

# Project Building Inther-WEB Interface



## Adresgegevens

### ***Student***

Naam: Gijs van Belkom  
Adres: Duinbeek 18  
Postcode: 5653 PM Eindhoven  
Telefoon: 040 2522060  
Mobiel: 06 52417456  
E-mail: [gvanbelkom@inthergroup.com](mailto:gvanbelkom@inthergroup.com)

### ***Bedrijf***

Naam: Inther Logistics Engineering B.V.  
Adres: Keizersveld 50  
Postcode: 5803 AN Venray  
Telefoon: 0478 502 575  
Website: [www.inthergroup.com](http://www.inthergroup.com)

### ***Bedrijfsbegeleider***

Naam: Roel Beckers  
Adres: Keizersveld 50  
Postcode: 5803 AN Venray  
Telefoon: 0478 502 575  
E-mail: [rbeckers@inthergroup.com](mailto:rbeckers@inthergroup.com)

### ***Docentbegeleidster***

Naam: Mariëlle Fransen  
Adres: Rachelsmolen 1  
Postcode: 5612 MA Eindhoven  
Telefoon: 0877 872270  
E-mail: [m.fransen@fontys.nl](mailto:m.fransen@fontys.nl)

---

## Voorwoord

Deze scriptie is geschreven in het kader van mijn afstudeerperiode als student aan de Fontys Hogeschool ICT te Eindhoven. De afstudeerperiode heeft plaatsgevonden bij het bedrijf Inther Logistics Engineering B.V. in de periode van 1 januari 2010 tot en met 2 juli 2010. Roel Beckers heeft mij begeleid tijdens de werkzaamheden aan de nieuwe webinterface.

Mijn dankbetuiging gaat uit naar de personen die mij op een positieve wijze hebben ondersteund tijdens dit project en daarmee ook hebben bijgedragen aan het succesvol afronden hiervan. Alle afdelingen van het bedrijf hebben hun inspiraties uitgesproken, waardoor ik goed in staat ben geweest om alle requirements duidelijk te definiëren.

In het bijzonder wil ik mijn bedrijfsbegeleider Roel Beckers en docentbegeleidster Mariëlle Fransen bedanken voor de regelmatige feedback en opbouwende kritiek gedurende mijn afstudeerperiode.

Venray, juni 2010

Gijs van Belkom

## Inhoudsopgave

Samenvatting .....	5
Summary .....	6
Verklarende woordenlijst .....	7
1. Inleiding.....	8
2. Het bedrijf .....	9
2.1. Introductie .....	9
2.2. Geschiedenis.....	10
2.3. Mijn plek binnen de organisatie .....	10
3. De opdracht .....	11
3.1. Introductie .....	11
3.2. Geschiedenis.....	11
3.3. Huidige situatie .....	11
3.4. Probleemstelling.....	13
3.5. Opdrachtomschrijving .....	13
3.6. Fasering.....	14
4. Initiatief & definitiefase .....	15
4.1. Plan van Aanpak .....	15
4.2. Document van Eisen .....	15
4.3. Graphical User Interface design .....	17
5. Onderzoeksfase.....	22
5.1. Java Framework onderzoek .....	22
5.2. Proof of Concepts .....	24
6. Ontwerpfase .....	27
6.1. Ontwerpspecificatie.....	27
6.2. Toelichting .....	27
7. Realisatie & Testfase .....	32
7.1. Realisatie.....	32
7.2. Testfase.....	35

---

8. Overdrachtsfase .....	39
8.1. Technische handleiding .....	39
8.2. Uitloop .....	39
Conclusie en aanbevelingen .....	40
Evaluatie.....	41
Literatuurlijst.....	42
Bijlagen.....	43
Bijlage I: Plan van Aanpak .....	43
Bijlage II: Document van Eisen .....	43
Bijlage III: Ontwerp Grafische User Interface.....	43
Bijlage IV: Java Framework onderzoek.....	43
Bijlage V: Ontwerpspecificatie .....	43
Bijlage VI: Testplan .....	43

## Samenvatting

Het afstudeerproject “Building Inther-WEB Interface” is uitgevoerd bij het bedrijf Inther Logistics Engineering B.V.. Inther is een jong, internationaal bedrijf dat zich als system integrator richt op het automatiseren van logistieke processen in magazijnen. Inther’s expertise omvat alle aspecten die aan bod komen in een magazijn. Denk hierbij aan opslagmethoden, gebruik van beschikbare ruimte, order verzameltechnieken en strategieën, beheer van gegevens, transport en distributie van goederen.

De uitvoering van het project heeft plaatsgevonden op de development afdeling, hier ligt de verantwoordelijkheid voor de ontwikkeling van Inther’s software suite genaamd Inther-LC. Inther-WEB is een onderdeel van de Inther-LC suite en bestaat uit een grafische gebruikersinterface waarmee eindgebruikers verschillende modules in Inther-LC kunnen monitoren en beïnvloeden via een browser. Deze gebruikersinterface bevat diverse overzichten die de gebruiker in staat stelt om gewenste gegevens op te vragen en functionaliteiten uit te voeren op het systeem.

De huidige Inther-WEB module is vooral ontworpen vanuit het oogpunt van een ontwikkelaar. Technisch gezien functioneert de webinterface naar behoren. Er zijn echter op vele punten verbeteringen en uitbreidingen denkbaar. Ook speelt moderne techniek een rol; in de afgelopen jaren zijn er wereldwijd vele web frameworks ontwikkeld, waardoor tegenwoordig meer functionaliteit en betere grafische weergave haalbaar is. Concreet is er gekeken naar de volgende aandachtspunten:

- Optimaal gebruikmaken van de beschikbare ruimte op het scherm
- Weergave en toegankelijkheid van functionaliteit
- Navigatiemogelijkheden
- Kleurgebruik
- Performance verbetering

Tijdens de Initiatief & Definitiefase zijn bovengenoemde aandachtspunten vertaald in specifieke requirements. Nadat de requirements bekend waren is hiervan een grafisch ontwerp opgezet. Deze fase heeft een aantal iteraties gekend met verschillende afdelingen binnen het bedrijf waardoor alle wensen besproken zijn. Gedurende deze periode bleek dat het niet haalbaar is om alle vastgestelde requirements te implementeren binnen de afstudeerperiode.

Vervolgens is een onderzoek van start gegaan waarin verschillende mogelijke implementatietechnieken vergeleken zijn om de best passende oplossing voor Inther-WEB te vinden. Aan de hand van de resultaten uit het onderzoek zijn een aantal Proof of Concepts gerealiseerd waarmee de gemaakte keuzes bevestigd zijn. Op basis van de requirements, het onderzoek en de opgedane kennis van de Proof of Concepts is een technisch ontwerp opgesteld.

Het resultaat van de afstudeerperiode is een goede basis om het project “Building Inther-WEB Interface” voort te zetten, zodat uiteindelijk alle gewenste requirements gerealiseerd worden.

## Summary

The graduation project “Building Inther-WEB Interface” has been carried out at the Inter Logistics Engineering B.V. company. Inther is a young international, company that focuses itself as system integrator to automate logistic warehouse processes. Inthers expertise encloses all aspects that can be found in a warehouse. Think of storage methods, use of available space, order picking techniques and strategies, data management, transport and distribution of goods.

The execution of this project took place at the development department, here lies the responsibility for developing Inthers software suite, named Inther-LC. Inther-WEB is a part of the Inther-LC suite and represents a graphical user interface which can be used by end-users to monitor and influence multiple modules in Inther-LC via a browser. This user interface contains multiple overviews which makes the user able to look up requested data and execute functionalities at the system.

Currently the Inther-WEB module was mostly designed from the developers point of view. Technically it is a well functioning environment, but one that could be improved and extended in a lot of ways. Also the availability of modern technology has a key role; In the past several years a lot of frameworks has been developed throughout the world. As a result of this more functionalities and better graphic looks can be achieved. Following points of attention have been looked into concretely:

- Optimized use of available screen space
- Accessibility and view of functionality
- Navigation possibilities
- Use of color
- Performance improvement

Points of attention that have been noticed above have been translated to specific requirements during the Initiative & definition phase. A graphical design has been created after finishing the requirement specification. This phase has had a number of iterations with different departments within the company to retrieve and discuss all wishes.

Thereupon an investigation was started to compare different possible implementation techniques to find the best fitting solution for Inther-WEB. According to the investigation results a number of Proof of Concepts were realized to confirm the made choice. A technical design was drawn up based on the requirements, the investigation and gained knowledge during the building of the Proof of Concepts.

The result of the graduation period resulted in a solid base to continue the project “Building Inther-WEB Interface” to eventually realize all wanted requirements.

## Verklarende woordenlijst

Term	Omschrijving
System integrator	Bedrijf dat verschillende verschillende hardware en software oplossingen integreert tot één functionerend systeem
Inther-LC	Inther Logistics Control is een verzamelnaam voor alle software producten ontwikkeld door Inther
Inter-WEB	Inther webinterface; een onderdeel van Inther-LC
Proof of Concept	Bewijsmateriaal dat een voorstel haalbaar is. Tijdens dit project kleine applicaties die gewenste werking van functionaliteit aantonen
Java	Java is een programmeertaal welke object georiënteerd is. Java frameworks zijn standaard voorzien van een grote hoeveelheid beschikbare functionaliteit, en worden continue verder ontwikkeld
IDE	Integrated Development Environment, een algemene naam voor een software ontwikkelomgeving als NetBeans
UML	Unified Modeling Language; een modelmatige taal om objectgeoriënteerde ontwerpen en analyses te kunnen maken
Framework	Bibliotheek van softwarecomponenten en standaarden die gebruikt kunnen worden tijdens het ontwikkelen van nieuwe software
GUI	Graphical User Interface. De grafische schil van een applicatie.
Component	Elk onderdeel van de GUI is een component
AJAX	Asynchroon Javascript And XML. Dit is een techniek om webpagina's gedeeltelijk te kunnen verversen
XML	Extensible Markup Language. Veel gebruikte standaard om gegevens gestructureerd op te slaan
Databasequery	Een commando waarmee gegevens uit een database opgevraagd kunnen worden
Look and Feel	Alle gebruikersinterface aspecten van een applicatie. Bepalend hoe de applicatie aanvoelt.
Exception	Softwarematige fout
CDI	Contexts and Dependency Injection is globaal omschreven een techniek die koppelingen (injections) verzorgt tussen een applicatie en een web project
Responsetijd	Tijd die het systeem nodig heeft om bepaalde functionaliteit uit te voeren
URL	Uniform Resource Locator. Een label dat verwijst naar een informatiebron, bijvoorbeeld een webpagina



## 1. Inleiding

Deze afstudeerscriptie is bedoeld als procesbeschrijving van het project “Building Inther-WEB Interface” welke uitgevoerd is bij het bedrijf Inther Logistics Engineering B.V.. Alle belangrijke beslissingen, problematiek en bevindingen worden in dit verslag uitvoerig omschreven en beargumenteerd.

