Technisch ontwerp

|  |  |
| --- | --- |
| Projectnaam : | Goodlookz |
| Bedrijfsnaam : | 4People |
| Soort document : | Technisch ontwerp |
| Datum : | 3/4/18 |
| Naam student : | R. Broens |
| Naam assessor : | J. van Amsterdam |
| Naam bpv-begeleider : | J. van Amsterdam |
| Naam praktijkopleider : | J. Franssen |
| Versie document : | 1.01 |

Goodlookz

# Voorwoord

Dit document neemt onderdeel uit aan het eindassessment van de opleiding applicatieontwikkelaar in het derde leerjaar van de opleiding. Het doel van dit document is om verdere informatie te bieden over het technische gedeelte van het ontwerp. Verder is dit document geschreven door Rebecca Broens, die momenteel stageloopt bij het bedrijf 4People in Nuenen.

Goodlookz is een mobiele applicatie die meiden uitnodigt om hun kleding te delen met vrienden. Zij kunnen hun kleding kast online zetten en deze delen met hun vrienden. Deze applicatie is ontwikkeld door Furkan Demirci, een eerdere stagiaire van 4People. Momenteel is dit al mogelijk, maar er is behoefte aan een uitbreiding van het programma.

# Versiebeheer

Hieronder ziet u welke aanpassingen er plaatsgevonden hebben over diverse versies van het document:

|  |  |
| --- | --- |
| Versie | Aanpassingen |
| V0.01 | Opzet document, structuur document, hoofdstuk 1 afgerond, mappenstructuur web api |
| V0.02 | Datamodellen toegevoegd |
| V0.03 | Mappenstructuur applicatie, klassendiagram toegevoegd |
| V0.04 | Usecases, sequence diagrammen |
| V0.05 | Schetsen, navigatiestructuur en flowcharts toegevoegd |
| V1.00 | Spellingscontrole |
| V1.01 | Hardware informatie ontwikkel systeem toegevoegd |

Inhoud

[Voorwoord 1](#_Toc509216047)

[Versiebeheer 1](#_Toc509216048)

[Inleiding 3](#_Toc509216049)

[Hoofdstuk 1: Techniekkeuze 4](#_Toc509216050)

[Programmeertalen 4](#_Toc509216051)

[Ontwikkelmethodiek 4](#_Toc509216052)

[Gebruikte software/websites 4](#_Toc509216053)

[Deelproducten 4](#_Toc509216054)

[Hoofdstuk 2: Gegevensmodel 5](#_Toc509216055)

[Klassendiagram 5](#_Toc509216056)

[Datamodel huidige database 6](#_Toc509216057)

[Toekomstige aanpassingen database 8](#_Toc509216058)

[Mappenstructuur Applicatie 10](#_Toc509216059)

[Mappenstructuur Web API 10](#_Toc509216060)

[Hoofdstuk 3: Functionaliteiten 12](#_Toc509216061)

[Use-cases 12](#_Toc509216062)

[Sequence diagrammen 14](#_Toc509216063)

[Flowcharts 21](#_Toc509216064)

[Hoofdstuk 4: Grafische user interface 26](#_Toc509216065)

[Navigatiestructuur 26](#_Toc509216066)

[Schetsen 29](#_Toc509216067)

[Nawoord 34](#_Toc509216068)

[Bijlage(s) 35](#_Toc509216069)

[Bijlage 1: Datamodel ontworpen door F. Demirci 35](#_Toc509216070)

[Bijlage 2: Datamodel huidige database 36](#_Toc509216071)

[Bijlage 3: Datamodel Aanpassingen database 41](#_Toc509216072)

[Bijlage 4: Complete mappen structuur 45](#_Toc509216073)

# Inleiding

Dit document is voornamelijk opgesteld om helderheid te bieden over het technische ontwerp van de applicatie. Daarnaast is de inhoud gericht om lezers, met kennis van ICT, te informeren over het technische gedeelte van de toekomstige uitbreidingen en/of aanpassingen.

Hoofdstuk 1, techniekkeuze, geeft de algemene afspraken weer over hoe de ontwikkeling tot stand komt. Er wordt informatie vrijgegeven over: de ontwikkelmethodiek, welke programmeertalen binnen de applicatie gebruikt worden en welke software en/of websites vereist zijn bij de ontwikkeling van het eindproduct. Daarnaast worden de specificaties bekendgemaakt van het ontwikkel systeem en wordt het product opgesplitst in deelproducten binnen hoofdstuk 1.

Vervolgens staat diverse informatie over de gegevens in hoofdstuk 2. Denk hierbij aan: een klassendiagram, datamodellen van de database en hoe de mappenstructuur er momenteel uit ziet of zal gaan zien.

Daarnaast worden de functionaliteiten uitgebreid uitgewerkt in flowcharts, use-cases en sequence diagrammen in hoofdstuk 3. Hierbij wordt er dieper ingegaan op de functionaliteiten in tegenstelling tot de flowcharts binnen het functioneel ontwerp.

Verder is er een navigatiestructuur weergegeven over hoe de huidige navigatie van de applicatie in elkaar zit in hoofdstuk 4. Daarnaast zal er een nieuwe versie van de toekomstige navigatiestructuur inbegrepen zijn in dit hoofdstuk. Vervolgens komen er nog enkele schetsen aan bod.

Tot slot komt het nawoord met hierin de contactgegevens van de auteur. Daarnaast zijn er bijlages terug te vinden na het nawoord. In bijlage 1 is het datamodel te zien ontworpen door F. Demirci. Vervolgens zijn in bijlage 2 en 3 de datamodellen te zien die door R. Broens ontworpen zijn. Deze gaat over de huidige situatie van de database en de aanpassingen die gedaan zullen worden. De gehele navigatiestructuur van de aanpassingen is te zien in bijlage 4.

# Hoofdstuk 1: Techniekkeuze

Aangezien het een toevoeging wordt op een bestaande applicatie, zal er gewerkt worden met de vooraf gekozen voorkeuren. Dit zorgt ervoor dat de applicatie coherent blijft ondanks de nieuwe aanpassingen die ontwikkeld worden bij de uitbreiding.

## Programmeertalen

Gezien het gaat om een mobiele applicatie wordt er momenteel gebruik gemaakt van Xamarin. Xamarin maakt het mogelijk om met één programmeertaal nieuwe applicaties te ontwikkelen die op verschillende platformen kunnen draaien, zoals: iOS, Windows Phone, UWP en Android.

Dit betekend dat de backend van de applicatie geschreven wordt in C# en de front-end in XAML. Daarnaast wordt er gebruik gemaakt van een webserver om te kunnen communiceren met de database. Om gegevens op te halen, te wijzigen of op te slaan wordt er gebruik gemaakt van PHP en JSON.

## Ontwikkelmethodiek

Er zal merendeels gewerkt worden via de watervalmethodiek tijdens de ontwikkeling van de applicatie, maar aangezien de opdrachtgever niet bekend is met ontwikkelmethodieken zullen er enige uitzonderingen ontstaan. Enkele methodes die bij tot de scrummethodiek behoren zullen gebruikt worden bij het realiseren van de applicatie.

De voortgang van het project wordt opgeslagen op de website “[Trello](https://trello.com/)”, een website die als tool gebruikt wordt bij een doorlopend project, waarbij verschillende borden gebruikt worden met kaarten. Dit zorgt ervoor dat de voortgang van het project duidelijk weergegeven worden voor de opdrachtnemer en opdrachtgever. Het kan voorkomen dat er meerdere functionaliteiten naderhand toegevoegd moeten worden bij de applicatie, deze zullen terug te vinden op het bord.

## Gebruikte software/websites

Bij het ontwikkelen van de applicatie wordt het programma “Visual Studio 2015” gebruikt. Deze zal tevens ook gebruikt worden bij het testen van de functionaliteiten. Zoals eerder aangegeven, er wordt een applicatie ontwikkeld waarbij gebruik gemaakt wordt van een webserver. Om deze code te schrijven, wordt het programma “Visual Studio Code” gebruikt.

Om de webserver te bereiken en bestanden aan te passen wordt [FileZilla](https://filezilla-project.org/index.php), een gratis ftp-software tool, gebruikt. Daarnaast zijn er enkele websites die gebruikt worden bij de ontwikkeling van de applicatie, zoals: [MySQL Syntax controleren](https://nl.piliapp.com/mysql-syntax-check/), [Trello](https://trello.com/), [PHP validator](https://phpcodechecker.com/), [W3Schools](https://www.w3schools.com/), [JSON validator](https://jsonlint.com/), enz.

