Stuffinder

Zoeken. Organiseren. Beheren.

Altijd je spullen bij de hand,
daarvoor staat Stuffinder garant

Thunder Ducks
DION SCHEPER – AMANDA HEERES – KIRSTEN KINGMA – LOTTE FEKKES

Voorwoord

Het bedenken van een app is een uitdagende taak. Het kan alles worden. Een spel is meestal het eerste wat naar boven schiet. Toch ga je wel iets langer nadenken en dan kom je bij dingen die dicht bij jezelf staan. Een app kan namelijk meer dan alleen voor amusement zorgen, hij kan ook dagelijkse problemen oplossen. Toen wij hierover aan het brainstormen waren hadden we echter een klein probleem. Amanda had haar laptop thuis laten liggen. Waardoor ze niets kon opzoeken. Toen kwamen we op een idee, want hoe vaak is ons dat zelf niet overkomen? Je laat je spullen liggen bij je ouderlijk huis of studentenkamer, bij je vader of bij je moeder. Of sterker nog: Je weet überhaupt niet meer waar je spullen zijn omdat je een aantal keer heen en weer hebt gereisd achter elkaar. Daarom hebben wij de app 'Stuffinder' bedacht. Om het leven van ons, en iedereen die onze app gaat gebruiken, makkelijker te maken.

Nadat wij hadden bedacht welke app wij wilden ontwikkelen, moest dit idee natuurlijk uitgewerkt worden. Dit is natuurlijk een heel proces van leren samenwerken, programmeren en uiteindelijk een goede app neerzetten. In dit verslag wordt dat proces nader toegelicht. Eerst gaan we uiteenzetten wat onze app nou precies moet kunnen doen en welke functies wij aan onze app hebben toegevoegd om de gewenste eigenschappen van de app te kunnen verkrijgen. Daarna gaan we dieper in op hoe we dit hebben aangepakt en wat er allemaal nodig is om de bedachte functionaliteiten te laten werken. Om dit verslag daarna goed af te sluiten gaan we terug kijken naar wat er goed ging, wat beter kon, waar problemen ontstonden en natuurlijk: wat wij allemaal hebben geleerd tijdens het ontwikkelen van 'Stuffinder'.

Inhoud

Beschrijving

Inleiding

De Stuffinder app is een aantal keer herzien tijdens het ontwikkelingsproces. Maar het uitgangspunt is hetzelfde gebleven. Er is gekeken naar de maatschappelijke behoefte om spullen te organiseren. Bij het ontwerpen van Stuffinder is er gekozen voor een drieslag aan functionaliteit: zoeken, organiseren en beheren.

1. In de app moet je spullen kunnen *zoeken* (Waar ligt mijn teddybeer?)
2. In de app moet je spullen kunnen *organiseren* op een logische wijze (Het 'verplaatsen' van spullen)
3. In de app moet je alle spullen afzonderlijk in een overzicht kunnen *beheren*.

AFBEELDING 1: DIT IS DE ACTION BAR MET DE KNOPPEN/FUNCTIONALITEITEN VAN DE APP. ZOEKEN (VERGROOTGLAS), ORGANISEREN (KOFFERTJE), BEHEREN (BLAADJE).

Naast deze drie eigenschappen van de app zijn er basisfunctionaliteiten toegevoegd:

