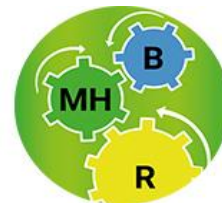


Ontwikkelen van een digitale toepassing voor Bureau MHR

Afstudeerverslag
31 mei 2019

Ilyaas Dhonre
20036095





De Haagse Hogeschool
Faculteit IT & Design
Informatica
Johanna Westerdijkplein 75
2521 EN Den Haag

Afstudeerder
Iljaas Dhonre

Afstudeeropdracht
Mobile Application Development en koppeling met
het registratiesysteem
31 mei 2019

Bureau Maatschappelijk Herstel en Rehabilitatie
Steenwijklaan 135
2541 RK Den Haag

Contactpersoon Bureau MHR
Karima Daoudi
Directrice Bureau MHR

Begeleider Bureau MHR
Mohamed Lamaiz
Software Engineer

Begeleiding De Haagse Hogeschool
Begeleidend examiner
Maarten Thissen

Expert examiner
Cobie van der Hoek

Referaat

Bureau MHR houdt zich bezig met het rehabiliteren van ex-gedetineerde jongeren bij hun herintreding in de maatschappij. Dit doen zij met behulp van een bewezen methodiek die zij zelf hebben ontwikkeld. Deze methodiek heet Forsa! en staat voor kracht en kans. Dit is wat Bureau MHR ex-gedetineerde jongeren wil bieden, nog een kans.

Net als vele andere bedrijven wil ook Bureau MHR meegaan met de veranderende technologie en zijn zij op zoek naar een manier om hun businessprocessen te versterken door middel van innovatie. Op deze wijze zijn zij op het idee gekomen om hun cliënten te motiveren en werknemers te ondersteunen met behulp van een mobiele applicatie. Deze applicatie moet Bureau MHR in staat stellen om de jongeren nog beter te bereiken en te activeren om een succesvolle herintreding in de samenleving te bewerkstelligen.

De hedendaagse trend laat zien dat steeds meer bedrijven innoveren en nieuwe technologieën aangrijpen voor het verbeteren van hun diensten.

Voorwoord

Als student van de opleiding Informatica kan ik met opgeheven hoofd terugkijken naar mijn studietijd. Waar ik in het eerste jaar nog geen stukje code kon schrijven heb ik voor mijn afstudeeropdracht een mobiele app gebouwd. Naarmate ik meer kennis ga toepassen in de praktijk des te meer ik erachter kom hoeveel ik in de afgelopen jaren geleerd heb. Als afsluiting van deze studietijd mocht ik mijn afstudeeropdracht uitvoeren bij Bureau MHR. Het was mij ter ore gekomen dat zij een mobiele applicatie wilden ontwikkelen en hier een afstudeerder voor zochten. Het dienstverlenende karakter en de hoge mate van maatschappelijke betrokkenheid spraken mij direct aan. Ik heb dan ook met veel enthousiasme aan deze opdracht gewerkt.

In dit verslag wil ik jullie meenemen in het proces dat ik heb doorlopen tijdens het uitvoeren van deze opdracht. Ik hoop dat jullie veel leesplezier zullen beleven.

Tot slot wil ik graag mijn dank betuigen aan de mensen die mij tijdens dit proces heel erg hebben gesteund. Allereerst mijn vrouw Mariam voor haar geduld en steun, mijn zoontje Dienn die mij regelmatig succes wenste, Karima Daoudi die in mij geloofde en mij deze kans heeft gegeven, mijn begeleiders Maarten Thissen en Coby van der Hoek en alle anderen die aan mij hebben gedacht tijdens dit proces.

Iljaas Dhonre
Rijswijk, 31 mei '19

Inhoudsopgave

REFERAAT	2
VOORWOORD	3
INLEIDING	5
DE ORGANISATIE	5
DE OPDRACHT	6
PROBLEEMSTELLING	6
DOELSTELLING	6
BEGINSITUATIE	7
AANPAK	7
ANALYSEFASE	8
ONTWERPFASE	9
ONTWIKKELFASE	9
TESTFASE	9
UITVOERING	10
ANALYSEFASE	10
<i>Brainstormsessie</i>	10
<i>Afnemen van interviews met stakeholders</i>	17
OPSTELLEN VAN MOCKUPS VOOR HET REGISTRATIESYSTEEM	18
<i>Opstellen van een context diagram en use case diagram</i>	19
<i>Opstellen van globale, geprioriteerde en gespecificeerde requirements</i>	20
<i>Use case beschrijvingen en requirements traceabilitymatrix</i>	21
<i>Validatie</i>	25
ONTWERPFASE	26
<i>Klassendiagrammen</i>	26
<i>Sequentiediagrammen</i>	30
ONTWIKKELFASE	36
<i>Mobiele applicatie</i>	36
<i>API</i>	37
<i>Systeemdeel Taak</i>	39
TESTFASE	39
AFWIJINGEN TEN OPZICHTE VAN HET AFSTUDEERPLAN	42
WIJZE VAN AANTONEN VAN DE COMPETENTIES/BEROEPSTAKEN	42
1.4 UITVOEREN ANALYSE DOOR DEFINITIE VAN REQUIREMENTS	42
2.1 OPSTELLEN GEGEVENSMODEL VOOR EEN DATABASE	42
2.2 ONTWERPEN, BOUWEN EN BEVRAGEN VAN EEN DATABASE	42
3.2 ONTWERPEN SYSTEEMDEEL	43
3.3 BOUWEN APPLICATIE	43
3.5 UITVOEREN VAN EN RAPPORTEREN OVER HET TESTPROCES	43
BRONNEN/LITERATUURLIJST	44
DEFINITIES	45

Inleiding

Dit afstudeerverslag is een uiteenzetting van het proces dat doorlopen is bij het uitvoeren van de afstudeeropdracht door de student. De opdracht omhelst het ontwikkelen van een mobiele applicatie voor het ondersteunen van werknemers en cliënten van de organisatie.

Allereerst wordt wat achtergrondinformatie beschreven met betrekking tot de organisatie, de beginsituatie en positie van de afstudeerder binnen de organisatie.

Vervolgens wordt uitgelicht wat de opdracht inhoudt en hoe deze is opgepakt. Daarnaast wordt kort beschreven welke mogelijkheden er zijn gevonden en voor welke aanpak is gekozen.

Daarna wordt toegelicht welke werkzaamheden zijn uitgevoerd, tegen welke problemen de student is aangelopen en de gemaakte keuzes hierbij. Afwijkingen die bij het uitvoeren van deze werkzaamheden ten opzichte van het afstudeerplan aan het licht zijn gekomen worden eveneens benoemd.

Tot slot worden de gekozen beroepstaken benoemd en de wijze waarop de competenties aan bod zijn gekomen bij het uitvoeren van de opdracht.

De organisatie

Bureau Maatschappelijk Herstel en Rehabilitatie (MHR) werkt vanuit het hart van aandachtwijken aan een effectieve rehabilitatie en re-integratie van ex-gedetineerde jongeren van niet-Westerse afkomst van 17 tot 27 jaar. Het programma waaronder dit gebeurt heet Forsa! Kenmerkend aan het programma Forsa! is dat ook de directe omgeving van de jongeren hierin wordt betrokken. Daarnaast initieert Bureau MHR herstelgesprekken met de direct betrokkenen en met de slachtoffers. Met deze aanpak leveren ze een succesvolle bijdrage aan het oplossen van maatschappelijke problemen als overlast en recidive. Aan Forsa! zijn enkele ondersteunende projecten gekoppeld, waarin betrokkenen met een niet-Nederlandse achtergrond worden opgeleid tot vertrouwenspersoon en 'maatjes'. Dat maakt het mogelijk dat maatjes uit de eigen cultuur de jongeren concreet begeleiden naar en tijdens een opleiding, stage of werkplek.

De aanpak die gebruikt wordt bij het ondersteunen van ex-gedetineerde jongeren bij hun re-integratie in de samenleving gebeurt aan de hand van een in-house methodiek ontwikkeld door Bureau MHR. De Forsa! methodiek. Hierbij staan negen leefgebieden centraal voor de re-integratie van de jongeren. Deze leefgebieden zijn:

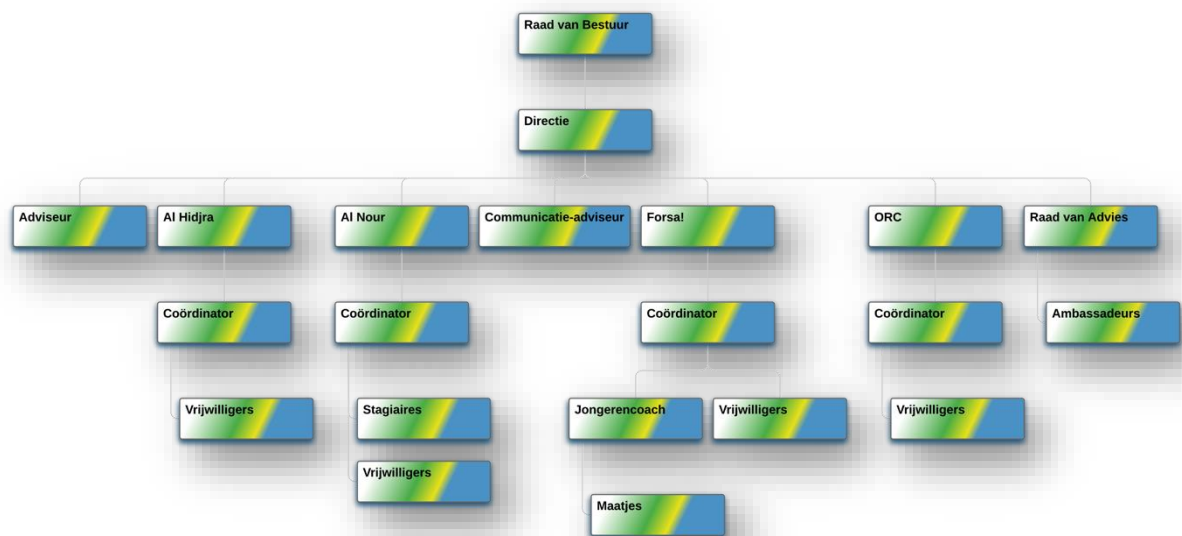
- Algemeen
- Wonen
- Werken
- Opleiding
- Financiën
- Gezondheid/welzijn
- Vrijtijdsbesteding
- Herstel netwerk
- Herstel slachtoffer

Een belangrijk kenmerk bij de Forsa! methodiek is dat deze begint in detentie. De gedetineerde jongere kan zich aanmelden voor het programma via de casemanager. Drie tot zes maanden voor vrijlating wordt de cliënt benaderd door een jongerecoach en zijn/haar toekomstige maatje. Tijdens dit eerste intakegesprek kan de jongere bepalen of hij/zij door wil gaan met het programma. Als het programma wordt voortgezet wordt bij een volgend gesprek een plan van aanpak opgesteld waarin de bovenstaande leefgebieden aan bod komen. De jongere en het maatje gaan samen kijken aan welke leefgebieden gewerkt moet worden om een succesvolle herintreding te maken in de maatschappij. In de meeste gevallen hoeft niet aan alle 9 leefgebieden gewerkt te worden. Deze afstudeeropdracht sluit aan op het plan van aanpak en dient als verlengstuk hiervan. Verderop wordt duidelijk wat hiermee bedoeld wordt. Het traject dat een jongere doorloopt duurt in totaal ongeveer anderhalf jaar.

Naast Forsa! zijn er ook programma's voor de naaste familieleden. Moeders en zussen kunnen deelnemen aan het empowermentprogramma Al Nour. Voor de vaders en broers is Horizon ontwikkeld. Zo krijgt de familie handvatten aangereikt, waarmee hun zoon/dochter/broer/zus na detentie beter kan worden opgevangen en ondersteund kan worden in het rehabilitatieproces. De recidivekans wordt hierdoor een stuk kleiner.

Van groot belang is ook het programma PreJob dat zich richt op preventie van criminaliteit onder jongeren door aansluiting te vinden op onderwijs en het bedrijfsleven.

Bureau MHR heeft een platte structuur met korte lijnen. Het hoofd van de stichting is de directrice Karima Daoudi. Onder haar werken de coördinatoren van de verschillende programma's die zich bezighouden met de dagelijkse gang van zaken. Bij Forsa! zijn er jongerecoaches die elk een team van maatjes begeleiden en ondersteunen. De maatjes zijn vrijwilligers of studenten die hun stage- of afstudeeropdracht afronden met het begeleiden en ondersteunen van ex-gedetineerde jongeren.



Figuur 1: Organogram van Bureau MHR

De opdracht

In samenwerking met ZonMw is een project opgestart voor het ondersteunen van ex-gedetineerde jongeren met een licht verstandelijke beperking. Een onderdeel van dit project is het toepassen van de Forsa! methodiek bij de herintreding van deze jongeren.

Probleemstelling

Een van de vraagstukken bij dit project is hoe de organisatie de jongere betrokken kan houden bij zijn/haar herintreding in de maatschappij. Met name in de eerste zes weken na vrijlating. Deze periode is van cruciaal belang, omdat een jongere in deze periode geen recht heeft op inkomen vanuit de overheid.

Doelstelling

Het betrokken houden van de jongere wil de organisatie bereiken door de jongere beter te ondersteunen, te motiveren en belonen. De organisatie wil dit bereiken middels een mobiele applicatie, omdat bijna elke jongere de beschikking heeft tot een smartphone.

Het doel van deze opdracht is het bouwen van een prototype van de mobiele applicatie met een koppeling met het bestaande registratiesysteem. Het gaat hierbij om een eenmalige opdracht onafhankelijk uitgevoerd door de afstudeerder. Hiervoor krijgt de afstudeerder een maandelijkse vergoeding.

Bureau MHR maakt regelmatig gebruik van de diensten van een software engineer die ingehuurd wordt als begeleider voor de afstudeerder.

Beginsituatie

Bureau MHR heeft reeds bij De Haagse Hogeschool een opdracht uitgezet voor het ontwerpen van het uiterlijk van de applicatie. Een groep CMD-studenten hebben dit als project opgepakt en uitgewerkt. Het gaat hierbij om een HIFI-prototype waarvan het ontwerp als basis is gebruikt bij het ontwikkelen van de mobiele applicatie.

Bij aanvang van de opdracht was er behalve het HIFI-prototype weinig bekend over de implementatie. Er waren geen requirements en ook de manier waarop de applicatie het proces en de gebruikers moet ondersteunen was nog niet helder. Het moest een mobiele applicatie worden welke door zowel Android als iOS systemen ondersteund wordt. De organisatie heeft geen IT-afdeling en er is bij aanvang geen technische documentatie beschikbaar.

Voor het achterhalen en helder krijgen van de wensen vanuit de organisatie en stakeholders is er in december 2018 door Bureau MHR een Google sprint georganiseerd. Hierbij heeft een multidisciplinair team een advies uitgebracht aan de opdrachtgever. De rol van de afstudeerder was vanuit een ontwikkelaarsperspectief kijken naar het probleem en adviseren. Er is uiteindelijk gekozen voor het ontwikkelen van een native mobiele applicatie met een koppeling met het registratiesysteem.

Voor het ontwikkelen van de mobiele applicatie is er gekozen voor Xamarin.forms als omgeving. Xamarin.forms maakt het mogelijk om een applicatie te bouwen welke op zowel Android als iOS ondersteund wordt. Er is echter weinig bekend over deze ontwikkelomgeving en de afstudeerder moet hiervoor veel vooronderzoek doen.

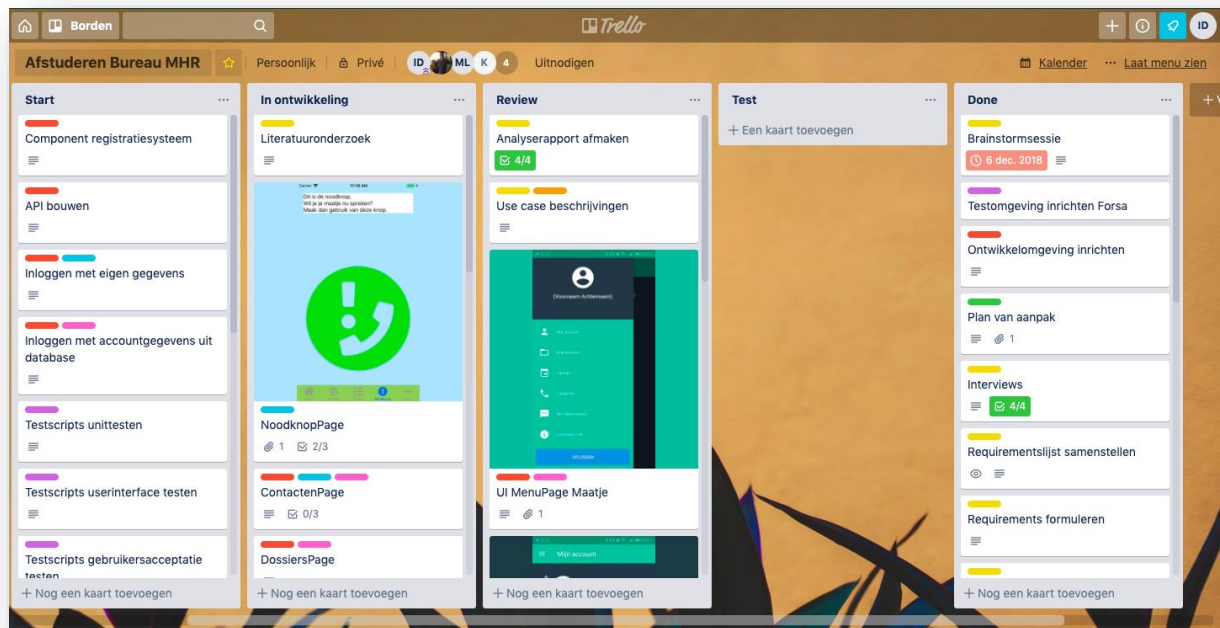
Aanpak

Voor het ontwikkelen van de applicatie zijn er geen restricties vanuit de organisatie wat betreft de aanpak en ontwikkelmethode. De afstudeerder krijgt hierin volledige vrijheid.

Voor aanvang van deze opdracht bij het opstellen van het afstudeerplan is er gekozen voor de KanBan methode als aanpak. Nadat naar voren is gekomen dat er weinig informatie beschikbaar is vanuit de organisatie, is er voor de watervalmethode gekozen. Deze methode zorgt voor een geordende structuur van het gehele proces. Het gehele proces is hierdoor opgedeeld in de fases:

- De analysefase
- De ontwerpfase
- De ontwikkelfase (implementatie)
- De testfase

Aan het eind van elke fase wordt een tussenproduct of deliverable opgeleverd. Hierbij gaat het om een analyserapport, een ontwerprapport en een testrapport. Voor elke stap binnen een fase worden de onderdelen gereviewd en eventueel aangepast. De productiefase/onderhoud is voor deze opdracht buiten scope gelaten. In combinatie met de watervalmethode is het planbord van de KanBan methode gebruikt voor overzichtelijkheid in de uit te voeren taken en te ontwerpen onderdelen van het systeem (zie figuur 2).



Figuur 2: KanBan planbord

De watervalmethode zorgt ervoor dat elke fase doorlopen en gereviseerd moet worden waarbij de requirements worden achterhaald, systeemontwerpen op basis van de requirements worden ontworpen en het gebouwde systeem vervolgens getest wordt.

Een agile methode is voor deze opdracht niet gekozen, omdat er eerder behoefte is aan een goed product dan aan korte iteraties met snel resultaat. Met het doorlopen en reviewen van elke fase kan dit bereikt worden. Tevens heeft de opdrachtgever voor verdere ontwikkeling van het systeem in de toekomst de beschikking tot de opgestelde tussenproducten als documentatie.

