



SOFTWARE ONTWIKKELING
EN PROFESSIONELE VAARDIGHEDEN

2DE BACHELORJAAR INFORMATICA

Pukkelpop: Analyseverslag

Groep 6:

Maria Hendrikx (1849886),
Wouter Grootjans (1848521),
Michiel Swaanen (1848709)

Begeleider
Balazs Nemeth

Coördinerend verantwoordelijke
prof. dr. Wim Lamotte

Jaar 2019-2020

Inhoudsopgave

1 Diepgaande beschrijving van het onderwerp	2
1.1 Algemene beschrijving	2
1.2 Een overzicht van de features	2
1.3 Algemene features	3
2 Gerelateerd werk: software, libraries en literatuur	4
2.1 Software	4
2.1.1 Locatie	4
2.1.2 Overzicht	5
2.2 Libraries	7
2.2.1 Visuele voorstelling van de architectuur	7
2.2.2 Libraries voor de webapplicatie en de smartphone applicatie	7
2.2.3 Libraries enkel voor de smartphone applicatie	8
2.2.4 Libraries alleen voor de webapplicatie	9
2.2.5 Libraries enkel voor de server	9
2.2.6 Externe services	10
3 Evaluatiecriteria	10
3.1 Features List	11
3.2 Prioriteitenlijst	13
3.3 Functionele vereisten	15
3.4 Niet-functionele vereisten	15
4 Analyse	16
4.1 Algoritmen en Datastructuren	16
4.1.1 Kortste pad	16
4.2 Klassendiagram en ontwerp	17
4.3 Data en databaseschema	19
4.3.1 Database	19
4.3.2 Data	19
4.3.3 Databaseschema	20
4.4 Bestandsformaat	23
5 Mockups	24
5.1 Algemene toelichting	24
5.2 Sector-verantwoordelijke	25
5.2.1 Toelichting	25
5.2.2 Afbeeldingen	26
5.3 Vrijwilliger	28
5.3.1 Toelichting	28
5.3.2 Afbeeldingen	28
5.4 Admin	29
5.4.1 Toelichting	29
5.4.2 Afbeeldingen	29
6 Taakverdeling en planning	35
6.1 De planning	35
6.2 De planning als tijdlijn	37
6.3 Mijlpalen	38
6.3.1 Toelichting	38
6.3.2 Mijlpalen	38
7 Bibliografie	39

1 Diepgaande beschrijving van het onderwerp

1.1 Algemene beschrijving

Pukkelpop stelt een groot aantal vrijwilligers te werk. Het controleren of deze vrijwilligers al dan niet op hun post zijn is een complexe taak. Pukkelpop wil checken of alle vrijwilligers wel degelijk aan het werk zijn, daarom stelt Pukkelpop een aantal sector-verantwoordelijke aan. Dit zijn mensen die moeten rondgaan en checken of de vrijwilligers zich nog op hun post bevinden.

Pukkelpop is daarom op zoek naar een applicatie om hun vrijwilligers buiten de festivalsite te lokaliseren en eventueel bij te sturen. Het primaire doel is om de rapportering digitaal te maken, waarbij er een duidelijk overzicht is van de invulling van de posten en de aan- en afwezigheid van de vrijwilligers op elk moment.

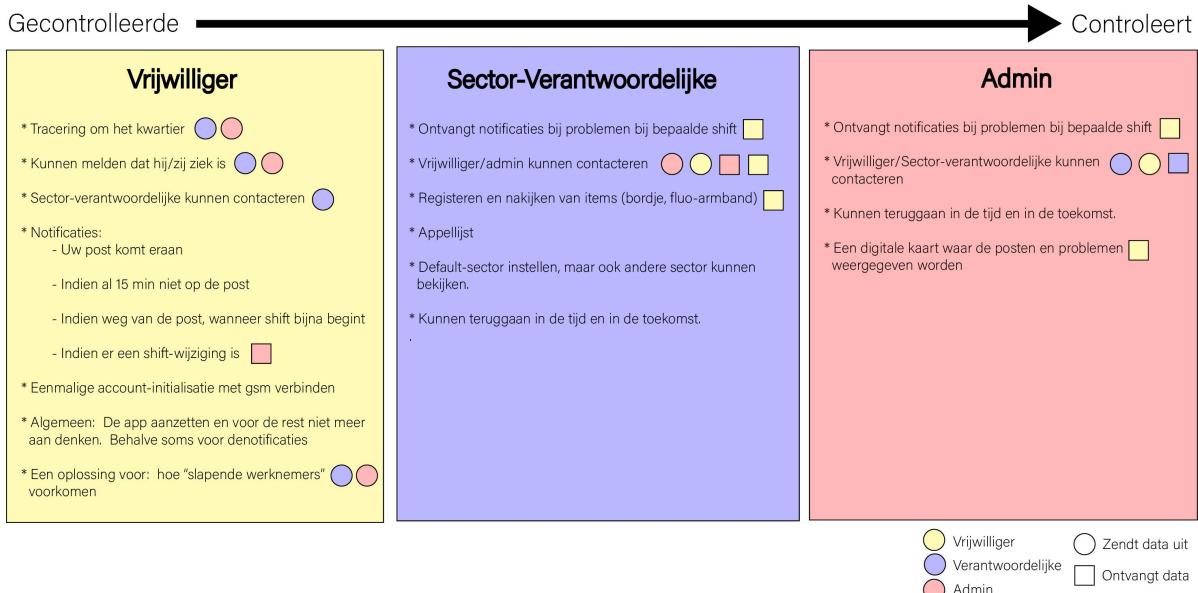
Dit wil Pukkelpop bereiken door een mobiele applicatie te laten maken, die de locatie van de vrijwilligers vergelijkt met de posten en zo nakijkt of hij/zij aanwezig is op zijn/haar post. De tracing van de vrijwilligers zou dan naar de controlerende bevoegdheden doorgestuurd worden, zodat zij ook meer gericht kunnen handelen wanneer er zich problemen voordoen. Dit kan op verschillende manieren gedaan worden. Pukkelpop zelf stelt voor om een digitale kaart te gebruiken, waarbij de probleem-gevende posten dan oplichten.

De applicatie bestaat uit 3 afzonderlijke modules, waarbij elke module een andere hoofd-functionaliteit heeft. Zo is de vrijwilliger-module eerder een passieve applicatie, waarbij de applicatie vooral bepaalde gegevens registreert, zoals bijvoorbeeld de locatie van de gebruiker. De sector-verantwoordelijke zal de applicatie eerder actief gebruiken. Zij zullen de mobiele applicatie vooral gebruiken om hun sector te monitoren. De administrator heeft een web applicatie die gebruikt kan worden voor een algemeen overzicht alsook het genereren van de rapportering.

Het doel van de applicatie is dat Pukkelpop op het einde van het festival de rapportering kan genereren van de al dan niet gepresteerde werk uren. Ook moet de applicatie het werk van de sector-verantwoordelijke digitaliseren en vereenvoudigen zodat zij gerichter kunnen werken.

1.2 Een overzicht van de features

De onderstaande features zijn de features vermeld door de opdrachtgever. Bij sectie "3.1 Feature List" zijn de features meer gedetailleerd uitgewerkt.



1.3 Algemene features

Omdat er gewerkt wordt met lijsten en mensen, zal er gebruik moeten gemaakt worden van een database. In totaal zullen er circa. 2500 mensen de applicatie gebruiken, waardoor de database hierop afgestemd kan worden. De data die zal worden verzameld zijn de locaties van de vrijwilligers, waardoor het programmeer-platform zeker locatie-features zal moeten ondersteunen, die daarbij in de achtergrond opgehaald kunnen worden.

Daarnaast moeten de gegevens onderling uitgewisseld kunnen worden, waardoor er ook beroep zal moeten gedaan worden op internet. Aangezien niet iedereen 4G heeft, moet er ook een oplossing gevonden worden voor het offline bewaren van de data, om die achteraf, eens er internet-connectie is, door te geven naar de algemene database. Aangezien de applicatie door ongeveer 250 mensen tegelijkertijd gebruikt zal worden, moeten de services hier ook op aangepast worden.

Voor de lay-out zijn er geen algemene vereisten vanuit Pukkelpop zelf, behalve dan dat de applicatie vooral functioneel moet zijn. Er moet rekening gehouden worden met het feit dat Pukkelpop zich in de zomer plaatsvindt, en het publiek vrij breed is (de leeftijd varieert van 18 tot 70+). Om te verifiëren dat zowel jong als oud met de applicatie overweg kan, zouden wij tijdens de ontwikkelingsfase de applicatie ook eens aan laten testen door een aantal mensen met een verschillende leeftijds categorie.

2 Gerelateerd werk: software, libraries en literatuur

2.1 Software

De applicatie dat Pukkelpop wil, is nog nergens verkrijgbaar. Dit komt omdat pukkelpop specifieke vereisten heeft. Dit heeft als gevolg dat de Pukkelpop applicatie niet vergeleken kan worden met andere applicaties. Wel kan er inspiratie gehaald worden uit afzonderlijke delen. Zo wil Pukkelpop bijvoorbeeld een digitale kaart weergegeven hebben. In de Play store zijn veel applicaties te vinden die een digitale kaart gebruiken, waarbij afhankelijk van de applicatie, verschillende features erbij aangeboden worden. Wij hebben even een paar applicaties onder de loep genomen.

2.1.1 Locatie

Google Maps

- Markers toevoegen
- Locatie in de achtergrond opvragen
- Een route instellen
- Zoeken naar een locatie
- ...

Aangezien we de Google Maps API gaan gebruiken, zal onze applicatie ook bepaalde designs en features van Google Maps overnemen. Zo zal de Pukkelpop applicatie gebruik maken van de markers, de map zelf, functionaliteiten zoals in- en uitzoomen, enzovoort.



2.1.2 Overzicht

InitLive [15]

InitLive is software bedoeld voor management. De applicatie heeft zowel applicaties voor de werknemers, alsook voor de werkgevers. Aangezien de applicatie van Pukkelpop ook onder categorie 'management' valt, kan InitLive als gerelateerde applicatie worden beschouwd.

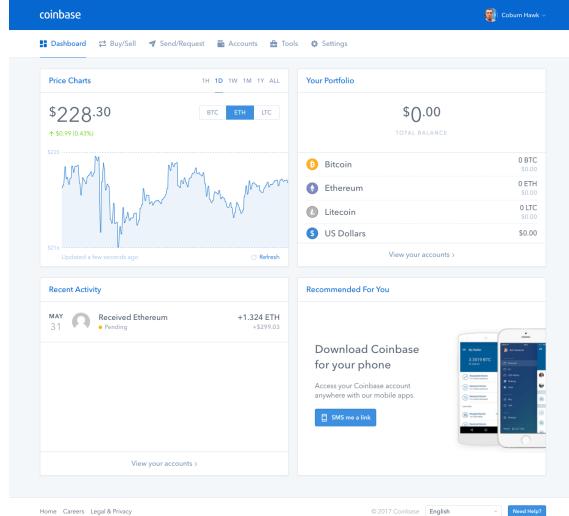
Een gedeelte dat InitLive heeft ontwikkeld, wat ook in onze software voor Pukkelpop voorkomt, is de data van het personeel weergeven. InitLive heeft dit onder andere toegepast door de standaard structuur van een database te behouden, waarbij er bepaalde functionele knoppen extra zijn toegevoegd, zoals bijvoorbeeld "see more"-knop. Daarnaast zijn er ook nog algemene functionaliteiten zoals filteren, zoeken, ... toegevoegd boven de weergave van de database.

De admin maakt bij de Pukkelpop-applicatie gebruik van hetzelfde: een database. Aangezien wij het weergeven van de data op deze manier een zeer overzichtelijke manier vinden, hebben wij ons hierop gebaseerd. Bij InitLive is deze sectie echter vrij geavanceerd. Aangezien Pukkelpop niet op zoek is naar een applicatie om de data te maken vanaf nul, moet het systeem voor de Pukkelpop-applicatie, niet zo geavanceerd zijn.

The screenshot shows the InitLive Staff interface. At the top, there's a navigation bar with icons for Home, My Organizations, Brendan Test Org, Events, Test Event, and Staff. Below the navigation is a search bar labeled 'Search Staff'. The main area is titled 'Staff' and contains a table with the following columns: Profile, Roles, Staff Email, Event Check-in, Role Status, Qualifications, Total Hours Scheduled, Total Number of Shifts, and M/N. There are 26 staff members listed. Each row shows a profile picture, the staff's name, their roles (e.g., First Aid, XYZ, Leader), their email, their event check-in status, their role status (In a Role), their qualifications (e.g., 50lbs), and their total hours scheduled and number of shifts. At the bottom right of the table, there are buttons for '26 Staff' and 'Roster Mode™'.

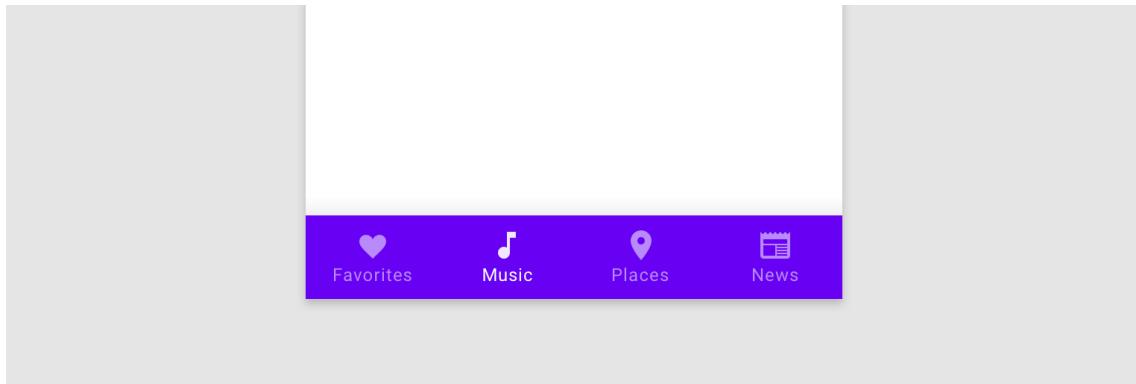
CoinBase Dashboard [1]

Qua overzicht hebben we inspiratie gehaald van “CoinBase Dashboard”. Waarbij we voornamelijk de layout-indeling van het dashboard van Coinbase overnomen. Qua functionaliteiten hebben we niet echt gekozen bij CoinBase.



