

Virtuele coach voor patiënten met hart- en vaatziekten

Door

Alan van Rossum
Wytze Elhorst

in gedeeltelijke vervulling van de eisen voor het diploma van

Bachelor of Science
in Computer Science

aan de Technische Universiteit Delft

Supervisors:	Coach:	Dr. W. P. Brinkman	TU Delft
	Client:	Dr. R. W. Treskes	LUMC
Bachelor project Coördinatoren:		Dr. Otto Visser	TU Delft
		Dr. Huijuan Wang	TU Delft

Inhoud

Inhoud	2
Samenvatting	4
Introductie	5
1.1 Structuur van dit Document	5
1.2 Ontwikkelingsmethode	5
Onderzoek	7
2.1 Overzicht	7
2.2 Probleemomschrijving en Analyse	7
2.3 Virtuele coach	8
2.3.1 Zelfzorg	8
2.3.2 Gebruik van virtuele coaches	9
2.3.3 Personalisatie	9
2.3.3 Ethische implicaties	9
2.4 Gesprekstechnieken	10
2.4.1. Fogg Method	10
2.4.2. Leary's Rose	11
2.5 Unity Talking Coach Framework	11
2.6 De Editor	12
Eisen	13
3.1 Overzicht	13
3.2 Ontwerpdoelen	13
3.2.1. Prestatie	13
3.2.2. Beveiliging	14
3.2.3. Onderhoud	14
3.3 Benodigdheden	14
3.3.1 Must haves	14
3.3.2 Should haves	14
3.3.3 Could haves	15
3.3.4 Won't haves	15
3.3.4 Samenvattend	15
3.4 Succes Criteria	16
Ontwerp	17
4.1 Overzicht	17
4.2 Problemen in het ontwerp	17
4.3 De Virtuele Coach	17

4.3.1 Initieel ontwerp	17
4.3.2 Uiteindelijk ontwerp	19
4.4 De Editor	20
4.4.1 Initieel ontwerp	20
4.4.2 Uiteindelijk ontwerp	21
Implementatie	23
5.1 Overzicht	23
5.2 De Coach	23
5.2.1 index.html	24
5.2.2 Map modules en Map Images	24
5.2.3 Map home	25
5.2.4 Map Build en Map TemplateData	25
5.3 De Editor	25
5.3.1 De Objecten	26
5.3.1 De Interface	27
5.4 Koppeling editor naar Coach	28
Testen en code kwaliteit	29
6.1 Overzicht	29
6.2 Testen	29
6.3 Code Kwaliteit	30
Evaluatie Eindproduct	31
7.1 Overzicht	31
7.2 Evaluatie van de benodigdheden	31
7.3 Evaluatie van de ontwerpdoelen	33
7.4 Aanbevelingen	34
7.5 Conclusie	34
 Appendix	
A - Handleiding Editor	36
B - Info Sheet	40
C - Project Omschrijving	42
D - Feedback SIG	44
E - XML voorbeeld	45
Referenties	50

Samenvatting

Het LUMC, het Leids Universitair Medisch Centrum, verzocht om een virtuele coach te ontwikkelen voor patiënten met hart- en vaatziekten. De coach zou deze patiënten thuis kunnen helpen om hen aan hun medicatie te houden. Daarvoor zou de patiënt de coach dagelijks kunnen opzoeken om zo feedback en motivatie te ontvangen van de coach. Ook zou de coach moeten kunnen helpen met veranderingen in levensstijl bevorderen.

Na veel onderzoek gedaan te hebben naar dit probleem en de benodigdheden hiervoor, zijn de volgende twee producten als oplossing bedacht. Ten eerste is er een webapplicatie gemaakt waarbij een virtuele 3d coach vragen stelt en beantwoordt over verschillende onderwerpen. Ten tweede is er een editor in Java gemaakt. Met deze editor is het mogelijk om de verschillende onderwerpen en vragen van de coach te creëren en aan te passen. Deze methode staat toe dat experts op het gebied van hart- en vaatziekten de inhoud van de coach kunnen schrijven. Ook zullen hierdoor veranderingen in medicatie gemakkelijk doorgegeven kunnen worden aan de coach.

Introductie

Het Leids Universitair medisch centrum (LUMC) omschrijft zichzelf als volgt [9]:

“Het LUMC is een modern universitair medisch centrum voor onderzoek, onderwijs en patiëntenzorg met een hoog kwaliteitsprofiel en een sterke wetenschappelijke oriëntatie. Met een unieke research praktijk van fundamenteel bèta medisch tot toegepast klinisch onderzoek, speelt het LUMC mee in de wereldtop. Zo kan het LUMC ook patiëntenzorg en onderwijs aanbieden naar de laatste internationale inzichten en standaarden. En bijdragen aan de verbetering van de geneeskunst en gezondheidszorg in eigen huis en daarbuiten.” Een belangrijk aspect van het LUMC is daarom om de patiënten ook in hun eigen huis te kunnen helpen, met als doel dat de patiënten zelfstandig hun medicijngebruik en levensstijl op goed peil kunnen houden.

Patiënten met hart en vaatziekten krijgen op dit moment een doos met boekjes en handleidingen mee, waarin ze informatie kunnen vinden over de pillen die zij moeten slikken en hoe zij hun levensstijl kunnen verbeteren. Het blijkt echter dat deze informatie vaak vergeten of genegeerd wordt en dat patiënten na een tijdje zonder toezicht stoppen met het nemen van medicijnen. Het LUMC wil nu dus een virtuele coach hebben waar de patiënt dagelijks naar toe gaat, om zo het gevoel te hebben met een echte dokter te praten.

1.1 Structuur van dit Document

In de eerste paar weken van het project is veel onderzoek gedaan naar virtuele coaches en het gebruik van deze. In hoofdstuk 2 staat alles wat wij tijdens deze onderzoeksfase hebben gevonden, zowel als de ideeën van ons en het LUMC en limitaties die deze met zich mee brachten. In hoofdstuk 3 zullen de doelen en benodigdheden van dit project worden neergezet. Hoofdstuk 2 en 3 vormen samen een aangepaste versie van het oorspronkelijke onderzoeksverslag. In hoofdstuk 4 staat alles over het uiteindelijke ontwerp waarvoor gekozen is en hoofdstuk 5 zal de implementatie-details hierover bevatten. Hoofdstuk 6 gaat over de manieren waarop ons programma is getest. In hoofdstuk 7 wordt gereflecteerd op de voorwaardes en ontwerpdoelen die tijdens het plan van aanpak en de onderzoeksfase zijn opgezet. Ook zullen hier aanbevelingen staan voor vergelijkbare projecten in de toekomst en zal een conclusie over het gehele project worden getrokken.

1.2 Ontwikkelingsmethode

In de eerste weken is nagedacht over de beste manier van software ontwikkelen. Aangezien er in een duo wordt gewerkt verloopt alle communicatie snel en gericht. Omdat er al vroeg

een demo werd verwacht en omdat er veel verschillende functies moesten worden toegevoegd is er gekozen voor een losse combinatie van SCRUM en Feature-Driven Development.

Het multidisciplinaire deel van SCRUM is niet gebruikt, omdat dat niet van toepassing was op het duo van informatici. Wel is er gezorgd voor dagelijkse communicatie en wekelijkse afspraken. Zowel SCRUM als Feature-Driven Development werken initieel naar een eerste 'overall model' toe. In het geval van de virtuele coach werd in de eerste weken de visuele weergave van het programma bedacht en overlegd met de opdrachtgever en TU Coach. Nadat hier enigheid over was ontstaan werd de eerste versie ontwikkeld in HTML en JavaScript.

Vervolgens werd met wekelijkse iteraties gewerkt aan het implementeren van de besloten functies, volgens het model van Feature-Driven Development.

Te zijner tijd werd ook besloten om af te stappen van vooraf gemaakte specifieke modules. Doordat het creëren van de editor pas later in het project begon, werden het ontwerpen en het ontwikkelen van de software vrijwel parallel uitgevoerd. Dit heeft er aan de ene kant tot geleid dat er een paar keer code herschreven moest worden, maar zorgde er aan de andere kant wel voor dat er gelijk een houvast was voor een editor die al snel in gebruik kon worden genomen.

2

Onderzoek

2.1 Overzicht

Dit hoofdstuk beschrijft al het onderzoek wat gedaan is tijdens dit project. Het grootste deel van het onderzoek vond plaats tijdens de eerste paar weken van het project. Dit hoofdstuk is later nog aangepast om nieuw onderzoek over te editor toe te voegen. In 2.2 zal naar de probleemstelling worden gekeken en hieruit wordt een duidelijk einddoel van het project bepaald. In 2.3 zal het goed gebruiken van een virtuele coach worden onderzocht. In 2.4 worden gesprekstechnieken onderzocht die de coach moet kunnen gebruiken. In 2.5 wordt naar het geleverde unity framework onderzocht om zo de mogelijkheden hiervan te bepalen. Tenslotte zal in 2.6 het gebruik en de mogelijkheden van een editor worden onderzocht.

2.2 Probleemomschrijving en Analyse

Tijdens een gesprek met onze cliënt van het LUMC, R. W. Treskes, werd het volgende probleem ter sprake gebracht:

Wanneer een patiënt met hart- en vaatziekten, zal deze ten eerste door een dokter zelf te horen krijgen wat er precies allemaal mis is en wat hieraan gedaan kan worden. De patiënt krijgt dan tot wel 5 verschillende medicijnen mee. De patiënt krijgt dan de volgende taak: Blijf deze medicijnen nemen en streef naar een gezondere levensstijl.

Om de patiënten te helpen zich aan deze taak te houden geeft het LUMC de patiënten een doos mee. In deze doos zitten boekjes met informatie over de medicatie en hoe men een gezondere levensstijl kan behalen. Ook zitten er enkele apparaten in de doos, zoals bloeddrukmeters, de patiënt kan deze gebruiken om zelf de staat van zijn hart en vaten in de gaten te houden.

Wat blijkt is dat deze informatie vaak maar één keer wordt gelezen en men deze dan langzaam weer vergeet. Daarnaast zijn de medicijnen bedoeld om continue ingenomen te worden om risico's te verkleinen en niet om pas ingenomen te worden wanneer de patiënt symptomen voelt aankomen, want dan is het eigenlijk al te laat. Hoe nadrukkelijk dit ook duidelijk wordt gemaakt op het LUMC en in de boekjes, vroeg of laat is de kans groot dat de patiënt toch stopt met het innemen van deze medicijnen. Daarnaast zorgen de medicijnen ook soms voor nare bijwerkingen, dit maakt het alleen maar minder aantrekkelijk voor de patiënt om ze te blijven innemen.

Het probleem is dus dat patiënten met hart- en vaatziekten er vaak niet in slagen om zich thuis aan hun medicatie te houden.

Een verandering in levensstijl kan net zo belangrijk zijn als het innemen van medicijnen. Zo kan alleen al stoppen met roken een groot effect hebben op de gezondheid van je hart [10]. Ook gezonde voeding en regelmatige lichaamsactiviteit hebben een goed effect op de gezondheid [11]. Zorgen dat men zo'n gezondere levensstijl adopteert en aanhoudt kan echter extreem lastig zijn. Zelfs in de lichtste gevallen is er een verandering in gedrag nodig, waarvoor ook de houding veranderd moet worden. In ergere gevallen, zoals rokende patiënten, moet deze ook nog van een verslaving afkomen. Een doos met informatie is daarom ook voor het bevorderen van een betere levensstijl niet voldoende [8]. Het LUMC heeft als visie dat deze patiënten thuis dagelijks gecoached worden door een virtuele coach. Deze coach moet de patiënt het gevoel geven dat zij met een echte dokter communiceren, dit zou namelijk een positief effect hebben op de motivatie van de patiënt [3]. Zo zal deze coach feedback geven op de acties van de patiënt, de patiënt helpen te herinneren om medicijnen in te blijven nemen en informatie geven over deze medicijnen wanneer de patiënt dit niet meer weet. Ook zal deze coach de patiënt steunen in het veranderen van zijn levensstijl door middel van advies en feedback. Het uiteindelijke doel van de coach is om gedragsverandering in de patiënt te stimuleren door middel van dagelijkse gesprekken. Door het menselijke aspect van de coach hoopt het LUMC dat de patiënt eerder geneigd is om zich te houden aan het advies van de coach en om deze dagelijks te bezoeken, de patiënt zou namelijk het gevoel krijgen dat hij zich moet verantwoorden tegen de coach [5].

Het einddoel van dit project is om een virtuele coach te leveren aan het LUMC die gebruikt kan worden als basis voor het behalen van de bovenstaande visie.

