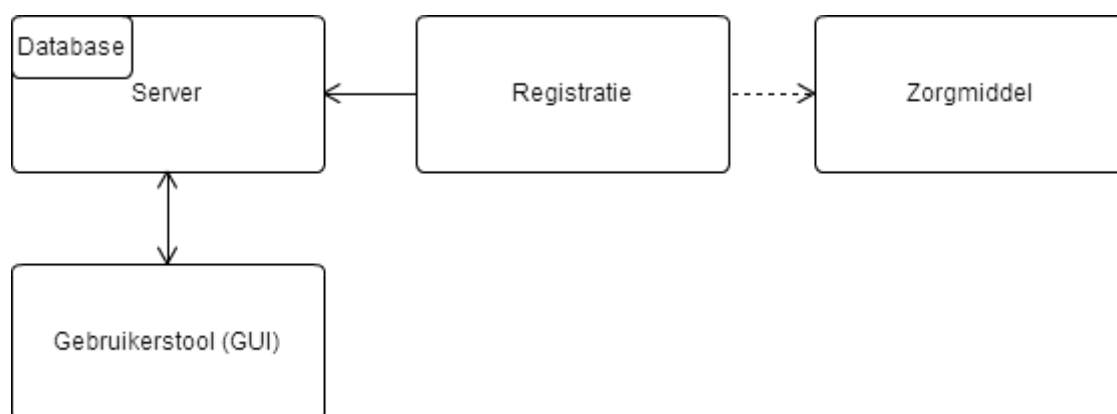
2.2 PROBLEEMSTELLING

De oplossing voor het eerder beschreven probleem is een geautomatiseerd plaatsregistratiesysteem, maar het is nog niet duidelijk **hoe** dit gerealiseerd kan worden.

Daarom moet onderzoek gedaan worden naar mogelijk te gebruiken technieken om dit systeem te maken.

2.3 DOEL & SCOPE

Het doel van het project is om een systeem en een tool te ontwerpen en ontwikkelen. Het systeem registreert automatisch de huidige plaats van de zorgmiddelen en de tool geeft gebruikers de mogelijkheid om zorgmiddelen op te zoeken. De flow van het systeem is als volgt weer te geven:



Figuur 25 – Flow van de ARTZ

Om de kosten van het project te beperken zal er een proof-of-concept (schaalmodel) worden gemaakt van het gewenste systeem.

De activiteiten die binnen de scope vallen zijn:

- Onderzoek naar bestaande systemen, binnen dit onderzoek wordt antwoord gegeven op de volgende vraag:
"Wat is een/zijn geschikte techniek(en) voor het registreren van zorgmiddelen?"
- Onderzoek naar draadloze communicatieprotocollen, binnen dit onderzoek wordt antwoord gegeven op de volgende vraag:
"Hoe werken de draadloze communicatieprotocollen van de technieken?"
- Onderzoek naar algoritmen voor locatiebepaling, binnen dit onderzoek wordt (afhankelijk van het gekozen communicatieprotocol) antwoord gegeven op de volgende vraag:
"Welk algoritme(n) geeft de gewenste nauwkeurigheid voor locatiebepaling en hoe werkt dit algoritme?"

- Ontwerpen van de ARTZ, deze activiteit bestaat uit de volgende deel activiteiten:
 - Onderzoek naar ontwikkelplatformen, binnen dit onderzoek wordt antwoord gegeven op de volgende vraag:
"Welk ontwikkelplatform(en) zijn nodig voor de ontwikkeling van de ARTZ?"
 - Onderzoek naar geschikte architectuur, binnen dit onderzoek wordt antwoord gegeven op de volgende vraag:
"hoe kan de architectuur van de ARTZ worden gerealiseerd?"
- Ontwikkelen van de ARTZ, de minimale functionaliteiten die de ARTZ moet hebben zijn:
 - Het automatisch registreren van zorgmiddelen
 - Het automatisch opslaan van de locatie van zorgmiddelen
 - Het kunnen inloggen als gebruiker via de tool
 - Het kunnen zoeken naar zorgmiddelen via de tool

De activiteiten die buiten de scope vallen zijn:

- Navigeren naar verloren apparaat
- Onderhoud van de ARTZ

2.4 DEELPRODUCTEN

De deelproducten die binnen het project worden opgeleverd zijn:

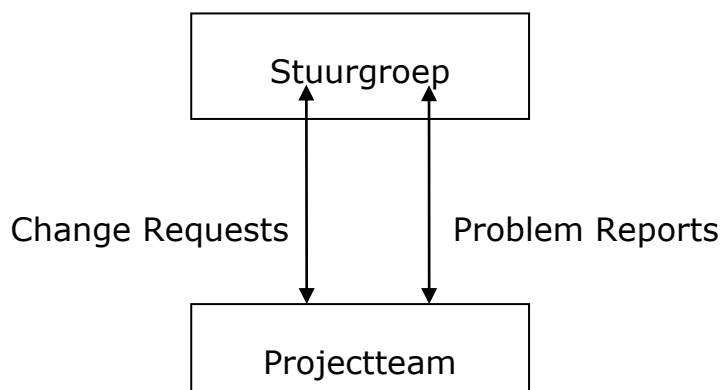
- Plan van aanpak
- Documentatie onderzoek(en)
- Functioneel ontwerp
- Technisch ontwerp
- De ARTZ
- Één scriptie (namens beide afstudeerders)

2.5 ORGANISATIE

Dit hoofdstuk beschrijft de structuur van het project en waarvoor de desbetreffende personen verantwoordelijk zijn. Ook wordt beschreven hoe de communicatie naar begeleiders zal verlopen.

2.5.1 ORGANISATIESCHEMA

Een kort overzicht van de organisatiestructuur van het project is te zien in Afb. 1.



2.5.2 STUURGROEP

De stuurgroep bestaat uit de volgende personen:

Rol	Naam	Functie in het project
Voorzitter	J. Zielman	Projectbegeleider
	B. Bilos	Technisch begeleider
Secretaris	F.C.D. van der Steen	Teamlid
Lid	P.N. de Wit	Teamlid
Vergaderfrequentie: 2-wekelijks		

De stuurgroep is verantwoordelijk voor:

- Periodieke voortgangscntrole in termen van tijd, geld en functionaliteit.
- Beslissingen over wijzigingen die de software requirements specificatie of het plan van aanpak beïnvloeden.
- Organisatorische zaken met betrekking tot resources, communicatie, milestones, leveringen, etc.
- Prioriteitsstelling binnen het project.
- Evaluatie van de resultaten van milestone checks, audits en de project evaluatie.

2.5.3 DE KLANT

De rol van klant zal tijdens dit project bekleed worden door J. Zielman. Met de klant zullen afspraken gemaakt worden over de functionele eisen van de ARTZ.

2.5.4 DE DOCENTBEGELEIDER

De functie van docentbegeleider zal tijdens dit project bekleed worden door M. Wensink. Met de docentbegeleider zullen afspraken gemaakt worden over zaken gerelateerd aan het afstuderen van de projectteamleden.

