



SOFTWARE ONTWIKKELING  
EN PROFESSIONELE VAARDIGHEDEN

2DE BACHELORJAAR INFORMATICA

## Pukkelpop: Eindverslag

*Groep 6:*

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

*Begeleider*

Balazs Nemeth

*Coördinerend verantwoordelijke*  
prof. dr. Wim Lamotte

Jaar 2019-2020

# Inhoudsopgave

<b>1</b>	<b>Executive summary</b>	<b>2</b>
<b>2</b>	<b>Inleiding</b>	<b>3</b>
<b>3</b>	<b>Functionaliteit</b>	<b>4</b>
3.1	Server . . . . .	4
3.2	Sectorverantwoordelijke . . . . .	4
3.3	Vrijwilliger . . . . .	5
3.4	Admin . . . . .	5
<b>4</b>	<b>Problemen</b>	<b>7</b>
4.1	Feedback van de opdrachtgever . . . . .	7
4.2	IOS, Android, Windows en MacOS . . . . .	7
4.3	Offline vs. Online . . . . .	7
4.4	Realtime database . . . . .	8
4.5	3 applicaties . . . . .	8
4.6	Beste route . . . . .	9
4.7	Excel inlezen . . . . .	9
4.8	Data invoegen . . . . .	10
4.9	Security . . . . .	10
4.10	Duidelijke structuur van de applicatie . . . . .	10
4.11	Lijsten . . . . .	10
4.12	Server vs. Client-side data afhandelen (tijdzones) . . . . .	11
<b>5</b>	<b>Revisie van de analyse</b>	<b>11</b>
<b>6</b>	<b>Mogelijke uitbreidingen</b>	<b>12</b>
<b>7</b>	<b>Ontbrekende functionaliteit</b>	<b>13</b>

# 1 Executive summary

Het doel van ons project was om een opvolging applicatie te maken voor de vrijwilligers op Pukkelpop. Hierbij komend was een applicatie voor de sector-verantwoordelijken en een website voor de admin.

We hebben dan ook 3 verschillende applicaties gemaakt, waarvan 2 voor gebruik op gsm en 1 voor een computer. De mobiele applicaties kunnen zowel op Android als op IOS gebruikt worden. Om te beginnen eerst de algemene verdeling van functies en verbindingen tussen de 3 applicaties.

De eerste applicatie die we hebben is degene voor de vrijwilligers, in grote lijnen dient deze voor de locatie van de vrijwilligers te vinden. Verder toont deze nog extra informatie, zoals welke shift hij/zij nu moet werken.

De tweede applicatie is voor gebruik door de sector-verantwoordelijken. Deze kan gebruik maken van de applicatie om te kijken waar er problemen zijn. Ook loopt het verder op de huidige functies van de sector-verantwoordelijke, het controleren van de vrijwilligers.

De laatste applicatie is de admin applicatie. Deze is bedoeld voor de administratie, zodat deze een overzicht hebben van wat er zich allemaal afspeelt op het terrein. Hiervoor toont deze applicatie de positie van posten, problemen en vrijwilligers.

De verschillende applicaties zijn verbonden met een server en database. Dit zodat informatie uitgewisseld kan worden. Op deze server bevindt zich de informatie over de verschillende shifts, gebruikers, planning, kortom al de informatie met betrekking tot Pukkelpop. Deze wordt dan nog tijdens het festival aangevuld met werk gegevens onder andere: waar bevinden de vrijwilligers zich, zijn ze op hun post, zijn er ergens problemen. Deze informatie is dan ook zichtbaar voor de sector-verantwoordelijke en administratie, zijn kunnen op elk moment zien wat er zich afspeelt op het terrein.

Nu volgt een beschrijving van de 3 applicaties in meer detail.

De eerste applicatie die er is, is degene voor de vrijwilliger. Deze persoon werkt op het veld en gebruikt de applicatie niet vaak actief. De applicatie heeft dan ook niet heel veel functies.

De belangrijkste is de tracking van de locatie van de gsm en gebruiker. Dit gebeurt continu, wanneer hij aan het werken is. Ook als de gsm van de gebruiker in slaap modes staat.

De app biedt de vrijwilliger de mogelijkheid om de informatie van zijn huidige en opkomende shift te zien. Zo weet hij of zij waar en wanneer hij op post moet zijn.

Als laatste feature heeft de vrijwilliger nog de mogelijkheid om berichten te sturen en problemen te melden.

De 2de applicatie is degene voor de sector-verantwoordelijke. Deze dient als een soort mini administrator die rondloopt op het terrein en de vrijwilligers inspecteert. Dit vertaalt zich dan in de applicatie.

De sector-verantwoordelijke krijgt een kaart te zien met daarop al de posten en problemen die er zijn in zijn sector. Hij kan desnoods ook andere sectoren zien.

Zoals het nu op Pukkelpop is, gaat de sector-verantwoordelijke rond op het terrein om de aanwezigheden van de vrijwilligers aan te duiden. Hiervoor heeft hij een overzicht van de verschillende posten en kan hij daarop aanduiden of een vrijwilliger aanwezig is of niet en of of de verschillende voorwerpen, zoals zaklampen aanwezig zijn.

