

# Afstudeerscriptie

## Realisatie informatieplatform iStip

René van Donselaar, studentnummer: 1537722

Information Engineering – Hogeschool Utrecht

### **Coaches:**

*Roald van Loon*

*Derk-Jan Winkel*

### **Eerste examiner:**

*Derk-Jan Winkel*

### **Tweede examiner:**

*Benny de Waal*



14 12 2010

## Voorwoord

Deze scriptie is geschreven in het kader van mijn afstudeerstage bij Dalen Business van September 2010 t/m Januari 2011. In de scriptie wordt verslag gedaan van de realisatie van het informatieplatform iStip, waarmee inzicht kan worden verkregen in de aanpak die genomen kan worden om een informatieplatform te realiseren als webapplicatie.

Deze periode is afsluitend geweest voor mijn bachelor studie Information Engineering aan de Hogeschool Utrecht.

Mijn dank gaat uit naar Dalen Business en in het bijzonder Gijsbert van Dalen die mij de kans heeft gegeven om binnen deze organisatie mijn afstudeerstage te lopen. Ook wil ik mijn bedrijfsbegeleider Roald van Loon bedanken voor zijn goede begeleiding en het opzetten van mijn werkomgeving tijdens dit project.

Binnen Hogeschool Utrecht wil ik graag mijn docentbegeleider Derk-Jan Winkel bedanken voor zijn begeleiding in deze periode.

René van Donselaar

December 2010



## Managementsamenvatting

Deze scriptie beschrijft het verloop van mijn afstudeerproject tijdens mijn stage bij Dalen Business van September 2010 tot Januari 2011.

Het project omvatte het creëren van een testomgeving voor het iStip platform.

iStip is een platform voor internationale studenten waar zij informatie kunnen vinden over het land of de stad waar zij willen studeren. Het platform voorziet erin dat alle partijen die nodig zijn voor een stage of studie in het buitenland gekoppeld zijn aan het platform waardoor de informatie centraal aangeboden kan worden. Het platform wordt beschikbaar gesteld in de vorm van een website waarbij de nadruk ligt op het integreren van bestaande functionaliteiten en informatie.

Het was mijn taak om ervoor te zorgen dat er keuzes werden gemaakt omtrent de techniek waarop het platform zou gaan draaien en om het platform gedeeltelijk te realiseren.

Om een goede keuze voor de techniek te kunnen maken heb ik mij zoveel mogelijk verdiept in de mogelijkheden die er zijn. Vervolgens is elke techniek intern besproken en beargumenteerd alvorens er een definitief besluit werd genomen.

De voornaamste functionaliteit die gerealiseerd moest worden was het publiceren van informatie op het platform.

Ik heb het publicatieproces in kaart gebracht in met een zogeheten activiteiten diagram.

Omdat het niet een uitbreiding was, maar het compleet van de grond af opbouwen van de applicatie, heb ik mij als eerst bezig gehouden met het opzetten van het platform in technische zin. Door een PHP framework te implementeren en uit te breiden tot het geschikt was voor het realiseren van het platform, heb ik ervoor gezorgd dat het platform en de toekomstige functionaliteiten gerealiseerd konden worden.

Omdat de realisatie van het platform parallel liep met het aantrekken van nieuwe partners was het van belang dat er snel een interface zou worden gebouwd waarmee een tijdelijke, en uiteindelijke het definitieve platform, kon worden gebouwd.

De interface ben ik gaan ontwerpen met usability in het achterhoofd en heb ik uiteindelijk ook getest door middel van usability testing.

Met behulp van wireframes heb ik de interface vormgegeven waarna ik er een werkend prototype van heb gemaakt.

Vanuit het prototype is de uiteindelijke testversie gerealiseerd.

Het project bleek een grote persoonlijke uitdaging vanwege de grote hoeveelheid programmeerwerk en de diepgaande techniek die ik niet vanuit mijn opleiding heb meegekregen. Juist om die reden is het een interessant project geworden met een zichtbaar resultaat dat ook in de toekomst nog flink uitgebreid zal worden.



**Inhoud**

Voorwoord .....	1
Managementsamenvatting .....	2
1. Achtergrond.....	5
2. Probleem analyse .....	5
2.1 Probleemstelling.....	5
2.2 Belanghebbenden .....	6
2.3 Diagnose .....	6
3. Opdrachtomschrijving .....	6
4. Aanpak.....	7
4.1 Software ontwikkeling.....	7
4.2 Usability .....	7
4.3 Projectbeheersing .....	8
4.4 Procesanalyse .....	8
4.5 Onderzoeksmethodes .....	8
5. Organisatie .....	9
5.1 Uitgangssituatie.....	9
6. Gebruikte technieken .....	10
6.1 Randvoorwaarden .....	10
6.2 Framework of CMS .....	11
6.2.1 Criteria .....	11
6.2.2 Methodes .....	11
6.2.3 Technieken .....	12
6.2.4 Weging en keuze .....	14
6.2.5 Conclusie .....	14
6.3 Keuze voor php-framework.....	15
6.3.1 Criteria .....	15
6.3.2 Methodes .....	15
6.3.3 Techieken .....	16
6.3.4 Weging en keuze .....	20
6.3.5 Conclusie .....	21
6.4 Keuzes binnen de afhandeling van MVC .....	22
6.4.1 CodeIgniter alleen is niet goed genoeg .....	22
6.4.2 Keuze voor ORM.....	23



6.4.3	Keuze voor template engine .....	27
7.	Projectbeheersing .....	30
7.1	Scrum.....	30
7.2	Prince2.....	32
7.3	Keuze .....	34
7.4	Conclusie .....	34
8.	Usability .....	35
8.1	Beperkt .....	35
9.	Realisatie .....	36
9.1	Procesbeschrijving.....	36
9.1.1	Oriëntatie .....	36
9.1.2	Realisatie .....	37
9.1.3	Testen en evaluatie .....	40
10.	Implementatie .....	41
10.1	Codelgniter .....	41
11.	Conclusies en aanbevelingen .....	42
12.	Evaluatie .....	43
12.1	Procesevaluatie .....	43
12.2	Productevaluatie .....	44
12.3	Zelfreflectie.....	44
13.	Literatuurlijst .....	46
Bijlages.....		47
Bijlage 1: Publicatieproces.....		47
Bijlage 2: Usecases .....		48
Bijlage 3: User needs .....		51
Bijlage 4: Home wireframe.....		55
Bijlage 5: Education wireframe .....		56
Bijlage 6: University wireframe .....		57
Bijlage 7: Home prototype .....		58
Bijlage 8: Country wireframe.....		59
Bijlage 9: City wireframe .....		60
Bijlage 10: General information wireframe.....		61
Bijlage 11: Eye tracking results.....		62



## 1. Achtergrond

In 2009 werd het concept MITIP door Dhr. Gijsbert van Dalen gepubliceerd. Het concept beschrijft een informatieplatform waar studenten informatie kunnen vinden over de steden waar zij willen gaan studeren of stagelopen.

MITIP moet in eerste instantie benaderbaar zijn via het internet als webapplicatie waar informatie verkregen en gedeeld kan worden tussen studenten, onderwijsinstellingen en bedrijven.

Om de technische mogelijkheden van het concept te onderzoeken heb ik samen met medestudent Robin van der Markt onderzoek gedaan naar het platform.

Tijdens het onderzoek is het concept omgezet naar duidelijke functionaliteiten (usecases) die stuk voor stuk technisch onderzocht en gerealiseerd zijn in de vorm van een prototype.

In mei 2010 heeft Dhr. Gijsbert van Dalen een onderneming (Dalen Business) opgericht om het MITIP platform onder de naam iStip te kunnen realiseren. Dalen Business is gevestigd in Utrecht INC op de Uithof, waar studenten met startende ondernemingen een kans krijgen hun concepten te verwezenlijken.

Vanaf 20 januari 2010 ben ik begonnen om bij Dalen Business mijn afstudeerstage te gaan lopen.

## 2. Probleem analyse

### 2.1 Probleemstelling

Dalen Business wil een pilot lanceren waarmee alleen Utrecht in kaart wordt gebracht. Deze pilot wil Dalen Business in februari 2011 van start laten gaan.

Om iStip in februari 2011 te kunnen lanceren moeten alle betrokken partijen content kunnen toevoegen, bewerken en bekijken via een publicatieproces. Het gaat om medewerkers van iStip die hiervoor bevoegd zijn, bedrijven en onderwijsinstellingen. Er hoeft geen administratietool ontwikkeld te worden waarmee alle informatie beheerd kan worden.

De data die ingevoerd wordt door de partners hoeft nog niet van een uitgebreide editor te zijn voorzien, waarmee ze tekstverwerkingsfuncties kunnen uitvoeren.

Wel

- *Het publicatieproces<sup>1</sup> in kaart brengen*
- *Technieken en methodes analyseren en keuzes maken*
- *Ervoor zorgen dat er een pilot kan worden gestart in februari 2011 en informatie voor die tijd gepubliceerd kan worden*

Niet

- *Een administratietool*
- *Geen editor<sup>2</sup>*
- *Onderzoek naar het product (ten aanzien van kosten, doelgroep enz.)*

---

<sup>1</sup> Het toevoegen van informatie op het platform, waarna het door bezoekers van iStip toegankelijk is

<sup>2</sup> Een tool om tekstverwerking te automatiseren



## 2.2 Belanghebbenden

### *Dalen Business*

iStip is de core business waarmee Dalen Business in de toekomst inkomsten wil genereren. Elke dag dat het platform langer duurt om te ontwikkelen, kost Dalen Business geld.

Een ander belang van Dalen Business is dat zij partners als universiteiten laten betalen voor de service die ze hen bieden, waardoor Dalen Business zich moet bewijzen om deze service te kunnen leveren. Als Dalen Business het platform niet tijdig of compleet kan leveren, loopt Dalen Business het risico dat partners zich terugtrekken waardoor het voortbestaan van het platform in gevaar komt.

### *Partners*

De partners die zich profileren op het iStip platform willen zoveel mogelijk bekend worden bij internationale studenten omdat zij voor hen potentiële waarde vertegenwoordigen. Het platform biedt hen een nieuwe manier om naar internationale studenten te communiceren.

Zonder iStip kunnen de partners nieuwe studenten mislopen die mogelijk een relatie met het partnerbedrijf aan hadden kunnen gaan.

### *De internationale student*

Zonder iStip betekent het voor de internationale student dat zij meer moeite moeten doen om de informatie te vinden die zij nodig hebben om in het buitenland te kunnen studeren of werken. Ze zullen hiervoor meerdere instanties en organisaties moeten raadplegen om zich te laten informeren, omdat er nog geen centrale plek is waar alle informatie gevonden kan worden.

## 2.3 Diagnose

De oplossing die geboden kan worden binnen mijn stageperiode is een testomgeving waarmee de data van partners toegevoegd, bewerkt en bekeken kan worden. Op deze manier kan ervoor gezorgd worden dat de partners tijdig de service krijgen om hun informatie op het platform te publiceren. Het zorgt er ook voor dat de informatie al toegevoegd is voor de applicatie afgebouwd is, waardoor het platform compleet oogt voor de gebruiker (internationale student) op het moment van lancering.

Voor Dalen Business betekent het dat het platform sneller gelanceerd kan worden omdat de informatie van partners eerder beschikbaar is en dat de kernfunctionaliteiten al gebouwd en getest zijn.

## 3. Opdrachtomschrijving

Het realiseren van een testomgeving voor het iStip platform waarmee informatie toegevoegd, bewerkt en bekeken kan worden door medewerkers van iStip, onderwijsinstellingen en bedrijven. Hiermee kan informatie worden toegevoegd aan een 'webstad'<sup>3</sup>, land of bedrijven/onderwijsinstellingen pagina.

---

<sup>3</sup> Term voor een informatiepagina die alleen betrekking heeft op een stad



## 4. Aanpak

Binnen dit project wil ik een aantal methodes gaan hanteren waarmee het project uitgevoerd kan worden. Hierbij wil ik voornamelijk gebruik maken van methodes die ik in mijn opleiding heb meegekregen en nog niet eerder in een groot project heb kunnen toepassen.

Ik heb de opdracht geanalyseerd om vervolgens een keuze te kunnen maken in de methodes die ik wil gaan gebruiken. De opdracht is op te delen in die grote delen, het ontwikkelen, usability en projectmanagement. Voor elk zijn er methodes beschikbaar om invulling en sturing te geven aan deze onderdelen.

### 4.1 Software ontwikkeling

Voor softwareontwikkeling wil ik **Agile** software development methode gaan hanteren. Het is een methode die ik eerder toegepast heb in kleinschalige opdrachten. Met Agile kan ervoor gezorgd worden dat er tijdens de ontwikkeling van de software bijgestuurd kan worden.

Dit wordt gedaan door tijdens de ontwikkeling feedback van de opdrachtgever direct te verwerken waardoor er geen scheefgroei kan ontstaan tussen het ontwikkelteam en de opdrachtgever.

Met Agile kunnen er verschillende rollen binnen een developmentteam zijn die gezamenlijk aan de ontwikkeling werken. Er kan gewerkt worden met zogenaamde '**usecases**' waarin een functionaliteit of een stukje functionaliteit wordt beschreven.

Om een zo goed mogelijke voorstelling te maken van de functionaliteit binnen de applicatie maak ik gebruik van **wireframes**.

Vervolgens kunnen de functionaliteiten ontwikkeld worden binnen een **iteratief** proces, dat wil zeggen dat er een klein proces is (meestal één tot twee weken) waarmee een functionaliteit ontworpen, gerealiseerd en getest kan worden. Dit proces kan worden herhaald tot alle functionaliteiten gerealiseerd zijn.

De functionaliteit wordt verwerkt in een **prototype** waarmee de werking van de functionaliteit in een demo aan de opdrachtgever kan worden getoond.

### 4.2 Usability

Om usability te kunnen testen wil ik gebruik maken van een methode die **eye-tracking** heet. Tijdens het bekijken van de website worden de oogbewegingen gevolgd. De informatie wordt verwerkt tot '**heat maps**' waarmee te zien is waar de testpersoon naar heeft gekeken en welke onderdelen het meeste aandacht gevraagd hebben.

Een andere testmethode die ik wil gaan gebruiken is de '**trunk test**' waarmee getest kan worden hoe goed gebruikers weten waar ze zich bevinden. De testpersoon wordt een pagina getoond waarna hij/zij een vragenlijst invult om aan te geven waar de testpersoon zich bevond. Met deze informatie kan inzicht worden verkregen in de duidelijkheid van elke pagina om de gebruiker feedback te geven over de locatie.





### 4.3 Projectbeheersing

Om het project te kunnen beheersen wil ik gebruik maken van Scrum<sup>4</sup>, een projectbeheersingsmethode die toegespitst is op softwareontwikkeling. Met deze methode is het mogelijk om met weinig overhead<sup>5</sup> een project te kunnen beheersen.

### 4.4 Procesanalyse

Om processen om te kunnen zetten naar functionaliteiten die gebruikt kunnen worden in een applicatie moet het proces inzichtelijk gemaakt worden. Ik wil dit gaan doen processen te modelleren in **activiteiten diagrammen**. Met deze diagrammen is het mogelijk om de activiteiten tussen verschillende gebruikersgroepen en systemen weer te geven.

Met **user templates** wil ik in kaart brengen welke gebruikersgroepen er bestaan en welke eigenschappen ze hebben die van belang zijn voor het systeem.

Om het proces om te kunnen zetten naar concrete functionaliteiten wil ik **usecases** gebruiken. Usecases beschrijven een stukje functionaliteit door de interactie tussen de gebruiker en het systeem tekstueel uit te schrijven. Een usecase kan betrekking hebben op één of meerdere gebruikersgroepen.

### 4.5 Onderzoeksmethodes

#### Delphi

Met deze methode kunnen andere specialisten geraadpleegd worden voor advies. Op basis van deze adviezen kunnen keuzes worden gemaakt. Ik wil Delphi gebruiken om een keuze tussen technieken te maken binnen dit project. De adviezen van experts kunnen hierbij van grote waarde zijn.

#### Multi Criteria Analysis

Een Multi Criteria Analysis is een systematische methode voor het maken van keuzes. Door vooraf criteria op te stellen en deze vervolgens te toetsen en te wegen kan er een keuze worden gemaakt. Weging kan schematisch worden weergegeven in tabelvorm waarmee de gegevens gekwantificeerd kunnen worden (indien mogelijk).

#### Literatuuronderzoek

Door de vakliteratuur en documentatie van verschillende technieken te lezen kunnen de voor- en nadelen ten opzichte van het project verkregen worden. Ik wil hiermee mijn kennis vergroten om belangrijke afwegingen te kunnen maken.

---

<sup>4</sup> Keuze tussen Scrum en PRINCE2 staat beschreven in 7

<sup>5</sup> Tijd en energie die niet wordt besteed aan de realisatie van het product



## 5. Organisatie

Dalen Business is een bedrijf dat is opgericht door Dhr. Van Dalen met als doel om een internationaal studenten platform op te zetten genaamd iStip. Het bedrijf is opgericht in mei 2010 en wordt bijgestaan door externe coaching op bedrijfsmatig gebied verzorgd door Utrecht INC en Dhr. Van Loon die zijn technische expertise aan het project verleent.



Tot februari 2011 wil Dalen Business zich bezig houden met het opzetten van het platform in Utrecht om daar een pilot te gaan draaien met de partners die daar een rol in spelen.

De huidige organisatiestructuur is als volgt:

**Director / Product owner**

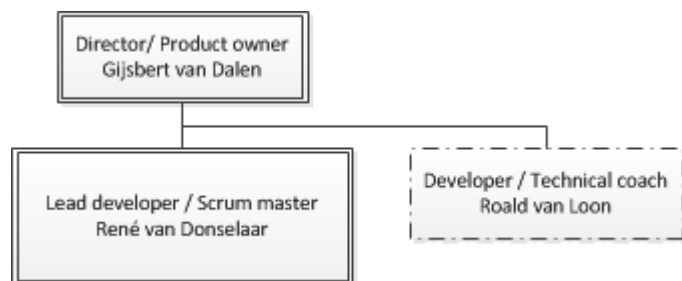
Gijsbert van Dalen

**Technical coach / Developer**

Roald van Loon

**Lead developer / Scrum master<sup>6</sup> (stagiair)**

René van Donselaar



Het bedrijf heeft de ambitie om na Januari 2011 een administrator tool te ontwikkelen waarmee het platform beheerd kan worden. Daarna wil Dalen Business het platform per stad uitbreiden door heel Nederland en op de lange termijn Europa, om vervolgens wereldwijd aanbod te leveren.

Dalen Business wil in de toekomst zoveel mogelijk met internationale studenten gaan werken omdat zij de doelgroep zijn en van waarde zijn als input binnen de ontwikkeling van het platform.

### 5.1 Uitgangssituatie

Voor het iStip platform is al een technisch onderzoek verricht om het concept (MITIP) uit te werken tot functionaliteiten en een technische invulling te geven.

Ook is er tijdens het onderzoek een prototype gemaakt met een aantal van de functionaliteiten die beschreven stonden in het onderzoek.

Het platform wordt van de grond af opgebouwd tijdens mijn stageperiode, dat betekent dat er volledige vrijheid is in het gebruiken van technieken en het vormgeven van functionaliteiten.

Er is één partner die zich al contractueel heeft verbonden met het platform, Hogeschool Utrecht. Dit wordt in de loop van het project uitgebreid met organisaties die betrokkenheid hebben met internationale studenten die naar Utrecht komen.

<sup>6</sup> Rol gedefinieerd binnen Scrum zie 7.1



## 6. Gebruikte technieken

Het platform is van de grond af opgebouwd in deze afstudeerperiode. Er was nog een vrije keus voor de technieken die gebruikt zouden worden om het platform te kunnen realiseren.

In dit hoofdstuk wil ik ingaan op alle belangrijke keuzes die er gemaakt zijn op het gebied van techniek. Ik beschrijf de randvoorwaarden waaraan elke techniek moet voldoen, en geef vervolgens aan wat de mogelijkheden zijn.

Vervolgens vergelijk ik de verschillende mogelijkheden en per techniek wordt aangegeven wat de voor- en nadelen zijn. Er volgt een weging en keuze met een conclusie bij elke techniek.

De aanpak kan per techniek verschillen en staat om die reden beschreven per techniek.

### 6.1 Randvoorwaarden

De randvoorwaarden zijn naar voren gekomen in de gesprekken met de opdrachtgever, met als belangrijkste vraag: Waar moet de applicatie aan voldoen? Om technische randvoorwaarden te kunnen formuleren heb ik mijn bedrijfsbegeleider betrokken bij deze gesprekken.

