

Afstudeerverslag: Personendetectie bij calamiteiten



Wouter de Vries,
Erik Hubers,
Dennis Heuven,
Jelle van Leeuwen.

Afstudeerbegeleiders: Dhr. ir. J.W.M. Stroet;
Dhr. P. Goolkate, MSc.

Opdrachtgever: Namens Kenniscentrum Design
& Technologie: Dhr. dr. ir.
R.J.W.T. Tangelder.

14 juni 2011
Versie 1.0

Inhoud

| | |
|--|----|
| Inhoud..... | 2 |
| Versiebeheer..... | 3 |
| Verklarende woordenlijst | 4 |
| 1 Inleiding | 4 |
| 2 Onderzoeksvraag & probleemstelling | 5 |
| 3 Projectopzet..... | 6 |
| 3.1 Scrum..... | 6 |
| 3.2 JIRA | 7 |
| 3.3 Versiebeheer | 11 |
| 3.4 Rolverdeling | 11 |
| 4 Vooronderzoek..... | 12 |
| 4.1 Kinect..... | 12 |
| 4.2 Deursensoren..... | 14 |
| 4.3 Warmte-/bewegingssensoren | 15 |
| 4.4 RFID..... | 15 |
| 5 Architectuur..... | 18 |
| 5.1 Website | 19 |
| 5.2 Bepaling prioriteit ruimte | 21 |
| 5.3 Server | 22 |
| 5.4 Gateways | 25 |
| 5.5 Communicatie tussen gateway en server..... | 27 |
| 5.6 Nodes..... | 28 |
| 5.7 Routeringsalgoritme..... | 30 |
| 6 Testen..... | 33 |
| 6.1 Samenstelling groep..... | 33 |
| 6.2 Testlocatie..... | 34 |
| 6.3 Conclusie | 34 |
| 6.4 Conclusies & Aanbevelingen | 34 |
| 7 Conclusie..... | 35 |
| 7.1 Met welke sensoren kunnen personen gedetecteerd worden | 35 |
| 7.2 Hoe kan de door de sensoren gegenereerde informatie verstuurd worden | 35 |
| 7.3 Hoe kan de door de sensoren gegenereerde informatie inzichtelijk worden gemaakt voor de eindgebruiker..... | 35 |
| 7.4 Wat is de betrouwbaarheid van de verkregen resultaten | 35 |
| 8 Aanbevelingen | 36 |
| 8.1 Schaalbaarheid | 36 |

| | | |
|--|---|----|
| 8.2 | Meer testen, vergroting betrouwbaarheid | 36 |
| 8.3 | Uitbreiding website | 37 |
| 8.4 | Stroomverbruik (UPS)..... | 37 |
| 8.5 | Compact PCB..... | 37 |
| 8.6 | Beveiliging XBee Communicatie | 37 |
| 8.7 | Meer sensortypes toevoegen..... | 37 |
| 9 | Referenties | 38 |
| Appendix I: Onderzoeksverslag Kinect..... | | 39 |
| Appendix II: Testresultaten | | 46 |
| Appendix III: Overzicht afgeronde user stories | | 48 |

Versiebeheer

Hieronder staat de tabel met alle versies en de bijbehorende bewerkingen.

| Versie | Datum | Auteur | Bewerkingen |
|--------|--------------|---------------------------------|---|
| V0.1 | 7 juni 2011 | J. van Leeuwen | Initiële opzet, inleiding, projectopzet |
| V0.2 | 7 juni 2011 | W. de Vries | Toevoeging Probleemstelling & onderzoeksvraag |
| V0.3 | 7 juni 2011 | D. Heuven | Toevoeging Warmte-/bewegingssensor |
| | | J. van Leeuwen | Toevoeging communicatie webserver/gateway |
| | | E. Hubers | Toevoeging Sheevaplugs |
| V0.4 | 8 juni 2011 | E. Hubers | Toevoeging RFID |
| | | W. de Vries | Toevoeging Kinect |
| | | D. Heuven | Toevoeging Deursensoren |
| V0.5 | 9 juni 2011 | J. van Leeuwen | Toevoeging Jira |
| | | D. Heuven | Toevoeging Nodes |
| | | W. de Vries | Toevoeging testen |
| | | E. Hubers | Toevoeging Architectuur algemeen |
| | | J. van Leeuwen | Toevoeging conclusie & begin discussie |
| V0.6 | 9 juni 2011 | J. van Leeuwen | Afronding eerste opleversie |
| V0.7 | 12 juni 2011 | J. van Leeuwen | Verwerking commentaar docenten |
| V0.8 | 14 juni 2011 | J. van Leeuwen & W. de Vries | Verwerking commentaar docenten |
| V0.9 | 14 juni 2011 | D. Heuven | Toevoegen Aanbevelingen |
| V1.0 | 14 juni 2011 | J. van Leeuwen | Definitieve versie |

Verklarende woordenlijst

| Woord | Verklaring |
|------------------|--|
| API-key | Beveiligingssleutel, benodigd om gebruik te maken van de webserver-services. |
| Burndown-chart | Grafiek die voortgang project weergeeft |
| FTDI | Future Technology Devices International, brug tussen RS232 en USB. |
| Increment | Planblok, er waren er in totaal 3. |
| Mesh-netwerk | Een netwerk van meerdere nodes, waarbij een node niet alleen begin en/of eindpunt is van een netwerk, maar ook wordt gebruikt voor het doorgeven van pakketten van andere nodes. Hiervoor is een routeringsprotocol benodigd. |
| Rebroadcast | Herhalings-broadcast. |
| Scrummaster | Het hoofd van het scrum-team. |
| Sprint | Planblok, onderdeel van increment. |
| Stand-up meeting | Dagelijks overlegmoment waarbij alle projectleden dienen te staan. |
| Subtask | Een kleinere taak die bij een user story hoort. |
| SVN-server | Server voor het versiebeheer van documenten en software. |
| User story | Wens van de opdrachtgever, behapbare (dag)taak. |
| WPAN | Wireless Personal Area Network. |

1 Inleiding

Dit verslag vormt de afstudeerscriptie bij het project voor het Kenniscentrum Design & Technologie: personendetectie bij calamiteiten. In dit project is er naar gestreefd een systeem te ontwikkelen waarbij in een openbaar gebouw per ruimte met een zekere grenzende waarschijnlijkheid gesteld kan worden hoeveel personen er per ruimte aanwezig zijn.

In dit verslag staat de werkwijze van de projectgroep, alle bevindingen en de bijbehorende conclusies en aanbevelingen.

Bij sommige hoofdstukken staat een of meerdere user stories vermeld, dit is een referentie naar Appendix III: Overzicht afgeronde user stories.

Dit project is uitgevoerd door Erik Hubers, Dennis Heuven, Wouter de Vries (studenten HBO Technische Informatica, Enschede) en Jelle van Leeuwen (student HBO Informatica, Enschede) in de periode februari – juni 2011.

2 Onderzoeksvraag & probleemstelling

In het geval van brand in een openbaar gebouw zal het aanwezige BHV-team en/of de brandweer het gebouw gedeeltelijk of geheel ontruimen. Dit doen zij door elke ruimte in het gebouw systematisch te controleren. In een worst-case scenario kan een drukke ruimte pas als een van de laatste ruimtes gecontroleerd worden. En gegeven is, dat de kans op ongevallen toeneemt naarmate de tijd verstrijkt.

De vraag is hoe er voor gezorgd kan worden dat het BHV-team en/of de brandweer doelgericht ruimten kan controleren en zo dus de kans op ongevallen kan verlagen.

De onderzoeksvraag luidt dan ook als volgt:

- Hoe kan, in een openbaar gebouw, gedetecteerd worden in welke ruimten zich personen bevinden
 - Met welke sensoren kunnen personen gedetecteerd worden;
 - Hoe kan de door de sensoren gegenereerde informatie verstuurd worden;
 - Hoe kan de door de sensoren gegenereerde informatie inzichtelijk worden gemaakt voor de eindgebruiker (Bijv. het BHV-team of de brandweer);
 - Wat is de betrouwbaarheid van de verkregen resultaten.

3 Projectopzet

3.1 Scrum

Tijdens het project is gebruik gemaakt van Scrum, een framework voor agile management van softwareontwikkeling (Wikipedia). Om dit framework vorm te geven, is gebruik gemaakt van JIRA. Dit systeem biedt de mogelijkheid scrum te beheren waardoor post-it's verleden tijd zijn. Naast scrum wordt ook de planning in JIRA beheerd.

In een notendop houdt Scrum in dat de requirements geformuleerd worden aan de hand van user stories. Elke story kan in een vooraf afgesproken tijdsperiode gekozen worden door de opdrachtgever. In **Error! Reference source not found.** staat een overzicht van al deze stories.

Elke story heeft een titel, een omschrijving, een how-to-demo en een aantal story points.

| | |
|---------------------|--|
| Titel | Implementeer login-functie |
| Story points | 25 punten |
| Omschrijving | De volledige implementatie van een login-functie op de website, exclusief accountbeheer. |
| How to demo | <i>Demonstreren dat login-functie werkt aan de hand van live-demo</i> |

Tabel 1: Voorbeeld user story

3.2 JIRA

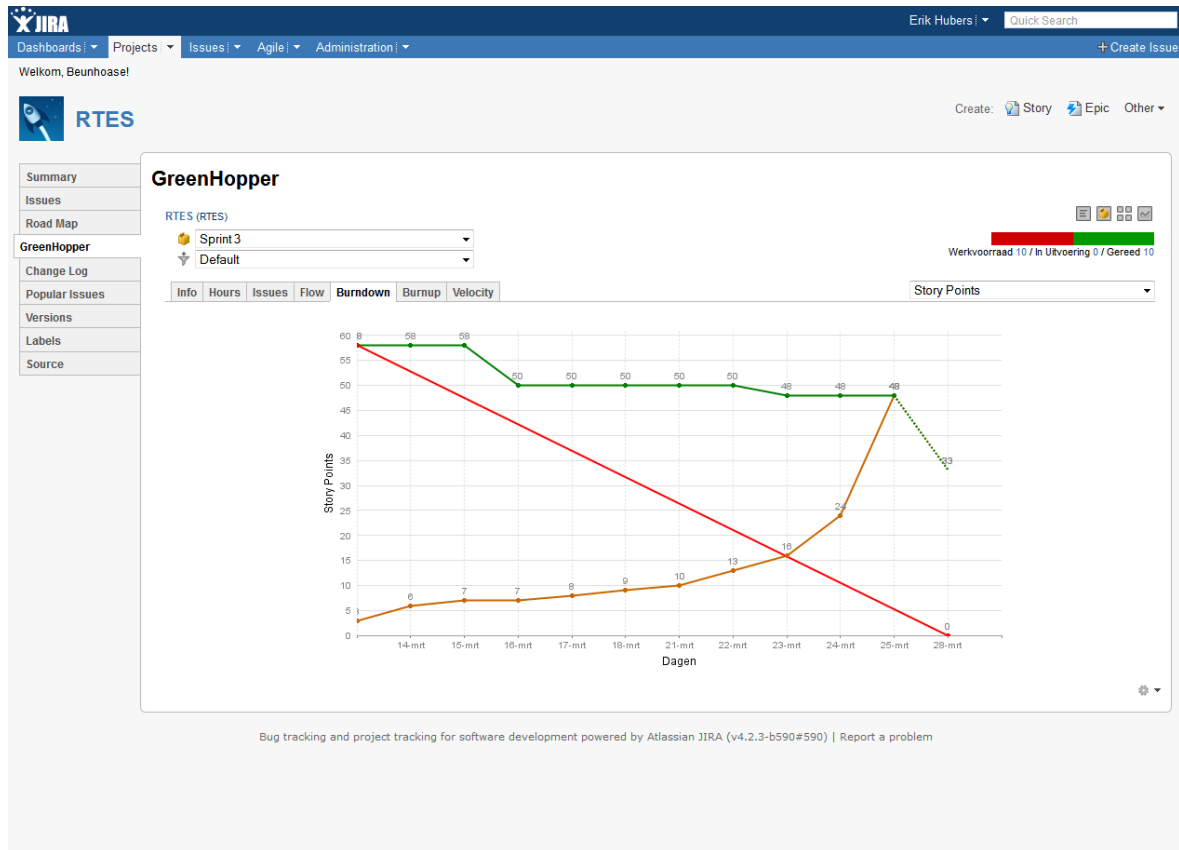
JIRA – de naam stamt af van ‘Goijra’, Japans voor Godzilla (Wikipedia) – is een zeer uitgebreide managementtool gemaakt door Atlassian. Een licentie voor maximaal 10 gebruikers kost eenmalig \$10,-, inclusief 12 maanden support en updates. Het is mogelijk de hosting bij Atlassian onder te brengen, echter voor dit project is gekozen dit lokaal op te zetten, omdat dit geen extra kosten met zich meebrengt en bovendien een uitdaging is.

JIRA biedt onder andere de volgende mogelijkheden (Atlassian), de punten gearceerd met een asteriks (*) geven aan dat de functionaliteit ook daadwerkelijk gebruikt is binnen dit project.