Analysefase

Om een duidelijk beeld te krijgen van de wensen hebben de opdrachtgever mevrouw Daoudi en projectleider mevrouw Schenk een brainstormsessie (Google sprint) georganiseerd met een multidisciplinair team. Tijdens de brainstormsessie zijn zaken zoals het beoogde doel, de customer journey, de mogelijke oplossingen en de te nemen vervolgstappen verder uitgelicht.

Er zijn twee oplossingen voortgekomen uit de brainstormsessie. Deze zijn een responsieve webapplicatie en een native applicatie. Na een afweging van voor- en nadelen van beide oplossingen is er gekozen voor het ontwikkelen van een native applicatie. Dit in verband met offline beschikbaarheid voor bepaalde gebruikers. Een Google sprint bestaat uit meerdere onderdelen verspreid over meerdere dagen of weken. Het doel van zo'n sprint is het vervaardigen van een product van begin tot eind. Vanwege het gebrek aan tijd en middelen zijn de belangrijkste onderdelen van een Google sprint in een brainstormsessie van één dag gestopt. Doordat de brainstormsessie niet voldoende informatie heeft opgeleverd met betrekking tot de implementatie van de applicatie en het gebruik ervan is er meer informatie opgehaald door de afstudeerder in de vorm van halfopen interviews. Deze interviews zijn afgenomen met de stakeholders.

Om na te gaan of de resultaten van de analysefase aansluiten op de wensen en behoeften van de business, hebben validatiemomenten plaatsgevonden. Bij deze opdracht zijn reviews en doorspreken gebruikt als validatietechnieken, omdat er nauw contact is met de vertegenwoordigers. Inspecteren is niet gebruikt, omdat er veelvuldig overleg plaatsvindt met de vertegenwoordigers.

Het resultaat van de analysefase is het analyserapport als tussenproduct.

Ontwerpfase

Het requirementsmodel en use case model uit de analysefase vormen de basis en input voor de ontwerpfase. Het product van deze ontwerpfase is het ontwerprapport. In dit rapport zijn de nodige ontwerpen zoals een analyse klassendiagram, design klassendiagram en sequentiediagrammen opgenomen. Naast deze diagrammen zijn ook databasemodellen opgenomen van de huidige centrale database, de aangepaste centrale database en een model voor de lokale database.

De centrale database bevat de dossiers van de jongeren. Aan elk dossier is een maatje gekoppeld welke meerdere dossiers kan beheren. De dossiers bevatten de leefgebieden waaraan de taken worden toegewezen. Om taken toe te wijzen aan leefgebieden moet de huidige database aangepast worden waarbij de leefgebieden een aparte entiteit worden.

Het over en weer verzenden van gegevens tussen de mobiele applicatie en het registratiesysteem gebeurt via een REST service. Deze service maakt de koppeling mogelijk tussen beide systemen. De mobiele applicatie gebruikt endpoints als adres om gegevens op te halen en te versturen. Figuur 11 is een visuele weergave van de datastromen die plaatsvinden tussen beide systemen en bijbehorende databases.

De gegevens worden als JSON-object verzonden. JavaScript Object Notification objecten zijn compact en makkelijk te verzenden via een REST service. De format van dit JSON-object is tevens opgenomen in het ontwerprapport. Dit object bevat relevante data vanuit de databases, zoals dossiernummers, persoonsgegevens, de leefgebieden en bijbehorende taken. Na het ophalen van het JSON-object wordt deze omgezet naar reële dataobjecten en opgeslagen in de lokale sqlite database zodat de gegevens ook offline beschikbaar zijn. In deze lokale database worden ook gegevens zoals afspraken, contacten en inloggegevens opgeslagen.

Ontwikkelfase

Voor het ondersteunen van zowel iOS en Android besturingssystemen is er gekozen voor Xamarin.forms als ontwikkeltechniek. Deze techniek maakt het mogelijk om een app te bouwen die op beide besturingssystemen draait.

Voor het ontwikkelen van de applicatie is er gekozen voor Model-View-ViewModel (MVVM) voor de mobiele applicatie als architectuur. Er is hiervoor gekozen, om een duidelijke scheiding aan te houden tussen de business logica en presentatie ervan. Tevens is het gebruik van MVVM 'good practice' binnen de Xamarin omgeving. Het 'binding' concept komt hiermee goed tot z'n recht. Daarnaast wordt een mediator pattern gebruikt voor de communicatie tussen verschillende ViewModels.

Voor het ontwikkelen van het systeemdeel 'Taken' van het registratiesysteem wordt een MVC architectuur gebruikt met een repository pattern. Deze pattern is bedoeld voor een duidelijke scheiding tussen de business logica en de database.

Testfase

Tijdens de testfase worden een testplan en detailtestplannen (DTP) opgesteld. In het testplan staat beschreven welke testsoorten er uitgevoerd worden, welke testtools er gebruikt zullen worden, welke testomgevingen er zijn en hoe bevindingen geregistreerd en afgehandeld worden.

Per testsoort wordt er een DTP opgesteld en wordt dieper ingegaan op een testsoort. In het DTP worden testscripts, het testdraaiboek en bevindingen opgenomen.

De testsoorten die aan bod komen tijdens het testproces zijn unittest en ketentest. Met deze testsoorten wordt het systeem op functionaliteit en koppeling met het bestaande systeem getest.

De testfase wordt afgesloten met de analyse van de bevindingen, het resultaat ervan en een advies voor de opdrachtgever.

Uitvoering

Alle werkzaamheden met betrekking tot het ontwikkelen van de mobiele applicatie zijn na de brainstormsessie van begin tot eind door de afstudeerder uitgevoerd.

Analysefase

In de analysefase zijn verschillende werkzaamheden uitgevoerd. Het gaat hierbij om:

- Deelnemen aan de brainstormsessie
- Afnemen van interviews met stakeholders
- Opstellen van mockups voor het registratiesysteem
- Opstellen van een context diagram en use case diagram
- Opstellen van globale, geprioriteerde en gespecificeerde requirements
- Opstellen van use case diagram en use case beschrijvingen
- Opstellen van een requirements traceability matrix

Brainstormsessie

De brainstormsessie is de eerste stap geweest in de analysefase voor het vergaren van informatie. Er is een beter beeld gecreëerd van de gewenste oplossing en het doel dat nagestreefd wordt.

Het doel van deze sessie is gezamenlijk vanuit meerdere invalshoeken na te denken over verschillende aspecten van een digitale oplossing en een advies uitbrengen naar de opdrachtgever. Tijdens deze sessie wordt er nog niet gesproken van een mobiele applicatie maar over een digitale oplossing.

Hoewel deze sessie niet door de afstudeerder is geïnitieerd past deze goed binnen het ontwikkelproces.

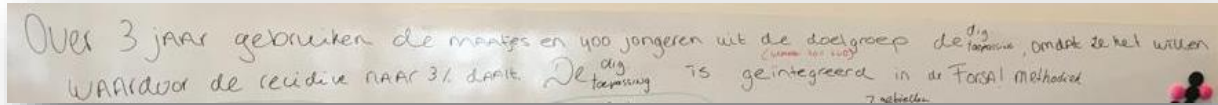


Figuur 3: Multidisciplinair team voor de brainstormsessie

Na een korte introductie van elk teamlid en indeling van de dag is het vraagstuk in kaart gebracht. De opdrachtgever heeft aangegeven dat de organisatie kampt met de vraag hoe ze een grotere bijdrage kan leveren aan het ondersteunen van ex-gedetineerde jongere in de eerste zes weken na vrijlating.

De scope van dit probleem is dat de jongeren direct na vrijlating in de eerste zes weken in een situatie terechtkomen waarbij ze geen of weinig hulp krijgen. De jongeren krijgen een kaartje voor de bus en een vuilniszak met hun eigen spullen mee. Ze komen op straat terecht en zijn volledig op zichzelf aangewezen.

De volgende stap was het bepalen wat er met de digitale oplossing op de lange termijn behaald moet worden. Hiervoor moet ieder lid een invulling geven aan hetgeen wat hij/zij denkt dat er met de oplossing over 3 jaar behaald is. Vanuit de aangedragen ideeën en suggesties wordt een doel geformuleerd. Dit doel moet bij elke stap van de sprint voor ogen gehouden worden. De opdrachtgever dient akkoord te gaan met het geformuleerd doel.



Over 3 jaar gebruiken de maatjes en 400 jongeren uit de doelgroep de digitale toepassing, omdat ze het willen. Waardoor de recidive naar 3% daalt. De digitale toepassing is geïntegreerd in de Forsa! methodiek.

Figuur 4: Geformuleerd doel dat behaald moet worden

Het doel dat hierbij is geformuleerd luidt: 'Over 3 jaar gebruiken de maatjes en 400 jongeren uit de doelgroep de digitale toepassing, omdat ze het willen, waardoor de recidive naar 3% daalt. De digitale toepassing is geïntegreerd in de Forsa! methodiek.'

Nadat er een doel geformuleerd is wordt de customer journey in kaart gebracht. Hierbij wordt er gekeken naar de reis die de gebruiker aflegt vanaf het moment dat hij/zij in contact komt met de organisatie. De gebruikers zijn de jongere en het maatje.

Jongere

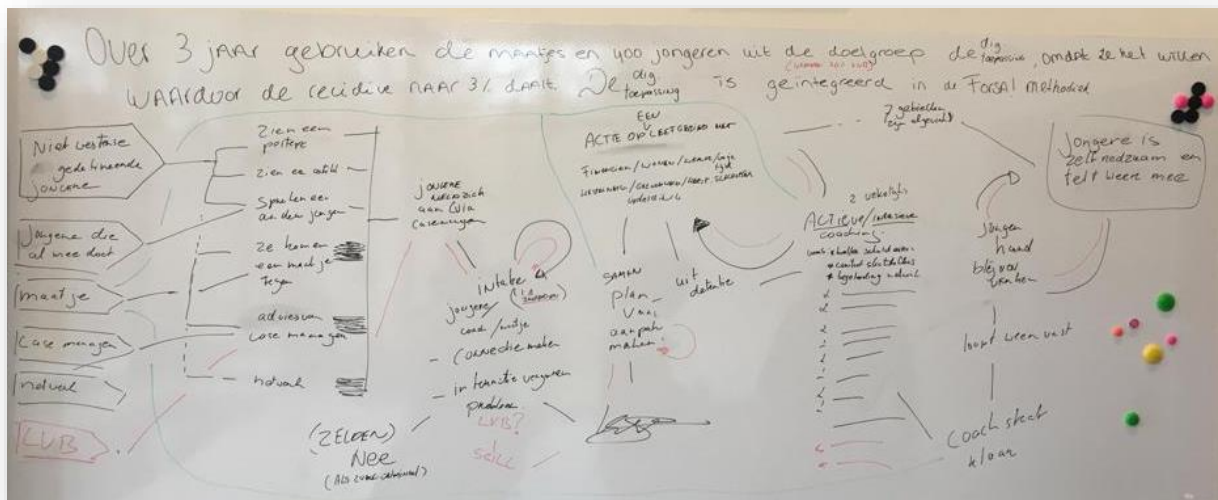
De jongere meldt zich aan via de casemanager om mee te doen met het Forsa! programma. De casemanager geeft dit door aan Bureau MHR. Vervolgens wordt de jongere in het detentiecentrum benaderd door een medewerker van Bureau MHR voor een intake. Tijdens dit intakegesprek wordt een verwachtingspatroon geschetst. Wat verwacht Bureau MHR van de jongere en vice versa. Als een jongere besluit door te gaan met het programma van Bureau MHR komen er vervolggesprekken. Tijdens een van de vervolggesprekken wordt er samen met het maatje een plan van aanpak opgesteld. Voordat dit plan wordt opgesteld wordt er per leefgebied gekeken naar de problemen waar een jongere tegenaan loopt. In het plan van aanpak wordt opgenomen hoe deze problemen worden aangepakt. Dit plan is een belangrijk document voor de rest van de doorlooptijd met Bureau MHR. Dit duurt, afhankelijk van de gemeente waar de jongere staat ingeschreven, één tot twee jaar.

Na vrijlating kan de jongere contact opnemen met het maatje om een contactmoment in te plannen. Tijdens dit contactmoment worden de gemaakte plannen nagelopen en twee wekelijkse gesprekken ingepland. Op deze manier vindt er coaching plaats en wordt de jongere ondersteund om zelfredzaam te worden. Op het moment dat de problemen van alle leefgebieden zijn aangepakt wordt er gesproken van een succes.

Maatje

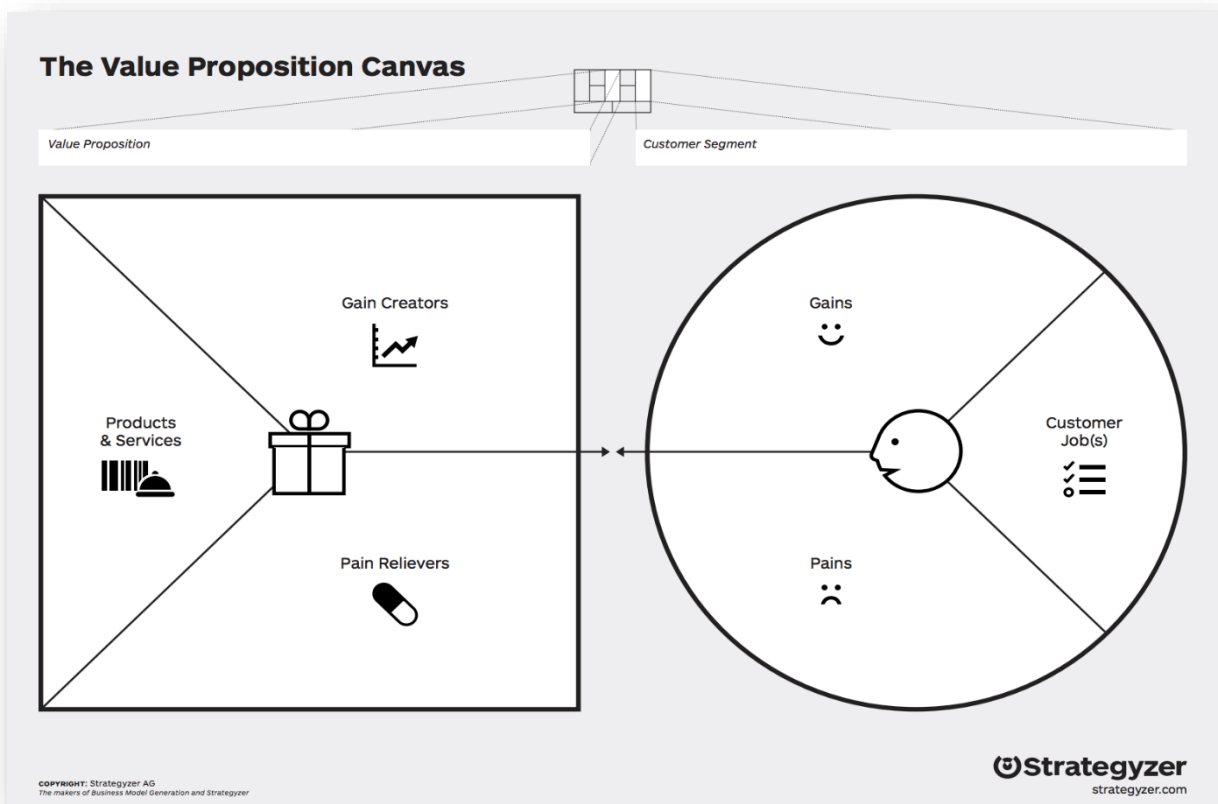
Een maatje krijgt van de jongerecoach te horen dat een cliënt zich heeft aangemeld voor het Forsa! programma. Het maatje gaat op het afgesproken moment naar de cliënt die op dat moment nog in detentie zit. Het eerste contactmoment is in detentie en ongeveer drie tot zes maanden voor vrijlating. Tijdens dit moment bespreekt het maatje de voorwaarden en verwachtingen met de jongere. Als de jongere hiermee akkoord gaat wordt er een plan van aanpak opgesteld. Het maatje noteert alle besproken leefgebieden met pen en papier. Het is verboden om een telefoon of ander elektronisch apparaat mee te nemen naar de jongere. Na het gesprek (intake- of vervolggesprek) gaat het maatje naar huis of terug naar Bureau MHR. De eerstvolgende keer dat het maatje inlogt op het registratiesysteem wordt het gesprek opgeslagen in het systeem. Hierbij worden per leefgebied de aandachtspunten genoteerd en een eventuele vervolgspraak.

Na detentie komt de jongere langs bij Bureau MHR om vervolgstappen te bespreken en eventuele nieuwe punten aan te kaarten. Tevens wordt er gekeken of punten uit het plan van aanpak opgepakt kunnen worden. Zoals het afspreken van een betalingstermijn met een schuldeiser bijvoorbeeld. Deze gesprekken en voortgangen worden eveneens gedocumenteerd. Dit proces loopt een maatje met elk cliënt na tijdens de doorlooptijd van het Forsa! programma.



Figuur 5: Customer journey

De volgende stap is 'The value proposition canvas'. Dit is een waarde propositie canvas met aan de rechterkant het klantsegment en links de waarde propositie vanuit de organisatie. Dit model is ontwikkeld door de Zwitser Alexander Osterwalder waarbij de klant en zijn wensen centraal worden gesteld. Het draait er bij een Value Proposition Canvas om om te achterhalen waarvoor de klant een leverancier nodig heeft, wat de klant extra waarde kan opleveren en wat de klant als vervelend of nadelig ervaart (Mulder, 2018).



Figuur 6: The Value Proposition Canvas van Osterwalder

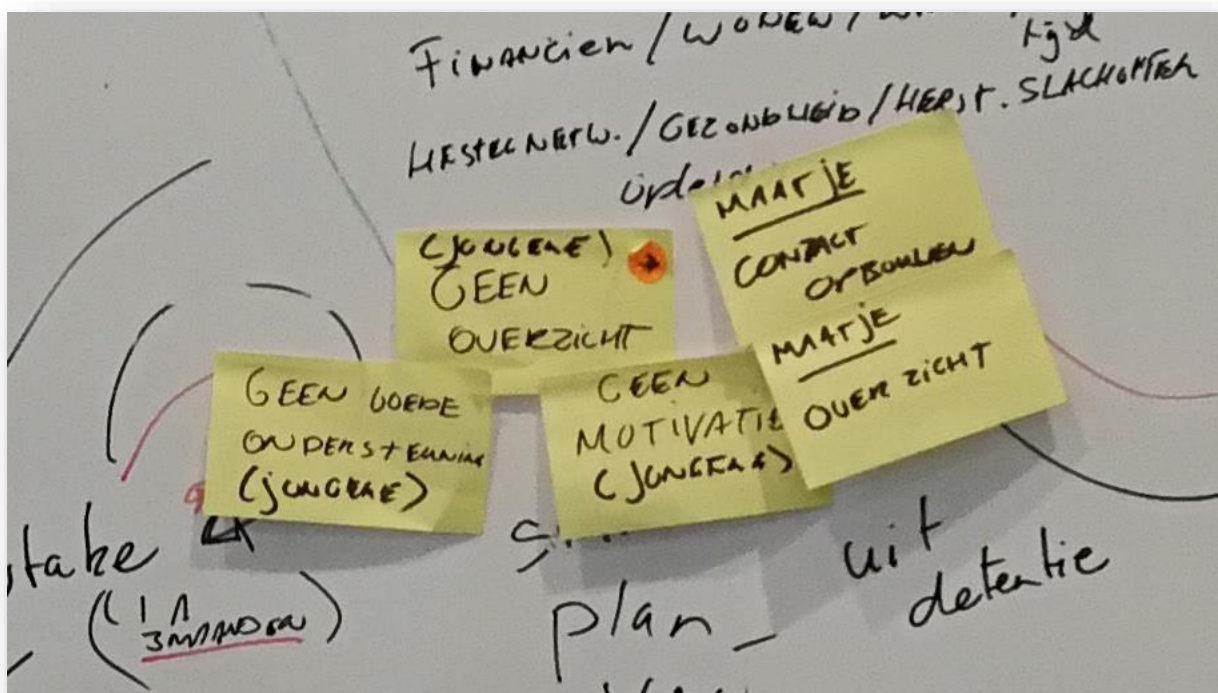
Dit model is tijdens de workshop gebruikt voor het achterhalen waar de gebruiker tegenaan loopt (de pains) tijdens de customer journey. Voor deze stap moet elk teamlid opschrijven waarvan hij/zij denkt wat de pains zijn waar de gebruiker mee kampt. Deze zijn vervolgens opgeplakt op grote ramen en gegroepeerd op overeenkomstigheid. Hierna moet elk lid drie gekleurde stickers opplakken bij de pains waarvan hij/zij denkt dat deze belangrijker zijn dan andere.