Opdrachtgever voor deze afstudeerstage is Inther Logistics Engineering, een jong, internationaal bedrijf dat zich als system integrator richt op het automatiseren en uitrollen van logistieke processen in magazijnen.

Gedurende de afstudeerperiode is een volledig nieuwe Inther-WEB applicatie ontworpen en gedeeltelijk geïmplementeerd. Huidige wensen zijn vastgesteld en geïmplementeerd met state of the art technologie om ook in de toekomst verzekerd te zijn van bruikbaar product. Om een impressie te krijgen van de opbouw van de scriptie is Figuur 1 opgenomen waarin een globale toelichting per hoofdstuk gegeven wordt.

Hoofdstuk 1	<b><i>Inleiding</i></b> Introductie van deze afstudeerscriptie
Hoofdstuk 2	<b><i>Het bedrijf</i></b> Introductie van het bedrijf
Hoofdstuk 3	<b><i>De opdracht</i></b> Gedetailleerde opdrachtomschrijving & fasering
Hoofdstuk 4	<b><i>Initiatief en definitiefase</i></b> Omschrijving van de eerste fase; opstellen van het Plan van Aanpak en vaststellen van gewenste requirements
Hoofdstuk 5	<b><i>Onderzoeksfase</i></b> Onderzoeken van beschikbare technologieën en de gevonden resultaten onderbouwen met Proof of Concepts
Hoofdstuk 6	<b><i>Ontwerpfase</i></b> Dit hoofdstuk beschrijft het technisch ontwerp zoals dat is opgezet aan de hand van de vastgestelde requirements
Hoofdstuk 7	<b><i>Realisatie &amp; Testfase</i></b> Hier wordt omschreven op welke manier het project geïmplementeerd is en welke knelpunten hierbij aan het licht kwamen
Hoofdstuk 8	<b><i>Overdrachtsfase</i></b> Uiteenzetting van de oplevering van alle producten
Hoofdstuk 9	<b><i>Conclusie en aanbevelingen</i></b> Conclusies, aanbevelingen en vervolgstappen naar aanleiding van de afstudeerstage
Hoofdstuk 10	<b><i>Evaluatie</i></b> Persoonlijk schrijven over project “Building Inther-WEB Interface”

Figuur 1: Globale hoofdstukomschrijving

## 2. Het bedrijf

### 2.1. Introductie

Inther Logistics Engineering, hierna te noemen Inther, is een jong, internationaal bedrijf dat zich als system integrator richt op het automatiseren en uitrollen van logistieke processen in magazijnen. Het hoofdkantoor is gevestigd in Venray, overige kantoren zijn gehuisvest in Amerika, Duitsland, Hongarije en Moldavië.

Inther's expertise omvat alle aspecten die aan bod komen in een magazijn. Denk hierbij aan opslagmethoden, gebruik van beschikbare ruimte, order verzameltechnieken en strategieën, beheer van gegevens, transport en distributie van goederen. Alle genoemde aspecten kunnen naar wens van de klant geautomatiseerd worden, waardoor de efficiëntie vergroot wordt. Het doel hiervan is de productiekosten te verminderen en de capaciteit te vergroten.

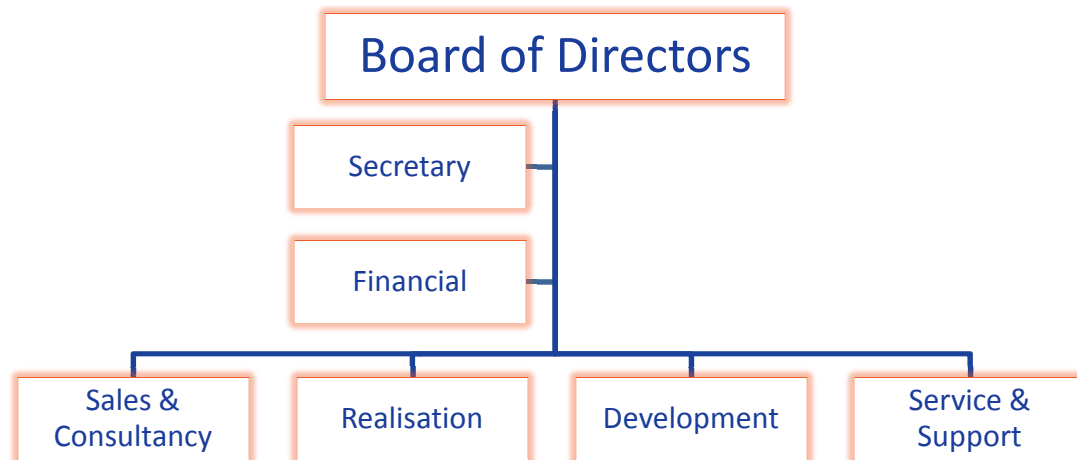
Alle geautomatiseerde logistieke onderdelen van Inther zijn opgenomen in diverse verschillende modules met als basis de Inther-LC library waarvan de overige componenten gebruik maken. Inther-LC staat voor Inther Logistics Control en biedt onder ander de volgende functionaliteiten:

- Warehouse Management System (WMS)
- Pick to Light (PTL)
- Pick to Cart (PTC)
- Material Flow Control (MFC)
- RF-Scanning
- Voice Picking
- Barcode scanning
- RFID scanning

Software alleen is niet voldoende om een logistiek proces te automatiseren. Inther maakt gebruik van onderaannemers om verschillende hardware te kunnen leveren. Onderstaande lijst geeft een impressie van de gebruikte materialen tijdens een project.

- Magazijn racking
- Scanners
- Terminals
- Transportbanden
- Commisioner systemen
- Inline print, verpakking en scanner apparatuur
- Programmable Logic Controller (PLC)
- Picklights

In totaal zijn momenteel ongeveer dertig medewerkers in dienst bij Inther verdeeld over de verschillende afdelingen binnen het bedrijf. Deze afdelingen zijn weergegeven in onderstaande organisatiestructuur.



Figuur 2: Organisatiestructuur van Inther

## 2.2. Geschiedenis

Tot 1998 werkten Martijn Herder en Paul Hermesen (directie) bij Witron Engineering in Zuid-Duitsland. Ze trokken veel met elkaar op en kwamen op een zeker moment tot de conclusie dat ze het zelf anders en beter konden doen dan Witron. Samen zijn ze in datzelfde jaar het bedrijf Inther Logistics Engineering gestart.

Martijn Herder en Paul Hermesen zijn begonnen in het huis van Paul. Naarmate het bedrijf groeide zijn ze verschillende malen verhuisd naar locaties met meer kantoorruimte. Sinds ruim twee jaar is het bedrijf gevestigd in een eigen pand genaamd "Keyserstaete" op Keizersveld 50 te Venray. Op dit moment zijn er zelfs al weer plannen voor een nieuw pand langs de A73 op industrieterrein "De Hulst" in Venray.

## 2.3. Mijn plek binnen de organisatie

Sinds 23 maart 2009 ben ik werkzaam bij Inther op de Service & Support afdeling. Deze afdeling houdt zich primair bezig met het operationeel houden van de magazijnautomatisering, neventaken zijn:

- Preventief onderhoud uitvoeren
- Bestellingen en reparaties verwerken
- Updates bij klanten uitrollen
- Software uitbreidingen en Request for Changes implementeren

Door werkzaam te zijn op deze afdeling heb ik snel de verschillende projecten en producten leren kennen. Bijna elke dag zie ik nieuwe code en problematiek waardoor het leergehalte erg hoog ligt. Denken in klantgerichte oplossingen staat hierbij centraal. Speciaal voor dit project ben ik verplaatst naar de Development afdeling om me volledig op mijn afstudeerproject te kunnen toelagen.

## 3. De opdracht

### 3.1. Introductie

De grafische gebruikersinterface, oftewel Graphical User Interface (GUI) wordt ingezet om klanten op hoog niveau controle te geven over geautomatiseerde logistieke processen. Met 'hoog niveau' wordt bedoeld dat gebruikers het systeem kunnen beïnvloeden zonder in contact te komen met hardware of directe aansturing daarvan. Om dit te realiseren wordt gebruik gemaakt van een website. Websites zijn eenvoudig op grote schaal in te zetten waardoor vele gebruikers tegelijkertijd toegang hebben tot hetzelfde systeem, een gewenste situatie binnen grote organisaties.

Men kan via de webinterface globaal de volgende zaken beheren:

- Diverse route en ordergegevens
- Gebruikersaccounts
- Magazijnindeling
- Transporteenheden welke door MFC of PTL getransporteerd en verwerkt worden
- Historische gegevens
- Statistieken
- Systeeminformatie

### 3.2. Geschiedenis

Medio 2002 is een project gestart om een website als interface te gebruiken met als doel grote aantallen gebruikers gezamenlijk toegang te verschaffen tot één algemeen systeem, welke vanaf standaard computers te gebruiken is. In 2004 is deze interface opnieuw herzien om deze een vernieuwd uiterlijk te geven waarbij de functionaliteit zoveel mogelijk hetzelfde is gebleven. Tot op heden is het huidige web project vanaf eind 2004 functioneel en grafisch bijna ongewijzigd.

### 3.3. Huidige situatie

Om een beeld te krijgen van de huidige situatie is Figuur 3 toegevoegd als screenshot van een routeoverzicht. Routes worden gebruikt om orders op een vooraf vastgelegde volgorde af te handelen. De gegevens die in het overzicht getoond worden zijn niet van belang, het doel is om een impressie te krijgen van de huidige situatie.