Bij het maken van de keuze speelde een aantal randvoorwaarden een rol:

- **De applicatie is in de toekomst goed schaalbaar**  
*Het iStip platform heeft de ambitie om in een later stadium verder in Europa uit te breiden. Het systeem zal mogelijk op meerdere servers moeten draaien, en meerdere databases aan moeten kunnen.*
- **De applicatie is in de toekomst goed uit te breiden**  
*Er zijn extra functionaliteiten zoals een stage zoekstelsel dat in de toekomst gerealiseerd gaat worden. iStip wil zoveel mogelijk registratiesystemen van bestaande organisaties waarmee de internationale student mee van doen heeft, in de toekomst aanbieden op het platform.*
- **Het MVC design pattern<sup>7</sup> is correct toegepast**  
*MVC is een algemeen geaccepteerd design pattern dat er voor zorgt dat de source code<sup>8</sup> goed beheersbaar blijft door middel van een duidelijke scheiding tussen presentatie, logica, en datamodellen.  
Deze voorwaarde is opgesteld om bij te dragen aan de beheersbaarheid van de code, maar ook om mijn kennis van het MVC design pattern te vergroten.*
- **De applicatie is geprogrammeerd in OOP<sup>9</sup>**  
*Door met objecten te werken kan veel redundantie worden voorkomen omdat functies en eigenschappen onderdeel zijn van objecten die maar éénmaal gedefinieerd moeten worden. OOP wordt veel gebruikt en om deze reden kunnen nieuwe developers snel verder bouwen op de bestaande applicatie.*
- **De applicatie is geschreven in PHP<sup>10</sup>**  
*Met PHP hebben zowel de bedrijfsbegeleider als ik het meeste ervaring. Het is ook een aanbeveling uit het eerdere onderzoek dat verricht is vanwege de grote community, dat het open source is en dat er veel goede ontwikkeltools kosteloos beschikbaar zijn.*

<sup>7</sup> Een patroon waarmee de code ingedeeld en gescheiden wordt

<sup>8</sup> De code waarmee het platform gebouwd is

<sup>9</sup> Object-oriented programming, gebruikt representaties van de werkelijkheid als objecten binnen de code

<sup>10</sup> Script/programmeertaal die veel gebruikt wordt voor webapplicaties



## 6.2 Framework of CMS

In dit hoofdstuk beschrijf ik mijn onderzoek naar de verschillen tussen framework en CMS systemen. Ik omschrijf wat een framework en CMS precies zijn en wat ze te bieden hebben.

Om duidelijk de verschillen te kunnen weergeven heb ik vooraf criteria opgesteld, die ik geformuleerd heb vanuit de doelstelling, probleemstelling, en randvoorwaarden van dit project.

De informatie die ik verkregen door de vakliteratuur te lezen heb ik verwerkt tot concrete voor- en nadelen die ik toets aan de vooraf gestelde criteria.

Na de toetsing volg ik met een aantal conclusies waarmee richting gegeven kan worden aan het project.

### 6.2.1 Criteria

Aan de volgende criteria heb ik de technieken getoetst:

- **Snelheid van de applicatie**  
*Hoe snel is de code die geproduceerd wordt door het framework of CMS.*
- **Snelheid van ontwikkeling**  
*Hoe snel kan de applicatie gebouwd worden.*
- **Flexibiliteit**  
*Hoe gemakkelijk kan het framework of CMS gewijzigd / aangepast worden.*
- **Toepassing van MVC**  
*Hoe goed worden de MVC regels nageleefd.*
- **Schaalbaarheid**  
*Hoe goed is de applicatie te verdelen over meerdere servers en databases.*
- **Uitbreidbaarheid**  
*Hoe gemakkelijk kan de applicatie uitgebreid worden met nieuwe functionaliteiten.*

### 6.2.2 Methodes

Ik heb de volgende methodes gehanteerd:

- **Literatuuronderzoek**  
*Vakliteratuur en documentatie is gebruikt om de technieken in kaart te brengen.*
- **Delphi**  
*Advies van experts is gebruikt om te leren hoe deze technieken worden toegepast in de praktijk.*
- **Multi criteria analyse (aangepast)**  
*MCA is gebruikt om de informatie te verwerken tot een keuze en visueel gekwantificeerd weer te geven. Dit is een versimpelede manier van MCA met een schaal van 1 t/m 5.*



## 6.2.3 Technieken

### 6.2.3.1 Framework

Een framework helpt bij de ontwikkeling door een set gestructureerde regels, patronen en methodes aan te bieden die de ontwikkelaar ondersteunen in het programmeren. Frameworks bieden over het algemeen 'best-practise' technieken aan, die in vrijwel elke applicatie een rol spelen. Om die reden is het verstandig om niet opnieuw het wiel uit te gaan vinden en voor een bestaand framework te kiezen.

#### Nadelen

- **Minder flexibel dan PHP zonder framework**  
*Een framework legt een manier van werken op aan de developers. Meestal zijn frameworks goed aan te passen, maar het is wel zo dat een framework een structuur heeft die al vast staat. Als de developers de bestaande structuur willen aanpassen, vergt dat veel tijd.*
- **Developers moeten om leren gaan met het framework**  
*Elk framework biedt zijn eigen unieke functies, naamgeving en structuur. De developers moeten om deze reden tijd investeren in het leren kennen van het framework ondanks dat ze al kennis hebben van de programmeertaal (PHP).*
- **Lagere snelheid**  
*De snelheid van een framework zal altijd lager liggen dan code die volledig geoptimaliseerd is zonder framework. Dat komt omdat het framework extra code toevoegt aan de applicatie om het framework te kunnen draaien.*
- **Veel programmeerkennis vereist**  
*De developer heeft kennis nodig van de programmeertaal (PHP), design patterns (MVC), OOP en het framework zelf.*

#### Voordelen

- **Snellere ontwikkeling van applicaties**  
*Sneller ontwikkelen dan bij sequentieel code schrijven. Dit geldt voor alle applicaties die een gemiddelde tot grote omvang hebben.*
- **Best practices**  
*Maakt gebruik van 'best practises', er hoeft niet opnieuw het wiel uitgevonden te worden. De development verloopt via bewezen methoden en structuren.*
- **Veelgebruikte functies zijn al aanwezig**  
*Veelgebruikte functies zijn vaak verwerkt in een framework, deze hoeven niet opnieuw geschreven te worden.*
- **Uitbreidbaarheid**  
*Een framework kan uitgebreid worden nieuwe modules (libraries, plugins). Hiermee kunnen extra functionaliteiten worden toegevoegd die al ontwikkeld zijn door andere developers.*
- **Schaalbaarheid**  
*Het framework legt geen beperkingen op ten aanzien van schaalbaarheid. Zo kunnen er engines voor views en models (ORM) geïntegreerd worden, die goed om kunnen gaan met verdeling over meerdere servers en databases.*
- **MVC**  
*In vrijwel alle frameworks wordt een nette MVC implementatie toegepast waar dat bij een CMS veel minder het geval is.*



### 6.2.3.2 CMS

Een CMS (content management system) helpt bij het ontwikkelen van een (web)applicatie. Het gaat verder dan een framework en zorgt ervoor dat de programmeerkennis minimaal kan zijn. Binnen het CMS wordt met modules gewerkt die toegevoegd en aangepast kunnen worden.

Met deze modules is het mogelijk om veelvoorkomende functionaliteiten, zoals een loginsysteem relatief snel te kunnen implementeren.

Wie een applicatie gaat bouwen waarbij de snelheid van de ontwikkeling voorop staat en het project bestaat uit veelvoorkomende functionaliteiten, doet er goed aan om een CMS te gebruiken.

#### Nadelen

- **Developers moeten leren werken met het CMS**  
*Het CMS heeft zijn eigen interface en beheertools, de developers moeten zich eigen maken met de functionaliteiten die het CMS hem biedt.*
- **Grotere hoeveelheid code dan bij een framework**  
*Omdat functionaliteiten al geschreven zijn binnen modules bieden ze vaak meer aan dan de developer nodig heeft, wat ervoor zorgt dat er meer overbodige code in de applicatie te vinden is.*
- **Zeer beperkt flexibel**  
*De keuzes die andere developers hebben gemaakt bij het schrijven van modules van het CMS maken het CMS minder flexibel.*
- **Beperkt schaalbaar**  
*Een CMS kan in een later stadium problemen opleveren bij de schaalbaarheid. Er zal enorm veel code omgevormd moeten worden om het systeem te configureren over meerdere servers en databases.*
- **Lagere snelheid dan geoptimaliseerde code**  
*Het CMS heeft een opbouw die niet nodig zou zijn voor de applicatie maar wel nodig is voor het CMS. Omdat het CMS altijd meer code produceert dan sequentieel geschreven code zal de snelheid lager liggen.*

#### Voordelen

- **Snelheid van development**  
*De ontwikkeling van functionaliteiten kan versneld worden door gebruik te maken van voorgeprogrammeerde modules die alleen geconfigureerd dienen te worden.*
- **Admin omgeving**  
*Bij een CMS hoort een admin omgeving waar de developer grafisch feedback krijgt van zijn acties. Ook is er geen programmeerkennis vereist om met deze admin omgeving te kunnen werken.*
- **Uitbreidbaarheid**  
*Nieuwe modules kunnen eenvoudig worden toegevoegd aan het CMS.*
- **Weinig/geen programmeerkennis vereist**  
*Het CMS werkt vanuit een interface die begrijpelijk is voor developers zonder al te veel ervaring met programmeren.*



### 6.2.4 Weging en keuze

Voor de keuze tussen het gebruik van een CMS of framework hebben een aantal factoren een belangrijke rol gespeeld: de snelheid waarmee de applicatie ontwikkeld kan worden, de uitbreidbaarheid en schaalbaarheid, maar ook het persoonlijk leerdoel om beter MVC en OOP toe te kunnen passen.

Omdat het project unieke functionaliteiten bevat en in de toekomst uitgebreid gaat worden met verdere functionaliteiten is hiervoor een framework het meest geschikt. De functionaliteiten kunnen hier zonder restricties van de grond af ontworpen en geprogrammeerd worden.

Om met een CMS hetzelfde te kunnen doen zullen modules dusdanig aangepast moeten worden dat de snelheidswinst van het CMS verloren gaat.

Op het gebied van schaalbaarheid is een framework het meest flexibel en kan in de toekomst problemen voorkomen. Gezien de doelstellingen van het iStip platform, om een groot internationaal platform te worden voor de internationale student, is dit een doorslaggevende factor om voor een framework te kiezen.

Uiteindelijk draagt het framework ook in grote mate bij aan mijn persoonlijke leerdoelen voor deze afstudeerstage, om beter te leren programmeren in MVC en OOP.

### 6.2.5 Conclusie

Ik heb de gegevens uit de analyse verwerkt in een tabel met 1 t/m 5 waardering om een weergave te geven van de bevindingen, zie Tabel 1 - Framework en CMS vergeleken.

Tabel 1 - Framework en CMS vergeleken

Schaal van 1 t/m 5

Criteria	Framework	CMS
Snelheid van code	++++	++
Snelheid van ontwikkeling	++++	+++++
Flexibiliteit	++++	++
Toepassing van MVC	+++++	+++
Schaalbaarheid	++++	+
Uitbreidbaarheid	++++	+++++

Op basis van deze bevindingen zijn er een aantal conclusies te trekken:

- Een framework biedt doorgaans de snelste applicatie
- Een CMS kan de snelste ontwikkeling bieden
- Een framework is het meest flexibel
- Een framework biedt de beste toepassing van MVC
- Met een framework kan de applicatie het meest schaalbaar gemaakt worden
- Een CMS kan het makkelijkste uitgebreid worden

### Hoofdconclusie

Een framework sluit het beste aan bij de behoeftes van Dalen Business om snel doch schaalbaar te ontwikkelen.



### 6.3 Keuze voor php-framework

In dit hoofdstuk beschrijf ik mijn onderzoek naar verschillende PHP frameworks. Ik heb aantal frameworks geselecteerd op basis van mijn eerdere ervaringen, ervaringen van andere experts en de aanwezigheid van een stabiele (non-beta) release.

De verschillen tussen de frameworks heb ik bekeken aan de hand van vooraf gestelde criteria, die ik heb ontleend aan de randvoorwaarden, inhoudelijke gesprekken met experts en mijn eerdere ervaring met frameworks.

#### 6.3.1 Criteria

Aan de volgende criteria heb ik de technieken getoetst:

- **Integratie ORM<sup>11</sup>**  
*In hoeverre is er een ORM geïntegreerd in het framework, en hoeveel mogelijkheden biedt deze.*
- **Integratie template engine<sup>12</sup>**  
*In hoeverre is er een template engine geïntegreerd in het framework, en hoeveel mogelijkheden biedt deze.*
- **Community<sup>13</sup>**  
*Hoe groot is de community, hoeveel blogs, tutorials en forumposts worden er gepubliceerd.*
- **Documentatie**  
*Hoeveel officiële documentatie is er beschikbaar en hoe toegankelijk is deze.*
- **Snelheid**  
*Hoe snel is een applicatie die gemaakt wordt binnen de verschillende frameworks.*
- **Uitbreidbaarheid**  
*Hoe gemakkelijk kan het framework uitgebreid worden met nieuwe functionaliteiten.*
- **Kleine footprint<sup>14</sup>**  
*Hoeveel code het framework zelf in beslag neemt.*

#### 6.3.2 Methodes

Ik heb de volgende methodes gehanteerd:

- **Literatuuronderzoek**  
*Vakliteratuur en documentatie is gebruikt om de technieken in kaart te brengen.*
- **Delphi**  
*Advies van experts is gebruikt om te leren hoe deze technieken worden toegepast in de praktijk.*
- **Multi criteria analyse (aangepast)**  
*MCA is gebruikt om de informatie te verwerken tot een keuze en visueel gekwantificeerd weer te geven. Dit is een versimplede manier van MCA met een schaal van 1 t/m 5.*

---

<sup>11</sup> Object Relational Mapper, zie 21

<sup>12</sup> Wordt gebruikt om views mee op te bouwen zie 25

<sup>13</sup> Developers die actieve bijdrage leveren aan het informeren van andere developers

<sup>14</sup> Hoe groot het framework is zonder dat de applicatie wordt meegerekend





### 6.3.3 Techieken

#### 6.3.3.1 CodeIgniter



CodeIgniter is een lichtgewicht framework dat een duidelijk MVC patroon aan de developer biedt. De regels die CodeIgniter biedt kunnen zo strikt aangehouden worden als de developer wil.

#### Nadelen

- **Mist authenticatie module**  
*Er zijn developer libraries die dit probleem oplossen, maar standaard is het niet aanwezig.*
- **Geen krachtig ORM geïntegreerd**  
*Maakt gebruik van 'active records', die redelijk beperkt zijn. Het is geen ORM maar een manier om queries verkort te schrijven.*
- **Geen krachtige template engine geïntegreerd**  
*Er is een template language geïntegreerd in CodeIgniter, maar deze is zeer beperkt en kan zich niet meten met een template engine als Smarty.*

#### Voordelen

- **Grote community (groeierende)**  
*Zeer populair onder PHP developers van midden en klein bedrijf.*
- **Veel documentatie**  
*De documentatie is uitgebreid en per functie is er een diepgaande beschrijving. De documentatie is compleet en duidelijk gestructureerd.*
- **Minimalistische opzet**  
*Het aantal functies dat CodeIgniter biedt is minimaal gehouden. Nieuwe functies kunnen worden toegevoegd via libraries en plugins. Om deze reden gebruikt CodeIgniter standaard weinig code.*
- **Gemakkelijk uitbreidbaar (libraries, plugins)**  
*Het framework is gemakkelijk uitbreidbaar met libraries en plugins. Ook de basis van CodeIgniter is gemakkelijk uitbreidbaar (hoe controllers werken enz.).*
- **Weinig configuratie nodig**  
*CodeIgniter hoeft amper geconfigureerd te worden. In één configuratiebestand dienen een aantal gegevens geconfigureerd te worden.*
- **Snelheid**  
*Omdat CodeIgniter standaard weinig code gebruikt is het een relatief snel framework.*



### 6.3.3.2 Zend Framework



Zend Framework onderscheidt zich van andere frameworks door de enorme featureset. Het framework is uitgebreid en biedt een duidelijk MVC patroon aan de developer.

#### Nadelen

- **Groot**  
*Het framework zelf bestaat uit veel code, het is een zwaargewicht onder de php frameworks.*
- **Snelheid**  
*Mede door de grote featureset die het framework biedt is het een van de minst snelle frameworks.*
- **Premium tools en libraries tegen betaling**  
*Zend Framework is een open source framework waarvoor niet betaald hoeft te worden, maar wordt ontwikkeld door het bedrijf Zend, dat commerciële tools en plugins ter beschikking stelt waarvoor betaald moet worden.*
- **Configuratie**  
*De configuratie van het framework is uitgebreid en bevat ook configuratie voor veel features die niet altijd gebruikt gaan worden.*
- **Geen krachtig ORM geïntegreerd**  
*Ondanks dat Zend Framework standaard zo uitgebreid is, is er geen krachtig ORM aanwezig.*
- **Geen krachtige template engine geïntegreerd**  
*Er is een simpele template engine geïntegreerd.*

#### Voordelen

- **Uitgebreid**  
*Standaard zijn er al veel features aanwezig die andere frameworks via extra libraries verkrijgen*
- **Redelijk grote community (dalende)**  
*Voorals populair onder grote bedrijven. Community wordt kleiner, en veel informatie is intussen outdated.*
- **Veel documentatie**  
*Op forums en blogs is veel informatie beschikbaar over Zend Framework.*
- **Veel plugins en libraries beschikbaar**  
*Er wordt actief gewerkt aan nieuwe plugins en libraries.*



### 6.3.3.3 CakePHP



CakePHP is een PHP framework dat zich baseert op de werking van Ruby On Rails. CakePHP gebruikt een commandline interface voor het genereren van scripts.

#### Nadelen

- **Snelheid**  
*De grote featureset zorgt ervoor dat de snelheid relatief laag ligt.*
- **Configuratie (commandline)**  
*De configuratie van CakePHP gaat deels via een commandline interface en vereist extra bestanden zoals php.exe. Dit maakt de configuratie van CakePHP wat meer tijdrovend dan bij sommige andere frameworks.*
- **Veel plugins en libraries zijn gedateerd**  
*De meeste development voor plugins en libraries was in 2006/7. Voor nieuwe plugins is het aanbod beperkt.*
- **Geen krachtig ORM geïntegreerd**  
*CakePHP heeft een ORM, maar kan zich niet meten in functionaliteit met een ORM als Propel of Doctrine.*
- **Geen krachtige template engine geïntegreerd**  
*Er is alleen een redelijk simpele template engine geïntegreerd binnen CakePHP.*

#### Voordelen

- **Redelijk grote community (dalende)**  
*De community van CakePHP is de afgelopen twee jaar niet zo actief meer. Het is op dit punt niet meer vergelijkbaar met bijv. CodeIgniter, dat een veel grotere en actievare community heeft.*
- **Veel documentatie**  
*De documentatie van CakePHP is uitgebreid. Via de 'bakery' kan veel informatie verkregen worden over functionaliteiten die het framework biedt.*





Yii is een nieuw php framework dat zich kenmerkt door snelheid en kleine footprint (omvang).

### Nadelen

- **Kleine community**  
*Yii is nieuw en moet zich nog bewijzen, er is nog onvoldoende tijd geweest om een grote community op te bouwen. De community groeit snel en produceert allerlei blogs, libraries en plugins.*
- **Matige hoeveelheid documentatie**  
*De documentatie is niet compleet en dat is te merken als je de diepte in gaat met het framework. De 'beginners'-guides zijn wel in orde.*
- **Weinig plugins en libraries**  
*Vanwege de kleine community zijn er ook nog weinig plugins en libraries.*
- **Geen krachtig ORM geïntegreerd**  
*Yii gebruikt 'active records' om queries mee te schrijven, maar is niet vergelijkbaar met een ORM als Propel of Doctrine.*
- **Geen krachtige template engine geïntegreerd**  
*Er is geen krachtige template engine aanwezig zoals bijv. Smarty.*

### Voordelen

- **Minimalistische opzet**  
*Yii is een klein framework met weinig overhead. Standaard biedt Yii een beperkt aantal functionaliteiten.*
- **Snelheid**  
*Het framework is één van de snelste van dit moment.*
- **Uitbreidbaarheid**  
*Het framework is gemakkelijk uit te breiden omdat Yii bestaat uit losse componenten die gemakkelijk aangepast en uitgebreid kunnen worden.*



### 6.3.4 Weging en keuze

Bij de keuze voor een framework was het van belang dat er voldoende documentatie te vinden was over het framework. Documentatie is cruciaal voor het gebruiken maar ook toekomstig ombouwen van het framework.

De community die het framework ondersteunt is van belang voor de actieve feedback van andere developers, de blogs en libraries. Een grote community kan tijdsbesparend werken op de development. Gezien de beperkte ontwikkeltijd is dit een punt dat zwaar weegt in de keuze voor een framework.

#### Interesse in Google Zoeken: codeigniter, zend framework, cakephp, yii

Wereldwijd, 2004 - heden

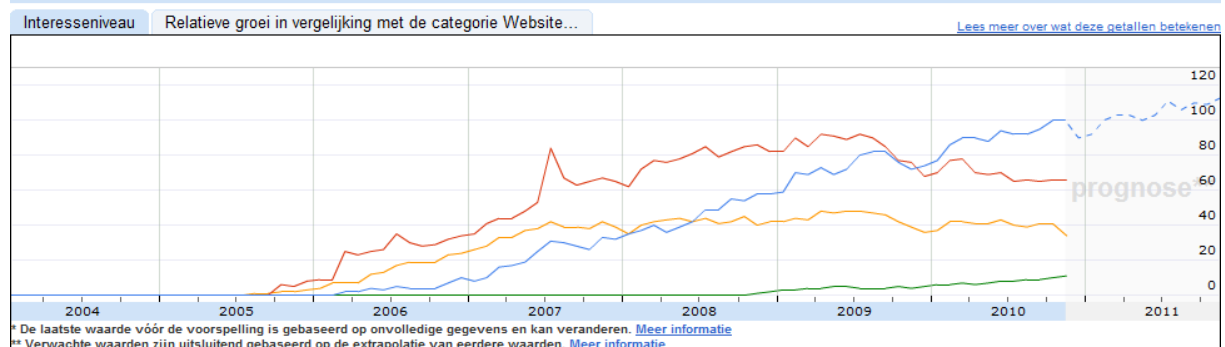
[Alle categorieën](#) > [Internet](#) > Websiteontwerp en -ontwikkeling

Minder dan 25% van de zoekopdrachten die uw zoektermen bevatten, behoort tot de categorie Websiteontwerp en -ontwikkeling. [Meer informatie](#)

Totalen	
codeigniter	42
zend framework	45
cakephp	25
yii	2

#### Interesse in de loop der tijd

☒ Prognose ☐ Nieuwskoppen



Figuur 2 - Interesse in frameworks binnen Google zoeken

Een minder grote rol speelt het ingebouwde ORM of de template engine, omdat er al grote libraries beschikbaar zijn. Het is van groter belang dat deze libraries gemakkelijk geïntegreerd kunnen worden binnen het framework.