Voor een jongere zijn de volgende pains naar voren gekomen;

- Geen goede ondersteuning
- Geen overzicht
- Geen motivatie

Voor een maatje zijn de volgende pains gedefinieerd;

- Contact opbouwen
- Geen overzicht



Figuur 7: De pains van een jongere en maatje

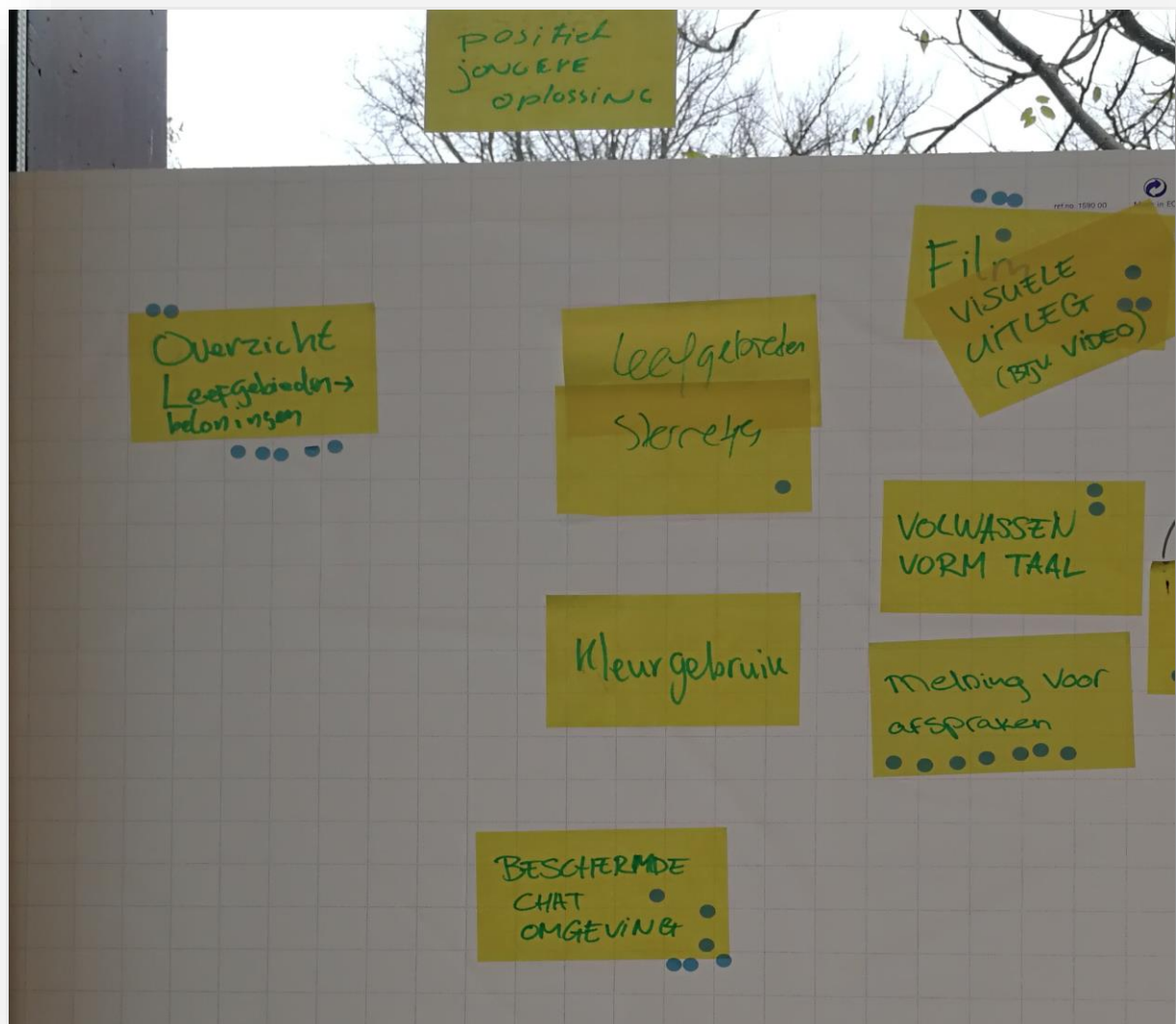
Door het gebrek aan tijd is er bij de volgende stap, het benoemen van zoveel mogelijk oplossingen, gebruik gemaakt van de HIFI-prototype voor een digitale toepassing die door cmd-studenten is gemaakt.

Voor het implementeren van de digitale oplossing en ontwerp van de cmd studenten zijn een tweetal mogelijkheden voorgelegd aan de opdrachtgever. Deze zijn een responsive webapplicatie en een native mobiele applicatie.

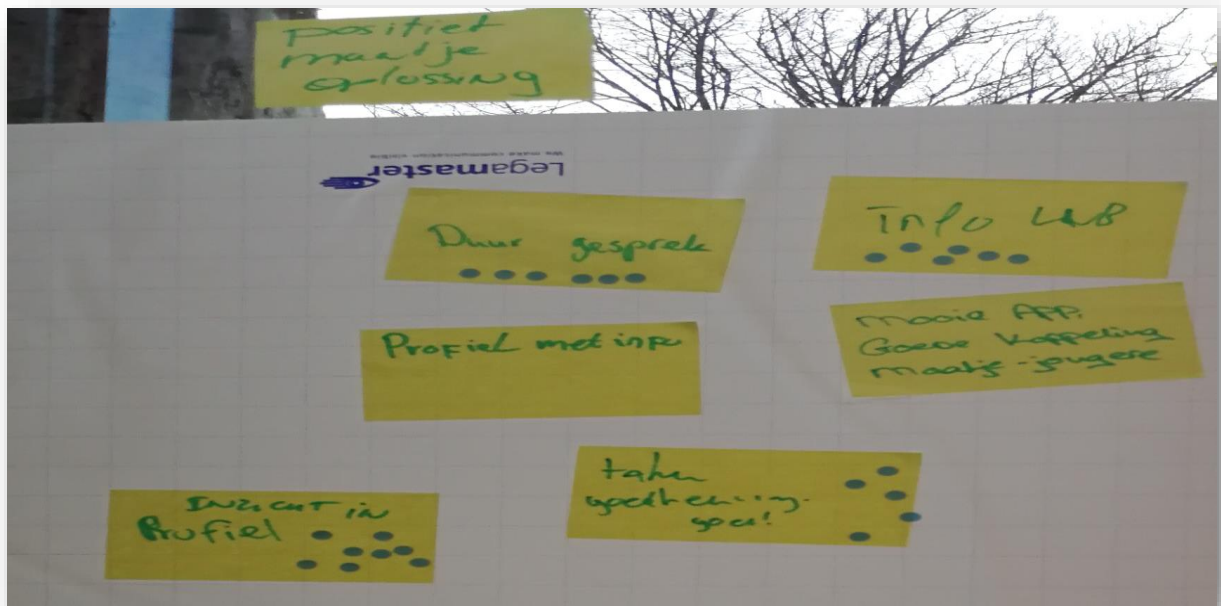
Op basis van het ontwerp van de cmd-studenten moet elk teamlid noteren wat er goed en slecht is aan het ontwerp. Nadat alle antwoorden zijn opgeplakt en besproken moet elk lid gekleurde stickers plakken op datgene wat in de digitale oplossing terug moet komen en welke weggelaten mag worden. Het aantal stickers geeft aan welke punten van het ontwerp positief zijn en welke beter kunnen. Op basis van dit puntensysteem met gekleurde stickers moet de opdrachtgever drie keuzes maken van hetgeen dat terug moet komen en wat beter kan. Deze keuzes zijn in Tabel 1: *Belangrijkste voor- en nadelen per gebruiker op basis van het cmd-ontwerp* weergegeven. In de figuren 8 tot en met 11 is te zien hoe deze stap is uitgevoerd. Daarnaast is er in Tabel 2 de voordelen opgenomen voor de implementaties van het cmd-ontwerp.

Gebruiker	Voordelen	Nadelen
Jongere	<ul style="list-style-type: none"> ▪ Overzicht per leefgebied ▪ Melding voor afspraken ▪ Beschermde chatomgeving 	<ul style="list-style-type: none"> ▪ Visuele vormgeving ▪ Goedkeuring volbrachte taak ▪ AVG-wetgeving
Maatje	<ul style="list-style-type: none"> ▪ Duur gesprek ▪ Info LVB ▪ Inzicht in profiel 	<ul style="list-style-type: none"> ▪ Overlapping registratiesysteem m.b.t informatie methodiek en LVB ▪ Het echte gesprek

Tabel 1: Belangrijkste voor- en nadelen per gebruiker op basis van het cmd-ontwerp



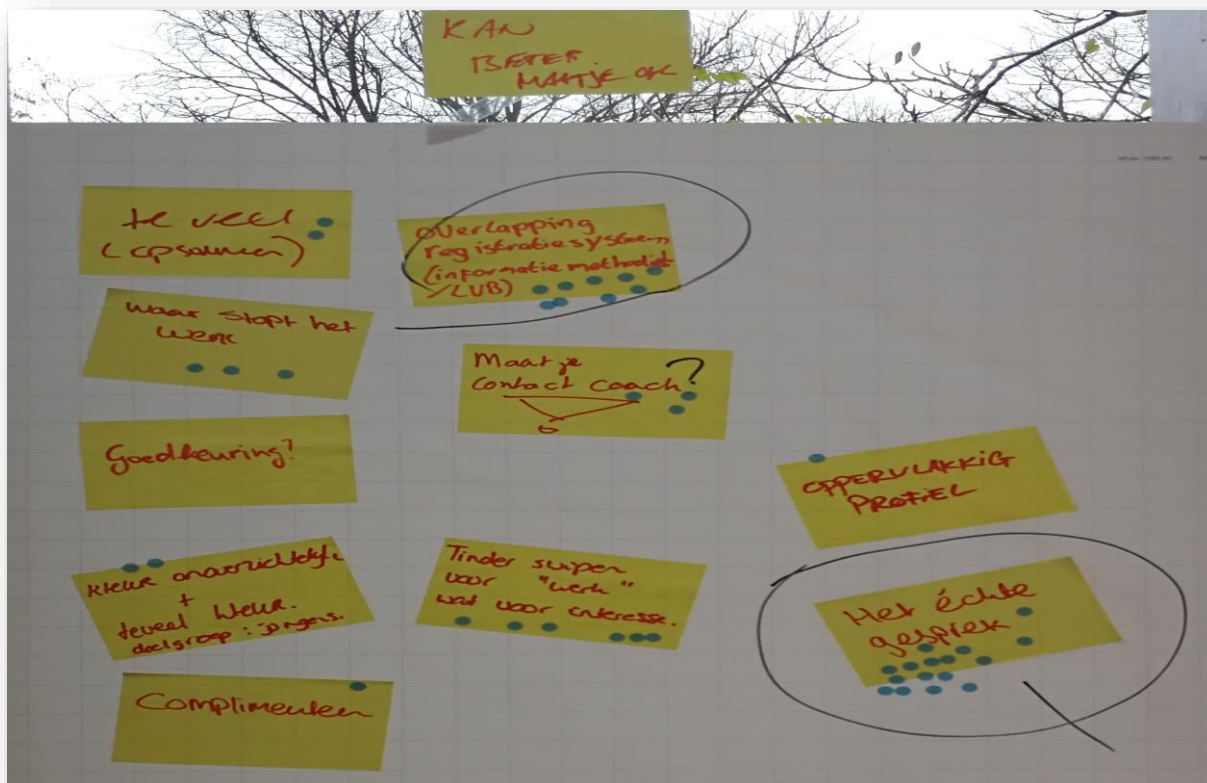
Figuur 8: Voordelen van het ontwerp voor gebruiker jongere



Figuur 9: Voordelen van het ontwerp voor gebruiker maatje



Figuur 10: Nadelen van het ontwerp voor gebruiker jongere



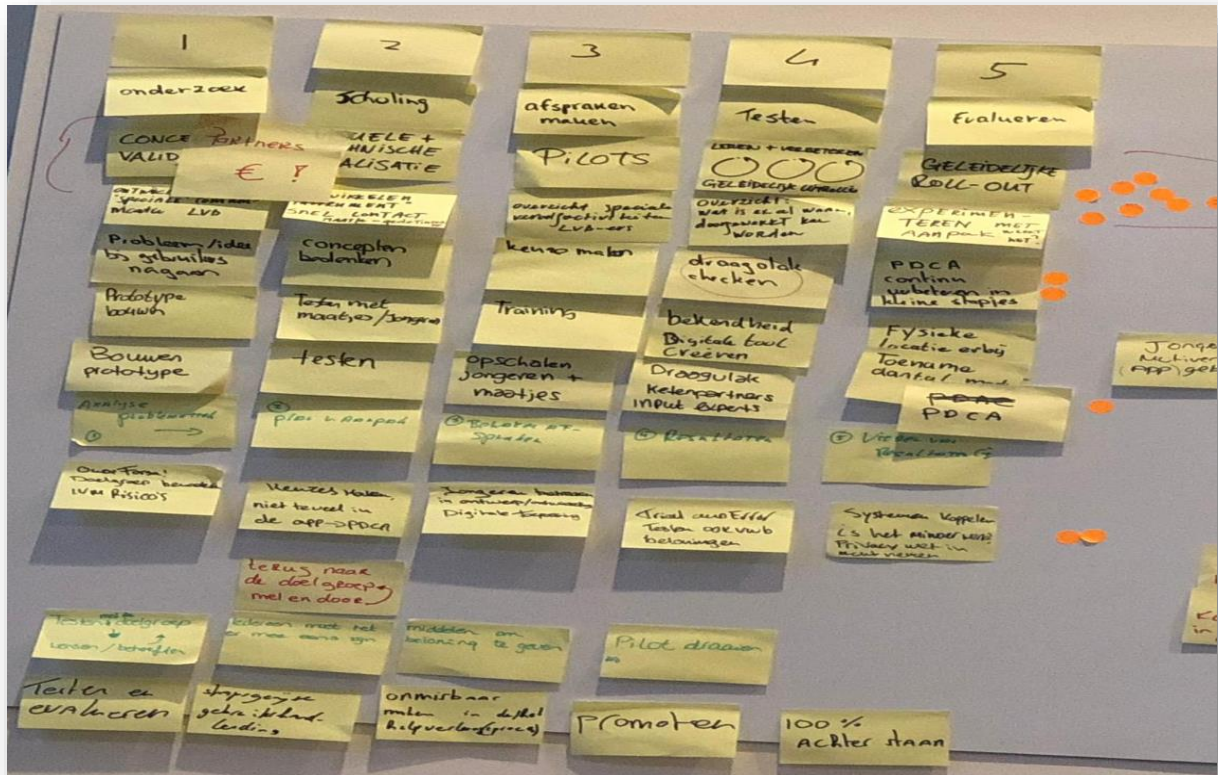
Figuur 11: Nadelen van het ontwerp voor gebruiker maatje

Responsive webapplicatie	Native mobiele applicatie
Benaderbaar in het RIC van het detentiecentrum	Gegevens zijn offline beschikbaar
Altijd online	Jongere kan altijd contact opnemen met maatje
Kan ook op computer benaderd worden	

Tabel 2: Voordelen van de implementaties van het cmd-ontwerp

De keuze is gevallen op een native mobiele applicatie omdat, de informatie ook offline beschikbaar moet zijn voor de jongere vanwege het feit dat bijna iedereen een mobiele telefoon tot zijn beschikking heeft, maar niet altijd met een internetverbinding.

De laatste stap van de workshop is het bepalen van de vervolgstappen. Elk lid mag vijf stappen adviseren aan de opdrachtgever waarvan hij/zij denkt wat de te nemen vervolgstappen zijn. Deze worden vervolgens per lid besproken en opgeplakt. Nadat iedereen is geweest mag elk lid een sticker opplakken bij een reeks van vijf stappen die voor hem/haar een logische reeks lijkt. Op basis van de keuzes van de leden kan de opdrachtgever een keuze maken voor de te nemen stappen.



Figuur 12: Vervolgstappen aangedragen door elk teamlid

In figuur 12 is te zien welke vijf stappen elk lid heeft aangedragen. De oranje stickers zijn de keuzes van elk lid waarmee een advies uitgebracht is naar de opdrachtgever. De keuze van de opdrachtgever omhelst de volgende vijf stappen;

1. Concept validatie
2. Visuele en technische specialisatie
3. Pilots
4. Leren, valideren en geleidelijk uitrollen
5. Geleidelijke roll-out

Afnemen van interviews met stakeholders

Voor het achterhalen van de requirements zijn er interviews afgenomen met de stakeholders. Hierbij gaat het om de opdrachtgever, jongerecoaches, maatjes, een jongere en de coördinator van Forsa! Hierbij is er maar één jongere geïnterviewd, omdat zij niet zo happig blijken te zijn hiervoor.

Bij het afnemen van de interviews is er gekozen voor halfopen interviews. Er is voor halfopen interviews gekozen, omdat de mockups en het ontwerp van de CMD-studenten als uitgangspunten gebruikt konden worden. De ontworpen functionaliteiten zijn gebruikt als te bespreken punten bij de interviews. Hierbij gaat het om:

- In- en uitloggen
- Profiel aanmaken
- Contacten
- Agenda
- Taken
- Chatten
- Noodknop
- Info LVB
- FAQ's

Bij het voeren van de gesprekken met de maatjes is naar voren gekomen dat er geen behoefte meer is aan een chatfunctie of agendafunctie voor de maatjes. Doordat ze nu de beschikking hebben over een zakelijke telefoon is het geen obstakel meer voor ze om Whatsapp te gebruiken als communicatiemiddel met de jongeren.

Voor het inplannen van afspraken en beheren van een agenda wordt organisatie breed gebruik gemaakt van de agenda functie van One.com. Zij beheren tevens het emailverkeer van Bureau MHR.

Na het voeren van de gesprekken heeft er een werkoverleg plaatsgevonden waarbij de opdrachtgever de voortgang wilde bekijken. Tijdens dit overleg is er een kleine presentatie geweest van de mobiele applicatie waarbij een gedeelte van de layout reeds ontwikkeld was. Daarnaast zijn de bevindingen van de interviews besproken en is besloten om de agendafunctie voor de maatjes en de chatfunctie niet te bouwen.

Opstellen van mockups voor het registratiesysteem

Voor visuele ondersteuning bij de interviews met de stakeholders zijn er mockups gemaakt van het registratiesysteem. Deze mockups zijn een weergave van het registratiesysteem nadat het systeemdeel 'Taken' is toegevoegd. Volgens de requirements moeten taken gekoppeld worden aan de leefgebieden van de jongere. Om deze reden is het toevoegen van taken alleen mogelijk in een leefgebied van een dossier. Het beheren van taken is op dezelfde manier gemodelleerd als het beheren van dossiers om uniformiteit te behouden binnen het registratiesysteem.