Enkele uiterlijke kenmerken zijn:

- Navigatiemenu aan linker zijde
- Resultaten getoond onder de beschikbare functionaliteit
- Zoeken op maximaal twee criteria
- Details worden weergegeven onder het hoofdvenster
- Gegevens in kolommen zijn mogelijk hyperlinks naar andere overzichten

http://localhost/index.html - Windows Internet Explorer

File Edit View Favorites Tools Help

http://localhost/index.html

Get by and

Order by

Page

Tools

Google

10THERLC

Route overview

Details

Ref

Route ID

Name

Dispatch time

Partner

Priority

Status

To Pricelabeling

Max Pallet Height

Min Full Pallet Height

Driving Time

Terminal

Driving Time

Route master

Dock

Emergency palletized

Article

Outbound

Routermaster overview

Route overview

Outbound Shipment overview

Order overview

Orderline overview

Transport Unit

Partner

Beer

User overview

Usergroup overview

Warehouse

Location overview

Flowthrough overview

PTL locations overview

Support

Priority overview

Loc overview

Inventory

Monitoring

Shipment monitoring overview

Order monitoring overview

Performance overview

Workstation monitoring overview

System overview

Palletize overview

History

Route monitoring overview

Shipment monitoring overview

Order monitoring overview

Transport unit overview

Inspection overview

Summarize Route Data Overview

\_logout

220641

120\_2009H15

120 INF ESPOO A 1,2

15-04-2009 04:00

SWAN LINE OY

11

ACTIVE

NO

5

4

1

2123013

15:00

120

DISPATCHH02

220772

120\_2009H15

120 INF ESPOO A 1,2

15-04-2009 04:00

SWAN LINE OY

1

ACTIVE

YES

5

4

1

2123013

15:00

120

DISPATCHH02

220658

121\_2009H15

121 INF ESPOO A 3,4

15-04-2009 04:00

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123013

15:00

121

DISPATCHH03

220773

121\_2009H15

121 INF ESPOO A 3,4

15-04-2009 04:00

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123013

15:00

121

DISPATCHH03

220774

122\_2009H15

122 INF ESPOO A 5,6

15-04-2009 04:00

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123013

15:00

122

DISPATCHH04

220779

122\_2009H15

122 INF ESPOO A 5,6

15-04-2009 04:00

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123013

15:00

122

DISPATCHH04

220860

191\_2009H14

191 UKO SWA PETIKK

14-04-2009 15:00

SWAN LINE OY

1

ACTIVE

YES

5

4

1

01:30

01:30

151

DISPATCHH10

220962

304\_2009H14

304 LDC JAMAKALA

14-04-2009 15:00

SWAN LINE OY

11

PROCESSED

NO

5

4

1

14:00

14:00

304

DISPATCHH03

220799

320\_2009H15

320 INF LEMPALA

15-04-2009 03:55

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123113

13:00

320

DISPATCHH07

220879

320\_2009H15

320 INF LEMPALA

15-04-2009 03:55

SWAN LINE OY

11

INITIAL

NO

5

4

1

2123113

13:00

320

DISPATCHH07

220854

321\_2009H15

321 INF LEMPALA

15-04-2009 03:55

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123113

13:00

321

DISPATCHH05

220859

321\_2009H15

321 INF LEMPALA

15-04-2009 03:55

SWAN LINE OY

11

INITIAL

NO

5

4

1

2123113

13:00

321

DISPATCHH05

220900

322\_2009H15

322 INF LEMPALA

15-04-2009 03:55

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123113

13:00

322

DISPATCHH05

220979

322\_2009H15

322 INF LEMPALA

15-04-2009 03:55

SWAN LINE OY

11

INITIAL

NO

5

4

1

2123113

13:00

322

DISPATCHH05

220954

324\_2009H15

324 INF SALO

15-04-2009 03:55

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123119

23:00

324

DISPATCHH05

220960

324\_2009H15

324 INF SALO

15-04-2009 03:55

SWAN LINE OY

11

INITIAL

NO

5

4

1

2123119

23:00

324

DISPATCHH05

220943

325\_2009H15

325 INF SALO

15-04-2009 03:55

SWAN LINE OY

11

ACTIVE

YES

5

4

1

2123119

23:00

325

DISPATCHH05

220961

325\_2009H15

325 INF SALO

15-04-2009 03:55

SWAN LINE OY

11

INITIAL

NO

5

4

1

2123119

23:00

325

DISPATCHH05

220959

420\_2009H15

420 INF KOUOLA 1

15-04-2009 08:30

SWAN LINE OY

11

INITIAL

NO

5

4

1

2123142

21:30

420

DISPATCHH08

220559

445\_2009H15

445 VALO VIENTIKOU

16-04-2009 12:00

SWAN LINE OY

10

PACKED

NO

5

4

1

445

DISPATCHH10

220503

445\_2009H16

445 VALO VIENTIKOU

16-04-2009 12:00

SWAN LINE OY

10

ADMIN

NO

5

4

1

445

DISPATCHH10

221319

640\_2009Z18

640 VALO JYASKALA

18-12-2009 08:30

SWAN LINE OY

1500

DISPATCHED

NO

5

4

1

17:00

640

DISPATCHH08

221494

2120074\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

4

RELEASING

NO

11

5718

221498

2120063\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

5

ACTIVE

YES

11

5704

221517

2119867\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

2

PROCESSED

NO

11

2684

221520

2120182\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

2

RELEASING

NO

11

5846

221527

2120116\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

1

RELEASING

NO

11

5769

221491

2120135\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

4

RELEASING

NO

11

5794

221497

2119008\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

1

RELEASING

NO

11

2706

221512

2118943\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

2

RF-FASTING

NO

11

2639

221516

2118655\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

2

RELEASING

NO

11

1705

221521

2120850\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

2

PROCESSED

NO

11

44043

221523

2120847\_220773\_15-ARR-09\_1

121\_2009H15\_1

121\_2009H15\_1

1

2

RF-FASTING

NO

11

44043

Figuur 3: Route overzicht in de huidige webinterface

### 3.4. Probleemstelling

De huidige Inther-WEB module is vooral ontworpen vanuit het oogpunt van een ontwikkelaar. Technisch gezien functioneert de webinterface naar behoren. Er zijn echter op vele punten verbeteringen en uitbreidingen denkbaar. Mede hierdoor voldoet de huidige webinterface op meerdere vlakken niet meer aan de hedendaagse verwachtingen van Inther en haar klanten, waardoor een vervangend product gewenst is. Ook speelt moderne techniek een rol; in de afgelopen jaren zijn er wereldwijd vele web frameworks ontwikkeld waardoor tegenwoordig meer functionaliteit en betere grafische weergave haalbaar is.

#### **Graphical User Interface**

Op grafisch vlak is het uiterlijk niet modern genoeg. Beschikbare ruimte op het scherm wordt niet optimaal benut doordat het navigatiemenu aan de linkerkant veel horizontale ruimte in beslag neemt. Verticaal ruimteverlies wordt veroorzaakt door een grote werkbalk die slechts twee zoekvelden bevat. Per overzicht worden altijd alle beschikbare functionaliteiten in een werkbalk weergegeven, zodat hier ook ruimte verloren gaat. Ook kleurgebruik en opmaak van de weergegeven componenten zijn niet meer van deze tijd.

#### **Gebruiksvriendelijkheid**

Gebruiksvriendelijkheid is een belangrijk issue. Navigatie tussen verschillende overzichten gebeurt niet efficiënt, omdat er weinig koppelingen tussen de overzichten beschikbaar zijn. Bij het implementeren van een overzicht moet de ontwikkelaar handmatig de gewenste links toevoegen. Regelmatig worden deze vergeten. Ook wordt bij iedere zoek- en navigatieopdracht de volledige pagina herladen, waardoor ingevulde tekstvelden leeg gemaakt worden. Hierdoor moeten gebruikers vaak onnodig dezelfde gegevens invoeren.

#### **Performance**

De performance van de webinterface is niet optimaal. Bij elk overzicht worden overzichtgegevens altijd volledig opgehaald, hetgeen onnodig database resources verbruikt. Zodra meerdere gebruikers tegelijk door grote overzichten navigeren, kan vertraging merkbaar zijn.

### 3.5. Opdrachtschrijving

project "Building Inther-WEB Interface" omvat het ontwikkelen van een nieuwe gebruikers interface weergegeven in een website. De website moet volledig configureerbaar zijn vanuit een centraal punt; een Oracle database, of Java code. De opdracht bestaat uit de onderstaande wensen:

- Modern uiterlijk
- Mogelijkheid tot bediening zonder muis
- Optimaal navigeren tussen menu's en submenu's
- Zoekresultaten gemakkelijk meenemen naar andere menu's
- Diverse applicaties van Inther integreren in de webinterface
- Inther dashboard performance tools integreren
- Services kunnen bedienen vanuit de webinterface
- De mogelijkheid om een touchscreen layer toe te voegen
- Overzichten vanuit één punt aanmaken; Oracle database of Java code

### 3.6. Fasering

Om de opdracht zoals in paragraaf 3.5 omschreven succesvol te voltooien, zullen afspraken gemaakt moeten worden over taken, op te leveren producten en planning. Figuur 3 geeft per fase aan hoe fasering oorspronkelijk opgezet is. Zie bijlage I voor het 'Plan van Aanpak'. Hierin kan de gedetailleerde fasering geraadpleegd worden.

Fase	Omschrijving	Deliverables
Initiatief & definitiefase	Vastleggen van communicatieplan, proces en planning. Uitzoeken en vastleggen van gedetailleerde requirements	Communicatieplan
		Plan van Aanpak
		Detailplanning
		Document van Eisen
		Graphical User Interface Design
Onderzoeksfase	Diverse frameworks onderzoeken en hiermee demonstratie applicaties bouwen	Java framework onderzoek
		Werkende demo applicaties
Ontwerpfase	Ontwerpen van Inther-WEB	Ontwerpspecificatie
Realisatie & Testfase	Implementatie en testen van het eindproduct	Inther-WEB Eindproduct
		Afstudeerscriptie
Overdrachtsfase	Afronden van het project	Uitloop om deliverables uit voorgaande fases af te ronden.
		Technische handleiding opleveren
		Bedrijfsdemonstratie
		Afstudeerzitting

Figuur 3: Fasering

## 4. Initiatief & definitiefase

### 4.1. Plan van Aanpak

Het Plan van Aanpak is opgezet om het verloop van het project zo goed mogelijk vast te leggen met als einddoel het succesvol opleveren van een werkend eindproduct. In het Plan van Aanpak wordt gekeken naar de haalbaarheid van de gestelde doelen, fasering en randvoorwaarden met als doel de focus op het project te behouden en afdwaling te voorkomen. Ook is als bijlage in het Plan van Aanpak een gedetailleerde planning opgenomen als extra controle op de gestelde fasering.

### 4.2. Document van Eisen

Het Document van Eisen omschrijft specifiek en gedetailleerd alle gewenste requirements. Om tot die requirements te komen zijn een aantal vergaderingen belegd met de afdelingen Sales & Consultancy, Development en Realisation. Deze vergaderingen waren vooral brainstorm sessies om zo veel mogelijk verschillende ideeën vast te stellen. Zie bijlage II voor het 'Document van Eisen' waarin de definitieve requirements zijn opgenomen.

#### **Prioriteit vaststellen**

Tijdens het vaststellen van de requirements blijkt dat er teveel wensen zijn om binnen de gestelde afstudeerperiode te realiseren. Aan de hand van een MoSCoW analyse is aan elke requirement een prioriteit toegekend. MoSCoW is een efficiënte en eenvoudige methode welke vaak binnen ICT gebruikt wordt om een gezamenlijk doel af te bakenen.

MoSCoW staat voor:

- Must have – Implementatie verplicht
- Should have – Implementatie wenselijk indien haalbaar
- Could have – Implementatie als geen andere onderdelen hierdoor in gevaar komen
- Would have – Geen implementatie binnen de huidige scope, wenselijk in de toekomst

#### **Requirements**

Ter ordening zijn alle requirements onderverdeeld in categorieën. De meest belangrijke functionele requirements zijn opgenomen in onderstaande omschrijvingen.

#### Inloggen

De gebruiker moet kunnen inloggen met een persoonlijk gebruikersnaam en wachtwoord. Zodra de gebruiker inlogt moet het laatst geopende venster getoond worden. Dit venster wordt geladen uit de profielhistorie.

#### Help

De helpfunctie verwijst naar een externe hulpbron waarin uitleg wordt gegeven over de gewenste component.