- * Planning, mogelijkheid tot plannen van increments en sprints;
- * Geautomatiseerde burndown-charts;
- * Beheer user stories;
- * Koppelen user story aan projectlid;
- * Besteedde tijd bijhouden op user-story-niveau;
- * Geautomatiseerde uitdraai voltooide en openstaande stories
- Bug-tracking;
- Koppeling SVN-server

3.2.1 Schermafbeeldingen

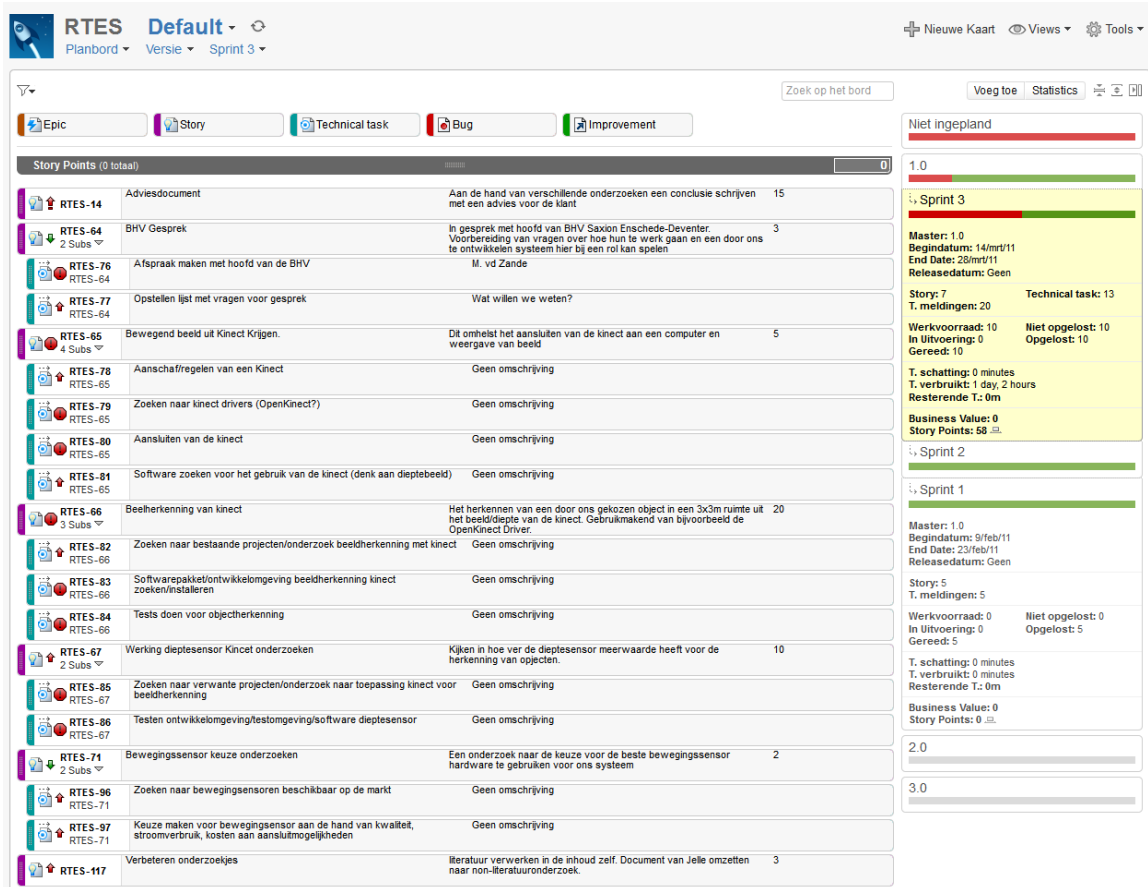
Hieronder staan enkele afbeeldingen die aangeven wat zoal de mogelijkheden van JIRA zijn. Slechts enkele belangrijke mogelijkheden zijn weergegeven.



Figuur 1: Burndown-chart JIRA

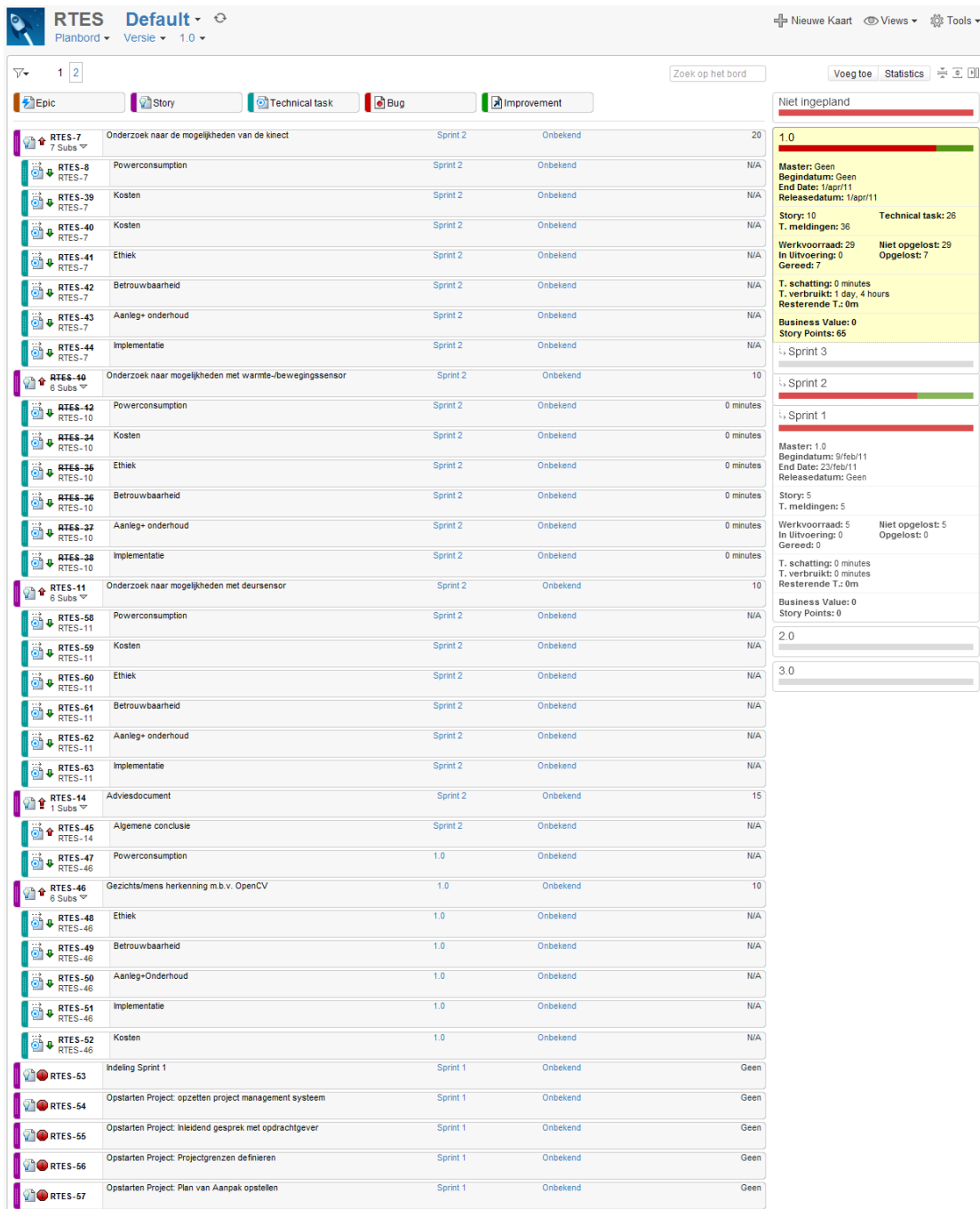
Noot: deze afbeelding dient slechts als voorbeeld.

Bovenstaande afbeelding geeft de burndown-chart-mogelijkheid weer. De rode lijn geeft de ideale lijn weer (elke dag evenveel werk verzetten totdat alles precies zoals gepland op het einde af is). De groene lijn geeft het daadwerkelijk verzette werk aan. De oranje lijn geeft een suggestie over hoeveel werk er vanaf dat moment nog verzet moet worden om weer op schema te komen.



Figuur 2: Overzicht openstaande user stories met subtasks

Deze afbeelding geeft een overzicht van alle openstaande user stories met bijbehorende subtasks, op een bepaald moment binnen het project. Rechts is de planning te zien, verdeeld in increments en sprints. De groenrode balkjes geven aan hoever dat deel afgerond is.



Figuur 3: Overzicht sprint

Deze afbeelding geeft een vergelijkbaar overzicht als de vorige afbeelding, echter geeft deze afbeelding alle stories van een bepaalde sprint weer.

3.3 Versiebeheer

Voor het versiebeheer (revision control) is gebruik gemaakt van een SVN-server. Hoewel het weliswaar mogelijk is JIRA te koppelen met de SVN-server, is daar niet voor gekozen. Het onderzoeken van JIRA was al erg tijdrovend, maar de koppeling met SVN zou nog veel meer tijd gaan kosten.

3.4 Rolverdeling

Binnen dit project zijn alle vier de projectleden in beginsel gelijk aan elkaar. Wouter de Vries is de scrummaster, wat inhoudt dat de leiding over de dagelijkse stand-up meetings voor hem waren. Erik Hubers eindverantwoordelijke voor documentatie. Dennis Heuven is de notulist bij alle overleggen, zowel met de begeleiding als met de opdrachtgever.

Door de dagelijkse stand-up meetings zijn de projectleden van elkaar op de hoogte. In geval van uitval zouden de werkzaamheden gemakkelijk opgevolgd kunnen worden. Afhankelijk van het soort werk, wordt het door degene overgenomen met de meeste kennis op dat gebied. De tabel hieronder geeft dit weer.

| Projectonderdeel | Projectlid | Fall-back projectlid |
|-------------------------------|----------------|----------------------|
| Website/webserver | J. van Leeuwen | W. de Vries |
| Nodes | D. Heuven | E. Hubers |
| Gateway | W. de Vries | D. Heuven |
| Documentatieverantwoordelijke | E. Hubers | J. van Leeuwen |

Tabel 2: Overzicht rollen en evt. overname werkzaamheden

4 Vooronderzoek

Voor de realisatie van het systeem zijn, in alle denkbare oplossingen, sensoren nodig voor het meten en/of tellen van personen in de te monitoren ruimten. Zonder nog de architectuur van het systeem bepaald te hebben, is er onderzoek gedaan naar verschillende detectiemogelijkheden. Deze onderzoeken zijn allemaal uitgevoerd aan het begin van het project. Er is onderzoek gedaan naar de Kinect, bewegings/warmte-sensoren, RFID en Deursensoren (objecttellers bij deuren). Aan de hand van deze onderzoeksresultaten is de verdere omzet van het systeem bezet.

4.1 Kinect

Om personen daadwerkelijk te kunnen detecteren en herkennen is een camera nodig. Veel andere sensoren zijn bijvoorbeeld slechts in staat beweging te detecteren of om de in- en uitgaande personen bij een deur te



Figuur 4: de Kinect

tellen. Ze hebben als bijkomend nadeel dat deze altijd aangezet moet worden met een bekende beginsituatie. Er is specifiek voor de

Kinect gekozen omdat hier op het moment van onderzoek veel mee gedaan werd. Er zijn een aantal open-source-projecten die er gebruik van maakten (OpenKinect) (OpenNi).

De Kinect is een camera van Microsoft die ontworpen is om gebruikt te worden met de spelconsole van hetzelfde bedrijf. Het combineert een infraroodcamera met een gewone camera en is daarmee in staat om hieruit een 3D-beeld te creëren. Oorspronkelijk was het apparaat bedoeld om digitale spellen mee te besturen.

4.1.1 Waarom de Kinect?

Het feit dat de camera al in meerdere open-source projecten wordt gebruikt. Hierdoor zijn er een aantal voorbeeldapplicaties met broncode beschikbaar. Daarnaast genereert deze camera diepte-informatie aan de hand van een infraroodsensor. Hiermee is het mogelijk een hogere betrouwbaarheidsgraad te behalen. Andere camera's met vergelijkbare mogelijkheden zoals de Asus Xtion Pro (Asus) zijn duurder of kennen minder support vanuit de open-source community.

4.1.2 Hoe werkt het?

Om een camera daadwerkelijk nuttig te laten zijn in dit project dient er beeldherkenningssoftware aan gekoppeld te worden. Hiervoor zijn verschillende mogelijkheden zoals IOImage en PrimeSense NITE (Zie Appendix I: Onderzoeksverslag

Kinect). Voor de beeldherkenningssoftware is een server nodig, hiervoor is de keuze zeer groot. De belangrijkste vereiste hiervoor is dat deze Windows of Linux draait en over een processorcapaciteit beschikt die voldoende is om een “redelijke” framerate te halen.

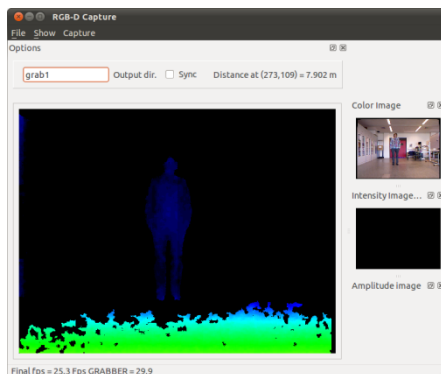
In dit project is gebruik gemaakt van het softwarepakket “Kinect RGB Demo v0.4.0” van Nicolas Burrus. Dit is een van de weinige vrij verkrijgbare softwarepakketten die momenteel beschikbaar zijn (Burrus).

4.1.3 Voordelen

- Door een camera te gebruiken is het mogelijk om op een willekeurig moment vast te stellen hoeveel personen waar zijn. Sommige andere sensoren dienen al aan te staan voordat de eerste mensen arriveren om zo een nulpunt te hebben;
- Een camera kan theoretisch ook personen herkennen die bijvoorbeeld bewusteloos op de grond liggen.

4.1.4 Nadelen

- De implementatie van een beeldherkenningsalgoritme is niet triviaal;
- Betrouwbaarheid van herkenning laat in de praktijk te wensen over (zie Appendix I: Onderzoeksverslag Kinect);
- Beeldherkenning kost veel rekenkracht, er zijn dus veel dure servers nodig;
- Betrouwbaarheid is sterk afhankelijk van de inrichting van de ruimte. Objecten zoals stoelen en whiteboards kunnen de herkenning lastiger of zelfs onmogelijk maken.