2.5.5 HET PROJECTTEAM

Het projectteam bestaat uit de volgende personen:

Naam	Periode	Niveau
F.C.D. van der Steen	Gehele project	Afstudeerder
P.N. de Wit	Gehele project	Afstudeerder

2.5.6 VERANTWOORDELIJKHEDEN

Hieronder is de lijst van initiële verantwoordelijkheden per projectteamlid.

Taak	Eindverantwoordelijke(n)
Onderzoek bestaande systemen en algoritmen voor locatiebepaling	F.C.D. van der Steen & P.N. de Wit
Onderzoek naar draadloze communicatieprotocollen	P.N. de Wit
Onderzoek naar ontwikkelplatformen	F.C.D. van der Steen
Onderzoek naar geschikte architectuur	F.C.D. van der Steen & P.N. de Wit
Functioneel ontwerp	P.N. de Wit
Technisch ontwerp	F.C.D. van der Steen & P.N. de Wit
Registratie zorgmiddelen	F.C.D van der Steen
Lokalisatie zorgmiddelen	P.N. de Wit
Tool met graphical user interface	P.N. de Wit
Scriptie	F.C.D van der Steen & P.N. de Wit
Testplan	F.C.D van der Steen & P.N. de Wit
Releases	F.C.D van der Steen

Tijdens de uitvoering van het project is het mogelijk dat verantwoordelijkheden wijzigen of aan een ander projectteamlid worden overgedragen. Indien er grote veranderingen zijn in de verantwoordelijkheden zal de docentbegeleider hierbij betrokken worden.

2.5.7 COMMUNICATIE EN EXTERNE INTERFACES

Het project zal gebruik maken van Scrum, dit is een iteratief en incrementeel agile software ontwikkelingsproces. Binnen het projectteam wordt een Scrum master aangewezen. De Scrum master begeleidt het team om zijn doelstellingen te behalen. Hij is niet de leider van het team, maar zorgt ervoor dat het scrumproces gevolgd wordt. Deze rol kan tijdens het project wijzigen van persoon, wat kan leiden tot een andere en mogelijk een (deels) betere aanpak. Binnen dit project is P.N. de Wit aangewezen als Scrum master.

Het project is opgedeeld in sprints van twee weken, binnen deze sprints wordt elke dag binnen het projectteam een daily scrum gehouden. Hierin worden de uitgevoerde en toekomstige activiteiten besproken. Ook wordt besproken welke problemen er zijn. Aan het einde van elke sprint vindt een voortgangsgesprek plaats met de project- en technisch begeleider. Eveneens zal binnen het projectteam een sprint review en sprint retrospective plaatsvinden. In de sprint review wordt de voortgang, opgeloste- en onopgeloste problemen, ingebouwde functionaliteiten en opgeleverde documenten besproken. In de retrospective wordt een korte evaluatie gegeven van de afgelopen sprint met wat goed ging en wat mogelijke verbeteringen zijn. Van de sprint review en de retrospective wordt een kort verslag gemaakt. Dit verslag wordt verstuurd naar de docentbegeleider, de projectbegeleider en de technisch begeleider.

2.6 MILESTONES

Onderstaande tabel geeft een overzicht van de project milestones zoals afgesproken met de stuurgroep:

Ref.	Milestone	Datum
#1	Goedkeuring plan van aanpak	01-03-2013
#2	Goedkeuring functioneel ontwerp	08-03-2013
#3	Oplevering onderzoeksrapport	22-03-2013
#4	Oplevering technisch ontwerp	05-04-2013
#5	Hardware levering	05-04-2013
#6	Hardware functioneert	12-04-2013
#7	Zorgmiddelen registreren	19-04-2013
#8	Zorgmiddelen lokaliseren	24-04-2013
#9	Versie Alpha ARTZ	24-04-2013
#10	Zorgmiddelen positie tonen aan gebruiker	03-05-2013
#11	Versie Bèta ARTZ	10-05-2013
#12	Succesvolle testcase uitvoeren	24-05-2013
#13	Acceptatie ARTZ	31-05-2013
#14	Oplevering ARTZ	31-05-2013
#15	Afstudeerzitting	17-06-2013

2.7 WERKPRODUCTEN/RELEASES

Dit hoofdstuk geeft de flow in het release- en acceptatieproces van producten weer.

- Een formele levering van een werkproduct vereist toestemming van de klant voor productacceptatie.
- Een tussentijdse levering vereist een onderlinge afspraak tussen het projectteam en de klant.

De onderstaande tabel geeft een overzicht van de werkproducten die formeel aan de klant worden geleverd:

Werkproduct / Release	Datum
Functioneel ontwerp	08-03-2013
Plan van aanpak	15-03-2013
ARTZ	31-05-2013
Technische documentatie	07-06-2013

De onderstaande tabel geeft een overzicht van de werkproducten die tussentijds aan de klant worden geleverd:

Werkproduct / Release	Datum
Concept functioneel ontwerp	15-02-2013
Concept plan van aanpak	18-02-2013
Concept onderzoeksrapport	22-03-2013
Alpha versie ARTZ	24-04-2013
Bèta versie ARTZ	10-05-2013

2.8 AFSTUDEERDOCUMENTATIE

Dit hoofdstuk geeft de flow in het acceptatieproces van afstudeerdocumenten weer.

- Een formele levering van een afstudeerdocument vereist toestemming van de docentbegeleider.
- Een tussentijdse levering vereist een onderlinge afspraak tussen het projectteam en de docentbegeleider.

De onderstaande tabel geeft een overzicht van de afstudeerdocumenten die formeel aan de docentbegeleider worden geleverd:

Werkproduct / Release	Datum
Plan van aanpak	15-03-2013
Scriptie	28-05-2013

De onderstaande tabel geeft een overzicht van de afstudeerdocumenten die tussentijds aan de docentbegeleider worden geleverd:

Werkproduct / Release	Datum
Sprint reviews en sprint retrospectives	Elke sprint
Concept plan van aanpak	18-02-2013
1 ^e concept van de scriptie	12-04-2013
2 ^e concept van de scriptie	26-04-2013
3 ^e concept van de scriptie	10-05-2013

2.9 BUDGET

Voor dit project is het volgende budget beschikbaar:

Omschrijving	Prijs
Manuren	€3150, -
Hardware	€1000, - (in overleg meer)
Totaal	€4150, -

2.10 METRICS

Tijdens het project zullen de volgende metrics worden verzameld:

Metric	Definitie	Frequentie
Milestone status	Zie Error! Reference source not found.	Periodiek
Effort	Zie Error! Reference source not found.	Wekelijks
Kosten	Zie Error! Reference source not found.	Wekelijks
Aantal risico's	Zie Error! Reference source not found.	Wekelijks
Test cases geïmplementeerd	Zie Error! Reference source not found.	Wekelijks
Test cases passed	Zie Error! Reference source not found.	Wekelijks

2.11 RAPPORTAGE

Er zal een 2-wekelijkse vergadering plaatsvinden met zowel mevr. J. Zielman als dhr. B. Bilos. Tijdens deze bespreking worden de uitgevoerde werkzaamheden, de komende werkzaamheden, de tegengekomen problemen en de problemen die verwacht worden besproken.

