

Odisee Hogeschool Gent  
Gebroeders De Smetstraat 1, 9000 Gent

EASYSHOP

Onderzoek en ontwikkeling van een winkelapplicatie

**Auteurs**

Niels De Bruyne, Piet Jacobs, Seppe Snoeck, Daan Zwaenepoel

**Mentoren**

Professionele Opleiding EO-ICT Sabine Martens Academiejaar 2016-2017 Katja Verbeeck



Odisee Hogeschool Gent  
Gebroeders De Smetstraat 1, 9000 Gent

EASYSHOP

Onderzoek en ontwikkeling van een winkelapplicatie

**Auteurs**

Niels De Bruyne, Piet Jacobs, Seppe Snoeck, Daan Zwaenepoel

**Mentoren**

Professionele Opleiding EO-ICT Sabine Martens Academiejaar 2016-2017 Katja Verbeeck

INHOUD

[Figurenlijst 3](#_Toc481537949)

[Tabellenlijst 4](#_Toc481537950)

[Codefragmentenlijst 5](#_Toc481537951)

[Afkortingenlijst 6](#_Toc481537952)

[Begrippenlijst 7](#_Toc481537953)

[Inleiding 8](#_Toc481537954)

[1 Mogelijke oplossingen 9](#_Toc481537955)

[1.1 Marktonderzoek 9](#_Toc481537956)

[1.1.1 Bespreking enquête 9](#_Toc481537957)

[1.1.2 Besluitvorming design webinterface 12](#_Toc481537958)

[1.2 Programmeertalen 12](#_Toc481537959)

[1.2.1 Frontend 12](#_Toc481537960)

[1.2.2 API 13](#_Toc481537961)

[1.3 Betaalmethoden 13](#_Toc481537962)

[1.4 Cloudoplossingen 14](#_Toc481537963)

[2 Gekozen oplossing 15](#_Toc481537964)

[2.1 Programmeertalen 15](#_Toc481537965)

[2.2 Betaalmethoden 15](#_Toc481537966)

[2.3 Cloudoplossing 15](#_Toc481537967)

[3 Technische uitwerking 16](#_Toc481537968)

[3.1 Bespreking hardware 16](#_Toc481537969)

[3.2 Bespreking software 16](#_Toc481537970)

[4 Risicoanalyse 17](#_Toc481537971)

[4.1 Clientside injection 17](#_Toc481537972)

[4.2 Risicovolle datatransmissie 17](#_Toc481537973)

[4.3 Betrouwbaarheid betaalmethoden 17](#_Toc481537974)

[5 Kostenraming en levensduur 18](#_Toc481537975)

[5.1 Totale kosten 18](#_Toc481537976)

[5.2 Levensduur software 18](#_Toc481537977)

[5.3 Levensduur hardware 18](#_Toc481537978)

[Conclusie 19](#_Toc481537979)

[Nawoord met kritische reflectie 20](#_Toc481537980)

[Literatuurlijst 21](#_Toc481537981)

[Bijlagenoverzicht 22](#_Toc481537982)

[Bijlage 1: Enquêtes 22](#_Toc481537983)

[Bijlage 2: … 22](#_Toc481537984)

# Figurenlijst

Tekst

# Tabellenlijst

Tekst

# Codefragmentenlijst

Tekst

# Afkortingenlijst

Tekst

# Begrippenlijst

Tekst

# Inleiding

Volgend project wordt uitgevoerd in opdracht van Projecten 1, onderdeel van fase twee in de opleiding elektronica-ICT aan de hogeschool Odisee Gent. Het omvat een webapplicatie genaamd Easyshop die als doel heeft om boodschappen te laten bezorgen door een andere gebruiker. Easyshop heeft twee doelgroepen. De eerste groep zijn de mensen die behoefte hebben aan thuislevering van hun boodschappen. De tweede groep zijn de mensen die de boodschappen aan huis leveren tegen een vergoeding.

Het is mogelijk om een boodschappenlijst te plaatsen, waarna een bezorger de aanvraag kan accepteren en uitvoeren. Wanneer een bezorger een ticket accepteert dan ontstaat er een ‘contract’ met de persoon die het ticket heeft geplaatst. Er zal gewerkt worden met een stappenplan waarbij de boodschappen, winkel en het tijdstip van de levering worden vastgelegd. Nadien levert de bezorger de boodschappen aan huis. Easyshop zal gebruik maken van een onlinebetaalservice. Deze service zal moeten voldoen aan de normen van de Payment Card Industry Data Security Standard.