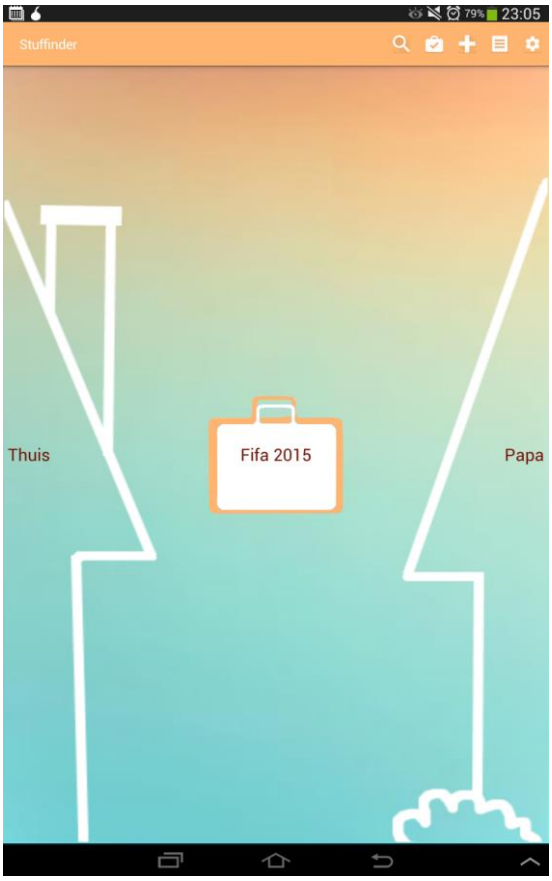
- Wijzigen van een item (EditItem)
- Toevoegen van een item (NewItem)
- Locatie wijzigen (Settings)
- Help overzicht met uitleg (Help)

Belangrijkste eigenschappen

AFBEELDING 2: HET ZOEKSCHERM, ALS BELANGRIJKSTE SCHERM GEKOZEN ALS OPENINGSSCHERM VAN DE APP. DE GEBRUIKER KAN EENVOUDIG ZOEKEN OP NAAM, ZOALS IN DE SCREENSHOT TE ZIEN IS MAAKT HEEFT DE APP AUTO-COMPLETE OM DE GEBRUIKER EEN HANDJE TE HELPEN.

Zoeken

Het openingsscherm van de app staat hier links. Dit is het zoekscherm en hier kan de gebruiker zoeken op een naam maar ook zoeken op 'locatie 1' of 'locatie 2'. Deze zijn als knop beschikbaar maar ook als zoekterm.



Organiseren

Dit is de functionaliteit die het meest is uit gediept. Hier links is het 'beheer' interface te zien.

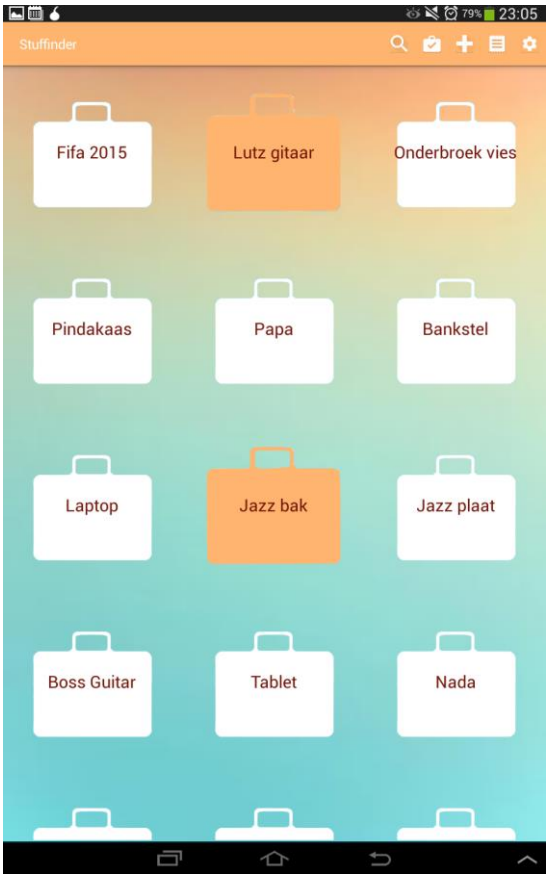
De koffers zijn te swipen over het scherm en kunnen ofwel links ofwel rechts van het scherm 'verdwijnen'.

De koffers hebben al de kleur van waar ze op dit moment zijn, dit geeft de gebruiker extra feedback op waar hij mogelijk heen zou willen swipen.

AFBEELDING 3: IN DIT VOORBEELD IS TE ZIEN DAT DE GEBRUIKER ER VOOR KAN KIEZEN OM 'FIFA 2015' BIJ THUIS OF BIJ PAPA TE PLAATSEN.

Beheren

Deze functionaliteit geeft de gebruiker meer controle over de items in zijn bezit. Met dit handige overzicht kan hij snel zien welke items waar zijn en eventueel wat aanpassen.



