> Rapportage UX onderzoek

Nu we weten hoe de webapplicatie er functioneel en technisch uitziet, is het wijsheid om te onderzoeken in welke mate gebruikers hun doelen met de webapplicatie effectief, efficiënt en naar tevredenheid kunnen bereiken. Dit doen we door de user experience van de webapplicatie in kaart te brengen m.b.v. de volgende vier hoofdvagen.

1. Doel: Waarom doen we dit onderzoek, wat hopen we eruit te halen?

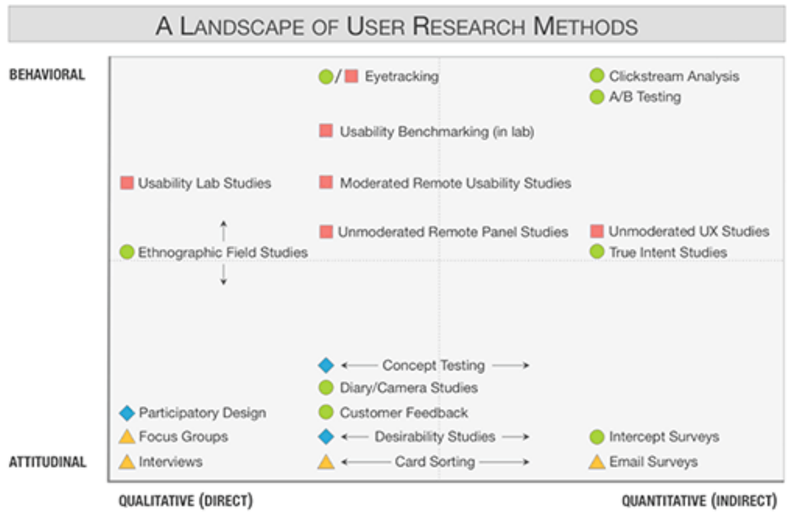
We willen in kaart brengen wie onze gebruikers zijn en wat hun behoeftes zijn, om zo tot de ultieme gebruikerservaring te komen.

2. Onderzoeksvragen: Wat zijn de specifieke vragen waarin we geïnteresseerd zijn?

> Wat zijn de algemene demografische gegevens van onze gebruikers?
> Wat is de toon die past bij onze gebruikers?
> Wat is het gedrag van onze gebruikers?
> Welke sites en platforms gebruiken onze gebruikers?
> Hoe serieus zijn onze gebruikers met muziek bezig?
> Via welke kanalen vinden de gebruikers de webapplicatie?
> Via welk device bezoeken gebruikers de huidige webapplicatie?
> Hoe intensief wordt de huidige webapplicatie gebruikt?
> Wie maken er nu gebruik van de huidige webapplicatie?
> Welke functies willen de gebruikers graag terugzien in de webapplicatie?

3. Methode: Welke specifieke methode willen we gebruiken om meer te weten te komen over de gebruiker?

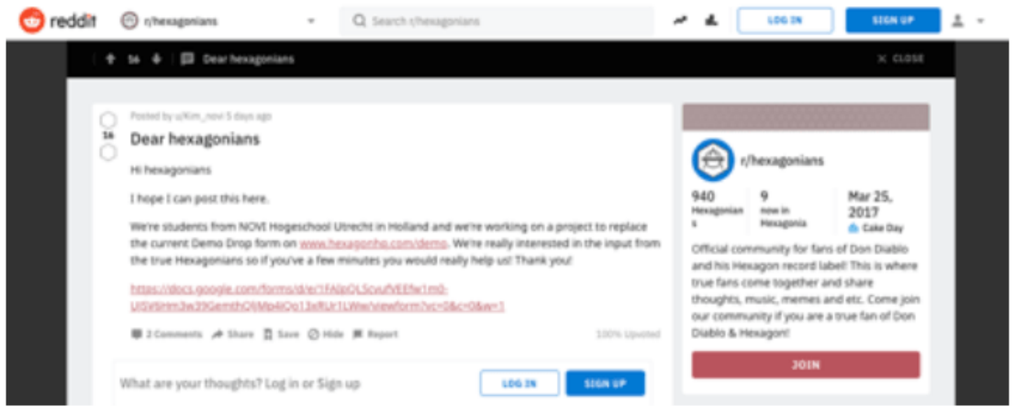
We hebben gekozen voor attitudinaal onderzoek in de vorm van een enquête. Het doel van attitudinaal (dat wat mensen zeggen) onderzoek is om de verklaarde overtuigingen van mensen te begrijpen en te meten. De enquête is daarbij uitermate geschikt om attitudes te meten en categoriseren en daarmee gegevens te verzamelen die kunnen helpen bij het volgen of ontdekken van belangrijke problemen die moeten worden aangepakt.