### Grafische weergave

Niet alle onderdelen van een menu of functionaliteiten hoeven standaard weergegeven te worden. De gebruiker kan kiezen om componenten naar eigen keuze weer te geven. Er is een menuknop beschikbaar waaronder alle functionaliteiten zijn opgeslagen op basis van gebruikersrechten.

### Notifier

Notifier is een melding aan de gebruiker. Wanneer in het magazijn een machine in storing valt zal een event-notifier bericht getoond worden. Als de gebruiker navigeert of functionaliteit uitvoert zal een web-notifier bericht verschijnen.

### Overzichten weergeven

Overzichten worden opgesplitst in een master- en detailweergave. Een master kan meerdere detailweergaves bevatten. Per overzicht zijn navigatielinks beschikbaar om snel tussen de verschillende overzichten te navigeren. Alle componenten op de pagina moeten dynamisch worden geüpdate. Dit houdt in: geen volledige refresh van een venster, maar alleen de gewenste component bijwerken.

De hoeveelheid beschikbare pagina's binnen een overzicht moet duidelijk weergegeven worden. Ook moet de gebruiker hier eenvoudig doorheen kunnen navigeren. Gebruikers kunnen functionaliteit uitvoeren, met of zonder geselecteerde resultaten.

Per getoonde kolom is het mogelijk een sortering op te geven, indien gewenst kan de gebruiker ook op meerdere kolommen tegelijkertijd sorteren. Zoeken gebeurt ook op kolomniveau; elke kolom kan één of meerdere zoekvelden bevatten. Zoeken op datum kan via een 'date-picker', ook kan een datumbereik worden opgegeven. Per zoekveld is het mogelijk een operator toe te kennen. Beschikbare operatoren zijn:

- =
- !=
- <
- >
- =<
- >=
- LIKE

### Externe applicaties

Inther-Dashboard is een webapplicatie die tabellen en grafieken tekent. Deze applicatie moet geïntegreerd worden in Inther-WEB. Werkdialogen zoals deze nu ingezet worden op diverse werkstations zijn ook beschikbaar vanuit de nieuwe webinterface. De wens bestaat om klanten de mogelijkheid te geven om Inther softwareprocessen te starten of stoppen via functionaliteit binnen een overzicht.

### Bediening

Om de webinterface gebruiksvriendelijker te maken is ervoor gekozen om volledige toetsenbordondersteuning te bieden. Dit houdt in dat met het toetsenbord de volledige webinterface bedient kan worden. Wanneer de gebruiker met de muis werkt is een 'rechtermuisknop menu' beschikbaar, ook wel context menu genoemd. Via de gebruikershistorie is het mogelijk de laatst bezochte pagina's te openen. De pagina wordt dan hersteld zoals deze als laatste gebruikt is.

### Standaard functionaliteit

Resultaten zijn exporteerbaar naar xls, pdf of csv bestandsformaten. Ook is het mogelijk om de weergegeven resultaten te printen.

## **4.3. Graphical User Interface design**

Requirements zeggen niet voldoende bij het ontwikkelen van een grafisch informatiesysteem. Om een gebruiksvriendelijke en moderne applicatie te bouwen is een grafisch ontwerp noodzakelijk. De Graphical User Interface (GUI) wordt ook wel presentatielaag genoemd. Hiermee wordt aangegeven dat deze laag de communicatie afhandelt tussen de applicatie en eindgebruiker. Het grafische ontwerp is met behulp van dezelfde stakeholders als het Document van Eisen tot stand gekomen.

Ter illustratie van het GUI ontwerp is Figuur 5 opgenomen. Deze toont het meest uitgebreide overzicht binnen de nieuwe webinterface applicatie met daaronder in Figuur 4 een toelichting per component. In de loop van deze scriptie zal het begrip component vaker voorkomen, er wordt dan altijd verwezen naar een grafisch item op de webpagina. Zie 'Graphical User Interface design' in bijlage III, hierin staat uitgebreid omschreven hoe de ontwerpen tot stand zijn gekomen.

Figuur 5: Ontwerp van de Graphical User Interface

Component	Omschrijving
Navigatiemenu	Bovenin de pagina bevindt zich het navigatiemenu. Gebruikers kunnen hiermee navigeren naar de verschillende pagina's van de applicatie. Het menu is drie lagen diep. Standaard worden alleen de eerste twee lagen getoond, waarbij de derde laag als een menu naar beneden uitklapt.
Huidige positie	Rechts bovenin wordt in een tekstvak het huidige submenu getoond zodat de gebruiker weet welke pagina getoond wordt. Dit is noodzakelijk omdat de derde laag van het navigatiemenu verborgen is.
Standaard functionaliteit	Onder de 'huidige positie' component zijn standaard functies beschikbaar; printen, gegevens exporteren, pagina herladen, browsergeschiedenis en help. Deze functies zijn met behulp van icoontjes beschikbaar gemaakt.
Functiemenu	Bijna alle overzichten hebben specifieke functionaliteit, bijvoorbeeld: vrijgave van orders, verwijderen van routes en aanmaken van gebruikers. Per overzicht zullen deze in het functiemenu beschikbaar zijn, weergegeven links onder het navigatiemenu.
Contextmenu	Het contextmenu is een meer technische naam voor 'rechter muisknop menu' en bevat dezelfde items als het functiemenu.
Datagrid	Met de component datagrid wordt de tabel aangeduid. De eerste kolom bevat altijd een 'checkbox' waardoor het mogelijk is regels te selecteren. Via het functiemenu kan de gebruiker een actie uitvoeren op de geselecteerde gegevens. Indien op de kolomnaam geklikt wordt, zal deze gesorteerd worden. Het is mogelijk om op alle kolommen gecombineerd te kunnen sorteren, zowel oplopend als aflopend. Een nummer en pijl geeft de volgorde en de orde van de sortering aan.
Zoekveld	Zoekvelden kunnen worden geopend door op het vergrootglas te klikken welke links boven elke kolomnaam getoond wordt. Er is geen limiet gesteld aan het aantal zoekvelden. Verschillende type zoekvelden zijn beschikbaar om op verschillende type gegevens te zoeken: <ul style="list-style-type: none"> <li>Datum selectie: In een kalender de juiste datum kiezen</li> <li>Invoer veld: Vrije invoer van tekst, nummers en tekens</li> <li>Drop-down list: Keuzes zijn op voorhand vastgelegd in een lijst</li> </ul> Indien meerdere zoekvelden gebruikt worden, zal een 'relatie' veld beschikbaar zijn. Deze geeft de zoekrelatie tussen de velden weer, gelimiteerd per kolom.
Paginator	Per overzicht kunnen vele duizenden rijen met gegevens beschikbaar zijn. Om dit te reguleren worden de gegevens verdeeld over meerdere pagina's. Navigatie tussen deze pagina's gebeurt met de paginator, weergegeven rechts onder de datagrid.
Detailweergave - Nested	Wanneer in de datagrid op een rij het '+' teken wordt ingedrukt, opent de detailweergave binnen het bestaande datagrid. Dit wordt ook wel nested weergave genoemd. Slechts één detailweergave kan per overzicht geopend zijn.
Detailweergave -	Binnen de detailweergave in het nested datagrid is een tabmenu opgenomen.

Tabmenu	Hierdoor is het mogelijk om meerder overzichten zichtbaar te maken in één detailweergave.
Detailweergave	Elke detailweergave bestaat uit een functiemenu, datagrid, paginator en zoekvelden met dezelfde specificaties als het normale overzicht. Binnen het detail datagrid is het niet mogelijk om nog een detailweergave te openen.
Detailweergave – Navigatielinks	Onderin het detailweergave zijn navigatielinks te vinden. Deze links navigeren naar het overzicht dat bij de detailweergave hoort.
Statusbalk – WEB notifier	In de statusbalk rechts naast het Inther logo bevindt zich de WEB notifier. Deze laat informatieve terugkoppeling zien op basis van gebruikersacties, bijvoorbeeld wanneer de gebruiker via het functiemenu een succesvolle of foutieve actie uitvoert.
Statusbalk – Event notifier	In de statusbalk links naast het klantlogo staat de Event notifier. Zodra extern een 'event' wordt aangemaakt, zal deze knipperend worden weergegeven zodat de operator direct op de hoogte is van een dringende situatie. Events worden vooral gebruikt bij machinestoringen.
Keyboardbediening	Een niet grafische onderdeel van de webinterface is de keyboardbediening. Door gebruik van toetsencombinaties is het mogelijk om naar verschillende onderdelen van de website te navigeren. Denk hierbij aan de sneltoetsen van Microsoft Windows.

Figuur 4: Beschikbare webinterface componenten

### Overweging

Op grafisch vlak zijn een aantal verbeteringen behaald met het nieuwe ontwerp. Het navigatiemenu is verplaatst naar de bovenkant van de pagina. Hierdoor ontstaat meer horizontale ruimte. Kolommen in de datagrid hebben daardoor meer ruimte zodat de getoonde resultaten overzichtelijker zijn. De derde laag van het navigatiemenu is standaard verborgen om ruimte te besparen, toch kan de gebruiker zien welk overzicht getoond wordt in de huidige positie component.

Gebruikers zullen de webinterface vaker gebruiken om informatie op te zoeken dan functionaliteit uit te voeren. Inther-LC is namelijk zo opgezet dat minimale gebruikersinteractie noodzakelijk is om te functioneren. Door deze gedachtegang is mogelijk de functionaliteiten te verbergen totdat de gebruiker deze nodig heeft. Via het functiemenu en contextmenu heeft de gebruiker snel toegang tot de beschikbare functies.

Een ander belangrijke tekortkoming aan de huidige webinterface zijn de zoekmogelijkheden. In het nieuwe ontwerp is gekozen om per kolom zoekvelden te kunnen gebruiken. Dit heeft een aantal voordelen:

- Het is meteen duidelijk bij welke kolom het zoekveld hoort
- Onbeperkte hoeveelheid zoekopdrachten
- Snel toegankelijk

Door het nested detailvenster is direct zichtbaar op welk master resultaat de weergave betrekking heeft. Ook creëert het nested detailvenster ruimte om een tabmenu weer te geven waarin dezelfde mogelijkheden beschikbaar zijn als in het hoofdvenster. Hierdoor hoeft de gebruiker niet continu van overzicht te wisselen om uitgebreid te zoeken of functionaliteit uit te voeren. Mocht de gebruiker alsnog willen navigeren naar een ander overzicht dan is het mogelijk via de beschikbare navigatielinks.

De huidige webinterface heeft de mogelijkheid om gecombineerd te kunnen zoeken, echter gaat dit alleen via een pop-up menu. Gebruikers hebben daarin de mogelijkheid om de gewenste sortering op te geven. In het nieuwe ontwerp is gekozen om dit rechtstreeks op de kolom te regelen, waardoor dit veel sneller toegankelijk is.

