



Afstudeerverslag

Collective Reputation Base (CRB)

Auteur	: Ali Jawad
Datum	: maandag 14 januari 2013
Versie	: v 1.0
Afstudeerbedrijf	: Logica Nederland B.V.
Divisie	: Professional Skills
Practice	: Microsoft
Bedrijfsmentor	: Dhr. Wouter Verwaard
Bedrijfsbegeleider	: Dhr. Arie Leeuwesteijn / Dhr. Hans Kappert
Opdrachtgever	: Dhr. Frank Titulaer
Onderwijsinstelling	: De Haagse Hogeschool
Opleiding	: Informatica
Afstudeercoördinator	: Mevr. Rianne Béchet
Examinator	: Dhr. Paul Smit

Logica is now part of CGI, a leading business and technology services company with 72,000 professionals operating in more than 40 countries across the Americas, Europe and Asia.

We have the presence, expertise and complete IT services to meet clients' business needs anywhere, anytime. Our innovative thinking, deep industry expertise, and end-to-end consulting, systems integration and outsourcing services enable us to serve as clients' full-service provider in improving all facets of their business and technology operations.



More information is available at
www.cgi.com | www.logica.com



Logica is now part of CGI.

Versiebeheer

Versie	Omschrijving
v 0.1	Concept afstudeerverslag
v 0.2	Referaat, Voorwoord, Inleiding, Organisatie, Opdrachtsomschrijving, methode en technieken, Requirements, Onderzoek CRB
v 0.3	Verwijderen h9 afwijkingen ten opzichte tot afstudeerplan, verplaatsen van MoSCoW paragraf van methoden naar Requirements, Gebruikte onderzoeksmethoden, verzamelen P2P-websites adressen, Onderzoeken reputatiesystemen, Enquete P2P-website beheerders, Architectuurontwerp, Gesprekken met de begeleiders en de opdrachtgever, Het opstellen van de context viewpoint, het opstellen van de functional viewpoint, het opstellen van de information viewpoint, het opstellen van de development viewpoint, het opstellen van de deployment viewpoint
v 0.4	Toegevoegd: Toevoegen organogram Logica en student positie binnen Logica, gebruikte methode requirements, Figuur 12, enquete resultaten, evaluatie enquete, communicatiemogelijkheden met de P2P-websites, Evalueren onderzoek, bespreken bevindingen onderzoek met de opdrachtgever, Concurrency View, Operational View, bespreken ontwerp met opdrachtgever en begeleiders, gevolgd cursussen, scrum sprints. Bijgewerkt: relevantie, UML, Software Systems Architecture, totstandkoming van de requirements, Tabellen 4 en 5, verzamelen P2P-websites, enquete respondenten, architectuurontwerp, Functional View, Information View, Development View.
v 0.5	Toegevoegd: Proof of Concept, Sprint 1. Bijgewerkt: opmerkingen Hans Kappert.
v 0.6	Toegevoegd: Sprint 2, Sprint 3, succesvolle uitgevoerde requirements, publiceren van het PoC op Windows Azure omgeving, taalcorrectie t/m pagina 33.
v 0.7	Toegevoegd: evaluatie. Bijgewerkt: afkortingen en begrippenlijst
v 1.0	Taalcorrectie, definitieve versie.

Tabel 1: Versiebeheer

Distributielijst

Naam	Instantie	Rol / Functie
Dhr. Wouter Verwaard	Logica	Bedrijfsmentor
Dhr. Arie Leeuwesteijn	Logica	Bedrijfsbegeleider
Dhr. Hans Kappert	Logica	Vervangende Bedrijfsbegeleider
Dhr. Frank Titulaer	Logica	Opdrachtgever
Mevr. Rianne Béchet	De Haagse Hogeschool	Begeleider
Dhr. Paul Smit	De Haagse Hogeschool	Examinator

Tabel 2: Distributielijst

Referaat

In dit verslag wordt ingegaan op het onderzoek en ontwerp voor het bouwen van een Proof-of-Concept van Collective Reputation Base (CRB). Het CRB is een centrale verzamelpunt voor gebruikersreputaties op P2P-websites. Dit is gedaan in het kader van het afstuderen van Ali Jawad bij Logica, nu onderdeel van CGI te Rotterdam.

- Afstudeerverslag
- Collective Reputation Base (CRB)
- Verzameling van de gebruikersreputaties
- Samenvoegen van verschillende beoordelingsvormen
- P2P-websites
- Particuliere gebruikers
- ASP.NET MVC 4
- Windows Azure
- Agile Scrum
- UML (Unified Modeling Language)
- Logica, nu onderdeel van CGI
- De Haagse Hogeschool

Voorwoord

In het kader van mijn afstuderen voer ik het project Collective Reputation Base (CRB) uit. De CRB is een systeem voor het verzamelen, samenvoegen en beschikbaar stellen van de gebruikersreputaties op verschillende P2P-websites. Dit zijn websites waar de particuliere aanbieders en afnemers elkaar kunnen vinden voor het aanbieden van artikelen en/of diensten.

Hierbij wil ik de volgende personen bedanken voor hun steun en waardige feedback en suggesties voor het succesvol afronden van dit project: dhr. Wouter Verwaard (bedrijfsmentor), dhr. Frank Titulaer (opdrachtgever), dhr. Arie Leeuwesteijn (bedrijfsbegeleider) en dhr. Hans Kappert (bedrijfsbegeleider).

Tevens wil ik ook mijn beste vrienden Waseem en Waiel Alkhateeb bedanken voor taalkundige correctie van alle documenten.

Ten slotte wil ik mijn vrouw, dochter en familie bedanken voor hun geduld en de mooie momenten die ik met hun hebben mogen doorbrengen.

Spijkenisse, 10 januari 2013

Ali Jawad

Inhoudsopgave

1.	Inleiding	8
2.	Organisatie.....	9
2.1	Afstudeerbedrijf.....	9
2.2	Projectorganisatie	10
2.3	Begeleiding	10
3.	Opdrachtschrijving	11
3.1	Aanleiding.....	11
3.2	Probleembeschrijving	11
3.3	Doelstelling	11
3.4	Relevantie	12
4.	Methode en technieken	13
4.1	Onderzoeksmethode (Wat is onderzoek?)	13
4.2	Software Systems Architecture	13
4.3	Systeem ontwikkel methode (Agile Scrum)	14
4.4	UML	15
5.	Requirements	17
5.1	Gebruikte methode	17
5.2	Gebruik MoSCoW	17
5.3	Totstandkoming van de requirements.....	18
6.	Onderzoek CRB.....	23
6.1	Onderzoeksvragen	23
6.2	Gebruikte onderzoeksmethoden.....	24
6.3	Verzamelen P2P-website adressen	24
6.4	Onderzoeken reputatiesystemen.....	25
6.5	Enquête P2P-website beheerders	28
6.6	Communicatiemogelijkheden met de P2P-websites	31
6.7	Evalueren onderzoek.....	33
6.8	Bespreken bevindingen met de opdrachtgever.....	33
7.	Architectuurontwerp	34
7.1	Gesprekken met de begeleider en de opdrachtgever	34
7.2	Het opstellen van de Context viewpoint	35
7.3	Het opstellen van de Functional viewpoint.....	37
7.4	Het opstellen van de Information viewpoint.....	40
7.5	Het opstellen van de Concurrency viewpoint.....	44
7.6	Het opstellen van de Development viewpoint.....	45
7.7	Het opstellen van de Deployment viewpoint	48
7.8	Het opstellen van de Operational viewpoint.....	49
7.9	Bespreken ontwerp met de opdrachtgever en de begeleiders	49
8.	Proof of Concept van CRB.....	50
8.1	Gevolgde cursussen	50
8.2	Scrum Sprints	50
8.3	Gebruik LINQ	59

8.4	Succesvol uitgevoerde requirements	59
8.5	Uitgevoerde non-functional requirements.....	60
8.6	Publiceren van het PoC op Windows Azure omgeving.....	60
9.	Evaluatie.....	62
9.1	Proces evaluatie	62
9.2	Producten evaluatie	63
9.3	Verantwoording van de beroepstaken.....	65
	Literatuurlijst.....	67
	Begrippenlijst.....	68
	Afkortingen.....	69
	Figurenlijst	70
	Tabellenlijst.....	71

1. Inleiding

Dit verslag is opgesteld om de uitgevoerde werkzaamheden bij het project Collective Reputation Base (CRB) in beeld te brengen en de keuzes te verantwoorden. Het verslag is een onderdeel van mijn afstudeertraject in de periode van 6 februari 2012 tot 14 januari 2013. Hierin wordt beschreven welk proces ik heb gevolgd tijdens het uitvoeren van het project en wat de belangrijkste producten van dit proces zijn. Tevens wordt beschreven welke keuzes ik heb gemaakt tijdens het project en deze keuzes worden verantwoord. Dit document is geschreven voor de examinatoren van de Haagse Hogeschool, de begeleiders/opdrachtgever van Logica (nu onderdeel van CGI) en de gecommitteerden tijdens mijn afstudeerzitting.

Tevens vindt er in dit document een vergelijking plaats tussen enerzijds de opgeleverde documenten en anderzijds het opgeleverde afstudeerplan.

Het afstudeerverslag is in de volgende hoofdstukken ingedeeld:

- **Organisatie:** in dit hoofdstuk worden het afstudeerbedrijf, de opdrachtgevers/begeleiders en de begeleiding van de student beschreven.
- **Opdrachtsomschrijving:** hierin wordt de aanleiding tot de opdracht, de probleembeschrijving, de doelstelling en de scope van de opdracht beschreven.
- **Methode en technieken:** in dit hoofdstuk worden de gebruikte methoden en technieken verantwoord en beschreven.
- **Requirements:** dit hoofdstuk gaat over de requirements. Hoe deze requirements tot stand zijn gekomen, de afstemming met de opdrachtgever en het gebruik van MoSCoW methode worden beschreven.
- **Onderzoek CRB:** hierin wordt de beantwoording van de onderzoeksvragen beschreven waaronder de dataverzameling, analyseren van de data en de evaluatie en resultaten van het onderzoek.
- **Architectuurontwerp:** dit hoofdstuk is geschreven in het kader van het architectuurontwerp. Als onderdelen van het ontwerp worden verschillende viewpoints beschreven: Functional, Information, Concurrency, Development, Deployment en Operational.
- **Proof of Concept van CRB:** het tot stand komen van het Proof of Concept wordt hierin beschreven. Daarnaast worden de verschillende gebruikte technieken benoemd en wordt er een verantwoording afgelegd waarom deze keuze zijn gemaakt.
- **Evaluatie:** Hierin staat een evaluatie over de procesgang en de producten. Verder worden de gekozen competenties verantwoord.

2. Organisatie

In dit hoofdstuk worden het afstudeerbedrijf, de projectorganisatie en de begeleiding beschreven. Het bedrijf en mijn positie binnen het bedrijf worden hier geïntroduceerd. Verder wordt de projectomgeving beschreven en hoe de begeleiding heeft plaatsgevonden.

2.1 Afstudeerbedrijf

Logica is nu onderdeel van CGI. CGI is nu een nieuwe wereldwijde technologie en zakelijke dienstverlener. Met 72.000 professionals in meer dan 40 landen bieden nu CGI en Logica meer expertise en diensten voor hun klanten in Noord- en Zuid-Amerika, Europa en de Asia Pacific regio.



Logica is now part of CGI.

Als echte lokale partner neemt CGI haar verantwoordelijkheid, tegelijkertijd biedt CGI wereldwijd capaciteit, talent en diensten die nodig zijn om aan de veranderende behoeften te voldoen.

De cultuur van CGI is gebaseerd op een ondernemingsgeest die haar experts flexibiliteit biedt om innovatieve en toepasbare oplossingen te bedenken die het beste passen bij behoeften van klanten. (1)

Zakelijke aanpak van CGI kent de volgende aspecten: kennis van de verschillende industrieën, unieke wereldwijde leveringsmogelijkheden en gekwalificeerde processen.

CGI heeft een uitgebreid portfolio van diensten zoals: IT-consulting, systeemintegratie, systeemarchitectuur, applicatieontwikkeling, applicatiebeheer, applicatieontwerp, uitgebreide infrastructuurdiensten, business proces service en het beheren daarvan en bedrijfskritische oplossingen.

Op dit moment heeft CGI kennis in de volgende gebieden: overheid, financiële diensten, productie, detailhandel en distributie, telecommunicatie, media, olie en gas en gezondheid. (2)

Binnen Logica maak ik deel uit van

Microsoft Practice. Deze practice is een onderdeel van de Professional Skills onderdeel van Logica, nu onderdeel van CGI.



Figuur 1: organogram Logica, nu onderdeel van CGI

2.2 Projectorganisatie

De projectorganisatie ziet er als volgt uit:

Naam	Instantie	Rol / Functie
Ali Jawad	De Haagse Hogeschool / Logica, nu onderdeel van CGI	Student
Dhr. Wouter Verwaard	Logica, nu onderdeel van CGI	Bedrijfsmentor
Dhr. Arie Leeuwesteijn	Logica, nu onderdeel van CGI	Bedrijfsbegeleider
Dhr. Hans Kappert	Logica, nu onderdeel van CGI	Vervangende bedrijfsbegeleider
Dhr. Frank Titulaer	Logica, nu onderdeel van CGI	Opdrachtgever
Mevr. Rianne Béchet	De Haagse Hogeschool	Begeleider
Dhr. Paul Smit	De Haagse Hogeschool	Examinator

Tabel 3: Projectorganisatie

2.3 Begeleiding

Tijdens het afstudeertraject wordt de begeleiding door Logica verzorgd door twee begeleiders. De procesbegeleiding en technische begeleiding wordt door de heer Wouter Verwaard verzorgd. Door de heren Arie Leeuwesteijn en Hans Kappert wordt het technische onderdeel ondersteund. Verder worden de resultaten besproken met de betrokkene en de keuzes worden gezamenlijk gedaan met begeleiders en opdrachtgever, de heer Frank Titulaer.

Om het proces optimaal te laten lopen wordt continu contact opgenomen met de begeleiders en de opdrachtgever. Elke twee weken wordt er een sessie gehouden met de heer Wouter Verwaard voor het bespreken van de voortgang.

Aan het einde van het afstudeertraject zijn er wijzigingen geweest door het vertrekken van de heer Leeuwesteijn. De heer heeft Kappert de begeleiding van hem overgenomen.

3. Opdrachtomschrijving

Dit hoofdstuk gaat over de opdracht in het algemeen. De aanleiding tot de opdracht wordt uitgelegd en de probleemstelling wordt vastgesteld. Daarna worden de doelstelling en de scope van de opdracht uitgewerkt.

3.1 Aanleiding

Het aantal P2P-websites neemt sterk toe. Door de voortschrijdende (mobiele) techniek wordt het voor mensen steeds gemakkelijker om allerlei soorten particulier vraag en aanbod te matchen. Technologie alleen is echter niet voldoende om ook daadwerkelijk tot transacties te komen. De menselijke factor (vertrouwen/reputatie) is net zo belangrijk. Door reputaties 'draagbaar' te maken tussen verschillende P2P-websites wordt het vertrouwen versneld opgebouwd en kunnen P2P-websites sneller groeien. Verder zijn er personen die op verschillende P2P-websites actief zijn; om het vertrouwen van zowel de aanbieders als de afnemers te verhogen is er behoefte aan het samenvoegen van de reputaties op verschillende P2P-websites.

3.2 Probleembeschrijving

Verschiedende P2P-websites hebben hun eigen reputatiesysteem, waarmee de aanbieders en soms de afnemers op deze websites op verschillende aspecten worden beoordeeld. Deze systemen worden steeds belangrijker voor deze P2P-websites door het grote aantal gebruikers dat deze websites bezoekt. Om de betrouwbaarheid van de aanbieders/afnemers te kunnen bepalen vragen deze bedrijven hun gebruikers om de verkopers te beoordelen. Een gebruiker kan op meerdere P2P-websites actief zijn, aangezien deze bedrijven niet onderling verbonden zijn, kunnen de afnemers geen beoordelingen van aanbieders op andere websites zien. Deze samengevoegde beoordeling is belangrijk om de betrouwbaarheid van de aanbieders te kunnen bepalen. Om dit laatste te kunnen bevorderen dient een systeem worden gebouwd voor het samenvoegen van de reputaties op verschillende P2P-websites tot één gezamenlijke score en deze beschikbaar te stellen voor deze websites.

3.3 Doelstelling

De centrale doelstelling van dit project is het ontwerpen en bouwen van Collective Reputation Base (CRB). Daaronder vallen de volgende doelstellingen:

- Onderzoeken van verschillende reputatiesystemen op de P2P-websites om het volgende vast te stellen:
 - achterhalen hoe de verschillende reputatiesystemen op de P2P-websites werken;
 - vaststellen hoe we de reputaties van de gebruikers van deze websites kunnen ophalen;
 - vaststellen hoe verschillende vormen van de beoordelingen op verschillende P2P-websites kunnen worden samengevoegd tot een totale beoordeling.
- Opstellen van architectuurontwerp aan de hand van vastgestelde requirements en het verrichtte onderzoek;
- Bouwen van een Proof-of-Concept van het systeem voor het tonen van de werking ervan.

3.4 Relevantie

Het realiseren van dit product kan voor verschillende partijen waardevol zijn. Voor de gebruikers op de P2P-websites is het een handig middel waarmee zij zich binnen een systeem kunnen presenteren waarin zij niet actief zijn. De afnemers krijgen een beter en completer inzicht in de betrouwbaarheid van de aanbieders. De waarde voor de P2P-websites is het verhogen van de betrouwbaarheid van de aanbieders op hun sites en voor de nieuwe websites kunnen de aanbieders sneller actief worden, omdat zij hun bestaande reputaties op andere websites mee kunnen nemen. Het is ook voor Logica, nu onderdeel van CGI, interessant om de technische en commerciële mogelijkheden van een dergelijk systeem te onderzoeken om zich in het hart van alle online customer-to-customer activiteiten te kunnen plaatsen.

Op dit moment zijn er soortgelijke systemen, maar deze systemen zijn speciaal voor het beoordelen van de bedrijven door middel van het direct beoordelen van de bedrijven en niet het ophalen, verzamelen en samenvoegen van de reputaties over deze bedrijven van verschillende websites.

De CRB onderscheidt zich van deze systemen door het verzamelen en samenvoegen van de beoordelingen van particulieren die op verschillende P2P-websites actief zijn. Daarnaast wordt bij de samenvoeging van de reputaties een totale reputatie berekend van de gebruikers en aangeboden aan de P2P-websites. De CRB is daarmee geen reputatiesysteem, maar een verzamelpunt voor de reputaties op internet.



4. Methode en technieken

In dit hoofdstuk wordt ingegaan op de gebruikte methoden en technieken voor het uitwerken van de afstudeeropdracht. Het project heeft twee hoofdonderdelen, namelijk: onderzoek en softwareontwikkeling.

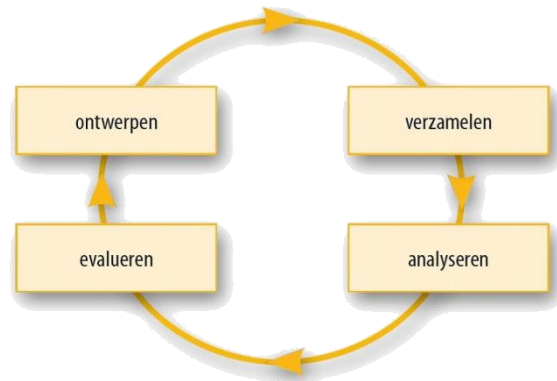
4.1 Onderzoeksmethode (Wat is onderzoek?)

Als onderzoeksmethode heb ik de methode van Nel Verhoeven gebruikt. Deze methode is beschreven in het boek "Wat is onderzoek?" (3). In dit boek zijn verschillende methoden en technieken te vinden voor het uitvoeren van de praktijkonderzoeken. Verder is een duidelijke fasering aangegeven voor deze methode. (Zie Figuur 2)

Tijdens mijn studie heb ik deze methode leren kennen en ik heb er goede ervaring mee gedaan. Dit waren de redenen geweest voor het kiezen van deze methode, naast de duidelijke fasering die er binnen deze methode is.

Naast het gebruik van het boek heb ik de website 'onderzoekinstappen.nl' gebruikt voor het toepassen van de "Wat is onderzoek?"-methode. Onderzoek in stappen is een onderzoekstool waarmee stapsgewijs een compleet onderzoek kan worden opgezet en uitgevoerd volgens de fases uit het boek *Wat is onderzoek?* van Nel Verhoeven. (4)

Ik heb me geabonneerd op deze website om daarvan gebruik te kunnen maken voor het uitvoeren van het onderzoek.

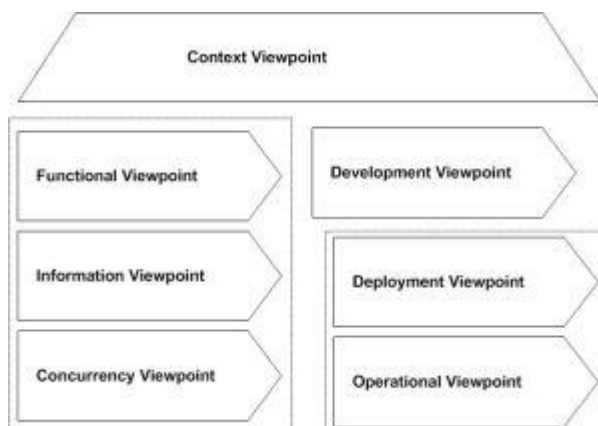


Figuur 2: fases "Wat is onderzoek?"-methode

4.2 Software Systems Architecture

Tijdens mijn studie bij blok (I-6) heb ik het boek "Software Systems Architecture" (5) gebruikt voor het schrijven van een software architectuurontwerp. Deze methode voor het schrijven van software architectuur wilde ik praktiseren tijdens mijn afstuderen zodat ik er meer ervaring mee op kan doen. Vooral het gebruiken van verschillende viewpoints die benoemd zijn in dit boek vond ik interessant, omdat vanuit verschillende kanten wordt gekeken naar een software architectuur.

Deze methode van software architectuur kent zes viewpoints, deze worden hieronder in het kort uitgelegd:



Figuur 3: viewpoints Software Systems Architecture

- **Functional viewpoint:** beschrijft de functionele elementen, hun verantwoordelijkheden, interfaces en primaire interacties van het systeem.
- **Information viewpoint:** beschrijft de manier waarop het systeem informatie opslaat, manipuleert, beheert en verspreidt.
- **Concurrency viewpoint:** beschrijft de samenwerkingen tussen verschillende onderdelen van het systeem en hoe deze samenwerking wordt gecoördineerd.

- **Development Viewpoint:** hierin wordt het software development proces beschreven en de invloed van de opdrachtgevers in het bouwen, testen en verbeteren van het systeem wordt aangegeven.
- **Deployment Viewpoint:** dit viewpoint beschrijft de omgeving waarin het systeem wordt geïnstalleerd/gehost.
- **Operational Viewpoint:** hierin wordt beschreven hoe het systeem wordt bediend, ondersteund en beheerd.

In de 2^e druk van het boek "Software Systems Architecture" (6) is een nieuwe viewpoint opgenomen. Deze viewpoint wordt hieronder uitgelgd:

- **Context viewpoint:** beschrijft de relaties, afhankelijkheden en de interactie tussen het systeem en zijn omgeving (de mensen, systemen en externe entiteiten waarmee samen wordt gewerkt).

Voor het schrijven van een architectuurontwerp voor CRB is gebruik gemaakt van de bovenstaande viewpoints. In hoofdstuk 7 wordt uitgebreid beschreven hoe deze viewpoints tot stand zijn gekomen binnen het architectuurontwerp document.

4.3 Systeem ontwikkel methode (Agile Scrum)

Scrum is een vorm van de Agile aanpak, welke als framework kan worden ingezet om software in teamverband te ontwikkelen. (7)

Voor het bouwen van het Proof-of-Concept van CRB is gebruik gemaakt van de Scrum methode.

4.3.1 Waarom Scrum?

Het kiezen van Scrum voor het uitvoeren van PoC onderdeel van dit project heeft de volgende redenen:

- Het product wordt op kortere periodes met de opdrachtgever afgestemd en dit verkleint het risico dat het opgeleverd product niet overeenkomt met de behoefte van opdrachtgever;
- Tijdens mijn werk heb ik te maken gehad met Scrum als software ontwikkelingsmethode en ik heb ervaring met deze methode;
- Ik heb ook een cursus gevolgd over Scrum bij Logica om de basis van Scrum te begrijpen. De cursusnaam was "Agile Scrum Awareness";
- De oplevering van een stuk werkende software bij scrum gebeurt in korte periodes (2-4 weken) en dit laatste was een wens van de opdrachtgever;
- Scrum biedt duidelijke inzicht in de voortgang van het project/software development.

4.3.2 Toepassing Scrum

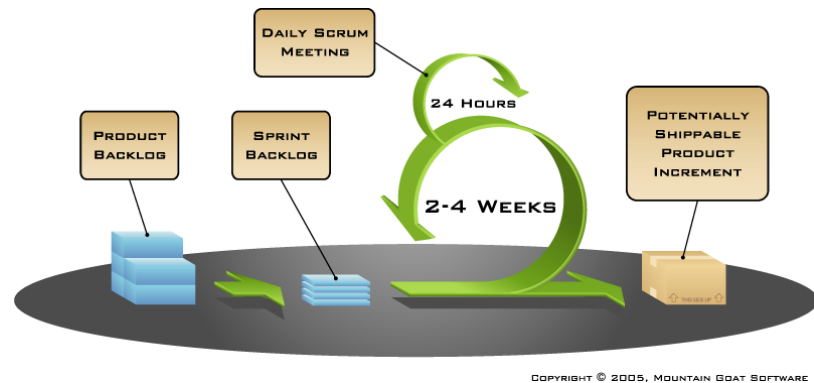
Bij het uitvoeren van deze methode zijn de volgende rollen/elementen gebruikt:

- **Product owner (PO):** de PO is de benoemde opdrachtgever (Dhr. Frank Titulaer), waarmee verschillende afspraken worden gemaakt voor het bespreken van de sprints, PBL en SPL. De bespreking vindt plaats samen met de Scrum master (SM) voor het vaststellen van de Definition of Done (DoD).
- **Scrum master (SM):** de SM is verantwoordelijk voor het vaststellen van DoD met de PO en het bewaken van het succesvol uitvoeren van de sprints. In het geval van dit project zal de student (Ali Jawad) de SM zijn, omdat het projectteam uit één persoon bestaat.
- **Sprint:** elk project wordt bij Scrum onderverdeeld in sprints. Bij elk sprint onderhandelen de PO en SM over welke requirements worden uitgevoerd en wat

de DoD (resultaat) van de sprint zal zijn. Dit project bestaat uit 3 sprints. De verdeling van de sprints wordt vermeld bij de activiteitentabel.

- **Product Backlog (PBL):** dit is de beschrijving van de op te leveren producten die met de PO afgesproken zijn.
- **Sprint Backlog (SBL):** hierin worden de requirements vastgesteld die tijdens een bepaalde sprint uitgevoerd worden. De SM en PO bespreken deze requirements en de SBL van een sprint wordt vastgesteld.

