



Professionele Bachelor Toegepaste Informatica



Selfie cadeaubon adverteren en cloud storage

Bram Van Vleymen

Promotoren:

Dhr. Tom Schuyten
Dhr. Bart Clijsner

Gift2Give
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2017-2018



Professionele Bachelor Toegepaste Informatica



Selfie cadeaubon adverteren en cloud storage

Bram Van Vleymen

Promotoren:

Dhr. Tom Schuyten
Dhr. Bart Clijsner

Gift2Give
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2017-2018

Dankwoord

Eerst en vooral wil ik Gift2Give bedanken voor het aanbieden van de stageopdracht, hier heb ik met volle moed en plezier aan gewerkt. Dhr. Tom Schuyten, mijn bedrijfspromotor, waar ik altijd naartoe kon als ik vragen had en een geweldige samenwerking mee achter de rug heb, zou ik nog eens apart willen bedanken.

Dhr. Bart Clijsner, mijn hogeschoolpromotor, zou ik ook willen bedanken voor zijn hulp bij alle administratieve taken die bij men stage hoorden. Natuurlijk wil mijn ouders en kotgenoten bedanken voor de steun tijdens mijn stage en het nalezen van mijn bachelor paper.

Als laatste wil ik de hogeschool PXL bedanken voor mij de kans te geven om op stage te gaan en zo ook mijn kennis te verbreden en ervaring op te doen.

Abstract

De stageopdracht wordt verdeeld in drie verschillende opdrachten, eerst en vooral de mobiele applicatie waar een gepersonaliseerde cadeaubon kan worden opgesteld en vervolgens aangekocht kan worden. Ook kunnen de handelaren hier de cadeaubon ontwaarden. Ten tweede wordt er ook een webapplicatie gemaakt waar samengelegd kan worden en een topup kan plaatsvinden. Al de betaal mogelijkheden worden ook voorzien in de webapplicatie en vervolgens geïntegreerd in de mobiele applicatie. Ten laatste wordt er een Spring boot applicatie gemaakt die het uploaden en ophalen van de foto's voor de mobiele applicatie beheerd.

In deze bachelorpaper vinden twee onderzoeken plaats, eerst wordt het verschil onderzocht tussen twee cloud storages, Amazon S3 en Cloudinary. Hierbij wordt er een vergelijking gemaakt tussen de kosten, de snelheid en de gebruiksvriendelijkheid. De nood kwam vanuit de stageopdracht, hier moesten foto's opgeslagen worden, de mogelijkheid was het gebruik van een cloud storage of fysieke servers.

Vervolgens wordt Facebook advertising onderzocht. In dit onderzoek wordt een breder beeld gegeven op de plaatsing van Facebook advertising, wat adverteerders precies is en hoe een succesvolle campagne is opgebouwd. Vervolgens wordt ook de Facebook pixel toegelicht.

Inhoudsopgave

Inhoud

Dankwoord	ii
Abstract	iii
Inhoudsopgave	iv
Lijst van gebruikte figuren	vii
Lijst van gebruikte tabellen	viii
Lijst van gebruikte afkortingen.....	ix
Inleiding	1
I. Stageverslag.....	2
1 Bedrijfsvoorstelling [1] [2].....	2
1.1 Situering	2
1.2 Wie is Gift2Give [4].....	2
2 Voorstelling stageopdracht	3
2.1 Probleemstelling.....	3
2.1.1 Situering van het probleem.....	3
2.1.2 Stakeholders	3
2.2 Doelstellingen.....	3
2.3 Architectuur beschrijving	4
2.3.1 High level overview	4
2.3.2 Presentation Layer Mobiele app	4
2.3.3 Service layer Mobiele app	5
2.3.4 Presentation layer web applicatie	5
2.3.5 Service layer web applicatie	5
2.3.6 Service layer IPMS	5
2.3.7 Technologieën algemeen	5
2.3.8 Technologieën Mobile app.....	6
2.3.9 Technologieën (Web applicatie).....	7
2.3.10 Technologieën (IPMS).....	7
3 Stage opdrachten	9
3.1 Uitleg opdrachten.....	9
3.1.1 Opdracht 1 (mobiele applicatie).....	9
3.1.2 Opdracht 2 (IPMS)	9
3.1.3 Opdracht 3 (Web APP).....	9
3.2 Integratie opdrachten	9

3.2.1	Order flow	9
3.2.2	Samenleggen flow	11
3.2.3	Topup flow.....	11
4	Uitwerking stageopdracht 1	12
4.1	Aanpak.....	12
4.2	Stage verloop.....	12
4.2.1	De selfie	12
4.2.2	Bouwen van de cadeaubon	12
4.2.3	Ontwaarden NFC/QR.....	13
5	Uitwerking stageopdracht 2	14
5.1	Stage verloop.....	14
5.1.1	Hoe werkt de IPMS?	14
5.1.2	Foto verwerking.....	14
5.1.3	Naam generatie	15
5.1.4	Upload foto naar Amazon S3.....	15
5.1.5	Opruimen.....	16
6	Uitwerking stageopdracht 3	17
6.1	Stage verloop.....	17
6.1.1	Samenleggen en topup.....	17
6.1.2	Betaling.....	17
6.1.3	Uitnodigen via Facebook [25].....	19
6.1.4	Opstellen cadeaubon.....	19
7	Besluit	20
II.	Onderzoekstopic.....	21
1	Cloud storage.....	21
1.1	Vraagstelling	21
1.2	Onderzoeksmethode	21
1.3	Uitwerking onderzoek	21
1.3.1	Inleiding	21
1.3.2	Kosten vergelijking	22
1.3.3	Snelheid Vergelijking	23
1.3.4	Gebruiksvriendelijkheid.....	26
1.4	Conclusie	26
2	Adverteren.....	27
2.1	Vraagstelling	27
2.2	Uitwerking onderzoek	27

2.2.1	Wat is adverteren?	27
2.2.2	Wat is de Facebook 'Ads Manager'?	27
2.2.3	Plaatsing Facebook advertising [32].....	28
2.2.4	Een succesvolle campagne	32
2.2.5	Facebook pixel	33
2.3	Conclusie	33
Bibliografie		36
Bijlagen		39

Lijst van gebruikte figuren

Figuur 1 High level diagram.....	4
Figuur 2 pakketten Cloudinary	22
Figuur 3 Upload tijd Amazon S3 en Cloudinary.....	24
Figuur 4 test resultaten Cloudinary unieke en niet unieke foto's.....	25
Figuur 5 test resultaten Amazon S3 unieke en niet unieke foto's	25
Figuur 6 Facebook aantal gebruikers per miljoen.....	28
Figuur 7 Dagelijkse/maandelijkse Facebook gebruikers	29

Lijst van gebruikte tabellen

Tabel 1 Prijzen Cloudinary	23
Tabel 2 Prijzen Amazon S3	23
Tabel 3 Vergelijking prijs Amazon S3 en Cloudinary	23
Tabel 4 Vergelijkingsmatrix gebruiksvriendelijkheid.....	26

Lijst van gebruikte afkortingen

APP	Applicatie
HTTPS	Hypertext Transfer Protocol secure
IP	Internet protocol
IPMS	Image processing micro service
JSON	JavaScript Object Notation
MVC	Model view controller
NFC	Near-field communication
PDF	Portable Document Format
POM	Project object model
UI	User interface
UUID	Universally unique identifier

Inleiding

In bachelorpaper wordt er een onderzoek gevoerd tussen cloud storages en Facebook advertising. Iemand die deze paper leest moet de vragen over cloud storages en Facebook advertising kunnen beantwoorden. Ook wil ik een geslaagde Facebook advertising campagne behalen.

Bij de cloud storages werd de vergelijkende onderzoeksmethode gebruikt. Hierbij hebben we een aantal aspecten onderzocht, zoals de snelheid en gebruiksvriendelijkheid.

De stage is bij het bedrijf Gift2Give, zij zijn gevestigd in Schoten. De stage zelf vindt plaats op de Corda Incubator. Vanwege mijn studentstartup. Het project werd volgens de agile methodologie ontwikkeld.

I. Stageverslag

1 Bedrijfsvoorstelling [1] [2]

Ten eerste wordt er een situering gemaakt van het bedrijf Gift2Give, dan een beknopte beschrijving over wie Gift2Give is.

1.1 Situering

Gift2Give is een startup gevestigd in Schoten, België. Het is ontstaan in 2013 en telt 6 werknemers.

Mijn stage vond plaats op de Corda Incubator, dit is een plaats voor startende ondernemers in de technologie en service sector. [3]

1.2 Wie is Gift2Give [4]

Gift2Give verleent volgende diensten:

- 'Smart gift cards' oplossingen;
- 'Smart loyalty' oplossingen;
- 'Smart city access' oplossingen;
- 'Smart apps, in-apps, app portals & mobiele website' oplossingen;
- 'Smart reservation' oplossingen;
- Alternatieve muntsystemen.

Ze maken gebruik van creatieve en slimme technologieën, ook maken ze gebruik van marketing tools. Ze willen innovatie bereiken door deze nieuwe concepten en inzichten te transformeren tot duidelijke bedrijfsresultaten.

2 Voorstelling stageopdracht

2.1 Probleemstelling

2.1.1 Situering van het probleem

Momenteel worden de cadeaubonnen fysiek verkocht op 'near-field communication' (NFC) kaarten of via een 'portable document format' (PDF) bestand. Dit PDF bestand kan dan worden afgedrukt. Wereldwijd wordt er dagelijks meer dan 1 miljoen ton papier gebruikt. [5] Om dit te verminderen moet iedereen zijn steentje bijdragen.

Hier speelt de digitale cadeaubon een rol. Er hoeft geen papier meer gebruikt te worden en deze optie is veel gebruiksvriendelijker omdat alles digitaal kan gebeuren. De digitale bon biedt een tal van opties aan. Zo kan hij bijvoorbeeld gepersonaliseerd worden met een foto.

De handelaren die de cadeaubonnen ontwaarden werken met een verouderde applicatie (APP), deze moet ook vernieuwd worden. Dit betekent dat de mobiele applicatie moet werken voor zowel de nieuwe digitale- als de fysieke cadeaubon.

Ook is er nog geen manier ter beschikking om gemakkelijk geld samen te leggen voor een cadeaubon. Nu moet men geld verzamelen en dan een cadeaubon gaan kopen. Deze handeling zou gemakkelijk geïntegreerd kunnen worden met de digitale cadeaubon, door een portaal te maken waar men kan samenleggen.

2.1.2 Stakeholders

De oplossing van dit probleem is voordelig voor diverse partijen. Aan de ene kant is de oplossing goed voor het milieu omdat er minder papier gebruikt wordt.

Aan de andere kant krijgen de handelaren een nieuwe mogelijkheid om de cadeaubonnen te ontwaarden. Ook voor wie een cadeaubon wil kopen heeft dit project een meerwaarde. De cadeaubon kan makkelijk worden gepersonaliseerd en het wordt makkelijker om samen te leggen voor een gelegenheid.

Ten laatste maakt deze applicatie ook nog een administratieve vereenvoudiging, de cadeaubonnen moeten niet meer opgestuurd worden.

2.2 Doelstellingen

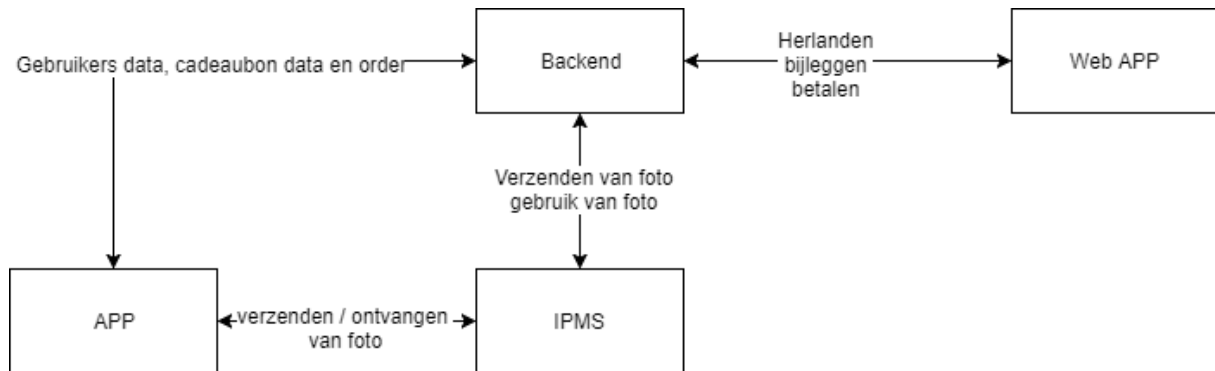
Dit project heeft enkele doelstellingen, ten eerste moet er een digitale cadeaubon kunnen gemaakt worden, met de mogelijkheid om deze te personaliseren, we noemen deze bon de Selfie bon. Vervolgens heeft de gebruiker een keuze om samen te leggen of niet.

Ook moet een handelaar zowel de fysieke als de digitale bon kunnen ontwaarden.

De handelaar krijgt ook nog een dashboard. Waarmee hij statistieken kan raadplegen over zijn ontwaarding.

2.3 Architectuur beschrijving

2.3.1 High level overview



Figuur 1 High level diagram

Een van de hoofd functionaliteiten van de mobiele applicatie is dat er foto's genomen kunnen worden, zodat deze vervolgens op de cadeaubon kunnen worden geplaatst.

Om deze foto's op te slaan wordt Amazon S3 gebruikt. De Java Spring backend voorziet endpoints om zo via 'Hypertext Transfer Protocol secure' (HTTPS) informatie te verkrijgen via 'JavaScript Object Notation' (JSON) formaat.

