

=DELIVIT

Thuis, wat en wanneer je wilt.



Deneubourg Mohamed Yassine

Multimedia & Communicatietechnologie

Erasmushogeschool Brussel | 2019 - 2020

Promotor: Johan Van Den Broek

BACHELORPROEF – FINAL WORK



Voorwoord

Deze **bachelorproef** (final work) is gemaakt met als doel het behalen van een **diploma Multimedia & Communicatietechnologie**, een opleiding aan de Erasmushogeschool in Brussel. Sinds september 2017 volg ik hier lessen en in de tussentijd ben ik enorm geëvolueerd.

Ik heb altijd mijn examens in eerste zittijd afgelegd en zo een (bijna) perfect traject gevolgda, hoewel ik Franstalig ben.

Ik beëindig mijn opleiding met een groot project: **DelivIt**, een mobiele applicatie die ik ook op de markt zal plaatsen. In September 2020 zal het gelanceerd worden op de Google Play Store en de Apple Store, indien dit juridisch mogelijk is.

Doorheen deze bachelorproef zal de naam "**DelivIt**" gebruikt worden als naam voor mijn applicatie en merk. De naam is afkomstig van de Engelse termen "Deliver" en "It". Dit betekent: "bezorg het".

Deze opdracht is erg belangrijk om af te studeren. Daarom wil ik graag volgende personen bedanken: eerst en vooral mijn promotor en IOS-professor, meneer **Johan van den Broek** voor de goedkeuring, de begeleiding van dit project en de waardevolle tips om tot een kwalitatief product te bekomen. Daarnaast wil ik ook **Eva-Nina Kunze** bedanken voor haar design-tips tijdens mijn eerste evaluatie-moment.

De docenten **Peter Dickx** en **Mike Derycke** wil ik ook bedanken omdat zij mij de mogelijkheid hebben aangeboden om tijdens de les Mobile App & Web de Flutter & Firebase zelfstandig te kunnen aanleren.

Ik wil ook **John-Nathan Michielsens** bedanken voor het advies, de testing en de hulp tijdens het ontwikkelen van mijn applicatie.

Obinna Odenigbo, Ibrahim El Ouakili, Ahmida Naimi en Ayoub Maskoub bedank ik voor hun geloof in mijn project en competenties.

Uiteindelijk bedank ik mijn verloofde **Samia Ait Slimane** en **mijn familie** voor de user testing van mijn applicatie en de dagelijkse motivatie, ook tijdens moeilijke tijden zoals de Coronapandemie.

Inhoudstafel

ONDERZOEK	6
Onderzoeksplan	7
Inleiding	7
Centrale onderzoeksvraag	7
Globale doel	7
Relevantie	7
Abstract	8
Marktanalyse	10
Concurrentieanalyse	11
Delhaize Home Delivery / My Delhaize App	11
Carrefour Home Delivery	12
Carrefour - ShipTo App	12
Uber	13
AirBnb	13
Doelgroepanalyse	14
Personas	15
Functionele-Analyse	16
MoSCoW-Analyse	20
Wat de applicatie zeker moet inhouden (Must)	20
Wat de applicatie gewenst moet inhouden (Should)	20
Wat de applicatie kan inhouden indien er nog tijd is (Could)	21
Wat de applicatie niet gaat inhouden (Won't)	21
Technische onderzoek	22
Flutter	23
React-Native	23
Kotlin	24
Keuze	24
ONTWERP	26
Huisstijl	27
Moodboard	28
Wireframes	29
Mockups	31

VERDIENSTRATEGIE.....	32
Business canvas	33
Verdienmodel	34
Opstartkosten	34
Maandelijkse kosten	34
Marge per verkochte eenheid	34
Enheden tot het break-even punt	34
Het break-even punt bereiken	34
Portefeuille – virtueel geld	35
ONTWIKKELING.....	36
Implementatie middelen	37
Framework Flutter	37
Visual Studio Code	38
XCODE – IOS (swift)	38
Android – Kotlin	38
GitHub	38
Trello	38
Firebase Authentication	39
Firebase Storage	39
Firebase Cloud Firestore	40
Node.js & Stripe	42
Softwarebibliotheeken en pakketten	44
Firebase	45
Flutter map & geolocator	45
Flutter local notifications	45
User testing	46
Moeilijkheden en oplossingen	47
Toekomstvisie	48
Besluit	49
Literatuurlijst	50



Onderzoeksplan

Inleiding

Vooraleer mijn onderzoeksvraag te definiëren, wil ik graag het probleem bespreken. In deze tijden hebben veel personen in België het moeilijk om hun boodschappen te doen, te kopen en tot hun woonplaats te brengen.

Dit gebeurt om verschillende redenen. Mensen zijn meestal bezig en hebben bijgevolg geen tijd. Anderen zijn niet mobiel, beschikken niet over een voertuig of hebben gezondheidsproblemen. Een andere groep mensen heeft gewoon geen zin om buiten te gaan. Dit is een probleem omdat het noodzakelijk is voor de mens om boodschappen te doen, de mens moet zich kunnen voeden.

In 2020 is het probleem nog groter geworden: wegens het coronavirus kon de Belg van maart 2020 tot en met mei 2020 niet altijd naar buiten gaan. Tijdens de lockdown werd men in België soms thuis opgesloten. Dit heeft een groot probleem gecreëerd in de voedingsmarkt.

Aangezien ik een technologische richting volg, wou ik dit probleem oplossen met technologie.

Centrale onderzoeksvraag

Hoe kan men met behulp van technologie ervoor zorgen dat mensen snel en veilig hun boodschappen betalen en ook thuis ontvangen?

Globale doel

Het doel is om te onderzoeken hoe ik mensen die met dit probleem geconfronteerd worden het leven kan vergemakkelijken. Een piste die ik voorstel is het ontwikkelen van een mobiele applicatie die de mogelijkheid biedt om boodschappen online te betalen en thuis te laten bezorgen. Daarnaast zou deze applicatie andere personen ook helpen om geld te verdienen.

Relevantie

Dit zal het leven vergemakkelijken van elke persoon in de doelgroep. Andere mensen kunnen daardoor ook geld verdienen, waardoor men dus een potentiële job gaat creëren.



Abstract

Vandaag zijn er steeds meer mensen die het moeilijk hebben om hun boodschappen te doen. Mensen hebben geen tijd, moeten afrekenen met mobiliteitsbeperkingen of hebben gewoon geen zin. Na onderzoek heb ik ontdekt dat dit probleem opgelost kan worden door technologie. Dit onderzoek ben ik begonnen in december 2019 en heb ik beëindigd in juni 2020 met een functionele oplossing (mobiele applicatie). Ondertussen zijn er allerlei elementen veranderd door sanitaire omstandigheden (Coronavirus).

Dit verslag is een onderdeel van mijn bachelorproef in de studierichting "Multimedia & Communicatietechnologie". Hieronder wordt heel het onderzoek, het proces en het eindresultaat uitgeschreven.

Tijdens mijn onderzoek, ben ik in januari tot de vaststelling gekomen dat een mobiele applicatie de beste oplossing is. Deze applicatie zal mensen helpen om hun boodschappen te doen en te bezorgen, en dit door andere personen. Dit is een community-gebaseerde applicatie waar iedereen aankoper of bezorger kan worden.

Er werd eerst onderzocht hoe de markt omtrent leveringsservices in elkaar zit, wie de concurrenten zijn en wie men precies wil bereiken en helpen.

Het proces van het ontwikkelen zelf zal ook aan bod komen, beginnend met het ontwerpen van de wireframes. Daarna gaan we de huisstijl bepalen en ook een logo ontwerpen. Ten slotte gaan we de technische aspecten onderzoeken met het doel een zo veilig en performant mogelijke applicatie te ontwikkelen.

Het concept van de applicatie is dat personen een aanvraag maken van producten die ter beschikking staan in een lijst (per categorie gefilterd). De bestelling-aanvraag gaat aan de bijliggende beschikbare bezorgers worden gestuurd. Deze bezorgers gaan een prijs en een leveringstijd aanbieden.

De aankoper ziet alle aanbiedingen van de bezorgers met een precieze prijs en tijd. Daarnaast kan hij het profiel van de bezorger bekijken, samen met de reviews van vorige bestellingen. De bezorger heeft ook een score: hoe hoger de score, hoe beter de service van de bezorger.

De aanbodprijs kan tot enkele euro's variëren, omdat elke bezorger zijn eigen prijslijst kan bepalen in de applicatie zelf. Hij kan daarmee zijn winst controleren of de kans dat hij gekozen wordt verhogen door de prijzen lager te zetten. De prijzen kunnen niet veel afwijken van de standaardprijs die is bepaald door een onderzoek op de markt.