AFBEELDING 4: HET BEHEER SCHERM, HIER IS TE ZIEN DAT DE GEBRUIKER DE ZELF TOEGEVOEGDE ITEMS KAN WIJZIGEN. OOK KAN HIJ AAN DE KLEUR VAN DE KOFFER ZIEN WAAR DIE ITEM ZICH BEVINDT.

Productverantwoording

Het product Stuffinder is nog niet op de Android markt te verkrijgen. Wel is er een vergelijkbare, en gelijknamige, app in ontwikkeling. Deze is te vinden op http://stuffinder-app.com/. Op deze website staat echter nog geen concrete info over hoe de app precies gaat werken en oordelend aan de opmaak en screenshots die er al wel zijn heeft het weinig met deze app te maken.

9292 is één van de inspiraties geweest bij het ontwerpen van de Stuffinder app. Vooral omdat deze ov app een intuïtieve bediening heeft. Deze bediening is dan ook zorgvuldig geanalyseerd en mede deze analyse heeft er toe geleid dat het 'zoek'-scherm nu het openingsscherm is van de Stuffinder app.

De gebruiker wil iets, wat wil hij? Dat is daarbij een belangrijke overweging geweest. De 9292-ov app speelt hier handig op in. De gebruiker zal vooral de app openen om een reis te plannen, daarmee is de openingsvraag: "waar wil je heen?" dus heel logisch gekozen. Zo'n zelfde soort aanpak zit in Stuffinder verwerkt. Alleen hier is de vraag: "wat zoek je?" of "wat ben je kwijt?".

Bij het nadenken over een handig organiseer scherm is er inspiratie gehaald uit de Tinder-app. Deze 'dating'-app brengt mensen in contact door middel van een foto keuring. De gebruiker kan deze foto naar links slepen als de foto hem/haar aanspreekt en als dat niet zo is swiped hij/zij deze foto eenvoudig naar rechts.

De Stuffinder app gebruikt deze gedachtegang ook. De gebruiker kan elke koffer met een item slepen naar links of naar rechts. Deze links en rechts krijgen dan een door de gebruiker gekozen naam. Standaard is dit 'Mama' en 'Papa'. De gebruiker kan dus spullen tussen huizen verslepen.

Al met al is er dus al een Stuffinder app ergens anders in de wereld in ontwikkeling. Echter bevindt deze zich nog niet in de app-store en vergelijkbare apps bieden niet hetzelfde soort functionaliteit als dat deze Stuffinder app doet.

Deze Stuffinder app laat daarnaast een nieuwe toepassing van touch technologie zien. Aangezien het typen steeds meer uit raakt en touch populairder wordt zal de gebruiker meer touch ervaringen willen hebben. De Stuffinder speelt hier op in met het 'Organiseer' scherm.

AFBEELDING 5,6,7,8: VAN LINKS NAAR RECHTS HET 9292 LOGO, TINDER LOGO, STUFFINDER-APP LOGO VAN EXTERNEN EN HET LOGO DAT IS GEMAAKT VOOR DEZE STUFFINDER APP.

Specificaties

Functionele eigenschappen:

1. **Item toevoegen**
   De gebruiker kan een item eenvoudig toevoegen door op het plusje in de action bar te drukken. Deze toont dan een nieuw scherm waar de gebruiker eenvoudig een naam kan intypen en een locatie kan selecteren. Daarna drukt hij op de knop 'Toevoegen' en met behulp van een melding zal hij op de hoogte worden gebracht dat deze is toegevoegd.
2. **Item zoeken**
   Aangezien de gebruiker nog al eens spullen kwijt lijkt te zijn kan hij met het openingsscherm meteen zoeken door een naam in te tikken. Dit scherm is ook te bereiken door op het vergrootglas icoon te drukken op de actionbar.
3. **Item wijzigen**
   Door in het 'beheer'-scherm ofwel het 'organiseer'-scherm op een koffer te drukken opent zich een nieuw scherm met de mogelijkheid om een item te wijzigen. De gebruiker kan dit vooral gebruiken om een naam aan te passen. Maar ook een individuele item aan te passen kan hiermee handig zijn.
4. **'Verhuizen'**
   De grootste functionaliteit wordt in de use-case 'verhuizen' genoemd. Dit komt omdat dit scherm (bij het klikken op het koffer icoon) vooral handig is bij het verhuizen. Als de gebruiker van zijn moeder naar vader vertrekt kan hij dit gebruiken om snel alle items bij langs te gaan en te controleren of hij deze bij mama wil laten of mee neemt naar de vader.
5. **Controleren**
   Het beheer scherm bevat alles om de verschillende items te kunnen controleren. Hier is een duidelijk overzicht te vinden van alle items. De gebruiker zal hier minder tijd door brengen maar het is een onmisbare functionaliteit. Zonder deze mogelijkheid zouden veel processen zich onzichtbaar voor de gebruiker afspelen.
6. **Alles verwijderen**
   Mocht de gebruiker nou privacy bewust zijn kan hij hier op klikken om alle data te verwijderen uit het programma. Door daarna het programma te sluiten zal de lege lijst opgeslagen worden.
7. **Help bekijken**
   De gebruiker kan ook door op het tandwiel icoon te drukken en daarna naar de help knop de navigeren een help scherm openen. Hier staan alle iconen uitgelegd met hun betekenis en extra toelichting over hoe de gebruiker het beste deze kan gebruiken.

Niet functionele eigenschappen

1. Opslaan van de lijst
2. Makkelijke navigatie (gebruiksgemak)

Verhuizen (uitwerking use-case):

De gebruiker gebruikt de app om bij te houden waar zijn spullen zijn. Hierbij wordt aangenomen dat de gebruiker tussen de 16 en 20 jaar oud is en elke week pendelt tussen zijn gescheiden ouders. De gebruiker heeft alle items die hij heeft al toegevoegd aan de Stuffinder app.

Wanneer de gebruiker nu van zijn vader naar zijn moeder vertrekt opent hij de Stuffinder app. Hij navigeert eenvoudig naar het 'organiseer'-scherm. Hij ziet op het eerste koffertje 'FIFA 2015' staan, en hij herkent aan de kleur dat deze bij papa licht. Hij denkt even kort na en herinnert zich dat hij maandag met de jongens bij zijn moeder FIFA wou spelen. Snel stopt hij het computerspel in zijn tas en swiped de koffer naar 'mama'.

Het volgende koffertje dat daar achter verschijnt bevat 'onderbroeken'. Hij telt snel zijn onderbroeken en hij heeft genoeg om mee te nemen naar zijn moeder. Snel naar 'mama' swipen dus. De volgende koffertjes kan hij zelf dus snel oordelen of hij dat al heeft ingepakt, of hij het überhaupt mee wil. Daarna swiped hij eenvoudig naar links of rechts afhankelijk van zijn keuze.

Als hij het laatste koffertje heeft geswiped geeft de app de melding dat hij 'klaar' is! Hij kan nu vertrekken met een gerust hart naar zijn moeder.

Zoeken (uitwerking use-case):

Voor het zoeken bestaan meerdere scenario's:

1. Zoeken op naam
2. Zoeken op locatie

1: Gebruiker zoekt 'teddybeer'
De gebruiker is een student en zit op zijn kamer. Hij vraagt zich af of hij zijn teddybeer nou wel of niet heeft meegenomen want uit de bende van zijn kamer wordt hij dat niet wijzer.

Bij het openen van de Stuffinder app krijgt hij meteen de vraag: "wat zoek je?". Door in het zoekveld teddybeer in te typen verschijnt tijdens het typen al het woord 'teddybeer' als zoeksuggestie. Hij klikt hier op en begint met zoeken. Binnen een tel heeft hij zoekresultaat in een overzicht, hieruit ziet hij dat hij hem inderdaad heeft meegenomen naar zijn kamer wat betekent dat hij kan beginnen met zoeken.

2: Gebruiker vraagt zich af wat hij ook al weer bij 'papa' had laten liggen.
De gebruiker is een studente op kamers met enkel haar vader als ouder. Ze vraagt zich af hoeveel kleren ze ook al weer bij haar vader had laten liggen.