2.3.2 Presentation Layer Mobile app

2.3.2.1 Sale view

De 'sale view' is de start pagina van de mobiele applicatie. Hier kan men de bon zien en de waarde er van veranderen. Ook kan men hier de foto bewerken en op de camera knop drukken om een nieuwe foto te maken.

2.3.2.2 Camera view

De 'camera view' is de pagina waar men een foto kan maken of een foto kan selecteren uit de bestanden op de mobiele telefoon.

2.3.2.3 Login view

De login pagina wordt gebruikt om de handelaar en/of gebruiker in te loggen. Hier kan men ook naar de registratie pagina gaan om een nieuwe gebruiker aan te maken.

2.3.2.4 Merchant view

Op de 'merchant view' kan de handelaar de QR-code van de bon scannen. Vervolgens kan de bon gecontroleerd of ontwaard worden. De handelaar kan hier zijn dashboard raadplegen en meer informatie opvragen over zijn recente transacties. De historiek van de ontwaardingen kan hier ook opgevraagd worden.

2.3.2.5 Register view

Hier kan de gebruiker een account registreren.

2.3.3 Service layer Mobiele app

Al de views die hierboven zijn vermeld hebben een controller, deze worden gebruikt om de view aan te spreken en te manipuleren.

2.3.4 Presentation layer web applicatie

De webapplicatie heeft twee views de 'topup view' en de 'contribution view'. Bij de 'topup view' kan de gebruiker de gegevens van de cadeaubon zien en een topup uitvoeren. Bij de 'contribution view' is het mogelijk om de reden van het samenleggen te raadplegen en vervolgens bij te leggen aan een cadeaubon. Ook kan de gebruiker hier zien wie heeft bijgelegd.

2.3.5 Service layer web applicatie

De webapplicatie heeft vier verschillende services, de 'IpmsRest' wordt gebruikt voor de communicatie tussen de webapplicatie en de IPMS. De 'RestService' wordt gebruikt voor de communicatie tussen de webapplicatie en de backend.

De 'MolliePayment' service wordt gebruikt om het betaalproces bij Mollie te starten, de service start de communicatie in de constructor en voorziet één functie. Dit is de 'pay' waar de betaling gebeurt.

De 'PaypalPayment' service wordt gebruikt om het betaalproces bij Paypal te starten, hier wordt in de constructor de communicatie gestart. Deze service bevat twee functies, de 'authorizePayment' waar de betaling geautoriseerd wordt. Vervolgens bevat de service ook nog een 'pay' functie waar de betaling plaatsvindt.

2.3.6 Service layer IPMS

De ipms bevat enkele services, de belangrijkste hiervan is de 'AwsUploadService' hier wordt de connectie gelegd tussen de IPMS en Amazon S3. Deze service bevat zes functies:

- 'getFile': hiermee kan een bestand worden opgehaald van Amazon S3 als de key gekend is;
- 'uploadFile': Hier wordt het bestand up ge load naar Amazon S3;
- 'deleteFile': Hier wordt een bestand verwijderd van Amazon S3;
- 'exists': Met deze functie wordt er gecontroleerd of het bestand bestaat;
- 'getAllInBucket': Hier wordt de volledige Amazon S3 bucket opgevraagd waardoor er een gemakkelijk overzicht kan worden gegeven van de bucket;
- 'moveFile': Deze functie verplaatst een bestand in de bucket.

De volgende service is de 'ImageProcessingService', deze service verwerkt een foto zodat die klaar is om up ge load te worden. De 'VideoProcessingService' verwerkt video's. Ten laatste is er ook nog de FileController, deze controller voorziet de endpoints zodat er gecommuniceerd kan worden via Rest calls.

2.3.7 Technologieën algemeen

2.3.7.1 Git

Git is een gratis en open-source versie beheer systeem. [6] In dit project zal Git gebruikt worden om al de code te beheren.

2.3.7.2 Jira

Jira is een software development tool voor issue tracking, bug tracking en project management. Gemaakt door Atlassian [7]. In dit project wordt Jira gebruikt om de taken in te delen en om de gespendeerde tijd te loggen.

2.3.8 Technologieën Mobile app

2.3.8.1 Titanium Appcelerator

Dit is het hoofd framework dat gebruikt wordt in de mobiele applicatie. Het framework staat toe om native applicaties te schrijven in web technologieën die daarna geconverteerd worden naar de native taal. Het is een open-source framework dus het mag vrij gebruikt worden door iedereen.

Er is een mogelijkheid op te betalen voor een 'pro seat', hierbij wordt de mogelijkheid toegestaan om de 'app designer' te gebruiken. Ook is het mogelijk om live chat en email ondersteuning te krijgen. Wanneer er niet voor betaald wordt is het alleen mogelijk om ondersteuning te krijgen met het publieke forum.

Dit framework heeft ook een platformonafhankelijke API. Hierdoor moet er meestal geen platform specifieke code opnieuw geschreven worden. [8]

2.3.8.2 Alloy

Alloy is een model view controller (MVC) framework voor Titanium Appcelerator [9]. Hierdoor blijven de user interface (IU) elementen gescheiden van de programmeer logica. Dit maakt de code een stuk overzichtelijker. Het geeft de mogelijkheid om een view op meerdere plaatsen in de mobiele applicatie te gebruiken met één lijn code.

2.3.8.3 Node.js

Omdat Appcelerator voornamelijk Javascript gebruikt, is het mogelijk om node.js packages toe te voegen die geen dependencies hebben. De volgende node.js packages zijn gebruikt.

Underscore JS voegt een hele bibliotheek van functies toe. Voornamelijk de 'throttle', hierdoor kan een knop niet meerdere keren ingedrukt worden. Om zo te voorkomen dat een functie meerdere keren uitgevoerd wordt, wanneer dit niet de bedoeling is. Het voegt ook een heleboel functionaliteiten toe bij collections. Deze bibliotheek maakt javascript gebruiksvriendelijker. [10]

RESTe, deze bibliotheek maakt het maken van REST calls overzichtelijker. De volledige instellingen van de calls staan op één plaats. Daarnaast kan er een globale fout afhandeling voorzien worden. Bijvoorbeeld als het toestel geen netwerk verbinding heeft. [11]

2.3.8.4 Ti.Barcode

Deze module wordt gebruikt om de QR-code van de bon te scannen zodat deze gecontroleerd en ontwaard kan worden. [12]

2.3.8.5 Alloy Toast Notification

Deze module wordt gebruikt om toast notificaties onder aan het scherm weer te geven wanneer er een actie gebeurt in de mobiele applicatie. [13]

2.3.9 Technologieën (Web applicatie)

2.3.9.1 PHP

“PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994” [14]

Deze programmeertaal wordt gebruikt om de logica te schrijven van de webapplicatie, deze keuze is vooral gemaakt omdat de originele website van Gift2Give op PHP werkt. Dit maakt het gemakkelijk omdat dit op dezelfde server kan werken.

2.3.9.2 Mollie SDK en Paypal SDK

Deze twee SDK's worden gebruikt om de betaalmogelijkheden toe te voegen aan de webapplicatie, de mobiele applicatie gebruikt dit deel van de webapplicatie ook voor de betalingen.

2.3.10 Technologieën (IPMS)

2.3.10.1 Java

“Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, [15] and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA)” [16]

Java is de programmeertaal die voor de Image processing microservice(IPMS) gebruikt wordt. De originele backend gebruikt ook Java. Dit geeft dus de mogelijkheid om de IPMS op dezelfde server te laten werken, ook omdat Java de mogelijkheid geeft om spring boot te gebruiken.

2.3.10.2 Apache Maven

“Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.” [17]

Apache Maven wordt in dit project gebruikt om de dependencies te beheren. Apache Maven maakt het heel gemakkelijk om dit te doen. Er moet namelijk alleen een kleine aanpassing gemaakt worden in het project object model (POM).

2.3.10.3 Spring Boot

Spring Boot is een framework wat gebouwd is om zo snel mogelijk operationeel te zijn. [18] In dit project gaat Spring Boot gebruikt worden om een representational state transfer application programming interface (REST API) op te zetten.

Deze REST API wordt dan gebruikt om endpoints ter beschikking te stellen. Hiermee kan de mobiele applicatie verbinding maken om data op te halen en te versturen.

2.3.10.4 Amazon S3

Amazon S3 is een simpele cloud based storage service waar diversen bestanden kunnen worden opgeslagen. [19] In onze applicatie gaat Amazon S3 gebruikt worden om de ingezonden foto's op te slaan, vervolgens gaan we deze foto's ook weer opnieuw opvragen.

2.3.10.5 AWS Java SDK

Aws java SDK staat voor Amazon web service java software developer kit. Deze wordt gebruikt om verbinding te leggen tussen Amazon S3 en onze applicatie.

3 Stage opdrachten

3.1 Uitleg opdrachten

3.1.1 Opdracht 1 (mobiele applicatie)

In de Mobiel-applicatie kan de gebruiker een gepersonaliseerde cadeaubon aanmaken. Vervolgens kan de gebruiker deze bestellen en mensen uitnodigen om bij te leggen.

De handelaar kan hier cadeaubonnen ontwaarden en controleren. Ook kan de handelaar data verkrijgen van zijn recente transacties.

3.1.2 Opdracht 2 (IPMS)

De image processing micro service (IPMS) verzorgt de verwerking en opslag van de foto's naar Amazon S3.

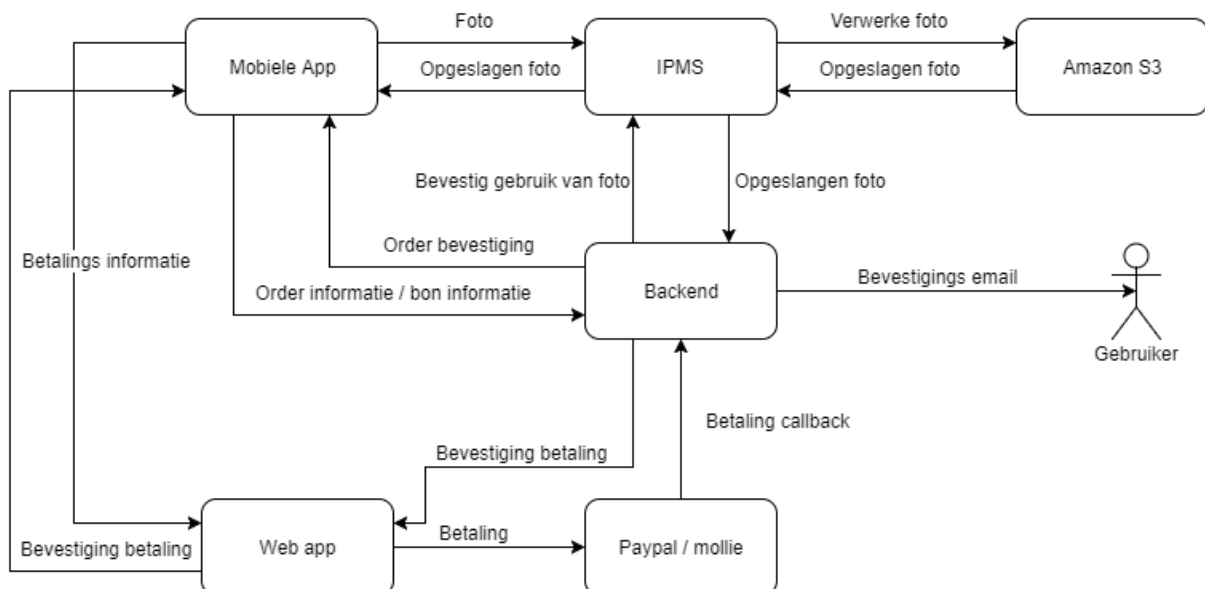
3.1.3 Opdracht 3 (Web APP)

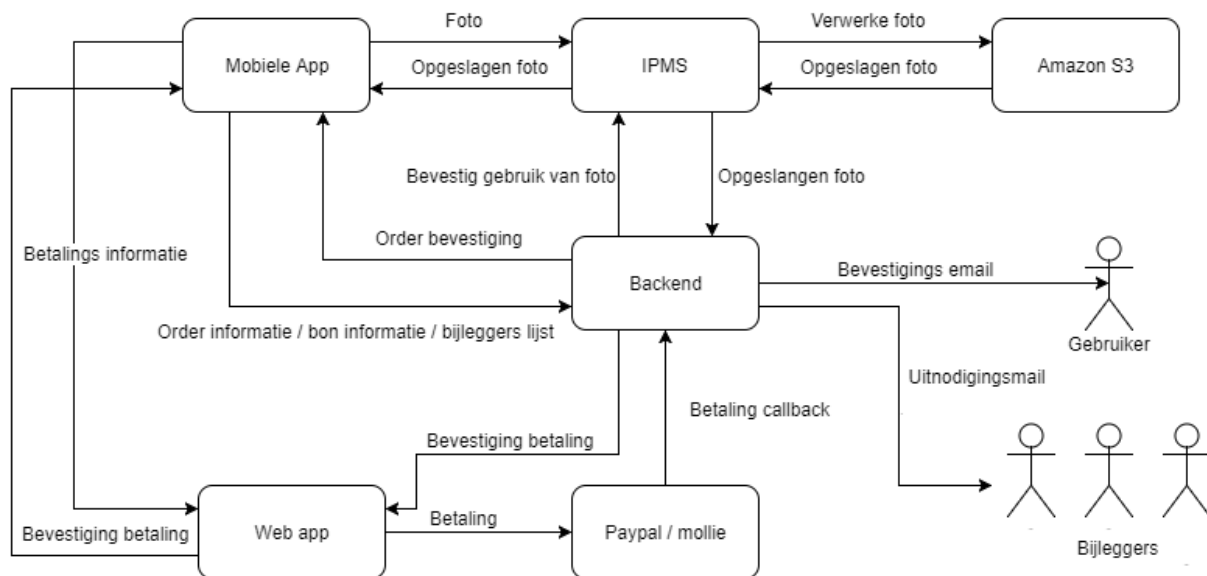
De webapplicatie geeft de mogelijkheid om bij te leggen en een cadeaubon te herladen. Ook voorziet de webapp een interface voor de handelaar waar de historiek opgevraagd kan worden en waar de handelaar zijn volgende uitbetalingen kan raadplegen.