Bottom Navigation [3]

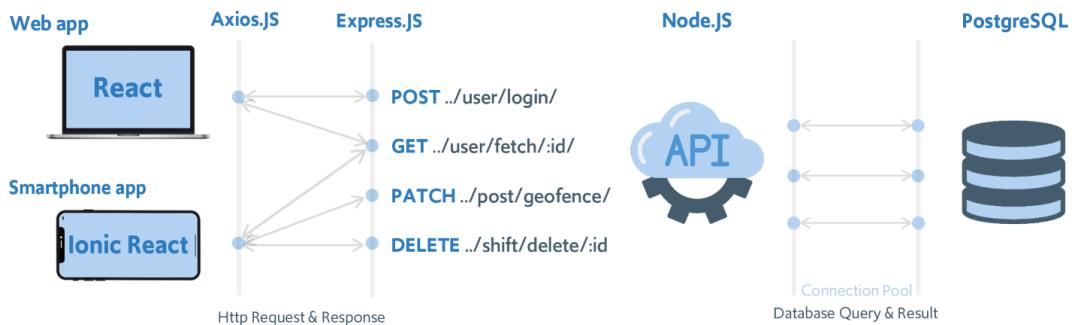
Voor de algemene lay-out van de mobile applicaties, hebben we gekozen voor een 'bottom navigation'. Deze layout komt vaker voor en wordt bij Android studio bijvoorbeeld ook voorgesteld als een manier van navigeren. Daarnaast zijn er verschillende applicaties die hier ook gebruik van maken, zoals bijvoorbeeld Instagram, Snapchat, enzovoort.



2.2 Libraries

Er worden verschillende libraries gebruikt voor het uitbuiten van bepaalde functionaliteit. Libraries voor het web platform alsook de smartphone applicatie worden beheerd door Node Package Manager (NPM). In de volgende sectie worden alle libraries en externe services beschreven die gebruikt worden binnen de smartphone applicatie, het web platform en onze server.

2.2.1 Visuele voorstelling van de architectuur



De web applicatie wordt gebruikt door de administrator van Pukkelpop. De administrator moet een goed overzicht hebben over alle meldingen en problemen. Daarom maakt de administrator gebruik van een web applicatie. Dit geeft de persoon meer schermruimte waardoor die meer informatie kan zien op het scherm.

De smartphone applicatie wordt gebruikt door de vrijwilligers en sector-verantwoordelijken. Deze gebruikers delen hun problemen en meldingen via hun smartphone, omdat zij werken op het terrein. Het zou dus niet handig zijn voor deze groep van gebruikers om een laptop mee te nemen tijdens hun werk.

2.2.2 Libraries voor de webapplicatie en de smartphone applicatie

- ReactJS: <https://reactjs.org/>

Om onze front-end op te bouwen hebben wij React gekozen. De voornaamste reden is dat er dan maar 1 programmeertaal gebruikt moet worden om zowel de smartphone applicatie alsook het web platform te ontwikkelen. Wij prefereren 1 programmeertaal voor het hele systeem, niet omdat wij lui of ongeïnteresseerd zijn, maar omdat dit handiger is voor de klant. Indien Pukkelpop ervoor kiest om de software uit te breiden of te onderhouden, moeten zij enkel een ontwikkelaar met kennis van TypeScript zoeken. Hierdoor kunnen zij de onderhoudskosten drukken en moeten zij niet op zoek gaan naar meerdere ontwikkelaars om hun systeem uit te breiden of te onderhouden.

De community van React is veel actiever dan de community van VueJS, hierdoor was de keuze voor ons snel gemaakt. AngularJS wordt meer binnen bedrijfs context gebruikt, hierdoor is de online support minder dan die van React.

- Redux: <https://www.npmjs.com/package/redux>

Er is gekozen om Redux te gebruiken omdat de logica en data flow binnen onze applicatie redelijk complex is. Redux versimpelt de 'data flow' van de applicatie. Redux is veruit de meest stabiele en best onderhouden library voor state management binnen React.

- Axios: <https://www.npmjs.com/package/axios>

Axios biedt een eenvoudig en gebruiksvriendelijk (developer experience) interface om te communiceren (via HHTP requests) met onze server (REST API). Axios heeft veel 'contributors' op GitHub, hierdoor kan er geconcludeerd worden dat de ondersteuning voor deze third-party library niet snel zal afzwakken.

- Redux Thunk: <https://www.npmjs.com/package/redux-thunk>

Redux Thunk word gebruikt omdat Redux out of the box geen asynchronous functionaliteit ondersteunt. Met de redux-thunk library kan er asynchronous functionaliteit toegevoegd worden aan het systeem. Een alternatief is redux-promise, maar uit onderzoek blijkt dat redux-thunk veel beter en vaker onderhouden wordt.

2.2.3 Libraries enkel voor de smartphone applicatie

- Ionic React <https://ionicframework.com/docs/react>

Ionic React word gebruikt om onze smartphone applicatie werkend te krijgen op iOS en Android. Hierdoor kan de smartphone applicatie in 1 programmeertaal geschreven worden. Dit bespaard ons veel tijd en is voor de klant makkelijk in onderhoud. De code wordt in TypeScript geschreven, deze wordt gecompileerd naar JavaScript die op zijn beurt gecompileerd wordt naar Java (Android) en Swift (iOS) door Ionic.

Gedurende de onderzoeksfase werden er verschillende concurrenten van Ionic React getest. In eerste instantie was Flutter onze favoriete keuze, echter waren we enorm veel problemen met het opvragen van de locatie. Een week en 5 libraries later was de achtergrond locatie voor onze smartphone applicatie nog steeds niet werkend. Wij zijn toen onderzoek beginnen te doen naar Ionic Cordova, React Native en Ionic React. Na 2 dagen werkte de achtergrond locatie met Ionic React en het was veel gebruiksvriendelijker dan Flutter of een andere concurrent. Ook biedt de Ionic background location library ons de beste precisie en opties om te configureren.

Ionic biedt ook out of the box native componenten aan, dit wilt zeggen dat het visuele aspect van de applicatie grotendeels overgelaten kan worden aan Ionic. Het is van groot belang dat de applicatie gebruiksvriendelijke en intuïtief te gebruiken valt, aangezien wij werken met een echte opdrachtgever.

Elke library die er gebruikt word binnen het Ionic React Framework wordt ondersteund, onderhouden en gedocumenteerd door Ionic. De reden dat er libraries gebruikt worden voor bepaalde functionaliteit is omdat wij niet beschikken over een Apple computer om native functionaliteit gaan te ontwikkelen.

- Background Geolocation: <https://ionicframework.com/docs/native/background-geolocation>

Een van de belangrijkste toepassingen binnen de smartphone applicatie is het traceren van de vrijwilliger op het terrein. Om dit zo gebruiksvriendelijk mogelijk te maken voor de gebruiker moet de locatie op de achtergrond opgevraagd worden. Het is redelijk tijdrovend om een eigen tracing systeem te schrijven die perfect de achtergrond werkt voor zowel Android als iOS.

- Sim Reader: <https://ionicframework.com/docs/native/sim>

Elke gebruiker van de applicatie krijgt een login toegewezen gebaseerd op zijn of haar gsm nummer. Elk gsm-nummer is uniek, zo kan elke persoon eenvoudig gekoppeld worden aan digitaal profiel. Deze library word gebruikt om het gsm nummer, en eventueel andere data, uit te lezen van de gsm van de gebruiker.

- Call number: <https://ionicframework.com/docs/native/call-number>

Wanneer een sector verantwoordelijke wil communiceren met een vrijwilliger kan hij of zij deze persoon bereiken door hem of haar te bellen. Er is gekozen voor deze aanpak omdat de vrijwilliger tijdens zijn/haar shift waarschijnlijk geen notificaties, e-mails of sms'en gaat lezen. Ook wanneer de gebruiker geen internet heeft, zal hij of zij niet bereikbaar zijn via notificaties en e-mail.

- Local notifications: <https://ionicframework.com/docs/native/local-notifications>
- Push notifications: <https://ionicframework.com/docs/native/push>

Er worden notificaties doorheen onze applicatie gebruikt om sector-verantwoordelijke en vrijwilligers te alarmeren voor bepaalde situaties en evenementen. Er word afwisselend gebruik gemaakt van push en lokale notificaties, omdat de applicatie vaak op de achtergrond runt.

- Toast messages; <https://ionicframework.com/docs/native/toast>

Het gebruik van toast messages vergemakkelijkt het bevestigen van een bepaalde actie. Wanneer een persoon een bepaalde actie onderneemt via de applicatie word er feedback met Toast messages.

2.2.4 Libraries alleen voor de webapplicatie

- Material-UI: <https://material-ui.com/>

Er word gebruik gemaakt van een third-party library voor de opmaak van het web platform. Omdat wij een echte opdrachtgever hebben willen wij ook dat onze webapplicatie er visueel sterk uit ziet. Concurrerende libraries zoals Bootstrap, Blueprint en Elemental zijn ook een goede keuze, echter werd er voor Material-UI gekozen, omdat de library speciaal voor React is gemaakt. Ook biedt het de meeste support online.

- Excel-to-JSON: <https://www.npmjs.com/package/convert-excel-to-json>

Voor het toevoegen van gebruikers word er gebruik gemaakt van excel files. De opdrachtgever kan deze uploaden via het administrator paneel om zo nieuwe gebruikers eenvoudig toe te voegen aan de applicatie. Er word gebruik gemaakt van de convert-excel-to-json library om deze functionaliteit mogelijk te maken.

2.2.5 Libraries enkel voor de server

- NodeJS: <https://nodejs.org/en/>

Voor de front-end wordt er gebruik van TypeScript (JavaScript). Het is van groot belang dat de applicatie zo makkelijk mogelijk te onderhouden is of uit te breiden valt. Daarom is er als back-end programmeertaal gekozen voor NodeJS. Dit is een eenvoudige programmeertaal die vaak gebruikt wordt om REST API's te ontwikkelen. Zowel de front-end: ReactJS en de back-end: NodeJS maken gebruik van de NGinx, hierdoor kan alles op 1 server instance runnen.

NodeJS maakt gebruik van NPM (Node Package Manager) om de dependencies binnen zijn ecosysteem te beheren. Hierdoor kan er gebruik gemaakt worden van al bestaande code die ons helpt om een veilige en efficiënte back-end omgeving te creëren.

Andere programmeertalen zoals Python (Flask / Django), Java (Spring Boot) of Go (Gorilla) zouden ook een goede keuze zijn als back-end. Echter zoals eerder besproken geven deze alternatieven programmeertalen de opdrachtgever extra kosten wanneer zij de software willen uitbreiden of laten onderhouden door een externe ontwikkelaar. Ze moeten dan zowel een ontwikkelaar vinden die TypeScript (front-end) kan onderhouden alsook een ontwikkelaar die de back-end kan onderhouden in een van de alternatieven programmeertalen. Om dit te vermijden werd er gekozen voor NodeJS, zo houden wij de development omgeving binnen 1 ecosysteem.

- Body-parser: <https://www.npmjs.com/package/body-parser>

De body-parser library is nodig om inkomende en uitgaande JSON te formatteren, zodat het leesbaar is voor de server. Met 12 miljoen downloads per week is body-parser de meest ondersteunde library die vandaag de dag verkrijgbaar is.

- Nodemon: <https://www.npmjs.com/package/nodemon>

Met deze library kan de de server hot-reloaden. Dit betekent dat de server niet meer handmatig aan en uit gezet moet worden om de laatste veranderingen te verkrijgen.

- Sequelize: <https://www.npmjs.com/package/sequelize>

Er word gebruik gemaakt van Sequelize, omdat dit een eenvoudige manier is om de database te queryen en te verbinden met de database. Hierdoor worden de fouten binnen queries gereduceerd en kan er een sneller development proces gegarandeerd worden. Ook maakt Sequelize gebruik van promises, hierdoor kan er elegante en logische code geschreven worden.

- PG: <https://www.npmjs.com/package/pg>

Door de PG library te gebruiken, kan er eenvoudig connectie gemaakt worden met onze PostgreSQL database. De reden dat er gekozen is voor deze library is omdat het door NodeJS zelf aangeraden wordt en het de enige library is die gebruik maakt van pool connecties. Pool connecties zorgen ervoor dat er minder connecties worden aangemaakt gedurende de lifecycle van de applicatie, dit leidt tot een sneller werkende applicatie.

- Bcrypt: <https://www.npmjs.com/package/bcrypt>

Bcrypt heeft de meeste ondersteuning op GitHub voor het encrypteren van verschillende soorten data. Er word gebruikt gemaakt van deze library om data geëncrypteerd te versturen en ontvangen data te decrypteren.