Easyshop wordt van scratch gemaakt en moet aan een aantal voorwaarden voldoen. Zo is een eerste voorwaarde de financiële veiligheid. Er moet op een veilige en liefst zo goedkoop mogelijke manier geld overgemaakt kunnen worden tussen drie verschillende partijen namelijk de gebruikers, de bezorgers en Easyshop. Een tweede voorwaarde is het vertrouwen in het betaalsysteem. Om dit te bevorderen maakt Easyshop gebruik van online betrouwbare betaalsystemen van externe services. Een volgende voorwaarde is het opbouwen van een vertrouwensrelatie met de gebruikers. De webinterface moet professioneel ogen om een eerste goede indruk te geven aan potentiële klanten en daarnaast moet er een controle zijn op het correct handelen van alle gebruikers.

De applicatie wordt gerealiseerd door enerzijds duidelijke gesprekken te voeren met de begeleidende technische docent en door anderzijds een literatuurstudie te doen over de norm volgens de Payment Card Industry Data Security Standard. Daarnaast moet er onderzoek verricht worden aan de hand van enquêtes naar de noden van de doelgroepen om deze zo goed mogelijk in te vullen. Dergelijk ontwerp van de webapplicatie ligt in de lijn van de kennis en vaardigheden reeds verworven in user-experience-design. Alvorens het programmeren moeten eerst enkele mogelijke programmeertalen afgewogen worden. Na het maken van de meest gunstige keuze wordt er tijd gespendeerd om de code zo robuust mogelijk te maken. De code zal dan ook vaak getest worden om aan deze voorwaarde te voldoen. Ook zijn er voldoende onlinebronnen voorhanden om tot mogelijke oplossingen te komen.

Er wordt een periode van één maand vanaf 1 februari 2017 voorzien om het onderzoek klaar te hebben. Voor de realisatie van de applicatie worden nog eens drie maanden gerekend.

Het rapport start met een opsomming van de mogelijke oplossingen. Na afweging van de voor- en nadelen wordt er een oplossing gekozen. Deze zal dan stap voor stap beschreven worden in de technische uitwerking. Nadien zal een risicoanalyse volgen die de veiligheid onder de loep neemt. Daaropvolgend wordt de kostenraming besproken. Tot slot wordt het volledige rapport samengevat in de conclusie en volgt er nog een nawoord met kritische reflectie.

# Mogelijke oplossingen

## Marktonderzoek

In dit onderdeel wordt het marktonderzoek besproken. Er wordt gebruik gemaakt van een enquête om informatie te verzamelen over de noden van mogelijke eindgebruikers. Deze informatie heeft een impact op de opbouw en mogelijkheden van de webinterface.

### Bespreking enquête

De enquête bestaat uit 16 vragen die het koopgedrag op het internet nagaat van 80 willekeurige personen. Hieronder worden een aantal verkregen antwoorden, die een weerslag kunnen hebben op het eindresultaat, grafisch voorgesteld en besproken.

De enquête is ingevuld door een grote variatie leeftijdsgroepen. Niemand is echter jonger dan 16 jaar (figuur 1). Zo’n 60% van de bevraagden zijn 40 of jonger, de overige 40% is dan vanzelfsprekend ouder dan 40, waarvan maar 3% ouder is dan 70 jaar.

Hieruit kan besloten worden dat boodschappen, al dan niet online, door alle leeftijdsgroepen gedaan worden.

Figuur 1: Leeftijd van de bevraagden

Uit de enquête blijkt ook dat 43% van de mensen gebruik maakt van een computer om te surfen op het internet (figuur 2). Een kleinere groep van 37% maakt vooral gebruik van een smartphone en 20% gebruikt een tablet om te surfen.

Om een zo breed mogelijk publiek aan te spreken, wordt er gekozen voor het ontwikkelen van een webinterface. Op deze manier kunnen gebruikers via alle mogelijke toestellen die over internet beschikken terecht op de webshop.

Figuur 2: Gebruik toestellen om te surfen op het internet

In de enquête is ook duidelijk geworden dat de meeste bevraagden een bakker, slager of een klein buurtwinkeltje beschikbaar hebben binnen een straal van 0 tot 3 kilometer (figuur 3). Om boodschappen in grotere hoeveelheden te kopen, moeten velen meer dan 3 kilometer afleggen.

