

I2Vote 2.0

Jonatan Ties Prins

Radiologie, UMCG
Hanzehogeschool Groningen, Informatica

Groningen, mei 2014

Studentenbureau UMCG



rijksuniversiteit
groningen

Universitair Medisch Centrum Groningen

I2Vote 2.0

Groningen, mei 2014

Auteur
Studentnummer

Jonatan Ties Prins
222578

Afstudeerscriptie in het kader van

Software Engineering
SICT Informatica
Hanzehogeschool Groningen

Oprachtgever

MSc. PhD. CHPHIT. P.M.A. van Ooijen
Radiologie, UMCG

Begeleider onderwijsinstelling

drs. G.P.C. Kerkhoff
Informatica
Hanzehogeschool Groningen

Begeleider UMCG

MSc. PhD. CHPHIT. P.M.A. van Ooijen
Radiologie, UMCG

© 2012 Studentenbureau UMCG Publicaties Groningen, Nederland.

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen, of enige andere manier, zonder voorafgaande toestemming van de uitgever.

Voor zover het maken van kopieën uit deze uitgave is toegestaan op grond van artikel 16B Auteurswet 1912 j° het Besluit van 20 juni 1974, St.b. 351, zoals gewijzigd in Besluit van 23 augustus 1985, St.b. 471 en artikel 17 Auteurswet 1912, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht. Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich tot de uitgever te wenden.

Trefw I2Vote, stelsysteem, afbeelding gebaseerd, mobile development, hci, aanwijsstrategieën, ontwikkelmethoden, Javascript, AngularJS, NodeJS, ExpressJS, SocketIO, realtime, web app, web applicatie, databases, smartphones, PDA, interactief, lesgeven, BYOD, Bring your own device.

VOORWOORD

Na een aantal jaren soms moeilijke, maar veelal leuke momenten en een uitstapje naar de lerarenopleiding basisonderwijs lijkt het einde van de opleiding nu dan toch in zicht te komen. Voor de verdiepingsstage ben ik in mobile development gerold, dit heb ik voor het afstuderen doorgezet en gecombineerd met verschillende onderzoeksaspecten bij het UMCG. Hieruit is deze scriptie ontstaan. Tijdens het schrijven ervan heb ik veel nieuwe dingen geleerd (zowel op medisch als ICT gebied) en leuke mensen ontmoet. Voor deze kans en de ondersteuning bij het schrijven van de stukken voor deze scriptie bedank ik P.M.A. van Ooijen en G.P.C. Kerkhoff.

Verder wens ik u veel plezier bij het lezen van deze scriptie.

Jonatan Ties Prins

Groningen, 1 mei 2014

INHOUDSOPGAVE

SAMENVATTING	1
1 INLEIDING.....	3
1.1 DE ORGANISATIE EN BETROKKENEN	3
1.2 KORTE INTRODUCTIE I2VOTE.....	4
1.3 DE PROBLEEM- EN DOELSTELLING	4
1.4 DE HUIDIGE SITUATIE.....	4
1.5 DE GEWENSTE SITUATIE.....	5
1.6 SCOPE, RANDVOORWAARDEN EN RISICO'S	5
2 RELEVANTIE EN VERANKERING	7
2.1 RELEVANTIE.....	7
2.2 THEORETISCH KADER	8
3 METHODIEK.....	11
3.1 AANWIJSSTRATEGIEËN	11
3.2 ONTWIKKELPLATFORMEN	11
3.3 DE ONTWIKKEL PERIODE	11
4 AANWIJSSTRATEGIEËN	13
4.1 PROBLEEMSTELLING.....	13
4.2 RELEVANTIE.....	14
4.3 METHODE.....	14
4.4 LITERATUURONDERZOEK	16
4.5 MOGELIJKE STRATEGIEËN.....	16
4.6 RESULTATEN	17
4.7 CONCLUSIE EN ADVIES.....	20
5 ONTWIKKELPLATFORMEN.....	23
5.1 NATIVE, WEB OF EEN HYBRID OMGEVING	23
5.2 AANBEVOLEN FRAMEWORKS EN BIBLIOTHEKEN	25
5.3 DATABASE.....	26
6 ARCHITECTUUR, FRAMEWORKS EN PLATFORMEN	29
6.1 INFRASTRUCTUUR EN INFORMATIE PADEN	29
6.2 SERVICE MET NODEJS.....	31

6.3 DE STUDENTENAPPLICATIE MET ANGULARJS.....	34
7 TECHNISCHE REALISATIE.....	37
7.1 BACK-END.....	37
7.2 MOBILE FRONT-END.....	38
8 CONCLUSIE & AANBEVELINGEN.....	41
8.1 CONCLUSIE.....	41
8.2 AANBEVELINGEN EN IDEEËN.....	41
8.3 HOE IS HET RESULTAAT ONTVANGEN.....	41
9 BIBLIOGRAFIE.....	43
BIJLAGE I LITERATUURONDERZOEK BIJ AANWIJSSTRATEGIEËN.....	45
BIJLAGE II LITERATUURONDERZOEK BIJ ONTWIKKELPLATFORMEN.....	47
BIJLAGE III USE CASES (STUDENT).....	49
BIJLAGE IV USE CASES (DOCENT).....	51
BIJLAGE V AUTHENTICATIE SUGGESTIES.....	53

SAMENVATTING

Dit onderzoek uitgevoerd voor het Radiologie R&D van het UMCG stelt de vraag of en hoe I2Vote toepasbaar is in een BYOD omgeving. I2Vote is een image-based voting system. Om dit te bepalen zijn een aantal deelonderzoeken uitgevoerd. Omdat er van een PDA naar een smartphone gemigreerd wordt, waarbij de aanwijspen uit het beeld verdwijnt, wordt er gekeken naar de verschillende aanwijsstrategieën voor op een touchscreen. Daarnaast worden verschillende ontwikkelplatformen vergeleken. Bij de keuze voor het ontwikkelplatform worden een aantal geschikte frameworks die bij dat platform passen geadviseerd. Aan de hand van de onderzoeken is een implementatie gegeven van een image-based vraag om aan te tonen dat de voorgestelde oplossingen werken, hierbij wordt ook gekeken naar de REST service.

Uit het onderzoek naar aanwijsstrategieën is gebleken dat de Offset strategie het meest aan alle gestelde eisen voldoet. Daarnaast zegt het onderzoek naar ontwikkelplatformen dat een web applicatie een geschikte oplossing is voor I2Vote met de huidige middelen.

De platformen en frameworks die daarbij worden geadviseerd zijn AngularJS en KineticJS voor de cliënten (docenten- en studentenapplicatie). Voor de service wordt een combinatie van NodeJS, ExpressJS en SequelizeJS geadviseerd met daarachter een MySQL database.

Met deze resultaten is een implementatie gegeven van een essentieel onderdeel van I2Vote, het beantwoorden van een image-based vraag is hierin verwerkt, evenals het beantwoorden van multiple-choice vragen. Er is een notificatie systeem opgezet en het is mogelijk om in de browser realtime van te voren geformuleerde vragen te ontvangen. Daarnaast zijn er voor het ontwikkelen van de REST service grote stappen gemaakt. De beschreven architectuur legt een aantal technische beslissingen uit met betrekking tot onder andere CORS, REST URI's en notificaties.

Uit de resultaten van het onderzoek en de gegeven implementatie blijkt dat I2Vote geschikt is voor een BYOD omgeving. Daarbij is een bruikbare applicatie structuur voorgesteld voor zowel de studentenapplicatie als de docentenapplicatie.

1 INLEIDING

Dit rapport beschrijft het project: de revisie van I2Vote. De revisie is uitgevoerd in opdracht van de Radiologie afdeling van Universitair Medisch Centrum Groningen (UMCG). Dit hoofdstuk beschrijft de organisatie waarin de opdracht wordt uitgevoerd. Daarna volgt een beschrijving van de context en de probleem- en doelstelling. Daarbij zijn de scope, randvoorwaarden en risico's beschreven. Gevolgd door de voorgestelde strategie en methodiek.

1.1 DE ORGANISATIE EN BETROKKENEN

1.1.1 HET UMCG

De opdracht wordt uitgevoerd voor het UMCG. Het ziekenhuis is één van de grootste van Nederland en dé grootste werkgever van Noord-Nederland. Er werken ruim 10.000 medewerkers in de patiëntenzorg en vooraanstaand wetenschappelijk onderzoek, waarbij de focus ligt op 'gezond en actief ouder worden'. Op gebied van wetenschappelijk onderzoek werkt het UMCG nauw samen met de Rijksuniversiteit Groningen (RUG). Er worden studenten opgeleid tot arts, artsen tot medisch specialist, tandarts of bewegingswetenschapper. Patiënten komen naar het UMCG voor basiszorg, maar ook voor specialistische diagnostiek, onderzoek of behandeling. Het UMCG houdt zich bezig met de kerntaken: zorg, onderwijs en onderzoek.

1.1.2 RADIOLOGIE

1.1.2.1 ALGEMEEN

De Radiologie valt onder Sector E van het UMCG. Radiologie houdt zich bezig met het maken en analyseren van beelden van patiënten en het doen van minimaal invasieve behandelingen. Hierbij worden technologieën gebruikt zoals röntgenstraling, CT-scans, MRI en echografie. Radiologie heeft het grootste deel van haar faciliteiten in het hoofdgebouw van het UMCG bij de Centrale Afdeling Radiologie (CAR).

1.1.2.2 RESEARCH & DEVELOPMENT

Naast het CAR is er een Research & Development afdeling, de afdeling is gevestigd in het Triade gebouw van het

UMCG. Ze doet onderzoek naar verschillende nieuwe technologieën. Zo is er een onderzoek gaande naar het verbeteren van de gebruikersinterface van de software waar de radiologen de foto's mee beoordelen, wordt er onderzoek gedaan naar cross-enterprise document sharing¹ en wordt er gewerkt aan een image-based voting system².

1.1.3 BETROKKENEN

Er zijn een aantal mensen betrokken bij het afstudeerproject zowel van de Hanzehogeschool en het UMCG. In tabel 1 is een overzicht te zien van de betrokkenen.

<i>Naam</i>	<i>Rol</i>
Dr. Ir. P.M.A. van Ooijen Ing. A. Broekema	Begeleider en opdrachtgever namens het UMCG. Betrokken bij het project, ontwikkelaar geweest van de huidige versie en af en toe beschikbaar voor het beantwoorden van (technische) vragen.
Dhr. G.P.C. Kerkhoff	Begeleider en eerste examinator van de Hanzehogeschool. Dhr. Kerkhoff is verantwoordelijk voor de begeleiding vanuit de Hanzehogeschool en het beoordelen van het afstuderen.
Dhr. J.W. Knobbe	Tweede examinator aangewezen door de Hanzehogeschool. Dhr. Knobbe is medeverantwoordelijk voor het beoordelen van het afstuderen.

Tabel 1 Betrokkenen

¹ http://wiki.ihe.net/index.php?title=Cross-enterprise_Document_Sharing_for_Imaging

² Een op afbeeldingen gebaseerd stemsysteem.

1.2 KORTE INTRODUCTIE I2VOTE

Tijdens de colleges tot bijvoorbeeld basisarts wordt er af en toe gebruik gemaakt van een ARS zoals Socrative³. De docent stelt een vraag en de studenten kunnen via de smartphone, laptop of tablet daar bijvoorbeeld A of B op antwoorden. Omdat tijdens radiologie colleges veel gebruik wordt gemaakt van afbeeldingen is de vraag ontstaan naar een image-based ARS. I2Vote is hiervoor in het leven geroepen door P.M.A. van Ooijen. In de publicatie (van Ooijen, Broekema, & Oudkerk, 2011) worden het ontwerp en de resultaten van de proeven met de eerste iteratie van het systeem beschreven. I2Vote is een normaal ARS met de gebruikelijke multiple-choice vragen. Waar I2Vote anders is dan bijvoorbeeld Socrative is haar mogelijkheid tot het stellen van image-based vragen. Waarbij de docent een afbeelding presenteert aan de studenten via de PDA zodat zij daar een lokatie op kunnen aanwijzen als antwoord.

1.3 DE PROBLEEM- EN DOELSTELLING

Het idee van I2Vote is goed en er is een goede applicatie neergezet, maar momenteel liggen de PDA's in de kast en staat de server stil. Uit het onderzoek van P.M.A. van Ooijen (van Ooijen et al., 2011) blijkt dat de behoefte aan een dergelijk image-based ARS er is. Maar zoals alle software moet ook I2Vote met de tijd mee om relevant te blijven. De PDA's van HP zijn niet actueel gebleven en hebben plaatsgemaakt voor tablets en smartphones. De verwachting is dat veel studenten tegenwoordig in het bezit zijn van een smartphone, tablet of laptop. Deze verwachting wordt ondersteund door een onderzoek uit 2012 op de Universiteit van Amsterdam (drs. Nynke Bos en drs. Nunke Kruidenink, 2012). Hieruit blijkt dat ruim 70% van de geënquêteerde een tablet of een laptop bezit en ruim 50% een smartphone. Dit was in 2012, twee jaar later is het aantal smartphone bezitters naar verwachting verder gegroeid.

De vraag is ontstaan om I2Vote te vernieuwen zodat het weer in gebruik genomen kan worden. Dit afstudeerproject richt zich op het toepassen van het ARS I2Vote binnen een BYOD omgeving. Is het technologisch haalbaar? Zo ja, hoe

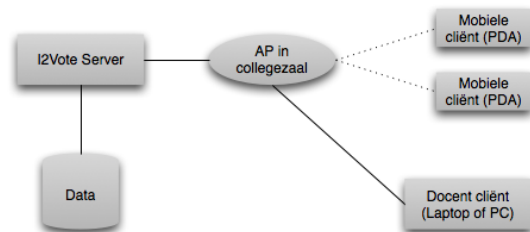
³ Een ARS als web applicatie <http://www.socrative.com>

kan dit dan het beste aangepakt worden zodat het grootste aantal platformen wordt bereikt met behoud van de originele functionaliteit? Zo nee, wat zijn dan de alternatieven? Het doel is om dit te beantwoorden door een basis te implementeren en advies te geven over hoe het verder uitgewerkt kan gaan worden.

1.4 DE HUIDIGE SITUATIE

I2Vote is momenteel geschikt voor de PDA en een Windows PC met daarop de docenten applicatie. In de docenten applicatie kan de docent zijn colleges opstellen en afspelen. De student heeft een PDA nodig om deel te nemen aan een college.

In figuur 1 is deze situatie te zien, hieruit is op te maken dat het systeem gekoppeld is aan een lokaal netwerk via een Access-Point⁴ (AP). Het mobiele apparaat (PDA⁵) is het apparaat waar de student de vragen op ontvangt en mee kan beantwoorden. Om de stof buiten de collegezalen te oefenen worden de vragen tijdens de colleges door het mobiele apparaat volledig gedownload. De vragen kunnen dan zonder het AP geoefend worden, de antwoorden worden opgeslagen op het mobiele apparaat en in de database gezet op het moment dat het apparaat weer op het netwerk van I2Vote zit. De I2Vote server is verantwoordelijk voor het bewaren en doorgeven van deze informatie.



Figuur 1 Huidige situatie I2Vote

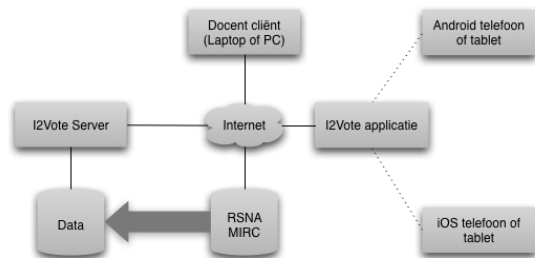
⁴ http://nl.wikipedia.org/wiki/Wi-Fi_Access_Point

⁵ PDA (PDA's)
<http://h20424.www2.hp.com/product/handhelds/nz/en/hp-ipaq-handhelds.asp>

1.5 DE GEWENSTE SITUATIE

Om een indicatie te geven van hoe het systeem eruit moet komen te zien is in figuur 2 de gewenste situatie geschetst. Hierin is te zien dat het systeem nu via internet werkt en dat de PDA's zijn vervangen door de nieuwe mobiele apparaten die via een nieuwe I2Vote applicatie communiceren met de I2Vote service. Voor het opstellen van de vragen is de docenten applicatie vernieuwd en uitgebreid zodat deze eenvoudig te gebruiken is.

In het huidige systeem moeten colleges of vragenlijsten voor een college handmatig samengesteld worden. Dit betekent zelf afbeeldingen zoeken, uploaden en daar vragen bij bedenken. Een mogelijke oplossing voor dit probleem is een koppeling met het MIRC systeem. Dit zou het aanmaken van vragenlijsten moeten vereenvoudigen door het mogelijk te maken om casussen te selecteren uit de MIRC database.



Figuur 2 Gewenste situatie I2Vote

In de gewenste situatie wordt er afgestapt van de PDA. Dit brengt niet alleen een verandering qua platform (Android of iOS). De PDA wordt bediend met een aanwijspen, met deze pen is het mogelijk om op de millimeter nauwkeurig een lokatie aan te wijzen. De moderne smartphones worden niet geleverd met een aanwijspen. Desondanks benadert de applicatie de nauwkeurigheid van de PDA met aanwijspen in de gewenste situatie.

1.6 SCOPE, RANDVOORWAARDEN EN RISICO'S

1.6.1 DE SCOPE

1.6.1.1 BINNEN DE SCOPE

- Het schrijven van een projectinitiatiedocument.
- Het schrijven van een eindschiptie.
- Onderzoek naar het meest geschikte ontwikkelplatform voor de mobiele studenten applicatie.
- Onderzoek naar de meest nauwkeurige en bruikbare aanwijsstrategie voor de studenten applicatie.
- Het implementeren van een prototype I2Vote studentenapplicatie voor mobile devices (met name de applicatie structuur en image-based vragen zijn hierbij belangrijk).
- Het implementeren van de I2Vote docentenapplicatie met een MIRC koppeling indien de tijd dit toelaat.
- De I2Vote service moderniseren en geschikt maken voor de nieuwe applicaties.

1.6.1.2 BUITEN DE SCOPE

- De beveiliging van het systeem. De voornaamste reden hiervoor is de complexiteit van het onderwerp beveiliging van BYOD systemen. Hier zou een volledige afstudeeropdracht aan besteed kunnen worden.
- Het systeem is niet bedoeld voor het afnemen van officiële examens.
- De I2Case applicatie(s).

1.6.2 RANDVOORWAARDEN

De volgende randvoorwaarden zijn van toepassing op het project:

- Voor de uitwerking van het project zijn twintig weken of vijf maanden beschikbaar.
- Er is een werkplek beschikbaar binnen het UMCG.
- Er is begeleiding vanuit het UMCG en vanuit de Hanzehogeschool.

1.6.3 Risico's

<i>Risico</i>	<i>Kans</i>	<i>Impact</i>	<i>Maatregel</i>
Uitlopen van de planning.	Groot.	Gemiddeld.	Duidelijke doelen met een realistische planning.
De benodigdheden zijn niet aanwezig.	Klein.	Groot.	In een vroeg stadium aangeven wat er nodig is. Zo is er meer ruimte om eventuele tegenslagen op te lossen.
Dataverlies door hardware falen.	Klein.	Groot.	Gebruik maken van versiebeheer zoals Git en Cloud-omgevingen.
Er is geen geschikt ontwikkelplatform.	Klein.	Groot.	Het onderzoek wordt in een vroeg stadium uitgevoerd.
Er is geen geschikte aanwijstrategie.	Klein.	Groot.	Het onderzoek wordt in een vroeg stadium uitgevoerd.
Ziekte (van Bechterew).	Midden.	Midden.	Goede balans vinden tussen rust, werk en bewegen (sport) om het risico te verkleinen.

