

De herimplementatie van de automatisering van  
stichting 'De Delftse Bedrijvendagen'

**S.J. Karger 1217526**

**R.P. Evers 1099752**

**B.J. Schaafsma 1150553**

*Faculteit Elektrotechniek, Wiskunde en Informatica*

*Technische Universiteit Delft*



# Inhoudsopgave

1	Inleiding . . . . .	5
1.1	Stichting ‘De Delftse Bedrijvendagen’ . . . . .	5
1.2	Doel Project . . . . .	6
1.3	Indeling rapport . . . . .	6
2	Projectplanning en -verloop . . . . .	7
3	Analyse huidige systeem . . . . .	8
3.1	Beschrijving huidige systeem . . . . .	8
3.2	Tekortkomingen huidige systeem . . . . .	10
4	Analyse requirements nieuwe systeem . . . . .	12
4.1	Requirements gathering . . . . .	12
4.2	MoSCoW analyse . . . . .	12
4.3	Prioriteitsanalyse requirements . . . . .	12
4.4	Keuze projectmethodologie . . . . .	12
5	Ontwerp . . . . .	13
5.1	Architectureel ontwerp . . . . .	13
5.2	Database ontwerp . . . . .	14
5.3	Gebruikte technieken en software pakketten . . . . .	16
5.4	Implementatie van het MVC pattern . . . . .	18
5.5	De mapping van subsystemen naar processoren en componenten . . . . .	20
5.6	Het identificeren en opslaan van persistente data . . . . .	20
5.7	Access control . . . . .	21
5.8	Global control flow . . . . .	22
5.9	Boundary conditions . . . . .	22
6	Implementatie verslag . . . . .	24
6.1	Leercurve . . . . .	24
6.2	HTML & CSS integratie . . . . .	24
6.3	Testen en kwaliteitscontrole . . . . .	24
7	Evaluatie . . . . .	26
7.1	Behaalde resultaten . . . . .	26
7.2	Third party tools en software . . . . .	27
7.3	Samenwerking met Stichting ‘De Delftse Bedrijvendagen’ . . . . .	27

7.4	Samenwerking met Shibby Projects . . . . .	28
8	Conclusie . . . . .	29
	Verklarende woordenlijst . . . . .	30
	Bibliografie . . . . .	32
A	Database model . . . . .	35
B	FormLib . . . . .	38
C	Requirements . . . . .	40

## 1. Inleiding

Dit verslag beschrijft het bachelor project van Bas Schaafsma, Ronald Evers en Steffan Karger. Het doel van dit project is de website en alle achterliggende systemen van stichting ‘De Delftse Bedrijvendagen’ opnieuw te implementeren in één systeem. Dit in tegenstelling tot het huidige systeem waarin vele losse systemen en scripts op een chaotische en niet onderhoudbare manier met elkaar samenwerken.

### 1.1 Stichting ‘De Delftse Bedrijvendagen’

Stichting ‘De Delftse Bedrijvendagen’ (DDB) is een initiatief van vijf studieverenigingen in Delft waaronder Wiskunde en Informatica Studievereniging ‘Christiaan Huygens’. Ieder van deze verenigingen vaardigt jaarlijks een bestuurder af om de stichting te besturen. DDB organiseert elk jaar diverse evenementen om studenten en bedrijven met elkaar in contact te brengen, waaronder een grote beurs in de aula van de TU Delft. DDB is in 12 jaar uitgegroeid tot een van de grootste technische bedrijvenbeurzen van Nederland. DDB stelt zich tot doel studenten aan de TU Delft in contact te brengen met bedrijven die hun mogelijk een baan, stage of afstudeerplaats kunnen bieden en bedrijven zich een beeld te laten vormen van afstuderend Delft.

De afgelopen jaren zijn er per editie vier evenementen: de Sollicitatietraining, de Presentatiedagen, de Workshops en de Gesprekkendagen.

Tijdens de Sollicitatietraining begin februari komen werknemers van verschillende bedrijven een sollicitatietraining geven aan deelnemers van DDB. De trainingen worden op één of twee dagen gegeven. Trainingen worden gegeven aan kleine groepen studenten, vinden simultaan plaats en studenten kunnen maar aan één training deelnemen.

Het hoofdevenement elk jaar wordt gevormd door de Presentatiedagen. Medio februari presenteren gedurende drie dagen circa 100 bedrijven zich op een beurs in de Aula. Elke dag staat een derde van de bedrijven op de beursvloer met een stand en kunnen dezelfde bedrijven presentaties geven. Vrijwel alle 1200 student-deelnemers aan DDB komen gedurende deze drie dagen langs.

Na de Presentatiedagen vinden in maart de Workshops plaats. Bedrijven kunnen zich opgeven om een workshop te geven, waarbij studenten een case kunnen doen die het bedrijf hun voorschotelt. De meeste workshops vinden plaats op locatie bij het bedrijf, DDB draagt zorg voor het vervoer. Daarnaast zijn er gedurende een dag medio maart enkele workshops in Delft. DDB verzorgt hiervoor de locatie, lunch, borrel en overige voorzieningen.

Tenslotte zijn er de Gesprekkendagen, waarbij deelnemers één op één in gesprek kunnen gaan met een bedrijf. Afhankelijk van de insteek van bedrijf en studenten kunnen deze gesprekken solliciterend of oriënterend van aard zijn. Deelnemers kunnen hun C.V. via de website invullen. Deze worden allemaal in een standaardopmaak in bundels aan de bedrijven toegezonden. Bedrijven selecteren op basis hiervan de studenten waarmee ze in gesprek willen treden. Deelnemers kunnen tenslotte gesprekken accepteren of afslaan. DDB verzorgt de locatie, de roostering, faciliteiten voor de C.V.’s en alle overige voorzieningen tijdens de dagen zelf.

Naast het DDB bestuur zijn er nog een aantal groepen mensen die meewerken aan het evenement. Ten eerst is er het Bedrijven Informatie Team (BIT). Het BIT ontfermt zich over het maken

van een informatiepakket voor de deelnemers en helpt bij de coördinatie op de beursvloer tijdens de Presentatiedagen. Het pakket bestaat over het algemeen uit een tasje, mapje, kladblok, pen, programmaboekje en een boek waarin alle bedrijven zich presenteren. Alle deelnemers krijgen dit pakket uitgereikt bij het inchecken voor de Presentatiedagen. Tijdens de Presentatiedagen zijn er daarnaast nog circa 90 studenten die helpen met het in goede banen leiden van de Presentatiedagen. Deze studenten worden commissarissen genoemd en staan onder leiding van het BIT (wat onder leiding van het DDB bestuur staat).

## **1.2 Doel Project**

Helaas is, door het uitblijven van een gestructureerde aanpak, de automatisering door de jaren heen rommelig verlopen. Op dit moment bestaat het systeem uit vele losse componenten en is het lastig te onderhouden. Daarnaast bestaan er wensen voor nieuwe functionaliteiten die lastig te implementeren zijn, zoals een koppeling met de NetID database van de TU Delft en betalingen voor studenten met iDeal.

Het is daarom wenselijk een nieuw systeem te ontwerpen en implementeren dat ruimte biedt voor de nieuwe wensen maar geen afbreuk doet aan de functionaliteiten van het oude systeem.

Het hoofddoel van dit project is dan ook een systeem te ontwerpen en implementeren dat de functionaliteit van alle losse scripts, Microsoft Access forms, webpagina's en andere tools die nu door DDB door elkaar heen gebruikt worden, aangevuld met de nieuwe wensen van DDB, integreert in één consistent geheel en deze zoveel mogelijk via één interface beschikbaar maakt.

## **1.3 Indeling rapport**

De rest van dit verslag is ingedeeld als volgt. In hoofdstuk 2 wordt de planning van het project besproken. Hoofdstuk 3 bespreekt de analyse fase van het project. Hier wordt onder andere een korte beschrijving van het huidige systeem gegeven en wordt gekeken naar de behoeften van een nieuw systeem. Het ontwerp van het project wordt besproken in hoofdstuk 5. Hoofdstuk 6 bespreekt het implementatie verslag van het project en hoe wij de kwaliteit van ons product garanderen. Het rapport wordt afgesloten met de evaluatie van het project en een conclusie in hoofdstuk 7 en 8. De appendix van het verslag bevat een overzicht van de requirements, het database ontwerp en een overzicht van het door ons ontwikkelde product FormLib. Daarnaast is apart meegeleverd een uitgebreide API specificatie van FormLib.

## 2. Projectplanning en -verloop

In beginsel hadden we de volgende planning voor het bachelor project gemaakt:

week 1 - 3: beschrijving huidige systeem  
week 3 - 5: analyse requirements  
week 5 - 8: ontwerp nieuwe systeem  
week 8 - 12: implementatie + testen  
week 12 - 14: uitloop

Het project zou volgens onze planning lopen van 25 juni 2007 tot en met 23 september 2007. Door meerdere vertragingen is het project uitgelopen en uiteindelijk (op het verslag na) een maand later afgerond, op 23 oktober. Op die datum is de nieuwe DDB website online gegaan met de algemene informatie en het bedrijvengedeelte.

Nadat we rond de vierde projectweek een uitgebreid requirements analysis document opgesteld hadden in samenwerking met DDB, bleek dat het te veel werk zou zijn voor een bachelorproject om alle requirements op te leveren. Later is hierover met de heer Kluit gesproken en is besloten de grenzen van het project te leggen bij het informatiegedeelte en de functionaliteit van het bedrijvengedeelte aangezien DDB dit als eerste nodig had.

De ontwerpfase van het project heeft langer geduurd dan gepland. Achteraf blijken we ons behoorlijk verslikt te hebben in de hoeveelheid energie die daar ingestoken moest worden. Hoewel ‘slechts’ het bedrijvengedeelte van het systeem geïmplementeerd zou gaan worden moest er wel een compleet, uitbreidbaar backend systeem opgeleverd worden.

Aansluitend op de ontwerpfase kwam direct de tentamenperiode in augustus. Vanwege deze tentamenperiode is de implementeerfase grotendeels uitgesteld tot het begin van het eerste kwartaal van het nieuwe collegejaar. De implementeerfase is in ongeveer acht weken doorlopen. Uiteindelijk hebben we de geplande tijdsduur (plus uitloop) met vijf weken overschreden.

Het verloop van dit project laat zich kenmerken door het continue jongleren van belangen: enerzijds het bachelorproject, anderzijds de behoeften van Stichting DDB, anderzijds onze persoonlijke belangen bij de tentamenperioden van het vierde kwartaal 2006-2007 en de hertentamens in augustus 2007.

### 3. Analyse huidige systeem

In deze sectie zullen we een globale omschrijving geven van het bestaande systeem dat gebruikt wordt door stichting ‘De Delftse Bedrijvendagen’ (DDB). We zullen dit doen door eerst van de verschillende onderdelen de functionaliteiten te beschrijven en daarna te kijken naar de tekortkomingen van het geheel.

#### 3.1 Beschrijving huidige systeem

##### WEBSITE

De website is de belangrijkste interface voor bedrijven en studenten. Voor persoon- of bedrijfspecifieke zaken moet er eerst ingelogd worden.

Via de website kunnen bedrijven zaken regelen die betrekking hebben op hun inschrijving voor de evenementen. Zij kunnen zich aanmelden voor de verschillende onderdelen, vacatures plaatsen, bedrijfsinformatie updaten, studenten selecteren en inzien wat de kosten voor hen zullen zijn.

Studenten kunnen zich ook online aanmelden, kunnen zich opgeven voor specifieke workshops, trainingen en gesprekken, vacatures van bedrijven inzien, hun contactinformatie updaten, hun C.V. online maken en inzien welke bedrijven hen uitnodigt voor gesprekken of workshops.

Bedrijven en studenten krijgen waarschuwingen (niet-automatisch) als ze bepaalde zaken nog moeten aanleveren of invullen.

Als laatste het DDB bestuur: er zijn weinig online tools voor het DDB bestuur. Veel zaken worden direct op de database geregeld. Wat het DDB bestuur wel kan, is bestaande webpagina's wijzigen (alleen textueel) en volgens een rooster offline en online laten gaan, de automatische reminder-emails aanpassen en per pagina interne notities maken.

##### DATABASE

De database is een MySQL systeem bestaande uit ongeveer 64 tabellen en is het hart van het DDB systeem. Alle data die gegenereerd wordt tijdens de loop van DDB wordt hier in opgeslagen. Er wordt op dit ogenblik nog geen gebruik gemaakt van geavanceerde mogelijkheden zoals het checken van sleutel integriteit. De enige gebruikers die direct gebruik maken van dit systeem zijn de DDB bestuurders en hun medewerkers. Aangezien niet alle gewenste data beschikbaar is via handige interfaces is het regelmatig nog nodig om handmatig SQL queries uit te voeren.

##### ROOSTER PROGRAMMA

De roostersoftware is een Python applicatie die als doel heeft het maken en aanpassen van het gesprekkenrooster voor de gesprekkendagen. Gesprekken tussen recruiters en studenten worden ingedeeld op dag, tijd en plaats.

Bij het maken van het rooster houdt de applicatie rekening met de beschikbaarheid van recruiters, voorkeursstudies van recruiters (sommige bedrijven sturen meerdere recruiters die elk een bepaald gebied op zich nemen) en een per recruiter in te stellen maximaal aantal gesprekken. Bedrijven kunnen een bepaalde recruiter aan een student koppelen.



Daarnaast houdt het programma bij het maken van een roostervoorstel rekening met het aantal beschikbare ruimten en probeert het de gesprekken van recruiters zoveel mogelijk achter elkaar in dezelfde ruimte te plannen.

Na het maken van een rooster is het mogelijk via een GUI het rooster aan te passen. Hier is het per gesprek mogelijk om de datum, tijd, ruimte en recruiter te wijzigen. Het rooster is op te slaan op een publieke plaats (de website) en intern als een soort kladversie.

Wanneer er conflicten optreden in het rooster waarschuwt de applicatie de gebruiker hiervoor.

#### LOSSTAANDE PHP SCRIPTS

Op de server van DDB staan een aantal PHP scripts. De meerderheid van deze scripts bestaat uit eenmalige fixes die nooit zijn weggehaald. Er zijn er echter ook een paar die blijvende functionaliteit verzorgen. Hieronder volgt een overzicht van de verzorgde functionaliteit van deze scripts.

- Activeren van bedrijven accounts.
- Een overzicht geven van betalingen van alle studenten.
- Het genereren van een inputfile voor Girotel voor het incasseren van betalingen van studenten.
- In de centrale database toevoegen van inschrijvingen van studenten die zich op de Presentatiedagen zelf (offline!) aangemeld hebben.
- Verwerken van alle ingeleverde C.V.'s van studenten tot een (per bedrijf specifiek) pakket in pdf formaat.

#### MICROSOFT ACCESS APPLICATIES

DDB gebruikt een verzameling Microsoft Access forms ter ondersteuning van Customer Relationship Management. Deze forms bieden de volgende mogelijkheden:

- Een nieuw bedrijf toegevoegen aan de database.
- Contactpersonen bedrijven beheren.
- Communicatiedagboek per bedrijf.
- Specifieke gegevens per bedrijf per evenement weergeven en aanpassen.
- Adresgegevens per student opvragen.
- Adresgegevens en functies per relatie (zoals bijvoorbeeld de Rector Magnificus) opvragen.

