

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](#_Toc477344206)

[Tabellenlijst 4](#_Toc477344207)

[Codefragmentenlijst 5](#_Toc477344208)

[Afkortingenlijst 6](#_Toc477344209)

[Begrippenlijst 7](#_Toc477344210)

[Inleiding 8](#_Toc477344211)

[1 Mogelijke oplossingen 10](#_Toc477344212)

[1.1 Marktonderzoek 10](#_Toc477344213)

[1.1.1 Bespreking enquête 10](#_Toc477344214)

[1.1.2 Besluitvorming design webinterface 10](#_Toc477344215)

[1.2 Programmeertalen 10](#_Toc477344216)

[1.2.1 Frontend 10](#_Toc477344217)

[1.2.2 Backend 10](#_Toc477344218)

[1.3 Betaalmethoden 10](#_Toc477344219)

[1.4 Cloudoplossingen 10](#_Toc477344220)

[2 Gekozen oplossing 11](#_Toc477344221)

[2.1 Programmeertalen 11](#_Toc477344222)

[2.2 Betaalmethoden 11](#_Toc477344223)

[2.3 Cloudoplossing 11](#_Toc477344224)

[3 Technische uitwerking 12](#_Toc477344225)

[3.1 Bespreking hardware 12](#_Toc477344226)

[3.2 Bespreking software 12](#_Toc477344227)

[4 Risicoanalyse 13](#_Toc477344228)

[4.1 Clientside injection 13](#_Toc477344229)

[4.2 Risicovolle datatransmissie 13](#_Toc477344230)

[4.3 Betrouwbaarheid betaalmethoden 13](#_Toc477344231)

[5 Kostenraming en levensduur 14](#_Toc477344232)

[5.1 Totale kosten 14](#_Toc477344233)

[5.2 Levensduur software 14](#_Toc477344234)

[5.3 Levensduur hardware 14](#_Toc477344235)

[Conclusie 15](#_Toc477344236)

[Nawoord met kritische reflectie 16](#_Toc477344237)

[Literatuurlijst 17](#_Toc477344238)

[Bijlagenoverzicht 18](#_Toc477344239)

[Bijlage 1: Enquêtes 18](#_Toc477344240)

[Bijlage 2: … 18](#_Toc477344241)

# 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 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 werd gebruik gemaakt van een enquête om informatie te verzamelen over de noden van mogelijke eindgebruikers. Deze informatie zal een impact hebben op de opbouw en mogelijkheden van de webinterface.

### Bespreking enquête

Hieronder worden de verkregen antwoorden op enkele vragen uit de enquête grafisch voorgesteld en besproken.

Figuur 1: Gebruik toestellen om te surfen op het internet

Uit de enquête blijkt dat 48% van de mensen gebruik maakt van een computer om te surfen op het internet. Een kleinere groep van 34% maakt vooral gebruik van een smartphone en 18% gebruikt een tablet om te surfen. Omwille van het grote aantal computergebruikers wordt er gekozen voor het ontwikkelen van een webinterface. Deze kan dan ook op elk toestel geopend worden zonder enig probleem.

(Hier komt nog een uitgebreid verslag van de enquête met grafieken en uitleg, maar de enquête wordt momenteel nog volop rondgestuurd)

### Besluitvorming design webinterface

(Enkele belangrijke doelstellingen en pijlers die zeker aanwezig moeten zijn in de webinterface)

## Programmeertalen

Voor deze toepassing wordt er gekozen voor een webinterface, wat reeds besproken werd. In dit hoofdstuk worden er een aantal mogelijke programmeertalen besproken om deze webinterface op te bouwen.

### 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 zal echter niet volstaan om aan alle nodige functionaliteiten van Easyshop te voldoen. Hiervoor zal er gebruik gemaakt worden 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 maakt. 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.

|  |  |  |  |
| --- | --- | --- | --- |
|  | React | Angular2 | Polymer |
| Ontwikkelaar | Facebook | Google | Google |
| Hoofddoel | Dynamische content | Beheren van data en de paginaroutering | Gebruiksvriendelijk en peformance |
| Community | Heel groot, veel documentatie en codevoorbeelden | Heel groot, veel documentatie en codevoorbeelden | Groot, voldoende documentatie en codevoorbeelden |
| Browser support | Meest gebruikte browsers | Meest gebruikte browsers | Meest gebruikte browsers |

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

In bovenstaande tabel 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 aanbienden. [5] Hier is de vergelijking beperkt tot de meest populaire framework, omdat deze het best gedocumenteerd zijn en voldoende codevoorbeelden bevatten. Easyshop is een webshop en deze pagina-inhoud is dan liefst ook zeer dynamisch. 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, enkele belangrijke eigenschappen worden in onderstaande tabel weergegeven.

|  |  |  |  |
| --- | --- | --- | --- |
|  | NodeJS | Php | Python |
| Laag | Frontend | Backend | Backend |
| LEEFTIJD | Nieuwer, meer moderne features | Oud, heel matuur, allomtegenwoordig | Oud en nog steeds het meest populair |
| Community | Heel groot, veel documentatie en codevoorbeelden | Heel groot, veel documentatie en codevoorbeelden | Heel groot, veel documentatie en codevoorbeelden |
| Frameworks | Hapi.js, Socket.io… | Laravel, CodeIgniter… | Django, Flask… |
| Complexiteit | Relatief eenvoudig en moderne syntax | Eenvoudig en snel | Heel eenvoudig en minimalistisch |

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

In de tabel zijn enkele belangrijke eigenschappen van enkele programmeertalen weergegeven waarmee er de API kan opgebouwd worden. Voor dit project wordt er gekozen voor een NodeJS in samenwerking met Python. NodeJS zal aan de hand van een dynamische pagina langs de frontend data verwerken. De communicatie met de database wordt dan via een Python-oplossing verwerkt.

## Betaalmethoden

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 (Fraude detectie kost extra) | Tussen €0,25 en €15% van de transactie |
| Eenvoud | Gebruikers moeten een account aanmaken | Gebruiksvriendelijk | Gebruiksvriendelijk | Gebruiksvriendelijk |

Tabel 3: Mogelijke online betaalmethoden [9]

Er zijn tal van providers die bijna alle mogelijke betaalmethoden aanbieden. De ene biedt al wat meer features aan dan de ander. Voor dit project zal er gekozen worden voor de meest gunstige prijzen, omdat dit project gelimiteerd is in kosten.

## 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 Odisee Hogeschool zelf 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, mits Easyshop gebruik maakt van een backend. [11] Een laatste oplossing die in dit hoofdstuk wordt besproken is Digital Ocean, afhankelijk van de benodigde features en hardware kan er gekozen worden tussen verschillende pakketten, elk met een eigen prijs. Het maakt gebruik van een gebruiksvriendelijke webinterface waar alle instellingen terug te vinden zijn. Een groot voordeel bij Digital Ocean is het feit dat er heel gemakkelijk internationale servers kunnen toegevoegd kunnen worden indien dit ooit nodig zou zijn. [12]

# Gekozen oplossing

Tekst

## Programmeertalen

Tekst

## Betaalmethoden

Tekst

## Cloudoplossing

# Technische uitwerking

Tekst

## Bespreking hardware

## Bespreking software

# Risicoanalyse

Tekst

## Clientside injection

## Risicovolle datatransmissie

## Betrouwbaarheid betaalmethoden

# 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: …