Wanneer de aankoper een aanbieding accepteert, zal het geld gedebiteerd en geblokkeerd worden. De bezorger zal dan de producten aankopen en leveren. Hij wordt daarvoor door de applicatie geholpen dankzij een producten-checklijst, de routebeschrijving naar de klant, additionele informatie, de mogelijkheid om de klant te contacteren, ...

Na bevestiging van de twee partijen wordt het geld dan aan de bezorger gestort. De laatste stap is dat de klant een score en een commentaar aangeeft over de service van de bezorger. Deze zal waarneembaar zijn op de profiel-pagina van de bezorger.



Marktanalyse

We gaan ons concentreren op de levering van boodschappen.

Sinds enkele jaren is de levering van maaltijden sterk geëvolueerd en deze praktijk wordt meer en meer gebruikt. Uit een studie blijkt dat 44% van de Belgen minstens één keer per 2 maanden voedsel bestellen.

Niet minder dan 3 bedrijven bieden deze service in België. Deze zijn Deliveroo, UberEats, en TakeAway. Dit zijn redelijk grote bedrijven die met kleine en grote restaurants samenwerken.

Wat de levering van boodschappen betreft, is deze momenteel in expansie. Er worden testen uitgevoerd bij grote Belgische winkels. Dit gebeurt bijvoorbeeld bij Carrefour, Delhaize en binnenkort ook bij Colruyt. Tijdens de pandemie van het Coronavirus zijn ze intensiever gaan inzetten op de ontwikkeling van deze methode.

Een ander systeem is het verzamelen van artikelen door de winkel zelf, waarbij de klant daarna de goederen zelf moet komen ophalen (Collect&Go Colruyt).

Dit wordt meer en meer gebruikt en er zijn hypotheses die stellen dat in de toekomst de goederen enkel nog door deze methode of via levering verkocht zullen worden.

Na de pandemie van het coronavirus zijn de mensen nog meer op zoek geweest naar middelen om boodschappen gemakkelijk, veilig en goedkoop thuis te ontvangen. Zeker tijdens de quarantaine, toen het moeilijk was om naar buiten te gaan.

Concurrentieanalyse

Er bestaan geen **community-gebaseerde** applicaties of services die ervoor zorgen dat klanten hun boodschappen thuis ontvangen.

Maar er bestaan wel applicaties en webservices van grote winkels die wel een leveringsservice bieden. Daarmee kan men alleen de producten aankopen die in specifieke winkels beschikbaar zijn.

Ze bieden deze service, maar zijn niet community-based.

Delhaize Home Delivery / My Delhaize App

Dit is een service op een app of website om boodschappen thuis te laten bezorgen, deze boodschappen moet men bij Delhaize doen.

Voordelen:

- Men kan zijn eigen datum en tijdstip bepalen voor de levering
- Web-only promoties verkrijgbaar
- Grote hoeveelheid van boodschappen is mogelijk

Nadelen:

- Delhaize kan niet overal leveren, deze service is beschikbaar in bepaalde regio's
- Niet de dag zelf geleverd
- Men moet minimum €60 uitgeven om geleverd te zijn
- Levering mogelijk van maandag t/m zaterdag, tussen 7u en 21u (afhankelijk van de regio).
- Leveringskosten variëren tussen €7,95 en €9,95
- (Per schijf van 60 liter dranken wordt een meerkost van €10 in rekening gebracht.)



Carrefour Home Delivery

Dit is een service op de website van Carrefour om boodschappen thuis te laten bezorgen, deze boodschappen moet men bij Carrefour doen.

Voordelen:

- Men kan zijn eigen datum en tijdstip bepalen voor de levering
- Grote hoeveelheid van boodschappen is mogelijk
- Gratis levering vanaf een bestelling van min. € 150

Nadelen:

- Levering alleen in het noordelijk deel van het land, Vlaams-Brabant, Brussel en Waals-Brabant mogelijk
- € 9,50 (€ 5 verzendkosten + € 4,5 voorbereidingskosten)
- Niet de dag zelf geleverd
- Levering alleen 17u - 22u

Carrefour - ShipTo App

Dit is een service op de Applicatie **SHIPTO** van Carrefour om eenvoudige boodschappen thuis laten bezorgen, deze boodschappen moet men bij Carrefour doen.

Voordelen:

- Levering de dag zelf (in 1u30)
- Tot 80 kg boodschappen
- 100% ecolevering met elektrische bakfietsen
- €5 leveringskosten

Nadelen:

- Levering alleen in Brussel mogelijk
 - Een Carrefour-winkel moet in een straal van 4 km liggen
- Alleen beschikbaar voor boodschappen bij Carrefour
- Bestelling & levering alleen mogelijk binnen de openingsuren van de winkels
 - 9u - 19u
 - 8u30 - 18u30

Er bestaan wel **community-gebaseerde** applicaties die een gelijkaardige service aanbieden. Deze kan men ook beschouwen als **concurrenten**.

Uber

Dit is een service op een app/website die chauffeurs en passagiers verbindt om de passagiers van punt A tot punt B te brengen.

Voordelen:

- Men kan zijn eigen datum en tijdstip bepalen voor de rit
- Overal beschikbaar in 67 landen
- Iedereen kan chauffeur worden (18+, rijbewijs en vergunning)
- Chauffeurs rijden wanneer en waar ze willen
- Goedkoper dan taxi's
- Veilig en snel betalingssysteem
- Chauffeurs maken geld en passagiers besparen geld

Nadelen:

- Door het gebrek aan strengere regulatie is er toch een risico op een onplezierige of onveilige rit

AirBnb

Dit is een service op een app/website die mensen (meestal reizigers) verbindt om snel en veilig een kamer te huren bij een lokale verhuurder.

Voordelen:

- Men kan zijn eigen datum en tijdstip bepalen voor de
- In de hele wereld beschikbaar
- Iedereen kan kamers huren
- Verhuurders kunnen wanneer ze willen kamers beschikbaar stellen
- Goedkoper dan hotels
- Ratingsysteem om de beste verhuurders te kunnen kiezen
- Veilig en snel betalingssysteem
- Verhuurders verdienen geld en huurders besparen geld

Nadelen:

- Risico dat de kamer niet zoals getoond is of problemen heeft
- Risico op diefstal
- Risico op schade aan meubilair

Doelgroepanalyse



De gebruiker moet eerst en vooral de mogelijkheid hebben om een smartphone/tablet te gebruiken of deze beschikken. Uit studies blijkt dat 77% van de Belgen (8.81 miljoen) over een smartphone beschikken. Deze cijfers zijn in januari 2018 gemeten.

De gebruiker moet over een bankkaart beschikken om online te kunnen bestellen of betaald te worden.

De aankoop-functie van deze applicatie is gericht op mensen die

- Geen tijd hebben om boodschappen te doen
- Geen zin hebben om buiten te gaan
 - Door weersomstandigheden (Te koud buiten,...)
 - Door licht & tijdsomstandigheden (Te donker buiten,...)
- Over beperkte mobiliteit beschikken
 - Gehandicapten
 - Mensen zonder auto
 - Mensen die ziek zijn
 - Mensen in quarantaine

De bezorging-functie van deze applicatie is gericht op mensen die

- Geld (zelfstandig) willen verdienen
- Andere mensen willen helpen (Humanitair)

Deze personen moeten

- Min. 16 jaar en bekwaam zijn
- Bereid zijn om boodschappen te leveren aan de hand van hun eigen transportmiddel
- Een minimumaantal geld beschikken (om producten te kopen en daarna te verkopen)

Personas



Elodie Van Scheppe

Student - 22 jaar - op kot - Brussel

Elodie is een student in marketing. Ze heeft veel vrije tijd en gebruikt dit om te gaan sporten of met haar vrienden uit te gaan.

Zij bezit natuurlijk een smartphone en wilt wel centjes verdienen. Ze heeft geen schrik van fietsen en heeft ook haar eigen wagen.

Elodie kan vrij geïnteresseerd zijn om boodschappen te leveren en daarmee geld te verdienen.



Hassan Muhan

Entrepreneur - 27 jaar - Vilvoorde

Hassan heeft zijn eigen bedrijf en beheert een 30-tal werknemers. Hij is meestal op werk en wilt zijn resterende tijd besteden aan meer stimulerende activiteiten zoals voetballen, uitgaan, ... Zijn vrouw werkt ook full-time.

Hij vindt geen tijd om zijn boodschappen te doen en is geavanceerd in technologische kennis.

Hassan kan geïnteresseerd zijn om haar boodschappen online te doen en te laten leveren.