Figuur 5: Persoon op 3 meter afstand

4.1.5 Conclusie

De betrouwbaarheid van de combinatie van de gebruikte software met de Kinect bleek lager dan verwacht. Daarnaast zijn de kosten voor de benodigde servers voor de verwerking van de beelden dusdanig hoog dat het lastig zou zijn dit in de praktijk uit te voeren. Het feit dat het zelf ontwikkelen van herkenningssoftware erg lastig is en buiten de scope van dit project ligt gaf de doorslag om de Kinect, en daarmee alle camera's in het algemeen, niet te gebruiken.

4.2 Deursensoren



Figuur 6: Deursensor

Een van de sensorkeuzeonderzoeken is gedaan naar de deursensoren. Het idee kwam doordat de aanwezigheid het makkelijkst bij de in- en uitgang te meten is. Door te tellen hoeveel mensen er in- en uit gaan is het mogelijk om aan te geven hoeveel mensen er nog aanwezig zijn.

De deursensor is een sensor die een constant signaal uitstuurt wat gereflecteerd wordt door een reflector. De deursensor controleert of het signaal niet wordt onderbroken.

Binnen het systeem zijn altijd 2 deursensoren gekoppeld, en deze zitten vlak naast elkaar. Doordat één van de 2 sensoren eerder onderbroken wordt, is bekend welke kant de beweging op gaat, zodat de teller verhoogd ofwel verlaagd wordt. Het is dus altijd duidelijk of de persoon de ruimte in of uit ging.



Figuur 7: Schematische weergave deursensor met reflector

4.2.1 Voordelen

- De deursensor is redelijk ongevoelig voor storingen, en het is duidelijk of de sensor een constant of een onderbroken signaal heeft.
- De deursensor is robuust, en zal niet snel kapot gaan.
- De deursensor heeft een bereik van 4 meter, waarmee het op de meeste deuren past.

4.2.2 Nadelen

- De deursensor gaat af bij elk object dat door de straal komt, dus ook niet-mensen zullen geteld worden. Hierdoor kan er door kwaadwillende geknoeid worden met het resultaat.

4.2.3 Conclusie

Deze sensor is gekozen omdat de nadelen niet opwegen tegen de voordelen. Met deze sensor is het mogelijk om een idee te krijgen hoeveel mensen (of objecten) in de ruimte aanwezig zijn. Deze sensor kan het best gecombineerd worden met een sensor die

bewegings- en warmtevelden meet om zeker te weten of het mensen zijn die binnen zijn gekomen.

4.3 Warmte-/bewegingssensoren

Een van de sensorkeuzeonderzoeken is gedaan naar bewegingssensoren. De nadruk is hierbij gelegd op de bewegingssensoren die de verplaatsing van hittevelden detecteren (er is dan beweging).

Dit is gedaan omdat dit type het laagste aantal foutieve meldingen geeft, en dat deze sensoren relatief goedkoop zijn.

De bewegingssensor is een sensor die de verplaatsing van warmtevelden detecteert. De sensor detecteert alleen objecten die warmtevelden hebben die niet gelijk zijn aan de omgevingstemperatuur. Hieronder vallen dus mensen en dieren, maar geen ventilatoren, vallende objecten of anderszins niet-warme objecten.

4.3.1 Voordelen

Het voordeel van de bewegingssensor is dat er een andere controle wordt toegevoegd in een ruimte waar ook andere sensoren hangen.

Het is vrijwel zeker dat er mensen zijn als deze sensor dit aangeeft, mits het bereik van de sensor alleen in de ruimte ligt.

4.3.2 Nadelen

Een nadeel van de bewegingssensor is dat het alleen informatie geeft over het feit of er beweging was. Het is niet mogelijk om via deze sensor te bepalen hoeveel mensen er daadwerkelijk zijn.

4.3.3 Conclusie

Deze sensor voegt veel extra betrouwbaarheid toe aan het systeem. Zeker in combinatie met bijvoorbeeld een personenteller, omdat er met deze sensor een extra zekerheid ingebouwd kan worden.

Er is daarom besloten om verder te gaan met deze sensor op weg naar het uiteindelijke product.

4.4 RFID

Een optie die in overweging is genomen voor de detectie van personen, is RFID (Radio frequency identification). Dit is een technologie om van een afstand informatie op te slaan in en af te lezen van zogenaamde RFID-tags. RFID-tags zenden een radiogolf met data uit als ze in de buurt komen van een RFID-lezer. De meestgebruikte RFID-tags zijn passief, dit

betekent dat ze de energie die nodig is om het signaal te zenden uit de uitgezonden radiogolf van de bron halen. Hierdoor is geen batterij nodig in de tag zelf.



Figuur 8: RFID grid

Er is gekeken naar een manier waarop mensen gedetecteerd kunnen worden door middel van RFID. Een vereiste is dat iedereen in het openbare gebouw in bezit is van een RFID tag. Er zijn twee manieren waarop een RFID tag gebruikt kan worden. De eerste optie is door het gebruik van lezers bij elke deur in het gebouw, zodat personen zich aan moeten melden alvorens een ruimte in te gaan. Een tweede optie is het opzetten van een RFID grid. Dit is een cluster van RFID lezers in bijvoorbeeld het systeemplafond. Hierdoor is nauwkeurige plaatsbepaling mogelijk zonder dat personen verplicht zijn een handeling uit

te voeren bij deuren. Dit door het meten van de sterkte van het RFID-tag signaal bij de verschillende readers, door een combinatie van deze signalen kan de locatie bepaald worden. Hieronder een illustratie van een dergelijk grid.

4.4.1 Voordelen

Het enige voordeel van een oplossing met RFID is de betrouwbaarheid. Het is immers een bewezen techniek die al jarenlang voor allerlei doeleinden gebruikt wordt.

4.4.2 Nadelen

Er zijn echter enkele kanttekeningen te plaatsen bij het gebruik van RFID tags. Ten eerste is het noodzakelijk dat ieder persoon een RFID-tag bij zich draagt. Dit is, zeker voor de doelgroep van dit onderzoek 'openbare gebouwen' een lastig te realiseren eis. Tevens moet bij het gebruik van toegangspasjes ieder persoon zich aanmelden ook als een deur al geopend is door een voorganger. Bij het gebruik van een grid structuur moeten in elke ruimte meerdere RFID-readers geplaatst worden wat hoge aanlegkosten met zich meebrengt. Een ander nadeel is de inbreuk op de privacy, immers wordt van iedereen bijgehouden wie waar is op welk moment, waar andere sensoren vaak anoniem meten.

4.4.3 Conclusie

Door deze nadelen is besloten de RFID-mogelijkheden niet verder te onderzoeken ofwel praktisch te testen.

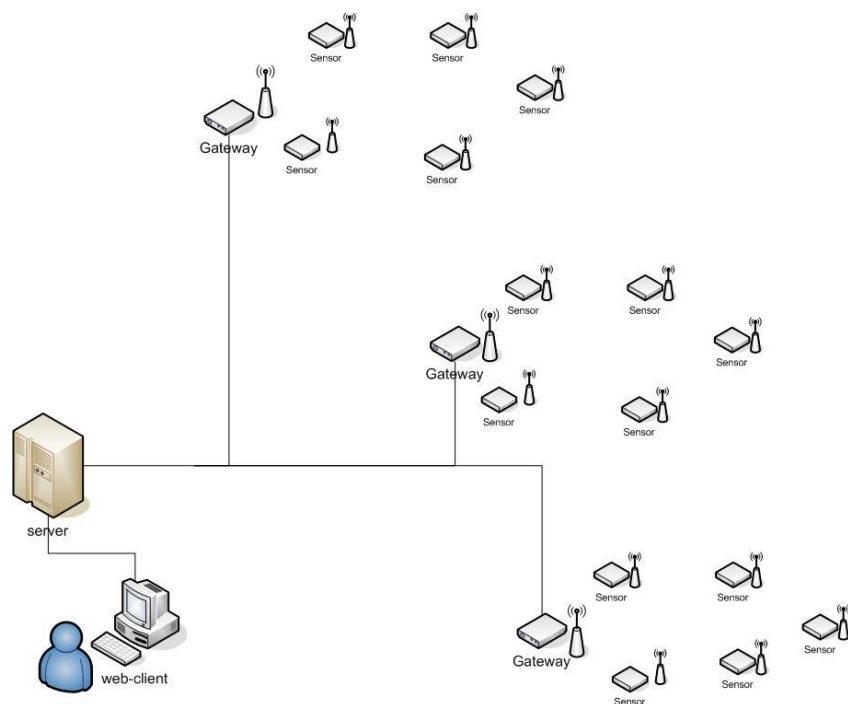
5 Architectuur

In dit hoofdstuk wordt de algehele architectuur uit de doeken gedaan. Er wordt verder ingegaan op het daadwerkelijke ontwerp van de implementatie van het systeem. Het hoofddoel van het project is het kunnen detecteren van personen in een openbaar gebouw en het inzichtelijk maken. Het systeem bestaat uit een aantal vereiste onderdelen om dit doel te bereiken.

- Er zijn sensoren nodig om de personen te detecteren;
- Er is een communicatie infrastructuur nodig voor het sturen van data van de sensoren naar de weergave;
- Er is een opslagmechanisme nodig om de sensordata op te slaan;
- Er is een weergave nodig voor de eindgebruikers.

Aan de hand van deze basisonderdelen is er een systeem ontworpen dat deze subonderdelen faciliteert om hiermee het doel, het detecteren van mensen en het weergeven aan eindgebruikers te bereiken. In de onderstaande afbeelding staat een uitwerking van het compleet uitgewerkte systeem van sensor tot weergave. In de volgende paragrafen wordt er

dieper ingegaan op de verschillende onderdelen.



De wolkjes geven de sensoren aan met bijbehorende gateway. Deze zijn bedraad verbonden met de server. Een web-client kan de server via een normale browser bereiken.

Figuur 9: Weergave algehele systeem

5.1 Website

Hieronder staan de meest essentiële websiteonderdelen beschreven.

5.1.1 Loginmodule

De loginmodule is voor de eindgebruikers (beheerders) van het systeem. Op dit moment worden de wachtwoorden nog niet geëncrypt opgeslagen, maar het is de bedoeling dit wel snel te doen.



Figuur 10: Loginscherm website

De loginmodule werkt als volgt:

- De gebruiker opent de website en ziet een login-scherm, deze vult hier zijn gegevens in;
- Gegevens worden verstuurd naar de Servlet HandleLogin, deze verifieert via de singleton Logins of de gebruiker in mag loggen!
- Als de gegevens correct blijken, wordt de gebruikersnaam in een sessie opgeslagen. Er wordt een hash gegenereerd van het IP-adres en gebruikersnaam. Deze hash wordt aan de serverkant opgeslagen;
- Bij elke pagina die de gebruiker laadt wordt de hash opnieuw gegenereerd met de gebruikersnaam (uit de sessie) en het IP adres. Als deze hash overeenkomt met de hash die opgeslagen is in Logins, is de gebruiker nog steeds ingelogd en wordt de pagina geladen (zometertijd verschijnt het login-scherm weer).

Doordat de hash elke pageload opnieuw gegenereerd wordt met het IP-adres van de client, is het niet mogelijk de hash te kopiëren naar een andere machine, wat het systeem veiliger maakt.

Bij alle formulieren die verzonden worden (naar een Servlet) wordt ook de hash mee verstuurd, dit om te voorkomen dat gebruikers van buitenaf onrechtmatig Servlets benaderen.

5.1.2 Accountbeheer

Via het accountbeheer is het mogelijk accounts van gebruikers te verwijderen en toe te voegen. Dit verloopt via de JSP accountcontrol.jsp. Het daadwerkelijke toevoegen en verwijderen gaat via de servlet AccountControl.

5.1.3 Kaartweergave

De bedoeling van de kaartweergave is een eenvoudig overzicht te geven van de situatie. Dit wordt gedaan door aan elke ruimte een kleur te geven die afhankelijk is van de door het beslissingsalgoritme gegenereerde score. Tevens wordt deze score letterlijk op de kaart gezet.



Figuur 11: Kaartweergave

De kaarten kunnen via de webinterface geüpload worden en het is de bedoeling dat via deze weg ook de ruimtes aangegeven kunnen worden. Momenteel moet dit rechtstreeks in de database omdat de story om dit verder uit te breiden niet gekozen is.

De kaartweergave bestaat uit twee servlets, MapHandler en MapDisplay. MapHandler is verantwoordelijk voor de pagina zelf, het uploaden van kaarten en in de toekomst ook het toevoegen van ruimten op de kaart. MapDisplay genereert op basis van de geüploadde kaart en de toegevoegde ruimten een afbeelding zoals te zien is in de figuur hierboven. Dit wordt gedaan m.b.v. Java's Image library.

Aangezien de ruimten uit een $n > 2$ aantal punten kunnen bestaan is er in de database een tabel die er voor zorgt dat er een arbitrair aantal coördinaten per ruimte kunnen worden toegevoegd. Over deze coördinaten wordt vervolgens een doorzichtige polygoon getekend. De coördinaten met de plek voor de tekst worden ook in de database opgeslagen maar in de tabel van de ruimte zelf.

5.2 Bepaling prioriteit ruimte

In dit hoofdstuk staat beschreven hoe de logica bepaald wordt binnen het systeem. Met andere woorden: hoe wordt bepaald welke ruimte een hogere prioriteit krijgt en daarmee hoger in de resultatenlijst komt.

De berekeningen zijn gebaseerd op een schatting, ze zijn zodanig gemaakt dat een ruimte met veel beweging en/of mensen een hogere waarde krijgt dan een ruimte met weinig beweging of weinig mensen. Er is getest en geverifieerd dat de uitkomsten kloppen, echter zijn de uitkomsten per ruimte niet met elkaar vergeleken. Zoals in de aanbevelingen ook omschreven staat, moet dit nog wel gebeuren.