- **Definition of Done (DoD):** bij DoD wordt beschreven hoe een product succesvol kan worden opgeleverd. Het is erg belangrijk om een goede DoD te schrijven en goed afstemmen met de PO voor het voorkomen van conflicten bij het leveren van verschillende producten.



Figuur 4: Scrum overview diagram

- **Daily Scrum:** aan het begin van elke dag wordt een teammeeting georganiseerd (max 15 minuten) om vast te stellen wat elk teamlid heeft gedaan, wat hij gaat doen en welke problemen hij tegenkomt. Het is van belang voor een algemeen beeld wat de status is van het uitvoeren van de requirements. Tijdens dit project wordt dit gedaan door de student (Ali Jawad) alleen, omdat hij de enige teamlid is bij dit project. Hij zal het werk dat verricht is dagelijks evalueren. Deze evaluatie wordt in een logboek opgenomen en aan de begeleider voorgelegd.

Tijdens het schrijven van het plan van aanpak zijn er gesprekken gevoerd met de opdrachtgever over de ideeën van de Proof-of-Concept. Tijdens de gesprekken is afgesproken dat er 3 sprints zullen worden uitgevoerd voor het bouwen van het PoC. Deze sprints zijn:

- Sprint 1 (Bouwen CRB PoC – Website CRB + basis functionaliteiten);
- Sprint 2 (Bouwen CRB PoC – verzamelen gegevens uit verschillende websites);
- Sprint 3 (Bouwen CRB PoC – afhandelen verzamelde gegevens en aanbieden van de totale reputatie).

4.4 UML

Om verschillende onderdelen binnen de viewpoints van het architectuur duidelijker te maken en goed te kunnen verbeelden heb ik de Unified Modeling Language (UML) gebruikt. UML is een taal die bestaat uit begrippen en diagrammen. Met behulp van de begrippen van deze taal worden de UML-diagrammen ontworpen. (9)

Tijdens mijn studie en werk heb ik de UML herhaaldelijk gebruikt, omdat de structuur is en het eenvoudig te gebruiken is voor verschillende projecten. Ik heb ervoor gekozen om het te gebruiken bij mijn afstudeertraject. UML speelt binnen dit project een belangrijke rol, vooral bij het opstellen van het architectuurontwerp. Hiermee doe ik een stap verder in mijn ervaring met UML.

UML kent onder andere de volgende diagrammen: use-case-diagram, klassediagram, objectdiagram, sequencediagram, communicatiediagram, toestandsdiagram, activiteitsdiagram, componentdiagram en deploymentdiagram.

De UML is toegepast bij het opstellen van requirements, door middel van de overall use-case diagram. Verder is deze moduleringsmethode gebruikt bij het opstellen van de volgende diagrammen en modellen op het Architectuurontwerp document:

- Sequentiediagrammen;
- Contextdiagram;
- Componentendiagrammen;
- Functioneel structuur model
- System datastructuur;
- EER-model;
- RRM-model;
- Concurrency model;
- Modulestructuur;
- Klassediagram;
- Deployment model

5. Requirements

Requirements spelen een belangrijke rol binnen dit project. Met behulp van de requirements worden verschillende onderdelen ontworpen en gebouwd. In dit hoofdstuk wordt uitgelegd hoe deze requirements tot stand zijn gekomen en welke stappen zijn genomen om tot een uiteindelijke versie van de requirements te komen. Verder zijn de requirements belangrijk voor het vaststellen van de Sprints tijdens het uitvoeren van Scrum werkwijze bij het bouwen van de PoC.

Als afwijking van het afstudeerplan wil ik hier melden dat ik in plaats van beroepstaak 1.3 (Selecteren van standaardsoftware) de beroepstaak 1.4 (Uitvoeren analyse door definitie van requirements) heb uitgevoerd. In dit hoofdstuk wordt vanuit deze beroepstaak gewerkt.

5.1 Gebruikte methode

Voor het opstellen van de requirements heb ik de SMART-methode⁽⁹⁾ gebruikt. Deze methode houdt in dat de opgestelde requirements **Specifiek**, **Meetbaar**, **Actueel**, **Realistisch** en **Tijdgebonden**¹ dienen te zijn. Deze methode heb ik toegepast op alle requirements en met behulp van de begeleidersfeedback heb ik de SMART requirements opgesteld.

5.2 Gebruik MoSCoW

Voor het prioriteren van requirements is de MoSCoW verdeling gebruikt. Ik heb daarvoor gekeken naar de beschikbare tijd voor het bouwen van de PoC, beschikbare kennis, beschikbare Hardware en Software, wensen van de opdrachtgever en de uit te voeren afstudeer beroepstaken. Aan de hand daarvan zijn de prioriteiten vastgesteld. Deze zijn ook besproken met de opdrachtgever en de begeleider en ook door hen ge-reviewd. Bij verschillende versies van het requirementsdocument zijn er een aantal van de prioriteiten bijgewerkt door middel van de reviews van de begeleiders en opdrachtgever.

De prioriteiten binnen MoSCoW kunnen we in vier categorieën verdelen:

- **M (Must have):** dit zijn de noodzakelijke requirements van de applicatie. Zonder deze onderdelen kan de applicatie niet geleverd worden en functioneert deze niet als gewenst.
- **S (Should have):** deze requirements zijn zeer gewenst. Voor een optimale werking van de applicatie dient deze functionaliteit aanwezig te zijn.
- **C (Could have):** deze requirements worden in principe niet meegenomen, maar als er tijd over blijft voor het einde van het project kunnen deze alsnog geïmplementeerd worden.
- **W (Would have):** deze requirements komen niet aan bod en worden voor een vervolgoopdracht voorgesteld. (8)

1 **Specifiek:** het gewenste gedrag zo concreet mogelijk beschrijven. **Meetbaar:** men geeft aan waaraan kan worden afgemeten of het leerdoel behaald is. **Actueel:** het leerdoel sluit aan bij de huidige situatie. **Realistisch:** het resultaat is haalbaar. **Tijdgebonden:** men geeft aan binnen welke periode het leerdoel behaald moet worden.

5.3 Totstandkoming van de requirements

Bij het opstellen van de requirements heb ik eerst een use-case-diagram gemaakt met behulp van de gegevens in het opgeleverde document door de opdrachtgever² en de gesprekken met hem over de functionaliteit van het systeem (Figuur 5). Daarnaast heb ik een lijstje opgesteld met enkele requirements die door mij zijn voorgesteld.

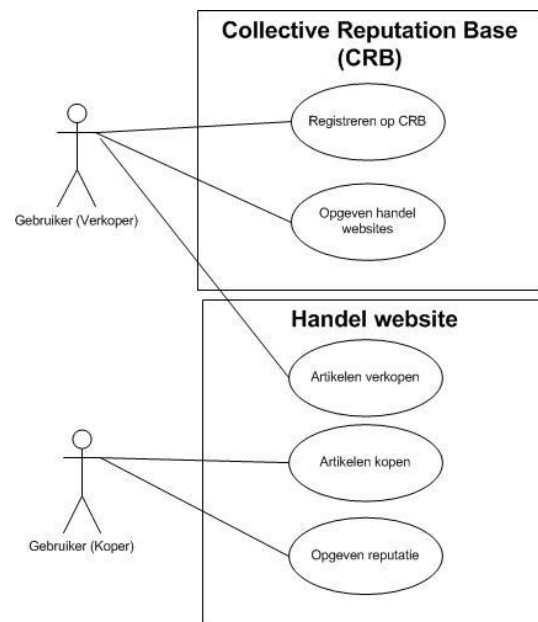
Met behulp van deze gegevens heb ik de eerste versie van het requirementsdocument opgesteld en naar de begeleiders en opdrachtgever gestuurd.

Hierover heb ik feedback gekregen. Het eerste wat moest gebeuren was dat er een uitgebreide use-case diagram moest worden gemaakt. Omdat het systeem zich niet beperkt tot de use-cases die genoemd zijn in het figuur hiernaast. Verder waren de requirements te algemeen en is het belangrijk dat ze zo SMART mogelijk worden opgesteld zodat er tijdens het ontwerp- en bouwphase geen problemen ontstaan en geen grote wijzigingen in de requirements voorkomen.

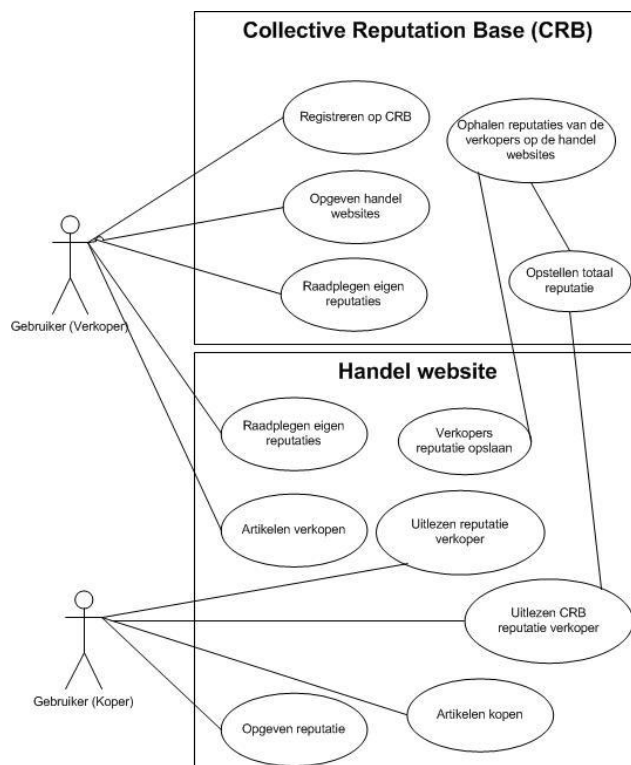
Aan de hand van de feedback en opmerkingen van de begeleiders en de opdrachtgever heb ik een uitgebreidere use-case diagram opgesteld en de requirements verder uitgewerkt.

In Figuur 6 is de nieuwe versie te zien van de use-case diagram. Hierin zijn er meer functionaliteiten toegevoegd. Deze tweede versie is ook gestuurd naar de begeleiders en de opdrachtgever. Hierover waren ook enkele opmerkingen. De belangrijkste opmerkingen waren:

- Opdrachtgever heeft het idee dat het systeem niet wordt beperkt tot verkopers (aanbieders). De beoordelingen van de afnemers moeten ook kunnen worden verzameld binnen de CRB. Het verzamelen van de afnemersreputaties geeft ook meer zekerheid voor de aanbieders en dit is een aspect waar niet veel aandacht aan is besteed in de online wereld.
- Het opnemen van "Verwijderen CRB account"-use-case.



Figuur 5: versie 1 use-case-diagram

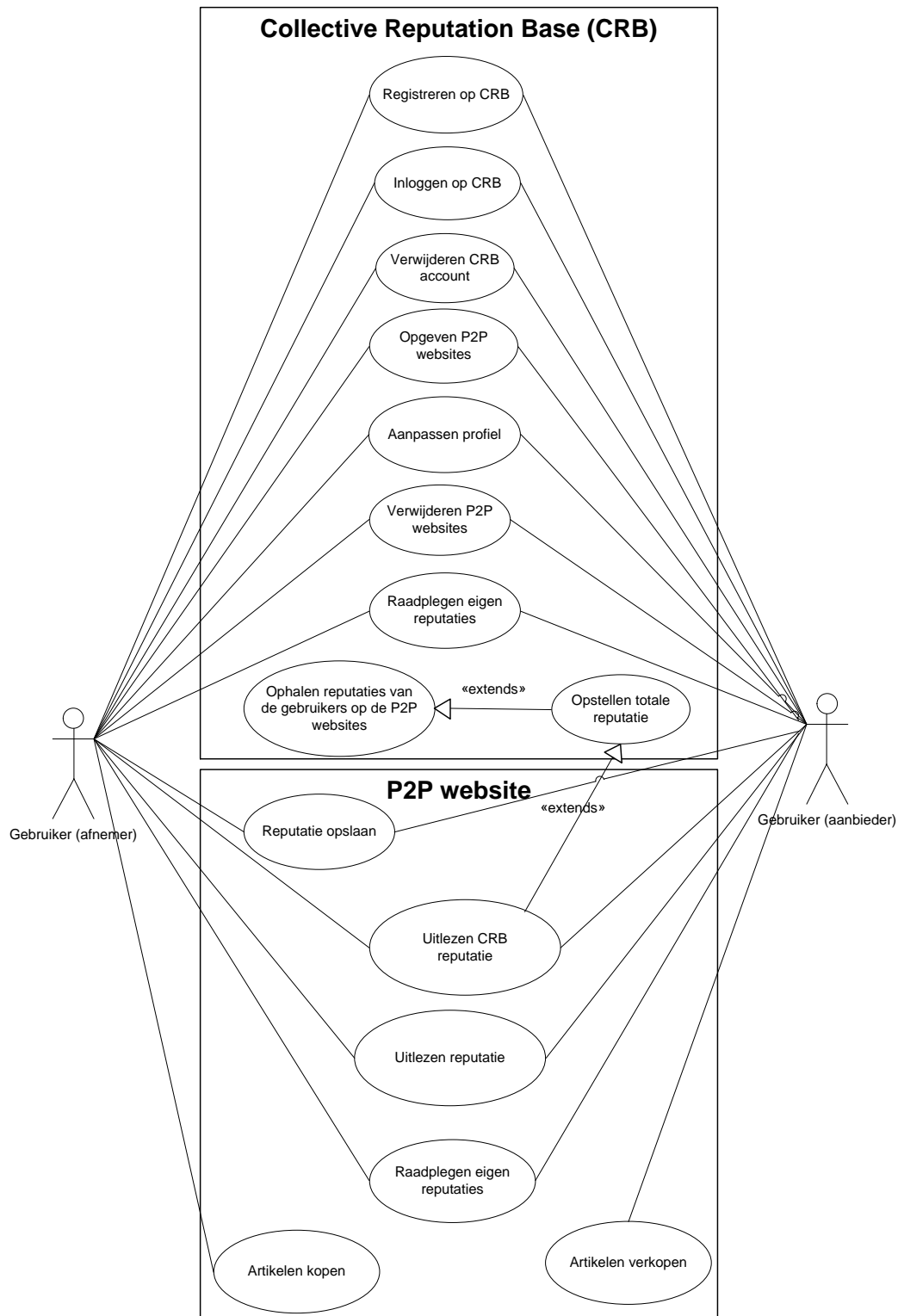


Figuur 6: versie 2 use-case-diagram

² WBSO – Project Mandate Feasibility – Collective Reputation Base, geschreven door Frank Titulaer (04-10-2011).

- Het opnemen van “Aanpassen profiel”-use-case.
- Verder was er door de opdrachtgever en de begeleiders gevraagd om enige basis functionaliteit in het systeem op te nemen zoals: inloggen op de CRB, verwijderen P2P-website.

Na het verwerken van deze opmerkingen en bedenken van nieuwe use-cases, is het use-case diagram in Figuur 7 tot stand gekomen. Het laatste diagram is doorgenomen met de begeleiders en de opdrachtgevers en zij waren hier allen akkoord mee.



Figuur 7: laatste versie use-case-diagram

Verder zijn in het requirementsdocument de use-case beschrijvingen opgenomen om de werking van elke use-case te verduidelijken. Hieronder is een voorbeeld te zien van deze beschrijvingen (Tabel 4).

Usecase nummer:	1
Usecase naam:	Registreren op CRB
Actor:	Gebruiker (Afnemer en Aanbieder)
Aannamen:	Hoofdpagina gestart
Usecase beschrijving:	Gebruiker registreert zich op het systeem
Procedure:	<ol style="list-style-type: none"> 1. Gebruiker opent de registratiepagina; 2. Gebruiker kiest het soort registratie, naast het classic registratie kan gebruiker zich registreren met zijn (Facebook, Twitter of Google) account; <ol style="list-style-type: none"> a. Registreren met klassieke authenticatie: <ol style="list-style-type: none"> i. Gebruiker voert zijn naam, emailadres en twee keer zijn gewenste wachtwoord in; ii. Gebruiker klikt op de "register" knop; iii. Gebruiker krijgt ter verificatie een email met een activatielink om zijn account te activeren; iv. Gebruiker klikt op de activatielink en activeert zijn account. b. Registreren met een bestaande (Facebook, Twitter of Google) account: <ol style="list-style-type: none"> i. Gebruiker kiest zijn voorkeur; ii. Gebruiker logt in met de inloggegevens; iii. Gebruiker geeft toestemming om gebruik te maken van zijn account.
Uitzondering	Als de ingevoerde wachtwoorden niet met elkaar overeenkomen wordt de gebruiker weer naar stap 1 gestuurd.
Resultaat:	Gebruiker geregistreerd.

Tabel 4: voorbeeld use-case beschrijvingen

Bij het opstellen van de requirements zijn de volgende bronnen geraadpleegd:

- Suggesties van de opdrachtgever;
- Suggesties van de begeleiders;
- Gesprekken met de opdrachtgever;
- Beschikbare systeembeschrijving³.

Bij elke requirement is aangegeven welk bron deze requirement heeft en welk prioriteit deze requirement krijgt.

Bij het prioriteren van de requirements heb ik een voorstel gedaan van de verdeling aan de hand van de beschikbare informatie over het project. Deze verdeling heb ik met de opdrachtgever en de begeleiders besproken, zodat er een uiteindelijke versie is ontwikkeld van de prioriteiten van de requirements. In het begin heb ik een voorstel gedaan voor de prioriteiten van de requirements en heb ik veel requirements op "Must

³ WBSO – Project Mandate Feasibility – Collective Reputation Base, geschreven door Frank Titulaer (04-10-2011).

have” gezet, maar na het bespreken van dit voorstel met de opdrachtgever en de begeleiders heb ik een andere verdeling toegepast waar minder “Must have” prioriteiten naar voren zijn gekomen. Deze nieuwe verdeling is verder besproken met de betrokkenen en zij vonden de nieuwe verdeling meer realistisch in verband met beschikbare tijd, kennis en Hardware en Software. Zij zijn akkoord gegaan met de nieuwe verdeling. De volledige verdeling is te vinden in het requirements document.

Bij het onderscheiden van de requirements heb ik twee groepen gemaakt, namelijk: Functional requirements en non-functional requirements. Deze twee groepen worden hieronder toegelicht.

5.3.1 Functional requirements

Deze requirements zijn een vertaling van de genoemde use-cases en hoe deze kunnen worden toegepast binnen de CRB. Deze requirements zijn onderscheiden aan de hand van de prioriteiten en de bronnen zijn bij elke requirement vermeld. Hieronder (Tabel 5) worden enkele requirements genoemd als voorbeeld. Het volledige overzicht is te vinden in het requirementsdocument.

#	Eis	Omschrijving	Prioriteit ⁴	Bron ⁵
F01	Registreren op CRB	Gebruikers moeten zich kunnen registreren in het CRB systeem met behulp van Facebook-, twitter- en/of google-account. Gebruiken OAuth technologie.	M	ST, BG
F03	Verwijderen CRB account	Gebruikers moeten zich kunnen verwijderen van het CRB systeem	C	OP
F05	Verwijderen P2P-websites	Gebruikers moeten kunnen opgeven op welke P2P-websites zij niet meer actief zijn.	S	ST
F07	Ophalen van de reputaties van de P2P-websites	De reputaties worden van de P2P-websites opgehaald wanneer er een nieuwe reputatie toegevoegd wordt.	W	OP, BG
F08	Periodiek ophalen van de reputaties van de P2P-websites	De reputaties worden met een bepaalde frequentie opgehaald van de P2P-websites.	M	ST

Tabel 5: voorbeelden functional requirements

5.3.2 Non-functional requirements

De non-functional requirements beschrijven de kwaliteitseisen van het systeem. Hiervoor is de ISO 9126⁶ gebruikt. Bij Tabel 6 zijn enkele non-functional requirements te vinden. Het volledig overzicht is te vinden in het requirements document.

#	Eis	Omschrijving	Soort	Prioriteit ⁴	Bron ⁵
NF01	Bescherming persoonsgegevens	Het systeem moet voldoen aan de voorwaarden in de wet Bescherming persoonsgegevens om de gegevens van de gebruikers te beschermen.	F	M	ST, OP

4 Afkortingen voor het gebruik van MoSCoW methode : **(M)**ust have, **(S)**hould have, **(C)**ould have en **(W)**ould like to have.

5 Afkortingen die bij de kolom Bron zijn gebruikt:

- ST: door de student opgesteld aan de hand van de beschikbare systeembeschrijvingen en gesprekken met de opdrachtgever;
- OP: Opdrachtgever dhr. Frank Titulaer;
- BG: begeleiders.

6 In de non-functional requirements tabel zijn de volgende afkortingen gebruikt:

F: Functionaliteit, BT: Betrouwbaarheid, BR: Bruikbaarheid, ON: Onderhoudbaarheid, OV: Overdraagbaarheid, EF: Efficiëntie.

NF05	ASP.NET MVC gebruiken voor het bouwen van het systeem	Het systeem wordt gebouwd met behulp van .NET technologie. Hiervoor wordt gebruik gemaakt van APS.NET MVC 4 C#.	OV	M	BG
NF06	Het systeem op de Azure cloud omgeving hosten	Het systeem moet in de Azure websites cloud omgeving worden gehost.	BR	M	BG, OP

Tabel 6: voorbeelden non-functional requirements

6. Onderzoek CRB

In dit onderdeel wordt op het onderzoek van CRB ingegaan. De werkzaamheden voor de totstandkoming van het onderzoek worden uitgelegd en er wordt aangegeven hoe ik tot de conclusies en onderzoeksresultaten ben gekomen. Het doel van het onderzoek kan worden samengevat in de volgende punten:

- het onderzoeken van de werking van de verschillende reputatie systemen op de P2P-websites;
- het vaststellen hoe wij de reputaties van de gebruikers van deze websites kunnen ophalen;
- het vaststellen hoe verschillende vormen van de beoordelingen op verschillende P2P-websites kunnen worden samengevoegd tot een totale beoordeling.

6.1 Onderzoeksvragen

Onderzoeksvragen zijn tijdens het opstellen van het plan van aanpak geformuleerd. Zoals in §4.1 is vermeld is voor het uitvoeren van het onderzoek de methode van Nel Verhoeven gebruikt: de "Wat is onderzoek?"-methode.

Daarnaast is een website geraadpleegd die zich aansluit bij het boek van Nel Verhoeven. Bij de website (www.onderzoekinstappen.nl) kunnen op een eenvoudige manier verschillende onderzoeksfases doorlopen worden door de controlevragen en checklijsten op deze website te raadplegen.

Om de bovengenoemde doelen van het onderzoek te bereiken dient er antwoord te worden gegeven op de hoofdvraag door middel van het formuleren en beantwoorden van de deelvragen.

De hoofdvraag was:

Hoe werken bestaande reputatie systemen en hoe kunnen we verschillende reputatiesoorten op verschillen websites naar een totale reputatie vertalen?

Met behulp van de website www.onderzoekinstappen.nl en definities van verschillende type vragen heb ik de deelvragen op de volgende manier verdeeld:

Vraagtype	Bijbehorende onderzoeksvragen	Te gebruiken methode (3)
Beschrijven	<ul style="list-style-type: none"> • Welke reputatiesystemen worden nu gebruikt door de verschillende P2P-websites? 	<ul style="list-style-type: none"> • Inhoudsanalyse: met deze methode wordt informatie verzameld over de reputatiesystemen op verschillende P2P-websites uit geschreven artikelen, interviews video-opnames en andere bronnen. • Casestudy: bij het gebruik van deze methode worden de reputatiesystemen op de P2P-websites en de werking daarvan één voor één onderzocht.
Definiëren	<ul style="list-style-type: none"> • Wat zijn de wegingsschalen die gebruikt worden door de verschillende P2P-websites? • Wat zijn de mogelijkheden voor dataverzameling (reputaties) indien een P2P-website geen medewerking biedt? 	<ul style="list-style-type: none"> • Analyse van bestaande materiaal: analyseren welke wegingsschalen gebruikt worden op de websites en hoe de reputaties worden bijgehouden. Dit laatste is belangrijk voor de dataverzameling zonder medewerking van de websites. • Inhoudsanalyse: hierbij wordt gezocht naar de geschreven artikelen en informatie over technieken om gegevens uit verschillende internetpagina op te halen.
Verklaren	<ul style="list-style-type: none"> • Hoe kunnen de verschillende reputatievormen tot een totale reputatie worden samengevoegd? • Hoe wordt de communicatie verzorgd tussen het CRB en de P2P-websites? 	<ul style="list-style-type: none"> • Observatie: observeren hoe gebruikers hun waarderings invoeren en wat hun wegingen zijn. • Experiment: uitzoeken of er overeenkomsten zijn tussen verschillende reputatievormen. • Enquête: de oprichters/beheerders van de P2P-websites enquêteren over CRB werking en

		mogelijke communicatie tussen hun websites en CRB.
Evalueren	<ul style="list-style-type: none"> In welke mate kunnen de P2P websites medewerking bieden voor het realiseren van CRB? 	<ul style="list-style-type: none"> Open interview: de oprichters/beheerders van de P2P-websites interviewen over de mogelijkheden op CRB en mogelijke medewerking. Enquête: de oprichters/beheerders van de P2P-websites enquêteren over CRB werking en mogelijke communicatie tussen hun websites en CRB.

Tabel 7: onderzoeksvragen en de types en te gebruiken methoden

6.2 Gebruikte onderzoeksmethoden

Tijdens het uitvoeren van het onderzoek heb ik de volgende onderzoeksmethoden gebruikt:

- **Inhoudsanalyse:** bij het gebruiken van deze methode is gezocht naar de bekende P2P-website met behulp van het zoeken op internet en op nieuws websites.
- **Casestudy:** verschillende reputatiesystemen en de werking daarvan heb ik met behulp van deze methode onderzocht.
- **Analyseren van bestaand materiaal:** met behulp van deze methode zijn verschillende wegingschalen die gebruikt worden voor de beoordelingen onderzocht.
- **Experiment:** deze methode is gebruikt bij het omrekenen van de wegingschalen op verschillende P2P-websites om deze wegingschalen met elkaar te kunnen samenvoegen.
- **Enquête:** de beheerders en/of oprichters van verschillende P2P-websites zijn geënquêteerd over de mogelijkheden tot samenwerking en om te kijken of er belangstelling is voor de CRB.
- **Open interview:** sommige enquêteerrespondenten waren niet bereid om de enquête in te vullen; daarvoor is deze methode gebruikt om hun mening alsnog te verkrijgen over de CRB.

6.3 Verzamelen P2P-website adressen

Voor de dataverzameling heb ik de genoemde methoden in §6.2 gebruikt. De toepassing van deze methoden wordt in de volgende paragrafen uitgelegd.

Voor het beantwoorden van verschillende deelvragen heb ik een lijst nodig van P2P-websites die op verschillende gebieden actief zijn. Ik heb via verschillende bronnen deze lijst verzameld:

- Een deel van de lijst heb ik via email van de opdrachtgever gekregen.⁷ Deze websites waren: toogethr.com, snappcar.nl, peerby.com, spullendelen.nl, mywheels.nl, plot-app.com, gidsy.com/city/amsterdam, justrent.it en thuisafgehaald.nl.
In de lijst van de opdrachtgever waren andere websites benoemd, maar na het bezoeken van die websites heb ik geconstateerd dat ze niet interessant waren voor het onderzoek door beperkingen bij het registreren of omdat ze nog niet actief waren, zoals sindo.nl .
- Een lijst die ik heb opgesteld van enkele grote spelers op het gebied van P2P-websites, zoals: ebay.nl, amazon.com, marktplaats.nl, tweedehands.nl en bol.com.
- Enkele suggesties van de begeleiders.
- Gevonden P2P-websites via google zoekmachine.