Doorslaggevend is de ervaring met een framework, omdat dit de ontwikkeltijd enorm beïnvloedt en elk framework technisch gezien geschikt is om de applicatie mee te bouwen.

Met alle bovenstaande frameworks heb ik gewerkt, maar verreweg de meeste ervaring heb ik opgedaan met CodeIgniter.

De grote hoeveelheid ervaring, community en documentatie hebben er uiteindelijk voor gezorgd dat ik voor CodeIgniter heb gekozen als framework.



### 6.3.5 Conclusie

Ik heb de gegevens uit de analyse verwerkt in een tabel met 1 t/m 5 waardering om een weergave te geven van de bevindingen, zie Tabel 2 - Vergelijking tussen PHP frameworks.

Tabel 2 - Vergelijking tussen PHP frameworks

#### Schaal 1 t/m 5

Criteria	CodeIgniter	Zend Framework	CakePHP	Yii
Integratie ORM	++	+	++	++
Integratie template engine	++	++	++	++
Community	++++	++++	+++	++
Documentatie	++++	++++	++++	++
Snelheid	++++	++	+++	+++++
Uitbreidbaarheid	++++	+++++	+++	+++++
Configuratie	+++++	++	++	++++
Kleine footprint	+++++	+	+++	+++++

Op basis van alle bevindingen zijn er een aantal conclusies te trekken:

- *Krachtige ORMs zijn niet geïntegreerd binnen frameworks*
- *Krachtige template engines zijn niet geïntegreerd binnen frameworks*
- *CodeIgniter en Zend Framework hebben de grootste community*
- *Yii is het snelste framework*
- *Zend Framework is het best uitbreidbaar*
- *CodeIgniter is het gemakkelijkst te configureren*
- *CodeIgniter en Yii hebben de kleinste footprint*

#### Hoofdconclusie

CodeIgniter biedt de meeste voordelen voor de realisatie van het iStip platform van alle onderzochte frameworks.



## 6.4 Keuzes binnen de afhandeling van MVC

In dit hoofdstuk wordt beschreven hoe MVC afgehandeld wordt binnen CodeIgniter en wat de voor- en nadelen zijn van deze implementatie.

Verder beschrijf ik hoe CodeIgniter kan worden uitgebreid om nog beter met MVC om te gaan.

Uiteindelijk trek ik conclusies over hoe MVC het best afgehandeld kan worden binnen CodeIgniter.

### 6.4.1 CodeIgniter alleen is niet goed genoeg

Binnen CodeIgniter wordt MVC al goed afgehandeld, maar het wordt niet geforceerd. Er kan dus prima business logic in de views komen.

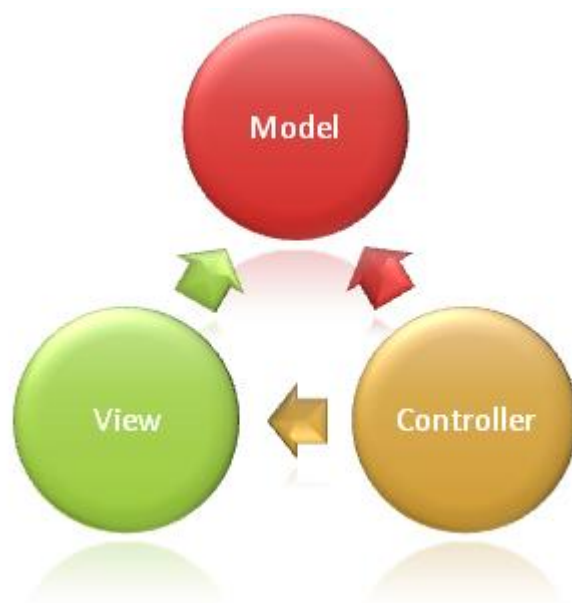
Standaard is CodeIgniter een geïntegreerd geheel dat beperkte mogelijkheden biedt voor grotere systemen ten aanzien van MVC.

Zo is de afhandeling van de database en het ontbreken van een krachtig ORM een probleem voor de toekomstige uitbreiding van het platform, dat zoveel mogelijk gescheiden dient te worden om flexibel om te kunnen gaan met schaling.

De toekomst kan ook uitwijzen dat een bepaald deel van een view enorm veel performance vraagt waardoor het beter kan draaien op een dedicated server.

Het opknippen van taken en verdelen over verschillende servers wordt standaard niet ondersteund door CodeIgniter.

Om deze redenen is ervoor gekozen om CodeIgniter uit te breiden met een aparte template engine voor de views en een ORM voor de models. CodeIgniter zal voornamelijk gebruikt worden als controller en basisstructuur van de applicatie.



Figuur 3 - MVC pattern met drie lagen



### 6.4.2 Keuze voor ORM

Een Object Relational Mapper zorgt ervoor dat objecten kunnen worden vertaald naar een relationele database. Dit zorgt ervoor dat de developer volledig OO kan programmeren ondanks dat er gebruik wordt gemaakt van een relationele database.

Een ORM kan veel logica bevatten waardoor er ook slimmer met de database kan worden omgegaan. Tijdens deze stageperiode is het voor mij belangrijk om goed OO te kunnen programmeren en het ORM draagt hier aan bij.

Ik heb twee ORMs geselecteerd om te gaan onderzoeken, deze zijn geselecteerd op basis van grootte. Het zijn de twee grootste ORMs met de meeste mogelijkheden van dit moment. Om de ORMs te kunnen vergelijken heb ik criteria opgesteld waaraan ik de ORMs heb getoetst door de voor- en nadelen in kaart te brengen.

#### 6.4.2.1 Criteria

Aan de volgende criteria heb ik de technieken getoetst:

- **Community**  
*Hoe groot is de community, hoeveel blogs, tutorials en forumposts worden er gepubliceerd.*
- **Documentatie**  
*Hoeveel officiële documentatie is er beschikbaar en hoe toegankelijk is deze.*
- **Configuratie**  
*Hoe gemakkelijk is het ORM te configureren.*
- **Integratie met CodeIgniter**  
*Hoe gemakkelijk kan het ORM geïntegreerd worden binnen CodeIgniter.*
- **Query's schrijven**  
*Hoe gemakkelijk is het om query's te schrijven en hoe leesbaar zijn ze.*

#### 6.4.2.2 Methodes

Ik heb de volgende methodes gehanteerd:

- **Literatuuronderzoek**  
*Vakliteratuur en documentatie is gebruikt om de technieken in kaart te brengen.*
- **Delphi**  
*Advies van experts is gebruikt om te leren hoe deze technieken worden toegepast in de praktijk.*
- **Multi criteria analyse (aangepast)**  
*MCA is gebruikt om de informatie te verwerken tot een keuze en visueel gekwantificeerd weer te geven. Dit is een versimplede manier van MCA met een schaal van 1 t/m 5.*





### 6.4.2.3 Propel



Propel is een van de grootste ORM's voor PHP en wordt veel gebruikt bij grote applicaties.

#### Nadelen

- **Development heeft vertraging gehad**  
*De ontwikkeling van Propel is langzamer gegaan sinds de komst van Doctrine, waar uiteindelijk Doctrine van geprofiteerd heeft. De ontwikkeling van Propel is nu weer op het oude niveau.*
- **Laatste versie (1.5) niet zo stabiel als voorgaande**  
*Heel veel nieuwe functionaliteiten die ervoor hebben gezorgd dat versie 1.5 niet zo stabiel is als versie 1.4.*
- **Weinig documentatie over integratie met CodeIgniter**  
*Er zijn minder blogs en tutorials te vinden over Propel door een wat kleinere community.*
- **Configuratie**  
*De configuratie van propel is redelijk uitgebreid en gebruikt 'phing' dat apart geïnstalleerd moet worden.*
- **Query's zijn moeilijker te lezen dan DQL**

#### Voordelen

- **Redelijk veel documentatie beschikbaar**  
*Er is documentatie te vinden over de huidige en alle voorgaande versies. De documentatie is redelijk uitgebreid en gaat in op alle functies van Propel.*
- **Query's zijn kort**  
*Door het gebruik van criteria's kunnen queries in Propel met weinig code worden geschreven.*



#### 6.4.2.4 Doctrine



Doctrine is een groot ORM dat de laatste jaren een groeispurt heeft meegemaakt.

#### Nadelen

- **Query's zijn langer**  
*De code die geschreven moet worden met DQL is doorgaans langer dan die van Criteria.*

#### Voordelen

- **Veel documentatie beschikbaar**  
*Doctrine heeft een uitgebreide documentatie die groter is dan die van Propel.*
- **Actieve community**  
*Er is een grote actieve community die veel blogs, tutorials en forumposts produceert.*
- **Veel documentatie te vinden over de integratie met codeigniter**  
*Er zijn meerdere tutorials geschreven die beschrijven hoe Doctrine geïntegreerd kan worden binnen CodeIgniter.*
- **Query's zijn gemakkelijk te lezen**  
*Door DQL is de code makkelijk te lezen en lijkt meer op SQL dan dat bij criteria het geval is.*
- **Configuratie**  
*Doctrine heeft weinig configuratie nodig en vereist geen externe toepassingen.*

#### 6.4.2.5 Weging en keuze

Beide ORM's hebben ongeveer een gelijke featureset en zijn technisch geschikt om gebruikt te kunnen worden binnen de iStip applicatie.

Op dit moment is Doctrine het voordeel door de grotere community en hoeveelheid documentatie.

#### Interesse in Google Zoeken: doctrine, propel

Wereldwijd, 2004 - heden

[Alle categorieën](#) > [Internet](#) > Websiteontwerp en -ontwikkeling

Minder dan 10% van de zoekopdrachten die uw zoektermen bevatten, behoort tot de categorie Websiteontwerp en -ontwikkeling.  
[Meer informatie](#)

**Totalen**

doctrine

propel

#### Interesse in de loop der tijd

☒ Prognose ☐ Nieuwskoppen

Interesseniveau Relatieve groei in vergelijking met de categorie Website...

[Hoe kan ik getallen bekijken?](#)



Figuur 4 - Interesse in ORM's binnen Google zoeken

Doorslaggevend is het feit dat mijn bedrijfsbegeleider veel ervaring heeft met Propel, waardoor de community, documentatie en configuratie van minder belang zijn.

Omdat er beperkte tijd is voor de development is het belangrijk dat er snel met het ORM gewerkt kan worden. De ervaring van mijn bedrijfsbegeleider op dit gebied zal de development doen versnellen.

Om deze reden heb ik uiteindelijk besloten om Propel te gaan gebruiken als ORM.

#### 6.4.2.6 Conclusie

Ik heb de gegevens uit de analyse verwerkt in een tabel met 1 t/m 5 waardering om een weergave te geven van de bevindingen, zie Tabel 3 - ORMs vergeleken.

Tabel 3 - ORMs vergeleken

##### Schaal 1 t/m 5

Criteria	Propel	Doctrine
Community	++++	+++++
Documentatie	++++	+++++
Configuratie	+++	++++
Integratie met CodeIgniter	++	++++
Query's schrijven	+++++	+++++

Op basis van alle bevindingen zijn er een aantal conclusies te trekken:

- **Doctrine heeft de grootste community**
- **Doctrine heeft de meeste documentatie**
- **Doctrine vereist de minste configuratie**
- **De integratie met CodeIgniter is gemakkelijker met Doctrine**
- **Beide frameworks hebben hun voordelen ten opzichte van query's schrijven**

#### Hoofdconclusie

Doctrine biedt op dit moment de meeste voordelen voor de realisatie van het iStip platform, maar Propel wordt gebruikt vanwege de betere interne begeleiding die daarmee mogelijk is.



### 6.4.3 Keuze voor template engine

Standaard is er in CodeIgniter een beperkte template engine te vinden. Deze engine is niet geschikt voor complexe systemen waarbij delen van de view opgesplitst moeten worden en op verschillende servers draaien.

Ook forceert CodeIgniter niet dat er geen php code in een view mag staan, wat kan leiden tot business logic binnen views waar alleen view logic zou mogen staan.

Om deze reden is ervoor gekozen om een template engine toe te voegen aan CodeIgniter.

In dit hoofdstuk vergelijk ik twee template engines. Omdat de meeste engines redelijk op elkaar lijken heb ik ze geselecteerd op diversiteit en populariteit. Smarty is op dit moment de grootste template engine en Savant3 gebruikt PHP i.p.v. een eigen template language.

Ik heb de twee template engines vergeleken op basis van een aantal criteria die ik vooraf heb opgesteld vanuit de ervaring met template engines en de randvoorwaarden.

#### 6.4.3.1 Criteria

Aan de volgende criteria heb ik de technieken getoetst:

- **Community**  
*Hoe groot is de community, hoeveel blogs, tutorials en forumposts worden er gepubliceerd.*
- **Documentatie**  
*Hoeveel officiële documentatie is er beschikbaar en hoe toegankelijk is deze.*
- **Snelheid**  
*Hoe snel kan de template engine de views produceren.*
- **Integratie met CodeIgniter**  
*Hoe gemakkelijk kan de template engine geïntegreerd worden binnen CodeIgniter.*
- **Naleving MVC**  
*In hoeverre zorgt de template engine ervoor dat MVC wordt nageleefd.*
- **Toegankelijkheid**  
*Hoeveel moet een onwikkelaar leren voor hij/zij aan de slag kan met de template engine.*

#### 6.4.3.2 Methodes

Ik heb de volgende methodes gehanteerd:

- **Literatuuronderzoek**  
*Vakliteratuur en documentatie is gebruikt om de technieken in kaart te brengen.*
- **Multi criteria analyse (aangepast)**  
*MCA is gebruikt om de informatie te verwerken tot een keuze en visueel gekwantificeerd weer te geven. Dit is een versimplede manier van MCA met een schaal van 1 t/m 5.*



### 6.4.3.3 Smarty



Smarty is de grootste en meest bekende template engine voor PHP.

#### Nadelen

- **Compiler**  
*Smarty templates moeten gecompileerd worden naar PHP, dit levert een prestatieverlies op.*
- **Template language**  
*Developers moeten een nieuwe template language leren om met Smarty te kunnen werken.*

#### Voordelen

- **Voorkomt business logic in view**  
*De smarty parser laat niet toe dat er php code in een template staat.*
- **Grote community**  
*Smarty heeft de grootste community van alle template engines.*
- **Veel documentatie**  
*Er is uitgebreide documentatie beschikbaar*
- **Caching**  
*Smarty heeft een complex cachingsysteem.*
- **Integratie met CodeIgniter**  
*De integratie met CodeIgniter staat goed beschreven in blogs van de community.*

### 6.4.3.4 Savant 3

# SAVANT3

## THE SIMPLE, ELEGANT TEMPLATE SYSTEM FOR PHP

Savant 3 is een kleine, op PHP gebaseerde template engine.

#### Nadelen

- **Kleine community**  
*Het aantal developers dat werkt met Savant 3 is beperkt waardoor de community klein is. Er zijn weinig blogs, libraries en forumtopics te vinden.*
- **Business logic in views is mogelijk**  
*Bij Savant 3 gaat wordt er vanuit gegaan dat er vertrouwen is tussen de programmeurs en de webdesigners om te voorkomen dat business logic in views terecht komt.*
- **Integratie met CodeIgniter**  
*Er is documentatie of tutorial te vinden waarin uitgelegd wordt hoe Savant 3 geïntegreerd kan worden binnen CodeIgniter.*



### Voordelen

- **Geen compiler nodig**  
*Savant 3 gebruikt PHP als template language, waardoor er geen compiler nodig is. De code van Savant 3 kan direct uitgevoerd worden.*
- **Snel te leren**  
*Omdat Savant 3 PHP gebruikt als template language is het voor developers snel te leren.*

#### 6.4.3.5 Weging en keuze

Beide template engines hebben een redelijk gelijke featureset en kunnen beide gebruikt worden binnen de iStip applicatie.

Voor het gebruiken van MVC is de scheiding tussen de drie lagen belangrijk en daar voorziet Smarty beter in.

Ook is de grootte van de community een belangrijk punt omdat ik nog beperkte ervaring heb met een template engine. Forumtopics, blogs en tutorials kunnen helpen bij het leren werken met de template engine.

De snelheid van en het compileren van de code is van minder belang omdat er bij Smarty goede caching aanwezig is.

Vanwege het goed (geforceerd) kunnen toepassen van MVC en de grote community heb ik gekozen om Smarty te gaan gebruiken als template engine.

#### 6.4.3.6 Conclusie

Ik heb de gegevens uit de analyse verwerkt in een tabel met 1 t/m 5 waardering om een weergave te geven van de bevindingen, zie Tabel 4 - Template engines vergeleken.

Tabel 4 - Template engines vergeleken

#### Schaal 1 t/m 5

Criteria	Smarty	Savant3
Community	+++++	++
Documentatie	+++++	+++
Snelheid	++++	+++++
Integratie met CodeIgniter	+++++	+++
Naleving MVC	+++++	+
Toegankelijkheid	+++	+++++

Op basis van alle bevindingen zijn er een aantal conclusies te trekken

- **Smarty heeft de grootste community**
- **Smarty heeft de meeste documentatie**
- **Smarty is het gemakkelijkst te integreren binnen CodeIgniter**
- **Smarty zorgt voor de beste naleving van MVC**
- **Savant3 is het meest toegankelijk**



## Hoofdconclusie

Smarty is het meest geschikt als template engine voor de realisatie van het iStip platform.

## 7. Projectbeheersing

Projectbeheersing is van beperkt belang in dit project omdat het een klein team betreft met maar één ontwikkelaar.

Het is voornamelijk belangrijk dat het producten, taken en tijdsverloop in kaart wordt gebracht tijdens de development.

Ik heb hiervoor twee projectontwikkelmethodes bekeken, Scrum: aangedragen door mijn bedrijfsbegeleider, en Prince2: een methode die ik al vaker heb gebruikt en waarvoor ik gecertificeerd ben.

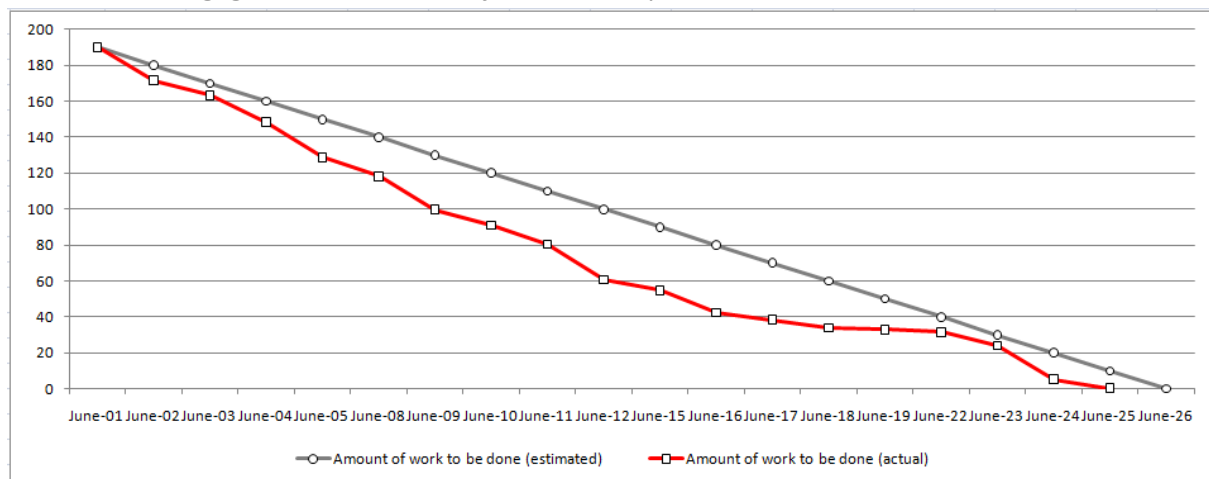
Met literatuuronderzoek heb ik bekeken hoe het karakter van beide projectmanagementmethodes is, en of deze past bij het karakter van het project.

### 7.1 Scrum



Een projectbeheersmethode die vooral gericht is op software development. Wordt in veel grote bedrijven gebruikt zoals Microsoft, Oracle, Sun en IBM. Een van de belangrijkste onderdelen is de burndown chart, waarmee inzichtelijk wordt gemaakt hoe de development verloopt.

In de burndown chart is te zien hoeveel tijd developers kwijt dachten te zijn aan een bepaalde taak en er wordt weergegeven hoe de werkelijkheid is verlopen.



Figuur 5 - Voorbeeld van een Scrum Burndown Chart



## Sprints

Scrum werkt met een iteratief proces die 'sprints' worden genoemd. Een sprint duurt doorgaans twee werkweken waarin er gewerkt wordt aan een functionaliteit uit het 'product backlog'. Het product backlog is een prioriteitenlijst van functionaliteit die wordt gevuld door de product owner (de opdrachtgever).

De functionaliteiten worden opgedeeld in taken die aan één of meerdere developers worden gekoppeld in een 'sprint backlog'. De developer schat vervolgens in hoeveel tijd hij/zij kwijt is met de taak. Vervolgens werkt de developer elke dag de uren die hij/zij aan de taak heeft besteed bij.