afb. 6

In de matrix is te zien dat de enquête (survey) zich rechtsonder bevindt. Dit betekent dat deze methode kwantitatief (indirect) van aard is. Dit hielp ons om in korte tijd zoveel mogelijk respondenten in ons onderzoek te betrekken om zo via de tool Google Formulieren een inzichtelijke analyse te maken van de verkregen gegevens. (de enquête incl. vragen en analyse is te vinden in de bijlage-map of via deze link: https://bit.ly/2u4uGZE).

De enquête genaamd: “Survey Demo Drop form Dj Don Diablo”, is gedeeld op de sociale nieuwswebsite Reddit in een besloten groep genaamd: *r/hexagonians* (afbeelding 7).



afb. 7

De enquête bevat 4 demografische vragen, 7 meerkeuzenvragen en 1 openvraag. De laatste 8 vragen hebben betrekking tot de muzikale affiniteiten van de respondenten en in hoeverre deze corresponderen met die van Don Diablo (en zijn management), op welke social media platformen zij actief zijn en in hoeverre zij bekend zijn met de huidige demo drop applicatie.

4. Deelnemers: wat is het gebruikersprofiel waarnaar we op zoek zijn?
Met behulp van de resultaten uit de enquête is deze persona opgesteld en dit voldoet daarmee aan ons gebruikersprofiel.

JORDY

DETAILS

Jordy is 20 jaar, woont in Rotterdam en volgt een HBO IT-opleiding.

Hij is gek op Future House, EDM en Dutch House, luistert dagelijks naar Don Diablo, heeft hem paar keer zien optreden en beschouwt zichzelf onderdeel van een nieuwe generatie 'Hexagonians'.

Jordy ambieert een carrière als dj en producer. In zijn vrije tijd besteedt hij zoveel mogelijk tijd aan het produceren en het zoeken naar geschikte tracks voor zijn dj sets. Jody is een beginner met talent.