Figuur 3: Afstand naar de winkel (in kilometer)

Er is ook gevraagd naar hoe lang iemand er over doet om boodschappen te doen (figuur 4). Zo’n 39% doet er minder dan 30 minuten over terwijl 30% er 30 tot 60 minuten over doet. Dan zijn er nog twee minderheidsgroepen, waarvan 22% er 60 tot 90 minuten over doet en 8% 90 tot 120 minuten. Er is ook één gebruiker die één maal per maand boodschappen doet en er meer dan 120 minuten over doet.

Easyshop kan de tijd dat mensen spenderen aan het doen van boodschappen verminderen, zo hoeven gebruikers zich niet meer te verplaatsen naar winkels die vaak op afstanden van meer dan 3 kilometer liggen.

Figuur 4: Tijdspanne van 1 boodschappensessie (in minuten)

Het gebruik van onlinebetaalmethoden is bij de bevraagden zeer eenduidig (figuur 5). Maar liefst 70% maakt gebruik van Bancontact. Alternatieve betaalmethoden worden aanzienlijk minder gebruikt.

Figuur 5: Gebruik onlinebetaalmethoden

### Besluitvorming design webinterface

Ten eerste moet de interface gebruiksvriendelijk zijn en zeker niet complex, het is tenslotte de bedoeling dat mensen van alle leeftijden gebruik kunnen maken van de webinterface om boodschappen te doen of hun diensten als leverancier aan te bieden. Een volgende belangrijke pijler is dat de webinterface op alle toestellen met internettoegang moet kunnen geopend worden zodat er geen potentiële gebruikers uitgesloten worden. Ten derde is boodschappen doen tijdrovend, zeker wanneer de winkels op grotere afstanden liggen, Easyshop moet snel en eenvoudig zijn om tijd te besparen. Tot slot moeten er voldoende betaalmogelijkheden beschikbaar zijn voor de potentiële gebruikers, een minimumvereiste is het implementeren van Bancontact.

## Programmeertalen

Voor deze toepassing wordt er gekozen voor een webinterface zoals hierboven besproken. In dit hoofdstuk worden er een aantal mogelijke programmeertalen vooropgesteld om deze webinterface op te implementeren.

### Frontend

Voor het aanmaken van een webpagina zijn er twee programmeertalen van essentieel belang, namelijk HyperText Markup Language (HTML) en Cascading Style Sheets (CSS). Beide vormen de basis van een eenvoudige webpagina, waarbij HTML zorgt voor de nodige structuur, terwijl CSS de opmaak verzorgt. [1] [2]

Deze basis volstaat echter niet om aan alle nodige functionaliteiten van Easyshop te voldoen. Hiervoor wordt er gebruik gemaakt van JavaScript, dit is een programmeertaal waarmee men het gedrag van een website kan programmeren. De mogelijkheden van JavaScript zijn echter eindeloos en worden snel complex. [1] Om het gebruik van deze krachtige programmeertaal te vereenvoudigen zijn er verschillende frameworks op deze taal gebaseerd, elk met een unieke syntax en functies die bepaalde opdrachten gemakkelijker maken. De webinterface zal dan ook gebruik maken van één of meerdere frameworks die de nodige tools aanbieden om te voldoen aan die doelstellingen. In volgende tabel zijn hiervoor een aantal mogelijke frameworks opgesomd die tegen elkaar worden afgewogen.

Tabel 1: Enkele eigenschappen van enkele mogelijke frameworks [3] [4]

|  |  |  |  |
| --- | --- | --- | --- |
|  | React | Angular2 | Polymer |
| Ontwikkelaar | Facebook | Google | Google |
| Hoofddoel | Dynamische content | Beheren van data en de paginaroutering | Gebruiksvriendelijk en performant |
| Community | Heel groot, veel documentatie en codevoorbeelden | Heel groot, veel documentatie en codevoorbeelden | Groot, voldoende documentatie en codevoorbeelden |

In bovenstaande tabel (tabel 1) zijn er een aantal belangrijke eigenschappen opgesomd van enkele populaire frameworks die bruikbaar zijn voor dit project. Naast deze frameworks zijn er nog veel die de nodige tools aanbieden. [5] Hier is de vergelijking beperkt tot de meest populaire frameworks, omdat deze het best gedocumenteerd zijn en voldoende codevoorbeelden bevatten. Ook worden deze in bijna alle moderne browsers ondersteund, wat van belang is om zo’n groot mogelijke doelgroep aan te spreken. Easyshop is een webshop, de pagina-inhoud wordt dan ook het best op een dynamische manier weergegeven. Omwille van deze reden wordt er voor React gekozen binnen dit project.