FORSAIClientvolgsysteem

Karima Daoudi

Karima Daoudi
Directeur

Intakegesprek

Dossiernr:
DNH0095

Naam cliënt:
Ali Bouati

Datum en tijd:
2/4/2019

Tijdsduur in uren:
1

Gesprekspartners:
Rachid, Nora, Karima

Type gesprek:
Pl bezoek

Algemeen Wonen Werk Opleiding Financiën Vrijtijdsbesteding Gezondheid/Welzijn Herstel netwerk Herstel slachtoffer

Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit dui vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum, praesent nascetur pulvinar sed, in dolor pede in aliquam, risus nec error quis pharetra. Eros metus quam augue suspendisse, metus rutrum risus erat in. In ultrices quo ut lectus, etiam vestibulum urna a est, pretium luctus euismod nisl, pellentesque turpis hac ridiculus massa. Venenatis a taciti dolor platea, curabitur lorem platea urna odio, convallis sit pellentesque lacus proin. Et ipsum velit diam nulla, fringilla vel tincidunt vitae, elit turpis tellus vivamus, dictum adipiscing convallis magna id. Viverra eu amet sit, dignissim tincidunt volutpat nulla tincidunt, feugiat est erat dui tempor, fusce tortor auctor vestibulum. Venenatis praesent risus orci, ante nam volutpat erat. cursus non mollis interdum maecenas, consequat imperdiet penatibus enim, tristique luctus tellus eos accumsan, ridiculus erat laoreet nunc. Lorem ipsum dolor sit amet, maiores ornare ac fermentum, imperdiet ut vivamus a, nam lectus at nunc. Quam euismod sem, semper ut potenti pellentesque quisque. In eget sapien sed, sit dui vestibulum ultricies, placerat morbi amet vel, nullam in in lorem vel. In molestie elit dui dictum.

Voeg een taak toe aan dit leefgebied:

Vraag studie financiering aan

Voeg toe

Openstaande taken Afgeronde taken

☐ Maak een lijst van interessante studies

☐ Bezoek open dagen

☐ Schrijfje in via Studielink

Tussentijdse evaluatie(pos/neg + motivering):

Nee

Datum volgende afspraak:

2/4/2019

Onderwerp volgend gesprek:

Here is a some text input.

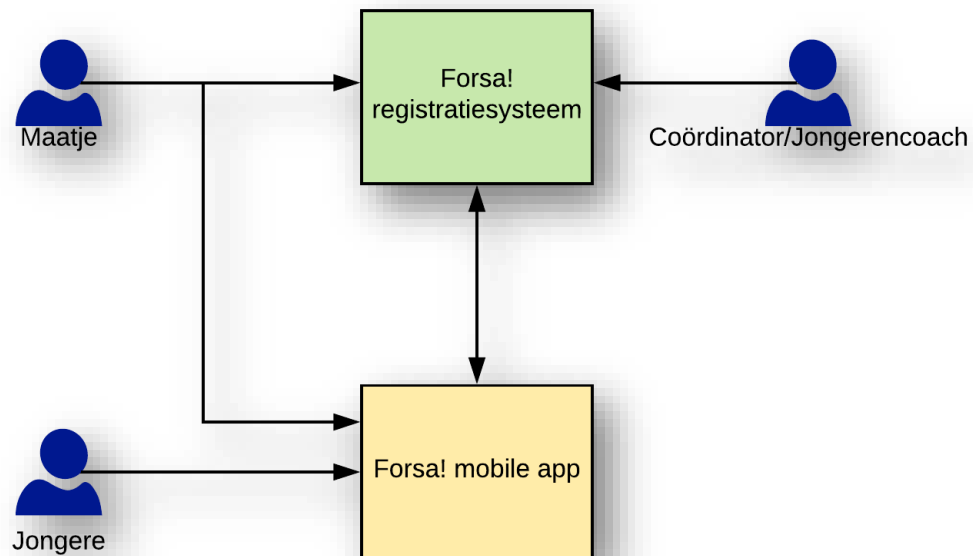
Here is another paragraph of input.

Alles opslaan Annuleren

Figuur 13: Het koppelen van taken aan een leefgebied in het registratiesysteem

Opstellen van een context diagram en use case diagram

Een context diagram geeft de interactie weer van het systeem met de omgeving. Dit diagram is gebruikt ter ondersteuning bij het afbakenen van het systeem en achterhalen van use cases samen met de stakeholders. Van hieruit is een globaal use case diagram opgesteld. Later in het proces is op basis van de geprioriteerde requirements een gedetailleerd use case diagram opgesteld met specifiekere use cases. Het globale use case diagram is gebruikt voor het opstellen van requirements. Het gedetailleerde use case diagram is gebruikt als input voor de ontwerpfase.



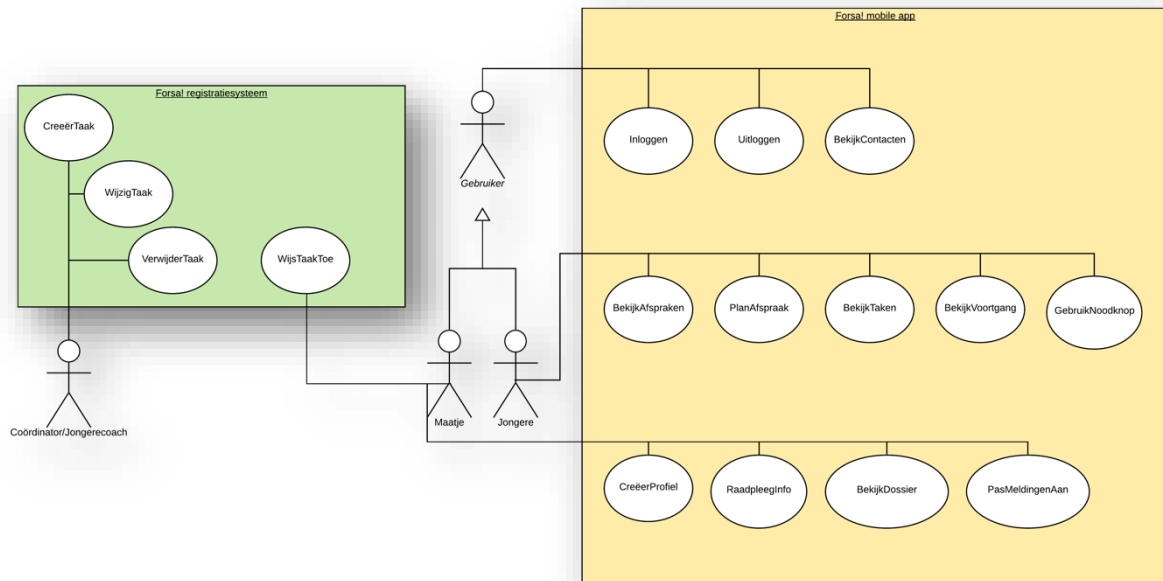
Figuur 14: Context diagram Forsa! Mobile

Forsa! mobile app

De mobiele applicatie is het te ontwikkelen systeem. De gebruikers van dit systeem zijn de jongeren en maatjes.

Forsa! registratiesysteem

Het Forsa! registratiesysteem is het bestaande systeem waar de mobile app aan gekoppeld moet worden. Deze koppeling zorgt ervoor dat er gegevens vanuit het registratiesysteem opgehaald kunnen worden en in de mobile app getoond worden.



Figuur 15: Use case diagram Forsa! Mobile en Forsa! Registratiesysteem

Opstellen van globale, geprioriteerde en gespecificeerde requirements

Na het voeren van gesprekken en afnemen van interviews met de stakeholders zijn de requirements opgesteld en de kwaliteitseisen zijn vastgesteld.

Allereerst zijn de business requirements en globale requirements opgesteld. Vervolgens zijn in samenspraak met de stakeholders deze globale requirements geprioriteerd op basis van het MoSCoW principe.

Tot slot worden de geprioriteerde requirements gespecificeerd. In het kader van duidelijkheid en eenvoud zijn de requirements atomair gedefinieerd. Dit houdt in dat deze requirements één eis representeren welke niet verder op te delen is. De Swart beschrijft een requirementspatroon waarbij softwarerequirements volgens een vast patroon worden gedefinieerd. Dit patroon is gehanteerd bij deze opdracht. De requirements zijn tevens voorwaarts en achterwaarts traceerbaar om het belang van een requirement te bepalen. Hieronder is een overzicht van de kwaliteitseisen en de lijst met de business requirements.

Productkwaliteit

Bruikbaarheid

Leerbaarheid

In het kader van het project in samenwerking met ZonMW moet de leercurve van de app laag zijn. Bij het ontwikkelen van de app moet er rekening gehouden worden met jongeren met een licht verstandelijke beperking. Het HIFI ontwerp van de CMD-studenten is hierop toegespitst.

Beveiligbaarheid

Vertrouwelijkheid

Het is van belang dat de gebruiker alleen die informatie te zien krijgt die voor hem/haar bestemd is. Er wordt met privacygevoelige informatie gewerkt.

Kwaliteit tijdens gebruik

Voldoening

Bruikbaarheid

Het systeem moet foutafhandeling op een dusdanige manier verwerken dat de gebruiker in een notificatie direct afleest wat er fout gaat, dit kan oplossen en verder kan gaan met het systeem.

Tevredenheid

Het is van belang dat het systeem de gebruiker voldoende motiveert en feedback geeft om het systeem te blijven gebruiken.

Businessrequirements

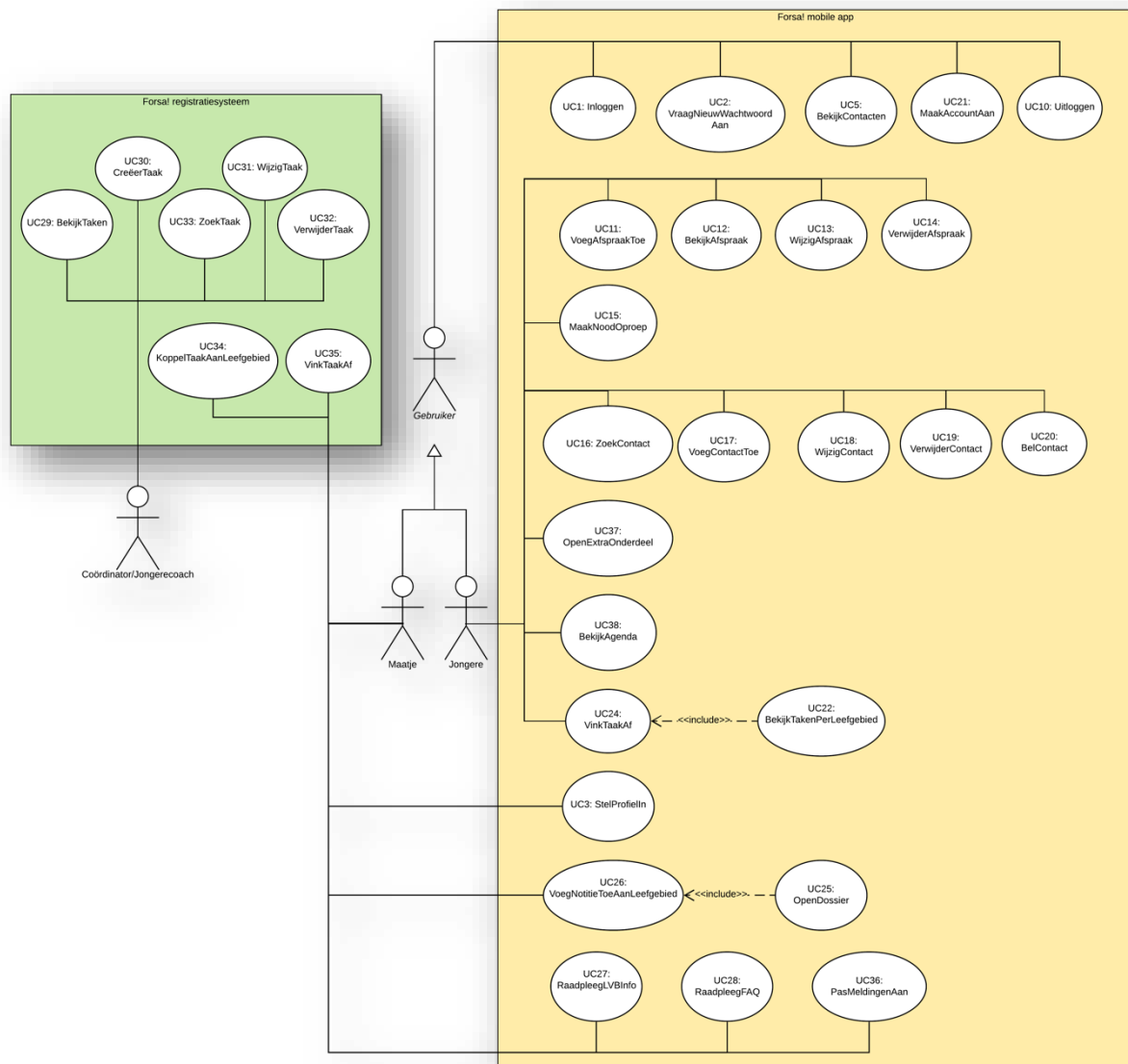
Code	Requirement
BR.1	De opdrachtgever wil dat het percentage recidive over 3 jaar daalt van 4% naar 3%.
BR.2	De opdrachtgever wil het contact tussen een jongere en maatje met 50% verhogen en verbeteren.
BR.3	De opdrachtgever wil een directe koppeling van de mobiele app met het huidige registratiesysteem voor het ophalen van de bestaande gebruikersgegevens en om redundantie en overbodig werk te voorkomen
BR.4	De opdrachtgever wil de jongeren ondersteunen en motiveren bij het uitvoeren van taken, managen van afspraken en contacten die nodig zijn om een herintreding te maken in de samenleving
BR.5	De opdrachtgever wil dat maatjes ondersteund worden middels werk gerelateerde informatie bij het uitvoeren van werkzaamheden met ex-gedetineerde jongeren
BR.6	De opdrachtgever wil de jongeren binden aan Bureau MHR door het personaliseren van de mobiele app met behulp van accountgegevens, dossiernummer en profiel.

Tabel 3: Opgestelde businessrequirements Bureau MHR

Use case beschrijvingen en requirements traceabilitymatrix

Voor elke use case uit het gedetailleerde use case diagram is een use case beschrijving opgesteld. Deze hebben als doel het uiteenzetten van de flow die een bepaalde actor/gebruiker met het systeem doorloopt per use case. Hierbij kunnen eveneens alternatieve flows uit voortvloeien. Deze zijn een beschrijving van de flow in een eventuele alternatieve situatie.

Een requirements traceabilitymatrix is een tool om na te gaan of alle functionele requirements uit het requirementsmodel te herleiden zijn naar het use case model en vice versa. Het gaat hier alleen om de functionele requirements, omdat deze een interactie tussen gebruiker en het systeem vertegenwoordigen. Op het moment dat een requirement niet gelinkt kan worden aan een use case, betekent het dat een use case ontbreekt. Bij het niet kunnen linken van een use case aan requirements betekent dat er requirements toegevoegd dienen te worden.



Figuur 15: Gedetailleerd use case diagram

Use case: Inloggen
ID: UC1
Korte omschrijving: Inloggen in de Forsa! Mobile app met een e-mailadres en wachtwoord
Primaire actoren: <ul style="list-style-type: none"> Maatje Jongere
Secundaire actoren: Geen
Precondities: <ul style="list-style-type: none"> De mobiele app dient geïnstalleerd te zijn. Een maatje heeft een account in het registratiesysteem. De gebruiker dient een account te hebben aangemaakt (UC21: MaakAccountAan).
Main flow: <ol style="list-style-type: none"> De use case start bij het opstarten van de app. De gebruiker krijgt een inlogscherm te zien en vult een e-mailadres in.

<p>2.1 While het e-mailadres niet de juiste conventie heeft krijgt de gebruiker de melding 'geen correct e-mailadres' te zien naast het invoerveld.</p> <p>3. De gebruiker vult het bijbehorend wachtwoord in.</p> <p>4. De gebruiker kiest voor [Inloggen].</p> <p>5. Het systeem controleert de gegevens.</p> <p>6. Als het account bestaat en het wachtwoord klopt</p> <p>6.1 Het systeem haalt de token op uit de database.</p> <p>6.2 Het systeem haalt de gebruiker op uit de lokale database.</p> <p>6.3 Als de gebruiker een 'Jongere' is</p> <p>6.3.1 Het systeem stuurt een verzoek naar het registratiesysteem met de inloggegevens en voegt hier de token uit de database aan toe</p> <p>6.3.2 Het registratiesysteem valideert de token</p> <p>6.3.3 Als de token valide is</p> <p>6.3.3.1 Het registratiesysteem controleert de inloggegevens met de gegevens uit de centrale database</p> <p>6.3.3.2 Het registratiesysteem stuurt een response terug.</p> <p>6.3.3.3 Als de response een jongere is</p> <p>6.3.3.3.1 Het systeem toont het beginscherm van de jongere.</p> <p>6.4 Als de gebruiker een 'Maatje' is</p> <p>6.4.1 Het systeem stuurt een verzoek naar het registratiesysteem met de inloggegevens en voegt hier de token uit de database aan toe</p> <p>6.4.2 Het registratiesysteem valideert de token</p> <p>6.4.3 Als de token valide is</p> <p>6.4.3.1 Het registratiesysteem controleert de inloggegevens met de gegevens uit de centrale database</p> <p>6.4.3.2 Het registratiesysteem stuurt een response terug.</p> <p>6.4.3.3 Als de response een jongere is</p> <p>6.4.3.3.1 Het systeem toont het beginscherm van de jongere.</p>
<p>Postcondities:</p> <ul style="list-style-type: none"> De gebruiker is ingelogd
<p>Alternative flow:</p> <p>InloggegevensNietCorrect</p>

Use case: MaakAccountAan
ID: UC21
<p>Korte omschrijving:</p> <p>Aanmaken van een account om in te loggen in het systeem.</p>
<p>Primaire actoren:</p> <ul style="list-style-type: none"> Jongere Maatje
<p>Secundaire actoren:</p> <p>Geen</p>
<p>Precondities:</p> <ul style="list-style-type: none"> De jongere dient een dossiernummer te hebben. Het maatje dient een account in het registratiesysteem te hebben.
<p>Main flow:</p> <ol style="list-style-type: none"> De use case start zodra de gebruiker klikt op de knop [Account aanmaken]. Het systeem toont het formulier voor een nieuw account. De gebruiker vult de benodigde gegevens in. De gebruiker klikt op de knop [Account aanmaken]. Het systeem controleert de gegevens. Als alle invoervelden ingevuld zijn en de wachtwoorden komen overeen. <ol style="list-style-type: none"> Het systeem controleert of het account reeds bestaat in de lokale database. Als het account niet bestaat. <ol style="list-style-type: none"> Het systeem controleert of het dossiernummer reeds bestaat in de lokale database. Als het dossiernummer niet bestaat.