Patricia Dierko

Gepensioneerd - 82 jaar - Brussel

Patricia is gepensioneerd en was professor in een universiteit. Zij heeft mobiliteitsbeperkingen. Ze woont alleen in Brussel en haar kinderen zijn in het buitenland.

Zij heeft het moeilijk om haar boodschappen te doen en kan een smartphone gebruiken.

Patricia kan geïnteresseerd zijn om haar boodschappen online te doen en te laten leveren.



Functionele-Analyse

Het is noodzakelijk om alle functionaliteiten te kunnen opsommen om deze effectief in de development-fase te kunnen implementeren.

❖ Authenticatie - mobile & desktop

- Account aanmaken met gsm-nummer
 - Adres kunnen terugvinden via lokalisatie
- Inloggen met e-mailadres en wachtwoord
- Inloggen met gsm-nummer en wachtwoord
- Meldingen krijgen indien een fout gemaakt wordt
- Vergeten wachtwoord kunnen overkomen

❖ Na inlog kiezen tussen Bezorger of Aankoper - mobile

❖ Buttonbar menu - mobile

- Knop naar overzicht kaart
- Kaart naar overzicht bestellingen
 - Tabs met Bezorgd/Aanbiedingen/Aan het bezorgen
- Voor de aankoper knop om een bestelling te plaatsen

Hamburger-menu - mobile

- Profiel foto, naam en email adres
- Bezorger-modus of Aankoper-modus switchen
- Link Profiel pagina
- Link Portefeuille
- Link Berichten
- Link Hulpdienst
- Knop om uit te loggen

❖ Profielpagina - mobile

- Profiel foto
- Naam
- E-mailadres
- Globaler rating met sterren
- Commentaar van mensen
- Aantal bestellingen besteld & bezorgd

❖ Portefeuille - mobile

- Aantal geld waarnemen
- Overzicht van transacties
- Knop geld aanvragen
- Knop geld toevoegen (Van PayPal, bank, ... In de portefeuille)

❖ Hulpdienst - mobile

- Informatie gebruik applicatie
- Contact opnemen met de eigenaar van de App

❖ Overzicht kaart - mobile - Aankoper

- Interactieve kaart
 - Eigen positie kunnen zien
 - Nabije beschikbare bezorgers kunnen zien
 - Bezorgers aanklikken en informatie krijgen
 - Link naar profiel-pagina bezorger
 - Menu & Knop om nieuwe een bestelling te plaatsen

❖ Bestelling plaatsen - mobile - Aankoper

- Keuze categorieën
- Lijst met producten
 - Productnaam
 - Productfoto
 - Productprijs
- Mogelijkheid om product aan de bestelling toe te voegen
- Aantal per unit kiezen
- Totale prijs zien van de bestelling
- Datum en tijd kiezen voor de levering
- Leveringsadres kiezen
- Melding indien fouten
 - Niet voldoende geldsaldo
- Indien de bestelling genomen werd, wordt het geld van de aankoper gedebiteerd.

❖ Bestellingenlijst - mobile - Aankoper

- Per bestelling
 - Bestelnummer
 - Datum en tijd
 - Status van bestelling
 - Aanbieding gekregen, onderweg, geleverd, ...
- Bestellingen zijn aanklikbaar om naar detailpagina te gaan

❖ Bestellingdetail - mobile - Aankoper

- Lijst met alle producten
- Datum en tijd
- Leveringsadres
- Aantal tijd voor de bezorging
- Kaart met lokalisatie van de bezorger
- Bestelling-status
 - Aanbiedingen gekregen
 - Lijst met bezorgers met prijsaanbiedingen
 - Producten verzamelen
 - Checklijst van alle producten die al verzameld zijn
 - Onderweg
 - Informatie over de bezorger
 - Bezorger contacteren via SMS/Chat
 - Map met huidig positie van de bezorger
 - Knop wanneer bestelling bezorgd is
 - Geleverd
 - Informatie over de bezorger
 - Informatie over de producten

❖ Overzicht kaart - mobile - Bezorger

- Interactieve kaart
 - Eigen positie kunnen zien
 - Nabije beschikbare bestellingen zien
 - Bestellingen aanklikken en informatie krijgen
 - Link naar detailpagina van de bestelling

❖ Bestellingsprijs aanbieden - mobile - Bezorger

- Bestelling in detail zien
 - Lijst van artikelen
 - Aantal artikelen
 - Leveringsadres
 - Leveringsdatum & uur
- Knop op de bestelling te nemen
 - Dit stuurt een aanbieding (offerte) aan de aankoper met totale prijs en bezorgingstijd

❖ Bestelling bezorgen - mobile - Bezorger

- Bestelling in detail zien
- Aanklikken wanneer een artikel genomen is
 - Aankoper kan zo zien hoe zijn bestelling voortgaat
- Knop bevestiging wanneer alle artikelen verzameld zijn
- Routebeschrijving getoond op de kaart naar het leveringsadres
- Timer leveringstijd getoond
- Knop wanneer bestelling bezorgd is

❖ Bestelling is goed geleverd - mobile

- Aankoper moet bezorger quoteren
 - Ranking systeem met sterren
 - Commentaar toevoegen
- Aankoper kan een tip geven
 - Knoppen met aantal geld te geven
- Geld van de bestelling en van eventuele tip wordt gestuurd naar de bezorger

❖ Prijslijst producten - desktop - bezorger

- Producten per categorieën zien
- Elk product een bepaalde prijs aangeven (tussen een range)
- Mogelijkheid om de gemiddeld prijs zien van elk product

Dit waren de technische-functionele vereisten, ik heb ook een analyse gemaakt op het globale systeem. Dit is dus hoe het systeem moet werken.

❖ Gebruiksvriendelijk

- ❖ **Alleen met internetverbinding gebruikbaar**
- ❖ **Toestel moet een GPS hebben en de toegang accepteren**
- ❖ **Toestel moet de notificaties accepteren**



MoSCoW-Analyse

Voor het starten met het project moet men zeker keuzes maken en goed de prioriteiten definiëren om een MVP (Minimum Viable Product) te bekomen. Hieronder de structuur.

Wat de applicatie zeker moet inhouden (Must)

- Beveiligde gebruikersaccount
 - Gebruiker **moet** via SMS zijn inschrijving bevestigen
 - Profielpagina hebben en deze kunnen wijzigen
- Schermen en functies die verschillend zijn voor Aankopers/bezorgers
- De klant krijgt een lijst met alle beschikbare nabije bezorgers
- De klant kan een aanvraag maken aan nabije bezorgers voor bepaalde producten
- De bezorger kan een prijs aanvragen voor de volledige bestelling
- Betalingssysteem (Contant? Bank? Paypal?)
- Berichten tussen bezorgers en klanten in de app

Wat de applicatie gewenst moet inhouden (Should):

- Een interactieve map met de beschikbare nabije bezorgers
- Een al gedefinieerde lijst van producten met een prijs-suggestie voor bezorgers
 - Bezorgers kunnen hun eigen prijslijst maken
 - Beveiligde Admin-website/app om eigen prijslijst te maken
- Functionele & beveiligde betalingssysteem met bankkaart/Paypal-account/Google Pay/Apple Pay
- De klant ziet in real-time de status van zijn bestelling en de positie van de bezorger
- Beoordelingssysteem voor klanten om bezorgers te beoordelen

Wat de applicatie kan inhouden indien er nog tijd is (Could):

- Web-versie
- Bevestiging van levering via QR-Code tussen bezorger & klant
- Goedkeuringssysteem voor bezorgers door de administrator
- Gamificatie zodat bezorgers snel de bestellingen bezorgen

Wat de applicatie niet gaat inhouden (Won't):

- Methode om de betaling onmiddellijk te sturen (zodat de bezorger altijd kan betalen) en te terugbetalen indien een probleem ontstaat



Technische onderzoek

Met de functionele- en MoSCoW analyse kon ik een goed overzicht hebben over wat ik nodig heb om deze applicatie te ontwikkelen.

Plaats nu aan de manier waarop ik de applicatie technisch wil ontwikkelen.

Dankzij het vak “Mobile App & Web” te volgen had ik de mogelijkheid om al een onderzoek te maken. Ik wil zeker een cross-platform applicatie maken zowel voor IOS als voor Android. Daarom moet ik de beste en voornaamste frameworks/programmeertalen gaan onderzoeken en de meest compatibele kiezen.