Bij het openen van de Stuffinder app is er een knop 'Papa'. Door op deze knop te drukken kan ze meteen naar het overzicht gaan met alleen de items die bij papa liggen. Door dit overzicht door te scrollen kan ze snel een beeld vormen van wat ze ook al weer had laten liggen.

Ontwerp

Globaal ontwerp

Stuffinder is opgedeeld in verschillende componenten die allemaal een verschillend doel hebben. Toch hebben ze wel veel met elkaar te maken. De componenten bestaan voornamelijk uit de verschillende functies die de app heeft:

- Het zoeken van een item
- Het organiseren
- Het beheren van items
- Items toevoegen
- Items wijzigen
- De naam van de locaties wijzigen
- Een help functie

Eigenlijk alle functies komen samen in de itemlist. Hier worden alle items in opgeslagen en alle functies hebben hier mee te maken. Het zoeken van een bepaald item, het wijzigen van een item of een item toevoegen wordt allemaal in of met de itemlist gedaan. Alleen de hulpfunctie heeft geen connecties met de andere functies behalve door de tekst die erin staat. Deze legt uit wat alle knoppen doen en dus wat de andere functies doen. Doordat deze lijst overal gebruikt word en overal dezelfde lijst is, kunnen alle componenten samen werken en zijn alle aanpassingen meteen van toepassing. Als de naam van een item gewijzigd is, is dit meteen te zien in de lijst bij het organiseren en als de locatie aangepast wordt, is dit bij het swipen ook te zien.

Detailontwerp

We hadden als eerste bedacht dat we alles met een MVC aanpak wilden implementeren. Dit is uiteindelijk niet meer helemaal naar voren gekomen, vooral omdat we gebruik hebben gemaakt van fragmenten. Bijna elk fragment (wat dus de view voorstelt samen met de xml) heeft een controller waar de methodes in staan die de functionaliteit maakt voor het fragment. Oftewel wat de functie moet doen, zoals het zoeken. Verder staat in de fragmenten welke buttons naar wat moeten luisteren en worden er zo ook methodes aangeroepen. Deze methodes staan meestal in de controller die bij dit fragment hoort. We zullen niet alle fragmenten bespreken, want in alle fragmenten staan ongeveer dezelfde methodes zoals onCreate en onCreateView die aangeroepen worden als het fragment aangemaakt wordt.

Zo hebben we als eerste de belangrijkste klasses van ons product: de item en itemList klasse. Hier wordt alles in opgeslagen en hier zijn alle methodes op gebaseerd. Zonder deze klasses zou het hele product niet werken. Een item heeft als attributen een naam en een locatie. Deze locatie wordt meteen bij het aanmaken van een item gegeven en kan dus ook niet leeg zijn.

De klasse ItemList heeft als attribuut een arraylist van Items en een string voor de naam van de file waar de lijst in opgeslagen gaat worden. Deze klasse heeft vooral methodes om de lijst op te slaan en te laden vanaf het mobiele toestel waar de app op staan. Andere methodes zijn er om de lijst aan te passen door items toe te voegen, te verwijderen of uit de lijst te halen (Bijvoorbeeld voor het zoeken).

Voor het zoeken hebben we de klasse SearchController en SearchFragment, waar in het fragment alle knoppen beschreven worden en gezegd wordt wat er moet gebeuren als het fragment aangemaakt word. In de controller zitten methodes die een item zoeken aan de hand van een

meegegeven string of een gekozen locatie. Hiervoor wordt een lijst aangemaakt die dan weergegeven kan worden.

Bij het organiseren van alle items doormiddel van het swipen hebben we gebruik gemaakt van een library[1] waarin een eenvoudige implementatie stond van het swipen naar links en rechts van blokjes en daar dan iets kunnen doen als het naar links is of als het naar rechts is. In de klasse SortController hebben we deze library dus geïmplementeerd. We moesten echter nog wel dingen aanpassen. Dat waren voornamelijk twee dingen: de opmaak en de adapter. Wij willen namelijk geen string maar een object meegeven. Ook kon het er wel wat vrolijker en meer in overeenstemming met het design van de rest van de app uitzien.