⁷ Email van dhr. Frank Titulaer op 4 april 2012.

In totaal zijn er 21 websites onderzocht. De volledige lijst is te vinden in het onderzoeksrapport.⁸

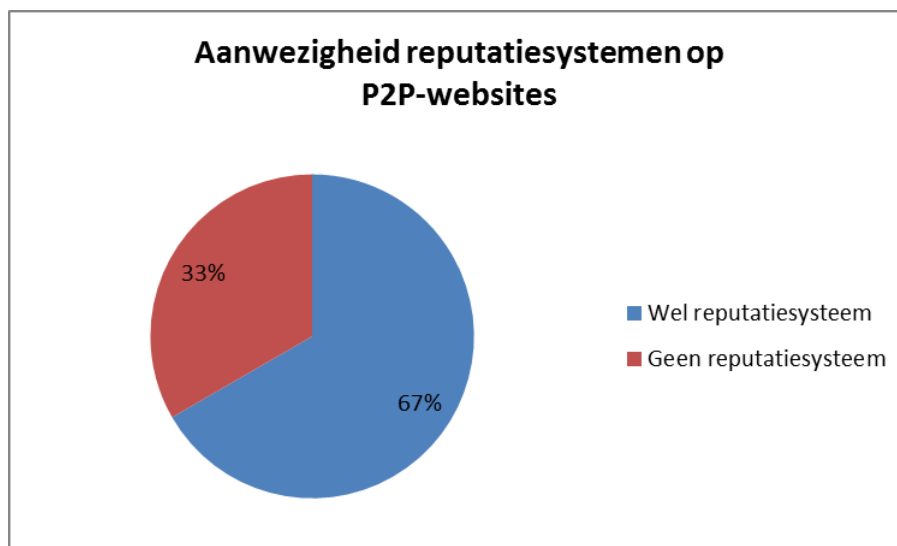
Naast de bovengenoemde bronnen zijn de volgende selectiecriteria gehanteerd voor het kiezen van de onderzochte P2P-websites:

- De geselecteerde websites moeten zowel grote als kleine spelers zijn binnen het gebied van P2P-websites om te kunnen vast stellen hoe belangrijk zulke systeem is voor deze websites;
- Enkele websites moeten over een reputatiesysteem beschikken, zodat wij deze systemen kunnen onderzoeken en kunnen bekijken hoe wij verschillende reputatievormen kunnen samenvoegen;
- De geselecteerde websites moeten op verschillende gebieden actief zijn, zodat wij de interesse voor de CRB binnen verschillende gebieden kunnen onderzoeken;
- Er dienen ook websites zonder reputatiesystemen onderzocht te worden om hun diensten te kunnen vergelijken met de websites die wel over een reputatiesysteem beschikken. Het is ook belangrijk om te weten of deze websites wel interesse hebben in de CRB.

6.4 Onderzoeken reputatiesystemen

Het onderzoeken van de reputatiesystemen heeft als doelstelling het vaststellen welke reputatievorm (wegingsschaal) het beste te gebruiken is en hoe verschillende reputatievormen kunnen worden samengevoegd voor het berekenen van de totale reputatie.

Bij het onderzoeken van de reputatiesystemen zijn alle aanwezige websites in de lijst bezocht. Eerst heb ik gekeken of deze websites wel over een reputatiesysteem beschikken. Het resultaat was dat 67% van deze websites een reputatiesysteem hebben. (Figuur 8)



Figuur 8: Aanwezigheid reputatiesystemen op de P2P-websites

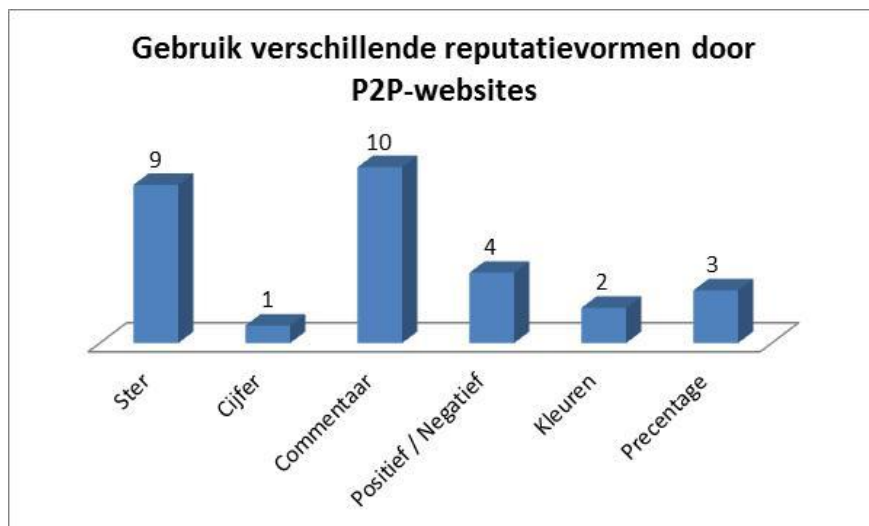
Daarna zijn al deze reputatiesystemen verder onderzocht en heb ik gekeken welke reputatievorm wordt gebruikt op deze websites. Bij elke vorm heb ik onderzocht wat de definitie van deze vorm is en een voorbeeld vermeld van een website die deze vorm gebruikt. Als voorbeeld is op (Figuur 9) de stervorm van de beoordeling te zien. De andere vormen zijn te vinden in het onderzoeksrapport.

⁸ Onderzoeksrapport, Tabel 4: Onderzochte P2P-websites.

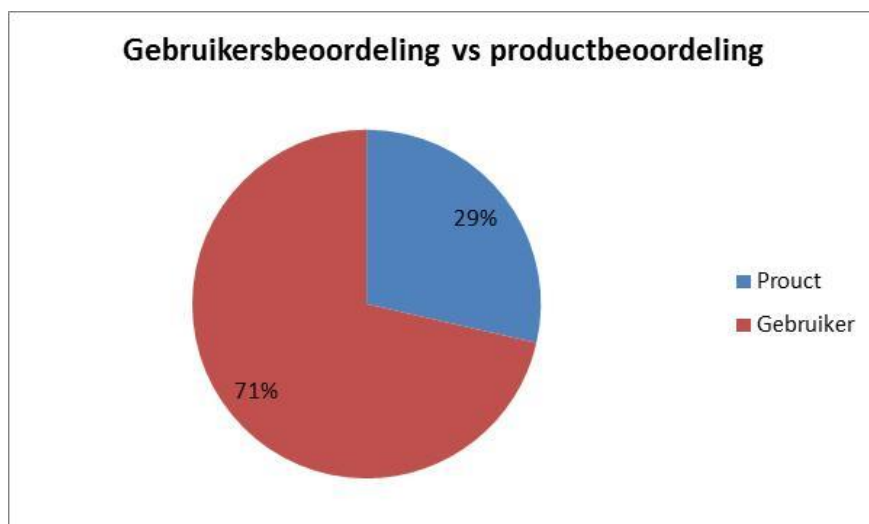
Criteria	Gemiddelde beoordeling	Aantal beoordelingen
Object voldoet aan beschrijving ★★★★★		397

Figuur 9: Ster beoordelingsvorm voorbeeld op ebay.nl

Van verschillende reputatievormen heb ik een lijst opgesteld en de resultaten daarvan zijn verwerkt in de grafiek hieronder (Figuur 10).

**Figuur 10: Gebruik verschillende reputatievormen door P2P-websites⁹**

Hierna is een derde aspect onderzocht en dit was hoe de beoordeling wordt gedaan op deze websites. Worden de gebruikers beoordeeld of de producten. Het antwoord daarop was dat op 71% van de websites de gebruikers worden beoordeeld en op de andere websites (29%) de producten. (Figuur 11) Omdat er weinig P2P-websites gebruiken van de productbeoordeling en is het moeilijk te combineren met de gebruikersbeoordelingen, is deze vorm buiten de scope gezet van de CRB.

**Figuur 11: Gebruikersbeoordeling vs productbeoordeling**

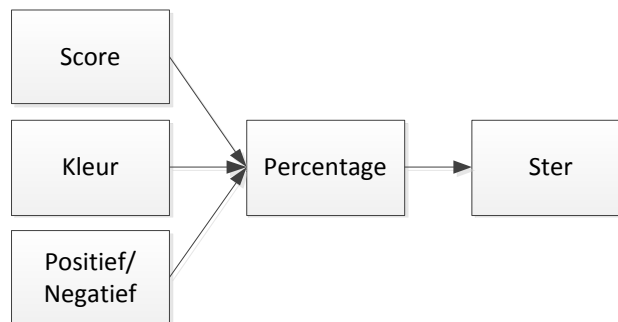
Na het verzamelen van de gegevens over de reputatievormen heb ik onderzocht hoe ik de verschillende vormen tot een totale reputatie kan samenvoegen.

⁹ Een P2P-website kan meerdere reputatiesvormen gebruiken.

Om dit laatste te bereiken is het belangrijk om te kiezen welke vorm ik ga gebruiken voor de CRB. De keuze is uiteindelijk op de ster vorm gevallen omdat:

- ster vorm een van de meest gebruikte vormen is op verschillende reputatiesystemen;
- het geen verrassingen meebrengt voor de gebruikers van de CRB, doordat er veel gebruik van deze vorm wordt gemaakt op verschillende P2P-websites,;
- het eenvoudig is weer te geven en te verwerken in verschillende componenten van het systeem;
- het omzetten van de verschillende vormen naar deze vorm eenvoudiger zou zijn dan het gebruiken van andere vorm.

Bij het omzetten van de beoordelingen van verschillende vormen naar de ster-vorm heb ik een procedure opgesteld. Deze procedure is in Figuur 12 te vinden.



Figuur 12: omzetten van verschillende reputatievormen naar de Ster model

Deze procedure houdt in dat 3 van de vormen (Positief/Negatief, Kleur en Score) overeenkomen qua inhoud en daardoor kunnen we deze vormen op dezelfde manier omzetten naar de Ster-vorm. We zetten eerst alle vormen om naar de Percentage-vorm en daarna wordt deze vorm naar de Ster-vorm omgezet.¹⁰

Hieronder is te vinden hoe verschillende vormen worden omgezet:

- **Percentage:** $\text{percentage} / 20 = 5$
- **Positief/Negatief:** $((\text{positief} / (\text{positief} + \text{negatief})) * 100) / 20 = 5$

Het is helaas niet mogelijk om de commentaar-vorm te converteren naar de Ster-vorm door de volgende redenen:

- Er zijn sommige websites die commentaar gebruiken alleen voor de positieve beoordelingen zoals: thuisafgehaald.nl. Met alleen positieve beoordelingen is het niet mogelijk om met andere vormen samen te voegen omdat er geen score en percentage berekend kan worden.
- Het is niet mogelijk om in een oogopslag te zeggen dat een commentaar positief of negatief is voordat we de gehele commentaar hebben gelezen en dit is niet eenvoudig te automatiseren.
- Er zijn bepaalde technieken waarmee een tekst kan worden gefilterd en aan de hand daarvan wordt vastgesteld of het een positieve of negatieve tekst is, maar deze technologieën zijn relatief nieuw en op dit moment ligt dit buiten de scope van het onderzoek.

¹⁰ Onderzoeksrapport §4.3.

6.5 Enquête P2P-website beheerders

Om de mening van de P2P-websites over de CRB te verzamelen heb ik een enquête opgesteld voor de beheerder/oprichter van deze websites. De enquête is belangrijk voor de beeldvorming over de visie van de websites over verschillende aspecten van CRB.

Deze aspecten zijn:

- Hoe groot deze websites zijn, hoeveel gebruikers en bezoekers zij hebben;
- Of deze websites over een reputatiesysteem beschikken;
- Indien een reputatiesysteem aanwezig is, wie wordt beoordeeld (aanbieder, afnemers of beide);
- Of deze websites medewerking kunnen bieden aan het tot stand komen van de CRB;
- Of ze de beoordelingen op hun websites beschikbaar kunnen stellen aan de CRB;
- Of deze website deel wil nemen aan de pilot fase;
- Of ze bereid zijn om de CRB te gebruiken als het er daadwerkelijk komt;
- Of ze bereid zijn om te betalen voor deelname en/of gebruik van de CRB;

Het tot stand komen van de enquête was als volgt. Ik had een eerste versie opgesteld van de enquête en opgestuurd naar de begeleiders en de opdrachtgever. Als commentaar kreeg ik de volgende punten over de eerste versie:

- Iets meer te vertellen over de voordelen van de CRB;
- De verschillende keuzemogelijkheden voor het aantal gebruikers en bezoekers was erg hoog, zeker voor de kleine P2P-websites. Er moest een andere verdeling worden gemaakt;
- Vervangen van de termen (Kopers en verkopers) met (Aanbieders en afnemers);
- Formulering bijwerken/aanpassen van enkele vragen;

Met behulp van deze opmerkingen heb ik een nieuwe versie van de enquête opgesteld en weer naar hun gestuurd. Er waren nog enkele opmerkingen die hieronder zijn vermeld:

- De verklaring van het geheimhouding van de verzamelde gegevens;
- Verschil tussen het deelnemen aan de CRB en het gebruiken daarvan;¹¹
- De voorgestelde abonnement op de CRB verduidelijken op basis van dag/maand/jaar.

Na het verwerken van deze opmerkingen is een definitieve versie opgesteld van de enquête.¹¹

6.5.1 Online enquête tool

Om het invullen van de enquête eenvoudiger te maken heb ik de (ThesisTools) online enquête tool gebruikt. Ik heb daarin de enquête geladen zodat het makkelijk te bereiken was voor de respondenten.¹²

6.5.2 Enquête respondenten

De populatie van deze enquête is erg klein en niet makkelijk te bereiken. Ik had ten minste 10 respondenten nodig om de gestelde aspecten goed te kunnen beoordelen. Het aantal 10 was gekozen in overleg met de begeleiders en opdrachtgever. We hebben

¹¹ Zie het onderzoeksrapport, Bijlagen, A Enquête

¹² De enquête is te vinden op de volgende link: <http://www.thesisools.com/crb>.

samen gekeken wat haalbaar is met deze populatie. Het benaderen van een P2P-websitebeheerder en/of oprichter en het overtuigen van deze persoon om de enquête in te vullen was een moeilijke taak. Daardoor is dit aantal gekozen in overleg met de betrokkenen.

Eerst had ik de enquête naar meer dan 20 websites gestuurd¹³. Van twee website heb ik response gekregen. Daarom heb ik besloten om de enquête naar veel meer websites te sturen zodat er meer response zou komen. Na lang zoeken op internet met behulp van google zoekmachine kon ik een lijst opstellen met 80 P2P-websites¹⁴ en de contactgegevens daarvan.

Na het opsturen van de enquête naar deze websites had ik de andere benodigde respondenten (8 respondenten) binnen gehaald, en daarmee kon ik de resultaten verder analyseren met in totaal 10 respondenten.

6.5.3 Versturen van de enquête

Het versturen van de enquête heb ik via e-mail gedaan. Daarvoor heb ik een standaard email bericht opgesteld waar in het kort de werking van de CRB, de voordelen daarvan en de doel van de enquête worden uitgelegd. Daarin heb ik ook de link geplaatst naar de online enquête.

Deze e-mail heb ik naar alle verzamelde contactgegevens van de P2P-websites gestuurd. In totaal waren het er meer dan 100 websites.

6.5.4 Open interview met marktplaats.nl

Om meer response te krijgen voor het onderzoek heb ik geprobeerd om naast de enquête ook open interviews te houden. Na het benaderen van meerdere websites kon ik in contact komen met één van de grote spelers op P2P gebied de marktplaats.nl. Na lang zoeken en proberen van verschillende media zoals e-mail, bellen en netwerksites, ben ik in contact gekomen met de eindverantwoordelijke voor de koper- en verkoperservaring dhr. Victor Croes⁽¹⁰⁾ via de netwerksite LinkedIn.

Ik had een poging gedaan om met hem in contact te komen door een bericht te sturen via LinkedIn. Dhr. Croes verraste mij met het bericht dat hij in gelegenheid is om met mij telefonisch te spreken over de CRB.

Na het maken van de afspraak hebben we elkaar gesproken en de resultaten van het gesprek waren als volgt:

- Op dit moment heeft marktplaats geen interesse in de CRB, om twee redenen: Eerste reden is dat marktplaats nu geen beoordelingssysteem heeft op alle rubrieken, behalve op de rubriek 'Diensten en Vakmensen'. De tweede reden is dat de CRB nu een startend project is, en ondersteuning zal bieden aan de concurrenten van marktplaats.
- Marktplaats zou mee kunnen doen in een later stadium als aan twee voorwaarden is voldaan:
 - De CRB heeft veel gebruikers en veel beoordelingen opgeslagen;
 - De CRB is verbonden aan een vertrouwde partij die bekend is bij mensen zoals: consumentenbond.
- Hij vond dat de CRB op dit moment vooruit loopt ten opzichte van de markt maar het is wel belangrijk om vast te stellen dat zulke systemen technisch mogelijk zijn. Op het moment dat de gebruikers tegelijkertijd op verschillende P2P-websites actief zijn en genoeg beoordelingen hebben zou dan de CRB een waardevol systeem kunnen zijn.

¹³ Zie de P2P-websites lijst op het onderzoeksrapport (Tabel 4).

¹⁴ Zie de bijlage B op het onderzoeksrapport.

- De CRB zou op dit moment waardevol zijn alleen voor kleine P2P-websites. Daarmee kunnen ze sneller groeien.
- Bij de realisatie van CRB zou Sociaal media zeker een grote rol kunnen spelen.

6.5.5 Enquête resultaten

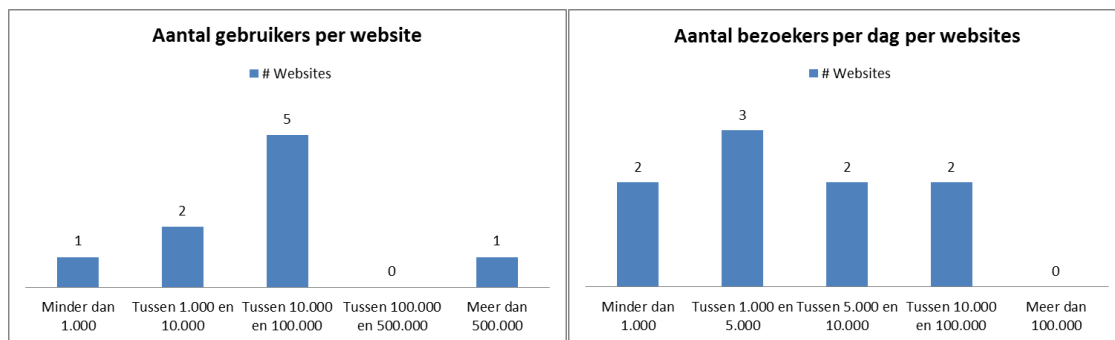
Na het behalen van de beoogde aantal respondenten heb ik de antwoorden van de enquête geanalyseerd.

Het uitvoeren van de enquête had langer geduurd dan ik had verwacht. Dat kwam voornamelijk door de moeilijk te bereiken respondenten. Daarom heb ik geprobeerd om zoveel mogelijk websites te bereiken om aan mijn benodigd aantal respondenten te komen.

De websites die hebben deelgenomen aan de enquête waren: Thuisafgehaald.nl, Peerby.com, Toogethr.com, Aanbodnet.nl, Marktplaats.nl, Snuffelbeurs.nl, Tweedehands.net, Speurmarkt.nl, Zoekertjes.net, Aanbodpagina.nl.

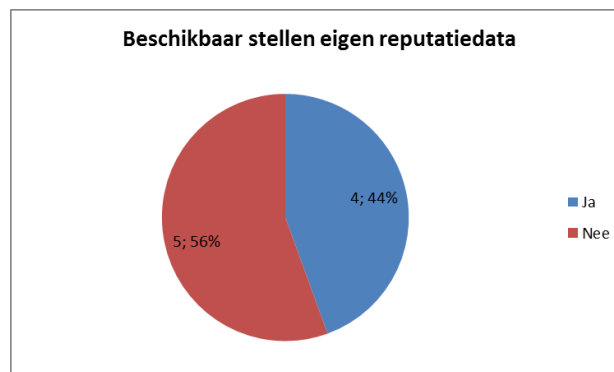
Aan de hand van de analyse die ik heb uitgevoerd, kan ik de resultaten in de volgende punten samenvatten:

- Websites die actief zijn binnen verschillende gebieden hebben deelgenomen aan de enquête;
- Er hebben twee grote P2P-websites deelgenomen aan de enquête¹⁵;



Figuur 13: aantal gebruikers en bezoekers per P2P-website

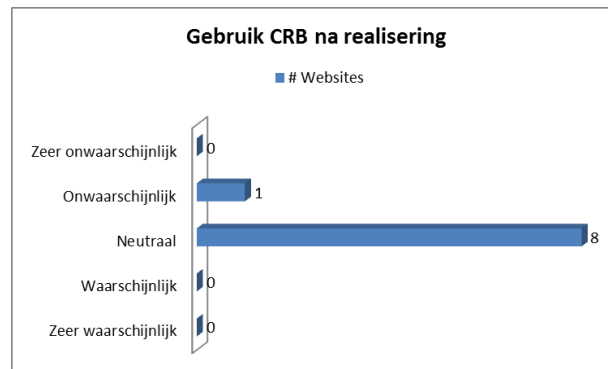
- De meeste respondenten (67%) hebben geen reputatiesysteem;
- De meeste websites zijn neutraal (weten nog niet) ten opzichte van medewerking bij het tot stand komen van de CRB;
- Bijna de helft (44%) is bereid om eigen reputaties beschikbaar te stellen voor de CRB;



Figuur 14: beschikbaar stellen reputatiedata voor de CRB

¹⁵ In Figuur 13 is één grote P2P-website genoemd, de reden daarvan is dat marktplaats.nl heeft geen enquête ingevuld en bij deze weergave zijn alleen de enquête resultaten opgenomen.

- Meer dan de helft was neutraal ten opzichte van het deelnemen aan de pilotfase;
- Bijna 90% was 'Neutraal' ten opzichte van het gebruik van de CRB na de realisatie;



Figuur 15: gebruik CRB na de realisering

- Bijna 80% was tegen het betalen om deel te nemen aan de CRB;
- 8 respondenten zouden niet willen betalen bij het gebruik van de CRB.

De volledige resultaten zijn te vinden in het onderzoeksrapport onder §5.5 en bijlage D.

6.5.6 Evaluatie enquête

De belangrijkste bevindingen van de enquête zijn te vinden in de volgende punten:

- Er is belangstelling voor de CRB door P2P-websites in verschillende gebieden;
- Op dit moment zijn er relatief weinig P2P-websites met een reputatiesysteem, maar uit de belangstelling die deze websites hebben getoond in de CRB, kunnen we concluderen dat na realisatie van de CRB er meer websites gebruik zullen maken van de CRB, en er meer belangstelling komt voor reputatiesystemen;
- Enkele websites (Aanbodpagina.nl, Toogethr.com, Aanbodnet.nl en Speurmarkt.nl) hebben aangegeven dat ze interesse hebben in de ontwikkelingen rondom de CRB. Daaruit kunnen we concluderen dat er na realisatie van het PoC nog meer belangstelling zal ontstaan;
- Er waren mogelijkheden tot samenwerking bij de pilotfase van de CRB, maar helaas alle website die zich open hadden gesteld aan deze mogelijkheid, hebben geen reputatiesystemen en daarmee kunnen ze niet deelnemen aan de pilotfase.

De volledige lijst van de bevindingen is te vinden in het onderzoeksrapport onder §5.6.

6.6 Communicatiemogelijkheden met de P2P-websites

Om de CRB te kunnen realiseren hebben we reputaties nodig van verschillende P2P-websites. Zonder de beoordelingen van deze websites kan de CRB geen service bieden. Daardoor heb ik twee scenario's vastgesteld voor het onderzoeken van de communicatie tussen de CRB en de P2P-websites. Deze zijn:

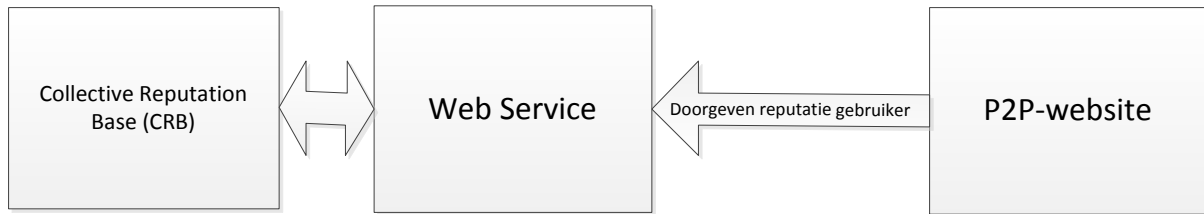
- Communicatiemogelijkheden bij medewerking van de P2P-website;
- Communicatiemogelijkheden zonder medewerking van de P2P-website.

Om een antwoord te kunnen vinden voor deze scenario's heb ik op internet gezocht naar de mogelijke oplossingen. De gevonden oplossingen zijn¹⁶:

¹⁶ Een uitgebreide uitleg is te vinden in het onderzoeksrapport §6.1 en §6.2.

- **Bij medewerking P2P-website:**

- **Bouwen van een Web Service op de CRB:** met behulp van deze oplossing wordt een Web Service op de CRB gebouwd waar de P2P-websites tegen kan spreken en de reputaties kan opsturen naar de CRB.



Figuur 16: Web Service op de CRB

- **Bouwen van een Web Service op de P2P-website:** bij deze oplossing wordt op de P2P-website een Web Service gebouwd de CRB tegen kan spreken om de benodigde reputaties ophalen.



Figuur 17: Web Service op de P2P-website

- **Periodiek file uitwisseling:** hiermee kunnen de P2P-websites om een bepaalde periode de reputatiegegevens opleveren bij de CRB.

- **Zonder medewerking P2P-website:**

- **Html scan:** bij deze oplossing kunnen de reputaties worden opgehaald door middel van het scannen van de html pagina waar de reputatie staat van de gebruiker. Deze oplossing is niet geschikt voor alle websites, er zijn enkele criteria waaraan voldaan moet zijn bij een website voordat de html scan kan worden gebruikt. Deze criteria zijn:
 - De P2P-website moet beschikken over een speciale pagina voor de gebruikersbeoordelingen met een vaste vorm voor de link. Bijvoorbeeld op ebay.nl ziet de gebruiker beoordelingenpagina link er als volgt uit: <http://myworld.ebay.nl/<gebruikersnaam>>
 - Op de P2P-website moet de gebruiker worden beoordeeld en niet het product;
 - De gebruikersnaam en/of de gebruikers-ID van de gebruiker moet bekend zijn bij de CRB;
 - Op de beoordelingspagina moet één van de volgende gegevens bekend zijn:
 - Het gemiddelde van de beoordelingen;
 - Aantal positieve en negatieve beoordelingen;
 - Alle positieve en negatieve beoordelingen van de gebruiker op één pagina;
 - Beoordelingspercentage van de gebruiker;
 - De beoordelingen pagina moet een openbare pagina zijn, dus zonder inloggen op de website de beoordelingen moeten zichtbaar zijn.