De verantwoordelijke heeft ook de mogelijkheid tot het sturen en ontvangen van berichten. Deze kan dit naar meer personen dan de vrijwilliger. Terwijl deze enkel de sector-verantwoordelijke kan contacteren, kan de verantwoordelijke zowel de andere verantwoordelijke als de admin als de vrijwilligers onder hem contacteren.

Hij krijgt ook de problemen die zich voordoen in zijn sector te zien, zodat hij kan rondgaan om ze op te lossen.

Als laatste hebben we dan nog de applicatie voor de administratie. Zoals eerder aangehaald dient deze voor een algemeen overzicht van de gebeurtenissen op het festival. Het hoofd deel van de admin applicatie is een pagina die we 'Overview' genoemd hebben. Zoals de naam doet ver-

moeden biedt deze een overzicht van wat er allemaal aan het gebeuren is. Dit ten eerste door het aanbieden van een kaart, daarover meer in de volgende alinea. Wat kan de admin hier nog zien? Hij ziet een oplijsting van al de onopgeloste problemen. Onder deze problemen vallen vrijwilligers die niet op post zijn en andere problemen gemeld door de sector-verantwoordelijken. Voor het overzicht te behouden worden niet al de problemen in 1 keer ingeladen, dit zouden er veel kunnen worden. Dus is er een knop om er meer in te laden. Ook heeft de admin hier een overzicht van al de verschillende posten met bij elke post de shiften waarop er gewerkt wordt. Als laatste is er hier nog een communicatie systeem voorzien, al de berichten verstuurd naar de administratie komen hier binnen en er is de mogelijkheid om zelf een bericht te versturen. Dit kan rechtstreeks naar iedereen, al de vrijwilligers, al de sector-verantwoordelijken, een specifieke post of naar een specifieke persoon. Ook zijn hier een aantal standaard berichten voorzien, die volgens ons vaak verstuurd zullen worden.

De admin heeft ook de mogelijkheid om een grote, zo goed als fullscreen, kaart te zien. Hierop staan al de posten aangeduid, samen met de locaties van de vrijwilligers en sector-verantwoordelijken. Als er zich een probleem voordoet op een post wordt enkel het probleem getoond op de locatie, dit om overlap te vermijden. Indien er zich meerdere problemen op dezelfde locatie bevinden, worden deze samen gezet onder de zelfde aanduiding op de map met de mogelijkheid om details te zien van een specifiek probleem.

Verder zijn er nog enkele specifieke pagina's met informatie over 1 categorie. Er is een oplijsting van al de problemen, een oplijsting van al de vrijwilligers en sector-verantwoordelijken een oplijsting van al de posten en één van de planning. Hierop is meer specifieke informatie zichtbaar en kan er gefilterd worden, bijvoorbeeld om enkel de posten in een bepaalde sector te laten zien.

Als laatste heeft de admin nog toegang tot de rapportering functie en toegang tot het inlezen van gegevens. De rapportering functie creëert een pdf met daarin een opsomming van al de problemen die tijdens het festival gebeurt zijn. De gegevens inlezen gaat via enkele Excel files, na het uploaden worden de gegevens hieruit in de database toegevoegd. Deze Excel files moeten voldoen aan bepaalde voorwaarden die ernaast vermeld staan.

## 2 Inleiding

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, die de locatie van de vrijwilligers vergelijkt met de posten en zo nakijkt of hij/zij aanwezig is op zijn/haar post. De tracering van de vrijwilligers zou dan naar de controlerende bevoegdheden doorgestuurd worden, zodat zij ook meer gericht kunnen handelen wanneer er zich problemen voordoen.

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, terwijl de sector-verantwoordelijke de applicatie eerder actief zal 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 en het invoegen van de data.

## 3 Functionaliteit

### 3.1 Server

De server is de plaats waar alle clients communiceren met de database. Wij hebben de volledige back-end opgedeeld in verschillende secties. Elke database tabel bestaat uit 4 zaken: een model, een controller, een router en een seeder. Deze 4 zaken zullen we voor de eenvoudigheid een sectie noemen. Elke sectie die wij geïmplementeerd hebben bevat de standaard CRUD functies. Elke sectie bevat buiten de CRUD functies ook nog specifiekere/geavanceerdere functies, deze zorgen ervoor dat er in de front-end met alle eenvoud ontwikkeld kan worden.

Buiten de functionaliteit die de server aan de front-end applicatie biedt heeft de server ook nog interne functionaliteit. Wij hebben ervoor gezorgd dat alle routes volledig beveiligd zijn en enkel aanspreekbaar zijn wanneer de client in bezet is van een authentication token. Deze token word in meer detail uitgelegd bij de 'Security' sectie. Ook kan de server excel files inlezen en deze importeren in onze database.