In de statusbar zijn de notifier componenten toegevoegd om terugkoppeling naar de gebruiker mogelijk te maken. Momenteel gebeurt dit alleen via een pop-up scherm dat een melding laat zien na het uitvoeren van functionaliteit. De nieuwe webinterface maakt het mogelijk dat op elke gebruikersactie een bericht verschijnt. Zo wordt de gebruiker bewust gemaakt van de uitgevoerde handeling. Meldingen weergegeven over andere Inther-LC modules was voorheen sowieso niet mogelijk. Event berichten zullen alleen aan bepaalde gebruikersgroepen getoond worden op basis van gebruikersrechten.

## 5. Onderzoeksfase

In de onderzoeksfase zal worden gekeken hoe de wensen zo succesvol mogelijk geïmplementeerd kunnen worden. Het onderzoek gaat dieper in op de huidige beschikbare technologieën. Zodra de meest bruikbare techniek geselecteerd is, zal dit door middel van Proof of Concepts bewezen worden.

### 5.1. Java Framework onderzoek

Een framework kan worden gezien als een bibliotheek waarin grote hoeveelheden software componenten beschikbaar zijn. Deze componenten kunnen door software ontwikkelaars gebruikt worden om nieuwe applicaties te ontwikkelen. Omdat bijna alle software van Inther in Java geprogrammeerd is, ligt de keuze voor deze programmeertaal tijdens dit project vast.

Java kent vele verschillende mogelijke technieken om een web project op te zetten. Al deze technieken worden door verschillende frameworks vertegenwoordigd. Momenteel zijn er wereldwijd honderden frameworks beschikbaar, waardoor een gedegen onderzoek naar de meest geschikte techniek noodzakelijk is. Later in deze paragraaf zal de methode en een samenvatting van de conclusie omschreven worden. Uitgebreide details over het onderzoek zijn beschikbaar in bijlage IV: 'Java framework onderzoek'.

#### **De methode**

Tijdens een bijeenkomst genaamd "JavaOne Technical Sessions" van het "Sun Developer Network" heeft een presentatie plaatsgevonden over welke stappen doorlopen moeten worden om een weloverwogen keuze te maken. De presentatie is bekend onder nummer: TS-6457, genaamd: "CHOOSING A JAVA WEB FRAMEWORK: A COMPARISON". In de literatuurlijst is de bron van deze presentatie terug te vinden. Deze methode zal tijdens dit onderzoek als leidraad gebruikt worden en bestaat uit de volgende stappen:

1. Beschikbare Web frameworks inventariseren
2. Traditionele criteria vaststellen
3. Wat zijn de benodigdheden
4. Geavanceerde criteria vaststellen
5. Afweging tussen frameworks
6. Conclusie

#### **Samenvatting**

Bij aanvang van de methode zal eerst een selectie gemaakt worden tussen verschillende frameworks. Er is gekozen uit een aantal frameworks welke sterk verschillen in implementatietechniek met als doel een zo breed mogelijk onderzoek op te zetten.

Door het invullen van de traditionele en geavanceerde criteria ontstaat een lijst van requirements welke concreet meetbaar tegen de gekozen frameworks uitgezet kan worden. Uiteindelijk blijkt het framework 'JavaServer Faces 2' (JSF 2) het beste te voldoen zodra deze wordt afgewogen tegen de belangrijkste requirements weergegeven in Figuur 5.

Requirement	Omschrijving
AJAX support	AJAX staat voor Asynchronous JavaScript And XML. Deze techniek wordt gebruikt om gedeeltes van een website opnieuw te laden zonder de rest van de pagina te beïnvloeden. Bijvoorbeeld kan een tabel opnieuw voorzien worden van gegevens zonder dat de pagina volledig herladen wordt.
Minimale configuratie via XML bestanden	Een aantal frameworks maakt intensief gebruik van configuratie. Deze configuratie bepaalt bijvoorbeeld de manier van navigeren tussen pagina's. Hoe meer configuratie nodig is hoe complexer de software achteraf aan te passen is. XML is een veel gebruikte opmaak om configuratie in op te slaan.
Push en Pull functionaliteit	Push wil zeggen dat een externe applicatie ervoor zorgt dat de website een actie uitvoert. Pull betekent dat de website zelf op basis van gebruikeracties gegevens ophaalt. Tijdens dit project zijn beide noodzakelijk.

Figuur 5: Belangrijkste requirements voor het framework onderzoek

Het JSF 2 framework bevat een groot aantal componenten welke op een webpagina weergegeven kunnen worden. Als aanvulling op het gekozen framework kan een Component Library gekozen worden. Deze libraries bevatten extra of uitgebreidere componenten die niet standaard in JSF 2 aanwezig zijn. Men kan denken aan een tabel welke standaard beschikbaar is in JSF 2, de Component Library heeft hierop bijvoorbeeld als uitbreiding horizontale en verticale scroll bars of sortering per kolom gerealiseerd. Ook kan het zijn dat de Component Library meer grafische effecten biedt. Denk hierbij aan transparantie, fade in, fade out, sliden en andere grafische effecten.

Component Libraries voor JSF 2 zijn minder beschikbaar, omdat JSF 2 recent is uitgebracht. Er zijn wel vele Component Libraries beschikbaar voor JSF 1 welke zijn omgebouwd tot JSF 2. Technisch zijn deze niet volledig compatible en daardoor niet gewenst binnen dit project. Om een goede keuze te maken welk Component Library geschikt is, is een deelonderzoek gestart. Voor dit deelonderzoek is beperkt dezelfde methode gebruikt als het Java framework onderzoek, dat wil zeggen:

1. Beschikbare Component Libraries inventariseren
2. Criteria vaststellen
3. Afweging tussen de Component Libraries
4. Conclusie

Als uitkomst blijkt 'Myfaces Trinidad 2.0.0 Alpha', hierna afgekort met Trinidad, de beste keuze te zijn. Helaas voldoet deze niet aan alle wensen, waardoor eventueel tijdens de implementatie besloten kan worden om een tweede Component Library te gebruiken. Dit verhoogt helaas wel de afhankelijkheid van externe partijen. Ook is het mogelijk om Custom Components te ontwikkelen waardoor wel voorzien wordt in de gewenste voorzieningen.



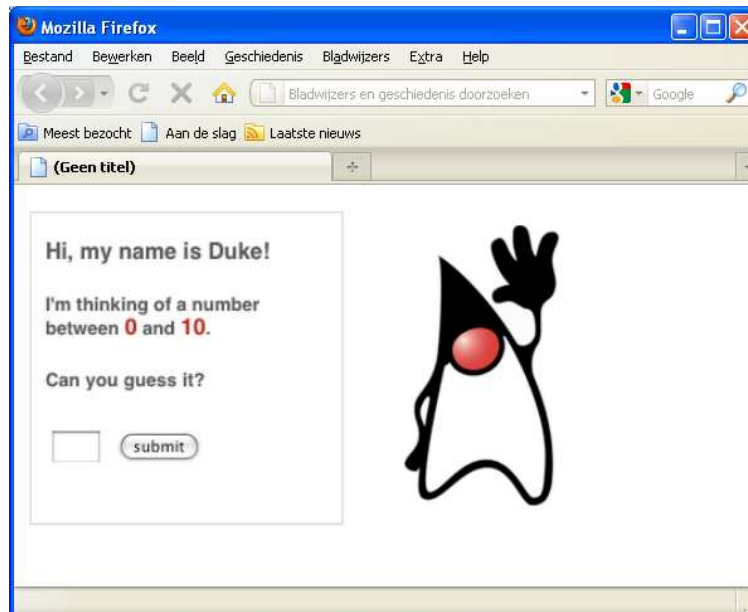
## 5.2. Proof of Concepts

Als bewijsvoering en kennismaking met de gekozen frameworks zijn drie Proof of Concepts gerealiseerd:

- Hello World project
- Tabel dynamisch opbouwen
- Custom Component bouwen

### *Hello World project*

Een Hello World project wordt in de software wereld gebruikt om aan te tonen dat een simpel stukje code functioneert om het vervolgens uit te bouwen tot een geavanceerde applicatie. Op de website van NetBeans staat een JSF 2 introductie project welke door het volgen van een handleiding uitgebouwd wordt tot een spelletje. Figuur 6 laat een screenshot zien van het eindresultaat. De eerste pagina vraagt om een cijfer van 0 t/m 10 in te vullen. Door op 'submit' te klikken wordt gecontroleerd of de ingevoerde waarde correct is geraden. Aan de hand van de uitkomst van de controle, goed of fout, wordt de vervolgpagina weergegeven met de tekst: "Yay, you got it" of "Sorry, [invoer] isn't it. Guess again...".



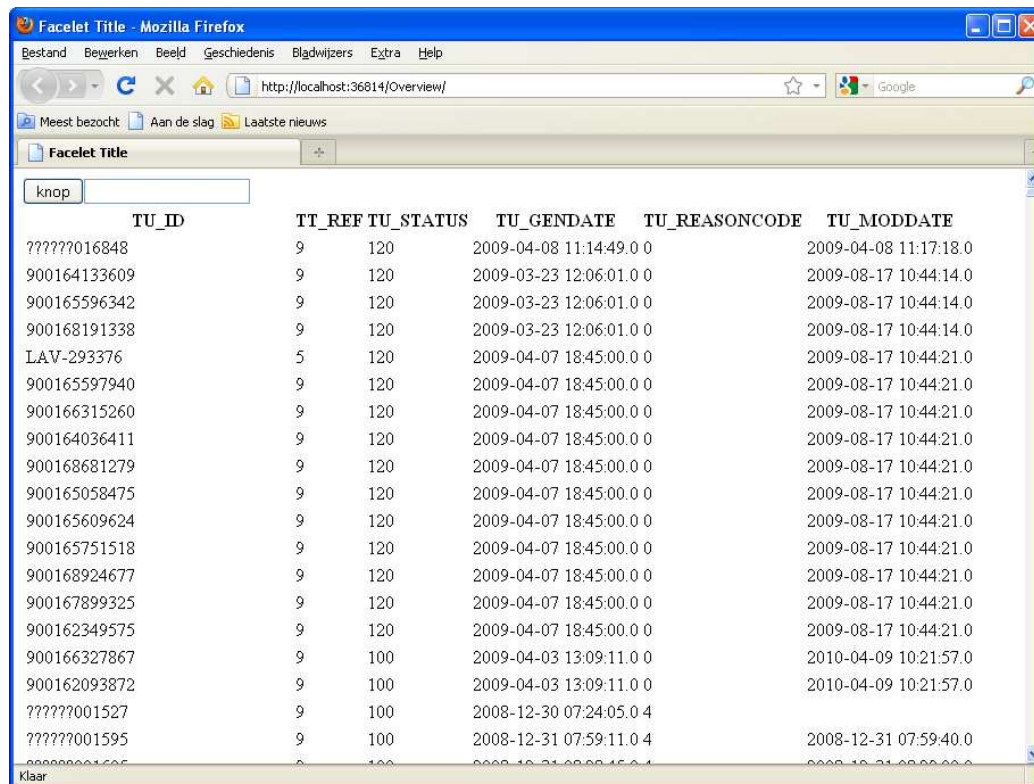
Figuur 6: Hello World project

Bovenstaand Hello World project is opgebouwd met standaard componenten van JSF 2. Om ook kennis te maken met de Trinidad is het invoerveld en 'submit' knop omgezet naar Trinidad componenten. Dit simpele project nam toch veel tijd in beslag omdat JSF 2 in combinatie met Trinidad lastiger integreert dan verwacht. Er zijn veel configuratiemogelijkheden die uitgezocht moeten worden. Ook maakt Trinidad gebruik van een groot aantal extra libraries die toegevoegd moeten worden.