Tijdens het uitvoeren van de enquête had ik enkele kandidaten voor medewerking, vooral in de pilotfase, maar zoals ik heb vermeld kon er helaas niet samengewerkt worden, omdat deze websites geen reputatiesysteem hebben. Daarom is de html scan overgebleven om tijdens het bouwen van het PoC te kunnen gebruiken.

Om dit laatste te kunnen testen en de mogelijkheden daarvan te onderzoeken, heb ik een console applicatie gebouwd. Deze applicatie haalt de reputatie van de gebruiker op met behulp van zijn gebruikersnaam en dit gegeven wordt omgezet in een cijfer waarna het kan worden gebruikt voor de Ster-vorm. De volledige omschrijving van deze applicatie is te vinden in het onderzoeksrapport §6.2.

6.7 Evalueren onderzoek

Als evaluatie van het onderzoek heb ik de antwoorden geformuleerd op de gestelde deelvragen. Hieronder zijn enkele punten te lezen die belangrijk zijn voor het opstellen van het architectuurontwerp en het PoC:

- Als reputatievorm wordt de Ster-vorm gebruikt; deze vorm wordt met 5 sterren weergegeven en aan de hand van de score worden de sterren gevuld;
- De methode die gebruikt kan worden voor het samenvoegen van verschillende wegingschalen;
- Het beoordelen van personen en niet de producten;
- Het gebruik van de Html scan voor het ophalen van de gebruikersreputaties.

De volledige resultaten en de aanbevelingen zijn te vinden in het onderzoeksrapport hoofdstuk 7.

6.8 Bespreken bevindingen met de opdrachtgever

Na het afronden van het onderzoek heb ik de bevindingen en de resultaten met de opdrachtgever besproken. Deze bespreking was een verificatie van de gemaakte keuzes voor het samenstellen van het architectuurontwerp en het bouwen van het PoC.

De opdrachtgever was tevreden met deze bevindingen vooral met het mogelijkheid om de reputaties te kunnen ophalen zonder medewerking van de P2P-website. Het was een belangrijk punt voor hem omdat met het gebouwde het PoC hij de waarde en het belang van de CRB kan laten zien.

7. Architectuurontwerp

In dit hoofdstuk wordt het tot stand komen van het architectuurdokument beschreven. Het architectuurdokument is een belangrijk onderdeel van het project. Hiermee wordt een vertaling gemaakt van de requirements naar het Proof-of-Concept.

Om te beginnen met dit document heb ik eerst de boek Software Systems Architecture⁽⁶⁾ geraadpleegd voor het onderbouwen van mijn architectuurdokument. Ik heb ook de template gebruikt die te verkrijgen is via website (<http://www.viewpoints-and-perspectives.info/>). Deze website is opgesteld door de schrijvers van dit boek.¹⁷

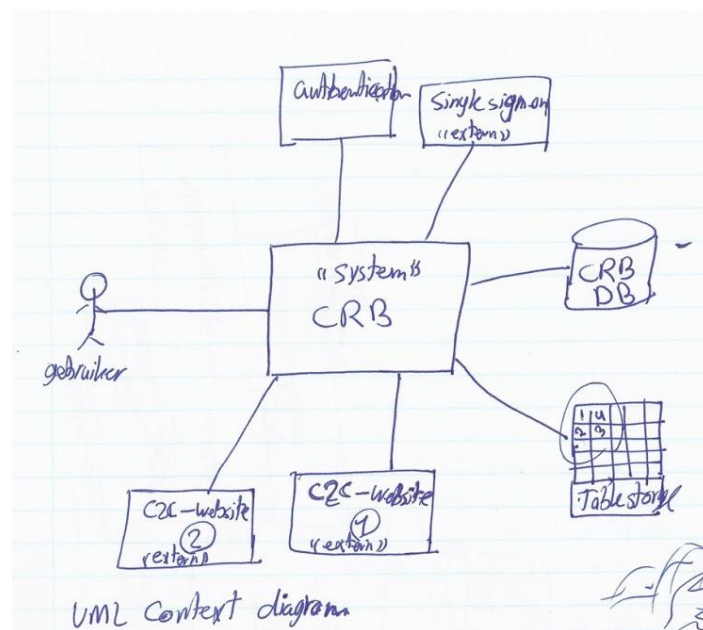
De template was in het Engels geschreven, om daarvan gebruik te kunnen maken heb ik het naar Nederlands vertaald. Alleen de namen van de Viewpoints en sommige onderdelen heb ik in het Engels gelaten vanwege de bekendheid in dit vakgebied.

Verder heb ik tijdens het opbouwen van het architectuurdokument meerdere bronnen geraadpleegd voor het beschrijven en opstellen van verschillende viewpoints zoals:

- Praktisch UML boek (8): voor het opstellen van verschillende UML-diagrammen heb ik dit boek gebruikt. De UML diagrammen die ik heb gebruikt zijn: context diagram, sequentiediagram, componentdiagram, functioneel structuur diagram, system data structuur, EER-model, RRM-model, concurrency model, module structuur, klasse diagram en deployment model.
- Principles van database boek (12): bij het opstellen van de database ontwerp in de information view heb ik dit boek gebruikt. Vooral voor het opstellen van de EER-RRM-model.
- Voorbeelden architectuurontwerp documenten op internet en voorbeelden van verschillende UML diagrammen.

7.1 Gesprekken met de begeleider en de opdrachtgever

Om te beginnen met het opstellen van het architectuurdokument heb ik afgesproken met dhr. Leeuwesteijn (begeleider/architect) en vóór het gesprek heb ik een paar schetsen gemaakt die bij verschillende viewpoints horen. Een voorbeeld daarvan is te vinden op (Figuur 18)



Figuur 18: Concept Context diagram CRB

¹⁷ Nick Rozanski en Eoin Woods.

Tijdens het gesprek hebben we de volgende onderwerpen behandeld:

- Reviewen van de gemaakte schets;
- Hoe de interacties worden verzorgd tussen CRB en de P2P-websites;
- Hoe moet het datamodel er uit zien.

Daarnaast heb ik ook inhoudelijke opmerkingen gekregen van de vervangende begeleider dhr. Hans Kappert.

Bij het modelleren van verschillende onderdelen van het architectuurdokument heb ik UML (Unified Modeling Language) gebruikt, en voor het tekenen van de UML-diagrammen heb ik de Microsoft Visio 2012 gebruikt.

Met behulp van de inhoudelijke opmerkingen van de begeleider en de schets ben ik begonnen met het schrijven van het architectuur document en tekenen van de schets in de Visio modeleringstool.

De eerste 4 hoofdstukken van het document geven een beeld van: doelen, doelgroepen, huidige situatie, systeem omschrijving, systeem stakeholders en requirements en de architecturale krachten en beperkingen.

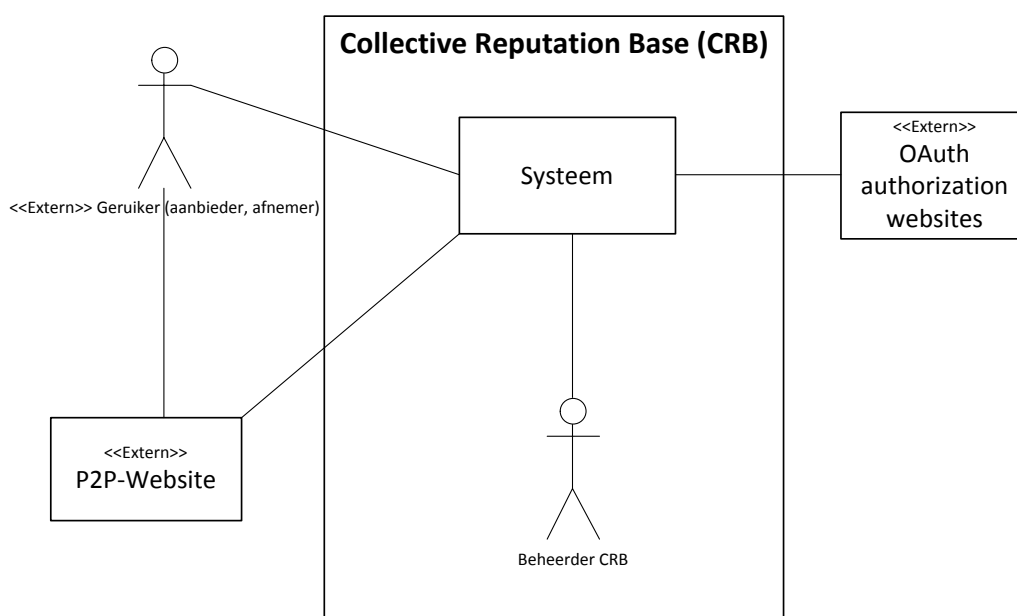
Daarna ben ik bezig geweest met het schrijven van verschillende viewpoints. Elk viewpoint is apart toegelicht in de onderstaande paragrafen.

7.2 Het opstellen van de Context viewpoint

De Context viewpoint beschrijft de relaties, afhankelijkheden en interactie tussen het systeem en zijn omgeving (de mensen, systemen en externe entiteiten waarmee samen wordt gewerkt).

Bij het opstellen van de Context viewpoint heb ik verschillende context diagrammen op internet en in het boek met elkaar vergeleken.

De schets die ik had gemaakt van de context had meer details dan wat nodig was voor dit soort diagrammen. Volgens dhr. Kappert moest het context diagram iets algemener worden opgesteld, zonder details. Dus bij het noemen van een systeem worden er geen componenten binnen dit systeem benoemd. Aan de hand daarvan heb ik de schets aangepast. De nieuwe versie van het context diagram is te vinden onder (Figuur 19).



Figuur 19: Context diagram

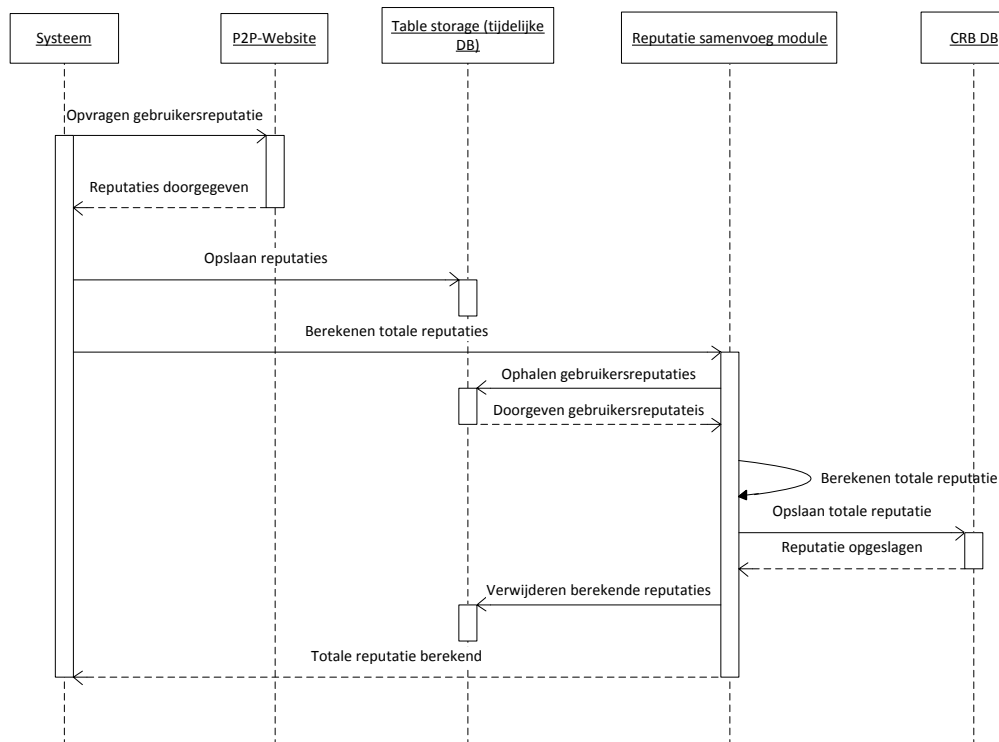
Naast het opstellen van het diagram heb ik alle benoemde elementen in het diagram, beschreven zodat er duidelijk wordt wat de functionaliteit is van dit element.

Hieronder zijn de beschrijvingen te vinden:

- **Gebruiker (aanbieder, afnemer):** dit zijn de gebruikers die op verschillende P2P-websites actief zijn en zich ook aanmelden bij de CRB om hun reputaties te kunnen verzamelen.
- **P2P-websites:** vraag-en-aanbod websites waar de hierboven genoemde gebruikers actief zijn en hun artikelen en diensten aanbieden en afnemen. P2P-websites kunnen zich aanmelden bij CRB als deelnemende websites zodat de gebruikersreputatie op deze websites kunnen worden opgehaald.
- **Systeem:** dit is het CRB systeem waarin de gebruikers zich aanmelden en aangeven op welke P2P-websites zij actief zijn. Met behulp van deze informatie worden hun reputaties verzameld uit deze websites. Deze reputaties worden samengevoegd zodat er een totale (gemiddelde) score berekend wordt voor de gebruiker. Met behulp van een Web API kunnen de P2P-websites de totale gebruikersreputatiescore opvragen en tonen op hun website.
- **Beheerder CRB:** dit is een interne actor die het CRB systeem beheert.
- **OAuth authorization websites:** met behulp van deze technologie kunnen de gebruikers zich registreren op de CRB met behulp van hun bestaande accounts op bijvoorbeeld Facebook, Twitter, Windows-Live en Google.

Na het opstellen en beschrijven van het context diagram heb ik enkele interactie scenario's opgenoemd, om de interactie tussen het systeem en de externe elementen te verduidelijken. Als voorbeeld heb ik het interactie scenario "Ophalen reputaties van P2P-website en berekenen totale reputatie op de CRB" opgenomen (Figuur 20). De andere scenario's zijn te vinden in het Architectuurontwerp document.

Ophalen reputaties van P2P-websites en
berekenen totale reputatie op de CRB



Figuur 20: sequentiediagram (Ophalen reputaties van P2P-websites en berekenen totale reputatie op de CRB)

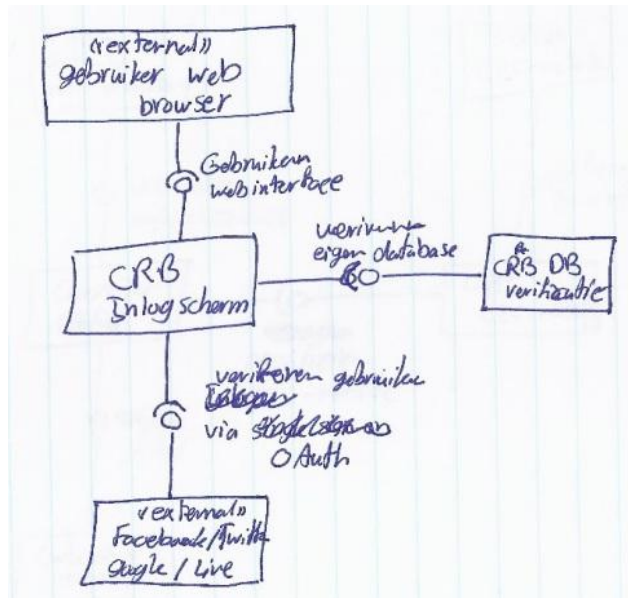
7.3 Het opstellen van de Functional viewpoint

De functional view van het systeem bepaalt de architectonisch belangrijke functionele elementen, de verantwoordelijkheden daarvan, de interfaces die zij aanbieden en de afhankelijkheden tussen de elementen.

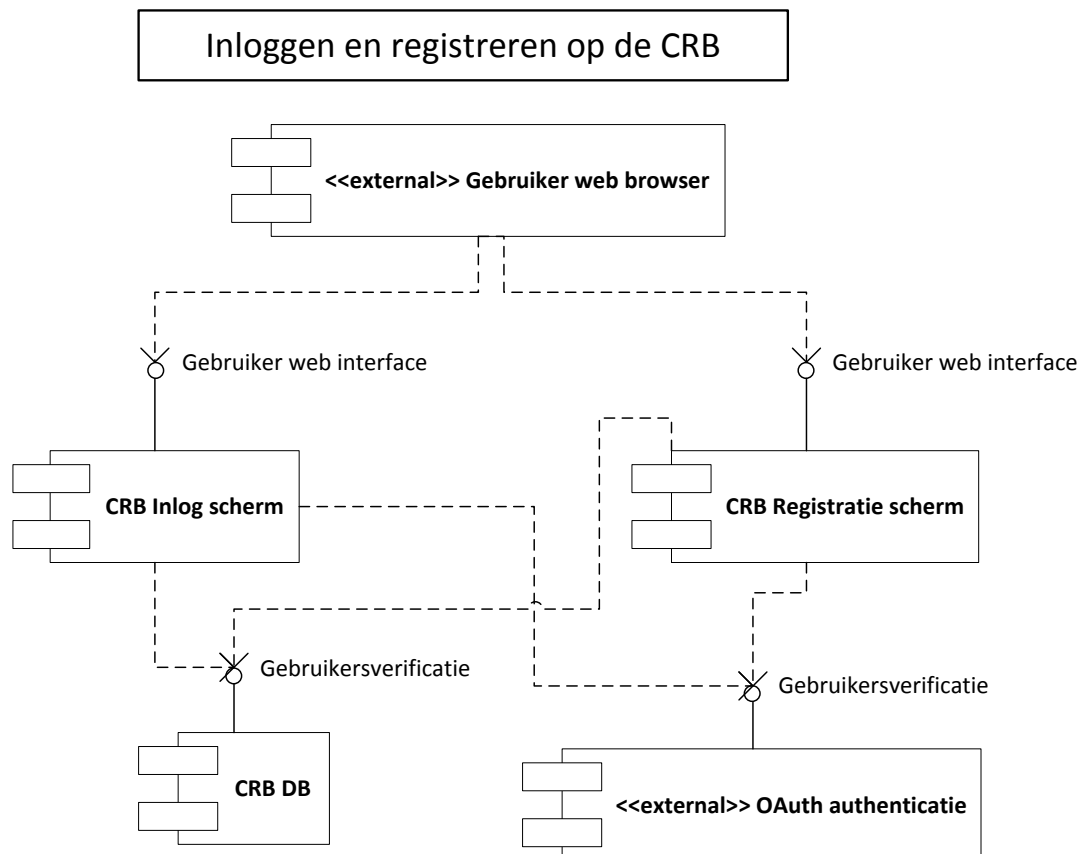
Om dit onderdeel te beschrijven had ik ook enkele schetsen gemaakt van component-diagrammen met het raadplegen van de use-case diagram en de functionele requirements die te vinden zijn in het requirements document.

Deze schets (zie voorbeeld hiernaast) heb ik besproken met dhr. Leeuwesteijn. Na het bespreken van deze schets heb ik hem uitgebreid met een registratie scherm (Figuur 22). Er zijn in totaal drie componenten-diagrammen opgesteld, namelijk:

- Inloggen en registreren op de CRB;
- Opgeven P2P-websites en wijzigen gebruikersprofiel;
- Ophalen reputaties van P2P-websites en berekenen en beschikbaar stellen van de totale reputatie.

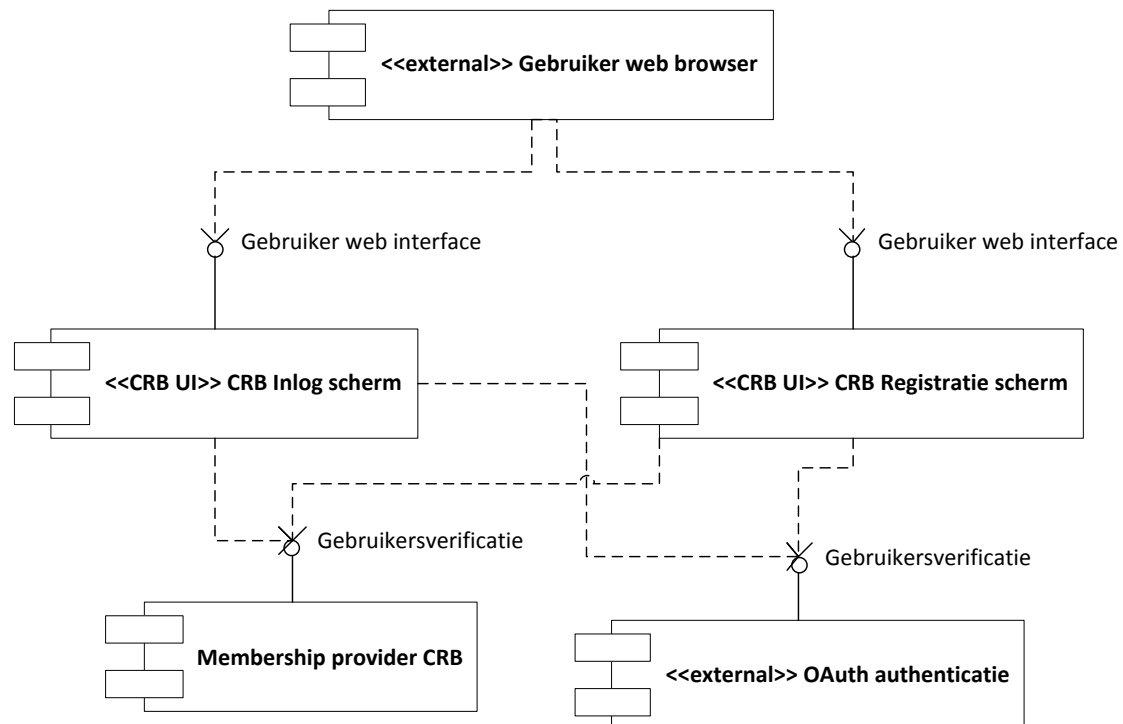


Figuur 21: voorbeeld schets componenten diagram



Figuur 22: 2^e versie component diagram (Inloggen en registreren op de CRB)

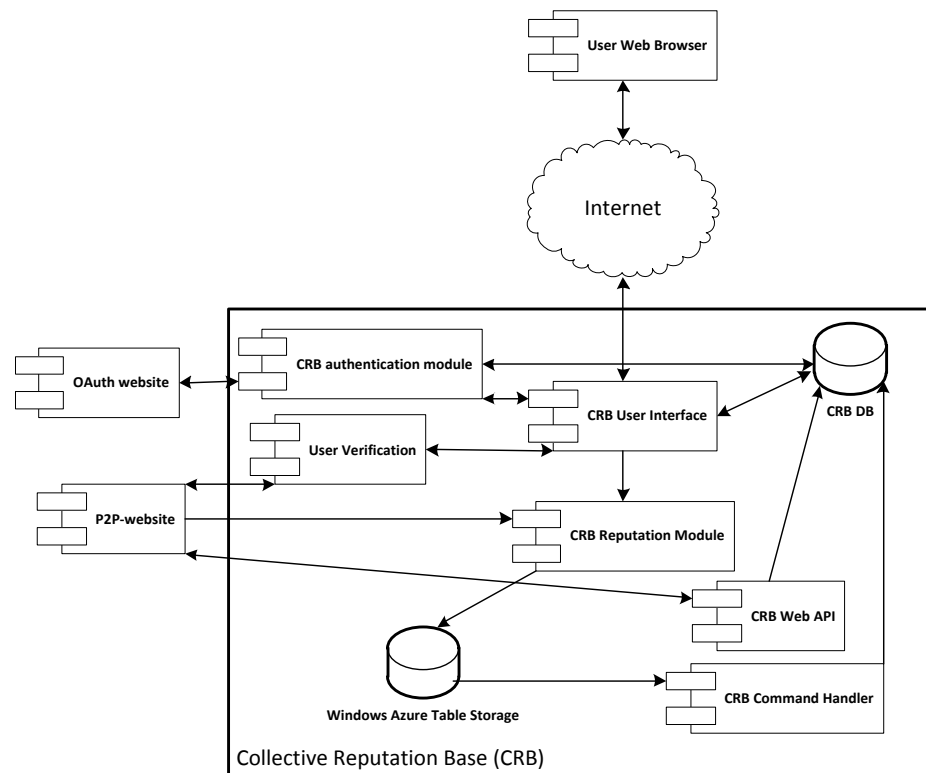
De 2^e versie is ook door de begeleiders beoordeeld en de feedback over Figuur 22 was dat de "CRB DB" geen component kan zijn. Dit is bijgewerkt in de laatste versie van dit diagram. (Figuur 23)



Figuur 23: laatste versie component diagram (Inloggen en registreren op de CRB)

Verder heb ik de functionele elementen vastgelegd aan de hand van de functionele requirements en de opgestelde componentendiagrammen. Voor het functioneel structuur model zijn in totaal twee versie gemaakt. De tweede versie heeft als toevoeging de 'CRB authentication module'.

Daarnaast heb ik de verantwoordelijkheden van deze elementen vastgesteld. Hieronder is de laatste versie van het functioneel structuur model (Figuur 24) te vinden inclusief de beschrijvingen van de verantwoordelijkheden van de elementen. Het volledig overzicht is te vinden in het Architectuurontwerpdokument onder Tabel 4.



Figuur 24: Functioneel structuur model

Voorbeeld elementen en verantwoordelijkheden:

Element naam	Verantwoordelijkheden
CRB User Interface	<ul style="list-style-type: none"> Dit is de userinterface waarmee gebruikers de volgende interacties kunnen uitvoeren: <ul style="list-style-type: none"> Registreren op CRB; Inloggen op CRB; Registreren P2P-websites; Verwijderen P2P-websites; CRB-Profiel bijwerken; CRB reputatie raadplegen; Verwijderen uit CRB.
CRB Reputation Module	<ul style="list-style-type: none"> Ophalen gebruikersreputaties van opgegeven P2P-websites; Opslaan van de reputaties op Windows Azure Table Storage.
OAuth website	<ul style="list-style-type: none"> Dit is een externe website waarmee gebruikers zich kunnen registreren op de CRB. Dit soort authenticatie is beschikbaar via verschillende websites, waaronder: Facebook, Twitter, Live-account en Google.

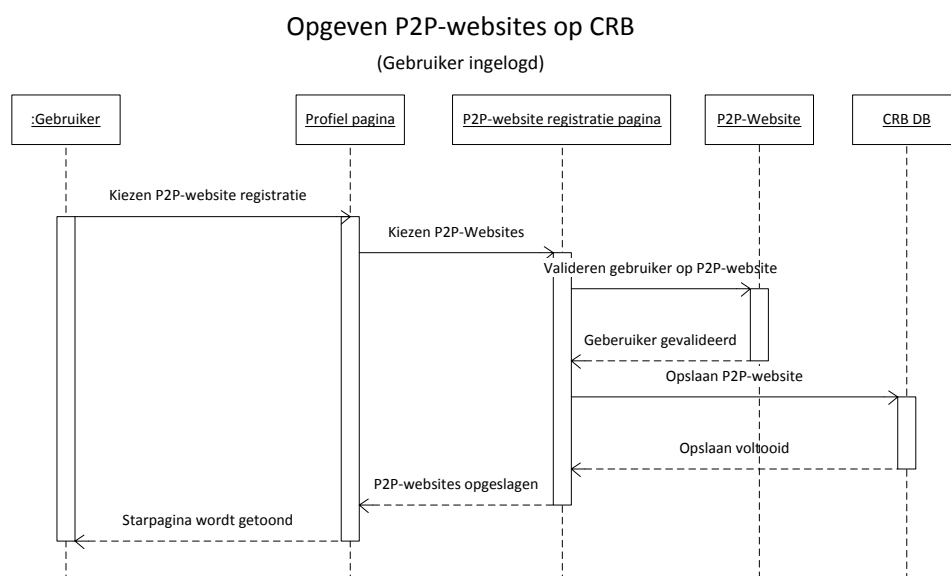
Tabel 8: Voorbeeld functionele elementen en hun verantwoordelijkheden

Als laatste onderdeel van de functional view heb ik enkele functionele scenario's weergegeven met sequentiediagrammen. Met behulp van deze diagrammen wordt vastgesteld hoe verschillende functionele elementen met elkaar integreren via hun interfaces. Enkele sequentiediagrammen die interacties hebben met externe elementen had ik ook in de interactie scenario's vermeld bij Context view.