3.2 Integratie opdrachten

De drie stageopdrachten zijn met elkaar verbonden en communiceren met elkaar om een werkend geheel te vormen. In dit hoofdstuk wordt uitgelegd hoe de communicatie tussen de verschillende onderdelen gebeurt en wat elk onderdeel doet, dit wordt gedaan door enkele processen.

3.2.1 Order flow





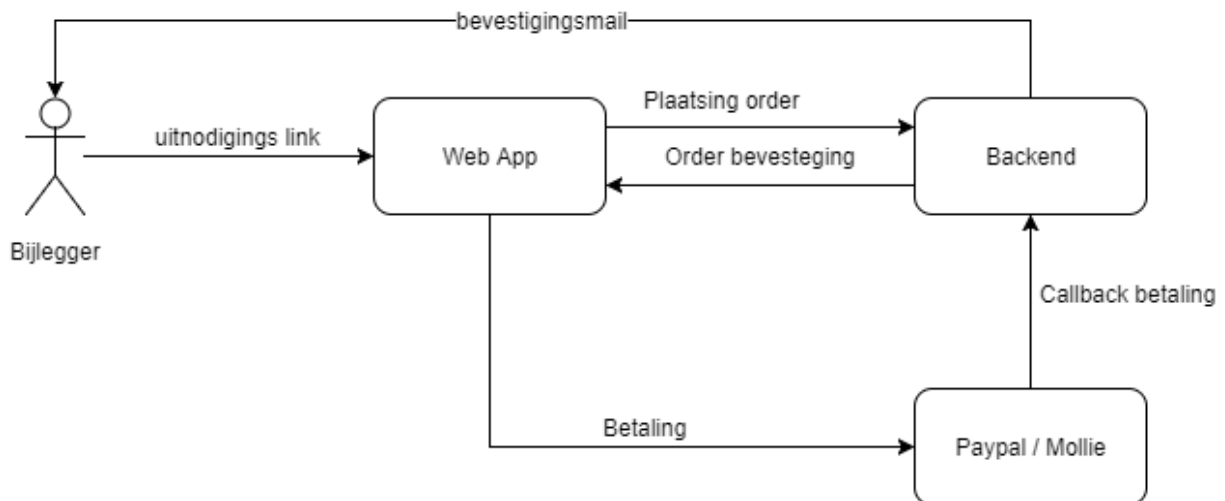
Elke nieuwe bestelling start in de mobiele applicatie, als eerste worden de stock foto's opgevraagd bij de IPMS. Deze foto's staan niet standaard in de applicatie zodat de stock foto's gemakkelijk gewijzigd kunnen worden zonder een update te doen voor de mobiele applicatie. Vervolgens kan de gebruiker beginnen met het opstellen van een cadeaubon, als de gebruiker beslist een nieuwe foto te nemen wordt deze verstuurd naar de IPMS. De IPMS verwerkt deze foto en slaat die op bij Amazon S3, de mobiele applicatie krijgt dan de verwerkte foto terug gestuurd.

Als de cadeaubon is opgesteld wordt de gebruiker doorverwezen naar het order proces waar de persoonlijke informatie ingevuld kan worden, wanneer dit gebeurd is wordt het order geplaatst bij de backend. De backend stuurt dan een bevestiging terug naar de mobiele applicatie dat het order geplaatst is, vervolgens kan de gebruiker zijn betaalwijze kiezen. Wanneer de betaalwijze gekozen is wordt de gebruiker herleid naar de web applicatie waar al de betalingen gebeuren.

De webapplicatie herleid de gebruiker dan naar de gekozen betaalwijze, Paypal of Mollie. Deze betalingsproviders sturen dan weer een callback naar de backend wanneer de betaling succesvol is. Wanneer de backend een succesvolle betaling krijgt, wordt de foto opgevraagd bij de IPMS om de cadeaubon op te stellen, deze foto wordt tegelijk op 'used' gezet omdat de foto gebruikt is, op deze manier is het gemakkelijk om de gebruikte en ongebruikte foto's te onderscheiden.

Als het een 'single order' is wordt de bevestigingsmail verstuurd naar de gebruiker en tegelijk wordt er in de mobiele applicatie vertoond dat de bestelling gelukt is. Bij een 'contribution order' oftewel een bestelling waarbij andere personen kunnen bijleggen worden de uitnodigingsmails verstuurd naar de bijleggers die vooraf zijn ingevuld in het order proces.

3.2.2 Samenleggen flow

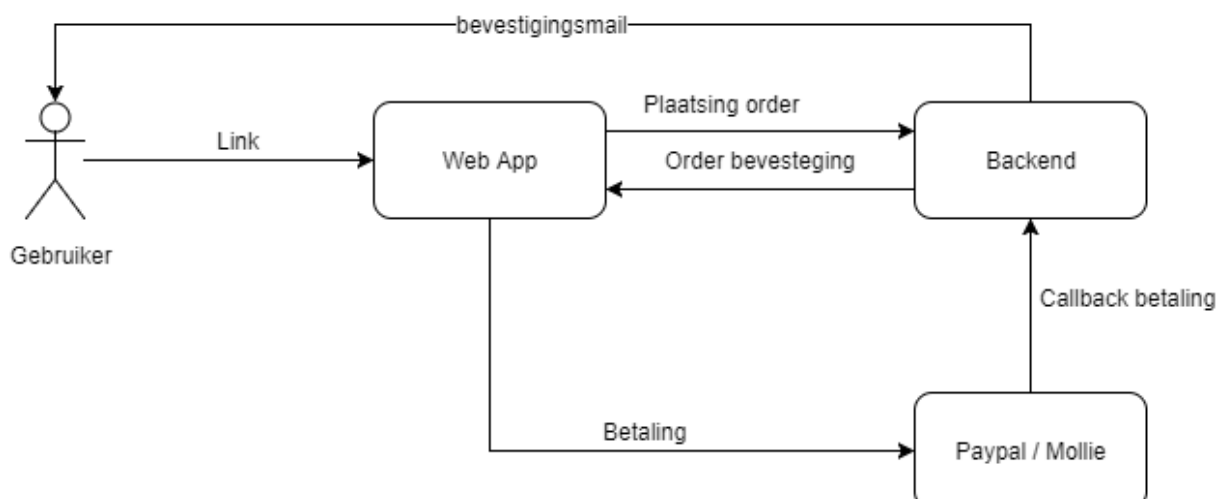


Er zijn drie mogelijkheden om op de samenleg pagina te komen:

- Door een uitnodigingsmail;
- Door een uitnodiging via facebook;
- Door het doorsturen van de link.

Als de ontvanger van de bon op deze pagina terecht komt kan er gekozen worden om de cadeaubon direct te ontvangen waardoor het bijleg proces wordt stop gezet. Andere gebruikers kunnen er voor kiezen om bij te leggen, hiervoor drukken ze op de knop bijleggen. Ze krijgen dan de mogelijkheid om een bedrag in te vullen en hun gegevens, hierna kiezen ze hun betaalwijze. Bij het kiezen van een betaalwijze wordt er een order geplaatst bij de backend, hierna wordt de gebruiker weer herleid naar de betaal pagina van de betalingsprovider. Wanneer de betaling succesvol is krijgt de backend een callback en de gebruiker krijgt een bevestiging.

3.2.3 Topup flow



De topup flow heeft dezelfde werkwijze als een persoon bijlegt alleen gebeurt het nu bij een bestaande cadeaubon en er worden geen extra kosten aangerekend bij de betaling.

4 Uitwerking stageopdracht 1

De eerste stageopdracht bestaat in het maken van een mobiele applicatie in Appcelerator. Hier kan de gebruiker een gepersonaliseerde digitale cadeaubon opstellen in enkele klikken. Om deze verder te personaliseren kan de gebruiker zelf een foto kiezen of er ter plaats een eentje maken. De foto moet opgeslagen worden in een cloud storage. Hiervoor wordt Amazon S3 gebruikt.

De handelaar moet de cadeaubon kunnen ontwaarden via NFC of door de QR-code te scannen. De handelaar krijgt een dashboard, waarin hij zijn ontwaardingen kan opvolgen en enkele grafieken met de historiek van deze ontwaardingen kan opvragen.

4.1 Aanpak

Bij dit project gebruiken we de agile methodologie. Dit wil zeggen dat er een lijst van user story's gemaakt wordt. Hierna wordt er gekozen welke user story's prioriteit hebben. Deze worden dan opgedeeld in sprints. [20]

Omdat een stage periode van 12 weken vrij kort is, lopen de sprints meestal over twee weken. Hierdoor kan om de twee weken een werkende versie van de applicatie getoond worden met de nieuwe features.

4.2 Stage verloop

In het stage verloop wordt uitgelegd hoe de mobiele applicatie tot stand is gekomen. Na een algemene inleiding volgt een overzicht van de sprints, om het verloop overzichtelijker te maken.

De stageopdrachten lopen wel door elkaar in de sprints omdat deze afgestemd moesten worden. Ook was dit nodig om na elke sprint een werkende versie te hebben. Hier wordt alleen het verloop besproken van stageopdracht 1, de mobiel applicatie.

4.2.1 De selfie

De selfie of de foto op de cadeaubon is een van de belangrijkste features in de mobiele applicatie zonder dit was er geen selfie cadeaubon. In de 'camera view' was het de bedoeling dat de gebruiker zowel een foto als een video kon nemen door de knop in te houden. Dit bleek zeer moeilijk te zijn om te bouwen en viel buiten de scope van het project.

In Android was het niet mogelijk om deze twee verschillende intents samen te starten zonder veel extra functionaliteit te maken specifiek voor Android. Ook was het niet mogelijk om aan de video intent mee te geven hoelang de video mocht zijn.

Hierdoor werd er besloten om alleen een foto toe te staan, de gebruiker kan een foto maken met beide camera's op het mobiele toestel of kan een foto inladen via de galerij. Die vervolgens wordt doorgestuurd naar de IPMS om verwerkt te worden.

4.2.2 Bouwen van de cadeaubon

Bij het bouwen van de cadeaubon zijn er verschillende opties, de gebruiker kan de kleuren van de cadeaubon kiezen en een gepersonaliseerde foto maken. Ook kan de gebruiker een bedrag kiezen voor de cadeaubon.

Het weergeven van deze gepersonaliseerde cadeaubon gaf enkele problemen. In het begin zou de backend telkens als er een wijziging was aan de bon een nieuwe bon opstellen en deze doorsturen naar de mobiele applicatie. Maar een andere optie was om de bon zelf aan te maken met de UI-

elementen, op deze manier moest de backend de bon maar één keer opstellen namelijk om het einde van een bestelling. Dit bleek een veel betere optie te zijn ook was deze sneller.

Bij een bestelling krijgt de backend een metadata map doorgestuurd waar de data van de cadeaubon in staat. Deze map bevat de kleuren en de key van de foto zodat die kan opgevraagd worden bij de IPMS. Het bedrag zelf wordt niet doorgestuurd maar komt binnen bij de callback van de betaling, hierdoor konden er geen foute gebeuren en kreeg de gebruiker altijd het bedrag op de bon dat betaald was.

4.2.3 Ontwaarden NFC/QR

De handelaar kan met deze mobiele applicatie de cadeaubon ontwaarden. Hiervoor zijn twee mogelijkheden als de cadeaubon NFC heeft kan de handelaar het toestel op de kaart leggen en dan het bedrag invullen dat ontwaard moet worden. Als de cadeaubon geen NFC heeft kan de handelaar de QR-code scanner openen en de QR-code op de bon scannen. Vervolgens kan de handelaar weer het bedrag invullen en ontwaarden.

De documentatie en test projecten van Appcelerator NFC waren niet meer up-to-date voor de nieuwe versie van het framework. De 'ForegroundDispatcher' moest uitgeschakeld worden wanneer de applicatie pauzeerde anders crashte de applicatie. Dit stukje code moest uitgevoerd worden op het 'pause' event maar in de nieuwe versie van appcelerator moest dit gebeuren op het 'onPause' event. Dit stond nergens gedocumenteerd en de test projecten waren ook niet aangepast naar de nieuwe versie. Ook was er nog een lijn code vergeten, de adapter werd niet rechtstreeks gestart hierdoor was het nog niet mogelijk om een NFC-tag te scannen.

Ik heb zelf een commit gedaan op de repository van het NFC test project met deze nieuwe aanpassingen, ook heb een wijzing van de wiki doorgestuurd naar de beheerders van de Appcelerator wiki. Hiermee hoop ik dat andere ontwikkelaars niet zolang vast zitten op dit probleem als ik.

5 Uitwerking stageopdracht 2

Stage opdracht twee bestaat uit het maken van een image processing micro service. Dit zorgt ervoor dat de gebruiker zijn foto, die in de mobiele applicatie gemaakt wordt, kan opslaan. Daarna kan deze opgehaald worden.

De IPMS zorgt voor de verbinding tussen Amazon S3 en de mobiele applicatie. Ook zorgt deze voor de verbinding tussen de backend en Amazon S3.