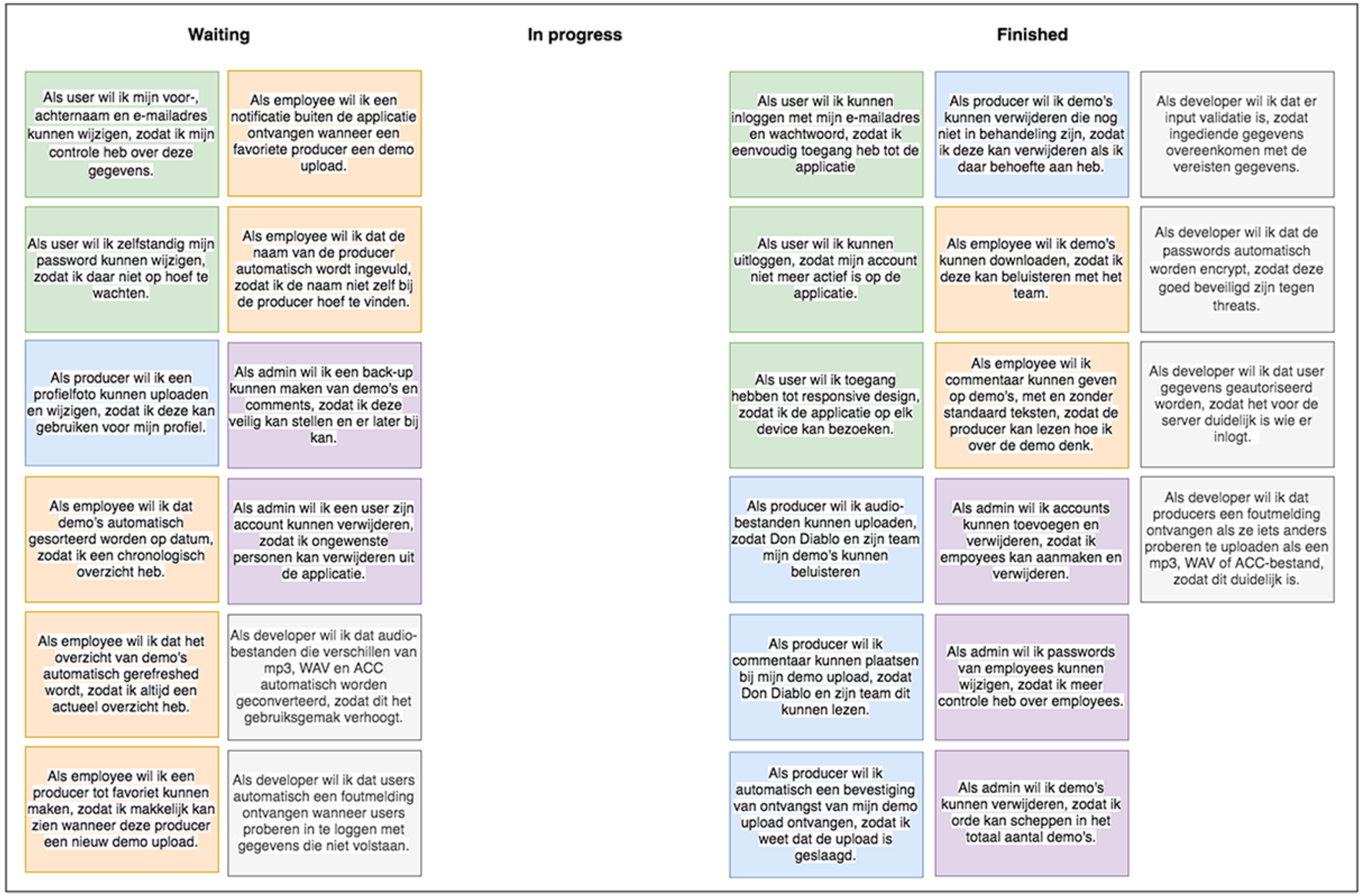
Online is hij voornamelijk te vinden op Instagram, Facebook en specifiek voor muziek vooral op Spotify, Youtube en Soundcloud.

Hij is bekend met de huidige demo drop applicatie en wil deze in de toekomst gaan gebruiken.

DOEL

"Mijn droom is om dance music te produceren dat gedraaid wordt door Don Diablo en uitgebracht wordt op zijn label Hexagon."

> User stories



afb. 8

User Stories
Omdat we een persona hebben gecreëerd en de actoren/eindgebruikers bij ons bekend zijn, kunnen we User Stories schrijven. Dit zijn eenvoudig beschreven behoeftes vanuit het oogpunt van de eindgebruikers. Een User Story is geen functionele beschrijving, maar maakt duidelijk wat een eindgebruiker wil, of nodig heeft en ook waarom dat nodig is. Het verschil met de eerder opgestelde requirements is dat User Stories meer context geven.

Product Backlog
Op deze Product Backlog (een gepriorteerde werkvoorraad) kunt u de User Stories vinden. Deze backlog vertegenwoordigt de totale waarde voor de eindgebruikers. De User Stories zijn onderverdeeld in kleuren en eindgebruikers: groen (users), blauw (producer), oranje (employee) en grijs (developer). Wegens tijdgebrek heb ik niet alles in werking kunnen krijgen voor de deadline, maar wel bijna alle High Priority requirements.