Voor de volgende functionele scenario's heb ik interactiediagrammen opgesteld:

- Registreren gebruiker op de CRB met OAuth;
- Inloggen gebruiker op de CRB met OAuth;
- Verwijderen gebruiker uit de CRB;
- Opgeven P2P-websites op CRB;
- Verwijderen P2P-websites op CRB;
- Aanpassen gebruikersprofiel op de CRB;
- Raadplegen eigen reputatie op de CRB;

In Figuur 25 is een voorbeeld te vinden van dit functionele scenario's.



Figuur 25: voorbeeld sequentiediagram (Opgeven P2P-websites op CRB)

Als opmerking op de functionele scenario's heb ik meegekregen dat ik bij enkele sequentiediagrammen het woord 'Klikken' had vermeld, en deze vermelding is niet gewenst op deze plek omdat het nog niet bekend is hoe het ontwerp gerealiseerd zal worden. Daardoor heb ik dit woord in de sequentiediagrammen vervangen door het woord 'Kiezen'.

7.4 Het opstellen van de Information viewpoint

In de Information viewpoint wordt beschreven hoe en waar de data wordt opgeslagen. Daarnaast worden de datastroom, informatie eigendom, informatie levenscyclus en de archivering beschreven.

Terugkijkend naar de non-functioneel requirements zien we dat het systeem op een Windows Azure omgeving moet worden gehost.¹⁸ Aan de hand daarvan heb ik naar de mogelijkheden voor dataopslag gezocht op de Azure omgeving. Op de Azure omgeving zijn er twee mogelijkheden, namelijk:

- **SQL Azure database:** dit een SQL database web editie. Deze database functioneert hetzelfde als de SQL server database op de desktop.
- **Windows Azure Table Storage:** deze component is een onderdeel van de Window Azure omgeving. Hierin kunnen grote hoeveelheden gestructureerde data worden opgeslagen. De service is een NoSQL datastore die geverifieerde aanvragen van binnen en buiten de Window Azure cloud accepteert. Windows Azure tabellen zijn ideaal voor het opslaan van gestructureerde, niet-relatieve gegevens.⁽¹¹⁾

Voor het opslaan van de reputaties die opgehaald worden van verschillende P2P-websites hebben we grote opslag nodig als de CRB veel gebruikers heeft die actief zijn op verschillende P2P-websites. Deze verzamelde gegevens worden door een command handler bewerkt en samengevoegd tot een score die op een de CRB database wordt opgeslagen. Daarna zijn deze gegevens niet meer nodig en kunnen verwijderd worden. Dus de levenscycli van deze informatie is niet lang. Daardoor is het handig om daarvoor de Table Storage variant te gebruiken.

Voor de andere zaken zoals: opslaan van gebruikersprofiel, opslaan van de totale reputatie en het registreren van de P2P-websties wordt de SQL Azure gebruikt.

Bij het opstellen van de datamodel heb ik drie modellen gebruikt, namelijk:

- **Systeem data structuur:** hierin heb ik een algemene data structuur laten zien met de entiteiten en de beschrijvingen daarvan. Daarnaast worden ook gedeeltelijk de relaties weergegeven.
- **EER-model (Enhanced Entity-Relationship):** met dit model wordt verder ingegaan op de relaties tussen de entiteiten en de attributen. Verder wordt een duidelijkere weergave van de relaties weergegeven
- **Relationeel Representatie Model:** bij dit model wordt dieper gekeken naar de data types van de attributen. Hierin worden ook de relaties weergegeven.

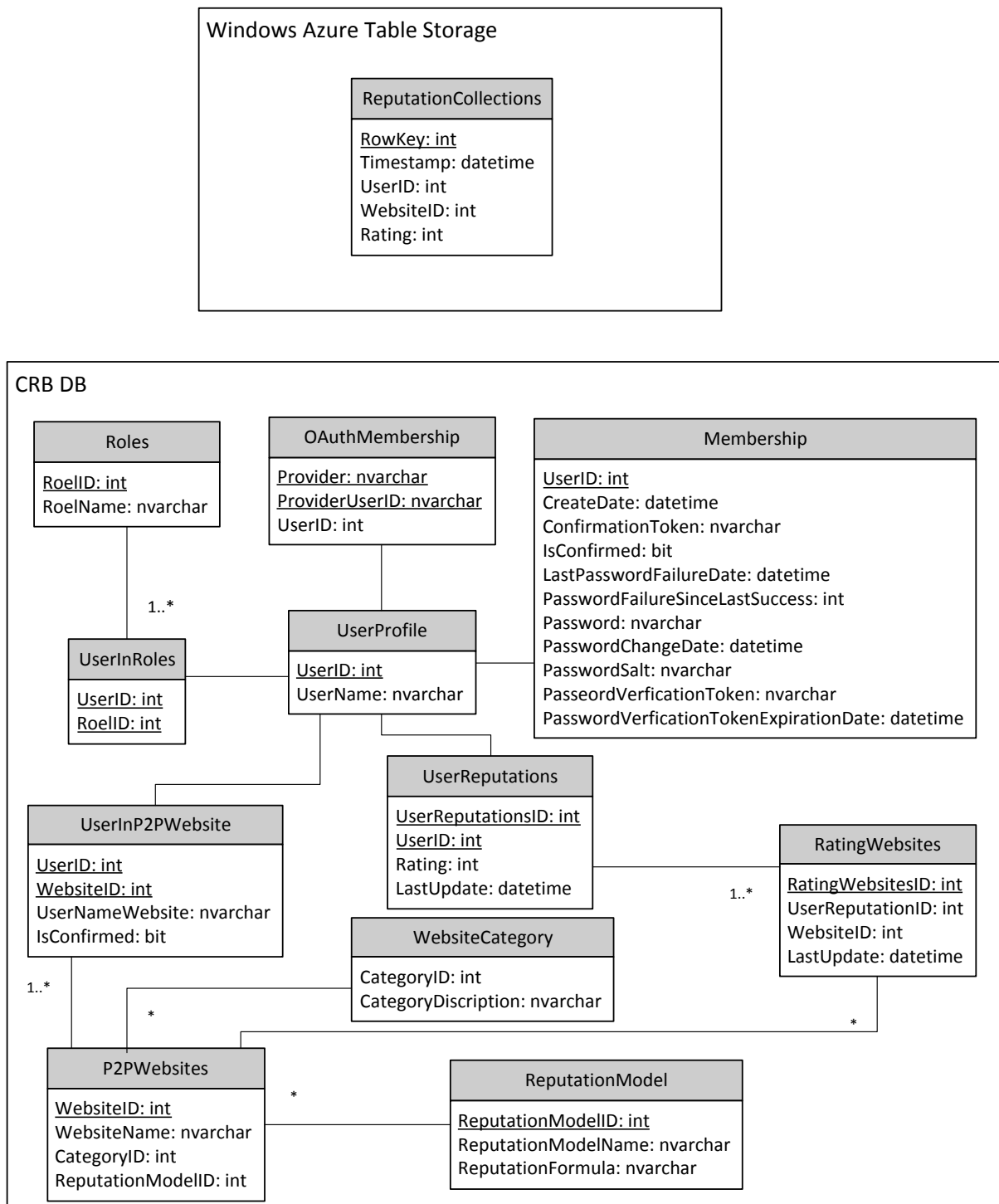
De modellen die weergegeven zijn bij de Information View bestaan uit twee groepen van entiteiten die relaties met elkaar hebben. Deze groepen zijn:

- **Standaard Membership entiteiten:** dit zijn de entiteiten die deel maken uit de Membership Provider van ASP.NET MVC. Deze Provider is een standaard authenticatie service binnen de .NET omgeving. Hierin hebben we de volgende entiteiten: UserProfile, Membership, OAuthMembership, Roles en UserInRoles.
- **CRB entiteiten:** deze entiteiten heb ik opgesteld aan de hand van de gedefinieerde systeembeschrijving, opgestelde requirements en de gesprekken

18 Zie de non-functional requirements nr. NF06 op het requirements document.

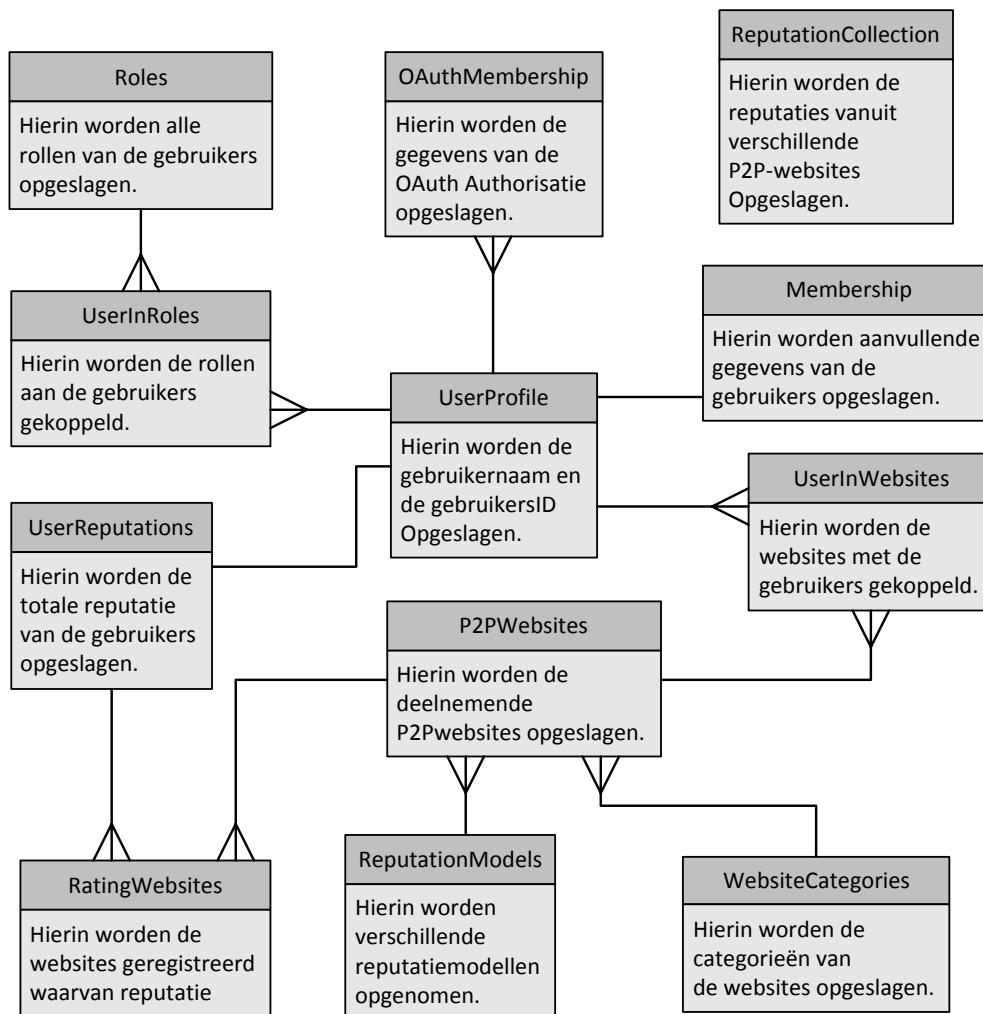
met de opdrachtgever. Deze entiteiten zijn: UserReputations, UserInWebsite, P2PWebsites, RatingWebsites, WebsiteCategories en ReputationModels.

In het begin had ik één model opgesteld voor de Information View, namelijk de Systeem data structuur (Figuur 26). Na het bespreken van dit model met de begeleider was hij van mening dat dit model te veel details bevatte terwijl het een algemeen model moet zijn.



Figuur 26: eerste versie systeem data structuur

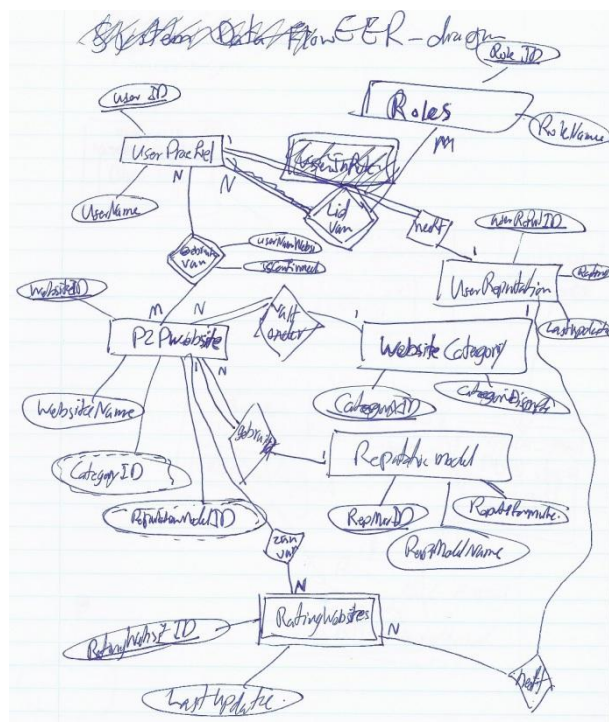
Na het raadplegen van het boek Software Systems Architecture (6), Information View hoofdstuk, heb ik een meer algemene model opgesteld waarbij de entiteiten worden genoemd met bijbehorende beschrijving. (Figuur 27)



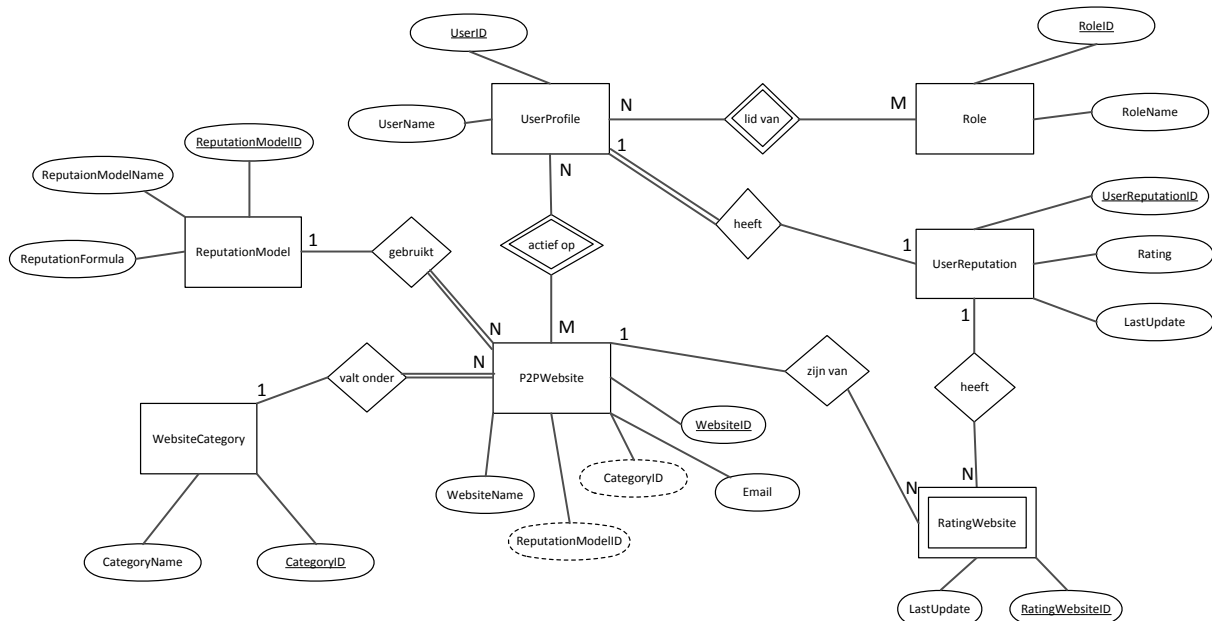
Figuur 27: laatste versie systeem data structuur

Dit model had ik opgestuurd naar de begeleiders en de opdrachtgever en ze waren eens met deze opzet.

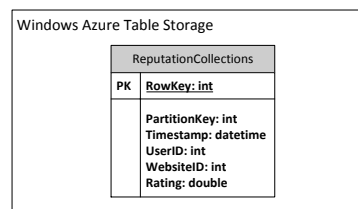
Daarna ben ik begonnen met het opstellen van de EER-model. Daarvoor heb ik het boek Principles van databases (12) gebruikt. Met behulp van de uitleg en de voorbeelden die vermeld zijn in dit boek heb ik een schets opgesteld van het EER-model (Figuur 28). Zoals hierboven is vermeld bevat dit model meer informatie over de entiteiten en de attributen die bij deze entiteiten horen. Daarnaast worden de relaties duidelijker weergegeven. Na het verwerken van de schets is het model ontstaan die in Figuur 29 te zien is. Bij dit model hadden de begeleiders geen opmerkingen. Daardoor is één versie beschikbaar van dit model.



Figuur 28: schets EER-model

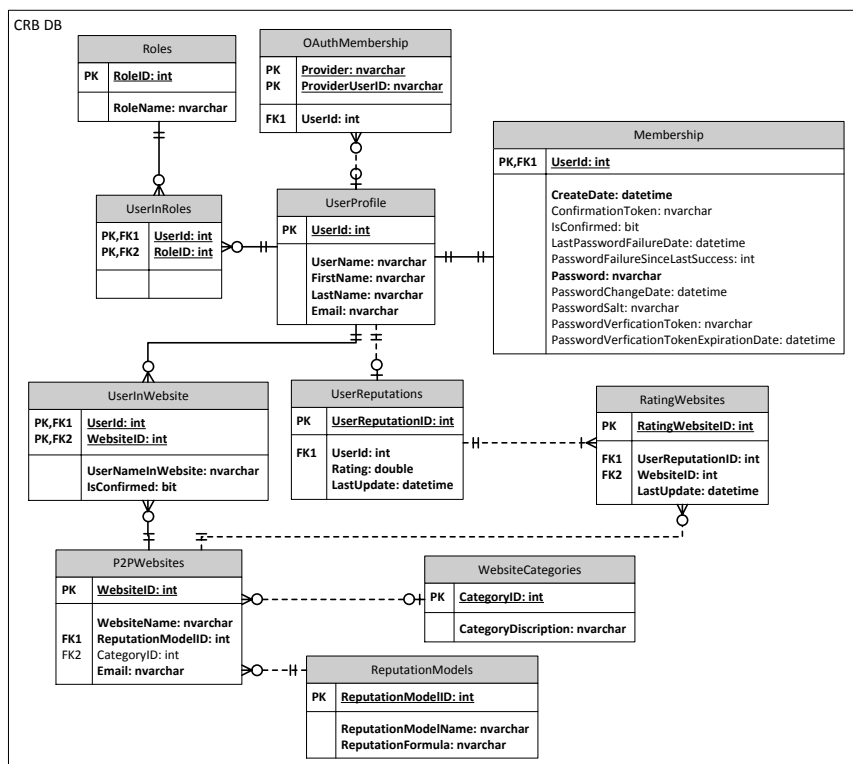


Figuur 29: EER-model van de CRB database



Daarna ben ik bezig geweest met het opstellen van de Relatieve Representatie Model. Als basis heb ik hiervoor het opgestelde model in Figuur 26 gebruikt. Zoals ik hiervoor heb vermeld bestaat het datamodel uit twee groepen.

Voor het opstellen van de Standaard Membership entiteiten heb ik een voorbeeld-project gemaakt in Visual Studio 2012¹⁹. Met behulp van dit project kon ik alle attributen die bij de Membership entiteiten horen noteren. Daarnaast heb ik de CRB entiteiten verder uitgebreid met de bijbehorende attributen die te vinden waren in het EER-model (Figuur 29) en ook in het model van Figuur 26.



Figuur 30: Relatieve Representatie Model van de CRB DB

¹⁹ Dit is een tool van Microsoft waarmee kan worden geprogrammeerd in verschillende programmeertalen, vooral in Microsoft programmeertalen die door het .NET framework wordt ondersteund.

Voor het verduidelijken van het model wil ik het volgende vermelden:

- De onderstreepte attributen zijn de Primary key's van de tabellen;
- De dikgedrukte attributen zijn verplichte attributen.

In deze viewpoint worden de data levenscycli en de data tijdslijnen opgesteld. Deze heb ik in overleg met de opdrachtgever opgesteld. Zie de tabellen hieronder.

Entiteit	Levenscycli
Gebruikersprofiel	Niet actieve gebruikers worden na 2 maanden gewaarschuwd dat zij niet actief zijn en 1 maand na de waarschuwing worden zij verwijderd uit de CRB indien ze nog niet actief zijn.
Reputaties op Table Storage	De opgeslagen gegevens op de Table Storage worden direct verwijderd na de succesvolle verwerking en opslag op de UserReputation tabel met behulp van de Command Handler.
P2PWebsite	Indien er 6 maanden na het registreren van de P2P-website geen activiteit wordt geconstateerd van deze website binnen de CRB (zoals: aangemelde gebruikers die op deze website actieve zijn en reputatie die van uit deze website binnen komt), wordt deze website gewaarschuwd en 2 maanden na de waarschuwing wordt het websiteprofiel verwijderd uit de CRB.

Tabel 9: Data levenscycli

Proces	Tijdslijn
Ophalen gebruikersreputaties van de P2P-websites	Dit proces wordt elke 24 uur uitgevoerd op het moment dat weinig gebruikers op de CRB actief zijn.
Samenvoegen van reputaties en opslaan van de totale reputatie	Dit proces wordt elke 24 uur uitgevoerd op het moment dat weinig gebruikers op de CRB actief zijn.

Tabel 10: Data tijdslijnen

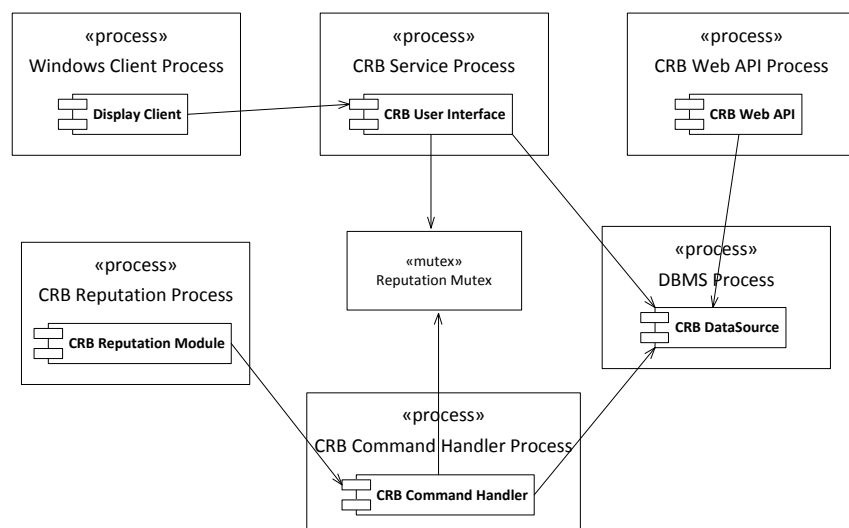
7.5 Het opstellen van de Concurrency viewpoint

Deze viewpoint beschrijft de samenwerking tussen verschillende onderdelen van het systeem en hoe deze samenwerking wordt gecoördineerd.

Bij het opstellen van deze viewpoint heb ik eerst het concurrency model (**Error! Reference source not found.**) opgesteld.

Daarmee zijn de relaties tussen verschillende processen te zien. Als feedback heb ik van de begeleider meegekregen dat het model tekortschiet voor het vervullen van de Concurrency View. Daardoor zou ik nog duidelijke beschrijvingen moeten schrijven over verschillende concurrency aspecten binnen de CRB.

De volgende concurrency aspecten heb ik gedefinieerd bij deze view:



Figuur 31: concurrency model CRB

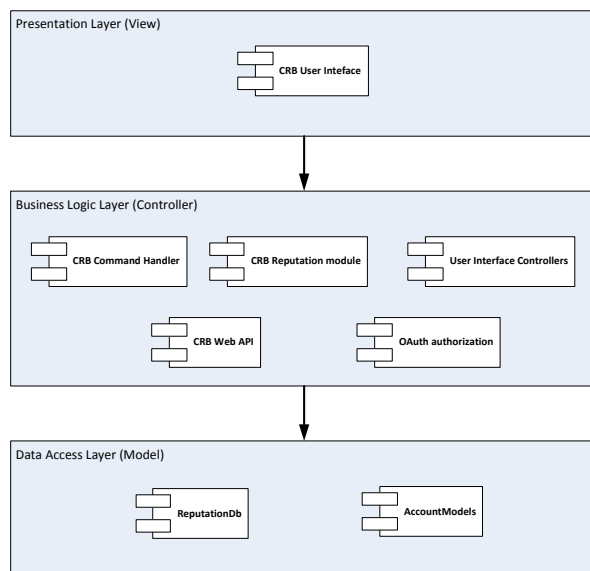
- Een geregistreerde gebruiker kan inloggen en tegelijkertijd kunnen zijn reputaties worden opgehaald met behulp van CRB Reputation Proces;
- Het samenvoegen van de reputaties door de CRB Command Handler Process kan uitgevoerd worden tijdens het inlogproces van de gebruiker;
- Om de reputaties te kunnen samenvoegen dient eerst het ophalen van de reputaties door de CRB Reputaion Process uitgevoerd te zijn. Dus het ophalen en samenvoegen van de reputaties kunnen niet tegelijkertijd worden uitgevoerd;
- Tijdens het samenvoegen van de reputaties door de CRB Command Handler Process zijn de processen CRB Web API Process en het raadplegen van eigen reputatie niet bereikbaar;
- Bij het verwijderen van een account worden alle reputaties en bij behorende gegevens mee verwijderd.

7.6 Het opstellen van de Development viewpoint

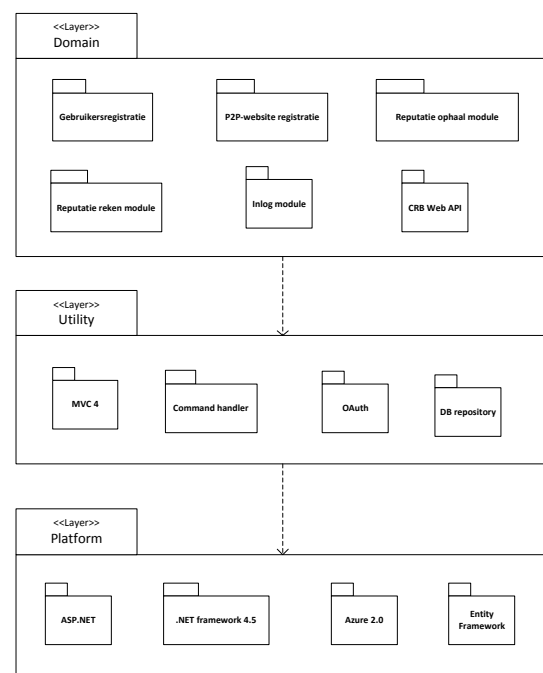
Deze viewpoint heb ik kunnen opbouwen met behulp van de requirements en genoemde functionele elementen in de Functional View.