## Hardware informatie ontwikkelsysteem

De applicatie zal gebouwd worden op een werkcomputer die de opdrachtgever levert. Hieronder staat een tabel met hierin de hardwarecomponenten van het systeem:

|  |  |
| --- | --- |
| Component | Benaming |
| Processor | Intel® Core™ i7-2600 CPU @ 3.40GHz |
| Geïnstalleerd geheugen (RAM) | 8 GB |
| Type systeem | 64-bits besturingssysteem, x64-processor |
| Opslagcapaciteit | Samsung SSD 850 EVO 250 GB, Intel Raid 0 Volume 1 TB |
| Videokaart | AMD Radeon HD 5800 Series |

## Deelproducten

Het product is opgebouwd uit verschillende deelproducten, zoals:

* Uitgebreide versie van de database
* Leen functies
* Verkoop functies
* Instelling functies
* Spiegel functie
* Kledingkast uitbreidingen
* Colours Health koppeling
* Uitgebreide versie van de web API

# Hoofdstuk 2: Gegevensmodel

## Klassendiagram

Er zijn momenteel een aantal klassen aanwezig binnen de applicatie, hieronder staat het klassendiagram met de classes die toegevoegd worden voor de nieuwe functionaliteiten:



Er zal op maat gekeken worden binnen de vriendenkledingkast, hiervoor zal de applicatie de maten van de gebruiker ophalen en deze in “UserSizes” bewaren. Daarnaast zal in de class “Sizes” de lijst weergegeven worden voor alle verschillende maten. Deze verschillen per geslacht (vrouw of man) en regio (Europa, Verenigd koninkrijk of Amerika). Deze classes worden gebruikt bij het opslaan van de maten in de database, het ophalen van kleding en het toevoegen van kledingstukken. Met de functie “getAllShopItems” wordt er een lijst gemaakt waarin alle winkelproducten in worden opgeslagen.

Bij het ophalen van winkelproducten is de class “ShopItems” vereist. Hierin staat algemene informatie over het kledingstuk, zoals: van welke winkel het is, hoeveel het kost en om wat voor soort type kleding het gaat. Om het type op te halen, wordt een enumeration gebruikt.

Verder moeten er notificaties opgehaald worden binnen de app. Een notificatie zal opgeslagen worden in de class: “Notification”. Hierin staat: het ID, type, van welke gebruiker het af komt en het bericht van de notificatie. Daarnaast wordt er een enumeration gebruikt om aan te geven om welk soort type notificatie het gaat. Er zullen twee verschillende functies binnen de class gebruikt worden: getNotifications en setMessage. Binnen “setMessage” wordt het bericht aangepast op het type notificatie en wie deze notificatie verzonden heeft. Vervolgens kunnen andere bestanden een lijst van deze notificaties opvragen die te maken hebben met het ID van de gebruiker.

Tot slot hebben we de class “Settings”, hierin worden de notificatie-voorkeuren van de gebruiker opgeslagen. Deze kunnen aangepast worden binnen de instellingen van de applicatie.

## Datamodel huidige database

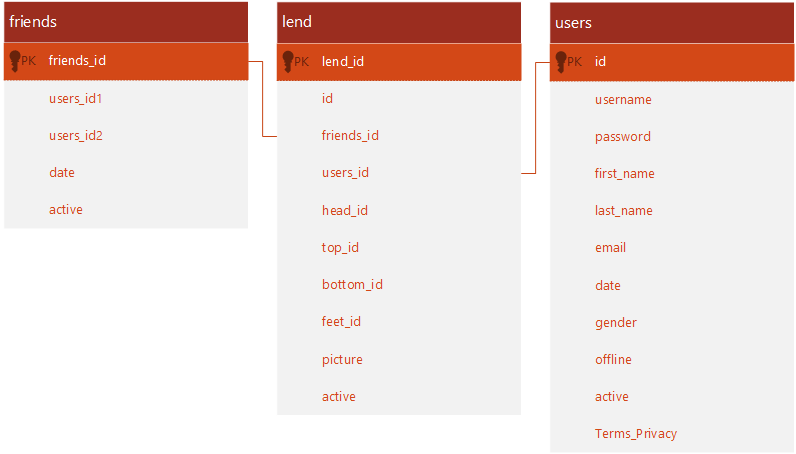
Het datamodel die ontworpen is door F. Demirci is terug te vinden in bijlage 1 op pagina 35. Aangezien dit model onoverzichtelijk, moeilijk leesbaar en verouderd is, is er een nieuw datamodel opgesteld met de huidige gegevens binnen de database. Daarna wordt er gekeken naar welke wijzigingen er zullen plaatsvinden om de nieuwe functionaliteiten toe te kunnen voegen.

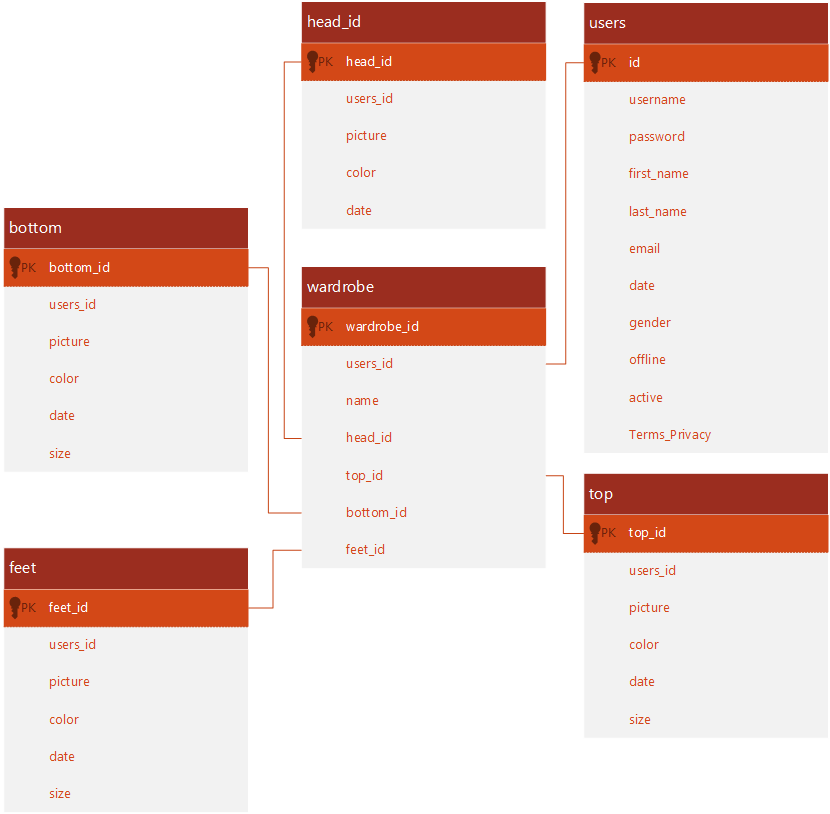
Bij vele van de database tabellen worden dezelfde waardes gebruikt (users\_id, head\_id, top\_id, bottom\_id en feet\_id), om het overzichtelijk te houden zijn deze tabellen niet altijd toegevoegd. De belangrijkste doel van dit document is om inzicht te bieden over hoe de applicatie technisch ontworpen worden, de datamodellen overzichtelijk houden is hierbij van groot belang.

Tevens zijn de tabellen onderverdeelt in verschillende datamodellen, om hierdoor verschillende functies weer te geven en de datamodellen leesbaar te houden. De datamodellen zitten in dit hoofdstuk inbegrepen, vergroten afbeeldingen van deze modellen zijn terug te vinden in bijlage 2 op pagina 36.

Hieronder ziet u het datamodel met de algemene gegevens van de gebruiker. Via het gebruikers ID (users\_id) zijn de tabellen met elkaar verbonden. In de “users” tabel worden voornamelijk de inloggegevens opgeslagen van de gebruiker. Daarnaast staan ok de voor- en achternaam in deze tabel. Gebruikers kunnen voorkeuren aangeven voor winkels, in de tabel “users\_shops” worden de relaties tussen de winkels en de gebruikers opgeslagen.

Informatie over deze winkel staan in twee verschillende tabellen: “rubric” en “shops”. De tabel “rubric” geeft aan in welke categorie de winkel valt. Algemene informatie over de winkel is terug te vinden in “shops”. Wanneer een vriendschap ontstaat tussen twee gebruikers, dan worden deze opgeslagen in de tabel “friends”. Hier worden de twee gebruikers ID’s opgeslagen. Doormiddel van een PHP-bestand wordt de longitude en latitude van de locatie, waar de gebruiker zich bevindt, gemeten, deze gegevens worden opgeslagen in de tabel “location”.

Hierlangs zie u de tabellen die vereist zijn voor de leenfunctie. Wanneer een gebruiker een item wilt lenen, wordt het verzoek opgeslagen in “lend”. Daarbij wordt het id van de eigenaar, id van de gebruiker en het id van het item opgeslagen. Dit kan het “head\_id”, “top\_id”, “bottom\_id” of “feet\_id” zijn. Daarnaast wordt het “friends\_id” ook in de “lend” tabel opgeslagen.