Figuur 6 - Scrum proces

Per dag wordt er een 'sprint meeting' (vergadering) gehouden waarin de developer aangeeft waar hij/zij mee bezig is geweest, de mogelijke problemen die ontstaan zijn en wat de developer die dag wil gaan doen. De vergadering wordt geleid door de 'scrum master' die ervoor zorgt dat het ontwikkelteam gefocust blijft op de taken die in de sprint zijn opgenomen. Aan het einde van elke sprint wordt er een presentatie gegeven van de ontwikkelde functionaliteit en er wordt een review gehouden waarin teruggekeken wordt op het verloop van de sprint, deze worden opgenomen in de 'sprint retrospectives'.

## Rollen

### Product owner

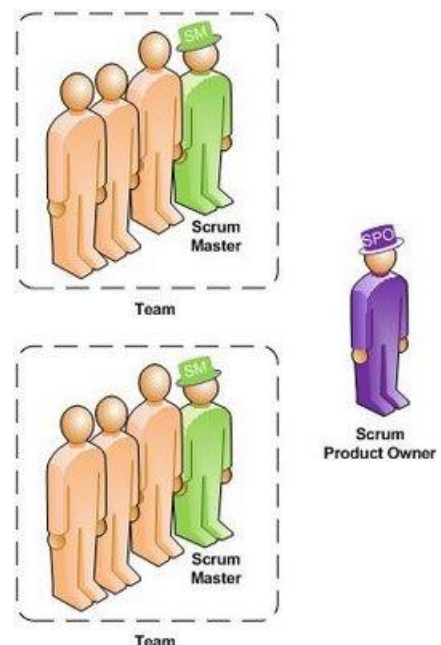
De product owner kan de opdrachtgever zijn en is verantwoordelijk voor de waarde van het product. De voornaamste taak van de product owner is om aan te geven welke producten hij/zij gerealiseerd wil hebben en er prioriteit aan toe te kennen in het product backlog. De product owner kan meerdere development teams aansturen.

### Scrum master

De scrum master leidt het development team door de sprints heen en zorgt ervoor dat het team gefocust blijft op de taken die gepland staan in de sprint backlog. Ook leidt hij de scrum meetings en de sprint reviews.

### Team

Het team zorgt ervoor dat alle taken die in het sprint backlog staan uitgevoerd worden. Zij maken een schatting van de tijd die het kost om een taak te voltooien en documenteren de voortgang per taak. Ook leveren ze de input tijdens alle meetings.



Figuur 7 - Rollen binnen Scrum





## 7.2 Prince2



PRINCE2 is een flexible projectbeheersmethode die gebaseerd is op de 'best practises' van bedrijven die hun eigen methodes hadden voor het beheersen van grote projecten.

Deze methode kan gebruikt worden voor software ontwikkeling maar is er niet speciaal op toegespitst, het kan gebruikt worden binnen elk project.

PRINCE2 reikt een hoop hulpmiddelen aan bij de beheersing van projecten, maar er zijn er maar een aantal cruciaal, de rest is optioneel.

Op deze manier kan PRINCE2 gevormd worden tot het aansluit bij de karakteristieken van het project.

### Business case

De business case is typerend voor PRINCE2 en verleent het project zijn bestaansrecht. Er wordt beschreven wat er gemaakt gaat worden, voor wie het product is en waar het project aan bij draagt, de zogenaamde 'benefits' en randvoorwaarden.

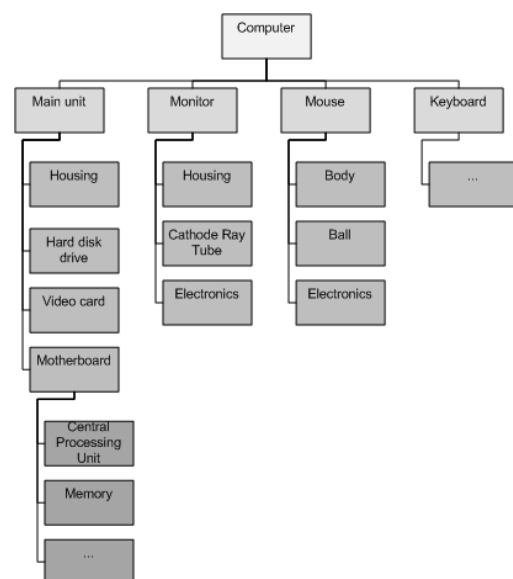
De projectleider distilleert de business case op basis van de input die gegeven wordt door de opdrachtgever. Uiteindelijk is het de opdrachtgever die akkoord moet gaan met de omschrijving van de business case.

Tijdens het project is het aan de projectleider om de business case te bewaken, als deze in gevaar komt neemt de projectleider contact op met de opdrachtgever, die uiteindelijk beslist of het project beëindigd wordt of dat de business case wordt aangepast. Een nieuwe versie van de business case vervangt altijd de oude business case die verworpen wordt.

### Product based

PRINCE2 is een product based projectbeheersingsmethode, dat wil zeggen dat producten de basis vormen van de planning binnen PRINCE2. Het eindproduct wordt opgedeeld in kleinere producten tot ze atomair zijn en niet meer opgedeeld kunnen worden. Dit wordt meestal visueel weergegeven in een 'product breakdown structure'.

De projectleider stelt op basis van de producten die gedefinieerd zijn een planning op. Vervolgens is het de taak van de projectleider om ervoor te zorgen dat het projectteam de producten volgens de planning aflevert. De acceptatie van de producten is aan de 'board' ofwel stuurgroep. De executive (opdrachtgever), senior user (hoofdgebruiker) en senior supplier (hoofdleverancier) maken deel uit van de board en zijn directe belanghebbenden van het eindproduct.



**Figuur 8 - Voorbeeld van een Product Breakdown Structure**

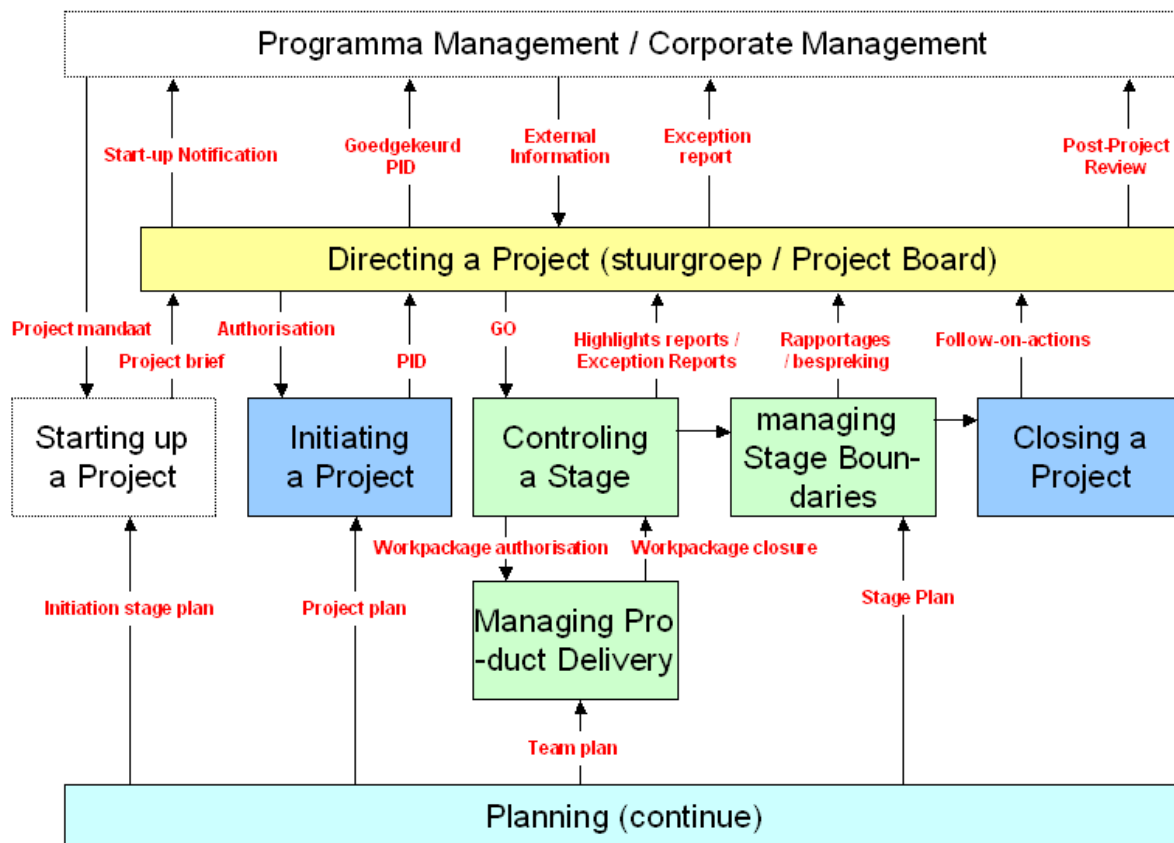


## Fases

Er zijn binnen PRINCE2 een aantal fases die doorlopen worden. De invulling van elke fase is aan de projectgroep, alle fases kunnen enorme overhead op het project produceren en het is verstandig vooraf te bepalen welke documentatie en methoden er nuttig zijn voor het project.

De eerste fase is 'starting up a project' waarmee het project wordt gestart. In deze fase verkrijgt de projectleider zijn projectmandaat van de opdrachtgever en gaat hij/zij van start met het vormen van een projectteam. De projectleider stelt een projectbrief op met daarin een omschrijving van de business case.

Vervolgens wordt er overgegaan op de fase 'Initiating a project'. Er wordt hier, na het verkrijgen van een projectmandaat, gestart met het schrijven van een project initiation document. In dit document wordt gedetailleerd vastgelegd hoe het project vormgegeven gaat worden en dient als onderlinge afspraak tussen de opdrachtgever en de projectleider over de invulling van het project.



Figuur 9 - PRINCE2 Proces

Controlling a stage / managing stage boundaries / managing product delivery maken deel uit van een iteratief proces waarin de (deel)producten ontwikkeld worden. Er worden werkpakketten verdeeld onder het team en de projectleider stelt een highlightrapport op waarin hij/zij de belangrijkste vorderingen beschrijft. Mogelijke uitzonderingen worden in het exception report opgenomen en op basis van dit document en het highlight report houdt de projectleider een bespreking met de board.



Het is de board die uiteindelijk beslist wat er moet gebeuren met de exceptions die hebben plaatstgevonden.

De laatste fase is 'closing a project' waar de projectleider zijn eindverslag schrijft met de huidige status en aanbevelingen voor de toekomst. Er wordt teruggekeken op het project door het 'lessons learned log' samen te stellen, waarmee problemen in die toekomst kunnen worden voorkomen.

Er is intussen wel een nieuwere versie van PRINCE2 beschikbaar, maar deze heb ik niet meegenomen omdat het geen verschil maakt in het karakter van PRINCE2, maar een uitbreiding/aanpassing van PRINCE2 betreft.

### 7.3 Keuze

De keuze voor een projectmanagementmethode heeft niet zozeer te maken met voor- en nadelen van de individuele methode, maar voornamelijk met het karakter van het project.

Mijn afstudeerproject is grotendeels gericht op software development waarbij het van belang is dat er goed met 'Agile' ontwikkeld kan worden met zo weinig mogelijk overhead.

In voorgaande projecten gebruikte ik PRINCE2 ook in gevallen waarbij het project software development betrof. Door PRINCE2 af te slanken tot een vorm waarmee het goed aansloot op het karakter van het project, bleek PRINCE2 flexibel genoeg om te gebruiken voor deze projecten. Het resultaat lijkt op wat Scrum al is, maar Scrum is nog beter toegespitst op software development.

De burndownchart en lage overhead zorgen ervoor dat scrum perfect aansluit op de behoefte van developers bij het beheersen van een ontwikkelproces.

Vanwege het karakter van het project heb ik uiteindelijk besloten om Scrum te gaan gebruiken als project management methode.

### 7.4 Conclusie

Vanuit de bevindingen die ik heb gedaan kan de volgende conclusie worden getrokken:

#### **Hoofdconclusie**

Scrum kan het best worden gebruikt als projectmanagement methode voor de realisatie van het iStip platform.



## 8. Usability

Met usability kan gekeken worden in hoeverre de menselijke interactie met een product gebruiksvriendelijk is. Usability wordt tegenwoordig veel gebruikt om de gebruikerservaring van websites mee te testen.

Voor het testen van usability is geen standaard ontwikkeld, er zijn wel boeken en artikelen die testen en methodes beschrijven.

Het testen van usability voor een website is kostbaar en wordt om die reden vaak overgeslagen of maar beperkt toegepast. Om goed te kunnen testen moeten er eerst tests worden opgesteld die later door een populatie van proefpersonen kunnen worden uitgevoerd.

Voor het testen zijn vele methodes ontwikkeld zo is er bijv. eye-tracking die de oogbewegingen van gebruikers volgt

bij het gebruiken van een interface.



Figuur 10 - Eye-Tracking binnen een website

Vaak wordt usability onbewust toegepast tijdens de development door in te beelden of elementen wel handig gepositioneerd zijn of door te vergelijken met andere websites. Deze manier van werken kan een prima resultaat hebben, maar kan niet garanderen dat usability overall consistent is toegepast. Het probleem dat hier ontstaat is dat de developer nooit kan overzien hoe een bepaalde doelgroep zal reageren op elementen binnen de interface.

### 8.1 Beperkt

Vanwege de korte ontwikkeltijd heb ik besloten om usability maar deels toe te passen. Door tijdens de development mijn ervaring van mijn cursus usability en de regels die gedefinieerd staan in het boek Don't Make Me Think van Steve Krug toe te passen kunnen veel valkuilen vermeden worden.

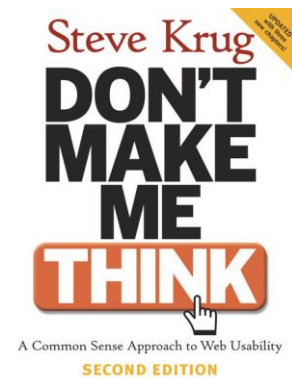
Om achteraf te testen of de applicatie goed bruikbaar is voor de gebruiker worden tests opgesteld waarmee inzicht gegeven kan worden in de usability van de applicatie. Met de beperkte tijd in ogenschouw worden er geen complexe testmethoden gebruikt, maar zal het een test betreffen in de vorm van een vragenlijst en een korte eye-tracking test.

De applicatie kan pas getest worden na dat de testomgeving is voltooid, wat betekent dat het afnemen van de tests en het interpreteren van de resultaten buiten de scope van dit project vallen.



Wel kan de applicatie vooraf getest worden door het naleven van conventies en regels zoals ze gedefinieerd zijn in het boek Don't Make Me Think van Steve Krug te testen. Daarbij worden de tests op conventies, layout en de trunk meegenomen. Deze tests worden niet door een populatie van de doelgroep uitgevoerd maar door de developer zelf.

De resultaten van deze tests kunnen geïnterpreteerd en gerapporteerd worden waarmee sturing gegeven kan worden aan de toekomstige ontwikkeling van de applicatie.



**Figuur 11 - Don't Make Me Think - Steve Krug**

## 9. Realisatie

De realisatie is een proces dat uit meerdere onderdelen bestaat. In dit hoofdstuk beschrijf ik hoe ik het realisatieproces doorlopen heb en hoe elke techniek, waarvoor eerder een keuze is gemaakt, toegepast is.

### 9.1 Procesbeschrijving

#### 9.1.1 Oriëntatie

##### *Gesprekken*

Allereerst heb ik gesprekken gevoerd met mijn opdrachtgever en begeleider om de opdracht duidelijk te kunnen formuleren. Ik heb besproken waar mijn ambities liggen binnen het project en wat een realistisch doel zou kunnen zijn voor de afstudeerperiode.

Uiteindelijk is er in het overleg getracht zo dicht mogelijk bij mijn persoonlijke leerdoelen te komen en de doelen van het bedrijf.

##### *Plan van aanpak*

In het plan van aanpak heb ik de opdracht verder uitgewerkt tot een plan. Ik heb hier een planning gemaakt, (sub)doelen gedefinieerd en randvoorwaarden opgesteld. Dit document heb ik besproken met mijn opdrachtgever en begeleider. Vervolgens heb ik het plan goed laten keuren door mijn docent/begeleider.

##### *Verdieping*

Tijdens de besprekingen met mijn begeleider heb ik vele technieken besproken die mogelijk gebruikt konden worden binnen het project. Deze technieken waren mij nog onbekend op dat moment en om mijn kennis te vergoten ben ik mij gaan verdiepen door documentatie, blogs en tutorials te gaan lezen.

Ook verscheidene concurrerende technieken heb ik bekeken om zo een goed beeld te krijgen van de mogelijkheden die er zijn.



## Keuzes

Om goede keuzes te kunnen maken voor technieken heb ik zoveel mogelijk informatie verzameld over de betreffende technieken. Uit de documentatie en vakliteratuur heb ik uiteindelijk de voor- en nadelen van elke techniek gedefinieerd.

Ik alternatieven tegenover elkaar gezet en de voor- en nadelen gewogen en getoetst aan de randvoorwaarden.

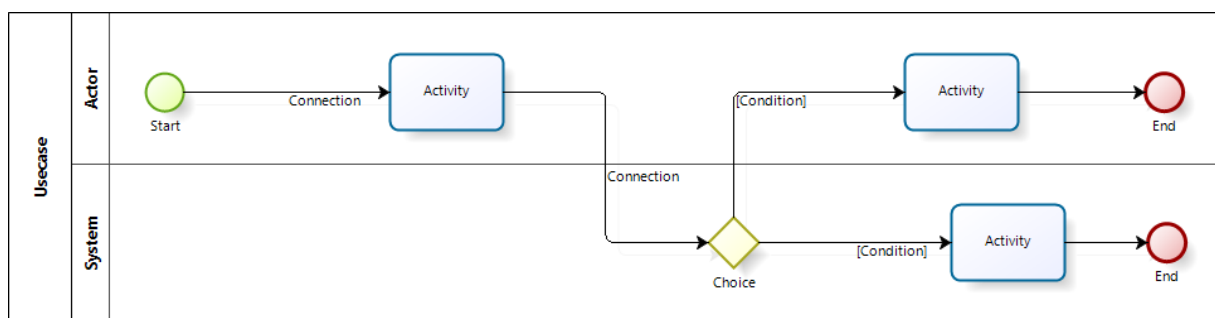
### 9.1.2 Realisatie

#### Framework installeren

Het framework heb ik geïnstalleerd door het te configureren en uit te breiden met enkele libraries. Ook de standaard mappenstructuur heb ik aangepast en uitgebreid.

#### In kaart brengen van het publicatieproces

Ik heb het publicatieproces in kaart gebracht door gesprekken te voeren met mijn opdrachtgever. Op deze manier werd duidelijk wat hij voor ogen had en verstond onder publiceren. Op basis van deze informatie heb ik een activiteitendiagram gemaakt dat het publicatieproces weergeeft. Bij het diagram heb ik usecases (functionaliteiten) gedefinieerd die beschrijven hoe de functionaliteiten opgebouwd zijn.



Figuur 12 - Voorbeeld activiteitendiagram

#### Gebruikers en gebruikersbehoeftes

Ik heb de gebruikersbehoeftes in kaart gebracht door een brainstormsessie te houden met de opdrachtgever en begeleider. De resultaten heb ik verwerkt in een document, waar ik voorstellen heb gedaan om in deze behoefte te kunnen voorzien.

Vanuit dit document heb ik wireframes ontworpen.

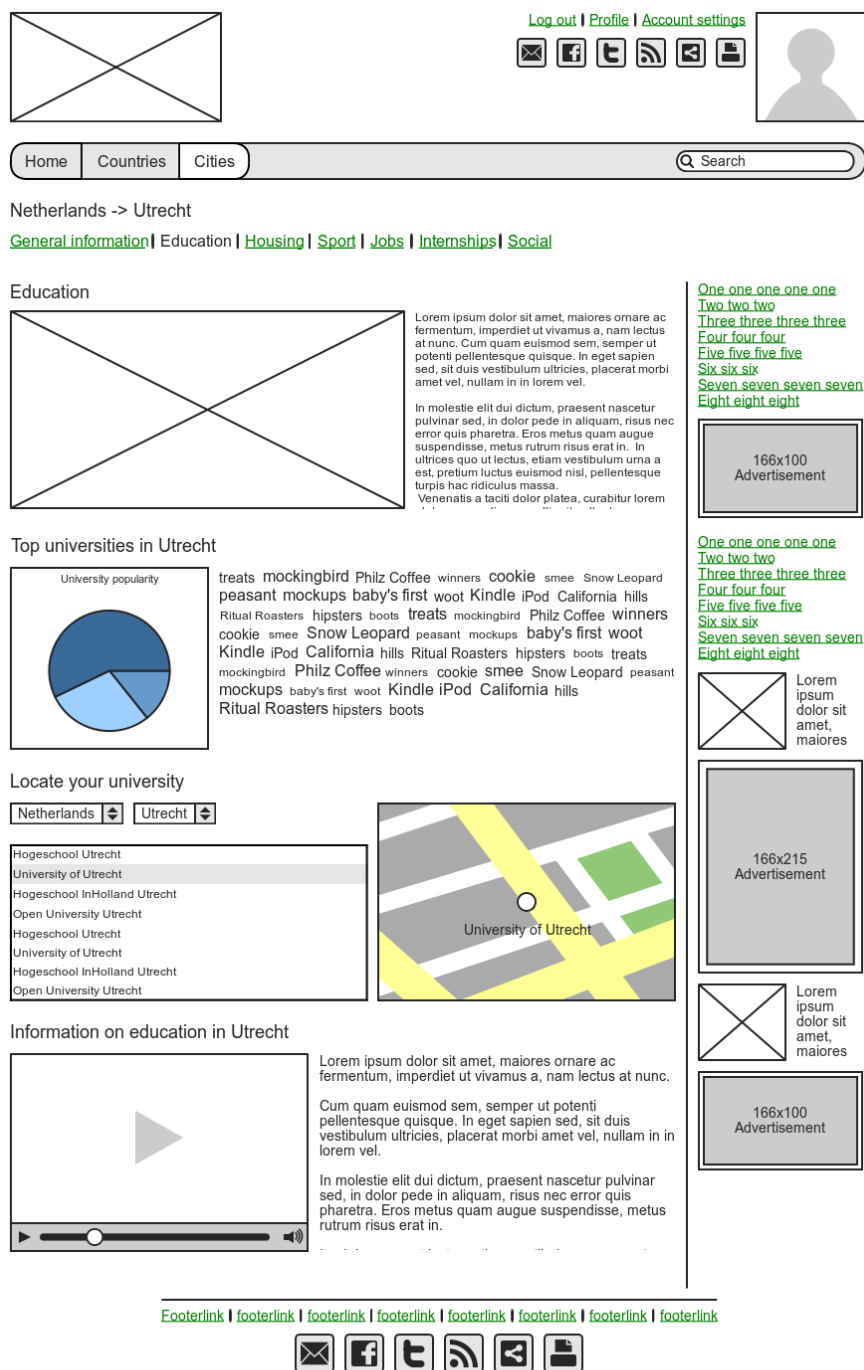
Wel heb ik aangemerkt dat behoeftes van een doelgroep in kaart brengen via een onderzoek naar de doelgroep zelf zou moeten gebeuren. De gekozen oplossing heeft te maken met tijdsdruk en de nadruk op ontwikkeling.