### API

Een Application Programming Interface (API) zorgt voor de communicatie tussen de verschillende lagen van de website en de achterliggende database. [6] Voor het opbouwen van de API zijn er opnieuw een aantal mogelijkheden, waarvan enkele belangrijke eigenschappen in onderstaande tabel worden weergegeven.

Tabel 2: Mogelijke programmeertalen voor het opbouwen van de API [7] [8]

|  |  |  |
| --- | --- | --- |
|  | Php | Python |
| Laag | Backend | Backend |
| LEEFTIJD | Oud (1995), heel matuur, allomtegenwoordig | Oud (1991) en nog steeds het meest populair |
| Frameworks | Laravel, CodeIgniter… | Django, Flask… |
| Complexiteit | Eenvoudig en snel | Heel eenvoudig en minimalistisch |

In de tabel zijn enkele belangrijke eigenschappen van de twee meest gebruikte programmeertalen weergegeven waarmee de API kan opgebouwd worden. Ze zijn beide voorzien van een uitgebreide documentatie met codevoorbeelden. Voor dit project wordt er gekozen voor NodeJS in samenwerking met Python. NodeJS verwerkt data langs de frontend aan de hand van een dynamische pagina. Het is een relatief nieuwe taal dat frequent gebruikt wordt sinds 2009. Hierdoor is het een taal die veel moderne features bevat en een moderne en relatief eenvoudige syntax heeft. De communicatie met de database wordt dan via een Python-oplossing verzorgd. Python is het doorgeefluik die voor de uitwisseling van de data zorgt tussen NodeJS en de database.

## Betaalmethoden

Om betalingen te doen moeten er enkele betaalmethoden voorzien worden. Hieronder worden er enkele mogelijke providers vergeleken met de aangeboden betaalmethoden.

Tabel 3: Mogelijke onlinebetaalmethoden [9]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Paypal | Multisafepay | Ogone | MOLLIE |
| Betaalmethoden | Paypal, Visa, Bancontact… | Visa, MasterCard, iDeal, Bancontact… | Visa, MasterCard, Bancontact… | Bancontact, Visa, Paypal… |
| Prijs | 3,4% + €0,35 per transactie | Opstartkost van €75, maandelijkse kost van €45 en transactiekosten tussen de €0,15 en 2.9% + €0,15 | Opstartkosten van €300, maandelijkse kost van €65 en transactiekost van €0,37 (Fraudedetectie kost extra) | Tussen €0,25 en €15% van de transactie |
| Eenvoud | Gebruikers moeten een account aanmaken | Gebruiksvriendelijk | Gebruiksvriendelijk | Gebruiksvriendelijk |

Er zijn tal van providers die bijna alle mogelijke betaalmethoden aanbieden. De ene biedt al wat meer features aan dan de andere en ook prijzen variëren. Omdat dit project gelimiteerd is in het maken van kosten, wordt er dus gekozen voor de goedkoopste oplossing. Dit zorgt voor een basisimplementatie van enkele betaalmethodes, in een latere fase van het project kan nog steeds gekozen worden voor een provider met meer features indien hier nood aan is.

## Cloudoplossingen

Om de website bereikbaar te maken voor iedereen moet deze gehost worden. Hieronder worden er enkele mogelijk cloudoplossingen besproken om de website te hosten.

Een eerste mogelijke cloudoplossing is cPanel, dit wordt door ikdoeict.be aangeboden voor studentenprojecten, zodat ze zelf niet moeten investeren in de nodige hardware. Het heeft een gebruiksvriendelijke webinterface met uitgebreide features en goede ondersteuning. [10]

Een andere mogelijke oplossing is GitHub Pages. Dit is volledig gratis, maar daarom ook beperkt in functionaliteit. Voor dit project is het dan ook niet bruikbaar aangezien GitHub Pages enkel in staat is statische pagina’s te hosten, Easyshop daarentegen maakt gebruik van een backend. [11]

Een laatste mogelijke oplossing die in dit hoofdstuk besproken wordt, is Digital Ocean. Afhankelijk van de benodigde features en hardware kan er gekozen worden tussen verschillende pakketten, elk met een eigen prijskaartje. Het maakt gebruik van een gebruiksvriendelijke webinterface waar alle instellingen terug te vinden zijn. Een groot voordeel van Digital Ocean is dat er heel gemakkelijk internationale servers toegevoegd kunnen worden indien de webinterface internationaal beschikbaar moet worden gemaakt. [12]

# Gekozen oplossing

De gekozen oplossing bestaat uit een combinatie van concepten die betrekking hebben tot de gekozen programmeertalen, de betaalmethoden en de cloudoplossing voor het bouwen van de webinterface. Bij het kiezen van de uiteindelijke oplossing wordt er rekening gehouden met de noden van de eindgebruikers die in de enquête aangehaald zijn, maar ook met de meest gunstige en voordelige oplossing voor de projectleiders.