#### LOSSTAANDE VISUAL BASIC SCRIPTS

Naast de forms in de Access applicatie zijn er ook Visual Basic scripts voor verschillende taken. Deze taken zijn

- Genereren van facturen voor bedrijven in Access reports.

- Tabel vullen met data waaruit de bevestigingsbrief voor bedrijven gemerged wordt met Microsoft Word.
- Tabel vullen met data waaruit de contracten gemerged worden met Microsoft Word.
- Sollicitatietraining indeling maken voor studenten.

### 3.2 Tekortkomingen huidige systeem

In deze sectie zullen we een opsomming geven van de tekortkomingen en knelpunten van het huidige systeem. Deze analyse hebben wij opgesteld aan de hand van ervaringen van de gebruikers (bedrijven, studenten en DDB-bestuur) en de onderhouders van het systeem. De opsomming zal uitgesplitst worden in een functioneel gedeelte en een niet-functioneel gedeelte.

#### FUNCTIONELE TEKORTKOMINGEN

- Het DDB bestuur kan geen nieuwe webpagina's aanmaken op de website.
- Studenten weten als ze mee willen doen aan de gesprekkendagen niet dat ze hun C.V. moeten invullen.
- Studenten weten als ze meedoen met de workshops niet dat ze geen C.V. hoeven in te vullen.
- Het is niet mogelijk om functionaliteit van een pagina gedeeltelijk te beperken, bijvoorbeeld als we de info van een HTML form willen laten zien maar gebruikers geen nieuwe informatie willen laten opsturen dan wordt nu de submit knop uitgecommenteerd in de code.
- Vele documenten (brieven) die ieder jaar gecreëerd worden, worden ieder jaar met de hand opnieuw opgesteld en gemerged.
- Het workshopoverzicht voor studenten wordt als zeer slecht beschouwd.
- 'What you see is what you get' concept mist soms, bijvoorbeeld als een bedrijf informatie aan een student doorgeeft via de site zien zij niet wat studenten zien.
- Sommige speciale features van het rooster programma —het aangeven van recruiter-student koppels, dagen waarop studenten niet kunnen— moeten rechtstreeks in de database gezet worden.
- Geen koppeling met NetID database TU.
- Geen betaling met iDeal mogelijk.

#### NIET-FUNCTIONELE TEKORTKOMINGEN

- De ODBC koppeling tussen Access en de MySQL server veroorzaakt onverwachte problemen.
- Door het ontbreken van een duidelijk ontwerp is dezelfde data en/of code op meerdere plaatsen te vinden.

- Er is geen goede beschrijving van de database, waardoor velden door verschillende stukken code op verschillende manieren worden geïnterpreteerd.
- Het ontwerp van de database veroorzaakt data-redundantie.
- Door het ontbreken van een duidelijk ontwerp is er een wildgroei aan ongedocumenteerde code ontstaan, waarvan een gedeelte al niet meer bruikbaar is door tussentijdse wijzigingen.
- Grote diversiteit aan gebruikte platformen (MS Access, PHP, Python, VB)
- Software evolutie wordt bemoeilijkt door de non-transparantie van sommige stukken code zoals de 'C.V. engine'.
- Er is geen laag tussen de interface en de data met als gevolg dat de interface van de site slechts moeilijk kan worden aangepast.

#### OVERIGE TEKORTKOMINGEN

Naast bovengenoemde tekortkomingen is er zo goed als geen technische documentatie over het systeem beschikbaar zodat nieuwe mensen een lange inwerktijd nodig hebben om het systeem te leren kennen. Daarnaast wordt het onderhoud gedaan door non-softwareexperts onder stress waardoor veranderingen niet worden gedocumenteerd en ongestructureerd plaatsvinden.

## 4. Analyse requirements nieuwe systeem

### 4.1 Requirements gathering

Zoals eerder gezegd, zijn we alle drie bekend met het organiseren van een editie van De Delftse Bedrijvendagen. Daarom zijn we allereerst zelf van start gegaan met het documenteren van de requirements die de basis-functionaliteit van het systeem omschrijven. Met deze eerste versie zijn we met DDB om tafel gaan zitten om te kijken of we alle basis-functionaliteit te pakken hadden en om te kijken wat DDB voor nieuwe requirements had. Hier zijn een aantal zaken uitgekomen zoals ondersteuning voor aanmelden met NetID en betaling met iDeal.

Nadat we dit allemaal in detail in een excel sheet hadden gezet, is deze met DDB nauwlettend doorgenomen om te kijken of er uiteindelijk alles in staat wat er in staan moet. In appendix C staat het uiteindelijke requirements analysis document.

### 4.2 MoSCoW analyse

Om tijdens het implementeren later in het project snel te kunnen besluiten wat te doen bij te veel of te weinig tijd, zonder opnieuw te moeten analyseren, hebben we een MoSCoW analyse [23] gedaan. Als startpunt hebben wij zelf elke requirement ingedeeld als een **Must have**, **Should have**, **Could have** of **Would like to have**. Dit voorstel hebben we tijdens een overleg aangepast en afgestemd op de wensen van DDB.

### 4.3 Prioriteitsanalyse requirements

Na het noteren van alle requirements werd al snel duidelijk dat er te veel gedaan moest worden om realistisch in de scope van een bachelor project uit te voeren. Daarom hebben we naast de MoSCoW analyse nog een andere analyse over de requirements gedaan. Hierin geven we per requirement aan wanneer hij opgeleverd dient te worden, een soort deadline-analyse.

Hier is uitgekomen dat, qua frontend, het bedrijvengedeelte het eerst af moest, gevolgd door het studenten deel. De studenten schrijven namelijk pas in januari in, de bedrijven in oktober. Daarnaast moet het DDB bestuur natuurlijk de nodige tools hebben om het geheel in goede banen te leiden. De gehele analyse is per requirement te vinden in de appendix over de requirements.

Besloten is dat het implementeren van het bedrijvengedeelte en daarmee dus ook de gehele backend, binnen de grenzen van het bachelor project zou vallen en de rest niet.

### 4.4 Keuze projectmethodologie

Hoewel we hebben geprobeerd de analyse zo secuur en compleet mogelijk te maken, is het waarschijnlijk dat er requirements missen, dat er requirements vervallen of aangepast moet worden. Wij hebben daarom gekozen niet voor een *Waterfall* model [3], maar voor een meer iteratief proces in de ontwikkeling van de software. Mocht er in een willekeurig stadium van het project een requirement blijken te ontbreken, dan dient de volgende procedure gevolgd te worden: bepaal MoSCoW status, bepaal opleverdeadline, pas waar nodig het ontwerp aan en ga over tot aanpassing implementatie.

## 5. Ontwerp

In dit hoofdstuk zullen wij het ontwerp van het door ons afgeleverde systeem bespreken. In 5.1 zullen wij bespreken welke ontwerpbeslissingen wij genomen hebben. Het ontwerp van de gebruikte database wordt besproken in 5.2. Hierna geven wij in 5.3 een overzicht van alle producten die wij gebruikt hebben in de loop van het project en in sectie 5.4 lichten wij toe welke rollen de gebruikte tools op zich nemen.

Hoewel onze aanpak niet helemaal overeenkomt met de aanpak, beschreven in *Object oriented software design* van Bruegge en Dutoit [2] worden hier wel belangrijke punten aangestipt die terugkomen in veel software-ontwerpprocessen. Daarom houden wij deze punten toch aan als leidraad voor ons ontwerp proces. In 5.5 bespreken wij de mapping van subsystemen naar processoren en componenten in ons systeem, in 5.6 bespreken wij hoe ons systeem omgaat met persistente data, in 5.7 bespreken wij access control. De global control flow word besproken in 5.8. Als laatste bespreken we hoe ons systeem met boundary conditions omgaat in 5.9.

### 5.1 Architectureel ontwerp

#### REPOSITORY ARCHITECTUUR

Na analyse van de in hoofdstuk 3 vergaarde requirements en het bestaande systeem hebben wij gekozen om het nieuwe systeem te ontwerpen volgens een repository architectuur. Onze argumenten voor deze keuze zijn als volgt:

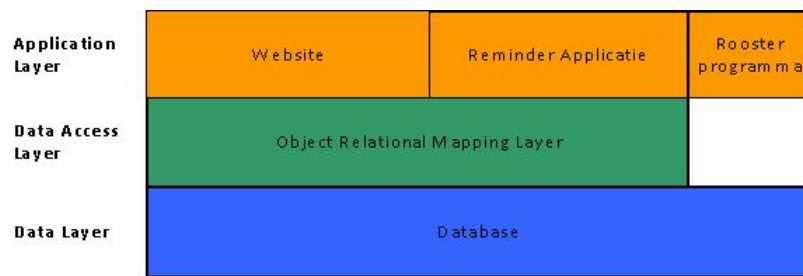
- Het vorige systeem had onbewust een repository architectuur.
- Het te creëren product bestaat voornamelijk uit onafhankelijke tools (webforms) die bewerkingen uitvoeren op één gemeenschappelijke dataset.

Een zeer belangrijk gevolg van deze beslissing is dat ons systeem een ‘data centered’ ontwerp heeft in plaats van een object georiënteerd ontwerp. We zullen ons database model uitgebreid bespreken in 5.2

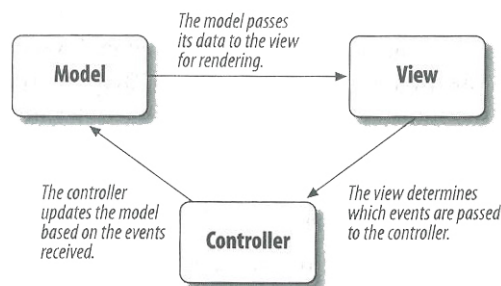
Het centraal zetten van de data wordt verder toegelicht in figuur 1. Hierin wordt getoond hoe de verschillende onafhankelijke applicaties gebruik zullen maken van de database. Om de onderhoudbaarheid van het systeem te bevorderen zal voor de meerderheid van de applicaties de database alleen toegankelijk zijn via een ‘Data Access Layer’. Wij voorzien dat het roosterprogramma direct contact met de database zal hebben omdat dit dusdanig veel queries uitvoert dat naar onze verwachting de overhead van object-creatie de snelheid in te grote mate nadelig zal beïnvloeden. Deze applicaties vallen buiten de scope van dit project en zullen ook niet verder besproken worden.

#### TOEPASSING VAN MODEL VIEW CONTROLLER ARCHITECTURAL PATTERN

Een belangrijk doel van het project is dat het eindproduct goed onderhoudbaar is. Daarom hebben we gekozen het Model View Controller [24] (MVC) pattern toe te passen in het ontwerp van de website. Dit pattern wordt veelvuldig toegepast in websites en is daardoor uitermate geschikt voor ons systeem. Figuur 2 geeft een abstracte representatie van dit pattern.



**Figuur 1.** Gelaagde opbouw van het systeem



**Figuur 2.** Abstracte representatie van het model view controller pattern

## 5.2 Database ontwerp

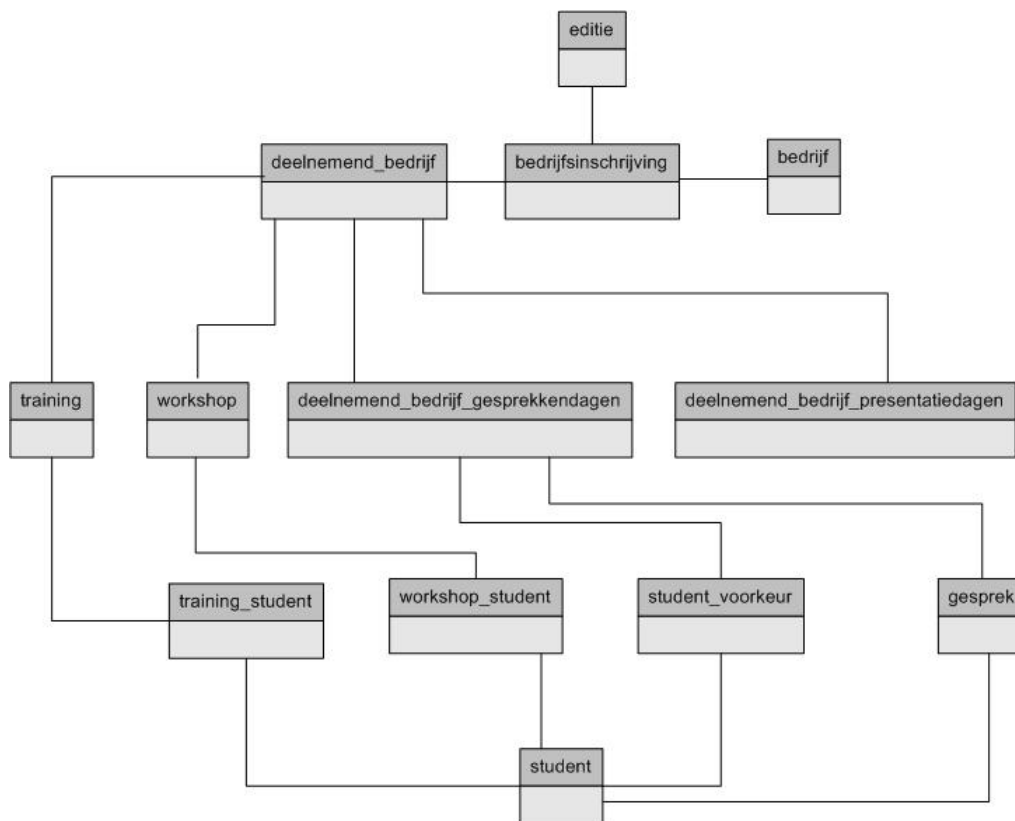
### HIGH LEVEL DESIGN

Hier zullen wij de grote lijnen van het ontwerp van onze database bespreken. Het gedetailleerde ontwerp van onze database kan gevonden worden in de appendix. Het abstractere model wordt getoond in figuur 3.

Zoals zichtbaar is in het model moet een bedrijf zich eerst inschrijven voor een editie van DDB. Daarna kan DDB een deelnemend bedrijf entry maken en met behulp van deze entry kunnen bedrijven zich gaan inschrijven voor de verschillende onderdelen die plaats vinden tijdens een editie van DDB. Studenten zijn, als ze zich aanmelden automatisch deelnemend aan DDB en zij kunnen zich dan ook rechtstreeks inschrijven voor evenementen.

Uit dit high level design zijn enkele elementen weggelaten die wel zichtbaar zijn in het gedetailleerde design:

- Sommige tabellen in de database horen niet bij het data ontwerp maar bij de website, dit zijn tabellen waarin data staat die gebruikt wordt door PHP, zoals de tabellen “String”, “Vraag\_actief” en “Pagina”.



**Figuur 3.** High level design van DDB database

- Eigenlijk bevat een bedrijfsinschrijving voor een evenement, zoals bijvoorbeeld de workshops, dusdanig veel info dat dit uiteindelijk moet worden opgeslagen in meerdere tabellen. Echter voor de overzichtelijkheid worden deze tabellen als één tabel weergegeven.
- Bij een inschrijving voor DDB wordt door bedrijven al info meegegeven die gebruikt gaat worden als een bedrijf daadwerkelijk geselecteerd wordt voor de DDB. Deze data wordt daarom gekoppeld aan de bedrijfsinschrijving en niet aan deelnemendbedrijf. De masterselectie voor de Presentatiedagen is hier een voorbeeld van. Dit is niet zichtbaar in het model in Figuur 3.