## Wireframes

Om de functionaliteiten binnen de interface te plaatsen heb ik wireframes gebruikt. De wireframes tonen hoe functionaliteiten zijn gepositioneerd en geven een goede indruk van de werking van de applicatie.

Wireframes heb ik zo basic mogelijk gehouden om te voorkomen dat er verwachtingen of verwarring zou ontstaan over het uiterlijk van de functionaliteiten. Ik heb ze gemaakt met de webapplicatie Mockingbird waarmee wireframes gemaakt en geëxporteerd kunnen worden.



Figuur 13 - Voorbeeld van een wireframe gemaakt met Mockingbird



## Prototypes

Met prototypes is het goed mogelijk om een beeld te schetsen van de uiteindelijke applicatie. Ik heb hiervoor dezelfde ontwikkelomgeving en tools gebruikt als voor de uiteindelijke realisatie. Bij het prototype heb ik functionaliteiten ingebouwd met volledige vormgeving.

Het verschil met een uiteindelijke realisatie is dat de fontend er weliswaar uitziet als het eindproduct, maar de backend minimaal is geïmplementeerd. Ook teksten en plaatjes zijn op dat moment opvulling en bevatten nog geen inhoud.



Figuur 14- Voorbeeld van een prototype





### **Software ontwikkeling**

Tijdens de software ontwikkeling heb ik Scrum gebruikt om het project te managen. Hiervoor neem ik een product uit de 'product-backlog' verdeel deze in taken. Vervolgens maak ik een inschatting van de hoeveelheid tijd die elke taak in beslag zal nemen.

Het realiseren zelf heb ik gedaan met behulp van de tool Netbeans, waarmee ik de applicatie uiteindelijk heb geprogrammeerd.

### **9.1.3 Testen en evaluatie**

#### **Tests opstellen**

Om de usability te kunnen meten heb ik tests opgesteld zoals ze beschreven staan in het boek Don't Make Me Think van Steve Krug.

Ik heb een trunk-test gemaakt waarmee wordt getest of de gebruiker ten alle tijden kan herleiden waar hij/zij zich bevindt.

#### **Testen**

Testen heb ik gedaan door een aantal personen te selecteren als tester. Zij maakten deel uit van de doelgroep en hadden het platform nog niet eerder gebruikt.

Ik heb de tests persoonlijk afgenomen om ervoor te zorgen dat deze ook direct uitgevoerd werden.

De resultaten heb ik verwerkt en geïnterpreteerd in een document. Het document bevat conclusies over de huidige implementatie van de interface. Op basis van de conclusies heb ik aanbevelingen gedaan over het verbeteren van usability in de toekomstige development.

#### **Evaluatie**

In de evaluatie heb ik terug gekeken op het verloop van het project en de resultaten. Om te kunnen evalueren heb ik een interne vergadering georganiseerd. De feedback van de vergadering heb ik verwerkt in een document.



## 10. Implementatie

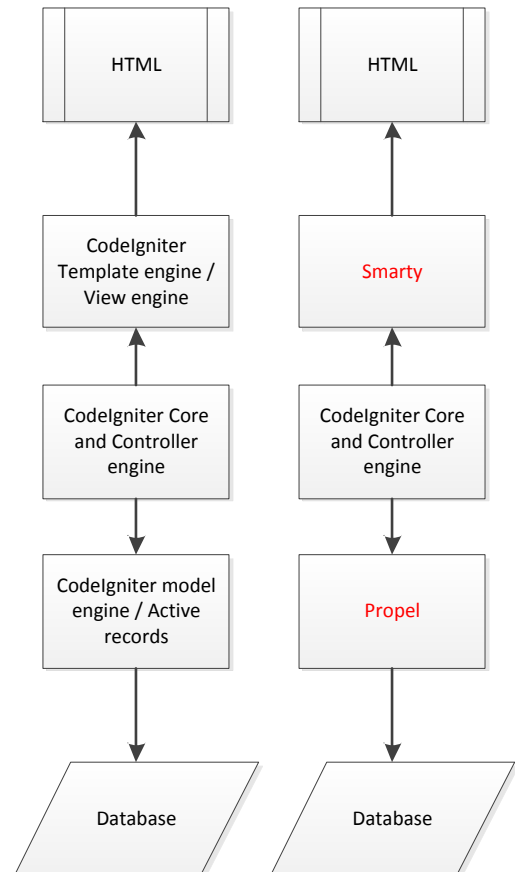
In dit hoofdstuk wordt beschreven hoe de applicatie is geïmplementeerd. Hierbij wordt een beschrijving gegeven van de aanpassingen op de standaard structuur van CodeIgniter. De aanpassingen worden visueel weergegeven en tekstueel verklaard.

### 10.1 CodeIgniter

De standaardstructuur van CodeIgniter bevatte alles dat nodig was voor een werkende webapplicatie. De applicatie is in staat business logic uit te voeren en views te tonen. Ook kan er standaard met CodeIgniter verbinding gemaakt worden met de database.

In de laag waar CodeIgniter standaard de views aanstuurt heb ik een library toegevoegd. Deze library heet Smarty en kan gebruikt worden als alternatieve template engine. Door Smarty te configureren en binnen CodeIgniter te laden kan de standaard view engine van CodeIgniter overschreden worden. Vanaf dat moment roept CodeIgniter in een controller Smarty aan om de views aan te bouwen en te tonen.

Voor de aansturing van de database, de models is ook een alternatief gebruikt. Deze is ingeladen via de library, zoals dat in CodeIgniter gebruikelijk is. Binnen de controllers kan vanaf nu propel benaderd worden om de models te gebruiken en de database aan te sturen.



Hiernaast is weergegeven hoe standaard CodeIgniter in elkaar zit en rechts daarvan de aangepaste situatie.

Deze implementatie zorgt ervoor dat CodeIgniter een stuk uitgebreider wordt in functionaliteit.



## 11. Conclusies en aanbevelingen

Op basis van het onderzoek dat ik verricht heb, zijn een aantal conclusies te trekken en aanbevelingen te doen ten aanzien van de realisatie van het iStip platform.

### *Conclusies op basis van het verrichte onderzoek*

- *Een framework sluit het beste aan bij de behoeftes van Dalen Business om snel doch schaalbaar te ontwikkelen.*
- *CodeIgniter biedt de meeste voordelen voor de realisatie van het iStip platform van alle onderzochte frameworks.*
- *Doctrine biedt op dit moment de meeste voordelen voor de realisatie van het iStip platform, maar Propel kan beter worden gebruikt vanwege de betere interne begeleiding die daarmee mogelijk is.*
- *Smarty is het meest geschikt als template engine voor de realisatie van het iStip platform.*
- *Scrum kan het best worden gebruikt als projectmanagement methode voor de realisatie van het iStip platform.*

### *Aanbevelingen voor de toekomstige development*

- **Meer onderzoek naar usability is wenselijk**  
*Het onderzoek naar de usability van de applicatie is maar in beperkte mate uitgevoerd. De prototypes die nu gemaakt zijn kunnen afwijken van het uiteindelijke product. Er zijn onvoldoende testpersonen gebruikt om sterke conclusies mee te trekken.*
- **Meer documentatie is nodig**  
*In deze periode lag de nadruk op het ontwikkelen van het platform, maar vanwege de beperkte middelen was het niet mogelijk om optimaal te documenteren. De keuze om veel aandacht te besteden aan de ontwikkeling is logisch in deze fase, maar in de toekomst zal de documentatieachterstand moeten worden ingelopen om ervoor te zorgen dat nieuwe developers goed inzicht kunnen krijgen binnen de huidige applicatiestructuur.*
- **Onafhankelijke tests zijn nodig**  
*In deze periode is de testversie gebouwd, maar nog onvoldoende getest. De ontwikkelaar zelf kan de applicatie onvoldoende testen omdat hij/zij de applicatie kent en onbewust anders reageert dan een onafhankelijk testpersoon.*



## 12. Evaluatie

Met de evaluatie kijk ik terug op het project en de resultaten. Ik beschrijf de goede punten in het project maar ook de leermomenten. De evaluatie is opgedeeld in drie gedeeltes, het proces, het product en mijn persoonlijke reflectie.

### 12.1 Procesevaluatie

Het proces heeft voor mij uit twee aspecten bestaan, namelijk: het projectmanagementproces en het realisatieproces. Door voor de projectmanagement Scrum te gebruiken als methode kon er sneller worden ontwikkeld dan ik voorheen gewend was.

De lage hoeveelheid overhead heeft ervoor gezorgd dat ik minder tijd kwijt was met documenteren dan voorheen het geval was. Ook is het handig geweest dat Scrum niet omgevormd hoefde te worden om aan te sluiten bij dit project, waar dat bij PRINCE2 wel het geval was.

Scrum blijkt een goede keuze die ik vaker zal gebruiken in toekomstige situaties.

Tijdens het opstarten van het project is het van groot belang geweest om de opdracht goed duidelijk te krijgen door gesprekken te voeren met de opdrachtgever en bedrijfsbegeleider. Deze communicatie is essentieel geweest om ervoor te zorgen dat ik de juiste sturing kon geven aan het project en zorgde ervoor dat er iets gerealiseerd zou worden waar Dalen Business ook in de toekomst nog mee zou kunnen werken.

Vanuit de gesprekken heb ik een plan van aanpak kunnen schrijven waarmee ik de invulling van het project concreet heb kunnen maken en het project heb kunnen starten. Het plan van aanpak bleek voornamelijk belangrijk om de scope van het project vast te stellen en te bewaken.

Voor er begonnen kon worden met enige realisatie was het noodzakelijk meer kennis op te doen van de technieken die mogelijk gebruikt konden worden. Door te beginnen met de vakliteratuur en documentatie door te nemen was ik beter in staat om keuzes te maken tussen verschillende technieken.

Het maken van keuzes heeft een grote rol gespeeld binnen dit project en heeft het meeste tijd in beslag genomen. Het project had natuurlijk sneller gerealiseerd kunnen worden door mijn eigen ervaringen met webdevelopment te gebruiken als basis, en geen onderzoek te doen naar andere methodes en technieken. Ik denk dat het uiteindelijk tijdbesparend zal werken door niet voor de snelle aanpak te kiezen, omdat er nu een goede basis is gecreëerd waarmee een hoop problemen al vroegtijdig zijn ondervangen, zoals bijv. schaling en uitbreidbaarheid.

Door het publicatieproces in kaart te brengen en te modelleren is het mogelijk geweest om goede functionaliteiten te creëren die het proces kunnen automatiseren. Activiteitendiagrammen bleken een goede manier om deze processen in kaart te brengen en te verduidelijken.

Het bleek lastig om uit te zoeken welke elementen er binnen een pagina gebruikt konden worden. De oplossing hiervoor was om onderzoek te doen naar de behoeftes van de doelgroep en vervolgens te kijken welke middelen daarin konden voorzien. Dit was een goede manier van werken waarmee op een doordachte wijze verschillende middelen konden worden ingezet om de gebruiker te bedienen.



Wireframes heb ik deze periode voor het eerst gemaakt met Mockingbird, het bleek een handige tool om te gebruiken. De wireframes zorgden ervoor dat er in een vroeg stadium gespeeld kon worden met de positionering van elementen, waarmee het als input diende voor gesprekken en brainstormsessies.

De prototypes zijn een onmisbaar onderdeel geweest van de realisatie. Het zorgde voor duidelijkheid bij de opdrachtgever en dat er vroegtijdig in de ontwikkeling gestuurd kon worden.

Het testen van de applicatie heb ik voornamelijk zelf gedaan en ik ben mij ervan bewust dat het onvoldoende is om van een representatief testresultaat te spreken. Het testen van de usability is ook in beperkte mate uitgevoerd vanwege de focus op ontwikkeling, maar het is wel de eerste keer dat ik nieuwe technieken als 'eye-tracking' heb toegepast.

'Eye-tracking' is een techniek die van waarde is geweest bij het analyseren en testen van usability en zal ik in de toekomst vaker en uitvoeriger gaan gebruiken.

## 12.2 Productevaluatie

Ik ben zelf erg tevreden met het eindproduct, het is ook meer geworden dan ik vooraf in de scope aan heb gegeven. De testversie is werkend en er kunnen gegevens worden gepubliceerd, bewerkt en bekeken worden.

Aan de interface heb ik veel aandacht besteed en het ziet al goed werkbaar uit. Er zijn uiteindelijk meer pagina's gemaakt dan alleen de publicatiepagina van de testversie. Er is nu een tijdelijke website die erop is gericht om nieuwe partners binnen te halen.

Ik heb het product tussentijds en afsluitend besproken met mijn opdrachtgever die erg tevreden was met het resultaat.

Het product heeft in deze periode echt vorm gekregen door constante bijsturing. Het eindproduct is een goed passend geheel waarmee iStip in de toekomst verder kan.

## 12.3 Zelfreflectie

Aan het begin van deze afstudeerperiode heb ik voor mezelf een aantal doelen gesteld. Één van de doelen was om me meer te verdiepen in techniek en beter te leren programmeren. De reden hiervoor was dat het binnen mijn opleiding (information engineering) niet veel aan bod is gekomen. Juist door dieper in de techniek te duiken denk ik dat het bij draagt aan de mate waarin ik andere programmeurs kan aansturen, in dat opzicht is deze stage aanvullend geweest aan mijn opleiding.

Ik wilde graag een product afleveren dat in de toekomst gebruikt zou worden en ook volgens de huidige normen van webdevelopment ontwikkeld werd. Dat heb ik gedaan door samen met mijn bedrijfsbegeleider ervoor te kiezen om complexe technieken niet te mijden, maar bijv. een framework met ORM en template engine te verkiezen boven de eenvoudigere route van een CMS.



Ik heb veel geleerd van deze afstudeerstage en nieuwe inzichten verkregen in het ontwikkelen van (web)applicaties. Het was ook leuk om aan een nieuw concept te werken waarbij alle opties nog open stonden en ik vrij was in mijn keuzes voor technieken en invulling van het project.

De goede begeleiding binnen het bedrijf heeft ervoor gezorgd dat ik gemotiveerd kon werken aan het product en problemen nooit lang voort hebben gesleept.

Ik ben blij met de uiteindelijke ontwikkeling die ik heb kunnen doormaken deze afstudeerperiode waarmee ik mijn kennisgebied zeer verbreed heb.



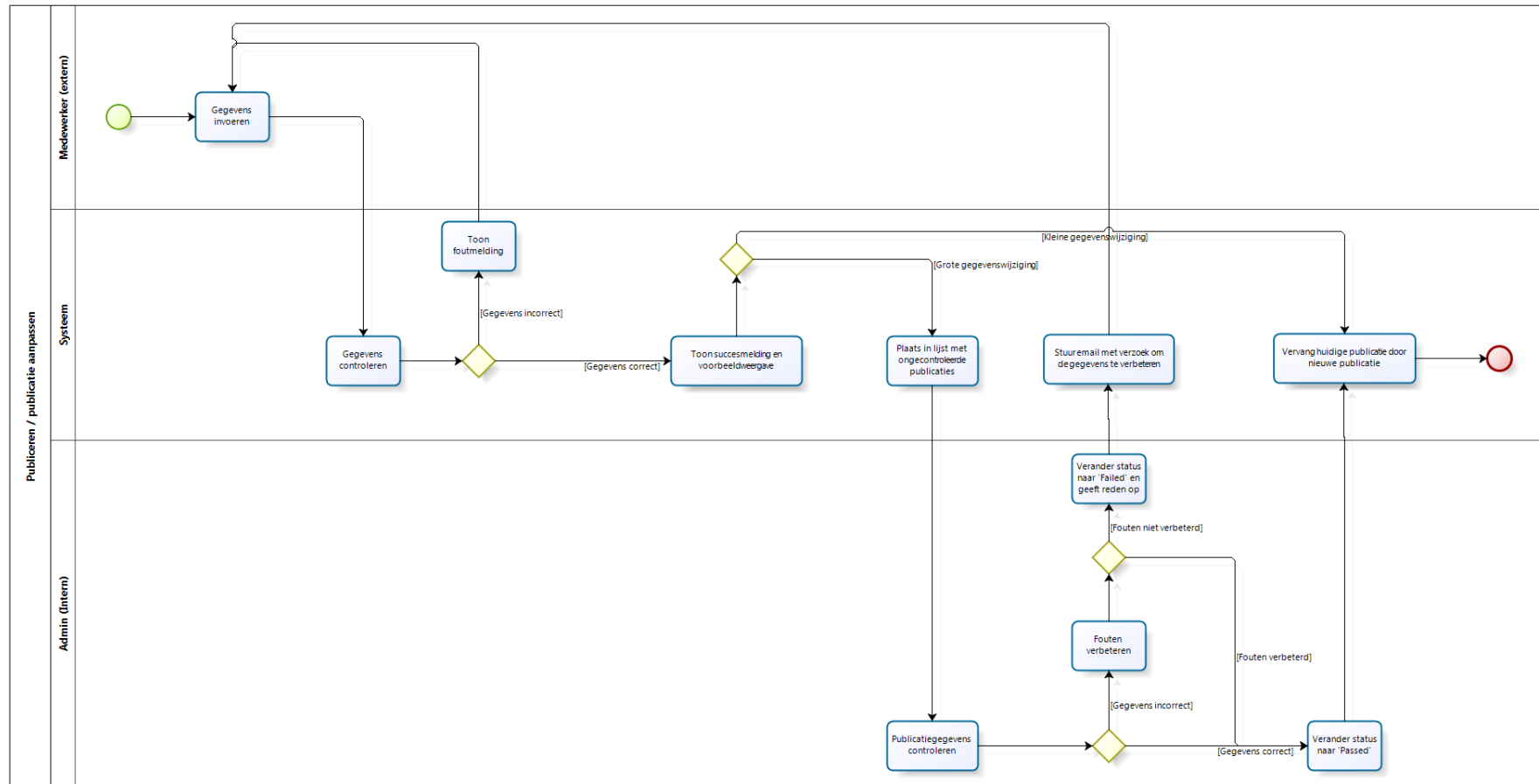
### 13. Literatuurlijst

- Smarty Manual | Smarty.* (2002). Opgeroepen op November, December 2010, van smarty.net:  
<http://www.smarty.net/docsv2/en/>
- Kijkgedrag op een menustructuur van een website.* (2007). Opgeroepen op December 2010, van igitur-archive.library.uu.nl: <http://igitur-archive.library.uu.nl/student-theses/2007-0620-200724/UUindex.html>
- My First Agile Project, Part 1: Doing 80% | Agile Software Development.* (2008, September). Opgeroepen op September 2010, van agilesoftwaredevelopment.com:  
<http://agilesoftwaredevelopment.com/blog/mattgrommes/matt-grommes-my-first-agile-project-part-1>
- Documentation/1.4 - Propel ORM.* (2009, November). Opgeroepen op Oktober, November, December 2010, van propelorm.org: <http://www.propelorm.org/wiki/Documentation/1.4>
- All things CakePHP :: The Cookbook.* (2010). Opgeroepen op Oktober, November 2010, van cakephp.org: <http://book.cakephp.org/>
- Doctrine - Object Relational Mapper Documentation (2.0).* (2010). Opgeroepen op Oktober, November, December 2010, van doctrine-project.org: <http://www.doctrine-project.org/projects/orm/2.0/docs/en>
- EyeTracking, Inc - Solutions - Interface Design.* (2010). Opgeroepen op Oktober 2010, van eyetracking.com: [http://www.eyetracking.com/solutions/interface\\_usability/Default.aspx](http://www.eyetracking.com/solutions/interface_usability/Default.aspx)
- PHP Frameworks.* (2010). Opgeroepen op September, Oktober 2010, van phpframeworks.com: <http://www.phpframeworks.com/>
- PRINCE2 Downloads - Process Model, PRINCE2 Mini Quiz and many more useful PRINCE2 Resources - ILX Group UK.* (2010). (ILX Group UK) Opgeroepen op September, Oktober 2010, van prince2.com: <http://www.prince2.com/nl/prince2downloads.asp>
- Savant | Documentation.* (2010). Opgeroepen op November, December 2010, van phpsavant.com: <http://phpsavant.com/docs/>
- Scrum Alliance - What Is Scrum?* (2010). Opgeroepen op September, Oktober 2010, van scrumalliance.org: [http://www.scrumalliance.org/pages/what\\_is\\_scrum](http://www.scrumalliance.org/pages/what_is_scrum)
- The Definitive Guide to Yii | Yii Framework.* (2010). Opgeroepen op Oktober, November 2010, van yiiframework.com: <http://www.yiiframework.com/doc/guide/1.1/en/index>
- Website wireframes: Mockingbird.* (2010). Opgeroepen op September, Oktober 2010, van gomockingbird.com: <https://gomockingbird.com/mockingbird/>
- Welcome to CodeIgniter: CodeIgniter User Guide.* (2010). Opgeroepen op September, Oktober, November, December 2010, van CodeIgniter.com: [http://codeigniter.com/user\\_guide/](http://codeigniter.com/user_guide/)
- Zend Framework.* (2010). (Zend) Opgeroepen op Oktober, November 2010, van framework.zend.com: <http://framework.zend.com/docs/overview>
- Krug, S. (2000). *Don't Make Me Think* (1st edition ed.). New Riders Press.