### **Tabel dynamisch opbouwen**

De tweede Proof of Concept betrof het dynamisch opbouwen van een tabel, ook wel datagrid genoemd. Om een zo generiek mogelijke opzet te hanteren bestaat de wens om één overzicht webpagina te ontwikkelen waarmee alle standaard overzichten weergegeven kunnen worden. Deze webpagina wordt vanaf nu overview genoemd. Diverse verschillende overzichten tonen logischerwijs diverse verschillende resultaten. Dit houdt in dat het aantal kolommen en rijen niet van te voren bekend is, waardoor de tabel flexibel opgebouwd moet worden.

Normaal gesproken wordt een datagrid statisch op de website aangemaakt, waarmee het aantal kolommen vast ligt. Met deze methode is het aantal getoonde rijen wel eenvoudig flexibel te vullen. Flexibel aanmaken van de gewenste hoeveelheid kolommen kan gerealiseerd worden door de datagrid dynamisch op te bouwen. In de webpagina wordt een koppeling gemaakt naar de achterliggende Java code waarin uiteindelijk het opbouwen plaats vindt. Figuur 7 laat het resultaat van deze Proof of Concept zien. De betekenis van de getoonde resultaten is niet van belang; er zijn willekeurige kolommen uit de database gebruikt. AJAX technologie is toegevoegd om de tabel opnieuw te tekenen wanneer er gezocht wordt op de kolom 'TU\_ID' via het invoerveld bovenaan de pagina.



TU_ID	TT_REF	TU_STATUS	TU_GENDATE	TU_REASONCODE	TU_MODDATE
?????016848	9	120	2009-04-08 11:14:49.0 0		2009-04-08 11:17:18.0
900164133609	9	120	2009-03-23 12:06:01.0 0		2009-08-17 10:44:14.0
900165596342	9	120	2009-03-23 12:06:01.0 0		2009-08-17 10:44:14.0
900168191338	9	120	2009-03-23 12:06:01.0 0		2009-08-17 10:44:14.0
LAV-293376	5	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900165597940	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900166315260	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900164036411	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900168681279	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900165058475	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900165609624	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900165751518	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900168924677	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900167899325	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900162349575	9	120	2009-04-07 18:45:00.0 0		2009-08-17 10:44:21.0
900166327867	9	100	2009-04-03 13:09:11.0 0		2010-04-09 10:21:57.0
900162093872	9	100	2009-04-03 13:09:11.0 0		2010-04-09 10:21:57.0
?????001527	9	100	2008-12-30 07:24:05.0 4		
?????001595	9	100	2008-12-31 07:59:11.0 4		2008-12-31 07:59:40.0
?????001505	9	100	2008-12-31 07:59:11.0 4		2008-12-31 07:59:40.0

Figuur 7: Dynamisch opgebouwde tabel

Gedurende het implementeren van deze Proof of Concept verscheen telkens een foutmelding die verwijst naar Trinidad renderkit. Een renderkit wordt gebruikt om componenten op het scherm af te beelden (renderen). JSF 2 heeft standaard een renderkit aan boord, maar wordt bij gebruik van een externe Component Library zoals Trinidad vervangen door de renderkit van de Component Library zelf. Renderen van een pagina wordt in onderliggende softwarelagen afgehandeld zonder dat de ontwikkelaar hier iets van ziet. Fouten die hiernaar verwijzen kunnen duiden op verkeerd gebruik van de component of een fout (bug) in de renderkit code. Beide fouten zijn zonder hulp van de ontwikkelaars van Trinidad moeilijk uit te sluiten. Om zeker te zijn van een goede oplossing is contact opgenomen met de ontwikkelaars van Trinidad. Ook is een vraag gesteld op een forum dat zich specialiseert in alle technieken die met Java te maken hebben. Trinidad's snelle reactie gaf de oplossing: er moest andere code gebruikt worden in combinatie met de datagrid component. Zodra de datagrid op de aangegeven manier gebruikt wordt, functioneert de Proof of Concept wel zoals gewenst.

### ***Custom Component***

Als laatste Proof of Concept wordt een zelfgemaakt, oftewel custom component gebouwd. Grafisch is alleen een invoerveld opgenomen, waardoor het niet interessant is hiervan een screenshot toe te voegen. Technisch zorgt de zelfgemaakte schil om het invoerveld ervoor dat alle functionaliteit van de component door de ontwikkelaar zelf te programmeren is in plaats van de opgelegde functionaliteit van de componenten in JSF 2 en Trinidad. Bij functionaliteit van een invoerveld kan gedacht worden aan tekst, kleur, lettertype en stijl. Natuurlijk zijn al deze functionaliteiten al vele malen geïmplementeerd door diverse organisaties. Dit project dient vooral als kennismaking met Custom Components waarbij alle stappen doorlopen worden om nieuwe componenten te bouwen en gebruiken, niet de technische uitbereiding van de component zelf. Deze techniek is bruikbaar wanneer blijkt dat een standaard component niet voldoet.

## 6. Ontwerpfase

Tijdens de ontwerpfase wordt de technische Ontwerpspecificatie opgesteld met behulp van de Unified Modeling Language, afgekort met UML, een wereldwijd gebruikte schematische taal om software ontwerpen weer te geven. Dit hoofdstuk zal het ontwerpproces omschrijven zonder diep op technische keuzes in te gaan. Voor de technische inhoud kan de Ontwerpspecificatie geraadpleegd worden, zie bijlage V.

### 6.1. Ontwerpspecificatie

project “Building Inther-WEB Interface” is bedoeld om over een lange periode inzetbaar te zijn. Om dit te realiseren zal de software zo ontworpen moeten worden dat deze toegankelijk en eenvoudig uit te breiden is. Over elk aspect zal goed nagedacht moeten worden. De ontwerpfase is voor Inther een van de belangrijkste fases tijdens dit project.

Het opzetten van de ontwerpspecificatie is een iteratief proces waarbij de bedrijfsbegeleider intensief betrokken is. Per week hebben een aantal iteraties plaatsgevonden totdat het model voldoende doordacht was. Om nog eens extra input te verwerven, hebben twee hoofdontwikkelaars ook deelgenomen aan een iteratieslag.

In de Ontwerpspecificatie zijn twee UML diagrammen opgenomen elk met tekstuele toelichting. Omdat de Ontwerpspecificatie te technisch is, zijn de UML diagrammen niet opgenomen in deze scriptie. Deze zijn terug te vinden in bijlage V. Met behulp van de opgedane kennis van de Proof of Concepts en vastgestelde requirements wordt eerst het Java ontwerp uiteengezet door middel van het UML Class diagram. Bestaande classes uit de Inther-LC library zijn niet opgenomen in het ontwerp aangezien dit niets toevoegt aan het ontwerp. Op de achtergrond zijn deze classes wel degelijk noodzakelijk, vooral om de database verbindingen en log bestanden correct af te handelen. Het tweede UML diagram betreft het database ontwerp, waarmee de Java code van gegevens wordt voorzien.

### 6.2. Toelichting

Om een zo helder mogelijke toelichting op de UML diagrammen te geven wordt de werking vanuit het oogpunt van de gebruiker bekeken, namelijk de grafische componenten op de web pagina's. Ter verduidelijking kan worden teruggekeken naar paragraaf 4.3 waarin het grafische ontwerp uitgewerkt is.

#### ***Inloggen - Filteren***

Zodra een gebruiker de webinterface opent, zal een sessie worden opgezet met de server. Een sessie bevat onder andere communicatie- en profielgegevens van de gebruiker. Ook zal in de sessie worden bijgehouden of de gebruiker correct ingelogd is. Indien dit niet het geval is zal de gebruiker automatisch doorgestuurd worden naar de login pagina. Elke keer als de gebruiker een actie uitvoert op de website, moet de sessie gecontroleerd worden op geldigheid. Technisch gezegd: elk request wordt gefilterd. Na het inloggen zal de gebruiker op de hoofdpagina uitkomen, ook bekend als index.

### ***Index***

De index bestaat uit twee statische componenten: het navigatie tabmenu en de statusbalk. Statisch wil zeggen dat deze geprogrammeerd zijn op de webpagina en ze daardoor altijd worden weergegeven. Het navigatie tabmenu wordt bovenaan de pagina weergegeven, onderaan vindt men de statusbalk. Tussen deze componenten is een grote lege ruimte welke opgevuld wordt door de specifieke overzichten. Gebruikers kunnen naar deze overzichten navigeren via het navigatie tabmenu.

### ***Menu***

Binnen een overzicht met een geopend detailvenster zijn vier menu's beschikbaar, namelijk:

- Navigatiemenu
- Functiemenu
- Tabmenu in detailweergave
- Functiemenu in detailweergave

In het ontwerp is gekozen om een zo generiek mogelijk menu op te bouwen zodat één implementatie voor alle menu's volstaat. Elk menu item bevat een koppeling naar een gewenste pagina of actie die uitgevoerd dient te worden. Dergelijke koppelingen worden Uniform Resource Locators genoemd afgekort met URL. Ook kan elk item een referentie naar een bovenliggend item bevatten waardoor het weergeven van meerdere lagen in een menu mogelijk is. Grafisch gezien klikt de gebruiker op een item in het menu waardoor via de URL het gewenste overzicht geladen wordt.

### ***Browser geschiedenis***

Bij elke navigatieactie naar een overzicht zal het vorige overzicht worden opgeslagen in de browser historie. Hierbij worden ook alle ingevulde zoekvensters en sorteringen meegenomen. Geselecteerde resultaten zijn uitgesloten van de geschiedenis omdat resultaten constant van inhoud veranderen.

### ***Overzicht***

De opbouw van het overzicht zal niet statisch zijn, maar dynamisch aangemaakt vanuit de Java code. Voor het overzicht zullen onderstaande objecten gegenereerd worden:

- Functiemenu
- Datagrid met sortering en zoekvelden
- Paginator

Zodra een detail in de datagrid geopend wordt, zullen extra componenten opgebouwd worden:

- Tabmenu
- Functiemenu
- Datagrid met zoekvelden en sortering
- Paginator

***Datagrid***

De datagrid component geeft resultaten, afkomstig uit de database, in tabelvorm weer. Deze resultaten zullen met een commando, oftewel databasequery opgehaald worden. Elk overzicht wordt door een eigen databasequery voorzien van gegevens, vanaf nu genaamd overviewquery. Overviewqueries zijn opgeslagen in de database met als voordeel dat ze centraal zijn opgeslagen en automatisch na een eventuele wijziging gebruikt worden. Elke keer dat een overzicht geladen wordt, zal de overviewquery ook opnieuw geladen worden.