De eerste versie van de module structuur heb ik ontworpen met behulp van de beschrijvingen in het boek 'Software Systems Architecture' en de non-functional requirements nr. NF05²⁰ (Figuur 32). Dit model was besproken met de begeleider. Hij was niet eens met de verdeling en was van mening dat het in een andere perspectief gezien moet worden, dus niet de verdeling wat ik had gebruikt. Ik heb verder gezocht naar verschillende voorbeelden van de module structuur en heb ik het model bijgewerkt tot de tweede versie(Figuur 33).

Bij het reviewen van dit laatste model heeft de begeleider aangegeven dat de verdeling van de MVC onderdelen (Model, View en Controller) op deze manier niet juist is. Daardoor ben ik verder onderzoek gegaan naar andere bronnen die een module structuur combineren met een MVC-model en



Figuur 33: tweede versie module structuur

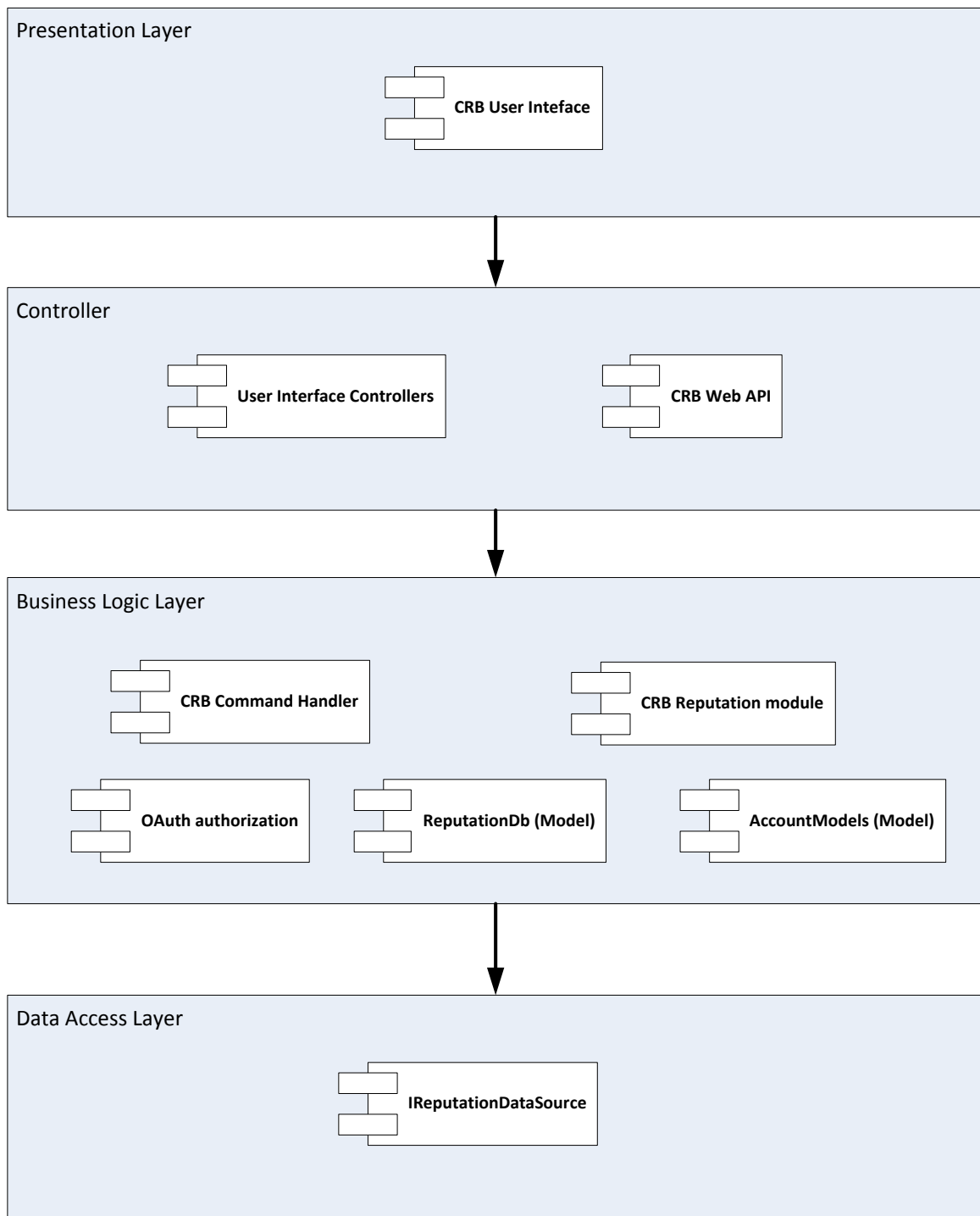


Figuur 32: eerste versie module structuur

uiteindelijk heb ik de laatste versie opgesteld van de module structuur. De nieuwe verdeling van de module structuur houdt in dat de controller geen deel uitmaakt van één van de lagen maar het is een koppeling tussen de Business Logic

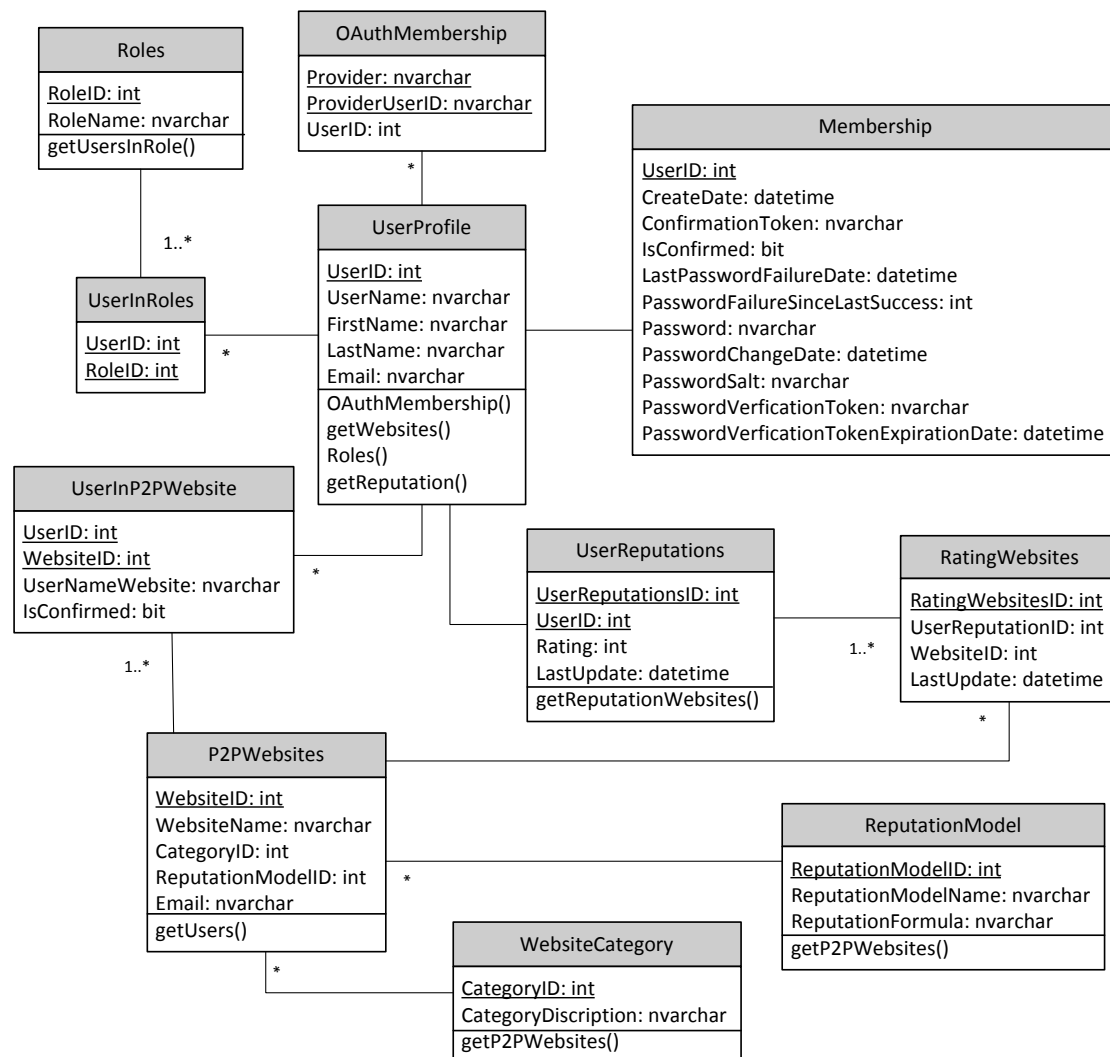
20 Requirements document, Non-functional requirements, nr. NF05.: ASP.NET MVC gebruiken voor het bouwen van het systeem

Layer en de Presentation Layer.(Figuur 34)



Figuur 34: laatste versie van de module structuur

Verder heb ik, met behulp van het RRM-model in de Information View en de beschikbare informatie over de werking van de CRB een klasse diagram opgesteld voor de CRB. In dit diagram zijn verschillende klassen te zien met de relaties tussen deze klassen en bij elk klasse zijn de bijbehorende methoden te zien.



Figuur 35: Klasse diagram CRB

In de Development View is verder over de algemene ontwerp gesproken. Daarvoor zal ik de MVC (Model View Controller) design pattern gebruiken. De ASP.NET MVC is een Microsoft .NET implementatie van deze design pattern. ⁽¹²⁾

Verder heb ik de volgende standaarden en tools genoemd die gebruikt zullen worden bij het bouwen van het PoC:

- Standaarden:
 - ASP.NET MVC 4: dit was een non-functional requirement en een randvoorwaarde van de opdrachtgever.
 - Windows Azure 2.0: dit was een non-functional requirement en een randvoorwaarde van de opdrachtgever.
 - C# .NET-programmeertaal: dit is één van de programmeer talen waarmee in een .NET omgeving kan worden geprogrammeerd. De andere optie was VB .NET, maar met deze taal heb ik geen ervaringen. Daardoor is C# .NET gekozen voor het programmeren van het PoC.
 - .NET framework 4.5: verschillende APS.NET MVC 4 functionaliteiten werken standaard met dit framework, en het was de laatste versie van .Net framework. Bij het gebruiken van de oudere frameworks was een risico dat

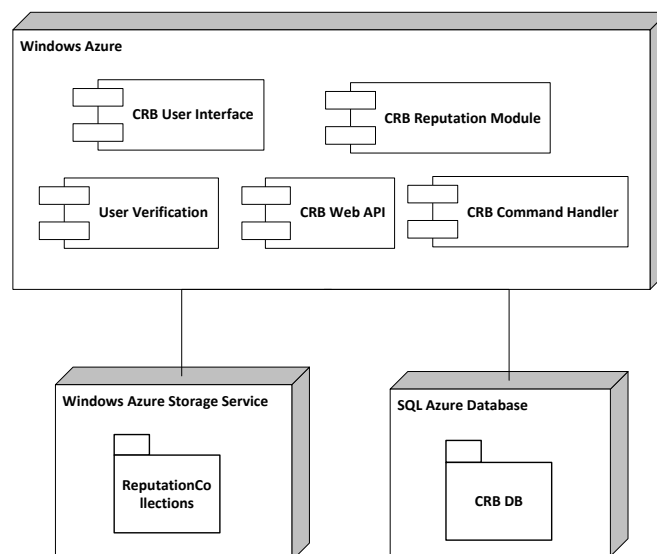
sommige functionaliteiten niet zouden werken en de nieuwe Windows Azure versie (2.0) is ook geoptimaliseerd voor dit framework.

- SQL Azure Database: deze keuze reeds verantwoord in §7.4.
- Windows Azure Table Storage: deze keuze reeds verantwoord in §7.4.
- Tools
 - Visual Studio 2012: om te kunnen programmeren met verschillende .NET programmeertalen en binnen de .NET omgeving, dient met Microsoft Visual Studio tool gewerkt te worden. De laatste versie van deze tool was Visual Studio 2012.
 - Microsoft SQL Server 2012: dit was ook de laatste versie van Microsoft SQL Server 2012. Deze was nodig voor het beheren en implementeren van de CRB database.
 - Azure Storage Explorer: om de Windows Azure Table Storage te kunnen beheren en implementeren op een desktop kan deze tool gebruikt worden. Het is makkelijker om de Table Storage vanaf je desktop te beheren dan via de online user interface van Windows Azure waarin weinig functionaliteiten aanwezig is.
 - manage.windowsazure.com: om de CRB te kunnen hosten op Windows Azure hebben we een online omgeving nodig op windowsazure.com. Bij het aanmelden op windowsazure.com was het mogelijk om zonder kosten voor 3 maanden een account aan te vragen waar alle functionaliteiten van Windows Azure beschikbaar worden gesteld.
 - tfs.visualstudio.com: om de source code te kunnen beheren en bewaren, is deze tool gebruikt. Het is een online Team Foundation Server die afgelopen maanden door Microsoft is geïntroduceerd. Hiervoor was het ook mogelijk om een proef abonnement van 3 maanden aan te vragen.

7.7 Het opstellen van de Deployment viewpoint

Dit viewpoint beschrijft de omgeving waarin het systeem wordt geïnstalleerd/gehost.

In het kader daarvan is een deployment model opgesteld. Dit model heb ik met behulp van de benoemde componenten in de Functional View opgesteld. Bij elke component wordt gekeken waar deze componenten wordt geplaatst. Na het bestuderen van deze componenten is de volgende model bestaan. (Figuur 36)



Figuur 36: Deployment model

Hierbij hoort ook de volgende table:

Functionele elementen	Deployment Node(s)
CRB User Interface	Windows Azure
CRB Reputation Module	Windows Azure
User Verification	Windows Azure
CRB Command Handler	Windows Azure
CRB Web API	Windows Azure
Windows Azure Table Storage (ReputationCollection)	Windows Azure Storage Service
CRB DB	SQL Azure Database

Tabel 11: Functionele elementen vs Deployment Node(s)

Verder heb ik de software afhankelijkheden als volgt vastgesteld in overleg met de begeleiders en de opdrachtgever:

- **Besturingssysteem:** Windows 7 of hoger is nodig om de benodigde tools te kunnen gebruiken, zoals: Visual Studio 2012, Microsoft SQL Server 2012 en Azure Storage Explorer.
- **Platform:** .NET Framework 4.5 om te kunnen programmeren in C#
- Visual C# 2012
- Web Developer Tools 1.0.30710.0
- Windows Azure Tools June 2012 SP1
- Team Explorer for Visual Studio 2012

7.8 Het opstellen van de Operational viewpoint

Bij dit Viewpoint wordt beschreven hoe het systeem wordt bediend, ondersteund en beheerd.

Om het systeem te kunnen installeren heb ik de verschillende onderdelen van het systeem onderscheid en per onderdeel heb ik aangegeven hoe dit onderdeel geïnstalleerd kunnen worden. Hieronder vermeld ik enkele voorbeelden daarvan. De volledige lijst is te vinden in het architectuurontwerp document onder § 5.7.

- **CRB database:** met behulp van de Publisher van de Visual Studio wordt het dataschema geladen op de SQL Azure database op de Windows Azure omgeving. Dit gebeurt door middel van het specificeren van de datasource link van de SQL Azure database in de publishing profile.
- **CRB Web Interface:** dit onderdeel wordt ook met behulp van de Publisher op de Windows Azure omgeving geïnstalleerd. Daarmee worden de benodigde .NET assemblies en de bestanden op de IIS (Internet Information Server) van de Azure omgeving geplaatst.
- **CRB Reputation Module:** deze module wordt als een .NET assembly samen met andere assemblies en bestanden geplaatst op de omgeving waar CRB Web Interface staat.

7.9 Bespreken ontwerp met de opdrachtgever en de begeleiders

Het opgestelde architectuurontwerp is meerdere malen besproken met de begeleiders en de opdrachtgever en hun feedback is gevraagd voor de ingeleverde onderdelen. Aan de hand van de opmerkingen en/of suggesties is het documenten een paar keer bijgewerkt totdat de uiteindelijke versie daarvan is ontstaan.

8. Proof of Concept van CRB

In dit hoofdstuk wordt beschreven hoe de PoC is tot stand gekomen, welke functionaliteiten geïmplementeerd zijn, welke technieken gebruikt zijn, welke voorbereidingen ondernomen zijn, en hoe de sprints zijn uitgevoerd.

8.1 Gevolgde cursussen

Als voorbereiding voor het bouwen van het PoC heb ik enkele cursussen en workshops gevolgd. Deze cursussen waren merendeels over Windows Azure en ASP.NET MVC 4. Ik had wel een algemene ervaring met de MVC maar dan niet met versie 4. Verder had ik niet veel ervaring met Windows Azure. De gevolgde cursussen zijn:

- **MUCA sessie Azure 2.0:** deze sessie was door Logica georganiseerd om de ontwikkelingen rondom de nieuwe Azure 2.0 uit te leggen;
- **Pluralsight:** dit is een online cursussen website met veel cursussen voor developers op verschillende gebieden. Logica heeft sinds ongeveer één jaar een contract met deze website waardoor voor elke Practice enkele abonnementen beschikbaar zijn. Met behulp van deze website heb ik de volgende cursussen gevolgd:
 - Building Applications with ASP.NET MVC 4
 - LINQ Fundamentals
 - Windows Azure Fundamentals
 - ASP.NET MVC 4 Fundamentals
 - Windows Azure Web Sites

Het volgen van deze cursussen was erg waardevol voor het gebruiken en toepassen van verschillende nieuwe technologieën zoals: gebruik van Windows Azure omgeving en bouwen van een ASP.NET MVC 4 applicatie. Ik heb deze cursussen ook als bron gebruikt voor het oplossen van sommige problemen tijdens het bouwen van PoC.

8.2 Scrum Sprints

Zoals in het Plan van aanpak is vermeld, heeft dit project 3 sprints, namelijk:

- Sprint 1 (Bouwen CRB PoC – Website CRB + basis functionaliteiten);
- Sprint 2 (Bouwen CRB PoC – verzamelen gegevens uit verschillende websites);
- Sprint 3 (Bouwen CRB PoC – afhandelen verzamelde gegevens en bieden van de totale reputatie).

Vóór het beginnen met een sprint had ik een bijeenkomst georganiseerd met de Product Owner (PO – de opdrachtgever), om de Definition of Done (DoD) te bespreken. Tijdens de bijeenkomst werd gekeken bij desbetreffende sprint welke requirements worden uitgevoerd en de requirements met succes konden worden opgeleverd na het aflopen van de sprint periode. Aan het einde van elke sprint worden de uitgevoerde werkzaamheden beoordeeld en geëvalueerd door de PO.

Het was afgesproken met de opdrachtgever dat in eerste instantie de Must-have requirements worden uitgevoerd en indien tijd overblijft wordt gekeken naar de Should-have's.

Hieronder wordt toegelicht hoe deze sprints zijn uitgevoerd.

8.2.1 Sprint 1 (07-12-2012 t/m 19-12-2012)

Tijdens het gesprek met de opdrachtgever over de 1^e sprint was afgesproken dat de volgende requirements zullen worden geïmplementeerd:

- F01: Registeren op CRB;
- F02: Inloggen op CRB;
- F03: Verwijderen CRB account;
- F04: Opgeven P2P-websites;
- F05: Verwijderen P2P-websites;
- F06: Aanpassen profiel.

De afspraak was dat eerst de requirements met "M" als prioriteit zullen worden geïmplementeerd, daarna requirements met "S" en als er tijd over blijft voor de sprint de requirements met "C" worden geïmplementeerd.

Ik had ook de opdrachtgever over enkele onderwerpen gevraagd om de DoD van de requirements te kunnen vaststellen. Deze onderwerpen en de antwoorden van de opdrachtgever kunnen worden samengevat in de volgende punten:

- Bij de registratie worden twee soorten authenticaties gebruikt, namelijk: klassieke authenticatie²¹, OAuth authenticatie²² (keuzen van de opdrachtgever voor deze service was: Facebook, Twitter en Google. Hij was van mening dat er bijna geen persoon is die niet actief is op één van deze websites.);
- Het registreren van de P2P-websites wordt uitgevoerd door middel van registratie door de CRB-gebruiker deze geeft zijn/haar gebruikersnaam aan de desbetreffende P2P-website;
- Bij het aanmaken van het gebruikersprofiel dient de gebruiker zijn voornaam, achternaam en email registreren;
- De datamodel van de CRB had ik uitgelegd en ik heb aangegeven welke gegevens worden opgeslagen en wat de relaties zijn tussen de verschillende tabellen.

Na het afronden van het gesprek met de opdrachtgever was ik gelijk aan de slag gegaan met voorbereidingen en uitvoeren van enkele taken die moesten worden uitgevoerd voordat de requirements kunnen worden geïmplementeerd. Hieronder zijn deze basistaken en de verrichte taken bij de implementatie van de requirements te vinden.

Vastleggen van verschillende onderdelen van de applicatie

Zoals het in het architectuur is vermeld heeft deze applicatie verschillende onderdelen. Deze onderdelen zijn: MVC applicatie, CRB Command Handler, CRB Reputation Module en de CRB database.

Om deze onderdelen te implementeren heb ik voor elk onderdeel een project aangemaakt:

- **ReputationBase.Web:** dit is de MVC web-applicatie en de User Interface waarin de verschillende functionaliteiten worden geïmplementeerd. Daarnaast is dit project een centrale punt waarmee de andere onderdelen met elkaar worden verbonden.

21 Dit is de standaard Membership Provider die binnen MVC wordt gebruikt. Het is een authenticatie service die beschikt over enkele basis authenticatiefuncties zoals: direct registratie met gebruikersnaam en wachtwoord, inloggen op de applicatie. Daarnaast wordt de beveiliging binnen de applicatie verzorgd door deze Provider door het toekennen van verschillende autorisaties aan de onderdelen van de applicatie.

22 Het is een authenticatie service die de gebruikers de mogelijkheid biedt om via een andere websites te kunnen registreren en inloggen op desbetreffende applicatie. Enkele belangrijke spelers die deze service bieden zijn: Facebook, Twitter, Google en Microsoft (Live-account).

- **ReputationBase.Domain:** in dit project worden de database entiteiten gedefinieerd. Met behulp van deze definities wordt de database geïmplementeerd binnen de SQL Server.
- **ReputationBase.RM:** dit is het Reputation Module project waarmee de verschillende reputaties van de gebruikers van verschillende P2P-websites worden opgehaald, omgezet in 5 sterren-vorm en opgeslagen in de Table Storage.
- **ReputationBase.CH:** dit is het Command Handler project waarmee de totale (gemiddelde) reputaties van de gebruikers worden berekend op basis van de gegevens van de gebruiker die in de Table Storage staat. Deze totale reputatie wordt opgeslagen in de CRB database en daarna worden de reputaties op de Tabel Storage die samengevoegd zijn verwijderd.

De reden dat ik deze verdeling heb gemaakt is dat verschillende onderdelen makkelijk te benaderen zijn en het is belangrijk om een duidelijke verdeling te maken voor de onderhoudbaarheid en de overdraagbaarheid van het project.

Bij het onderhouden van de database is deze verdeling een voordeel, dit voordeel wordt verder in dit hoofdstuk verantwoord.

Verbinden met de versiebeheer tool TFS van de Microsoft²³

Na het aanmaken van de basisproject heb ik verbinding gemaakt met eigen online TFS (<https://crb.visualstudio.com>). Na het verbinden met de TFS heb ik de gehele Solution²⁴ ingecheckt op de server.

Het implementeren van het datamodel

Zoals hierboven is vermeld is een apart project aangemaakt voor de datamodel (ReputationBase.Domain).

Het datamodel heeft twee onderdelen, namelijk:

- **Membership Provider entiteiten:** dit zijn de standaard tabellen die bij deze provider horen. De tabellen zijn: UserProfile, Roles, UserInRoles, Membership, OAuthMembership.
- **CRB entiteiten:** dit zijn de tabellen die nodig zijn voor de implementatie van de web applicatie van CRB. De tabellen zijn: P2PWebsites, UserInWebsite, UserReputations, RatingWebsites, WebsiteCategories, ReputationModels en ReputationCollections.

De Membership Provider tabellen worden standaard door de MVC applicatie gemaakt. Om mijn datamodel te kunnen implementeren moest ik de CRB tabellen aanmaken. Bij het aanmaken van de tabellen heb ik voor elk tabel een class aangemaakt waarin de definities van de entiteit worden vermeld en de attributen worden gedefinieerd. Indien sprake is van een relatie wordt de relaties ook gedefinieerd. Voor het definiëren van de entiteiten, attributen en de relaties heb ik de RRM-model gebruikt die in het architectuur was vermeld (Figuur 30).

Bijvoorbeeld de definitie van P2PWebsites en WebsiteCategories ziet er als volgt uit:

```
public class P2PWebsite
{
    public int Id { get; set; }
    public string Name { get; set; }
    public int CategoryId { get; set; }
    public int ReputationModelId { get; set; }
}
```

²³ Deze tool is benoemd in §7.6. TFS staat voor Team Foundation Server. Deze tool is een online tool die te vinden is op tfs.visualstudio.com.

²⁴ Solution is een projecten verzameling die binnen de Visual Studio kunnen worden aangemaakt.

```
public class WebsiteCategory
{
    public int Id { get; set; }
    public string Name { get; set; }
    public ICollection<P2PWebsite> P2PWebsites { get; set; }
}
```

Na het definiëren van alle entiteiten (behalve ReputationCollections)²⁵ is een class aangemaakt voor het definiëren van een interface waar alle entiteiten in staan. Met behulp van deze interface wordt het mogelijk om alle entiteiten via een centrale punten te benaderen en query's naar de database te sturen.

Verder heb ik in het project 'ReputationBase.Web' een referentie geplaatst van het project 'ReputationBase.Domain' en een class gemaakt voor het definiëren van de datamodel in de Web applicatie.

Na het verbinden van de model heb ik de database Migratie geactiveerd binnen de Web applicatie. Database migratie aanzetten binnen een project houdt in dat bij wijzigingen binnen de datamodel de database kan worden geüpdate. Bij het uitvoeren van de update wordt rekening gehouden met de data en de relaties tussen de tabellen zodat er later geen fouten ontstaan. Naast het updaten van de database bij database Migratie kunnen we sommige basisdata invoeren die in de database worden gezet, in geval van de CRB zou het om de volgende gegevens gaan: gebruikersnamen (b.v. Admin), rollen (administrator rol), website categorieën en website reputatie modellen.