- dotenv: <https://www.npmjs.com/package/dotenv>

Met de dotenv library kunnen er eenvoudig zogenaamde 'environment variables' gebruikt worden binnen onze code die compatibel zijn met elk OS. Met 'environment variables' kunnen er eenvoudig variabelen doorheen de code gespecificeerd worden, deze variabelen kunnen eenvoudig beheert worden in een .env file. Hierdoor kan er bijvoorbeeld gemakkelijk en snel gewisseld worden van debug modus naar productie modus.

- Express.JS: <https://www.npmjs.com/package/express>

De routing voor de server wordt afgehandeld door Express.JS. Deze library zal alle routing requests en middleware verwerken en zal alle requests doorverbinden met de juiste controller. Express.JS is makkelijk te leren en heeft een grote community, dit maakt het ideaal om te leren en online hulp te zoeken.

2.2.6 Externe services

- Google Maps API: <https://developers.google.com/maps/documentation>

Het gebruik van de Google Maps API is cruciaal voor onze applicatie. Deze betaalde service biedt ons een goede tracking en visualisatie van de vrijwilligers op het terrein. Google Maps is veruit de meest gebruiksvriendelijke en best ontwikkelde digitale kaart op de markt.

- Google Directions API: <https://developers.google.com/maps/documentation/directions/start>

Als extra functionaliteit willen wij voor de sector-verantwoordelijke de kortste route naar een specifieke vrijwilliger/post genereren. Dit doen wij aan de hand van het langste pad algoritme. Echter maken wij alreeds gebruik van de Google Maps API die dit proces drastisch kan vereenvoudigen. De Google Directions API helpt ons met de kortste route te vinden naar de vrijwilliger/post.

3 Evaluatiecriteria

Voor Pukkelpop zijn er drie verschillende gebruikers. Een prioriteitenlijst met bullet-points is niet echt overzichtelijk, daarom hebben wij ervoor gekozen om de features per gebruiker apart te noteren. Omdat de prioriteiten per gebruiker door elkaar lopen, hebben we daarnaast ook een prioriteitenlijst gedefinieerd waarbij de features in prioriteit gesorteerd zijn.

3.1 Features List

- Vrijwilliger
 - Aanwezigheid
 - * Opvragen locatie om de x minuten (achtergrond + voorgrond).
 - * Controleren of hij of zij op post is.
 - * Tracing locatie, een kwartier voor de shift begint t.e.m de shift eindigt.
 - * Notificatie sturen: "Bent u nog wakker", waar de vrijwilliger binnen het halfuur op moet reageren.
 - Meldingen
 - * Melding dat de shift binnen x minuten begint.
 - * Melding indien de gebruiker zich niet binnen de post-radius bevindt, tijdens zijn shift.
 - * Melding indien de shift al begonnen is en na x tijd nog niet aanwezig is.
 - * Melding indien de gebruiker na y tijd (y tijd is x + ...) nog steeds niet aanwezig is, en vragen naar de reden waarom ze er nog niet zijn.
 - * Melding dat shift verandert (via bericht, mail of notificatie).
 - * Melding van globale informatie.
 - Melden
 - * Melden van een probleem (technisch, materialistisch, ...).
 - * Melden van het verlaten van de post om x reden.
 - Authenticatie
 - * Registratie van vrijwilligers (check op gsm-nummer).
 - Info
 - * Informatie over de shift (adres, uur, met wie, wat, ...)
 - * Een map naar de locatie
- Sector-verantwoordelijke
 - Appel-lijst en Posten weergeven
 - * Per post de vorige en volgende shiften bekijken.
 - * Posten weergeven via een digitale kaart.
 - * Posten met problemen visueel zichtbaar maken (zowel op de digitale kaart als in de lijst).
 - * Filteren op sector bij de digitale kaart (per sector tonen, bepaalde sectoren, alle sectoren tonen)
 - * Posten weergeven via een lijst.
 - * Sorteren van de lijst met sectoren: alfabetisch, kortste afstand.
 - * Filteren van de lijst op sector (per sector tonen, bepaalde sectoren, alle sectoren tonen)
 - * Filteren van de lijst op naam
 - * Filteren van de lijst op probleem
 - Contact
 - * Zichtbare gegevens of een soort sneltoets om de vrijwilliger/administrator te contacteren (bellen / email).
 - Meldingen
 - * Vrijwilliger
 - . Melding bij afwezigheid.
 - . Melding bij probleem.

- Melding bij ongeval.
- * Administrator
 - Melding bij probleem.
 - Melding bij globale melding.
- Controle
 - * Aanwezigheid van 'items' checken.
 - * Mogelijkheid om bij post en/of medewerkers nota's te maken.
- Extra instellingen
 - * Mogelijkheid om default een sector in te stellen.
 - * Uit/aanzetten van notificaties/meldingen van de gsm
 - * Light/Dark modus
- Administrator
 - Data
 - * Inlezen / veranderen van planning (via Excel?)
 - * Inlezen / veranderen van posten en sectoren
 - * Inlezen / veranderen van vrijwilligers en verantwoordelijken
 - * Veranderen van doorgegeven activiteit van de vrijwilliger (check-in, problemen, ...)
 - * Instellingen
 - De centrale check-tijd manueel kunnen aanpassen (om de hoeveel tijd de vrijwilligers worden gecheckt).
 - Appel-lijst en Posten weergeven
 - * Per post de vorige en volgende shiften kunnen bekijken.
 - * Shiften weergeven via een digitale kaart.
 - * Shiften met problemen visueel zichtbaar maken (zowel op de digitale kaart als in de lijst).
 - * Filteren op sector bij de digitale kaart (per sector tonen, bepaalde sectoren, alle sectoren tonen)
 - * Shiften weergeven via een lijst.
 - * Filteren van de lijst op sector (per sector tonen, bepaalde sectoren, alle sectoren tonen)
 - * Filteren van de lijst op naam
 - * Filteren van de lijst op probleem
 - Overzicht
 - * Gemelde problemen
 - * Items
 - * Details over de vrijwilligers en verantwoordelijken
 - * Vrijwilligers kunnen deactiveren
 - * Pop-up wanneer werknemer begint en eindigt aan zijn/haar shift
 - Contact
 - * Planning-wijzigingen naar vrijwilligers
 - * Globale informatie naar alle / specifieke vrijwilligers (via notificatie, sms of e-mail)
 - * Zichtbare gegevens of een sneltoets om vrijwilliger/sector-verantwoordelijke te contacteren (bellen / email)
 - Rapportering
 - * Genereren van pdf
 - * Opdeling wie afwezig was tijdens zijn shift
 - * Wie, welke sector, welke post, wanneer

3.2 Prioriteitenlijst

1. Vrijwilliger → Aanwezigheid - Opvragen locatie om de x minuten (achtergrond + voorgrond)
2. Vrijwilliger → Aanwezigheid - Controleren of hij of zij op post is
3. Vrijwilliger → Aanwezigheid - Tracing locatie, een kwartier voor de shift begint t.e.m de shift eindigt
4. Administrator → Data - Inlezen / veranderen van planning (via Excel?)
5. Administrator → Data - Toevoegen / veranderen van posten en sectoren
6. Administrator → Data - Inlezen / veranderen van vrijwilligers en verantwoordelijken
7. Administrator → Rapportering - Onderverdeling wie afwezig tijdens zijn shift
8. Administrator → Rapportering - Wie, welke sector, welke post wanneer
9. Administrator → Rapportering - genereren van pdf
10. Administrator → Appel-lijst en Posten weergeven - Shiften weergeven via een digitale kaart.
11. Administrator → Appel-lijst en Posten weergeven - Shiften weergeven via een lijst.
12. Administrator → Appel-lijst en Posten weergeven - Shiften met problemen visueel zichtbaar maken (zowel op de digitale kaart als in de lijst).
13. Administrator → Appel-lijst en Posten weergeven - Filteren op sector bij de digitale kaart (per sector tonen, bepaalde sectoren, alle sectoren tonen)
14. Administrator → Appel-lijst en Posten weergeven - Filteren op sector bij de lijst (per sector tonen, bepaalde sectoren, alle sectoren tonen)
15. Verantwoordelijke → Appel-lijst en Posten weergeven - Shiften weergeven via een digitale kaart.
16. Verantwoordelijke → Appel-lijst en Posten weergeven - Shiften weergeven via een lijst.
17. Verantwoordelijke → Appel-lijst en Posten weergeven - Shiften met problemen visueel zichtbaar maken (zowel op de digitale kaart als in de lijst).
18. Verantwoordelijke → Appel-lijst en Posten weergeven - Filteren op sector bij de digitale kaart (per sector tonen, bepaalde sectoren, alle sectoren tonen)
19. Verantwoordelijke → Appel-lijst en Posten weergeven - Filteren op sector bij de lijst (per sector tonen, bepaalde sectoren, alle sectoren tonen)
20. Vrijwilliger → Authenticatie - Registratie van vrijwilligers (check op gsm-nummer)
21. Verantwoordelijke → Meldingen - Verkrijgen waar problemen zijn: bij afwezigheid
22. Verantwoordelijke → Meldingen - Versturen van problemen: shift te laat begonnen
23. Administrator → Overzicht - Gemelde problemen
24. Vrijwilliger → Meldingen - Melding indien de gebruiker minstens x meter van de shift verwijderd is, en niet richting de shift aan het gaan is, maar de shift wel bijna begint.
25. Vrijwilliger → Meldingen - Melding dat de shift binnen 15 minuten begint, tenzij hij/zij al een soortgelijke melding heeft gekregen.
26. Vrijwilliger → Meldingen - Melding indien de gebruiker zich niet binnen de post-radius bevindt, tijdens zijn shift.

27. Vrijwilliger → Meldingen - Melding indien de gebruiker na y tijd (y tijd is x + ...) nog steeds niet aanwezig is, en vragen naar de reden waarom ze er nog niet zijn.
28. Verantwoordelijke → Controle - Aanwezigheid van vrijwilligers aanduiden
29. Verantwoordelijke → Controle - Aanwezigheid van ‘items’ checken
30. Administrator → Appel-lijst en Posten weergeven - Per post de vorige en volgende shifts kunnen bekijken.
31. Verantwoordelijke → Appel-lijst en Posten weergeven - Per post de vorige en volgende shifts kunnen bekijken.
32. Vrijwilliger → Melden - Melden van technisch en materieel probleem
33. Vrijwilliger → Melden - Melden van verlaten van de post om x reden
34. Verantwoordelijke → Meldingen - Verkrijgen waar problemen zijn: bij technisch/materialistisch probleem
35. Verantwoordelijke → Meldingen - Verkrijgen waar problemen zijn: bij ongeval
36. Verantwoordelijke → Meldingen - Versturen van problemen: bij ongeval
37. Administrator → Appel-lijst en Posten weergeven - Filteren van de lijst op naam.
38. Administrator → Appel-lijst en Posten weergeven - Filteren van de lijst op probleem
39. Verantwoordelijke → Appel-lijst en Posten weergeven - Filteren van de lijst op naam.
40. Verantwoordelijke → Appel-lijst en Posten weergeven - Filteren van de lijst op probleem
41. Administrator → Overzicht - items
42. Administrator → Data - Veranderen van doorgegeven activiteit van de vrijwilliger (check-in, problemen, ...)
43. Administrator → Zichtbare gegevens of een soort sneltoets om de vrijwilliger/administrator te contacteren(bellen / email).
44. Verantwoordelijke → Zichtbare gegevens of een soort sneltoets om de vrijwilliger/administrator te contacteren(bellen / email).
45. Administrator → Contact - Planningswijzigingen naar vrijwilligers
46. Vrijwilliger → Meldingen - Melding dat shift verandert (via bericht, mail of notificatie)
47. Administrator → overzicht - Pop-up wanneer werknemer begint en eindigt aan zijn/haar shift.
48. Administrator → Overzicht - Vrijwilligers kunnen deactiveren
49. Administrator → Overzicht - Details over de vrijwilligers en verantwoordelijken
50. Administrator → Instellingen - De centrale check-tijd manueel kunnen aanpassen (om de hoeveel tijd de vrijwilligers worden gecheckt).
51. Administrator → Contact - Globale informatie naar alle / specifieke vrijwilligers
52. Vrijwilliger → Meldingen - Meldingen van globale informatie
53. Vrijwilliger → Meldingen - Meldingen dat shift verandert
54. Vrijwilliger → Aanwezigheid - Notificatie sturen: "Bent u nog wakker", waar de vrijwilliger binnen het halfuur op moet reageren.
55. Verantwoordelijke → Extra opties - Uit/aanzetten van notificaties/meldingen van de gsm

3.3 Functionele vereisten

- Er wordt gecontroleerd of vrijwilligers zich nog op hun post bevinden. Dit door hun locatie te vergelijken met de locatie van de posten.
- De administrator kan data invoeren via een Excel bestand.
- De administrator kan na het festival de gebeurtenissen uitlezen naar een pdf bestand.
- Problemen worden gemeld aan de administrator en de sector-verantwoordelijke.
- De locatie van de vrijwilligers is zichtbaar op een map voor de administrator.
- De vrijwilliger kan problemen melden, bijvoorbeeld afwezigheid van materiaal.
- Sector-verantwoordelijke kan aanwezig en afwezigheden van materiaal en personen aanduiden.
- De administrator kan planningswijzigingen invoeren.
- vrijwilligers kunnen slechts 1 toestel koppelen aan hun account. Zodat niet iemand anders in hun plaats kan gaan.
- de vrijwilligers krijgen verschillende meldingen, waaronder een melding wanneer hun shift verandert, een melding dat hun shift gaat beginnen en globale berichten van de administrator.