2.12 SPRINT SETUP

Voor het totale project is een planning gemaakt van de komende sprints. Er wordt uitgelegd wat in elke sprint wordt gedaan. In alle sprints zal aan de scriptie worden gewerkt, die in sprint 7 zal worden opgeleverd.

Sprint 0 – Initiële sprint

04-02 – 15-02

Behaalde milestones: -

Sprint 0 is de initiële sprint, de werkzaamheden zijn compleet uitgelegd in **Error! Reference source not found..**

Sprint 1 – PMP sprint

18-02 – 01-03

Behaalde milestone: #1

Sprint 1 heeft als doel een goedgekeurd plan van aanpak op te leveren. In deze sprint zal gewerkt worden aan het functioneel ontwerp. Daarnaast zal in deze sprint ook gewerkt worden aan de onderzoeken die in sprint 3 opgeleverd moeten worden.

Sprint 2 – FO sprint

04-03 – 15-03

Behaalde milestone: #2

Sprint 2 heeft als doel een goedgekeurd functioneel ontwerp op te leveren. Daarnaast zal in deze sprint ook gewerkt worden aan de onderzoeken die in sprint 3 opgeleverd moeten worden.

Sprint 3 - Onderzoekssprint

18-03 – 28-03

Behaalde milestone: #3

Sprint 3 heeft als doel de onderzoeksfase af te sluiten. Hierbij moeten alle onderzoeken worden afgerond en moet het besluit worden genomen welke hardware besteld moet worden voor de implementatie van de ARTZ. Deze deadline is hard omdat hardware levering lang kan duren en er niet veel tijd is voor uitloop.

Aan de hand van de resultaten van de onderzoeken wordt een globaal technisch ontwerp opgesteld dat verder wordt uitgewerkt in de tijd dat wordt gewacht op de hardware die is besteld.

Sprint 4 – Hardware sprint

02-04 – 12-04

Behaalde milestones: #4, #5, #6

In sprint 4 zal als eerste gewerkt worden aan het verder uitwerken van het technisch ontwerp dat in sprint 3 is gemaakt. Daarnaast zou de hardware tijdens deze sprint geleverd moeten worden, waarna het team kan beginnen met het begrijpen van de werking van deze hardware. Aan het einde van de sprint zou de hardware werkend moeten zijn en zou het team genoeg kennis moeten hebben om te beginnen met de realisatie van de basisfunctionaliteiten van de ARTZ.

Sprint 5 – Alpha sprint

15-04 – 24-04

Behaalde milestones: #7, #8, #9

In sprint 5 zal gewerkt worden aan het implementeren van de basisfunctionaliteiten van de ARTZ. Aan het einde van deze sprint zal een tussentijdse versie van de ARTZ worden opgeleverd.

Sprint 6 – Bèta sprint

02-05 – 10-05

Behaalde milestones: #10, #11

Sprint 6 heeft als doel het opleveren van een tweede tussentijdse versie van de ARTZ. In deze tweede versie zullen (veel) fouten van de alpha versie zijn opgelost en zullen alle basisfunctionaliteiten van de ARTZ zijn geïmplementeerd.

Sprint 7 – ARTZ opleversprint

13-05 – 31-05

Behaalde milestones: #12, #13 #14

Sprint 7 heeft als doel het verbeteren van fouten en het implementeren van extra functies in de bèta versie van de ARTZ. Aan het einde van sprint 7 zal de eindversie van de ARTZ worden opgeleverd en kan een demonstratie worden gegeven van de mogelijkheden van het systeem.

Sprint 8 - Afrondingssprint

03-06 – 14-06

Behaalde milestone: #15

Dit is de laatste sprint van het project, in deze sprint zal gewerkt worden aan de eindpresentatie, wat tevens de afstudeerzitting is en zal er technische documentatie worden opgeleverd van de ARTZ.

3 RANDVOORWAARDEN

3.1 APPARATUUR, INFRASTRUCTUUR EN TOOLS

Onderstaand een lijst met apparatuur en software die benodigd is om het project te kunnen uitvoeren.

Naam	Aantal	Datum	Toelichting	Aanwezigheid
Onderzoeksapparatuur	n.v.t.	n.v.t.	Zonder onderzoeksapparatuur kan het onderzoek niet worden uitgevoerd	Ja
Hardware	n.v.t.	05-04-2013	Voor een werkend proof-of-concept is benodigde hardware vereist	Nee
Ontwikkelomgeving	2	08-04-2013	Voor de ontwikkeling van software is een ontwikkelomgeving nodig	Onbekend
Agile projectmanagement tool	1	18-02-2013	Voor het ondersteunen van het Scrum proces is een tool vereist	Onbekend

3.2 RISICO'S

Onderstaand een lijst van risico's met bijbehorende acties om de hinder te beperken.

Risico	Actie
Lange hardware levertijd (of geleverde hardware is kapot)	Vroegtijdig bestellen
Registratie werkt niet waardoor lokalisatie onmogelijk wordt	Goed onderzoek doen naar de te gebruiken hardware en technieken
Hardware kan niet meer worden geleverd of is voor lange tijd uit het assortiment	Andere leverancier als reserve zoeken die dezelfde hardware ook verkoopt (tegenover een evt. hogere prijs)
Teamlid kan langdurig niet werken aan project	Prioriteiten geven aan alle taken waardoor taken met lagere prioriteit (in overleg met klant) niet worden uitgevoerd
Hardware is te duur	Optie bespreken met klant om een schaalmodel op te leveren

4 SOFTWARE TESTPLAN

Dit hoofdstuk beschrijft hoe ARTZ globaal wordt getest.

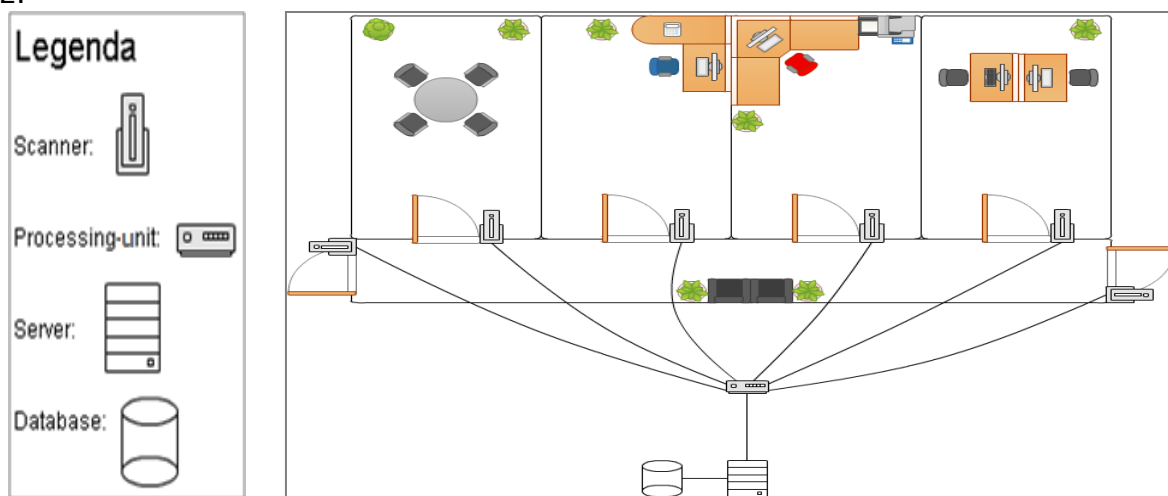
4.1 SCOPE

De ARTZ bestaat uit meerdere hoofdonderdelen, die allemaal apart zullen worden getest. Het systeem bevat de volgende hoofdonderdelen:

- Registratie zorgmiddelen
- Lokalisatie zorgmiddelen
- Tool met de graphical user interface (GUI)

4.2 TESTOPSTELLING

De hardware, de software omgeving en de tooling die nodig zijn voor het testen zijn opgenomen in paragraaf 3.1 (Apparatuur, infrastructuur en tools). De opstelling voor de eindversie (en mogelijk eerdere versies) van de ARTZ kan worden geïllustreerd met Afb. 2.



De scanners uit Afb. 2 zijn referentiepunten waarvan de locatie vooraf bekend is. De te volgen zorgmiddelen dragen een markering, waardoor de scanners in staat zijn om de zorgmiddelen te kunnen detecteren als deze zich in hun buurt bevinden. De scanners zijn gekoppeld aan een processing unit die de gedetecteerde zorgmiddelen van de scanners verwerkt en doorstuurt naar een server. Op de server draait een database waar de informatie over de gedetecteerde zorgmiddelen wordt opgeslagen. Door middel van de tool en de GUI kan de database op de server door gebruikers worden geraadpleegd om informatie van het systeem op te vragen.

4.3 AANPAK

De software wordt uitvoerig getest door zowel het projectteam als de klant. De verschillende onderdelen van de ARTZ zullen als volgt getest worden:

Onderdeel	Stappenplan
Registratie zorgmiddelen	Koppel de markering van een zorgmiddel aan het zorgmiddel type in de database van de server. Beweeg dit zorgmiddel door de test omgeving en controleer of het systeem het zorgmiddel heeft "gezien". Herhaal dit meerder malen. Controleer of de informatie van het gedetecteerde zorgmiddel bij de server is aangekomen. Controleer of de informatie in de database op de server is bijgewerkt.
Lokalisatie zorgmiddelen	Beweeg een zorgmiddel naar een locatie binnen de test omgeving. Controleer of de locatie opgeslagen op de server overeenkomt met de daadwerkelijke locatie.
Tool met graphical user interface (GUI)	Test de werking van alle functionaliteiten die zijn gespecificeerd in het functionele ontwerp.

De bevindingen van deze tests zullen worden opgelost/gedocumenteerd alvorens de versie beschikbaar te stellen voor de klant. Bij de eindversie van de ARTZ zal een testopstelling worden opgezet (zie paragraaf 4.2).

5 CONFIGURATIEMANAGEMENT PLAN

Binnen TASS wordt gebruik gemaakt van Trac. Trac is een webapplicatie waarmee webdocumenten gezamenlijk kunnen worden bewerkt. Ook beschikt Trac over een issue tracking systeem voor de ontwikkeling van softwareprojecten¹ en biedt het een interface naar Subversion (en andere version control systemen), een geïntegreerde wiki en rapportagemogelijkheden. Omdat TASS gebruik maakt van Trac zal eveneens het projectteam voor dit project gebruik maken van dit systeem.

Binnen Trac zal gebruik gemaakt worden van Subversion voor het beheren van de source code. Deze keuze is gemaakt, omdat TASS dit als standaard version control systeem gebruikt.

Als cliënt zal gebruikt worden gemaakt van TortoiseSVN. De standaardinrichting van de repository bestaat uit de volgende folders: trunk, branches en tags. De trunk bevat naast de code ook documentatie van het project.

In de folder tags, komen de releases van de ARTZ te staan (alpha, bèta, eindrelease). Belangrijke tussentijdse versies worden eveneens getagged. De versie nummering heeft de volgende vorm:

[Stage]-[major].[minor]

Onderdeel	Bevat	Nummering
Stage	Bevat het stadium waarin de ARTZ zich bevindt.	Alpha (a), bèta (b), release (r)
Major	Bevatten nieuwe features en/of breiden functionaliteiten uit.	0-9
Minor	Bevatten verbeteringen van bestaande features.	0-9

Twee voorbeelden van een mogelijk versienummer zijn: a-1.0 en b-1.2.

¹ Trac website - <http://trac.edgewall.org>

6 KWALITEITSWAARBORGING

Dit hoofdstuk beschrijft hoe de kwaliteit van de ARTZ wordt gewaarborgd.

6.1 PROCES

Het project wordt uitgevoerd volgens de richtlijnen en eisen zoals beschreven in het TASS Handboek Projecten.

De drie hoofdonderdelen van de ARTZ (zoals beschreven in paragraaf 4.1) zullen zowel apart als gecombineerd worden getest. Bij het afronden van elk hoofdonderdeel wordt dit uitvoerig getest (voordat het wordt samengevoegd met de andere hoofdonderde(e)l(en)).

Daarnaast zal eerdert de eindversie van de ARTZ wordt opgeleverd, een alpha en een bèta versie worden opgeleverd. Deze twee 'tussen'-versies bevatten beide extra functionaliteiten ten opzichte van de daarvoor opgeleverde versie en zullen door de klant worden getest om te bevestigen dat de tool tot zo ver naar believen is.

Als laatste wordt ook elke twee weken een voortgangsmeting georganiseerd om zo het project vroegtijdig bij te sturen indien het afwijkt van de wensen van de klant.

6.2 REVIEWS

Voor alle uitgebrachte versies van de ARTZ is een review vereist. Dit betekent dat in totaal drie reviews plaats zullen vinden, één tussentijdse voor de alpha versie en één tussentijdse voor de bèta versie en een formele voor de eindversie.

De procedure voor de reviews is als volgt. Eerst zullen de afstudeerders aan de hand van een kort testplan (zoals aangegeven in hoofdstuk 4) de ARTZ testen. Na het uitvoerig getest (en indien nodig verbeterd) te hebben wordt voor de klant een demonstratie gegeven.

Ook zal een code review plaatsvinden door de technisch begeleider bij de oplevering van de ARTZ. Daarnaast kunnen mogelijk code reviews plaatsvinden van andere afstudeerders (die werken aan andere projecten bij TASS).

7 ACCEPTATIE ARTZ

Dit hoofdstuk beschrijft de acceptatie van het projectteam en de klant voor zowel de tussentijdse als de eindversie van de ARTZ.