Tabel 2 Risico's

2 RELEVANTIE EN VERANKERING

Dit hoofdstuk gaat in op de relevantie van het project en zal het theoretisch kader beschrijven.

2.1 RELEVANTIE

Waarom is de uitvoering van de revisie relevant? De volgende paragrafen beschrijven de relevantie voor de verschillende afdelingen en vakgebieden.

2.1.1 UNIVERSITAIR MEDISCH CENTRUM GRONINGEN, RADIOLOGIE

Tijdens proeven met het prototype (proof of concept op de PDA) van I2Vote is gebleken dat studenten er baat bij hebben (van Ooijen et al., 2011). De PDA's gebruikt tijdens deze proeven zijn inmiddels achterhaalde technologie. Een mogelijke oplossing is om de software geschikt te maken voor apparaten die wel worden gebruikt middels een revisie.

In het rapport van P.M.A. van Ooijen wordt aangetoond dat studenten baat hebben bij het systeem I2Vote, de interactie tijdens colleges maakt ze interessanter en verhoogt daardoor de participatie. De enquête gehouden tijdens de proeven toont aan dat de deelnemers vinden dat de stof beter blijft hangen door het gebruik van het systeem. Daarbij vindt meer dan de helft van de deelnemers dat er tijdens de proeven meer gebruik gemaakt had mogen worden van het systeem. Naast de duidelijke voordelen die de deelnemers ervaren, biedt I2Vote de docenten meer inzicht in de prestaties en het kennisniveau van zijn studenten. Dit komt voornamelijk doordat klassikale vragen op deze manier door alle studenten individueel beantwoord worden. De uitkomst van deze afstudeeropdracht, de revisie, zorgt ervoor dat het I2Vote weer onderdeel kan uitmaken van de radiologie colleges.

Bij de originele opzet van I2Vote is er gekozen voor de PDA als medium. Er zijn een aantal PDA's aangeschaft en ook een periode ingezet. Uit enquêtes bleek dat er toentertijd maar weinig studenten in het bezit waren van een PDA of

enige andere vorm van een smartphone. De behoefte om bijvoorbeeld thuis of in de bus te oefenen is er volgens de zelfde enquête wel. Het meegeven van PDA's van de onderwijsinstelling is geen optie, PDA's kunnen makkelijk kwijtraken of kapot gaan.

Tegenwoordig is de smartphone aanzienlijk gestegen in populariteit (drs. Nynke Bos en drs. Nunke Kruidenink, 2012). Een groter aantal studenten is in het bezit van een eigen smartphone. Deze trend zou de behoefte om thuis of in de bus te kunnen oefenen mogelijk kunnen maken door de afhankelijkheid van de PDA weg te nemen.

Deze stijgende populariteit van de smartphone maakt het mogelijk om I2Vote via deze apparaten tijdens colleges in te zetten. De opleidingsinstantie hoeft niet meer in het bezit te zijn van eigen apparaten. Dit scheelt kosten en tijdens de colleges tijd, de PDA's hoeven immers niet meer uitgedeeld of klaargelegd te worden.

Door de afhankelijkheid van de PDA weg te halen gaan de kosten voor het gehele systeem omlaag en zou het inzetten binnen basisscholen en middelbare-scholen laagdrempeliger worden.

Daarnaast vereenvoudigt een koppeling met het MIRC systeem het samenstellen van casussen en colleges. Dit komt met name doordat de docent niet meer verplicht wordt om zelf afbeeldingen te uploaden en achtergrond informatie te bedenken bij zijn vragen, deze kunnen uit de MIRC database geladen worden.

2.1.2 MOBILE DEVELOPMENT

Voor het vakgebied Mobile Development zou het onderzoek naar de verschillende aanwijsstrategieën die gebruikt kunnen worden om een lokatie op een afbeelding zo exact mogelijk aan te geven interessant kunnen zijn. Wanneer aanwijsprecisie van belang is zou de verzameling aan informatie die het onderzoek oplevert wellicht de keuze van de ontwikkelaars voor een ontwikkelplatform kunnen vereenvoudigen.

2.1.3 AUDIENCE RESPONSE SYSTEMS MET BRING YOUR OWN DEVICE

De pilot van I2Vote breidt het ARS principe uit met de mogelijkheid tot het stellen van image-based vragen. Deze vragen zijn niet gesloten, een antwoord kan elke willekeurige plek op een plaatje zijn. De pilot heeft aangetoond dat het werkt en dat er behoefte is aan een dergelijk systeem zoals I2Vote. Nu migreert I2Vote van de PDA (hardware speciaal aangeschaft voor dit systeem) naar de smartphone van de student. Doordat de software op het apparaat van de student zelf wordt uitgevoerd kan I2Vote geclassificeerd worden als een Bring Your Own Device (BYOD) omgeving. Met het resultaat van dit afstudeerproject zou een antwoord gegeven kunnen worden op de vraag of BYOD een geschikte oplossing is binnen een ARS omgeving.

2.2 THEORETISCH KADER

Er zijn een aantal theoretische kaders waar dit afstudeerproject binnen valt. Dit hoofdstuk beschrijft deze kaders en hoe de revisie van I2Vote daarin past.

2.2.1 BRING YOUR OWN DEVICE

Bring Your Own Device (BYOD) betekent je eigen apparaat meenemen naar en gebruiken op je werk- of onderwijsplek. Met het doel om deze apparaten daar te gebruiken om de informatie van de instelling te benaderen. Voor de werknemer betekent dit dat er bijvoorbeeld werk mail op de mobiel ontvangen kan worden of dat er presentaties met de de eigen iPad gegeven kunnen worden. Voor een onderwijsinstelling betekent dit dat studenten bijvoorbeeld examens kunnen doen op hun eigen laptop, telefoon of tablet.

Eén van de grote voordelen is dat een eigen apparaat vaak het prettigst werkt, het is een omgeving waar de werknemer zich in kan vinden. Buiten werktijden of collegetijden heeft hij het apparaat ook bij zich, dat biedt mogelijkheden. Zoals het in de bus digitaal oefenen van een tentamen voor studenten. Voor bedrijven kan het de bereikbaarheid van werknemers vergroten.

Eén van de grootste gevaren bij BYOD is beveiliging. Omdat er gebruik gemaakt wordt van het eigen apparaat tij-

dens het werken of het volgen van een opleiding zou daar gevoelige informatie op kunnen staan van deze organisatie. Het grootste probleem waardoor de beveiliging in het geding zou kunnen komen is dat het je persoonlijke apparaat is en de organisatie niet de volledige controle over de informatiestroom op dat apparaat heeft.

Dit afstudeerproject richt zich op het aanpassen en herontwerpen van I2Vote zodat het binnen een BYOD omgeving gebruikt kan worden. BYOD zal dan ook een grote rol spelen bij de keuze voor het ontwikkelplatform.

2.2.2 CROSS-PLATFORM DEVELOPMENT

Cross-platform development sluit aan bij het BYOD principe. Studenten kiezen zelf hun eigen telefoon uit, de één kiest telefoon van Apple en de ander een telefoon van Samsung. Dit betekent dat er een aantal verschillende platformen in omloop zijn waar de mobiele applicatie van I2Vote op moet gaan werken.

Cross-platform ontwikkeling is het schrijven van programma's voor meerdere besturingssystemen (platformen) zoals Windows en OS X, maar ook Android en iOS. Er zijn een aantal manieren om dit voor elkaar te krijgen. Er kan gekozen worden om voor elk platform een nieuwe applicatie te schrijven. Maar er zijn ontwikkelaars geweest die vonden dat dat anders moest. Platformen zoals Xamarin en PhoneGap zijn onder andere ontwikkeld voor de mobiele markt. Xamarin biedt overigens tegenwoordig ook de mogelijkheid om applicaties voor de Windows of OS X te ontwikkelen. De manier van ontwikkelen bepaalt niet of een programma cross-platform is, het enige criterium is dat het op meerdere platformen werkt. De afstudeeropdracht valt precies in dit theoretische kader, I2Vote heeft te maken met de verschillende mobiele devices van de studenten waarvoor de applicatie geschikt gemaakt moet worden.

2.2.3 HUMAN-COMPUTER INTERACTION

Human-Computer Interaction (HCI) is een vakgebied binnen de informatica dat zich bezig houdt met onderzoek naar de interactie tussen mens en machine. De gebruikersinterface bepaalt voor een groot deel de bruikbaarheid van een applicatie. Het onderzoek naar de verschillende

aanwijsstrategieën op een touchscreen valt binnen het kader van HCI.

2.2.4 AUDIENCE RESPONSE SYSTEMS

Een ARS stelt een groep (bijvoorbeeld een lezing) in staat om op een onderwerp of vraag te stemmen. Het stemmen werkt vaak via een speciaal daarvoor ontwikkeld stemapparaat. Het is gebruikelijk dat deze apparaten de antwoorden naar een centrale server sturen. De antwoorden kunnen dan direct teruggekoppeld worden naar de deelnemers via bijvoorbeeld het apparaat waar ze mee hebben gestemd of een groot scherm (zoals een projector) waar de presentatie op wordt gegeven.

ARS heeft een aantal voordelen. Zo bevordert het gebruik de actieve participatie van het publiek aan bijvoorbeeld een college. Dit zou een positief effect kunnen hebben op wat deelnemers onthouden van een college (Hoyt et al., 2010).

Een nadeel is dat er vaak apart hardware voor aangeschaft moet worden. Bij het bezitten van deze apparatuur hoort (vaak duur) onderhoud. De vragen van een ARS zijn veelal gesloten of multiple-choice vragen. Het stellen van open vragen is meestal niet mogelijk.

I2Vote is een ARS. I2Vote biedt de mogelijkheid, net als met een gebruikelijk ARS, vragen te stellen aan het publiek en streeft het ernaar om de vraagsoorten uit te breiden met image-based vragen.

3 METHODIEK

Om de revisie op de baan te houden zijn er een aantal stappen geformuleerd. De stappen zijn geformuleerd vanuit de probleemstelling in hoofdstuk 1 en de gewenste situatie in paragraaf eveneens in hoofdstuk 1. Het afstudeerproject is opgebouwd uit twee onderzoeken en een ontwikkelperiode. De ontwikkelperiode wordt gebruikt om de applicatie structuur op te zetten en de belangrijkste onderdelen te implementeren. Daaraan vooraf gaan beide onderzoeken.

3.1 AANWIJSSTRATEGIEËN

Het eerste onderzoek heeft te maken met de verandering van de manier van aanwijzen. De PDA heeft een aanwijspen, de moderne smartphone heeft dat niet. Dit onderzoek moet een aanwijsstrategie adviseren die geschikt is voor gebruik op een smartphone zonder aanwijspen. Dit wordt conform een standaard opbouw van een onderzoek uitgevoerd. Het begint met een probleemstelling, daarna zal de methodiek worden beschreven gevolgd door de uitwerking van de in de methodiek beschreven methodes. Het onderzoek zal daarna de resultaten op een rij zetten gevolgd door de conclusie en aanbevelingen.

3.2 ONTWIKKELPLATFORMEN

Het tweede belangrijke onderzoek dat gedaan moet worden is het onderzoek naar ontwikkelplatformen. Hier gaat het over welk platform het meest geschikt is om het doel uit de probleemstelling te realiseren. Dit onderzoek bepaalt waarmee de I2Vote mobiele applicatie gebouwd gaat worden. Het gaat hierbij ook om het selecteren van bijvoorbeeld een geschikt framework. De opbouw van dit onderzoek is gelijk aan de in de vorige paragraaf beschreven structuur.

3.3 DE ONTWIKKEL PERIODE

Het vinden van een geschikt platform en een geschikte aanwijsstrategie zijn een goede stap richting een antwoord op de hoofdvraag. Maar het moet ook nog aangetoond

worden en daarbij moet een juiste applicatie structuur geadviseerd worden. De ontwikkelperiode is er om de gekozen platformen te bestuderen en uit te proberen om zo tot een geschikte applicatie structuur te komen. Daarnaast is het van belang om zoveel mogelijk de belangrijkste onderdelen zoals het beantwoorden van een image-based vraag te implementeren. Wanneer dit namelijk niet lukt met het gekozen platform moet er gekeken worden naar alternatieven.

Wanneer de voorgaande onderzoeken voltooid zijn, kan er begonnen worden om met het geadviseerde ontwikkelplatform te gaan ontwikkelen. Deze fase dient als een bevestiging van het resultaat uit het onderzoek ontwikkelplatformen.

Tijdens deze periode zal er zoveel mogelijk gebruik gemaakt worden van de Rational Unified Process methode (RUP). RUP is verkozen boven alternatieven zoals Agile en Waterfall. Agile vereist veelal veel kennis van de omgeving waarin wordt gewerkt en die is niet gegarandeerd bij dit afstudeerproject. Waterfall is niet flexibel genoeg doordat voor elke wijziging in het ontwerp weer bovenaan begonnen moet worden in het waterfall proces.

RUP iteraties zijn wel geschikt voor deze fase omdat het hier potentieel gaat om nieuwe stof. Wanneer er een ontwerp wordt gemaakt voor een bepaalde applicatie of voor een bepaald onderdeel van een applicatie door een ontwikkelaar die beperkte kennis heeft van het platform en/of framework waarmee wordt gewerkt is de kans groot dat dit ontwerp niet optimaal is. In dit geval zou het opnieuw ontwerpen (van bepaalde onderdelen) met de kennis van de vorige iteratie een uitkomst kunnen bieden om toch een degelijke applicatie structuur neer te zetten.

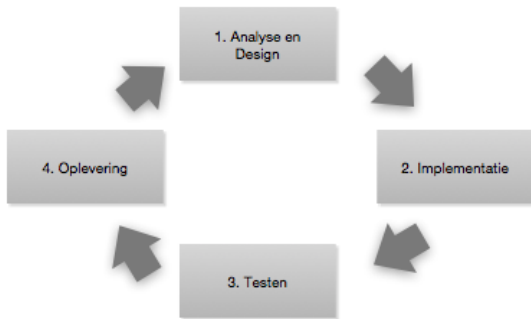
3.3.1 RUP FASEN

RUP kent vier verschillende basis fasen: Inception, Elaboration, Constructen en Transition. Elk van deze fasen heeft een hoofdtak, maar ze mogen elkaar in de tijd wel overlappen. In de Inceptionfase wordt bepaald wat de grenzen

van het project zijn en waarom het project gedaan moet worden. De tweede fase, Elaboration, is er om de technische haalbaarheid vast te stellen en om de functionele eisen te bepalen. In principe wordt in deze fase het uiteindelijke projectplan ook opgesteld. Construction, de derde fase, is bedoeld voor de uiteindelijke implementatie. Hier wordt het grootste deel van de ontwikkeling gedaan. De laatste fase is de Transition, de overdracht of oplevering van het geheel.

3.3.2 ITERATIES

Elke iteratie bestaat uit een aantal stappen. Het gaat om de stappen: Analyse en design, Implementatie, Testen en Oplevering. Elke iteratie is te vergelijken met een (klein) RUP project. In figuur 3 is dit proces te zien. Na elke iteratie start het proces opnieuw met het resultaat en kennis van de vorige iteratie totdat het eindresultaat is bereikt.



Figuur 3 Voorbeeld van een RUP iteratie

3.3.2.1 ANALYSE EN DESIGN

In deze fase van de iteratie wordt er gekeken naar hoe de oude situatie werkt. Nadat dit in kaart is gebracht (door de Use Cases te beschrijven) zal er een ontwerp gemaakt worden voor de nieuwe applicatie. Elke iteratie bouwt verder op de kennis van de vorige.

3.3.2.2 IMPLEMENTATIE

De implementatie fase is er om het ontwerp te implementeren. Testen vindt ook in deze fase plaats.

3.3.2.3 TESTEN

Testen gaat vrijwel gelijk op met het implementeren. In deze fase worden de resultaten of deelresultaten van de implementatie getest. Bij voorkeur gaat het testen middels Unit Testen (UT) binnen een test-driven development (TDD) omgeving. Binnen de afdeling Radiologie is geen continuous integration (CI) omgeving aanwezig. Dit is niet raar, Radiologie specialiseert zich immers niet op het ontwikkelen van Software. Het onderzoeken en opzetten van een CI omgeving valt niet binnen de scope van dit project.

3.3.2.4 OPLEVERING

Aan het einde van elke iteratie wordt er geprobeerd om een werkend onderdeel op te leveren. Bij voorkeur werkt het gehele systeem dan ook nog steeds samen. Dit blijft helaas afhankelijk van hoeveel extra werk het zou zijn om het I2Vote op deze manier in de ontwikkelfase draaiende te houden.

4 AANWIJSSTRATEGIEËN

In de vorige hoofdstukken is I2Vote geïntroduceerd en is het doel beschreven. Het systeem I2Vote wordt geschikt gemaakt voor nieuwe apparaten.

Nieuwe mobiele apparaten (zoals de iPhone 5s, Samsung Galaxy S4 en Google Nexus) worden, in tegenstelling tot de PDA, niet met een aanwijspen geleverd. Het migreren van een applicatie op een apparaat met pen bediend touchscreen naar een apparaat met vinger bediend touchscreen lijdt tot het stellen van de vraag of de aanwijsprecisie van het nieuwe aanwijsgereedschap (de vinger) voldoende is voor I2Vote.

Het is voor te stellen dat het gewenst is dat het te markeren gedeelte een groter gebied bestrijkt dan een punt. De vernieuwing van I2Vote zal in de eerste instantie inzetten op de bestaande functionaliteit, daar zit het selecteren van een gebied niet bij en is daarom geen onderdeel van dit onderzoek.

Dit onderzoek gaat in op de vraag of de precisie van de met vinger bediende touchscreen voldoende is voor het systeem I2Vote en welke aanwijsstrategie de voorkeur heeft. Om dit te bepalen worden een aantal mogelijke oplossingen naast elkaar gezet. Van belang is dat de gebruikerservaring verbetert ten opzichte van het oude systeem en dat de aanwijsprecisie nauwkeurig blijft.

Dit hoofdstuk begint met de probleemstelling gevolgd door de onderzoeksmethode waarin wordt uitgelegd hoe het onderzoek is aangepakt. Na de methode wordt de literatuur bekeken gevolgd door een beschrijven van de potentiële oplossingen. Dit hoofdstuk wordt afgesloten met de resultaten, een conclusie en het advies.

4.1 PROBLEEMSTELLING

De precisie van aanwijzen is met de vinger minder dan met een aanwijspen, simpelweg omdat het raakoppervlak van de pen vele malen kleiner is (rond 0.5mm diameter tegen

over ongeveer 1 cm). De interfaces zijn door de jaren veranderd om deze wijziging te accommoderen. Met name in de vorm van grotere knoppen en afbeeldingen. Met de I2Vote applicatie gaat het om het aanwijzen van lokaties op een afbeelding, de lokaties kunnen erg klein zijn.