Voor het beheren hebben we verschillende klasses voor de verschillende functies.
Voor het toevoegen van een item is er het fragment NewItemFragment. We hebben hier niet apart een controller bij gemaakt omdat er maar twee methodes geïmplementeerd hoefden te worden en deze waren duidelijker te maken als deze in het fragment stonden.
Dit geldt ook voor het wijzigen van een item in het EditItemFragment. Alle functies staan in de onClick methode omdat deze hele functionaliteit gebaseerd is op input van de user en er verder geen ingewikkelde dingen hoeven te gebeuren.
Om alle items goed weer te geven zodat alles overzichtelijk is hebben we ook alleen een fragment gebruikt: GridListFragment. Hierin worden alle items weergegeven in een gridlist, zodat alle items netjes naast en onder elkaar komen te staan.

Om de namen van de locaties aan te passen hebben we een SettingsController en Fragment. In het settingsmenu zit ook een button om het HelpFragment te openen. In de SettingsController zitten methodes om je setting op te slaan zodat elke keer als je de app opent je je eigen instellingen krijgt en je dit niet steeds opnieuw hoeft in te stellen. De andere methodes halen de locatienamen op of veranderen de namen in de nieuw ingegeven namen.

Alle fragmenten worden geladen in de MainActivity. Hierin wordt de actionbar geladen die de hele tijd in beeld blijft, waardoor er gemakkelijk genavigeerd kan worden door de app. Doordat er maar één activity is, kan de itemList heel makkelijk doorgegeven worden aan de fragments zonder al te veel moeite.

Ontwerpverantwoording

9292 indeling qua menu -> eigen onderzoek (proefpersonen zijn mensen uit de omgeving) wees uit dat dat makkelijker in gebruik is

We hebben qua indeling en navigatie van onze app goed gekeken naar andere apps en naar wat mensen fijn vinden. We hebben aan mensen gevraagd wat ze een makkelijk te gebruiken app vonden en waar ook makkelijk in te navigeren was. Hierbij kwamen we uit op de 9292ov app. Dit is een populaire app die mensen erg makkelijk in het gebruik vinden. Als je deze app opstart kom je meteen bij het hoofddoel van de app: je reis plannen. Dit idee paste goed bij onze app en we hebben dus besloten om dit principe te gebruiken in onze app. Als Stuffinder opgestart word kom je meteen op het zoekscherm terecht. De gebruiker kan dan snel iets opzoeken wat hij kwijt is.

Een andere mogelijke designkeuze was om de actionbar onderaan of bovenaan het scherm te plaatsen. We hebben uiteindelijk gekozen om deze bovenaan te plaatsen, omdat onderaan ons niet

[1] https://github.com/Diolor/Swipecards

erg gebruiksvriendelijk leek. Er bestaat dan namelijk de kans dat een gebruiker vaak per ongeluk ergens op klikt omdat hij zijn telefoon vasthoudt en dan met zijn hand op een knop komt. Ook zijn gebruikers het gewend dat het menu bovenaan zit, omdat dit bij heel veel apps zo is en dit dus natuurlijker aanvoelt dan een menu onderin het scherm.

Reflectie

Toen we aan dit project begonnen hadden wij er alle vier erg zin in. Het klinkt natuurlijk ook goed: je eigen app bouwen. Toch bleek het allemaal niet rozengeur en maneschijn. Het in elkaar zetten van een plan waarin staat hoe we het gaan aanpakken ging goed. We hebben duidelijke afspraken gemaakt en een helder idee beschreven naar elkaar van wat we willen bereiken met deze app. Helaas was het uitvoeren van deze plannen wat lastiger. Het eerste probleem kwam toen we wilden werken met gitlab. Dat ging niet even soepel omdat we niet goed wisten hoe git werkt. Na een paar uurtjes uitproberen en om hulp vragen hebben we git echter toch goed laten werken. Het was erg belangrijk voor ons dat we met git konden werken want daardoor werd het doorsturen van de app enorm versoepeld. Daarna kwam de basis van de app: We moesten het model dat we hebben opgesteld implementeren. De klasses en functies werden aangemaakt, maar wat hadden we sommige dingen toch verkeerd ingeschat. Wij hebben alle vier geen ervaring met programmeren in android studio (behalve de kleine puzzel app) en de errors die we kregen waren vaak onbegrijpelijk. We hadden met een aantal dingen echt problemen:

- De taakbalk interface was erg lastig te implementeren. Android had namelijk bepaalde regels opgesteld voor de taakbalk waar we tijdens ons ontwerp niet rekening mee hebben gehouden. Ook moest de lijst met spullen telkens worden doorgegeven aan de andere activities, wat niet goed lukte. Hier hebben we erg lang mee vast gezeten maar uiteindelijk hebben we de android development internetpagina en verschillende forums helemaal uitgezocht en zijn we op een oplossing gekomen: Fragments. De balk bovenin zit een activity en wat telkens verandert is alleen een fragment. De gegevens zitten dus in de overkoepelende activity.
- Het opslaan van de instellingen en de lijst met spullen op de telefoon van de gebruiker. Wij hadden zoals is gezegd geen ervaring met het programmeren van apps, dus hadden geen idee hoe we dat moesten aanpakken. Ook hier hebben we weer het internet afgestruind en zijn we ook hier op deze manier op een oplossing gekomen. We maken een filestream aan die de gegevens laad bij het maken van een fragment en de gegevens opslaat wanneer het fragment wordt afgebroken. Zo zijn de gegevens altijd up-to-date. Helaas ging dit niet goed. Want het werkte maar niet: de gegevens konden niet geladen en opgeslagen worden. Het probleem lag bij de gegevens. Deze waren niet parseerbaar en konden dus helemaal niet opgeslagen worden. Dit hebben we ook opgelost dus nu werkt het opslaan en laden helemaal naar onze wens.
- Hoe we het swipen van de koffers naar links en naar rechts moeten implementeren. Gelukkig hebben we hiervoor een library konden vinden op het internet. Deze hebben we natuurlijk zelf aangepast naar onze app. Toch was het wel enorm prettig om dat te kunnen gebruiken, aangezien we op die manier heel veel tijd konden besparen aan onderzoek hoe dat werkt en veel meer aan de algemene functionaliteiten van de app.

Natuurlijk hebben we nog wel meer problemen gehad, maar deze hebben we even uitgelicht omdat dit de meest opvallende problemen waren en een goed voorbeeld zijn voor hoe we de rest van de problemen hebben opgelost.

Daarna komt natuurlijk het design: hoe ziet het totale plaatje eruit, want dat is natuurlijk erg bepalend voor een (toekomstige) gebruiker. Hier zijn we dan ook erg serieus mee bezig geweest en alhoewel er nog steeds dingen mooier kunnen zijn we erg tevreden over hoe dit overleg ging en wat het resultaat hiervan ging.

We zijn er dus achter gekomen dat een goed plan opstellen en goede communicatie van gegevens en wie waar mee bezig is erg belangrijk is. Door gitlab te gebruiken hebben we dit ook kunnen verwezenlijken. Ook hebben we geleerd flexibel te zijn, soms gaan dingen namelijk nu eenmaal niet zoals we hadden verwacht en dan moet er een plan B worden ingezet. Dit is ook een goede eigenschap op de werkvloer: werkgevers hebben weinig aan werknemers die niet met tegenslagen of veranderingen om kunnen gaan. We hebben onszelf ook beter leren kennen. Met zo'n groot project wordt het wel duidelijk waar je interesses en goede (en slechte) eigenschappen zitten. Dit heeft ons allemaal ook dichter bij een conclusie over ons werkgebied gebracht, namelijk wat we later nog veel meer willen doen.

Wat wij, de Thunder Ducks, nu uit dit project kunnen concluderen is dat we, als we allemaal goed communiceren en ons goed inzetten, een fantastisch team zijn. Iedereen heeft zijn of haar mindere momenten en zwakkere punten, maar als we elkaar aanvullen, komen daar fantastische resultaten uit. Stuffinder is hier natuurlijk een prachtig voorbeeld van. Want:

"Altijd je spullen bij de hand, daarvoor staat Stuffinder garant!"