7.1 ACCEPTATIE DOOR HET PROJECTTEAM

Alvorens een release uit te brengen van het product, zal het projectteam het product onderwerpen aan een acceptatietest. Hiervoor geldt:

- Het projectteam zal een acceptatietest uitvoeren van een vooraf vastgesteld testplan (zie hoofdstuk 4).
- Het projectteam zal de resultaten van de acceptatietest rapporteren via een kort rapport, waarin onder andere alle gevonden defecten worden vermeld.

7.2 ACCEPTATIE DOOR DE KLANT

Binnen twee weken na de overdracht van het product zal de klant het product onderwerpen aan een acceptatietest. Hiervoor geldt:

- De klant zal een acceptatietest uitvoeren op basis van een vooraf door de klant vastgesteld testplan
- De klant zal de resultaten van de acceptatietest rapporteren via een acceptatietestrapport, waarin onder andere alle gevonden defecten worden vermeld

Defecten die een normaal gebruik van het product in de weg staan worden binnen een afgesproken termijn door het projectteam hersteld, waarna de klant de acceptatietest geheel of gedeeltelijk zal herhalen.

Zodra geen defecten meer worden gevonden zal de klant schriftelijk bevestigen dat hij het product accepteert. Indien de klant binnen twee weken na de laatste aflevering niet reageert wordt het product geacht door hem te zijn geaccepteerd.

B. PERSOONLIJKE EVALUATIE PATRICK

Deze bijlage gaat dieper in op wat de afstudeerder Patrick de Wit (1573954) tijdens zijn afstudeerperiode bij Tass Technology Solutions heeft geleerd.

Er is tijdens de afstudeerperiode door de betreffende student gewerkt aan meerdere onderdelen van de ARTZ, maar daarnaast is er ook veel tijd gestoken in het ontwerpen van de ARTZ en ook aan het voorafgaande onderzoek. Voor zowel de technische onderdelen als de niet technische onderdelen zal besproken worden wat er is geleerd.

Bij de start van het project is veel tijd gestoken in het in kaart brengen van de wensen en eisen van de klant door veel te communiceren. Deze wensen zijn gedocumenteerd aan de hand van een functioneel ontwerp die ter review is opgestuurd naar de klant. Daarnaast is er een plan van aanpak gemaakt welke hielp bij een projectmatige aanpak.

Persoonlijke review: In mijn mening samen met de onderzoeken de belangrijkste stappen van het project. In deze stappen moest veel gecommuniceerd worden met de klant en werd er van zowel de projectleden als de klant inzet verwacht. Dit resulteerde in dat er af toe van zowel de klant als de projectleden iets werd verwacht wat niet of niet compleet werd uitgevoerd. Naast dit minpunt verliep de start van het project snel en zelfs sneller dan volgens de planning wat een comfortabele speling gaf in de planning tegen de afsluiting van het project. Ook heb ik geleerd om voortgangsvergaderingen met de klant beter voor te bereiden aan de hand van een overzicht met de gemaakte uren en de uitgevoerde werkzaamheden en een agenda met te bespreken punten voor tijdens de vergadering.

De volgende stap was het uitvoeren van onderzoeken voor het kiezen van de geschikte technieken, methoden en hardware voor het proof-of-concept welke uiteindelijk opgeleverd moest worden. Nadat de resultaten van het onderzoeksverslag bekend waren is er een technische verslag opgeleverd met daarin de technische aspecten van de ARTZ.

Persoonlijke review: Zoals eerder vermeld vond ik het uitvoeren en voorbereiden van onderzoeken één van de belangrijkste onderdelen van dit project. Het verzamelen van informatie voor het maken van beslissingen met de klant over de ARTZ werd gestructureerd uitgevoerd en na het opleveren van het onderzoeksverslag is hier door andere studenten nog naar gevraagd, omdat ze deze wilden gebruiken voor hun eigen project. We hebben uiteindelijk maar één tegenslag gehad in deze stap en dat kwam door een fout in de datasheet voor een hardware module die is gebruikt in de ARTZ. Door deze fout kunnen niet velen tags tegelijk worden uitgelezen, maar blijft de scanner dezelfde tag opnieuw uitlezen. Deze fout kon echter op een andere (minder nette) manier worden opgelost. Hierdoor heb ik geleerd dat hoe duidelijk informatie ook in de datasheet van de module staat, deze niet altijd hoeft te kloppen. In de toekomst zal ik

dus beter zoeken naar reviews van andere gebruikers voordat de modules worden besteld.

De laatste stap was het opleveren van de ARTZ zelf. Hiervoor moest de registratie van de tags ontwikkeld worden, maar ook moest de registratiedata worden opgeslagen en moest deze data worden getoond aan de gebruiker. Als laatste is er ook een kleine Android applicatie ontwikkelt om niet alleen aan te tonen dat de API werkt, maar ook om gebruikers op een intuïtieve manier favoriete zorgmiddelen toe te laten voegen aan hun ARTZ account.

Persoonlijke review: In mijn mening de leukste stap van het project, het was in deze stap zeer belangrijk om een goede taakverdeling voor het team te maken en bij te houden om niet teveel of dubbel werk te verrichten. In deze stap veel geleerd, waaronder de taal Ruby als het gebruiken van het framework genaamd Ruby on Rails. Ook heb ik sinds jaren weer een keer gewerkt met het versiebeheersysteem Apache Subversion. Als laatste heb ik ook geleerd om een beter software ontwerp te maken voor de webapplicatie, het ontwerp bleek halverwege het project niet volledig volgens de Ruby on Rails standaarden waardoor deze voor een groot deel gewijzigd moest worden.

Over de gehele projectperiode heb ik naast projectgerelateerde zaken ook veel meer geleerd. Ik heb bij de start van het project besloten, omdat Tass Technology Solutions gevestigd is in Eindhoven, om te verhuizen naar Eindhoven. Hier heb ik samen met twee andere (mede) afstudeerders voor een half jaar gewoond, voor ons alle drie was het voor de eerste keer om niet meer thuis te wonen. Ook heb ik het bedrijf Tass Technology Solutions en veel van haar medewerkers mogen leren kennen.

Conclusie: Het project is gestructureerd en beheerst uitgevoerd. De scheiding in werkzaamheden was uitermate goed geregeld en ik ben zeer trots met wat ik zelf en wat we als team hebben kunnen bereiken. Het was een zeer leerzaam en leuk project en daarnaast een geweldige ervaring!

C. PERSOONLIJKE EVALUATIE DICK

Tijdens de afstudeerperiode is door de student gewerkt aan meerdere onderdelen van de ARTZ. In het project speelde onderzoek een grote rol. Samen met medestudent Patrick de Wit is de opdracht uitgevoerd. Er is besloten aan het begin van de afstudeerperiode om samen met Patrick en Arjan (medestudent technische informatica) in Eindhoven te gaan wonen. Voor ons alle drie een unieke ervaring waar veel van genoten is.