Uiteindelijk was het tijd om de twee onderdelen van de database (Membership Provider entiteiten en CRB entiteiten) met elkaar te samenvoegen. De sleutel entiteit die beiden onderdelen kan verbinden was de UserProfile entiteit. Bij UserProfile hebben we de UserId attribueert die bij verschillende tabellen wordt gebruikt op de beide onderdelen. Om dit te bereiken heb ik de UserProfile entiteit opnieuw gedefinieerd binnen het 'ReputationBase.Domain' project en verder in de DataSource opgenomen en in de database definitie op het 'ReputationBase.Web' project. Met deze actie kon ik de beide onderdelen van de database samenvoegen en één database van maken. Uiteraard moest ik nog enkele configuraties wijzigen binnen het 'ReputationBase.Web' project, om dit laatste te kunnen bereiken heb ik de cursus 'Building Applications with ASP.NET MVC 4' geraadpleegd op de Pluralsight.com. Deze koppeling was niet eenvoudig en heeft veel tijd gekost.

Als laatste onderdeel van de datamodel heb ik de 'ReputationCollections' tabel aangemaakt met behulp van de tool 'Azure Storage Explorer'.

Aanmaken Windows Azure account

Om het PoC op een Windows Azure omgeving te kunnen hosten had ik een Windows Azure account nodig. Dit was te regelen via windowsazure.com website. Bij het aanmelden met een Live-account krijgt men de mogelijkheid om 3 maanden gebruikt te kunnen maken van de diensten van Windows Azure onder een proefabonnement. Dit was een voldoende tijd voor mij voor het bouwen en implementeren van de PoC van de CRB.

Aanpassen van de basis onderdelen van MVC web applicatie

Bij het aanmaken van de MVC web applicatie moest ik kiezen uit verschillende Project Templates. De beschikbare keuzes waren: Empty, Basic, Inter Application, Intranet Application, Mobile Application en Web API. Mijn keuze was Internet Application. Deze keuze was door de beschikbaarheid van de Membership Provider. Dit laatste heb ik nodig voor het implementeren van de klassieke authenticatie binnen de CRB.

Na het aanmaken van het project heb ik de naam van het project veranderd op verschillende onderdelen en configuraties van de Web Applicatie. Bijvoorbeeld op de hoofdpagina stond een standaard tekst over MVC Web applicatie deze heb ik vervangen

²⁵ Deze tabel wordt in Windows Azure Table Storage gemaakt.

met een afbeelding die gerelateerd is aan de CRB en verder is de benaming op de logo en de copyright gedeeld bijgewerkt tot de CRB tekst. Verder zijn er enkele vertalingen gemaakt voor onderdelen van de Membership Provider die in het engels waren.

Na het afronden van al deze voorbereidingen ben ik begonnen met het vertalen van de beoogde requirements voor Sprint 1.

De requirements F01 en F02 die over het registreren en inloggen op de CRB gaan hadden twee onderdelen. Het eerste onderdeel genaamd de klassieke authenticatie²⁶ was geïmplementeerd tijdens de voorbereidingen.

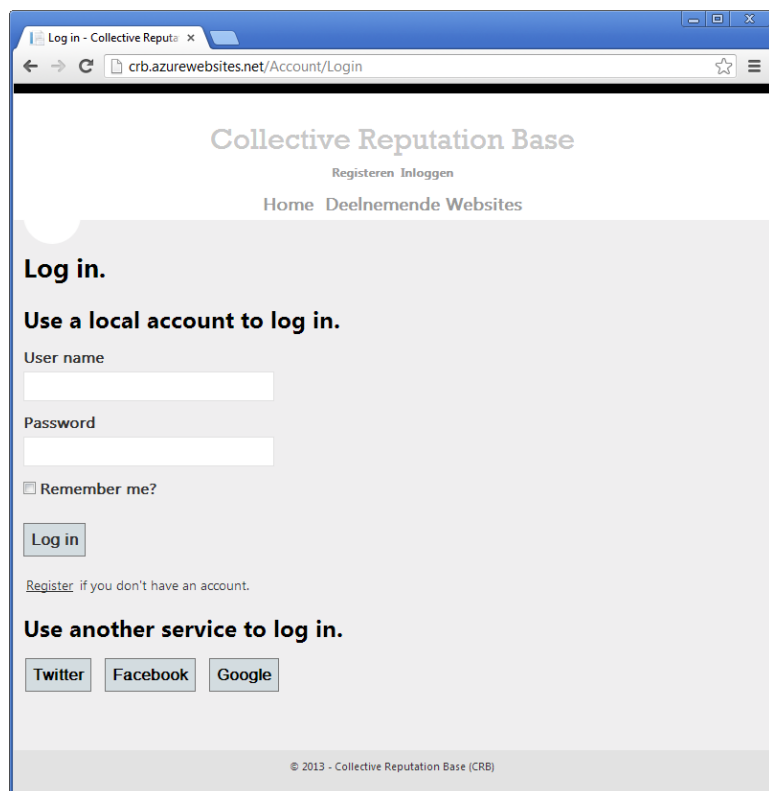
Het tweede onderdeel ging over de OAuth authenticatie. Zoals we²⁷ hadden afgesproken bij DoD van dit requirement geldt dat bij dit onderdeel de volgende drie websites worden gebruikt: Facebook, Twitter en Google.

Om het functionaliteit mogelijk te maken binnen de CRB dient eerst aangemeld te worden op de developers website van deze websites. Ik heb me aangemeld bij Facebook developers website en bij Twitter developers website. Voor het gebruik van de Google-account bij authenticatie hoeft er geen aanmelding plaats vinden bij de developers website van Google.

Na het invoeren van de benodigde gegevens op Facebook en Twitter kreeg ik een ID en een key voor mijn service op deze websites. Met het invoeren van deze gegevens op mijn Web Applicatie kon ik deze service inschakelen.

Het inschakelen van deze service was eenvoudig, zeker door de cursus die ik heb gevolgd op Pluralsight.com. In de Web Applicatie staat een configuratiebestand waar deze codes kwijt kan en de code die nodig was voor deze functionaliteit was als commentaar gezet bij het weghalen van de commentaar en invoeren van de codes worden deze functionaliteiten geactiveerd.

Alle bovengenoemde acties worden in de database opgeslagen. Bij de registratie door de klassieke authenticatie worden de gebruikersnaam en het wachtwoord opgeslagen in 'UserProfile' en 'Membership' tabellen. En bij de OAuth authenticatie worden de gebruikersnaam en de OAuth-website gegevens van desbetreffende persoon opgeslagen. Deze worden opgeslagen in 'UserProfile' en 'OAuthMembershop' tabellen. Het is mogelijk dat een persoon beide authenticatie methoden gebruikt en daarmee wordt zijn lokale account gekoppeld aan zijn OAuth authenticatie en worden alle drie genoemde tabellen gebruikt.



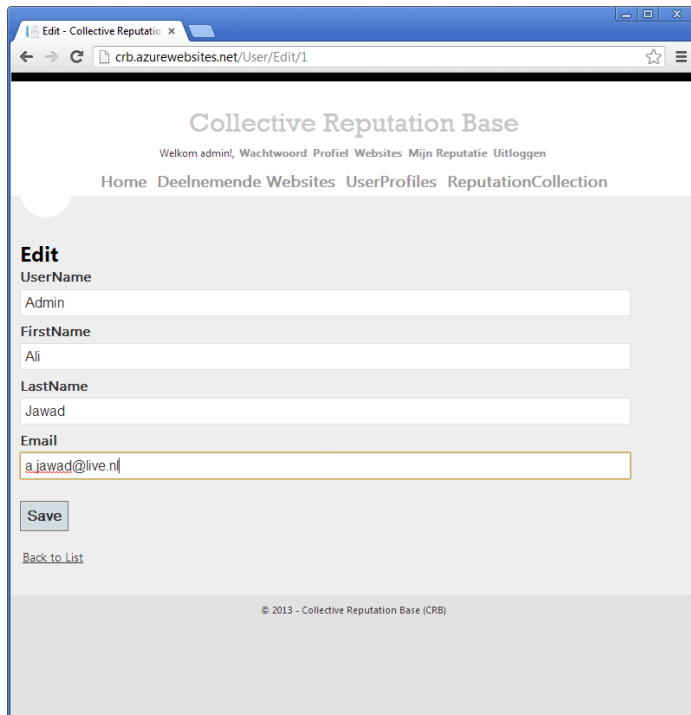
Figuur 37: inlog pagina met OAuth mogelijkheden

²⁶ Membership Provider.

²⁷ Ik zelf en de opdrachtgever.

Met het activeren van de OAuth authenticatie en het implementeren van de klassieke authenticatie heb ik deze requirements uitgevoerd.

Bij het implementeren van de requirement F04 (Opgeven P2P-website) had ik moeite met het koppelen van websites aan de gebruikers. Om deze requirement te kunnen uitvoeren moest ik via de user interface de 'P2PWebsite' tabel koppelen met de 'UserProfile' tabel met behulp van 'UserInWebsite' tabel. Deze koppeling was niet eenvoudig. Daardoor kwam ik in tijdsnood om het in Sprint 1 af te ronden. In overleg met de opdrachtgever heb ik besloten om dit onderdeel in Sprint 2 alsnog uit te voeren. De requirement F05 was afhankelijk van requirement F04, daardoor is deze ook verplaatst naar Sprint 2. Het tot stand komen van deze twee requirements wordt ook in het onderdeel van Sprint 2 verantwoord.



Figuur 38: profiel pagina

De requirement F06 had ik ook tijdens Sprint 1 uitgevoerd omdat het te regelen was met een eenvoudige formulier voor het aanpassen van de gebruikersprofiel.

De requirement F03 is door tijdsnood uitgevallen van Sprint 1.

Tijdens Sprint 1 was veel tijd besteed aan de implementeren van het datamodel. Tijdens de implementatie waren enkele problemen die veel tijd hebben gekost. Zoals bij het koppelen van de twee onderdelen van de database met behulp van 'UserProfiel' tabel. Zoals ik hiervoor heb gemeld was de koppeling niet eenvoudig en moest ik op verschillende onderdelen puzzelen om dit op te lossen. De moeilijkheid die hierbij ontstond was dat de standaard tabellen die gemaakt zijn door de MVC applicatie samenvoegd moesten worden met de tabellen die

ik zelf heb gemaakt en dit had als risico dat de functionaliteiten die verboden zijn aan deze standaard tabellen niet meer werkbaar kunnen zijn. Daardoor moest ik alle kanten goed bekijken en goed testen zodat ik in latere stadia geen niet tegen problemen zou lopen.

Ik had ook veel tijd besteed aan het koppelen van het 'ReputationBase.Domain' project aan het 'ReputationBase.Web' project. De moeilijkheid hierbij had ook te maken met de Membership Provider. Voor deze provider is een standaard datamodel gemaakt in de MVC applicatie en om mijn datamodel in te werken moest ik veel puzzelen vooral bij het database migratie gedeelte zodat ik altijd mijn gehele model up-to-date kan houden.

Na het afronden van deze werkzaamheden heb ik een gesprek georganiseerd met de opdrachtgever voor het bespreken en evalueren van de uitgevoerde werkzaamheden. Hij was tevreden met de resultaten die behaald zijn binnen deze periode en we hebben verder over Sprint 2 gesproken.

8.2.2 Sprint 2 (20-12-2012 t/m 02-01-2013)

Tijdens het evaluatiegesprek van Sprint 1 met de opdrachtgever hebben we de requirements voor Sprint 2 gekozen. De gekozen requirements waren:

- F04: Opgeven P2P-websites (uit Sprint 1);

- F05: Verwijderen P2P-websites (uit Sprint 1);
- F08: Periodiek ophalen van de reputatie van de P2P-websites;
- F10: De verzamelde reputatiegegevens op een database opslaan.

Voor het vaststellen van de DoD van deze requirements zijn de volgende punten besproken:

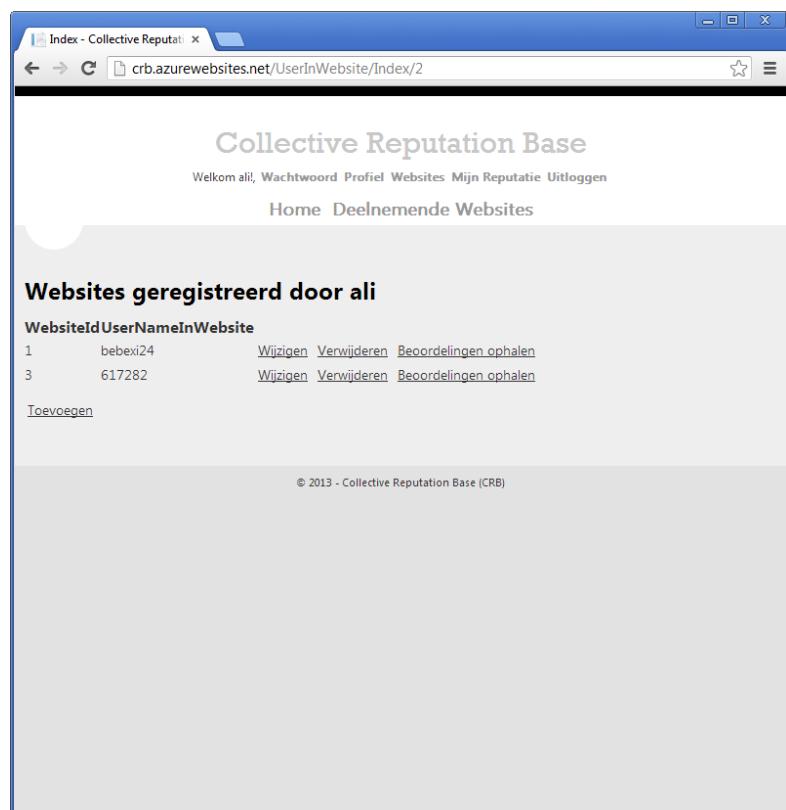
- Tijdens het uitvoeren van de enquête konden we helaas geen P2P-website vinden waarmee we de pilotfase kunnen doen. Daarom is in overleg met de opdrachtgever besloten om de Html scan oplossing²⁸ te implementeren voor het verzamelen van willekeurige reputatiegegevens en deze reputaties met elkaar samen te voegen om de werking van het CRB PoC te kunnen laten zien.
- We hebben ook mijn keuzes van de websites waarop de Html scan zal worden uitgevoerd, besproken. De twee websites waren: ebay.nl en tweedehands.nl. De reden dat ik deze twee websites heb gekozen is dat we verschillende vormen van de reputatie nodig hebben om te kunnen laten zien hoe deze verschillende vormen samengevoegd kunnen worden. Ebay.nl had verschillende vormen beschikbaar waarvan ik de percentage-vorm wilde gebruiken en op tweedehands.nl was de Positief/Negatief-vorm aanwezig. Beide websites voldoen ook aan de eisen die gesteld zijn voor de websites waarop Html scan kan worden uitgevoerd. (§6.6)

Eerst ben ik begonnen met de requirements F04 en F05. Om te kunnen beginnen met deze requirements heb ik enkele websites toegevoegd aan de database om mee te kunnen werken als basis voor deze requirements. De toegevoegde websites waren: Ebay.nl, Speuders.nl, Tweedehands.nl.

Na lang zoeken naar een oplossing voor deze requirements heb ik ook via een cursus op de Pluralsight.com een oplossing kunnen vinden.

Binnen de MVC het is eenvoudig om een view te maken van een tabel in de database. Maar bij requirements F04 hebben we een combinatie van drie tabellen die benaderd worden, namelijk: UserProfile, P2PWebsites en UserInWebsites.

Om een view te kunnen maken van de UserInWebsites-tabel met behulp van de UserProfile-tabel en de P2PWebsites-tabel heb ik een gecombineerde view gemaakt waarbij van de gebruiker een view wordt gemaakt (UserProfile-tabel) en aan de hand daarvan een query wordt uitgevoerd op de UserInWebsites-tabel. Hierbij kunnen de benodigde gegevens van



Figuur 39: websites registratie pagina

²⁸ Kijk bij de communicatie mogelijkheden zonder medewerking P2P-websites §6.6.

desbetreffende gebruiker worden toegevoegd, opgehaald, bewerkt en verwijderd. Daarmee zijn deze twee requirements opgelost.

Voor het uitvoeren van de requirement F08 heb ik de test Console Applicatie (6.6) gebruikt als basis voor het implementeren van deze functionaliteit. Ik heb deze applicatie omgezet in een C# class die binnen het project 'ReputationBase.RM' was geplaatst. De werking van dit onderdeel was als volgt:

De gebruiker voert zijn gegevens in het gedeelte die bij requirement F04 was uitgevoerd. Bij de invoer van de gebruiker worden de volgende gegevens ingevoerd: GebruikersId²⁹, WebsiteId en Gebruikersnaam in de P2P-website.

Na het invoeren van de website en de gebruikersnaam wordt men naar een pagina doorverwezen waarin hij zijn invoer kan inzien (Figuur 39).

Om de werking van deze requirement te laten zien heb ik een link geplaatst bij elk ingevoerde website (Beoordelingen ophalen). Met behulp van deze link worden de beoordelingen van desbetreffende website opgehaald en opgeslagen in de database in ReputationCollections-tabel.

Bij het klikken op de link 'Beoordelingen ophalen' worden de websiteID en de gebruikersnaam naar 'ReputationBase.RM' gestuurd en daarin wordt de link waar de beoordelingen staan van de gebruiker opgesteld en wordt de Html pagina opgehaald en in een Data Stream³⁰ omgezet. Met behulp van de HtmlAgilityPack³¹ wordt een analyse uitgevoerd op de html code waarmee de benodigde beoordeling wordt gefilterd. Elke website heeft een eigen filter en eigen model. Bijvoorbeeld bij Tweedehands.nl worden de afbeeldingen gefilterd waar de titel daarvan 'Positief' of 'Negatief' is, deze gefilterde afbeeldingen worden met elkaar geteld en met behulp van de formule die opgesteld is voor deze website wordt de omgezette score naar de 5 sterren-model teruggegeven.

Na het berekenen van de score wordt deze score opgeslagen in de database (requirements F10). Hierbij wil ik iets melden over het gebruik van de Windows Azure Table Storage. Het is mij helaas niet gelukt om met de Table Storage te kunnen verbinden en de verzamelde data daarin op te slaan. Daardoor is in overleg met de begeleiders en opdrachtgever voor dit onderdeel een tabel aan de database toegevoegd met de naam 'ReputationCollections'. Deze tabel is zonder moeite geïmplementeerd door middel van het gebruiken van de database migratie die ik eerder heb toegelicht.

Bij een succesvolle actie wordt een pagina getoond met de berekende score.



Figuur 40: voorbeeld opgeslagen score

29 Door een tekort aan tijd was het niet mogelijk om in plaats van de Id's de website namen te kunnen gebruiken. Bij de gebruiker dient ook zijn Id te worden ingevoerd.

30 Een Data Stream is een bestand die geladen wordt in het geheugen en wordt beschikbaar gesteld voor verwerkingen binnen een applicatie. Na het aflopen van de benodigde verwerkingen wordt deze Data Stream verwijderd van het geheugen.

31 Dit is een Html analyzer die een document object model maakt voor het lezen/schrijven van de html. (18)

Na het afronden van deze functionaliteiten heb ik de resultaten besproken met de opdrachtgever. Hij had als opmerking dat bij het invoeren van de Website en de gebruikersnaam het niet handig is om een WebsiteId in te voeren in plaats van het kiezen van de Website op basis van website naam. Mijn antwoord was dat deze manier wel mogelijk is, maar op dit moment door tijdsgebrek het niet mogelijk is om uit te voeren. Bij het uitbreiden van het PoC zou het altijd mogelijk zijn. Verder had hij geen opmerkingen en tot zo ver tevreden met het opgeleverde werk.

8.2.3 Sprint 3 (03-01-2013 t/m 10-01-2013)

Dit was de laatste Sprint binnen het project. Bij deze Sprint is met de opdrachtgever afgesproken dat twee requirements zullen worden uitgevoerd, namelijk:

- F11: Samenvoegen van de reputaties;
- F12: Raadplegen eigen reputatie.

Om deze requirements uit te voeren hebben we het volgende afgesproken:

- Het berekenen van de totale reputatie (samenvoegen) en verwijderen van de deel reputaties (verzamelde reputaties bij 'ReputationCollections') gebeurt door het klikken op een link die te vinden is bij de gebruikersgegevens pagina, die alleen voor de administrators beschikbaar is;
- Tonen van eigen reputatie gebeurt door middel van de knop 'Mijn Reputatie' waar de gebruiker op kan klikken.

Voor het uitvoeren van requirement F11 heb ik de Command Handler geïmplementeerd. De Command Handler is gebouwd in het project 'ReputationBase.CH'. Zoals ik in de bovenstaande afspraken heb geschreven wordt de functie van de Command Handler uitgevoerd met behulp van een link op de gebruikersbeheer pagina. Deze link is aangemaakt om de werking van de Command Handler te kunnen laten zien. Bij het klikken op deze link wordt een query uitgevoerd op de 'ReputationCollection'-tabel waarbij de verzamelde reputaties van desbetreffende gebruiker worden opgehaald. Deze reputatie collectie wordt naar de 'ReputationBase.CH' gestuurd voor het berekenen van de totale (gemiddelde) reputatie van deze persoon. Na deze berekening wordt de totale reputatie naar de 'ReputationBase.Web' terug gestuurd waarbij deze reputatie op de 'UserReputations'-tabel wordt opgeslagen. Als resultaat van succesvolle uitvoer wordt een pagina getoond met de opgeslagen samengevoegde beoordeling.

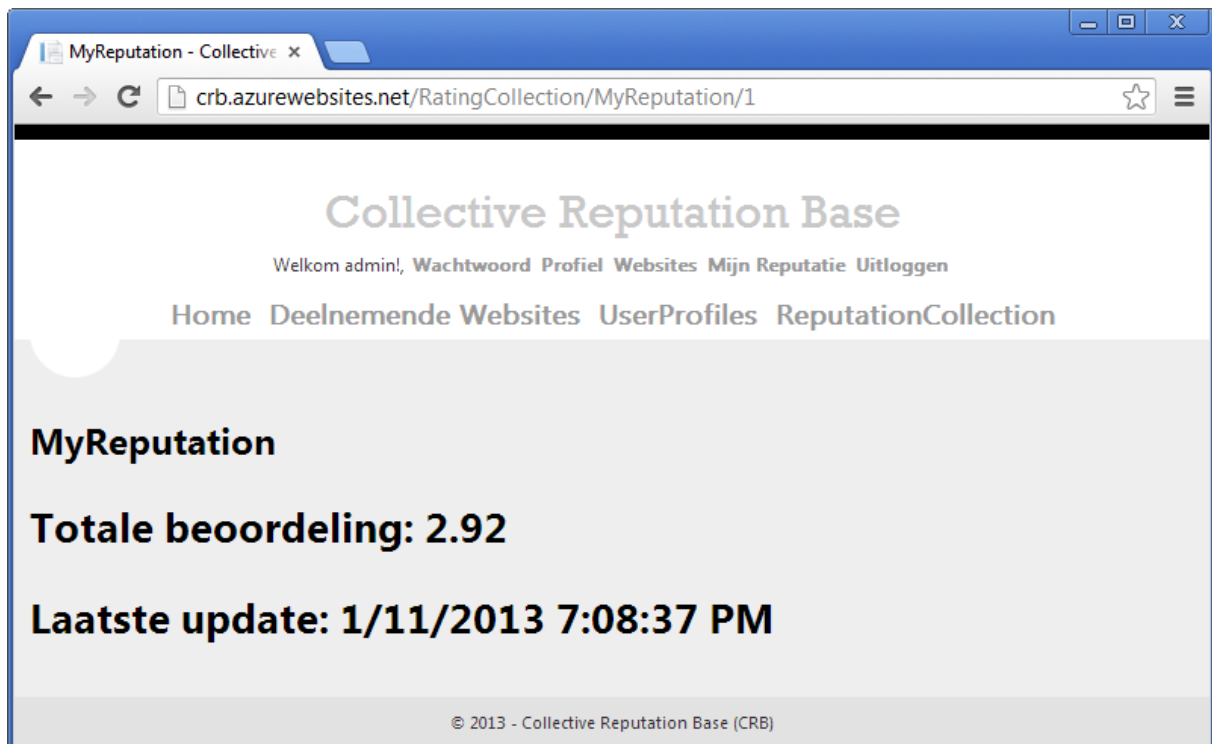


Figuur 41: gebruikersbeheerpagina (link), resultaatpagina na samenvoegen (recht)

Als laatste onderdeel heb ik een pagina gemaakt waarin de reputatie wordt getoond van de gebruiker en wanneer de laatste update is geweest. (Figuur 42)

Binnen deze Sprint heb ik geen problemen gehad met implementeren van de functionaliteiten.

Tevens wil ik hier melden dat bij de succesvolle implementatie van elke functionaliteit, de source code wordt ingecheckt naar het gemaakte project op (<https://crb.visualstudio.com>)



Figuur 42: voorbeeld Mijn Reputatie pagina

8.3 Gebruik LINQ³²

LINQ is een geïntegreerde query in een programmeertaal. Het is een set van functies die gebruikt kan worden binnen een programmeertaal zoals C#. LINQ was voor het eerst geïntroduceerd in Visual Studio 2008. (15)

Ik heb LINQ gebruikt bij voor het bevragen van de CRB database voor het ophalen van de reputaties van de P2P-websites en bij het samenvoegen van de reputaties. Hieronder is een voorbeeld te zien van de query waarbij de verzamelde reputaties van een gebruiker wordt opgevraagd op basis van zijn gebruikersId.

```
var query = from u in db.ReputationCollections
            where u.UserId == userId
            select u.Rating;
```

8.4 Succesvol uitgevoerde requirements

Hieronder is een overzicht te zien van de uitgevoerde functional requirements:

#	Eis	Omschrijving	Prioriteit
F01	Registreren op CRB	Gebruikers moeten zich kunnen registreren in het CRB systeem met behulp van Facebook-, twitter- en/of google-account. Gebruiken OAuth technologie.	M
F02	Inloggen op CRB	Gebruikers moeten kunnen inloggen op CRB met eigen inloggegevens of met OAuth autorisatie.	M
F04	Opgeven P2P-websites	Gebruikers moeten kunnen opgeven op welke P2P-websites zij actief zijn.	M
F05	Verwijderen P2P-websites	Gebruikers moeten kunnen opgeven op welke P2P-websites zij niet meer actief zijn.	S

³² Language-Integrated Query.

F06	Aanpassen profiel	De gebruikers moeten hun profiel kunnen aanpassen.	C
F08	Periodiek ophalen van de reputaties van de P2P-websites	De reputaties worden met een bepaalde frequentie opgehaald van de P2P-websites.	M
F10	De verzamelde reputatiegegevens op een database opslaan	De reputatiegegevens die verzameld zijn, worden op een database opgeslagen.	M
F11	Samenvoegen van de reputaties	Verschillende soorten reputaties tot een algemene type samenvoegen	M
F12	Raadplegen eigen reputaties	De gebruikers moeten eigen reputaties kunnen raadplegen.	S

Tabel 12: uitgevoerde functional requirements

8.5 Uitgevoerde non-functional requirements

Bij het bouwen van het PoC heb ik de volgende non-functional requirements toegepast:

#	Eis	Omschrijving	Soort ³³	Prioriteit
NF01	Bescherming persoonsgegevens	Het systeem moet voldoen aan de voorwaarden in de wet Bescherming persoonsgegevens om de gegevens van de gebruikers te waarborgen.	F	M
NF05	ASP.NET MVC gebruiken voor het bouwen van het systeem	Het systeem wordt gebouwd met behulp van .NET technologie. Hiervoor wordt gebruik gemaakt van APS.NET MVC 4 C#.	OV	M
NF06	Het systeem op de Azure cloud omgeving hosten	Het systeem moet in de Azure websites cloud omgeving worden gehost.	BR	M

Tabel 13: toegepaste non-functional requirements

Er is nog één non-functional requirement die de "Must have" prioriteit had (NF04: Commentaar programmacode). Deze is door mij niet uitgevoerd door de beperkte tijd die nog beschikbaar was in overleg met de opdrachtgever.