Van belang is dat er een geschikte oplossing wordt gevonden voor de manier van lokatiebepaling op afbeeldingen in I2Vote die exact en eenvoudig genoeg is. Het streven is naar een aanwijsstrategie die door de gebruiker als intuïtief ervaren wordt zonder dat het conflicten veroorzaakt met de standaard (of native) gebaren van het apparaat. Onder conflict wordt het vervangen van een native gebruikersinterfacegebaar van het toestel verstaan zoals: zoomen wordt gedaan door met twee vingers op het scherm in tegenovergestelde richting te bewegen. De criteria zijn als volgt:

- De leercurve van de aanwijsstrategie moet door de proefpersonen als relatief laag worden ervaren.
- De gekozen aanwijsstrategie moet de nauwkeurigheid van de aanwijspen benaderen, om het haalbaar te houden is een maximale afwijking van 1 mm gesteld.
- Er mogen geen conflicten optreden met de gebaren van de native gebruikersinterface.

Het hoeft geen nieuwe strategie te zijn, het gaat hier om een literatuuronderzoek naar de verschillende ideeën over aanwijsstrategieën op een met vinger bediend touchscreen en de evaluatie daarvan om de beste eruit te kiezen. De hoofdvraag die hierbij luidt: welke aanwijsstrategie is de meest nauwkeurige en eenvoudige voor het gebruik met I2Vote op moderne mobiele apparaten (smartphones)?

Om deze hoofdvraag goed te kunnen beantwoorden zijn de volgende deelvragen geformuleerd:

- 1 Welke publicaties zijn er op dit gebied van aanwijsstrategieën?
- 2 Welke geschikte aanwijsstrategieën kunnen hieruit afgeleid worden?
- 3 Welke aanwijsstrategie is het meest precies?
- 4 Welke aanwijsstrategieën vindt de gebruiker het beste werken?

4.2 RELEVANTIE

Het beantwoorden van het probleem is interessant voor gebruikers (studenten en docenten) van I2Vote, de ontwikkelaar(s) en de opdrachtgever(s). Daarnaast kan het interessant zijn voor vergelijkbare projecten. Voor de ontwikkelaar(s) en opdrachtgever(s) is de keuze onderbouwd en vergeleken met de alternatieven. Dit geeft vertrouwen in de gekozen oplossing.

Voor de gebruikers van het systeem levert het een goed onderbouwde bruikbare aanwijstrategie op. De precisie van het aanwijzen heeft direct invloed op de betrouwbaarheid van de antwoorden. Deze betrouwbaarheid geeft docenten die gebruik maken van het systeem het vertrouwen dat er feedback gegeven wordt op antwoorden die de studenten hebben gegeven.

Daarnaast kan het op gebied van HCI een extra bevestiging zijn van de bruikbaarheid van de gedocumenteerde strategieën. Uit de literatuur worden steeds de meest geschikte oplossingen gekozen en tegenover elkaar gezet. Het zou kunnen dat er strategieën worden vergeleken of gecombineerd worden waar dat nog niet eerder mee is gebeurd en dat zou nieuwe inzichten kunnen geven.

4.3 METHODE

4.3.1 INFORMATIE VERZAMELEN EN ANALYSEREN

Om uit te zoeken welke geschikte strategieën er bestaan wordt er een literatuuronderzoek gedaan. Hierin worden publicaties over het onderwerp verzameld en geraadpleegd. Daarnaast wordt er overlegd met een HCI specialist. Met deze stap wordt deelvraag 1 beantwoord.

4.3.2 MOGELIJKE OPLOSSINGEN

Nadat de informatie is verzameld worden er een aantal mogelijke oplossingen gekozen en beschreven. Daarbij wordt rekening gehouden met de criteria uit de probleemstelling en kunnen de strategieën aangepast worden om hier aan te voldoen. Deelvraag 1 wordt hiermee beantwoord.

4.3.3 ONTWERP VAN DE PROEF

Om te bepalen welke strategie het meest geschikt is worden een aantal kandidaten gevraagd. De kandidaten voeren een aantal proeven uit en zullen naderhand gevraagd worden om hier een aantal stellingen over te beoordelen. (zie paragraaf de aanwijaak voor een beschrijving van de aanwijaak)

De kandidaten worden gevraagd om per aanwijastrategie acht keer een doel (punt) op een afbeelding aan te wijzen. Een korte uitleg over hoe de strategie werkt wordt gegeven bij elke strategie. De eerste vier keren worden beschouwd als leerproeven en de laatste vier keren worden opgeslagen als metingen. Bij een meting wordt per kandidaat per aanwijaak per strategie opgeslagen hoe lang ze over het aanwijzen hebben gedaan, hoeveel pogingen er zijn gedaan om het doel aan te wijzen en de uiteindelijke afstand tot het doel. Om de proeven te onderscheiden wordt er per resultaat opgeslagen tot welke strategie deze behoort. Dit komt neer op de volgende velden:

- X-coördinaat van aanwijaadoel.
- Y-coördinaat van aanwijaadoel.
- Werkelijke X-coördinaat.
- Werkelijke Y-coördinaat.
- Pogingen.
- Verstreken tijd.
- Strategie.

Met behulp van deze gegevens kunnen de verschillende strategieën vergeleken worden op gebied van de afstand van het doel, de tijd die nodig is om het doel aan te wijzen en het aantal pogingen. Per kandidaat wordt het gemiddelde uiterekend, deze data wordt weer per strategie geordend zodat de verschillende strategieën vergeleken kunnen worden.

Om er voor te zorgen dat elke strategie een keer als eerste wordt getest, wordt de volgorde van de strategieën per proefpersoon veranderd. De beginstrategie wordt bij elke volgende kandidaat achteraan gezet:

- $A \rightarrow B \rightarrow C$
- $B \rightarrow C \rightarrow A$
- $C \rightarrow A \rightarrow B$

A, B en C zijn in dit voorbeeld aanwijs strategieën en 1, 2 en 3 stellen drie kandidaten voor. In het voorbeeld zijn drie aanwijs strategieën gebruikt, in dit geval zit het ideale aantal kandidaten op een meervoud van drie. Zouden er twee strategieën zijn, dan zit het aantal op een meervoud van twee. Zo komt elke strategie even vaak op een bepaald moment aan de beurt.

Het doel van de stellingen is om te bepalen welke strategie de gebruiker als het meest eenvoudig ervaart. Daarnaast moeten de stellingen inzicht geven in hoeverre de volgens de gebruiker de meeste exacte strategie overeenkomt met de meest exacte strategie volgens de resultaten.

De resultaten van deze proef beantwoorden deelvragen 3 en 4.

4.3.4 DE AANWIJSTAAK

De aanwijaak bestaat uit het aanwijzen van een stip op een afbeelding. De stip krijgt een highlight zodat deze beter zichtbaar is en zodat de kandidaat niet hoeft te zoeken naar de locatie van de stip. Hoe de kandidaat de taak moet uitvoeren is afhankelijk van de strategie. In de paragraaf mogelijke strategieën wordt de strategie en hoe de aanwijaak daarbij uitgevoerd moet worden uitgelegd.

4.3.5 APPARATUUR

Om de proeven uit te voeren is een smartphone met een touchscreen nodig. Het gaat om een HTC One S met een 4,3" scherm van 960 bij 540 pixels, 256 ppi. Elke gegeven mogelijke oplossing wordt uitgewerkt om te gebruiken met de proefafbeelding. De HTC One S wordt uitsluitend in portrait-mode gebruikt om verschil in de resultaten als gevolg van verschillende gebruikte standen te voorkomen. Om de data weg te schrijven is er een werkstation ingericht met een database.

4.3.6 MATERIAAL

Het proefmateriaal bestaat uit een afbeelding, een proefapplicatie en een vragenlijst. Op de afbeelding wordt het aanwijsdoel duidelijk aangegeven. Deze afbeelding krijgt aan elke kant een marge van 10 pixels zodat de zijanten eenvoudig aan te raken zijn. De originele afbeelding is 562x461 pixels. De proefafbeelding wordt weergegeven via de HTC One S in portrait-mode. Het scherm is dan 540 pixels breed, de afbeelding $540 - 10 - 10 = 520$ pixels breed.

Een millimeter op het scherm komt overeen met 10,08 pixels waardoor de afbeelding op het scherm $520/10,08 = 51,59$ mm breed is. De verhouding van de afbeelding blijft gelijk.

Als aanwijsdoel wordt gebruikt gemaakt van een stip. Bij het kiezen van de proefafbeelding is rekening gehouden met de mogelijke aanwijs doelen die voor kunnen komen in I2Vote. Om het formaat van de punt te bepalen is er gekeken naar de afbeeldingen die zijn gebruikt in de pilot van I2Vote. Er is bewust gekozen om het aanwijsdoel overduidelijk aan te geven op de proefafbeelding, omdat de kennis van de kandidaten geen invloed mag hebben op het resultaat. De afstand tot het doel wordt berekend vanaf het centrum van de stip. Het doel heeft een diameter van 4 pixels op het scherm, wat ongeveer gelijk is aan een halve millimeter.

Een proefapplicatie met een implementatie van de verschillende strategieën wordt gebruikt om de afbeelding te tonen. Deze wordt volledig geschreven met behulp van NodeJS, HTML5, Javascript en MongoDB. De proef kan daardoor volledig worden uitgevoerd in Google Chrome voor Android.

Na het doen van elke proef (strategie) wordt de kandidaat de volgende stellingen voorgelegd:

- Strategie x was eenvoudig om mee te werken.
- Ik kon de aanwijaak nauwkeurig uitvoeren met strategie x.
- Strategie x was eenvoudig om te leren.

Waarbij x het nummer van de strategie of proef is. De beoordeling wordt gegeven op de volgende schaal: helemaal niet mee eens, niet mee eens, neutraal, mee eens, helemaal mee eens.

4.3.7 PROEFPERSONEN

De groep proefpersonen bestaat uit zes personen. Een kandidaat mag zelf bepalen welke vinger(s) er gebruikt word(t)(en) zolang deze gedurende de proef maar dezelfde is. De groep kandidaten bestaat uit mannen tussen de 25 en 35.

4.4 LITERATUURONDERZOEK

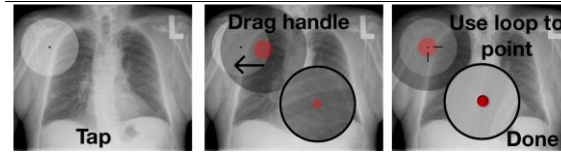
Het literatuuronderzoek bestaat voornamelijk uit interpretaties van bestaande literatuur, voor het resultaat hiervan wordt verwezen naar bijlage II.

4.5 MOGELIJKE STRATEGIEËN

Aan de oplossing zijn een aantal criteria gesteld. De oplossing mag niet botsen met de gebaren van de gebruikersinterface van het apparaat en moet de methode nauwkeurig genoeg kunnen aanwijzen om de I2Vote vragen mee te beantwoorden. De kandidaat oplossingen moeten hier potentieel aan voldoen. De strategieën zijn gekozen aan de hand van welke in de verschillende onderzoeken het beste (de meeste potentie hebben) naar voren kwamen.

4.5.1 SHIFT

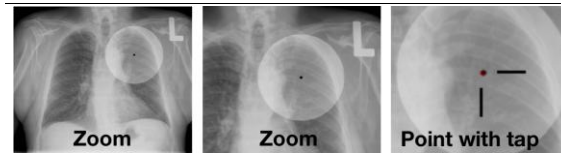
Een mogelijke oplossing is het toepassen van een door Shift geïnspireerde strategie. Shift botst op het eerste gezicht niet met de gebaren van de native interfaces van iPhone en Android. Door de vinger op het scherm te houden wordt er naast de vinger een vergrote versie van de ruimte onder de vinger getoond. Door gebruik te maken van een vergrote versie van een gedeelte van de afbeelding kan er nauwkeuriger een locatie aangewezen worden. Er wordt na het selecteren de mogelijkheid gegeven om de locatie te bevestigen of aan te passen. Het aanpassen gaat door middel van opnieuw een locatie te selecteren en het bevestigen zal gefaciliteerd worden door een submit-knop. Deze strategie wordt iets anders geïmplementeerd dan hierboven beschreven. Het vergrote gedeelte verschijnt op het scherm na een tap, na een tweede tap verdwijnt het weer. Naast het vergrote gedeelte op de plek waar het scherm is aangeraakt wordt een handvat getekend. Door het handvat over de afbeelding te slepen verplaatst de aanwijslocatie (Vergelijkbaar met de Offset strategie uit paragraaf Offset). Figuur 4 probeert dit grafisch uit te leggen. Onder een poging wordt bij Shift verstaan het verslepen van het handvat.



Figuur 4 Instructie bij uitvoering aanwijstaak met Shift

4.5.2 ZOOM-POINTING

Zoom-Pointing zou een oplossing kunnen zijn. Door de mogelijkheid om oneindig in te zoomen is deze strategie enorm nauwkeurig. Hiermee is de kans groot dat er voldaan wordt aan de nauwkeurigheidseis. De strategie beschrijft dat het vergroten gebeurt door een selectie te maken van een bepaald gebied. Om niet te botsen met de native gebaren en om de gebruikerservaring te verbeteren zou hier het zoom-gebaar van het apparaat overgenomen kunnen worden. Wederom lijkt het een goed idee om hier de gebruiker de mogelijkheid te geven om het antwoord aan te passen voor er bevestigd wordt. In figuur 5 wordt hier een grafische uitleg bij gegeven.



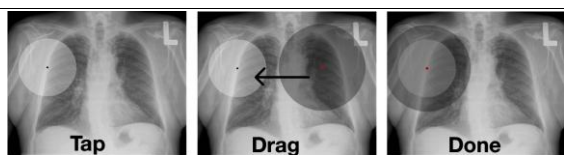
Figuur 5 Instructie bij uitvoering aanwijstaak met Zoom-Pointing

Deze strategie heeft volgens Pär-Anderson Albinsson en S. Zhai (Albinsson & Zhai, 2003) als nadeel dat er relatief veel arm en hand beweging nodig is voor het selecteren van meerdere doelen, dat moet volgens F. Wang en X. Ren (Wang & Ren, 2009) ook vermeden worden. Gezien er in de applicatie van I2Vote per vraag maar één doel is, is de verwachting dat dit effect minimaal is. Omdat er wordt gezegd dat Zoom-Pointing minder goed werkt bij grote doelen is het vergroten van een gebied geen voorwaarde voor het kunnen selecteren. Bij Zoom-Pointing wordt elke tap beschouwd als een aanwijspoging.

4.5.3 OFFSET

De laatste optie is een variatie op de Offset methode, maar dan zonder het Take-Off effect. In eerste instantie is het

een directe strategie. De gebruiker raakt met zijn vinger een lokatie aan (tap) en de contactplek wordt gebruikt als aanwijslokatie. Op de aanwijslokatie wordt een vizier getekend. Dit vizier kan dan verplaatst worden. Het vizier moet groot genoeg zijn zodat de gebruiker er mee kan schuiven zonder dat de vinger het centrum van het vizier (de aanwijslokatie) blokkeert. Als de gebruiker tevreden is met de aangewezen lokatie kan er bevestigd worden met behulp van een submit-knop. Deze strategie zou kunnen werken zonder zoom, omdat het centrum van het vizier klein kan zijn. Daarmee kan dan heel precies aangewezen worden. Figuur 6 geeft hier een grafische uitleg bij. Bij de offset strategie wordt het verslepen van het vizier beschouwd als een aanwijspoging.



Figuur 6 Instructie bij uitvoering aanwijstaak met Offset

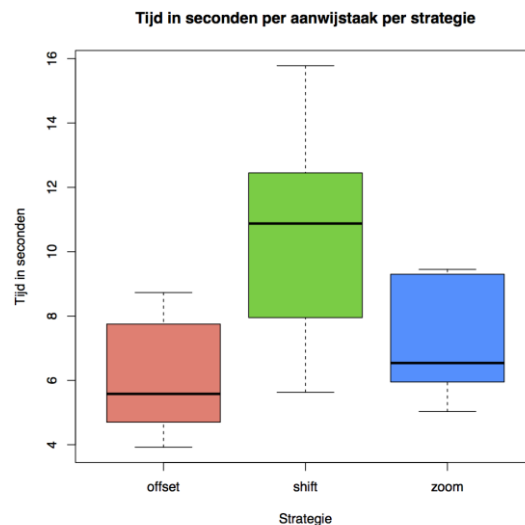
4.6 RESULTATEN

De meetresultaten worden beschreven met behulp van box plot diagrammen. De metingen worden per strategie vergeleken, van elke kandidaat wordt het gemiddelde genomen van haar meetresultaten en dit wordt vervolgens geordend per strategie. De beoordelingen op de stellingen worden per stelling vergeleken.

4.6.1 METINGEN

De meetresultaten bestaan uit diagrammen met daarbij een interpretatie van deze diagrammen.

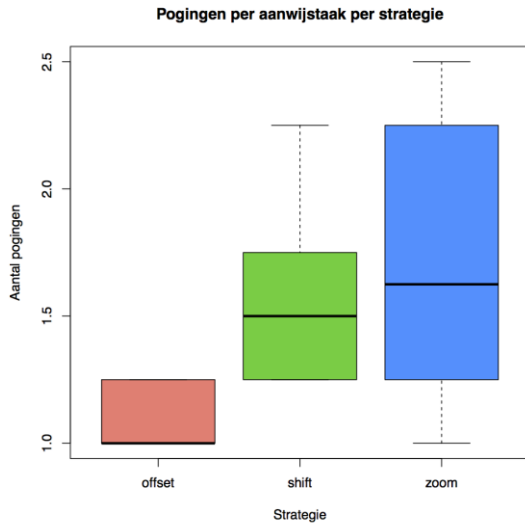
4.6.1.1 TIJD VERGELEKEN



Figuur 7 Snelheid van de uitvoering aanwijstaak

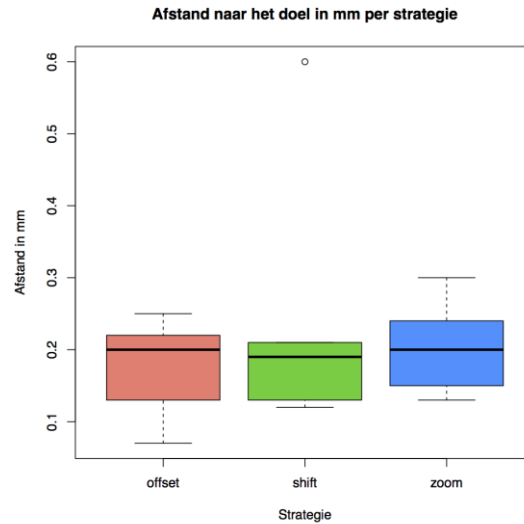
Uit de resultaten in figuur 7 van de proef blijkt dat Offset en Zoom-Pointing een vergelijkbare hoeveelheid tijd kostte om de aanwijstaak uit te voeren. Offset was hierin gemiddeld iets sneller. Ook kun je zien aan het hogere maximum, hogere minimum en spreiding van de tijdresultaten dat een aanwijstaak met Zoom-Pointing consequent langer duurder dan Offset. Shift kostte veruit de meeste tijd met de grootste spreiding.