Gebruikers hebben de mogelijkheid om setjes op te slaan binnen de applicatie. Om deze gegevens op te slaan is de tabel “wardrobe”. Gebruikers kiezen een item van iedere categorie (head, top, bottom en feet). De ID’s van deze items worden opgeslagen in de tabel met het gebruikers id en de naam van de outfit.

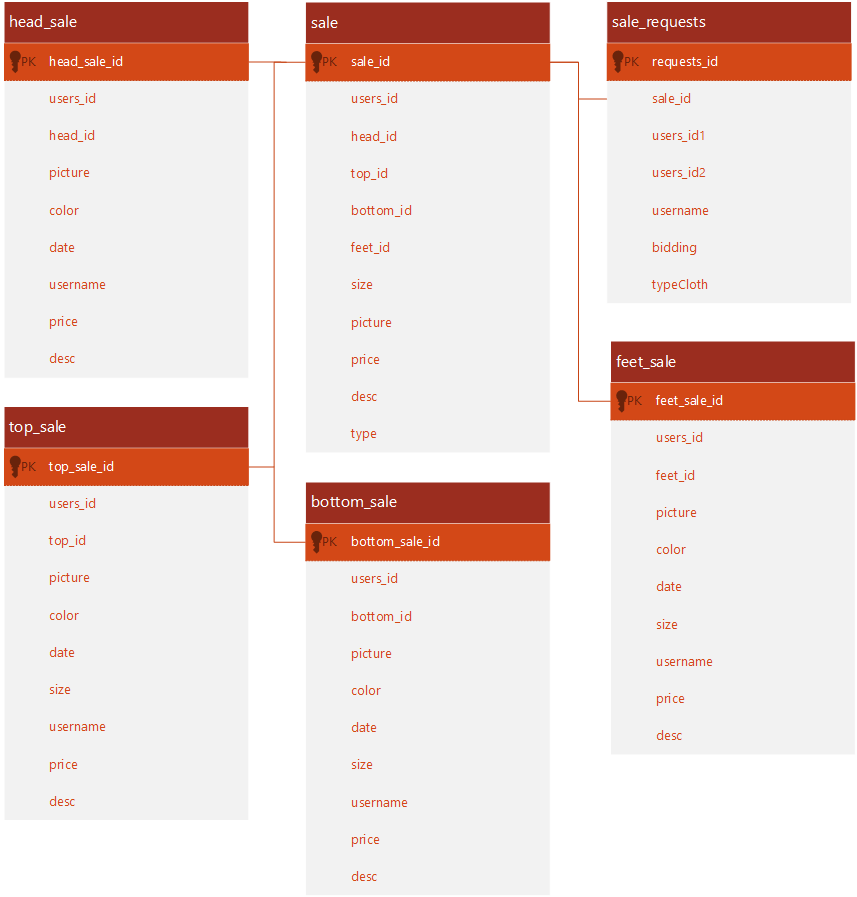
Deze tabel wordt overigens ook gebruikt bij het ophalen van de setjes.

Hier is waar het datamodel van F. Demirci niet overeenkomt met de werkelijke inhoud van de database. In het datamodel van F. Dimirci zijn er twee tabellen voor het verkopen van tweedehands kleding: “sale” en “sale\_requests”. In de huidige database zijn nog vier verschillende tabellen toegevoegd, genaamd: “head\_sale”, “top\_sale”, “bottom\_sale” en “feet\_sale”. Momenteel wordt de verkoopitems toegevoegd en opgehaald in deze tabellen. Maar bij het wijzigen van verkoopitems wordt de oude tabel nog steeds gebruikt. Dit heeft als gevolg dat niet alle functionaliteiten momenteel getest kunnen worden om te kijken of ze werkend zijn.

Tot slot staat hieronder het datamodel voor het opslaan van de sale en de kledingstukken. De kledingstukken worden opgeslagen in een van de volgende tabellen: “head”, “top”, “bottom” of “feet”. Deze tabellen worden overigens ook gebruikt bij het ophalen van kleding.

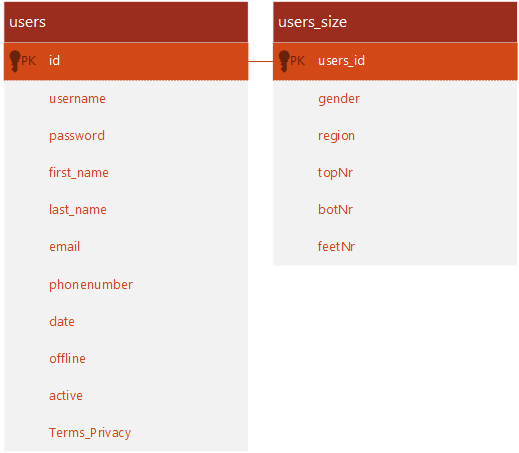
De “sale\_id” die opgeslagen worden in deze tabellen verschillen van de “sale\_id” binnen de “sale” tabel. Dit geeft de indruk dat hier geen relatie tussen zit.

## Toekomstige aanpassingen database

Er zullen enige wijzigingen plaatsvinden binnen de applicatie om de huidige te verbeteren en uit te breiden. Alleen de tabellen die toegevoegd of gewijzigd worden zullen hier besproken worden. Vergroten datamodellen van deze afbeeldingen zijn overigens in de bijlage(s) terug te vinden.

De code zal aangepast worden, zodat de sale\_id’s van de diverse tabellen overeen zullen komen met elkaar. Dit zorgt ervoor dat de functionaliteiten niet compleet opnieuw gemaakt moeten worden en alleen de toevoegfunctie gewijzigd wordt.

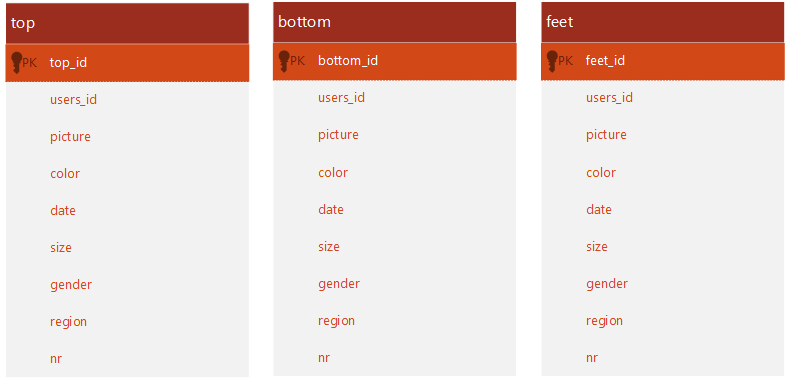
Overigens wordt er ook een rij toegevoegd aan de tabel “sale”. Deze rij, genaamd “type”, geeft aan om welk type kleding het gaat.

Wanneer de gebruiker in een kledingkast van zijn vrienden kijkt, laad de applicatie kledingstukken. Deze kledingstukken moeten dicht bij de maat liggen van de gebruiker, anders kan hij/zij deze kledingstukken niet lenen. Daarom zullen er enige wijzigingen plaatsvinden binnen de database.

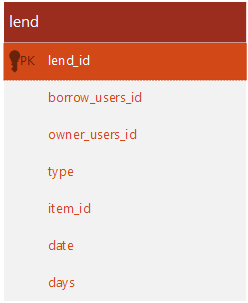
Aangezien de applicatie Engelstalig is, is ervan uitgegaan dat de applicatie internationaal bereikbaar zal zijn. Er zijn verschillende soorten maten die wereldwijd gebruikt worden. Om de applicatie zo universeel mogelijk te maken, worden er meerdere soorten maten gebruikt. Maten kunnen overigens ook verschillen per geslacht.

Bij de eerste keer opstarten, zal de applicatie de maten van de gebruiker opvragen, deze zullen in de “users\_size” tabel opgeslagen worden.

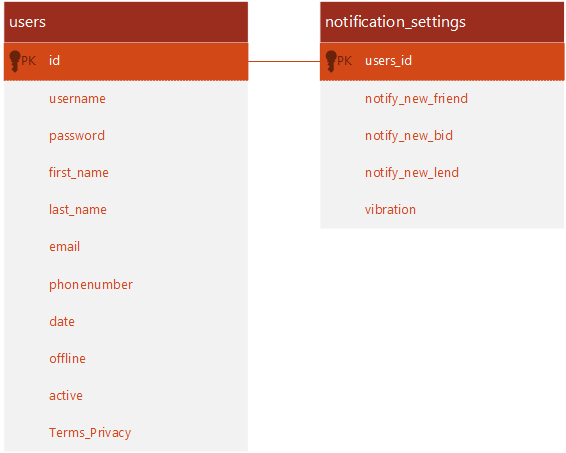
Daarnaast wordt de rij “gender” uit de tabel “users” gehaald en komt er een nieuwe rij voor in de plaats. Deze rij zal het telefoonnummer van de gebruiker opslaan, deze is vereist om ervoor te zorgen dat gebruikers elkaar kunnen contacteren.