De opstart van het project

Aan het begin was nog niet precies duidelijk wat het systeem moest kunnen. Door aan de hand van een vragenlijst tijdens gesprekken met de klant (in dit project Josee Zielman) de eisen en wensen van het systeem vast te leggen, kon er vlot begonnen worden met het maken van het functioneel ontwerp (hier was snel een concept versie van). Tijdens de start moest ook het Plan van Aanpak (PvA) worden gemaakt. Tass gebruikt hier een standaard template voor genaamd het Project Management Plan (PMP). Door de eisen van het bedrijf en school samen te voegen in een plan, kon goed gepland worden wat er allemaal moest gebeuren voor de opdracht. De opstart ging sneller dan verwacht en hierdoor is een speling in de tijd ontstaan. Ik heb het zeer prettig ervaren om via een vragenlijst de eisen en wensen van de klant in kaart te brengen. Door dit te gebruiken moest er worden nagedacht over wat er besproken moest worden, maar ook welke eigen inbreng belangrijk was. De communicatie tussen klant en teamleden was gedurende de start van het project van groot belang. Dit kostte veel tijd aan beide kanten waardoor sommige afspraken niet volledig werden nagekomen. Ik heb geleerd dat de eisen en wensen van de klant in kaart brengen met behulp van een vragenlijst, een hele goede gestructureerde aanpak biedt voor het maken van een functioneel ontwerp. Wat ik ook heb geleerd is het beter voorbereiden van wat er besproken gaat worden tijdens een vergadering (bijvoorbeeld met behulp van agendapunten). Door dit te doen weten beide partijen wat verwacht kan worden tijdens een vergadering en komt niemand voor onverwachte verrassingen te staan.

Onderzoek

Voor de ARTZ moest onderzoek gedaan worden voor het kiezen van een geschikte draadloze communicatie techniek, welke architecturen er nodig waren maar ook welk hardware nodig was om de ARTZ te realiseren. Er is gestructureerd en objectief gezocht naar oplossingen voor de vragen die moesten worden beantwoord. Alle resultaten van de onderzoek zijn in een onderzoeksrapport vastgelegd. Uit deze onderzoeken zijn onder andere de keuze voor RFID en Ruby on Rails gemaakt. Het onderzoek doen is heel gestructureerd en objectief uitgevoerd. Hierdoor ontstaat er geen twijfel over de keuzen die gemaakt zijn. Het belangrijkste leerpunt tijdens het onderzoek was dat objectief zijn tijdens het onderzoek doen erg belangrijk is, hierdoor sluit je niks direct uit en wordt het

onderzoek compleet. Door alle resultaten in een rapport op te slaan kon er ook geen onderzoek verloren gaan en dit is zeker iets wat ik weer doe bij volgende projecten,

Ontwikkeling van de ARTZ

Buiten alleen onderzoek naar de ARTZ moest het ook ontwikkeld worden. De onderdelen die moesten worden ontwikkeld waren het registreren van tags, het vertalen van de registratie van een tag naar locatie en het tonen van tags aan een gebruiker. Er is een duidelijke verdeling gemaakt tussen de twee teamleden. Waarbij ik meer aan de hardware kant gewerkt heb en Patrick aan de User Interface. Tijdens de ontwikkeling wordt ontdekt dat de gekozen RFID modules uit het eerdere onderzoek niet aan de specificaties van de datasheet voldeden. Hieruit is geleerd dat soms de specificatie mooier is dan de werkelijkheid. Het was zeer prettig om een goede verdeling tussen de teamleden te hebben. Hierdoor konden we ons allebei focussen op een bepaald gedeelte en hulp vragen waar nodig. Dit heeft de ontwikkeling zeer bespoedigd. Tijdens de ontwikkeling heb ik erg veel geleerd over hoe een Arduino werkt en hoe deze kan worden gebruikt om randapparatuur aan te sturen. Ook is er geleerd hoe Ruby on Rails werkt en hoe hiermee webapplicaties kunnen worden gemaakt.

Conclusie

Het project is er goed uitgevoerd. Door een goede structuur en planning aan te houden is het project geslaagd. We mogen terugkijken met trots op een product dat mogelijk ooit ook een toevoeging kan worden in de maatschappij.

D. SYSTEEMTEST REGISTRATIE ZORGMIDDELEN

Koppel een tag aan een zorgmiddel

- Verwachte uitkomst: Een zorgmiddel heeft een RFID tag aan zich gekoppeld.
- Uitkomst: De tag is gekoppeld aan een zorgmiddel.

Beweeg dit zorgmiddel door de testomgeving en controleer of het systeem het zorgmiddel heeft "gezien". Herhaal dit meerder malen.

- Verwachte uitkomst: De software in de middleware tier toont dat een zorgmiddel (met een bepaalde tag) langs een bepaalde scanner is gekomen (dit wordt getoond in de console).
- Uitkomst: Het Java programma in de middleware tier heeft de output gegeven zoals te zien is in figuur 26. Dit betekent dat de tags gezien is bij de scanners die gepasseerd zijn.

```
[13:49:45] tag: f31c554f seen at scanner with serialnumber e24b67311408b07a  
[13:49:49] tag: f31c554f seen at scanner with serialnumber 7a1a5f5d6c0f837c  
[13:49:51] tag: f31c554f seen at scanner with serialnumber 409493ee93b371b7
```

Figuur 26 – Middleware tier output registratie zorgmiddel

Controleer of de informatie van het gedetecteerde zorgmiddel bij de server is aangekomen

- Verwachte uitkomst: De webserver van de Ruby on Rails applicatie ziet een aanvraag naar zijn webservice binnen komen en zet dit in de console output.
- Uitkomst: De server heeft de output gegeven zoals te zien is in figuur 27. Dit betekent dat de informatie bij de server is aangekomen is.

```
Started POST "/api/v1/asset_registrations.json" for 10.10.4.90 at 2013-05-17 13:  
35:20 +0200  
Processing by Api::V1::AssetRegistrationsController#create as JSON  
Parameters: {"tag_serial_number"=>"f31c554f", "scanner_serial_number"=>"7a1a5f  
5d6c0f837c"}
```

Figuur 27 - Webserver output registratie zorgmiddel

E. SYSTEEMTEST LOKALISATIE ZORGMIDDELEN

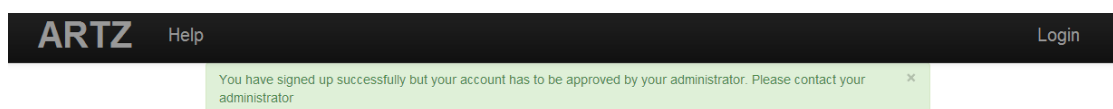
Beweeg een zorgmiddel naar een locatie binnen de testomgeving en controleer of de locatie opgeslagen op de server overeenkomt met de daadwerkelijke locatie

- Verwachte uitkomst: De locatie die is opgeslagen op de server is hetzelfde als waar het zorgmiddel naar toe is verplaatst binnen het schaalmodel.
- Uitkomst: Deze kan tot nog toe niet worden uitgevoerd, omdat het schaalmodel nog onder constructie is.