3.4 Niet-functionele vereisten

- De Controle van de vrijwilligers gebeurt om de 15 minuten
- Veranderingen van een shift worden binnen 1 minuut doorgegeven aan de vrijwilligers
- De server moet een constant aantal requests aankunnen van ongeveer 300 personen
- De administrator kan bepaalde vrijwilligers uitzetten, zodat hij hiervan geen afwezigheden of probleem-meldingen kan krijgen. Dit omdat de administrator niet continue een melding krijgt, indien deze vrijwilliger geen internet verbinding heeft.
- Het uitgelezen pdf bestand is overzichtelijk opgesteld, waarop zonder verwarring te zien is wie hoeveel uur afwezig is geweest.
- Er wordt documentatie meegeleverd met instructies over hoe ons systeem te gebruiken.
- De Vrijwilliger applicatie is gemakkelijk in gebruik. De vrijwilliger moet zonder veel extra uitleg aan de slag kunnen gaan bij Pukkelpop.
- Bij de administrator en de sector-verantwoordelijke is efficiënt kunnen werken belangrijker dan het gebruiksgemak.
- Wanneer de vrijwilliger geen internetverbinding heeft tijdens zijn gebruik van onze administrator, zal de locatie data aan de server gegeven worden wanneer hij terug verbinding heeft.
- Onze applicatie is aangepast aan mensen met beperkingen, zoals kleurenblindheid.
- 00:00 wordt 23:59 om verwarring te voorkomen.

4 Analyse

4.1 Algoritmen en Datastructuren

4.1.1 Kortste pad

Een van de extra's in het project is het implementeren van een kortste pad functie voor de sector-verantwoordelijken, zodat deze de minimale afstand moet afleggen om al de posten in hun sector te bezoeken. Als we dit willen doen zonder het gebruik van libraries zullen we gebruik moeten maken van een graaf structuur. Dit aangezien we Dijkstra gebruiken en Dijkstra lost het kortste pad probleem op in een graaf.

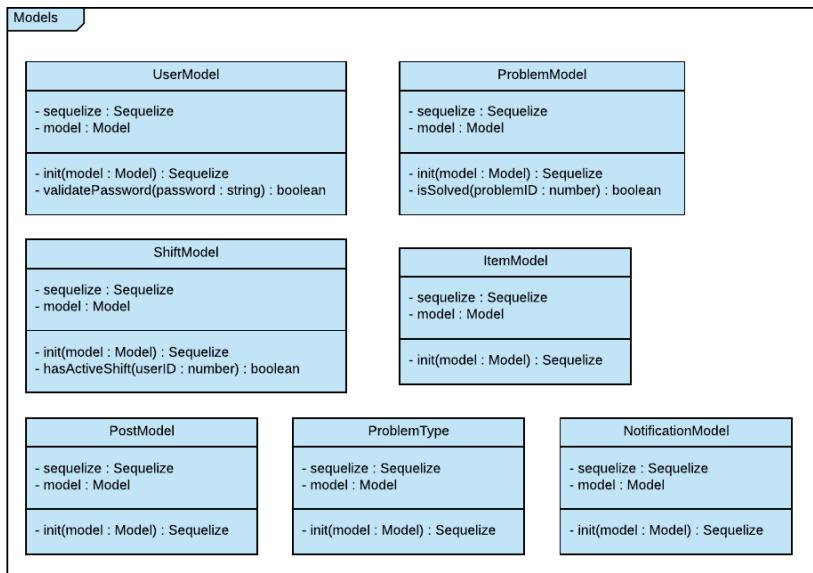
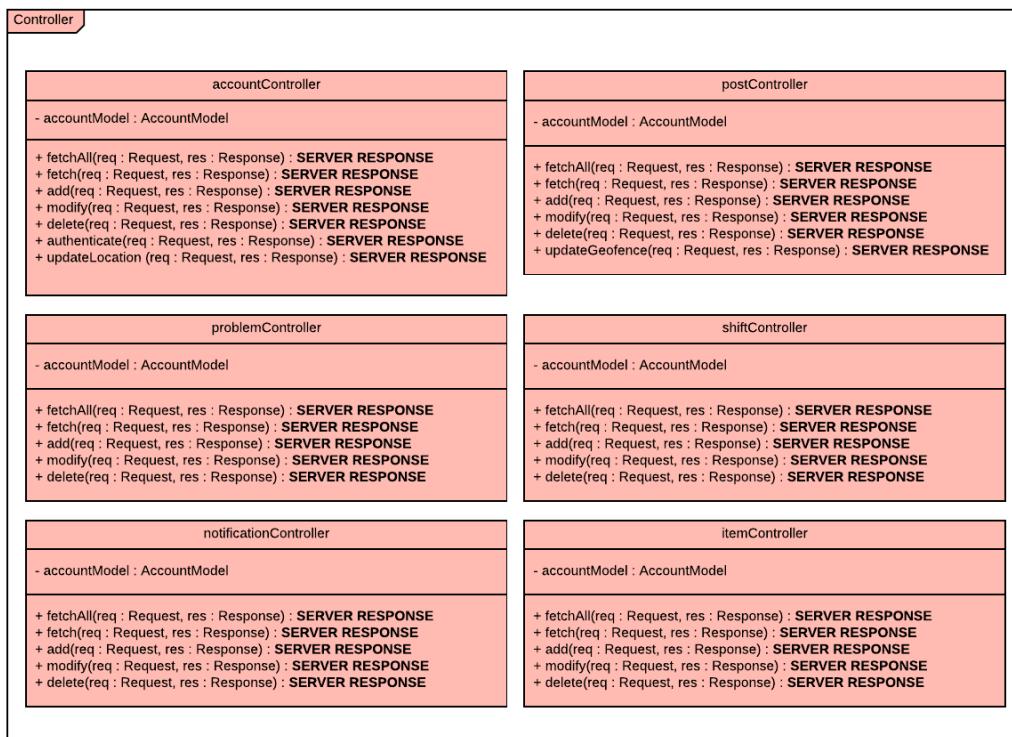
Deze graaf bestaat uit al de straten die zich in de buurt van pukkelpop bevinden. Dit zijn de straten waarop de posten gesitueerd zijn. Om meer exact te zijn, zijn er 2 soorten nodes die we beschouwen. We nemen de verschillende posten als nodes, omdat de sector-verantwoordelijke deze moet bezoeken. Ook de kruispunten van straten beschouwen we als nodes. De gewichten van de bogen tussen de verschillende nodes is de afstand tussen deze 2 nodes.

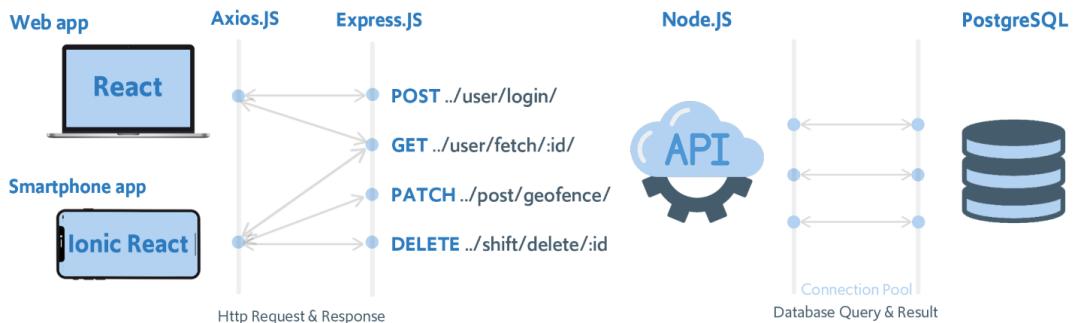
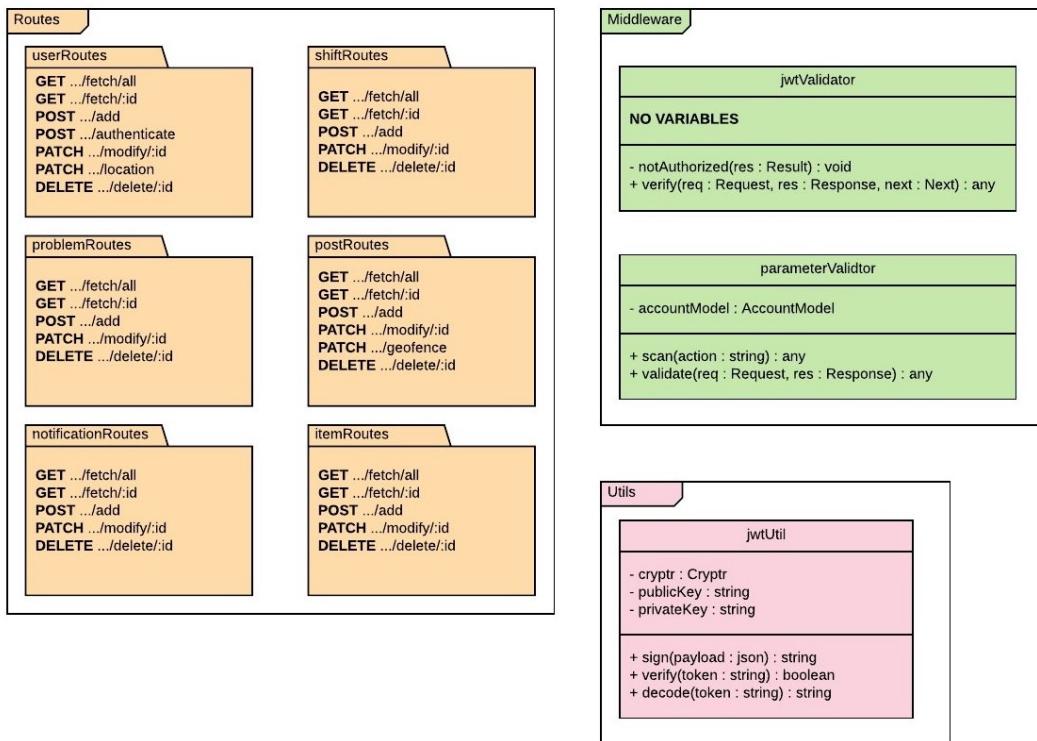
Het algoritme dat we gebruiken is een aangepaste versie van het Dijkstra algoritme. We gebruiken Dijkstra algoritme omdat dit het 'single-source shortest path' berekent in een graaf. Dit is wat wij moeten berekenen, vanaf de gebruiker zijn huidige locatie naar een andere locatie over een zo kort mogelijke afstand. Een verschil waar wij rekening mee moeten houden is dat de sector-verantwoordelijke al zijn posten moet bezoeken, dus zijn er een aantal nodes die in het pad moeten liggen. Hierdoor zullen we het Dijkstra algoritme niet zomaar kunnen toepassen, maar moeten we enkele veranderingen doorvoeren.

Het eerste idee dat we kunnen gebruiken is om al de posten te ordenen over toenemende afstand van het begin punt in vogelvlucht. Waarna we één voor één de kortste paden kunnen berekenen. Dit garandeert echter niet dat de totale afstand het korst is.

Voor de rest worden in het project geen complexe algoritmen of datastructuren gebruikt. Er zal wel gebruik gemaakt worden van de standaard structuren zoals bv. een array.

4.2 Klassendiagram en ontwerp





De front-end en back-end worden van elkaar gescheiden door gebruik te maken van een zogenaamde REST API. Een REST API is een server die de client via HTTP requests kan ‘commanderen’. Elke correct HTTP request naar de REST API stuurt data terug of voegt data toe aan de database.

Het dataformaat, dat gebruikt wordt om data uit te wisselen tussen de client en de server, is JSON.

Een groot deel van de functionaliteit wordt uit handen genomen door de library Sequelize. Dit is een library die het verbinden met de database tot het queryen van de tabellen op zich neemt.

Sequelize valt ook onder de ORM (Object-Relation-Mapper) categorie. Dit houdt in dat de library ervoor zorgt dat wij geen native SQL queries moeten schrijven. In plaats kunnen wij gebruik maken van eenvoudige functies om onze database te queryen.

Door gebruik te maken van Sequelize moeten wij enkel nog een route (bv: .../users/fetch/all) koppelen aan een zelfgemaakte functie. Deze zelfgemaakte functies bevatten de logica van elk

HTTP request, hierbij wordt dus gebruik gemaakt van Sequelize om de database te queren.

Door het gebruik van Sequelize wordt er heel veel complexe functionaliteit uit onze handen genomen. Hierdoor kunnen wij de architectuur van het programma heel simpel houden. Dit is de reden waarom wij gebruik maken van een functionele opbouw en niet gekozen hebben om object georiënteerd te programmeren. Wij vermijden bewust onnodig complexe situaties, zo ontwikkelen wij een robuust en eenvoudig onderhoudbaar systeem voor onze opdrachtgever.

4.3 Data en databaseschema

4.3.1 Database

Er werd gekozen voor PostgreSQL, omdat de support binnen JavaScript beter is voor PostgreSQL dan voor MySQL. PostgreSQL steekt boven MySQL met kop en schouders bovenuit als het gaat om performance wanneer er reads en writes moeten gebeuren. PostgreSQL is veel sneller in het verwerken van JSON-input dan MySQL, dit was het doorslaggevende argument om PostgreSQL te gebruiken.

Wij hebben onderzoek gedaan naar NoSQL databases zoals MongoDB, RethinkDB en Firebase. Ondanks dat zij ideaal zijn voor het verwerken van JSON zijn er toch enkele negatieve aspecten aan deze databases.