Dat Zoom-Pointing langzamer is dan Offset is te verklaren door het feit dat Zoom-Pointing de extra handeling zoomen vereist, zoomen kost blijkbaar meer dan het verslepen van een vizier bij Offset. Daarnaast verlies je bij zoomen het totaal overzicht zodra je teveel vergroot. De slechte resultaten van Shift lijken veroorzaakt door de te grote vergrotingsfactor in het vergrote gebied. De snelheid van de aanwijzer in het vergrote gebied is hierdoor hoger waardoor de gebruiker het gevoel krijgt dat de cursor steeds voorbij het aanwijsdoel springt. Hierdoor moet de gebruiker zich meer concentreren op de aanwijstaak en kost het meer moeite en tijd.



Figuur 8 Aantal benodigde pogingen bij de aanwijstaak

Het aantal pogingen dat een gebruiker nodig had om een de aanwijstaak uit te voeren lag bij alle strategieën volgens figuur 8 redelijk bij elkaar in de buurt. Offset heeft hier wederom de beste resultaten met de kleinste spreiding, gevolgd door Shift en als laatste Zoom-Pointing. Dit zou te verklaren kunnen zijn doordat de gebruiker bij Zoom-Pointing het doel eerst probeert aan te wijzen voordat er verder wordt vergroot en dit elke keer wanneer er vergroot wordt herhaalt totdat de gewenste en aanwijsprecisie is bereikt. De slechte resultaten van Shift zouden opnieuw verklaard kunnen worden door het verspringen van de cursor in het vergrote gebied.



Figuur 9 De afstand naar het doel per aanwijstaak

Volgens de resultaten in figuur 9 is te zien dat Shift, ondanks het verspringen van de cursor, de meest nauwkeurigste is gevolgd door Offset en daarna Zoom-Pointing. Let wel op dat er bij Shift één resultaat ver boven de rest uitspringt. Dit komt waarschijnlijk door een kandidaat die te snel probeerde aan te wijzen. Hiermee lijken alle strategieën vrij nauwkeurig en in ieder geval onder de grens van 0.5 mm die is gesteld in de probleemstelling.

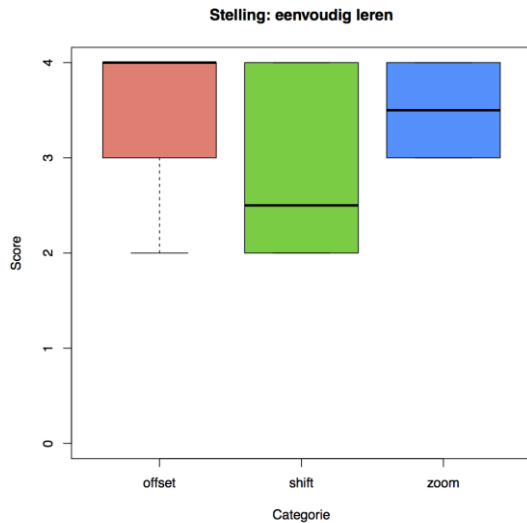
4.6.1.4 SAMENVATTEND

Neem je alle resultaten samen dan lijkt Shift het nauwkeurigst, maar ten koste van de tijd. Daarnaast ligt het aantal pogingen ten opzichte van Offset hoger. Offset lijkt daarop het nauwkeurigst met de laagste tijd en het laagste aantal pogingen. Het lage aantal pogingen en tijd bij Offset suggereren dat deze strategie eenvoudig is in gebruik. Zoom-Pointing komt als minst nauwkeurig uit de proef waarbij het aantal pogingen behoorlijk varieert en over het algemeen hoger ligt dan bij de overige resultaten. De tijd die nodig is bij Zoom-Pointing ligt tussen Shift en Offset in.

4.6.2 STELLINGEN

Om de stellingen te vergelijken is de schaal van 'Helemaal niet mee eens' naar 'Helemaal mee eens' omgezet naar een numerieke schaal van 0 naar 4. Hier staat 0 gelijk aan 'Helemaal niet mee eens' en 4 gelijk aan 'Helemaal mee eens'. Het numeriek uitdrukken van de resultaten geeft een duidelijker en makkelijker af te lezen resultaat en is eenvoudiger te vergelijken dan de originele schaal.

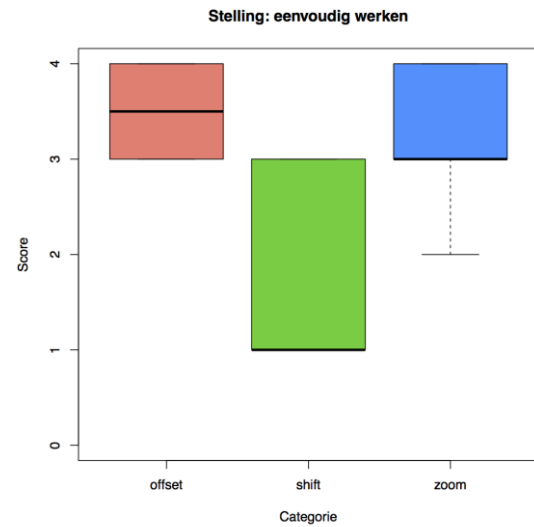
4.6.2.1 DEZE STRATEGIE WAS EENVOUDIG OM TE LEREN



Figuur 10 Beoordeling stelling leren per aanwijstaak

Uit de grafiek in figuur 10 blijkt dat Zoom-Pointing en Offset ongeveer even eenvoudig zijn om te leren waarbij Zoom-Pointing iets mindere beoordelingen lijkt te krijgen. Shift lijkt veruit het moeilijkst te leren met de meest verdeelde meningen. Offset leunt meer naar een positievere beoordeling dan Zoom-Pointing, Offset heeft een lagere laagste beoordeling dan zoom, waardoor je kunt stellen dat Offset misschien niet voor iedereen even eenvoudig is om te leren.

4.6.2.2 DEZE STRATEGIE WAS EENVOUDIG OM MEE TE WERKEN

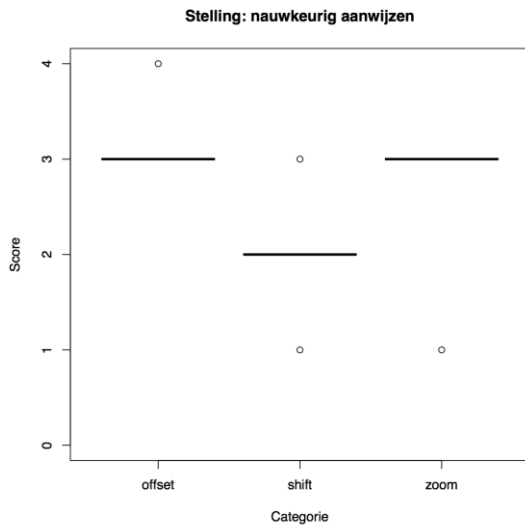


Figuur 11 Beoordeling stelling werken per aanwijstaak

Figuur 11 laat zien dat de verhoudingen tussen de strategieën hier vergelijkbaar zijn met de verhouding bij de stelling: deze strategie was eenvoudig om te leren. Offset lijkt als beste beoordeeld te worden gevolgd door Zoom-Pointing. Shift krijgt hier de slechtste beoordeling, met de mediaan zelfs onder het midden van de schaal waardoor deze implementatie van Shift onvoldoende eenvoudig is in gebruik.

Als dit naast de tijd resultaten van de tijd wordt gezet zou er gesteld kunnen worden dat er een verband is tussen de tijd dat het aanwijzen kost en de gebruikerservaring: hoe meer tijd het kost, hoe moeilijker het aanwijzen met deze strategie is.

4.6.2.3 IK KON DE AANWIJSTAAK NAUWKEURIG UITVOEREN MET DEZE STRATEGIE



Figuur 12 Stelling vertrouwen in nauwkeurigheid per aanwijstaak

Figuur 12 geeft een verhouding vergelijkbaar met de vorige stellingen. Offset lijkt als beste beoordeeld te worden gevolgd door Zoom-Pointing en Shift. De resultaten liggen iets dicht bij elkaar met minder uitzonderingen. Shift heeft de meest gevarieerde resultaten met de laagste score. Bij Offset was iedereen even tevreden over de nauwkeurigheid met één uitzondering naar boven toe. Bij Zoom-Pointing lijken de meningen ook gelijk te zijn met één vrij negatieve uitzondering.

4.6.2.4 SAMENVATTEND

Als we naar de resultaten van de stellingen kijken lijkt Offset over het algemeen beter beoordeeld te worden door de kandidaten. Zoom-Pointing komt wel steeds dichtbij of haalt vergelijkbare scores. Shift scoort op alle stellingen relatief laag en lijkt de minst populaire strategie. Offset lijkt hiermee de favoriet te zijn van de testgroep, de kandidaten hadden vertrouwen in de nauwkeurigheid van Offset en vonden de strategie eenvoudig om te leren en om mee te werken.

4.6.3 OPMERKINGEN

De kandidaten vonden dat het vergrote gedeelte van het scherm af en toe versprong. Hierdoor werd de aanwijstaak als moeilijk ervaren. Dit lijkt veroorzaakt door een te grote vergroting. Dit zou opgelost kunnen worden door een vertraging in te bouwen. Een kandidaat zou bijvoorbeeld de mogelijkheid moeten krijgen om het aanwijs gereedschap van Shift ook middels het vergrootglas te kunnen verplaatsen. De gebruiker sleept de cursor dan over het vergrote gedeelte in plaats van de volledige afbeelding. De snelheid ligt hierdoor een stuk lager en daarmee zou het zichtbare verspringen van de cursor verminderd kunnen worden. In eerder onderzoek (Vogel & Baudisch, 2007) wordt dit aangeraden. Bij het huidige onderzoek liet de beschikbare tijd het niet toe om dit alsnog te implementeren. Het gevolg hiervan lijkt te zijn dat de beoordelingen bij Shift erg laag liggen vergeleken met de overige strategieën. Uit de notities van de observaties van de kandidaten tijdens de proef viel op dat de kandidaten probeerden om het vergrootglas te verslepen in plaats van het handvat (figuur 4 geeft een voorbeeld van Shift).

Daarnaast werd er bij Shift de opmerking gemaakt dat doordat het vergrootglas de randen van het scherm vermijdt, het vergrootglas af en toe onder de vinger wordt getekend. Om dit te voorkomen zou een slimmer algoritme bedacht kunnen worden die rekening houdt met de positie van de hele vinger in plaats van alleen de plaats waar het scherm wordt aangeraakt.

4.7 CONCLUSIE EN ADVIES

Geen van de voorgestelde aanwijsstrategieën botst met de gebaren van de native-interface van het testtoestel. Daarnaast waren alle strategieën ruim voldoende nauwkeurig, met een afwijking van minder dan een 0,5 mm. Alle kandidaten hadden alle strategieën redelijk snel door. Hiermee is voldaan aan alle eisen uit de probleemstelling uit de probleemstelling, maar is de hoofdvraag nog niet beantwoord.

Offset scoort qua aanwijs pogingen en tijd beter dan Zoom-Pointing en Shift. Alle strategieën zijn nauwkeurig genoeg om de aanwijspen te vervangen, maar Shift is het meest nauwkeurig. Offset is iets nauwkeuriger dan Zoom-

Pointing. Zoom-Pointing wordt door de kandidaten positief beoordeeld en komt in de buurt van Offset. Offset wordt als meest eenvoudig en nauwkeurig beoordeeld door de kandidaten. Shift komt niet in de buurt en wordt op alle stellingen lager beoordeeld. In gebruik kreeg Shift zelfs een onvoldoende.

Offset is de meeste eenvoudige in gebruik, eenvoudig om te leren en wordt als meest nauwkeurig ervaren. Er kan gesteld worden dat Shift de meeste geschikte strategie is wanneer uitsluitend de aanwijs precisie het enige criterium is. Zodra de tijd, het aantal benodigde pogingen en de gebruikerservaring mee tellen zijn Offset en Zoom-Pointing het meest geschikt voor een aanwijstaak.

Rekening houdende met het aantal pogingen, de tijd die een kandidaat nodig had om de taak uit te voeren, de nauwkeurigheid van de strategie en de beoordelingen op de stellingen wordt geadviseerd om de Offset strategie te gebruiken bij I2Vote. Daar kan eventueel afhankelijk van de afbeelding een vergrotingselement aan toegevoegd worden vergelijkbaar met die in de Zoom-Pointing strategie.

Offset voldoet door haar nauwkeurigheid en eenvoud aan alle in de probleemstelling gestelde eisen. Offset heeft een afwijking van minder dan een 0,5 millimeter op een scherm van 4,3" inch en is eenvoudig te gebruiken en te leren.

5 ONTWIKKELPLATFORMEN

5.1 NATIVE, WEB OF EEN HYBRID OMGEVING

5.1.1 PROBLEEM- EN DOELSTELLING

Het huidige ontwikkelplatform is niet geschikt voor de meeste moderne mobiele apparaten. Daarom moet er een vergelijking komen van de geschikte beschikbare platformen waaruit één gekozen kan worden die in de hoogste mate aan alle eisen voldoet. De eisen die aan het platform zijn gesteld zijn:

- De ontwikkeltijd is belangrijk, korter is beter.
- Er moet voldoende ondersteuning zijn voor het platform. Een actieve community met de waarschijnlijkheid dat deze niet snel zal verdwijnen.
- Het platform moet ondersteuning bieden voor automatisch testen.
- De functionaliteit van I2Vote mag er niet op inleveren. (Met name met het oog op de image-based vragen)

Hierbij is de hoofdvraag geformuleerd: welk ontwikkelplatform past het best bij I2Vote als er rekening gehouden wordt met de ontwikkeltijd, onderhoudbaarheid, kosten en de functionaliteit van I2Vote.

5.1.2 RELEVANTIE

Dit literatuuronderzoek kan interessant zijn voor ontwikkelaars die bezig zijn met een vergelijkbaar project zoals een ARS, waarbij er gemigreerd moet worden van een verouderd platform of uitgebreid moet gaan worden naar meerdere moderne mobiele apparaten. Het aantal opties is enorm, daarom is dit onderzoek noodzakelijk voor dit afstudeerproject om onderbouwd een keuze te maken voor een geschikt ontwikkelplatform.

Zijn er dan nog geen onderzoeken gedaan naar de verschillende ontwikkelplatformen? Waarschijnlijk een heleboel, maar elk project is weer anders. Waardoor het bij elk project noodzakelijk is om opnieuw te evalueren welk platform het meest geschikt is.

5.1.3 METHODE

Om uiteindelijk tot een geschikt ontwikkelplatform te komen zal er eerste een literatuuronderzoek plaatsvinden. Het gaat hier voornamelijk om het informatie in winnen via blogs, discussieforums, (voor zover mogelijk wetenschappelijke) artikelen en de informatie die op de websites van de platformen zelf beschikbaar is. Uit dit literatuuronderzoek zullen een aantal mogelijke oplossingen worden beschreven. De mogelijke platformen kunnen dan vergeleken worden met behulp van de eisen gesteld in de probleemstelling bij dit onderzoek. In deze vergelijkingen worden de voor- en nadelen naast elkaar gezet om zo te bepalen met welk platform verder gewerkt gaat worden.

Om te meten tegen eisen als de ontwikkeltijd en de verwachte gebruikerservaring zal gekeken worden naar andere projecten die met het betreffende ontwikkelplatform zijn gedaan. Hieruit zal dan een schatting gemaakt worden per platform in hoeverre het aan deze eis voldoet.

Wanneer je wilt toetsen of alle functionaliteit met het betreffende platform valt te ontwikkelen is het belangrijk om exact te weten wat deze eisen zijn. Daarom zullen er aan de hand van de huidige implementatie van I2Vote requirements worden opgemaakt waarmee dit getoetst kan worden. Over het algemeen valt voor elk probleem met elk platform wel een oplossing te vinden, daarom moet er gekeken worden naar de meest kritieke requirements en de eventuele mogelijke oplossingen. De meest betrouwbare methode om dit te toetsen is simpelweg het implementeren van deze functionaliteit, maar qua tijd is dit niet praktisch. Om te bepalen of het implementeren van de requirements mogelijk is met een bepaald platform zal gezocht worden naar applicaties met vergelijkbare functionaliteiten of wordt er een beroep gedaan op de community of eigenkennis van een platform.

5.1.4 REQUIREMENTS

Bij I2Vote horen een aantal use cases die zijn beschreven in bijlage III.

Uit de use cases zijn de onderdelen gehaald waarvan de verwachting is dat de implementatie daarvan kan verschillen per platform.

- De applicatie stelt image-base vragen.
- De applicatie ondersteunt realtime colleges. Waarbij de docent de volgende vraag bij de studenten opstart.
- Het aanwijzen op een afbeelding moet met een betrouwbare nauwkeurigheid gedaan kunnen worden (met een afwijking van maximaal rond de 0.5mm zoals in hoofdstuk 4.

De overige requirements in de use cases kunnen naar verwachting ongeacht het platform relatief eenvoudig geïmplementeerd worden.

5.1.5 LITERATUURONDERZOEK

Het literatuuronderzoek bestaat voornamelijk uit interpretaties van bestaande literatuur, voor het resultaat hiervan wordt verwezen naar bijlage III.

5.1.6 MOGELIJKE PLATFORMEN

Uit het literatuuronderzoek blijkt dat web based of hybrid oplossingen afhankelijk van de use case een veel gemaakte keuze is. Niet alle hybrid oplossingen lijken van dezelfde kwaliteit. Hier zijn op basis van het literatuuronderzoek een aantal platformen geselecteerd. Van elk type (hybrid, native, web applicaties en runtime) is geprobeerd om een veelbelovend en populair platform mee te nemen. De platformen komen niet persé voor in het literatuuronderzoek en zijn geselecteerd op het type platform en op basis van gebruikerservaringen op community pagina's zoals Stackoverflow⁶. Daarnaast is de enquête van Vision Mobile (Mobile., 2012) bepalend geweest bij de selectie van de verschillende mogelijke platformen.

5.1.6.1 PHONEGAP

PhoneGap is een wrapper, dat wil zeggen dat PhoneGap een native applicatie waarin een Webview draait die een HTML/CSS/Javascript applicatie (web applicatie) weergeeft. Om de Native API's te benaderen is er een brug geschreven tussen de Webview en de Native applicatie. Via

deze brug kunnen zaken zoals de camera, GPS en contacten benaderd worden.

5.1.6.2 XAMARIN

Xamarin is een cross-platform ontwikkelomgeving waarbij er in .NET wordt ontwikkeld. Xamarin biedt de native look and feel door de ontwikkelaar per platform de front-end te laten schrijven. De back-end of de core van de applicatie kan wel gedeeld worden. De performance wordt gehaald door gebruik te maken van een .NET runtime/virtual machine (bijvoorbeeld Mono voor Android en iOS). De .NET code kan via een abstractielaag bij de native API's en interface elementen. Er is voor het ontwikkelen van een GUI daarom kennis nodig van de architectuur van de platformen waar je voor ontwikkeld. Zoals applicatie lifecycles en interface structuur. Omdat de runtime in de applicatie wordt verwerkt is deze net iets groter (ongeveer 2MB) dan een native variant.

5.1.6.3 WEB APPLICATIONS

Een web applicatie is een website geschikt voor de kleinere scherm formaten en browsers van mobiele apparaten. Net als bij PhoneGap heb je hier de keuze uit tal van web frameworks. Web applicaties worden via de browser van het mobiele apparaat gebruikt. Op dit platform is in principe alle code gedeeld net als bij PhoneGap.