2.3 Virtuele coach

In dit deel zullen belangrijke aspecten die present moeten zijn in onze virtuele coach worden onderzocht. Hiervoor zal worden gekeken naar het gebruik van virtuele coaches op vergelijkbare gebieden en, gebruikmakend van de resultaten van onderzoeken, zullen de doelen, aanpak en werking van onze coach worden gevormd.

2.3.1 Zelfzorg

Een belangrijk doel van de coach is het bevorderen van de zelfzorg voor patiënten en zal daarom goed toegankelijk moeten zijn voor elke patiënt. In veel gebieden van geneeskunde blijkt namelijk dat meer betrekking van de patiënt veel voordelen met zich meebrengt [7]. Er komt daarom steeds meer nadruk te liggen op de zelfzorg. De zelfzorg doelt erop om patiënten thuis zelf te activeren om gezonder te worden. Zo probeert het de patiënt te motiveren en kennis te geven over gezond leven. Ook moet de coach het gevoel geven dat er een echt gesprek met een mens plaatsvindt. Uit onderzoek blijkt namelijk dat, wanneer alleen gelaten, zelfs patiënten met genoeg kennis over gezond leven en de gevolgen van ongezond leven uiteindelijk zich vaak toch niet meer aan hun behandeling houden en terugvallen op oude gewoontes [8]. Voor effectieve zelfzorg zullen de patiënt continu gemotiveerd moet worden door externe bronnen.

Het LUMC krijgt veel patiënten met verschillende achtergronden [9], deze patiënten zullen ook niet allemaal Nederlands spreken. Om te zorgen dat de coach ook deze mensen kan

helpen in de zelfzorg is het nodig dat de coach in meerdere talen beschikbaar is. Zo kunnen ook mensen zonder beheersing van de nederlandse taal de coach optimaal gebruiken.

2.3.2 Gebruik van virtuele coaches

Uit 2.3.1 blijkt dat het motiveren van de patiënt een belangrijk element is van de zelfzorg. Het gebruik van een coach om patiënten met medische problemen te motiveren heeft positieve gevolgen, dit blijkt uit een onderzoek waarbij een virtuele coach werd gebruikt om mensen met ruggengraat beschadiging te helpen [3]. Deze coach informeerde de patiënten en gaf advies over hoe patiënten het beste konden handelen om serieuze bijwerkingen te voorkomen. Uit dit onderzoek bleek dat de coach een positieve invloed had op de motivatie en gezondheid van de patiënt. De onderzoekers en patiënten zagen de coach als een goede manier om zelfzorg te bevorderen.

Daarnaast blijkt het ook dat virtuele coaches een goed medium zijn om mensen te helpen met gedragsverandering en het informeren van de patiënt over hun conditie, zo is er een onderzoek [5] uitgevoerd waar mensen met overgewicht een online dagboek invulden over hun levensstijl waarbij een avatar feedback en informatie gaf. Uit het onderzoek blijkt dat door de avatar mensen vaker hun dagboek invulden, hun motivatie tot verbetering minder snel afnam, en uiteindelijk meer lichaamsgewicht verloren. Ook zorgde het ervoor dat men meer kennis had over het behouden van een gezonde levensstijl.

Het menselijke aspect van de coach brengt ook voordelen met zich mee, wanneer dit op een realistische manier wordt afgebeeld. Zo is er onderzoek gedaan naar het gebruik van een 'conversational agent' om de communicatievaardigheden van tieners met autisme te verbeteren [2]. Deze conversational agent gaf op een menselijke manier feedback, hierdoor bleken de proefpersonen meer op te steken van de gesprekken, omdat het als een echt gesprek voelde.

2.3.3 Personalisatie

Een ander belangrijk punt van onze coach is de personalisatie. Als de coach persoonlijk wordt voor patiënten, zoals het onthouden van informatie uit vorige sessies, zal dit een zeer positief effect hebben op het behouden van interesse in de coach. Het verband tussen personalisatie van de coach en het behouden van interesse in de coach blijkt uit een onderzoek uit 2013 [6]. In dit onderzoek liet men kinderen met diabetes omgaan met een neutrale en een gepersonaliseerde robot. Deze robots converseerde met de kinderen en deed spellen en quizen met ze. Uit het resultaat bleek dat de kinderen na een bepaalde tijd hun interesse verloren in de neutrale robot. De gepersonaliseerde robot bleek echter even interessant te blijven en bleek ook beter kennis door te geven aan de kinderen. De kinderen begonnen zelfs de robot na te doen.

2.3.3 Ethische implicaties

Het gebruik van virtuele coaches brengt echter ethische implicaties met zich mee. Zo zullen virtuele coaches de baan van menselijke coaches kunnen innemen, zo valt het gebruik van virtuele coaches onder de robotisatie en automatisatie van banen [20]. Nu zal dit project op

zichzelf nog niet voldoende zijn om banen over te nemen van mensen, maar is dit project wel een stap in de richting van het universele gebruik van virtuele coaches en dus de automatisering van de zorg. Het gevolg hiervan kan zijn dat er in de toekomst minder werkplekken in de zorg nodig zijn. Aan de andere kant [20] wordt beargumenteerd dat het aantal banen niet zal afnemen door automatisatie, maar dat er ander soort banen nodig zullen zijn. Daarnaast zal de automatisatie in de zorg een hoop extra mogelijkheden kunnen bieden aan patiënten en kunnen deze bijvoorbeeld op elk moment worden gemonitord [21].

Automatisatie wordt vaak als iets onvermijdelijks gezien en iets waar de samenleving zich maar aan moet aanpassen [20]. Daarnaast brengt automatisatie in de zorg veel voordelen met zich mee voor patiënten [21]. Hierdoor kan geconcludeerd worden dat het gebruik van virtuele coaches en dus ook dit project ethisch verantwoord is.

2.4 Gesprekstechnieken

Elk gesprek dat een patiënt kan hebben met een arts of dokter heeft een doel. Bijvoorbeeld informeren, overtuigen of iets aanleren. De manier waarop zo'n doel het beste kan worden bereikt hangt af van het doel zelf. Daarom is het belangrijk dat de software meerdere gespreksvormen kan ondersteunen. Zoals een arts op verschillende manieren gesprekken kan hebben met een patiënt, zal de arts deze zelfde technieken ook moeten kunnen verwerken in het gedrag van de virtuele coach.

Op het moment dat deze gesprekstechnieken onderzocht werden, was het plan om de tekst van de coach te schrijven als deel van het project en daarbij de onderstaande gesprekstechnieken te gebruiken. Later in het process is besloten om niet zelf de tekst van de coach te schrijven, maar om een editor te maken, zoals gespecialiseerde dokters en psychologen de inhoud van de coach kunnen maken. Als gevolg hiervan zullen deze gespreksmethodes niet direct in dit project worden toegepast, maar zal er wel rekening mee worden gehouden dat de coach in staat moet zijn om deze gesprekstechnieken te gebruiken. Verder kunnen deze gesprekstechnieken ook functioneren als inspiratie voor de persoon die uiteindelijk de tekst van de coach zal schrijven.

2.4.1. Fogg Method

De methode van Fogg is een universele methode ontworpen door BJ Fogg [12]. Hij beschrijft in zijn werken de beste manier om gedragsveranderingen te veroorzaken. Hij staat achter een systematische methode van drie stappen, ieder kort en concreet.

Stap 1: Wees specifiek. Het moet duidelijk zijn voor de gebruiker wat het gewenste gedrag is dat hij of zij moet vertonen. Om dit te doen moeten de doelen, zoals dagelijks medicijnen innemen, worden vertaald naar gedrag, zoals een stabiel en consistent dagritme hebben. De coach zal dus concrete opdrachten geven aan de patiënt en duidelijke feedback geven zodat de patiënt weet of het wel of niet goed gaat.

Stap 2: Maak het makkelijk. Hoe makkelijker je het uitvoeren van het gewenste gedrag maakt, des te makkelijker de overstap naar het nieuwe gedrag wordt. Voor de virtuele coach betekent dit niet alleen dat de inhoud van de gesprekken met de coach een makkelijke overgang garanderen en de coach dus geen onrealistische doelen stelt, maar ook dat de software makkelijk te gebruiken is. Door alleen de nodige knoppen weer te

geven, en een overzichtelijk en herkenbaar beeld te creëren, zullen gebruikers minder bezig zijn met het leren kennen van het programma en meer met het daadwerkelijke gesprek met de virtuele coach.

Stap 3: Zet het gedrag in gang. Gedrag gebeurt niet zonder 'trigger'. Een trigger kan natuurlijk zijn, zoals de koelkast open doen wanneer je door de keuken loopt. Een trigger kan ook extern zijn, bijvoorbeeld een alarm. Een succesvolle gedragsverandering heeft plaatsgevonden als het gewenste gedrag door natuurlijke triggers wordt uitgevoerd. Dus in plaats van een wekkertje die de patiënt elke dag herinnert medicijnen in te nemen, is het doel om bijvoorbeeld de inname te koppelen aan een andere natuurlijke gebeurtenis. Een voorbeeld is het aanleren aan de patiënt om altijd na het tandenpoetsen aan de in te nemen medicijnen te denken.

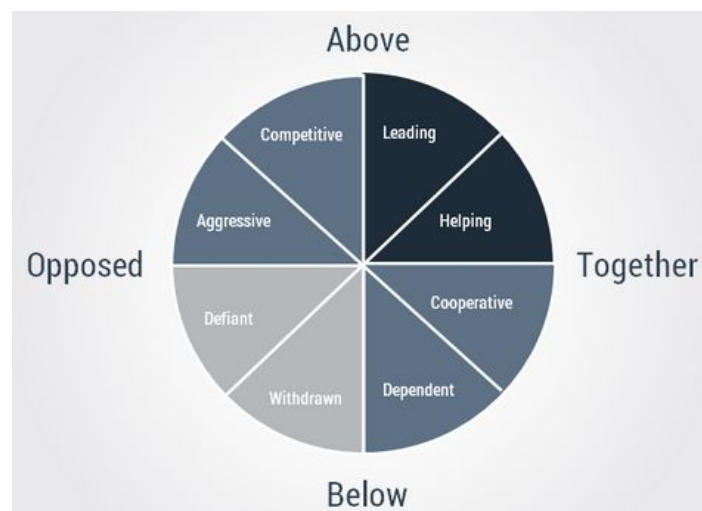
De coach moet dus concrete en realistische opdrachten en duidelijke feedback kunnen geven en het aanvullen en aanpassen van de bestaande levensstijl met nieuwe handelingen.

2.4.2. Leary's Rose

De roos van Leary[zie afbeelding 1] is een diagram dat wordt gebruikt in strategisch communiceren[13].

Timothy Leary deelde het menselijk gedrag op in vier categorieën. De theorie van Leary's Rose is dat bepaald gedrag ander gedrag uitlokt.

Dit diagram kan gebruikt worden als een extra richtlijn voor gedragsverandering. De eerste stap is het bepalen van het gewenste gedrag van de patiënt. De gedragsverandering zal vaak van afhankelijk naar coöperatief moeten gaan.



Figuur 1: Leary's Rose

Zo zal de patiënt uiteindelijk op eigen benen kunnen staan zodra hij of zij de software vaak genoeg heeft gebruikt. Volgens Leary moet de aanpak van zulke patiënten dus niet alleen leidend, maar ook vooral helpend zijn. De virtuele coach moet dus naast een leidinggevende bron van informatie te zijn ook de patiënt op persoonlijk niveau helpen met toekomstige problemen en vragen, opdat de patiënt met samenwerkend gedrag reageert.

2.5 Unity Talking Coach Framework

Om een pratend 3d model van de coach te verkrijgen moest het al bestaande Talking Coach framework [13] worden gebruikt. Dit framework is geschreven in unity en bevat 3d modellen

van coaches met verschillende achtergronden. Via javascript calls kan dit programma met behulp van de ingebouwde text-to-speech van de browser de coach laten praten. Het programma zal dus dit unity framework moeten kunnen implementeren en vervolgens moet het met javascript de coach kunnen laten communiceren. Verder kunnen ook nog de additionele opties van het framework worden gebruikt, zoals het veranderen van het uiterlijk van de coach, of de achtergrond.

2.6 De Editor

In de eerste paar weken van het project was de veronderstelling dat er een enkel product moest worden afgeleverd, namelijk een virtuele coach die patiënten met hart- en vaatziekten begeleidt en coacht. Er werd al onderzoek gedaan naar de medicijnen die de patiënten gebruikten, de informatie waar ze al over beschikten en wat er precies nodig was om deze informatie goed aan te vullen. De cliënt zou er tevreden mee zijn als we het project met zo'n coach zouden afronden.