Naast het toevoegen van een nieuwe tabel, zullen er ook enkele tabellen aangepast worden. De tabellen die aangepast moeten worden, zijn: “top”, “bottom” en “feet”. Bij het opslaan van deze items wordt ook het geslacht, de regio en het nr van de maat opgeslagen.

Er zullen drie nieuwe rijen toegevoegd worden aan de tabellen, genaamd: “gender”, “region” en “nr”. De normale maat zal alsnog opgeslagen worden, zodat andere functionaliteiten niet aangepast hoeven te worden.

Er zullen ook enkele wijzigingen plaatsvinden rondom het leensysteem. Enige rijen worden verwijderd of zullen door andere rijen vervangen worden. Aangezien het onduidelijk wat er in de rijen “id” of “users\_id” staan, worden benamingen aangepast. Hierdoor krijgt een buitenstaander meer duidelijkheid over welke gebruikers id bij welke gebruiker hoort. Hierdoor zullen een aantal functionaliteiten gewijzigd moeten worden.

Daarnaast zal het “friends\_id” niet nodig zijn bij het ophalen van de leenverzoeken. Het type kleding wordt opgegeven, waardoor er maar één rij nodig voor het kledingstuk ID i.p.v. vier. Verder geeft de persoon die iets wilt lenen aan op welke dag hij/zij het kledingstuk wilt lenen en hoeveel dagen het kledingstuk geleend wordt.

Tot slot komt er nog een nieuwe tabel in de database, deze zal de notificatie instellingen opslaan per gebruiker. Bij het aanmaken van een gebruiker zal een record aangemaakt moeten worden voor de gebruiker.

## Mappenstructuur Applicatie

Bij het ontwikkelen van de applicatie zullen er enkele mappen toegevoegd worden aan de huidige mappenstructuur. Hierlangs ziet u een overzicht van hoe de mappenstructuur in elkaar zit.

De map “Behavior” zal toegevoegd worden, hierin worden de behaviors toegevoegd die te gebruiken zijn bij bijvoorbeeld invoervelden. Hiervoor kan er gezorgd worden dat er een kleine e-mail validatie toegevoegd kan worden.

De map “Models” en “Views” zitten momenteel al in de applicatie. Binnen de map “Models” zullen nieuwe classes toegevoegd worden. Binnen de map “Views” staan de pagina’s van de applicatie. Er zullen enkele nieuwe pagina’s toegevoegd worden, zoals voor het opslaan van maten, het weergeven van de spiegelfunctie, enz.

Voor enkele van deze pagina’s is een sub map toegevoegd, hierdoor zal de applicatie overzichtelijk blijven.

## Mappenstructuur Web API

Momenteel zijn er al een aantal mappen aanwezig op de webserver, hiervan is een overzicht gecreëerd. Deze kunt is hieronder terug te zien:

Zoals hierlangs weergegeven zijn er diverse mappen die gebruikt worden op de web API. Aan de hand van deze gegevens zijn er genoeg mappen aanwezig om de functionaliteiten hierin toe te voegen, er hoeven dus geen nieuwe mappen gemaakt te worden op de webserver.

Daarnaast is het van belang om kennis te bemachtigen over wat er in iedere map geplaatst is. In de map “Account” komen de PHP-bestanden die te maken hebben met functionaliteiten omringt accountgegevens.

Vervolgens heb je de map “Email”, momenteel bevindt zich één PHP-bestand hierin. Deze is voor het versturen van mailberichten (deze functie wordt niet uitgevoerd binnen de applicatie). Binnen de map “Friends” zitten de functionaliteiten die te maken hebben met de vriendenlijst. Denk hierbij aan: het toevoegen van vrienden, verwijderen van vrienden, vriendenverzoeken versturen, enz. Vervolgens staat er in de map “Includes” het PHP-bestand die verbinding maakt met de database. Deze zijn vereist bij verschillende functionaliteiten binnen de webserver en een referentie naar dit bestand zal meerdere malen terug te vinden zijn in andere bestanden. In de map “Lend” zitten functies voor het versturen, verwijderen en ophalen van leenverzoeken. Binnen de map “Location” wordt de locatie van de gebruiker opgehaald. Vervolgens staat in “Notifications” een PHP-bestand die vriendverzoeken ophaalt.

Binnen de map “Sale” is een sub map toegevoegd: Type. In deze map zitten de verschillende kledingtypes, zoals: bottom, feet, head en top, als sub mappen. De map “Shops” heeft een map met hierin afbeeldingen van de verschillende winkels die de gebruiker kan kiezen als voordeel.

Ook bij de map “Wardrobe” zijn er enkele mappen met sub mappen te vinden. Hierbij worden de afbeeldingen van een kledingtype opgeslagen in de “Img” map en zijn er meerdere PHP-bestanden te vinden voor het verwijderen, ophalen en toevoegen van een kledingstuk.

De mappenstructuur van de webserver zal niet aangepast worden bij het toevoegen/aanpassen van de functionaliteiten. Indien dit wel het geval blijkt te zijn, zullen de aanpassingen genoteerd staan in de technische documentatie die opgesteld aan het einde van de realisatiefase.

Tot slot zijn er nog twee mappen te vinden binnen de mappenstructuur, WardrobeFriends en WardrobeSet. Binnen deze mappen worden de kledingstukken van vrienden opgehaald en de opgeslagen setjes.

# Hoofdstuk 3: Functionaliteiten

Binnen dit hoofdstuk zullen de hoofdfunctionaliteiten die gewijzigd/toegevoegd worden uitgewerkt staan in verschillende diagrammen of schematische weergaves. Hierdoor wordt er duidelijkheid gecreëerd over hoe de functionaliteiten in elkaar gaan zitten.

## Use-cases

Er zijn een aantal use-cases ontworpen, deze zijn overigens verdeelt onder vier verschillende categorieën: algemene functies, maat functies, kleding functies en instelling functies.

Bij de algemene functies behoren twee functionaliteiten: licht instellen bij de spiegelfunctie en de achtergrondkleur aanpassen op gevoel. De gebruiker zal zich terug kunnen zien in een spiegel functie, waarbij de camera gebruikt wordt met hieromheen een wit randje. Het zal mogelijk zijn om de intensiviteit van de witte rand aan te passen aan de voorkeur van de gebruiker.

Om een koppeling te maken met Colours Health zal de gebruiker een optie krijgen om een kleur op te vragen aan de hand van een gevoel die hij/zij wilt voelen.

Binnen de applicatie zal het mogelijk worden om je eigen maat op te geven. Wanneer een gebruiker voor het eerst inlogt binnen de applicatie, moet hij/zij zijn winkel voorkeuren opgeven. Na deze pagina zal de gebruiker vereist zijn om zijn kledingmaten op te geven. Er zullen verschillende soorten maten opgeslagen worden aan de hand van het geslacht en de regio.

Bij het opslaan van kleding zullen deze maten opnieuw ingevoerd worden, zodat deze kledingstukken gefilterd kunnen worden binnen de vrienden kledingkast.

Er wordt gekeken naar de opgegeven maat van de gebruiker. De kleding stukken die dezelfde maat hebben of een maat groter of smaller worden opgehaald binnen de vrienden kledingkast.

Daarnaast worden er enkele kleding functies toegevoegd en aangepast. Het lenen van kleding wordt gewijzigd, er zal een ander scherm gemaakt worden die meer zal lijken op het scherm voor het verkopen van tweedehands kleding. Verder worden er enkele wijzigingen gemaakt bij het contact opnemen.

Kleding kan niet alleen gefilterd worden op maat, maar ook op kleur. Dit wordt gedaan door een type kleding op te vragen aan de gebruiker en hij/zij vervolgens een kleur te laten kiezen. Hierna zal er een zoekfunctie plaatsvinden en de lijst met kleding wordt aangepast.

Tot slot komen de instelling functies aan bod, deze zijn nieuwe functionaliteiten die toegevoegd worden. Een gebruiker zal aan kunnen geven over welke notificaties zij een bericht willen hebben, deze voorkeuren zullen standaard op aan staan. Dit wordt vervolgens opgeslagen in de database.