#### DATA NORMALISATIE PROCES

Waar mogelijk hebben wij vaak gekozen om onze tabellen in normaal vorm op te stellen. Dit werd gedaan om de integriteit van onze database te bevorderen. In de oude database was het bijvoorbeeld mogelijk om als niet ingeschreven bedrijf mee te doen aan de presentatiedagen. Dit soort situaties kan nu niet meer voorkomen.

In sommige gevallen bleek het helaas niet mogelijk omdat het normalisatie proces zou resulteren in een onoverzichtelijk aantal tabellen. In deze niet genormaliseerde tabellen wordt data opgeslagen met behulp van één of meerdere velden met booleans of integers. Data integriteit moet in dat geval gegarandeerd worden in een hogere laag in het systeem. Deze oplossing hebben wij bijvoorbeeld gekozen voor de tabel “Gesprek”. Deze tabel heeft een boolean die aangeeft of het gesprek al dan niet door gaat en een tekstveld waar de reden voor afmelding kan worden opgegeven. Het leek ons niet nuttig om een extra tabel met afgemelde gesprekken te maken.

### 5.3 Gebruikte technieken en software pakketten

#### POSTGRESQL

Als database engine kiezen we tussen PostgreSQL [19] en MySQL [16], de twee meest gebruikte, gratis beschikbare en welbekende open-source DBMS's. Postgres zien we als de volwassene van de twee. Waar MySQL bijvoorbeeld pas sinds versie 5.x enige ondersteuning heeft voor foreign keys, heeft Postgres deze ondersteuning al veel langer. Postgres biedt ook ondersteuning om op tabelniveau autorisatie toe te kennen aan gebruikers. Dit hebben we nodig voor access control. We hebben ook ervaring met PostgreSQL en MySQL en op basis van de genoemde argumenten en onze ervaring kiezen we voor Postgres versie 8.x (laatste major release).

#### APACHE WEB SERVER

Als web server maken wij gebruik van de Apache web server [7]. Wij hebben gekozen voor deze server omdat hij al draaide op de DDB server en gemakkelijk aan te passen was voor het nieuwe systeem. Daarnaast heeft Apache zich door de jaren heen ruimschoots bewezen met een marktaandeel rond de 50% [8].

#### PHP

Om dynamisch webpagina's te genereren, hebben wij een keuze moeten maken voor een server-side scripting taal. We hebben de keuze uit verschillende talen zoals ASP.NET [6], PHP [17], Ruby on Rails [21] en ColdFusion [14]. Wij willen een taal gebruiken die a) open source is, b) waar wij bekend mee zijn, c) ondersteuning geniet van de huidige ddb server en d) geen financiële investering vereist. Op basis van deze eisen hebben wij gekozen voor de taal PHP.

#### PROPEL

Voor de data access layer naar de website en eventuele andere php applicaties willen we een tool gebruiken om de database vanuit php te kunnen manipuleren zonder SQL queries te hoeven schrijven. Onze zoektocht naar zo een tool heeft ons naar de Object Relational Mapping tool Propel [10] geleid. De keuze voor deze tool is beïnvloed door het feit dat het een open-source tool is, een goede duidelijke website met duidelijke documentatie en voorbeelden heeft en een actieve community.

Propel beeldt de tabellen van de database één op één af op PHP klassen. Dit betekent dat we eigenlijk niet een object georiënteerd ontwerp hebben, maar dat we het data ontwerp een laag



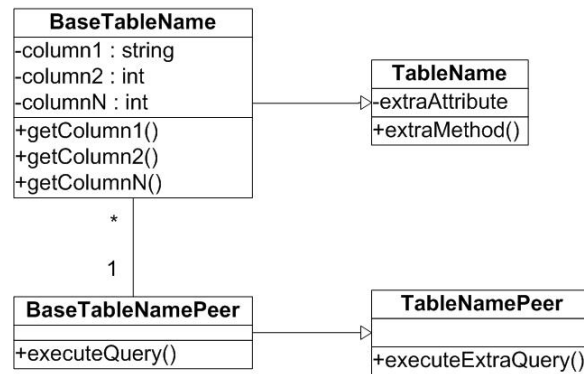
omhoog tillen, naar PHP objecten. Hierdoor hoeven we geen SQL meer te coden in de applicatie laag en dat is het doel van deze laag.

De afbeelding van de tabellen op PHP objecten gebeurt volgens het UML diagram in figuur 4. Propel genereert zelf de objecten nadat een beschrijving van de database is ingevoerd. Er worden per tabel 2 klassen en 2 stubs gegenereerd.

De klasse *BaseTableName* representeert één entry in de tabel *TableName* en heeft alle methodes om in deze entry kolommen waardes toe te kennen. Tevens worden er methodes gegenereerd die gebruikt kunnen worden om gerelateerde entries (foreign keys) uit de database op te kunnen halen.

De complete tabel *TableName* wordt gerepresenteerd door de klasse *BaseTableNamePeer*. Door middel van methodes in deze klasse kunnen SELECT, UPDATE, INSERT en DELETE queries uitgevoerd worden over deze tabel in de database.

De twee gegenereerde stubs extenden respectievelijk de *Base* en de *BasePeer* klasse. De gebruiker kan in deze twee klassen extra functionaliteit toevoegen.



**Figuur 4.** Afbeelding database tabellen op PHP klassen door Propel

## SMARTY

Om de MVC structuur (zie onder apart kopje MVC) die we willen implementeren te ondersteunen, gebruiken we een templating engine. De viewers (in het MVC paradigma) zullen verzorgd worden door templates die de templating engine rendert tot HTML. Er bestaan een paar van dit soort tools, waaronder Smarty [11]. Met Smarty zijn wij bekend en hebben we goede ervaringen. Daarnaast wordt deze tool gesponsord door php.net zelf, heeft Smarty een grote, vriendelijke userbase, duidelijke voorbeelden en documentatie.

## HTML\_QUICKFORM EN FORMLIB

Een vaak terugkomend element in websites zijn HTML forms en hun bijbehorende operaties (submitten, validaten, vullen, showen). Om de onderhoudbaarheid en overzichtelijkheid van onze code te bevorderen, besloten wij om al deze operaties te automatiseren en uit te laten voeren door PHP klassen.

Ons eerste instinct was om hiervoor een bestaand softwareproduct te gebruiken, zodat wij geen tijd en moeite zouden moeten steken in ons eigen ontwerp en implementatie voor zo'n product. Het enige product wat wij konden vinden wat voldeed aan onze specificatie was het open-source HTML\_QuickForm [9].

Nadat wij dit product geïmplementeerd hadden, bleek de HTML door QuickForm op een dusdanige manier gegenereerd te worden dat het onmogelijk was ons CSS schema te integreren. De enige oplossing was om ons eigen product te creëren wat het zelfde zou kunnen doen als QuickForm, maar ook HTML zou kunnen creëren die om zou kunnen gaan met ons CSS schema.

Uiteindelijk hebben wij onze eigen lichtere herimplementatie van QuickForm geschreven, FormLib. Zie Appendix B of de los meegeleverde FormLib API specificatie voor meer informatie over FormLib.

## SUBVERSION

Om met het hele team tegelijkertijd aan de code te kunnen werken, versie beheer te ondersteunen en regelmatig backups van de code te maken hebben wij besloten om een Subversion (SVN) server [12] te installeren op de DDB server en deze te gebruiken voor de code ontwikkeling van het project. Wij hebben gekozen voor deze versiebeheer software omdat wij hier bekend mee waren en het gemakkelijk te installeren is onder Debian.

## MANTIS EN TRAC

Zowel tijdens als na het project zullen er bugs optreden. Om deze bugs gecoördineerd te kunnen documenteren en op te lossen, maken we gebruik van de professionele bugtracker Mantis [15]. In de loop van het project zijn we echter overgestapt op Trac [13] omdat deze tool ook web-based SVN repository browsing biedt en een geïntegreerde Wiki wat mogelijkheden biedt voor online documentatie.

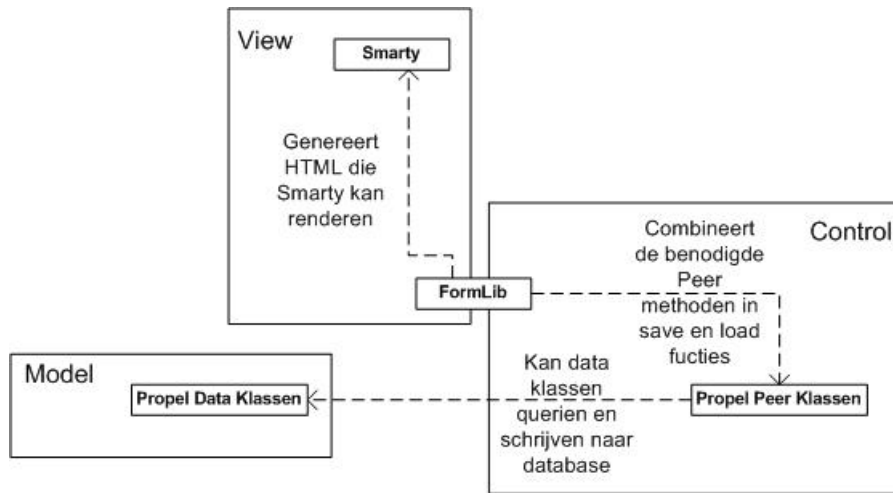
## HAPPY FISH

We maken gebruik van het database modeling programma Happy Fish [18] om onze database creatie code te genereren en te updaten. Dit heeft als voordeel dat wijzigingen in onze database goed overdacht en gedocumenteerd worden. Tevens is er te allen tijde een kloppend grafisch model van de database beschikbaar.

## 5.4 Implementatie van het MVC pattern

Hieronder volgt een beschrijving hoe wij het MVC pattern geïmplementeerd hebben, met de software producten genoemd in 5.3. Het Model gedeelte van MVC wordt gevormd door de klassen die Propel genereert aan de hand van onze database. Control wordt deels gedaan door de verwerk methoden in de afzonderlijke forms en de *Peer* klassen van Propel, deze kunnen data objecten schrijven en opvragen uit de database. Als Viewer gebruiken wij Smarty. Deze tool rendert templates naar HTML. We gebruiken FormLib als lijm tussen deze drie producten. FormLib biedt save en load

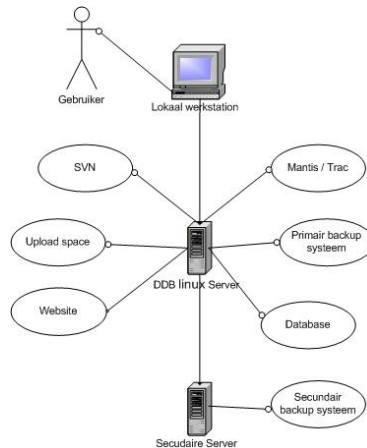
stubs, waar Peer methoden gebruikt kunnen worden om ingevoerde data te verwerken. Tevens kan FormLib HTML genereren die wordt gebruikt door Smarty. Dit wordt geïllustreerd in Figuur 5.



Figuur 5. MVC implementatie in systeem

## 5.5 De mapping van subsystemen naar processoren en componenten

Het systeem van DDB draait in principe op drie verschillende typen hardware nodes waar wij software systemen naar toe mappen. In de eerste plaats hebben wij de linux server waar de website, Mantis/Trac, bedrijven upload ruimte, database, svn en het primaire backup systeem draaien. Ten tweede hebben wij lokale computers waarop recruiters, studenten en bestuurders gebruik maken van het systeem. Ten derde hebben wij een secundair backupsysteem waar wekelijks een versie van de code en de database naar gebackupd worden. Deze mapping wordt geïllustreerd in figuur 6.



Figuur 6. De mapping van subsystemen in het DDB systeem

## 5.6 Het identificeren en opslaan van persistente data

WELKE DATA IS PERSISTENT IN ONS SYSTEEM?

Zoals gemeld in 5.1 heeft ons systeem een ‘data centered design’ dus alle data waarmee gewerkt wordt, wordt gepersisteerd.

WAAR WORDT PERSISTENTE DATA OPGESLAGEN?

Persistente data wordt veelal gezien als een performance bottleneck in systemen, echter performance is niet een heel belangrijk ontwerp doel van ons systeem. Als een bewerking meerdere seconden duurt is dat niet snel een probleem. Hierdoor is een ‘simpele’ oplossing voldoende en zelfs wenselijk en hebben wij besloten om alle persistente data op te slaan in één centrale PostgreSQL database die gebruikt zal worden door alle processen.

VOORKOMEN DATA CORRUPTIE

Een bekend probleem van persistente data is het consistent houden van deze data. Om deze corrosie van data tegen te gaan dwingen wij ons database model af door gebruik te maken van ‘foreign keys’ zodat waarden in tabellen die verwijzen naar andere tabellen altijd naar geldige waarden wijzen.

Het database model is zoveel mogelijk genormaliseerd binnen de grenzen van het praktische. Maar er zijn enkele velden die met een integer verschillende states coderen. Hoewel op deze kolommen column constraints toegepast zijn, zou het mogelijk zijn dat een veld een waarde gaat krijgen die niets representeert in de bovenliggende lagen. Mocht dit in de toekomst optreden dan kunnen we onze Propel klassen uitbreiden met invarianten om dit probleem op te lossen.

## 5.7 Access control

### DE KLASSE AUTH

Zoals gemeld kent het systeem verschillende gebruikers. Om je als gebruiker aan te melden moet je inloggen. Deze login wordt doorgegeven aan de klasse Auth. Deze klasse bepaalt dan via de database of deze login al dan niet bestaat en welk type gebruiker hoort bij deze login en zet deze waarden in de PHP sessie, zodat bij vervolg requests via dezelfde klasse Auth de waarden van de gebruiker kunnen worden opgevraagd.

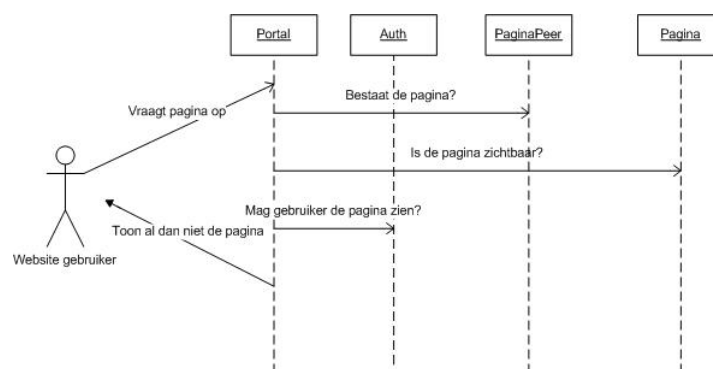
### PORTAL

Alle webpagina's die gebruikers opvragen lopen via een centraal access point. Deze portal voert de volgende controles uit:

- A) Bestaat de opgevraagde pagina in de database? (m.b.v. Propel PaginaPeer)
- B) Heeft de gebruiker toegang tot deze pagina? (m.b.v. Auth)
- C) Is deze pagina reeds zichtbaar? (m.b.v. Propel Pagina)

Indien A, B of C niet geldig zijn dan wordt een nette fout getoond aan de gebruiker. Zijn A en B en C wel geldig dan wordt de opgevraagde content getoond. Figuur 7 illustreert deze sequence of events. Een gebruiker die ingelogd is als DDB'er kan altijd alles zien.