> Prototype

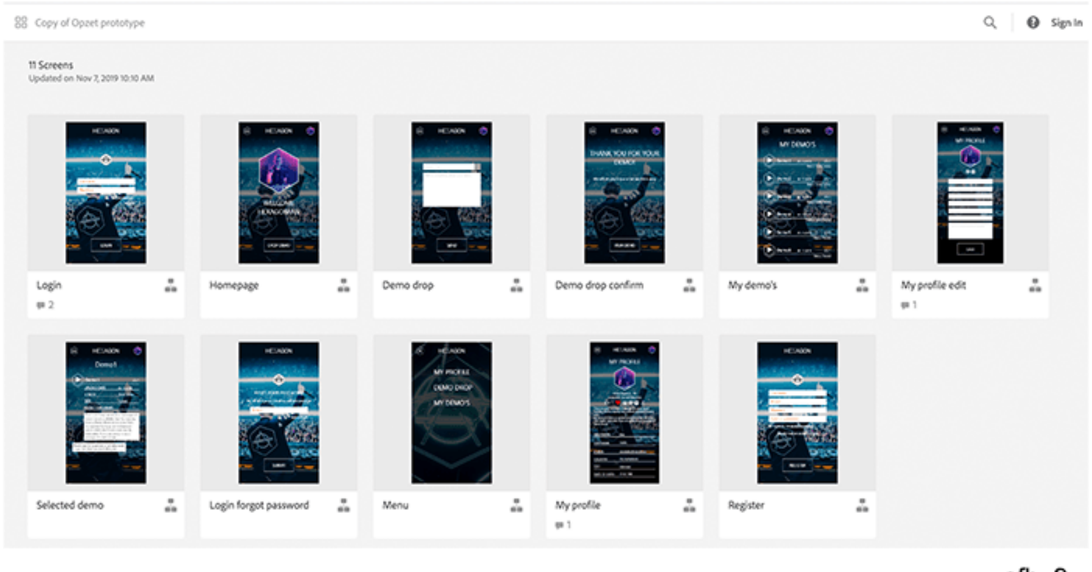
Nadat we het onderzoek naar het gebruiksgemak van de te ontwikkelen webapplicatie hadden afgerond, zijn we aan de slag gegaan met het ontwerpen van het prototype. Alle opgedane kennis en ervaringen uit het UX onderzoek hebben we hierin verwerkt.

> Het nut van prototyping
Het prototype draagt bij om te controleren of we op de juiste weg zijn voordat de webapplicatie daadwerkelijk ontwikkeld wordt. Via simulaties testen we een aantal aspecten van de webapplicatie, zodat getest kan worden hoe de applicatie werkt, wat het kan en hoe het interacteert.

Ook hebben we tussentijds de look en feel van het prototype met de woordvoerder kunnen afstemmen op de wensen van Don Diablo. Design keuzes zijn veelal gebaseerd op de huidige huisstijl van Don Diablo; afbeeldingen, lettertypes, layout, kleurgebruik etc. Ook de wens voor responsive design hebben we hierin meegenomen. Alles is ontworpen in AdobeXD.