Vervolgens kan de gebruiker ook enkele gegevens wijzigen, zoals: zijn wachtwoord, persoonlijke gegevens of gebruikersnaam. Alhoewel deze functionaliteiten alle drie over gegevens wijzigen gaat, zijn er diverse uitzonderingen en afwijkingen binnen het wijzigen van de gegevens. Vandaar dat deze functies in aparte use-cases weergegeven worden.

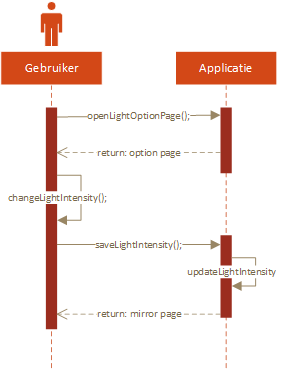
Bij het wijzigen van het wachtwoord wordt de gebruiker ten eerste vereist het oude wachtwoord op te geven. Hierna kan hij/zij een nieuw wachtwoord invoeren.

Wanneer een gebruiker zijn persoonlijke gegevens wilt wijzigen, zal allereerst de huidige gegevens opgehaald worden binnen de database, hierbij worden de invoervelden gevuld. De gebruiker kan deze vervolgens aanpassen en opslaan. De applicatie zal eerst een check doen om te kijken of alle vereiste informatie niet leeg is, als alle vereiste gegevens is meegegeven worden de wijzigingen toegepast binnen de database.

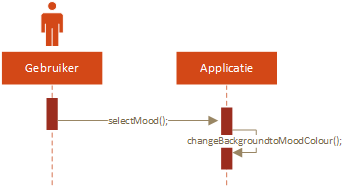
Daarnaast krijgt de gebruiker de mogelijkheid om zijn gebruikersnaam te wijzigen. Het is van belang dat de applicatie controleert of de gebruikersnaam al bestaat binnen de applicatie, dit geldt ook voor het wijzigen van een e-mailadres. Wanneer de gebruikersnaam voldoet aan de eisen zal deze veranderd worden binnen de applicatie. Aangezien de gegevens van de gebruiker momenteel in een class opgeslagen worden, zal gegevens ook gewijzigd moeten worden in de class.

## Sequence diagrammen

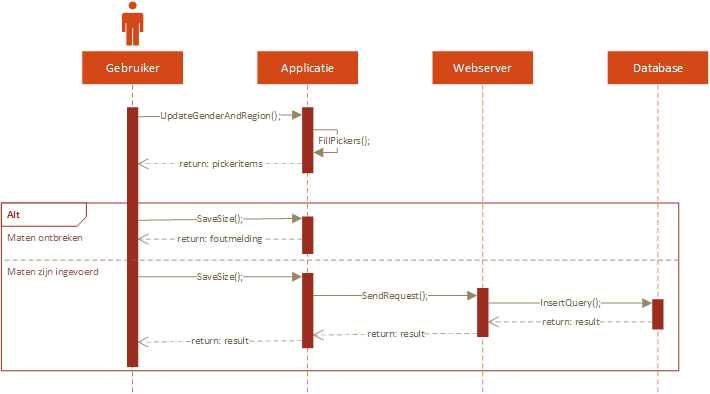
Aan de hand van de use-cases zijn er enkele sequence diagrammen ontworpen om duidelijkheid te schetsen over welke objecten met elkaar communiceren bij de functionaliteiten.



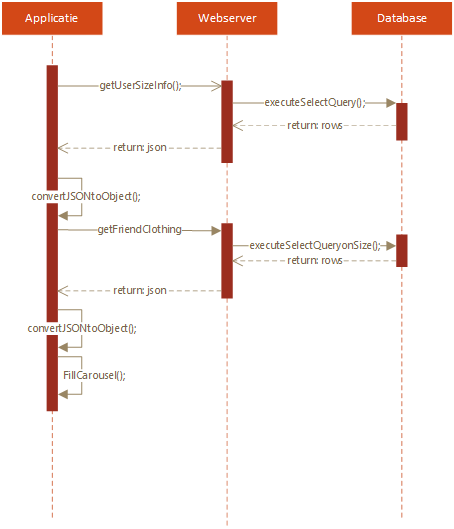
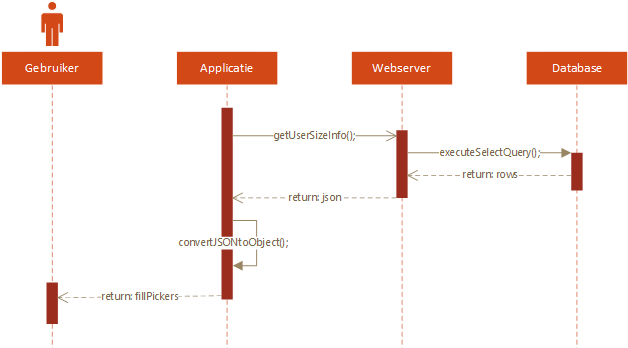
Zoals al eerder aangegeven bij de use-cases, zullen gebruikers de mogelijkheid krijgen om de witte rand rondom de camera in te stellen. Hierlangs is te zien hoe dit proces te werk gaat en welke interacties er plaatsvinden.



Bij het selecteren van een emotie zal de applicatie de achtergrondkleur van de pagina aanpassen. Er twee verschillende objecten die met elkaar communiceren: de gebruiker en applicatie. Hierlangs is de sequence diagram van het achtergrond instellen te zien.

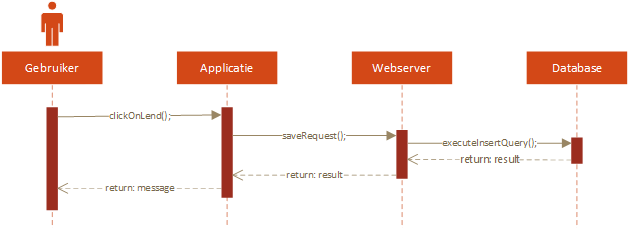
Hieronder is de sequence diagram voor het opslaan van de maten te zien. Zoals te zien is zijn er meerdere objecten die met elkaar communiceren in deze diagram. De gebruiker, applicatie, webserver en de database communiceren onderling met elkaar om de functionaliteit uit te voeren. De webserver is vereist voor het   
uitvoeren van database functies en voor het ophalen van de resultaten.

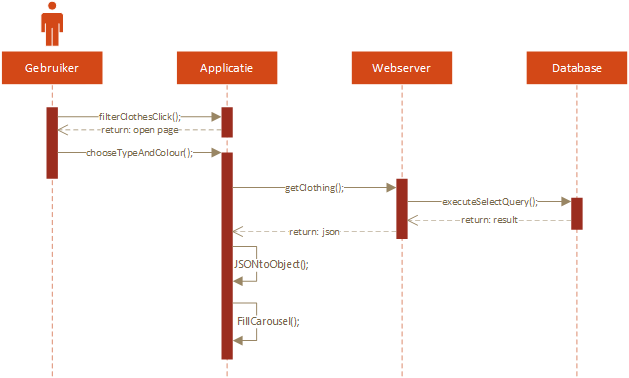
Wanneer de gebruiker een kledingstuk wilt toevoegen in zijn kledingkast, zal er allereerst opgehaald worden welk geslacht de gebruiker is en uit welke regio hij/zij komt. Deze gegevens worden opgevraagd bij de webserver die vervolgens een select query uitvoert. Het resultaat wordt teruggegeven in een JSON-formaat en deze worden hierna door de applicatie verwerkt tot een object. De applicatie zal de juiste waardes selecteren in de “pickers”. Wanneer de gebruiker deze wijzigt, zullen de maten in de pickers gewijzigd worden.

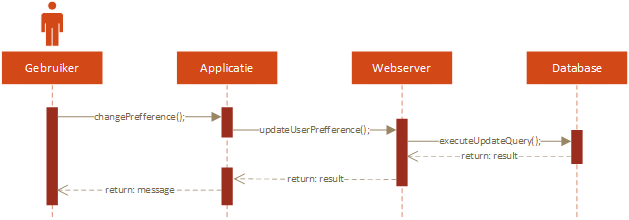
Gebruikers kunnen de kleding binnen de kledingkast van hun vrienden bekijken. Aangezien de applicatie om het lenen van kleding gaat, zal er op de maat van het kledingstuk gelet worden.

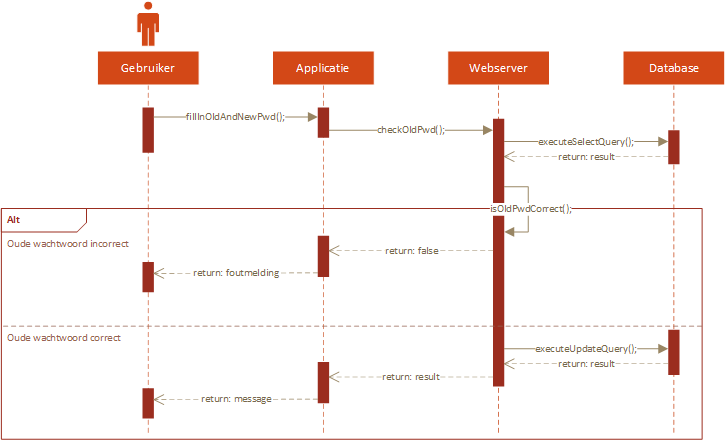
Alleen de kleding die dezelfde maar of een maat groter of smaller is worden weergegeven, hierlangs is te zien hoe dit met interacties gebeurt.