***Paginator***

Niet alle opgehaalde resultaten kunnen worden weergegeven op één pagina; overviewqueries kunnen duizenden resultaten opleveren. De paginator berekent het aantal beschikbare pagina's door de totale hoeveelheid resultaten te delen met de hoeveelheid gewenste resultaten op één pagina. Op de webinterface is deze paginator grafisch zichtbaar gemaakt zodat de gebruiker naar de gewenste pagina binnen het overzicht kan navigeren.

***Overviewquery parsing***

Gebruikers bepalen welke resultaten in de datagrid worden weergegeven. Dit is mogelijk door onderstaande acties uit te voeren:

- Openen van een bepaald type overzicht
- Zoeken op één of meerdere kolommen
- Sorteren op één of meerdere kolommen
- Navigeren via de paginator

Weergeven van gewenste resultaten in de datagrid kan technisch gezien op twee manieren gerealiseerd worden. Bovengenoemde gebruikersacties spelen hierbij een rol:

1. Alle gegevens worden bij het openen van een overzicht eenmalig in het geheugen van de server geladen met behulp van de overviewquery. Bij elke vervolgopdracht worden de beschikbare gegevens in het geheugen van de server gebruikt om het overzicht opnieuw van resultaten te voorzien. Mogelijke vervolgopdrachten zijn: zoeken, sorteren en navigeren via de paginator.
2. Zodra de gebruiker één van de bovenstaande acties uitvoert, zal de overviewquery aangepast en uitgevoerd worden. Hierdoor worden resultaten altijd uit de database geladen. Aanpassen van een overviewquery houdt in dat alle zoekvelden, sorteringen en huidige pagina worden toegevoegd. Hierdoor ontstaat een nieuwe overviewquery die specifiek voor de huidige situatie de juiste resultaten ophaalt.

In de techniek wordt het aanpassen van een commando ook wel parsen genoemd, en stamt af van het Engelse werkwoord to parse. Letterlijk wordt dit vertaald met ontleden. Hoe parsen van een overviewquery in zijn werk gaat, wordt verderop in deze paragraaf toegelicht. Om een keuze te maken tussen de twee mogelijkheden zijn drie criteria bekeken:

- Performance: Hoe lang duurt het voordat de juiste gegevens getoond worden?
- Courantheid van resultaten: Welke mogelijkheid zorgt ervoor dat altijd de meest accurate gegevens getoond worden?
- Implementatie: Wat moet er gedaan worden om de mogelijkheden te realiseren?

**Overviewquery parsing - Performance**

Mogelijkheid 1 zal bij het openen van een overzicht iets minder snel reageren dan mogelijkheid 2 omdat alle resultaten in het geheugen geladen moeten worden. Bij het zoeken of sorteren zal deze techniek juist sneller zijn. De hoeveelheid resultaten in het geheugen is vele malen kleiner dan de totale hoeveelheid gegevens in de database. Hoe minder gegevens beschikbaar zijn, hoe sneller zoeken en sorteren uitgevoerd kan worden. De performance van mogelijkheid 2 hangt sterk af van de overviewquery. Wanneer de overviewquery complex is, en/of er door grote hoeveelheden gegevens gezocht wordt, kan deze traag reageren. Het parsen van de overviewquery zal geen grote invloed hebben op de performance.

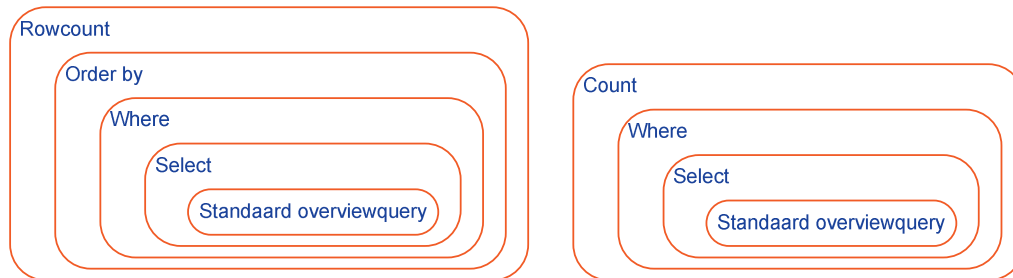
**Overviewquery parsing – Courantheid van de resultaten**

Mogelijkheid 1 maakt gebruik van tijdelijke gegevens in het geheugen. Zodra gegevens in de database door een andere Inther-LC module wordt aangepast ontvangt de webinterface hier geen bericht van. Doordat gegevens onopgemerkt veranderen, is geen garantie te geven over de courantheid van de getoonde resultaten. Mogelijkheid 2 voert altijd de geparste overviewquery uit. Dit zorgt ervoor dat bij elke actie de meest courante gegevens gebruikt worden.

**Overviewquery parsing – Implementatie**

Deze criteria is als bijzaak opgenomen: een ingewikkeldere implementatie is geen directe reden om een techniek af te wijzen. Mogelijkheid 1 behoort tot een conventionele oplossing: Laden en bewerken van gegevens met behulp van databasequeries is een algemeen gebruikte techniek. Implementatie hiervan zal naar verwachting geen moeilijkheden opleveren.

Parsen van overviewqueries is ingewikkelder. De opbouw van een databasequery, ook wel syntax genoemd, moet precies goed zijn omdat anders een foutmelding volgt. Er zijn vele uitzonderingen die getest moeten worden. Verder levert deze techniek een ander struikelblok op, namelijk: de paginator. Eerder in deze paragraaf is omschreven dat de paginator de hoeveelheid beschikbare pagina's berekent aan de hand van het totale aantal beschikbare resultaten. Een geparste overviewquery zal slechts de resultaten voor één pagina ophalen, waardoor de totale hoeveelheid niet bekend is. Als oplossing voor dit probleem wordt een tweede geparste overviewquery toegevoegd. Deze telt het totaal aantal van de resultaten met behulp van een count commando. Ter verduidelijking laat Figuur 8 een schematische weergave zien van het geparste resultaat. De linker overviewquery is noodzakelijk om de gewenste gegevens op te halen. De rechter overviewquery zal alleen de hoeveelheid resultaten tellen. Figuur 9 geeft per commando een korte uitleg.



Figuur 8: Parsed databasequeries

Commando	Omschrijving
Rowcount	Selecteren van de gewenste resultaten voor één pagina. Hierin wordt een resultaten bereik opgegeven.
Order by	Sortering per kolom, oplopend of aflopend
Where	Zoekvelden toevoegen
Select	Selecteren van alleen de gewenste kolommen voor dit overzicht. Hierop heeft de eindgebruiker geen invloed; deze zijn door de ontwikkelaar in overleg met de klant vastgelegd tijdens het ontwikkelen van het project
Count	Tellen van alle resultaten die de databasequery oplevert. Sorteren is hierbij niet nodig omdat dit geen beperking heeft tot de geselecteerde data, alleen wordt de volgorde beïnvloed
Standaard overviewquery	Databasequery welke gegevens ophaalt om deze vervolgens grafisch weer te geven in de datagrid

Figuur 9: Commando-omschrijving bij parsed databasequeries

### Overviewquery parsing – Conclusie

In de Ontwerpspecificatie is gekozen voor het parsen van de overviewquery. Doorslaggevend was de wens om altijd de meest courante resultaten weer te geven. Of de theoretische traagheid voor praktische problemen gaat zorgen, moet blijken tijdens de testfase. Mocht na afloop van de afstudeerfase blijken dat niet alle overzichten snel genoeg reageren dan bestaat de mogelijkheid om mogelijkheid 1 alsnog te implementeren naast de gekozen techniek. Per overzicht moet dan configureerbaar zijn in de database met welke methode het overzicht geladen moet worden.



## 7. Realisatie & Testfase

Alle requirements, grafische ontwerpen en technische details zijn bekend; er kan een start worden gemaakt met de implementatie van Inther-WEB. Ook zal worden toegelicht welke tests er uitgevoerd zijn ter controle van het behaalde resultaat. Dit hoofdstuk omschrijft de ondernomen stappen en moeilijkheden tijdens deze fase.

### 7.1. Realisatie

Implementeren is een iteratief proces waarbij de bedrijfsbegeleider betrokken wordt. Aangezien het ontwerp gereed is, zullen de iteraties betrekking hebben op de code zelf. Er zal met name gekeken worden naar efficiëntie, netheid en documentatie.

#### **Component**

Onderstaande lijst stelt de volgorde van implementatie van de diverse componenten vast. Om de voortgang te bewaken is de gekozen route in de wekelijkse voortgangsbespreking meegenomen.

1. Datagrid met nested functionaliteit
2. Zoekvelden
3. Sorteren
4. Menu's
5. Standaard functionaliteit (iconen)
6. Contentmenu
7. Login
8. Filter & sessie
9. Browser geschiedenis
10. Statusbar
11. Keyboard bediening
12. Navigatie links

#### **Compatibiliteit**

Implementatie met behulp van JSF 2 vereist logischerwijs ondersteuning vanuit de ontwikkelomgeving, ook wel bekend als Integrated Development Environment (IDE). Inther heeft als standaard gekozen voor het gebruik van NetBeans als IDE. Om JSF 2 in NetBeans te kunnen gebruiken moet de meest recente versie geïnstalleerd zijn, namelijk NetBeans 6.8.

JSF framework is gebaseerd op Java 2 Platform, Enterprise Edition oftewel J2EE architectuur welke te complex is om in dit document uit te werken. Het doel van J2EE is onder meer het vereenvoudigen en standaardiseren van implementatie, database verbindingen, beveiliging, ondersteuning van web services en connecties naar andere enterprise gegevensbronnen. Door het gebruik van NetBeans 6.8 is automatisch de laatste nieuwe versie J2EE 6 beschikbaar.

NetBeans 6.8 resulteert ook in het gebruik van de laatste nieuwe webserver; het programma dat de Inther-WEB applicatie draait en beschikbaar maakt via een internet browser. NetBeans 6.8 is uitgebracht met de GlassFish v3 als webserver. Binnen Inther wordt tot nu toe hooguit gebruik gemaakt van GlassFish v3 Prelude, de voorganger van GlassFish v3.

J2EE en GlassFish zijn binnen NetBeans 6.8 beide ook beschikbaar in een eerdere release versie. Hierdoor kan de ontwikkelaar, indien gewenst, ook gebruik maken van oudere technieken. In het kader van Research and Development en het oog op de toekomst is het wenselijk om gebruik te maken van de meest recente techniek.

### **Compatibiliteitsprobleem**

Bij de start van de realisatie van Inther-WEB is een nieuw project aangemaakt in NetBeans. Zoals hierboven omschreven maakt de nieuwe webinterface gebruik van de volgende technieken:

- JSF 2
- J2EE6
- GlassFish v3

Verder is aan het project de Inther-LC library toegevoegd om standaard Inther functionaliteit beschikbaar te hebben. Zodra dit project uitgevoerd wordt, treedt een exception op. Exceptions zijn foutmeldingen die de applicatie opwerpt. Aan de hand van deze fouten kan normaal gesproken de oorzaak achterhaald worden. In dit geval blijkt dat de Inther-LC library een fout veroorzaakt, verdere informatie uit de exception blijkt niet zinvol. Om deze fout op te lossen zijn de volgende stappen doorlopen:

- Zoeken op internet naar een vergelijkbaar probleem
- Controleren van configuratie in het project, JSF 2, Trinidad en de webserver
- Opnieuw een Inther-WEB project aanmaken
- Fout in Inther-LC uitsluiten door deze library van een live project te kopiëren.