5.1 Stage verloop

In het stage verloop wordt er uitgelegd hoe de IPMS tot stand gekomen is. Hier wordt alleen het verloop besproken van stageopdracht twee, de IPMS.

Eerst wordt er uitgelegd hoe de IPMS werkt, hierna wordt de verwerking van de foto uitgelegd en wat er daarna mee gebeurt. Dan volgt er informatie over hoe de foto wordt geüpload naar Amazon S3. Ten laatste wordt de opruim functie besproken.

5.1.1 Hoe werkt de IPMS?

De IPMS is gemaakt in Java spring. Dit is gedaan zodat de IPMS kan worden aangesproken via een RESTfull API. De IMPS bevat 2 REST calls.

De '/upload' hiermee kan de app een foto uploaden. Daarna wordt die verwerkt en vervolgens geüpload naar AmazonS3. De mobiele applicatie krijgt dan het ID terug van de foto waarmee deze opgevraagd kan worden.

De '/get/image/{key}' hiermee kunnen de backend en de mobiele applicatie de foto opvragen, het wordt terug gegeven als een Base64 string, deze kan dan opnieuw geconverteerd worden naar een byte array.

5.1.2 Foto verwerking

De IPMS kan ook video's verwerken maar dit deel wordt niet gebruikt. De foto komt binnen als een Base64 string, deze moet eerst terug omgezet worden naar een byte array.

```
public static BufferedImage createImageFromBytes(byte[] imageData) {
    ByteArrayInputStream bais = new ByteArrayInputStream(imageData);
    try {
        return ImageIO.read(bais);
    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

Vervolgens wordt de byte array omgezet naar een 'Buffered image' zodat deze bewerkt kan worden. [21]

```
Graphics2D g2d = outputImage.createGraphics();
g2d.drawImage(inputImage, 0, 0, scaledWidth, (int) (scaledWidth / aspectRatio),
null);
g2d.dispose();
```

De 'buffered image' wordt gebruikt om de ingestuurde foto te resizen, de 'bufferd image' wordt omgezet naar een 'Graphics2D' object. Dit heeft de mogelijkheid om de foto te tekenen naar de correcte afmetingen, hierbij wordt de originele beeldverhouding behouden.

```
BufferedImage dest = new BufferedImage((int) rect.getWidth(), (int)
rect.getHeight(), BufferedImage.TYPE_INT_ARGB);
Graphics g = dest.getGraphics();
```

```
g.drawImage(src, 0, 0,
            (int) rect.getWidth(), (int) rect.getHeight(), (int) rect.getX(), (int)
            rect.getY(),
            (int) rect.getX() + (int) rect.getWidth(), (int) rect.getY() + (int)
            rect.getHeight(), null);
g.dispose();
```

Vervolgens moet de foto geknipt worden zodat die op de cadeaubon past. De cadeaubon heeft vaste resolutie. Namelijk 310px breed en 110px hoog. Maar hier gebruiken we het dubbele zodat de kwaliteit van de foto behouden wordt. Hierna wordt de foto naar een bestand geschreven en uploadt naar AmazonS3.

5.1.3 Naam generatie

Als een foto geüpload wordt naar Amazon S3 moet deze een unieke key hebben.

```
Date date = new Date();
DateFormat df = new SimpleDateFormat("MM-dd-YY-HH-mm-ss-SSS");
String reportDate = df.format(date);

String[] split = reportDate.split("-");
String fileName = "";

for (String value : split) {
    fileName += encodingTable[Integer.parseInt(value) % 64].toString();
}
return fileName;
```

Als eerste gebruikte we een Date die een formaat heeft van 'MM-dd-YY-HH-mm-ss-SSS', oftewel een timestamp. Deze timestamp encoderen we dan met een Base64 encoding tabel. Hierbij werd er ook nog gecontroleerd of de key al bestond, als dit zo was werd er een nieuwe key gegenereerd.

Deze methode had enkele problemen, namelijk als er foto's binnenkwamen binnen de 16ms dan werd dezelfde key gegenereerd. Omdat deze nog niet geüpload waren bij Amazon S3. Op dat moment ging de validatie functie van de key melden dat deze nog niet bestond en dan werd de foto overschreven bij Amazon S3.

Dit hebben we opgelost door een universally unique identifier (UUID) te gebruiken. De kans dat er eenzelfde UUID wordt gemaakt is niet nul, maar klein genoeg zodat deze verwaarloosbaar is. [22] Hierdoor waren de keys van de bestanden wel langer maar dit was geen probleem want de key moest nooit handmatig ingetypt worden.

5.1.4 Upload foto naar Amazon S3

Het uploaden naar Amazon S3 is heel makkelijk uitgevoerd met de SDK die Amazon aanbied.

```
AWSCredentials credentials = new BasicAWSCredentials(accessKey, secretAccessKey);
s3client = AmazonS3ClientBuilder.standard()
    .withCredentials(new AWSSStaticCredentialsProvider(credentials))
    .withRegion("eu-west-2").build();
```

Er moet een 'AmazonS3ClientBuilder' object aangemaakt worden met de toegangscode en de geheime toegangscode in de parameters.

```
s3client.putObject(new PutObjectRequest(bucketName, file.getName(), file));
file.delete();
```

Daarna kan het bestand worden opgeslagen door de 'putObject' methode. Waar de naam van de Amazon S3 bucket, de key en het bestand zelf wordt meegegeven in de parameters. Het bestand wordt opgeslagen in een aparte map via het mac-adress van het mobiele toestel. Hierdoor weten we altijd welk apparaat de foto's verstuurd heeft. Hierna verwijderen we de file op de server van de IPMS, want die hebben we niet meer nodig.

5.1.5 Opruimen

Er worden meer foto's verzonden naar de IPMS dan dat er gebruikt worden. Het kan zijn dat de gebruiker een foto maakt in de mobiele applicatie en dan nooit zijn bestelling afmaakt. Om deze foto's te onderscheiden van de foto's die gebruikt worden op een cadeaubon moet er een stukje code geschreven worden.

Eerst was het de bedoeling op elke woensdag om 3:00h een cron job te laten runnen die al de orders van de week binnenhaalde en dan controleerde in de bucket welke foto's gebruikt werden en dan de andere naar een vuilbak directory te verplaatsen. Omdat de IPMS nog geen data van de backend bezitten werd er een andere oplossing besloten.

Als er een nieuwe foto wordt upgeloadt naar de IPMS wordt deze automatisch in de 'unused' directory geplaatst. Vervolgens wanneer de backend deze foto opvraagt, wordt de foto in de 'used' directory geplaatst. Als de backend de foto opvraagt, wilt dit zeggen dat het order compleet is en dat de foto gebruikt gaat worden. Vervolgens heeft de backend ook de mogelijkheid om de foto weer terug te verplaatsen naar de 'unused' directory wanneer de cadeaubon vervalst.

Hierdoor is er een overzicht van welke foto's gebruikt worden en welke niet. Dit geeft de mogelijkheid om 'lifecycle rule' in te stellen in de bucket die de ongebruikte foto's na een bepaalde tijd verwijderd of naar een goedkoper type opslag te verhuizen.

6 Uitwerking stageopdracht 3

Stageopdracht 3 bestaat uit de webapplicatie, in de web applicatie zijn er enkele functionaliteiten:

- Samenleggen;
- Topup;
- Betalen met Paypal;
- Betalen met mollie;
- Opstellen van een cadeaubon;
- Weergeven foto's voor de administrators.

6.1 Stage verloop

In het stage verloop wordt er uitgelegd hoe de webapplicatie tot stand gekomen is. Hier wordt alleen het verloop besproken van stageopdracht 3, de webapplicatie.

Eerst wordt er uitgelegd hoe het bijleggen werkt vervolgens hoe de topup werkt. Dan wordt er uitgelegd hoe de betaling is opgebouwd en hoe de cadeaubon kan worden opgebouwd in de webapplicatie. Ten laatste wordt er besproken hoe administrator kant is opgebouwd.

6.1.1 Samenleggen en topup

De webapplicatie voorziet de webinterface om bij te leggen en een topup te doen. Deze functionaliteiten werken ongeveer via hetzelfde proces, behalve dat bij een topup geen extra kosten worden aangerekend.

Eerst en vooral krijgt de gebruiker een overzicht van het order, hierbij kan de gebruiker zien wie al bijgelegd heeft en voor wat er bijgelegd wordt. Ook is het mogelijk om een preview van de cadeaubon te zien. Bij de topup krijgt de gebruiker ook een preview te zien van de kaart.

Vervolgens kan de gebruiker beslissen of ze willen bijleggen, hierna wordt de gebruiker doorverwezen naar een pagina waar de persoonlijke gegevens kunnen invullen. Daarna kan de gebruiker kiezen welke betaalwijze ze willen gebruiken, voordat de webapplicatie in het betaal proces gaat, wordt een nieuwe 'contributor' aangemaakt. Hier controleert de backend of deze persoon uitgenodigd was, als dit zo is wordt dat record in de database aangepast maar niet als de persoon voor de tweede keer bijlegt of niet uitgenodigd was dan wordt er een nieuw record gemaakt. Hierna wordt de gebruiker doorverwezen naar het betaal proces.

De persoon voor wie de cadeaubon bestemt is krijgt een speciale link waar er een optie is om het geld nu te gebruiken, dan krijgt de gebruiker de cadeaubon automatisch doorgestuurd en wordt het samenleg proces beëindigd.

6.1.2 Betaling

In de webapplicatie worden twee betalingsproviders gebruikt, Mollie en Paypal. Al de betalingen van het project gaan door de webinterface, dit is omdat de SDK's voor appcelerator van paypal verouderd was en niet meer werkten na behoren. Voor Mollie was er zelfs geen ter beschikking. Deze SDK's waren beschikbaar in PHP.

6.1.2.1 Mollie [23]

Eerst wordt een nieuwe 'Mollie api client' aangemaakt met de API key die Mollie aanbied. Vervolgens moest er een makkelijke manier zijn om het betalen van de extra kosten uit te schakelen, dit is

gedaan door een class variabele te declareren 'buyerPaysFee' wanneer deze op true staat worden de extra kosten van de betaling aangerekend bij de klant.

Voor een topup betaald Gift2Give de extra kosten, dit is gedaan om de topup te promoten. Vervolgens wordt het order id, contribution id, het bedrag, de extra kosten, het totaal en of het een topup is in de 'metaDataArray' geplaatst. Dit was de extra data die de backend nodig had voor het verwerken van het order.

Dan wordt het payment object aangemaakt waar het bedrag, de beschrijving, de redirect url, webhook url en de 'metaDataArray' wordt meegegeven. Vervolgens wordt de gebruiker herleid naar de betaalpagina van Mollie waar de betaling kan plaatsvinden. Wanneer de betaling gebeurt is krijgt de backend een callback op de 'webhookUrl' met de data van de betaling en de 'metaDataArray' waarbij het order afgewerkt kan worden.

6.1.2.2 Paypal [24]

De betaling bij paypal is iets complexer dan bij Mollie, eerst moet een nieuwe 'API context' aangemaakt worden met de 'client Id' en de 'client secret' die Paypal aanbied. Vervolgens moet ook de mode van de betaling gezet worden, deze zijn 'sandbox' of 'live'. Waarbij 'sandbox' de test omgeving van paypal is.

Daarna moet er een autorisatie request naar Paypal worden gestuurd met de betalingsgegevens. Zoals bij Mollie moesten de extra kosten gemakkelijk door- of niet doorgerekend worden. Dit is gedaan met dezelfde variabele als bij Mollie, de 'buyerPaysFee'.

```
$payer = new Payer();
$payer->setPaymentMethod("paypal");
$item = new Item();
$item->setName('Online GiftCard')
    ->setCurrency('EUR')
    ->setQuantity(1)
    ->setSku("10069")
    ->setPrice($price);
```

Om een autorisatie te starten moet eerst een 'Payer' object aangemaakt worden waar de betaal methode wordt gezet. Dan moet een artikel aangemaakt worden, met de naam 'online GiftCard'. Vervolgens moet het valuta gezet worden zodat Paypal weet welk valuta ze moeten gebruiken, ook moet de hoeveelheid en het artikel id gezet worden. Ten laatste moet bij het artikel de prijs gezet worden. Dit artikel moet dan toegevoegd worden aan een lijst van artikelen.

Vervolgens moet een 'Details' object worden aangemaakt waar de transportkosten, belasting en het subtotaal wordt geplaatst. Hier wordt de 'handeling fee' ook nog toegevoegd als de klant de extra kosten moet betalen. Het 'Amount' object wordt hierna aangemaakt waar opnieuw het valuta wordt meegegeven, de totaal prijs en het 'Details' object wordt hier ook in geplaatst.

```
$transaction->setAmount($amount)
    ->setItemList($itemList)
    ->setDescription("Online GiftCard")
    ->setInvoiceNumber(uniqid())
    ->setCustom(json_encode($metaDataArray));
```

Hierna wordt de 'metaDataArray' weer aangemaakt met dezelfde data als bij een betaling met Mollie. Dan wordt er een nieuwe transactie aangemaakt waarbij al de vorige data samenkomt. Dan volgen ook de 'redirect urls' met een 'return url' en een 'cancel url'.