Echter, na meer onderzoek en overleg bleek dat dit toch niet de optimale aanpak was. Het grootste nadeel was namelijk hoe weinig dynamisch het resultaat zou zijn. Er vindt regelmatig verandering plaats in medicatie en behandelmethodes [17], de coach loopt daarom het risico om al snel gedateerd te worden, wanneer deze niet wordt aangepast. Het zou voor de cliënt onmogelijk worden om zonder extra technische handleidingen de inhoud van de coach aan te passen. Verder zou de inhoud, een gesimuleerd gesprek met een medisch expert, niet van voldoende kwaliteit zijn, tenzij deze inhoud werd opgesteld door een daadwerkelijk medisch expert.

Daarom is besloten om over te stappen op een nieuw plan. Het schrijven van de inhoud van de modules, de vragen en de gesprekken zou worden overgelaten aan de cliënt zelf. Om dit te verwezenlijken moest het voor mensen zonder technische achtergrond mogelijk zijn om de inhoud aan te passen. De oplossing hiervan is een Editor die het project kan creëren, openen, aanpassen, en weer uitschrijven.

De Editor zal gebruikt moeten worden door één of meerdere experts bij het LUMC. Dit betekent dat de editor minder restricties heeft wat betreft gebruiksvriendelijkheid dan de coach. Desalniettemin zal de editor gebruikt moeten kunnen worden door mensen zonder enige programmeer achtergrond. Dit betekent dat de editor vanzelfsprekend genoeg moet zijn dan iemand, na enig onderzoek over de functies van de editor, direct aan de slag kan.

De Editor is in feite een vorm van 'Content Management System' [18] die het mogelijk maakt om mensen zonder technische kennis toch de inhoud van een technische applicatie te laten bewerken. De editor moet het mogelijk maken om de modules van de coach in te laden, op te slaan, te bewerken en compleet nieuwe modules moeten toegevoegd kunnen worden [18].

Voor het opslaan en laden van de modules moet informatie worden opgeslagen in een XML-bestand. XML is een manier om informatie op te slaan, te identificeren en over te zetten [15].

XML brengt als voordelen met zich mee dat het een gratis en zeer bekende manier is van data opslag, het makkelijk te lezen en schrijven is voor programma's en het goed uitbreidbaar is [15].

3

Eisen

3.1 Overzicht

In dit hoofdstuk zullen, op basis van het onderzoek van hoofdstuk 2, doelen en benodigdheden worden opgesteld. In 3.2 worden verschillende doelen bedacht. In 3.3 zullen alle benodigdheden via MoSCoW [19] worden gerangschikt.

3.2 Ontwerpdoelen

3.2.1. Prestatie

De prestatie van het eindproduct is een belangrijk ontwerpdoel. De prestatie van de applicatie kan op twee manieren opgesplitst worden, de inhoud en de interface. Verder is het eindproduct ook nog op te delen in de coach en de editor.

Inhoud

De coach moet de al bestaande informatie over hart- en vaatziekten weten over te brengen. Daarbij moet het deze informatie op een menselijke manier via een dialoog kunnen leveren. Verder moet de coach naast de informatie ook de patiënt blijvend kunnen motiveren, de patiënt moet dus een reden hebben om herhaaldelijk de coach op te zoeken.

De editor moet in staat zijn om de inhoud van de coach te kunnen creëren. De editor moet deze inhoud kunnen omzetten naar de web-applicatie zonder dat de gebruiker hiervoor hoeft te programmeren.

Interface

De interface moet door iedere patiënt die beschikt over een computer gebruikt kunnen worden. Dit betekent dat de interface intuïtief en gebruiksvriendelijk moet zijn. Een perfecte coach is alsnog niet bruikbaar wanneer de patiënten niet begrijpen hoe ze met de coach kunnen communiceren. Daarbij moet de interface goed uitbreidbaar zijn, zodat veranderingen in de interface makkelijk pasbaar zijn.

De editor moet door medische specialisten gebruikt kunnen worden. De interface moet daarom een goed overzicht geven van de structuur van de modules, maar er kan wel vanuit worden gegaan dat de gebruiker van de editor enig onderzoek heeft gedaan naar het gebruik. Verder hoeft de interface van de editor niet uitbreidbaar te zijn, want er kan vanuit worden gegaan dat deze niet hoeft worden veranderd.

3.2.2. Beveiliging

Er komen veel risico's kijken het behouden van de privacy van de cliënten. Een doel van de applicatie is namelijk om de cliënt een persoonlijke hulp te bieden, dit betekent dat de applicatie bepaalde medische informatie moet onthouden om cliënten zo persoonlijk mogelijk te kunnen helpen. Dit is echter erg gevoelige informatie en valt volgens de Wet bescherming persoonsgegevens (Wbp) dan ook onder bijzondere persoonsgegevens. Wettelijk is het (exclusief bepaalde uitzonderingen) verboden om bijzondere persoonsgegevens te verwerken of delen. De applicatie moet daarom goed genoeg beveiligd zijn om te kunnen garanderen dat de informatie secuur blijft. Wanneer dit niet het geval is, zal deze applicatie niet gebruikt kunnen worden door het LUMC.

3.2.3. Onderhoud

In de medische wereld worden elk jaar nieuwe technieken en medicijnen ontdekt, waardoor behandelmethodes erg kunnen veranderen. Om de coach dus langer dan een paar maanden relevant te houden, is het belangrijk dat de inhoud van de coach makkelijk aan te passen is, ook door mensen met weinig of geen programmeerervaring. Hier is de editor voor nodig. Ook moeten ontwikkelaars onze code makkelijk kunnen begrijpen en aanpassen, zodat de applicatie uitgebreid kan worden wanneer dat nodig is. Dit geldt beide voor de coach en de editor.

3.3 Benodigdheden

Het eindproduct zal aan bepaalde voorwaarden moeten voldoen. De functionaliteit van de applicatie moet worden afgebakend om een duidelijke omschrijving te krijgen van wat het product wel en niet zal hebben. Met behulp van de MoSCoW-methode [19] wordt het eindproduct beschreven.

3.3.1 Must have's

De must-have's zijn de absolute kernpunten van de applicatie, die hoe dan ook in het eindproduct moeten plaatsvinden.

De basis van het product is een web-based interface waarin de gebruiker kan communiceren met een virtueel persoon. De virtuele coach is een geanimeerd 3D personage afkomstig van de geleverde unity framework. De coach moet met de patiënt kunnen converseren met behulp van gesprekstechnieken. Ook moet het eindproduct makkelijk aan te passen zijn door middel van de editor. De editor moet het creëren van modules en het omzetten ervan naar javascript mogelijk maken.

3.3.2 Should have's

De should-have's zijn zeer gewenste eigenschappen van het product, die niet strikt noodzakelijk zijn voor het functioneren van de applicatie.

Allereerst is ondersteuning voor meerdere talen een optie die zeer gewenst is. Zo kunnen gebruikers die geen of slecht Nederlands spreken de applicatie ook gebruiken. Ten tweede is het van groot belang dat de coach ook daadwerkelijk gepersonaliseerd is. Zo kan de coach informatie uit eerdere sessies onthouden, en zal de coach de gebruiker niet telkens

weer om zijn naam en gewoontes hoeven vragen. Dit bevordert de relatie tussen gebruiker en coach. Vereist voor deze functionaliteit is dat er een centrale database wordt bijgehouden, met daarin entries voor gebruikers en een paar van hun kenmerken, zoals rookgewoonten, leeftijd, eetgewoontes en eventuele andere klachten. Naast het simpelweg vertellen wat de gebruiker moet doen, zoals medicijnen innemen, zal de coach ook over een uitleggende functie beschikken. Zo kan de gebruiker vragen naar extra informatie over medicijnen en gevolgen van veel roken en ongezond eten en drinken. De editor zou bestaande modules moeten kunnen inladen en deze zou hij moeten kunnen aanpassen, dit zorgt ervoor dat modules niet steeds opnieuw gemaakt hoeven te worden als er slechts kleine veranderingen gemaakt hoeven worden.

3.3.3 Could have's

De could-have's zijn functionaliteiten die aan bod komen mits er genoeg tijd is. Als er tijd over is na het implementeren van alle must- en should-have's, zal er allereerst worden gekeken naar een vorm van een smartphone-cliënt. Dit hoeft niet per se een volledige kopie van het web-based framework te zijn. De eerste variant zal een app zijn die een dagelijkse herinnering notificatie stuurt naar de eigenaar. Deze notificatie zal de gebruiker herinneren om zijn medicijnen in te nemen en vragen een bezoek te brengen aan de website.

Andere functionaliteiten waaraan kan worden gewerkt zijn een geavanceerde coach, die naast het communiceren met tekst en spraak ook communiceert met handgebaren, gezichtsexpressies en andere lichaamstaal. Dit kan als resultaat hebben dat de coach een algemene positieve invloed heeft op de gesteldheid van de gebruiker.

Daarnaast zouden er ook nog verbeteringen van de stem van de coach kunnen plaatsvinden. Zo zou deze bijvoorbeeld ingesproken kunnen worden of komt er een optie om de stem te vertragen voor slechthorende patiënten. Dit zou de coach menselijker laten overkomen en hem beter verstaanbaar maken.

De editor zou elke module als een boom kunnen weergeven met de link tussen de vragen van de module als takken. Dit zou helpen bij het overzichtelijk maken van de modules.

3.3.4 Won't have's

De won't-have's zullen tijdens dit project niet aan bod komen, maar zijn interessant voor een vervolgproject. Als dit product een succes blijkt is de eerste stap, na het internationaal verspreiden naar andere ziekenhuizen, het customiseren van het product voor andere ziektes. Hart- en vaatziekten zijn namelijk niet de enige kwalen die kunnen worden bestreden met behulp van gedragsverandering en dagelijkse persoonlijke coaching. Door het product toe te passen op andere ziektes kunnen ook deze ziektes preventief worden bestreden.

Ook zou het uiterlijk van de coach kunnen worden verbeterd, zodat deze nog meer op een mens lijkt. Dit heeft tot gevolg dat het makkelijker is voor gebruikers om de coach te identificeren als persoon. Dit maakt het scheppen van een vertrouwensband makkelijker.

3.3.4 Samenvattend

Hieronder staat een lijst die alle benodigdheden samenvat.

Must have's:

- Web interface met een geanimeerde coach.
- Coach communiceert op een menselijke manier.
- Inhoud van de Coach moet gemaakt kunnen in de editor.
- De editor maakt het creëren van modules voor de coach mogelijk.

Should have's:

- De coach spreekt meerdere talen.
- De coach is gepersonaliseerd.
- De coach legt uit over de in te nemen medicijnen en vragen van de patiënt.
- De editor kan bestaande modules in laden en deze bewerken.

Could have's:

- Smartphone cliënt voor de web-applicatie.
- De Coach maakt gebruik van handgebaren en lichaamstaal.
- Verbetering van de stem van de coach.
- De Coach kan naast tekst ook plaatjes naar de persoon sturen.
- De editor geeft een visuele boom weer die de link tussen alle vragen van een module laat zien

Won't have's:

- Uitbreiding van coach voor andere ziektes en medicijnen.
- Het uiterlijk van de coach verbeteren en menselijker maken.

3.4 Succes Criteria

In dit deel zal een definitie worden bepaald voor wanneer het project als succesvol kan worden beschouwd. Hiervoor zullen de benodigdheden en ontwerpdoelen worden gebruikt als criteria. Voor een succesvol eindproduct moeten de ontwerpdoelen zeker gevolgd worden. Het eindproduct moet presteren zoals beschreven in 3.2.1. Ook moet het eindproduct goede beveiliging hebben, anders kan het niet gebruikt worden. Tenslotte moet het eindproduct ook goed onderhoud kunnen worden, zodat het product met de tijd mee kan gaan.

Van de benodigdheden moet het eindproduct per definitie alle 'must have's' bevatten om succesvol te zijn. Ook stellen wij dat het product een merendeel van de 'should have's' moet bevatten. De rest van de benodigdheden zullen optioneel zijn voor het slagen van dit project.

4

Ontwerp

4.1 Overzicht

In dit hoofdstuk wordt het ontwerp van beide kanten van het project weergegeven. Allereerst wordt in sectie 4.2 de problemen bij het ontwerpen beschreven. Daarna wordt in sectie 4.3 de ontwerpfase van de virtuele coach beschreven. Tot slot komt in sectie 4.4 de visuele weergave van de Editor aan bod.

4.2 Problemen in het ontwerp

Een groot probleem met het ontwerpen en implementeren van de coach was de privacy. Dit was het probleem van privacy en de Wet bescherming persoonsgegevens. Meer hierover is te lezen in hoofdstuk 2.5.2. De meest voorliggende oplossing van dit probleem is om gebruik te maken van de database server van het LUMC, het LUMC heeft namelijk al ervaring met het gebruiken van databases om zult soort informatie op te slaat [15]. Het gebruik van de server van het LUMC bleek echter niet mogelijk te zijn.