### 3.2 Sectorverantwoordelijke

Het doel van de sectorverantwoordelijke is om de vrijwilligers te controleren en bij te sturen waar nodig. De volgende features zijn allemaal geïmplementeerd:

- Algemeen
  - User-feedback : dialogs / toast waar nodig.
- Appel-lijst en posten weergeven
  - Appel-lijst weergeven
    - \* Posten weergeven via een digitale kaart
    - \* Posten weergeven via een lijst
    - \* Posten met problemen visueel zichtbaar maken (kaart)
    - \* Posten met problemen visueel zichtbaar maken (lijst)
    - \* Filteren op sector bij de digitale kaart
    - \* Filteren op sector bij de lijst
    - \* Sorteren van de lijst op: alfabetisch, beste route en kortste afstand
    - \* Kleurtjes toegevoegd om de sectoren te onderscheiden
    - \* De default sector wordt automatisch geladen
  - Posten weergeven
    - \* Algemene shift info per post
    - \* De huidige shift wordt automatisch weergegeven, waardoor men enkel moet navigeren naar een bepaalde shift, indien er iets nagekeken moet worden.
    - \* Per post de vorige en volgende shiften bekijken, hierbij zijn er meerdere opties toegevoegd, waarbij men zowel kan sliden, via een knop of via sneltoets alle shiften kan bekijken en vervolgens de juiste selecteren.
    - \* Checklist om de items aan/uit te checken, deze staan standaard op aangecheckt om tijd te besparen voor de gebruiker
    - \* Problemen log, waar men eventueel een extra probleem kan toevoegen, een probleem kan markeren als afgehandeld en foutieve problemen kan verwijderen.
    - \* Mogelijkheid om bij post en/of medewerkers nota's te maken (een probleem melden wordt gezien als een nota maken)
- Contact
  - Zichtbare gegevens of een soort sneltoets om de vrijwilliger/administrator te contacteren (bellen / email).

- Meldingen
  - Meldingen kunnen worden ontvangen, waarbij er onderscheid gemaakt wordt tussen de problemen van de eigen sector, en notificaties om de gebruiker gewoon te verwittigen over iets.
  - Berichten en problemen kunnen ook worden aangemaakt en verzonden worden.
  - Meldingen kunnen worden gemarkeerd als gelezen
  - Er is ook een algemeen overzicht van het totaal aantal meldingen, om een beter beeld te hebben van de meldingen

### 3.3 Vrijwilliger

Het doel van de vrijwilliger is om bepaalde gegevens te verzamelen en door te sturen naar de leidinggevende. Daarnaast zijn er nog subdoelen, zoals bijvoorbeeld de mogelijkheid geven dat de vrijwilliger de nodige informatie kan zien en eventueel ook de sector-verantwoordelijke kan bereiken indien er zich problemen voordoen.

- Aanwezigheid
  - Opvragen locatie om de x minuten (achtergrond + voorgrond).
  - Controleren of hij of zij op post is
  - Tracering locatie, een kwartier voor de shift begint t.e.m de shift eindigt. (Is geïmplementeerd via een startknop)
- Meldingen
  - Automatisch
    - \* Melding indien de gebruiker zich niet binnen de post-radius bevindt, tijdens zijn shift.
  - Handmatig
    - \* Melding dat de shift binnen x minuten begint.
    - \* 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.
    - \* Algemene meldingen (shift verandering, globale informatie, ...)
    - \* Mogelijkheid om bericht te versturen naar zijn/haar sectorverantwoordelijke
- Authenticatie
  - Login beveiligd op de gsm-nummer
- Info
  - Info wordt weergegeven, indien de shift nog niet is gestart
  - Info wordt weergegeven tijdens de shift
  - Een map naar de locatie

### 3.4 Admin

Het doel van de admin is om een overzicht van het gehele festival te hebben. het merendeel van de functionaliteit van de admin website dient dan ook om dit algemeen overzicht te geven.

- Overview pagina

Deze pagina geeft een overzicht van al de zaken die de admin moet kunnen zien, Hij kan hier een olijsting van de onopgeloste problemen zien, deze worden per 5 ingeladen om overzichtelijk te blijven. De admin kan via de knop 'Toon meer problemen' de volgende 5 laden.

Verder is er een olijsting van al de posten, met de verschillende shiften op deze post, al de berichten die naar de admin zijn verstuurd. Hij kan deze berichten aanduiden als gelezen, zodat ze niet meer te zien zijn.

Ook is hier een map zichtbaar, met daarop de locatie weergegeven van posten, problemen, vrijwilligers en verantwoordelijken.

Als laatste kan hij hier ook zelf berichten sturen. De admin heeft hier de keuze om zijn bericht naar iedereen, al de vrijwilligers, alle verantwoordelijken, een specifieke post of naar iemand specifiek te sturen. Als hij kiest om naar een post te sturen krijgt hij een olijsting te zien van de beschikbare posten. Ook kan hij hier gebruik maken van vooraf bepaalde standaard berichten.

- Map

Deze pagina dezelfde map als te zien is in de overview pagina, enkel is neemt deze de volledige pagina in voor wanneer de admin een beter overzicht wilt.

- Rapportering

Dit genereerd de rapportering van de gebeurde problemen tijdens het festival. Dit is een pdf die al de problemen, gesorteerd per vrijwilliger bevat. Deze pdf wordt geopend in een nieuw browser tablat. In deze pdf zullen al de problemen zichtbaar zijn, met daarbij de shift waarop hij plaatsvond met het specifieke uur, ook staat erbij vermeld of het probleem opgelost was of niet.