Deze portal initialiseert overigens ook Smarty, Propel en FormLib zodat elke pagina deze libraries gebruiken kan.



Figuur 7. Ophalen van pagina's uit database door portal

## INDIVIDUELE PAGINA'S

Soms is het niet wenselijk een gehele pagina offline te halen, maar slechts een gedeelte van de functionaliteit van een pagina uit te schakelen. Dit wordt in ons systeem geregeld via het *vraag\_actief* mechanisme. Door een *vraag\_actief* te evalueren kan gekeken worden of een stuk code al dan niet moet worden uitgevoerd. Een voorbeeld van dit concept is: “Kunnen de waardes in een form nog veranderd worden of mogen deze alleen nog maar opgevraagd worden”. Zoals aangegeven in de requirements moet het mogelijk zijn om uitzonderingen toe te voegen. Ook deze feature is geïmplementeerd.

## 5.8 Global control flow

### THREAD CONTROL FLOW

Het DDB systeem bestaat (voor de meerderheid van de gebruikers) uit procedures die het mogelijk maken om webpagina's op te vragen en forms te versturen door meerdere gebruikers tegelijk. De meest logische keuze is dan een *Thread control flow* [23]. Om het makkelijker te maken voor de ontwikkelaars delegeren we de taak van thread creatie en beheer naar de webserver.

### CONCURRENT USERS AFHANDELING

Hoewel het systeem geen mutual exclusion ondersteunt, verwachten wij zeer weinig concurrency problem. Het bewijs hiervoor is dat in de afgelopen zes jaar, de architectuur van het systeem (in het opzicht van concurrency) hetzelfde is gebleven en dat er zich in die tijd nooit grote problemen hebben voorgedaan die toegeschreven kunnen worden aan concurrent use.

## 5.9 Boundary conditions

### SYSTEEM INITIALISATIE EN SHUTDOWN

Het starten en stoppen van het systeem komt neer op het starten en stoppen van de web server en database server. Zowel Apache als PostgreSQL hebben hun eigen opstart- en afsluitscripts die standaard worden uitgevoerd bij het opstarten en afsluiten van de fysieke server. Bij onderhoud is het soms nodig handmatig het systeem af te sluiten en opnieuw op te starten. Dit komt dan neer op het uitvoeren van voornoemde scripts. Het geheel duurt minder dan een minuut.

Het kan voorkomen dat de databaseserver afgesloten wordt terwijl er aan het gesprekkenrooster gewerkt wordt of mensen op de website dingen aan het invullen zijn. Het is echter makkelijk tijdstippen te kiezen voor onderhoud waarbij het gebruik van het systeem minimaal is. Daarnaast is er slechts een kleine groep mensen die taken als roosterbewerking kunnen uitvoeren en is met hen gemakkelijk te coördineren wanneer onderhoud zal plaats vinden.

### INCORRECTE USER INVOER

Een van de belangrijkste verantwoordelijkheden van een software ontwikkelaar is dat incorrecte invoer niet leidt tot 'system failures'. In ons systeem ondervangen wij verkeerde invoer in forms met

het intern ontwikkelde FormLib. Elk veld in een form kan bepaalde regels krijgen waaraan de invoer moet voldoen. Mocht de invoer niet voldoen, dan wordt het form automatisch opnieuw getoond aan de gebruiker met foutmeldingen bij de verkeerde invoer zodat hij deze kan corrigeren en opnieuw insturen.

#### SQL INJECTION

Het probleem van SQL injection is een veel voorkomend probleem in database driven systemen [1]. Dit wordt voorkomen in ons ontwerp door het gebruik van Propel. Propel genereert alle queries en controleert alle variabele data opdat geen injecties kunnen voorkomen.

#### CODE CORRUPTION

Mocht een developer een stuk code beschadigen, bijvoorbeeld door een tikfout te introduceren of erger, dan kan de originele code teruggezet worden vanuit SVN. Mocht het SVN repository corrupted raken, dan hebben we wekelijkse on- en off-site backups en in het ergste geval dat we de backups niet kunnen terugzetten, checkouts op de pc's van de ontwikkelaars en de checkout die online staat (de website zoals die nu draait is ook een checkout uit SVN). Dan is de history wel verloren, maar de meest recente versie is bijna altijd nog te redden.

#### DATABASE CRASHES

Het DDB bestuur moet vanwege het flexibele karakter van 'De Delftse Bedrijvendagen' directe toegang hebben tot de database. Dit heeft als gevolg dat verkeerde of gevaarlijke queries op dit systeem uitgevoerd kunnen worden. Om de gevolgen van dit soort crashes te beperken wordt de database ieder kwartier gebackuped op de server. Mocht er een database crash optreden dan kan de database in minimale tijd handmatig hersteld worden. Als de databaseserver crasht, dan zal die automatisch opnieuw opstarten en met behulp van zijn interne log controleren of de laatste transacties goed gegaan zijn en eventueel deze opnieuw uitvoeren om consistentie te garanderen. Mocht dit fout gaan, dan kan handmatig een backup van maximaal een kwartier geleden teruggezet worden.

#### STROOMUITVAL EN HARDWAREPROBLEMEN

Data en code in ons systeem wordt regelmatig gebackuped op een secundair backup systeem. Daarom verwachten wij niet dat hardware crashes zullen leiden tot data verlies. Hardware crashes zullen wel leiden tot downtime, maar door het secundaire backup systeem kunnen we wel het systeem opnieuw installeren op een andere server met minimale downtime. Let op: er is dus geen hot backup systeem dat de plaats van het primaire systeem direct kan overnemen als het fout gaat. Dit is echter een kwestie van geld en niet van techniek.

## 6. Implementatie verslag

In de requirement gathering fase is gebleken dat het inwilligen van alle wensen van DDB te veel werk is om in het kader van het bachelor project uit te voeren. Om een coherent, onderhoudbaar systeem te produceren is voor de ontwerpfase wel alles meegenomen, maar is besloten het bachelor project qua implementatie te laten eindigen bij het bedrijvengedeelte van de website.

### 6.1 Leercurve

Hoewel de individuele technieken die we gebruiken vrij toegankelijk zijn en in sommige gevallen ook bekend, is het toch een moeilijke opgave gebleken om alles te integreren tot een geheel. Vooral alle onderdelen voor de website integreren is moeilijker gebleken dan gedacht werd. Problemen die hierbij spelen zijn bijvoorbeeld de opmaak van de website en het veilig coderen van inlogprocedures.

### 6.2 HTML & CSS integratie

Het bedrijf wat door DDB gebruikt wordt voor grafische zaken, Shibby Projects levert CSS code en voorbeeld HTML voor het nieuwe uiterlijk van de website. Door het gebruik van een templating engine (zie Smarty, p.17) om de HTML te genereren was het gemakkelijk om het ontwerp door te voeren aangezien we door het gebruik van een *master layout template* maar op weinig plaatsen wijzigingen hoefde door te voeren die dan gelijk voor de gehele website zichtbaar werden.

Zoals in paragraaf 5.3 al vermeld, bleek het onmogelijk de CSS toe te passen op de forms van HTML\_QuickForm omdat de HTML die HTML\_QuickForm genereert niet compatibel is met de voorbeeld HTML van Shibby Projects en ook niet gemakkelijk compatibel te krijgen was<sup>1</sup>. Daarop hebben we onze eigen form library geschreven: FormLib. Deze library is een tiende van de size van HTML\_QuickForm, biedt alle functionaliteit die we nodig hebben en is simpeler in opzet en daardoor gemakkelijker te begrijpen. Zie Appendix B of de meegeleverde FormLib API specificatie voor meer informatie over FormLib.

### 6.3 Testen en kwaliteitscontrole

Om de kwaliteit van het geleverde systeem te garanderen hebben wij het grondig getest. In deze sectie rapporteren wij hoe wij het systeem getest hebben.

#### PROPEL

Een van de fundamentele aannames in ons systeem is dat de Data Access Layer correct is, dat Propel goed werkt. Dit hebben wij getest door basis SQL queries te draaien op onze database en deze te vergelijken met de Propel variant die dezelfde query beschrijft. Hier is uitgekomen dat voor onze doeleinden Propel goed werkt.

---

1. HTML\_QuickForm genereert forms die voor layout in een HTML <TABLE> attribuut zitten. Wij hebben tabelloze layout nodig zodat CSS gemakkelijker en vrijer de layout kan aanpassen.



## FORMLIB

De form library is een relatief klein stuk software en het is gemakkelijk alle mogelijkheden te overzien en te testen. We hebben hiervoor een voorbeeld of test PHP pagina gemaakt met daarop een form dat alle mogelijkheden uitput, dat wil zeggen, elke control wordt een keer gebruikt. Als dit form gesubmit wordt, laat het zien wat na verwerking gesubmit is. Dit kan vergeleken worden met de door de tester ingevoerde testdata.

## WEBSITE

De website heeft qua functionaliteit drie aspecten die getest moeten worden:

- authenticatie en autorisatie,
- de verschillende forms en
- het publieke informatieve gedeelte.

Alle authenticatie en autorisatie wordt geregeld door de entry point van de webapplicatie: de portal (zie 5.7 onder portal). Het is dus relatief gemakkelijk dit te testen. We hebben alle randgevallen bekeken. Hier zijn enkele cosmetische fouten uitgekomen en die zijn gecorrigeerd.

De verschillende forms worden na goedkeuring opgeslagen in de database. Wanneer de gebruiker teruggaat naar de pagina, zal hij zien welke waarden hij volgens het systeem ingevuld heeft; die worden dan namelijk weer uit de database opgehaald en getoond. Het volstaat daarom om per form testdata in te vullen, te submitten, het form opnieuw te openen en dan te kijken of de waarden in het form overeenkomen met de ingevulde testdata. Als dit niet het geval is, gaat er iets fout bij het opslaan, ophalen of tonen van de waarden. Omdat dit per veld gaat en de form code door gebruik van de form library erg compact is, is het gemakkelijk fouten op te sporen en te corrigeren.

Aan het publieke gedeelte, dat gedeelte van de website dat eenieder kan benaderen zonder in te loggen, valt weinig te testen aangezien er geen interactie met de gebruiker plaatsvindt op deze pagina's. Wel is samen met het bestuur van DDB gekeken of alle benodigde pagina's aanwezig zijn en of die de juiste informatie bevatten. Overigens kan het DDB bestuur de teksten zelf aanpassen zonder interventie van ons.

## CONVERSIE DATA VOORGAANDE EDITIES

De contactinformatie van bedrijven die al in het bestand van DDB staan, is overgenomen in de nieuwe database. Het testen van dit punt is uit handen gegeven aan stichting 'De Delftse Bedrijvendagen' omdat zij aan het begin van het jaar alle bedrijven nabelt om te controleren of de gegevens nog kloppen voordat de uitnodigingen verstuurd worden. Uiteindelijk zijn de resultaten van deze acceptatietest bij mondelinge overeenkomst goedgekeurd door het bestuur van DDB.

## 7. Evaluatie

In deze sectie evalueren we ons project. We bespreken de behaalde resultaten, de gebruikte third party software oplossingen en de samenwerking met Shibby Projects en het bestuur van DDB.

### 7.1 Behaalde resultaten

#### OPGELOSTE TEKORTKOMINGEN

De in sectie 3.2 besproken tekortkomingen zijn nog niet allemaal opgelost. Dit is voornamelijk te wijten aan het feit dat de meeste van deze requirements betrekking hebben op het studentengedeelte van de website en pas in januari af hoeven te zijn. Echter, de basis is gelegd zodat deze nu snel geïmplementeerd kunnen worden.

Wel is het inmiddels mogelijk de functionaliteiten van een pagina gedeeltelijk te beperken door velden uit forms uit te schakelen. Het is nog niet mogelijk om eenvoudig nieuwe pagina's aan te maken.

Van de niet-functionele tekortkomingen is een groot gedeelte inmiddels aangepakt:

- De problemen veroorzakende ODBC koppeling is verdwenen uit het systeem en alleen read-only beschikbaar gebleven tot het nieuwe systeem alle functionaliteiten volledig biedt.
- De wildgroei van code wordt tegengegaan doordat alle code nu één goed gedocumenteerd ontwerp en één architectuur volgt en doordat het databaseontwerp goed is vastgelegd. Tevens zijn tijdens het project de requirements van de afgelopen jaren en nabije toekomst geïndexeerd, waardoor we weinig nieuwe eisen aan het systeem verwachten.
- De veldnamen van de database zijn in de meeste gevallen zelfbeschrijvend. Bij mogelijke twijfel is commentaar bij het veld geplaatst met uitleg. Deze commentaren zijn in het model opgenomen, waardoor ze ook bij nieuwe versies van de database zullen blijven bestaan. Daarnaast is het databasemodel zo veel mogelijk in normaalvorm gehouden, waardoor we dataredundantie sterk beperken ten opzichte van het oude systeem.
- Het nieuwe systeem is geheel in PHP5 geschreven, waarmee vrijwel alle Visual Basic code is vervangen.
- Op basis van een 'Model-View-Controller'-model zijn de data-, logica- en presentatielagen van elkaar gescheiden, zodat deze eenvoudiger afzonderlijk van elkaar aangepast kunnen worden.

Een niet-functionele tekortkoming die nog aangepakt moet worden, is de ondoorzichtigheid van de 'C.V. engine'. Deze viel echter buiten de scope van dit project en zal later dus nog aangepakt worden.

#### REQUIREMENTS

Tijdens de analyse (hoofdstuk 3) is een lijst requirements opgesteld, waaraan het systeem moet voldoen. Tijdens de evaluatie is gekeken aan welke van deze requirements tegemoet gekomen is. Het

systeem is een week voordat het online zou gaan opgeleverd aan het DDB bestuur, zodat deze alle functionaliteiten na konden lopen en aangeven of alle requirements naar hun zin geïmplementeerd waren. Hier zijn enkele kleine verzoeken uit naar voren gekomen, die niet in de oorspronkelijke requirements stonden. Het systeem is na enige kleine aanpassingen geaccepteerd door het DDB bestuur en online gezet.

Tabel 7.1 toont statistieken over de geïmplementeerde requirements. Zoals gezien kan worden, zijn alle *Must Haves* die in september klaar moesten zijn op één na af (of voldaan in het geval van een niet-functionele requirement). De niet geïmplementeerde requirement bleek verkeerd geclassificeerd te zijn en wordt waarschijnlijk in januari geïmplementeerd.

Type requirement	Aantal	Af of Voldaan
Compleet project	104	32
Requirements met deadline in september	54	32
Must Have requirements met deadline in september	24	23
Should Have requirements met deadline in september	15	7
Could Have requirements met deadline in september	11	2
Won't Have requirements met deadline in september	4	0

**Tabel 1.** Overzicht geïmplementeerde requirements

Opgemerkt wordt dat hier zowel de functionele als niet functionele requirements meegeteld zijn. Zoals het systeem er nu staat, komt het alle niet functionele requirements voldoende na. Het is de bedoeling hieraan te blijven voldoen terwijl de rest van de functionele requirements geïmplementeerd worden in de komende maanden.