Een server van de TU Delft gebruiken was een andere optie. Hier zitten voor het LUMC echter ook nog risico's aan, want de medische informatie is dan alsnog verkrijgbaar door de TU. De applicatie kan echter wel tijdelijk deze servers gebruiken, zonder gebruik te maken van echte patiënten. Wanneer het LUMC deze dan wil gaan gebruiken voor echte patiënten kan de applicatie worden aangepast om gebruik te maken van de LUMC server. Dit zou de uiteindelijke overdracht echter lastig kunnen maken, want het project zou dan alsnog aangepast moeten worden om te werken op de LUMC servers.

Uiteindelijk zijn we tot de conclusie gekomen dat het het veiligst was om helemaal geen informatie op te slaan. In plaats van een 'slimme' coach die je vorige sessie onthoudt, zullen de modules zelf moeten beginnen met enkele vragen over de voortgang van de patiënt, om als het ware het geheugen van de coach op te frissen. Dit zal de personalisatie van de coach wel erg limiteren, maar het brengt verreweg de minste risico's met zich mee.

4.3 De Virtuele Coach

4.3.1 Initieel ontwerp

Voor het eerste ontwerp van het uiterlijk van de virtuele coach werd vooral nadruk gelegd op simpliciteit. De patiënten waar het LUMC mee te maken heeft zijn van alle leeftijden, en juist patiënten met hart- en vaatziekten zijn overwegend ouder. Daarom was het uiterst belangrijk

dat elke gebruiker in één oogopslag kan zien wat er allemaal voor functies zijn. Daarom hebben we voor een tweedeling gekozen, zie figuur 2.



Figuur 2: Voorkant van het initiële ontwerp

Aan de rechterkant is weer een tweedeling gemaakt met boven de 3D avatar van de coach, met daaronder algemene knoppen voor taalverandering en algemene gebruikersvragen. Zodra er aan de linkerkant een module zou worden geselecteerd zou de layout veranderen naar figuur 3.



Figuur 3: Inhoud van een module

Zoals te zien in het figuur verandert de taalselectie naar algemene vragen over de huidige module. De coach blijft op dezelfde plek, voor consistentie. Links is de inhoud van de

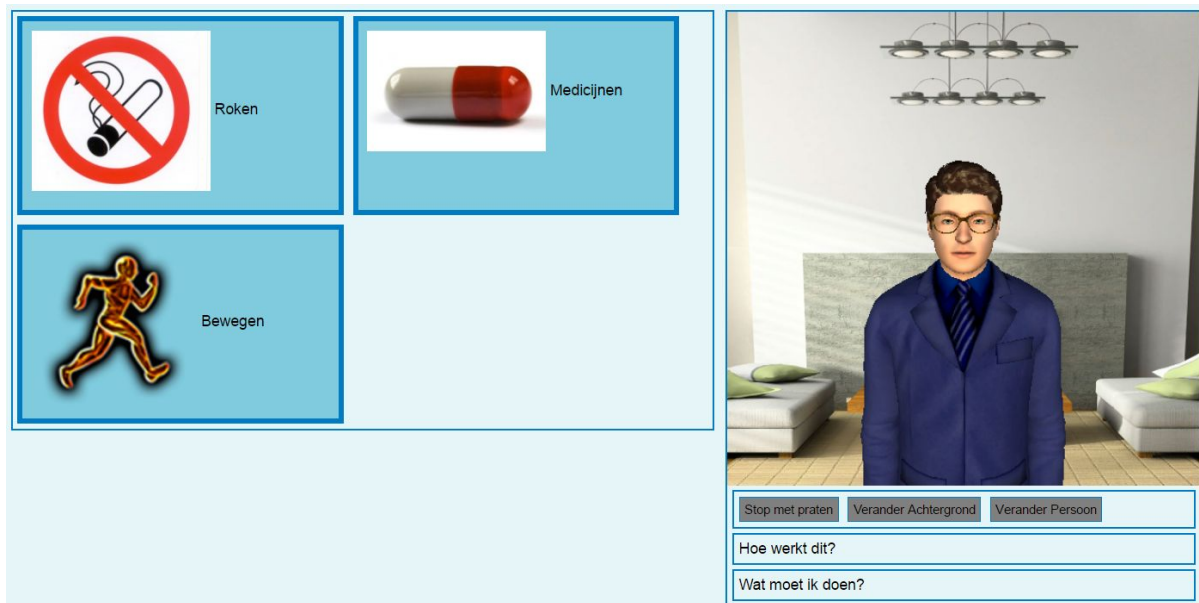
module te vinden, in het voorbeeld een bron van informatie over een medicijn, met knoppen om verder en terug te gaan.

De layout zoals te zien in deze figuren werd voorgelegd aan de TU Coach en de cliënt, die beiden genoeg aan te merken hadden. Er waren meerdere feedbackpunten die hebben geleid tot veranderingen in het ontwerp.

Ten eerste was de kleurkeuze van grijs en zwart te donker. Voor het vernieuwde ontwerp is gekozen voor een kleurenpalet van wit en lichtblauw, net zoals de website van het LUMC[9]. Ten tweede werd de inhoud van een module ondervonden als te chaotisch. Links bevindt zich te veel kleine tekst, en de knoppen zijn te groot en lelijk. De middenscheiding met links de inhoud en rechts de algemenere objecten is besloten te houden. Een belangrijke suggestie voor de vormgeving van het linkerdeel was het veranderen naar een chat-gespreksvorm. In plaats van grote stukken tekst en willekeurige knoppen, zou het mooi zijn als het meer zou worden weergegeven in een dialoogvorm. Dit zou de immersie versterken en de gesprekken met de virtuele coach persoonlijker maken.

4.3.2 Uiteindelijk ontwerp

Het uiteindelijke ontwerp lijkt qua indeling nog op het initiële ontwerp. Wel is de feedback doorgevoerd, en is alles iets gebruiksvriendelijker. Zie figuur 4 voor het uiteindelijke resultaat.



Figuur 4: Voorkant van het uiteindelijke ontwerp

Zoals te zien zijn de kleuren veranderd, de knoppen mooier en is de avatar een daadwerkelijk 3D-model in Unity. Zodra aan de linkerhand een module wordt geselecteerd gaat de gebruiker als het ware een module binnen. In een module is links een gespreks-scherm en rechts nog steeds de avatar, met vragen betreffende de huidige module, zie figuur 5.

Welkom bij de module: Stoppen met roken. Eerst wil ik graag een paar vragen stellen om inzicht te krijgen in uw situatie. Heeft u al eens een keer geprobeerd te stoppen met roken?

Goed dat u het nog een keer probeert, stoppen met roken lukt vaak nooit in één keer, gemiddeld stopt men voor goed na 3 pogingen.

Klik hieronder om terug te gaan naar het thuis scherm.

Stop met praten Verander Achtergrond Verander Persoon

Hoe lang duurt stoppen met roken gemiddeld?

Waarom zou ik stoppen met roken?

Figuur 5: Inhoud van een module

Doordat links is gekozen voor een chat-weergave wordt meer de suggestie gewekt dat de gebruiker daadwerkelijk omgaat met een persoonlijke coach. Verder zijn de knoppen gecustomiseerd en kleiner gemaakt, om minder chaos te veroorzaken. Zodra aan de linkerkant het gesprek te lang wordt, ontstaat er automatisch een scrollbar om het gesprek goed in beeld te houden. Hetzelfde geldt als er zeer veel modules worden toegevoegd in het startscherm.

4.4 De Editor

4.4.1 Initieel ontwerp

Zoals al eerder genoemd, liep het ontwerp van de editor parallel aan de implementatie. Over de implementatie kunt u meer te weten komen in hoofdstuk 4. Tijdens de implementatie van de editor werd een achterliggende SCRUM mentaliteit gebruikt en dit reflecteert zich in de totstandkoming van het ontwerp. Zo waren er in de eerste werkende versie van de editor nog niet veel functies en was het ontwerp daarom nog vrij overzichtelijk. Elke week daarna kwamen er echter meer functies bij waardoor het ontwerp moest veranderen om overzichtelijk te blijven.

Een belangrijk punt van het ontwerp van de editor was om alles overzichtelijk te houden. Er is daarom voor gekozen om alles op één scherm te houden. Zo kan in één blik worden gezien waar precies mee gewerkt wordt en kan er ook gemakkelijk van de ene functie naar de andere functie gesprongen worden. Dit bracht wel met zich mee dat alle knoppen, lijsten en teksten allemaal op één scherm moesten passen. Daarnaast moesten alle elementen op het scherm ook duidelijk weergeven bij welke functie deze hoorden. Dit was initieel een lastige taak, maar met behulp van feedback is de editor uiteindelijk een stuk overzichtelijker gemaakt.

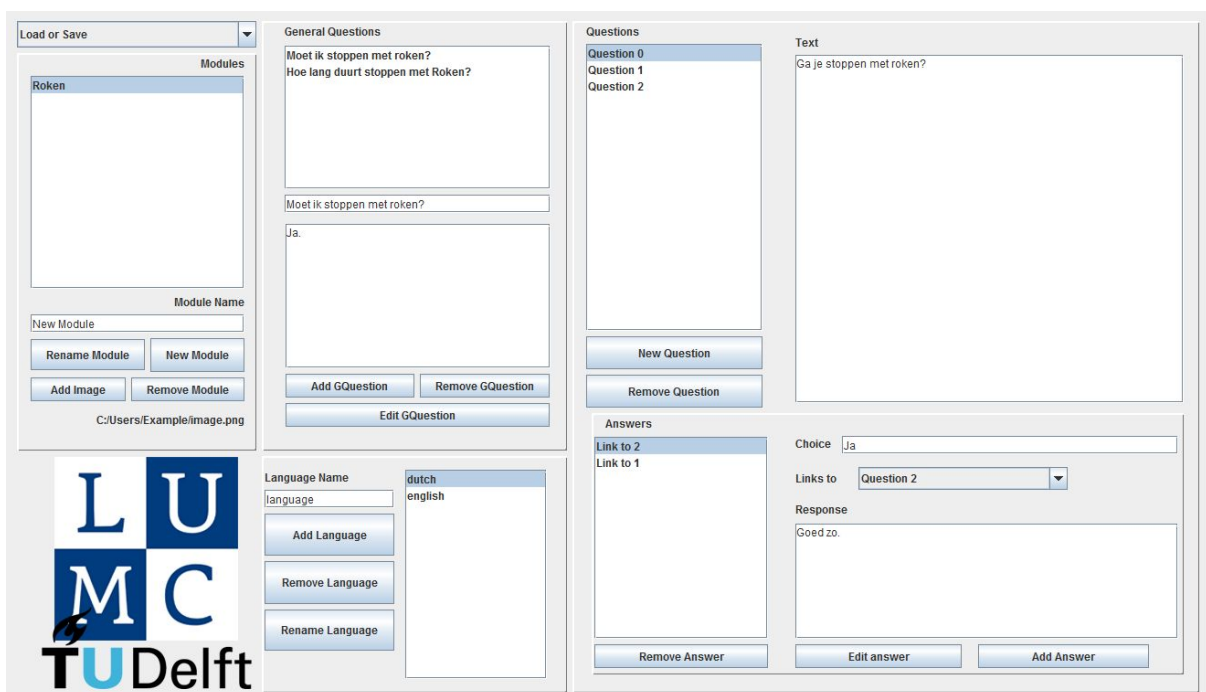
De feedback die werd ontvangen kwam ondermeer van de projectcoach, de opdrachtgevers van het LUMC, maar ook hebben mensen buiten het project hebben de editor getest en feedback geleverd. Er is hiermee veel feedback verkregen. Deze feedback kan worden samengevat in de onderstaande punten:

- Het is onduidelijk waar precies begonnen moet worden en waar verder moet worden gegaan.
- Van sommige knoppen wist men niet wat de functie was, sommige namen waren onduidelijk.
- Het selecteren van een element uit een lijst zou makkelijker zijn als je er gewoon op kon klikken.
- Wanneer iets in een tekstveld wordt getyped zou het een stuk handiger zijn als dit automatisch wordt opgeslagen.
- Het programma moet vragen of ik op wil slaan als ik het sluit met onopgeslagen veranderingen.
- De laad en opslag knoppen werden linksboven in een dropdown-menu verwacht zoals bij veel standaard programma's zoals Microsoft Word.

Deze punten zijn allemaal verwerkt en zo goed mogelijk opgelost in het uiteindelijke ontwerp.