```
$payment = new Payment();
$payment->setIntent("sale")
    ->setPayer($payer)
```

```
->setRedirectUrls($redirectUrls)
->setTransactions(array($transaction));
$payment->create($this->apiContext);
```

Vervolgens wordt een 'Payment' object aangemaakt waar de intent, de payer, de redirect urls en de transactie wordt in geplaatst. Hierna wordt het order verstuurd naar Paypal. Paypal stuurt dan een 'Approval link' terug waar de gebruiker de betaling kan autoriseren.

Wanneer de autorisatie gebeurt is stuur paypal de gebruiker terug naar de webapplicatie. Hier wordt er gecontroleerd of de autorisatie succesvol was. Wanneer de autorisatie succesvol is, finaliseert de webapplicatie de betaling door een 'payment execution' te sturen. Ten laaste krijgt de backend een callback van Paypal met de status van de betaling, als de betaling succesvol is kan het order afgehandeld worden.

6.1.3 Uitnodigen via Facebook [25]

De webapplicatie voorziet de functionaliteit om personen uit te nodigen via facebook. Hiervoor wordt de facebook API gebruikt. Dit is gemakkelijk te gebruiken, eerst en vooral moet er een 'app id' aangevraagd worden bij Facebook. Vervolgens moet de webapplicatie redirecten naar Facebook met de link die verzonden moet worden, app id en de redirect url. Het bericht kan niet gezet worden dit moet de gebruiker zelf doen. Dan kan de gebruiker op Facebook zelf selecteren naar wie het bericht verzonden moet worden, daarna wordt de gebruiker weer herleid naar de bijleg pagina.

6.1.4 Opstellen cadeaubon

In de webapplicatie is er ook een mogelijkheid om een nieuwe cadeaubon aan te maken, dit deel van de webapplicatie kan gebruikt worden op meerde websites om de cadeaubon te promoten. Hier kan de gebruiker een foto uploaden en de kleuren van de cadeaubon kiezen, vervolgens kan de gebruiker zijn gegevens invullen en een 'solo order' of een 'contribution order' starten. Hierna wordt de gebruiker weer herleid naar het betaalproces.

7 Besluit

Toen in het begin van het project de requirements werden opgesteld dacht ik dat ik te weinig te doen ging hebben tijdens mijn stage, het blijkt dat om een project af te werken en de laatste bugs op te lossen er toch veel tijd en werk in moet gestoken worden. Omdat ik op de Corda incubator zat zonder directe begeleiding blonk ik uit in zelfstandig werk, ik had evenzeer een directe communicatie lijn met mijn bedrijfspromotor wanneer nodig. Ik werkte ook heel snel dit was vooral omdat ik in een professionele omgeving zat met weinig geluid.

Bij dit project moest ik enkele nieuwe frameworks leren, dit verliep snel en zonder problemen. Dit kwam omdat er duidelijke en uitgebreide documentatie ter beschikking was. Wanneer de deadlines dichterbij kwamen kreeg ik een boost waardoor ik veel sneller en gestructureerder kon werken, hieruit blijkt dat ik deadlines nodig heb. Ook was er totaal geen spraken van stress tijdens de deadlines.

Omdat ik gewend ben on zelfstandig te werken verliep de communicatie soms wat stroef, dit wil niet zeggen dat Gift2Give niet wist waarmee ik bezig was. Ik plaatste namelijk dagelijks mijn vooruit gang in Slack en Jira. Ik wilde veel te snel starten aan het project hierdoor ben ik enkele problemen tegen gekomen die gemakkelijk verholpen konden worden door meer research te doen naar Appcelerator en de components voor het starten met programmeren.

De programmeertalen zoals PHP en java die ik op de PXL heb geleerd kwamen zeer goed van pas in dit project, ook de kennis van Spring boot die we in de les gezien hadden kwam goed van pas. Hierdoor kon ik snel de endpoints van de IPMS opzetten. Ook heb ik een architectuurplan moeten maken dit had ik ook al eerder aan de PXL gezien en verliep hierdoor zonder problemen nadat ik de presentaties nog eens opnieuw bekeken had. De Agile methodologie die wordt aangeleerd bij de PXL kwam van pas, vanuit het It-project heb ik geleerd om Jira te gebruiken, dit was ook de issue tracking tool die ik gebruikt heb tijdens mijn stage.

Tenslotte ben ik zeer blij met het eindproduct dat ik kan opleveren na mijn stage het was zeker een leerrijke ervaring die in de toekomst van pas gaat komen.

II. Onderzoekstopic

1 Cloud storage

1.1 Vraagstelling

In dit project moeten de foto's die de gebruiker maakt opgeslagen worden zodat ze vervolgens gebruikt kunnen worden op de cadeaubonnen. Er wordt een vergelijking gemaakt tussen Cloudinary en Amazon s3 waarbij de functionaliteit bij Amazon zelf opgezet moet worden.

De vragen die dus gesteld worden zijn:

- Welke cloud storage is de snelste;
- Welke cloud storage is de goedkoopste;
- Welke cloud storage biedt de meeste vrijheden;
- Welke cloud storage is het gemakkelijkste te implementeren?

1.2 Onderzoeksmethode

Het doel van de stageopdracht is om een mobiele applicatie te maken waarmee gepersonaliseerde cadeaubonnen opgesteld kunnen worden.

Het onderzoek maakt een vergelijking tussen Cloudinary en Amazon S3 voor het opslaan van de foto's van de gebruiker. Beide oplossingen worden opgenomen in het ontwikkelingsproces en worden dan vergeleken aan de hand van verschillende normen.

Het onderzoek is geslaagd wanneer de bovenstaande vragen beantwoord zijn. Het doel van het onderzoek is om meer duidelijkheid te scheppen over de manieren waarop foto's kunnen worden opgeslagen en een duidelijk advies te formuleren voor de gebruiker.

1.3 Uitwerking onderzoek

1.3.1 Inleiding

In de inleiding wordt er uitgelegd wat cloud storage, Cloudinary en Amazon S3 precies zijn.

1.3.1.1 Wat is cloud storage?

Cloud storage is een model van dataopslag, waarbij de data meestal op verschillende servers en vaak plaatsen staan. Deze worden meestal beheerd door een hosting bedrijf dat er voor zorgt dat deze data altijd en overall beschikbaar zijn. Mensen en organisaties huren of kopen opslag ruimte om deze dan te gebruiken voor het opslaan van hun data. [26]

1.3.1.2 Wat is Cloudinary? [27]

Cloudinary is een foto verwerkende backend voor web en mobiele ontwikkelaars. Het geeft de mogelijkheid om foto's te uploaden via een API, die dan verder opgeslagen worden in een cloud storage. Ook is het mogelijk om deze foto's te manipuleren door bijvoorbeeld te schalen en filters toe te passen.

Cloudinary heeft ook nog een 'powerful Web-based management console'. Deze geeft de mogelijkheid om informatie te zien over de geüploade foto's.

Bijvoorbeeld:

- Hoeveelheid nieuwe foto's;
- Hoeveelheid foto's in totaal;
- Hoeveelheid opslagruimte er gebruikt wordt;
- Hoeveelheid bandbreedte er gebruikt wordt;
- Mogelijkheid om foto's te bekijken.

1.3.1.3 Wat is Amazon S3? [19]

Amazon S3 is een simpele cloud based storage service waar diversen bestanden kunnen worden opgeslagen, waaronder foto's. De bestanden worden opgeslagen als objecten in 'buckets'. Er kan een ongelimiteerd aantal bestanden in een 'bucket' worden opgeslagen. Deze bestanden kunnen een omvang hebben van wel 5 terabyte.

Key features:

- Eenvoud: Amazon S3 is gebouwd om makkelijk bruikbaar te zijn. Met een volledige REST API en SDK's kan dit gemakkelijk geïntegreerd worden met derde partij applicaties;
- Duurzaamheid: Amazon S3 is beschikbaar in veel regio's van de wereld met globale redundancy in elke regio;
- Schaalbaarheid: Uitgaves kunnen van dag tot dag veranderen en een globale deployment kan worden uitgevoerd in een enkele minuten;
- Security: De datatransfer gebeurt over SSL en wordt automatisch geëncrypteerd op het moment dat het bestand succesvol geüpload is.

1.3.2 Kosten vergelijking

Hier wordt de kostprijs van Amazon S3 vergeleken met die van Cloudinary.

1.3.2.1 Cloudinary [28]

Cloudinary heeft 3 pakketten:

- Het gratis pakket bevat 300 000 foto's, 20 000 maandelijkse transformaties, 10GB opslagruimte en 20GB bandbreedte;
- Het 'plus' pakket kost \$89 per maand en dit bevat 2 miljoen foto's, 150 000 maandelijkse transformaties, 50GB opslagruimte en 150GB maandelijkse bandbreedte;
- Het 'advanced' pakket bedraagt \$224 per maand en dit bevat 5 miljoen foto's, 400 000 maandelijkse transformaties, 120GB opslagruimte en 400GB maandelijkse bandbreedte.

FREE No time limit Fully Featured	Plus \$89 / Month	Advanced \$224 / Month
300,000 Total Images & Videos	2 Million Total Images & Videos	5 Million Total Images & Videos
20,000 Monthly Transformations	150,000 Monthly Transformations	400,000 Monthly Transformations
10 GB Managed Storage	50 GB Managed Storage	120 GB Managed Storage
20 GB Monthly Net Viewing Bandwidth	150 GB Monthly Net Viewing Bandwidth	400 GB Monthly Net Viewing Bandwidth

Figuur 2 pakketten Cloudinary

Bij Cloudinary telt een transformatie als een upload of als er een bewerking met een foto gebeurt. Telkens als er iets met een foto gebeurt, telt dit dus als een transformatie.

1.3.2.2 Amazon S3 [19]

Bij Amazon is de prijs afhankelijk van de plaats, in dit geval Londen.

Voor de eerste 50TB wordt \$0.024 per GB aangerekend. Voor de volgende 450TB betaal je \$0.023 per GB en meer dan 500TB kost \$0.022 per GB. Al deze prijzen zijn op maandelijkse basis.

Vervolgens zijn er ook nog kosten verbonden aan de requests van de service, namelijk \$0.01 per 1,000 PUT, COPY, POST, or LIST requests en \$0.01 per 10,000 GET en alle andere requests.

Er zijn ook nog kosten voor het ophalen van data; deze bedragen \$0.01 per GB.

1.3.2.3 Vergelijking

Hier gaan we de exacte kosten berekenen. We gebruiken de pakketten van Cloudinary omdat deze een vaste prijs hebben.

		Cloudinary Opslag (GB)	Cloudinary transformations	Cloudinary bandbreete (GB)		Totaal Prijs
FREE	Hoeveel	10	20,000	20		\$ -
PLUS	Hoeveel	50	150,000	150		\$ 89.00
ADVANCED	Hoeveel	120	400,000	400		\$ 224.00

Tabel 1 Prijzen Cloudinary

Op tabel 1 ziet u de prijzen van Cloudinary zoals hierboven vermeld, weergegeven in een tabel.

		Amazon Opslag (GB)	Amazon POSTS	Amazon bandBreete (GB)	GETS	Totaal Prijs
FREE	Hoeveel	10	20,000	20	60000	
	Prijs	\$ 0.24	\$ 0.20	\$ 0.20	\$ 0.06	\$ 0.70
PLUS	Hoeveel	50	150,000	150	450000	
	Prijs	\$ 1.20	\$ 1.50	\$ 1.50	\$ 0.45	\$ 4.65
ADVANCED	Hoeveel	120	400,000	400	450000	
	Prijs	\$ 2.88	\$ 4.00	\$ 4.00	\$ 0.45	\$ 11.33

Tabel 2 Prijzen Amazon S3

Op tabel 2 ziet u de prijzen van Amazon S3. Hier hebben we geprobeerd dezelfde pakketten te bekomen zoals bij Cloudinary. Één transformatie staat bij Cloudinary gelijk aan één upload, hieruit hebben we afgeleid dat één transformatie gelijk staat aan een POST. Het aantal GET requests staat niet vermeld bij Cloudinary. Hiervoor hebben we het aantal POST's vermenigvuldigt met 3.

Vergelijking	Amazon S3	Cloudinary	Verskil
FREE	\$ 0.70	\$ -	\$ 0.70
PLUS	\$ 4.65	\$ 89.00	\$ 84.35
ADVANCED	\$ 11.33	\$ 224.00	\$ 212.67

Tabel 3 Vergelijking prijs Amazon S3 en Cloudinary

In tabel 3 ziet u het prijsverschil tussen de pakketten. Het prijsverschil tussen beide cloud storages is heel groot. Cloudinary is een stuk duurder om foto's op te slaan. Natuurlijk bieden ze een waaier van transformaties aan om een foto te bewerken. Maar als er alleen naar de prijs wordt gekeken is het financieel veel voordeliger om foto's op te slaan bij Amazon S3. Maar niet als het gratis pakket op Cloudinary voldoende is.

1.3.3 Snelheid Vergelijking

Bij de snelheidsvergelijking gaan we het snelheidsverschil meten tussen Cloudinary en Amazon S3. Er worden enkele verschillende testen gedaan. Deze testen worden gedaan op verschillende internet verbindingen en verschillende dagen zodat er een groter beeld gegeven kan worden over de snelheid van beide cloud storages. De testen vinden wel allemaal plaats in België.

De eerste test gaat een upload zijn van 200 keer dezelfde foto. Daarna testen we of er een verschil is bij een breed assortiment van foto's. Hierbij maken we gebruik van 15 verschillende foto's.

1.3.3.1 Upload 200 identieke foto's

Bij deze test gaan we 200 keer dezelfde foto uploaden naar Cloudinary en Amazon S3. Hierbij maken we gebruik van een klein programma in Java. In dit programma wordt de tijd gemeten voor een upload. Daarna wordt het gemiddelde berekend, de snelste en traagste tijd wordt ook bijgehouden.