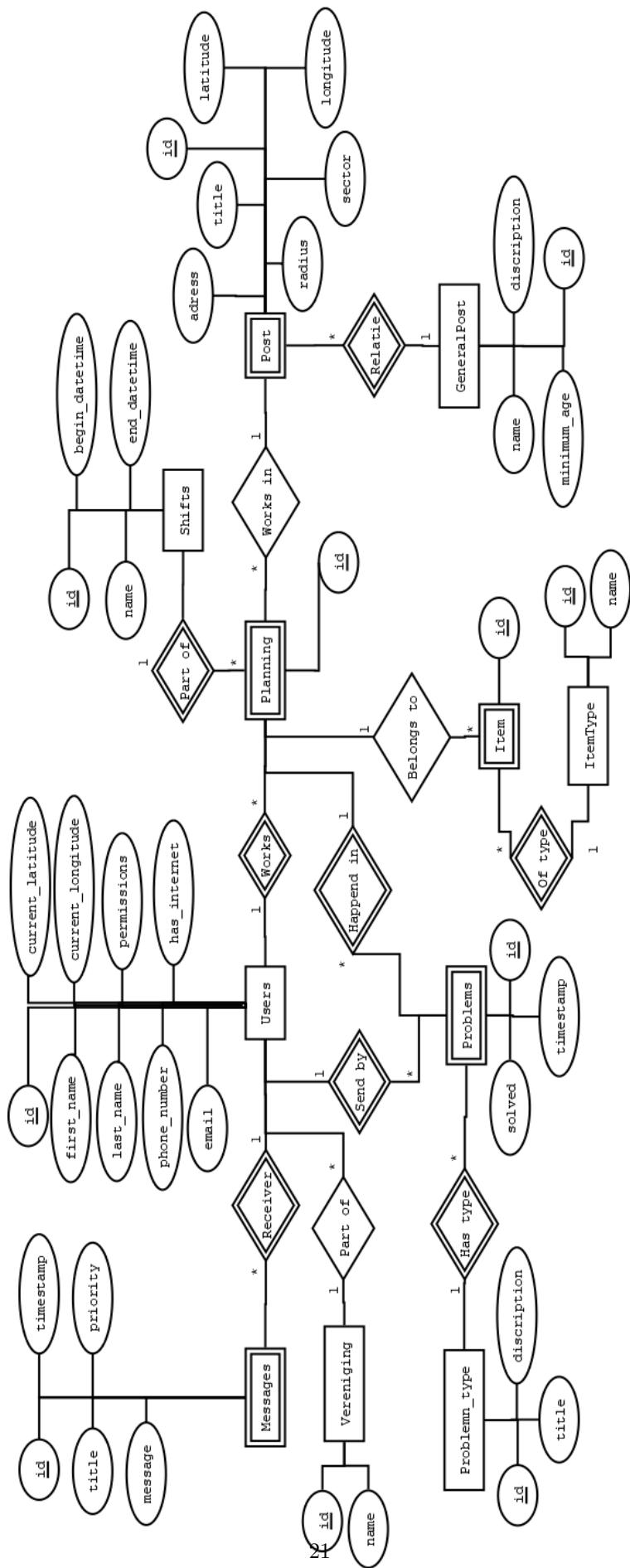
Firebase is je back-end volledig in de cloud, je moet al de functionaliteit creëren via Cloud Functions. Wij maken in het project gebruik maken van complexe logica, hierdoor zou Firebase op een bepaald moment een blok aan ons been worden. Onze klant maakt gebruik van SQL-databases, wij willen daarom hem niet gaan belasten met een NoSQL database. Hierdoor is voor ons NoSQL geen optie meer.

4.3.2 Data

Pukkelpop wilt een opvolging applicatie voor de verschillende vrijwilligers die voor hun werken, hiervoor moeten we data van de vrijwilligers bijhouden, tonen en vergelijken met de verzamelde data. We hebben dus een minimale hoeveelheid aan persoonlijke data nodig. Ook hebben we de opstelling van de planning van Pukkelpop nodig. Deze data zal worden weergegeven aan de mensen om hun tewerkstelling te zien en te zien of iedereen op post is. Onder deze informatie valt ook een opsomming van al de verschillende posten en sectoren die Pukkelpop gebruikt. Hier zullen we dus de huidige locatie van de vrijwilligers vergelijken met de locatie van de verschillende posten. Zodat we kunnen weergeven of de vrijwilligers nog op post zijn en dit kunnen melden.

4.3.3 Databaseschema





Voor onze database hebben wij voorlopig 11 tabellen gedefinieerd, dit zijn volgens ons de voor-naamste tabellen met de belangrijkste data. Echter kunnen er nog veranderingen optreden. De tabellen kunnen nog veranderen afhankelijk van de specifieke input data die we van Pukkelpop krijgen en uiteraard kunnen er tijdens ons project nog tabellen of kolommen veranderen en bijkomen.

We maken in al onze tabellen gebruik van een primary key, id.

De eerste tabel die wij hebben is de ‘Users’ tabel. Deze representeert de verschillende gebruikers, users, die onze applicatie zullen gebruiken. Onder gebruikers vallen zowel de vrijwilligers als de sector-verantwoordelijken. We houden naam en id bij om de gebruiker te kunnen identificeren. Email en telefoonnummer kunnen gebruikt worden om de mensen te contacteren, maar telefoonnummer wordt ook opgeslagen om elke account aan een specifieke gsm te koppelen. Ook wordt bijgehouden bij welke vereniging ze horen, zodat kan bekijken worden hoeveel elke vereniging gewerkt heeft. Dit gebeurt via een verwijzing naar een aparte tabel ‘Vereniging’. Voor het controleren of de vrijwilligers zich op hun post bevinden houden we ook hun locatie bij. Als laatste is er het attribuut ‘Has_internet’, wat zegt of een vrijwilliger internet heeft of niet. Dit heeft de functie dat als een vrijwilliger geen internet heeft en dus ook zijn locatie niet kan doorsturen. De admin geen melding krijgt dat hij of zij niet op post is.

De tabel ‘Shifts’ bevat de verschillende shiften die de vrijwilligers kunnen invullen gedurende Pukkelpop. Onder een shift valt een werkperiode met een specifiek begin- en einduur. Meerder mensen kunnen op dezelfde shift werken. Verschillende shiften kunnen overlappen. Kortom is ‘Shifts’ een verzameling van alle verschillende begin- en einduren.

De ‘Post’ tabel bevat de info van al de posten. ‘Latitude’ en ‘longitude’ zijn de locatie van de post. Samen met de kolom ‘radius’ kunnen we controleren of vrijwilligers zich op een post bevinden. ‘Title’ en ‘adress’ dienen om extra informatie beschikbaar te stellen voor de gebruikers. De sector waarbij deze post hoort, wordt ook bijgehouden. Ten laatste is er ook een verwijzing naar de tabel ‘GeneralPost’, die extra info weergeeft over een specifieke taak.

De tabel ‘GeneralPost’ dient om algemene informatie bij te houden die van toepassing is bij verschillende posten. Hieronder valt een specifieke naam van de post, bijvoorbeeld parking, straat afzetten of catering. Verder wordt hier ook een kleine beschrijving gegeven van de taken en de minimumleeftijd gegeven.

De ‘Planning’ tabel wordt gebruikt om uit te drukken wie, waar en wanneer moet werken. Hij bevat een entry voor elke user per shift hij of zij werkt. Hieraan is dan ook nog verbonden op welke post hij of zij deze shift moet uitvoeren.

‘ItemType’ bevat al de verschillende objecten die nodig kunnen zijn op het festival, bijvoorbeeld een fluohesje, een zaklamp om er een paar op te noemen.

Hiervan maakt de tabel ‘Items’ gebruik om weer te geven welk soort item het is. Men kan namelijk meerdere fluohesjes hebben, deze zijn unieke items, maar verwijzen naar hetzelfde itemtype. ‘Items’ bevat ook nog een verwijzing naar ‘Planning’, zodat kan gecontroleerd worden welke items de vrijwilligers nodig hebben en wanneer en door wie er problemen optraden.

De tabel ‘Problems’ geeft al de problemen weer die kunnen optreden tijdens Pukkelpop. Het type probleem verwijst naar de ‘Problem_type’ tabel. Een probleem treedt op tijdens het werk van een vrijwilliger, waardoor we de relatie tussen ‘Problems’ en ‘Planning’ hebben. Ook wordt er bijgehouden wie het probleem gemeld had of over wie het probleem gaat, hierdoor de relatie tussen ‘Problems’ en ‘User’. ‘Solved’ gebruiken we om aan te duiden of het probleem opgelost is, zodat we ook de data van de opgeloste problemen nog kunnen laten zien. Ten laatste houden we een timestamp bij, zodat de problemen doorheen de tijd gezien kunnen worden.

De ‘Problem_Type’ tabel definieert de verschillende types om de soorten problemen te onderscheiden. Als enkele voorbeelden: ‘Niet op post’, ‘Materiaal afwezig’. Priority laat ons toe een

onderscheid te maken tussen belangrijke en dringende problemen en minder belangrijke problemen.

Als laatste hebben we de tabel 'Messages'. Deze slaat de verschillende notificaties en berichten op die de admin verstuurd naar de users. We houden hier ook bij welke user de notificatie ontvangt. Ook hier dient priority voor een onderscheid volgens het belang van het bericht.

4.4 Bestandsformaat

Voor het inlezen van de planning en gerelateerde data maken we gebruik van Excel. De bestanden worden momenteel gegenereerd door Pukkelpop zelf en kunnen we dus niet veel aan veranderen. Ze maken gebruik van verschillende Excel files. De eerste 'Shiften' bestaat uit 3 tabellen, de shift naam met begin- en einduur. De tweede 'Functies' bevat de verschillende posten die Pukkelpop gebruikt met daarbij extra informatie. Het bevat een beschrijving, de sector, het adres en de minimum leeftijd. Een derde bestand 'Hoofdfuncties' verbindt de shiften met hun bijhorende posten. Als laatste krijgen we nog een file die de planning van Pukkelpop weergeeft. Deze beschrijft wie op welke post moet werken.

Ook moeten we na afloop van Pukkelpop onze verzamelde data uitschrijven naar een pdf bestand. De opmaak van dit bestand is voor Pukkelpop niet van belang. We kunnen hierover slechts zeggen dat we voor een overzichtelijk bestand gaan waarop een duidelijke opsomming is van de problemen die zijn voorgekomen tijdens het festival en de gepresteerde uren van de verschillende vrijwilligers hun verenigingen.

Om de communicatie tussen de verschillende modules in onze software te vergemakkelijken maken we ook gebruik van json.

5 Mockups

5.1 Algemene toelichting

Het uiterlijk van de applicatie zou gebruiksvriendelijk moeten zijn. Om dit te bereiken hebben wij rekening gehouden met de volgende aspecten:

- Het aantal stappen tot een bepaald scherm beperkt houden.
 - Het maximaal aantal stappen tot een bepaald scherm is drie.
- Elk scherm heeft een specifiek doel, waarbij het aantal functies per scherm minimaal wordt gehouden. Indien er meerdere functies zijn, zullen dit eerder hulp-functies zijn, die het hoofddoel zullen helpen te bereiken.
- Het uiterlijk is gelijkaardig aan andere applicaties. Zo wordt er niet plots gebruik gemaakt van een nieuw design waar de gebruiker niets van herkend.
- Componenten met dezelfde functie (zoals bijvoorbeeld knoppen) hebben een gelijkaardig layout.
- Indien men dieper gaat in de structuur, worden de details specifieker. Zo zal er eerst een overzicht getoond worden, waarna men de afzonderlijke componenten gedetailleerder kan bekijken, indien men op dit component klikt.
- Er wordt gebruik gemaakt van kleuren om bepaalde componenten van elkaar te onderscheiden.
 - Zo zijn er bijvoorbeeld meerdere sectoren op een digitale kaart, om dit visueel zichtbaar te maken, is er gebruik gemaakt van een kleur per sector.
- Sneltoetsen.
- Geen conflicterende kleuren.
- Verduidelijkende iconen.
- De grootte en verhoudingen van de componenten.
- Het device waarop de applicatie gebruikt zal worden (smartphone versus desktop).
 - Aangezien een smartphone veel kleiner is dan een desktop, is het aantal functies zichtbaar per scherm op de smartphone veel minder talrijk dan het aantal functies op een desktop. Aangezien een desktop groter is, kunnen er bijvoorbeeld gerelateerde componenten naast elkaar gezet worden.
- Hoe het device gebruikt zal worden: al zittend versus onderweg.
 - Indien de gebruiker zit, zijn er meer details zichtbaar dan indien de gebruiker de applicatie al lopend gebruikt.
- Waar het device gebruikt zal worden: buiten in de nacht, buiten overdag of binnen onder een dak.
 - Indien de zon schijnt, is een donkere achtergrond moeilijk zichtbaar. Daarom wordt er standaard gebruikt gemaakt van een lichte achtergrond. Indien het donker is, wordt een donkere achtergrond als aangenamer beschouwt door sommige mensen. Aangezien de applicatie ook in het donker gebruikt zal worden, zal de gebruiker de keuze krijgen om over te schakelen naar nachtmodus.