In een ruimte hangen een of meerdere sensoren. Per ruimte worden deze met elkaar gecombineerd. Hiermee wordt zowel rekening gehouden met de betrouwbaarheid als met het aantal mensen dat in de ruimte is.

Hoe hoger de uitkomst, hoe groter de kans dat er mensen in een ruimte zijn.

5.2.1 Berekening

Voor elke sensor geldt een andere berekening.

Noot: negatieve eindresultaten zijn niet mogelijk, omdat er per ruimte geen negatieve deelresultaten mogen zijn.

5.2.1.1 Deursensors

(Het aantal mensen die vandaag naar binnen zijn gegaan – het aantal mensen dat vandaag naar buiten is gegaan) * 10.

Indien er meerdere deursensoren in een ruimte aanwezig zijn dienen de in- en uitgaande waarden bij elkaar opgeteld te worden.

5.2.1.2 Bewegingssensors

500 – het aantal seconden dat de laatste beweging gemeten is.

Indien er meerdere bewegingssensoren in een ruimte aanwezig zijn telt alleen de sensor die de meest recente beweging gedetecteerd heeft.

5.2.2 Voorbeelden

1 deursensor, 2 bewegingssensoren

- De deursensor heeft 5 mensen binnen zien komen;
- De deursensor heeft 2 naar buiten zien gaan;
- Bewegingssensor 1 heeft anderhalve minuut geleden voor het laatst beweging gemeten;
- Bewegingssensor 2 heeft 3 minuten geleden voor het laatst beweging gemeten.

$$((5 - 2) * 10) + (500 - 90)$$

$$(5 - 2) * 10 = 30$$

$$(500 - 90) = 410$$

$$30 + 410 = 440$$

2 deursensoren, 2 bewegingssensoren

- De deursensor 1 heeft 25 mensen binnen zien komen;
- De deursensor 1 heeft 2 mensen naar buiten zien gaan;
- De deursensor 2 heeft 0 mensen naar binnen zien gaan;
- De deursensor 2 heeft 20 mensen naar buiten zien gaan;
- Bewegingssensor 1 heeft anderhalve minuut geleden voor het laatst beweging gemeten;
- Bewegingssensor 2 heeft 6 minuten geleden voor het laatst beweging gemeten.

$$((25+0)-(2+20)) * 10 + (500 - 90)$$

$$(25-22) * 10 = 30$$

$$(500 - 90) = 410$$

$$30 + 410 = 440$$

User stories: 129, 145, 146

5.3 Server

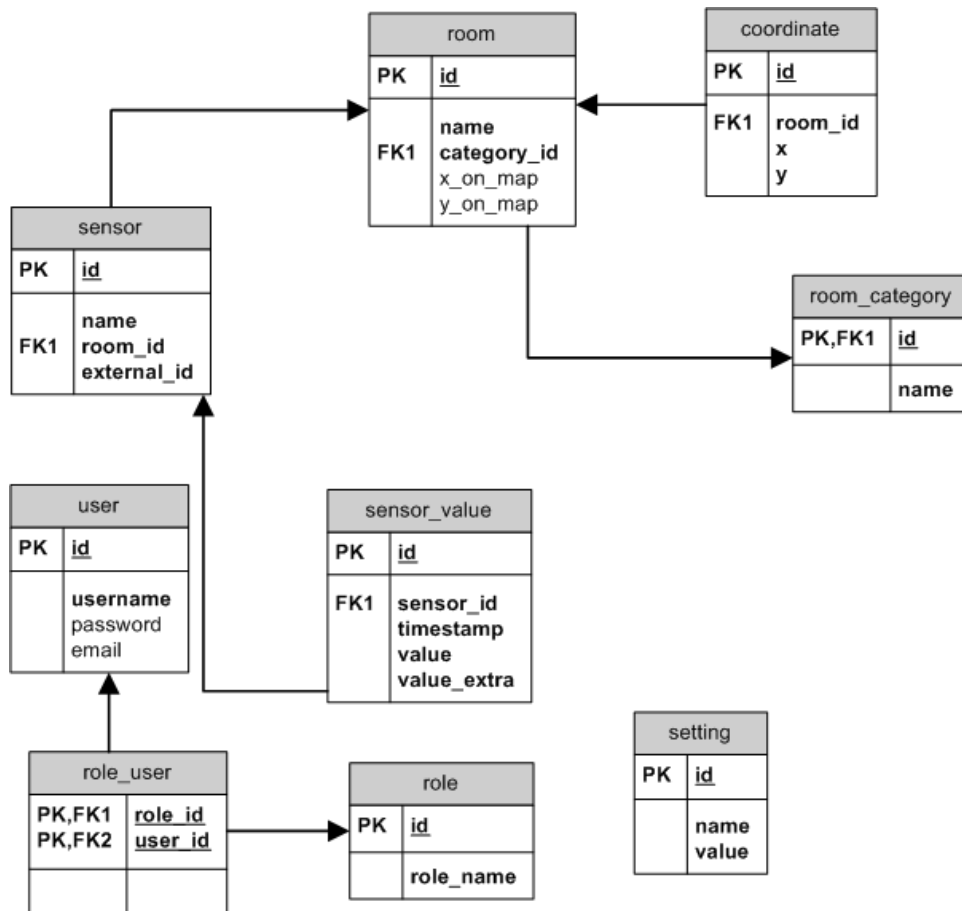
De server is het hart van het systeem, alle logica wordt afgehandeld op dit onderdeel. Op de server komt alle data vanuit de gateways binnen. De server draait een MySQL-server (v 5.5.8) en Apache Tomcat 7.0.

5.3.1 Database

Hieronder staat de databasestructuur weergegeven. De databasestructuur is bedacht op het moment dat duidelijk werd dat er vanuit verschillende plekken data binnen zou komen en de wens was dit bij elkaar te krijgen.

Vanuit gesprekken met de BHV op het Saxion is gebleken dat er rekening gehouden moest worden met het feit dat ontruimingen vaak per afdeling of vleugel gaan, hierdoor is de scheiding van categorie aangebracht.

Het ontstaan van de overige tabellen spreekt voor zich. Verderop in dit hoofdstuk staat per tabel de preciese functie nader verklaard.



Figuur 12: Databaseschema

| | |
|----------------------|--|
| User | Tabel met alle gebruikers die in kunnen loggen op de website. |
| Role | Verschillende rollen die de gebruikers kunnen hebben op de website (N:N-relatie met koppeltabel role_user) |
| Sensor | Opslag van de verschillende sensoren |
| Sensor_value | Opslag van ruwe sensordata. Bevat enum met als waarden IN, OUT, MOVEMENT, NO_MOVEMENT en COUNT (bij COUNT wordt nog een extra waarde ingevuld) |
| Room | Overzicht met ruimtes waarin een sensor geplaatst kan zijn (x_on_map en y_on_map worden gebruikt voor de mapweergave) |
| Room_category | Categorie (vleugel/afdeling) waar de ruimte zich bevindt |
| Coordinate | coördinaten van een ruimte (voor gebruik mapweergave) |
| Setting | Opslag van instellingen (niet in gebruik); |

Tabel 3: Verklaring databasetabellen

5.3.2 ORM: Hibernate

Als ORM (object relational mapping) is gebruikt gemaakt van Hibernate. Omdat voor alle projectleden Hibernate nog onbekend was, was het een uitdaging en kostte het veel tijd dit uit te zoeken.

Voor elke tabel is een java-klasse gemaakt en bijbehorend een mappingfile. Zowel de klasse als de mappingfile staan in de package hibernate. Hibernate's configuratiefile staat in WebContent\WEB-INF\classes\hibernate.cfg.xml.

Door hibernate te gebruiken kan vanuit de java-programmatuur gewerkt worden met objecten en hoeven er niet voor elke losse actie queries geschreven te worden. Gelinkte tabellen worden gelinkt aan de hand van andere objecten, en niet aan de hand van ID's.

5.4 Gateways

Om de sensordata van de sensoren te verzenden naar de webserver wordt de data verstuurd door het meshnetwork. Er is gekozen om een gateway binnen het meshnetwork te plaatsen om de data te versturen naar de webserver. Dit om de grootte van het meshnetwork te beperken, zodat een pakket van een sensor niet een heel gebouw door gestuurd hoeft te worden om uiteindelijk uit te komen bij de webserver. Dit zorgt voor lange bezorgtijden en veel dataverkeer door het meshnetwork, gezien elke node in het meshnetwork de data rebroadcast. Tevens biedt het gebruik van een gateway de mogelijkheid om de webserver buiten het pand waar het meshnetwork zich bevindt te plaatsen.



Figuur 13: De Sheevaplug

Er is een keuze gemaakt uit verschillende systemen die voldoen aan de eisen. Enkele eisen die gesteld zijn aan de gateway zijn : de beschikking over een ethernet-poort (voor de communicatie naar de webserver); een compact formaat (voor gemakkelijke plaatsing boven bijvoorbeeld een systeemplafond) en een USB-aansluiting (voor het aansluiten van een node om data te ontvangen van het meshnetwork). Verder bij voorkeur een laag stroomverbruik. De keuze is uiteindelijk gevallen op de Sheevaplug. Dit is een Linux-systeem ter grootte van een blikje frisdrank.

Er is een keuze gemaakt uit verschillende systemen die voldoen aan de eisen. Enkele eisen die gesteld zijn aan de gateway zijn : de beschikking over een ethernet-poort (voor de communicatie naar de webserver); een compact formaat (voor gemakkelijke plaatsing boven bijvoorbeeld een systeemplafond) en een USB-aansluiting (voor het aansluiten van een node om data te ontvangen van het meshnetwork). Verder bij voorkeur een laag stroomverbruik. De keuze is uiteindelijk gevallen op de Sheevaplug. Dit is een Linux-systeem ter grootte van een blikje frisdrank.

Hieronder enkele specificaties van de Sheevaplug.

| | |
|-----------------------|--|
| CPU | 1.2 Ghz ARM (Marvell Kirkwood 88F6281) |
| RAM | 512MB |
| Flash-geheugen | 512MB |
| Vermogen | 2.3W min / 7W max |
| OS | Debian |
| JTAG | Ja (mini-usb formfactor) |
| I/O Aparatus | USB 2.0, SD slot, 1GE, JTAG mini USB |

Tabel 4: Specificaties Sheevaplug

Om data van het meshnetwerk uit te lezen op de gateway wordt er, zoals eerder vermeld, een node aangesloten op de Sheevaplug. Dit gebeurt doormiddel van een USB kabel. De XBee nodes die hiervoor gebruikt worden communiceren serieel over USB. Om serieel over usb te kunnen communiceren is het noodzakelijk dat er een nieuwere versie van de debian kernel geflashed wordt op de Sheevaplug. Dit gezien de kernel die wordt meegeleverd op de Sheevaplug geen FTDI support heeft. FTDI support is benodigd voor de USB-naar-Serieël-converter die in veel adapters gebruikt wordt.

Voor het verzenden van data naar de server draait een programma geschreven in Java. De keuze voor Java is gemaakt gezien Java platformonafhankelijk is waardoor het door middel van weinig of geen aanpassingen ook op een ander systeem of platform kan draaien. Het programma luistert continu naar de seriële interface van de node en stuurt ontvangen, valide pakketten door naar de webserver over een SSL-verbinding (HTTPS). Tevens worden alle valide datapakketten opgeslagen in een circular buffer van 10 items, dit om te voorkomen dat een rebroadcast van een node er voor zorgt dat een pakket twee keer wordt verzonden naar de webserver.

5.5 Communicatie tussen gateway en server

De communicatie tussen de server en de gateway verloopt via webservices op de server. Deze is aan te roepen via een HTTP POST-request. Deze is te bereiken op <server-adres>/InsertSensorData en verwacht de volgende parameters

| Parameter | Omschrijving | Evt. opmerking |
|------------|---|------------------------------------|
| sensor | Hardware-ID van de XBee | |
| value | MOVEMENT als de sensor beweging heeft gedetecteerd; | |
| | NO_MOVEMENT als de sensor heeft gedetecteerd dat er de beweging opgehouden is; | |
| | COUNT als de sensor een aantal personen geteld heeft; | Wordt op dit moment niet gebruikt. |
| | IN als de sensor gemeten heeft dat iemand de ruimte is binnengekomen; | |
| | OUT als de sensor gemeten heeft dat iemand de ruimte heeft verlaten. | |
| extravalue | Alleen te gebruiken bij value.COUNT om het precieze aantal mensen aan te geven. | |
| apikey | Zie beveiliging | |

Tabel 5: Parameters bij data-invoer webserver

De server geeft altijd een van de volgende HTTP-status-codes terug.

| | |
|------------------|--|
| 200 OK | De invoer is verwerkt |
| 400 Client Error | Syntaxfout |
| 503 Server Error | Er is een fout opgetreden aan de serverkant, invoer niet of foutief verwerkt |
| 403 Not Allowed | API-key is ongeldig. |

Tabel 6: HTTP-status-codes na data-invoer webserver

5.5.1 Beveiliging

De beveiliging tussen de gateway en de server is geregeld via een API-key. Deze API-key is bekend op de gateway en dient bij elke POST meegestuurd te worden naar de server (parameter 'apikey').

Daarnaast wordt er gebruikt gemaakt van een SSL-verbinding (HTTPS), waardoor de API-key niet achterhaald kan worden.

5.6 Nodes

User stories: 125, 132, 135, 166

Om mensen te detecteren worden sensoren gebruikt, deze sensoren worden uitgelezen en de gemeten gegevens worden verzonden naar de webserver. Hiervoor zijn de sensornodes bedacht. Een sensornode bestaat uit een sensor, een wireless-module en een PCB om deze met elkaar te verbinden.