4.4.2 Uiteindelijk ontwerp

Nadat alle functies waren geïmplementeerd en alle feedback was verwerkt is het uiteindelijke ontwerp van de editor ontstaan. In figuur 6 is dit ontwerp te zien.



figuur 6: Uiteindelijk ontwerp van de editor

Wat opvalt is dat het ontwerp is opgedeeld in een aantal panelen. Deze panelen representeren de vijf objecten die zijn gebruikt tijdens de implementatie: Taal, Module, Algemene Vraag, Vraag en Antwoord. Elke knop en veld op zo'n paneel heeft dan te maken met het object van het paneel. Zo kan makkelijker gezien worden waar elke knop bij hoort en wat zijn functie is.

Ook zijn initiëel nog wat knoppen gedeactiveerd. Deze knoppen activeren pas wanneer ze gebruikt kunnen worden. Zo kan gemakkelijk worden gezien waar gestart moet worden en hoe kan worden verder gegaan. Zo begint een programma maar met slechts één geactiveerde knop: 'add language'. Wanneer een taal wordt toegevoegd en geselecteerd zullen er knoppen bij het module paneel worden geactiveerd. Zo weet men dat een module maken de volgende stap is.

In eerdere versies van de editor had elke lijst een knop waarmee ze een element uit de lijst konden selecteren. Uit feedback bleek dat die onhandig was en zijn deze knoppen daarom verwijderd. Om nu een element uit een lijst te selecteren hoeft alleen maar op element geklikt te worden. Dit werd als een stuk handiger en intuïtiever beschouwd.

Voor het tekstveld van een vraag was ook een knop om deze tekst op te slaan. Naast dat deze knop verwarring bracht omdat de functie onduidelijk was, werd hij ook als overbodig gezien, omdat het automatisch opslaan van de tekst een stuk makkelijker was. Deze knop is daarom dus ook verwijderd en de tekst slaat nu automatisch op.

De laad- en opslag knoppen waren eerst grote knoppen die rechts onderin stonden. Deze namen veel ruimte in. Daarnaast was hun plaats, vooral van de laad knop, erg onintuïtief. Omdat het programma van links naar rechts werkt en iets inladen meestal aan het begin gebeurt, was rechtsonderin een vreemde plek voor deze knop. Deze twee knoppen zitten daarom nu in een dropdown menu links bovenin. Ook vraagt het programma of je op wil slaan wanneer je deze probeert te sluiten.

5

Implementatie

5.1 Overzicht

In dit hoofdstuk wordt implementatie de code uitgelegd. Eerst komt in 5.2 de HTML en JavaScript van de coach aan bod. Daarna wordt in 5.3 het bewerkingsprogramma voor de inhoud, de Editor, besproken. Tot slot wordt in 5.4 uitgelegd hoe de connectie tussen de twee tot stand komt.

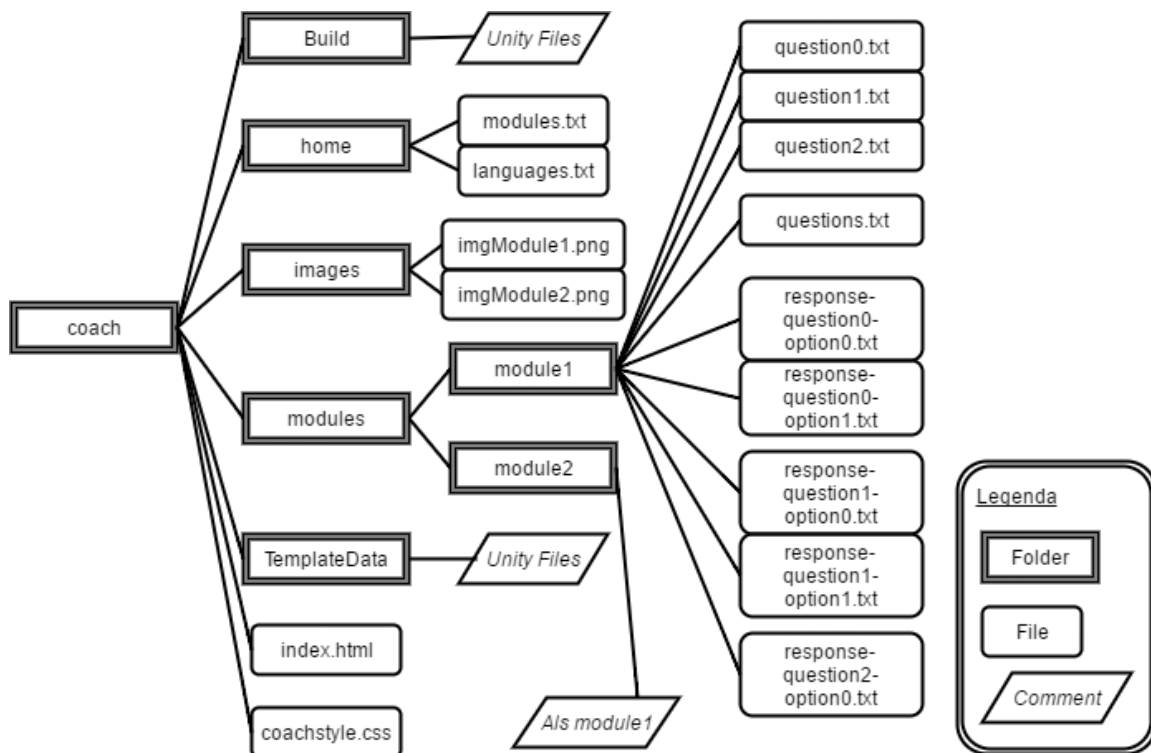
5.2 De Coach

De Coach is geïmplementeerd met Unity, HTML en JavaScript. Verder is CSS gebruikt voor de opmaak. De 3D avatar met spraakfunctie is ter beschikking gesteld door de Interactive Intelligence Group. Het is een framework dat in een kader van je browser, in dit geval een afgebakend stuk van 500x500 pixels, een sprekende avatar weergeeft. Met JavaScript kunnen functies van de avatar worden aangeroepen. De belangrijkste hiervan is

```
Coach.SendMessage('TalkingCoach', 'convertToSpeech', 'Hello');
```

Deze functie neemt als derde argument een String die moet worden uitgesproken. Dit wordt gedaan door zowel de mond van de avatar te laten bewegen als door de ingebouwde text-to-speech functie van de browser te gebruiken.

Het bestandssysteem is weergegeven in figuur 7.



Figuur 7: Bestandssysteem van de Virtuele Coach

Door het gebruik van meerdere mappen ontstaat er onderscheid tussen de verschillende delen van de coach. De belangrijkste bestanden en mappen worden verder uitgelegd.

5.2.1 index.html

De index is het hoofdbestand van de coach. Hij bepaalt wat de gebruiker te zien krijgt op het scherm. Hij zorgt ervoor dat de avatar wordt weergegeven, en hij gebruikt JavaScript functies om de coach te laten praten. Taalknoppen zorgen ervoor dat de gebruiker dynamisch kan wisselen tussen meerdere talen.

De index gebruikt div onderdelen met onclick-functies om functies aan te roepen. Functies kunnen zijn het veranderen van de taal, het direct vragen om een antwoord van de coach of het opvragen van het volgende deel van de module. Omdat het Unity deel opnieuw moet worden geladen als een nieuwe pagina wordt geopend, was het niet mogelijk meerdere html bestanden te gebruiken per module. Dan zou de avatar namelijk telkens afsluiten en weer opstarten. Dit is opgelost door slechts een enkel html bestand te gebruiken, en de inhoud aan te passen met behulp van XMLHttpRequests. Dit biedt ons de mogelijkheid de pagina te updaten zonder de pagina te herladen. De mogelijke teksten die moeten worden weergegeven bevinden zich in .txt bestanden in de andere mappen in het bestandssysteem. De functies in de index bepalen wat er wordt weergegeven en wat er wordt uitgesproken.

5.2.2 Map modules en Map Images

De modules bevatten de inhoud van de coach. De coach ondersteunt een oneindig aantal modules en talen. Een module bevat uit een afbeelding die wordt weergegeven in het module selectiescherm, ook wel hoofdscherm of startscherm. Verder bestaat een module uit vragen, responses en algemene vragen. Elke van deze bevindt zich in zijn eigen txt bestand

in de bijbehorende module map. Elke vraag en elke optie is gelinkt aan een functie in de index.

Zodra vanuit het startscherm een module wordt geselecteerd, begint deze module met het weergeven en uitspreken van de eerste vraag, `question0`. De mogelijke antwoorden van de gebruiker worden weergegeven, en elk triggert een aparte functie. Een functie moet namelijk een respons op het antwoord weergeven en uitspreken, en na 1 seconde de vraag weergeven waar het antwoord naar verwijst. Er kunnen dus vragen worden overgeslagen, en de volgorde hoeft niet altijd oplopend te zijn. Tijdens het aanmaken van modules kan ervoor worden gekozen om door te verwijzen naar vraag nummer -1. In zo'n geval wordt de gebruiker weer teruggeleid naar het startscherm.

5.2.3 Map home

Na het beëindigen van een module of bij het starten van de coach wordt het startscherm weergegeven. De inhoud hiervan wordt bepaald door de inhoud van de home map. Rechts worden de selecteerbare talen weergegeven, en links het plaatje en de naam van elke module. Elke module is verbonden met een `onclick` event dat ervoor zorgt dat de inhoud van de betreffende module wordt weergegeven.

De talen zijn geïmplementeerd met behulp van de `display` functie van classes. Elk mogelijk stuk tekst bevat een `<span>` voor elke taal, met als inhoud de betreffende tekst in die taal. Met behulp van `display = "none";` en `display = "inline-block";` wordt alleen de taal weergegeven die is geselecteerd. Naast het weergeven van de juiste taal moet ook de juiste taal worden gebruikt door het spraakprogramma van de coach. Dit wordt gedaan door een array van String mee te geven aan de `speak(q)` functie. Deze functie kijkt wat de huidige taal is en stuurt aan de hand daarvan een enkele string uit rij `q` door naar de coach.

5.2.4 Map Build en Map TemplateData

De mappen Build en TemplateData zijn niet aangepast. Deze worden automatisch gecreëerd door Unity met behulp van WebGL. In TemplateData zit een `textToSpeech.js` bestand dat het praten van de coach mogelijk maakt.

5.3 De Editor

De editor is geïmplementeerd via Java. Voor de grafische interface is Swing gebruikt. In figuur 8 is een UML te zien van de implementatie.

GQuestion is een algemene vraag. Dit zijn de vragen die bij een module rechts onderin komen te staan. Een GQuestion bestaat uit twee lijsten met Strings, één lijst bevat de vraag en de ander bevat het antwoord wat de coach zal geven.

5.3.1 De Interface

Naast deze 5 klassen heeft de editor ook nog een aantal klassen die de interface en zijn functionaliteiten vormen.

Window

De klasse Window vormt de grafische interface. In deze klasse komen alle panelen, knoppen, text velden, etc tot stand en worden op de juiste plaats gezet. Er wordt hier echter geen functionaliteit toegevoegd.

WindowController

De klasse WindowController geeft de laad en opslag functionaliteit aan de interface. Daarnaast bewaard deze klasse ook de huidige structuur van de objecten wordt bewaard. Zo bevat zijn attribuut 'modules' een lijst met alle modules van de huidige sessie en een ander attribuut 'languages' bevat een lijst met alle talen. Ook bevat deze klasse de attributen 'curmod', 'curquest', 'curopt', 'curgquest' en 'curlang'. Deze attributen zijn respectievelijk de huidige geselecteerde module, vraag, optie, algemene vraag en taal. In het laatste geval is curlang alleen een Integer dat de index van de taal in de lijst weergeeft. Omdat WindowController al deze data bijhoudt, komt de klasse veel in verband met andere klassen, dit is te zien in het UML, waar WindowController in het midden staat en met elke klasse wel op een manier gelinkt is. Om te zorgen dat WindowController hierdoor geen 'GodClass' werd, zijn zoveel mogelijk verantwoordelijkheden uit deze klasse verwijderd.

LanguageController

De LanguageController klasse voegt functionaliteit toe aan alle elementen die zich bevinden op het talen paneel. Meer informatie over het talen paneel is te vinden in Appendix A.

ModuleController

De ModuleController klasse voegt functionaliteit toe aan alle elementen die zich bevinden op het module paneel. Voor meer informatie over de functionaliteiten van het module paneel, zie Appendix A.

GQuestionController

De GQuestionController klasse voegt functionaliteit toe aan alle elementen die zich bevinden op het algemene vragen paneel. Meer informatie over dit paneel is te vinden in Appendix A.

QuestionController

De QuestionController klasse voegt functionaliteit toe aan de elementen die zich bevinden op het algemene vragen paneel, maar die niet op het Optie paneel zitten. Meer informatie over de functionaliteiten van het vragen paneel is te vinden in Appendix A.

OptionController

De OptionController klasse voegt functionaliteit toe aan alle elementen die zich op het Optie Paneel bevinden. Het optie paneel maakt deel uit van het vragen paneel, voor meer informatie over de functionaliteiten van het optie paneel, zie Appendix A.