6.2.2.1 Het systeem maakt nieuw Login object aan. 6.2.2.2 Als het dossiernummer is ingevuld. 6.2.2.2.1 Het systeem maakt een nieuw jongere object aan. 6.2.2.2.2 Het systeem koppelt het Login object aan het jongere object. 6.2.2.3 Anders het systeem maakt een nieuw maatje object aan. 6.2.2.4 Het systeem koppelt het Login object aan het maatje object. 6.2.3 Het systeem schrijft de objecten naar de database. 6.2.4 Het systeem toont een succes bericht. 6.2.5 Het systeem toont het inlogscherf.
Postcondities: <ul style="list-style-type: none"> Het account is aangemaakt en de gebruiker kan inloggen (UC1:Inloggen).
Alternative flow: GegevensOntbreken DossiernummerKomtNietOvereen IngevoerdeNaamKomtNietOvereen

Use case: KoppelTaakAanLeefgebied
ID: UC34
Korte omschrijving: De actor voegt een taak toe aan het leefgebied van een jongere.
Primaire actoren: Maatje
Secundaire actoren: Geen
Precondities: <ol style="list-style-type: none"> Het maatje moet ingelogd zijn in het registratiesysteem. De betreffende taak moet reeds zijn toegevoegd in de centrale database. Het dossier van de jongere moet gekoppeld zijn aan het maatje.
Main flow: <ol style="list-style-type: none"> De use case start zodra het maatje een taak selecteert uit de dropdownlist van taken. Het maatje klikt op de knop [Voeg toe]. Het systeem voegt de taak toe aan de takenlijst van het bijbehorende leefgebied.
Postcondities: <ol style="list-style-type: none"> De taak is toegevoegd aan het leefgebied in het dossier van een jongere. De jongere krijgt de toegevoegde taak in zijn dossier in de mobiele app te zien.
Alternative flow: Geen

Validatie

Reviewen

Voor het reviewen van de specificaties is elke stap van deze analysefase samengevat en samen met de resultaten en producten opgenomen in een validatierapport. Een dergelijk rapport is naar elke vertegenwoordiger opgestuurd voor review. Bij elk punt in het rapport heeft de vertegenwoordiger ruimte voor op- of aanmerkingen.

De heer Y. Ouaicha heeft de plaats van mevrouw H. Oualed ingenomen als vertegenwoordiger voor de maatjes en de jongeren. Mevrouw Oualed was ten tijde van het validatiemoment niet meer werkzaam bij Bureau MHR.

Doorspreken

Er is een validatiebijeenkomst gehouden waarbij de validatierapporten zijn nagelopen en doorgesproken. Bij deze bijeenkomst is de requirementsanalist samen met de vertegenwoordigers alle punten van het rapport nagelopen en hebben de vertegenwoordigers de tijd en ruimte gehad om hun op- of aanmerkingen te bespreken.

Resultaat

Op basis van bovenstaande validatietechnieken zijn onderstaande punten naar voren gekomen. De manier waarop dit wel of niet wordt meegenomen in het ontwikkelproces is eronder vermeld.

- Het tonen van begrijpelijke foutmeldingen bij het werken met de app.
 - o *Foutafhandeling wordt verwerkt in het systeem, details m.b.t. dit punt zijn opgenomen bij kwaliteitseisen.*
- Het afvinken van een taak door een jongere is een requirement met prioriteit 'Must have'.
 - o *Requirement is opgenomen en tevens verwerkt in het use case diagram.*
- Is het mogelijk om emoticons toe te voegen aan een chatgesprek.
 - o *Dit wordt nader onderzocht.*
- Een jongere moet alleen contacten kunnen toevoegen die relevant zijn voor het traject.
 - o *Bij het toevoegen van een contact moet de jongere de rol van het contact aangeven, dit is opgenomen in het use case diagram. UR.19 is aangepast.*
- Een jongere moet een gesprek met een ketenpartner van Bureau MHR kunnen beoordelen in de app. Deze beoordeling moet inzichtelijk zijn voor een maatje om in een persoonlijk gesprek met de jongere op voort te borduren.
 - o *Dit wordt meegenomen in een volgende oplevering, omdat dit een nieuwe requirement betreft. Het komt in de lijst met 'Won't have' requirements.*
- De noodknop wordt gebruikt om naar het hoofdnummer van Bureau MHR te bellen en niet meer naar het maatje.
 - o *De betreffende requirement UR.27 is aangepast.*
- Alleen een jongere moet afspraken kunnen inplannen, voor het maatje moet een synchronisatie plaatsvinden met de huidige One.com agenda
 - o *De geprioriteerde requirements zijn aangepast. Maatjes moeten eigen agenda synchroniseren met One.com agenda.*

Ontwerpfase

Voor de ontwerpfase is er gekeken naar de benodigde diagrammen om de te bouwen systemen te ontwerpen. Hierbij gaat het om de Forsa! Mobile app en het systeemdeel 'Taken' in het Forsa! Registratiesysteem. Voor de mobiele applicatie zijn zowel een analyse klassendiagram als een design klassendiagram ontworpen. Voor het systeemdeel 'Taken' alleen een design klassendiagram, omdat het hier om een klein onderdeel gaat. Een analyse klassendiagram heeft in dit geval geen toegevoegde waarde. Daarnaast zijn er databasemodellen opgenomen van de oude centrale database en de aangepaste centrale database. Voor de lokale sqlite database is een entiteitsrelatiediagram opgenomen. Voor het in kaart brengen van het dataverkeer is een information viewpoint toegevoegd.

Klassendiagrammen

Analyse klassendiagram

Als input voor het analyse klassendiagram zijn het requirementsmodel en het use case model gebruikt. Dit diagram bestaat uit de modellen die voorkomen in de mobiele applicatie. Elke klasse heeft een id voor het wegschrijven naar de database. De attributen van de klassen zijn public, omdat voor elk attribuut automatisch een private backingstore aangemaakt wordt in c# mits de set methode leeg blijft. Figuur 16 en 17 zijn een voorbeeld hiervan.

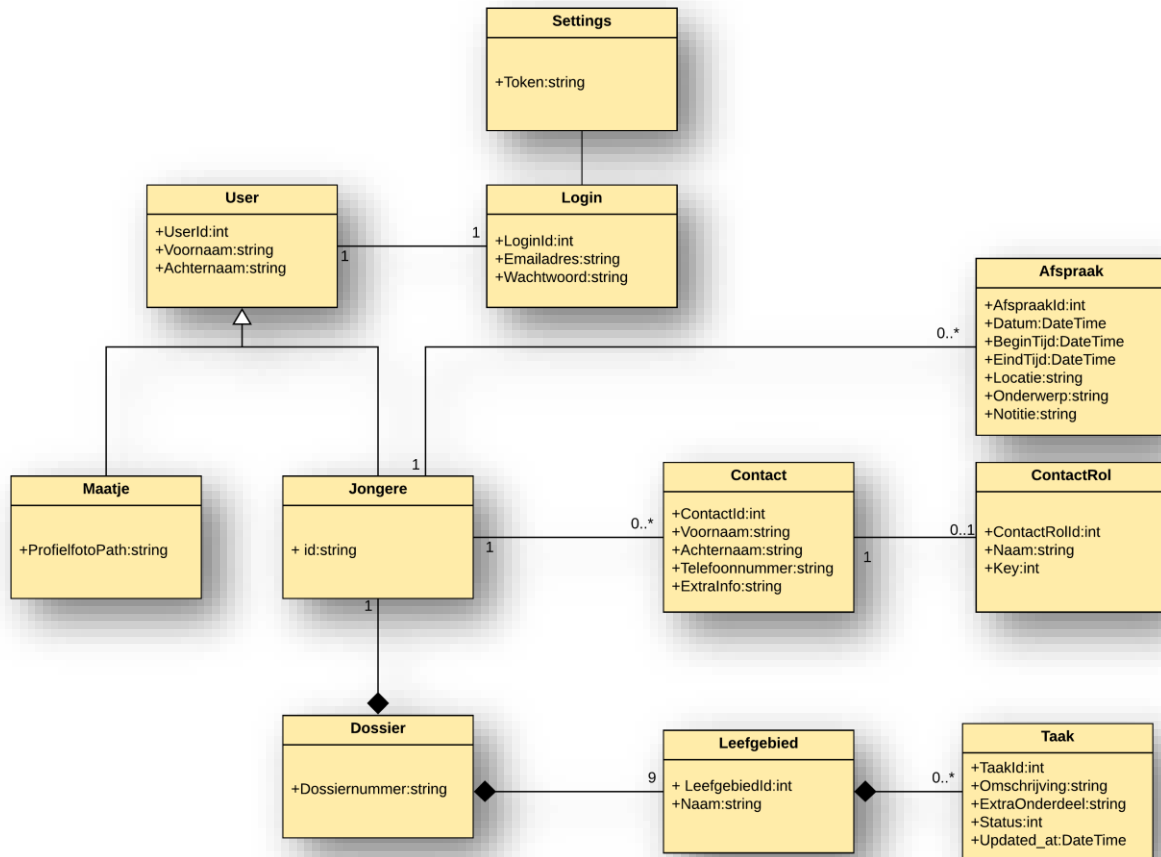
```
private string _locatie;  
  
public string Locatie  
{  
    get  
    {  
        return _locatie;  
    }  
    set  
    {  
        _locatie = value;  
    }  
}
```

Figuur 16: zelf definiëren van een backingstore

```
public string Locatie { get; set; }
```

Figuur 17: property waarbij de backingstore op de achtergrond gedefinieerd wordt

Figuur 18 is een weergave van het analyse klassendiagram. Daaronder volgt een beschrijving van de modellen en hun onderlinge relatie.



Figuur 18: Analyse klassendiagram Forsa! Mobile

User

Deze klasse is gemodelleerd als superklasse van de klassen Maatje en Jongere. Het is een abstracte klasse, omdat er geen objecten van worden aangemaakt in de app. Zowel Jongere als Maatje zijn users, omdat Maatje een verwijzing bevat naar de profielfoto en Jongere een composiet koppeling heeft met Dossier wordt er een onderscheid gemaakt tussen beide. Zonder dossiernummer wordt er geen Jongere aangemaakt. De User klasse heeft een 1op1 associatie met de Login klasse. Dit houdt in dat elke User een Login moet hebben. Bij het aanmaken van een account wordt een User als Jongere of Maatje met een Login object aangemaakt.

Maatje

Het maatje heeft een ProfielfotoPath voor het opslaan van een profielfoto. Dit adres wordt opgeslagen in de 'Settings' optie van de app. Dit is een simpele manier om persistente data van klein formaat lokaal op de telefoon op te slaan.

Jongere

De jongere heeft een id attribuut welke gedefinieerd wordt bij het aanmaken van een account. Deze id wordt gekoppeld aan het dossiernummer van het dossier. De id wordt verkregen door de laatste vijf cijfers van het dossiernummer te nemen. Dit is geen Foreign key, maar een id die in het JSON-object meegestuurd wordt met de request van de API. Daarnaast heeft een jongere een composiet relatie met een dossier, omdat een jongere altijd een dossiernummer moet hebben. Zonder dossiernummer kan er geen jongere aangemaakt worden.

Settings

Voor de applicatie is eenmalig een token gegenereerd. Deze token moet bij elke API request meegestuurd worden om toegang te krijgen tot de gegevens van het registratiesysteem. De token is opgeslagen in de Settings helper klasse

Afspraak

Een afspraak wordt aangemaakt door een jongere en heeft naast een eigen id een foreign key naar het id van de jongere. Een afspraak bevat een datum, een begin- en eindtijd, locatie, onderwerp en er kan een notitie toegevoegd worden.

Contact

Een contact wordt aangemaakt door een jongere en heeft naast een eigen id een foreign key naar het id van de jongere. Een contact heeft een voornaam, achternaam, telefoonnummer en extra info zoals een emailadres. Daarnaast heeft het een 1op1 relatie met ContactRol. Bij het toevoegen van contacten moet een rol gedefinieerd zijn.

Dossier

Een dossier hoort altijd bij een jongere. Daarnaast bestaat het uit 9 leefgebieden. Het dossiernummer wordt gebruikt om een dossier te herkennen bij het ophalen van gegevens uit het registratiesysteem.

Leefgebied

Er zijn negen verschillende leefgebieden welke altijd bij een dossier horen. Een leefgebied kan een lijst van taken bevatten.

Taak

Uit de requirements is te herleiden dat een taak gekoppeld wordt aan een leefgebied. Om deze reden hoort het altijd bij een leefgebied en is dit gemodelleerd middels de composiet relatie met 'Leefgebied'.

Design klassendiagram Forsa! Mobile

Om te beginnen is er een App klasse welke de root klasse is van elke app. Hierin wordt het startscherm gedefinieerd en globale constante gegevens zoals, kleuren en stijlen. Daarnaast wordt de lokale database als statisch attribuut meegegeven met de App klasse. Hiermee is de database altijd bereikbaar via de App klasse en hoeft er niet elke keer een Database object aangemaakt te worden wanneer deze nodig is.

Om de MVVM-architectuur aan te houden is er voor elk Model object een of meerdere Views en ViewModels. Het model object beschrijft de data, zoals Login, User, Contact etc. De views zijn de userinterfaces die de gebruikers te zien krijgen met elk hun XAML attributen zoals een invoerveld, een textlabel of button. De viewmodels bevatten alle logica om de models met de views te verbinden middels het binding concept. Op deze manier blijft de logica gescheiden van de views. Dit wordt beschouwd als 'good practice' binnen Xamarin.forms.

Naast de MVVM indeling zijn er nog een aantal andere onderdelen terug te vinden in het design. Dit zijn de data, helpers en services onderdelen.

In het dataonderdeel wordt de 'Database' klasse gedefinieerd. In deze klasse wordt aangegeven welke tabellen er aangemaakt moeten worden en hoe bepaalde informatie met betrekking tot de verschillende modellen opgehaald kan worden.

Een helper klasse is de 'Settings' klasse welke is opgenomen voor het persistent opslaan van data van klein formaat, zoals de token voor toegang tot het registratiesysteem. Hiervoor moet overigens eerst een plug-in geïnstalleerd worden alvorens deze gebruikt kan worden.

Onder het services onderdeel valt de 'ApiServices' klasse. Deze klasse is verantwoordelijk voor het declareren van de endpoints en de manier waarop er gecommuniceerd wordt met het Forsa! Registratiesysteem. In deze klasse worden vier endpoints gedeclareerd. Deze zijn:

1. "https://forsa.bureaumhr.nl/public/api/v1/checkuser"
2. "https://dev.forsa.bureaumhr.nl/public/api/v1/dossier/"
3. "https://dev.forsa.bureaumhr.nl/public/api/v1/dossiers/"
4. "https://dev.forsa.bureaumhr.nl/public/api/v1/statusupdate/"

De 'checkuser' endpoint(1) is verantwoordelijk voor het controleren van de rol van gebruiker. Bij het aanroepen van deze endpoint of versturen van deze request wordt in het geval van het maatje een emailadres meegestuurd en in het geval van een jongere de voornaam, achternaam en id (laatste vijf cijfers van het dossiernummer). Het antwoord op deze request is jongere, maatje of onbekend. Op deze wijze vindt er een controle plaats op het account dat door de gebruiker is aangemaakt in de app.

De 'dossier' endpoint(2) is verantwoordelijk voor het ophalen van dossiergegevens. Hierbij wordt het emailadres van een jongere meegestuurd. Als antwoord worden alle dossiergegevens van deze jongere geretourneerd. Waaronder de leefgebieden en bijbehorende taken.

De 'dossiers' endpoint(3) is verantwoordelijk voor het ophalen van alle dossiers waar een maatje aan gekoppeld is. Als antwoord worden alle dossiers van het maatje geretourneerd. Waaronder de leefgebieden en bijbehorende taken.

De 'statusupdate' endpoint(4) is verantwoordelijk voor het updaten van de status van een taak. In het registratiesysteem kan een taak vier statussen hebben. Deze zijn:

- 0 = Taak is aangemaakt in het registratiesysteem.
- 1 = Open, taak is gekoppeld aan leefgebied maar niet afgevinkt.
- 2 = In afwachting, taak is afgevinkt in de app en in afwachting van goedkeuring.
- 3 = Taak is goedgekeurd en afgevinkt.

Bij het versturen van de request 'statusupdate' kan de status van de taak gewijzigd worden naar status 1 of 2, afhankelijk van de status op dat moment in de app. Middels een onzichtbaar attribuut 'Updated_at' vindt er een controle plaats om na te gaan welke statusverandering nieuwer is. Hiermee wordt ervoor gezorgd dat de status van de taak chronologisch correct is.

Binnen de applicatie wordt er ook gebruik gemaakt van interfaces. Deze zorgen voor een lagere koppeling tussen verschillende klassen. Hierbij gaat het om de IMessenger interface en de IToast interface.

De IMessenger interface is verantwoordelijk voor de communicatie tussen verschillende viewmodels. De manier waarop dit werkt is dat een viewmodel een bericht verstuurd en dat een ander viewmodel registreert voor hetzelfde bericht. Het geregistreerde viewmodel ontvangt vervolgens het bericht. Middels zo'n bericht kan elk gegeven verstuurd worden. Hierbij kan het gaan om een object zoals een User, een string of een getal.

De IToast interface is verantwoordelijk voor het tonen van de feedback berichten aan de gebruiker. Hoewel dit bij Android standaard is ingebouwd moet deze interface apart gedefinieerd worden in Xamarin.forms om ook op iOS apparaten te zien. Hierbij kan het gaan om een 'Account is aangemaakt' bericht of 'Goed gedaan, ga zo door' bericht na het voltooien en afvinken van een taak.

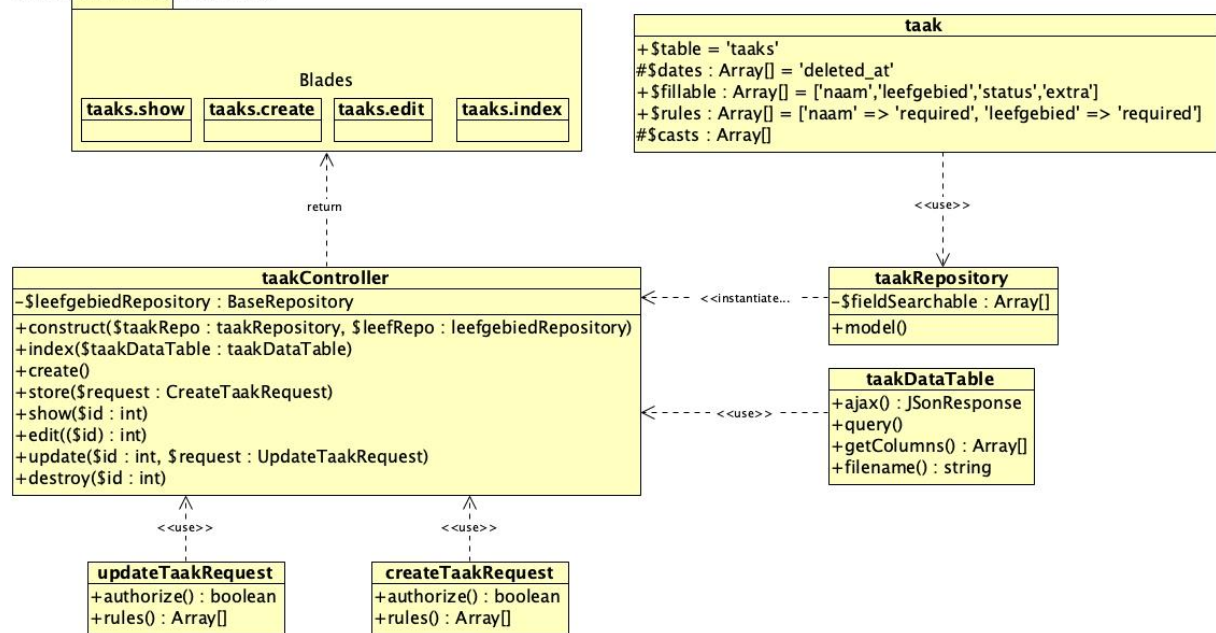
Design klassendiagram Forsa! Registratiesysteem.taak

Het design klassendiagram van het systeemdeel 'Taak' voor het registratiesysteem is op basis van de architectuur van het registratiesysteem ontworpen. Het gaat hier om een MVC-architectuur(Model-View-Controller) met een repository pattern.