	Flutter	React-Native	Kotlin
Maker	Google	Facebook	JetBrains
Programmeertaal	Dart	Javascript	Kotlin
Native	Ja, directe toegang tot het toestel (IOS & Android)	Ja maar niet voor alle API's	Ja
Cross-platform	Ja, IOS, ANDROID & WEB	Ja, IOS, ANDROID & WEB	Ja, IOS, ANDROID & WEB maar gaat verschillende code hebben.
Uitgebracht in	December 2018	Maart 2015	2011
Libraries/ Bibliotheken	Niet veel, vrij nieuwe framework	Zeer veel & grote libraries	Veel libraries

Dit zijn werd onderzocht in December 2019.

Flutter

Voordelen:

- Code schrijven die voor zoals Android als IOS zeer performant werkt
- Hot reload (Zeer vlug compileren en renderen van de modificaties op de applicatie) → Zo kan men snel en eenvoudig experimenteren, UI's bouwen, functies toe te voegen en bugs oplossen
- Dart is een gemakkelijke programmeertaal

Nadelen:

- Dit is een vrij nieuw taal dus de “community” is **momenteel** niet zeer groot
- Libraries moeten worden ingeladen en er zijn **momenteel** niet enorm veel libraries die Flutter ondersteunen.

React-Native

Voordelen:

- Code schrijven die voor zoals Android als IOS zeer performant werkt
- Hot reload (Zeer vlug compileren en renderen van de modificaties op de applicatie)
- Veel bibliotheken (libraries) beschikbaar die de ontwikkeling vergemakkelijkt
- Grote “community” beschikbaar die deze framework gebruiken en kennen, dus veel mensen die kunnen helpen

Nadelen:

- Moeilijk te leren
- Zeer traag om te herstarten dus traag om te ontwikkelen, bugs te regelen, functies toevoegen, ...



Kotlin

Voordelen:

- Gemakkelijke taal, minder code voor veel werk, efficiëntie teamwork
- Ondersteund door de meerderheid van de IDE's, Androidstudio helpt zeer veel
- Veel bibliotheken (libraries) beschikbaar die de ontwikkeling vergemakkelijkt
- Grote "community" beschikbaar die deze framework gebruiken en kennen, dus veel mensen die kunnen helpen

Nadelen:

- Compilatietijd is trager dan JAVA

Keuze

Uiteindelijk heb ik mijn keuze gemaakt. Ik zal verschillende elementen nodig hebben om mijn applicatie te ontwikkelen. Namelijk werken met Firebase, Maps — Localisatie, Betalingsmethodes, ...

Ik wil daarvoor een zeer efficiënte applicatie in term van performance (snelheid, geheugen, ...), een snelle development flow en zeker veiligheid.

Daarom heb ik voor Flutter gekozen. Ik vind dat dit een goede optie is, omdat het een nieuwe framework is van GOOGLE.

Dit heeft al in 1 jaar een groot succes; men merkt al meer dan 60 000 projecten op GitHub tegen 80 000 voor React-Native die 3 jaar ervoor uitgekomen is.

Flutter is nu één van de top 20 actieve frameworks op GitHub (13e plaats op 3 juni 2019), wat bewijst dat de developers het echt gebruiken en dus er vaak verbetering is.

Ik wou een framework die native,cross-platform en performant is, en Flutter voldoet aan deze aspecten.

Zoals vermeld zal **Firebase** in dit project gebruikt worden.

Firebase is een ontwikkelplatform van de onderneming Google.

Voor veiligheid zal ik de hele authenticatie-proces aan de Firebase-authenticatie module linken.

Als database zal ik de **Cloud Firestore** gebruiken van Firebase (Google), dit is een document-gebaseerde database die in real-time synchroniseert.

Aangezien ik met Flutter werk dat ook van Google is, gaat de samenwerking en ontwikkeling vlot gebeuren.

In de applicatie zijn er een aantal functies nodig, hieronder een lijst van **geteste** libraries die ik heb gebruikt voor het prototype.

Mapbox

Dit is een library die zich aan het platform Mapbox linkt om een kaart weer te geven. https://pub.dev/packages/flutter_map

Geolocator

Dit geeft de toegang tot de GPS van het toestel en geeft in real-time de positie van de gebruiker. <https://pub.dev/packages/geolocator>

Datetime picker

Dit is een library om op een mooie, snelle en efficiënte manier een specifiek datum en -tijd te kiezen.

https://pub.dev/packages/flutter_datetime_picker

Square payment

Dit is een library die een beveiligde link maakt met "Squareup" (<https://squareup.com/>). Een bedrijf dat betalingssystemen aanbiedt. https://pub.dev/packages/square_in_app_payments

Image picker

Een library om gemakkelijk afbeeldingen maken of kiezen in de galerij. https://pub.dev/packages/image_picker

Deze zijn allemaal getest in de **prototype-applicatie** die ik als **voorbereiding** ontwikkeld heb. Deze kan men in bijlage terugvinden.





Huisstijl



Example of a header

Montserrat - Black : #F3D511



#F3D511

Example of a subhead

Montserrat - Bold : #F7E710



#F7E710

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi at pulvinar eros, vitae accumsan ante. Nunc erat ligula, auctor et diam eu, porttitor efficitur ligula. Nulla pretium quam ac tellus tincidunt iaculis. Nam semper urna posuere purus posuere, a maximus nisl dapibus. Nullam non ipsum libero. Suspendisse sodales ac arcu sit amet ornare. Fusce sit amet condimentum dui, in ultrices felis. Sed consequat ipsum lacus, ac interdum est egestas vitae. Nam nec interdum est.

Montserrat - Regular : #000000

Button



#EFEFEF

Button



#CCCACA

Button



#717070

Button



Moodboard



Wireframes

Kies een functie
Dit kan je ook later in je profielpagina wijzigen.

Bezorger
Jij gaat geld maken door de boodschappen op tijd te leveren.

Aankoper
Jij wilt je allerlei boodschappen laten leveren.

OVERZIJKT MAP

Bezorger : Beschikbaar ★★★☆☆

PRODUCT BESTELLEN

BESTELLING 2

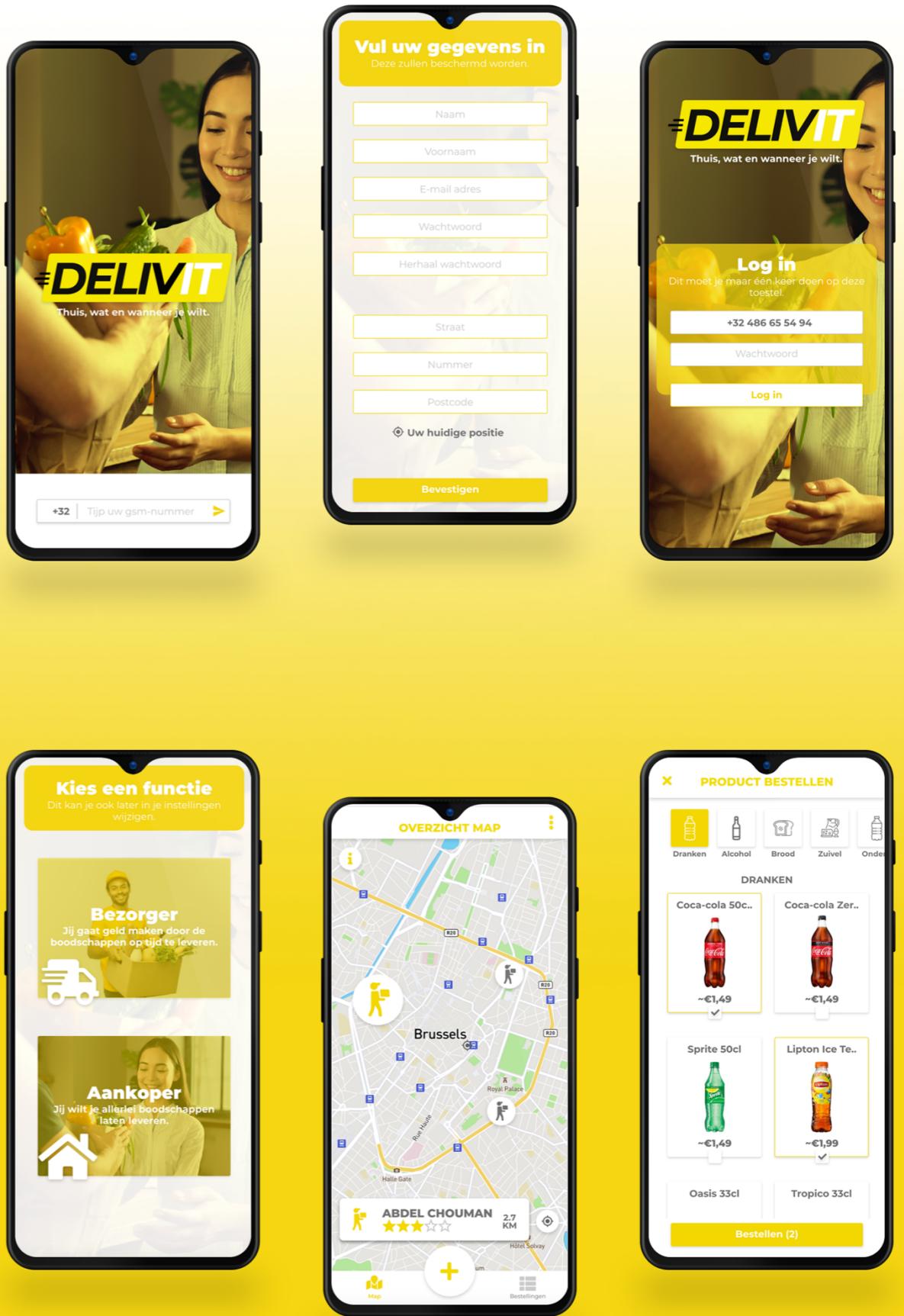
BESTELLINGEN



Mockups



Dit is een fragment van de wireframes. De volledige en prototype versie
<https://xd.adobe.com/view/621cdcbe-e219-461a-6927-fbb274ef8bc7-142d/>
 Of in bijlage beschikbaar in een Adobe XD-bestand.