- Data pagina

Deze pagina dient voor het indienen van de gegevens via Excel. De admin moet hier 5 verschillende bestanden indienen en uploaden naar de server. Ook is er de mogelijkheid om de huidige gegevens uit de database te verwijderen. Bij het uploaden van een file wordt deze verwerkt. De nuttige informatie wordt er uit gehaald en omgezet in Json, waarna deze Json wordt doorgestuurd naar de server. Op de server wordt deze dan aan de database toegevoegd.

- Overzicht pagina's

Ook zijn er enkele specifieke overzicht pagina's voorzien voor problemen, vrijwilligers en verantwoordelijken, posten en de planning. Hier kan men een olijsting zien van al de problemen of al de posten bijvoorbeeld. De admin kan op deze pagina's filteren indien hij op zoek is naar een specifiek item.

Bij de posten kan hij filteren de naam van de post of om enkel de posten uit een specifieke sector te laten zien.

Bij de vrijwilligers en verantwoordelijken kan de admin filteren op voornaam, achternaam, of het een vrijwilliger is of een verantwoordelijke, op de vereniging en of dat hij/zij een internet verbinding heeft.

Bij het overzicht van de planning kan de gebruiker selecteren welke shift hij/zij wilt zien. Waarna daarna gefilterd kan worden op postnaam of op sector.

Bij de vrijwilligers en verantwoordelijken kan de admin aanduiden of de betrokken persoon een internet verbinding heeft of niet.

Bij het overzicht van de problemen is er ook de mogelijkheid om direct aan te duiden of het probleem opgelost is of niet.

- Specifieke pagina's

Voor zowel de problemen als de posten is er nog een specifieke pagina. Hierop ziet de gebruiker de details van deze post of het probleem. Deze pagina's zijn bereikbaar na het klikken op markers op de map, of als men op het probleem/post klikt in hun olijsting op de overzicht pagina's.

Het is ook hier dat men bij een probleem kan aanduiden of het opgelost is of niet.

## 4 Problemen

### 4.1 Feedback van de opdrachtgever

Wij hebben tijdens de tussentijdse presentatie redelijk veel feedback gekregen. Wij hebben deze feedback serieus genomen en al deze features nog geïmplementeerd. Daarnaast hebben wij ook nog een vergadering gehad met de werkelijke opdrachtgevers, en zij leken vrij tevreden over de applicatie.

### 4.2 iOS, Android, Windows en MacOS

Pukkelpop wil een applicatie die zowel voor iOS alsook op Android functioneert. Aangezien beide systemen anders functioneren, is het niet mogelijk om met 1 taal een applicatie te laten draaien op beide operating systems. Daarnaast kwam nog het extra probleem dat apple een gesloten systeem is, en niemand van ons in het bezit was van apple-producten. Wij zijn dus op zoek moeten gaan naar een oplossing hiervoor.

Een mogelijke oplossing zou zijn geweest om voor zowel android als iOS apart te coderen (in respectievelijk Android-studio en Swift), maar aangezien dit te veel tijd in beslag zou nemen en we daarnaast niet kunnen coderen in swift, omdat we niet in het bezit zijn van MacOS, was deze optie al gauw aan de kant geschoven. Een andere oplossing, waarvoor we uiteindelijk ook voor hebben gekozen, is door gebruik te maken van frameworks die toelaten om met 1 algemene taal te programmeren, waarbij het framework zelf dan vervolgens de code omzet voor de specifieke OS.

Zo kwamen wij al gauw uit bij Flutter, een open-source UI software development kit van Google. Echter waren er 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 doen naar alternatieven zoals Adobe PhoneGap, Apache Cordova, Vue, Apache Cordova, enzovoort. Uiteindelijk kwamen we uit op Ionic. Na 2 dagen werkte de achtergrond locatie met Ionic React en het was veel gebruiksvriendelijker dan Flutter of een andere concurrent. Ook bood de Ionic background location library ons de beste precisie en opties om te configureren.

Het eerste deelprobleem was dus al reeds opgelost hiermee, maar toen kwam het tweede deelprobleem: om de applicatie te compilen naar iOS, moet men in het bezit zijn van MacOS, en deze OS draait enkel op apple-hardware. Wanneer men op zoek gaat naar de mogelijkheid om MacOS te runnen via een virtual machine, komt men erachter dat dit blijkbaar niet zo makkelijk is. Ten eerste heb je de OS zelf nodig (bv. macOS Catalina), maar deze kan enkel gedownload en omgezet worden naar een ander windows/linux-vriendelijk formaat op een apple zelf. Ten tweede is macOS blijkbaar afhankelijk van bepaalde hardware. Dus afhankelijk van of je net macOS-compatibele hardware hebt, zal de OS kunnen runnen in de virtual machine.

Een virtual machine was dus blijkbaar niet de oplossing. Helaas willen mensen meestal ook niet hun laptop / computer voor een paar dagen uitleenen, dus deze optie viel eigenlijk ook weg, maar bleef wel in ons achterhoofd. Een Mac kopen voor enkel dit project is tot slot ook niet echt een oplossing. Uiteindelijk zijn we terechtgekomen bij MacInCloud. Dit is een service die toelaat om via de cloud een Mac te runnen op een OS zoals bv. linux of windows. De kosten zijn hiervoor ongeveer 20 euro gemiddeld per week/maand, afhankelijk van het abonnement, maar de eerste 24u is een gratis testperiode. Wij hebben deze proefperiode gebruikt om te kijken of de belofte van Ionic (dat het ook compiled voor iOS) waar is. Onze applicatie runde na dezelfde stappen die we moeten doen om het op Android te compilen, ook op een Iphone simulator, waarbij er een paar details anders waren, maar in het algemeen werkte zoals we wilde.