```
float uploadAmount = 100f;
Long fastestCloud = 100000L;
Long slowestCloud = 0L;
Long cloudTotal = 0L;
for (int i=0; i<uploadAmount; i++){
    System.out.println("Uploading Cloudinary Image " + (i + 1));
    long start = System.currentTimeMillis();
    cloudinaryUploadService.Upload(file);
    long stop = System.currentTimeMillis();
    long timeSpent = stop-start;
    cloudTotal += timeSpent;
    if(timeSpent>slowestCloud){
        slowestCloud = timeSpent;
    }
    if(timeSpent<fastestCloud){
        fastestCloud = timeSpent;
    }
    System.out.println("Time spend: " + timeSpent + "ms");
}
System.out.println("---CLOUDINARY---");
System.out.println("Total time spend for upload: " + (cloudTotal)+ "ms");
System.out.println("Average time spend uploading: " + (cloudTotal/uploadAmount) +
"ms");
System.out.println("The fastest time was: " + fastestCloud + "ms");
System.out.println("The slowest time was: " + slowestCloud + "ms");
```

De totale tijd die wordt opgenomen is alleen de tijd gespendeerd aan het uploaden. Dit is voorbeeld code voor Cloudinary. Voor Amazon S3 wordt dezelfde structuur gebruikt.

---CLOUDINARY---	---CLOUDINARY---
Total time spend for upload: 262132ms	Total time spend for upload: 111466ms
Average time spend uploading: 1310.63ms	Average time spend uploading: 1114.61ms
The fastest time was: 1039ms	The fastest time was: 838ms
The slowest time was: 2974ms	The slowest time was: 2311ms
---AMAZON S3---	---AMAZON S3---
Total time spend for upload aws: 106659ms	Total time spend for upload aws: 23504ms
Average time spend uploading: 533.255ms	Average time spend uploading: 235.02ms
The fastest time was: 364ms	The fastest time was: 125ms
The slowest time was: 2088ms	The slowest time was: 1297ms

Figuur 3 Upload tijd Amazon S3 en Cloudinary

Op figuur 3 kunt u de resultaten waarnemen. Deze zijn verrassend. Bij de totale gespendeerde uploadtijd is Amazon S3 bijna 2,5 keer sneller. Bij de 2^e test is Amazon S3 zelfs 4,7 keer sneller. Dit is omdat de server van Cloudinary zich bevindt in de Verenigde Staten van Amerika. [29] Dit is nergens te vinden op hun website, ik ben hier achter gekomen door de API te pinggen en dan het internet protocol (IP) adres op te zoeken.

Onze Amazon S3 server die we gebruiken bevindt zich in Londen, dit hebben we kunnen selecteren bij het maken van de bucket. Hierdoor hebben we een veel snellere verbinding.

1.3.3.2 Upload 15 verschillende foto's

Omdat de vorige test een heel groot snelheidsverschil gaf tussen Amazon S3 en Cloudinary wordt hier alleen nagegaan of er een verschil is tussen het uploaden van dezelfde foto en verschillende foto's zowel bij Amazon S3 als bij Cloudinary.

De totale upload grootte is ongeveer hetzelfde zodat de resultaten representatief zijn.

```

List<File> fileList = new ArrayList<File>();

File singleImage = new File("C:\\Users\\BramV\\Desktop\\image.jpg");
//Loading images
for(int i = 0; i<15; i++){
    fileList.add(new File("C:\\Users\\BramV\\Desktop\\Pictures\\" + (i + 1) +
".jpg"));
}

int j = 0;
Long totalTime = 0L;
for (File file: fileList) {
    long start = System.currentTimeMillis();
    cloundinaryUploadService.Upload(file);
    Long stop = System.currentTimeMillis();
    totalTime += (stop-start);
}
System.out.println("Total time spend uploading 15 unique images: " + totalTime +
"ms.");

Long totalTimeSame = 0L;
for(int i = 0; i<15; i++){
    long start = System.currentTimeMillis();
    cloundinaryUploadService.Upload(singleImage);
    Long stop = System.currentTimeMillis();
    totalTimeSame += (stop-start);
}
System.out.println("Total time spend uploading 15 the same images: " +
totalTimeSame + "ms.");

```

Eerst en vooral worden alle foto's ingeladen zodat dit niet meer moet gebeuren voor de upload. Hierdoor kan er geen vertraging voorkomen. Daarna worden eerst de 15 unieke foto's upload waarbij we de tijd meten. Vervolgens wordt de tijd opgeteld, zo bekomen we de totale upload tijd.

Vervolgens wordt dezelfde foto die ook eerder geladen is 15 keer geupload. De grootte van de foto die hetzelfde blijft is 90,1kb. Dus de totale grootte is 1351,5kb. De geaccumuleerde grootte van de 15 unieke foto's is 1360,5 kb, er is dus maar een verschil van 9kb.

```

Total time spend uploading 15 unique images: 17284ms.
Total time spend uploading 15 the same images: 18078ms.

```

Figuur 4 test resultaten Cloundinary unieke en niet unieke foto's

Figuur 4 toont de resultaten van Cloundinary, er is niet veel verschil te merken. De 794ms zijn verwaarloosbaar. Dit kan te maken hebben met de verschillende groottes tussen de bestanden.