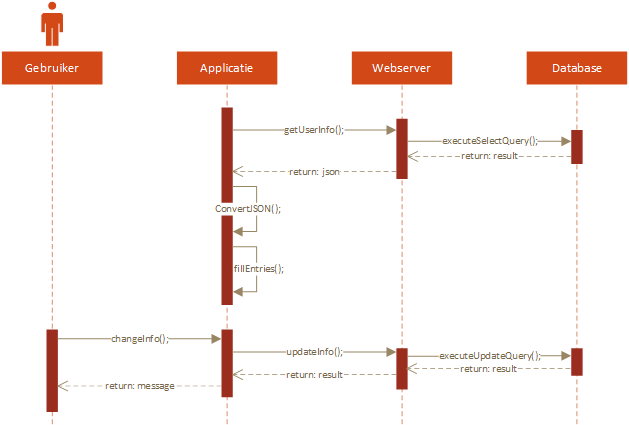
Ten eerste worden de gegevens van de gebruikers opgehaald, deze gegevens worden vervolgens gebruikt bij het ophalen van de kleding. In de select query wordt gekeken welke kleding bij de zoekopdracht past. Beide resultaten worden teruggegeven in JSON-formaat en de applicatie verwerkt deze informatie tot een object.

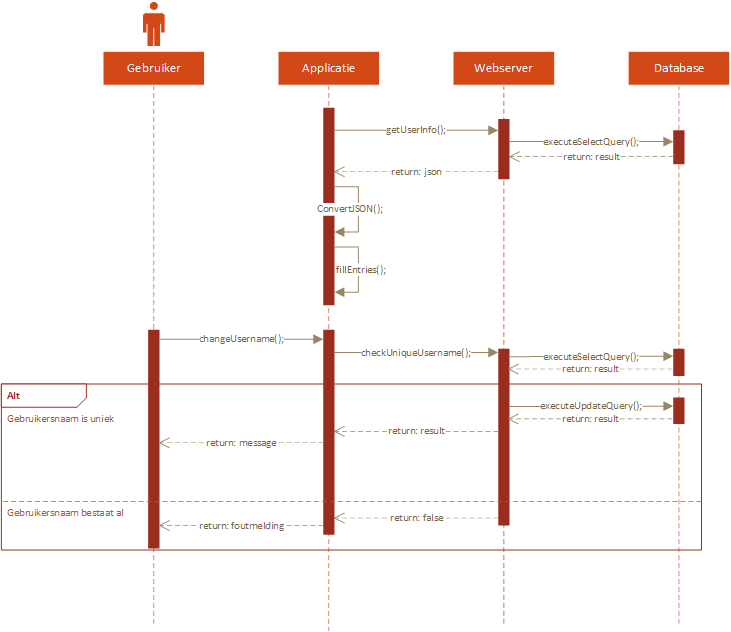
Bij het versturen van een leenverzoek zal de applicatie via de webserver het verzoek opslaan in de database. Aan de hand van het resultaat die de applicatie terugkrijgt, krijgt de gebruiker hier een melding over. Het verzoek kan succesvol zijn of gefaald zijn.

Wanneer kleding op kleur gefilterd wordt de gebruiker ten eerste gevraagd om aan te geven op welk type en welke kleur zij willen filteren. Hierna zal de applicatie via de webserver een select query uitvoeren. De gegevens van de opgehaalde kledingstukken worden vervolgens via JSON verwerkt en teruggestuurd naar de applicatie. Hier worden ze verwerkt in een lijst en vervolgens in de “carouselview” weergegeven.

Gebruikers krijgen de mogelijkheid om notificaties aan of uit te zetten. Dit kunnen ze doen door de voorkeur aan te passen en op te slaan. De applicatie haalt de gegevens op, stuurt deze naar de webserver en de webserver zal de record in de database aanpassen via een update query. De database zal aangeven of het gelukt of mislukt is en geeft dit door aan de webserver. Die vervolgens het resultaat teruggeeft aan de applicatie. Aan de hand van het resultaat zal de applicatie een gepaste melding weergeven aan de gebruiker.Hieronder ziet u de sequence diagram van het wachtwoord wijzigen. Er zal allereerst gecheckt worden of het opgegeven oud wachtwoord overeenkomt met het wachtwoord in de database. Als dit het geval is, dan wordt de update query past uitgevoerd. Ook hier worden de database functies uitgevoerd via de webserver. Het kan zijn dat het opgegeven wachtwoord niet overeenkomt met het huidig wachtwoord, de gebruiker zal hier een melding over krijgen en het juiste wachtwoord op moeten geven.



Als allereerst zal de applicatie gegevens opvragen aan de webserver, de webserver voert vervolgens een select query uit op de database. De opgehaalde gegevens worden via JSON verwerkt en naar de applicatie teruggestuurd. De applicatie zal deze gegevens omzetten en in diverse entries plaatsen. Hier kan de gebruiker wijzigingen aan passen en op “opslaan” drukken. Vervolgens zal er een update query uitgevoerd worden op de database en het resultaat zal terugkomen bij de gebruiker in een melding.

Tot slot het laatste sequence diagram die ontworpen is, deze geeft weer hoe de interacties tussen verschillende objecten tot stand komen bij het wijzigen van een gebruikersnaam. Allereerst zal de huidige gebruikersnaam opgehaald worden via de database en deze wordt vervolgens in de entry geplaatst. Vervolgens zal de gebruiker een ander gebruikersnaam invoeren en deze proberen op te slaan. Ten eerste wordt er een check uitgevoerd om er zeker van te zijn dat er geen dubbele gebruikersnamen ontstaan.

Wanneer hij niet uniek is, zal de gebruikersnaam niet gewijzigd worden en krijgt de gebruiker een foutmelding. Als de gebruikersnaam wel uniek is, zal deze via een update query gewijzigd worden. Het resultaat wordt via een melding weergegeven aan de gebruiker.

## Flowcharts

In het functioneel ontwerp, hoofdstuk 3, zijn een aantal processen uitgewerkt in flowcharts. Deze flowcharts waren gericht op de gebruikers en enige klanten die het document zouden lezen. Enige flowcharts van het functioneel ontwerp zijn overgenomen en uitgebreid.

Er zijn zes verschillende flowcharts opgesteld om de functionaliteiten te verduidelijken. Hieronder staan deze flowcharts met hierbij verdere uitleg:

**Lichtfunctie**

Bij het opstarten van de MirrorPage zal er eerst gekeken worden of er al een lichtwaarde is opgegeven, deze staat standaard op null.

Wanneer dit het geval is, zal de applicatie geen andere lichtwaarde instellen. Als er wel een lichtwaarde is opgeslagen, dan zal de applicatie het witte randje aanpassen.

Wanneer een gebruiker de lichtwaarde wilt wijzigen, zal hij/zij eerst moeten navigeren naar de AdjustLightPage. Hier kunnen zij via een schuifbalk het licht aanpassen. De applicatie slaat deze waarde op in de LoginCredentials en stuurt de gebruiker terug naar de MirrorPage.



**Colours Health koppeling**

De gebruiker krijgt een lijst met verschillende gevoelens of emoties voor zich, wanneer hij een gevoel kiest zal de applicatie de achtergrondkleur van de huidige pagina aanpassen. Dit doet de applicatie door te checken of de SelectedIndex is veranderd.



**Kleurenfilter op de kledingkast**

Gebruikers selecteren allereerst een type en kleur en vervolgens drukken zij op “filter”. De applicatie zal hierna een check uitvoeren of er een type en kleur is gekozen. Hierna zal hij deze gegevens opsturen naar de webserver die een select query uitvoert.

Hier zullen alleen de kledingstukken die de gekozen kleur hebben teruggestuurd worden via JSON-formaat.

Deze zal de applicatie vervolgens verwerken in een lijst en zullen de zoekresultaten terug te vinden zijn in de carouselview.



Naast het ophalen van zoekresultaten zal er ook een “reset” optie gemaakt worden. Het zal mogelijk zijn voor de gebruikers om de filters te verwijderen en alle kleding te zien krijgen.



**Leenverzoek versturen**

Bij het versturen van leenver-zoeken geeft de gebruiker de datum en het aantal dagen op wanneer en hoelang hij/zij het kledingstuk wilt lenen.