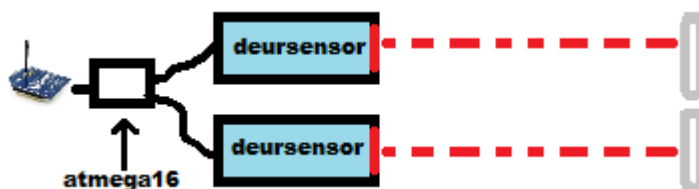
Daarnaast is er ook nog een simpelere variant, een gewone node. Deze heeft maar één taak: het doorsturen van de ontvangen gegevens. Deze heeft geen sensoren en genereert zelf dus geen sensordata (al worden er wel alive-messages gestuurd).

5.6.1 Analyse & ontwerp

Om de sensoren uit te lezen is een microcontroller nodig. Om de mobiliteit zo hoog mogelijk te houden voor de nodes is het gewenst dat de data draadloos wordt verzonden, waarvoor een module nodig is die draadloos kan verzenden. Ook is het wenselijk dat de nodes zo klein mogelijk zijn, zodat de nodes overal simpel geplaatst kunnen worden. Voor dit doel is er een klein bordje nodig waarop alle componenten van de nodes zitten.

Als microcontroller is er voor de Atmega16 microcontroller van Atmel (Atmel Corporation) gekozen omdat deze chips voldoende rekenkracht hebben om de taken uit te voeren, goedkoop zijn en bovendien ruim voorradig waren.

Als draadloze module is er gekozen voor de Digi XBee S1 van SparkFun Electronics (SparkFun Electronics) omdat deze module goedkoop is en een makkelijk te gebruiken API heeft.



Figuur 14: Ontwerp node

5.6.2 Inhoud van de nodes

De nodes bevatten 1 Atmel Atmega16 microcontroller (Atmel Corporation) voor de berekeningen en 1 Digi XBee S1 (SparkFun Electronics) voor de draadloze communicatie en eventueel een sensor.

De combinatie van de XBee, de atmega16 en eventueel de sensor worden geplaatst op het ontwikkelde sensorbordje (PCB).

5.6.3 Wireless module

Om de gegevens draadloos te versturen zijn draadloze modules nodig.

Er is gekozen voor de Digi XBee S1, omdat deze goedkoop zijn en een makkelijk te gebruiken API heeft.

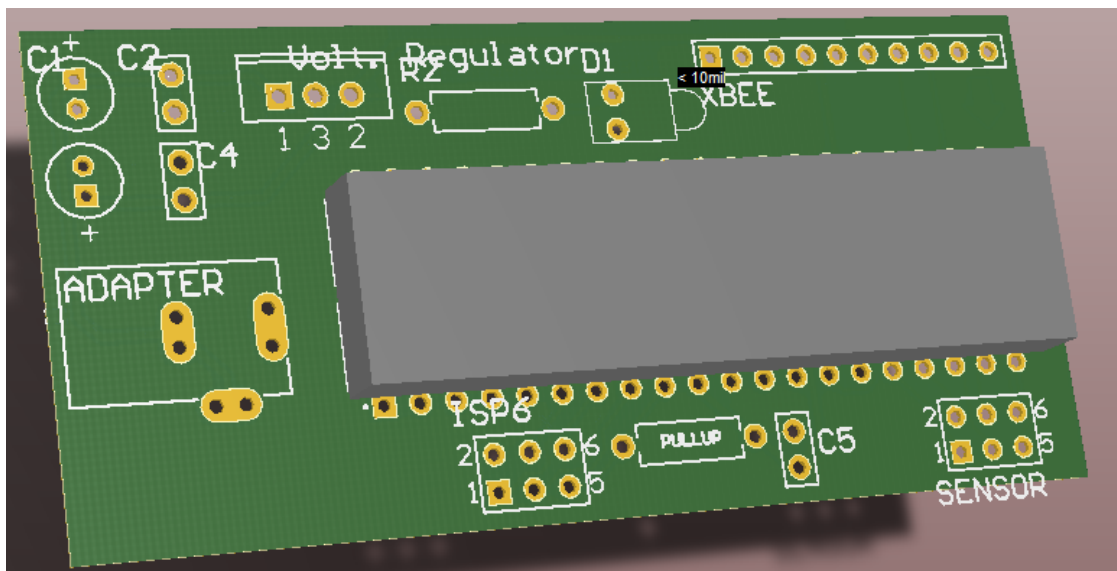
Deze module heeft een bereik tot 90 meter (zonder obstakels/stoorzenders), en heeft een zend capaciteit tot 250 kb/s.



Figuur 15: XBee-module

5.6.4 PCB

Om de nodes zo klein mogelijk te maken - en de kosten te drukken - is een printplaat ontworpen voor de nodes. Deze printplaat bevat alle benodigde pinnen voor een node. Op dit bordje worden de XBee, de sensoren en de microcontroller geplaatst. Het geheel wordt gevoed door een 12V-adapter.



Figuur 16: Het sensorbordje (zonder geplaatste onderdelen)

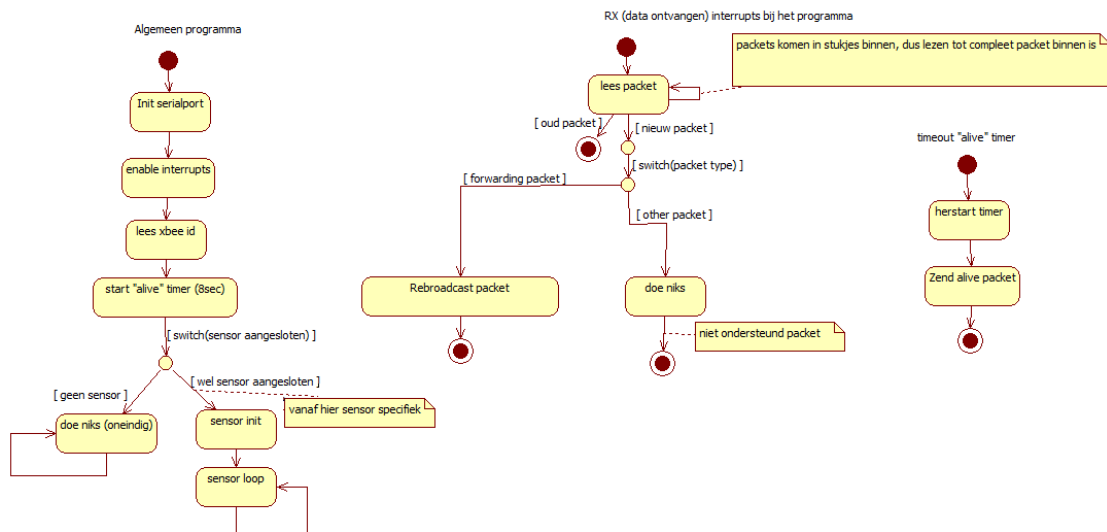
5.6.5 Atmega16 programmatuur

Het programma voor de Atmel Atmega16 heeft de volgende taken:

- Aansturing communicatie (i.c.m. wireless module, ons protocol)
- Uitlezen sensoren

Ook is er gaandeweg het project een extra requirement bijgekomen waaraan de Atmega16 moest voldoen: Een heartbeat versturen (controle of het systeem nog werkt).

Uit deze eisen is het volgende ontwerp gekomen:



Figuur 17: ontwerp atmega16

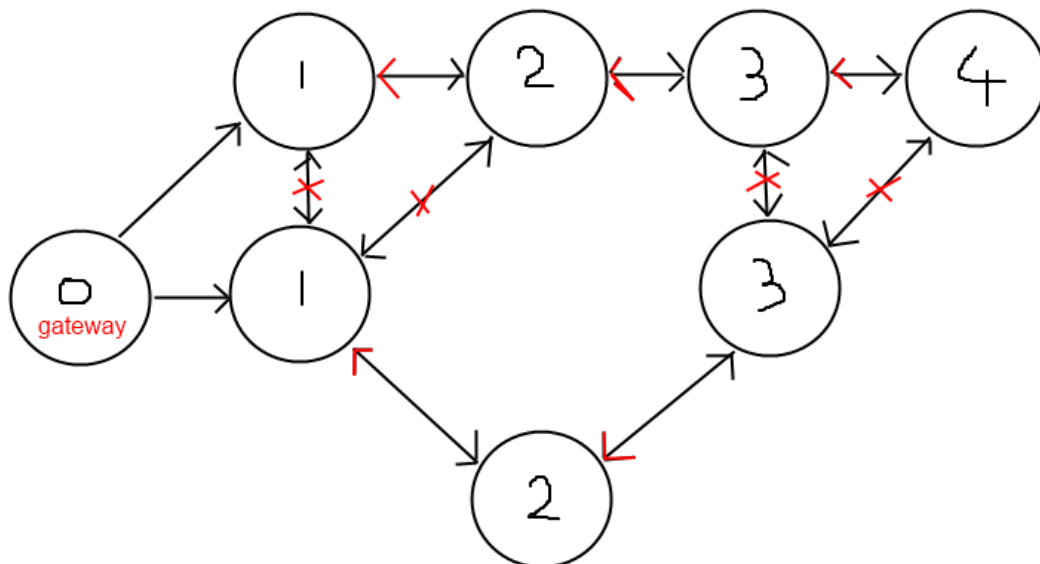
5.7 Routeringsalgoritme

User stories: 143, 144

In eerste instantie is uitgegaan van een routingsalgoritme met een routingstabel. Later bleek dat dit in zijn huidige opzet te onbetrouwbaar was. Dit is getest door nodes met vaste tussenafstanden te verspreiden door een gang. Bij meer dan 1 à 2 hops bleek de kans op een succesvolle verzending van een pakket dicht bij 0 te liggen. Dit komt waarschijnlijk doordat het kortste pad ook vaak onbetrouwbaar is. Als er bijvoorbeeld 4 nodes op een rij liggen zal node 1 node 3 proberen te bereiken zonder hierbij gebruik te maken van node 2. Door gebruik te maken van een floodingprotocol zonder routingstabel wordt dit probleem opgevangen omdat elke node het pakket sowieso doorstuurt.

5.7.1 Met routingstabel

Omdat de gekozen transceivers geen ingebouwde ondersteuning voor mesh networking heeft, moest er een eigen algoritme m.b.v. een microcontroller geïmplementeerd worden. Hiervoor zijn zeer veel (complexe) mogelijkheden. Om het eenvoudig te houden is er gekozen voor een algoritme dat gebaseerd is op Dijkstra. Met dit algoritme kan data alleen richting de gateway bewegen en routingpakketten alleen in de tegenovergestelde richting.



Figuur 18: Routeringsalgoritme

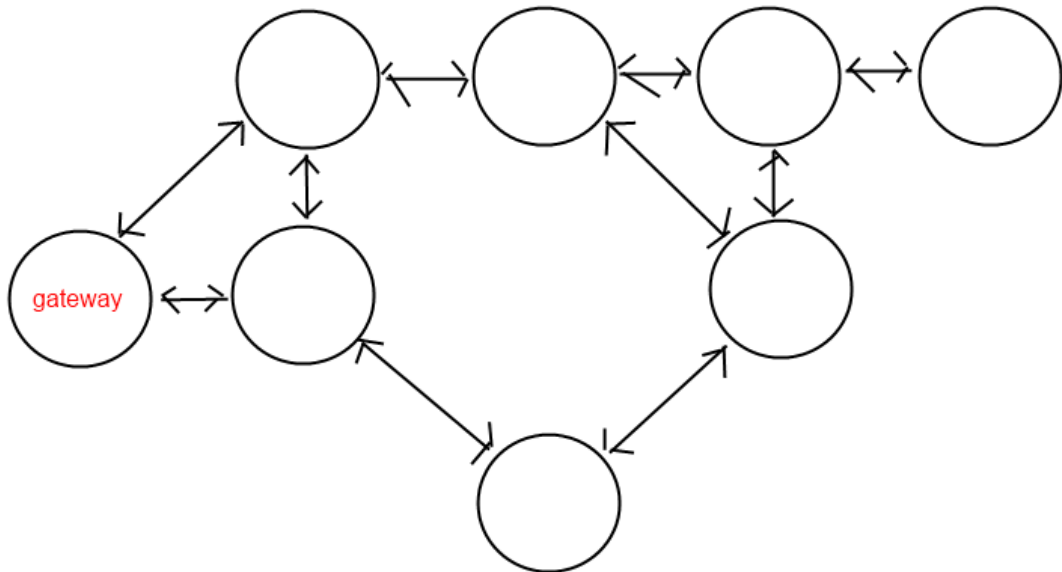
In het bovenstaande afbeelding is de richting van data goed te zien. De rode pijlen geven telkens de richting van data aan en de zwarte pijlen de richting van routingpakketten. Een rood kruis geeft aan dat dit pad niet gebruikt wordt (bijvoorbeeld omdat de routing cost daarmee hoger zou worden). Doordat het algoritme zo beperkt is, is het eenvoudig te implementeren en heeft het weinig geheugen nodig.

In pseudocode ziet dit er als volgt uit:

- Node heeft de variabelen *shortestPath* en *address*;
- Gateway broadcast "route" pakket met *cost 0*;
- Node ontvangt pakket en vergelijkt het met zijn huidige *shortestPath*;
- *Indien lager:*
 - *update de address variable met de bron van het ontvangen pakketje.*
 - *Verhoog de cost met 1*
 - *Rebroadcast het pakket*
- *Indien gelijk aan of hoger:*
 - *Doe niks.*

5.7.2 Zonder routingtabel

Een nog eenvoudiger routeringsalgoritme is het zogenaamde flooding-algoritme. Hierbij stuurt elke node elk pakket – mits 'ie het nog niet heeft ontvangen - door. Pakketten kennen hiertoe een pakketnummer welke in combinatie met het adres van de verzendende node de sleutel vormt. Het aantal pakketten wat het systeem onthoudt is eenvoudig aanpasbaar en is afhankelijk van de grootte van het netwerk. In de in dit project gebruikte testscenario's was een buffer van 10 afdoende.