### 4.3 Offline vs. Online

Het internet werkt niet altijd even goed. Op sommige plaatsen kan de mobiele data zelfs zo slecht zijn, dat de applicatie niet met de server kan verbonden worden. Daarom is het van belang dat



deze app ook offline kan draaien. Eens er terug netwerk is, moeten de verrichte acties dan wel doorgegeven worden aan de database.

Om dit te verwezelijken zijn we op zoek gegaan naar offline mogelijkheden. Aangezien onze data behandeld wordt via redux, hebben we ook gezocht onder 'offline redux'. We kwamen hierbij uit op 'redux-persist'. Deze library was vrij vervelend en echt controle hadden we er ook niet over. Dus hebben we verder gezocht naar alternatieven, waardoor we uiteindelijk uitkwamen op de standaard-library: localStorage van ionic react zelf. Het voordeel bij deze library is dat we zelf de controle hebben over wat en hoe de data lokaal opgeslagen kan worden. Wat het voor ons dus ook mogelijk maakt om bijvoorbeeld de data te hashen.

Omdat de applicatie voornamelijk online werkt, maar ook in offline-status moet kunnen werken, hebben we ervoor gekozen de zichtbare data altijd via de localStorage te laten zien. Hierdoor moeten we maar op 1 locatie aangeven of we offline of online zijn, en moeten we voor de rest in de applicatie hier weinig van aantrekken, behalve indien men nieuwe data naar de database wil versturen.

Dit brengt op zich wel enkele nadelen met zich mee: het opvragen van de data is geconcentreerd in 1 functie, terwijl anders het opvragen van de data eerder apart gebeurde. Voor de vrijwilliger is dit geen probleem, aangezien de applicatie van de vrijwilliger maar beperkte data moet opvragen, maar voor de sector-verantwoordelijke kan dit wel problemen opleveren, aangezien de sector-verantwoordelijke de mogelijkheid moet hebben om niet alleen zijn eigen sector te bekijken, maar ook die van de anderen. Met andere woorden: de sector-verantwoordelijke vraagt bijna alle data op van de database.

#### 4.4 Realtime database

Een belangrijke ontwerpbeslissing in de applicatie voor pukkelpop, is de keuze qua database. Tijdens de eerste kennismakings-vergadering werd ons duidelijk dat de snelheid van de datawisseling niet prioritair was. De applicatie moest vooral dienen om de rapportering online te maken, en niet via papieren zoals ze in het verleden deden. Hierdoor hebben wij besloten om niet voor een Realtime Database te gaan zoals bijvoorbeeld Firebase van google, maar eerder voor een passievere database.

#### 4.5 3 applicaties

Een moeilijkheid bij het maken van de applicatie voor pukkelpop, is dat er 3 verschillende applicaties nodig zijn, met elk een anders publiek, die onderling goed met elkaar moeten communiceren, en die ook nog eens op alle operating systems moeten kunnen draaien (IOS, Android, Windows, ...).

Het probleem met drie aparte applicaties, waarvan twee voor de smartphone en 1 voor op de computer, is dat men specifieke problemen heeft per applicatie. Zo moet men bij de mobiele applicaties er bijvoorbeeld rekening mee houden dat de locatie aangevraagd moet worden als permissie aan de smartphone, waarbij er daarnaast ook nog eens verschillen zijn tussen IOS en Android.

Een ander puntje dat anders is, is bijvoorbeeld de layout. De website mag groot, meerdere schermen op 1 scherm bevatten, terwijl de informatie van de mobiele applicaties eerder compact moeten weergegeven worden. Een structuur die mooi past voor een mobiele applicatie, is niet peré een goede structuur voor een website, enzovoort.

Tot slot zijn bepaalde features per applicatie specifiek. Zo zijn er libraries speciaal voor mobiele applicaties geschreven, bijvoorbeeld ionic react, maar voor de website bestaan er alternatieven die meer gericht zijn op het maken van een website, bijvoorbeeld Material-ui. Hierdoor is er meer leerstof te verwerken, dan wanneer er maar 1 applicatie gemaakt moest worden.

## 4.6 Beste route

Een functie die wat meer aandacht vraagt is het implementeren van een "beste route" optie. Het is namelijk zo dat indien men echt de allerbeste route wil calculeren, men alle mogelijke paden moet afgaan, wat qua tijdscomplexiteit exponentiele tijd vraagt. Indien we het probleem vertalen naar een meer algemeen probleem, staat dit probleem beter bekend als het "Traveling Salesman Problem". Hiervan is bewezen dat dit NP-compleet is, of met andere woorden: er is tot vandaag de dag nog geen efficiënt algoritme voor gevonden.