5.1.6.4 NATIVE

Native ontwikkeling volgt de bij het platform horende regels en structuur. Hiermee schrijf je een volledige applicatie per platform in de programmeertaal die daarbij hoort en wordt er geen code gedeeld.

5.1.7 OPLOSSING

Om een goede vergelijking te maken zijn de genoemde criteria in de probleemstelling van dit onderzoek in een tabel gepresenteerd.

⁶ <http://stackoverflow.com>

Ondersteuning	Ontwikkeltype	Functioneleit	Testen
Web Community afhankelijk van het gekozen framework. Mogelijk uitbreiden.	Niet meer uitgegeven van Javascript en is de versie die de meeste browsers kan draaien.	HTML5 Canvas + Javascript. Ook gebruikt in het onderzoek naar andere ontwerpen.	Manie, Jasmine. UnitJS met behulp van bijvoorbeeld GruntJS
PhoneGap Gelijk met Web.	Gelijk met Web.	Gelijk met Web.	Gelijk met Web.
Xamarin Community en betaald. Wordt gesponsord door grote namen zoals Microsoft, Microsoft uitbreiden.	Uit per platform, ook (Business) Layer. Niet elk is een gelijkende versie. Alhoewel .NET kan een voordeel zijn.	Native graphics manipulatie, een Webview die wordt gebruikt in PhoneGap is hier ook beschikbaar.	Ingebouwd in de IDE van Xamarin (Xamarin Studio)
Native Community. Zelfs in platformen bestaan en lokale versies, veel uitbreiden.	Alhoewel apart per platform. Elk platform heeft een ontwikkelingsomgeving. C#, Objective C.	Zelfs als Xamarin.	Hoort meestal bij de IDE van de verschillende platformen.

Referenties:
 Mochte: <http://www.mochte.com/>
 jQuery: <http://jquery.com/>
 Jasmine: <http://jasmine.github.com/>
 Manie: <http://manie.com/>
 UnitJS: <http://unitjs.com/>

Tabel 3 Vergelijking ontwikkelplatformen

Uit tabel 3 is te lezen dat PhoneGap gelijk is aan een web oplossing. PhoneGap biedt meer mogelijkheden om de native API's aan te spreken dan web, maar voor de toepassing I2Vote lijkt de web oplossing voldoende functionaliteit te ondersteunen. De web applicatie bereikt over het algemeen de meeste platformen met de minste code. Daarbij komt dat een PhoneGap een wrapper is voor een web applicatie. Mocht uiteindelijk blijken dat er toch API's aangesproken moeten worden, kan de web applicatie eenvoudig in een PhoneGap applicatie gezet worden. Xamarin heeft als voordeel over native ontwikkeling dat voor alle platformen in .NET geschreven kan worden. Voor een enkele ontwikkelaar zou dit een groot voordeel zijn, maar wanneer elk platform haar eigen ontwikkelaar(s) heeft is dit voordeel minimaal. Er hoeft dan namelijk niet geschakeld te worden tussen bijvoorbeeld Java en Objective C. Als er maar één platform ondersteund moet worden dan valt de keuze al snel op native ontwikkeling. Moeten er meerdere platformen ondersteund worden, zouden Xamarin en PhoneGap een goede oplossing zijn. Wanneer er zoveel mogelijk platformen bereikt moeten worden met minimale middelen (één ontwikkelaar, tijd) dan zou een web applicatie met eventueel PhoneGap een goede oplossing zijn.

I2Vote valt op het moment van schrijven in de laatste categorie waarbij de web applicatie de meest geschikte oplossing lijkt.

5.2 AANBEVOLEN FRAMEWORKS EN BIBLIOTHEKEN

5.2.1 ANGULARJS

Als er een web applicatie gebouwd gaat worden kan er 'from scratch' begonnen worden, naar dat hoeft niet. Er zijn een aantal frameworks die een heleboel basis functionaliteit bieden. Bij basis functionaliteit kan gedacht worden aan navigatie onderdelen van een applicatie en life cycles⁷ van schermen. Daarnaast is HTML statisch, een Javascript framework kan de manipulatie van de DOM⁸ vereenvoudigen.

Voor het project I2Vote is gekozen voor AngularJS. Vanwege de betrokkenheid van Google, de functionaliteit, de duidelijke voorbeelden en de documentatie is AngularJS verkozen boven andere bewezen frameworks zoals BackboneJS en EmberJS.

AngularJS biedt de mogelijkheid om een Model-View-Controller (MVC) structuur aan te brengen in de applicatie. Daarnaast is AngularJS een opensource project van Google en wordt het gebruikt onder andere voor de Youtube applicatie⁹ op de PS3 van Sony en voor de website van Vevo¹⁰. Gezien grote bedrijven zoals Sony en Vevo gebruik maken van het systeem lijkt het erop dat de support voor AngularJS niet snel zal verdwijnen. Aan de activiteit op Github¹¹ is te zien dat er actief ontwikkeld wordt aan dit project. Daarmee is AngularJS momenteel relevant en lijkt dat de komende tijd te blijven.

Daarnaast maakt Google ook haar eigen webbrowser, zowel voor mobiel als voor besturingssystemen zoals Windows 8 en OS X. AngularJS wordt daar ook uitgebreid in getest. Helaas is de support voor Internet Explorer 8 met een recente versie verdwenen. Dat wil zeggen, de tests die AngularJS deed met Internet Explorer 8 worden niet meer gedraaid voor een release. Dit is geen probleem. De applicatie wordt in de eerste plaats ontwikkeld voor mobiele apparaten. PC's die nog gebruik maken van Internet Explorer kunnen eenvoudig een browser zoals Google Chrome

⁷ bijvoorbeeld op- en afbouw van een bepaald scherm

⁸ DOM is de document object map, ofwel, alle de HTML elementen

⁹ <http://us.playstation.com/youtube/>

¹⁰ <http://www.vevo.com>

¹¹ <https://github.com/angular/angular.js/commits/master>

installeren. De belangrijkste redenen blijven de uitgebreide documentatie en de verwachte blijvende ondersteuning.

5.2.2 KINETICJS

AngularJS doet veel voor de applicaties structuur, maar het is niet geschikt voor het implementeren van de voorgestelde aanwijs strategie. Daarom wordt de bibliotheek KineticJS ingezet om dit op te lossen. Voor het onderzoek naar aanwijs strategieën is succesvol geëxperimenteerd met KineticJS. Daarom is er opnieuw gekozen om deze bibliotheek in te zetten.

5.2.3 NODEJS EN EXPRESSJS

De back-end van de applicatie is erg belangrijk. In het geval van I2Vote moet deze aan een aantal eisen voldoen. In eerste instantie wordt er een mobiele web applicatie ontwikkeld, de web applicatie moet 'realtime' kunnen communiceren met de service doormiddel van polling of push berichten. Belangrijk is dat er ondersteuning is voor deze two-way communicatie, daarnaast moet er duidelijke documentatie zijn bij de back-end zodat er in een later stadium op doorgebouwd kan worden. De technologie van de front-end zal sneller veranderen dan de server technologie. Dat neemt niet weg dat de back-end ook met haar tijd mee moet. Er is in eerste instantie gekeken naar de huidige back-end van I2Vote. Die is geschreven voor het prototype dat is gebruikt sinds 2008. Echter, deze applicatie is volledig met inmiddels verouderde .NET technologie geschreven, daarnaast zijn de eisen aan de back-end gewijzigd en is er weinig documentatie bij de service.

In de oude situatie werden casussen volledig gedownload door de mobiele apparaten en daarop afgespeeld. In de nieuwe situatie zijn er mobiele web applicaties. Een web applicatie is per definitie afhankelijk van een netwerk/internet verbinding. De data, zoals afbeeldingen zullen dus niet opgeslagen worden op het mobiele apparaat, maar per keer geladen worden vanaf de service. Daarnaast moet er een mogelijkheid komen met de service voor periodieke updates, zoals het laden van een nieuwe vraag op initiatief van de professor.

Daarom is gekozen om de service te vernieuwen. Voor de vernieuwing is gekozen voor een relatief nieuwe technolo-

gie: NodeJS¹². NodeJS is een (server side) Javascript platform en biedt snelle asynchrone communicatie. Met name geschikt voor realtime web applicaties. Het is een relatief jong platform, maar desondanks wordt het in productie gebruikt door grote namen zoals PayPal¹³ en Walmart¹⁴.

Zowel NodeJS als AngularJS werken met Javascript. Dit betekent dat er tijdens het ontwikkelen niet geschakeld hoeft te worden tussen contexten (van bijvoorbeeld Objective C naar .NET, Java of Scala).

Daarnaast biedt NodeJS een uitgebreid package management systeem, Node package manager (NPM). Vergelijkbaar met de apt-get opdracht van Linux en RubyGems van Ruby On Rails. Omdat NodeJS en haar modules en packages open source zijn, is er een enorm aanbod aan packages en modules die vrij (wel even de documentatie van de packages erop nalezen) gebruikt kunnen worden¹⁵. Heeft de applicatie een ORM nodig dan is de kans groot dat iemand er al een keer één heeft gemaakt. Via NPM dan eenvoudig te gebruiken: 'npm install sequelize --save' gevolgd door 'npm install mysql'.

5.3 DATABASE

I2Vote heeft een database nodig. De vragen, afbeeldingen en college's moeten bijgehouden en opgeslagen worden. Als het gaat om database zijn er een aantal keuzes die gemaakt moeten worden. Ten eerste SQL of NOSQL, relationeel of niet. Op het gebied van relationele database zijn er de gebruikelijke spelers zoals MySQL en MsSQL. Daartegenover staan de niet relationele databases. Die zijn er in een aantal smaken. Zoals Redis, een in memory database. Heel erg snel. Redis schrijft eens in de zoveel tijd de wijzigingen weg naar de HDD (of SSD). Het zou daarbij dus kunnen gebeuren dat er data verloren gaat in het geval van een stroomuitval. Redis lijkt echter niet geschikt om relaties in te beschrijven en qua persisteren van data zijn er betere

¹² <http://nodejs.org>

¹³ <https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/07-03-2014>

¹⁴ [http://venturebeat.com/2012/01/24/why-walmart-is-using-node-js/07-03-2014](http://venturebeat.com/2012/01/24/why-walmart-is-using-node-js/)

¹⁵ <https://www.npmjs.org>

alternatieven zoals MySQL of MongoDB. Dan zijn er nog varianten zoals MongoDB. MongoDB is een document gestructureerde database. Dat wil zeggen dat documenten (of objecten) in haar volledigheid, bijvoorbeeld inclusief relaties, opgeslagen worden in één document. Dit sluit goed aan bij javascript omdat JSON (javascript objecten) rechtstreeks passen in het MongoDB model. Een I2Vote docent heeft een aantal vragenlijsten op zijn naam staan. In document gestructureerde databases zou in elke vragenlijst de gegevens van een docent opgeslagen staan. In een relationele database wordt er alleen een unieke identifier van de docent bij de vragenlijst geplaatst. In relationele databases worden tabellen en structuren van te voren aangegeven en wordt het aantal velden van een rij van te voren bepaald. Bij een NoSQL zoals MongoDB database worden dit soort constraints niet opgelegd en is het mogelijk om bijvoorbeeld tijdens het opslaan van een document een veld toe te voegen. Zou er in MongoDB in een later stadium een veld (kolom) toegevoegd worden aan een collectie (tabel met SQL), dan kan dat zonder dat alle andere documenten in de collectie aangepast hoeven te worden, net als dat er velden weggelaten kunnen worden. Met SQL kan een tabel niet uitgebreid worden bij een insert. Dit moet van te voren aangegeven worden en als een waarde weggelaten wordt maakt het nog wel deel uit van de rij (het veld wordt null of een andere standaard waarde).

```
questionList: {
  title: "...",
  user: {
    name: "...",
    email: "...",
    etc: "...",
  }
}
```

```
questionList: {
  title: "...",
  user: 1,
}
```

```
user: {
  id: 1,
  name: "...",
  email: "...",
  etc: "...",
}
```

Voorbeeld 1 Document structured versus relational

Het is mogelijk om relaties zoals in voorbeeld 1 een NoSql database zoals MongoDB te implementeren. De database is hier alleen zelf niet van op de hoogte en het koppelen en naleven van deze regels komen dan voor de rekening van de programmacode. Het nadeel van de regels en relaties naleven in de programmacode is, naast de extra programmacode die ervoor geschreven moet worden, dat het onnodig werk is. Een relationele database heeft deze functionaliteit meestal ingebouwd. Daarnaast ondersteunt MongoDB geen transacties. Bijvoorbeeld een vraag heeft antwoorden opgeslagen in een aparte collectie (tabel). Ten eerste moet hier de vraag verwijderd, maar ook de antwoorden die bij deze vraag horen. De antwoorden gelden immers alleen voor deze vraag. Nu gaat er na het verwijderen van de antwoorden tijdens het uitvoeren van de code iets mis voordat de vraag wordt verwijderd. De database zal nu niet automatisch het geheel terug draaien, het heeft immers geen kennis van de relatie tussen de twee collecties.

In een document gestructureerde database is het mogelijk om ook deze relaties aan te leggen, maar omdat er dan relaties aangelegd worden in een niet relationele database is er voor gekozen gebruik te gaan maken van een relationele database, namelijk MySQL. Dit lijkt de meest eenvoudige oplossing en bepaalde structuren van de oude database kunnen overgenomen worden. MongoDB schaal goed, alleen is dat in de eerste instantie met I2Vote niet op die schaal nodig. Het idee is om het niet onnodig complex te maken.

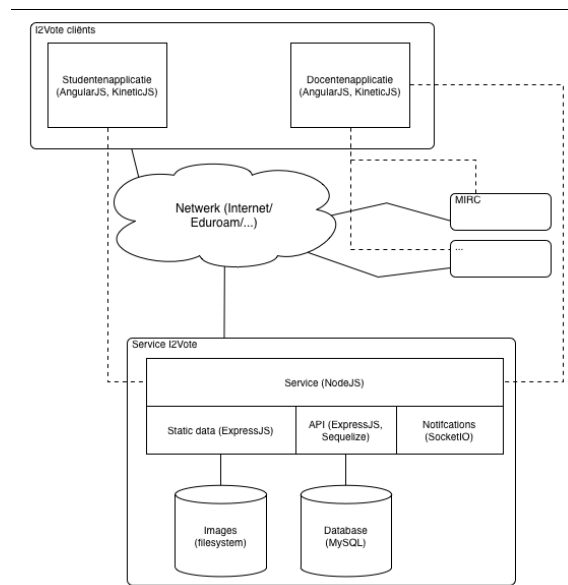
6 ARCHITECTUUR, FRAMEWORKS EN PLATFORMEN

6.1 INFRASTRUCTUUR EN INFORMATIE PADEN

Er zijn drie applicaties: de studentenapplicatie, de docentenapplicatie en de service-applicatie. Deze applicaties vormen samen I2Vote ook te zien in figuur 13. De docentenapplicatie is onder andere verantwoordelijk voor het aanmaken en het afspelen van vragenlijsten of casussen. De studentenapplicatie stelt de gebruiker in staat om de vragen uit een vragenlijst te beantwoorden. De docentenapplicatie bepaalt in deze welke vraag er beantwoord moet worden en de service voorziet beide partijen van informatie. Daarnaast biedt de service de mogelijkheid voor beide partijen om op de hoogte te blijven van een bepaalde sessie (het afspelen van een vragenlijst of casus). Het op de hoogte houden van beide applicaties gaat in tegenstelling tot polling gebruikt uit het prototype met push berichten. De cliënt applicaties (zowel de docent- als de studentenapplicatie) melden zich bij het deelnemen aan een college aan bij de service voor notificaties van de sessie. Op deze manier hoeven de cliënten pas actie te ondernemen als er een notificatie wordt gestuurd. Als de docent een sessie begint met een vragenlijst krijgt de student de mogelijkheid om hier aan deel te nemen via de studentenapplicatie. Deze schrijft zich in bij de service voor de sessie. Op het moment dat de docent een nieuwe stap (met een vraag) aanmaakt krijgt de studentenapplicatie een bericht: stap x is klaargezet voor sessie y. Verder krijgt de studentenapplicatie geen informatie. De studentenapplicatie doet vervolgens een verzoek bij de service voor de inhoud van deze vraag (vraag, vraagsoort, afbeeldingslocatie en eventueel mogelijke antwoorden). Op deze manier gaat het verkrijgen van de informatie altijd via de daarvoor bestemde routes. De authenticatie hoeft dan maar op één plek plaats te vinden, namelijk bij het ophalen van de informatie. De notificatie service kan eventueel bij aanvang een authenticatie doen, maar omdat er verder inhoudelijk geen informatie beschikbaar is over deelnemers, vragen en antwoorden via deze notificatie service is het niet noodzakelijk.

Omdat het I2Vote een image-based systeem is wordt er veelvuldig gebruik gemaakt van afbeeldingen. De afbeeldingen komen vooral uit de MIRC database. In de originele

opzet van I2Vote werd er op de afbeelding een lokatie gemarkeerd als het juiste antwoord. MIRC heeft deze informatie niet, daarom moet de docent bij het aanmaken van een image-based vraag dit markeren. Dit betekent dat de docentenapplicatie een koppeling krijgt met het MIRC systeem, daar kan een casus uit gekozen worden middels de MIRC API. Wanneer een casus is gekozen en daarbij een vraag is geformuleerd met een antwoord wordt de vraag inclusief MIRC casus via de I2Vote API opgeslagen in het I2Vote systeem. De I2Vote API slaat de vragen en antwoorden op in de MySQL database en de afbeelding op een aparte lokatie los van de database. De lokatie van de afbeelding wordt opgeslagen bij de vraag in de database.



Figuur 13 Architectuur overzicht I2Vote

De studentenapplicatie en de service zijn twee verschillende onderdelen, ieder met een eigen domein (bijvoorbeeld service.i2vote.edu en student.i2vote.edu). De studentenapplicatie wordt in een browser uitgevoerd en vraagt data op van de service. De Same-Origin Policy vormt hierbij een probleem.

6.1.1 SAME-ORIGIN POLICY

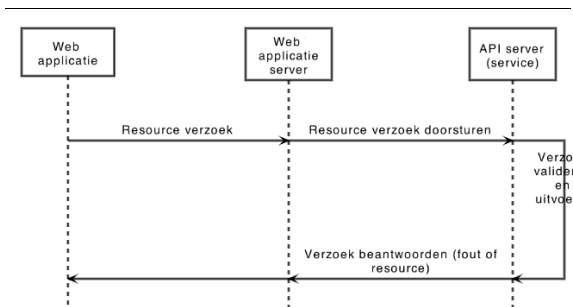
De Same-Origin policy is geïntroduceerd rond 1995 om te voorkomen dat er in een internet browser informatie wordt opgevraagd bij een domein anders dan het domein waar de pagina van is geladen. Als `\url{http://www.google.nl}` open staat en die pagina probeert gegevens op te vragen bij `\url{http://www.yahoo.com}` dan zal de browser dit tegenhouden.

6.1.2 DE OPLOSSINGEN

Voor dit probleem dat veel voorkomt bij web applicaties zijn een aantal oplossingen. De volgende paragraaf beschrijft de meest gebruikte oplossingen.

De meest voorkomende oplossingen zijn: CORS, JSONP en Proxy. Afhankelijk van de situatie kan er gekozen worden voor een van deze oplossingen.