## 7.2 Third party tools en software

Wij zijn erg tevreden over de gebruikte tools Mantis en later Trac, Propel en Smarty die wij gebruikt hebben gedurende dit project. Hoewel het tijd en moeite kost om deze tools onder de knie te krijgen zijn de voordelen dusdanig groot dat wij in toekomstige projecten eerst zullen gaan kijken of er reeds een tool bestaat voordat we zelf gaan coden.

HTML\_QuickForm heeft helaas teleurgesteld, dat hebben we opgelost met onze eigen library FormLib (zie appendix B).

## 7.3 Samenwerking met Stichting ‘De Delftse Bedrijvendagen’

Een belangrijk aspect van het bouwen van een systeem is goede kennis van de organisatie en het begrijpen van de wensen van de opdrachtgever. Zowel Bas, Ronald als Steffan hebben in het bestuur van DDB gezeten, Ronald in 2003-2004, Bas in 2005-2006, Steffan in 2006-2007. Daarom zijn we zeer bekend met de organisatie van DDB. Deze heeft ons geholpen in het stroomlijnen van de communicatie met DDB. Dit komt bijvoorbeeld goed tot uiting bij het overleggen over de requirements: omdat we het project kennen, weten we waar het bestuur mee bezig is en waar het bestuur het over heeft als ze een bepaalde requirement voorleggen. Het brengt natuurlijk ook het gevaar mee te veel

beslissingen zelf te nemen omdat we “wel weten hoe het werkt”. Maar omdat we vooral de bestaande functionaliteit netter implementeren heeft dit volgens ons geen problemen opgeleverd.

#### **7.4 Samenwerking met Shibby Projects**

Shibby Projects (Shibby) is de huis-grafische ontwerper van DDB. Shibby is al sinds ongeveer 2002 betrokken bij DDB. Het is een klein bedrijf waar we een vaste contactpersoon hebben in Martijn van der Heijden. Om de website van DDB een professionele uitstraling te geven hebben we Shibby nu voor het eerst ingeschakeld om ook het ontwerp van de website te maken.

Aangezien Martijn zelf de ontwerper is, hadden wij verwacht efficiënt en snel met Shibby te kunnen communiceren. In het begin van het project was dit echter niet het geval. Wij hadden contact met René Pinggen van DDB, die had contact met Jon Reijneveld van DDB, die op zijn beurt contact onderhield met een collega van Martijn van Shibby, die op zijn beurt Martijn aanstuurde.

Dit was onwerkbaar en zodra dit aan het licht kwam, heeft Ronald het contact met Shibby overgenomen en voor het verdere verloop van het project direct met Martijn gecommuniceerd. Het voordeel hiervan was dat het contact nu direct van de ene naar de andere implementerende partij liep en niet meer via niet-technici zodat snel duidelijk was wat op implementatiegebied van elkaar moest en kon worden verwacht.

## 8. Conclusie

In dit document hebben wij de herimplementatie van de automatisering van stichting ‘De Delftse Bedrijvendagen’ beschreven.

Op de omvang van het project hebben we ons verkeken. Daarom is niet de volledige set van requirements opgeleverd, maar hebben we ons beperkt tot het bedrijvengedeelte van de website en de backend. Zoals in de evaluatie te lezen is, zijn alle requirements die als *Must Haves* geclassificeerd zijn voor het bedrijvengedeelte en de backend, geïmplementeerd. Daarnaast zijn de helft van de *Should haves* en twee van de *Could haves* geïmplementeerd.

Hoewel dit verslag een eerste aanzet is voor complete documentatie van het systeem zijn er nog steeds enkele features die nog niet gedocumenteerd zijn. Dit zal in de loop der maanden gebeuren.

De eerste ontvangst van het product is door zowel DDB als bedrijven als zeer positief ervaren. Wij verwachten dat de nog te implementeren requirements geen grote problemen meer zullen opleveren aangezien nu voldoende ervaring aanwezig is met de gebruikte tools. In de komende maanden zullen de overige *Must Have* requirements geïmplementeerd worden.

# Verklarende woordenlijst

**Bedrijvenboek** Boekwerk dat elke editie uitgegeven wordt met informatie over alle deelnemende bedrijven. De studenten gebruiken het bedrijvenboek om te kiezen welke bedrijven ze benaderen tijdens DDB.

**BIT** Het Bedrijven Informatie Team assisteert het bestuur van DDB tijdens de Presentatiedagen en maakt het bedrijvenboek.

**DDB** Stichting ‘De Delftse Bedrijvendagen’ organiseert jaarlijks vier evenementen, de Sollicitatietraining, de Presentatiedagen, de Workshops en de Gesprekkendagen, om studenten en bedrijven in contact te brengen met elkaar.

**Gesprekkendagen** Vierde evenement van DDB. Studenten vullen online hun C.V. in en kunnen gesprekken aanvragen met bedrijven. Bedrijven kiezen daarop de studenten die ze willen spreken. Wanneer beide partijen bevestigen, wordt een gesprek gepland. De Gesprekkendagen duren afhankelijk van het aantal gesprekken doorgaans tien dagen. In deze tijd worden in tien tot twintig ruimtes ongeveer 500+ gesprekken gevoerd.

**iDeal** Online betalingssysteem van de Nederlandse banken ABN Amro, Postbank, Fortis, Rabobank en SNS Bank.

**NetID** “Algemene authenticatiemiddel van de TU Delft”, zie <http://netid.tudelft.nl>)

**Presentatiedagen** Tweede evenement van DDB. Drie daags evenement in de aula van de TU Delft waar 100+ bedrijven zich presenteren met een stand en eventuele bedrijfspresentatie aan 1000+ studenten.

**Recruiter** Alle medewerkers die namens een bedrijf met DDB te maken hebben.

**Sollicitatietraining** Twee daags evenement waar een groeiend aantal recruitmentbureau’s en consultants trainingen geven met betrekking tot het sollicitatieproces.

**W.I.S.V. ‘Christiaan Huygens’** Wiskunde en Informatica Studievereniging ‘Christiaan Huygens’ is de studievereniging voor wiskunde en informatica aan de faculteit Elektrotechniek, Wiskunde en Informatica van de Technische Universiteit Delft.

**Workshops** Derde evenement van DDB. Bedrijven kunnen een datum kiezen in maart om een workshop te organiseren op locatie bij het bedrijf. Bedrijven kunnen er ook voor kiezen een workshop in Delft te organiseren. DDB verzorgt dan een ruimte.

# Bibliografie

- [1] S. Boyd and A. Keromytis. Sqlrand: Preventing sql injection attacks, 2004.
- [2] Bernd Bruegge and Allen A. Dutoit. *Object-Oriented Software Engineering; Conquering Complex and Changing Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.
- [3] Alan Dennis and Barbara Haley Wixom. *Systems Analysis and Design in Action*. John Wiley & Sons, Inc., New York, NY, USA, 1999.
- [4] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Design patterns: Abstraction and reuse of object-oriented design. *Lecture Notes in Computer Science*, 707:406–431, 1993.
- [5] Ian Gilfillan. Postgresql vs mysql: Which is better?, 2003.
- [6] <http://asp.net/>. Asp.net 2.0.
- [7] <http://httpd.apache.org/>. The apache http server.
- [8] [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html). Netcraft: Web server survey archive.
- [9] [http://pear.php.net/HTML\\_QuickForm](http://pear.php.net/HTML_QuickForm). Html\_quickform html form library.
- [10] <http://propel.phpdb.org/>. Propel object relational mapping tool.
- [11] <http://smarty.php.net/>. Smarty templating engine.
- [12] <http://subversion.tigris.org/>. Svn version control.
- [13] <http://trac.edgewall.org/>. The trac project.
- [14] <http://www.coldfusionplanet.com/>. Coldfusion web development tool.
- [15] <http://www.mantisbt.org/>. Mantis bug tracker.
- [16] <http://www.mysql.com/>. Mysql database management system.
- [17] <http://www.php.net/>. Php scripting language.
- [18] <http://www.polderij.nl/happyfish/>. Happy fish database modeling tool.

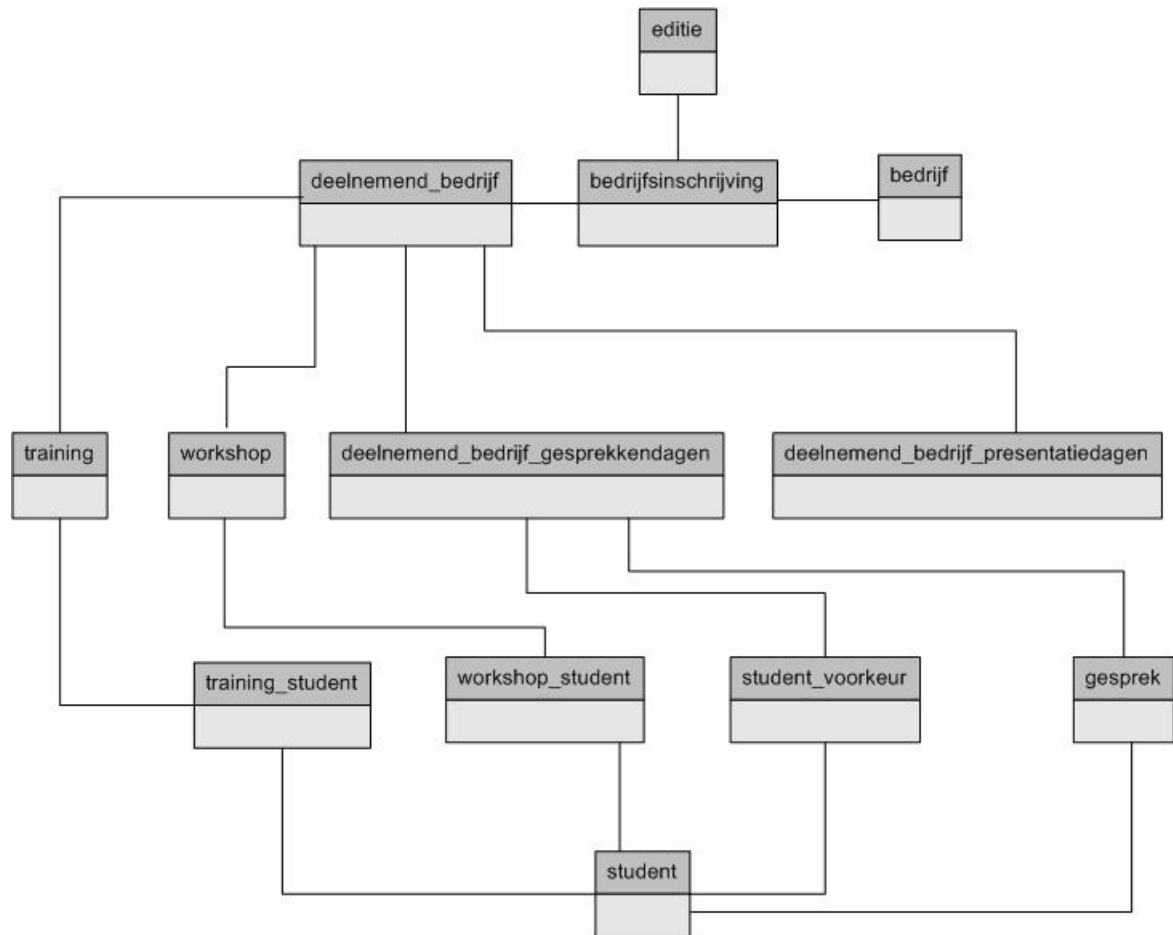


- [19] <http://www.postgresql.org/>. PostgreSQL database management system.
- [20] <http://www.quirksmode.org/css/forms.html>. Css 2 - tableless forms.
- [21] <http://www.rubyonrails.org/>. Ruby on rails framework.
- [22] Brian Jepson. PostgreSQL vs. mysql, 2001.
- [23] Timothy Lethbridge and Robert Laganriere. *Object-Oriented Software Engineering: Practical Software Development using UML and Java*. McGraw-Hill, 2002.
- [24] Ph.D. Steve Burbeck. Applications programming in smalltalk-80(tm): How to use model-view-controller (mvc).



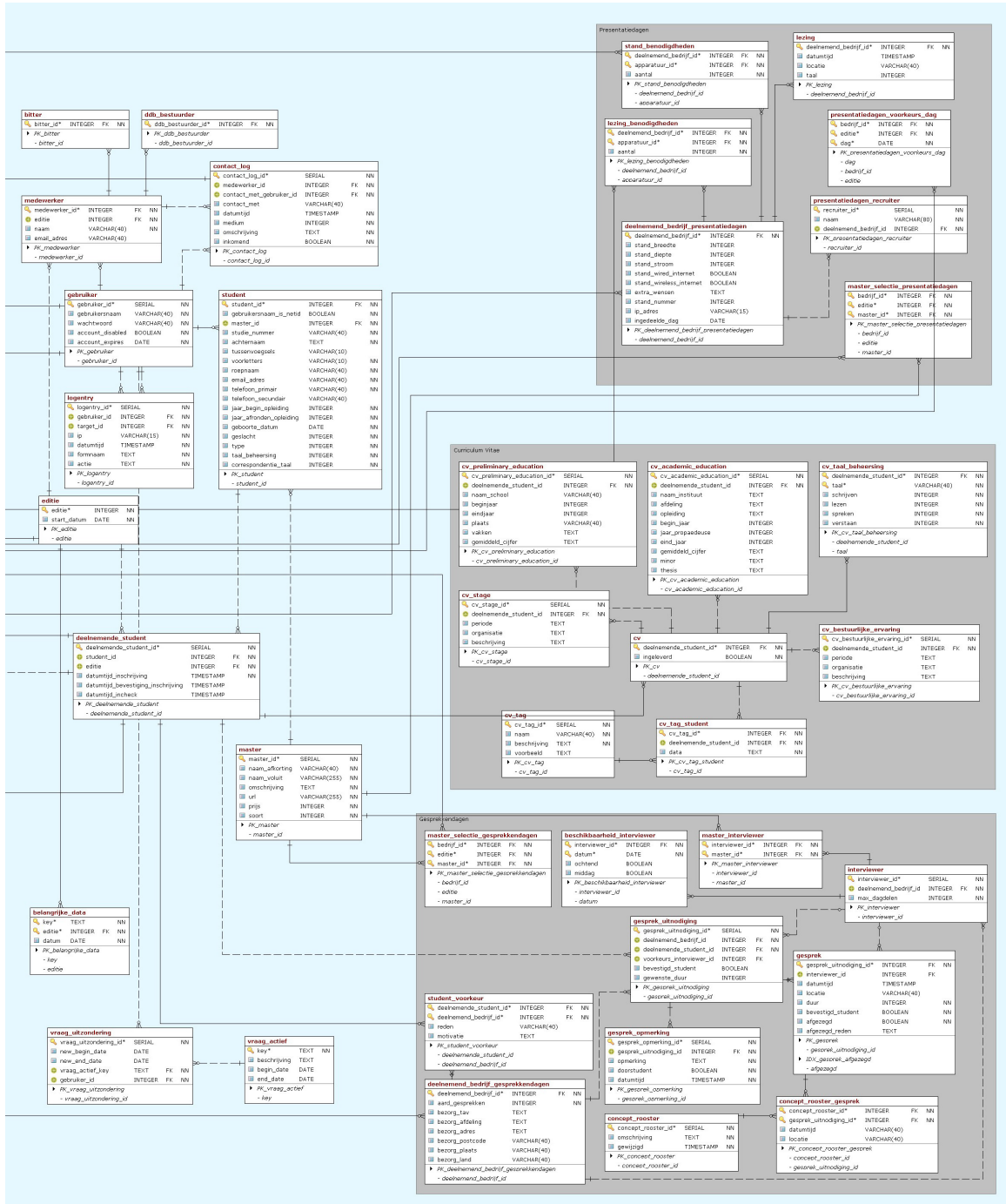
## Appendix A. Database model

Op de volgende twee pagina's staat gedetailleerde schema van het database model van de DDB site. Ter verduidelijking geven wij ook nog het abstracte model, voor het eerst gepresenteerd in 5.2.