Aangezien een exponentiele tijdscomplexiteit zeer traag is, en indien de lijst te groot is, dit zelfs kan oplopen tot jaren wachten, wordt het algoritme meestal wat lossier geïmplementeerd. Men heeft dan misschien niet de garantie dat het de allerbeste route is, maar wel de garantie dat het een degelijke route, die in een aangename tijd te berekenen valt, ook wel heuristischeken genaamd.

Bij het zoeken naar een gepaste heuristiek, zijn er verschillende mogelijkheden. Sommige algoritmen zijn al reeds bekend onder de computerwetenschappers, zoals bijvoorbeeld het 'greedy' algoritme, of het 'Nearest Neighbour' algoritme. Wij hebben gekozen om zowel het 'greedy' alsook 'nearest neighbour' te implementeren, waarbij we dan vervolgens de maximale afstand met elkaar vergelijken om dan uiteindelijk de beste van de twee als gesorteerde lijst te nemen.

De twee algoritmes even samen geven:

Greedy

- Maak bogen van de locaties.
- Sorteer deze bogen van klein naar groot.
- Selecteer de kortste boog en voeg het aan de lijst toe indien:
  - Het geen cirkel vormt, tenzij de route al reeds N bogen bevat.
  - Het niet verbonden is met meer dan 2 bogen.

Nearest Neighbour

- Neem een post(1) en vind de dichtstbijzijnde post(2) tot deze post(1)
- Indien deze nieuwe post nog niet is toegevoegd aan de lijst, voeg deze dan toe.
- Maak nu van post(2), post(1) en herhaal de vorige stappen, tot men een lijst met N knopen heeft.

## 4.7 Excel inlezen

Een eerste beslissing die we moesten maken was of we de Excels naar de server gingen sturen of, dat we de parsing aan de client-side zouden doen. Hierbij trad echter een probleem op waardoor onze keuze snel gemaakt was, als wij de volledige Excel file doorstuurden naar de server kwam deze niet aan. We hebben niet gevonden wat hierbij het probleem was en hebben dus gekozen om eerst de Excel te parsen op de website zelf en de data in Json formaat door te sturen.

Een ander probleem dat we natuurlijk hadden was hoe we de correcte data gingen halen uit de gegeven Excel files. Deze waren niet altijd goed gestructureerd om door een computer uit gelezen te worden. Er was ook veel repetitie van gegevens en bepaalde gegevens die nog ontbraken. We hebben er dan ook voor gekozen om een paar aanpassingen te doen aan het gegeven formaat. Dit hebben we ook aangehaald bij de opdrachtgever en zij waren hier mee oké.

Zo moet Pukkelpop nu bijvoorbeeld een gekend adres ingeven, indien ze dit willen gebruiken. Het is niet mogelijk om de locatie van adressen zoals SKJ of Parking 1 te vinden. Een ander voorbeeld is dat ze nu bij hun opsomming van de planning de voor- en achternaam plus telefoon nummer moeten ingeven gescheiden door een spatie.

## 4.8 Data invoegen

Nadat we de Excel files geparsed hadden, traden er echter nog problemen op.

We maken gebruik van een database object, dat al de connecties met de server regelt. Voor het parsen van de Excel maken we ook gebruik van een eigen class. Nu trad er een probleem op wanneer we de gegevens van het parsing object via de database class wouden doorsturen naar de server. Hoewel de gegevens correct aankwamen in het database object kwamen ze niet aan op de server. De oplossing die we hiervoor hebben gebruikt is om rechtstreeks vanuit de parser class verbinding te maken met de server.

Een volgend probleem met het versturen van de gegevens naar de server is dat er een limiet staat op hoe groot de gegevens in 1 post request mogen zijn. We konden natuurlijk deze limiet verhogen, maar dan waren we nooit honderd procent zeker dat er geen te grote file zou worden gebruikt. Dus hebben we besloten om de gegevens in verschillende requests door te sturen.

Dit probleem trad enkel op bij het invoegen van de planning. Deze dataset is vele malen groter dan de andere bestanden. We hebben dan ook de keuze gemaakt om het enkel bij dit te implementeren, om het aantal request beperkt te houden.

## 4.9 Security

Voor de beveiliging tussen de client-side en de server te verzekeren maken wij gebruik van RFC 7519 internet standaard. Deze staat beter bekend als JWT of voluit JSON Web Tokens. Voor bijna elke operatie van client-side naar de server-side moet er een JWT meegegeven worden. Wanneer dit niet gebeurt krijg je geen toegang tot de CRUD en geavanceerde operaties van de server. Indien je wel een JWT meegeeft, via de Authorization header, checkt de server of de token nog niet vervallen is en of dat de token nog geldig is qua handtekening (signature). Deze geldigheid wordt gecheckt door een RS256 public en private key, indien de token is veranderd door de client-side gebruiker zal dit gedetecteerd worden en krijgt de gebruiker geen toegang tot de operaties van de server.

## 4.10 Duidelijke structuur van de applicatie

De bedoeling van de applicatie is om door een bepaalde groep mensen 3-4 dagen intensief gebruikt te worden. Dit betekent dat de applicatie niet te ingewikkeld mag zijn, aangezien het gebruik van de applicatie maar van korte duur is. Daarnaast is er nog een verschil tussen het gebruik van de 3 applicaties: de sector-verantwoordelijke zal de applicatie actief gaan gebruiken, terwijl de vrijwilliger eerder passief. Dit heeft als gevolg dat de vrijwilliger best niet te veel opties heeft om te doen, terwijl de sector-verantwoordelijke dit juist wel moet hebben. Veel opties heeft echter als gevolg dat het overzicht wel eens verloren kan gaan.