Business canvas

Key Partners	Key Activities	Value Propositions	Customer Segments
Groothandelaren Kleinhandelaren Stripe (Betalingsysteem) Google (Firebase,Flutter)	Constante verbetering klantervaring Data analyseren en onderhoud app&web Bezorger & Aankoper laten communiceren	VOOR AANKOPERS Snelle service Comfortabel boodschappen doen Transparente ratingsysteem Kan kiezen wie de bezorging maakt Veilig betaling	VOOR AANKOPERS: Personen vanaf 16 jaar die hun boodschappen thuis willen krijgen. Geen tijd hebben om boodschappen te maken Geen zin hebben om boodschappen te maken weer-omstandigheden Te donker buiten Andere redenen
Customer Relationships AANKOPER : Promotie BEZORGER : Bonus Gebruikers support	Key Resources	VOOR BEZORGERS Inkomens genereren Flexibele werkuren Eigen baas Gemakkelijke aanwerving Veilig betaling	Beperkte mobiliteit hebben om boodschappen te maken Gehandicapten Mensen zonder auto Mensen die ziek zijn
Channels Sociale media & web Google & Apple App Stores Mond-tot-mondreclame Lokale campagnes	Branding Mobile/Web applicatie Database Productenlijst Actieve gebruikers (bezorgers & Aankopers)		VOOR BEZORGERS Geld (zelfstandig) willen verdienen Andere mensen willen helpen (Humanitair)
Cost Structure Advertenties sociale media Apple Store Developer account Google Play Developer account Inschrijving KBO + BTW nummer	€65,0 per maand €8,25 per maand €22,90 eenmalig €195 eenmalig	Revenue Streams DelivIt wint 20% van elke transactie op het platform.	Revenue Streams Deze 20% zijn berekend op de totaliteit van de producten, maar niet op de leveringskosten.
	Totaal opstartkosten(eenmalig): €217,90 Totaal per kosten per maand: €73,25		Inkomsten door (Google) advertenties in de applicatie.

Verdienmodel

Opstartkosten

Voor de onderneming op te starten heb ik de volgende zaken nodig.

- Google Play Developer account: € 22.90
 - Dit is om de DelivIt-app op de Play Store te kunnen publiceren.
- Inschrijving KBO + BTW nummer : € 195

Dit geeft dus een totaal van **€ 217.19** om DelivIt op te starten.

Maandelijkse kosten

- Apple Store Developer Account: € 8.25 per maand
- Advertenties: € 65 per maand
 - Flyers distributie
 - Instagram, Snapchat & Facebook

Er moet **€ 73,25** worden betaald per maand.

Marge per verkochte eenheid

Aankopers besteden (na analyse) ongeveer € 39,99 per maand om boodschappen online te maken en deze te laten bezorgen. Per gemiddeld transactie wordt € 7,99 winst gemaakt (**20%** van de producten)

De marge is dus **€ 7,99** per eenheid.

Eenheden tot het break-even punt

Aangezien ik € 73,25 vaste maandelijkse kosten heb en mijn eenheid €7,99 is, moet ik maandelijks **10** actieve leden hebben tot het break-even punt.

Het break-even punt bereiken

$$15 - 10 = 5 \rightarrow 5 \times € 7,99 = € 39,95 \rightarrow € 217,19 : € 39,95 = (5,43) \text{ } \mathbf{5,5}$$

Met **18** actieve gebruikers die ongeveer € 39,99 per maand besteden bereikt DelivIt het break-even punt na **5,5** maanden (22 weken).

Het break-even punt bereiken wilt zeggen dat alle kosten gedeckt zijn.

De onderneming heeft op dit moment geen winst en geen verlies.

Portefeuille – virtueel geld

Op het platform worden de betalingen met virtueel geld uitgevoerd.

Dit geld is eigenlijk in de portefeuille van de gebruiker. Om deze te bezitten moet de gebruiker zijn portefeuille vullen via een beschikbare betalingsmethode (bancontact, Paypal, kredietkaart). Hij zal minimum €10 in zijn portefeuille moeten storten en dit kan hij meerdere keren doen.

Met het geld in de portefeuille zal de gebruiker (aankoper) bestellingen maken, het geld word dus ook virtueel gestort aan de bezorger. DelivIt zal het geld manueel storten aan de bezorger in de komende dagen.

Dit procedure kan gewijzigd worden door technologie, in de toekomst zal men geautomatiseerd het geld storten op de PayPal of bank-account van de bezorger.

De bezorger kan met zijn portefeuille-geld zich in aankoper-modus zetten en ook bestellingen maken. DelivIt gaat daarop nog 20% service-kosten nemen.

Hieronder een voorbeeld van een bestelling.

1. De Aankoper stort € 20 op in zijn portefeuille via bancontact.
 2. De aankoper maakt een bestelling en betaald **€ 15.30**
(Hij heeft nu nog € 6.70 in zijn portefeuille).
 3. Het geld wordt tijdelijk tussen de aankoper en de bezorger geblokkeerd tot dat de bezorging voltooid is.
- Hier neemt DelivIt **20%** van de prijs van de producten, dus zonder de levering. DelivIt wint **€ 2**
4. Wanneer de bezorging correct voltooid is, ontvangt de bezorger **€ 13.50** in zijn portefeuille, dit geld kan hij gebruiken om een bestelling te maken of zal worden gestort op zijn Paypal of bankaccount.

DelivIt heeft hier **€ 2** winst gemaakt. Indien de bezorger het geld gebruikt om een bestelling te maken zal DelivIt nog **20%** winst maken op de bestelde producten.





ONTWIKKELING

Implementatie middelen

In dit hoofdstuk zal ik het hebben over de tools en methoden die ik gebruik heb om tot de eindversie van de DelivIt applicatie te bekomen.

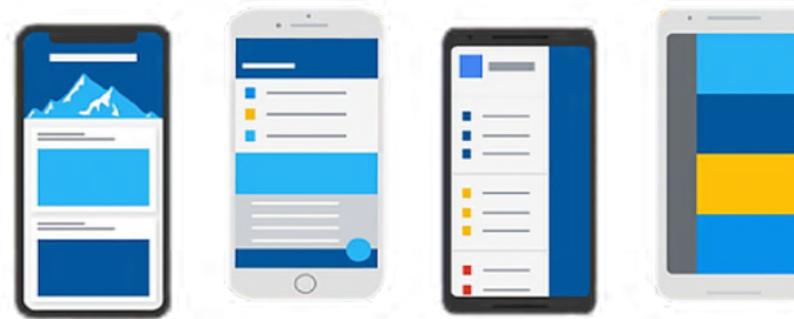
Framework Flutter

Zoals vermeld, heb ik de framework Flutter gebruikt om de applicatie te ontwikkelen. Dit is een framework van Google om native applicaties te ontwikkelen voor iOS, Android en sinds mei 2020 ook voor web.

Deze framework gebruikt de programmeertaal **Dart**. Ik ben begonnen met online lessen te volgen om stap per stap deze framework en taal te begrijpen en te kunnen gebruiken. (Ondertussen heb ik zeer veel geëxperimenteerd en heb al enkele jobaanbiedingen als flutter-developer gekregen).