5.4 Koppeling editor naar Coach

De editor is ontwikkeld voor mensen met lichte technische kennis, en de coach is ontworpen voor mensen met geen tot weinig technische kennis. De koppeling tussen de twee moet ervoor zorgen dat wat wordt aangemaakt in de editor daadwerkelijk kan worden weergegeven in HTML voor de gebruiker. Dit wordt in feite op twee manieren gedaan.

Ten eerste maakt de editor gebruik van een XML bestand dat kan worden uitgelezen, aangepast en weer opgeslagen. Alle benodigde informatie bevindt zich in dit bestand. Dit is handig voor overdracht en het bewerken van de inhoud. Ten tweede bevat de editor een klasse DirectoryCreator, die de huidige modules om kan schrijven naar een bestandssysteem zoals in figuur 7. Dan kan het mapje dat door deze klasse wordt aangemaakt op de server geplaatst worden, zodat de online coach direct wordt geüpdatet. Zo kan de inhoud van de coach dus worden uitgelezen uit XML, bewerkt worden in Java, en dan weer opgeslagen in zowel een XML bestand als een direct bruikbaar mapje voor de server.

6

Testen en code kwaliteit

6.1 Overzicht

In dit hoofdstuk zullen de testen en de kwaliteit van de code naar boven komen. In 6.2 zullen de verschillende manieren van het testen van de coach en de editor worden besproken. 6.3 zal gaan over de kwaliteit en structuur van de code, hierbij zal ook de feedback van de eerste SIG upload ter sprake komen.

6.2 Testen

Om te garanderen dat onze applicaties goed werken, moeten deze goed getest zijn.

Het testen van de coach is echter vrij lastig, de structuur van de javascript staat namelijk niet vast, omdat deze door de editor moet worden gemaakt. Aan de andere kant is de coach gemakkelijk handmatig te testen door zelf door alle modules heen te gaan en op alle knoppen te drukken. Elke optie van coach laat namelijk de coach spreken of laat iets op het scherm verschijnen, wat dus meteen te controleren is. De coach is dus uiteindelijk handmatig getest door door de inhoud van een paar kleine en grote modules te gaan. Vrijwel elke verschillende structuur mogelijkheid is hier aan bod gekomen.

De editor is een stuk makkelijker te testen. De editor heeft een vaste back-end code die niet verandert. Deze kon dus door junit tests getest worden. De gui is grondig getest met behulp van de java-klasse 'robot', hiermee wordt al het ware een gebruiker gesimuleerd die op verschillende knoppen in de gui klikt, daarna wordt getest of dit de gewenste effecten heeft. Het enige nadeel aan deze manier van testen is dat wanneer de knoppen van plaats veranderen, de tests niet meer zullen slagen. We kunnen er echter vanuit gaan dat de opmaak van de editor niet zal veranderen, waardoor dit voor dit project geen probleem vormt. Uiteindelijk heeft de editor een line-coverage boven de 90% behaald, zoals te zien in figuur 9.

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
▲ BEP	 94.9 %	10,382	563	10,945
▲ src	 94.9 %	10,382	563	10,945
▸ main.java	 92.7 %	7,092	558	7,650
▸ test.java	 99.8 %	3,290	5	3,295

Figuur 9: Coverage van de editor

6.3 Code kwaliteit

Een ander belangrijk punt van de applicaties is dat de code goed gestructureerd is, zodat de code goed te begrijpen en bewerken is door andere informatici. De structuur van de coach zal op eerste indruk lichtelijk gecompliceerd zijn, deze wordt namelijk opgesplitst in vele bestanden, gebaseerd op de inhoud gemaakt in de editor. Echter hebben al deze bestanden een duidelijke hiërarchie en functie die te zien is in hun naam. Verder is de inhoud van deze bestanden niet lastig te begrijpen, deze inhoud is vaak kort en bevat slechts een paar regels code. De structuur van de coach is vooral zo gemaakt zodat het makkelijk is de inhoud aan te passen. In de eerste SIG review werden geen op- of aanmerkingen gemaakt over de code van deze coach.

Het implementeren van de editor kwam pas vrij laat in het project naar voren. Dit had als resultaat dat bij de eerste SIG review, de editor pas in zijn allereerste versie zat. Hierdoor was er nog niet veel code om te reviewen, maar over deze code was alsnog goede feedback geleverd. Er waren twee grote problemen met de code. Het eerste probleem was dat de initiatie van de gui en de werking van de knoppen stonden in dezelfde klasse, hierdoor werd dit een heel grote en onduidelijke klasse. Deze zijn uiteindelijk gesplitst in 7 verschillende klassen, namelijk 'window' voor de instantie van de interface en de 6 '-controller' klassen die functies geven aan de elementen van de interface.

Het tweede probleem was dat de code nog niet getest werd door junit tests. In 6.2 werd duidelijk dat de editor nu wel goed getest wordt. Aan deze feedback is dus ook gewerkt. Verder is ook nog javadoc toegevoegd aan elke klasse en methode behalve de setters en getters, dit zal men helpen om snel te zien wat een klasse en zijn methodes precies doet.

Aangezien de code van de editor nu een stuk groter is, zal de tweede review van SIG een stuk meer feedback kunnen leveren, deze feedback zal echter pas na de deadline van dit verslag komen, dus zal deze hier nog niet worden besproken.

Evaluatie Eindproduct

7.1 Overzicht

In dit hoofdstuk zal het eindproduct worden geëvalueerd, deze evaluatie zal naar verschillende criteria kijken. Zo zal er in 6.2 gekeken worden hoeveel van de benodigdheden die zijn vastgesteld in hoofdstuk 2.6. In 6.3 zal het eindproduct worden vergeleken met de ontwerpdoelen die zijn vastgesteld in hoofdstuk 2.5. Daarna zullen in 6.4 aanbevelingen worden gedaan voor mensen die een vergelijkbaar project starten. Tenslotte zal er in 6.5 een conclusie worden getrokken over het gehele project.

7.2 Evaluatie van de benodigdheden

In dit deel zal het eindproduct worden vergeleken met de benodigdheden die waren vastgesteld. In de tabel hieronder staat bij elke benodigdheid of deze wel of niet behaald is en een uitleg hiervan.

Benodigdheid	behaald	Uitleg
Web interface met een geanimeerde coach.	ja	Ons eindproduct bevat een webapplicatie met een geanimeerde pratende coach.
Coach communiceert op een menselijke manier.	ja	De coach spreekt met de patiënt en stelt en beantwoordt vragen.
Inhoud van de Coach moet gemaakt kunnen worden in de editor.	ja	Met de editor kan alle tekst van de coach worden gemaakt.
De editor maakt het creëren van modules voor de coach mogelijk.	ja	Zie hierboven.
De coach spreekt meerdere talen.	ja	De coach werkt in zoveel talen als je invult.
De coach is gepersonaliseerd.	nee	Het personaliseren van de coach was helaas niet mogelijk, dit komt omdat de data die daarvoor opgeslagen moest worden absoluut privé moest blijven. Het zou een te groot risico vormen om deze data buiten de LUMC servers te

		bewaren. Het gebruiken van LUMC servers bleek ook geen mogelijkheid.
De editor kan bestaande modules in laden en deze bewerken.	ja	De editor kan deze bestanden opslaan en inladen als XML bestand
De coach legt uit over de in te nemen medicijnen.	ja	Met de editor kunnen specifieke vragen naar uitleg worden gemaakt (zoals de algemene vragen).
Smartphone cliënt voor de web-applicatie.	nee	De cliënt gaf al snel aan dat dit niet nodig was en het idee is daarna snel vervallen.
De Coach maakt gebruik van handgebaren en lichaamstaal.	nee	Door de limitatie van het unity framework was dit niet realistisch.
Verbetering van de stem van de coach.	nee	Omdat de coach een oneindig groot en veranderend dialoog kan hebben, is het inspreken van dialoog geen valide optie. De coach gebruikt de text-to-speech van de browser van de gebruiken, deze kan dus alleen in de browser zelf worden aangepast en niet via de applicatie.
De Coach kan naast tekst ook plaatjes naar de persoon sturen.	nee	Dit zou uiteindelijk te lastig worden om te gebruiken met een editor. De vrijheid die de editor geeft voor de plaatjes maakt het te lastig om het in het ontwerp van de coach te passen.
De editor geeft een visuele boom weer die de link tussen alle vragen van een module laat zien	nee	De vragen in de editor kunnen vrij naar elkaar linken waardoor er in plaats van een boom een chaotische graaf zou ontstaan, het zou er niet duidelijker van worden.
Uitbreiding van coach voor andere ziektes en medicijnen.	ja	Omdat alle dialoog en vragenstructuur aan te passen is, zou de coach in theorie voor veel meer medicijnen gebruikt kunnen worden.
Het uiterlijk van de coach verbeteren en menselijker maken.	nee	Door de limitatie van het unity framework was dit niet realistisch.

7.3 Evaluatie van de ontwerpdoelen

In dit deel zal naar de verschillende ontwerpdoelen worden gekeken en in hoeverre het eindproduct deze doelen behaalt.

Prestatie

De coach kan alle benodigde informatie overbrengen en zou zelfs door slim gebruik van vragen stellen erachter kunnen komen hoe vaak de patiënt al met de coach heeft gesproken en waar in het process de patiënt zit. Zo zou een coach kunnen vragen of de patiënt al vaker een module heeft bezocht en zo ja, welk onderwerp vorige keer was besproken.

De editor kan de inhoud van de coach creëren en aanpassen, ook kan hij deze met één druk op een knop omzetten naar javascript voor de web-applicatie.

De interface van de coach is vrij simpel en gebruiksvriendelijk. Er is niet veel onnodige informatie op het scherm die de patiënt zou kunnen afleiden. De patiënt zelf hoeft tijdens de module alleen maar vragen te beantwoorden en krijgt zo al de uitleg die zij nodig hebben. Ook is de interface goed uitbreidbaar. Met de huidige layout zouden oneindig veel modules gemaakt kunnen worden, ook al moet hiervoor dan wel gescrolld worden. Realistisch passen er genoeg modules op één scherm dat scrollen niet nodig is. De inhoud van de modules kunnen ook zo lang zijn als het maar kan.

De interface van de editor is wat gecompliceerder dan die van de coach. Dit is, zoals eerder vermeld, echter geen probleem. Wel is nog zo veel mogelijk gedaan om een duidelijke structuur in de editor te stoppen.

Qua prestatie heeft het eindproject zijn doelen dus behaald.

Beveiliging

De applicatie slaat absoluut geen informatie op van de patiënt, hierdoor is de applicatie erg secuur en kan er geen gevoelige informatie verkregen worden door hackers. Ook al lijdt de personalisatie van de applicatie hieronder, het veiligstellen van deze gevoelige persoonsgegevens is vele malen belangrijker.

De editor is volledig offline, dus zijn modules die hierop gemaakt worden en nog niet online gezet zijn dus net zo goed beveiligd als alle andere bestanden op de computer.

Het eindproduct heeft dus ook de beveiligings doelen behaald.

Onderhoud

De applicatie is erg makkelijk te onderhouden. Via de editor is het heel simpel om kleine of grotere veranderingen aan modules te maken. Ook het toevoegen en verwijderen van modules is goed mogelijk. Wel wordt geadviseerd om een back-up te maken van de huidige modules in het geval dat er tijdens het editen iets verkeerd gaat (zoals het per ongeluk verwijderen van een module). Het back-up bestand kan dan worden ingeladen om dit probleem op te lossen. Het veranderen van de editor zelf is wel programmeerkennis voor nodig, maar iedereen die dit heeft zou niet veel moeite hebben met het aanpassen van de editor. De code wordt namelijk goed getest en elke methode heeft documentatie.

Dit doel is dus ook bereikt door het eindproduct.

7.4 Aanbevelingen

In dit deel zullen er aanbevelingen worden gedaan voor mensen die met een vergelijkbaar project beginnen. Dit zullen dingen zijn die wij graag hadden willen zien in ons eindproduct, maar er uiteindelijk niet in zijn gekomen of dingen die wij anders zouden hebben gedaan als we het project opnieuw moesten doen.

Ten eerste zal het personaliseren van de coach heel veel mogelijkheden met zich mee brengen. Dit zou een grote stap naar voren zijn voor het gebruik van virtuele coaches. Wanneer de veiligheid van informatie geen probleem is (door bijvoorbeeld het hebben van secure servers, of het handelen van niet gevoelige informatie), is het zeker de moeite waard om personalisatie aan de coach toe te voegen.