De applicatie zal nakijken of beide gegevens ingevuld zijn, voordat er een verzoek gestuurd wordt naar de webserver.

Vervolgens zal de webserver een insert query uitvoeren en het resultaat teruggeven aan de applicatie.

De applicatie geeft tot slot een melding met hierin het resultaat.

**Maat opslaan**

Bij het opslaan van de maat zal de gebruiker via pickers zijn geslacht en regio aangeven. De maten binnen de overige drie pickers zullen zich aanpassen op geslacht en regio.

Hierna kiest de gebruiker zijn drie maten. De applicatie checkt na of deze werkelijk ingevuld zijn en stuurt het verzoek vervolgens door aan de webserver. Deze slaat de nieuwe gegevens op via een insert query en geeft het resultaat terug aan de applicatie. De applicatie geeft tot slot een melding aan de gebruiker en geeft aan of het gelukt of mislukt it.

**Gegevens wijzigen**

Bij het wijzigen van de gegevens wordt dit proces gebruikt. Er zullen kleine uitzonderingen zijn bij het wijzigen van een wachtwoord of gebruikersnaam, maar deze stappen worden merendeels doorlopen.

De applicatie haalt gegevens op en vult deze in binnen de juist entries. Hierna kan de gebruiker deze gegeven aanpassen door andere waardes in te voeren.

Wanneer de gebruiker op “save changes” drukt zullen de gegevens binnen de database via de webserver gewijzigd worden. Tot slot zal er een melding weergegeven worden.

# Hoofdstuk 4: Grafische user interface

## Navigatiestructuur

De navigatiestructuur is verdeeld in diverse categorieën. De gehele mappen structuur is terug te vinden in bijlage 4 op pagina 45. Daarnaast bestaat de navigatiestructuur alleen over de wijzigingen en functionaliteiten die aangepast/toegevoegd worden. Dit is niet de gehele navigatiestructuur van de applicatie.

**Maat opslaan**

Wanneer een gebruiker voor de eerste keer inlogt, zal de applicatie nagaan of hij/zij al een voorkeur heeft opgegeven voor winkels. Zo niet, opent de pagina “ChoseShopPage” waarbij hij zijn voorkeuren kan kiezen. Hierna zal de gebruiker op de pagina “SaveSizePage” komen en zijn maten opgeven. Wanneer een gebruiker wel zijn winkel en maten heeft opgeslagen, dan opent de pagina “MenuPage”.

**Menu navigatie**

Er komen drie nieuwe pagina’s die te bereiken zijn via de “MenuPage”, genaamd: ColoursHealthPage, NotificationPage en MirrorPage. Daarnaast worden er enkele wijzigingen toegepast aan de volgende bestaande pagina’s: WardrobePage en SettingPage.



**Leensysteem en Filterfunctie**

Er wordt momenteel via de volgende stappen genavigeerd naar het leensysteem: SignPage -> LoginPage -> MenuPage -> WardrobePage -> WardrobeChoose -> WardrobeFriends -> LendPage. Dit is zo onduidelijk geweest dat bij het testen van de applicatie de testpersoon deze pagina niet heeft kunnen vinden. Daarom zal er veranderingen plaatsvinden om naar deze pagina te navigeren.

Daarnaast zullen gebruikers via de WardrobePage ook naar de FilterWardrobe pagina kunnen navigeren.



**Notificatiescherm**

Via de MenuPage kan er genavigeerd worden naar de NotificationPage. Hierbij kunnen er drie mogelijkheden worden gemaakt, er wordt genavigeerd naar: WardrobeSaleRequests, FriendRequest of LendRequests.

Dit heeft te maken met welke notificatie de gebruiker op klikt.



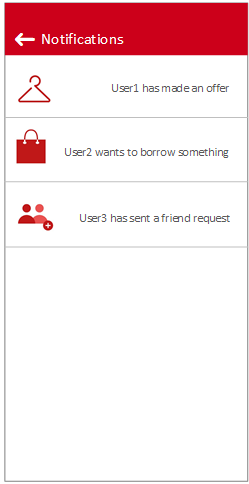
**Instellingen**

Tot slot worden er enkele wijzigingen toegepast in de structuur van de instellingen. Er zullen meerdere pagina’s toegevoegd worden aan de applicatie, zoals: EditAccount, EditNotifications, ChangeBackground, Help, Privacy en Logout.

Bij de pagina EditAccount zitten nog drie verschillende pagina’s aan vast. ChangeInfo, ChangePassword en ChangeUsername zijn te bereiken via EditAccount.

## Schetsen

Deze schetsen zijn nogmaals terug te vinden in hoofdstuk 6 van het Functioneel Ontwerp met uitleg voor gebruikers/klanten.

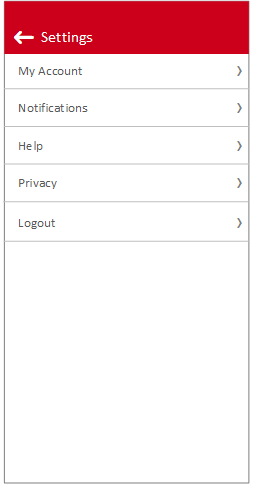


**Notificatiescherm**

Er zal een nieuwe pagina ontworpen worden voor het weergeven van de notificaties binnen de applicatie. Tevens zal er een klein icoontje toegevoegd worden op de startpagina van de applicatie, hierdoor kan een gebruiker naar het notificatiescherm navigeren.

Er zijn momenteel drie verwachte notificaties: iemand stuurt een vriendenverzoek, iemand wilt iets lenen of iemand heeft een bod gedaan op een kledingstuk die verkocht wordt. Er zullen drie verschillende icoontjes geplaatst worden voor het bericht aan de hand van het type notificatie.

De gebruikersnaam zal weergegeven staan in het bericht waarvan het verzoek vandaan komt, daarnaast komt een gepast bericht te staan. Wanneer een gebruiker op een notificatie klikt, zal de applicatie de juiste pagina openen, zoals: FriendsPage, WardrobeSelling of LendRequests.



**Instelling pagina(s)**

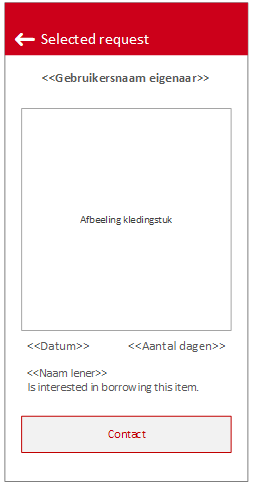
De huidige instelling pagina zal compleet gewijzigd worden aangezien er verder ook geen functionaliteiten binnen deze pagina plaatsvinden als alleen de uitlog functie. De instellingen zullen in diverse categorieën verdeeld worden.

Zoals hierlangs te zien is, zal de gebruiker op een van deze categorieën kunnen klikken en de instelling functies uit kunnen voeren. Er zullen dus enkele nieuwe pagina’s toegevoegd moeten worden aan de hand van de instellingen. In hoofdstuk 3 onder het kopje “use-cases” zullen de instelling functionaliteiten uitgelegd staan.

**Leenverzoeken**

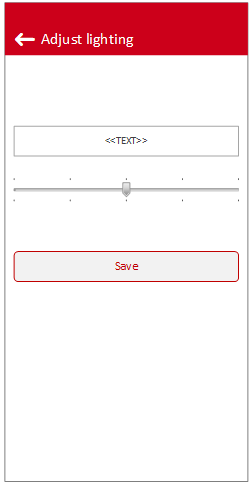
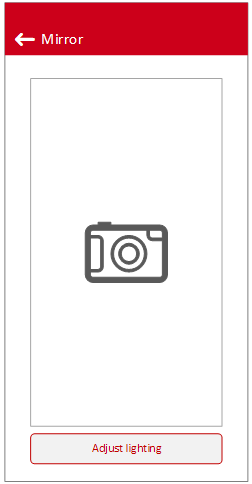
De pagina’s voor de leenverzoeken zullen overeenkomen met de verkooppagina’s die momenteel gebruikt worden binnen de applicatie. Dit zal de indruk geven dat de applicatie samenhangend is en niet de indruk geeft dat verschillende applicatieontwikkelaars eraan gewerkt hebben.

De gebruiker zal een afbeelding krijgen om gelijk in te zien om welk kledingstuk het gaat. Verder komt de gebruikersnaam van de lener, die het kledingstuk wilt lenen, op het request te staan. Verder zal onderaan de datum te zien zijn wanneer de lener het kledingstuk wilt lenen.



Wanneer de gebruiker op het verzoek klikt, zal er een nieuwe pagina geopend worden. Als gebruikers een verzoek willen verwijderen, zullen ze (net zoals bij het verwijderen van een koopverzoek) het verzoek in moeten drukken en vervolgens op “delete” drukken.