**Figuur 8.** High level design van DDB database





Figuur 10. Database model systeem 'De Delftse Bedrijvendagen' - deel 2 van 2

## Appendix B. FormLib

FormLib is een minimalistische library die het vergemakkelijkt HTML forms te ontwerpen, te verwerken en te tonen. In tegenstelling tot `HTML_QuickForm` [9] genereert FormLib tableless forms [20] die daardoor makkelijker en veel vrijer te layouten zijn met behulp van CSS.

De library bestaat uit een Validator en een tree van klassen die HTML elementen voorstellen. Deze hebben allen `Element` als superclass. In figuur 11 is in een uitgebreid klassediagram te zien hoe de verschillende klassen zich relateren tot elkaar en welke methoden beschikbaar zijn.

Een complete API specificatie van FormLib wordt meegeleverd in een apart document en is in te zien op de development website van DDB<sup>2</sup>.

Een 'hello world' form wordt als volgt gemaakt:

```
<?php
require_once 'formlib.php';

class HelloWorldForm extends Form {
    public __construct() {
        super('helloworldform');
    }
public function loadData() {
    return array();
}
public function saveData($values) {
    // called after successful validation
    echo ''Hello world!'';
}
}

$form = new HelloWorldForm();
$form->addElement(new TextField('myTextField'));
$form->addRule('myTextField', 'required');

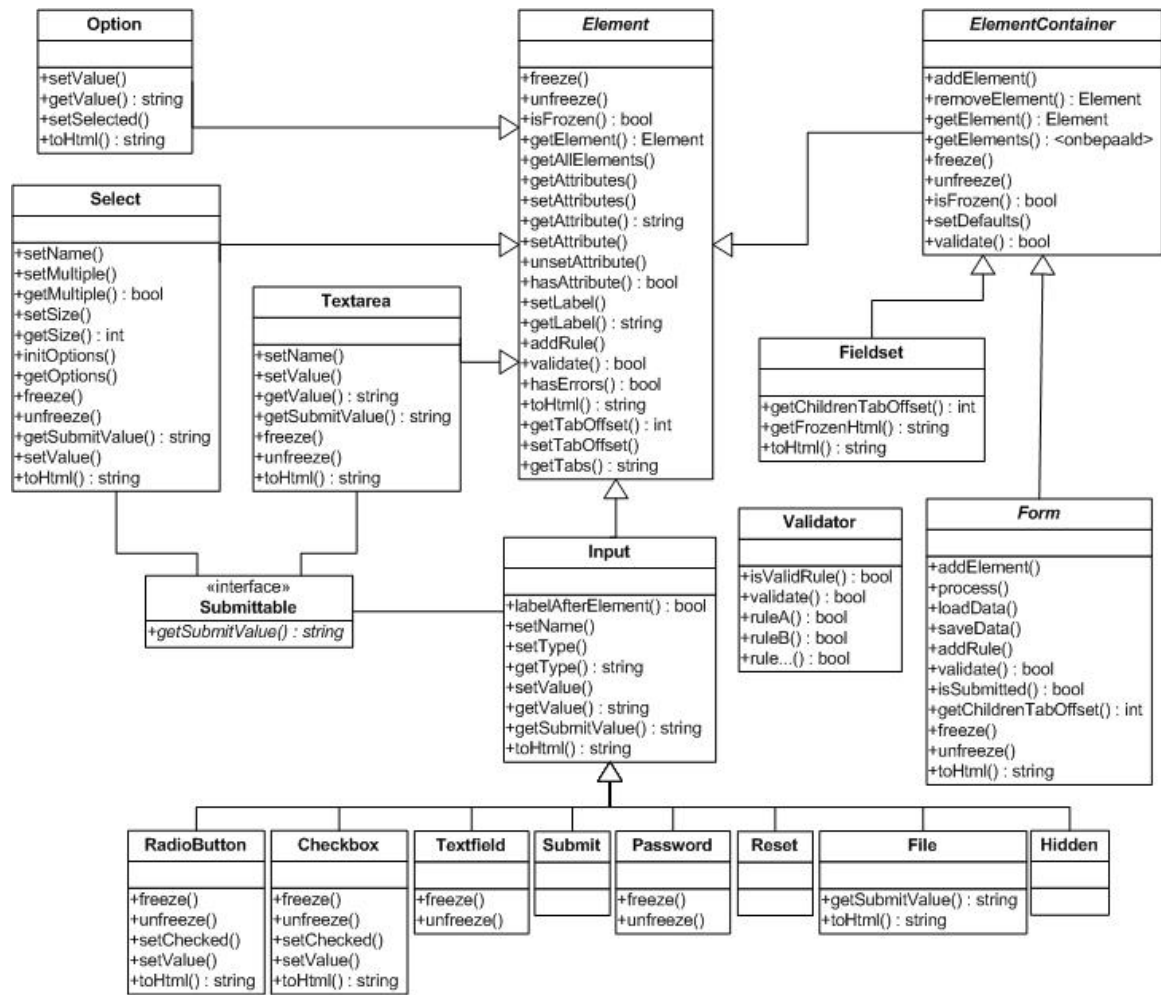
if ($form->isSubmitted() && $form->validate()) {
    $form->saveData();
}

echo '<html><head></head><body>' . $form->toHtml() . '</body></html>';
?>
```

---

2. URL: <https://ddb.tudelft.nl:8081/trac> - hiervoor is speciale toegang vereist.

Figuur 11. Klasse Diagram van FormLib



## Appendix C. Requirements

### Requirement 1.1.1

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Must have	sep-07	Af
Beschrijving:				
De gehele website dient tweetalig uitgevoerd te worden.				

### Requirement 1.1.2

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Should have	sep-07	Af
Beschrijving:				
Alle teksten op de website moeten eenvoudig via de website gewijzigd kunnen worden door het bestuur van Stichting De Delftse Bedrijvendagen.				

### Requirement 1.1.2.1

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Could have	sep-07	Open
Beschrijving:				
Wanneer een meermaals gebruikte tekst wordt aangepast, wordt hier een waarschuwing voor gegeven bij het aanpassen.				

### Requirement 1.1.3

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Should have	sep-07	Af
Beschrijving:				
Elke actie van elke gebruiker die een verandering in de staat van de database teweegbrengt, dient in een log bijgehouden te worden. Opgeslagen wordt: user, datum/tijd, ip, wijziging/actie.				

### Requirement 1.1.3

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Should have	sep-07	Af
Beschrijving:				
Het dient mogelijk te zijn gemakkelijk gegevens en statistieken over voorgaande jaren te achterhalen dan wel te aggregeren.				
Opmerkingen:				
Gerealiseerd door het mogelijk te maken dezelfde database te gebruiken voor meerdere edities.				



**Requirement 1.1.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Should have	sep-07	Open

---

## Beschrijving:

Het moet mogelijk zijn bepaalde delen van invoerformulieren uit te schakelen afhankelijk van een datum / tijd.

---

**Requirement 1.1.6**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Could have	dec-07	Open

---

## Beschrijving:

Notificatiesysteem dat berichten kan versturen via SMS, email of posts op de website.

---

**Requirement 1.1.7**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Could have	sep-07	Ontwerp gereed

---

## Beschrijving:

Todo / deadline systeem voor bedrijven en studenten. De todo's en deadlines moeten altijd zichtbaar zijn als is ingelogd. Bedrijf of student moet op een todo / deadline kunnen klikken om naar het scherm te gaan waar ze de acties kunnen uitvoeren die nodig zijn voor de todo of deadline. Het systeem moet ook 'deadlines' voor het DDB bestuur weer kunnen geven. Denk aan het bekendmaken van de standindeling of het gesprekkenrooster. Deze deadlines moeten naar keuze bij studenten of bedrijven in het overzicht komen.

---

**Requirement 1.1.7.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Niet actor specifiek	Could have	sep-07	Ontwerp gereed

---

## Beschrijving:

Op een deadline kunnen uitzonderingen worden gemaakt. Zowel studenten als bedrijven moeten deze kunnen krijgen.

---

**Requirement 1.2.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

## Beschrijving:

Niet ingelogde studenten kunnen de volgende informatie opvragen op de website:

- Lijst met deelnemende bedrijven
  - Details van de deelnemende bedrijven (bedrijfsprofiel, etc)
  - Contactinformatie van Stichting De Delftse Bedrijvendagen
-

**Requirement 1.2.1.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Should have	dec-07	Ontwerp gereed

---

Beschrijving:

Bedrijven die de master van een student geselecteerd hebben, worden benadrukt zodat deze voor de student gemakkelijk te onderscheiden zijn.

---

**Requirement 1.2.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

TU Delft studenten en promovendi kunnen inloggen met hun NetID, aspirant deelnemers die geen NetID hebben dienen ook te kunnen deelnemen indien ze zijn uitgenodigd.

---

**Requirement 1.2.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

De in 1.2.2 genoemde partijen dienen zich via de website in te kunnen schrijven voor De Delftse Bedrijvendagen. Hierbij worden de volgende gegevens gevraagd (te verifiëren):

- Personalialia (aan te passen via NetID)
  - Adresgegevens (vervallen, postadres niet van belang)
  - Geboortedatum (afkomstig uit NetID)
  - Geslacht (afkomstig uit NetID)
  - Beginjaar studie
  - Verwacht jaar van beëindigen opleiding
  - Telefoonnummers (primair telefoonnummer verplicht, secundair optioneel)
  - Emailadres (afkomstig uit NetID, wel aan te passen)
  - Studienummer (afkomstig uit NetID)
  - Master / richting
  - Student type (Bsc, Msc, Phd)
  - Taalbeheersing
  - Indien taalbeheersing zowel Nederlands als Engels, correspondentietaal
- 

**Requirement 1.2.3.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Na het inschrijven wordt een bevestigingsmail verzonden, waarin de inschrijfggegevens vermeld worden. Om de inschrijving definitief te maken, dient men op een link in deze mail te klikken. Hiermee wordt het emailadres geverifieerd.

---

**Requirement 1.2.3.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

## Beschrijving:

Na het bevestigen van de inschrijving kan de student betalen. Het bedrag van inschrijving kan afhankelijk zijn van de inschrijfdatum en kan geschieden via iDeal, creditcard of contant. Eventueel is het ook mogelijk de betaling uit te stellen tot een ander moment. Contante betalingen worden aan de balie bij DDB verricht, deze kunnen door DDB medewerkers ingevoerd worden.

---

## Opmerkingen:

De mogelijkheid tot betalen met creditcard heeft een lage prioriteit.

---

**Requirement 1.2.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

## Beschrijving:

Over het verloop van de inschrijfprocedure worden de volgende gegevens bijgehouden:

- Moment van aanmelden
  - Moment van bevestiging aanmelding (per email)
  - Status betaling
  - Inchecktijdstip
- 

**Requirement 1.2.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

## Beschrijving:

Aanmelden voor de onderdelen van DDB is enkel mogelijk wanneer betaald is.

---

**Requirement 1.2.6**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

## Beschrijving:

Studenten kunnen hun gegevens aanpassen via NetID, de volgende gegevens kunnen echter aangepast worden. In dat geval worden de gegevens uit de eigen database gebruikt.

- Telefoon
  - Email
  - Jaar van begin studie / doctoraal
  - Verwacht jaar van afstuderen / promoveren
  - Master / richting
  - Student type (Bsc, Msc, Phd)
  - Taalbeheersing
  - Taalinstellingen
-

**Requirement 1.3.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Voor bedrijven is het mogelijk op de website informatie te vinden over De Delftse Bedrijvendagen, zonder in te loggen.

---

**Requirement 1.3.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Bedrijven kunnen inloggen met een gebruikersnaam en wachtwoord welke aan het begin van elk jaar toegezonden worden. Wanneer dit wachtwoord vergeten wordt, kan dit naar het emailadres van de hoofdcontactpersoon verzonden worden via een ‘wachtwoord vergeten’ link.

---

Opmerkingen:

De ‘wachtwoord vergeten’ link heeft lage prioriteit.

---

**Requirement 1.3.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Na ingelogd te zijn, kunnen bedrijven zich inschrijven voor De Delftse Bedrijvendagen, of aangeven dat ze afzien van deelname.

---

**Requirement 1.3.3.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Should have	sep-07	vervallen

---

Beschrijving:

Wanneer een bedrijf afziet van deelname, kunnen ze hiervoor een motivatie opgeven.

---

**Requirement 1.3.3.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

## Beschrijving:

Wanneer een bedrijf aangeeft deel te willen nemen aan De Delftse Bedrijvendagen, wordt om de volgende gegevens gevraagd:

- Naam bedrijf zoals in correspondentie naar bedrijf
  - Naam bedrijf zoals die naar studenten toe gebruikt zal worden
  - Gegevens van de hoofdcontactpersoon
  - Keuze tot deelname aan de Sollicitatietraining (in geval van een recruitmentbureau, het aantal trainingen dat het bedrijf wil geven)
  - Voorkeursdagen voor de Presentatiedagen (niet voor recruitmentbureau's)
  - Voorkeur tot lezing tijdens presentatiedagen (niet voor recruitmentbureau's)
  - Selectie van de masters waarin het bedrijf geïnteresseerd is
  - Voorkeur met betrekking tot het type workshop dat een bedrijf wil geven (geen, in Delft of op locatie)
  - Voorkeur met betrekking tot deelname aan de Gesprekkendagen
  - Indien bedrijf wil deelnemen aan Gesprekkendagen een selectie van de masters waarin het bedrijf geïnteresseerd is studenten uit te spreken.
- 

**Requirement 1.3.3.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Could have	sep-07	Ontwerp gereed

---

## Beschrijving:

Na de inschrijving wordt een overzicht van de gekozen optie getoond en verstuurd per email. Onderdeel van de email is een bevestigingslink, waarmee het emailadres gecontroleerd wordt. Een inschrijving is pas definitief wanneer via de link het emailadres bevestigd is. Het getoonde overzicht is zo weergegeven dat dit eenvoudig uitgeprint kan worden.

---

**Requirement 1.3.3.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Could have	sep-07	Af

---

## Beschrijving:

Als het bedrijf al eerder deelgenomen heeft aan DDB, worden de contactgegevens van de oude inschrijving overgenomen. Deze kunnen dan verbeterd en/of aangevuld worden als dat nodig is.