We hebben dit opgelost door de structuur van algemeen naar specifieker te laten gaan. Het algemene overzicht is bijvoorbeeld beschikbaar door op de bottom-navigation naar een bepaald scherm te gaan, waarbij men dan in dit scherm al een overzicht heeft. Om een bepaald detail dan specifieker te bekijken, kan men dan op dat item klikken.

Ondanks dat het overzicht niet verloren mag gaan, moet de applicatie echter ook nog efficiënt werken, waarbij de gebruiker niet te veel moet doen om zijn/haar doel te voltooien. Een voorbeeld bij de sector-verantwoordelijke is het feit dat hij van post naar post moet gaan en moet controleren of alles in orde is. Dit betekent dus een vrij repetitieve actie. Aangezien deze actie waarschijnlijk het meest gebruikt zal worden, moet deze dan ook zo efficiënt mogelijk gemaakt worden. Wij hebben dit opgelost door een lijst met posten te hebben, waarbij de sector-verantwoordelijke enkel op de juiste post moet klikken, waarna hij gewoon de informatie binnen deze postview moet afgaan, om klaar te zijn met de desbetreffende post.

## 4.11 Lijsten

De applicatie is grotendeels opgebouwd uit lijsten: appellijsten, contactlijsten, enzovoort. Een gevolg van lijsten is dat deze te groot kunnen worden, waardoor de content van de applicatie niet goed meer kan laden. Om dit om te lossen hebben wij gewerkt met "Meer laden-knoppen". De

eerste keer dat een pagina geladen wordt, zijn er maar een paar elementen van de lijst beschikbaar, indien men dan op "meer laden" klikt, rendert de applicatie er 10 extra bij.

## 4.12 Server vs. Client-side data afhandelen (tijdzones)

De server en client-side maken gebruik van 2 verschillende tijdzones. Standaard maakt de server-side gebruik van de UTC tijdzone, deze loopt 2 uur achter op onze lokale tijd. Na veel blogpost 's te lezen, die natuurlijk up-to-date waren, kwam ik tot de conclusie dat de tijdzone van de server altijd in UTC vorm moet staan. Door deze techniek toe te passen kunnen wij onze applicatie flexibel uitbreiden voor internationale gebruikers.

## 5 Revisie van de analyse

Wanneer we het algemene proces en geïmplementeerde features vergelijken met het analyse verslag, dan zien we dat deze vrij nauwkeurig gevolgd is. Er zijn echter wel een paar puntjes anders verlopen. We sommen ze even op.

- De database is lichtjes veranderd, het nieuwe database schema vindt u onderaan dit document.
  - Bij al onze tabellen houden we 2 nieuwe variabelen bij, zijnde `updated_at` en `created_at`.
  - We hebben een extra tabel voor de sector aangemaakt. Deze vervangt de variabele die in ons analyse verslag in de post tabel stond. We hebben dit gedaan om bij te kunnen houden welke sector-verantwoordelijke bij welke sector hoort.
  - Ook bij de user hebben we een variabele vervangen door een gehele tabel. Hier de Association tabel. Dit zodat dezelfde association naam niet telkens herhaald moet worden bij elke vrijwilliger.
  - Enkele toevoegingen die we gedaan hebben, omdat we deze variabelen nodig hadden voor functionaliteit.
    - \* Bij de planning houden we nu ook bij of de gebruiker ingecheckt is of niet.
    - \* Bij item houden we nu bij of het verloren is of niet.
    - \* Bij Message houden we nu bij of het bericht gezien is of niet.
- Features
  - In het analyse verslag werd er vermeld dat we de gsm op nacht modus zouden kunnen zetten, dit is echter niet geïmplementeerd, omdat dit op zich geen extreem belangrijke feature is. Een feature die bijvoorbeeld wel als extra kan gezien worden, is bijvoorbeeld de mogelijkheid om alle shiften van de huidige post te bekijken en in 1 klik naar deze shift te navigeren. Een ander voorbeeld van een feature die oorspronkelijk niet in het verslag stond, is de mogelijkheid om zowel te sorteren op beste route, alsook op kortste afstand.
- Libraries
  - De genoemde libraries in het analyse verslag zijn grotendeels ook effectief gebruikt geworden. Sommige libraries, zoals bijvoorbeeld de "Call Number-library van Ionic react, zijn echter tijdens het implementeren veranderd naar een beter alternatief. Daarnaast zijn er ook enkele extra libraries geïmplementeerd geworden, zoals bijvoorbeeld 'localStorage' van ionic react.
  - We hebben de overstap gedaan van Google-maps naar OpenStreetMap via Leaflet, deze verandering was omdat het gebruik van de Google-maps API niet gratis is in productie en OpenStreetMap wel. Terwijl ze beide dezelfde functionaliteit aanbieden.
  - Om dezelfde reden hebben we ook geen gebruik meer gemaakt van Google directions. Dit hebben niet vervangend door een library, maar een externe servis, online ArcGIS geocoding service. Deze servis is gratis wanneer men enkel de opgevraagde coördinaten toont op de map.