In deze pagina komt een overzicht met informatie over het leenverzoek. Er wordt duidelijk weergegeven op welke datum de lener het kleding stuk wilt lenen, voor hoeveel dagen hij/zij deze wilt lenen, een afbeelding van het kledingstuk en er komt een optie om contact op te nemen met de persoon die het kledingstuk wilt lenen.

**Spiegelfunctie**

De camera van de telefoon wordt weergegeven met hieromheen een witte rand. Deze is in te stellen op de rechterpagina. De gebruiker zal via een schuifbalk het licht kunnen instellen en terug navigeren naar de “mirror” pagina.



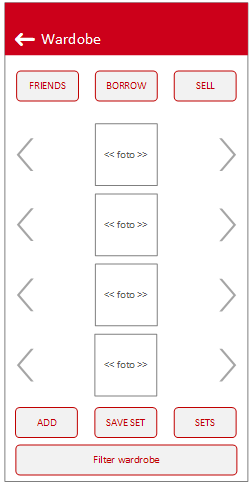
**Achtergrondkleur op gevoel**

Er zal een koppeling gemaakt worden met de website “Colours Health”. Colours Health is een website ontwikkeld voor mensen te helpen en deze beter te laten voelen. Door middel met het gebruik van kleuren worden er prikkels verzonden en deze kunnen pijn verlichten of helpen met het voelen van een emotie. Bezoek [de website](http://colourshealth.com/#nl/) voor meer informatie over Colours Health.

De opdrachtgever wilt dat Goodlookz een koppeling krijgt met Colours Health door een soortgelijke functionaliteit in de applicatie te bouwen.

Hierlangs ziet u het verwachte uiterlijk van de toekomstige pagina. Er zal een kleine uitleg gegeven worden over Colours Health en een optie om naar de website te navigeren.

Daarnaast zullen gebruikers de mogelijkheid krijgen om een gevoel uit te kiezen en deze bekijken. Langs deze opties zullen voorbeelden van deze kleuren komen staan.

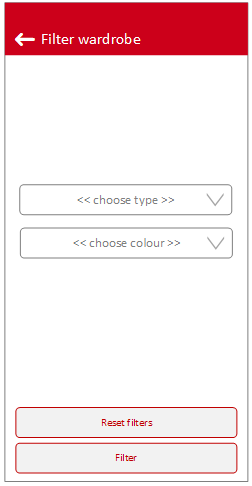


**Kleurenfilter kledingkast**

De filterfunctie wordt ingebouwd als een uitbreiding in de virtuele kledingkast, er zullen enkele knoppen toegevoegd worden op de kledingkast pagina. Dit is mede doordat de nagivatiestructuur voor de leenfuncties onhandig gemaakt zijn. Hierlangs is de nieuwe opbouw van de wardrobe pagina te zien.

Er worden twee knoppen toegevoegd aan deze pagina: “Filter wardrobe” en “Borrow”. Enkele teksten binnen de knoppen worden tevens ook aangepast. De “Borrow”-knop zal gebruikt worden bij het leensysteem die ontwikkeld wordt.

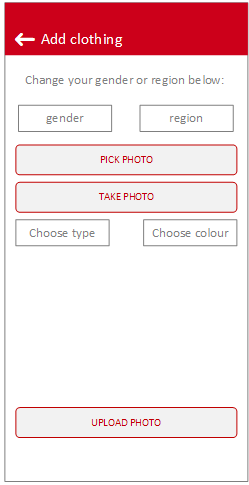
Wanneer gebruikers op de knop “Filter wardrobe” drukken, zal de applicatie de volgende pagina openen:



Gebruikers zullen eerst aangeven op voor welk type kleding zij willen filteren via een picker of combobox, hierna worden de kleuren weergeven in de volgende combobox. Er worden twee gegevens meegegeven aan de zoekopdracht: type kleding en kleur.

Als de gebruiker aan heeft gegeven welk kledingtype en welke kleuren hij/zij op wilt filteren, kunnen ze op de knop “filter” drukken. De applicatie brengt de gebruiker terug op de “wardrobe” pagina maar de kledingstukken worden nu gefilterd. De huidige lijst van het kledingstuk wordt leeggemaakt en gevuld met de opgehaalde zoekresultaten.

Daarnaast kunnen gebruikers alle filters resetten via de knop “reset filters”.

****

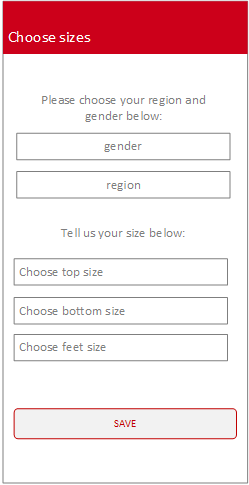
**Vrienden kledingkast filteren op maat**

Bij het ophalen van kleding van de vrienden kledingkast zal alleen de kleding weergegeven worden die een maat groter, een maat smaller of dezelfde maat als de gebruiker zijn.

Allereerst worden er gegevens opgehaald van de gebruiker, zoals: de regio en het geslacht. Deze worden hierna gekozen in de twee bovenstaande pickers.

Gebruikers kunnen deze natuurlijk aan passen, indien nodig. Maar het is handig als deze automatisch op de juiste waardes staan.

Hierna kunnen gebruikers het type kleding kiezen en hierna een maat opgeven die voldoen aan de twee bovenstaande waardes.



Daarnaast zal er een nieuw scherm ontworpen worden voor het opslaan van de maten. Bij het eerste keer opstarten, moet de gebruiker voorkeuren aangeven van winkels. Na deze pagina zal er nog een pagina openen waarbij de gebruiker vereist is om zijn regio en geslacht op te geven. Daarnaast geeft de gebruiker aan welke maten hij/zij heeft voor een shirt, broek en schoenen. Deze zullen later gebruikt worden bij het ophalen van kledingstukken.

Aan de hand van het gekozen geslacht en regio veranderen de maten die de gebruiker kan kiezen. De gebruiker moet bij iedere invoerveld een maat kiezen. Wanneer de gebruiker een invoerveld vergeet, zal de applicatie dit melden als de maat opgeslagen moet worden.

Daarnaast zal de “terug” knop uitgeschakeld zijn om enige fouten te voorkomen.

# Nawoord

Hopelijk heeft dit document u voldoende geïnformeerd over de technische gedeelte van de toekomstige functionaliteiten. Als u nog enige opmerkingen en/of vragen heeft, dan kunt u deze doormailen naar het onderstaande mailadres:

* [rebeccabroens@gmail.com](mailto:rebeccabroens@gmail.com)

# Bijlage(s)

## Bijlage 1: Datamodel ontworpen door F. Demirci

## Bijlage 2: Datamodel huidige database



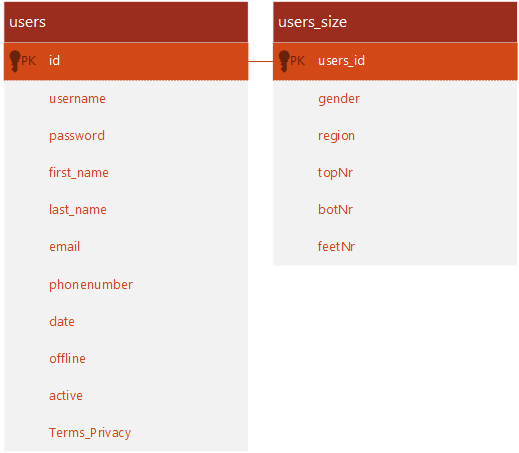






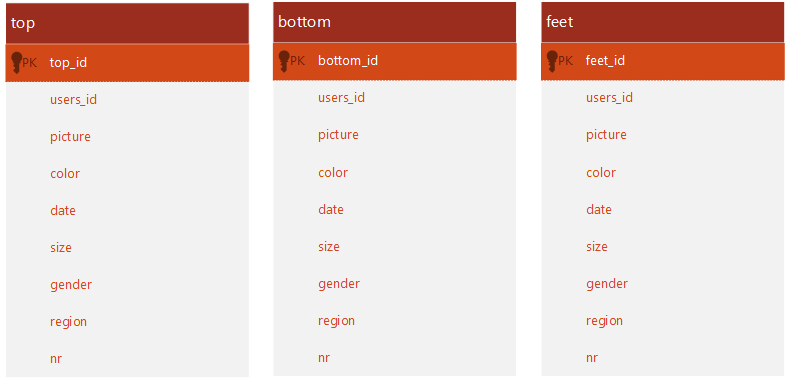


## Bijlage 3: Datamodel Aanpassingen database











## Bijlage 4: Complete mappen structuur