8.6 Publiceren van het PoC op Windows Azure omgeving

Het publiceren van de Web Applicatie op de Windows Azure omgeving is momenteel eenvoudiger geworden dan vroeger. Bij het introduceren van de Visual Studio 2012-tool is het mogelijk geworden om direct een Web Applicatie te kunnen publiceren op de Windows Azure omgeving door het kiezen van Publish functionaliteit en het importeren van de Publishing Profile van desbetreffende website. Vroeger moest men een Windows Azure publicatie project toevoegen aan zijn Web applicatie waarmee het project op Windows Azure kan worden gepubliceerd.

Om de PoC te publiceren heb ik eerst in de gemaakte Windows Azure omgeving een website aangemaakt met een database en de Publishing profile gedownload. Met behulp van de publishing profile worden alle benodigde gegevens vooraf ingevuld op de Publish pagina van de Visual Studio.

Na het kiezen van Publish en importeren van de Publishing Profile heb ik de PoC succesvol kunnen publiceren. De home pagina was gestart zonder problemen, maar bij het klikken op de andere linkjes kreeg ik een onbekende foutmelding. Ik heb daarvoor verschillende onderdelen onderzocht om te onderzoeken of er iets fout is gegaan. Lokaal werkte de applicatie zonder problemen.

³³ In de non-functional requirements tabel zijn de volgende afkortingen gebruikt:

F: Functionaliteit, BT: Betrouwbaarheid, BR: Bruikbaarheid, ON: Onderhoudbaarheid, OV: Overdraagbaarheid, EF: Efficiëntie.

Na het zoeken naar de mogelijke oplossingen en puzzelen tussen verschillende onderdelen van het PoC, ben ik erachter gekomen dat de probleem komt door de database die niet goed gepubliceerd is in SQL Azure.

Daarna heb ik enkele scenario's uitgetest, zoals: het aanpassen van de `ConnectionString`³⁴, controlleren van de SQL Azure database of er iets er in staat en nog andere scenario's.

Uiteindelijk heeft één van de scenario's resultaat opgeleverd. Ik had een database server aangemaakt op de Windows Azure omgeving en met behulp van de Microsoft SQL Server 2012-tool was ik ingelogd op de SQL Azure database. Binnen SQL Server-tool is een mogelijkheid om een database te publiceren naar Windows Azure. Dit was ook een nieuwe functionaliteit van SQL Server bij deze nieuwe versie. Ik kon uiteindelijk de lokale database naar de SQL Azure database publiceren en daarmee was het PoC volledig werkend.

Het PoC van de CRB is op de volgende URL online gezet: <http://crb.azurewebsites.net>

³⁴ `ConnectionString` bestaat uit enkele gegevens (zoals: database servernaam, database naam, gebruikersnaam en wachtwoord) waarmee de Web Applicatie verbinding kan maken met de database.

9. Evaluatie

In dit hoofdstuk zal ik het geheel traject van mijn afstuderen evalueren. Het proces wordt eerst geëvalueerd, daarna de opgeleverde producten en uiteindelijk een wordt verantwoording gelegd voor het aantonen van de uitgevoerde beroepstaken.

9.1 Proces evaluatie

Over het proces ben ik in het algemeen tevreden. Zeker bij het aflopen van het project. Ik ben in februari 2012 met dit project begonnen en ik zou het uitvoeren naast mijn fulltime werk bij Logica. Maar helaas was er weinig tijd naast mijn werk om aan dit project te werken. Daardoor heb ik besloten om vanaf oktober 2012 een onbetaald verlof aan te vragen bij Logica om fulltime met mijn afstuderen bezig te zijn. Mijn manager bij Logica en de begeleiders op de HHS waren positief daarover en hebben mijn verteld dat het een verstandige keuze is. Er zijn enkele onderwerpen die binnen proces evaluatie vallen die ik wil behandelen. Deze worden hieronder toegelicht.

9.1.1 Uitvoering Scrum

Ik was bekend met de Scrum softwareontwikkelingsmethode door het toepassen hiervan op enkele project die ik bij Logica heb uitgevoerd en ik heb ook een workshop (Agile Scrum Awareness) bijgewoond bij Logica over deze methode. Daarnaast heb ik literatuur daarover geraadpleegd bij het opstellen van de afstudeerplan.

Bij het uitvoeren van deze methode tijdens het bouwen van het PoC heb ik me meer verdiept in deze methode en heb ik enkele termen binnen de Scrum meer leren kennen zoals: Definition of Done, Product Owner en andere termen.

Ik heb helaas niet alle onderdelen van Scrum kunnen ervaren. De reden daarvoor is dat een Scrum team uit meerdere personen bestaat die met elkaar samenwerken. Maar tijdens dit project ik was de enige persoon die deel maakte van het Scrum team, uiteraard naast de opdrachtgever die de Product Owner is.

Ik heb ook de Daily Scrum bijeenkomsten gemist, waarmee ik zeker meer ervaringen zou krijgen met het uitvoeren van dit methode.

In de toekomst wil ik zeker meer projecten uitvoeren met behulp van deze methode omdat ik vind dat het een waardevolle middel is voor het opleveren van succesvolle projecten.

9.1.2 Planning

Zoals ik hierboven heb aangegeven had mijn afstudeerperiode enkele knelpunten waardoor ik de planning moest aanpassen. De planning is niet één van mijn sterke punten, daarom had ik een beetje moeite met het plannen van sommige taken. Hiervoor heb ik hulp gevraagd bij mijn bedrijfsmentor en kon ik daarmee gelukkig taken beter plannen. Het was een belangrijke leerpunt voor mij en het zal zeker invloed hebben op mijn toekomstige werkzaamheden bij Logica of waar dan ook.

9.1.3 Voortgangsgesprekken

In het begin waren de toegangsgesprekken met de bedrijfsmentor niet in een bepaalde frequentie. Soms maandelijks of via de telefoon. Maar nadat ik in oktober 2012 was begonnen met fulltime afstuderen, hadden wel elke twee weken een toegangsgesprek waarbij het proces werd besproken en de voortgang tot zo ver. Tijdens deze gesprekken werden ook documenten doorgenomen en gereviewed. Naar mijn mening waren de gesprekken waardevol, doorslaggevend en belangrijk.

9.1.4 Feedback begeleiders en opdrachtgever

We³⁵ hadden afgesproken dat elke nieuwe versie van een door mij opgestelde document direct opgestuurd wordt naar de betrokkenen, zodat ze feedback kunnen geven op het document zodat ik deze opmerkingen kan verwerken en opnemen in een nieuwe versie. Meestal werden de documenten via email gestuurd en kreeg ik ook hun feedback via de email terug. Met behulp van deze feedback kon ik verschillende documenten op een juiste manier opstellen, zodat deze documenten zo compleet mogelijk zijn en voldoen aan de verwachtingen van de opdrachtgever en mijn opleidingsniveau.

9.1.5 Gevolgde cursussen

Zoals ik in hoofdstuk 8 heb vermeld heb ik verschillende cursussen gevolgd via Pluralsight.com. Met het praktiseren van deze cursussen kon ik veel problemen oplossen bij het bouwen van het PoC. Het was een waardevolle middel die Logica voor mij beschikbaar had gesteld, waardoor ik veel nieuwe onderwerpen heb geleerd niet alleen voor mijn afstuderen maar ook voor mijn werkzaamheden als Software Engineer.

9.2 Producten evaluatie

Over de producten die opgeleverd zijn waren de opdrachtgever en de begeleiders positief en daarmee ben ik ook tevreden. Ik heb mijn best gedaan om producten op te leveren die uiterst compleet zijn, wellicht bij sommige producten was dit wat minder door de tijdsnood die ik heb opgelopen door enkele problemen van het PoC.

9.2.1 Afstudeerplan

Het afstudeerplan was opgesteld door mij en kende meerdere versies. Deze versies zijn ontstaan door de reviews van dit documenten door meerdere personen. In het begin had ik mijn plan besproken met een Architect binnen Logica en we hebben samen de beroepstaken gekozen die uitgevoerd zullen worden. Na het bespreken van deze keuzes met de begeleiders op school en ook bij Logica zijn er enkele wijzigingen aangebracht.

In het algemeen heb ik dit plan goed kunnen uitvoeren, behalve enkele afwijkingen die ontstaan zijn door het proces en de methode die ik had gebruikt. De afwijkingen waren niet te vermijden omdat sommige dingen niet helder waren aan het begin van het traject, maar in ieder geval is de hoofdlijn van het project het zelfde gebleven. Deze afwijkingen kan ik samenvatten in de volgende punten:

- In het afstudeerplan staat dat verschillende cloud mogelijkheden worden onderzocht en aan de hand van het onderzoek wordt eentje gekozen. Maar tijdens het schrijven van het Plan van aanpak heeft Logica het gebruik van Windows Azure Cloud omgeving als randvoorwaarde aangegeven, en daardoor vervalt dit onderzoek en daarmee vervalt ook de eerste gekozen beroepstaak (1.3: selecteren van standaardsoftware). In plaats daarvan had ik veel tijd besteed aan het opstellen van de requirements en daardoor heb ik de beroepstaak '1.4: Uitvoeren analyse door definitie van requirements' gekozen om uit te voeren. Dit laatste was ook een suggestie van de HHS begeleiders;
- In het afstudeerplan staat dat een van de op te leveren producten de Gegevensmodel database zou zijn. Dit document is een onderdeel geworden van het architectuurdokument onder Information View;
- Ik had de Scrum Sprints over het geheel project verspreid bij het opstellen van het afstudeerplan. Deze verdeling is aangepast zodat de Sprints worden toegepast alleen tijdens het bouwen van PoC. De reden daarvan is dat dit een softwareontwikkelingsmethode is en geen projectmanagementmethode.

35 Ik zelf, bedrijfsbegeleiders en de opdrachtgever.

9.2.2 Plan van aanpak

In het plan van aanpak had ik meer duidelijkheid kunnen geven over het project ten opzichte van het afstudeerplan. Zoals ik heb vermeld zijn er enkele afwijkingen tov het afstudeerplan wat betreft de Sprints verdeling en het vastleggen van de randvoorwaarde 'Het gebruik van Windows Azure als cloud omgeving'. Verder is de planning enkele keren aangepast vanwege veranderingen in de beschikbare tijd voor het uitvoeren van het project. Ik ben tevreden over dit document omdat het duidelijke verdeling geeft over het uitvoeren van de verschillenden onderdelen van het project.

9.2.3 Requirements

Het Requirementsdocument is één van de documenten waar ik erg tevreden mee bent. Door de belangrijke feedback die ik heb gekregen van de opdrachtgever en de begeleiders kon ik een duidelijke beeld vormen over mijn voorstel voor de functional en non-functional requirements. Met behulp van dit document kon ik goed het architectuurontwerp schrijven en het PoC bouwen. De opdrachtgever en de begeleiders zijn ook positief daarover.

9.2.4 Onderzoeksrapport

Bij het opstellen van het onderzoeksrapport heb ik goede overwegingen en voorstellen kunnen doen voor het samenvoegen van de reputaties en dit was een belangrijk onderdeel die als basis is gebruikt bij het bouwen van de PoC. Ik wilde meer tijd besteden voor het communicatie gedeelte van het onderzoek, maar helaas was dat niet mogelijk doordat ik veel tijd kwijt ben geraakt bij het uitvoeren van de enquête. Maar in het algemeen zijn er waardevolle resultaten verkregen met het onderzoek en die zijn gebruikt bij het architectuurontwerp en de PoC.

9.2.5 Architectuurontwerp

Ik heb geprobeerd om zo compleet mogelijk een architectuur op te stellen. Ik denk dat het mij wel gelukt is, dat geven ook de begeleiders aan. In het opgestelde architectuurontwerp is een goede vertaling te vinden van de requirements en met het ontwerp is het duidelijk hoe de PoC gebouwd kan worden. Ik zou zeker sommige views wat dieper willen uitschrijven zoals Development View, maar ik vind het zo ver voldoende voor dit project.

9.2.6 Proof of Concept CRB

Ik ben tevreden met het resultaat dat ik heb bereikt. Er zijn onderdelen waar ik meer aandacht aan wil besteden, zoals: het gebruiken van de Table Storage en betere weergave van de views. Ik had weinig tijd besteed aan de lay-out van de Web Applicatie. Maar in iedere geval is voor mij belangrijk dat alle "Must have" requirements naast twee "Shoul have" en één "Could have" requirements zijn geïmplementeerd en kon ik mijn opgestelde architectuur goed vertalen naar de werkelijkheid. Ik had een demo gegeven aan de begeleider bij Logica en hij was positief over het opgeleverde werk.

9.3 Verantwoording van de beroepstaken

Hieronder wordt verantwoording afgelegd over hoe verschillende beroepstaken zijn uitgevoerd. Zoals ik eerder heb gemeld in plaats van beroepstaak 1.3 heb ik de beroepstaak 1.4 uitgevoerd, daardoor is 1.3 niet vermeld in de onderstaande paragrafen.

9.3.1 Uitvoeren analyse door definitie van requirements (1.4)

Bij het uitvoeren van deze beroepstaak heb ik mij gehouden aan de richtlijnen die genoemd zijn onder 1.4 bij de beroepstaken document van de opleiding Informatica. Als resultaat van deze beroepstaak kan ik verwijzen naar het requirementsdocument en hoofdstuk 5 van dit document. De beroepstaak was uitgevoerd voor één applicatie waarvoor veel requirements zijn geschreven met een beperkt aantal stakeholders. Zowel de functional en non-functional requirements zijn geschreven en er zijn wijzigingen geweest in de requirements tijdens het traject. Deze heb ik allemaal zelfstandig uitgevoerd en daarmee denk ik dat ik de eindniveau '3' heb bereikt.

9.3.2 Opstellen gegevensmodel voor database (2.1)

Deze beroepstaak heb ik uitgevoerd tijdens het opstellen van het architectuurontwerp. Met behulp van Visio heb ik enkele UML-diagrammen opgesteld, zoals: EER-model en domain model. Dit datamodel bevat verschillende soorten relaties zoals: één op veel en één op één relaties. Er zijn ook enkele objecten opgesteld die afhankelijk zijn van relaties tussen twee objecten. En ook een combinatie tussen de Membership Provider entiteiten en de CRB entiteiten. De resultaten daarvan zijn te vinden in het architectuur document in 'Information View' en in dit document onder §7.4. Het opgesteld model had ik ook besproken met de begeleiders en ook afgestemd met de opdrachtgever. Aan de hand van de werkzaamheden die ik heb verricht binnen deze beroepstaak geef ik ook voor deze beroepstaak '3' als eindniveau door de zelfstandigheid en de lastigheid van dit onderdeel.

9.3.3 Ontwerpen, bouwen en bevragen van een database (2.2)

Bij deze beroepstaak behoort het opstellen van het RRM-model en het implementeren van de database via het 'ReputationBase.Domain' project die duidelijk is uitgelegd in §8.2.1. Voor deze implementatie zijn de SQL Server, SQL Azure en Table Storage gebruikt. Bij het bevragen van de database is binnen de Web applicatie gebruik gemaakt van LINQ. De gebouwde database wordt door verschillende onderdelen van het PoC gebruikt en bij het bevragen van data worden meerdere tabellen benaderd zoals bij het toevoegen, wijzigen en verwijderen van de P2P-websites bij een bepaalde gebruiker. Daarom heb ik voor deze beroepstaak het eindniveau tussen de '3' en '4' kunnen plaatsen door de sturende rol die ik hierin had voor een lastige opdracht.

9.3.4 Ontwerpen softwarearchitectuur (3.1)

Het architectuur die ontworpen is voor de CRB is een ontwerp voor een gedistribueerde applicatie. De CRB heeft meerdere onderdelen die verschillende functionaliteiten bevatten en daarnaast wordt binnen de CRB contact gelegd met externe applicaties. Bij het opstellen van de architectuur zijn de opgestelde functional en non-functional requirements geraadpleegd en is gebruik gemaakt van de MVC design pattern, objectgeoriënteerde ontwerp en UML modellering voor verschillende onderdelen. Deze zijn allemaal te vinden in het architectuurontwerpdocument en de verantwoording daarvan is te vinden in hoofdstuk 7 van dit document. De eindniveau van deze beroepstaak ligt ook tussen de '3' en '4' door de zelfstandigheid en het bouwen van een lastig tot een complexe architectuur met aanwezigheid van verschillende viewpoints.

9.3.5 Bouwen applicatie (3.3)

Voor het tonen van deze beroepstaak heb ik een web applicatie gebouwd met behulp van ASP.NET MVC 4 en door het gebruiken van C# programmeertaal met .NET framework 4.5. De ontwikkelomgeving bestond uit: Visual Studio 2012, SQL Server

2012, Azure Storage Explorer en de online TFS Server van Microsoft tfs.visualstudio.com. Bij het bouwen van de PoC heb ik rekening gehouden met herbruikbaarheid, programmeerstandaarden, wijzigbaarheid en onderhoudbaarheid. Bij het bouwen zijn enkele 3^d-party Libraries gebruikt zoals HtmlAiliyPack. Met de bovenstaande kenmerken van de gebouwde PoC plaats ik deze beroepstaak ook tussen eindniveau '3' en '4' door de lastigheid en de sturende rol die ik daarin had.

Literatuurlijst

1. **CGI.** Informatie | Over Logica. *Logica.nl*. [Online] [Citaat van: 01 11 2012.] <http://www.logica.nl/we-are-logica/about-logica/>.
2. —. CGI at a glance. *CGI*. [Online] [Citaat van: 01 11 2012.] http://www.cgi.com/sites/cgi.com/files/brochures/cgi_broc02_cgi_at_a_glance.pdf.
3. **Verhoeven, Nel.** *Wat is onderzoek?* Den Haag : Boom Lemma uitgevers, 2011. ISBN 978 90 5931 671 3.
4. Onderzoek in stappen. *Onderzoekinstappen.nl*. [Online] [Citaat van: 11 11 2012.] <http://www.onderzoekinstappen.nl/>.
5. **Nick, Rozanski en Eoin, Woods.** *Software Systems Architecture*. sl : Addison-Wesley Pearson Education, 2005. ISBN: 9780321112293.
6. —. *Software Systems Architecture*. 2nd. sl : Addison-Wesley, 2011. ISBN:9780321718334.
7. **Prowareness.** Wat is Scrum. *Scrum.nl*. [Online] [Citaat van: 11 11 2012.] <http://www.scrum.nl/site/Wat-is-Scrum-agile-scrum>.
8. **Jos, Warmer en Anneke, Kleppe.** *Praktisch UML*. 3de. sl : Pearson Education Benelux, 2006. ISBN: 9789043008129.
9. **coaching, Academy.** Gebruik SMART methode voor haalbare en meetbare doelen. *Academy coaching*. [Online] [Citaat van: 01 11 2012.] <http://www.academycoaching.nl/blog/?p=109>.
10. **Haughey, Duncan.** MoSCoW Method. *ProjectSMART.co.uk*. [Online] [Cited: 01 11 2012.] <http://www.projectsmart.co.uk/moscow-method.html>.
11. **Marktplaats.nl.** Victor Croes - Marktplaats Perskamer. *Marktplaatsperskamer*. [Online] [Citaat van: 6 11 2012.] <http://marktplaatsperskamer.nl/mediatheek/marktplaats-gezichten/victor-croes/>.
12. **Tré, de Guy.** *Principes van databases*. Amsterdam : Pearson Education Benelux bv, 2007. ISBN: 978-90-430-1302-4.
13. **Microsoft.** How to use the Table Storage Service. *Windows Azure*. [Online] [Citaat van: 2 12 2012.] <http://www.windowsazure.com/en-us/develop/net/how-to-guides/table-services/#what-is>.
14. **Microsoft, Web Development Tools.** .NET Web Development and Tools Blog. *MSDN*. [Online] [Citaat van: 5 12 2012.] <http://blogs.msdn.com/b/webdev/archive/2012/08/15/visual-studio-2012-rtw.aspx>.
15. LINQ (Language-Integrated Query). *Visual Studio*. [Online] [Citaat van: 25 12 2012.] <http://msdn.microsoft.com/en-us/library/vstudio/bb397926.aspx>.
16. **Fowler, Martin and Scott, Kendall.** *UML Distilled: A brief guide to the standard Object Modeling Language*. 2000. ISBN: 0-201-65783-X.
17. **Thompson, Kevin.** When to Use Scrum for software projects? *Cprime*. [Online] [Cited: 03 01, 2011.] <http://www.cprime.com/community/articles/whentousescrum.html>.
18. **Resnick, Paul, et al., et al.** Reputation systems. *Communications of the ACM*. December 2000, Vol. 43, 12, pp. 45-48.
19. **DathObiwan.** Html Agiliy Pack. *CodePlex.com*. [Online] [Citaat van: 25 12 2012.] <http://htmlagilitypack.codeplex.com/>.

Begrippenlijst

Woord	Beschrijving
.NET C#	C# is een object georiënteerde programmeertaal op basis van het .NET Framework.
.NET framework	.NET is een applicatie framework ontwikkeld door "Microsoft". Hiermee kunnen de applicaties worden ontwikkeld en gepresenteerd die geprogrammeerd zijn met behulp van de Microsoft gerelateerde programmeertalen.
Aanbieders	De aanbieders zijn personen die hun artikelen en/of diensten aanbieden op de P2P-websites.
Afnemers	De afnemers zijn diegene die de artikelen en/of diensten afnemen van de aanbieders.
ASP.NET MVC 4	Is de laatste versie van de Microsoft implementatie van de MVC (Model View Controller) Design Pattern.
Command Handler	Het is een geautomatiseerde interne proces binnen een systeem die wordt uitgevoerd voor het afhandelen van bepaalde werkzaamheden.
Membership provider	Is een interne ASP.NET manier voor validatie en opslag van de gebruikersgegevens.
OAuth authorization	Registreren op een website via een andere website zoals Facebook, Twitter, Live en Google.
P2P-website	Vraag-en-aanbod website waar particulieren met elkaar worden gematched.
SQL Azure	Is een Microsoft cloud implementatie van de SQL Server. De SQL Azure is een onderdeel van Windows Azure omgeving.
SQL Server	Is een Microsoft informatie platform en een relationeel database management systeem.
Web API	De Web API kan omschreven worden als een interface van een applicatiecomponent die toegankelijk is via standaard web protocollen (HTTP) en waarbij wordt gecommuniceerd via XML zonder menselijke tussenkomst.
Windows Azure	Is een platform van Microsoft voor cloud computing waarmee een aantal internetdiensten aangeboden kunnen worden via het internet of binnen de omgeving van eigen bedrijf.
Windows Azure Table Storage	deze component is een onderdeel van de Windows Azure omgeving. Hierin kunnen grote hoeveelheden gestructureerde data worden opgeslagen. De service is een NoSQL datastore die geverifieerde aanvragen van binnen en buiten de Windows Azure cloud accepteert. Windows Azure tabellen zijn ideaal voor het opslaan van gestructureerde, niet-relatieve gegevens. ⁽²⁾

Afkortingen

Woord	Beschrijving
CRB	Collective Reputation Base
DoD	Definition of Done
EER	Enhanced Entity-Relationship, is een model waarmee de entiteiten, attributen en de relaties van een database ontwerp wordt gemoduleerd.
LINQ	Language-Integrated Query
MVC	Model View Controller
OAuth	Open Authentication
PO	Product Owner
PoC	Proof-of-Concept. Een demo van het te bouwen software.
SMART	Specifiek, Meetbaar, Actueel, Realistisch en Tijdsgebonden
SQL	Structured Query Language
TFS	Team Foundation Service
UML	Unified Modeling Language

Figurenlijst

Figuur 1: organogram Logica, nu onderdeel van CGI	9
Figuur 2: fases "Wat is onderzoek?"-methode	13
Figuur 3: viewpoints Software Systems Architecture	13
Figuur 4: Scrum overview diagram	15
Figuur 5: versie 1 use-case-diagram	18
Figuur 6: versie 2 use-case-diagram	18
Figuur 7: laatste versie use-case-diagram	19
Figuur 8: Aanwezigheid reputatiesystemen op de P2P-websites.....	25
Figuur 9: Ster beoordelingsvorm voorbeeld op ebay.nl	26
Figuur 10: Gebruik verschillende reputatievormen door P2P-websites	26
Figuur 11: Gebruikersbeoordeling vs productbeoordeling	26
Figuur 12: omzetten van verschillende reputatievormen naar de Ster model.....	27
Figuur 13: aantal gebruikers en bezoekers per P2P-website.....	30
Figuur 14: beschikbaar stellen reputatiedata voor de CRB	30
Figuur 15: gebruik CRB na de realisering	31
Figuur 16: Web Service op de CRB	32
Figuur 17: Web Service op de P2P-website	32
Figuur 18: Concept Context diagram CRB.....	34
Figuur 19: Context diagram	35
Figuur 20: sequentiediagram (Ophalen reputaties van P2P-websites en berekenen totale reputatie op de CRB).....	36
Figuur 21: voorbeeld schets componenten diagram	37
Figuur 22: 2 ^e versie component diagram (Inloggen en registreren op de CRB)	37
Figuur 23: laatste versie component diagram (Inloggen en registreren op de CRB).....	38
Figuur 24: Functioneel structuur model	38
Figuur 25: voorbeeld sequentiediagram (Opgeven P2P-websites op CRB)	39
Figuur 26: eerste versie systeem data structuur.....	41
Figuur 27: laatste versie systeem data structuur	42
Figuur 28: schets EER-model	42
Figuur 29: EER-model van de CRB database	43
Figuur 30: Relationeel Representatie Model van de CRB DB	43
Figuur 31: concurrency model CRB	44
Figuur 32: eerste versie module structuur	45
Figuur 33: tweede versie module structuur.....	45
Figuur 34: laatste versie van de module structuur	46
Figuur 35: Klasse diagram CRB	47
Figuur 36: Deployment model	48
Figuur 37: inlog pagina met OAuth mogelijkheden.....	54
Figuur 38: profiel pagina	55
Figuur 39: websites registratie pagina.....	56
Figuur 40: voorbeeld opgeslagen score	57
Figuur 41: gebruikersbeheerpagina (link), resultaatpagina na samenvoegen (recht)	58
Figuur 42: voorbeeld Mijn Reputatie pagina	59

Tabellenlijst

Tabel 1: Versiebeheer	3
Tabel 2: Distributielijst.....	3
Tabel 3: Projectorganisatie	10
Tabel 4: voorbeeld use-case beschrijvingen	20
Tabel 5: voorbeelden functional requirements	21
Tabel 6: voorbeelden non-functional requirements	22
Tabel 7: onderzoeksvragen en de types en te gebruiken methoden	24
Tabel 8: Voorbeeld functionele elementen en hun verantwoordelijkheden.....	39
Tabel 9: Data levenscycli.....	44
Tabel 10: Data tijdslijnen	44
Tabel 11: Functionele elementen vs Deployment Node(s)	49
Tabel 12: uitgevoerde functional requirements.....	60
Tabel 13: toegepaste non-functional requirements	60