- De bibliotheek Excel-to-json gebruiken we ook niet. Dit omdat deze geen beschikbaar type definities had voor het gebruik in Typescript. We hebben nu gebruik gemaakt van XLSX.
- Mockups
  - De mockups zijn ook vrij nauwkeurig opgevolgd. Een opvallende verandering is de 'bottom-navigation', waarbij er 1 component minder in is verwerkt. Dit is gedaan, omdat de layout op verschillende toestellen moet werken, waarbij 5 elementen in de 'bottom-navigation' op bepaalde toestellen problemen gaf. Het voordeel is dat men hierdoor ook een extra lijst heeft kunnen toevoegen: namelijk een lijst waarin alle problemen van de sector-verantwoordelijke bij elkaar zijn gezet.
- Planning
  - Iets wat niet nauwkeurig is opgevolgd geweest van de analyse, is de planning. Dit komt doordat tijdens de implementatie bepaalde zaken afhankelijk waren van elkaar, waardoor bepaalde functionaliteiten nog niet geïmplementeerd konden worden.

## 6 Mogelijke uitbreidingen

Enkele mogelijke uitbreidingen, die de gebruikers ervaring verhogen, zijn:

- De mogelijkheid om veranderingen in shift door te brengen. Onder andere het veranderen van welke vrijwilliger op een bepaald moment op een gegeven post staat. Dit zouden we implementeren door bij het overzicht van de planning, op de admin website, de admin de keuze te geven om shiften te wijzigen. We denken hier aan een input veld dat de namen van vrijwilligers auto complete. Deze veranderingen dan ook echt doorvoeren zo via een patch request naar de server, waarna de database een update query zou uitvoeren.
- Een andere uitbreiding zou zijn om opmerkingen te schrijven bij gemelde problemen, zodat Pukkelpop na het festival nog bepaalde extra info heeft over de gemelde problemen. Om dit toe te voegen moeten we in onze database natuurlijk deze extra informatie bijhouden. Waarna we deze kunnen toevoegen bij het aanmaken van een probleem of de admin en de sector-verantwoordelijke de mogelijkheid geven deze gegevens aan te passen.
- Gps-functionaliteit. Op dit moment is de kaart eerder een overzicht van de posten, maar heeft het verder niet veel functionaliteit. Een mogelijke uitbreiding zou zijn om deze kaart meer gps-functionaliteiten te geven: bijvoorbeeld een route-planner toevoegen, en de eigen locatie.
- Bij het versturen van berichten door de admin naar specifieke vrijwilligers of verantwoorde-lijke een auto complete functionaliteit te implementeren. Zo moet de admin niet de namen van de ontvanger onthouden.
- Geavanceerdere log-in functionaliteit zodat personen zelf hun log-in beter kunnen beheren. Wanneer ze hun wachtwoord vergeten zijn moet de admin dit nu voor hun aanpassen. Wij zouden nog graag een e-mail gebaseerde 'Wachtwoord vergeten' functionaliteit toevoegt hebben. Echter was dit geen requirement van de opdrachtgever en lagen onze prioriteiten op andere features. Ook zouden wij voor het admin paneel een log-in creëren zodat zij a.d.h.v email makkelijk en dynamisch kunnen inloggen.
- In sommige delen code hebben wij gebruik gemaakt van redelijk grote functies die niet overeenkomen met de kwaliteit van de code die wij willen neerzetten. Wij zouden nog graag extra tijd gehad hebben om deze stukken code te herbekijken en aan te passen.
- Het settings menu dat we vermeld hadden in ons analyse verslag hebben we niet meer geïmplementeerd. Dit had enkel de optie om te check-up-tijd van de gebruikers locatie

te veranderen. Deze feature was geen prioriteit, merendeels aangezien hier geen volledige vrijheid in is. IOS laat maar een minimum toe van 15 minuten tussen 2 check-ups. Indien we nog extra tijd hadden, was deze feature misschien wel geïmplementeerd.

## 7 Ontbrekende functionaliteit

- Standaard sector voor sector-verantwoordelijken

We voorzien voor elke sector-verantwoordelijke een standaard sector, dit is de sector waar hij of zij verantwoordelijk voor is. Wat niet geïmplementeerd is, is het invoegen van deze data. Indien we dit implementeren stoten we op een probleem met de opbouw van onze database. Nu houden we bij elke sector bij welke de verantwoordelijke hiervan is. Dit laat echter niet toe om meerdere verantwoordelijke aan dezelfde sector te koppelen, dus we kunnen hiervan geen shiften maken. Dit zal echter wel het geval zijn in een festival, waar 1 persoon niet continu werkt.

Om deze feature te implementeren moeten we dus structuur van onze database iets aanpassen. Een nieuwe tabel met verantwoordelijke shiften, waarin de verantwoordelijke op een specifiek tijdstip verbonden wordt met een sector. Deze veranderingen zouden dan ook doorgevoerd moeten worden naar de sector-verantwoordelijke applicatie om de standaard sector op te halen. Als laatste moet deze gegevens dan ook ingeladen worden en zal daarom het inladen van de Excel nog aangepast moeten worden.