De Proxy oplossing maakt op het domein van de web applicatie een interface of een link die naar de service wijst. Op deze manier worden alle resource verzoeken van de web applicatie gedaan op haar eigen server, op haar eigen domein. De server van de web applicatie stuurt het verzoek door naar service. Deze gegevens worden vervolgens weer terug gegeven aan de server die op haar beurt antwoord geeft op het verzoek van de web applicatie. Figuur 14 beschrijft de Proxy oplossing in een sequence diagram.



Figuur 14 Proxy strategie

De I2Vote implementatie maakt gebruik van push notificaties voor het aanbieden van nieuwe vragen, bij het eventueel invoeren van de Proxy strategie moet daar rekening

mee gehouden worden. Daarnaast moet met eventuele authenticatie via bijvoorbeeld HTTP Basic-Auth rekening worden gehouden.

Een andere oplossing is JSONP. JSONP ofwel 'JSON with Padding', maakt gebruik van Javascript mogelijkheid om dynamisch scripts in te laden van andere domeinen.

Web applicatie

```

<script type='text/javascript'>
  function JSONPcallback(data) {
    //Doe iets met de data
    alert(data.veld1 + data.veld2);
  }
</script>
  
```

...

```

<script type='text/javascript'
src='http://anderdomein/<resource>/<id>?callback=JSONPcallback'></script>
  
```

Service antwoord

```
JSONPcallback({veld1: "...", veld2: "..."})
```

Voorbeeld 2 De werking van JSONP

Voorbeeld 2 laat de JSONP strategie zien. De web applicatie doet middels het dynamisch laden van een script een verzoek bij de service. De `?callback=` parameter laat de service weten dat de aanvrager graag antwoord wil middels de JSONP strategie. Dit betekent dat de service een script genereert waarin de callback functie wordt uitgevoerd. Als parameters geeft de service de data waar om is verzocht. Bij deze strategie is het van belang dat de callback functie bestaat. Deze strategie heeft als voordeel dat het in oude browsers werkt. De mogelijkheid tot foutafhandeling is echter beperkt. Wanneer de web applicatie hier bijvoorbeeld een aanvraag doet voor een resource die niet bestaat, krijgt zij niets terug. Dit is eventueel op te vangen door handmatig een timeout te geven aan het verzoek, maar hierdoor wordt het ophalen van een resource gecompliceerder dan nodig lijkt. Een pakket zoals jQuery zou hierbij uitkomst kunnen bieden omdat daar de functionaliteit is ingebouwd.

Daarnaast biedt deze strategie de mogelijkheid voor de service om heel eenvoudig code uit te voeren binnen de web applicatie middels het opgevraagde script. Dit zou potentieel een veiligheidsprobleem kunnen zijn als de service niet in eigen beheer is. JSONP biedt alleen de mogelijkheid tot het aanroepen van HTTP GET resources. Dit betekent dat de web applicatie geen resource kan aanmaken op een REST service middels een POST request. Om deze strategie te implementeren moet de service daar rekening mee houden en kan zij in ieder geval geen REST protocol aanhouden.

CORS¹⁶, oftewel Cross Origin Resource Sharing, is in het leven geroepen om het Cross Domain Resource probleem op te lossen. Dit is de simpelste en meest elegante oplossing. De browser waarin de web applicatie wordt uitgevoerd zet bij het verzoek de origin header. De service leest deze uit en bepaalt of de web applicatie het verzoek mag doen vanaf het domein. Als het akkoord is, stelt de service een header in, waarmee wordt aangegeven dat het verzoek is toegestaan. Zo niet, dan krijgt de web applicatie een foutmelding terug. Het enige wat voor deze strategie geïmplementeerd moet worden is het zetten van de ACAO header uit voorbeeld 3 aan de server kant.

Request header van de cliënt

Origin: voorbeeld.domein.tld

Response header van de service

Access-Control-Allow-Origin: voorbeeld.domein.tld

Voorbeeld 3 De CORS HTTP header velden

Moderne browsers ondersteunen de nieuwe HTTP header waarin aangegeven kan worden dat het verzoek is toegestaan. Dit kan eventueel gekoppeld worden aan een domein. Deze strategie ondersteunt alle HTTP methodes en daarmee zou een REST service volledig benaderd kunnen worden, waardoor het protocol en de foutafhandeling eenvoudiger wordt. Er worden geen scripts dynamisch geladen waardoor de service geen code kan uitvoeren op de cliënt.

¹⁶ <http://www.w3.org/TR/cors/>

CORS werkt met behulp van nieuwe HTTP headers die niet door legacy browsers worden ondersteund. Dit betekent dat dit geen oplossing is voor oude browsers. Echter, sinds Android 2.1 en Internet Explorer 9 wordt CORS vrijwel op alle platformen ondersteund¹⁷. Gezien Android 2.2 (Froyo) volgens Google Dashboards¹⁸ al bijna niet meer wordt gebruikt (minder dan 2% van de Android gebruikers met Google Play¹⁹).

6.2 SERVICE MET NODEJS

Bij het ontwerpen van de service is zowel de interface als de achterliggende applicatie structuur belangrijk. De interface moet namelijk voor ontwikkelaars helder en consequent zijn opgebouwd met duidelijke regels zodat deze snel te begrijpen is. De achterliggende applicatie structuur moet eveneens helder zijn, wanneer er bijvoorbeeld uitbreidingen gedaan moeten worden is het belangrijk dat een ontwikkelaar zo snel mogelijk aan de gang kan met het schrijven van de uitbreiding.

De service is verantwoordelijk voor het bijhouden van de data voor de applicaties, de service is stateless (elke request heeft een vraag en een antwoord en daarna is het klaar, de server houdt bijvoorbeeld geen inlog status bij middels http sessies) en biedt de mogelijkheid aan cliënten om op de hoogte gehouden te worden van aanpassingen van onderdelen zoals het aanbieden van een nieuwe vraag.

Dat is samen te vatten in twee taken, namelijk: resource management en een notificatie service.

6.2.1 RESOURCE MANAGEMENT

Om het resource management probleem op te lossen is gekozen voor het REST protocol. REST is een bewezen en goed gedocumenteerd protocol met een duidelijke structuur. Vooral deze structuur is een doorslaggevend argument, het maakt het schrijven van de documentatie

¹⁷ <http://caniuse.com/cors>

¹⁸ Google haar statistieken bij het gebruik van de verschillende versies van Android [htt-
ps://developer.android.com/about/dashboards/index.html](https://developer.android.com/about/dashboards/index.html)

¹⁹ De app store voor Android

eenvoudiger en als een ontwikkelaar al bekend is met het REST protocol is het gebruik van de service relatief eenvoudig.

Service URI				
Resource	POST	GET	PUT	DELETE
/sessions	Create a session.	Get all sessions.	x	x
/sessions/1	x	Get a single session.	Edit a single session.	Delete a single session.
/sessions/1/steps	Create a new step.	Get all steps for this session.	x	x
/sessions/1/steps/1	x	Get a single step belonging to this session.	Edit a single step.	Delete a single step.
/steps/1/answers	Create a new answer.	Get all answers for this step.	x	x
/steps/1/answers/1	x	Get a single answer.	Edit a single answer.	Delete a single answer.
/catalogue	Create a new catalogue.	Get all catalogues.	x	x
/catalogue/1	x	Get a single catalogue.	Edit a single catalogue.	Delete a single catalogue.
/questions	Create a new question.	Get all questions.	x	x
/questions/1	x	Get a single question.	Edit a single question.	Delete a single question.
/questions/1/options	Create a new option.	Get all options belonging to this question.	x	x
/question/1/options/1	x	Get a single option.	Edit a single option.	Delete a single option.

Figuur 15 Service URI design

32

Omdat geen standaard is gevonden voor het opbouwen van de REST URI's zijn de volgende richtlijnen gedefinieerd:

- URI's zijn niet dieper dan 2 niveau's (/level/1/level/2).
- Als een entiteit op zichzelf kan bestaan dan krijgt zij haar eigen base URI.
- Wanneer een entiteit een directe relatie heeft met een andere entiteit volgen ze elkaar op in de URI.
- Sommige entiteiten hebben een relatie (afhankelijkheid van) met een andere maar kunnen ook op zichzelf bestaan of tot meerdere entiteiten behoren. Deze entiteit krijgt in dit geval een eigen base URI (/entiteit/1).

Dan is er nog de vraag, hoe er van een dergelijke REST service bij bijvoorbeeld een Question de antwoorden er bij worden gegeven zonder nog een tweede verzoek te doen bij de service. Daar komen queries van pas. Een query is een uitbreiding op het verzoek, hiermee wordt er bijvoorbeeld aangeven dat alle 'heeft één' of 'heeft meerdere' relaties ook in het resultaat worden verwerkt.

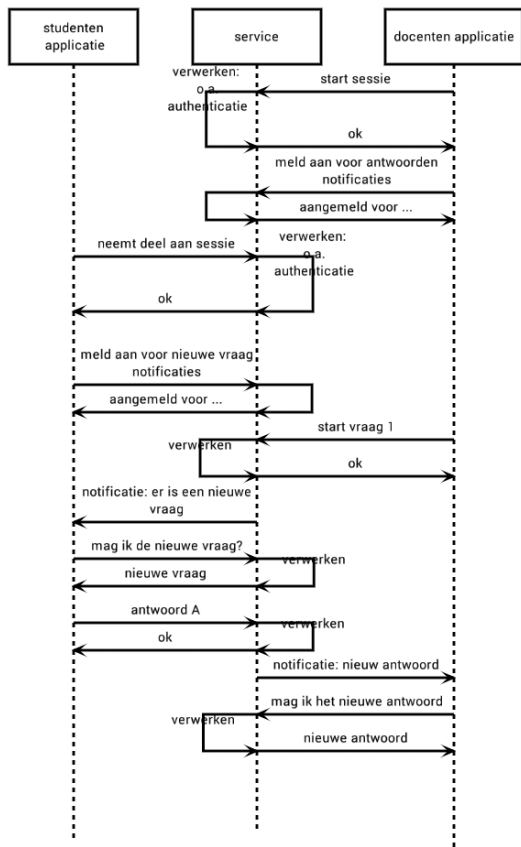
<http://api.i2vote.tld/questions/1?include=atalogues,options>

Voorbeeld 4 URI met query

In voorbeeld 4 wordt de vraag met het id 1 opgevraagd. Hieraan worden alle catalogues waar deze vraag in voorkomt en alle antwoorden die bij deze vraag horen toegevoegd.

6.2.2 NOTIFICATIE SERVICE

De notificatie service is verantwoordelijk voor het uitzenden van push notificaties. Aan de hand van een push notificatie kan een cliënt weer een resource opvragen. Er is bewust gekozen om de resource niet met de notificatie mee te sturen. Door de resources gescheiden te houden van de notificaties, raken de informatie stromen niet met elkaar in de knoop en hoeft bijvoorbeeld de authenticatie niet gedaan te worden op de notificatie service. De notificatie service heeft immers niet de mogelijkheid om gevoelige gegevens te versturen.



Figuur 16 Notificatie service sequence diagram

Tegenwoordig is het mogelijk om via de web browser een socket verbinding op te zetten, I2Vote maakt hier gebruik van. De service heeft naast de HTTP server een socket verbinding open staan. Hier kunnen cliënten mee verbinden en op aangeven aan welke sessie (college, lezing, presentatie) deel wordt genomen. Wanneer er voor een sessie door de docent een nieuwe vraag wordt gestart krijgen alle cliënten die zich hier op hebben aangemeld een notificatie met daarin het nummer van de nieuwe vraag. De cliënt kan nu van de server de nieuwe vraag ophalen. Ditzelfde principe wordt aangeboden voor docenten. Docenten kunnen zich aanmelden bij de notificatie service als docent bij een ses-

sie. De scheiding wordt aangebracht door de verschillende type notificatie cliënten onder te brengen in verschillende kanalen. Bijvoorbeeld voor een sessie met id 2, wordt het kanaal waar de vragen notificaties over worden gestuurd '2'. De antwoord notificaties worden dan gestuurd naar '2:answers'. Nu zou een studentenapplicatie zich aan kunnen melden voor het antwoorden kanaal. Door de scheiding schiet de applicatie daar echter weinig mee op, omdat alleen het nummer van het antwoord wordt mee gegeven. Er moet daarna nog steeds een verzoek gedaan worden om het volledige antwoord op te halen en daar zit weer de authenticatie tussen. In figuur 16 is dit scenario geschetst en voorbeeld 5 geeft hierbij de voorgestelde applicatie structuur.

6.2.3 APPLICATIE STRUCTUUR

```

/i2vote-service
  controllers
  models
    sequelize
  node_modules
  query-parser
  socket
  commands
  events
  tests
  sh
  
```

Voorbeeld 5 Service applicatie structuur

Wederom wordt hier gebruik gemaakt van models en controllers. De output van de service calls zou als view beschouwd kunnen worden. De folder controllers bevat alle controllers, de controllers zijn hier verantwoordelijk voor het afhandelen van de service calls. Neem bijvoorbeeld de service call /sessions?include=steps, deze wordt afgehandeld door de functie list van de controller sessions. De folder models bevat een folder sequelize. Dit kan eventueel gebruikt worden om scheiding aan te brengen in tussen de verschillende databases. Wanneer er bijvoorbeeld om performance redenen gekozen wordt om gebruik te gaan maken van Redis voor enkele models, kan daar apart een folder voor aangemaakt worden. De node_modules bevat alle npm modules zoals ExpressJS en SequelizeJS. De folder query-parser bevat de code voor het parsen van query's die gebruikt kunnen worden bij de verschillende service calls.

De notificatie commands en events staan in de socket folder. De verschillende service calls kunnen gebruik maken van de commands in de commands folder en verbonden cliënten kunnen gebruik maken van de events in de events folder.

Alle folders worden automatisch geladen, wanneer er een bestand new-step.js aan de commands folder toegevoegd wordt krijgen de verschillende service calls toegang krijgen tot de newStep() functie. De folder tests spreekt voor zich.

6.3 DE STUDENTENAPPLICATIE MET ANGULARJS

De studentenapplicatie wordt gerealiseerd als web applicatie, dit betekent dat er geen door de fabrikant voorgestelde richtlijnen zijn voor het opzetten van de applicatie. Dit stuk gaat over de opbouw van de studentenapplicatie met AngularJS. AngularJS biedt de mogelijkheid tot het implementeren van het MVC principe, vrij gangbaar tegenwoordig, maar omdat de hele applicatie in Javascript wordt geschreven is het wel anders dan met een traditionelere Object Oriented programmeertaal zoals Java voor Android of Objective C voor iOS. Om de ontwikkelomgeving in te richten voor de studentenapplicatie is gebruik gemaakt van een scaffolding applicatie, Yeoman. Yeoman heeft een aantal modules waaronder een voor AngularJS. Deze module suggereert een applicatie structuur waarop verder is gebouwd (paragraaf applicatie structuur}. Daarbij wordt er gebruik gemaakt van Grunt. Grunt is een Javascript Task Runner. Heel simpel, het voert een aantal van te voren gedefinieerde taken uit. Te vergelijken met bijvoorbeeld een ANT buildscript. Grunt maakt het mogelijk om alle unit tests uit te voeren en code te controleren op fouten alvorens de code te compileren. Nu wordt de Javascript niet echt gecompileerd, maar simpelweg samengevoegd in één bestand zonder spaties en line breaks. Hierdoor wordt het bestand kleiner wat weer gunstig is voor de snelheid waarmee de web applicatie laadt. Daarnaast maakt Grunt het mogelijk om de kwaliteit van de code te controleren. Een debugger voor Javascript. De engine die daarvoor is gebruikt is JSHint. JSHint detecteert potentiële typefouten in de codebase en controleert of de vooringestelde coding conventions worden gevolgd. Net als bij het bijvoorbeeld compileren van Java

of Objective C wordt de applicatie niet uitgevoerd als er dergelijke fouten inzitten.

6.3.1 APPLICATIE STRUCTUUR

```
/i2vote-student
  app
    bower_components
    fonts
    images
    scripts
      controllers
      directives
      models
      resources
      services
    styles
    views
  dist
    bower_components
    fonts
    images
    scripts
    styles
    views
  node_modules
  test
    spec
      controllers
      directives
      services
```

Voorbeeld 6 Studenten applicatie structuur

Voorbeeld 6 laat de voorgestelde applicatie structuur zien van de studentenapplicatie bij I2Vote. Zoals hierboven aangegeven is deze structuur tot stand gekomen met behulp van het scaffolding systeem Yeoman. De app folder bevat alle componenten van de applicatie. De dist folder bevat dezelfde componenten alleen dan door GruntJS gecontroleerd en opgeschoond. De test folder bevat de unit tests van de applicatie. De _components en _modules bevatten de components en modules voor het project die beheert worden door bower en npm respectievelijk. Waar het interessant wordt is de folder scripts. Hierin wordt de applicatie gedefinieerd. De folder controllers spreekt voor zich, hierin staan de controllers, de controllers koppelen de data uit een model in models aan een view uit views.

De directives folder is interessanter, hierin worden AngularJS directives geplaatst. Een directive beschrijft een DOM

element of attribuut. Hiermee kan bijvoorbeeld KineticJS worden geïmplementeerd. Dit kan in de view aangeduid worden met bijvoorbeeld `<kineticjs-element></kineticjs-element>` of `<div kineticjs-attribute></div>`.

De models in models zijn vanzelfsprekend. De models bevatten de data waarmee wordt gewerkt in de controllers en die wordt getoond in de views. De resources folder bevat de koppeling met de service laag. Elk model maakt gebruik van één resource of meerdere, de controller koppelt het model aan de view en bepaalt wat de view allemaal kan veranderen aan het model.

7 TECHNISCHE REALISATIE

Om aan te tonen dat het advies wat dit rapport op moet leveren gegrond is. Is er een gedeelte van I2Vote geïmplementeerd. Dit hoofdstuk beschrijft de implementatie en legt enkele keuzes die daarin zijn gemaakt uit.

7.1 BACK-END

De back-end is conform de keuzes uit de paragraaf applicatie structuur opgezet. Uit de use-cases zijn de belangrijkste elementen gehaald en geïmplementeerd. De notificatie service voor het deelnemen aan een college of sessie, een persistentie laag en het resource protocol (REST).