5.2 Sector-verantwoordelijke

5.2.1 Toelichting

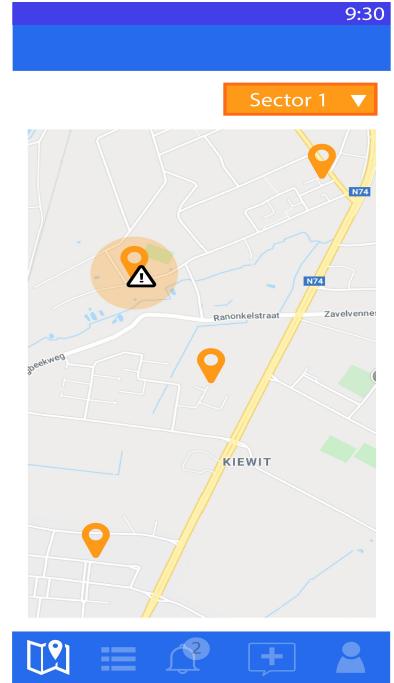
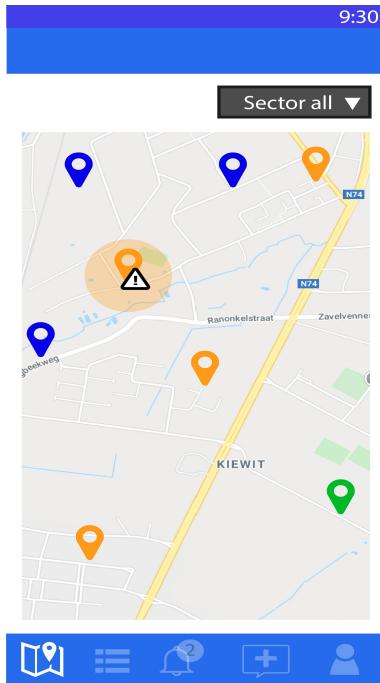
De sector-verantwoordelijke gebruikt de applicatie wanneer hij/zij onderweg is naar een bepaalde post. Daarom is er een scherm met enkel een digitale kaart, waar zowel de sector-verantwoordelijke zijn/haar huidige locatie alsook de locatie van de posten opstaan. Er kan hier eventueel nog een feature extra bijkomen, waarbij de sector-verantwoordelijke de applicatie een voorgestelde weg kan laten genereren om deze vervolgens te nemen. Indien de sector-verantwoordelijke een lijst gemakkelijker vindt om zijn/haar tour te doen, is deze ook ter beschikking.

Eens aangekomen bij de post, kan de sector-verantwoordelijke ofwel op de marker klikken op de digitale kaart, ofwel via de lijst met posten naar de desbetreffende post gaan. Er wordt dan de info weergegeven die bij de specifieke post hoort. Daarnaast zijn de functies die moeten uitgevoerd worden bij de post ook ter beschikking op hetzelfde scherm. Zo moet de sector-verantwoordelijke bijvoorbeeld checken of de vrijwilligers en de materialen aanwezig zijn. Daarnaast moet er ook gecheckt worden dat er geen fout-gemelde problemen zijn, of dat er bepaalde problemen niet doorgekomen zijn. Indien de sector-verantwoordelijke een extra notitie wil maken bij de desbetreffende post, kan hij dit doen door het te melden als een probleem.

Naast de MapView en de ListView zijn er in het menu nog drie andere sneltoesten, namelijk: notificaties, melding sturen en contacten. Het notificatie scherm geeft een overzicht van alle meldingen. Hierbij wordt er duidelijk vermeld van wie de melding komt. Indien de melding reeds bekeken is, is de achtergrond grijs. Indien de melding nieuw is, heeft de melding een witte achtergrond. Naast meldingen ontvangen, kan een sector-verantwoordelijke ook meldingen versturen. Hiervoor is een apart scherm voorzien, waarbij de verantwoordelijke een ontvanger kan kiezen, een titel moet invullen en de melding beschrijven.

Tot slot is er ook nog een apart scherm voor de contacten. De lay-out is hetzelfde als de manier waarop contacten worden weergegeven in een smartphone. In het overzicht zijn er sneltoetsen om de persoon te contacteren. Indien men niet zeker is dat men de juiste persoon voor zich heeft, kan er nog geklikt worden op de persoon, waarna men in een scherm komt met meer details over deze persoon.

5.2.2 Afbeeldingen



Figuur 4: ListView van de posten: sector op 1.



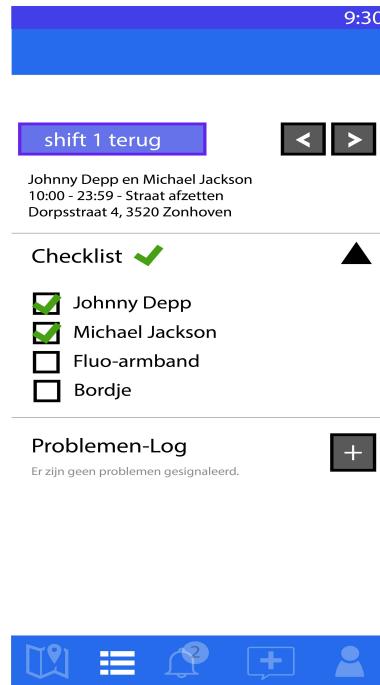
Figuur 5: ListView van de posten: sector op all.



Figuur 6: ListView van de posten: filteren.



Figuur 7: ListView van de posten: sorteren.



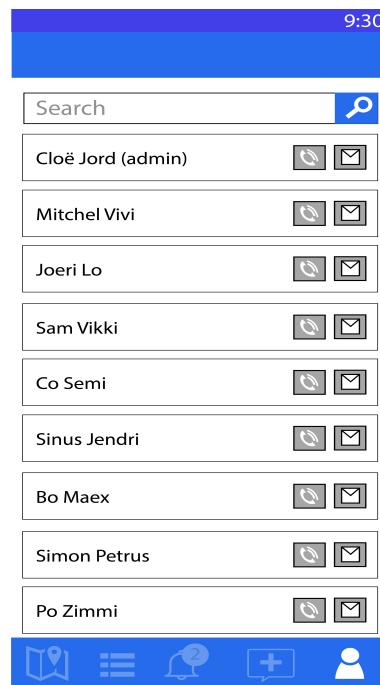
Figuur 8: PostView: Indien men op een post klikt.



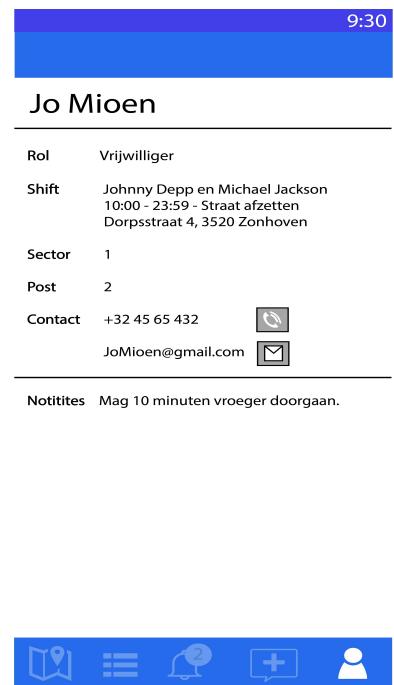
Figuur 9: Notificaties: grijs is reeds bekeken.



Figuur 10: Een notificatie verzenden naar ...



Figuur 11: Contacten: lijst met sneltoetsen en zoekbalk.



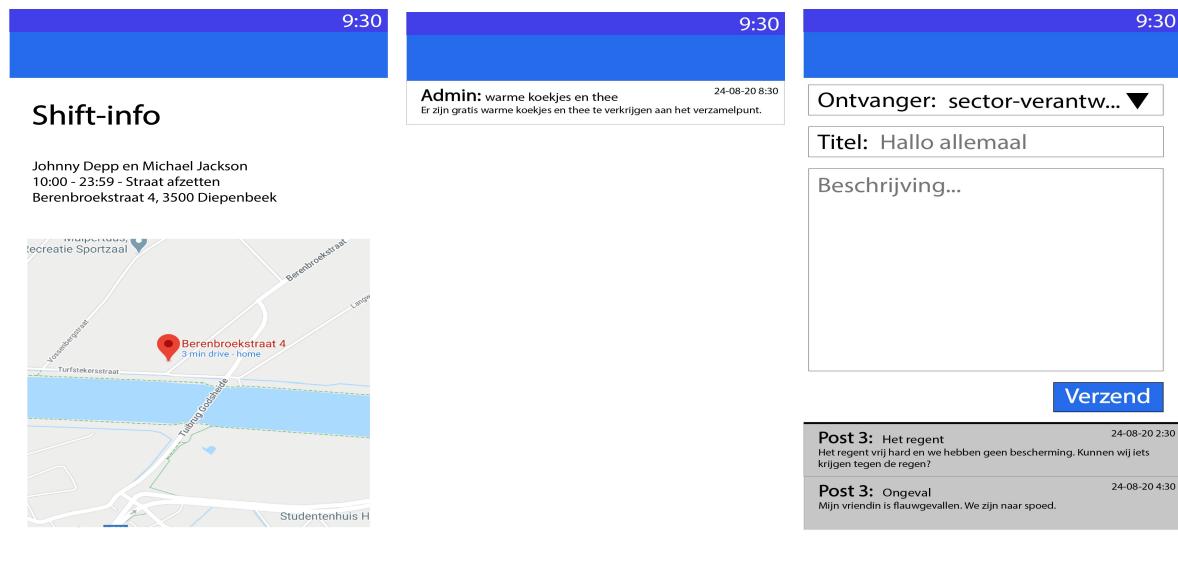
Figuur 12: Contacten: een individueel contact.

5.3 Vrijwilliger

5.3.1 Toelichting

Aangezien de vrijwilliger de applicatie niet actief moet gebruiken, heeft applicatie ook een beperkt aantal functies. De vrijwilliger ziet als hoofdscherm de info voor zijn/haar shift. Daarnaast kan de vrijwilliger ook notificaties ontvangen en worden deze bijgehouden in het notificatie-scherm. Tot slot is er ook nog een mogelijkheid om problemen te melden aan de sector-verantwoordelijke.

5.3.2 Afbeeldingen



Figuur 13: Het hoofdscherm van de vrijwilliger.

Figuur 14: De gekregen notificaties.

Figuur 15: De verantwoordelijke te verwittigen.

5.4 Admin

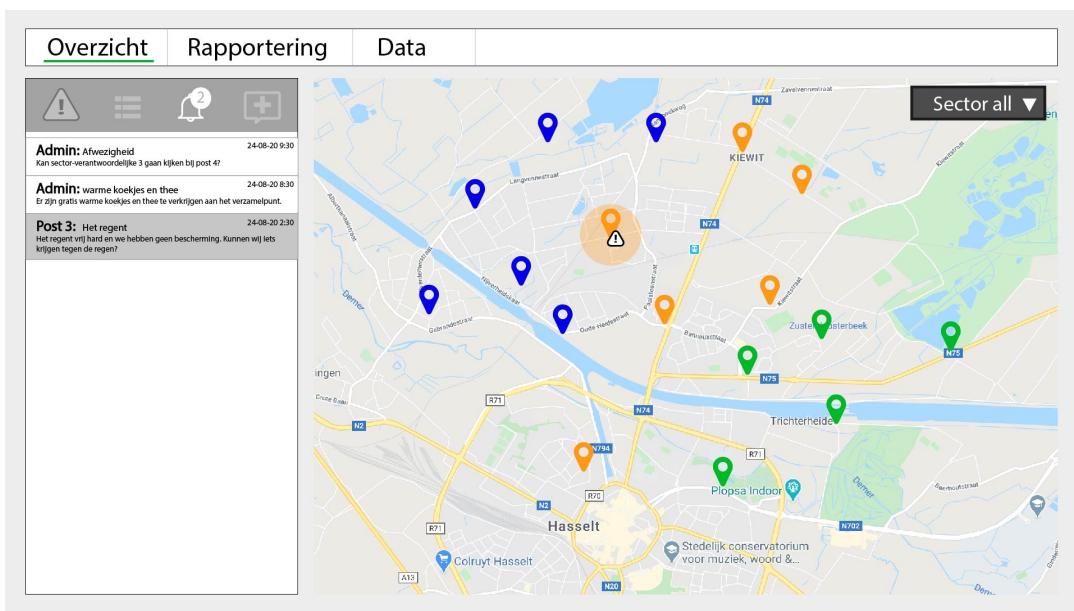
5.4.1 Toelichting

De admin heeft eerder een controlerende rol. Er is voor de applicatie van de admin gekozen om bepaalde functionaliteiten hetzelfde te integreren zoals bij de applicatie van de sector-verantwoordelijke. Bij de tab "Overzicht" is het beeld namelijk opgedeeld in twee. Links een bijna letterlijke kopie van de applicatie van de sector-verantwoordelijke, en rechts een digitale map waarbij de admin een overzicht heeft van alle sectoren en eventuele problemen.

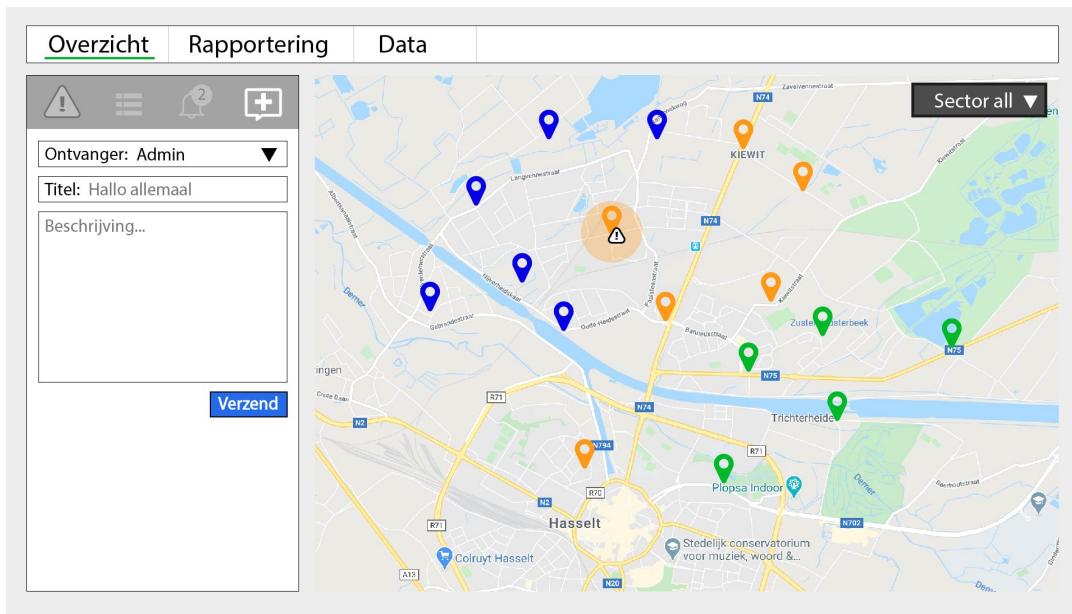
Daarnaast heeft de admin ook de mogelijkheid om alles te initialiseren qua data. Dit kan dan door een excel file te uploaden, waarbij de desbetreffende data verwerkt wordt door de applicatie en ook toegevoegd wordt aan de database. Vervolgens kan de admin de data op meerdere manieren bekijken in de tabs hiervoor voorzien.