Het gebruik van een ander framework voor de coach kan ook veel voordelen hebben. Het framework wat voor dit project is gebruikt maakte veel dingen een stuk makkelijker, zo stond de pratende avatar al bijna meteen klaar. Aan de andere kant bracht het ook veel limitaties met zich mee. Het laden van de coach kan lang duren op minder snelle computers. Ook was het menselijker maken van de coach erg gelimiteerd. De coach gebruikte de text-to-speech van de browser, waardoor de stem niet aangepast kon worden en ook het maken van handgebaren is nog niet mogelijk in dit framework.

Voor dit project was dit framework nog erg handig om te gebruiken, maar in de toekomst zal het zeker waard zijn om ook naar alternatieven van dit framework te kijken.

Verder zijn er nog veel benodigdheden in 'could have' en 'won't have', waar niet aan toegekomen was. Naar deze benodigdheden zouden ook gekeken kunnen worden als inspiratie latere projecten. Een smartphone-cliënt zou bijvoorbeeld heel handig kunnen zijn in de toekomst.

7.5 Conclusie

In dit deel zal een conclusie worden getrokken over het eindproduct, deze conclusie wordt getrokken door te kijken naar de bovenstaande evaluaties en deze te vergelijken met de succescriteria die zijn opgesteld in 3.4. Verder zal ook nog gekeken worden of het eindproduct voldoet aan de originele opdracht van het LUMC.

Volgens de succescriteria moest het eindproduct alle ontwerpdoelen volgen, in 7.3 wordt hiernaar gekeken en blijkt dat het eindproduct dele doelen ook behaalt. Verder moeten alle 'must have' benodigdheden en een merendeel van de 'should have' benodigdheden behaald worden. In de tabel in 7.2 kan worden opgemaakt dat de 'must have' benodigdheden allemaal zijn geïmplementeerd en de enige 'should have' benodigheid die niet is geïmplementeerd is de personalisatie van de coach, het merendeel zit er dus wel in. De succescriteria die zijn opgezet blijken dus behaald te zijn.

De originele opdracht was om een virtuele coach te maken die patiënten zou helpen bij het aanhouden van hun medicatie. Niet alleen is zo'n coach gemaakt, maar ook is een editor

gemaakt waardoor deze inhoud aangepast kan worden. Het eindproduct voldoet dus ook nog aan de opdracht van het LUMC en kan dus als geslaagd worden beschouwd.

Appendix A

Handleiding Editor

De editor is op te delen in 5 verschillende delen: Language, Module, GQuestion, Question en Option. Deze 5 delen hebben ieder een eigen paneel op de editor waarop al hun functies staan. Hieronder worden alle functies uitgelegd per paneel. Tenslotte wordt het laden en oplaan van bestanden uitgelegd.

Language

De volgende functionaliteiten bevinden zich op het talen paneel:

De knop 'Add Language' voegt een taal toe aan de lijst met talen. De naam van deze taal is de inhoud van het tekstveld onder 'Language Name'.

De knop 'Remove Language' verwijdert geselecteerde taal uit de lijst met talen.

De knop 'Rename Language' verandert de naam van de geselecteerde taal met de inhoud van het tekstveld onder 'Language Name'.



Talen paneel

De lijst rechts van deze knoppen laat alle bestaande talen zien. Wanneer op één van deze talen wordt geklikt zal deze taal geselecteerd worden. Alle tekst op de interface verandert dan ook naar de tekst van die taal.

Verder zorgen de knoppen 'Add language' en 'Remove Language' er ook voor dat de lijsten met Strings voor tekst in verschillende talen (bijvoorbeeld de titels van de modules, of de text van een vraag) altijd even lang zijn als de lijst met talen. 'Add Language' voegt dus bij deze lijsten aan het einde een nieuwe String toe. Remove Language haalt bij deze lijsten de String weg met dezelfde index als de te verwijderen taal.

Module

De volgende functionaliteiten bevinden zich op het module paneel:

Module paneel

De knop 'New Module' voegt een nieuwe module toe aan de lijst met modules, de titel van deze module (in elke taal) is gelijk aan de inhoud van het tekstveld onder 'Module Name'.

De knop 'Rename Module' verandert titel van de geselecteerde module met de inhoud van dit zelfde tekstveld. Dit kan gebruikt worden om per taal een andere titel te geven.

De knop 'Remove Module' verwijdert de geselecteerde module uit de lijst met modules.

De knop 'Add Image' opent een 'file explorer' waarmee een plaatje kan worden geselecteerd. De url van dit plaatje wordt dan toegevoegd aan de geselecteerde module.

Het selecteren van een module gaat net zoals het selecteren van een vraag, namelijk door te klikken op de gewenste module uit de lijst met modules.

GQuestion

De volgende functionaliteiten bevinden zich op het algemene vragen paneel:

Het bovenste witte vlak laat alle algemene vragen zien van de geselecteerde module, de inhoud van deze lijst zal dus veranderen wanneer van module wordt veranderd. Het selecteren uit deze lijst werkt hetzelfde als de andere lijsten.

'Add GQuestion' voegt een nieuwe algemene vraag toe aan de lijst met als vraag en antwoord de inhoud van het 'Question' tekstveld en 'Answer' tekstveld respectievelijk.

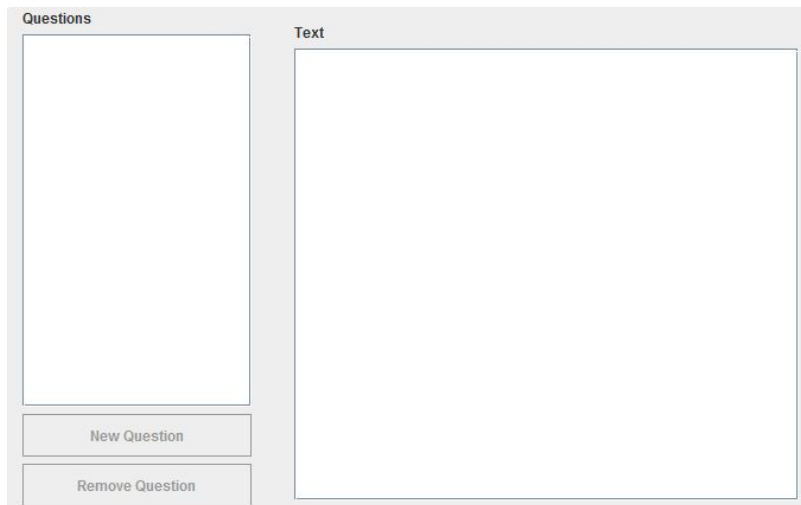
'Edit GQuestion' verandert de Vraag en Antwoord van de geselecteerde vraag met de inhoud van dezelfde twee tekstvelden.

'Remove GQuestion' verwijdert de geselecteerde vraag uit de lijst.

Algemene vraag paneel

Question

De volgende functionaliteiten bevinden zich op het vragen paneel:



Linksboven staat een lijst met de vragen van de huidige module. Deze lijst verandert dus ook als van module wordt veranderd. Ook het selecteren uit deze lijst werkt net zoals de andere lijsten.

'New Question' voegt een nieuwe vraag toe. Deze vraag heeft nog geen tekst of antwoorden.

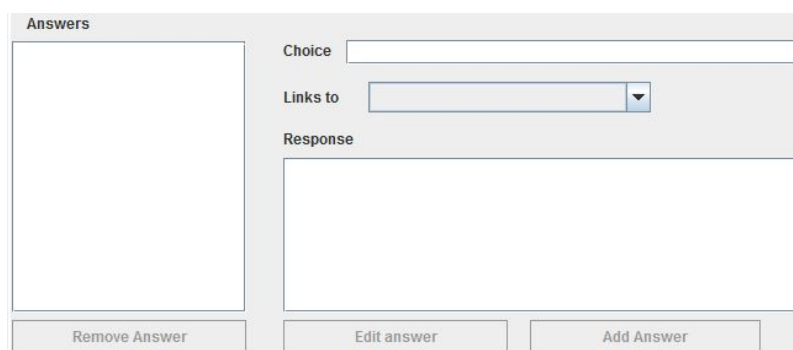
Vragen Paneel, zonder Optie Paneel

Het grote vlak rechts laat de text zien van de geselecteerde vraag. Deze tekst kan hier bewerkt worden en wordt dan meteen opgeslagen in de vraag zelf.

'Remove Question' verwijdert de geselecteerde vraag uit de lijst met vragen.

Option

De volgende functionaliteiten bevinden zich op het optie-paneel:



Aan de linkerkant staat een lijst met 'Options' (ook wel antwoorden) van de huidige vraag. Deze lijst verandert wanneer van vraag wordt veranderd en het selecteren werkt als elke andere lijst.

Optie Paneel

'Add Answer' voegt een optie toe aan de huidige vraag, de data van de optie wordt als volgt bepaald:

- de keuze is de inhoud van het tekstveld achter 'Choice'
- het antwoord van de coach is de inhoud van het tekstveld onder 'Response'

- De vraag waar het antwoord naar linkt kan worden gekozen uit het drop-down menu achter 'Links to'. Dit menu bevat een lijst met alle vragen van de module.

'Edit Answer' past de geselecteerde optie aan door deze de data te geven die op dezelfde manier als 'Add Answer' wordt verkregen.

'Remove Answer' verwijdert de geselecteerde optie uit de lijst met opties.

Laden en Opslaan

Verder is het ook nog mogelijk om de modules op te slaan als xml bestand. Deze modules kunnen daarna ook weer worden ingeladen. Om dit te doen moet linksboven op de dropdown-list 'load and save' worden geklikt. Nu verschijnen de opties om een bestand te laden (load) of een bestand op te slaan (save). Beide zullen een file explorer openen waar je een bestand kunt kiezen om in te laden, of een locatie kunt kiezen om op te slaan.

B

Info Sheet

Algemene informatie

Titel van het project: Virtuele coach voor patiënten met hart- en vaatziekten

Naam van de cliënt organisatie: LUMC

Datum van de eindpresentatie: 5 juli 2017

Beschrijving

De hoofd uitdaging van het project was het creëren van een virtuele coach die patiënten met hart- en vaatziekten helpt met hun medicatie. Hiervoor is veel onderzoek gedaan naar het gebruik van virtuele avatars in de zorg en op andere gerelateerde gebieden. Ook is onderzoek gedaan naar gesprekstechnieken die de coach zou kunnen gebruiken.

Halverwege het project bleek het belangrijk te zijn dat de inhoud van de coach aangepast kon worden. Dit veranderde de probleemstelling naar het maken van een algemene virtuele coach met een editor. Voor dit probleem zijn de volgende twee producten als oplossing bedacht. Ten eerste is er een webapplicatie gemaakt waarbij een virtuele 3d coach vragen stelt en beantwoordt over verschillende onderwerpen. Ten tweede is er een editor in Java gemaakt. Met deze editor is het mogelijk om de verschillende onderwerpen en vragen van de coach te creëren en aan te passen. Deze methode staat toe dat experts op het gebied van hart- en vaatziekten de inhoud van de coach kunnen schrijven. Ook zullen hierdoor veranderingen in medicatie gemakkelijk doorgegeven kunnen worden aan de coach. De coach zal niet direct gebruikt worden door het LUMC, aangezien ze de inhoud eerst zelf moeten invullen met behulp van de bijgeleverde editor, daarna kan het worden gebruikt. Daarnaast zal dit product een goede stap zijn naar het gebruik van virtuele coaches in de zorg.

Leden van het project team

Naam: Wytze Elhorst

Interesses: Java Programming en Design

Contributie en rol: Hoofdprogrammeur van de editor.

Naam: Alan van Rossum

Interesses: Web Development en Algoritmes

Contributie en rol: Hoofdprogrammeur van de webapplicatie



Cliënt

Naam: R. W. Treskes

Affiliatie: LUMC

Coach

Naam: W. P. Brinkman

Afdeling: Interactive Intelligence Group

Contact

Alan van Rossum, alanvanrossum@gmail.com

Wytze Elhorst, welhorst@hotmail.com

Het eindverslag van dit project kan worden gevonden op <http://repository.tudelft.nl>

De code van dit project kan worden gevonden op

<https://github.com/wytzeelhorst/coach-editor>

Project Omschrijving

Roken, weinig bewegen, veel en ongezond eten en teveel alcohol zijn typische voorbeelden van een ongezonde leefstijl. Inmiddels is uitvoerig bewezen dat een ongezonde leefstijl de kans op het krijgen van hart & vaatziekten enorm kan verhogen. Het tegenovergestelde is ook waar: als patiënten na een hartinfarct gezond leven, kunnen ze de kans op een nieuw infarct verkleinen. Het aanpassen van leefstijl is echter heel moeilijk en veel patiënten vallen terug in oude (ongezonde) patronen. Op dit moment worden patiënten vaak alleen geïnformeerd over gedragsverandering en de mogelijkheden tot het volgen van cursussen. Dit blijkt in de praktijk helaas onvoldoende. Het coachen van patiënten blijkt effectief, maar is tijdrovend en erg arbeidsintensief.