## Bijlages

### Bijlage 1: Publicatieproces





## Bijlage 2: Usecases

### Usecase 1: Onderwijsinstelling bekijken

Actor(s): Bezoeker, Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor**, **systeem**

Vereiste: Is op de onderwijsinstellingenpagina –pagina.

1. Actor klikt op naam onderwijsinstelling.
2. Systeem toont onderwijsinstelling.

### Usecase 2: Onderwijsinstelling onderwerp toevoegen

Actor(s): Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor**, **systeem**

Vereiste: Is ingelogd en op de betreffende onderwijsinstellingpagina.

1. Actor klikt op onderwerp toevoegen
2. Systeem toont formulier met lege velden en dropdownmenu met mogelijke onderwerpen en de mogelijkheid om zelf een onderwerp te definiëren.
3. Actor vult gegevens in en klikt op 'toevoegen'.
4. Systeem controleert de velden.
  - 4.1 [Niet succesvol] Systeem toont foutcode en zet focus op betreffende veld, terug naar 1.
  - 4.2 [Succesvol] Systeem voegt gegevens toe aan de database. Toont melding met een voorbeeld van het onderwerp en geeft optie om nog een onderwerp toe te voegen of terug te gaan naar de site van de onderwijsinstelling.

### Usecase 3: Onderwijsinstelling onderwerp wijzigen

Actor(s): Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor**, **systeem**

Vereiste: Is ingelogd en op de betreffende onderwijsinstellingpagina.

1. Actor klikt op onderwerp wijzigen
2. Systeem toont formulier met gevulde velden die de huidige gegevens weergeven.
3. Actor wijzigt de gegevens en klikt op 'Wijzigen'.
4. Systeem controleert de velden.
  - 4.1 [Niet succesvol] Systeem toont foutcode en zet focus op betreffende veld, terug naar 1.
  - 4.2 [Succesvol] Systeem voegt gegevens toe aan de database. Toont melding en voorbeeld van het onderwerp en geeft optie om het onderwerp te bekijken, er nog maals te wijzigen of terug te gaan naar de site van de onderwijsinstelling.

### Usecase 4: Onderwijsinstelling onderwerp verwijderen

Actor(s): Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor**, **systeem**

Vereiste: Is ingelogd.

1. Actor klikt op onderwerp verwijderen
2. Systeem toont een melding waar om bevestiging wordt gevraagd voor het verwijderen.
  - 3.1 [Actor klikt op verwijderen] Door naar 4
  - 3.2 [Actor klikt op annuleren] Door naar 5

4. Systeem verwijdert het onderwerp (wordt op inactief gezet).
5. Systeem toont de pagina van de onderwijsinstelling.

### Usecase 5: Onderwijsinstelling bestand downloaden

Actor(s): Bezoeker, Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor, systeem**

Vereiste: Is op de onderwijsinstellingpagina

1. Actor klikt op het betreffende bestand.
2. Systeem haalt het betreffende bestand op.

### Usecase 6: Onderwijsinstelling bestand toevoegen

Actor(s): Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor, systeem**

Vereiste: Is ingelogd en bij de pagina onderwerp toevoegen.

1. Actor klikt op bestand toevoegen.
2. Systeem toont file browser.
3. Actor selecteert het gewenste bestand en drukt op toevoegen.
4. Systeem controleert het bestand op grootte en extensie.
  - 4.1 [Niet succesvol] Systeem toont foutcode , terug naar 1.
  - 4.2 [Succesvol] Systeem slaat het bestand op en toont melding naar de gebruiker.

### Usecase 7: Onderwijsinstelling bestand verwijderen

Actor(s): Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor, systeem**

Vereiste: Is ingelogd en op onderwijsinstellingen pagina.

1. Actor klikt op bestand verwijderen
2. Systeem toont een melding waar om bevestiging wordt gevraagd voor het verwijderen.
  - 3.1 [Actor klikt op verwijderen] Door naar 4
  - 3.2 [Actor klikt op annuleren] Door naar 5
4. Systeem verwijdert het bestand en toont de onderwijsinstellingen pagina.
5. Systeem toont de pagina van de onderwijsinstelling.

### Usecase 8: Onderwijsinstelling video toevoegen

Actor(s): Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor, systeem**

Vereiste: Is ingelogd en bij de pagina onderwerp toevoegen.

1. Actor voegt het ID in van de gewenste video en klikt op toevoegen.
2. Systeem controleert het ID van de video.
  - 4.1 [Niet succesvol] Systeem toont foutcode , terug naar 1.
  - 4.2 [Succesvol] Systeem slaat het bestand op en toont melding naar de gebruiker.



## Usecase 9: Onderwijsinstelling video verwijderen

Actor(s): Admin, Onderwijsinstelling (eigenaar)

Kleur: **Actor**, **systeem**

Vereiste: Is ingelogd en op onderwijsinstellingen pagina.

1. Actor klikt op video verwijderen
2. Systeem toont een melding waar om bevestiging wordt gevraagd voor het verwijderen.
  - 3.1 [Actor klikt op verwijderen] Door naar 4
  - 3.2 [Actor klikt op annuleren] Door naar 5
4. Systeem verwijdert de video en en toont onderwijsinstellingen pagina.
5. Systeem toont de pagina van de onderwijsinstelling.



## Bijlage 3: User needs

### Home-pagina

De home-pagina is een landingspagina die als doel heeft om gebruikers door te linken naar de informatie waar de gebruiker naar op zoek is. Dit is de eerste pagina die de gebruikers te zien krijgen, om deze reden is het van belang dat de home-pagina de uitstraling heeft die bij het platform past.

[vervolgens de uitstraling die bij het platform past, zoals: betrouwbaar, informatief, compleet, dynamisch, modern]

Gebruikers van deze pagina zijn te onderscheiden in verschillende gebruikersgroepen, namelijk:

- *Studenten*
- *Docenten [krijgt later invulling]*
- *Medewerkers van bedrijven, organisaties en onderwijsinstellingen*

Voor de groepen studenten en docenten geldt dat er een onderscheid wordt gemaakt tussen 'nieuwe gebruikers' en 'terugkerende gebruikers'.

### Studenten – Nieuwe gebruikers

Studenten zijn de belangrijkste doelgroep van het iStip platform en vertegenwoordigen waarde voor het platform en de derde partijen die samenwerken met het iStip. Het doel van het iStip platform is om zoveel mogelijk studenten van goede, betrouwbare en complete informatie te voorzien.

Om ervoor te zorgen dat studenten daadwerkelijk de informatie vinden die zij nodig hebben moet er op de home-pagina in de volgende behoeftes worden voorzien:

- *Uitleg geven over wat het iStip platform is*
- *Uitleg geven over hoe het iStip platform gebruikt kan worden*
- *Overtuigen dat het iStip platform een betrouwbare informatiebron is*

### Uitleg geven over wat het iStip platform is

Deze informatie moet direct beschikbaar zijn voor de gebruiker, omdat het waarschijnlijk de eerste vraag is die de gebruiker zich stelt en hij al snel kan afhaken als het doel van de website niet duidelijk is. Om de gebruiker van deze informatie te kunnen voorzien moet er op de homepage een button met 'What is iStip' prominent aanwezig zijn.

Ook door middel van een korte tekst of slogan kan uitleg gegeven worden over het platform.



***Uitleg geven over hoe het iStip platform gebruikt kan worden***

Kan gedaan worden door een uitleg te geven bij de pagina 'What is iStip'. Bij de stedenkiezer kan ook kort aangegeven worden welke stappen de gebruiker moet ondernemen bijv.:

1. *Pick a country*
2. *Pick a city*
3. *Find information on your city of choice*

***Overtuigen dat het iStip platform een betrouwbare informatiebron is***

Dit kan door te verwijzen naar betrouwbare partners, zoals universiteiten en internationale organisaties.

Het is ook van belang dat de informatie op de home-pagina up-to-date is, oude informatie op de home-pagina geeft een weinig betrouwbare indruk voor de gehele website.

Het uiterlijk draagt ook bij aan de perceptie van betrouwbaarheid. In veel gevallen wordt een commerciële uitstraling gezien als een partijdige informatiebron, het is daarom een goed idee om geen of advertenties van betrouwbare corporaties en organisaties op de home-pagina te plaatsen.

***Studenten – Terugkerende gebruikers***

Terugkerende studenten hebben de website eerder geraadpleegd en weten hoe ze de site moeten gebruiken. Ze komen terug naar de website met een gerichte zoekvraag en willen zo snel mogelijk deze zoekvraag beantwoord hebben.

Voor terugkerende studenten moet de home-pagina in de volgende behoeftes voorzien:

- *Direct kunnen navigeren naar een land of stad*
- *Direct kunnen zoeken*

***Direct kunnen navigeren naar een land of stad***

Om direct naar een land of stad te kunnen navigeren kan op de home-pagina gebruik gemaakt worden van een steden/landen kiezer. Een kaart van Nederland of Europa waar binnen twee klikken de bestemming kan worden geselecteerd.

Deze steden/landen kiezer mag aanzienlijk veel ruimte innemen en moet direct zichtbaar zijn boven aan de pagina, omdat de steden en landen de kern zijn van de informatievoorziening van het iStip platform.



***Direct kunnen zoeken***

De student met een gerichte zoekvraag wil zo snel mogelijk antwoord op zijn vraag. Hij kent de navigatie en weet welke informatie er gevonden kan worden op het platform.

Aan deze behoefte, om direct de zoekvraag beantwoord te hebben zonder daarvoor teveel te hoeven navigeren, kan een zoekbalk gebruikt worden. Hier geeft de gebruiker zijn zoekterm op met een paar steekwoorden en krijgt vervolgens een lijst met resultaten.

Op deze wijze kan de student direct toegang krijgen tot de juiste informatie.

***Medewerkers van bedrijven, organisaties en onderwijsinstellingen***

Deze groep gebruikers zijn interessant voor het iStip platform, omdat zij mogelijk een samenwerkingsverband willen aangaan met iStip. Voor nieuwe gebruikers van deze doelgroep dient de home-pagina in de volgende behoeftes te voorzien:

- *Uitleg geven over wat het iStip platform is*
- *Tonen welke partijen er al samenwerken met het iStip platform*

***Uitleg geven over wat het iStip platform is***

Deze informatie moet direct beschikbaar zijn voor de gebruiker, omdat het waarschijnlijk de eerste vraag is die de gebruiker zich stelt en hij al snel kan afhaken als het doel van de website niet duidelijk is. Om de gebruiker van deze informatie te kunnen voorzien moet er op de homepagina een button met 'What is iStip' prominent aanwezig zijn.

Ook door middel van een korte tekst of slogan kan uitleg gegeven worden over het platform.

***Tonen welke partijen er al samenwerken met het iStip platform***

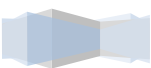
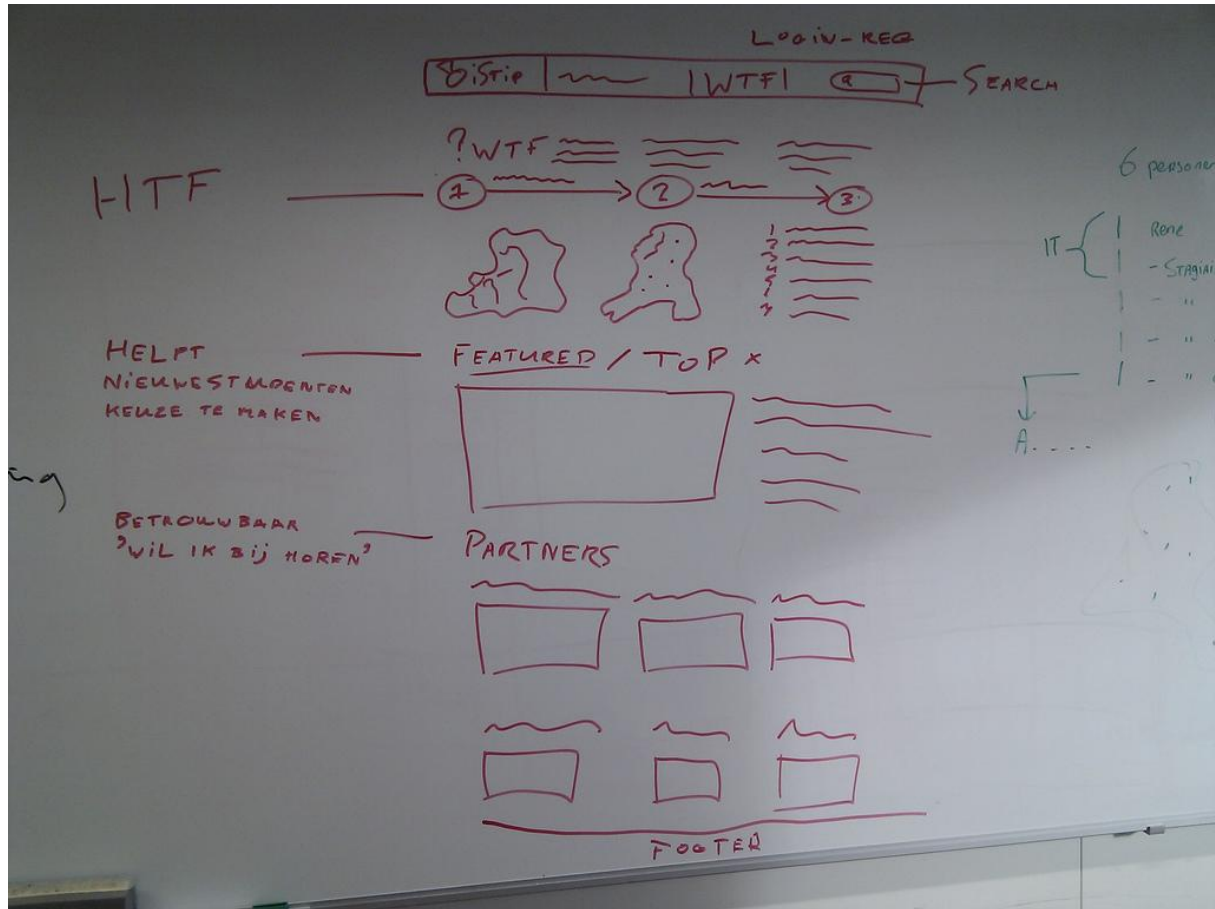
Hiermee wordt aangetoond dat iStip een platform is dat een betrouwbare partner kan zijn om mee samen te werken. Tevens is het zo dat het sterk een gevoel oproept van 'you are missing out'. Als de andere partijen al betrokken zijn, dan moet het wel meerwaarde hebben voor die partijen.

Deze informatie kan getoond worden door op de home de logo's van de partijen weer te geven die al betrokken zijn bij het iStip platform. In één oogopslag is het duidelijk dat iStip een platform is waar je als bedrijf, organisatie of onderwijsinstelling bij wilt horen.

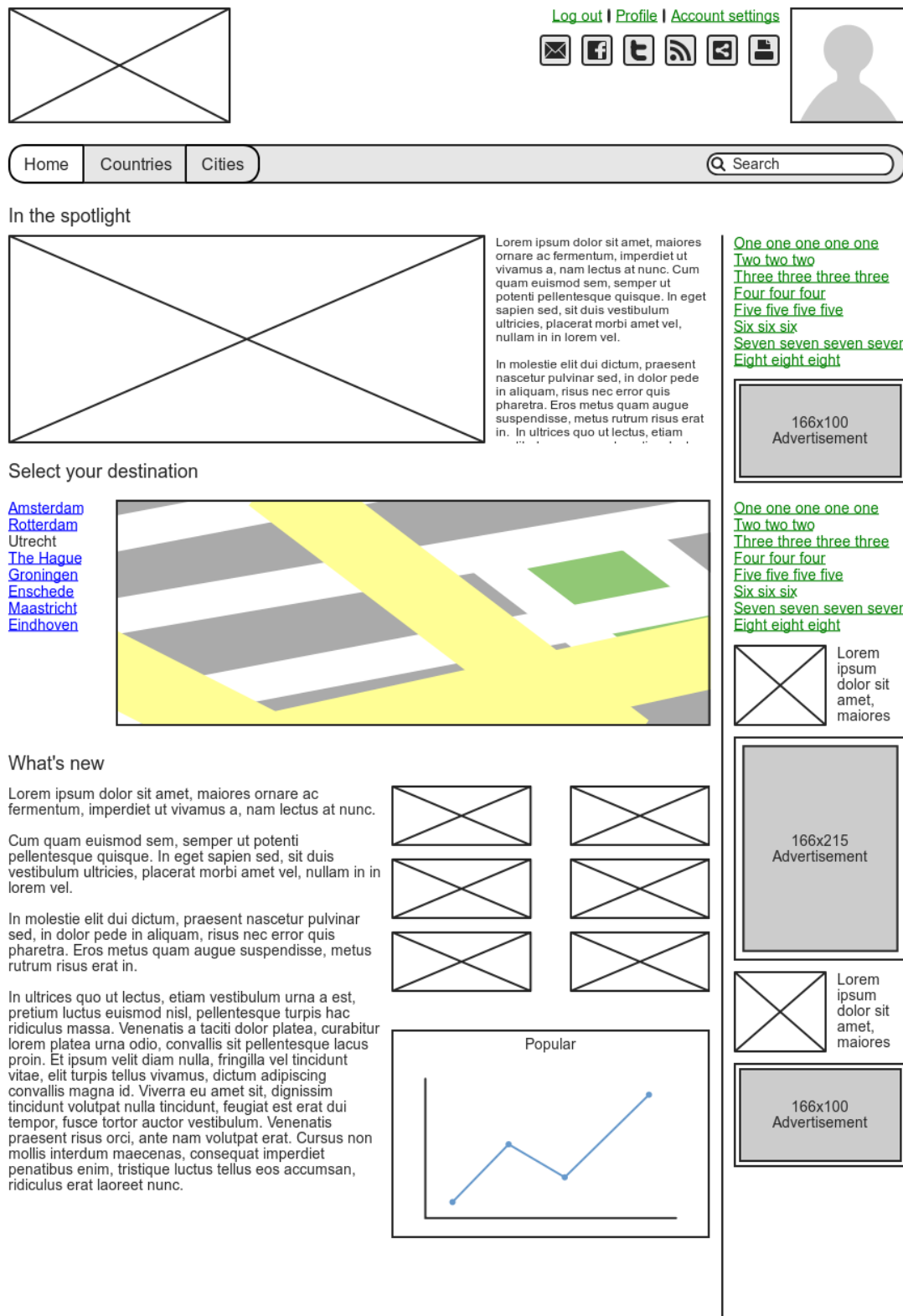
Gedetailleerdere informatie over de lopende samenwerking kan opgenomen worden in een aparte pagina waar een link naartoe gaat vanaf de home pagina, bijv. 'What is iStip'.