Figuur 19: Routeringsalgoritme

De reden om hiervoor te kiezen is de hoge betrouwbaarheid. Een pakket wordt immers via alle mogelijke routes verstuurd. Een nadeel hiervan is dat het slecht schaalbaar is omdat het netwerk snel zwaar belast wordt, bovendien is het energie-onzuiniger.

In pseudocode ziet er als volgt uit:

- Node ontvangt pakket;
- Pakket zit al in ontvangenpakketlijst:
 - Pakket wordt genegeerd;
- Pakket zit nog niet in ontvangenpakketlijst:
 - Pakket wordt gerebroadcast;
 - Pakket wordt in de ontvangenpakkettenlijst gezet;
 - Indien de grootte van de ontvangenpakkettenlijst de maximum overschrijdt:
 - Verwijder oudste pakket uit de lijst.

6 Testen

Er hebben tijdens het project twee grote testen plaatsgevonden. De eerste was om de werking en betrouwbaarheid van de Kinect in combinatie met de beeldherkenningssoftware vast te stellen. De tweede test was om de betrouwbaarheid van de deursensoren te bepalen.

Bij de Kinect-test is eerst een enkele persoon uit de projectgroep gedetecteerd op 1, 2 en 3 meter afstand. De resultaten hiervan staan in Appendix II: Testresultaten. 1 meter bleek hierbij de minimale afstand te zijn en 3 meter de maximale afstand. Vervolgens hebben de verschillende personen uit de projectgroep verschillende situaties nagebootst waaronder het zitten aan een bureau. Hierbij is het aantal gedetecteerde personen vergeleken met het werkelijke aantal personen. Deze situaties zijn zo gekozen dat ze vergelijkbaar zijn met situaties die in het echt voorkomen.

Voor de deursensoren zijn de personen uit de projectgroep met verschillende snelheden door de deur gelopen, zowel in als uit en is telkens het aantal gemeten personen vergeleken met het werkelijke aantal personen.

De deursensor hangt op 94 centimeter van de grond.

6.1 Samenstelling groep

| Nr. | Lengte | Heuphoogte |
|-----|--------|------------|
| 1 | 1,97m | 104cm |
| 2 | 2,03m | 106cm |
| 3 | 1,76m | 92cm |
| 4 | 1,75m | 97cm |
| 5 | 1,80m | 94cm |

Tabel 7: Groepssamenstelling

Het is van belang rekening te houden met de heup- en sensorhoogte omdat dit invloed heeft op de meting. Er is gekozen om ongeveer heuphoogte aan te houden bij het plaatsen van de sensor omdat er bij de heup geen dubbele metingen kunnen plaatsvinden. Dit is wel het geval bij benen en armen. Bij bijvoorbeeld een klas schoolkinderen zou een andere sensorhoogte aangehouden moeten worden, omdat de testgroep een andere optimale sensorhoogte heeft. De resultaten van deze test zijn uitsluitend van toepassing op groepen van een soortgelijke lengte en heuphoogte.

Indien een grondigere test zou plaatsvinden, zal er gezocht moeten worden naar een groep die wat lengte en heuphoogte betreft meer varieert. Zie ook hoofdstuk 8. Aanbevelingen.

6.2 Testlocatie

De testen zijn gedaan in een lokaal van Saxion met 2 ingangen waarvan er 1 in gebruik is.



Figuur 20: Testlocatie Saxion Enschede

6.3 Conclusie

Bij het testen van de Kinect is gebleken dat de betrouwbaarheid van de beeldherkenning erg laag is. Voor het uitgebreide testresultaat zie Appendix II: Testresultaten.

6.4 Conclusies & Aanbevelingen

Indien een grondigere test zou moeten plaatsvinden, zal er gezocht moeten worden naar een groep die wat lengte betreft meer variëren. Daarnaast zal er een locatie gebruikt moeten die meerdere in- en/of uitgangen heeft. Ook zou de tijdsduur van de testen dan verhoogd moeten worden.

7 Conclusie

Nu, aan het einde van het project, kan worden gesteld dat er een sterk prototype staat. Zowel technisch als praktisch is het een gedegen systeem.

De onderzoeksvraag luidt als volgt:

Hoe kan, in een openbaar gebouw, gedetecteerd worden in welke ruimten zich personen bevinden? Hieronder staat voor elke deelvraag het concluderende antwoord. Daarmee wordt ook de hoofdonderzoeksvraag beantwoordt.

7.1 Met welke sensoren kunnen personen gedetecteerd worden

Na verschillende mogelijkheden onderzocht te hebben, is naar boven gekomen dat het werken met de Kinect geen haalbare oplossing is binnen de huidige projectgrenzen. De 2 sensoren die positief uit het onderzoek kwamen – en die ook gebruikt zijn in het prototype – zijn de deursensor en warmte-/bewegingssensor. Ook is gebleken dat RFID geen praktische oplossing is.

7.2 Hoe kan de door de sensoren gegenereerde informatie verstuurd worden

De gegevens die gegenereerd worden door de sensoren worden via de nodes met behulp van een flooding-protocol verstuurd naar de webserver. Dit is een betrouwbare methode.

7.3 Hoe kan de door de sensoren gegenereerde informatie inzichtelijk worden gemaakt voor de eindgebruiker

De gegevens zijn inzichtelijk via de website die draait op de server. Dit overzicht werkt en is voorzien basisfunctionaliteiten. Deze weergaves kunnen nog uitgebreider. De tijd liet het niet toe dit verder uit te breiden.

7.4 Wat is de betrouwbaarheid van de verkregen resultaten

De betrouwbaarheid van de verkregen resultaten zijn getest door middel van enkele kleine tests. Ook hier liet de tijd het niet toe uitgebreidere en diepgaandere tests uit te voeren. Hier ligt dus nog verbeterwerk.

8 Aanbevelingen

Binnen het project is er ruimte voor discussie. In de volgende paragrafen staan verschillende discussiepunten omschreven.

8.1 Schaalbaarheid

De schaalbaarheid is – zoals bij alle WPAN's – een discussiepunt. Op dit moment is het routeringsalgoritme nog gebaseerd op flooding, maar daarvan is bekend dat het niet schaalbaar is.

8.1.1 Routeringsprotocol

Zoals ook in het hoofdstuk 5.7 is beschreven, heeft de grootte van de circular buffer invloed op de schaalbaarheid. Om de schaalbaarheid te vergroten moet in ieder geval de circular buffer vergroot worden, ondanks dat dit ten koste gaat van de performance.

8.2 Meer testen, vergroting betrouwbaarheid

De betrouwbaarheid van het systeem is getest, echter is dit nog niet uitgebreid genoeg. Critici hebben recht van spreken als zij stellen dat de betrouwbaarheid niet gewaarborgd is. Als projectgroep is het dan ook aan te raden eerst verdere gedegen tests uit te voeren, alvorens uitbreidingen te maken aan het systeem ofwel aan de uitrol te beginnen.

Tevens kan de betrouwbaarheid van het systeem vergroot worden door meer sensoren per ruimte toe te voegen, maar dit zal de prijs per ruimte uiteraard verhogen.

8.2.1 Resultaat per ruimte

De formule die bepaald wat de uitkomst is per ruimte (alle sensoren in een ruimte bij elkaar) is, is gemaakt op basis van een schatting wat logische waarden zouden zijn. Bij uitgebreidere tests moet hier ook nog nader naar gekeken worden.

8.3 Uitbreiding website

De website kent nog veel verbeteringsmogelijkheden. Deze staan hieronder beschreven.

8.3.1 Gebruikerservaring webinterface verbeteren

Door de website beter vorm te geven wordt de gebruikerservaring verbeterd. Kleurenschema's en stijlschema's zijn op dit moment niet optimaal.

8.3.2 Loginmodule

Er is een loginmodule met rechtensysteem, echter zijn er nog geen mogelijkheden voor het beheer van deze rechten.

8.3.3 Encryptie wachtwoorden

Wachtwoorden worden nu nog niet opgeslagen met encryptie. Dit moet – voordat het systeem in productie gaat - absoluut wel gebeuren.

8.4 Stroomverbruik (UPS)

Om het systeem draaiende te houden als er stroomuitval is, is er een noodvoeding nodig. Deze zal dan moeten worden aangesloten op de sensorbordjes, de gateway en de server. Hier en daar kan ook gekeken worden naar de energiezuinigheid, zodat het systeem langer met de UPS kan blijven draaien.

8.5 Compact PCB

Er is een ontwerp gemaakt voor de compacte sensorbordjes, maar deze zijn nog niet gemaakt. Het is aangeraden om de bordjes te fabriceren en deze te gebruiken in plaats van de grote ontwikkelborden die nu gebruikt worden.

8.6 Beveiliging XBee Communicatie

De communicatie tussen de XBee's onderling is niet beveiligd. Dit betekent dat de pakketten zonder veel moeite uitgelezen of vervalst kunnen worden. Het wordt dus aangeraden om de beveiliging tussen de XBee's te realiseren.

8.7 Meer sensortypes toevoegen

Het systeem ondersteunt meerdere typen sensoren. Zo is het mogelijk om nieuwe typen sensoren toe te voegen. Zo zouden er een bijvoorbeeld video of infrarood camera's kunnen worden toegevoegd.

9 Referenties

Asus. (n.d.). *Xtion Pro*. Retrieved juni 8, 2011, from Asus eStore:

http://us.estore.asus.com/index.php?l=product_detail&p=3397

Atlassian. (n.d.). *Full Features List*. Retrieved juni 9, 2011, from Atlassian JIRA:

<http://www.atlassian.com/software/jira/full-features.jsp>

Atmel Corporation. (n.d.). *Atmega16*. Retrieved juni 8, 2011, from Atmel:

http://www.atmel.com/dyn/products/product_card.asp?part_id=2010

Burrus, N. (n.d.). *Kinect RGB Demo v0.4.0*. Retrieved juni 8, 2011, from Nicolas Burrus

Homepage:

<http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV4?from=Research.KinectRgbdDemoV3>

OpenKinect. (n.d.). *Main Page*. Retrieved juni 9, 2011, from OpenKinect: <http://openkinect.org>

OpenNi. (n.d.). Retrieved juni 9, 2011, from Introducing OpenNi: <http://openni.org>

SparkFun Electronics. (n.d.). *XBee 1mW Wire Antenna*. Retrieved juni 8, 2011, from

SparkFun Electronics: <http://www.sparkfun.com/products/8665>

Wikipedia. (n.d.). *JIRA*. Retrieved 6 9, 2011, from Wikipedia: <http://en.wikipedia.org/wiki/JIRA>

Wikipedia. (n.d.). *Scrum (softwareontwikkelmethode)*. Retrieved juni 7, 2011, from Wikipedia:

http://nl.wikipedia.org/wiki/Scrum_%28softwareontwikkelmethode%29

Appendix I: Onderzoeksverslag Kinect

Onderstaand document is het onderzoeksverslag zoals opgeleverd in het project.

Inleiding

In dit document zullen de verschillende aspecten van het gebruik van de Kinect en andere technologieën als middel voor persoon detectie uiteen worden gezet. De Kinect is een camera met ingebouwde diepteherkenning die gemaakt is voor de Xbox 360. In de Xbox wordt de Kinect gebruikt om spellen te spelen zonder controller maar met een zogenaamde natural user interface, via gestures.

Er zijn ook veel professionele oplossingen op de markt voor persoonsdetectie. Dit vindt veel toepassing bij beveiliging van gebouwen en bouwplaatsen. Er bestaan zowel all-in one oplossingen bestaande uit camera's, infrastructuur en software als ook softwarepakketten die met gangbare camera's, infrarood camera's en TOF camera's overweg kunnen. In dit document zullen we de verschillende technologieën op een rij zetten en de voor en nadelen benoemen.

Ethiek

De privacy implicaties bij gebruik van videocamera's zijn groot. Dit komt doordat er door een camera veel meer gegevens waargenomen en dus ook opgeslagen kunnen worden. Uit deze gegevens kunnen vervolgens de identiteit van personen, locatie, gesprekspartner, etc., etc. afgeleid worden. Ook is de vraag in hoeverre een camera toelaatbaar is in bijvoorbeeld een openbare toiletruimte.

Volgens de directeur van de Vigilat groep zijn camera's in principe toegestaan mits dit op een duidelijke manier kenbaar gemaakt wordt aan personen die het gebouw betreden. Dit heeft puur betrekking op het maken en opnemen van de beelden. Hetgeen wij willen gaan doen met de Kinect, het registreren van personen, gaat een stap verder. Omdat dit een relatief



onontgonnen gebied is is het lastig om daar op dit moment wat over te zeggen. Het is nu wel zo dat bijvoorbeeld de KLPD veel kritiek krijgt op hun kentekenscan systemen die voor langere tijd onthouden waar bepaalde auto's hebben gereden.

Al met al heeft dit systeem (video-opname en herkenning) de potentie om grote inbreuk te maken op de privacy. Om dit te voorkomen zullen we er voor moeten zorgen dat gegevens over personen niet te lang bewaard blijven en geanonimiseerd zijn.