Tot slot is er ook nog de mogelijk om een pdf te genereren met de rapportering.

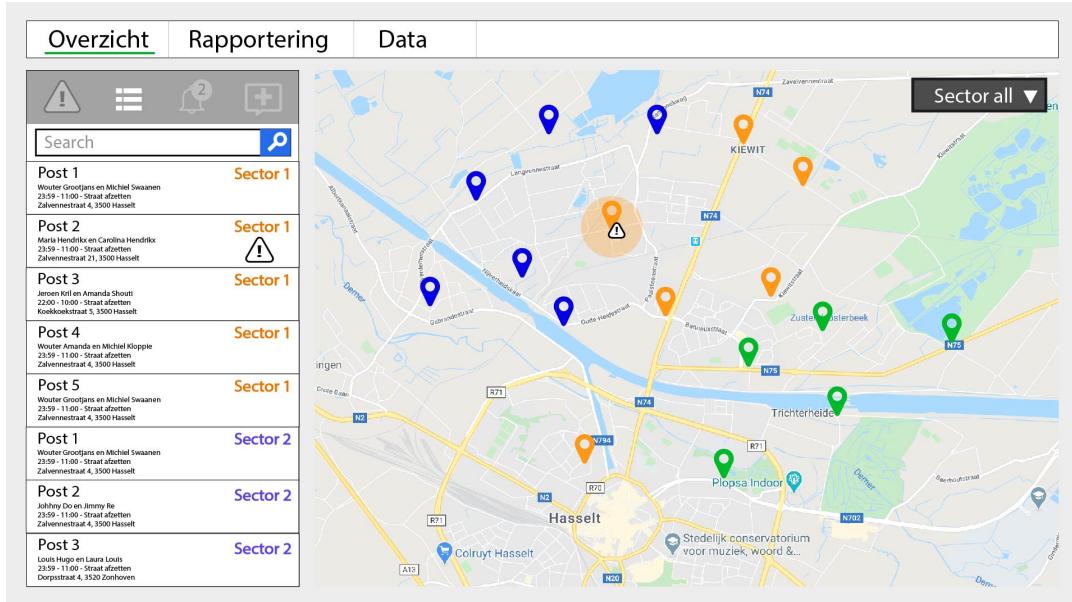
5.4.2 Afbeeldingen



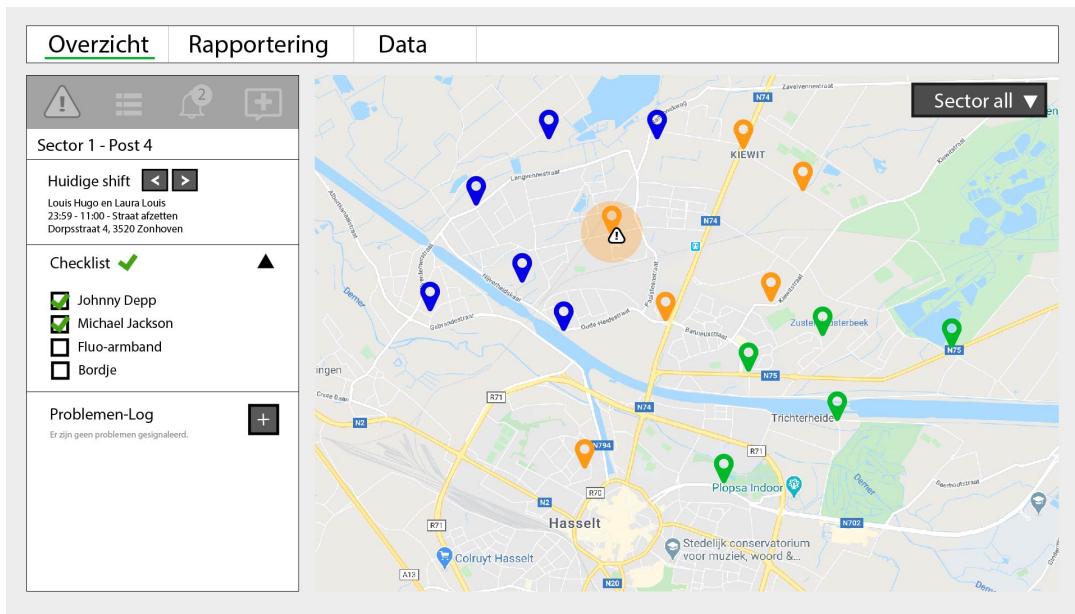
Figuur 16: De map is altijd zichtbaar, waarbij men links dezelfde functionaliteiten heeft als de sector-verantwoordelijke.



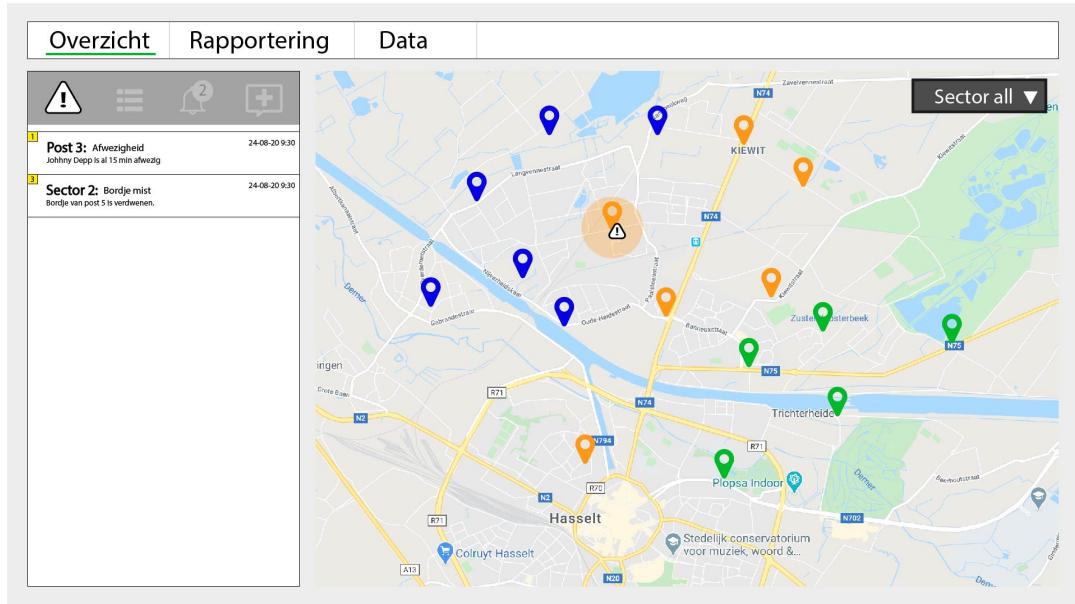
Figuur 17: Notificatie verzenden.



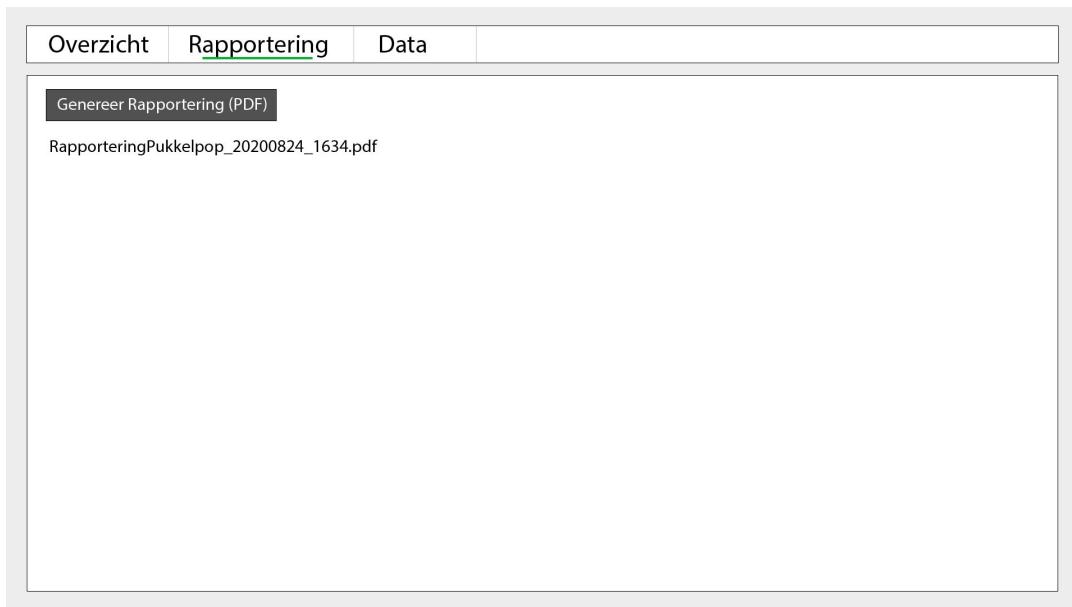
Figuur 18: ListView en MapView langs elkaar.



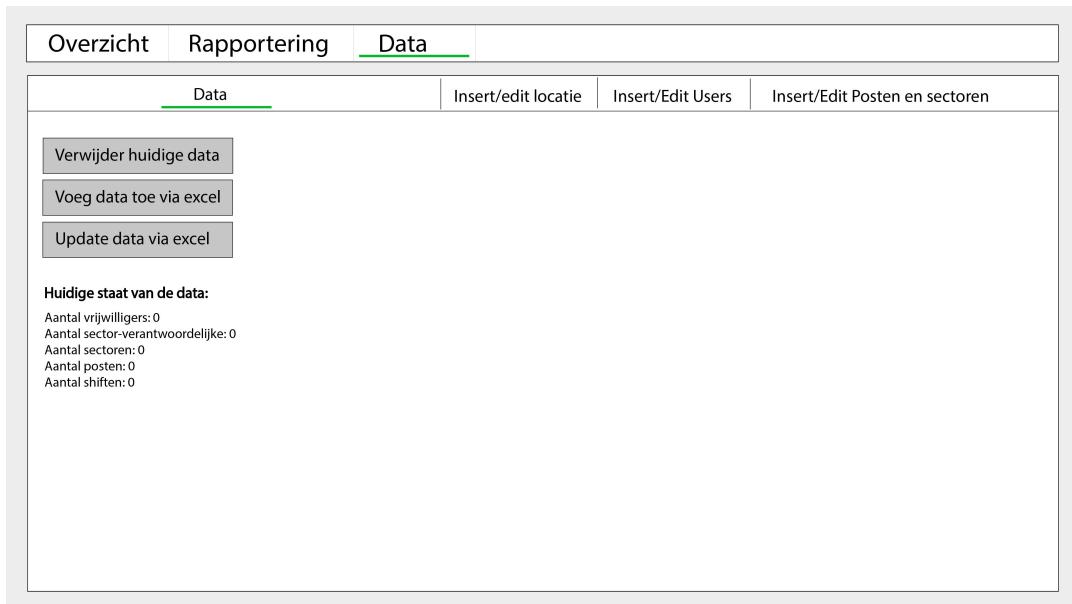
Figuur 19: Indien men op een post duwt (via ListView of via MapView).



Figuur 20: De problemen worden weergegeven in een lijst.



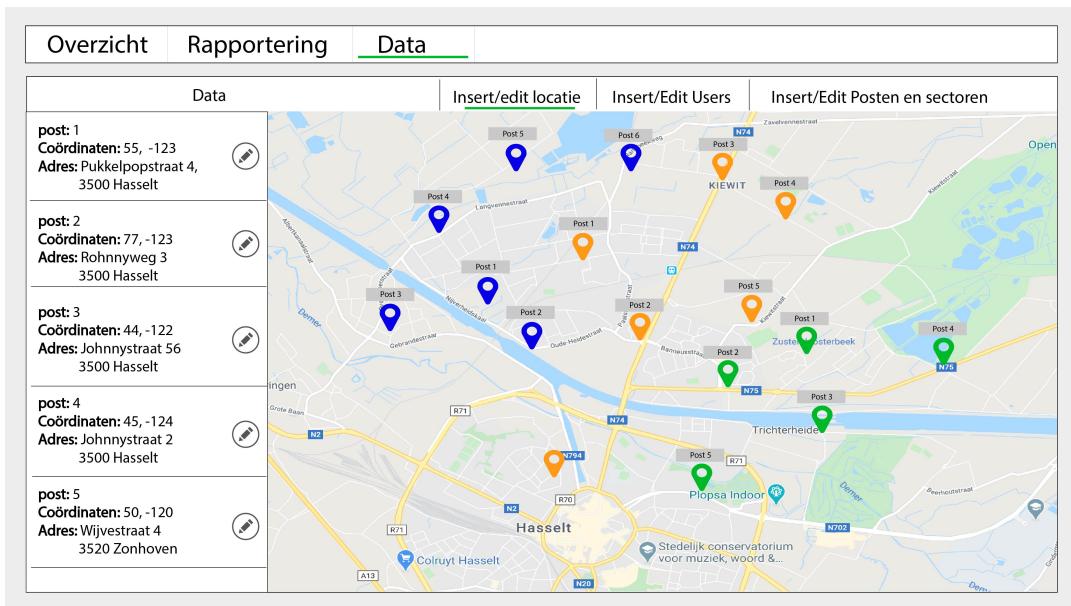
Figuur 21: Een mogelijkheid om de gekregen data om te zetten in een pdf.