Het modelobject is 'Taak' en beschrijft de onderdelen waaruit een taak bestaat. Middels de repository pattern vindt de communicatie plaats met de database en de controller waardoor het modelobject afgeschermd blijft.

Voor de view zijn verschillende views ontworpen. Het gaat hier om een view voor het tonen van een overzicht van alle taken, een view voor het aanmaken van taken en een view voor het bewerken van taken. Conventioneel worden de views in Laravel 'blades' genoemd.

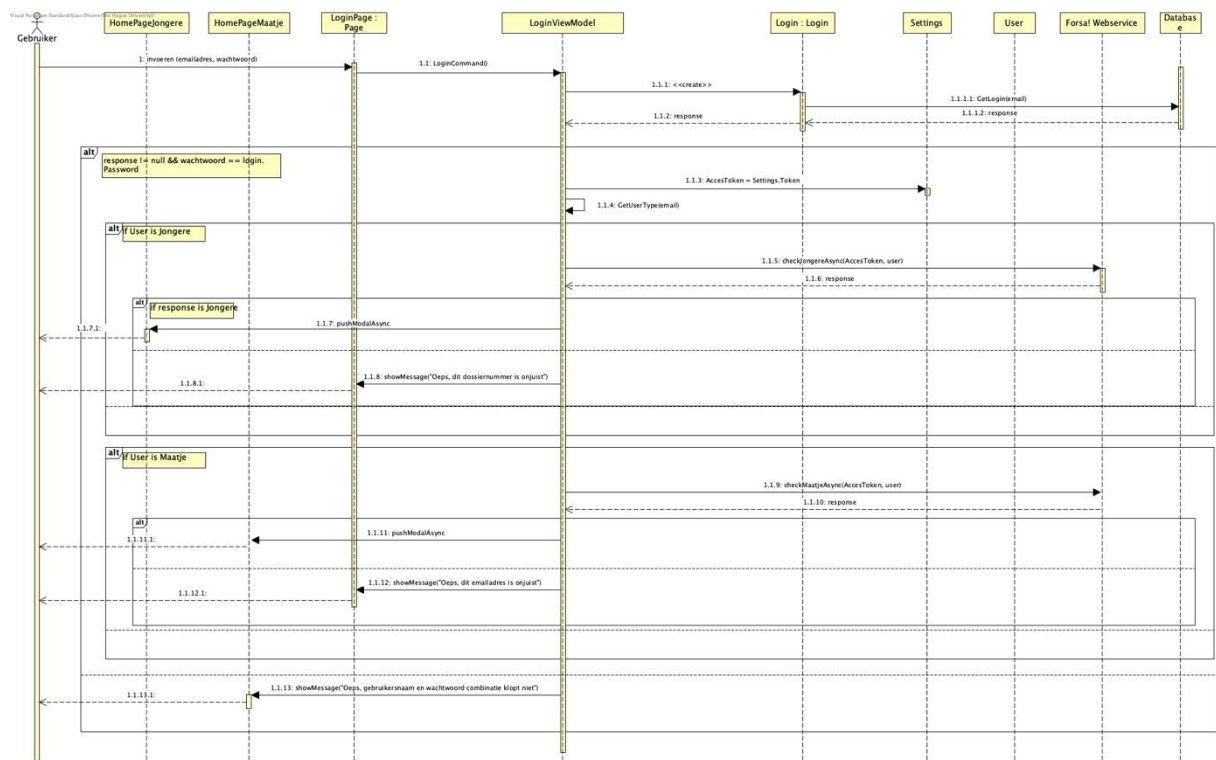
De controller is verantwoordelijk voor het uitvoeren van alle handelingen die mogelijk zijn. Zoals het creëren, bewerken, opslaan en verwijderen van een taak.



Figuur 19: Design klassendiagram van Registratiesysteem.Taak

Sequentiediagrammen

Voor het beschrijven van het gedrag in detail zijn sequentiediagrammen opgesteld. Hiervoor zijn use case beschrijvingen in combinatie met analyse klassen gebruikt om bijbehorende sequentiediagrammen op te stellen. Voor de beeldvorming zijn twee diagrammen in dit verslag opgenomen met een korte beschrijving ervan. Voor een overzicht van alle sequentiediagrammen wordt u verwezen naar het ontwerprapport.

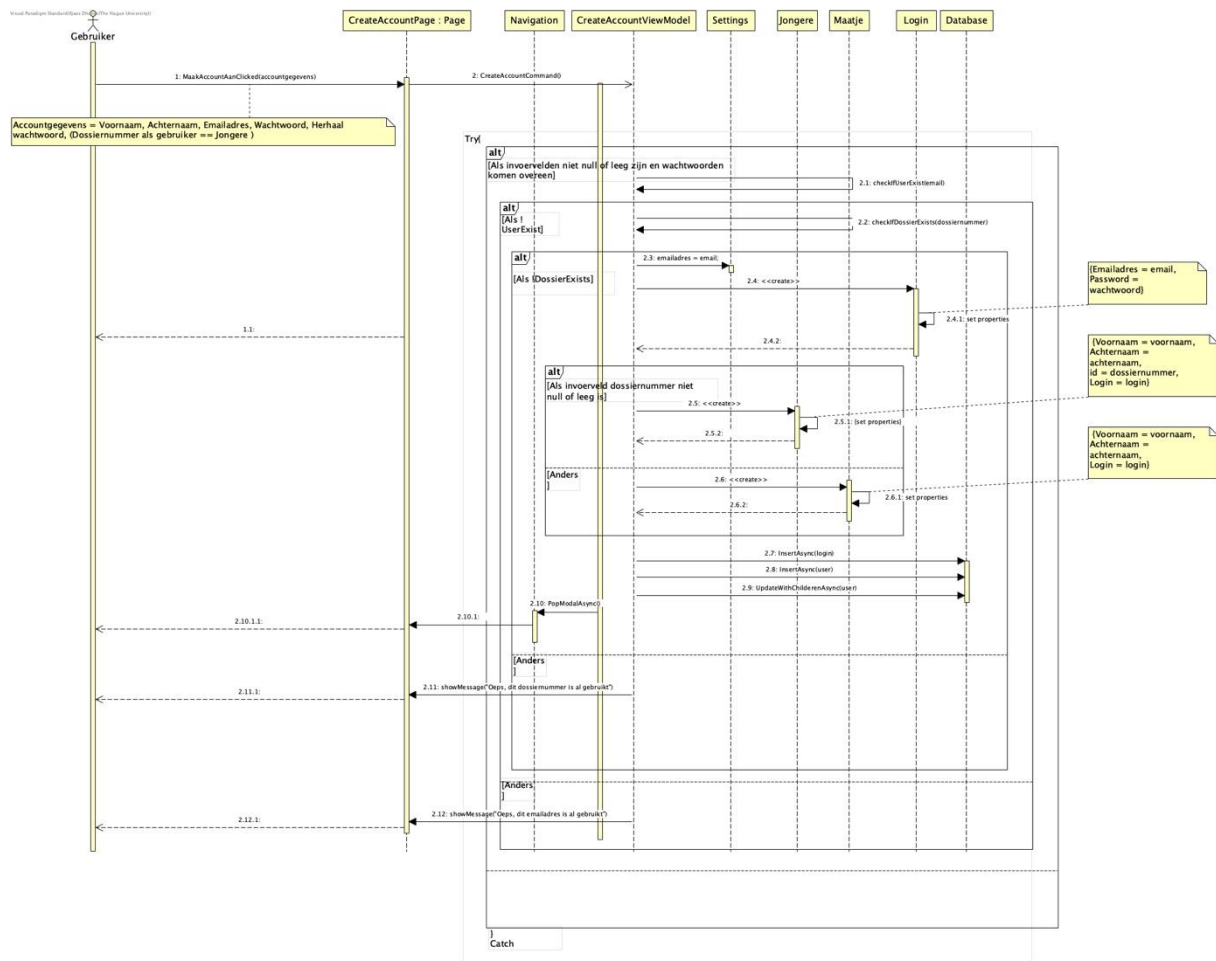


Figuur 20: Sequentiediagram UC1_Inloggen

Het inloggen gebeurt aan de hand van een emailadres en wachtwoord. Vervolgens wordt het LoginCommand asynchroon aangeroepen. Daarna wordt er gecontroleerd of het account bestaat in de database. Als het account bestaat wordt de token opgehaald en gekeken naar het type gebruiker. Als de gebruiker een jongere is wordt de checkJongereAsync(AccesToken, user) methode van de ApiService aangeroepen. Deze service geeft een response. Als de response een jongere is wordt het startscherm van de jongere getoond.

Als de gebruiker een maatje is wordt de checkMaatjeAsync(AccesToken, emailadres) aangeroepen. Als de response een maatje is wordt het startscherm van het maatje getoond.

Als de response onbekend is wordt een error getoond. Als er geen account gevonden wordt in de database wordt er eveneens een error getoond.



Figuur 21: UC21_MaakAccountAan

Bij het aanmaken van een account worden de voornaam, achternaam, emailadres, wachtwoord en herhaal wachtwoord opgegeven. In het geval van een jongere wordt ook het dossiernummer opgegeven. Als de invoervelden ingevuld zijn wordt en de wachtwoorden komen overeen wordt er gecontroleerd of het account al voorkomt in de database middels het emailadres. Als het account bestaat wordt er een error getoond. Anders wordt er gekeken of het dossiernummer al voorkomt in de database. Als het dossiernummer voorkomt in de database wordt er error getoond. Als het dossier niet voorkomt wordt er een inlog object aangemaakt en het emailadres wordt persistent in de Setting helper klasse opgeslagen. Als het invoerveld dossiernummer niet null of leeg is wordt er een account aangemaakt voor een jongere en het login object wordt hieraan gekoppeld. Als het dossiernummer invoerveld null of leeg is wordt er een maatje account aangemaakt en het login object wordt hieraan gekoppeld. Tot slot worden het login object en gebruikersaccount opgeslagen in de database met hun onderlinge relatie.

Databases

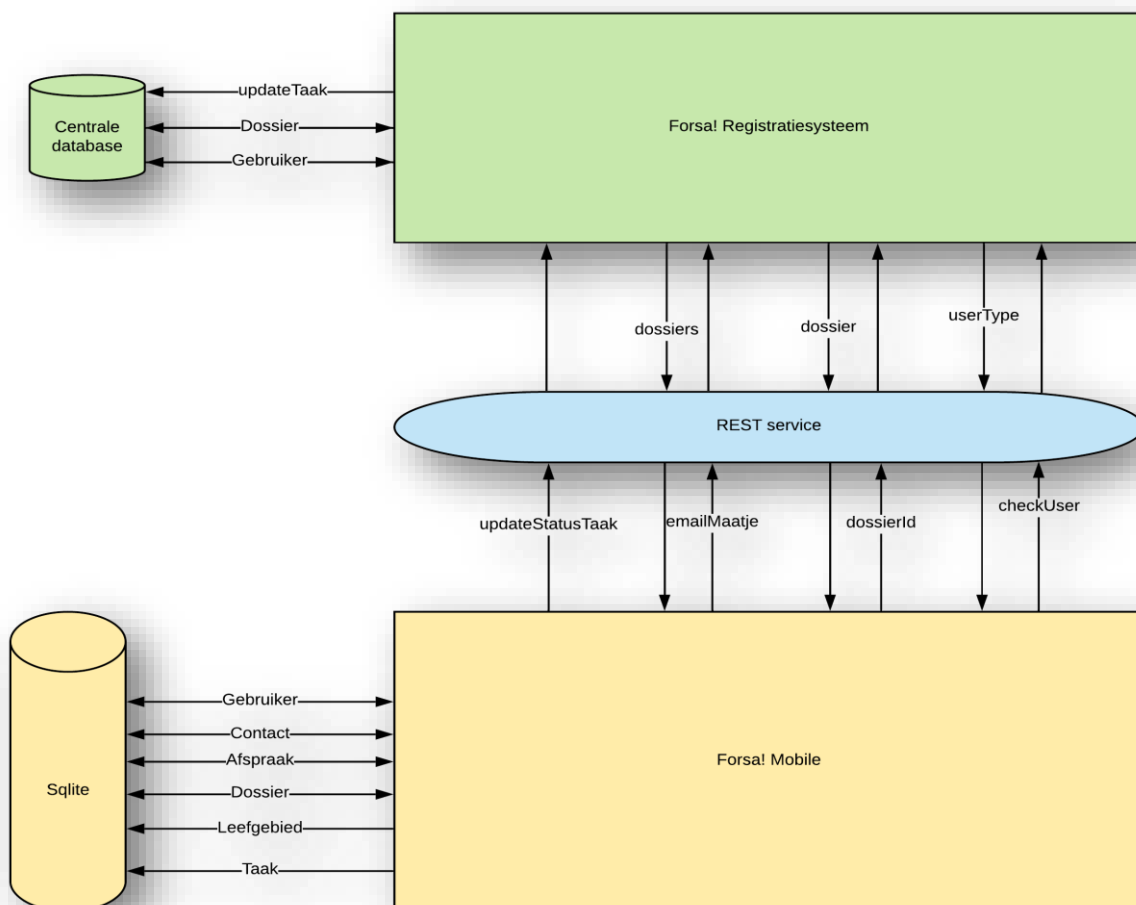
Het Forsa! registratiesysteem maakt gebruik van een centrale database voor het opslaan van de dossiers en gebruikers. De Forsa! Mobile app maakt gebruik van een lokale sqlite database voor het opslaan van inloggegevens, gebruikersgegevens, contacten, afspraken, dossiers, leefgebieden en taken.

Bij het koppelen van een login object aan een gebruiker is gebleken dat sqlite geen onderlinge relaties ondersteunt. Hierdoor is het niet mogelijk om middels het emailadres een gebruiker object op te halen. Na verder te zoeken zijn er een aantal alternatieven opgedoken. Dit zijn de Realm database en Sqlite extensions. Er is bij deze opdracht voor Sqlite extensions gekozen, omdat deze de mogelijkheden biedt voor het opslaan van objecten en onderlinge relaties.

De Realm database is een object georiënteerde database en vergt nader onderzoek voor de implementatie ervan. De Realm database is wat betreft snelheid en compactheid wel interessant om in de toekomst te gebruiken.

Bij het ontwerpen van de huidige centrale database is het probleem naar voren gekomen dat de entiteit 'Leefgebied' een attribuut is van andere entiteiten. Op deze wijze is het niet mogelijk om taken te koppelen aan leefgebieden. Om dit te bewerkstelligen moeten er aanpassingen gedaan worden aan de database. Deze worden verderop nader besproken.

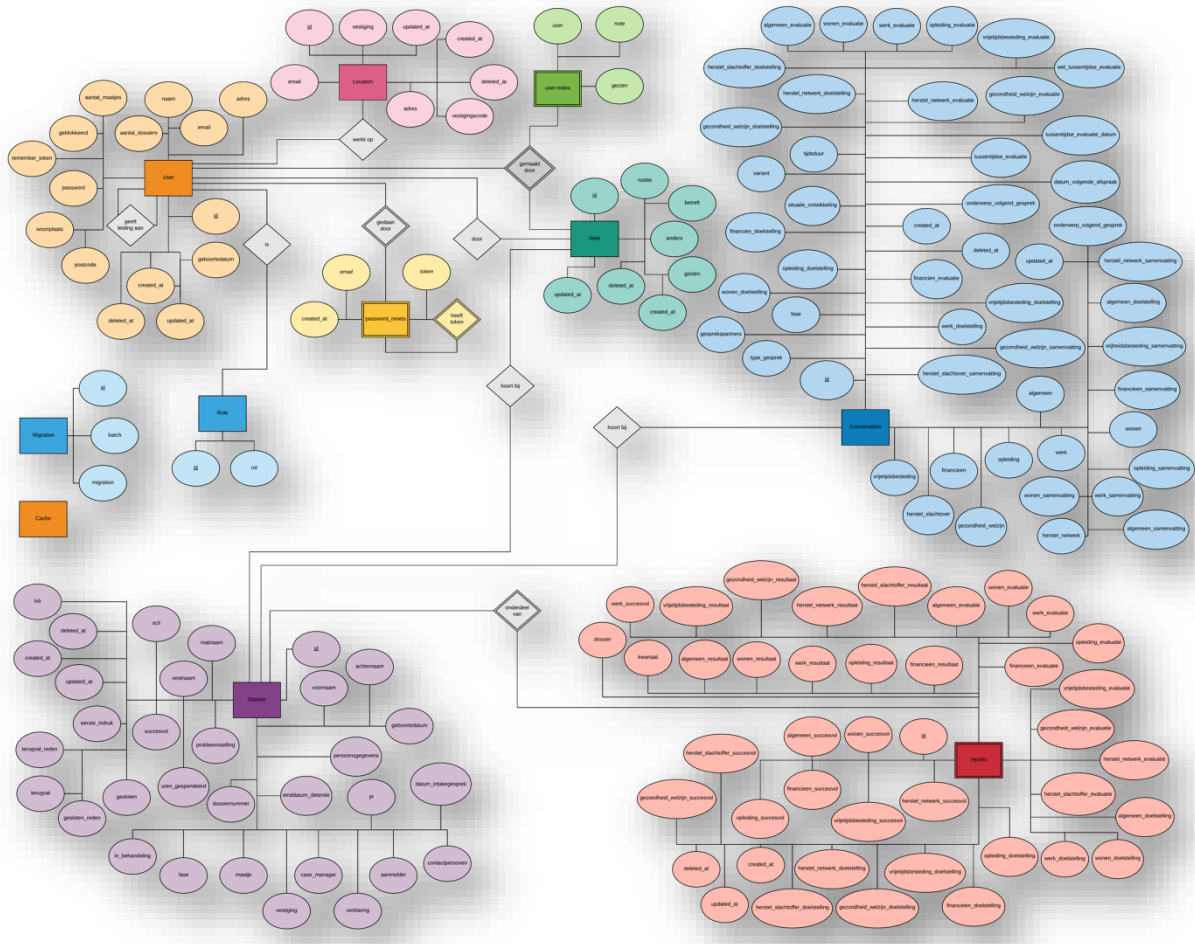
Hieronder is een diagram opgenomen waarin te zien is hoe de data tussen beide systemen verloopt. De REST service tussen het Forsa! Registratiesysteem en Forsa! Mobile app zorgt voor het afvangen van de requests en het opvragen van de juiste gegevens.



Figuur 22: Dataverkeer tussen Forsa! Mobile en het Forsa! Registratiesysteem

Huidige database

De huidige database is een relationele database welke gekoppeld is aan het registratiesysteem. Voor het modelleren van deze database zijn de inloggegevens van de phpMyAdmin omgeving overhandigd door de beheerder. Bij het modelleren hiervan is alleen een enhanced entity relationship (EER) diagram gebruikt, omdat dit voldoende inzicht geeft in de huidige situatie.