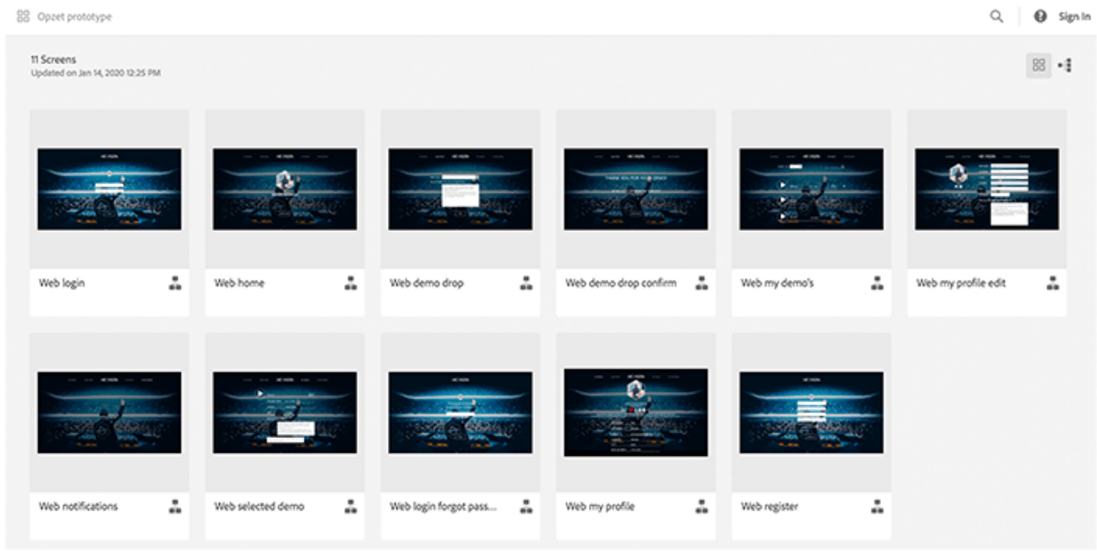
Uiteindelijk hebben we ons prototype laten testen, opnieuw door leden van de *r/hexagonians* groep op Reddit. De vergaarde feedback hebben we opnieuw afgewogen en wat nuttig is, is verwerkt in het eindresultaat (zie afbeelding 9 en 10). Via deze link vind u het klikbare prototype: https://adobe.ly/2uUxJUN

Mobile design



afb. 9

Desktop design



afb. 10

4. FRONT-END & BACK-END DEVELOPMENT () {

> Webapplicatie

Al het voorafgaande werk is nodig om een eindproduct neer te zetten dat zo goed mogelijk voldoet aan de eisen en verwachtingen van de opdrachtgever en zo goed mogelijk aansluit bij de best mogelijke gebruikservaring. Wel nu presenteer ik u het eindresultaat.

> Broncode

De gehele webapplicatie is te vinden in de map “hexagon” De broncodes kunt u vinden via de volgende file-paths:
Java-code en back-end: *hexagon/src/main/java/com/unizef/login/...*
Front-end: *hexagon/src/main/recources/...*

Ook kunt u de webapplicatie inzien en downloaden via mijn repository op **Github:** *unizef/novi-fullstack-bootcamp*

De gebruikte versies, handleiding en het instructiefilmpje zijn te vinden in de map “unizef”.

> Gebruikte technieken

In deze paragraaf leg ik uit met welke technieken, tools en security toepassingen de webapplicatie tot stand is gekomen. Ook vindt u hier het *datamodel*. In de map “unizef” vindt u het verantwoordingsdocument; een meer beknopte en gedetailleerde lijst van de belangrijkste toegepaste technieken voor de onderdelen front-end en back-end. Hierin vindt u tevens ook de verantwoording voor de onderdelen secure software en ontwerp.

> Front-end

De front-end is betreft design gebasseerd op de prototypes in de ontwerpfase. De front-end is gebouwd in de broncode-editor *Visual Studio Code*.

Bootstrap
Om de layout en interfaceonderdelen van de applicatie te bouwen, heb ik gekozen voor Bootstrap als framework. Hiermee werd het mogelijk om efficiënt en betrouwbaar de nodige webtoepassingen vorm te geven.

Thymeleaf
Deze *template engine* heb ik gebruikt met HTML5. Doordat hiermee logica ook in de template gebruikt kan worden, hoeft niet alles alleen in de back-end te gebeuren. Dit werkt voor mij prettig omdat ik dankzij de duidelijke syntax van Thymeleaf in de front-end (view) kan zien hoe dit zich logisch verhoudt tot de Java- en Spring Boot code in de back-end.

> Back-end

De back-end is betreft technisch ontwerp gebasseerd op het klassendiagram. De back-end is deels ontwikkeld in Eclipse Spring Tool Suite, uiteindelijk was het voor mij praktischer om dit net als de front-end in Visual Studio code voort te zetten.