Flutter is een zeer snelle, performante framework waarmee men zeer mooie en precieze designs kunnen ontwerpen. Deze is gebaseerd op "widgets". Flutter/Dart heeft nu in 2020 al een zeer uitgebreide bibliotheek (<https://pub.dev/>) en heeft in enkele jaren meer actieve projecten op GitHub dan React-native. Na onderzoek blijkt dat het meer en meer gebruikt zal worden in de bedrijfswereld.

Wat zeer interessant is met Flutter is dat de 'hot reload' zeer performant is, ik kon daardoor met meerdere apparaten ter gelijkertijd werken en zeer vlugge modificaties maken in mijn code. Deze werden onmiddellijk zichtbaar op de applicatie



Visual Studio Code

Dit is een bekende open-source code editor ontwikkeld door Microsoft die een zeer groot aantal talen ondersteunt dankzij extensies. Het ondersteunt autocompletion, syntax highlighting, debugging en git commando's.

Ik heb daarmee de Dart en de Flutter extensie geïnstalleerd. Zo kon ik iOS & Android emulators lanceren en daarop mijn applicatie testen en ontwikkelen.

XCODE – IOS (swift)

Dankzij de kennis die ik had met Swift 5 kon ik enkele bugs oplossen in de iOS applicatie van mijn project. Soms kan het zijn dat de Dart code alleen op Android werkt en enkele problemen maakt op de iOS-emulator, meestal was het door nieuwe dependencies, compatibiliteitsproblemen of nodige autorisaties (notificaties, localisatie, ...).

De release versie van de applicatie zal in de toekomst ook moeten ingesteld worden op XCODE, zodat ik deze op de Apple Store invoeg.

Android – Kotlin

Ik had nog nooit met Kotlin gewerkt, daarom moest ik tutorials gaan volgen op het internet. Dit werd alleen gebruikt om bugs op te lossen of nodige autorisaties te vragen (notificaties, localisatie, ...).

GitHub

Als back-up heb ik sinds januari een persoonlijke GitHub repository, ik upload daar mijn documenten, maar zeker ook mijn Flutter/Dart project.

Dit heeft me veel geholpen, omdat er soms enorme problemen waren bij het compilen of het installeren van een nieuwe package. Ik heb altijd gestructureerd gewerkt en correcte namen gegeven aan mijn commits.

Trello

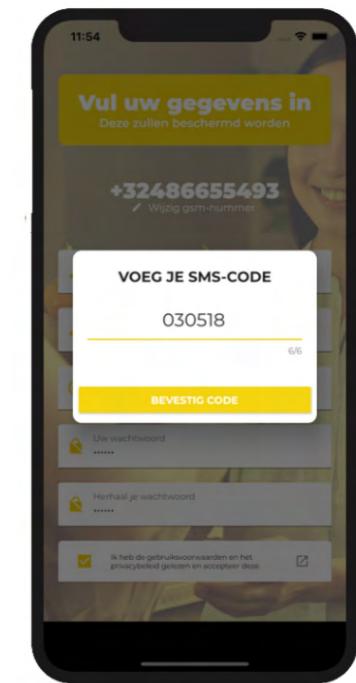
Om mijn ontwikkelingsproces goed in het oog te houden gebruikte ik Trello. Dit werd door mijn promotor aangeraden maar, ik gebruikte het toch vooraf. Dit is super om de opdrachten te structureren en snel te zien waar ik in mijn project ben. Ik zet mijn opdrachten in de volgende structuur: TE DOEN , IN PROCES, GEDAAN.

Firebase Authentication

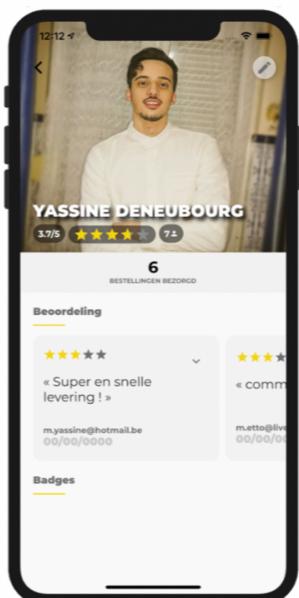


Aangezien Google een veilig authenticatie-systeem bezit, heb ik voor deze gekozen. Daarmee kan ik veilig wachtwoorden opslagen (bij Google) en gebruikers authentificeren.

Wanneer de gebruiker zijn wachtwoord vergeten is, krijgt hij de mogelijkheid om deze te veranderen. Dit gebeurt via het e-mailadres die op zijn account geregistreerd is. Ook in de instellingen kan hij zijn wachtwoord wijzigen. Om de GDPR-wetgeving te respecteren, biedt DelivIt de mogelijkheid om zijn account volledig te verwijderen.



Om meer veiligheid te geven en de multiple-accounts te vermeiden heb ik een SMS-verificatie systeem geïmplementeerd. De gebruiker moet dus altijd zijn gsm-nummer intikken om zich in te loggen of in te schrijven. Hij krijgt dan een verificatie-code via SMS om de inschrijving te finaliseren.



Firebase Storage

Om afbeeldingen en andere bestanden op te slaan gebruik ik de storage van Firebase. Deze heeft een zeer groot opslagcapaciteit en is schaalbaar.

Firebase Cloud Firestore

Als database heb ik de Cloud Firestore gebruikt. Dit is een NoSQL en document-gebaseerde database waarmee men gegevens voor mobiele en web apps kan opslaan, synchroniseren en opvragen. Real-time functies zijn ook beschikbaar. Dit wil zeggen dat er 'luisteraars' zijn die altijd de code zal vernieuwen wanneer iets gewijzigd wordt in de database.

Ik kan deze database gebruiken op wereldwijde schaal.

Ik heb de database zo efficiënt mogelijk gemaakt. Men al nu de entiteiten bespreken.

Globaal

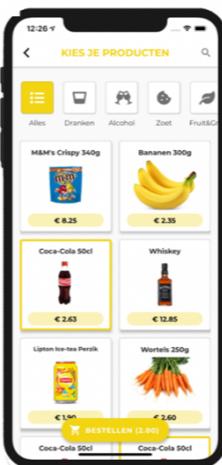
Hier sla ik globale variabelen, dit groepeert het percentage commissie die ik gebruik om winst te maken, de leveringskosten en de bepaalde uren om de avonds-kosten te definiëren. In de toekomst zal ik daar ook de verschillende verdienbare badges opslaan. Dit wordt allemaal in één uniek document geplaatst. Ik gebruik dit om vlug belangrijke elementen te kunnen wijzigen.

Gebruiker

Hier sla ik alle nodige informatie van de gebruiker. Altijd met inzicht op de GDPR-wetgeving. Ik sla de naam, voornaam, gsm-nummer, e-mailadres, online-status en portefeuille-geld op. Voor de bezorgers sla ik ook de positie (latitude & longitude), de review-berichten, review-score en de individuele prijslijst.

Conversaties

Ik sla hier alle conversaties tussen de gebruikers. Dit wil zeggen: de email-adressen van de gebruikers en alle berichten met datum en tijd. Ik sla daarnaast ook de datum en tijd van het laatste bericht om notificaties te implementeren.

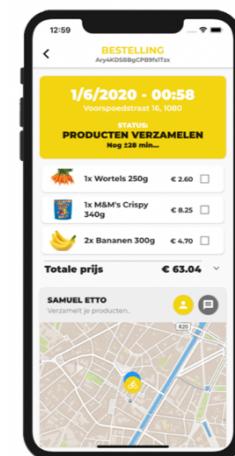


Producten

Hier sla ik alle producten die beschikbaar zijn voor de aankopers.

Ik sla de titel, de beschrijving, de categorie, een afbeelding van het product, de standaardprijs en ook de gemiddelde prijs.

De gemiddelde prijs wordt gewijzigd wanneer bezorgers hun eigen prijslijst wijzigen.



Bestellingen

Voor de structuur van de bestellingen heb ik zeer veel aanpassingen moeten doen. Hier sla ik alle nodige informatie van een bestelling. Deze moeten in praktijk door een bezorger bij een aankoper bezorgd worden.

Hier sla ik een lijst van de nodige producten met de hoeveelheid daarvan, het adres en de locatie (latitude & longitude) waar het bezorgd moet zijn.

Ten slotte ook het email-adres van de aankoper en de bezorger, de totale prijs van de producten, de leveringsprijs en de commissie-prijs voor Delivlt (winst) en de beschikbaarheid van de bestelling.

Natuurlijk ook de status van de bestelling. Deze wordt veranderd met interactie van de bezorger en de aankoper (Aanvraag, Aanbieding gekregen, Producten verzamelen, Producten leveren, Confirmatie aankoper, Bezorgd).