7.1.1 DE NOTIFICATIES

In de architectuur staat de notificatie service als een apart onderdeel beschreven. Tijdens de implementatie is de node module Socketjs gebruikt om de notificatie service te implementeren. SocketJS werkt zoals de naam al impliceert met sockets. Naast de HTTP server draait een sockets module die wacht op inkomende verbindingen. Deze worden op een asynchrone manier afgehandeld. Dus geen apart thread voor een nieuwe verbinding. SocketJS werkt op basis van events. Een event heeft een naam en een data object, zowel de cliënt als de server kunnen events versturen. Om te zorgen dat het ontvangen van de events en het versturen van de events op een overzichtelijke manier los van bijvoorbeeld de resource code komt te staan, worden events en commands dynamisch geladen middels het script uit voorbeeld 7. Deze code staat in het bestand index.js in een folder. Middels de functie require('.') van Nodejs wordt deze automatisch geladen. In de voorbeeld code staat de term module.exports, deze geeft aan de de functie die hierna volgt terug gegeven wordt op het moment dat deze code geladen wordt.

```
'use strict';

module.exports = function (io) {
  var commands = {};

  var fs = require('fs');
  var dir = fs.readdirSync(__dirname);
  var files = dir.filter(function (file) {
    return (file.indexOf('.') == 0) &&
    (file == 'index.js');
  });
  files.forEach(function (file) {
    var name = file.replace('.js', '');
    name = name.replace(/(-)/g, function
(x) {return x[1].toUpperCase(); });
    commands[name] = function (data) {
      var command = require('./' + file);
      return command(io, data);
    };
  });

  return commands;
};
```

Voorbeeld 7 Dynamisch laden socket commands

Het dynamisch laden gebeurt op deze manier voor de models, controllers, events en query's. De commands die geladen worden zijn heel eenvoudige korte stukken code die niet meer doen dan een object versturen met een bepaald event te zien in voorbeeld 8.

```
'use strict';

module.exports = function (io, step) {
  var event = 'newStep';
  var channel = step.sessionId;
  var message = {
    stepId: step.id
  };

  io.sockets.in(channel).emit(event, message);
};
```

Voorbeeld 8 Eenvoudig socket command

De voorgestelde architectuur spreekt over het aanmelden bij een sessie. Dit is geïmplementeerd middels dezelfde strategie als de commands. De commands worden echter

aan de hand van resource verzoeken en wijzigingen uitgevoerd en de events aan de hand van binnenkomende berichten op de sockets. De event `subscribeTo` wordt bijvoorbeeld uitgevoerd door een cliënt of docent om zich aan te melden bij een bepaald kanaal ('1:answers' of '1'). Er zitten hier in de huidige implementatie geen restricties op de aanmeldingen. Dat was ook niet nodig want de notificaties bevatten geen inhoudelijke informatie zoals antwoorden of namen.

De huidige implementatie voegt commands die aangeroepen worden aan de hand van resource verzoeken toe aan het HTTP request object via middleware²⁰. Op deze manier kan er eenvoudig vanuit elke resource een notificatie verstuurd worden. Dit zou eventueel selectief doorgegeven kunnen worden, maar hierdoor wordt de implementatie ingewikkelder. Dit is alleen nuttig als blijkt dat de huidige manier een (performance) bottleneck vormt.

7.1.2 DE PERSISTENTIE LAAG

Hier is gewerkt met `SequelizeJS`. `SequelizeJS` is een SQL object relational mapper voor `NodeJS` en `ExpressJS`. Hierin worden de entiteiten of objecten gedefinieerd en de relaties aangegeven. `SequelizeJS` zorgt er voor dat de database wordt aangemaakt. Daarnaast is `SequelizeJS` verantwoordelijk voor het ophalen en opslaan van gegevens. De models worden evenals de notificatie commands doormiddel van middleware toegevoegd aan de HTTP request zodat deze bij alle resource calls eenvoudig te benaderen is.

7.1.3 RESOURCES

`SequelizeJS` ondersteunt transacties. Er is geprobeerd het geheel te implementeren middels deze transacties. Echter de asynchrone aard van Javascript en `NodeJS` maken deze opzet een stuk ingewikkelder dan nodig lijkt. In een transactie worden meerdere database wijzigingen uitgevoerd. Als het ergens halverwege verkeerd gaat kan de transactie volledig terug gedraaid worden. Omdat elke database handeling via `sequelize` asynchroon is, kunnen de handelingen niet simpelweg achterelkaar uitgevoerd worden. Wanneer de ene handeling van de andere handeling afhankelijk is

moet daar op gewacht worden. Dit kan bijvoorbeeld gedaan worden door elke callback van een database handeling terug te geven aan een functie die een lijst bij houdt met alle handelingen die gedaan moeten worden. In deze lijst staat het resultaat van elke functie, op het moment dat een functie klaar is kan de eerst volgende functie die afhankelijk is van de zojuist uitgevoerde functie uitgevoerd worden. Op het moment dat er iets verkeerd gaat, kan dit binnen de transactie aangegeven worden als een fout. Het aangeven van een fout resulteert een een rollback. Als transacties nodig zijn lijkt dit een goede oplossing, maar het resulteert wel in veel code. Via npm kan er een package (`npm install async --save`) gebruikt worden die het geheel overzichtelijker maakt.

Het alternatief is om het eenvoudig te houden en de transacties (nog) niet te implementeren. Door de resource calls eenvoudig te houden zijn er minder database handelingen per call nodig. Transacties zijn daardoor niet nodig. Bijvoorbeeld wanneer een vraag wordt aangemaakt. In plaats van de antwoorden en de vraag in één keer aan te maken zijn er verschillende resource calls nodig. Mocht het zo zijn dat de cliënt tijdens het aanmaken van een vraag de vraag al wel heeft opgeslagen maar nog niet de antwoorden. Kan daar gewoon mee verder gegaan worden door alle vragen op te vragen zonder antwoorden en daar alsnog antwoorden aan toe te voegen. Alles in één keer, zou eventueel als er bijvoorbeeld performance problemen ontstaan door de eenvoudigere methode, geïmplementeerd kunnen worden.

De Same-Origin Policy wordt hier opgelost wederom door via middleware de juiste HTTP headers toe te voegen aan het antwoord van de service.

7.2 MOBILE FRONT-END

De front-end heeft twee verschillende eind gebruikers, namelijk de docent en de student. Deze implementatie heeft zich geconcentreerd op de studentenapplicatie. Omdat de implementatie hiervan nuttiger is bij het beantwoorden van de hoofdvraag. Deze applicaties komen namelijk terecht op de mobiele apparaten van de studenten.

²⁰ Een serie functies die uitgevoerd worden op een request object voordat de request bij de resource laag aan komt.

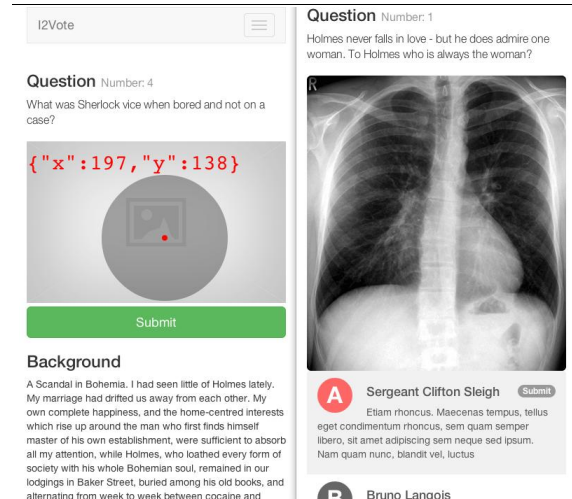
Voor de implementatie is zoals voorgesteld in de paragraaf applicatie structuur AngularJS gebruikt in combinatie met KineticJS voor de image-based vragen. Om te beginnen heeft de applicatie manier nodig om te schakelen tussen verschillende controllers en views. iOS heeft hiervoor bij-voorbeeld de navigation bar of een tabbar evenals Android. Deze componenten bevatten de functionaliteit om te schakelen tussen verschillende views en controllers. Met AngularJS gaat dat anders. AngularJS werkt met modules, modules bestaan weer uit onder andere services, factory's, directives en controllers. AngularJS heeft een aantal navigatie modules, deze implementatie gebruikt de `ui.router`²¹ module.

```
angular.module('i2voteMobileYoApp', [
  ...
  'ui.router'
])
.config(function ($urlRouterProvider,
  $stateProvider) {
  $urlRouterProvider.otherwise('/login');
  $stateProvider
    .state('login', {
      url : '/login',
      templateUrl : 'views/login.html',
      controller : 'LoginCtrl',
      authenticate: false
    })
    .state('search', {
      ...
    });
});
```

Voorbeeld 9 RouterUI voorbeeld

Voorbeeld 9 laat zien hoe dit in de praktijk werkt. De `router.ui` module werkt met states in plaats van urls en biedt de mogelijkheid om views en controllers te nesten in andere controllers. De veronderstelling was dat dit zou helpen bij het implementeren van het scherm dat gebruikt gaat worden om de vragen te beantwoorden te zien in figuur 17 echter werd het geheel er alleen maar ingewikkelder van. Daarom is het in de eerste instantie eenvoudig gehouden door het maar één niveau diep te houden. Dit werkt prima, maar de officiële router module van AngularJS zelf werkt als het op één niveau blijft vergelijkbaar en zou hier daarom ook gebruikt kunnen worden.

²¹ <https://github.com/angular-ui/ui-router>



Figuur 17 Links image-based, rechts multiple-choice

In plaats van het nesten van de verschillende views en controllers is bij de image-based vragen zoals voorgesteld in de architectuur gebruik gemaakt van AngularJS directives. De directive is in de huidige implementatie geschreven als een nieuw HTML element. Echter kan dit in de productie versie van de applicatie het beste als attribuut toegepast worden. De implementatie verandert hier minimaal door en het verbetert de ondersteuning voor oudere browsers. De image-based directive gebouwd met behulp van KineticJS is opgebouwd uit een view en een controller. Het model wordt bij gehouden via de data binding mogelijkheden van AngularJS. In figuur 17 zijn de coördinaten van de aanwijzer te zien, daar worden ze live bijgehouden in het model. In de productie versie van I2Vote is het aan te raden om dit niet te doen, dit heeft namelijk een behoorlijke impact op de performance. Een beter oplossing is om de coördinaten te te updaten op het moment dat de aanwijzer stil staat.

Om de resources op te halen van de service zijn er een tweetal strategieën geïmplementeerd. De eerste is de meest voor de hand liggende, het gebruik van een AngularJS service die per controller wordt meegegeven. Deze service heeft functies voor elke service call, voorbeeld 10 geeft een voorbeeld van. De functie `getSessions()` geeft een promise terug. Een promise heeft kan gebruikt worden om bijvoorbeeld foutafhandeling te implementeren. De foutaf-

handeling wordt benaderd door functies te chainen²² waarbij de er aangeknoopte functies de callback functies bevatten die worden uitgevoerd. Foutafhandeling gebeurt hier ook voornamelijk in de controller. In de controller wordt namelijk de promise opgehaald om daar vervolgens de callbacks aan vast te knopen. In deze callbacks wordt de data naar de view gestuurd.

```
return {
  getSessions: function() {
    ...
  },
  getOpenSteps: function(sessionId) {
    ...
  }
};
```

Voorbeeld 10 Service benadering enkele AngularJS service stijl

Alles qua service benadering zit zo op één plek. Is er een nieuwe niet bestaande service call nodig? Dan kan dit eenvoudig toegevoegd worden. De code in deze AngularJS service wordt er echter niet overzichtelijker van.

Daarom is er een tweede methode voorgesteld, deze gebruikt de Resource module van AngularJS. Deze module is speciaal voor services die het REST protocol implementeren. Voor deze methode wordt gebruik gemaakt van de voorgestelde applicatie structuur in de paragraaf applicatie structuur. Elk model maakt gebruik van één of meerdere resources. Een AngularJS resource heeft een aantal functies. Dit zijn functies voor bijvoorbeeld het aanmaken van een resource (POST /questions), ophalen van een lijst met resources (GET /questions) of een enkele resource (GET /questions/1).

Zo kan er in de controller tegen het model gezegd worden dat het de data moet gaan laden. Het model vraagt aan de resource die daar aan gekoppeld is of deze de data wil ophalen. De resource vult aan de hand van de gekregen data het model weer in. Door de data-binding van AngularJS wordt de view automatisch geupdate aan de hand van het model. Deze laatste mogelijkheid is meer in lijn met het

MVC principe met een extra laag, de Resource laag (of een Data Access Layer).

²² Het aan elkaar plakken van functies: `getSessions().success(function(result) {}).error(function(error) {})`

8 CONCLUSIE & AANBEVELINGEN

8.1 CONCLUSIE

In de probleemstelling is een vraag geformuleerd, namelijk of I2Vote in te zetten is in een BYOD omgeving. In de voorgaande hoofdstukken is er gekeken naar verschillende aanwysstrategieën en ontwikkelplatformen. Er is een implementatie gegeven met de voorgestelde aanwysstrategie, ontwikkelplatformen en frameworks van de core functionaliteit van I2Vote, namelijk het beantwoorden van image-based vragen.

Uit het onderzoek naar aanwysstrategieën blijkt dat Offset het meest aan alle gestelde eisen voldoet.

Aan de hand van het onderzoek naar de verschillende ontwikkelplatformen is voorgesteld om de applicatie als web applicatie te ontwikkelen, omdat hiermee de meeste platformen bereikt worden. Er is een reële verwachting dat alle functionaliteit hiermee gebouwd kan worden. Deze reële verwachting wordt bevestigd door de implementatie van de hoofdfunctionaliteit van I2Vote met de aangewezen technologieën.

Bij de implementatie zijn de voorgestelde frameworks gebruikt. Zo is AngularJS samen met KineticJS gebruikt om een studentenapplicatie implementeren. NodeJS is samen met ExpressJS, SequelizeJS en SocketIO gebruikt om de service mee te ontwikkelen. Daarbij is zowel voor de studentenapplicatie als de service applicatie de voorgestelde architectuur gebruikt uit de paragraaf applicatie structuur.

Aan de hand van deze implementatie, demonstratie en de resultaten van onderzoeken kan gesteld worden dat I2Vote zeker geschikt is voor een BYOD omgeving. Maar er moet nog veel gedaan worden, er is namelijk niet gekeken naar de docentenapplicatie. De service en studentenapplicatie zijn niet productie klaar. Denk hierbij aan zaken zoals de beveiliging en een integratie met het authenticatie systeem van instituut. Daarnaast is er in dit onderzoek niet gekeken naar zaken zoals de netwerk capaciteit van het instituut.

8.2 AANBEVELINGEN EN IDEEËN

Naast de resultaten van het onderzoek zijn er in het proces een aantal ideeën en aanbevelingen geformuleerd. Deze ideeën en aanbevelingen zijn gebaseerd op persoonlijke inzichten en meningen.

8.2.1 AUDIO-VISUEEL COMPONENT

I2Vote is in de eerste instantie een stem systeem. Maar het systeem zou met een kleine aanpassing ook in te zetten zijn om colleges op afstand te volgen. Door bijvoorbeeld tijdens het geven van het college tegelijkertijd een livestream uit te zenden met audio en de sheets van de powerpoint presentatie. Door dit te combineren met I2Vote zou er actief deelgenomen kunnen worden aan een college op afstand.

8.2.2 TOEKOMST

Tijdens de HBO opleiding Informatica wordt er bijna drie jaar lang vooral in project groepen gewerkt. Een afstudeeropdracht en een verdiepingstage worden veelal individueel uitgevoerd. Door dit contrast is de realisatie tot stand gekomen dat het werken in een groep, al is het maar met twee personen, veel voordelen heeft. Er kan informatie uitgewisseld worden en werk kan gecontroleerd worden door teamgenoten waardoor veel fouten voorkomen kunnen worden. Hierdoor kan er sneller gewerkt worden en een hogere kwaliteit geleverd worden.

Daarnaast wordt I2Vote een mobiele applicatie in een BYOD omgeving met een heleboel verschillende apparaten. Een nieuwe versie van een browser of besturingssysteem zou zomaar nieuwe fouten kunnen introduceren. Het lijkt dus aan te raden om hier altijd een team of ontwikkelaar voor te hebben om dit op te kunnen vangen.

8.3 HOE IS HET RESULTAAT ONTVANGEN

Het afstudeerproject is afgesloten met een presentatie voor de opdrachtgever en collega's. Tijdens de presentatie is een demo gegeven van de implementatie. Voor de demo is gebruik gemaakt van de mobiele apparaten van het publiek.

Het publiek kreeg een aantal image-based vragen te beantwoorden die vervolgens op het scherm getoond werden. Het aantal vragen is beperkt gehouden met het oog op de tijd, maar het publiek en de opdrachtgever reageerde positief en waren enthousiast.

9 BIBLIOGRAFIE

- Albinsson, P.-A., & Zhai, S. (2003). High precision touch screen interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 105–112).
- Amatya, S., & Kurti, A. (2014). Cross-Platform Mobile Development: Challenges and Opportunities. In *ICT Innovations 2013* (pp. 219–229). Springer.
- Benko, H., Wilson, A. D., & Baudisch, P. (2006). Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 1263–1272).
- Biolchini, J., Mian, P. G., Natali, A. C. C., & Travassos, G. H. (2005). Systematic review in software engineering. *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES, 679*(05).
- Charland, A., & Leroux, B. (2011). Mobile application development: web vs. native. *Communications of the ACM, 54*(5), 49–53.
- drs. Nynke Bos en drs. Nunke Kruidierink. (2012). Hoe gebruiken studenten ICT (in het onderwijs).
- Grossman, T., & Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 281–290).
- Hartmann, G., Stead, G., & DeGani, A. (2011). Cross-platform mobile development. *Mobile Learning Environment, Cambridge*.
- Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2013). Evaluating cross-platform development approaches for mobile applications. In *Web Information Systems and Technologies* (pp. 120–138). Springer.
- Hoyt, A., McNulty, J. A., Gruener, G., Chandrasekhar, A., Espiritu, B., Ensminger, D., ... Naheedy, R. (2010). An audience response system may influence student performance on anatomy examination questions. *Anatomical Sciences Education, 3*(6), 295–299.
- Mobile., V. (2012). Cross-Platform Developer Tools. Retrieved from <http://www.visionmobile.com/product/cross-platform-developer-tools-2012/>
- Outtier, B., Leuven, K. U., & Leuven, B. (n.d.). Vergelijkende studie van cross-platform tools voor de ontwikkeling van native mobiele applicaties.
- Potter, R. L., Weldon, L. J., & Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 27–32).
- Sears, A., & Shneiderman, B. (1991). High precision touchscreens: design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies, 34*(4), 593–613.
- Trice, A. (2012). PhoneGap advice on dealing with Apple application rejections. Retrieved from <http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm>
- Van Ooijen, P. M. A., Broekema, A., & Oudkerk, M. (2011). Design and implementation of I2Vote--An interactive image-based voting system using windows mobile devices. *International Journal of Medical Informatics, 80*(8), 562–569.
- Vogel, D., & Baudisch, P. (2007). Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 657–666).
- Wang, F., & Ren, X. (2009). Empirical evaluation for finger input properties in multi-touch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1063–1072).
- Xamarin. (2013). Xamarin Platform FAQ. Retrieved from <http://xamarin.com/faq#q3>

BIJLAGE I LITERATUURONDERZOEK BIJ AANWIJSSTRATEGIEËN

Dit is het literatuuronderzoek gedaan voor het onderzoek naar de meest geschikte aanwijsstrategie voor I2Vote 2.0. De referenties die hieronder worden gegeven hebben betrekking op de bibliografie uit het hoofddocument (de scriptie).

Hier worden een aantal technieken beschreven uit verschillende onderzoeken. De informatie is gericht op de precisie van het aanwijzen en de overstap van high-resolution hardware (de aanwijspen) naar de vinger (low-resolution).

De meest eenvoudige manier van aanwijzen is het direct aanwijzen, ofwel Land-On. Maar die strategie is helaas niet altijd even nauwkeurig.