```

Total time spend uploading 15 unique images: 5115ms.
Total time spend uploading 15 the same images: 2854ms.

```

Figuur 5 test resultaten Amazon S3 unieke en niet unieke foto's

Figuur 5 toont de resultaten van Amazon S3, hier is er duidelijk wel een verschil te merken. Het uploaden van dezelfde foto is ongeveer 1,8 keer sneller. Dit betekent dat Amazon S3 een soort van caching gebruikt. Hier wordt niet rechtstreeks iets van gezegd in de documentatie. Er is wel een mogelijkheid om een storage gateway aan te maken waardoor de caching nog sneller en efficiënter gebeurt. [30]

1.3.4 Gebruiksvriendelijkheid

Functionaliteiten	Amazon S3	Cloudinary
SDK	JA	JA
Duidelijke documentatie	JA	JA
Dashboard		
Weergaven hoeveelheid bestanden	JA	JA
Bandwith tracker	NEE	JA
Totaal storage	NEE	JA
Hoeveelheid calls op een file	NEE	JA
Bestanden verwijderen	JA	JA
Bestanden ophalen	JA	JA
Snel bestanden weergeven	NEE	JA
Folders	JA	JA
Zoeken	JA	JA
Foto verwerking	NEE	JA
Automatische handelingen	JA	NEE

Tabel 4 Vergelijkingsmatrix gebruiksvriendelijkheid

In de gebruiksvriendelijkheid loopt Cloudinary voor, dit is omdat bij Amazon de meeste services softwarematig moeten ingebracht worden en niet via het dashboard. Amazon heeft dan wel weer de mogelijkheid om automatische handelingen in te voeren zoals het verwijderen van bestanden na een aantal dagen.

Ook is het bij Amazon niet mogelijk om foto's in bulk te zien, ze moeten telkens apart geopend worden.

1.4 Conclusie

Amazon S3 blijkt 19 keer goedkoper te zijn dan Cloudinary, tenzij het gratis pakket van Cloudinary wordt gekozen. Amazon S3 is ook 4,5 keer sneller dan Cloudinary, dit is omdat Cloudinary hun servers in de Verenigde Staten staan. Cloudinary heeft een meer gebruiksvriendelijk dashboard waardoor de statistieken van de dataopslag veel duidelijker zijn. Het is wel niet mogelijk om automatische handelingen in te stellen bij Cloudinary.

Voor dataopslag is Amazon S3 de beste keuze, maar als het gaat over foto's en er zijn veel manipulaties nodig wint Cloudinary, dit is dan ook wat zij extra aanbieden.

2 Adverteren

Elk project heeft een vorm van marketing nodig, bij dit project werd besloten om Facebook advertising te gebruiken. Wegens de korte periode van de stage is het onderzoek niet in de praktijk toegepast, maar een theoretisch plan uitgewerkt zodat dit naderhand kan toegepast worden.

2.1 Vraagstelling

Marketing is een heel groot aspect van een succesvolle applicatie, hierdoor komt de applicatie namelijk in de handen van de gebruiker. In dit onderzoek wordt er gefocust op Facebook advertising, de onderzoeksvragen die gesteld worden zijn:

- Waar plaatst Facebook advertising zich op de markt;
- Hoe is een succesvolle Facebook advertising campagne opgebouwd;
- Hoe weet ik of mijn campagne succesvol is;
- Welke tools zijn er ter beschikking;

2.2 Uitwerking onderzoek

2.2.1 Wat is adverteren?

Adverteren is een geluids- of visuele vorm van marketing, dit bevat een bericht voor het promoten van een product, service of idee. Het is verschillend van PR omdat de adverteerder controle heeft over het bericht en hier ook voor betaald. [31]

Adverteren gebeurt op verschillende platformen:

- Kranten;
- Televisie;
- Radio;
- Reclame borden;
- Social media;
- Websites.

De presentatie hiervan wordt een advertentie genoemd oftewel een 'ad'.

2.2.2 Wat is de Facebook 'Ads Manager'?

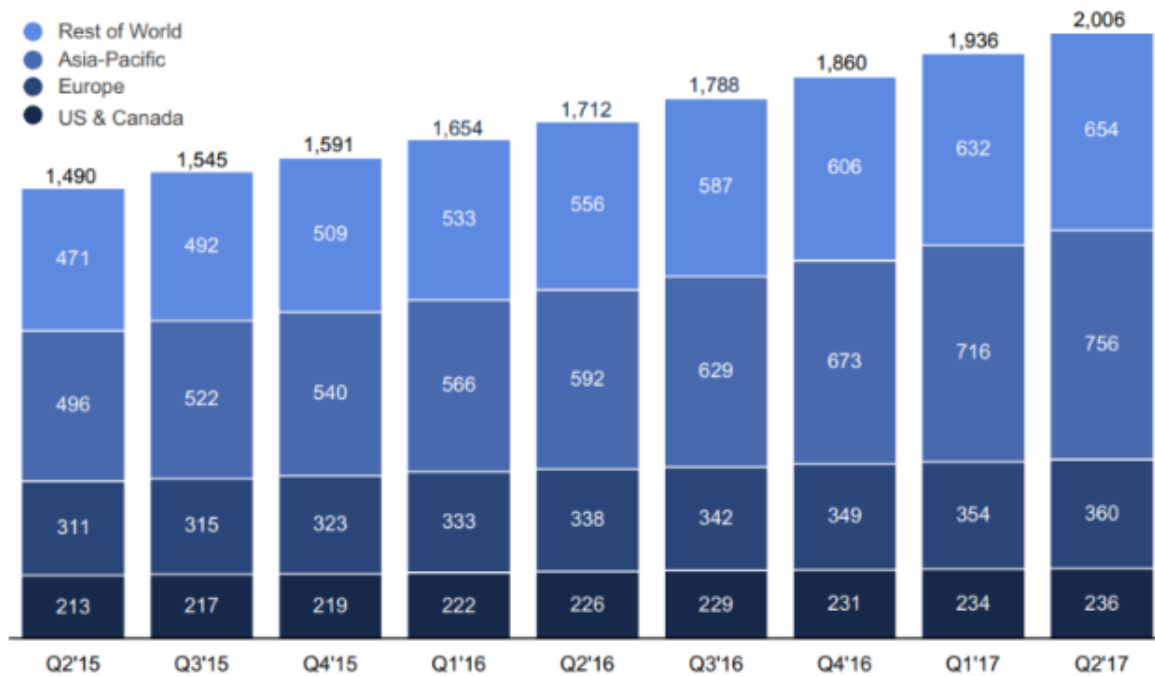
De Facebook 'Ads manager' is een all-in one tool voor het maken van ads, beheren en activeren van campagnes. Het is gebouwd voor alle adverteerders met veel of weinig ervaring. [32]

De 'Ads manager' geeft ook de mogelijkheid om statistieken te raadplegen van campagnes. De 'Ads manager' kan gebruikt worden op mobiele toestellen en desktop.

2.2.3 Plaatsing Facebook advertising [33]

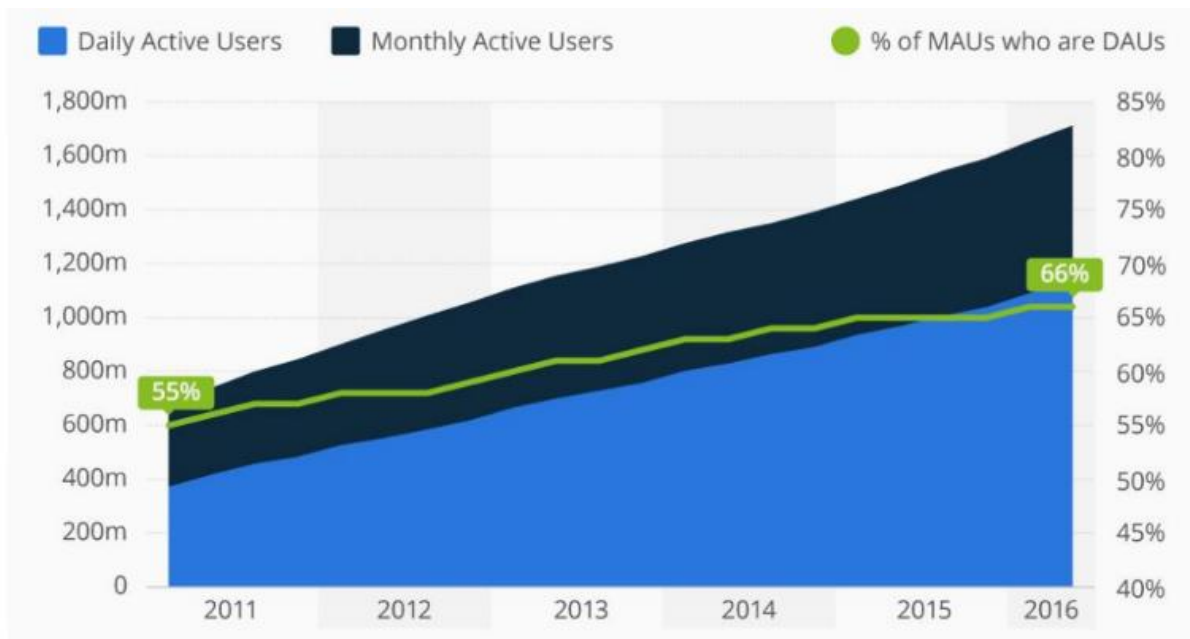
2.2.3.1 Cijfers Facebook

Facebook is op het moment de meest gebruikte social media platform, in kwartaal twee van 2017 telt Facebook meer als twee miljard actieve gebruikers.



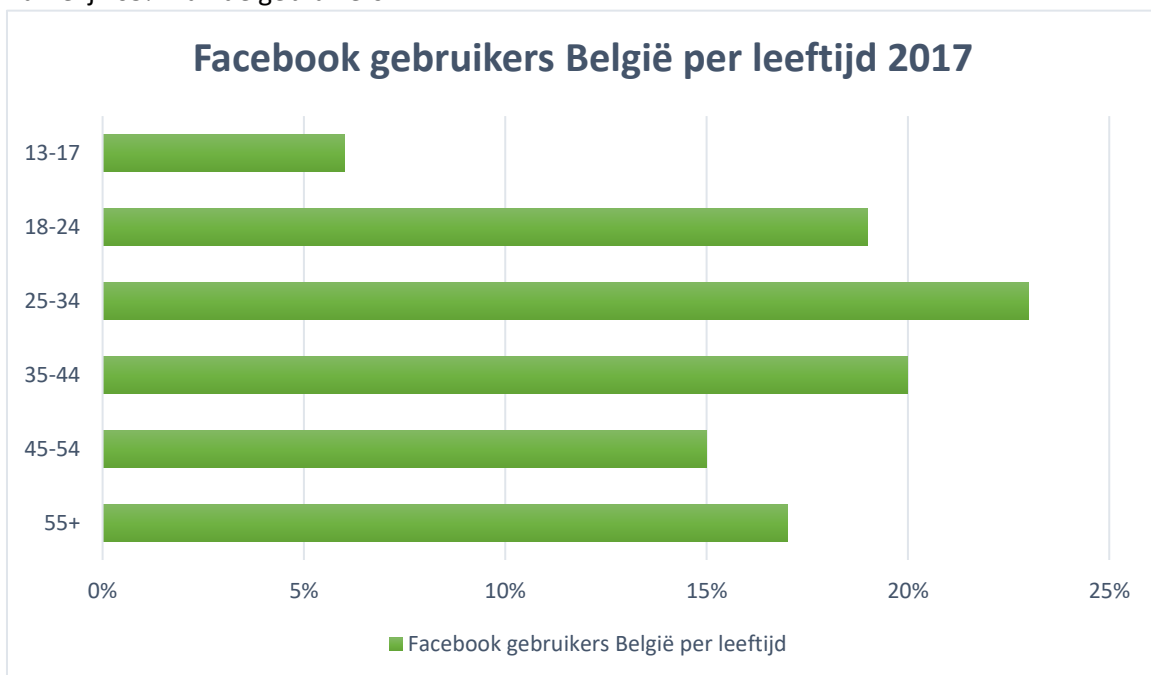
Figuur 6 Facebook aantal gebruikers per miljoen

Facebook blijft ook heel sterk groeien, zoals op figuur 5 te zien is. In België telt Facebook iets minder dan 6,5 miljoen actieve gebruikers. Ongeveer 66% van deze gebruikers zijn dagelijks actief. [34] Het dagelijkse gebruik is met de jaren ook zeer gestegen. In 2011 waren 55% van de Facebook gebruikers dagelijks actief, in 2016 was dit 66%.



Figuur 7 Dagelijkse/maandelijke Facebook gebruikers

Een gemiddelde gebruiker spendeert dagelijks 35 minuten op Facebook en voor Instagram gebruiker is dit 15 minuten. Facebook wordt tegenwoordig veel gebruikt voor het vergaren van nieuws, namelijk 65% van de gebruikers.



Facebook geeft de mogelijkheid om al de leeftijdscategorieën te bereiken met een adverteer campagne, opvallend is dat de 55-plussers tegenwoordig veel meer actief zijn op Facebook. Deze leeftijdscategorie heeft namelijk een stijging van 20% vergeleken met 2016, de 15-17 categorie heeft een lichte daling van 8%. De andere leeftijdscategorieën zijn min of meer onveranderd, in België gebruiken evenveel mannen als vrouwen Facebook.

Een Europese gebruiker is ongeveer \$ 20 waard, dit bedrag is vooral de inkomsten van het adverteren. Een Amerikaanse gebruiker is ongeveer \$ 66 waard, dit is vooral omdat het adverteren daar veel duurder is. Het organisch bereik is ook sterk gedaald, dit is omdat er veel meer content

gemaakt wordt dan vroeger. Een gemiddelde Facebook gebruiker heeft gemiddeld meer dan 1,500 berichten, hier wordt maar 20% van weergegeven in de nieuwsfeed.

2.2.3.2 Waarom adverteren op Facebook?

Facebook is een platform wat gebruikt wordt door 2 miljard gebruikers, het adverteren werkt zowel op facebook als Instagram. Ruim 1 op de 2 internetgebruikers zijn bereikbaar door Facebook ads, dit geeft een heel groot voordeel vooral bij digitale producten. Maar Facebook advertising wordt ook gebruikt voor van allerlei lokale services, zoals tuinmannen en schoonmaak bureaus. De mogelijkheden om personen te bereiken zijn eindeloos.

Het adverteren op searchengines wordt ook steeds duurder, Facebook advertising is op het moment nog niet zo populair, hierdoor is het goedkoper. In vergelijking met adverteren op searchengines heeft facebook ook een hogere click-through-rate(CTR), dit wil zeggen dat er meer personen op de advertentie klikken. Dit wilt niet zeggen dat er ook meer conversies plaats vinden, de aankoopintentie is namelijk lager omdat de persoon in kwestie niet actief aan het zoeken is naar het product. Dit is op te lossen door remarketing toe te passen, hier later meer over.

De cost-per-mill(CPM), het bedrag dat betaald wordt per 1000 impressies (aantal personen die de advertentie zien) is ook lager. Deze data hangt af van campagne tot campagne maar over het algemeen is Facebook goedkoper.

Facebook werkt ook met people-based targeting, de searchengines gebruiken cookie-based targeting, als de cookies van een browser gereset worden moet dit volledig opnieuw beginnen. People-based targeting werkt op basis van het Facebook profiel van de persoon in kwestie om zo te selecteren welke ads het meeste succes gaan hebben en deze wordt dan weergegeven. Dit systeem is ook uniek en wordt nu alleen door facebook gebruikt.

Facebook gebruikers krijgen verschillende labels toegevoegd aan hun profiel die verder weer gebruikt worden om te selecteren welke advertenties er worden weergegeven.

Het is ook mogelijk om heel specifiek te gaan selecteren welke facebook gebruikers de advertentie zien, zo kan er bijvoorbeeld gebruik worden gemaakt van de interesses en het device dat er gebruikt wordt. Ook is er de mogelijkheid om plaatsen te selecteren, dit kan tot specifieke steden en dorpen.

2.2.3.3 Advertenties

Facebook plaats de advertenties op heel veel verschillende plaatsen:

- Nieuwsfeed ads;
- Rechterkolom ads;
- Expresartikelen;
- In-stream video ads;
- Voorgestelde video's;
- Instagram verhalen;
- Instagram ads;
- Audience network;
- Messenger ads.

Bij het maken van een advertentie kan er gekozen worden waar de advertenties geplaatst worden, de positie van een advertentie verandert ook de prijs. Een rechterkolom ad is veel goedkoper dan een nieuwsfeed ad, maar deze is een stuk kleiner en heeft ook een lage click-through-rate. De expresartikel ad is meer gericht op posts die te maken hebben met het artikel en niet voor het verkoop van artikelen.

Het 'audience network' is een nieuwe soort advertentie die Facebook aanbied, hierbij wordt de advertentie geplaatst op websites die onafhankelijk zijn van Facebook. Het is ook niet mogelijk op te selecteren op wat voor soort website de advertentie terecht komt, de click-through-rate is hier zeer laag en wordt op het moment afgeraden door experts.

Er zijn ook verschillende type advertenties [35]:

- Foto ad;
- Video ad;
- Tekst ad;
- Link ad;
- Link video ad;
- Carousel ad;
- Lokale bekendheid ad;
- Event ad;
- App installatie ad;
- Lead ad;
- Canvas ad;
- Dynamische product ads;
- Collections ad.

Deze advertenties hebben allemaal hun eigen doeleindes, een lead ad is bijvoorbeeld om gebruikers hun gegevens te ontvangen of om hun een korte vragenlijst te laten invullen. De video ad, foto ad, dynamische product ad zijn goed voor het promoten van producten. Er is geen verschil in kostprijs bij het type ad. De adverteerder betaalt evenveel voor een foto ad als voor een video ad. Maar de click-through-rate verschilt.

Onrechtstreeks heeft het type ad wel invloed op de prijs, een video ad wordt veel beter ontvangen door de gebruikers dan een tekst ad, hierdoor gaat de adverteerder uiteindelijk meer moeten betalen voor een tekst ad met hetzelfde bereik.

2.2.4 Een succesvolle campagne

2.2.4.1 Campagne doel

Een succesvolle Facebook advertising campagne start met het selecteren het correcte campagne doel, Facebook optimaliseert namelijk voor dit doel. De doelen zijn:

- Merkbekendheid;
- Lokale bekendheid;
- Bereik;
- Verkeer;
- Betrokkenheid;
- Appinstallaties;
- Videoweergaven;
- Leads genereren;
- Conversies;
- Verkopen uit productcatalogus;
- Winkelbezoeken;

De meeste doelen zijn bijna vanzelfsprekend, betrokkenheid is wat moeilijker te begrijpen. Hierbij optimaliseert facebook de campagne voor interactie van de gebruiker, bijvoorbeeld in het commentaar.

2.2.4.2 Doelgroep

Net even belangrijk als het campagne doel is de doelgroep, dit zijn de gebruikers die de advertentie gaan zien. Hierbij zijn twee type doelgroepen, de koude doelgroep en de warme doelgroep. De koude doelgroep zijn de gebruikers die niet verbonden zijn met je bedrijf of product. Met een koude doelgroep is het niet de bedoeling om conversies te krijgen, hier wordt vooral de bekendheid mee vergroot en het genereren van leads.