Java
De webapplicatie is geprogrammeerd in de krachtige objectgeoriënteerde programmeertaal Java. Dankzij Java en samenhangde technieken was het mogelijk om een stabiele en veilige *MVC-applicatie* te ontwikkelen. **Maven** is gekozen als Java *build systeem* om externe modules, componenten en plug-ins te gebruiken.

Spring Boot
Dit applicatie framework werkt naaldloos samen met Java, Maven, Thymeleaf en MySQL. Spring Boot beschikt over een groot aantal *dependencies* verpakt in *Starters.* Deze Starters bieden oplossingen voor veel voorkomende toepassingen en zijn eenvoudig te implementeren in het project. Hierdoor was ik in staat om in afzienbare tijd mijn eerste multi-tier applicatie op te leveren.

Spring Boot beschikt ook over een eigen model-view-controller (MVC). Deze software design pattern verdeelt de programma logica in de drie elementen van het MVC-patroon. Dit hielp om het overzicht te houden, belangrijke componenten te ontkoppelen en om code hergebruik en parallelle ontwikkeling mogelijk te maken.

MySQL
Met dit managementsysteem en *SQL* als standaardtaal is de *relationele database* ontwikkeld, bestaande uit 5 tabellen. Spring Boot nam ook hier een grote rol in door te kiezen voor de Starters: *Data JPA* en *MySQL-connector-java*.

> Security

Met behulp van de risico- en beveiligingsanalyse heb ik de nodige mitigerende maatregelingen kunnen treffen om een goed beveiligde webapplicatie op te leveren. Wederom heeft Spring Boot hier geholpen, dit keer met de *Security Starter*. Hierdoor zijn alle passwords *encrypt* en zijn alle eindgebruikers via *JDBC-authenticatie* geauthenticeerd en via *HTTPS* geauthoriseerd. De *inputvalidatie* is geprogrammeerd aan de server-side en *SQL-injection* wordt voorkomen door het ORM-framework Hibernate.

> Unit testing

Unit testing is een methode om softwaremodulen of stukjes broncode (units) afzonderlijk te testen. Het doel is om te controleren of delen van de software goed werken of blijven werken gegeven bepaalde invoer (correct of foutief). Bij unit tests worden voor iedere unit een of meerdere tests ontwikkeld. Hierbij worden dan verschillende *testcases* doorlopen. In het ideale geval zijn alle testcases onafhankelijk van andere tests. Eventueel worden hiertoe *mockobjecten* gemaakt om de unit tests gescheiden uit te voeren.

Om tot een werkende applicatie te komen, heb ik vanzelfsprekend delen van de broncode moeten testen om uiteindelijk tot coherent geheel van code te komen om zo een werkzame applicatie te kunnen opleveren. Dit heb ik gedaan door stukjes code, waar nodig, te isoleren en te testen. Door gebrek aan tijd en kennis heb ik hier geen genormaliseerde strategie voor gebruikt en heb ik geen testrapport kunnen opleveren.

> Gebruikerstest

Met gebruikerstesten wordt gecontroleerd, in samenwerking met of de gebruiker alleen, of de software reageert op de manier zoals de gebruiker verwacht dat hij zou reageren.