Volgens sommigen is het occlusion-probleem één van de grootste belemmerende factoren bij het precies aanwijzen van een doel op een Touchscherm met de vinger (Vogel & Baudisch, 2007). Het occlusion-probleem is dat een gebruiker het zicht op het aanwijsdoel verliest doordat de vinger of hand in de weg zit. Er is ook wel gedacht dat de precisie te wensen over liet door het gebrek aan feedback (Potter, Weldon, & Shneiderman, 1988), onder de vinger ziet de gebruiker niet wat er precies gebeurt. Wanneer de afbeelding of het aanwijsdoel te klein is kan dit bij I2Vote een probleem zijn. Doordat je niet precies ziet waar je vinger terecht komt kan de cursor naast het voorgenomen gebied terecht komen. Een gevolg kan zijn dat het antwoord daarvoor fout is of dat de gebruiker het een paar keer moet proberen om de juiste lokatie te raken.

Een besproken strategie is Offset, of Take-Off (Potter et al., 1988). Offset of Take-Off tekent een cursor op een bepaalde afstand van het punt waar je het scherm aanraakt. Zolang je het scherm aanraakt wordt er geen muisklik of selectie doorgegeven. De gebruiker kan de cursor zo naar het doel bewegen en van het scherm afhalen om het punt te selecteren. In principe is dit een oplossing voor het occlusion-probleem. Door de constante terugkoppeling over de lokatie van de cursor en het zicht op het doel kan er preciezer een lokatie aangewezen worden. Maar de tijd die het

kost wordt groter in tegenstelling tot het direct aanwijzen, de cursor verschijnt immers pas na de aanraking. Een kritiek op deze strategie is dat je moeite zou kunnen hebben met doelen aanwijzen dichtbij de randen van het scherm, welke randen dit zijn is afhankelijk van de Offset richting. Offset zou kunnen werken in de I2Vote setting, maar voor het probleem aan de randen van de schermen zou een oplossing gevonden moeten worden. Misschien zou de gebruiker zelf de regie moeten krijgen over de richting van de

Offset zoals H. Benko met zijn Dual-Finger Midpoint strategie voorstelt (Benko, Wilson, & Baudisch, 2006). H. Benko houdt alleen, door zijn onderzoeksdoel, minder rekening met de kleine schermen waar dit onderzoek zich op richt. Deze strategie gebruikt het middelpunt tussen de twee vingers als aanwijslokatie. De afstand tussen de vingers bepaalt de Offset.

H. Benko beschrijft meerdere strategieën die twee vingers gebruiken. Zo is er een strategie waar de rechter vinger een cursor bestuurt (net als bij een muis) en de afstand tussen de linker vinger en de rechter vinger bepaalt hoe snel de cursor beweegt ten opzichte van de rechter vinger. Wanneer de vingers dicht bij elkaar staan staat de cursor vast, wanneer ze ver uit elkaar staan beweegt de cursor net zo snel als de vinger. Een nadeel hiervan is dat je veel schermruimte nodig hebt om dit uit te voeren.

Een andere strategie van H. Benko is de strategie waarbij de tweede vinger bepaalt of er gebruik wordt gemaakt van een vaste Offset. Een tweede vinger op het scherm schakelt de Offset in. De Dual-Finger methode is ook gecombineerd met een vergroot functie. De rechter vinger bepaalt direct de aanwijspositie en de linker vinger bepaalt of het gebied ontstaan vanuit de aanwijspositie vergroot (zoomen) moet worden en hoeveel.

Dat het succesvol raken van een doel eenvoudiger wordt wanneer je het vergroot is eerder beschreven. Zoom-Pointing beschreven door Pär-Anderson Albinsson en S. Zhai is een strategie waarbij er het gebied met het doel wordt geselecteerd door er een vierkant omheen te tekenen (Albinsson & Zhai, 2003). Het geselecteerde gebied

wordt uitvergroot waardoor het raakoppervlak van het doel groter wordt. De gebruiker kan blijven zoomen totdat het doel niet meer gemist kan worden. Een nadelige bijkomstigheid is dat de gebruiker door het zoomen geen totaaloverzicht meer heeft. Als je dit zou vertalen naar een moderne smartphone zou je het tekenen van een vierkant kunnen vervangen door het native zoomgebaar van het mobiele apparaat.

Een strategie die Take-Off combineert met Zoom-Pointing en een aantal van haar problemen oplost is Shift door D. Vogel (Vogel & Baudisch, 2007). Shift detecteert doelen in het gebied waar de gebruiker het scherm aanraakt. Als blijkt dat deze doelen door het occlusion-probleem niet meer te zien zijn (te klein zijn), wordt er een kopie van het gebied onder de vinger getekend op een plek waar het wel zichtbaar is. Deze plek wordt net als met de Take-Off (Offset) bij de vinger geplaatst en verplaatst zich zo dat de kopie van het beeld nooit buiten de randen van het scherm komt. Net als bij de Take-Off methode wordt de selectie gemaakt op het moment dat de vinger wordt opgetild. Om de precisie van deze strategie te vergroten, wordt het gekopieerde gebied vergroot. Zo houdt de gebruiker het totaal overzicht en mik je bij het aanwijzen, in tegenstelling tot de Take-Off strategie, wel direct op je doel. Uit een aantal proeven blijkt dat Shift prettig werkt en de voorkeur heeft boven een aantal alternatieven zoals Offset.

A. Sears en B. Shneiderman spreken in hun onderzoek ook over een Gravity strategie (Sears & Shneiderman, 1991). Hierbij zou het aanwijsdoel de cursor naar zich toe kunnen trekken. Neem bijvoorbeeld de kaart van Nederland waar een gebruiker de hoofdsteden van de provincies aan moet wijzen. Het iets naast Groningen wijzen maar wel in de buurt wijzen wordt geïnterpreteerd als Groningen selecteren met de Gravity strategie. Dit doet het door de cursor naar Groningen toe te trekken omdat dat het dichtstbijzijnde aanwijsdoel is. Dit zou gebruikt kunnen worden in het I2Vote systeem wanneer het gaat om dergelijke topografische vragen. Echter bij röntgenfoto's is deze strategie wellicht minder interessant. Het aantal potentiële doelen is hier, in tegenstelling tot het aantal hoofdsteden in Nederland, immers oneindig. Een strategie die hierop lijkt is de Bubble-cursor. Deze wordt beschreven door T. Grossman en Ravin Balakrishnan (Grossman & Balakrishnan, 2005) en

is gericht op het gebruik als cursor bij een computermuis, maar maakt gebruik van een vergelijkbaar idee en kan ook als inspiratie dienen bij een eventuele implementatie.

Volgens F. Wang en X. Ren mag de radius van het doel oppervlak niet kleiner zijn dan 5.76mm voor een cirkelvormig doel bij directe aanraking (Wang & Ren, 2009). Daarbij wordt er geadviseerd dat het aantal arm bewegingen, het veelvuldig tappen (klikken) en draaibewegingen maken met de pols vermeden moeten worden. Dit vanwege de vermoeidheid die daar bij optreedt. Ook zij benadrukken nogmaals dat het aanwijsdoel groot genoeg moet zijn. In het onderzoek van A. Sears en B. Shneiderman (Sears & Shneiderman, 1991) wordt het zoomen bij de Gravity strategie niet behandeld maar wel genoemd als een onderdeel dat verkend moet worden. Het lijkt erop dat het zoomen steeds weer terug komt en dat dit bij I2Vote, waar de afbeelding groot zijn (veel informatie bevatten) en de aanwijsdoelen potentiëel heel klein zijn, een onderdeel van de oplossing zou kunnen zijn.

BIJLAGE II LITERATUURONDERZOEK BIJ ONTWIKKELPLATFORMEN

Dit is het literatuuronderzoek gedaan voor het onderzoek naar de verschillende ontwikkelplatformen voor I2Vote 2.0. De referenties die hieronder worden gegeven hebben betrekking op de bibliografie uit het hoofddocument (de scriptie).

Om een goed advies te geven moet er eerst de nodige informatie verzameld worden. Omdat de technologie snel veranderd en de gegevens van onderzoeken over cross-platform development van vier jaar geleden bijvoorbeeld al niet meer relevant zijn, is het van belang om recente publicaties te nemen. Er is gekozen om publicaties vanaf 2011 en later te nemen.

Tijdens het verzamelen van informatie kom je een aantal verschillende platformen tegen, veel Open Source en een aantal betaald en gesloten. De opensource platformen lijken veelal geld te verdienen met service contracten en abonnementen. Een aantal opvallende platformen zijn: Xamarin, PhoneGap, Appcelerator Titanium Mobile, web (een mobiele website) en native (per platform een applicatie schrijven).

In onderzoek gedaan door Heitkötter (Heitkötter, Hanschke, & Majchrzak, 2013) wordt een vergelijking gemaakt tussen hybrid, web en native applicatie ontwikkeling. Er wordt gesteld dat het platform web applicaties een lage leercurve en ontwikkeltijd heeft en relatief eenvoudig te onderhouden is. Een hybride platform zoals PhoneGap heeft als grote voordeel boven een web applicatie dat het de hardware van de mobiele apparaten kan gebruiken en dat de performance beter lijkt. Daarnaast kan volgens het onderzoek de laadtijd van een web applicatie hoger liggen dan bij een hybride applicatie. Native applicaties werken het snelst, maar nemen de meeste ontwikkeltijd in beslag, daarbij lijkt het onderhoud van de applicaties ingewikkelder. Een web applicatie is op een later moment relatief eenvoudig om te zetten naar een PhoneGap applicatie om toegang te krijgen tot meer hardware van het mobiele apparaat. Wanneer men met native applicaties meerdere platformen wil ondersteunen betekent dit veelal dat de

applicatie volledig herschreven moet worden per platform. Het onderzoek van Heitkötter stelt dat wanneer applicaties gepubliceerd worden er rekening gehouden moet worden met de regels van de betreffende applicatie stores, zoals de App store van Apple of de Play store van Google.

Trice (Trice, 2012) haalt een belangrijk punt aan dat betrekking zou kunnen hebben op I2Vote, namelijk dat een web applicatie wanneer het met behulp van bijvoorbeeld. PhoneGap wordt gepubliceerd de applicatie niet mag aanvoelen als een website. Dit betekent dat zaken zoals het pinchen om te zoomen om content te kunnen bekijken niet mag. I2Vote zou de pinch zoom mogelijkheid van de browser wellicht kunnen gebruiken bij het aanwijzen op een afbeelding.

Open Source oplossingen zijn veelal gratis (Heitkötter et al., 2013) om te gebruiken. Daar bestaat het verdien model uit het leveren van bijvoorbeeld support abonnementen. Closed Source oplossingen bieden vaak een gratis versie aan waarmee ontwikkeld en getest kan worden, maar zodra er gepubliceerd moet worden is er een duurder abonnement nodig. Of zoals bij Xamarin: er kan gratis gebruik gemaakt worden van het systeem totdat de applicatie boven een bepaalde grootte uitkomt.

Volgens Charland (Charland & Leroux, 2011) komt hardware support steeds meer naar de mobiele browsers toe. Dit zou kunnen betekenen dat op den duur de voordelen van een hybrid zoals PhoneGap tegenover een web applicatie steeds kleiner worden. Waardoor PhoneGap uiteindelijk zou kunnen verdwijnen. Dit lijkt echter nog ver weg. Alhoewel zaken zoals het gebruiken van de lokatie bepaling en gyroscope data via veel moderne browsers te benaderen zijn²³.

Hartmann (Hartmann, Stead, & DeGani, 2011) beschrijft ontwikkelplatformen en frameworks geschikt voor mobile learning. Ze stellen dat het beste platform erg afhankelijk is van de functionaliteit van een applicatie. De web based aanpak (zowel hybrid als web applicaties) wordt veel be-

²³ <http://caniuse.com> 14-03-2014

sproken en krijgt in veel gevallen de voorkeur boven native applicaties. Een native applicatie zou volgens Hartmann beter zijn wanneer er veel zware berekeningen worden gedaan of wanneer de applicatie veel grafische rekenkracht eist. Outtier (Outtier, Leuven, & Leuven, n.d.) stelt dat Appcelerator Titanium langzamer is dan Xamarin. Xamarin is volgens eigen zeggen (Xamarin., 2013) net zo snel als een native applicatie.

Door Kurti (Amatya & Kurti, 2014) is een literatuuronderzoek uitgevoerd waarin ze aan de hand van een door Biolchini (Biolchini, Mian, Natali, & Travassos, 2005) geïnspireerde methode artikelen hebben verzameld en gekozen. Uit de resultaten blijkt dat web based applicaties in opkomst zijn en alleen op performance en hardware ondersteuning achterlopen op de native applicaties. Kurti zegt dat de hybrid platformen zoals PhoneGap niet alle API's van verschillende mobiele platformen (Android, iOS, Blackberry, Windows, etc) even goed ondersteunen. Daarbij stelt hij dat de slogan "code once, deploy everywhere" meer "code once, debug everywhere" wordt. Charland (Charland & Leroux, 2011) stelt dat er tussen web browsers op verschillende platformen ondanks dat ze dezelfde achterliggende technologieën gebruiken toch kleine verschillen zitten. Hierdoor lijkt het nog steeds wel noodzakelijk om met de verschillende browsers en platformen te testen. Hier lijkt een kern van waarheid in te zitten, Outtier zegt namelijk over Appcelerator dat het lastig was om de code overal aan de praat te krijgen en dat er veel verloren gaat aan het debuggen ("debug everywhere").

BIJLAGE III USE CASES (STUDENT)

Use Case: realtime college volgen

Actors:

- Student
- Studenten applicatie
- Servicelaag
- Docenten applicatie

Omschrijving: de student kiest ervoor om een realtime college te volgen. De code die hij van de docent krijgt wordt ingevoerd en het juiste college wordt door het systeem (studenten applicatie) getoond. Het systeem wacht totdat de er een bericht komt van de docent applicatie met de eerste vraag. De student krijgt de vraag te zien en heeft een bepaalde tijd om daar antwoord op te geven. Wanneer de student tevreden is met het antwoord of wanneer de tijd om is wordt de vraag stop gezet wordt het antwoord opgeslagen.

Use Case: gearchiveerd college volgen (voormalig offline)

Actors:

- Student
- Studenten applicatie
- Servicelaag

Omschrijving: de student kiest ervoor om een realtime college te volgen. De code die hij van de docent krijgt wordt ingevoerd en het juiste college wordt door het systeem (studenten applicatie) getoond. Het systeem begint met de eerste vraag. De vraag blijft actief totdat de student antwoord geeft of de tijdslimiet is verstreken. Vervolgens kan de student zelf aangeven wanneer de volgende vraag begint. De student kan zien hoeveel vragen er nog beantwoord moeten worden voordat het college voorbij is, daarnaast De antwoorden worden opgeslagen. De student kan gearchiveerde colleges oneindig vaak uitvoeren.

Verschillende vragen in een college

De eenvoudige multiple-choice vraag

Deze vraagsoort bestaat alleen uit tekst. Er is een vraag met daarbij een aantal antwoorden, waarvan er in principe on-

eindig kunnen zijn. De studenten applicatie krijgt het juiste antwoord niet direct.

Voorbeeld:

I can run but not walk. Wherever I go, thought follows close behind. What am I?

A: A sheep

B: A horse

C: A nose

De multiple-choice afbeeldingsvraag met een afbeelding
Deze vraagsoort bestaat uit een tekst en een afbeelding. De vraag bestaat uit een vraag en een afbeelding. De antwoorden zijn gebaseerd op de afbeelding. Antwoorden kunnen ook een afbeelding zijn.

Voorbeeld:

Welke is het juiste antwoord?

A: Plaatje 1

B: Plaatje 2

C: Plaatje 3

De image-based vraag

Deze vraag bestaat uit een text en een afbeelding. Een locatie op de afbeelding is in dit geval het antwoord.

Voorbeeld:

Waar ligt Groningen?

Use Case: resultaten bekijken

Actors:

- Student
- Studenten applicatie
- Servicelaag

Omschrijving: de student kiest ervoor om de resultaten van een college te bekijken. Het systeem geeft de student een lijst met colleges (met datum/tijd) waaruit gekozen kan worden. Na het kiezen van een college wordt de score per vraag weergegeven. Het juiste antwoord kan hierbij worden getoond. Daarnaast wordt de totaalscore weergegeven.

BIJLAGE IV USE CASES (DOCENT)

Use Case: aanmaken van een college

Actors:

- Docent
- Docent applicatie
- Servicelaag

Omschrijving: na het inloggen kan de docent er voor kiezen om een college aan te maken. Een college is een serie vragen. Na het kiezen voor het aanmaken van een college kiest de docent een naam voor het college. Vervolgens worden er vragen aangemaakt of geselecteerd uit eerder gestelde vragen. Dit kunnen image-based vragen zijn maar ook gewone multiple-choice vragen. De volgorde van de vragen kan worden aangepast. Na het aanmaken van het college wordt het college met de bijbehorende vragen opgeslagen zodat de docent deze later kan opstarten.

Use Case: image-based vraag toevoegen

Actors:

- Docent
- Docent applicatie
- Servicelaag

Omschrijving: de docent kiest bij het aanmaken van een college voor het toevoegen van een image-based vraag. De docent kan zelf een afbeelding kiezen van zijn computer of kiezen uit een database met eerder gebruikte afbeeldingen. Na het kiezen van de afbeelding wordt er een vraag ingevoerd bij de afbeelding en wordt met behulp van een teken tool op de afbeelding de lokatie van het goede antwoord aangegeven. De docent kan een tijdslimiet op de vraag instellen. Na de invoer wordt de de vraag opgeslagen en wordt deze toegevoegd aan het college waar deze bij is aangemaakt.

Use Case: multiple-choice vraag toevoegen

Actors:

- Docent
- Docent applicatie
- Servicelaag

Omschrijving: de docent kiest bij het aanmaken van een college voor het toevoegen van een multiple-choice vraag. De docent formuleert een vraag en noteert daarbij de antwoorden. Vervolgens wordt het juiste antwoord geselecteerd. Hierna wordt de vraag opgeslagen en toegevoegd aan het college. Optioneel zou bij deze vraag een afbeelding toegevoegd kunnen worden en een tijdslimiet ingesteld kunnen worden. De docent kan een tijdslimiet op de vraag instellen.

Use Case: college geven

Actors:

- Docent
- Docent applicatie
- Studenten applicatie
- Servicelaag

Omschrijving: nadat een college is aangemaakt en is opgeslagen, kiest de docent voor het geven van een college. Het college wordt opgestart. Het systeem (de docent applicatie) geeft de unieke code van het college, deze kan gebruikt worden om via een studenten applicatie deel te nemen aan het college. De docent kan zien hoeveel deelnemers er zijn en als er voldoende zijn, kan hij de eerste vraag starten. De vraag wordt naar alle deelnemers gestuurd en getoond in de docent applicatie. De docent of de tijdslimiet bepaald wanneer de vraag wordt gestopt of overgeslagen, om de docent daarbij te helpen wordt er een teller getoond waaraan te zien is hoeveel mensen hebben geantwoord. Als de vraag is gestopt, kan de docent er voor kiezen om alle gegeven antwoorden op de afbeelding te tonen. Wanneer de docent klaar is om de volgende vraag te starten wordt deze weer naar alle deelnemers gestuurd. Dit proces herhaalt zich totdat de docent het college vroegtijdig stopt of totdat alle vragen zijn beantwoord. Het college kan dan gearchiveerd worden. Daarna kan de docent weer een nieuw college starten.

BIJLAGE V AUTHENTICATIE SUGGESTIES

- SAML 2.0 <https://npmjs.org/package/passport-saml>
- <https://github.com/bozzltron/express-saml/blob/master/README.md>