De warme doelgroep daarentegen zijn de gebruikers die wel verbonden zijn met het bedrijf of product, of gebruikers die geretarget worden. Deze doelgroep kan wel gebruikt worden voor het generen van conversies.

Dit wil niet zeggen dat een koude doelgroep niet gebruikt kan worden voor conversies maar de CTR zal hierdoor lager zijn omdat de gebruiker nog niet klaar is voor het kopen van een product.

Ook geeft facebook de mogelijkheid om een doelgroep aan te maken op basis van een klantenbestand. Hiermee wordt de privacy van de gebruiker niet geschonden omdat de dataset die ingestuurd wordt eerst gehashed wordt, hierdoor krijgt Facebook geen nieuwe data binnen gestuurd en kunnen ze alleen maar controleren op hun dataset. Ongeveer 60% van een klantenbestand komt overeen de dataset van Facebook.

De mogelijkheden voor het maken van een doelgroep is eindeloos. Het is mogelijk om gebruikers te targeten via Demografie, plaats, interesses, gedrag en veel meer. [36] Het is ook mogelijk om een lookalike doelgroep te maken, hieruit wordt er gestart van uit een bestaande doelgroep. Bijvoorbeeld het klantenbestand, hierna gaat facebook andere gebruikers proberen te matchen met het klantenbestand en hier een nieuwe doelgroep voor aanmaken.

Bij de doelgroep is het ook mogelijk om gebruikers met bepaalde criteria uit te sluiten, bijvoorbeeld als de gebruiker al een product gekocht heeft.

2.2.4.3 Advertentie

Het type advertentie is ook heel belangrijk bij het maken van een succesvolle campagne, de advertentie moet namelijk aantrekkelijk en relevant zijn voor de gebruiker. Als een advertentie goed ontvangen wordt door de gebruikers is er een voordeel bij de veiling. De CPM is lager dan de advertentie goed ontvangen wordt omdat de gebruikers deze graag zien.

Facebook wilt hun platform 'fun' houden zelfs met de advertenties, het blijkt ook dat minder professionele advertenties beter presteren omdat het voor de gebruiker dan lijkt alsof vrienden dit gedeeld hebben.

Video advertenties zijn zeer goed voor koude doelgroepen, hierdoor leert de gebruiker het product kennen via beeld formaat, hierna kan er een warme doelgroep worden aangemaakt met de gebruikers die de video bekeken hebben.

Als een advertentie het goed doet en gebruikers beginnen deze te delen, wordt er ook organisch bereik gegenereerd, hiervoor moet niet betaald worden. Dit geeft de mogelijkheid om een groot bereik te genereren met een klein budget. Zo heeft Samsung een bereik gehaald van 500,000 gebruikers met een budget van \$ 100. [33]

2.2.4.4 A/B Testen

Het is belangrijk om verschillende advertenties te testen op verschillende doelgroepen met een klein budget, zo is het mogelijk om te zien welke advertenties goed werken op een bepaalde doelgroep. Facebook geeft hiervoor de relevantie score een cijfer op 10, dit cijfer wordt gemaakt nadat de advertentie 500 keer bekeken is geweest. Een cijfer tussen de 6-10 is goed onder de 6 wordt aangeraden om de campagne te stoppen en een nieuwe doelgroep of ander soort advertentie uit te proberen.

2.2.5 Facebook pixel

De facebook pixel is een analytics tool die de mogelijkheid biedt om het succes van de advertentie te meten door de acties die de gebruiker maakt op de website waar de pixel actief is. [37]

Het toevoegen van de pixel gebeurt door een stukje javascript code te kopiëren op de website waar de pixel actief moet zijn. De voordelen van de pixel zijn dat het gemakkelijk is om een doelgroep te maken van je website en het meet de resultaten van de advertenties.

2.3 Conclusie

Facebook biedt een tool aan die gemakkelijk te gebruiken is voor zowel de nieuwe adverteerder als de ervaren adverteerder. Vervolgens zijn de instellingen voor het doelpubliek eindeloos en kan elke facebook gebruiker bereikt worden, in tegenstelling met het adverteren op seachengines is het doelpubliek niet warm om producten te kopen omdat ze niet actief aan het zoeken zijn.

Hiervoor moet er eerst aandacht besteed worden aan het warm maken van het doelpubliek zodat er naderhand een conversie kan plaatsvinden. Facebook geeft ook heel snel een weergaven van hoe de advertentie presteert hierdoor kan er snel ingegrepen worden als de advertentie het niet goed doet. Adverteren op facebook is ook goedkoper en levert op lange termijn meer kliks en conversies op, wanneer een campagne gemaakt wordt, geeft Facebook ook weer hoeveel gebruikers de advertentie gaat bereiken en hoeveel kliks het gaat opleveren, deze cijfers zijn voor 90% van de advertenties correct.

Als een advertentie het goed doet en gebruikers beginnen deze te delen, wordt er ook organisch bereik gegenereerd, hiervoor moet niet betaald worden. Zo is het mogelijk om een heel groot bereik te krijgen op een klein budget.

Zoals bij elke marketing strategie is het testen van verschillende advertenties en doel publieke heel belangrijk om succesvol te zijn, elke advertentie is uniek en presteren niet altijd even goed. Ten laatste geeft de Facebook pixel heel veel controle over de statistieken van de campagne, het is mogelijk om te zien waar de gebruikers het meeste na toe gaan op je website en zo voort.

De Facebook ad manager is een zeer sterke tool en Facebook hun people-based targeting technologie is uniek en de toekomst van adverteren.

Bibliografie

- [1] Gift2Give, *persoonlijke communicatie*, Hasselt, 2017.
- [2] Gift2Give, „gift2give,” gift2give, 2017. [Online]. Available: <https://www.gift2give.be/>. [Geopend 2017].
- [3] Corda, „Welcome bij corda INCubator,” [Online]. Available: <http://www.cordacampus.com/nl/corda-incubator/welkom>. [Geopend 2017].
- [4] Gift2Give, „Brochure Gift2Give,” [Online]. Available: https://www.gift2give.be/uploads/documents/Brochure_Gift2Give.pdf. [Geopend 2017].
- [5] Bpost, „Parier: een hernieuwbare en duurzame bron?,” [Online]. Available: http://www.medattest.be/site/nl/postgroup/greenpost/docs/Papier_eeen_hernieuwbare_en_duurzame_bron_WWF.pdf. [Geopend 2017].
- [6] Git, Software Freedom Conservancy, „Git,” [Online]. Available: <https://git-scm.com/>. [Geopend 2017].
- [7] Wikipedia, the free encyclopedia, „Jira (software),” [Online]. Available: [https://en.wikipedia.org/wiki/Jira_\(software\)](https://en.wikipedia.org/wiki/Jira_(software)).
- [8] Wikipedia, the free encyclopedia, „Appcelerator Titanium,” [Online]. Available: https://en.wikipedia.org/wiki/Appcelerator_Titanium.
- [9] Appcelerator, „Alloy,” [Online]. Available: <https://github.com/appcelerator/alloy>.
- [10] Underscore.js, „Underscore.js,” Underscore.js, [Online]. Available: <http://underscorejs.org/>.
- [11] jasonkneen, „RESTe,” [Online]. Available: <https://github.com/jasonkneen/RESTe>.
- [12] Hansenmann, „Ti.Barcode,” [Online]. Available: <https://github.com/appcelerator-modules/ti.barcode>.
- [13] FokkeZB, „nl.fokkezb.toast,” [Online]. Available: <https://github.com/FokkeZB/nl.fokkezb.toast>.
- [14] Wikipedia, „PHP,” [Online]. Available: <https://en.wikipedia.org/wiki/PHP>. [Geopend 2017].
- [15] J. Gosling, B. Joy, G. Steele, G. Bracha en A. Buckley, „The Java® Language Specification,” 2014. [Online]. Available: <https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf>.
- [16] Wikipedia, the free encyclopedia, „Java (programming language),” Wikipedia, 2017. [Online]. Available: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)). [Geopend 2017].
- [17] The Apache Software Foundation, „Welcome to Apache Maven,” 25 11 2017. [Online]. Available: <https://maven.apache.org/>.
- [18] Pivotal Software, Inc, „Spring,” [Online]. Available: <https://spring.io/>. [Geopend 2017].

- [19] Amazon, „Amazon S3 Product Details,” [Online]. Available: <https://aws.amazon.com/s3/details/>. [Geopend 2017].
- [20] Lean Six Sigma.nl, „Wat is agile,” [Online]. Available: <https://www.sixsigma.nl/wat-is-agile>. [Geopend 2017].
- [21] Stackoverflow, „How to convert a byte[] to a BufferedImage in Java?,” 3 oct 2015. [Online]. Available: <https://stackoverflow.com/questions/12705385/how-to-convert-a-byte-to-a-bufferedimage-in-java>. [Geopend 2017].
- [22] Wikipedia, the free encyclopedia, „Universally unique identifier,” [Online]. Available: https://en.wikipedia.org/wiki/Universally_unique_identifier. [Geopend 2017].
- [23] Mollie, „Mollie API client for PHP,” 2017. [Online]. Available: <https://github.com/mollie/mollie-api-php>.
- [24] Paypal, „Paypal,” 2017. [Online]. Available: <https://developer.paypal.com/docs/api/payments/#payment>.
- [25] Facebook, „Send Dialog,” 2017. [Online]. Available: <https://developers.facebook.com/docs/sharing/reference/send-dialog>. [Geopend 2017].
- [26] Wikipedia, the free encyclopedia, „Cloud storage,” [Online]. Available: https://en.wikipedia.org/wiki/Cloud_storage.
- [27] Cloudinary, „Cloudinary,” [Online]. Available: <https://cloudinary.com/>. [Geopend 2017].
- [28] Cloudinary, „PRICING & PLANS,” Cloudinary, [Online]. Available: <https://cloudinary.com/pricing>. [Geopend 2017].
- [29] Neustar, „WHOIS IP Lookup Tool,” [Online]. Available: <https://www.ultratools.com/tools/ipWhoisLookupResult>.
- [30] Amazon, „What Is AWS Storage Gateway?,” [Online]. Available: <http://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html>. [Geopend 2017].
- [31] Wikipedia, „Advertising,” Wikipedia, 2018. [Online]. Available: <https://en.wikipedia.org/wiki/Advertising>. [Geopend 2018].
- [32] Facebook, „facebook ads reporting ads manager,” 2018. [Online]. Available: <https://www.facebook.com/business/learn/facebook-ads-reporting-ads-manager>.
- [33] geheimvandesmith, „Training Facebook Ads,” <https://geheimvandesmith.nl/>, Amsterdam, 2018.
- [34] Statista, „Forecast of Facebook user numbers in Belgium from 2015 to 2021 (in million users),” 2018. [Online]. Available: <https://www.statista.com/statistics/568747/forecast-of-facebook-user-numbers-in-belgium/>.
- [35] Facebook, „how ads show,” 2018. [Online]. Available: <https://www.facebook.com/business/products/ads/how-ads-show>.

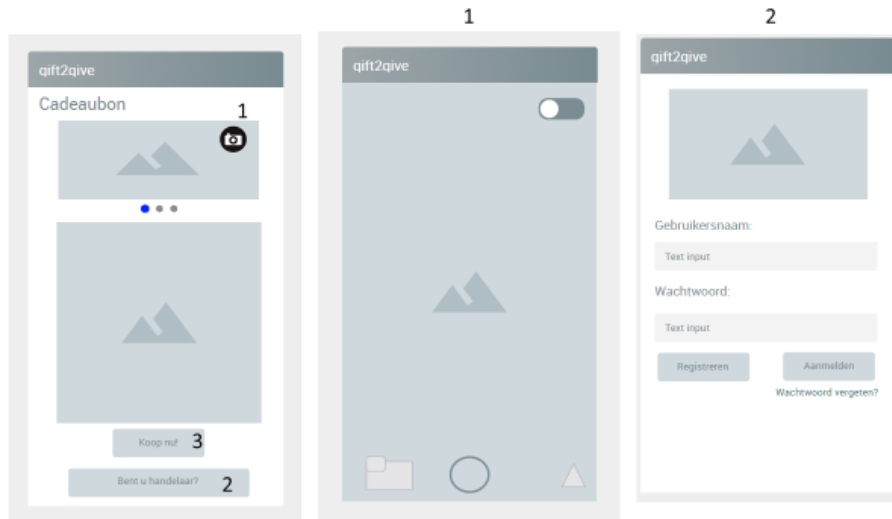
- [36] Facebook, „Choose your audience,” 2018. [Online]. Available: <https://www.facebook.com/business/products/ads/ad-targeting>.
- [37] Facebook, „About Facebook Pixel,” 2018. [Online]. Available: <https://www.facebook.com/business/help/742478679120153>.
- [38] Chart.JS, „Chart.JS,” [Online]. Available: <http://www.chartjs.org/>.

Bijlage

A. Mockups

A. Mockups

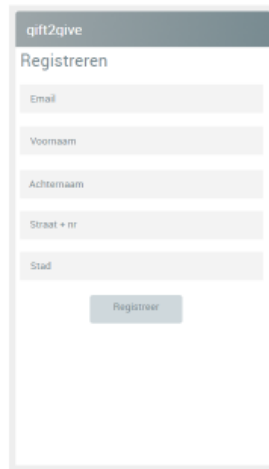
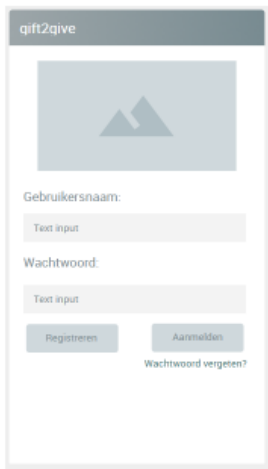
Index



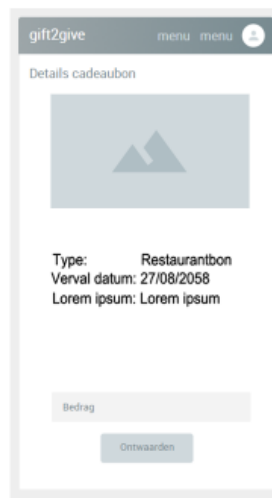
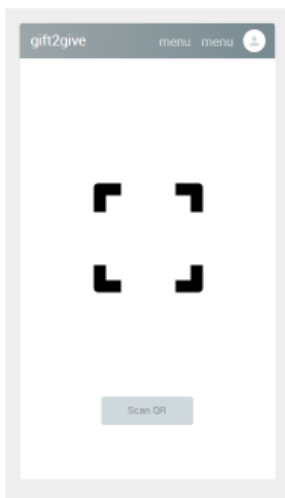
Betaling



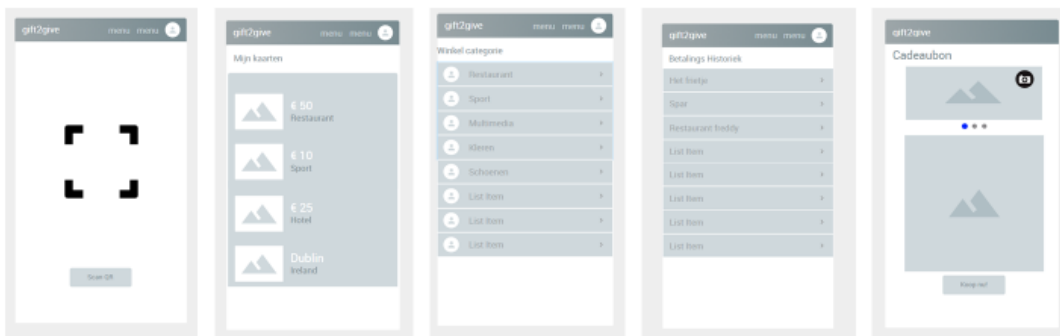
Register



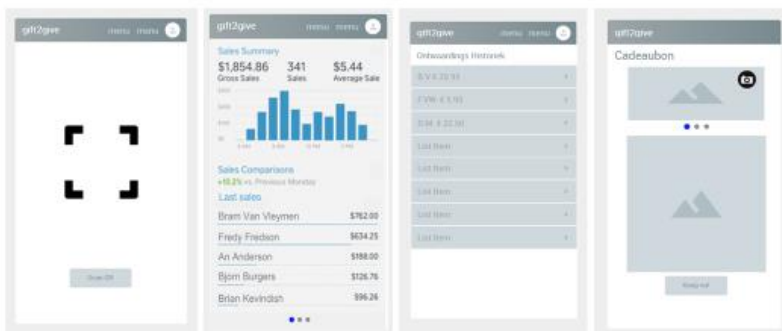
Qr scan



Tab menu (user)



Tab menu (handelaar)



Winkels