Figuur 21: Voorbeeld camera waarschuwing

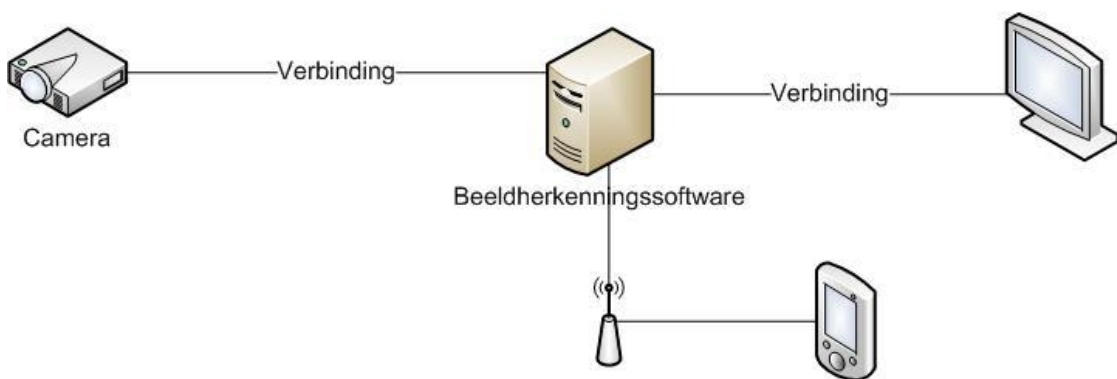
Betrouwbaarheid

Herkenningsoftware

De betrouwbaarheid van een beeldherkenningssysteem verschilt natuurlijk per implementatie. Fabrikanten zijn niet bepaald openhartig te noemen betreft deze gegevens. Veel verder dan het melden van een “goede” herkenning komen de meesten niet. De betrouwbaarheid van de herkenningsoftware zal dan ook grotendeels uit praktijktesten moeten blijken.

Infrastructuur

Een andere vraag is hoe de gegevens van de camera's naar een apparaat getransporteerd worden die ze analyseert en van daar verder.



Figuur 22: Verbindingen

Stroom

De decentrale onderdelen van het systeem zoals de camera's zullen voorzien moeten worden van een betrouwbare stroomvoorziening. Dit kan door brandwerende kabels te gebruiken of elke camera te voorzien van een accu.

Netwerk

Om de camera aan de herkenningsoftware te koppelen en vervolgens aan de gebruikers is een netwerk nodig. Om dit netwerk zo betrouwbaar mogelijk te maken kan er gebruik worden gemaakt van brandwerende kabels of een draadloze oplossing. In combinatie met een accu zal een draadloze camera waarschijnlijk de meest betrouwbare zijn, de camera is dan tijdens calamiteiten nergens van afhankelijk. Wat hierbij dan wel speelt is de afstand die een draadloze camera betrouwbaar kan overbruggen.

Implementatie

Een beeldherkenningssysteem bestaat uit twee componenten. Te weten de camera en de beeldherkenningssoftware.

Camera

Kinect

De Kinect is een camera van Microsoft die bedoeld is om in samenwerking met een Xbox360 gebruikers spelletjes te laten spelen m.b.v. hun armen, benen, etc, etc. De Kinect is in staat om met behulp van een infrarode laser en twee camera's niet alleen een gewoon camera beeld te leveren maar ook een waarop te zien is hoe ver weg iets is. Hierdoor is het met een Kinect een stuk makkelijker om mensen te herkennen. Er is inmiddels een ware Kinect Community ontstaan die de mogelijkheden van de Kinect onderzoeken en software hiervoor openbaar beschikbaar maken.



Figuur 23: Kinect

Gewone camera

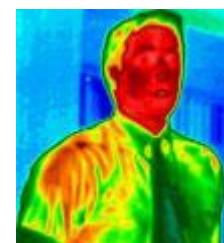
Een gewone camera heeft als voordeel dat er veel keuze in is. Zo kan een gewone camera een veel hogere resolutie hebben en het daardoor makkelijker maken om hier herkenning algoritmen op los te laten. Ook heeft deze camera als voordeel dat er niet perse van USB gebruik gemaakt hoeft te worden. Verder zijn deze camera's ook een stuk eenvoudiger te integreren in het interieur. Waar een Kinect vrij groot is zijn er voor normale camera's honderden commercieel verkrijgbare modellen die goed weg te werken zijn in een ruimte.



**Figuur 24:
Dome camera**

Thermische camera

Het nadeel van een gewone camera is dat deze het in het donker erg slecht doet. Er valt dan praktisch niets te zien. Een camera die dit nadeel niet heeft is een thermische camera. Deze camera's reageren op warmte (infrarood straling). Een mens is over het algemeen warmer dan zijn omgeving en zal dus eenvoudig te herkennen zijn. Deze camera biedt ook betere herkenningmogelijkheden wanneer een persoon bijvoorbeeld op de grond ligt.



**Figuur 25:
Thermisch**

Software

Er zijn verscheidene softwarepakketten voor beeldherkenning en het gebruik van de Kinect.

IOImage

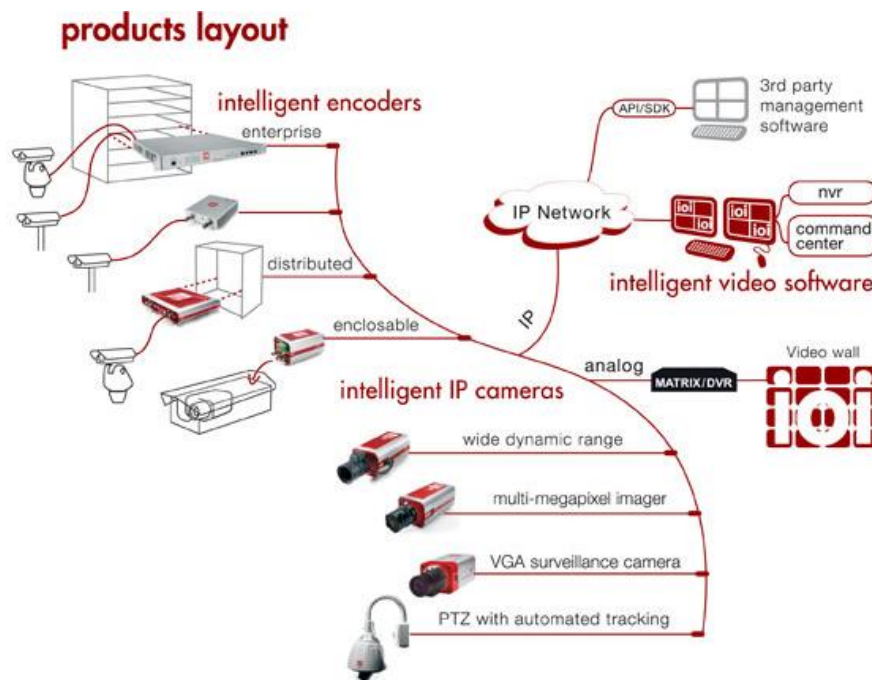
Dit softwarepakket wordt gebruikt door de Vigilat groep. Deze software is in staat om mensen en objecten te herkennen. Het wordt voornamelijk gebruikt in de beveiligingswereld maar de mogelijkheden die het heeft zijn ook binnen dit project toepasbaar. Een nadeel van dit pakket is dat het niet opensource is en de mogelijkheden om het naar wens aan te passen dus vrij beperkt zijn. IOImage biedt behalve centrale oplossingen ook stand alone camera's met ingebouwde beeldanalyse, hieronder een illustratie van de productrange van IOImage.

PrimeSense NITE

Deze software is gemaakt door medeontwikkelaars van de Kinect. Het is in staat mensen en hun interactie met de omgeving te detecteren. Het is opensource en kan dus naar wens aangepast worden.

Kinect SDK

In eerste instantie zijn door Microsoft alleen drivers voor de Xbox360 gemaakt. Momenteel zijn zij bezig om voor Windows een SDK te maken waarmee de Kinect eenvoudig aangesloten kan worden op een standaard computer. Tot deze gereleased wordt moet er gebruik gemaakt worden van opensource alternatieven die er momenteel zijn. Dit zijn er momenteel twee: OpenKinect en OpenNI waarbij de laatstgenoemde ontwikkeld is door het bedrijf wat ook de Kinect medeontwikkeld heeft.



Figuur 26: ProductRange van IOImage

Aanleg en Onderhoud

De aanleg en het onderhoud van een dergelijk systeem is erg afhankelijk van de implementatie. Er zit veel verschil tussen verschillende camera systemen en hun bereik. Ook hangt het veel af van de manier waarop het beeld verwerkt gaat worden hoe de infrastructuur er uit ziet. Er kan verschil gemaakt worden tussen centrale, semi-locale en lokale verwerking van de videodata.

Centrale Verwerking

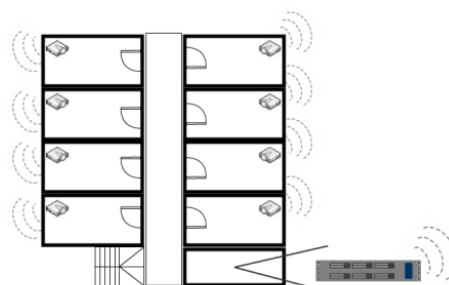
Bij centrale verwerking wordt alle videodata van de bronnen naar een centrale plek gestuurd waar de beelden worden geanalyseerd door een software systeem dat personen kan detecteren. Deze centrale plek kan op een toegankelijke plek binnen het gebouw gebeuren, bijvoorbeeld in een brandwerende ruimte bij de hoofdingang. Maar kan ook gebeuren op een externe locatie. Om de data op een centrale plek te verwerken zal de video op een of andere manier daar terecht komen. Hiervoor zal vanaf elke bron een kabel gelegd moeten worden naar de centrale verwerking. Een nadeel hiervan is dat er een single point of failure is, als de centrale verwerking uitvalt is er totaal geen verwerking meer.

Semi-Centrale verwerking

Een andere optie is om de data van meerdere bronnen in clusters te verwerken. Hierdoor is er minder bekabeling nodig van de bron tot de verwerking. Zo zou er verwerking per verdieping of per vleugel gedaan kunnen worden en de resultaten daarvan, een aanzienlijk kleinere datastroom, doorgestuurd naar een centrale plek. Bij deze vorm van werkerking is een draadloze implementatie ook mogelijk gezien er in veel gevallen een stuk kleinere afstand is tussen de bron en de verwerker. Dit zorgt voor een minder ingrijpende oplossing voor de aanleg. Ook zorgt dit voor een hogere beschikbaarheid aangezien er geen single point of failure is zoals bij centrale verwerking.

Lokale Verwerking

De meest decentrale oplossing die mogelijk is is lokale verwerking, hierbij kun je denken aan een camera met automatische persoonsdetectie ingebouwd in de camera of in een klein systeem/controller die aan de camera vast zit en data over het aantal personen bedraad of wireless verstuurd naar een centrale plek. Hieronder een klein voorbeeldje. De server stelt de verwerking voor van de camera's die in de verschillende ruimtes op die verdieping hangen.



Figuur 27: Voorbeeld lokale verwerking

Kosten

De kosten van het gebruik van camera technologie voor het herkennen van personen ligt gemiddeld een stuk hoger dan het gebruik van bijvoorbeeld deursensoren of bewegingsensoren. Dit komt voornamelijk door het feit dat de hardware voor een deursensor eens stuk goedkoper is dan een camera. Echter biedt een camera wel meer functionaliteit dan het meten van objecten op een plek en kunnen meerdere personen tegelijkertijd gedetecteerd worden.

Zoals bij de implementatie al te zien is zijn er veel verschillende oplossingen voor persoon detectie met camera's . Een Kinect camera kost bijvoorbeeld ongeveer 150 euro en heeft een bereik van enkele meters (aldus de originele specificaties icm de Xbox360). Voor het monitoren van een grote ruimte zullen dus meerdere Kinects nodig zijn en dit zorgt ervoor dat de kosten per ruimte relatief hoog zijn. Professionele camera's die geleverd worden in combinatie met bijvoorbeeld IOImage kosten tussen de 1000 en 3000 euro maar hebben wel een beter analyse bereik. Dit betekent dat er een veel grotere oppervlakte kan worden gemonitord. Hieronder staat een voorbeeld van een IOImage camera die de verwerking van het beeld al in de camera zelf doet en dus geen aparte verwerking nodig heeft voor het analyseren van de beelden. Dit scheelt in de licentiekosten en maakt het systeem ook een stuk schaalbaarder.



Figuur 28: IOImage Intelligent-Video IP Camera



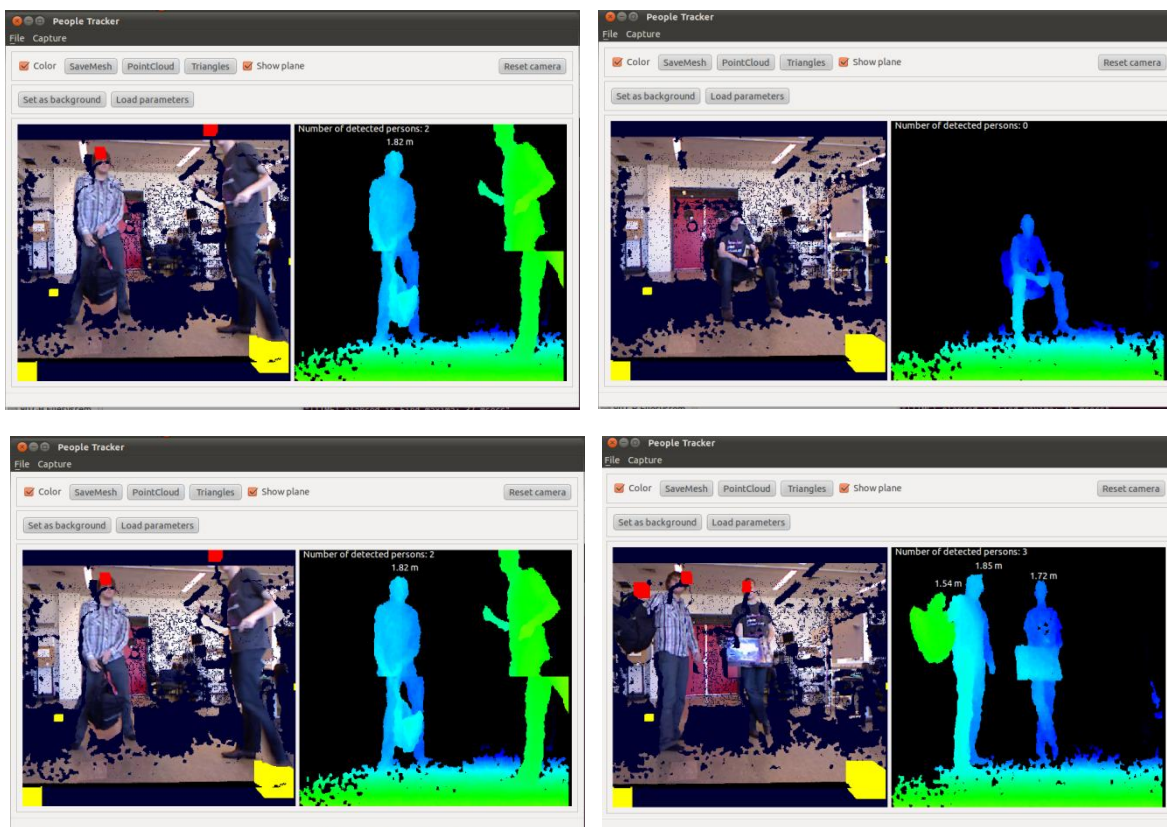
Figuur 29: NITE Standalone SOC Camera



Figuur 30: Kinect Camera

Appendix II: Testresultaten

Bij het testen van de Kinect is gebleken dat de betrouwbaarheid van de beeldherkenning erg laag is. Hieronder staan de volledige resultaten.