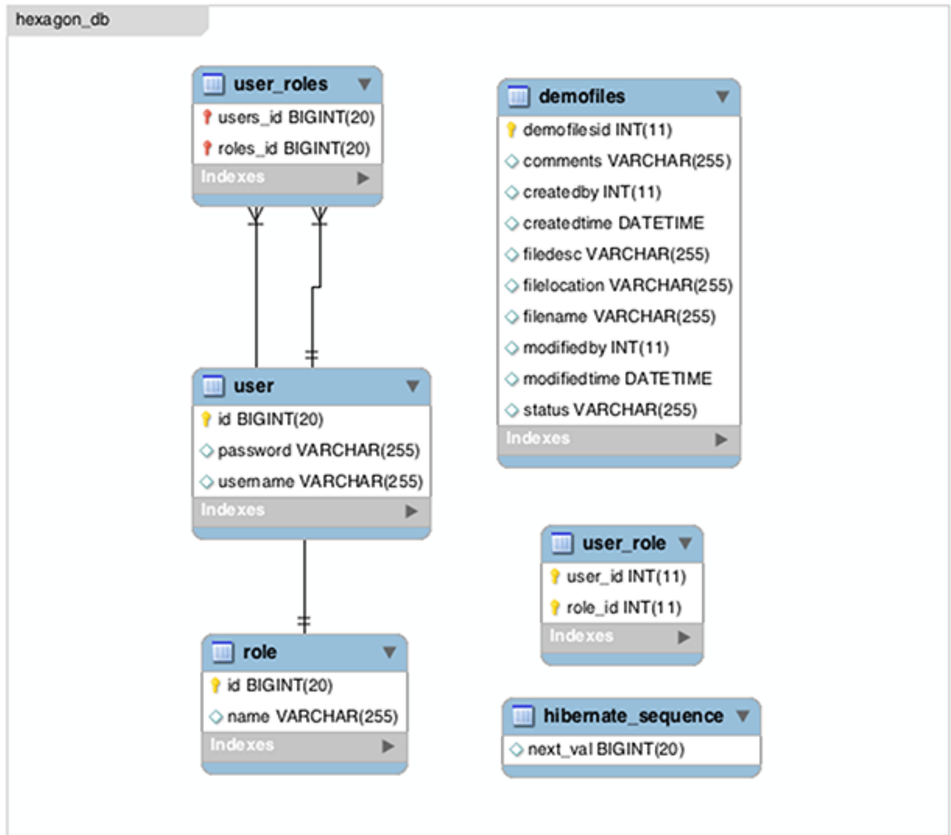
Hiervoor heb ik een aantal gebruikerstesten laten uitvoeren door vrienden en familie. De resultaten zijn positief te noemen. De webapplicatie is stabiel, betrouwbaar en de belangrijkste functies werken.

> Datamodel

Het datamodel is een soort stroomdiagram dat illustreert hoe *entiteiten* (personen, voorwerpen of concepten) met elkaar verbonden zijn. Datamodellen worden vaak gebruikt om (rationele) databases te ontwerpen en beschrijven.

Het datamodel, genaamd: hexagon_db, is feitelijk een weergave van hoe de gegevens in de database van de Demo Drop webapplicatie gestructureerd zijn. Het betreft een persistente *relationele database* bestaande uit de volgende 5 entiteiten/tabellen:
> user_roles
> user_role
> user
> role
> demofiles
> hibernate_sequence is puur ondersteunend.

De *relaties* en *kardinaliteiten* zijn beschreven d.m.v. Hanepoot- of Martin-stijl. De gele sleutels vormen de *primary keys*, de rode sleutels de *foreign keys* en deze vormen, evenals de overige attributen, kolommen in de database.