## Wireframe

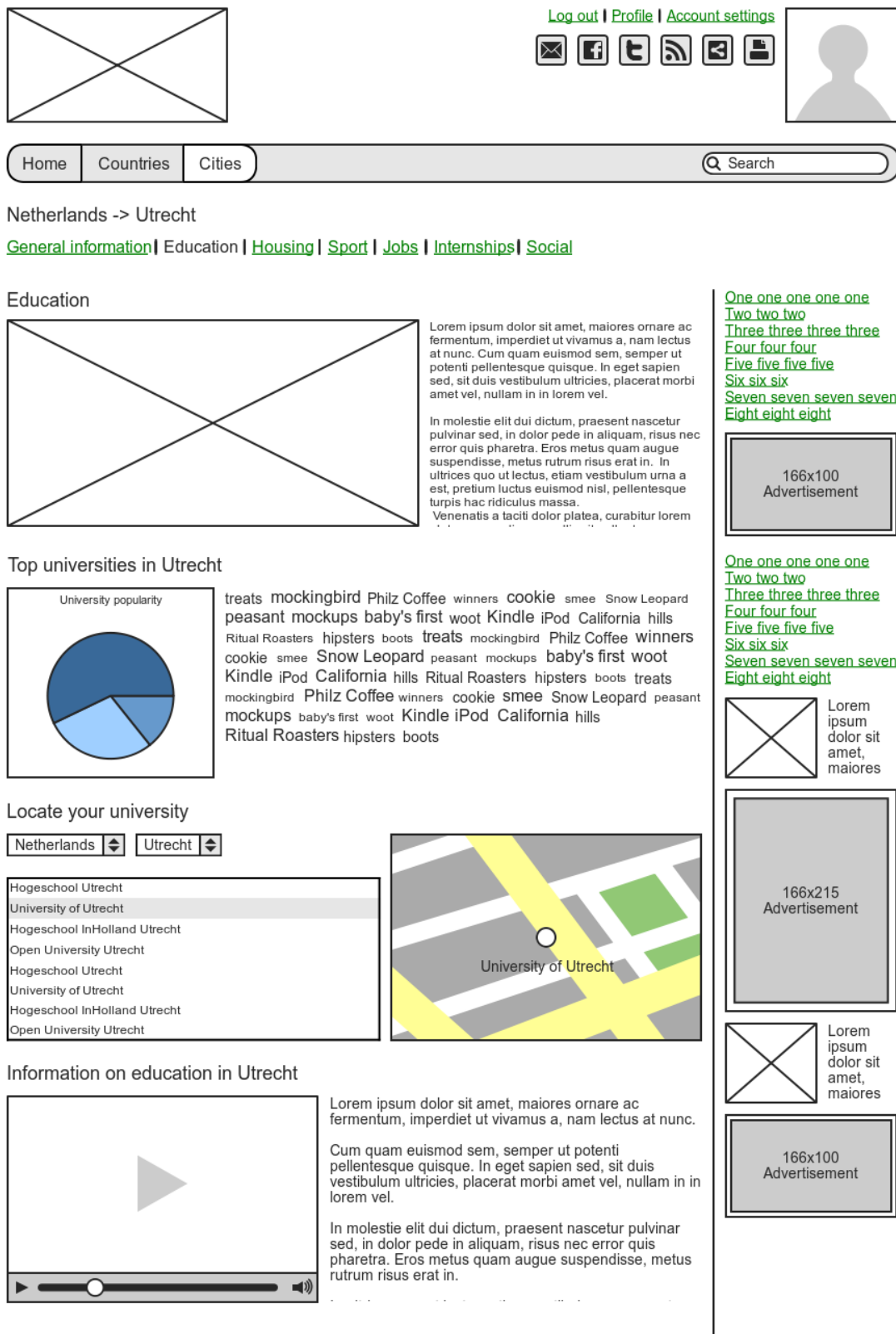


## Bijlage 4: Home wireframe

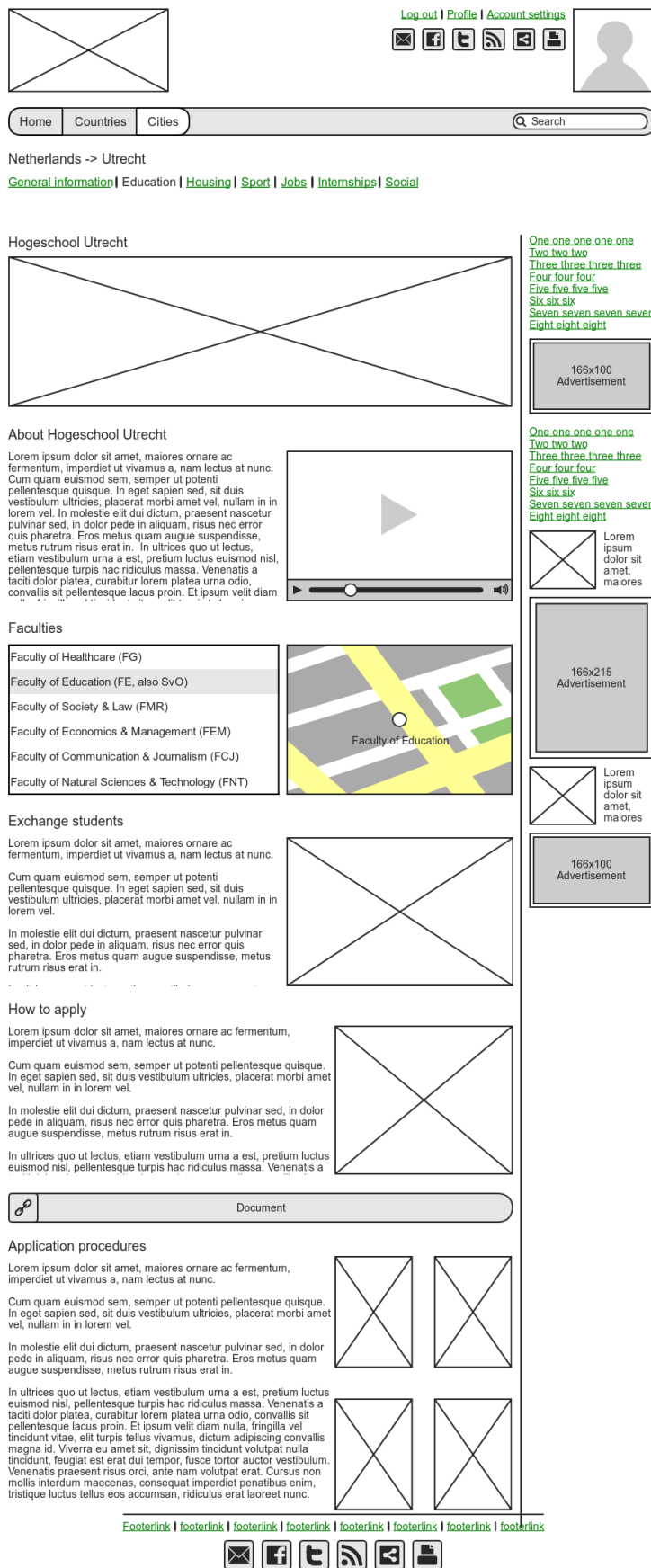




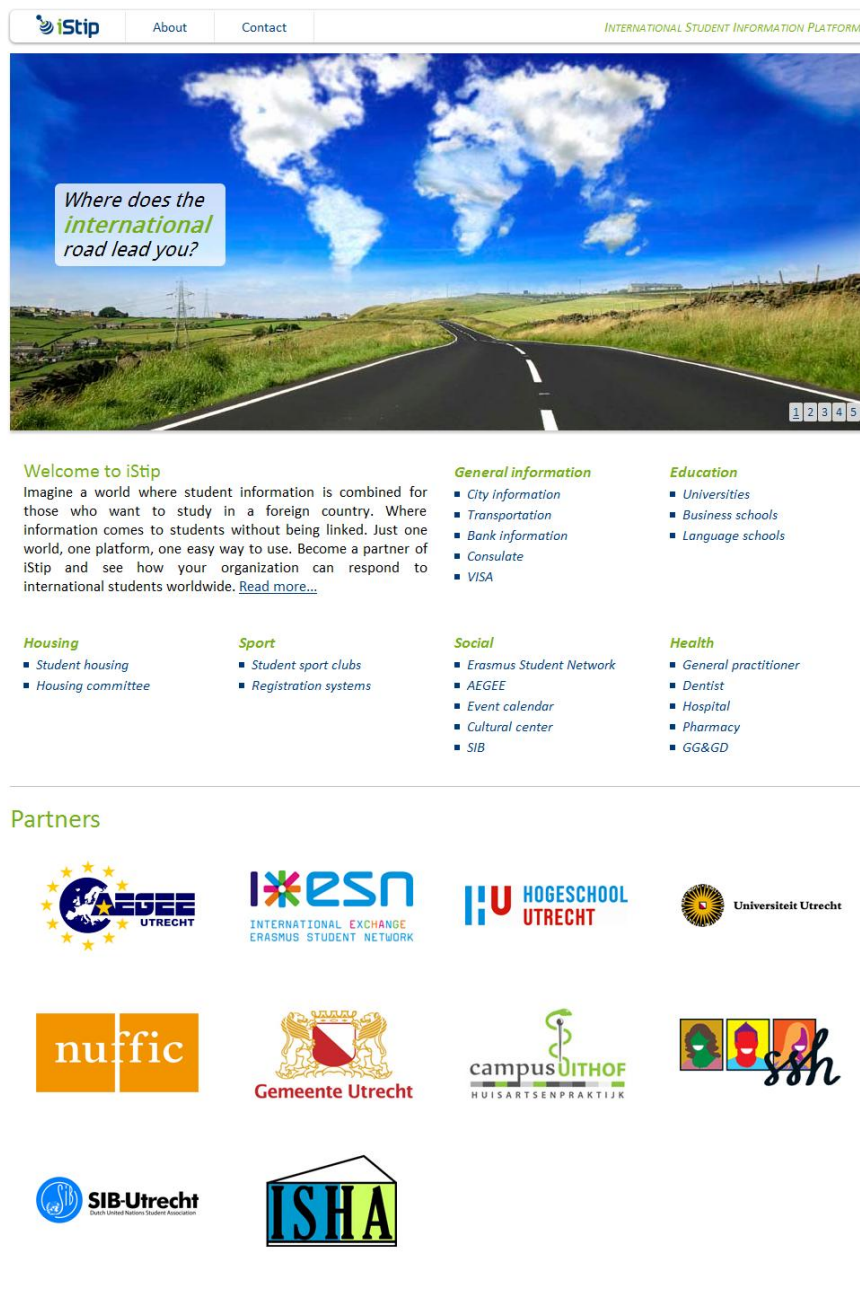
## Bijlage 5: Education wireframe



## Bijlage 6: University wireframe



## Bijlage 7: Home prototype

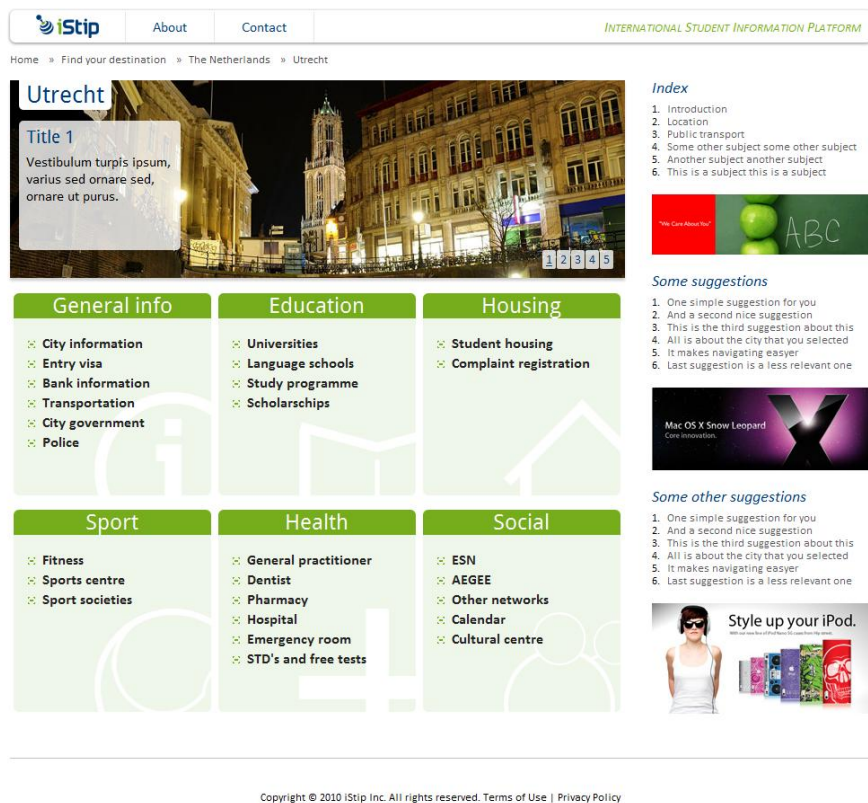


Copyright © 2010 iStip Inc. All rights reserved. Terms of Use | Privacy Policy

## Bijlage 8: Country wireframe



## Bijlage 9: City wireframe





## Bijlage 10: General information wireframe


[About](#)
[Contact](#)

INTERNATIONAL STUDENT INFORMATION PLATFORM

[Home](#) » [Find your destination](#) » [The Netherlands](#) » [Utrecht](#) » [General information](#)

[General information](#) | [Education](#) | [Housing](#) | [Sport](#) | [Jobs](#) | [Internships](#) | [Social](#)

### 1. Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum turpis ipsum, varius sed ornare sed, ornare ut purus. Nulla lobortis dictum nibh nec dapibus. Maecenas tempor tristique orci ut ultrices. Sed aliquet risus ac leo facilisis tincidunt. Aenean tempus nisi ac nunc fringilla in convallis lectus laoreet. Fusce ornare ultricies nunc, vel dapibus justo fermentum sed.

Am adipiscing ultricies feugiat. Fusce a libero tellus. Etiam condimentum molestie imperdiet. Integer luctus condimentum volutpat. Nam semper, eros vel viverra euismod, metus enim pharetra diam, eget consequat tortor dui feugiat mi. Aliquam ullamcorper tincidunt lorem, in convallis mauris volutpat a. Am adipiscing ultricies feugiat. Fusce a libero tellus. Am adipiscing ultricies feugiat.



**Utrecht introduced by boat**

Donec nec interdum nibh. Etiam at lectus cursus arcu pellentesque imperdiet id vitae nisi. Mauris vel mi magna. Ut fermentum aliquam neque, at eleifend libero lobortis at. Vestibulum accumsan eros et orci fermentum pretium.

### 2. Location

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum turpis ipsum, varius sed ornare sed, ornare ut purus. Nulla lobortis dictum nibh nec dapibus. Maecenas tempor tristique orci ut ultrices.

Sed aliquet risus ac leo facilisis tincidunt. Aenean tempus nisi ac nunc fringilla in convallis lectus laoreet. Fusce ornare ultricies nunc, vel dapibus justo fermentum sed.

Integer luctus condimentum volutpat. Nam semper, eros vel viverra euismod, metus enim pharetra diam, eget consequat tortor dui feugiat mi.



### 3. Public transport

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum turpis ipsum, varius sed ornare sed, ornare ut purus. Nulla lobortis dictum nibh nec dapibus. Maecenas tempor tristique orci ut ultrices. Sed aliquet risus ac leo facilisis tincidunt. Aenean tempus nisi ac nunc fringilla in convallis lectus laoreet. Fusce ornare ultricies nunc, vel dapibus justo fermentum sed.

Am adipiscing ultricies feugiat. Fusce a libero tellus. Etiam condimentum molestie imperdiet. Integer luctus condimentum volutpat. Nam semper, eros vel viverra euismod, metus enim pharetra diam, eget consequat tortor dui feugiat mi.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum turpis ipsum, varius sed ornare sed, ornare ut purus. Nulla lobortis dictum nibh nec dapibus. Maecenas tempor tristique orci ut ultrices. Sed aliquet risus ac leo facilisis tincidunt. Aenean tempus nisi ac nunc fringilla in convallis lectus laoreet. Fusce ornare ultricies nunc, vel dapibus justo fermentum sed. Fusce ornare ultricies nunc, vel dapibus justo fermentum sed. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing.



**Rush hour in Utrecht**

Donec nec interdum nibh. Etiam at lectus cursus arcu pellentesque imperdiet id vitae nisi. Mauris vel mi magna. Ut fermentum aliquam neque, at eleifend libero lobortis at. Vestibulum accumsan eros et orci fermentum pretium. Vestibulum accumsan eros et orci fermentum pretium.

#### Index

1. Introduction
2. Location
3. Public transport
4. Some other subject some other subject
5. Another subject another subject
6. This is a subject this is a subject



#### Some suggestions

1. One simple suggestion for you
2. And a second nice suggestion
3. This is the third suggestion about this
4. All is about the city that you selected
5. It makes navigating easier
6. Last suggestion is a less relevant one



#### Some other suggestions

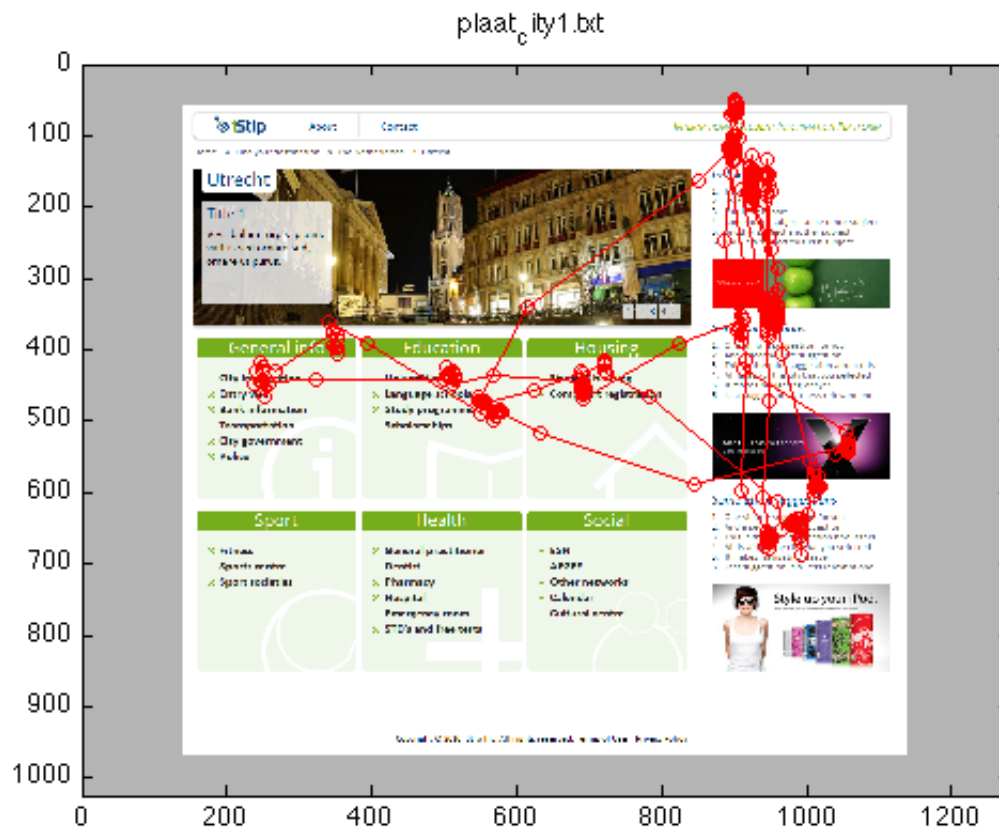
1. One simple suggestion for you
2. And a second nice suggestion
3. This is the third suggestion about this
4. All is about the city that you selected
5. It makes navigating easier
6. Last suggestion is a less relevant one