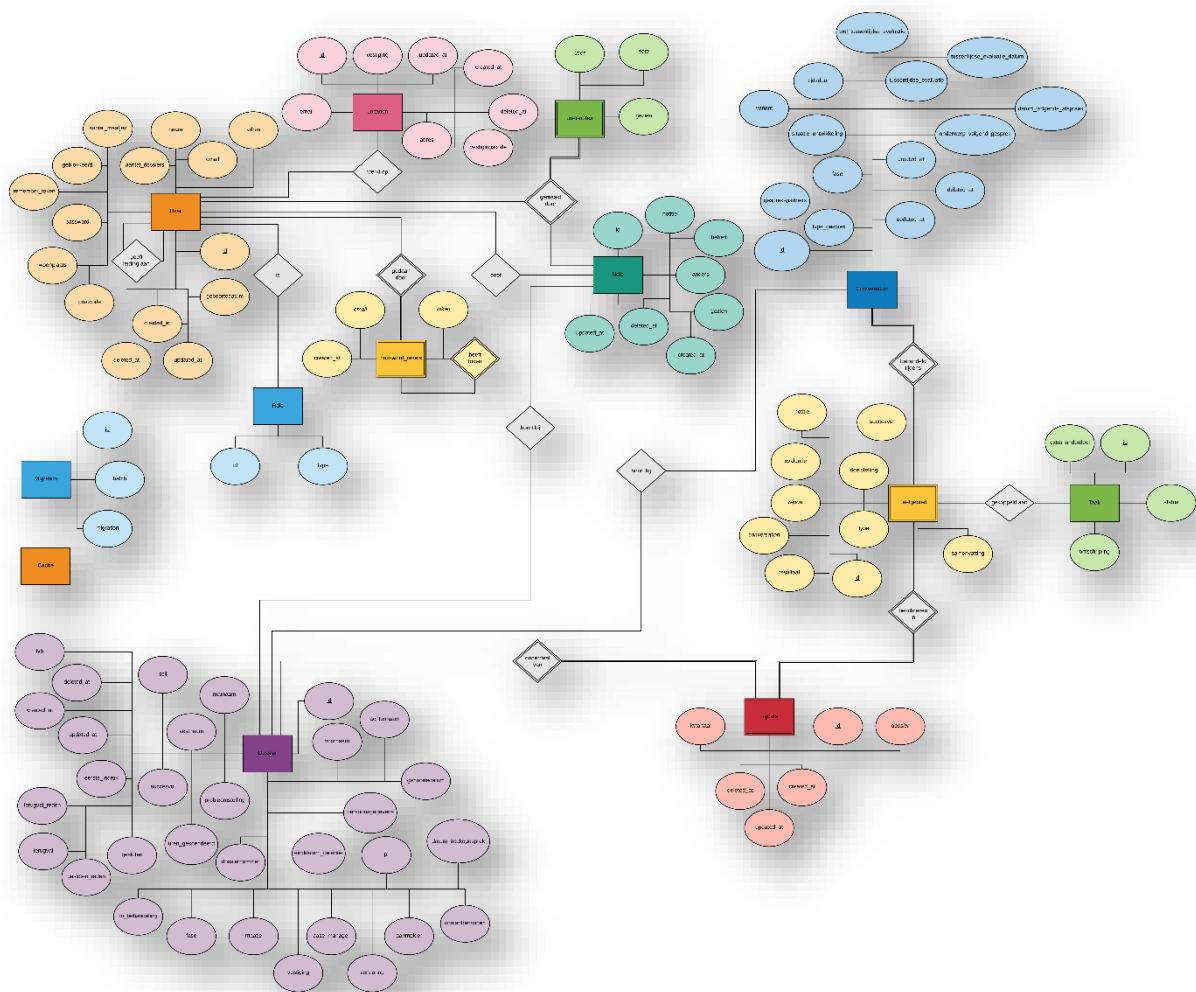
Figuur 23: Centrale database voor aanpassingen

Aanpassingen aan de database

Voor het toevoegen van taken aan leefgebieden in het registratiesysteem worden aanpassingen gedaan aan de database. Leefgebieden zijn in de huidige situatie gemodelleerd als attribuut van de tabellen 'conversation' en 'reports' zoals hierboven te zien is in de rode en blauwe entiteiten. In de aangepaste versie worden nieuw tabellen genaamd 'leefgebied' en 'taak' opgenomen. Leefgebied heeft vervolgens een relatie met 'conversation' en 'reports'. 'Taak' heeft een relatie met 'Leefgebied'.

Op basis van de requirements moeten taken gekoppeld worden aan leefgebieden. In het kader van 'good practice' en met het oog op aanpasbaarheid in de toekomst is 'Leefgebied' een aparte entiteit geworden met een koppeling naar de entiteiten waar deze een relatie mee heeft in plaats van 'Taak' als entiteit opnemen met een attribuut 'leefgebied'.

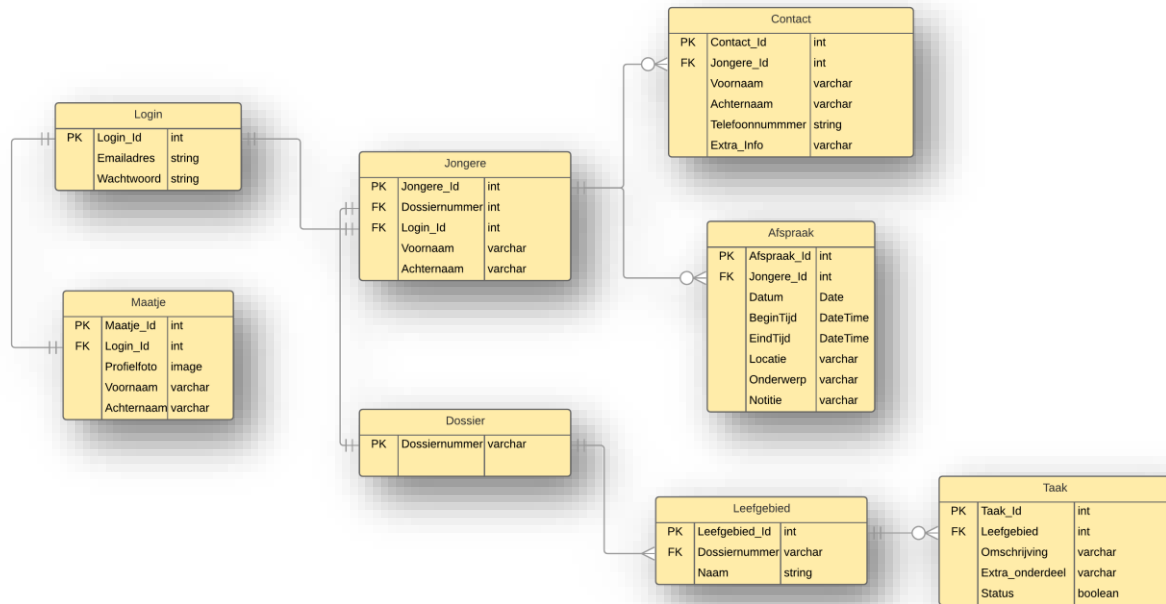
Voor taken is tevens een entiteit 'Taak' opgenomen in het aangepaste model met een relatie met 'Leefgebied'. Voor het modelleren van deze database zijn een EER, entity relationship diagram (ERD), relationeel representatie model (RRM) en relationeel implementatie model (RIM) ontworpen, omdat deze aanpassingen doorgevoerd worden in het registratiesysteem.



Figuur 24: Centrale database na aanpassingen

Lokale database

De lokale database wordt lokaal opgeslagen op de telefoon. Deze wordt gebruikt voor het opslaan van gebruiker specifieke gegevens en het beschikbaar stellen van de gegevens in de offline modus van de telefoon. Hieronder is een er-diagram opgenomen van alle objecten en hun onderlinge relatie.

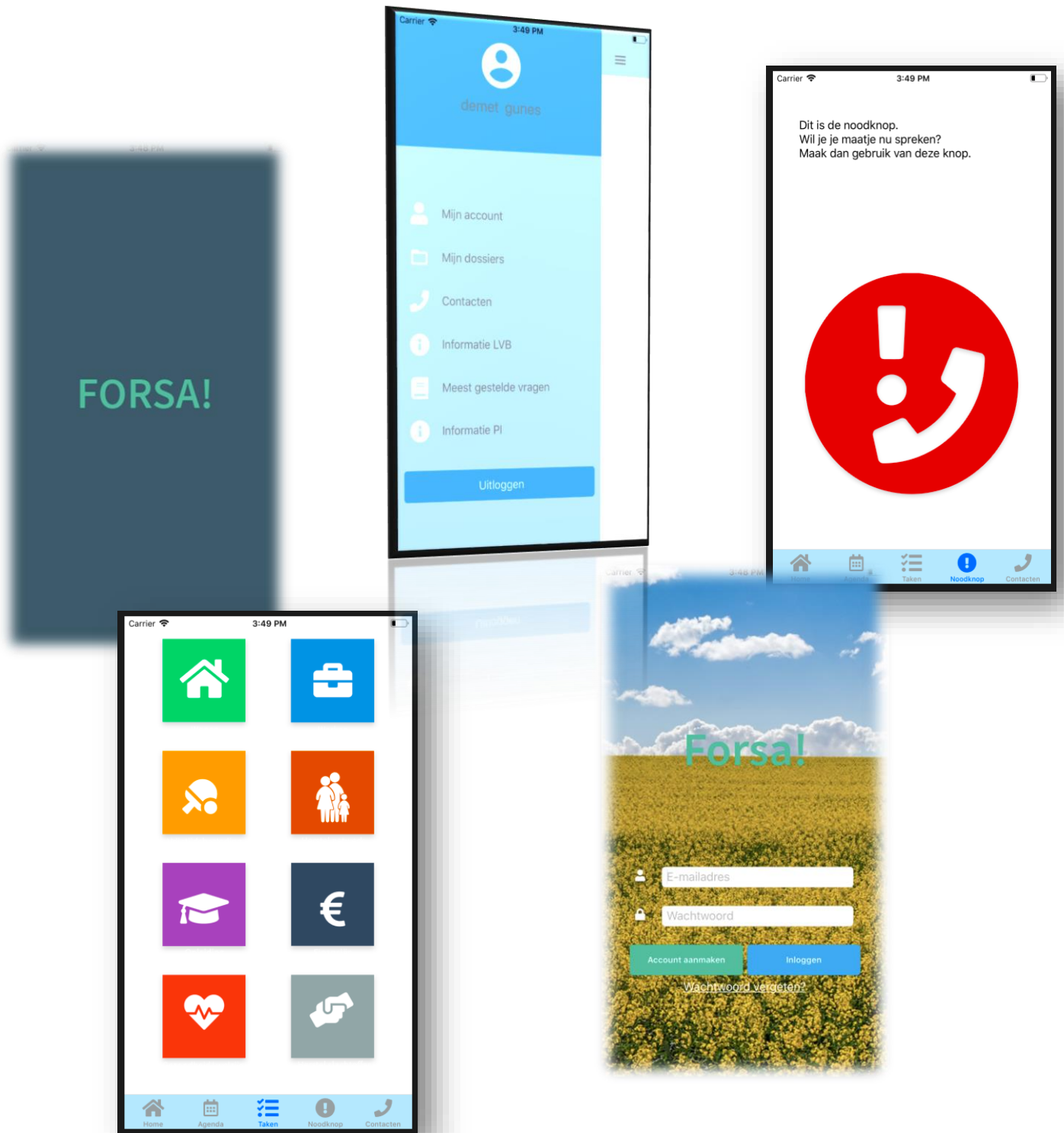


Figuur 25: Lokale database Forsa! Mobile

Ontwikkelfase

Mobiele applicatie

Op basis van het ontwerprapport worden de verschillende systemen ontwikkeld. Allereerst zijn de verschillende userinterfaces gebouwd op basis van het ontwerp van de CMD-studenten. Hieronder volgen een aantal screenshots van de app.



Na het ontwikkelen van de userinterfaces is de sqlite database ontwikkeld. Deze bestaat uit de tabellen Login, Maatje, Jongere, Contact, Appointment, Dossier, Leefgebied en Taak. De sqlite extension biedt een library aan vooraf ingestelde methodes voor het bevragen van de database, opslaan van gegevens en onderlinge relaties aangeven.

Tot slot is er overgegaan tot het bouwen van de functionaliteit van de app. Doordat er nog weinig bekend was over Xamarin.forms bij de afstudeerder zijn er heel wat uren gestoken in het lezen en bekijken van tutorials. De afgesproken deadline met de opdrachtgever van 25 maart is hierdoor in overleg ook verschoven naar 30 juni.

Nu de deadline van de scriptie er is zijn er nog wat punten die gebouwd moeten worden. Hierbij gaat het om de wachtwoord vergeten functionaliteit, het afvinken van een taak, het synchroniseren van notities tussen jongere en maatje en de voortgangsbalk.

Zaken die reeds af zijn zijn het aanmaken van een account, het inloggen als maatje of jongere, het beheren van afspraken, het beheren van contacten, het bekijken van taken, het bekijken van dossiers.

Een volledig werkende app zal uiteindelijk 30 juni opgeleverd worden aan de opdrachtgever.

API

Voor de API zijn vier endpoints gedefinieerd. Deze endpoints moeten aangeroepen worden en de response moet afgehandeld worden in de mobiel app. Hieronder ziet u een aantal afbeeldingen hoe dit bewerkstelligd wordt in de Forsa! Mobile app.

```
namespace Forsa
{
    public class ApiService
    {
        const string CHECK_USER_URI = "https://dev.forsa.bureaumhr.nl/public/api/v1/checkuser";
        const string GET_DOSSIER = "https://dev.forsa.bureaumhr.nl/public/api/v1/dossier/";
        const string GET_DOSSIERS_MAATJE = "https://dev.forsa.bureaumhr.nl/public/api/v1/dossiers/";
        const string UPDATE_TAAK_STATUS = "https://dev.forsa.bureaumhr.nl/public/api/v1/statusupdate/";
    }
}
```

Hierboven is een afbeelding van de endpoints naar de testomgeving. Om een connectie te maken met de live omgeving te maken moet het woordje 'dev' uit de url gehaald worden.

```
public async Task<string> CheckUserAsync(string token, Emailadres emailadres)
{
    httpClient = new HttpClient();
    httpClient.DefaultRequestHeaders.TryAddWithoutValidation("Authorization", token);
    var json = JsonConvert.SerializeObject(emailadres);
    HttpContent content = new StringContent(json, Encoding.UTF8, "application/json");
    HttpResponseMessage response = await httpClient.PostAsync(CHECK_USER_URI, content);
    var receiveStream = await response.Content.ReadAsStreamAsync();
    string user = (string)JsonConvert.DeserializeObject(receiveStream);
    return user;
}
```

Hierboven is de implementatie van het controleren of het maatje ook als maatje in het Forsa! Registratiesysteem staat genoteerd. De methode heet CheckUserAsync en neemt een string token en een

Emailadres object als parameters. Het antwoord van het registratiesysteem is vervolgens 'maatje' of 'onbekend'.

```
public async Task<string> CheckJongereAsync(string token, User jongere)
{
    httpClient = new HttpClient();
    httpClient.DefaultRequestHeaders.TryAddWithoutValidation("Authorization", token);
    var json = JsonConvert.SerializeObject(jongere);
    HttpContent content = new StringContent(json, Encoding.UTF8, "application/json");
    HttpResponseMessage response = await httpClient.PostAsync(CHECK_USER_URI, content);
    string receiveStream = await response.Content.ReadAsStringAsync();
    string user = (string)JsonConvert.DeserializeObject(receiveStream);
    return user;
}
```

De CheckJongereAsync methode is een request om na te gaan of de jongere ook als jongere in het Forsa! Registratiesysteem staat genoteerd. De methode neemt een token en een User object als parameters. Het antwoord van het registratiesysteem is 'jongere' of 'onbekend'.

```
public async Task<Dossier> GetDossierAsync(int dossiernummer, string accessToken)
{
    httpClient = new HttpClient();
    httpClient.DefaultRequestHeaders.TryAddWithoutValidation("Authorization", accessToken);
    HttpContent content = new StringContent("", Encoding.UTF8, "application/json");
    var json = await httpClient.PostAsync(GET_DOSSIER + dossiernummer, content);
    var receiveStream = await json.Content.ReadAsStringAsync();
    Dossier dossier = JsonConvert.DeserializeObject<Dossier>(receiveStream);
    return dossier;
}
```

De derde methode is GetDossierAsync en is bedoeld voor het ophalen van een dossier van een jongere. De methode neemt een dossiernummer en token als parameters en als antwoord wordt het dossier van de jongere teruggestuurd. Hierin zitten persoonsgegevens vermeld en de leefgebieden met bijbehorende taken.


```

public async Task<ObservableCollection<Dossier>> GetDossiersMaatjeAsync(string token, string emailadres)
{
    httpClient = new HttpClient();
    httpClient.DefaultRequestHeaders.TryAddWithoutValidation("Authorization", token);
    HttpContent content = new StringContent("", Encoding.UTF8, "application/json");
    var json = await httpClient.PostAsync(GET_DOSSIERS_MAATJE + emailadres, content);
    var receiveStream = await json.Content.ReadAsStringAsync();

    Portfolio portfolio = JsonConvert.DeserializeObject<Portfolio>(receiveStream);
    ObservableCollection<Dossier> dossiers = new ObservableCollection<Dossier>(portfolio.dossiers);
    return dossiers;
}

```

De laatste methode is de GetDossiersMaatjeAsync en haalt alle dossiers van een maatje op. Hierin staan eveneens alle persoonsgegevens, leefgebieden en taken van elke jongere in vermeld.

Systeemdeel Taak

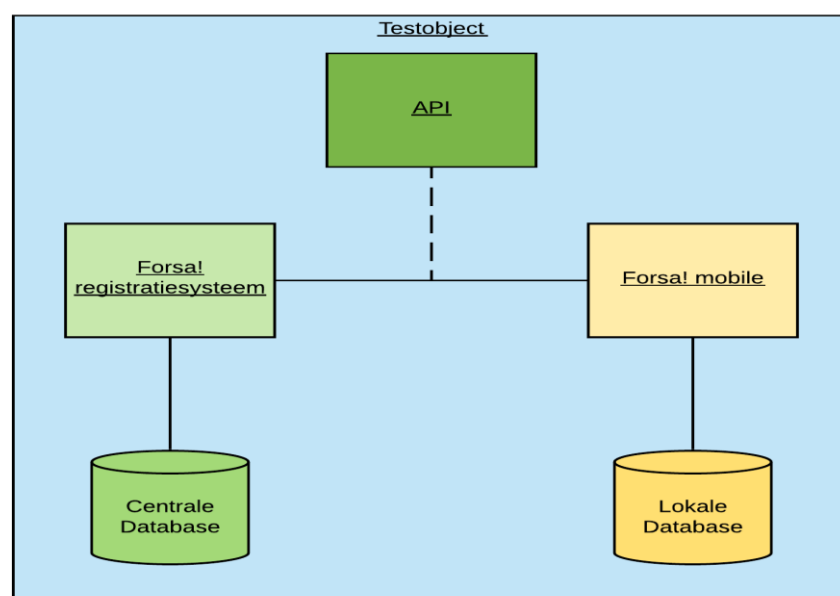
In beginsel was het nog onduidelijk hoe het systeemdeel dusdanig gebouwd moest worden om deze zonder al te veel aanpassingen te implementeren in het bestaande systeem. Uiteindelijk is er eerst gekeken naar het huidige systeem en in overleg met de ontwikkelaar gekeken naar de makkelijkste manier.

Het systeemdeel taak is dusdanig gebouwd dat het dezelfde conventie heeft als bestaande onderdelen van het registratiesysteem. Hierbij gaat het niet alleen om de MVC architectuur, maar ook de gebruikte jQuery framework voor het genereren van datatables bijvoorbeeld of het gebruikmaken van een repository tussen de database en de models.