Figuur 22: Indien men data wil invoegen.

Overzicht	Rapportering	<u>Data</u>												
<table border="1"> <tr> <td>Data</td> <td>Insert/edit locatie</td> <td>Insert/Edit Users</td> <td>Insert/Edit Posten en sectoren</td> </tr> <tr> <td> <input type="button" value="Verwijder huidige data"/> <input type="button" value="Voeg data toe via excel"/> <input type="button" value="Update data via excel"/> </td> <td colspan="3"></td> </tr> <tr> <td colspan="4">Huidige staat van de data: Aantal vrijwilligers: 3000 Aantal sector-verantwoordelijke: 100 Aantal sectoren: 5 Aantal posten: 40 Aantal shiften: 11</td> </tr> </table>			Data	Insert/edit locatie	Insert/Edit Users	Insert/Edit Posten en sectoren	<input type="button" value="Verwijder huidige data"/> <input type="button" value="Voeg data toe via excel"/> <input type="button" value="Update data via excel"/>				Huidige staat van de data: Aantal vrijwilligers: 3000 Aantal sector-verantwoordelijke: 100 Aantal sectoren: 5 Aantal posten: 40 Aantal shiften: 11			
Data	Insert/edit locatie	Insert/Edit Users	Insert/Edit Posten en sectoren											
<input type="button" value="Verwijder huidige data"/> <input type="button" value="Voeg data toe via excel"/> <input type="button" value="Update data via excel"/>														
Huidige staat van de data: Aantal vrijwilligers: 3000 Aantal sector-verantwoordelijke: 100 Aantal sectoren: 5 Aantal posten: 40 Aantal shiften: 11														

Figuur 23: Wanneer men data invoegt, krijgt de gebruiker feedback door de samenvatting



Figuur 24: Een mogelijkheid om de locaties manueel te veranderen. Men kan dit doen door de juiste locatie op de map aan te duiden, of door het adres/coördinaten met eigen waarden in te vullen.

Overzicht Rapportering <u>Data</u>								
Data				Insert/edit locatie		<u>Insert/Edit Users</u>		Insert/Edit Posten en sectoren
ID	Gedeactiveerd	Naam	Voornaam	Gsm-nr	E-mail	Adres		Rol
1	X	Hendrickx	Isabelle	/	Isabell@gmail.com	Lodewijkstraat 4	Vrijwilliger	
2		Hendrickx	Roger	04852666666	Roger@gmail.com	Lodewijkstraat 4	Vrijwilliger	
3		Lo	Karen	04563213545	Karen@gmail.com	Pinostraat 3	Vrijwilliger	
4		Lo	Evelien	04321654556	123@gmail.com	Pinostraat 3	Vrijwilliger	
5		Swane	Mitchel	04832132165	Swane@gmail.com	Pinostraat 3	Vrijwilliger	
6		Grootie	Wotte	04894321325	Grootie@gmail.com	Pinostraat 3	Vrijwilliger	
7	X	Hohoi	Louis	04835616548	Hohi@gmail.com	Yoyostrat 654	Sector-verantwoordelijk	

Voeg gebruiker toe

Figuur 25: Een mogelijkheid om de gebruikers (users) te bewerken.

Overzicht Rapportering <u>Data</u>									
Data					Insert/edit locatie		<u>Insert/Edit Users</u>		Insert/Edit Posten en sectoren
ID	Sector	Post-ID	Post-naam	Post Locatie	Post verantwoordelijk		Vereniging		
1	1	1	A-B	Rogerstraat 4, 3520 Zonhoven	Koen Wauters	Chiro Kiewit			
2	1	2	Po-We	Hohiaweg 4, 3500 Hasselt	Pe Roger	Chiro Kiewit			
3	1	3	A-D	Koboltweg 3, 3500 Hasselt	Kaat Vanny	Chiro Kiewit			
4	2	1	U-V	Superstraat 5, 3500 Hasselt	Jozefien Vroemie	Chiro Zonhoven			

Voeg een Sector/Post toe

Figuur 26: Een mogelijkheid om de sectoren en posten te bewerken.

6 Taakverdeling en planning

6.1 De planning

28/03/2020	04/04/2020	Maria	Sector-verantwoordelijke: Algemene Layout
		Maria	Sector-verantwoordelijke: Het weergeven van de posten via een lijst
		Maria	Sector-verantwoordelijke: Het weergeven van de posten via een kaart
		Maria	Sector-verantwoordelijke: filter functie op de weergaven van de posten
		Wouter	Admin: Inlezen posten en sectoren uit excel en invoegen in de database
		Wouter	Admin: Inlezen vrijwilligers en verantwoordelijke en invoegen in de database
		Wouter	Admin: basis admin website met verschillende tabs
		Wouter	Admin: de mogelijkheid tot het inlezen van de data
		Michiel	Hulp: helpt Wouter en Maria bij eventuele problemen van React
		Michiel	Hulp: helpt Wouter en Maria eventueel bij taakjes die niet op tijd afgeraken.
		Michiel	Database aanmaken
		Michiel	Backend maken
		Maria	Verslag aanvullen
		Wouter	Verslag aanvullen
		Michiel	Verslag aanvullen
04/04/2020	11/04/2020	Maria	Sector-verantwoordelijke: notificaties-layout
		Maria	Sector-verantwoordelijke: notificaties kunnen ontvangen
		Maria	Sector-verantwoordelijke: notificaties kunnen versturen
		Maria	Sector-verantwoordelijke: Lijst van contacten
		Maria	Sector-verantwoordelijke: Contacten kunnen weergeven
		Maria	Sector-verantwoordelijke: ListView van posten kunnen sorteren
		Maria	Sector-verantwoordelijke: Het weergeven van een probleem op de digitale kaart.
		Maria	Sector-verantwoordelijke: het weergeven van een probleem in de lijst
		Wouter	Admin: representatie van user data
		Wouter	Admin: Rapportering: uitschrijven van data naar pdf
		Wouter	Admin: representatie van de posten
		Michiel	Vrijwilliger: de locatie opropen in de achtergrond
		Michiel	Locatie van vrijwilliger vergelijken met locatie van de posten
		Michiel	Vrijwilliger: de tracing van de vrijwilliger
		Michiel	Vrijwilliger: de basis layout
		Allermaal	Tussentijdse besprekking over de stand van zaken
		Allermaal	Tussentijdse besprekking om de planning eventueel aan te passen
		Allermaal	Tussentijdse besprekking over bepaalde problemen
11/04/2020	18/04/2020	Maria	Sector-verantwoordelijke: De PostView in orde maken (layout)
		Maria	Sector-verantwoordelijke: eventuele statische gegevens, dynamisch maken.
		Maria	Sector-verantwoordelijke: eventuele statische gegevens, dynamisch maken.
		Maria	Sector-verantwoordelijke: Instellingen: mogelijkheid om default-sector te selecteren.
		Maria	Sector-verantwoordelijke: Instellingen: mogelijkheid om notificaties aan/uit te zetten
		Wouter	Admin: Planning uit excel inlezen en invoegen in de database
		Wouter	Admin: algemene layout van de tab "overzicht"
		Wouter	Admin: Pop-up indien een werknemer begint en eindigt aan zijn/haar shift
		Michiel	Vrijwilliger: Versturen van meldingen aan sector-verantwoordelijke
		Michiel	Vrijwilliger: Ontvangen van notificaties
		Michiel	Hulp: helpt Wouter en Maria bij eventuele problemen van React
		Michiel	Hulp: helpt Wouter en Maria eventueel bij taakjes die niet op tijd afgeraken.
		Maria	Verslag aanvullen
		Wouter	Verslag aanvullen
		Michiel	Verslag aanvullen

18/04/2020	25/04/2020	Allemaal Een eerste prototype laten uittesten door een externe student Allemaal Een eerste prototype laten uittesten door een externe persoon (leeftijd ouder dan 50) Allemaal Een eerste prototype uittesten en debuggen Allemaal Tussentijdse rapportering maken Allemaal Samenkomen om de tussentijdse rapportering te bespreken Allemaal Tussentijdse rapportering Allemaal Tussentijdse bespreking over de stand van zaken Allemaal Tussentijdse bespreking om de planning eventueel aan te passen Allemaal Tussentijdse bespreking over bepaalde problemen Wouter Admin: digitale kaart met posten erop weergeven Wouter Admin: Het weergeven van de posten via een lijst Maria Admin: notificaties-layout Maria Admin: notificaties kunnen ontvangen Maria Admin: notificaties kunnen versturen Michiel Admin: Het weergeven van een probleem op de digitale kaart.
25/04/2020	02/05/2020	Wouter Admin: Het weergeven van de posten via een kaart Wouter Admin: filter functie op de weergaven van de posten Maria Admin: Lijst van contacten Maria Admin: Contacten kunnen weergeven Maria Admin: ListView van posten kunnen sorteren Michiel Admin: De PostView in orde maken (layout) Maria Admin: eventuele statische gegevens, dynamisch maken. Wouter Admin: eventuele statische gegevens, dynamisch maken. Michiel Admin: Instellingen: mogelijkheid om default-sector te selecteren. Michiel Admin: Instellingen: mogelijkheid om notificaties aan/uit te zetten Maria Admin: Instellingen: mogelijkheid voor Light/Dark Modus. Allemaal Een eerste prototype laten uittesten door een externe student Allemaal Een eerste prototype laten uittesten door een externe persoon (leeftijd ouder dan 50) Allemaal Een eerste prototype uittesten en debuggen Allemaal Tussentijdse rapportering maken Allemaal Samenkomen om de tussentijdse rapportering te bespreken Allemaal Tussentijdse rapportering
02/05/2020	09/05/2020	Allemaal Tussentijdse bespreking over de stand van zaken Allemaal Tussentijdse bespreking om de planning eventueel aan te passen Allemaal Tussentijdse bespreking over bepaalde problemen Allemaal Tussentijdse bespreking Extra week ruimte, zodat indien de planning te strak is, of slecht is ingeschat, deze Allemaal week nog als backup kan gebruikt worden.
09/05/2020	16/05/2020	Allemaal Documentatie afwerken Allemaal Debuggen Allemaal Verslag aanvullen
16/05/2020	23/05/2020	Allemaal Tussentijdse bespreking voor het verslag Allemaal Eventuele nutteloze code verwijderen Allemaal Eventuele extra's implementeren Extra week ruimte, zodat indien de planning te strak is, of slecht is ingeschat, deze
23/05/2020	30/05/2020	Allemaal week nog als backup kan gebruikt worden.
30/05/2020	03/06/2020	Allemaal Bespreking eindpresentatie Allemaal Maken van eindpresentatie Allemaal Inoefenen van de eindpresentatie

6.2 De planning als tijdlijn



6.3 Mijlpalen

6.3.1 Toelichting

De mijlpalen staan al vrij vroeg ingepland. Dit omdat er in mei nog verschillende andere projecten zijn bij de andere vakken, en de examens daarnaast ook dichterbij komen. Aangezien dit allemaal samen vrij veel stress oplevert, lijkt het ons beter om deze applicatie al reeds in begin mei klaar te hebben. Indien dit niet lukt, hebben wij eventueel nog extra tijd.

6.3.2 Mijlpalen

- 11/04/2020 Admin: Data, Rapportering klaar hebben
- 11/04/2020 Sector-verantwoordelijke: ListView, MapView en PostView klaar hebben
- 18/04/2020 Vrijwilliger: klaar hebben
- 30/04/2020 Sector-verantwoordelijke: Alles (behalve de extra's) klaar hebben
- 10/05/2020 Admin: alles klaar hebben

7 Bibliografie

Referenties

- [1] CoinBase, Retrieved from: <https://www.pinterest.com/pin/42291683983214113/>.
- [2] Strava, Retrieved from: <https://strava.com>.
- [3] Bottom Navigation afbeelding, Retrieved from: <https://material.io/components/bottom-navigation/>.
- [4] NativeScript, Retrieved from: <https://en.wikipedia.org/wiki/NativeScript>.
- [5] Ionic Framework, Retrieved from: [https://ionicframework.com/](https://ionicframework.com).
- [6] CodeName One, Retrieved from: https://en.wikipedia.org/wiki/Codename_One.
- [7] Flutter Release, Retrieved from: <https://developers.googleblog.com/2018/02/announcing-flutter-beta-1.html>
- [8] Cordova and Ionic, Retrieved from: <https://stackshare.io/stackups/apache-cordova-vs-ionic>
- [9] Hybrid definition, Retrieved from: <https://searchsoftwarequality.techtarget.com/definition/hybrid-application-hybrid-app>
- [10] Mobile Frameworks, Retrieved from: https://en.wikipedia.org/wiki/Mobile_development_framework
- [11] Flutter documentation, Retrieved from: <https://flutter.dev/docs>
- [12] React documentation, Retrieved from: <https://reactjs.org/docs/getting-started.html>
- [13] Ionic documentation, Retrieved from: <https://ionicframework.com/docs>
- [14] 2nd Quadrant, Retrieved from: <https://www.2ndquadrant.com/en/postgresql/postgresql-vs-mysql/>
- [15] InitLive, Retrieved from: <https://www.initlive.com/>