---

**Requirement 1.3.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Een ingelogd bedrijf kan de volgende gegevens wijzigen:

- Contactpersonen en bijbehorende contactgegevens
  - BTW nummer en PO nummer
  - Voorkeuren mbt de verschillende evenementen
- 

**Requirement 1.3.4.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Should have	sep-07	Ontwerp gereed

---

Beschrijving:

Bij aan- of afmelding van een van de evenementen moet er een notificatie-email naar het DDB bestuur verstuurd worden.

---

**Requirement 1.3.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Een ingelogd bedrijf kan de volgende informatie vinden:

- Overzicht kosten deelname (volgens opgegeven voorkeuren)
  - Frequently Asked Questions
- 

**Requirement 1.3.6**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Ontwerp gereed

---

Beschrijving:

Een bedrijf kan van de volgende typen zijn:

- Regulier bedrijf
  - Recruitmentbureau
- 

Opmerkingen:

De recruitmentbureauoptie is mogelijk, maar niet gebruikt, het DDB bestuur heeft aangegeven voorlopig af te willen zien van deze opsplitsing. De mogelijkheden zijn wel opgenomen in het ontwerp en de database, maar er is geen code die functionaliteit hiervoor biedt geschreven.

---

**Requirement 1.4.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	sep-07	Gedeeltelijk af

---

## Beschrijving:

Via een 'instellingen' scherm moeten allerlei data instelbaar zijn. Denk aan de data van de evenementen, het maximale aantal recruiters per bedrijf op de presentatiedagen.

---

## Opmerkingen:

Alleen de mogelijkheid tot het invoeren van speciale datums is geïmplementeerd.

---

**Requirement 1.4.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Should have	sep-07	Open

---

## Beschrijving:

Elke email die de website automatisch verstuurt (bevestiging van aanmelding, wachtwoord vergeten voor bedrijven en dergelijke), moet via één interface op de website aanpasbaar zijn.

---

**Requirement 1.4.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	dec-07	Open

---

## Beschrijving:

Het moet mogelijk zijn in te stellen wanneer en via welk medium automatische notificaties / herinneringen (bijvoorbeeld een sms aan studenten de dag voor een workshop) verstuurd worden en hiervoor de templates creëren en aanpassen.

---

**Requirement 1.4.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	sep-07	Af

---

## Beschrijving:

Alle correspondentie die gegenereerd wordt kan in zowel Engels als Nederlands gegenereerd worden.

---

**Requirement 1.4.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	sep-07	Ontwerp gereed

---

## Beschrijving:

Het DDB bestuur moet een lijst van ingeschreven bedrijven kunnen oproepen. In deze lijst kan zij dan bedrijven accepteren dan wel afwijzen.

---

**Requirement 1.4.6**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	sep-07	Open

---

Beschrijving:

Een aantal jaarlijks terugkerende brieven/mails moeten via de website gegenereerd kunnen worden, onder andere:

- Uitnodiging bedrijven
  - Bevestigingsbrief geaccepteerde bedrijven
  - Afwijzingsbrief naar niet geaccepteerde bedrijven
  - Contracten bedrijven
  - Facturen bedrijven (en bijbehorende herinneringen / aanmaningen)
- 

**Requirement 1.4.7**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	sep-07	Ontwerp gereed

---

Beschrijving:

De templates voor correspondentie (zie lid 1.4.6 voor voorbeelden) dienen via de website te kunnen worden aangepast door het DDB bestuur.

---

**Requirement 1.4.8**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Would like to have	sep-07	Open

---

Beschrijving:

Per bedrijf moeten extra aanpassingen gemaakt kunnen worden in de correspondentie zoals in de twee leden hiervoor bedoeld.

---

**Requirement 1.4.9**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Would like to have	sep-07	Open

---

Beschrijving:

Het DDB bestuur kan verzendlijsten oproepen van studenten of bedrijven die aan bepaalde criteria voldoen. Voorbeelden van deze criteria zijn: ingeschreven, geaccepteerd, afgewezen, betaald, deelname evenement, subevenement (specifieke training/workshop), dag(en), bedrijf. Het is mogelijk meerdere criteria te stellen voor één verzendlijst.

---

**Requirement 1.4.10**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Would like to have	sep-07	Open

---

Beschrijving:

Het DDB bestuur kan de resultaten van de in 1.4.9 genoemde verzendlijsten als CSV downloaden (voor Word merges bijvoorbeeld).

---



**Requirement 1.4.11**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Would like to have	sep-07	Ontwerp gereed

---

Beschrijving:

Via het systeem is het mogelijk een massaemail te verzenden aan een aangemaakte verzendlijst. De email wordt automatisch toegevoegd aan het CRM en eventueel ook opgeslagen in de sent-mail box van ddb@ddb.tudelft.nl.

---

**Requirement 1.4.12**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	sep-07	Af

---

Beschrijving:

Medewerkers van DDB (Bestuur, BIT, commissarissen, websiteboys) krijgen een medewerkers-account.

---

**Requirement 1.4.12.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Should have	sep-07	Af

---

Beschrijving:

Een DDB bestuurder moet een bedrijf of student kunnen selecteren en de website te zien krijgen zoals de geselecteerde gebruiker deze zou zien. Hierin kunnen ook direct aanpassingen worden gemaakt. Eventueel zouden deze mogelijkheden ook aan andere groepen gebruikers toegekend kunnen worden.

---

**Requirement 1.4.12.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Should have	sep-07	Ontwerp gereed

---

Beschrijving:

Blokkeringen wegens verstreken deadlines moeten door het DDB bestuur (tijdelijk) opgeheven kunnen worden met één druk op de knop, wanneer zij ingelogd zijn als een bedrijf of student.

---

**Requirement 1.4.13**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	sep-07	Af

---

Beschrijving:

Een medewerker kan na het selecteren van een bedrijf via het CRM (Customer Relation Management systeem) contactmomenten met een bedrijf of student opvragen, wijzigen, verwijderen en toevoegen. Hierbij wordt onderscheid gemaakt tussen de verschillende gebruikerstypen en krijgen zij verschillende rechten op het systeem.

---

**Requirement 1.4.13.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	sep-07	Af

---

Beschrijving:

In een contactmoment worden de volgende gegevens opgeslagen:

- DDB medewerker
  - Gebruiker waarmee contact is geweest (bedrijf/student/medewerker)
  - Medium (Telefoon, email, post, fax, overig)
  - Inkomend of uitgaand contact
  - Datum en tijd van contactmoment
  - Omschrijving inhoud contactmoment
- 

**Requirement 1.4.13.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	sep-07	Ontwerp gereed

---

Beschrijving:

Bedrijfsspecifieke gegevens (denk aan: factuurnummer, contractstatus, evenement specifieke gegevens, etc) die niet via de reguliere website geraadpleegd kunnen worden, moeten via dit systeem beschikbaar en bewerkbaar zijn.

---

**Requirement 1.4.14**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Would like to have	dec-07	Uitgesteld

---

Beschrijving:

Het DDB bestuur kan enquêtes online zetten, voor zowel bedrijven als studenten. Enquêtes kunnen gekoppeld worden aan een student of bedrijf, maar ook anoniem ingevuld worden. Wanneer enquêtes anoniem ingevuld worden, wordt niet bijgehouden wie welke enquête ingevuld heeft, maar wel wie reeds een enquête ingevuld heeft.

---

**Requirement 2.1.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Een bedrijf kan gegevens voor in het bedrijvenboek aanleveren via de website. Deze gegevens bestaan uit:

- Een stuk tekst verdeeld in 5 stukken – Bedrijfsprofiel, Profiel student, Start functies, Carrière mogelijkheden en Sollicitatieprocedure.
- Advertentie
- Bedrijfslogo
- Indien deelnemend aan gesprekkendagen: aard van de gesprekken (oriënteren / solliciterend)
- Aantal werknemers in Nederland
- Aantal werknemers wereldwijd
- Vestigingsland
- Aantal academici in dienst in Nederland
- Aantal stage / afstudeerplaatsen
- Omzet wereldwijd
- Branche

Alle ingevulde gegevens en geuploadde bestanden moeten te bekijken zijn via de website.

---

**Requirement 2.1.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Should have	sep-07	Open

---

Beschrijving:

Een bedrijf moet goed zichtbare, doch niet-indringende waarschuwingen krijgen bij het inloggen op de site wanneer de gegevens uit 2.1.1 nog niet zijn ingevuld.

---

**Requirement 2.2.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	BIT	Must have	sep-07	Af

---

Beschrijving:

Het BIT moet gemakkelijk bij alle via de website ingestuurde advertenties, logos en teksten kunnen.

---

**Requirement 2.2.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	BIT	Should have	sep-07	Ontwerp gereed

---

Beschrijving:

Het BIT moet gemakkelijk inzicht kunnen krijgen in de nog ontbrekende gegevens.

---

**Requirement 2.2.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	BIT	Must have	sep-07	Af

---

Beschrijving:

Het BIT moet kunnen zien welke masters een bedrijf gekozen heeft voor de presentatiedagen en gesprekkendagen.

---

**Requirement 3.1.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Een student moet een overzicht kunnen krijgen van de aangeboden sollicitatietrainingen. Dit overzicht omvat per training in ieder geval een korte omschrijving van de inhoud, welk bedrijf de training geeft, in welke taal de training wordt gegeven, wanneer (ochtend of middag) en eventueel waar de training gegeven wordt en of het nog mogelijk is om aan te melden (vol of inschrijfperiode verstreken).

---

**Requirement 3.1.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Een student moet zich kunnen aanmelden voor een sollicitatietraining tenzij de training al vol zit of de inschrijfperiode verstreken is. In dat geval moet de student hier een melding van krijgen.

---

**Requirement 3.1.2.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Should have	dec-07	Ontwerp gereed

---

Beschrijving:

Wanneer een sollicitatietraining vol zit, kan een student zich op de wachtrij voor een sollicitatietraining zetten. Hij kan zich dan niet ook nog voor een andere training inschrijven.

---

**Requirement 3.1.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Should have	dec-07	Ontwerp gereed

---

Beschrijving:

Een student moet een waarschuwing krijgen indien een training, waarvoor hij probeert in te schrijven, gegeven wordt in een taal waarvan de student niet heeft aangegeven dat hij hem beheerst. Na aan te geven hiervan op de hoogte te zijn en geen bezwaar te hebben, kan de student doorgaan met de aanmeldprocedure.

---

**Requirement 3.1.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Would like to have	dec-07	Ontwerp gereed

---

Beschrijving:

Als een training vol zit kan een student plaats nemen op de wachtlijst. Hij kan op een wachtlijst komen per training of op een algemene wachtlijst.

---

**Requirement 3.1.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Could have	dec-07	Ontwerp gereed

---

Beschrijving:

Een deelnemende student wordt van te voren (meestal de avond voor het evenement) herinnerd aan zijn inschrijving voor een training door middel van een sms en als dit niet mogelijk is (door ontbreken van mobiel telefoonnummer) een email.

---

**Requirement 3.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Bedrijven kunnen zich aan- en afmelden voor de Sollicitatietraining tot aan een bepaalde datum.

De volgende gegevens worden ingevuld:

- Titel en beschrijving van de workshop
  - Maximum aantal deelnemers per dagdeel/sessie
  - Taal waarin de training gegeven wordt
  - Benodigde apparatuur
  - Namen van afgevaardigden die de training zullen geven
- 

**Requirement 3.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	dec-07	Ontwerp gereed

---

Beschrijving:

Het DDB bestuur kan overzichten inzien en uitprinten met per training of bedrijf de namen van de mensen die de training komen verzorgen en de namen van de studenten die zich aangemeld hebben, inclusief bijbehorende telefoonnummers.

---

**Requirement 4.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Studenten kunnen de dagindelingen voor de Presentatiedagen zien. Hierop is te vinden welke bedrijven er die dag op de beurs zullen staan, op welke stand op de beurs de bedrijven zullen staan, of de bedrijven een lezing zullen geven en indien ze een lezing geven, in welke taal.

---

**Requirement 4.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Bedrijven kunnen de volgende informatie vinden nadat ze zijn ingelogd:

- Op welke dag ze zijn ingedeeld (zodra bekend) / voorkeursdagen (aanpasbaar tot opgegeven datum)
  - Masterselectie (aanpasbaar tot opgegeven datum)
  - De standindeling (zodra bekend)
  - Tijdstip van hun presentatie (zodra bekend)
  - Gewenste apparatuur (aanpasbaar tot opgegeven datum)
  - Opgegeven presentatietaal (aanpasbaar tot opgegeven datum)
  - Opgegeven standgrootte (aanpasbaar tot opgegeven datum)
  - Gewenste voorzieningen bij stand (stoelen, tafels, statafels, schermen (lacetpanelen), stopcontacten, extra vermogen, internetaansluiting (wired/wireless) (aanpasbaar tot opgegeven datum)
  - Namen van medewerkers die namens het bedrijf naar de Presentatiedagen zullen komen (aanpasbaar tot opgegeven datum)
- 

**Requirement 4.3.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Would like to have	dec-07	Ontwerp gereed

---

Beschrijving:

Het DDB bestuur kan een dataset exporteren, die geschikt is voor het printen van bedrijvenbadges.

---

**Requirement 4.3.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Het DDB bestuur kan studenten opzoeken op naam, studienummer of NetID.

---

**Requirement 4.3.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

DDB medewerkers met voldoende rechten kunnen de betaalstatus van studenten controleren en betalingen registreren. Bij betaling wordt ook de methode (PIN/cash) geregistreerd.

---

**Requirement 4.3.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

DDB medewerkers met voldoende rechten kunnen badges voor studenten uitprinten.

---

**Requirement 4.3.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	dec-07	Ontwerp gereed

---

Beschrijving:

Het DDB bestuur kan overzichten uitdraaien van onder andere:

- Per presentatiedag een lijst bedrijven met standnummer, gegevens contactpersoon en de namen van de recruiters.
  - Alle informatie die bedrijven aangeleverd hebben, waarin zichtbaar is of alles juist is ingevuld.
  - De presentaties per dag
- 

**Requirement 5.1.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Studenten kunnen informatie opvragen over de beschikbare workshops:

- Bedrijfsnaam, titel van de workshop, beschrijving van de workshop
- Voertaal van de workshop
- Plaats van de workshop (plaatsnaam indien op locatie, zaal indien in Delft)
- Verzamelpunt (indien op locatie)
- Datum en tijd waarop de student aanwezig dient te zijn op verzamelpunt voor vertrek
- Datum en tijd waarop de student terug zal zijn op het verzamelpunt

Wanneer een student informatie opvraagt die nog niet beschikbaar is gesteld door een bedrijf, dient de student hier een melding van te krijgen in plaats van lege velden.

---

**Requirement 5.1.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Studenten moeten zich kunnen inschrijven voor een workshop. Hierbij moeten zij een workshop selecteren en een motivatie kunnen opgeven, waarom juist zij voor deze workshop zouden moeten worden uitgenodigd.

---

**Requirement 5.1.2.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

De inschrijvingsmogelijkheid voor workshops staat open vanaf een bepaalde datum en tot een bepaalde datum.

---

**Requirement 5.1.2.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Een student moet, wanneer een bedrijf hem/haar geselecteerd heeft, deze uitnodiging kunnen bevestigen dan wel afwijzen. Dit is mogelijk tot een bepaalde datum.

---

**Requirement 5.1.2.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Het afwijzen van de uitnodiging van een bedrijf geschiedt onder opgaaf van reden.