Figuur 31: Screenshots Kinect-software

De deursensoren bleken in de testen een betrouwbaarheid te hebben tussen de 92% en 100%.

| Wat? | Hoeveel personen in de ruimte | Gemeten aantal personen in de ruimte |
|-------------|-------------------------------|--------------------------------------|
| Beginaantal | 4 | 4 |
| 4x uit | 0 | 0 |
| 3x in | 3 | 3 |
| 1x uit | 2 | 2 |
| 2x in | 4 | 4 |
| 4x uit | 0 | 1 |
| 4x in | 4 | 5 |
| 1x uit | 3 | 4 |
| 2x uit | 1 | 2 |
| 3x in | 4 | 5 |

Accuraatheid in: 12/12 100%
 Accuraatheid uit: 11/12 92%

| Wat? | Hoeveel personen in de ruimte | Gemeten aantal personen in de ruimte |
|-------------|--------------------------------------|---|
| Beginaantal | 0 | 0 |
| 30x in | 30 | 29 |
| 30x out | 0 | -1 |

Accuraatheid in: 29/30 97%
 Accuraatheid uit: 30/30 100%

| Wat? | Hoeveel personen in de ruimte | Gemeten aantal personen in de ruimte |
|-------------|--------------------------------------|---|
| Beginaantal | 0 | 0 |
| 3x in | 3 | 0 |
| 1x in | 4 | 4 |
| 2x uit | 2 | 2 |
| 2x in | 4 | 4 |
| 4x uit | 0 | 0 |
| 6x in | 6 | 6 |
| 1x uit | 5 | 5 |
| 2x uit | 3 | 3 |
| 3x in | 6 | 6 |

Accuraatheid in: 15/15 100%
 Accuraatheid uit: 9/9 100%

Appendix III: Overzicht afgeronde user stories

In deze appendix staan alle afgeronde user stories. De verschillende stories waar in dit verslag naar gerefereerd wordt, staan hieronder opgesomd.

| Sprint | Titel | Story Points | Omschrijving | How to demo |
|--------|--|--------------|--|--|
| 1.0 | [RTES-10] Onderzoek naar mogelijkheden met warmte-/bewegingssensor | 10 | | |
| 1.0 | [RTES-46] Gezichts/mens herkenning m.b.v. OpenCV | 10 | | |
| 1.0 | [RTES-11] Onderzoek naar mogelijkheden met deursensor | 10 | | |
| 1.0 | [RTES-7] Onderzoek naar de mogelijkheden van de kinect | 20 | | |
| 1.0 | [RTES-53] Indeling Sprint 1 | | | |
| 1.0 | [RTES-54] Opstarten Project: opzetten project management systeem | | | |
| 1.0 | [RTES-55] Opstarten Project: Inleidend gesprek met opdrachtgever | | | |
| 1.0 | [RTES-56] Opstarten Project: Projectgrenzen definiëren | | | |
| 1.0 | [RTES-57] Opstarten Project: Plan van Aanpak opstellen | | | |
| 1.0 | [RTES-64] BHV Gesprek | 3 | In gesprek met hoofd van BHV Saxion Enschede-Deventer. Voorbereiding van vragen over hoe hun te werk gaan en een door ons te ontwikkelen systeem hier bij een rol kan spelen | Notulen gesprek. |
| 1.0 | [RTES-65] Bewegend beeld uit Kinect | 5 | Dit omhelst het aansluiten van de kinect | Een applicatie die het bewegende beeld van |

| | | | | |
|------------|--|----|---|---|
| | Krijgen. | | aan een computer en weergave van beeld | de Kinect weergeeft |
| 1.0 | [RTES-71] Bewegingssensor keuze onderzoeken | 2 | Een onderzoek naar de keuze voor de beste bewegingssensor hardware te gebruiken voor ons systeem | Een keuze voor de best te gebruiken bewegingssensor voor grootschalige toepassing en eerder genoemde criteria |
| 1.0 | [RTES-14] Adviesdocument | 15 | Aan de hand van verschillende onderzoeken een conclusie schrijven met een advies voor de klant | Verslag |
| 1.0 | [RTES-117] Verbeteren onderzoeken | 3 | literatuur verwerken in de inhoud zelf. Document van Jelle omzetten naar non-literatuuronderzoek. | |
| 1.0 | [RTES-67] Werking dieptesensor Kinect onderzoeken | 10 | Kijken in hoe ver de dieptesensor meerwaarde heeft voor de herkenning van objecten. | Een verslag na de toepasbaarheid van de dieptesensor Kinect door praktisch onderzoek |
| 1.0 | [RTES-66] Beelherkenning van Kinect | 20 | Het herkennen van een door ons gekozen object in een 3x3m ruimte uit het beeld/diepte van de Kinect. Gebruikmakend van bijvoorbeeld de OpenKinect Driver. | Demonstratie van de Kinect die een object kan herkennen |
| Sprint 2.1 | [RTES-126] Prototype Webserver Installeren | 8 | Het installeren van Apache Tomcat | Een omgeving waarop Apache Tomcat geïnstalleerd staat |
| Sprint 2.1 | [RTES-132] Basisweergave beweging in ruimte (webinterface) maken | 8 | Weergave sensordata in tabelvorm | Demonstratie weergave |
| Sprint 2.1 | [RTES-127] Webservices maken | 8 | Het aanmaken van een simpel protocol zodat de gateways of nodes via HTTP hun waardes kunnen inschieten. | Demo gaat via een van de weergaves. |
| Sprint 2.1 | [RTES-70] [C] Deursensor keuze onderzoeken | 8 | Een klein onderzoekje naar de keuze voor de beste deursensor hardware te gebruiken voor ons systeem | Een keuze voor het best te gebruiken deursensor voor grootschalige toepassing en eerder beschreven criteria |

| | | | | |
|------------|--|----|---|--|
| Sprint 2.1 | [RTES-128] Basisweergave aantal personen in ruimte (webinterface) maken. | 8 | Weergave sensordata in tabelvorm | Demonstratie weergave |
| Sprint 2.1 | [RTES-134] Databasestructuur ontwerpen | 15 | Geschikte databasestructuur maken voor het opslaan van sensoren, gateways en sensorwaardes | Uitdraai van structuur |
| Sprint 2.1 | [RTES-131] [C] Met bewegingssensor beweging detecteren en uitlezen | 15 | | Laten zien dat het uit te lezen is op de computer |
| Sprint 2.1 | [RTES-135] [C] Gevonden nodes praktisch testen | 60 | Praktische test met nodes, evt. multihop in een ruimte. | Werkend mesh-netwerk met aan twee kanten een invoer/uitvoer mogelijkheid met behulp van bijv. een computer. |
| Sprint 2.1 | [RTES-125] [C] Geschikte nodes zoeken | 50 | Uitzoeken naar geschikte wirelesscommunicatiemethoden en chip (multihop / mesh / Zigbee?) | Beknopt verslag van bevindingen |
| Sprint 2.1 | [RTES-130] [C] Gateway hardwarekeuze bepalen | 8 | Kijken naar beste oplossing voor gateway (systeem per gang/afdeling). Ook OS-keuze | Beknopt verslag van beste keuze hardware/OS |
| Sprint 2.2 | [RTES-142] Gateway communicatie met webservice | 10 | Het doorsturen van (dummie) sensordata naar de server. Dit gebeurt door middel van HTTP | Het door kunnen sturen van (dummie)sensordata van de gateways naar de server. Dit wordt verwerkt in de database van de server of een servlet |
| Sprint 2.2 | [RTES-144] Sensordata doorsturen over XBee | 20 | Gegevens van een sensor inlezen en doorsturen over het mesh netwerk | Sensordata is te zien op een PC. |
| Sprint 2.2 | [RTES-141] Hardware aanschaffen | 4 | Het bestellen van de verschillende benodigde hardware. Dit omhelst onderandere de aanschaf van de gekozen gateway, ev XBees en deursensoren | n.v.t. |
| Sprint | [RTES-152] Gateway installeren | 10 | Dit omhelst het uitpakken en installeren | n.v.t. |

| | | | | |
|------------|---|----|---|--|
| 2.2 | | | van de benodigde software op de gateway. | |
| Sprint 2.2 | [RTES-143] Gatewaycommunicatie met XBee | 35 | Data uit kunnen lezen van het XBee netwerk op de gekozen gateway. Dit omhelst het aansluiten van de XBee adapter aan de gateway (via usb) en het schrijven van een stukje code om de communiceren met de aangesloten XBee | Het uit kunnen lezen van de data van het XBee meshnetwork op de gateway |
| Sprint 2.2 | [RTES-145] Loginmodule website | 15 | Loginsysteem voor website | Om op de website te kunnen komen moet eerst worden ingelogd. |
| Sprint 2.2 | [RTES-138] Beslissingsalgoritme ontwerpen | 10 | Brainstormen over beslissingsalgoritme/logica/prioriteit | Notulen van brainstormsessie/uitwerking in document |
| Sprint 2.2 | [RTES-68] [C] Met deursensor beweging detecteren en uitlezen | 10 | Het plaatsen van een deursensor en uitlezen van de sensor vanaf bijvoorbeeld een computer | Counter aantal voorbijgaande objecten van deursensor demonstreren |
| Sprint 2.2 | [RTES-69] [C] Met deursensors richting bewegend object bepalen/uitlezen | 3 | Het plaatsen van 2 deursensoren en het uitlezen van de sensoren op een computer om de richting te bepalen van een voorbijgaand object | Een counter van het aantal ingaande en uitgaande personen bij een deur |
| Sprint 2.2 | [RTES-146] Accountbeheer website | 10 | Module om accounts te beheren | Een website waar accounts beheerd kunnen worden. |
| Sprint 2.2 | [RTES-129] Kaartweergave combinatie alle sensoren webinterface | 25 | Weergave van sensordata op een kaart van een afdeling/vleugel. Nog geen mogelijkheid zelf kaarten in te voeren | Demonstratie weergave |
| Sprint 2.2 | [RTES-139] Beslissingsalgoritme implementatie | 25 | Implementeren van story 'Beslissingsalgoritme ontwerp' | Lijstweergave van ruimtes waar volgens beslissingsalgoritme nog mensen zijn (incl. prioriteit) |
| Sprint 3.1 | [RTES-166] Compact productieboardje sensornodes | 15 | Het ontwerp van een compact PCB voor de XBee en sensoren | Schema voor PCB en PCB Design in PDF formaat. |

| | | | | |
|------------|---|----|---|--|
| Sprint 3.1 | [RTES-153] Aanleg testomgeving | 20 | Installeren meshnetwerk, gateway en uitrusten ruimte met sensoren. | Aangelegde testomgeving |
| Sprint 3.1 | [RTES-167] Beheren van sensoren/nieuwe sensoren toevoegen | 35 | Toevoegen/verwijderen/wijzigen van ruimtes. Mogelijkheid tot koppelen van nieuw aangemelde sensoren bij een ruimte. Mogelijkheid tot instellen ruimte van een bestaande sensor | Mogelijkheid tot aanmelden nieuwe sensoren / beheren ruimtes |
| Sprint 3.1 | [RTES-170] "Werkt alles nog?" - heartbeat | 20 | Nieuwe story bedacht ahv sprintopstart. Checken of er gateways e/o nodes uitgevallen zijn. | |
| Sprint 3.1 | [RTES-158] Implementatie bewegingssensor in mesh-netwerk | 10 | Gezien de gekozen bewegingssensor anders functioneerd dan de deursensor waar oorspronkelijk mee getest is moet de code voor het uitlezen van de bewegingssensor worden herschreven. | |
| Sprint 3.1 | [RTES-150] Beveiliging gateway/server ontwerpen | 25 | Onderzoeken wat de mogelijkheden zijn qua beveiliging webserver/gateway (HTTPS, encryptie, API Keys) | Beveiligde verbinding |
| Sprint 3.1 | [RTES-168] Meshnetwerk meerdere gateways | 20 | | Het testen van het meshnetwerk met meerdere gateways. Testen of pakketten allemaal aankomen bij gebruik van meer dan 1 gateways in het meshnetwerk |
| Sprint 3.1 | [RTES-164] Persoontelling testomgeving | 15 | Het meten van het aantal personen die in en uit de testomgeving gaan | Rapportage van de getelde personen |
| Sprint 3.1 | [RTES-165] Verifiëren persoontelling testomgeving | 15 | Het controlleren van de metingen die gedaan worden door de sensoren in de testomgeving | Document met vergelijking tussen gemeten aantal personen en werkelijke aantal personen (met de hand geteld). |