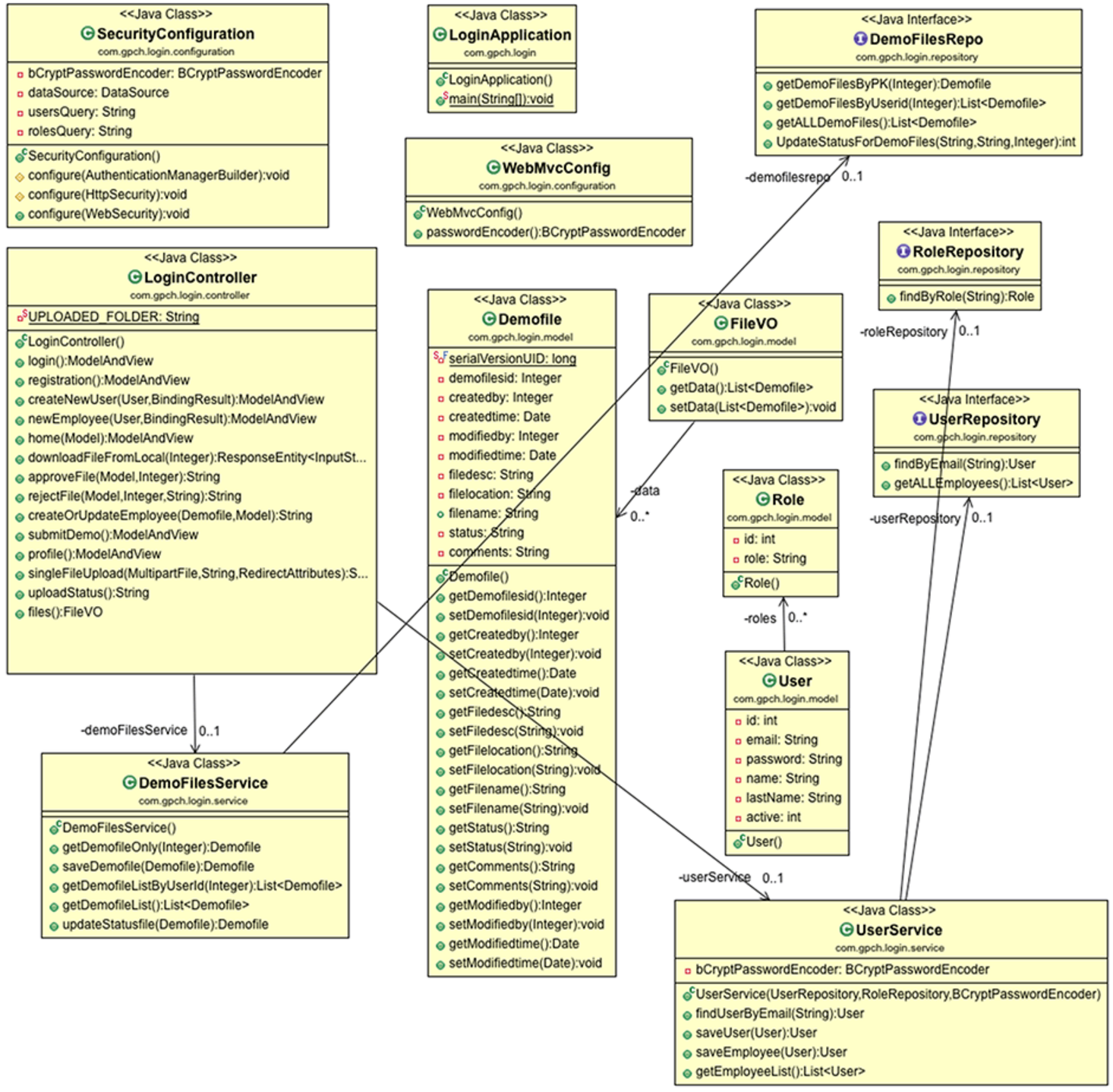


> Omgekeerd gegenereerde klassendiagram

Het klassendiagram uit de ontwerp fase (pagina 8) wijkt af van het klassendiagram dat u op deze pagina ziet.

Omdat de webapplicatie het MVC-patroon volgt en is geprogrammeerd met Spring Boot, is de uitkomst per definitie verschillend.

Het eerdere technisch ontwerp dient als uitgangspunt en dit is daar het eindresultaat van, met de belangrijkste requirements in acht genomen.



> Slotwoord

De afgelopen 5 maanden waren voor mij leerzaam en intens. Als ik terugkijk naar de analyes, het vooronderzoek en de ontwerpfase, moet ik concluderen dat niet alles is opgeleverd. In de product backlog zijn dan ook de user stories te vinden die het niet gehaald hebben voor de deadline (veel daarvan zijn ook zelf opgesteld met projectgroep SuperClass). Toch zijn de belangrijkste requirements werkzaam gemaakt. De opdrachtgever heeft in ieder geval een product in handen dat werkt, veilig is en dat gebruikt kan worden door alle gewenste eindgebruikers.

Voor een eeste MVC-webapplicatie ben ik zelf in ieder geval trots.

Zef Stribos

}