Copyright © 2010 iStip Inc. All rights reserved. Terms of Use | Privacy Policy

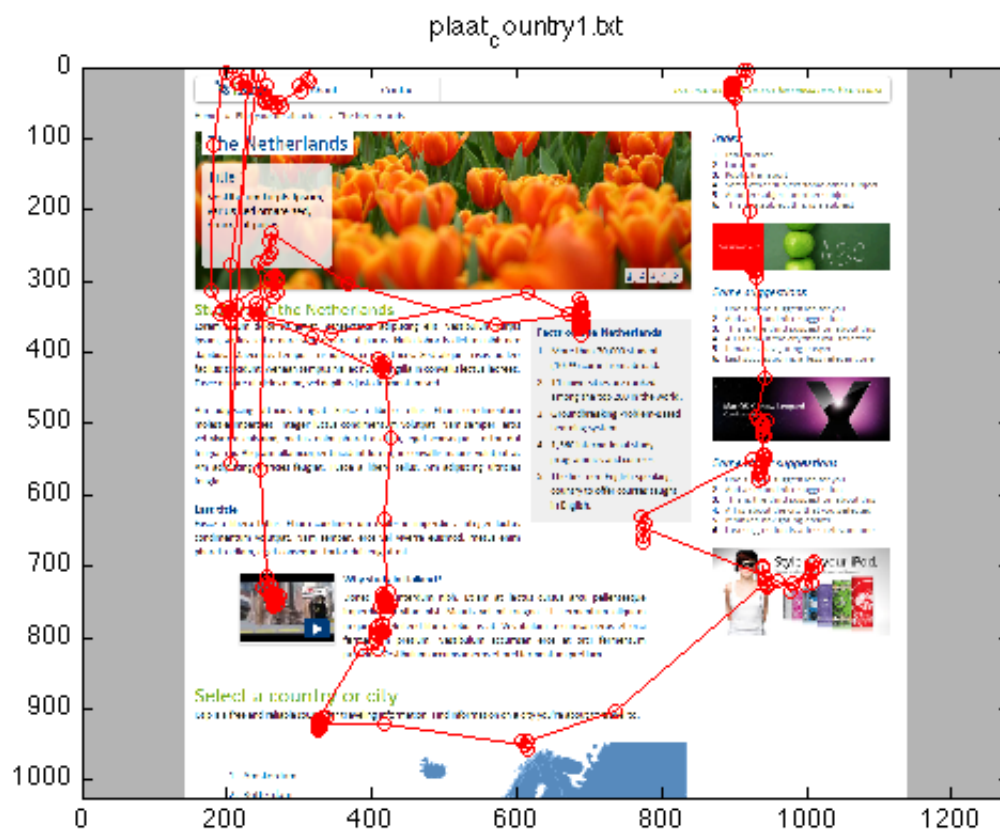


## Bijlage 11: Eye tracking results



Figuur 15 - Expert, city

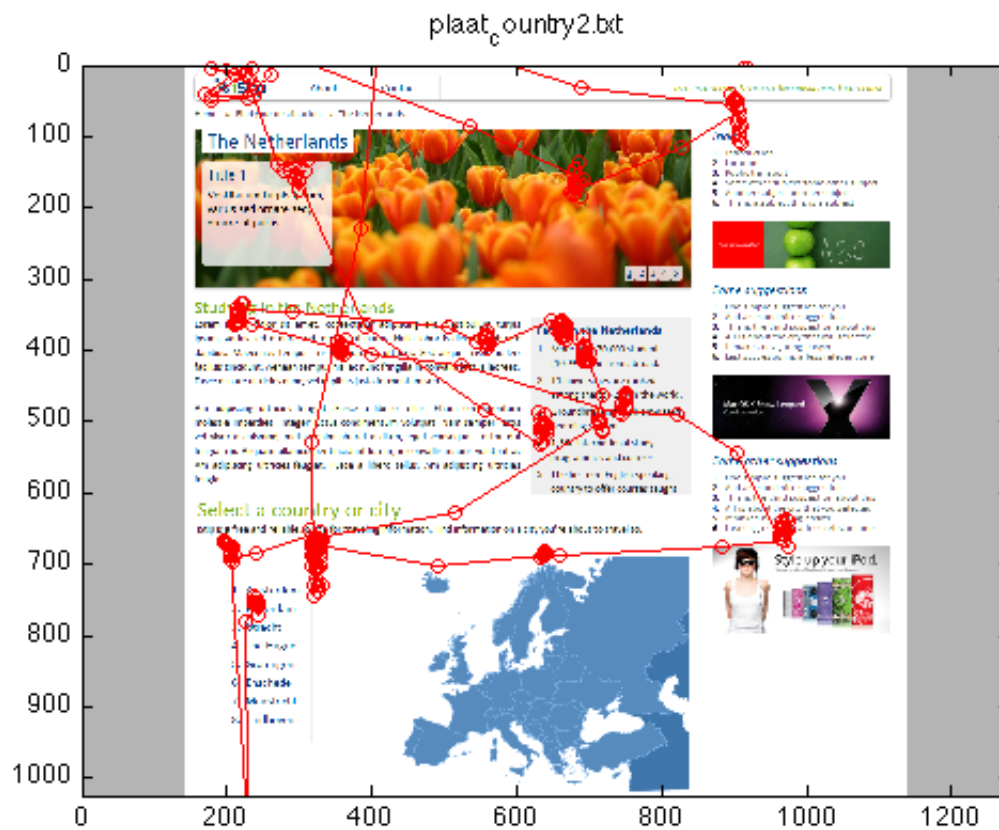




Figuur 16 - Expert, country

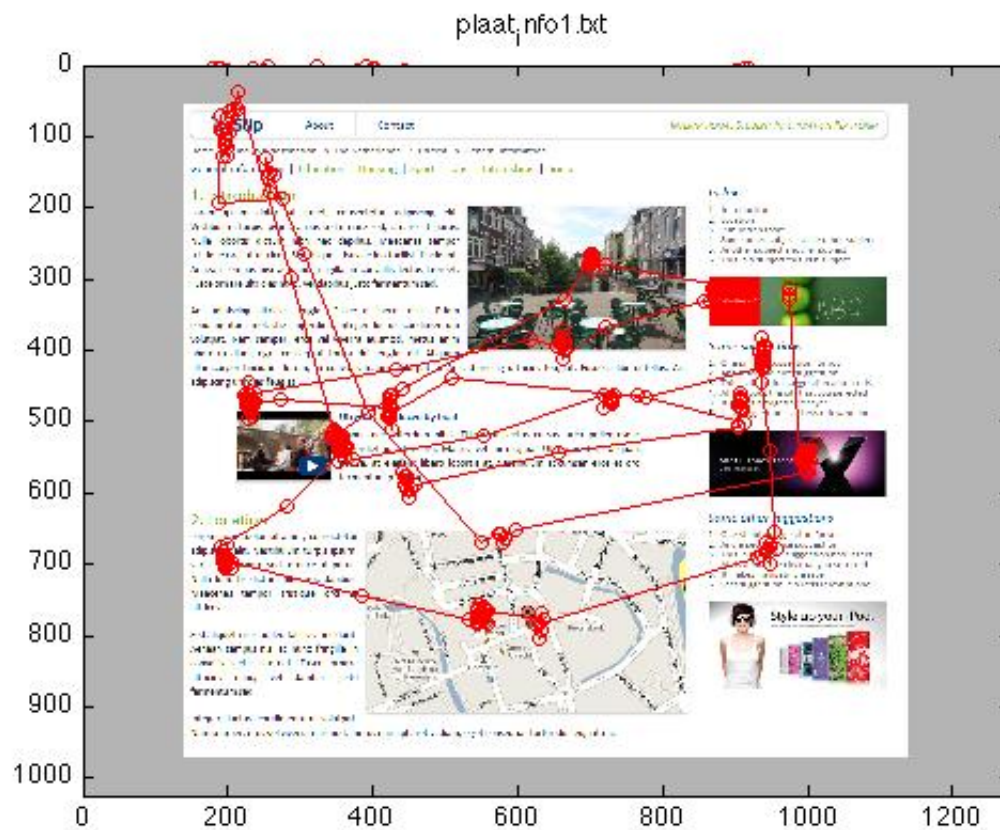






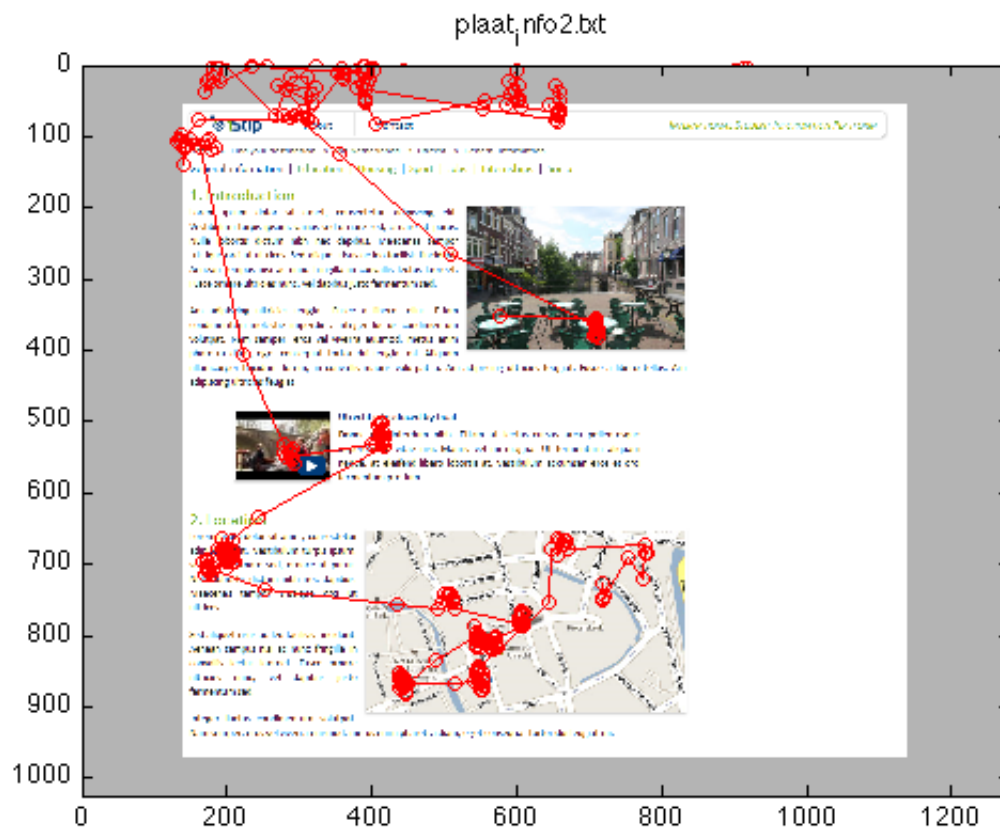
Figuur 17 - Expert 2, country



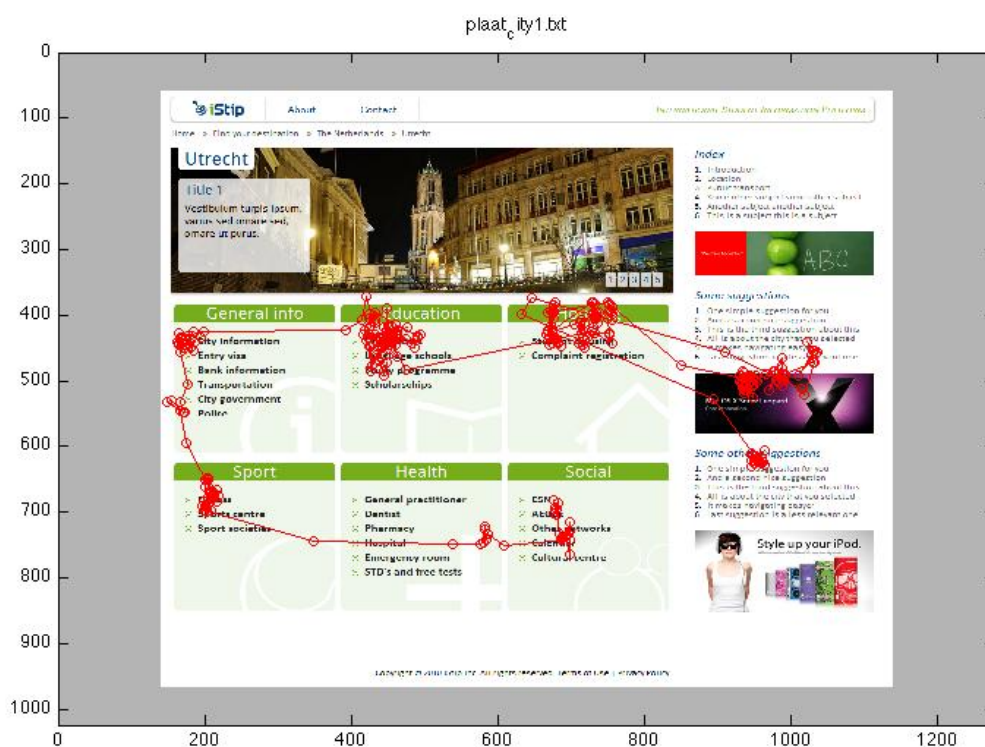


Figuur 18 - Expert, general information

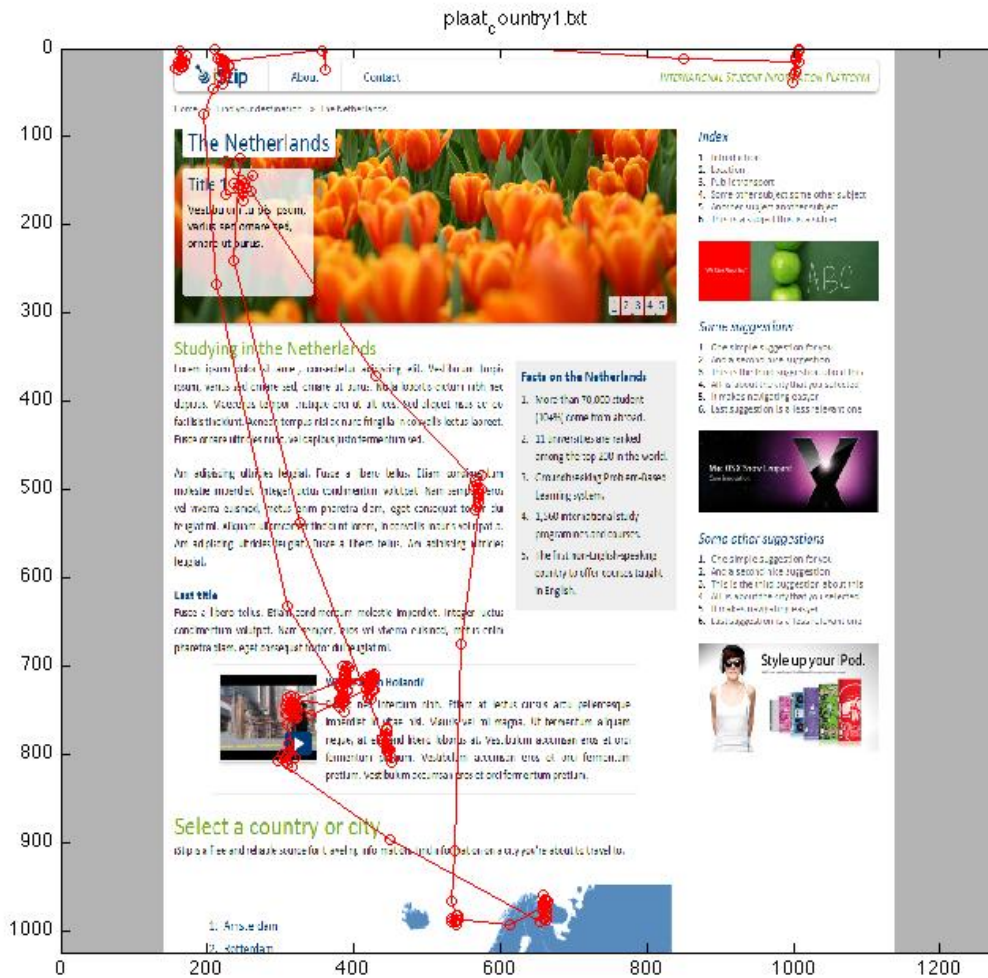




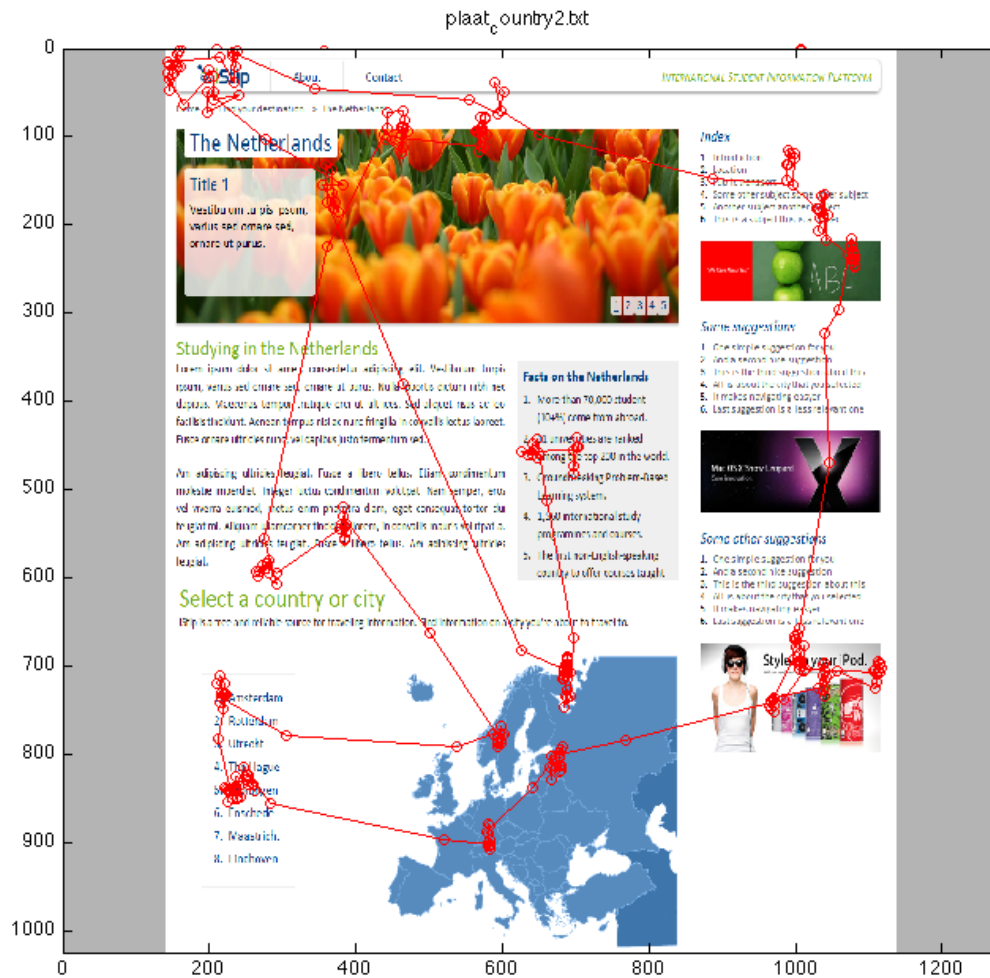
Figuur 19 - Expert 2, general information



Figuur 20 - Novice, city



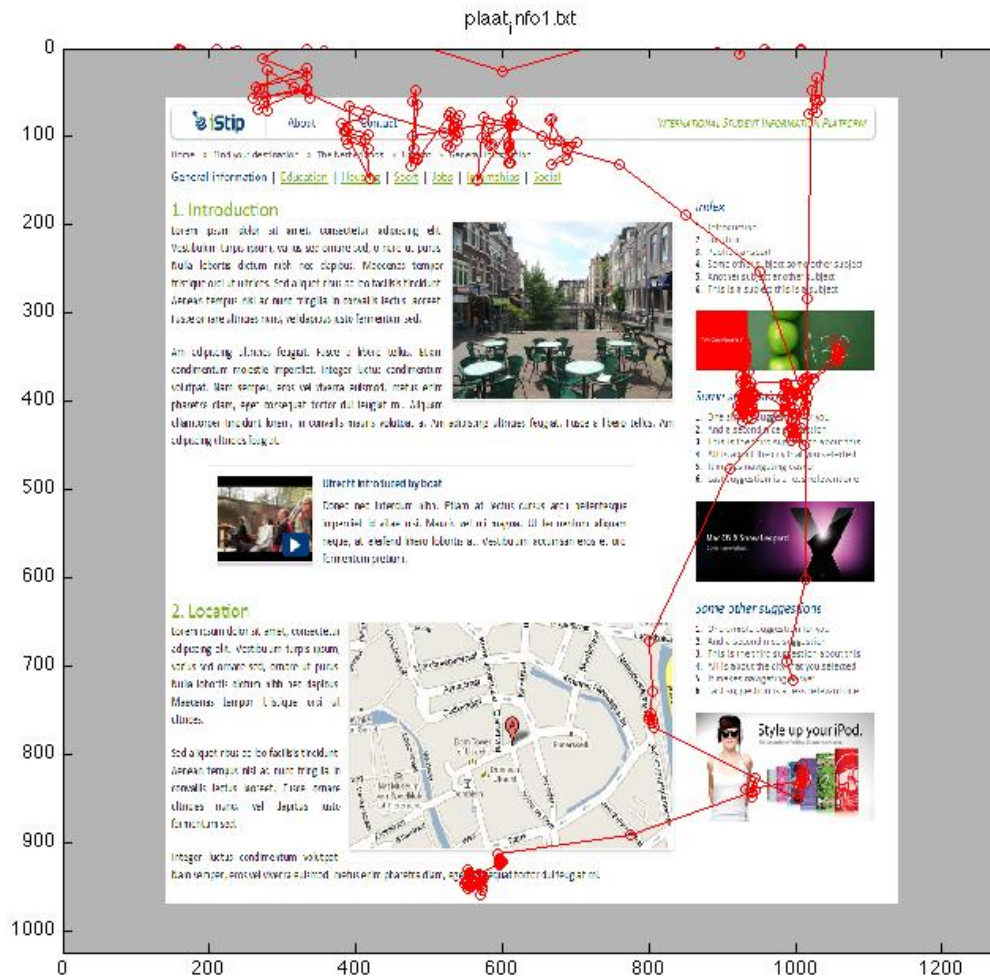
**Figuur 21 - Novice, country**



Figuur 22 - Novice 2, country

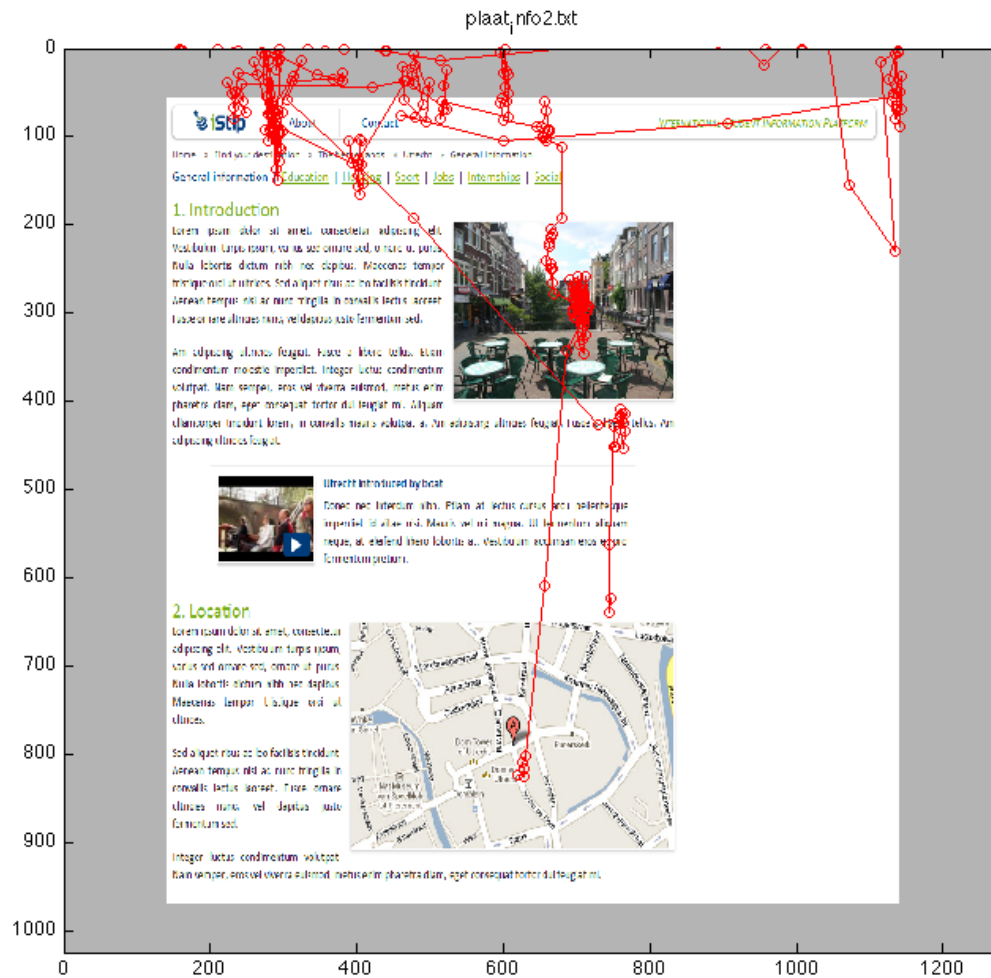






Figuur 23 - Novice, general information





### Figuur 24 - Novice 2, general information