F. SYSTEEMTEST GEBRUIKERSTOOL

Probeer een nieuw account aan te maken

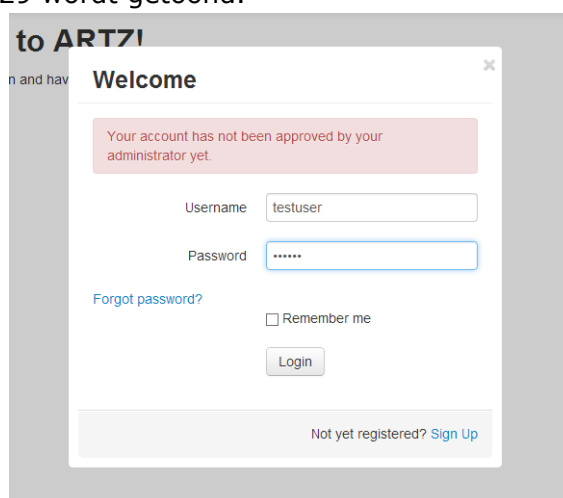
- Verwachte uitkomst: Het nieuwe account is aangemaakt en staat in de database van de ARTZ. Ook wordt er een scherm getoond met de melding dat het aanmaken succesvol is uitgevoerd, maar het account nog eerst moet worden goedgekeurd voor het in gebruik van worden genomen.
- Uitkomst: het nieuwe account is aangemaakt en staat in de database van de ARTZ. Het scherm uit figuur 28 wordt getoond.



Figuur 28 - Aanmaken nieuw account

Probeer met het nog niet geactiveerde account in te loggen

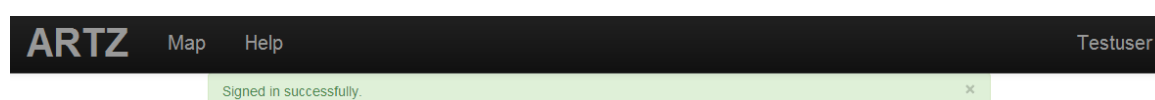
- Verwachte uitkomst: In het inlog scherm wordt getoond dat het account wel bestaat, maar nog niet geactiveerd is.
- Uitkomst: Het account kan niet inloggen omdat het nog niet is geactiveerd. Het scherm uit figuur 29 wordt getoond.



Figuur 29 - Inloggen ongeactiveerd account

Probeer dit account te activeren in de adminomgeving en probeer nogmaals in te loggen.

- Verwachte uitkomst: De webapplicatie wijzigt de status van het account naar geactiveerd en in de database is te zien dat de actie is voltooid. Ook kan de gebruiker nu wel inloggen.
- Uitkomst: Het account is nu geactiveerd en er kan op worden ingelogd. Het scherm uit figuur 30 wordt getoond.



Figuur 30 - Inloggen geactiveerd account

Probeer nieuwe tags aan te maken

- Verwachte uitkomst: Na het correct invullen van de gegevens van een nieuwe tag en het drukken op de knop om deze toe te voegen, verschijnt er een nieuwe rij in de tabel met alle tags en deze nieuwe tag is daarnaast ook te vinden in de database.
- Uitkomst: De nieuwe tag is in de tagmanagementpagina succesvol toegevoegd. Het verschil tussen voor en na is te zien in figuur 31.

Tag Type	Serial Number	Created At	Updated At	
Laundry	22F9E539	2013-05-02 07:26:46 UTC	2013-05-02 07:26:46 UTC	✗
Laundry	92FCE739	2013-05-02 07:26:46 UTC	2013-05-02 07:26:46 UTC	✗
Laundry	82B95ABF	2013-05-02 07:26:46 UTC	2013-05-02 07:26:46 UTC	✗
Card	F31C554F	2013-05-02 07:26:47 UTC	2013-05-02 07:26:47 UTC	✗
Card	63A3564F	2013-05-02 07:26:47 UTC	2013-05-02 07:26:47 UTC	✗
Card	0BF730D6	2013-05-02 07:26:47 UTC	2013-05-02 07:26:47 UTC	✗
Card	0B8412D6	2013-05-02 07:26:47 UTC	2013-05-02 07:26:47 UTC	✗
<input type="text"/>	<input type="text"/>			✓

➔

Tag Type	Serial Number	Created At	Updated At	
Laundry	22F9E539	2013-05-02 07:26:46 UTC	2013-05-02 07:26:46 UTC	✗
Laundry	92FCE739	2013-05-02 07:26:46 UTC	2013-05-02 07:26:46 UTC	✗
Laundry	82B95ABF	2013-05-02 07:26:46 UTC	2013-05-02 07:26:46 UTC	✗
Card	F31C554F	2013-05-02 07:26:47 UTC	2013-05-02 07:26:47 UTC	✗
Card	63A3564F	2013-05-02 07:26:47 UTC	2013-05-02 07:26:47 UTC	✗
Card	0BF730D6	2013-05-02 07:26:47 UTC	2013-05-02 07:26:47 UTC	✗
Card	0B8412D6	2013-05-02 07:26:47 UTC	2013-05-02 07:26:47 UTC	✗
Card	ABCD1234	2013-05-17 13:38:19 UTC	2013-05-17 13:38:19 UTC	✗
<input type="text"/>	<input type="text"/>			✓

Figuur 31 – Invvoegen van nieuwe tag

Probeer nieuwe zorgmiddelen aan te maken en koppel deze aan een tag

- Verwachte uitkomst: Na het correct invullen van de gegevens van een nieuw zorgmiddel en het drukken op de knop om deze toe te voegen, verschijnt er een nieuwe rij in de tabel met alle zorgmiddelen en dit nieuwe zorgmiddel is daarnaast ook te vinden in de database.
- Uitkomst: Het nieuwe zorgmiddel is in de zorgmiddelmanagementpagina succesvol toegevoegd. Het verschil tussen voor en na is te zien in figuur 32.

Assettype	Tag	Name	Image	Description	
Bed	22F9E539	—		—	✕
None	0BF730D6	—		—	✕
None	0B8412D6	—		—	✕
<div> [Select type] [Select tag] [Enter name] Bestand kiezen Geen bestand gekozen [Enter description] ✓ </div>					

Assettype	Tag	Name	Image	Description	
Bed	22F9E539	—		—	✕
None	0BF730D6	—		—	✕
None	0B8412D6	—		—	✕
Walker	ABCD1234	Walker		It's a walker	✕
<div> [Select type] [Select tag] [Enter name] Bestand kiezen Geen bestand gekozen [Enter description] ✓ </div>					

Figuur 32- Invoegen nieuw zorgmiddel

Probeer een nieuw zorgmiddeltype aan te maken

- Verwachte uitkomst: Na het correct invullen van de gegevens van een nieuw zorgmiddeltype en het drukken op de knop om deze toe te voegen, verschijnt er een nieuwe rij in de tabel met alle zorgmiddeltypes en dit nieuwe zorgmiddeltype is daarnaast ook te vinden in de database.
- Uitkomst: Het nieuwe zorgmiddeltype is in de zorgmiddeltypemanagementpagina succesvol toegevoegd. Het verschil tussen voor en na is te zien in figuur 33.