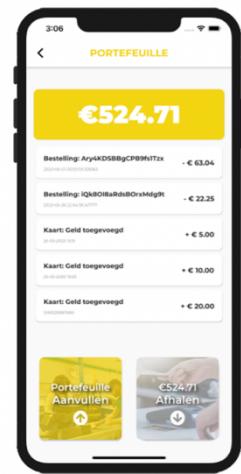
Node.js & Stripe

Om een veilig betalingssysteem te beschikken, zodat mijn app in productie-modus gebruikt kan worden, ben ik op zoek geweest naar de veiligste en performantste manier om een betalingsmethode te bouwen.

Na onderzoek kon ik meerdere betalingssystemen gebruiken zoals PayPal, Adyen, WireCard, MultiSafePay of Square, maar ik heb voor **Stripe** gekozen.

Stripe bouwt de krachtigste en meest flexibele instrumenten voor de internetcommercialisering. Of men nu een abonnementsdienst, een on-demand marktplaats, een e-commerce winkel of een crowdfunding platform creëert, Stripe kan dit zorgvuldig ontwerpen aan de hand van de zijn sterke API's. Miljoenen technologiebedrijven schalen sneller en efficiënter door hun bedrijf met Stripe te bouwen.

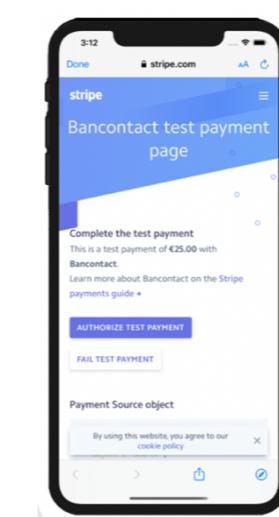
Voor developers is er documentatie ter beschikking om veilig en snelle betalingsmethoden te ontwikkelen op websites of applicaties.



Ik heb dus een account gemaakt en de test-omgeving gebruikt om dit op de DelivIt-app te implementeren. Eerst kon ik aan de hand van een library, dit onmiddellijk op Flutter functioneel krijgen. Maar voor meer veiligheid heb ik beslist dit om mijn **eigen server** te hosten.



The screenshot shows the Stripe dashboard under the 'Paiements' section. It displays a table of transactions with columns for MONTANT (Amount), DESCRIPTION, CLIENT (Email), and DATE. A warning message at the top states: "Les informations du propriétaire de l'entreprise figurant sur votre compte doivent être vérifiées. Vous ne pourrez pas recevoir de virements tant que les informations concernant le propriétaire de votre entreprise n'auront pas été examinées et vérifiées." A 'Vérifier' button is present. The table includes entries for successful payments (Réussi) and one for a payment method (Paiement).



Ik ben aan de hand van LinkedIn-learning platform begonnen met **Node.js** aan te leren. Daarna ook de package **Express.js** te gebruiken om mijn server te ontwikkelen. Daarop heb ik backend code ontwikkeld die de gebruiker de mogelijkheid geeft om veilig en vlug te betalen. Hij zal dan in zijn portefeuille op de DelivIt applicatie onmiddellijk geld ontvangen.

Dit is voor testing gehost op **heroku**:
<http://delivitapp.herokuapp.com/>



Softwarebibliotheeken en pakketten

In Flutter en Dart heb ik gebruik gemaakt van allerlei ‘packages’ die beschikbaar zijn op <http://pub.dev>. Deze zijn door grote bedrijven of gewone ontwikkelaars gemaakt en beschikbaar gesteld voor het groot publiek.

Bibliotheken/pakketten zijn ook libraries of packages genoemd in het Engels. Dit is een verzameling code die voor bepaalde algemene bewerkingen zorgt. Het voordeel is dat men als programmeur/developer niet meer de code moet herschrijven maar onmiddellijk functies van deze libraries kan gebruiken. Meestal is er een documentatie om de library op een project te installeren en deze goed te gebruiken.

Op de hele mobiele applicatie te ontwikkelen heb ik een 40 tal packages gebruikt. Deze kan men hieronder waarnemen.

```
# inschrijving en login + FIREBASE: # Algemene:  
international_phone_input: ^1.0.2 flutter_launcher_icons: "^\n0.7.3"  
firebase_core: ^0.4.4+3 auto_size_text: ^2.1.0  
firebase_auth: ^0.15.5+3 font_awesome_flutter: ^8.5.0  
cloud_firestore: ^0.13.4+2 toast: ^0.1.5  
firebase_storage: ^3.0.8 icon_shadow: ^1.0.1  
# Kaart: flutter_spinkit: "^\n4.1.1"  
flutter_map: any curved_navigation_bar: ^0.3.1  
geopoint: ^0.7.0 flutter_rating_bar: ^3.0.1+1  
geolocator: ^5.1.5 dotted_border: ^1.0.4  
location_permissions: ^2.0.5 dio: ^3.0.8  
app_settings: ^3.0.0+1 flutter_staggered_grid_view: ^0.3.0  
datetime_picker_formfield: ^1.0.0 android_intent: ^0.3.5  
bubble: ^1.1.9+1 url_launcher: ^5.4.10  
#NOTIFICATIONS: keyboard_avoider: ^0.1.2  
flutter_local_notifications: ^1.4.3 lottie: ^0.3.2  
rxdart: ^0.24.1 expandable: ^4.1.4  
flutter_app_badger: ^1.1.2 snaplist: ^0.1.8  
#AFBEELDINGEN flutter_mapbox_navigation: 0.0.7  
image_picker: ^0.6.7 cupertino_icons: ^0.1.2  
flutter_native_image: ^0.0.5 flutter_web_browser: ^0.11.0
```

Deze packages komen van de website <http://pub.dev>. Ze zijn zoals gezegd door bedrijven of individuele developers ontwikkeld en publiek gemaakt.

Sommige packages zijn er om de applicatie esthetieker te maken, ik heb packages zoals bubble, dotted border, lottie, icons, spinkit en anderen gebruikt om de user interface attractief te creëren.

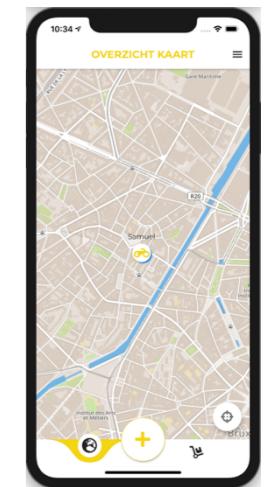
Men zal nu de belangrijke packages bekijken.

Firebase

Zoals vermeld in de implementatie middelen, is Firebase een kern van de applicatie. Ik gebruik hier de packages Firebase core, Firebase auth, Firebase storage en Firebase cloud firestore om de hele applicatie met de database te laten functioneren.

Flutter map & geolocator

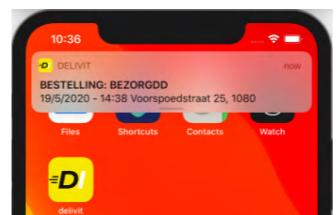
Dit gebruik ik om de kaart weer te geven en de posities van de gebruiker te kunnen vragen en opslagen. Andere functionaliteiten zoals adresvalidatie is hier ook van toepassing.



Packages zoals location permission, geopoint en app settings worden hier gebruikt om de nodige autorisaties van de toestellen Android & iOS aan te vragen om de gebruiker te mogen lokaliseren.

Flutter local notifications

Deze plug-in gebruik ik om notificaties te sturen wanneer er een nieuw bericht gekregen is. Dit werkt samen met de cloud firestore die constant aan het luisteren is wanneer er een nieuw bericht in de database wordt gestuurd. Dit gaat dus de notificaties aan de nodige gebruiker sturen.



Notificaties worden ook gestuurd wanneer de status van de bestellingen veranderd. Bijvoorbeeld wanneer een bestelling geaccepteerd, aan het aankomen of bezorgd wordt.

Hier wordt ook de package rxdart gebruikt voor snelheid en de flutter app badger om notificaties op de app-icon te laten zien.



User testing

Alvorens ik de applicatie volledig operationaliseerde, heb ik de wireframes laten testen door enkele dichte personen. Familieleden en vrienden hebben dit serieus genomen en mij eerlijke feedback bezorgd. Ik heb dankzij hun feedback de wireframes kunnen verbeteren, om een gebruiksvriendelijke ervaring (ux) te bieden.

Tijdens het ontwikkelen van de applicatie met Flutter, heb ik meerdere tests uitgevoerd. Dit leverde steeds opbouwende feedback, waardoor ik mijn applicatie performanter en efficiënter kon maken.

De test-personen zijn personen tussen de 16 en 64 jaar. Deze leeftijdscategorie stemt overeen met mijn doelpubliek. Kinderen behoren niet tot het doelpubliek omdat zij, in de meeste gevallen, niet diegene zijn die de boodschappen maken en of bezorgen.