## Programmeertalen

De programmeertalen zijn essentieel voor het implementeren van de webinterface. Er is gekozen voor HTML om de nodige webstructuur te vormen in combinatie met CSS voor het opmaken van de pagina.

Om het geheel dynamisch en programmeerbaar te maken, wordt er gebruik gemaakt van React, een krachtig JavaScript framework dat het mogelijk maakt om pagina’s dynamisch bij te werken. Dit zal samen met NodeJS de frontend van de pagina verzorgen. NodeJS verwerkt dan de data langs de frontend en geeft deze door aan de backend die gebruik maakt van Python.

Python wordt gebruikt voor de implementatie van de API, deze zorgt dan weer voor de communicatie tussen de frontend en de achterliggende database. Deze laatste zal gebruik maken van MySQL in combinatie met de InnoDB engine omdat deze voor een goede beveiliging van de data zorgt.

## Betaalmethoden

De betaalmethoden hangen vast aan een bepaalde provider. Dit project maakt gebruik van Mollie, één van de goedkoopste providers voor de betaalmogelijkheden die Easyshop zal aanbieden. Bovendien zijn er geen opstartkosten aan deze diensten verbonden. [13]

Per Bancontact-transactie rekent Mollie €0,25 + 1,5%, voor PayPal geldt €0,10 + PayPal-aandeel, Mastercard en Visa kosten €0,25 + 1.8% voor Europese betalingen. Mollie voorziet ook een API-implementatie voor NodeJS en Python waar de webinterface gebruik zal van maken voor betalingen.

## Cloudoplossing

Als cloudoplossing wordt er gekozen voor de oplossing van Digital Ocean, het is een goedkope oplossing die veel te bieden heeft en vooral flexibel is met uitbreidingsmogelijkheden. [12] Het is een zeer respectabel bedrijf dat garant staat voor hoge beschikbaarheid en betrouwbaarheid. Bovendien kunnen studenten gratis gebruik maken van hun services.

Voor dit project wordt er gestart met een eenvoudige linuxbox die volledig naar eigen hand kan worden gezet door middel van uitgebreide configuratiemogelijkheden en verschillende tools die beschikbaar zijn via de eenvoudige webinterface van Digital Ocean.

# Technische uitwerking

Andere indeling (subtitels)?

Vermelden en opbouwen vanuit de enquête?

Technische uitwerking API verschillend van gekozen oplossing? 🡪 aanpassen?

Onderschrift bijlagen op het einde van het document?

Codefragmentenlijst, figurenlijst, tabellenlijst, afkortingenlijst, begrippenlijst etc… Hoe zien deze eruit? Met of zonder afbeeldingen?

## Bespreking hardware

Linux-server voor database en webhost met de juiste software voorzieningen: ...

Hoort dit dan al bij software?

## Bespreking software

Programmeertalen, structuur, opbouw, communicatie tussen de verschillende lagen...  
Grafische aspect

...

In het tweede hoofdstuk werd er gekozen om een de API te implementeren met Python. Na onderzoek is gebleken dat dit echter niet haalbaar is. Er was geen tool te vinden die het mogelijk maakte om de API te genereren, dit is noodzakelijk wegens de vooropgestelde termijn. Een volledige API zelf schrijven voor deze applicatie is zeer tijdrovend. Hierdoor is er gekozen voor een eenvoudigere aanpak met php. In deze programmeertaal zijn er wel verschillende tools beschikbaar die het mogelijk maken om alle mogelijke API-calls te genereren. Dit bespaart veel tijd, die gebruikt wordt voor de verdere ontwikkeling van de applicatie te.

# Risicoanalyse

Tekst

## Clientside injection

## Risicovolle datatransmissie

## Betrouwbaarheid betaalmethoden

Grotere spelers vs goedkope kleine providers

# Kostenraming en levensduur

## Totale kosten

## Levensduur software

## Levensduur hardware

# Conclusie

Tekst

# Nawoord met kritische reflectie

Tekst

# Literatuurlijst

|  |  |
| --- | --- |
| [1] | Bron 1 |
| [2] | Bron 2 |

# Bijlagenoverzicht

## Bijlage 1: Enquêtes

## Bijlage 2: …