Testfase

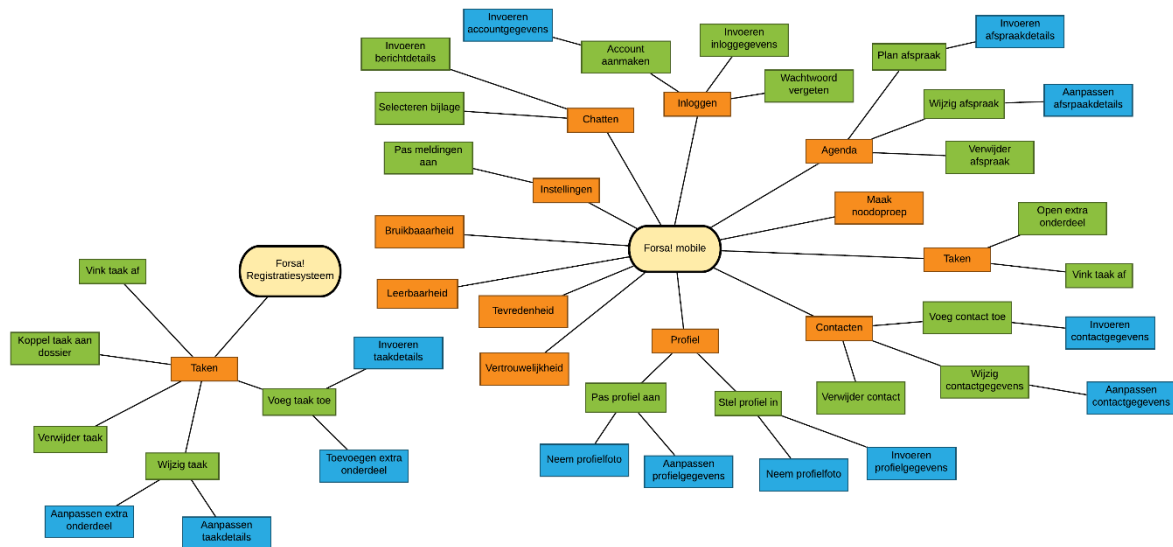
Het op te leveren product in deze fase is het testrapport. In dit rapport zijn een testplan en detailtestplannen (DTP) opgenomen.

In het testplan is het testobject opgenomen in de vorm van een contextdiagram. Het contextdiagram uit het analyserapport is niet toereikend als het gaat om de te testen systemen. In het analyserapport zijn de databases buiten beschouwing gelaten, omdat ze een intern onderdeel zijn van de twee systemen. Om deze reden is er een nieuw contextdiagram gebruikt voor het beschrijven van het testobject.



Figuur 26: Testobject

Op basis van de requirements en het use case diagram is een testboom gemaakt. Deze testboom is een mindmap van de systemen, functies en sub-functies.



Figuur 27: Testboom

Daarnaast is een testbasis beschreven. De testbasis is het geheel aan beschrijvingen en wordt als basis voor het testen gebruikt. Tevens is er een Testrisicoanalyse (TRA) opgenomen.

Voor de TRA is een 1-D TRA-sessie gehouden met verschillende stakeholder. Hiermee wordt duidelijk welke functies belangrijk zijn voor het bedrijf. Het belang van de functies is om na te gaan welke functies voorrang behoeven bij het testen. Zodat niet alle functies met dezelfde diepgang worden getest.

Bij de TRA-sessie worden de functies/risicogebieden uit de testboom in tabelvorm weergegeven. De stakeholders mogen vervolgens per functie om beurten een score van 1,3,5 of 9 punten geven met een maximum van 100 te vergeven punten. Tijdens de sessie blijkt dat elke functie van belang is voor de stakeholders. Aankaarten dat het om een relatief belang gaat en meerdere malen opnieuw beginnen met scoren heeft uiteindelijk geleid tot onderstaande 1-D test risicoanalyse.



Inloggen - Account aanmaken	9	9	9	9
Inloggen - Invoeren inloggegevens	9	9	9	9
Inloggen - Wachtwoord vergeten	9	9	9	9
Agenda - Plan afspraak	9	9	9	9
Agenda - Wijzig afspraak	9	9	9	9
Agenda - Verwijder afspraak	9	9	9	9
Maak noodoproep	9	9	9	9
Taken - Open extra onderdeel	9	9	9	9
Taken - Vink taak af	9	9	9	9
Contacten - Voeg contact toe	9	9	9	9
Contacten - Wijzig contact	9	9	9	9
Contacten - Verwijder contact	9	9	9	9
Profiel - Stel profiel in - neem foto	9	9	9	9
Profiel - Stel profiel in - invoeren profielgegevens	9	9	9	9
Instellingen - Pas meldingen aan	9	9	9	9
Bruikbaarheid	9	9	9	9
Leerbaarheid	9	9	9	9
Tevredenheid	9	9	9	9
Vertrouwen	9	9	9	9
	97	97	89	67

Figuur 28: TRA-sessie met stakeholders, v.l.n.r: Rachid Ben Meftah, Youssef Ouaicha, Tadyen Elsayed en Karima Daoudi.

Uit de TRA-sessie is het volgende uitgekomen,

Op 15 mei '19 is een risicoanalyse uitgevoerd door:

- Iljaas Dhonre, testcoördinator
- Karima Daoudi, opdrachtgever
- Rachid Ben Meftah, coördinator Forsa!
- Youssef Ouaicha, jongerecoach
- Tadyen Elsayed, maatje

De volgende productrisico's met hun relatief belang zijn erkend:

Risicocategorie	Risicogebied	Relatief belang
Kritisch	Taken-Voeg taak toe	324
	Inloggen-Account aanmaken	288
Hoog	Taken-Koppel taak aan dossier	288
	Inloggen-Invoeren inloggegevens	216
	Taken-Wijzig taak	216
	Taken-Vink taak af	216
Midden	Inloggen-Wachtwoord vergeten	144
	Taken-Open extra onderdeel	126
	Taken-Vink taak af	108
	Agenda-Plan afspraak	72
	Maak noodoproep	50
	Taken-Verwijder taak	45
Laag	Agenda-Wijzig afspraak	42
	Contacten-Voeg contact toe	36
	Agenda - Verwijder afspraak	24
	Profiel-Stel profiel in-Neem profielfoto	5
	Contacten-Wijzig contact	3
	Contacten-Verwijder contact	3
	Profiel-Stel profiel in-Invoeren profielgegevens	3
	Instellingen-Pas meldingen aan	1

Op basis van deze risicogebieden zijn de testscripts opgesteld en is de diepgang bepaald. In de testscripts staan de testgevallen en de mate waarin deze getest zullen worden. In verband met de deadline en prioritering van onderdelen zijn alleen de risicogebieden uit de risicocategorieën 'hoog' en 'kritisch' getest. De overige gebieden worden op een later moment getest.

Afwijkingen ten opzichte van het afstudeerplan

Bij het modelleren is de ontdekking gedaan dat leefgebieden niet zijn opgenomen als zijnde entiteiten, maar als attribuut van andere entiteiten waar leefgebieden een rol bij spelen. In overleg met de ontwikkelaar van dit systeem is naar voren gekomen dat hiervoor gekozen is, omdat leefgebieden als zijnde tekst worden opgeslagen en verder geen toegevoegde waarde hebben. Dit vertaalt zich door naar het registratiesysteem waarbij leefgebied alleen een stukje tekst is. Nu het een aparte entiteit is geworden dient het registratiesysteem in z'n geheel aangepast te worden. Dit valt buiten de scope van de opdracht en wordt opgepakt door de ontwikkelaar van het systeem en zorgt voor vertraging bij het koppelen van de systemen. In het afstudeerplan is ervan uitgegaan dat de opdracht in 17 weken afgerond zou zijn en het afstudeerverslag op 25 maart 2019 ingeleverd zou worden. Deze periode en deadline zijn na 5 weken verlengd met nog eens 10 weken waardoor de deadline is verplaatst naar 31 mei 2019.

Wijze van aantonen van de competenties/beroepstaken

1.4 Uitvoeren analyse door definitie van requirements

Het inventariseren, analyseren en beschrijven van de gewenste informatievoorziening is gedaan in de analysefase. Tijdens de stappen 1 'Positioneer het systeem binnen het businessdomein' en 2 'Definieer de gewenste oplossing' van het requirementsproces is dit veelvuldig aan bod gekomen.

Het inventariseren van de gewenste informatievoorziening is tijdens de brainstormsessie naar voren gekomen. Hier is duidelijk geworden waar de customer van de business tegenaan loopt en waar deze behoefte aan heeft. Het analyseren en beschrijven van de gewenste informatievoorziening is eveneens behandeld tijdens de brainstormsessie bij het bekijken van de voor- en nadelen van de gewenste oplossingen. Tevens is dit aan bod gekomen vanuit gesprekken en interviews met de stakeholders. Op basis hiervan zijn een contextdiagram en globaal use case diagram opgesteld. Daarnaast zijn de belangrijkste kwaliteitseigenschappen besproken en beschreven.

Analyseren, opstellen en beschrijven van de requirements is tijdens de stappen 2, 3 en 4 van het requirementsproces aan bod gekomen. Hierbij zijn er globale requirements opgesteld op basis van gesprekken en interviews. Deze zijn vervolgens geprioriteerd in samenspraak met de stakeholders.

Het specificeren en de samenhang zijn in de laatste stap behandeld. De samenhang van requirements is in de tabel aangegeven met een eigen kleur voor de requirements die bij elkaar horen. Tevens zijn ze traceerbaar gemaakt om de samenhang met de businessrequirements en use cases aan te geven.

Het toetsen van de requirements op volledigheid, duidelijkheid en consistentie is gedaan door deze meerdere malen met de stakeholders te bespreken tijdens periodieke werkoverleggen en nogmaals tijdens een validatiemoment te reviewen en bespreken.

De consistentie van requirements is bereikt door deze atomair op te stellen voor duidelijkheid en eenduidigheid. Door middel van de requirements traceabilitymatrix kan de consistentie met de andere requirements en use cases gecontroleerd worden. Doordat het linken van de requirements met use cases inzicht geeft in de requirements die overbodig zijn of use cases die nog niet gerepresenteerd worden door een set requirements.

2.1 Opstellen gegevensmodel voor een database

Deze beroepstaak komt terug in deze opdracht door het opstellen van een gegevensmodel voor een te ontwikkelen database (de lokale database) en uitbreiding/aanpassing van de bestaande database.

Als modelleringstechniek wordt een EER model gemaakt met als basis het klassendiagram.

2.2 Ontwerpen, bouwen en bevragen van een database

Voor deze beroepstaak zijn de gegevensmodellen, benoemd in de vorige beroepstaak, omgezet naar een implementeerbaar model. Dit model is geïmplementeerd in een relationeel DBMS van phpMyAdmin.

Voor het omzetten van het conceptueel gegevensmodel zijn een RRM en RIM opgesteld. Tevens is er gebruik gemaakt van timestamping (Updated_at) t.b.v. concurrency. Doordat er data tussen de mobiele applicatie en het registratiesysteem vervoerd wordt is het van belang dat deze up to date en correct blijft.

3.2 Ontwerpen systeemdeel

Het ontwerpen van de mobiele applicatie gebeurt met een design klassendiagram. Hierbij wordt rekening gehouden met de design structuur MVVM voor de applicatie en Singleton voor de lokale database. Er wordt een object georiënteerd ontwerpstrategie MVVM voor ogen gehouden door een duidelijke verantwoordelijkheid en scheiding van taken neer te leggen bij de juiste objecten.

Als ontwerpstechniek is UML gebruikt bij het modelleren van het design klassendiagram en lucidchart als case tool.

Voor het ontwerpen van het systeemdeel Taak is een MVC architectuur voor ogen gehouden met een repository pattern.

3.3 Bouwen applicatie

Het bouwen van de mobiele applicatie is gedaan met de programmeertools Visual studio. Hierbij is het framework Xamarin.forms gebruikt met C# als programmeertaal. Als versiebeheertools is Github gebruikt.

Voor het bouwen van het systeemdeel Taak is er rekening gehouden met de conventie waarin het registratiesysteem is gebouwd. Bij het bouwen van de mobiele applicatie moet er rekening gehouden met Ivb jongeren. Dit houdt in dat er veel met afbeeldingen en feedback gewerkt moet worden.

3.5 Uitvoeren van en rapporteren over het testproces

Voor het uitvoeren van deze beroepstaak zijn testscripts opgesteld en opgenomen in de detailtestplannen voor alle uit te voeren tests. Hierbij gaat het om unittesten met behulp van xUnit.net en end-to-end tests.

Bevindingen, dekkingsgraad en advies zijn in het testrapport opgenomen.

Bronnen/Literatuurlijst

- Arlow, J., & Neustadt, I. (2005). *UML 2.0 and the Unified Process: Practical Object-oriented Analysis and Design*. Upper Saddle River, Verenigde Staten: ADDISON WESLEY Publishing Company Incorporated.
- De Grood, D. J. (2008). *TestGoal: leerboek resultaatgedreven software testen*. Den Haag, Nederland: Academic Service.
- De Swart, N. (2010). *Handboek Requirements*. Delft, Nederland: Eburon Business.
- De Tré, G. (2007). *Principes van databases*. Amsterdam, Nederland: Pearson Education.
- Elicitatie - definitie - Encyclo. (z.d.). Geraadpleegd op 3 april 2019, van <https://www.encyclo.nl/begrip/Elicitatie>
- House of Control. (z.d.). Geraadpleegd op 19 november 2018, van <http://www.house-of-control.nl/valkuilen-softwareontwikkeling.html>
- Ketenpartners - definitie - Encyclo. (z.d.). Geraadpleegd op 3 april 2019, van <https://www.encyclo.nl/begrip/Ketenpartners>
- Kivit, J. (z.d.). Software-ontwikkeling: welke methoden zijn er en waarin verschillen ze? - Emerce. Geraadpleegd op 17 november 2018, van <https://www.emerce.nl/best-practice/software-ontwikkeling-welke-methoden-zijn-er-en-waarin-verschillen-ze>
- Van der Wardt, R. (2018, 25 november). Wat is een Google Design Sprint? (Nederlandstalige uitleg & tips per stap). Geraadpleegd op 3 april 2019, van <https://designthinkingworkshop.nl/google-design-sprint/>
- Wat is Kanban? Een volledige uitleg van de Kanban Methode | Agile Scrum Group. (2018, 4 april). Geraadpleegd op 20 november 2018, van <https://agilescrumgroup.nl/wat-is-kanban-methode/>
- Wikipedia-bijdragers. (2019a, 29 maart). Mock-up - Wikipedia. Geraadpleegd op 3 april 2019, van <https://nl.wikipedia.org/wiki/Mock-up>
- Wikipedia-bijdragers. (2019b, 29 maart). ISO 25010 - Wikipedia. Geraadpleegd op 3 april 2019, van https://nl.wikipedia.org/wiki/ISO_25010
- ZonMw. (z.d.). Geraadpleegd op 3 april 2019, van <https://www.zonmw.nl/nl/>

<http://www.raamstijn.nl/eenblogjeom/index.php/requirements/198-requirements-engineering>
<https://www.json.org/json-nl.html>
https://www.tutorialspoint.com/design_pattern/index.htm

Definities

Forsa!

Het programma en de methodiek van Bureau MHR voor het ondersteunen van de cliënten (ex-gedetineerden) bij hun herintreding en rehabilitatie.

Maatje

Een maatje is werkzaam bij Bureau MHR als de vertrouwenspersoon van de cliënten.

Cliënt

De cliënten van Bureau MHR zijn ex-gedetineerde jongeren met een niet westerse achtergrond.

Jongerecoach

Maatschappelijk medewerker werkzaam bij Bureau MHR voor het ondersteunen en begeleiden van de maatjes.

Coördinator

Beleidsmedewerker werkzaam bij Bureau MHR.

Registratiesysteem

Het registratiesysteem van Bureau MHR is een webapplicatie die gebruikt wordt om de dossiers van de cliënten bij te houden.

API

Application **P**rogramming **I**nterface is een verzameling definities waardoor een computerprogramma kan communiceren met een ander programma. In dit geval gaat het om de communicatie van de mobiele applicatie met het registratiesysteem.

OTAP-omgeving

Ontwikkel, **T**est, **A**cceptatie en **P**roductie omgevingen dienen ingericht te worden voor het ontwikkelen, testen en lanceren van het product. Hierbij zijn er respectievelijk drie omgevingen, de ontwikkelomgeving, testomgeving en productieomgeving. Bij acceptatie gaat het om goedkeuring van de stakeholders.

Kanban

Kanban is de gekozen softwareontwikkelingsmethode om dit project in goede banen te leiden. Volgens een artikel op agilscrumgroup.nl zit de kracht van Kanban 'm vooral in het onzichtbare (kenniswerk) zichtbaar maken, en in de kunst om niet alles tegelijk op te pakken.

IDE

IDE staat voor **I**ntegrated **D**evelopment **E**nvironment en is bedoeld voor het ontwikkelen van verschillende soorten applicaties. Voor deze opdracht worden PhpStorm en Visual Studio gebruikt.

Xamarin.forms

Een cross-platform userinterface toolkit voor .Net ontwikkelaars. Xamarin maakt het mogelijk om native apps te ontwikkelen voor meerdere platformen zoals Android en iOS tegelijk.

Baseline

Een set afgesproken en goedgekeurde requirements horende bij een specifieke release van het product.

Businessrequirement

Een verbetering in een (intern of door klanten uitgevoerd) proces of een nieuw proces die een belanghebbende uit de business met het systeem wil realiseren.

Gebruikersrequirement

Een proces (of subproces of taak of activiteit) die de gebruiker met behulp van het systeem wil uitvoeren.

Softwarerequirement

Gedrag dat het systeem moet vertonen om in een behoefte te voorzien van een belanghebbende uit de business.

Elicitatietechniek

Techniek voor het uitlokken van reacties of het onttrekken van moeilijk verkrijgbare informatie.

Google Design Sprint

Een Google Design Sprint is een intensief proces van een aantal dagen (vaak een werkweek) waarin een fundamenteel vraagstuk binnen een organisatie wordt opgelost.

RIC

Re-integratie Centrum voor ex-gedetineerde jongeren.

Mockup

Een mock-up is een tijdens de ontwerp- of productiefase op schaal of op ware grootte gemaakt model van een ontwerp of product.

Traject

Het traject van anderhalf jaar dat een cliënt van Bureau MHR doorloopt na detentie.

ISO-norm 25010

De ISO-norm 25010 vervangt sinds 2011 de ISO-norm 9126 en beschrijft de kwaliteitskenmerken van software.

MoSCoW prioritering

Een prioriteringstechniek van voor requirements. Het acroniem staat voor Must have, Should have, Could have, Won't have.

Ketenpartner

Persoon of organisatie, buiten de eigen organisatie, die een bijdrage levert aan de totstandkoming en of levering van het product of professioneel betrokken is bij de cliënt.

Actor

Een specificatie van een rol die een externe entiteit overneemt bij directe interactie met het systeem.

Binding

Een concept gehanteerd binnen de Xamarin.forms omgeving voor het koppelen van eigenschappen tussen twee objecten waarbij een van de objecten normaliter een userinterface object is.

ZonMW

Instelling die zich bezighoudt met het financieren van gezondheidsonderzoek én stimuleren van het gebruik van de ontwikkelde kennis om daarmee de zorg en gezondheid te verbeteren.

RUP

Rational Unified Process of RUP is een iteratief softwareontwikkelingsproces. Dit proces is een uitgebreide verfijning van het generieke Unified Process (UP) en is ontwikkeld door Rational Software Corporation dat nu een divisie is van IBM.

Singleton

Singleton is een ontwerp patroon om het aantal objecten van een bepaalde klasse tot één te beperken.

Responsieve webapplicatie

Webapplicatie waarvan de layout zich aanpast aan het scherm waar een gebruiker mee kijkt.

Native applicatie

Een applicatie die eerst op de telefoon geïnstalleerd dient te worden alvorens deze gebruikt kan worden.

Requirements engineering

Requirements engineering gaat over het definiëren, documenteren en onderhouden van requirements gedurende de levenscyclus van de ontwikkeling van een informatiesysteem.

JSON

JSON (JavaScript Object Notation) is een lichtgewicht formaat voor data-uitwisseling.

Design pattern

Design patterns staan voor 'best practices' gebruikt en ontwikkeld door ervaren object georiënteerde softwareontwikkelaars voor het oplossen van terugkerende problemen.