Ik bouwde altijd een APK-versie van de applicatie en stuurde het naar de test-personen via e-mail. Zij installeerden de applicatie en maakte dan video's waar zij de applicatie gebruikten. Dit gebeurde ongeveer elke week tot dat ik een bijna perfecte eindversie had.

Ik wisselde elke week van persoon om de doeltreffendheid van mijn applicatie te versterken.

Tijdens het bekijken van hun video's, voegde ik onmiddellijk in mijn TRELLO nieuwe taken aan in de rubriek "to do's".

Met een klasgenoot was de testing iets verschillend, deze testte mijn applicatie en zette onmiddellijk "to do's" in mijn code.

De testing was van grote nut, omdat men als ontwikkelaar niet altijd weet of de gebruikers de applicatie zoals verwacht zullen gebruiken.

De meeste aanpassingen werden op de bestelling-detailpagina gemaakt. Daar moest de gebruiker precieze informatie terugvinden.

Moeilijkheden en oplossingen

De uitwerking van deze bachelorproef verliep niet altijd van een leien dak. In het begin was het moeilijk om het platform en de functionaliteit te definiëren. Gaandeweg heb ik mijn idee aangepast tot dit eindresultaat, namelijk: DelvIt.

De moeilijkheden die ik in het begin ervaarde, hadden te maken met mijn (onredelijke) wil om alle producten van de Belgische supermarkten in de applicatie op te nemen. Ik wilde namelijk per winkel een exhaustieve lijst opstellen. De bedrijven die ik hierin wilde opnemen waren Colruyt, Aldi, Lidl, Delhaize, Carrefour en Cora.

Ik kwam snel tot het besef dat het niet realistisch was omdat zij hun prijzen elke dag veranderen en om de simpele reden dat zij hun prijslijst niet met mij gaan willen delen. En vooral, omdat zij zelf ook werk maken van de ontwikkeling van eigen leveringsservices.

Deze obstakels besprak ik met mijn promotor en enkele medestudenten en besloot dan een eigen productenlijst te maken met veelgebruikte producten.

Ik kon me dus baseren op de productenlijst van een nachtwinkel in Brussel. Deze lijst was van ongeziene hulp voor definiëring van mijn eigen productenlijst. De prijzen van de producten van DelvIt zijn ook gebaseerd op prijzen van de nachtwinkel. Voor meer flexibiliteit heb ik de bezorgers de optie gegeven om de prijzen te kunnen wijzigen.

Een andere uitdaging waarmee ik geconfronteerd werd, was het betalingssysteem. In mijn prototype maakte ik gebruik van 'Square', maar er staan helaas enkel betalingen in dollars of in ponden toe. Ik voerde daardoor onderzoek naar de beste betalingsservice in Europa. Ik wal uiteindelijk terecht bij 'Stripe'. Deze keuze dwong me ook om Node.js aan te leren, zodoende ik mijn eigen veilige server kon ontwikkelen. Bij het ontwikkelen op Flutter/Dart had ik een aantal moeilijkheden met het implementeren van de notificaties. Maar dit werd in enkele dagen opgelost.



Toekomstvisie

Delivlt is nu een functionele applicatie die klaar is om in productiefase te gaan. Dit betekent dat de applicatie op Google Play Store & Apple Store geplaatst kan worden.

Nu moet ik volop inzetten op Marketing en het product bekend maken bij het gewenste doelpubliek.

Ik heb de ambitie om de applicatie nog te verfijnen en efficiënter te maken. Een aantal technische details dienen te worden aangepast. Daarnaast wens ik ook een web-versie te ontwikkelen, om de diensten ook op een website (via desktop) aan te bieden. Dankzij Firebase is het project volledig schaalbaar.

Vervolgens wil ook een jurist.e aanwerven, om alle juridische aspecten van de applicatie te onderzoeken.. Dankzij de lokalisatie-functies kan de applicatie overal in de wereld gebruikt worden. Het platform moet daarvoor alleen de correcte vertalingen en munteenheid gebruiken.

Omdat ik Flutter gebruik zal dit zeker geen probleem zijn om andere ontwikkelaars aan te werven, omdat Flutter een revolutie is in het ontwikkelen van crossplatform native applicaties.

Besluit

Dit project heeft enkele veranderingen ondergegaan op vlak van design en functionaliteiten, vaak na elke test-fase tijdens het ontwikkelen. Volgens mij is het oorspronkelijke doel van het onderzoek behaald.

Op de centrale vraag 'hoe kan men met behulp van technologie zorgen dat mensen snel en veilig hun boodschappen betalen en ook thuis ontvangen?' kan nu een antwoord worden geformuleerd. Ik heb uiteindelijk een kant-en-klare mobile applicatie kunnen ontwikkelen die de noden van mijn doelgroep ontmoet.

Alles werd goed bedenkt en getest om iedereen tevreden te maken. De aankopers gaan meerdere aanbiedingen krijgen en zo dus een keuze hebben omtrent de bezorging. Zij zullen zonder moeite boodschappen kunnen doen en betalen via hun smartphone of tablet. Zij krijgen dan alle nodige informatie over het bezorgen van hun bestelling.

De bezorgers worden geholpen door de applicatie om klanten te vinden. Zij hebben flexibiliteit in hun prijslijst en verdienen voldoende geld. Zij verdienen dus € 3.50 (of € 4.50 's nachts) per bezorging en kunnen daarnaast extra verdienen dankzij het prijsverschil. Zij kunnen daarnaast kiezen wanneer zij werken.

Delivlt verdient 20% op elke transactie. Delivlt is een intuïtieve applicatie die aankoper en de bezorger met elkaar in contact brengt en elkaar noden vervult. Delivlt creëert een gemeenschap en zal door de feedback zich dagelijks verbeteren.

Het uiteindelijke resultaat van dit project is uiteindelijk het afgelopen proces. Op basis van de onderzoeksraag kunnen nu een- bijna- perfecte applicatie aanbieden die klaar is om geplaatst te worden op iOS en Android, indien dit juridisch mogelijk is en in samenspraak met de Erasmushogeschool.

Dankzij dit project ben ik als individu ook gegroeid in mijn vakgebied. Ik kan mezelf eindelijk beschouwen als een competente Flutter-ontwikkelaar. Ik heb ondertussen 3 jobaanbiedingen gekregen als mobile-ontwikkelaar met Flutter.

Literatuurlijst

TestAchats, (29/10/2019). Plateforme de livraison de repas : un service qui roule. Geraadpleegd op 02/11/2019 via <https://www.test-achats.be/famille-prive/supermarches/news/plateforme-de-livraison-a-domicile>

Sudinfo, (28/9/2019). Colruyt: vos courses livrées à domicile dès 2019... mais attention. Geraadpleegd op 02/11/2019 via
<https://www.sudinfo.be/id77135/article/2018-09-28/colruyt-vos-courses-livrees-domicile-des-2019-mais-attention>

Delhaize. Geraadpleegd op 10/11/2019 via <https://www.delhaize.be/nl-be/CP:Hdintothekitchen>

Carrefour. Geraadpleegd op 11/11/2019 via
<https://drive.carrefour.eu/fr/livraison-levering>
<https://www.carrefour.eu/fr/services/courses-en-ligne.html>

Shipto. Geraadpleegd op 11/11/2019 via <https://www.shipto.be/>

Wotrich, V (31/3/2019). Branded apps en aankoopgedrag. Geraadpleegd op 14/11/2019 via <https://dare.uva.nl/search?identifier=7388f366-35d1-4343-8ec0-a75be826620c>

Gagnon, M.C (9/10/2019). Uber: "Bon pour l'environnement". Geraadpleegd op 17/11/2019 via <https://www.jurnaldequebec.com/2019/10/09/uber-bon-pour-l-environnement-plaide-francois-legault>

Bartosz, S (11/12/2019). Flutter vs React Native – what to choose in 2020? Geraadpleegd op 13/12/2019 via
<https://www.thedroidsonroids.com/blog/flutter-vs-react-native-what-to-choose-in-2020>

Stock-beelden in de applicatie en deze magazine van Pexels,
<https://www.pexels.com/>

Afbeeldingen persona's van FreePik,
<https://freepik.com/>

Afbeeldingen moodboard op FreePik en Google Image,
<https://freepik.com/>
<https://images.google.com/>

Icons in deze magazine van Flaticon,
<https://www.flaticon.com/>

Icons in de applicatie van FontAwesome,
<https://fontawesome.com/>





Deneubourg Mohamed Yassine
FINAL WORK 2020 - ERASMUSHOGESCHOOL