Name	Image	Description	
Bed		A piece of furniture used as a place to sleep or relax.	✕
Walker		A tool for disabled or elderly people who need additional support to maintain balance or stability while walking.	✕
Human		Primates of the family Hominidae, and the only extant species of the genus Homo.	✕
<div> [Enter name] Bestand kiezen Geen bestand gekozen [Enter description] ✓ </div>			


Name	Image	Description	
Bed		A piece of furniture used as a place to sleep or relax.	✕
Walker		A tool for disabled or elderly people who need additional support to maintain balance or stability while walking.	✕
Human		Primates of the family Hominidae, and the only extant species of the genus Homo.	✕
Testtype		Test type for report.	✕
<div> [Enter name] Bestand kiezen Geen bestand gekozen [Enter description] ✓ </div>			

Figuur 33 - Invoegen nieuw zorgmiddeltype

Probeer een nieuwe map aan te maken

- Verwachte uitkomst: Na het correct invullen van de gegevens van een nieuwe map en het drukken op de knop om deze toe te voegen, verschijnt er een nieuwe rij in de tabel met alle maps en deze map is daarnaast ook te vinden in de database.
- Uitkomst: De nieuwe map is in de mapmanagementpagina succesvol toegevoegd. Het verschil tussen voor en na is te zien in figuur 34.

Maps			
Name	Description	Floors	
ARTZ Map	Map for scale model	3	✕
Name	Description		✓



Maps			
Name	Description	Floors	
ARTZ Map	Map for scale model	3	✕
Test Map	Map for testing	0	✕
Name	Description		✓

Figuur 34 - Invoegen nieuwe map

Probeer nieuwe verdiepingen aan de map toe te voegen


- Verwachte uitkomst: Na het correct invullen van de gegevens van een verdieping en het drukken op de knop om deze toe te voegen, verschijnt er een nieuwe rij in de tabel met alle verdiepingen en deze verdieping is daarnaast ook te vinden in de database.
- Uitkomst: De nieuwe verdieping is in de mapmanagementpagina succesvol toegevoegd aan een map. Het verschil tussen voor en na is hieronder te zien. Het verschil tussen voor en na is te zien in figuur 35.

Maps			
Name	Description	Floors	
ARTZ Map	Map for scale model	3	✕
Test Map	Map for testing	0	✕

Name	Image	Description
Name	<div>Bestand kiezen</div>	Geen bestand gekozen
		Description ✓

Name	Description	✓
------	-------------	---

Maps			
Name	Description	Floors	
ARTZ Map	Map for scale model	3	✕
Test Map	Map for testing	1	✕

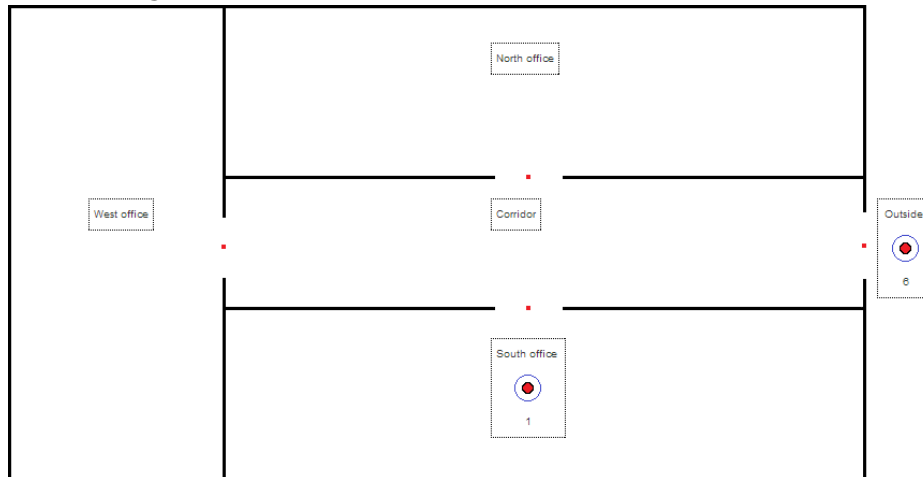
Name	Image	Description
Test Floor	<div></div>	Test floor for testing
Name	<div>Bestand kiezen</div>	Geen bestand gekozen
		Description ✓

Name	Description	✓
------	-------------	---

Figuur 35 - Invoegen nieuwe verdieping

Probeer het zoeken van alle zorgmiddelen

- Verwachte uitkomst: Alle zones waarin één of meer zorgmiddelen zich bevinden, worden gemarkeerd met een speciaal teken en daaronder een getal wat duidt op het aantal zorgmiddelen dat zich daar bevindt.
- Uitkomst: Het zoeken naar alle zorgmiddelen is succesvol uitgevoerd. Het resultaat is in figuur 36 te zien.



Figuur 36 - Resultaat alle zorgmiddelen zoeken

Probeer het zoeken van alle zorgmiddelen die als favoriet genoteerd zijn





- Verwachte uitkomst: Alle zones waarin één of meer zorgmiddelen zich bevinden die zijn toegevoegd als favoriet van de ingelogde gebruiker, worden gemarkeerd met een speciaal teken en daaronder een getal wat duidt op het aantal favoriete zorgmiddelen dat zich daar bevindt.
- Uitkomst: Het zoeken naar alle zorgmiddelen die als favoriet genoteerd zijn is succesvol uitgevoerd. Het resultaat is in figuur 37 te zien.



Figuur 37 - Resultaat zoeken favorieten

Probeer het opvragen van de gedetailleerde zoekresultaten

- Verwachte uitkomst: De markering op een verdieping na het zoeken van zorgmiddelen is door gebruikers aanklikbaar. Nadat een gebruiker dit opvraagt, wordt deze in een tabel onder de afbeelding van de verdieping getoond.
- Uitkomst: Het tonen van de gedetailleerde zoekresultaten is succesvol uitgevoerd. Het resultaat is in figuur 38 te zien.

	<p>Serial number: 22F9E539</p> <p>Type: Bed</p> <p>Description: A piece of furniture used as a place to sleep or relax.</p>
	<p>Serial number: 92FCE739</p> <p>Type: Bed</p> <p>Description: A piece of furniture used as a place to sleep or relax.</p>
	<p>Serial number: 82B95ABF</p> <p>Type: Bed</p> <p>Description: A piece of furniture used as a place to sleep or relax.</p>
	<p>Serial number: 63A3564F</p> <p>Type: Walker</p> <p>Description: A tool for disabled or elderly people who need additional support to maintain balance or stability while walking.</p>
	<p>Serial number: 0BF730D6</p>
	<p>Serial number: 0B8412D6</p>

Figuur 38 - Gedetailleerde zoekresultaten