---

**Requirement 5.1.2.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Een inschrijving is pas definitief als het bedrijf de student uitnodigt nadat hij zich ingeschreven heeft en de student de uitnodiging dan weer bevestigt.

---

**Requirement 5.1.2.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Should have	dec-07	Ontwerp gereed

---

Beschrijving:

Bij onvoldoende aanmeldingen kan de inschrijfmogelijkheid voor een workshop langer open gehouden worden.

---

**Requirement 5.1.2.6**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Na het bevestigen van de uitnodiging van een bedrijf voor een workshop, kan een student nog steeds afmelden tot een bepaalde datum en onder opgaaf van reden.

---

**Requirement 5.1.2.7**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Na het verstrijken van de sluitingsdatum voor afmelding, wordt de uitschrijfmogelijkheid geblokkeerd en vergezeld van een tekst die de student doorverwijst naar het DDB bestuur in het geval dat hij wil afmelden ter voorkoming van no-shows.

---



**Requirement 5.2.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Bedrijven kunnen, nadat ze ingelogd zijn en opgegeven hebben een workshop te willen doen, informatie aanleveren over hun workshop:

- Titel en beschrijving van de workshop
  - Het maximum aantal deelnemers per sessie
  - Door welke personen de workshops gegeven zal worden
  - De plaats waar de workshop zal plaatsvinden
  - De voorkeur voor het dagdeel waarop de workshops plaats zal vinden (indien workshop in Delft)
  - De benodigde apparatuur (indien workshop in Delft)
  - De datum waarop de workshop zal plaatsvinden (indien workshop op locatie). Bij het kiezen van de datum wordt een overzicht gegeven waarop zichtbaar is op welke data reeds workshops gegeven zullen worden.
  - Adres (indien op locatie)
  - Of er lunch, borrel en/of diner inbegrepen is (indien op locatie)
  - Begintijd op locatie (oftewel, de tijd waarop de studenten verwacht worden op de locatie) (indien op locatie)
  - Eindtijd van het programma (oftewel, de tijd waarop de bus terug zal vertrekken) (indien op locatie)
- 

**Requirement 5.2.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	dec-07	Ontwerp gereed

---

Beschrijving:

Bedrijven kunnen studenten selecteren voor hun workshops. Om deze selectie te kunnen maken, krijgen bedrijven een overzicht te zien van de studenten die hebben opgegeven geïnteresseerd te zijn voor de workshop, met daarbij hun motivatie. Naast het direct selecteren voor deelname kunnen bedrijven ook studenten op de reservelijst zetten. Zodra een direct geselecteerde student zich afmeldt, zal de persoon bovenaan de reservelijst geselecteerd worden. Het selecteren van studenten is tot aan een bepaalde datum mogelijk. Per workshop moet kunnen worden opgegeven of de studentselectie definitief is. Zodra de selectie definitief is, kunnen studenten zien of ze geselecteerd zijn of niet.

---

**Requirement 5.3.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	dec-07	Ontwerp gereed

---

Beschrijving:

Het DDB bestuur moet presentielijsten uit kunnen draaien per workshop. Op deze lijst staan de deelnemers inclusief hun telefoonnummer en status. Tevens staan de reservedeelnemers vermeld op de lijst.

---

**Requirement 5.3.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Could have	dec-07	Ontwerp gereed

---

## Beschrijving:

Het DDB bestuur kan eenvoudig alle informatie uitdraaien die meegenomen dient te worden door een van de deelnemers naar de workshop:

- Contactgegevens DDB
  - Contactgegevens bedrijf + contactpersoon voor workshop
  - Adresgegevens workshoplocatie
  - Kaartje van de aankomstlocatie
- 

## Opmerkingen:

Kaartje van aankomstlocatie heeft lage prioriteit

---

**Requirement 5.3.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	dec-07	Ontwerp gereed

---

## Beschrijving:

Het DDB bestuur kan via de website de inschrijving van een bedrijf aanvullen met:

- Verzamelpunt (vervallen, beleidswijziging DDB bestuur)
  - Datum en tijd van vertrek vanaf het verzamelpunt
  - Datum en tijd van terugkomst op verzamelpunt
  - Geschatte reistijd
  - Studenten die aangeven op eigen gelegenheid naar de workshop te gaan
- 

**Requirement 6.1.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	jan-08	Ontwerp gereed

---

## Beschrijving:

Voor deelname aan de gesprekkendagen moet een student online een CV maken. De CV is standaard ingedeeld in een aantal kopjes (bijvoorbeeld 'werkervaring', 'opleidingen', 'stages', 'taalbeheersing', etc), de student kan kiezen welke kopjes hij hiervan gebruikt en de bijbehorende gegevens invullen. Het is mogelijk een niet-definitieve versie van het verslag op te slaan en hier later aan verder te werken. Zodra de student klaar is met zijn CV kan hij deze 'definitief' maken, hierna is wijzigen niet meer mogelijk. De student kan een overzicht opvragen waarin de bedrijven die zijn CV zullen ontvangen.

---

## Opmerkingen:

Het overzicht waarin de student ziet welke bedrijven zijn CV zullen ontvangen heeft lage prioriteit.

---

**Requirement 6.1.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	jan-08	Ontwerp gereed

---

## Beschrijving:

Een student kan voordat de bedrijven studenten gaan uitnodigen een aantal voorkeursbedrijven opgeven. Deze bedrijven zullen de CV van de student ontvangen en de student kunnen uitnodigen voor een gesprek, ongeacht of ze de studie van de student geselecteerd hebben. De student kan opgeven wat zijn reden is om het bedrijf als voorkeursbedrijf op te geven (oriëntatie, stage, baan, etc) en heeft de mogelijkheid om een motivatie te schrijven.

---

**Requirement 6.1.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Studenten	Must have	jan-08	Ontwerp gereed

---

## Beschrijving:

Nadat de bedrijven studenten uitgenodigd hebben, kan een student een overzicht opvragen van de bedrijven die hem uitgenodigd hebben voor een gesprek. In dit overzicht is het tot een bepaalde datum mogelijk om gesprekken te accepteren of af te wijzen. Reeds bevestigde gesprekken kunnen alleen onder opgaaf van reden nog geannuleerd worden, het is daarna niet meer mogelijk het gesprek opnieuw te bevestigen. Zodra het rooster voor de gesprekken bekend is, zijn ook de ingedeelde gesprekken en bijbehorende informatie (tijd, datum, plaats, interviewer) zichtbaar. Ingedeelde gesprekken kunnen bevestigd of afgemeld worden.

---

**Requirement 6.2.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	jan-08	Ontwerp gereed

---

## Beschrijving:

Een bedrijf kan tot op een bepaald moment opgeven welke personen gesprekken komen afnemen namens het bedrijf. Per 'interviewer' wordt aangegeven: naam, beschikbaarheid per dagdeel (voor de gehele periode van de gesprekkendagen), maximaal aantal in te zetten dagdelen, met studenten van welke studies deze interviewer gesprekken wil voeren en eventueel met welke studenten specifiek een interviewer gesprekken wil voeren.

---

**Requirement 6.2.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Should have	sep-07	Uitgesteld

---

## Beschrijving:

Een bedrijf kan per gesprek aangeven hoe lang het duurt tot een bepaald maximum.

---

## Opmerkingen:

Vaste gesprekslengte gevraagd door DDB bestuur

---

**Requirement 6.2.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	mrt-08	Ontwerp gereed

---

Beschrijving:

Een bedrijf kan een PDF bestand downloaden, waarin alle CV's die het heeft ontvangen te vinden zijn.

---

**Requirement 6.2.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	mrt-08	Ontwerp gereed

---

Beschrijving:

Een bedrijf kan tot een bepaald moment studenten selecteren voor een gesprek. Het is mogelijk om een student uit te nodigen voor meerdere gesprekken.

---

**Requirement 6.2.5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	mrt-08	Ontwerp gereed

---

Beschrijving:

Zodra het gesprekkenrooster gemaakt is, kunnen bedrijven het rooster inzien op de website.

---

**Requirement 6.2.6**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Should have	mrt-08	Ontwerp gereed

---

Beschrijving:

Bij elke student in zowel het selectieoverzicht als het overzicht van geplande gesprekken kan een bedrijf een opmerking plaatsen, die zichtbaar is voor de student. Via deze weg kan een bedrijf verzoeken aan de student te kennen geven. Studenten kunnen via hun interface tevens een bericht terugzenden naar het bedrijf.

---

**Requirement 6.2.7**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Af

---

Beschrijving:

Het bedrijf kan de selectie van de masters voor de gesprekkendagen aanpassen tot een bepaald moment. Het is mogelijk geen masters te selecteren. Het bedrijf krijgt dan alleen de CV's van studenten die het bedrijf kiezen.

---

**Requirement 6.2.8**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	Bedrijven	Must have	sep-07	Ontwerp gereed

---

Beschrijving:

Een bedrijf kan het bezorgadres voor het CV pakket opgeven.

---

**Requirement 6.3.1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	mrt-08	Ontwerp gereed

---

Beschrijving:

Het DDB bestuur kan PDF's genereren voor de CV pakketten die aan bedrijven toegezonden zullen worden en ter download beschikbaar gesteld worden. Een CV pakket bevat alle CV's van studenten die een studie doen die het bedrijf geselecteerd heeft of het bedrijf opgegeven hebben als voorkeursbedrijf.

---

**Requirement 6.3.2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Must have	mrt-08	Ontwerp gereed

---

Beschrijving:

Het DDB Bestuur kan een gesprekkenrooster genereren. Na het genereren is het mogelijk gesprekken handmatig te verplaatsen, kladversies van het rooster op te slaan en versies te publiceren op de website (naar zowel bedrijven als studenten). Aanpassingen kunnen ook in het reeds gepubliceerde rooster gemaakt worden.

---

**Requirement 6.3.3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Should have	mrt-07	Ontwerp gereed

---

Beschrijving:

Het DDB bestuur kan dagoverzichten uitprinten: Wanneer, waar, welke gesprekken plaatsvinden en wie de gesprekken zullen voeren.

---

**Requirement 6.3.4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Functional	DDB bestuur	Should have	mrt-07	Ontwerp gereed

---

Beschrijving:

Het DDB bestuur kan lijsten met contactgegevens van bedrijven en studenten uitprinten per dag.

---

**Requirement NF 1**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Non functional	Niet actor specifiek	Must have	sep-07	Voldaan

---

Beschrijving:

Zekerheid: Het systeem moet op regelmatige basis, ongeveer per 15 minuten, gebackuppeld worden.

---

Opmerkingen:

Behalve deze backup procedure die onafgebroken loopt vanaf begin oktober, wordt het systeem tevens dagelijks op een andere fysieke locatie gebackuppeld.

---

**Requirement NF 2**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Non functional	Niet actor specifiek	Should have	sep-07	Voldaan

---

Beschrijving:

Snelheid: Het systeem moet voor studenten en bedrijven direct reageren. Ingewikkeldere handelingen van DDB medewerkers mogen maximaal enkele minuten duren.

---

Opmerkingen:

Alle functionaliteiten die via webpagina's geboden worden, bleken na ondervraging van bedrijven en bestuurders, een voor de gebruiker acceptable responstijd te hebben. De enige uitzondering is het uploaden van advertenties, waarbij de verbindingssnelheid van de gebruiker de responstijd bepaald. Dit geval is acceptabel, want bestanden uploaden is een activiteit die weining -twee tot vijf keer per jaar- wordt uitgevoerd.

---

**Requirement NF 3**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Non functional	Niet actor specifiek	Must have	sep-07	Voldaan

---

Beschrijving:

Beschikbaarheid: Het systeem moet elk jaar van september tot mei 99% van de tijd online zijn, inclusief onderhoud. Groot onderhoud, waarvoor het systeem offline moet, dient in de overige maanden plaats te vinden.

---

Opmerkingen:

Het systeem heeft naast de online 'productie' omgeving een ontwikkelomgeving, welke niet voor de eindgebruiker zichtbaar is. Deze omgeving draait afzonderlijk van de productieomgeving en gebruikt een eigen database. Nieuwe onderdelen en aanpassingen kunnen in deze omgeving ontwikkeld en getest worden, zonder dat de productieomgeving offline hoeft. Zodra een nieuw onderdeel geaccepteerd is, worden de wijzigingen of toevoegingen ook op de productieomgeving toegepast. Op deze manier veroorzaakt onderhoud geen downtime.

Het systeem wordt geserveerd door een machine in de serverruimte van faculteit TNW. De machine is beschermd van stroomuitval door middel van een UPS (uninterruptable power supply) en zal na eventuele langdurige stroomuitval automatisch opnieuw starten.

Alle bovenstaande opmerkingen worden ondersteund door het feit dat het systeem online is sinds begin oktober 2007 en in die periode nog geen noemenswaardige downtime heeft gehad.

---

**Requirement NF 4**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Non functional	Niet actor specifiek	Should have	sep-07	Open

---

Beschrijving:

Gelijktijdig gebruik: Het systeem moet berekend zijn op gebruik van enkele honderden studenten en enkele tientallen bedrijven die gedurende dezelfde dag willen werken.

---

Opmerkingen:

Een test met deze aantallen gebruikers is nog niet uitgevoerd. Echter aan de hand van een schatting, op basis van de huidige belasting van het systeem, worden geen problemen verwacht met grote aantallen gebruikers.

---

**Requirement NF 5**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Non functional	Niet actor specifiek	Must have	sep-07	Open

---

Beschrijving:

Robuustheid: Het systeem zal bij eventuele problemen een nette foutmelding moeten geven.

---

Opmerkingen:

Sinds eind oktober tot december 2007, heeft het systeem in bijna alle gevallen een nette foutmelding gepresenteerd. In enkele specifieke gevallen was een Propel error zichtbaar, echter de meerderheid van de gevallen wordt netjes afgehandeld.

---

**Requirement NF 6**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Non functional	Niet actor specifiek	Could have	sep-07	Voldaan

---

Beschrijving:

Modieus: De interface dient er aantrekkelijk uit te zien en ‘hippe effecten’ te bevatten om de gebruiker te imponeren en professionaliteit uit te stralen.

---

Opmerkingen:

Het grafisch ontwerp is uitbesteed aan een professionele ontwerper en de feedback van de onder-vraagde gebruikers (bedrijven en bestuurders) is zeer positief.

---

**Requirement NF 7**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Non functional	Niet actor specifiek	Should have	sep-07	Voldaan

---

Beschrijving:

Overzichtelijkheid: De interface moet zonder uitleg direct te begrijpen zijn en intuïtief werken.

---

Opmerkingen:

Tot december 2007 zijn er geen noemenswaardige klachten gekomen vanuit het bestuur van DDB of vanuit de bedrijven, die zouden aangeven dat de interface van de site niet begrijpbaar is

---

**Requirement NF 8**

---

Type:	Actor:	MoSCoW:	Deadline:	Status:
Non functional	Niet actor specifiek	Must have	sep-07	Voldaan

---

## Beschrijving:

Onderhoudbaarheid: De code achter en het ontwerp van het systeem dient goed gedocumenteerd te worden, zodat de kwaliteit van het systeem zo min mogelijk zal verslechteren door onderhoud.

---

## Opmerkingen:

Wij zijn van mening dat de door ons geleverde phpdoc van FormLib en de interne documentatie in de rest van de code voldoende is.

---