Om dit probleem op te lossen wordt er gedacht over het gebruik maken van een web-based virtuele coach. Dit zijn interactieve personages die vaak antropomorfische (menselijke) elementen hebben om (sociale) reacties/gedrag op te roepen die mensen normaal vertonen wanneer ze communiceren met een ander persoon. Vanwege zijn natuurlijke karakter is de communicatie met een virtueel karakter laagdrempelig en wordt er minder ICT kennis vereist van de gebruiker. Dit laatst is van belang vanwege de diversiteit van patiënten waarmee het LUMC te maken heeft. Het doel is daarom een virtuele coach te bouwen die patiënten gaat helpen bij het gewend raken van hun nieuwe levensstijl. Een belangrijk element daarbij is het medicijn gebruik. Als een preventie maatregel, is het namelijk wenselijk dat deze chronische patiënten een aantal medicijn gaan gebruiken voor de rest van hun leven. Vaak bestaat er verwarring hierover bij de patiënten, bijvoorbeeld waar deze medicijnen voor dienen, en dat het niet de bedoeling is dat deze na verloop van tijd afgebouwd worden. Naast deze levensstijl verandering zijn er ook andere die bijvoorbeeld te maken hebben met lichaamsbeweging en voeding. Het succes van de virtuele coach bestaat er uit dat hij een bijdrage levert aan de uiteindelijke levensstijlverandering, bijvoorbeeld dat de patiënt een routine opbouwt voor dagelijks medicijn gebruik. Een belangrijke uitdaging bij het ontwikkelen van deze virtuele coach is het implementeren van "persuasive" strategies die de kans verhogen dat patiënten het gewenste gedrag gaan vertonen, door bijvoorbeeld: empathie uit te dragen, het makkelijk maakt voor patiënt zijn gedrag te veranderen, of door de patiënt te belonen voor gewenst gedrag.

De TU Delft Interactive Intelligence groep is bezig met het ontwikkelen van een framework voor het visualiseren van een virtuele coach, deze is geïmplementeerd in Unity. Het framework stelt programmeurs in staat personages te visualiseren en teksten door deze personages te laten uitspreken. Het is de bedoeling dat dit project gebruik maakt van dit framework.

De volgende taken kunnen onderscheiden worden in dit project

1. literatuur onderzoek naar bestaande virtuele coaches, mogelijk interactie stijlen, en persuasieve strategieën.
2. Ontwerp(en) maken van een virtuele coach en deze reviewen met experts
3. Implementeren van de virtuele coach en de onderhoudsomgeving
4. Invoeren van de levensstijl informatie in de virtuele coach
5. Beta test en gebruikerstest van de virtuele coach

Company description

Het LUMC is een tertiair ziekenhuis waar hoogcomplexere zorg wordt geleverd. De afdeling Cardiologie is één van de grootste afdelingen van het ziekenhuis. Jaarlijks worden er 25000 polibezoeken gedaan. Naast zorg voor hartpatiënten doet de afdeling onderzoek, zowel basaal onderzoek als klinisch onderzoek. Tevens is de afdeling betrokken bij onderwijs voor verschillende opleidingen (onder andere verpleegkunde, geneeskunde, hartfunctielaboranten, verpleegkundig specialisten en cardiologen). De afdeling is zeer vooruitstrevend op het gebied van digitale toepassingen in de zorg. Zo was het één van de eerste afdelingen in Nederland dat met een elektronisch patiëntendossier ging werken. Nu werkt het volop met e-Health toepassingen als telemonitoring van hartinfarct patiënten, patiënten met een implanteerbare defibrillator en patiënten met een cryptogeen herseninfarct.

Feedback SIG

Onderstaande tekst is de feedback die is ontvangen van de eerste SIG upload:

Het eerste wat opvalt is dat jullie nog erg weinig code hebben geschreven, zoals we al per mail hebben besproken. We geven daarom nog geen score op basis van ons onderhoudbaarheidsmodel.

Wat al wel opvalt is dat jullie vrij een user interface in Swing hebben geschreven. Het is daarbij goed om op te letten dat de definitie van de interface niet door elkaar gaat lopen met het gedrag dat aan de interface is verbonden. In andere woorden: zorg dat de code voor het aanmaken van knoppen en de code voor als je op die knop klikt niet door elkaar staan. Op dit moment is dat namelijk wel zo:

```
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (!(list_m.isSelectionEmpty())) {
            curmod = list_m.getSelectedValue();
            Question quest = new Question(curmod.questions.size(), "", new ArrayList<Option>());
            curmod.questions.add(quest);
            modelq.addElement(quest);
        }
    }
});

btnNewModule.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (txtModuleName.getText().equals("")) {
            Module mod = new Module("newModule", new ArrayList<Question>());
            modules.add(mod);
            modelm.addElement(mod);
        } else {
            Module mod = new Module(txtModuleName.getText(), new ArrayList<Question>());
            modules.add(mod);
            modelm.addElement(mod);
        }
    }
});
```

Dit is nu nog geen probleem, omdat de hoeveelcode code voor het afhandelen van een "action" vrij beperkt is. Over het algemeen is het echter zo dat je steeds meer knoppen en steeds meer acties krijgt, en dan werkt deze aanpak niet meer. Het is daarom beter om de code binnen de ActionListener naar een aparte methode te verplaatsen, op die manier hou je de verantwoordelijkheden beter gescheiden en de code beter leesbaar.

Als laatste nog de opmerking dat er geen (unit)test-code is gevonden in de code-upload. Het is sterk aan te raden om in ieder geval voor de belangrijkste delen van de functionaliteit automatische tests gedefinieerd te hebben om ervoor te zorgen dat eventuele aanpassingen niet voor ongewenst gedrag zorgen.



XML voorbeeld

```
<core>
<module>
<moduleimagesrc>
C:/Users/Example/image.png
</moduleimagesrc>
<moduletitle>
<dutch>
Roken
</dutch>
<english>
Smoking
</english>
</moduletitle>
<leftsidequestion>
<index>
0
</index>
<question>
<dutch>
Ga je stoppen met roken?
</dutch>
<english>
Will you quit smoking?
</english>
</question>
<answer>
<value>
<dutch>
Ja
</dutch>
<english>
Yes
</english>
</value>
<respons>
<dutch>
```

Goed zo.

</dutch>

<english>

Nice one.

</english>

</respons>

<nextquestionindex>

2

</nextquestionindex>

</answer>

<answer>

<value>

<dutch>

Nee

</dutch>

<english>

No

</english>

</value>

<respons>

<dutch>

Probeer toch maar.

</dutch>

<english>

You should try to.

</english>

</respons>

<nextquestionindex>

1

</nextquestionindex>

</answer>

</leftsidequestion>

<leftsidequestion>

<index>

1

</index>

<question>

<dutch>

Klik op de knop om terug te gaan naar het hoofdscherm.

</dutch>

<english>

Click on the button to return to the homescreen.

</english>

</question>

<answer>

<value>

```

<dutch>
Terug.
</dutch>
<english>
Back.
</english>
</value>
<respons>
<dutch>
Stilte?
</dutch>
<english>
Stilte? Dit kan evt ook leeg. Nextquestionindex van -1 betekent: terug naar home.
</english>
</respons>
<nextquestionindex>
-1
</nextquestionindex>
</answer>
</leftsidequestion>
<leftsidequestion>
<index>
2
</index>
<question>
<dutch>
Rookt u nu nog?
</dutch>
<english>

</english>
</question>
<answer>
<value>
<dutch>
Ja
</dutch>
<english>
Ja
</english>
</value>
<respons>
<dutch>
Probeer dit dan geleidelijk af te nemen.
</dutch>
<english>

```

Probeer dit dan geleidelijk af te nemen.

</english>

</respons>

<nextquestionindex>

1

</nextquestionindex>

</answer>

<answer>

<value>

<dutch>

Nee

</dutch>

<english>

Nee

</english>

</value>

<respons>

<dutch>

Lekker bezig! Ga zo door!

</dutch>

<english>

Lekker bezig! Ga zo door!

</english>

</respons>

<nextquestionindex>

1

</nextquestionindex>

</answer>

</leftsidequestion>

<rightsidequestion>

<question>

<dutch>

Moet ik stoppen met roken?

</dutch>

<english>

Should I quit smoking?

</english>

</question>

<respons>

<dutch>

Ja.

</dutch>

<english>

Yes.

</english>

</respons>

```
</rightsidequestion>
<rightsidequestion>
<question>
<dutch>
Hoe lang duurt stoppen met Roken?
</dutch>
<english>
How long does quitting take?
</english>
</question>
<respons>
<dutch>
Dat hangt heel erg af aan de persoon, maar gemiddeld
</dutch>
<english>
it depends
</english>
</respons>
</rightsidequestion>
</module>
</core>
```

Referenties

1. S. Kang et al (2017), *Social influence of humor in virtual human counselor's self-disclosure*, Computer Animation & Virtual Worlds, DOI: 10.1002/cav.1763 <http://onlinelibrary.wiley.com/tudelft.idm.oclc.org/doi/10.1002/cav.1763/full>
2. Razavi S.Z., Ali M.R., Smith T.H., Schubert L.K., Hoque M.. (2016) The LISSA Virtual Human and ASD Teens: An Overview of Initial Experiments. In: Traum D., Swartout W., Khooshabeh P., Kopp S., Scherer S., Leuski A. (eds) *Intelligent Virtual Agents. IVA 2016. Lecture Notes in Computer Science*, vol 10011. Springer, Cham https://link.springer.com/chapter/10.1007%2F978-3-319-47665-0_55
3. A. Shamekhi et al (2016), *A Virtual Self-care Coach for Individuals with Spinal Cord Injury*, ASSETS '16 Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility, doi: 10.1145/2982142.2982199 <http://dl.acm.org/citation.cfm?doid=2982142.2982199>
4. Liu HaoWen et al (2013), Design of online pseudo-code editor and code generator, Conference Anthology, IEEE, DOI: 10.1109/ANTHOLOGY.2013.6784941 <http://ieeexplore.ieee.org/document/6784941/>
5. Olivier A. Blanson Henkemans et al (2009). *An online lifestyle diary with a persuasive computer assistant providing feedback on self-management*, Technology and Health Care 17 (2009) 1–15 DOI 10.3233/THC-2009-0545
6. Blanson Henkemans OA, et al. *Using a robot to personalise health education for children with diabetes type 1: A pilot study*. Patient Educ Couns (2013), <http://dx.doi.org/10.1016/j.pec.2013.04.012>
7. S. Maes and P. Karoly, Self-regulation Assessment and Intervention in Physical Health and Illness: A Review, *Applied Psychology An International Review* 54(2) (2005), 267–299(33).
8. S. van Dulmen, E. Sluijs, L. van Dijk, D.T.D. de Ridder, E.R. Heerdink and J. Bensing, *Patient Adherence to Medical Treatment: A Review of Reviews*, Health Services Research 7 (2007), 55.
9. LUMC, Centrum van medische vernieuwing, <https://www.lumc.nl/over-het-lumc/organisatie-en-missie/>
10. Hartwijzer: Roken, <http://www.hartwijzer.nl/Roken.php>
11. Hartstichting, *Gezond Leven*, <https://www.hartstichting.nl/gezond-leven>
12. B.J. Fogg, *The Fogg Method*, <http://www.foggmethod.com/>
13. A. Karpel, *Roos van Leary*, <http://wij-leren.nl/roos-van-leary-gedrag.php>
14. Talking Coach Unity Framework, <https://github.com/ruuddejong/talkingcoach>
15. The XML FAQ, <http://xml.silmaril.ie/whyxml.html>
16. LUMC Patiëntenportaal <https://www.lumc.nl/patientenzorg/patienten/digid/>
17. Verandering van medicatie. Onderzoek onder patiënten en mantelzorgers, <http://www.vmdb.nl/wp-content/uploads/2012/02/Onderzoek-TNS-healthcare.pdf>
18. Baxter, S. Vogt, L.C. (2002), Content management system, Google Patents, <https://www.google.com/patents/US6356903>

19. Hans van Vliet, *Software Engineering: Principles and Practice*, third edition, Wiley, Chichester (UK), 2008, p. 63
20. Smith, A., & Anderson, J. (2014). AI, Robotics, and the Future of Jobs. *Pew Research Center*, 6.
<http://www.fusbp.com/wp-content/uploads/2010/07/AI-and-Robotics-Impact-on-Future-Pew-Survey.pdf>
21. Noury, Norbert, et al. "New trends in health smart homes." *enterprise networking and computing in healthcare industry, 2003. Healthcom 2003. Proceedings. 5th international workshop on*. IEEE, 2003.