Na het doorlopen van de genoemde stappen was nog niet duidelijk wat de exception veroorzaakt. Dieper onderzoek is noodzakelijk om een scherpere visie op het probleem te krijgen. Eerst moet zekerheid ontstaan dat de exception niet optreedt door nieuw toegevoegde code die gedurende dit project is toegevoegd. Bij het verdere onderzoek is gebruik gemaakt van een Hello World project welke alleen de gelijknamige tekst laat zien op een blanco pagina. Het onderzoek heeft als volgt plaatsgevonden:

#	Project	Inther-LC	Framework	Web server	J2EE	Succes?
1	Hello World 1	Nieuw	JSF 2	GlassFish v3	J2EE6	NEE
2	Hello World 2	Bestaand	JSF 2	GlassFish v3	J2EE6	NEE
3	Hello World 3		JSF 2	GlassFish v3	J2EE6	JA

Figuur 10: Testcase 1, 2 en 3

Uit testcase 1, 2 en 3 blijkt dat zodra Inther-LC niet aanwezig is het project wel functioneert. Waarom Inther-LC niet compatibel is met de huidige settings moet nog blijken. Het lijkt erop dat JSF 2 geen invloed heeft op de exception, maar dit moet uitgesloten worden door testcase 4. Het project vervolgen zonder de aanwezigheid van de Inther-LC library is geen optie.

#	Project	Inther-LC	Framework	Web server	J2EE	Succes?
4	Hello World 4	Nieuw		GlassFish v3	J2EE6	NEE
5	Hello World 5	Nieuw	JSF 2	Tomcat v6.0	J2EE5	JA
6	Hello World 6	Nieuw	JSF 2	GlassFish v3	J2EE5	JA

Figuur 11: Testcase 4, 5 en 6

Testcase 4 sluit definitief uit dat JSF 2 met de exception van doen heeft. Na alle negatieve tests blijkt testcase 5 wel succesvol. Om volledig onderbouwd een conclusie te kunnen trekken moet uitgesloten worden of webserver GlassFish v3 of J2EE6 roet in het eten gooit. De webserver Tomcat is nog niet uitgebracht met J2EE6 dus blijft alleen de optie open om GlassFish v3 te gebruiken met J2EE5. Hieruit volgt dat testcase 6 wel functioneert met Inther-LC. Nu is duidelijk geworden dat J2EE6 niet compatible is met Inther-LC.

### J2EE6 Release notes

Om de exception op te lossen moet duidelijk worden waarom J2EE6 niet compatible is met Inther-LC en de voorgaande versie J2EE5 wel. Met andere woorden: wat zijn de verschillen tussen J2EE6 en J2EE5? Het beantwoorden van deze vraag kan met behulp van de release notes van J2EE6; dit is een document waarin de producent alle toegevoegde functionaliteiten omschrijft. In de release notes blijkt dat een nieuwe techniek genaamd "Contexts and Dependency Injection for Java EE", afgekort met CDI is toegevoegd aan het framework. Globaal omschreven is CDI een techniek die koppelingen (injections) verzorgt tussen een applicatie en een web project. Vanwege de zeer technische inhoud, en beperkte meerwaarde valt de omschrijving van CDI buiten de scope van dit verslag.

Het resultaat van CDI is dat alle classes die aanwezig zijn in het project, dus ook de Inther-LC library, gescand worden op mogelijke injections. Testen of CDI het probleem veroorzaakt kan eenvoudig gebeuren door een configuratie bestand te verwijderen. Zodra de configuratie niet aanwezig is, zal CDI niet gestart kunnen worden. Testcase 7 ziet er als volgt uit:

#	Project	Inther-LC	Framework	Web server	J2EE	Succes?
7	Hello World 4	Nieuw	JSF2	GlassFish v3	J2EE6	JA!

Figuur 12: Testcase 7

Duidelijk is dat CDI de veroorzaker is van de exception. Uiteindelijk blijkt dat het probleem op te lossen is door Inther-LC uit te sluiten van de CDI scan.

## 7.2. Testfase

Tijdens een Research and Development project zijn normaalgesproken drie teststadia benoembaar, namelijk: code test, functionele test en acceptatietest. Het testen van de Java code zal door de software ontwikkelaar gebeuren die de applicatie geïmplementeerd heeft. Dit kan eventueel uitgebreid worden met een collega ontwikkelaar die de code nog niet gezien heeft om zeker te zijn dat alle uitzonderingen getest zijn. Functionele tests bepalen of de applicatie voldoet aan de gestelde specificaties waarbij de testpersoon het liefst alleen functioneel op de hoogte is van de te testen applicatie. Acceptatietests worden uitgevoerd als het product beschikbaar is op de acceptatieomgeving bij de klant. Hierdoor benaderen deze tests zo reëel mogelijk de productieomgeving waarin de applicatie uiteindelijk zal functioneren.

### Code test

Geprogrammeerde code zal aanvankelijk nooit volledig foutloos zijn, hierdoor is testen noodzakelijk. Het testen van de geschreven Java software is een continu proces dat tijdens het implementeren plaatsvindt. In het begin van de implementatiefase is nog geen webinterface beschikbaar om de achterliggende Java code te kunnen testen, waardoor tijdelijk een testomgeving noodzakelijk is. Figuur 13 laat een screenshot zien van de testapplicatie. Deze wordt gebruikt om onderstaande aspecten te testen:

- Initialiseren van achterliggende classes
- Overview gegevens ophalen
- Zoekvelden
- Sortering
- Overviewqueries parsen en uitvoeren
- Paginator
- Verwerken van de opgehaalde resultaten

Figuur 13: Testomgeving van Inther-WEB

In de code test is ook een performancetest opgenomen. Met deze performancetest wordt teruggeblikt op paragraaf 6.2 waarin de specificatie van het overviewquery parsen omschreven staat. Hoeveel tijd er nodig is om alle stappen tussen het initialiseren van een overzicht tot en met het verwerken van de resultaten wordt, blijkt uit onderstaande metingen. De gemeten tijd zal een goede indicatie zijn van de reactietijd van de complete applicatie wanneer een gebruiker een overzicht opent, zoekactie uitvoert of kolommen sorteert.

Na het meten van de responsetijd kunnen de waardes vergeleken worden:

- Een responsetijd kleiner dan 0,1 seconde wordt beschouwd als direct. Mensen ervaren dit als hetzelfde als 0 seconde.
- Bij een responsetijd tot 1 seconde blijven gebruikers geconcentreerd bij hun taak. Dit is kritiek bij ingewikkelde taken waarbij concentratie vereist is.
- Een responsetijd vanaf 10 seconde levert een afgeleide gebruiker op. Als het tegen zit stopt de gebruiker met het werk.

Deze gegevens zijn gebaseerd op een onderzoek uit 1968 door Robert Miller. Het onderzoek is al gedateerd, maar wordt nog steeds als betrouwbaar bestempeld. Het onderzoek richtte zich op de werking van de menselijke hersenen welke niet veranderd is. In 1997 voegde Peter Bickford een vierde criteria toe specifiek voor webpagina's.

- De webpagina moet binnen 8,5 seconde geladen zijn, anders verlaat de helft van de onderzochte internetgebruikers de website.

Aan de nieuwe webinterface wordt de eis gesteld dat gebruikers deze continu kunnen blijven gebruiken zonder afgeleid te worden door vertraging. Maximaal mag de responsetijd dus 1 seconde bedragen volgens bovenstaand onderzoek. Deze houdt echter geen rekening met eventuele extra vertragingen zoals een zwaar belast computernetwerk binnen een productieomgeving of het gebruik van een minder snelle cliënt. Ter compensatie wordt 20% gekort op de maximum waarde. Dit levert een controlegetal op van 0,8 seconde of 800 milliseconde.

In de huidige situatie wordt voor een aantal overzichten standaard geen gegevens getoond. Van deze overzichten is de overviewquery erg complex. In combinatie met veel beschikbare gegevens resulteert dit in onvoldoende performance om initieel alle beschikbare waardes op te halen. Wanneer een gebruiker een dergelijk overzicht wil bekijken moet eerst een zoekcriteria opgegeven worden om de hoeveelheid te doorzoeken gegevens te beperken. Hoe minder gegevens doorzocht moeten worden, hoe sneller de overviewquery zal functioneren. Door het gebruik van zoekcriteria is de performance zodanig dat deze een 'normaal' overzicht benaderd. Deze filosofie wordt ook toegepast in de nieuwe webinterface. In de database is per overzicht configureerbaar of de resultaten bij openen geladen dienen te worden.

### Meting 1:

Meting 1 is lokaal uitgevoerd met als doel vooral de Java code doorlooptijd te meten. Er is gebruik gemaakt van een eenvoudige overviewquery en beperkte hoeveelheid beschikbare informatie in de database. Hierdoor zal weinig databasevertraging ontstaan.

- Maximum: 93 ms
- Gemiddelde: 71 ms
- Minimum: 62 ms

### Meting 2:

Meting 2 benadert een real-life situatie. Vanaf een werkstation waarop de testapplicatie draait, is verbinding gemaakt naar een developmentserver waarop de database geïnstalleerd staat. Gedurende de test wordt een bestaande overviewquery gebruikt die een gemiddelde verwerkingstijd nodig heeft vanuit de database.

- Maximum: 95 ms
- Gemiddelde: 78 ms
- Minimum: 66 ms

### Meting 3:

Meting 3 beschrijft een performancetest voor de meest trage overviewquery die wel standaard gegevens laat zien wanneer deze geopend wordt. Ook deze meting is opgezet zoals meting 2.

- Maximum: 667 ms
- Gemiddelde: 642 ms
- Minimum: 621 ms

Uit meting 1 en 2 kan geconcludeerd worden dat het parsen van overviewqueries ruimschoots snel genoeg is om ingezet te gaan worden in dit project. Ook meting 3 valt binnen de gestelde grens, hoewel bij deze meting minder speling over is voor extra vertraging.

**Functionele test**

Het Testplan in bijlage VI is een stappenplan dat bedoeld is als handleiding tijdens de functionele test. Geteste requirements kunnen door de testpersoon eenvoudig worden afgevinkt in het stappenplan. Dit stappenplan is gebaseerd op het Document van Eisen welke terug te vinden is in bijlage II. Alle gerealiseerde en niet gerealiseerde functionaliteiten van de overzichtspagina zullen omschreven worden. Overige functionaliteiten als Dashboard, Inther applicaties en services zijn niet in het testplan opgenomen omdat deze buiten de scope van het project vallen. De functionele test wordt intern uitgevoerd door een niet-technische collega. Er is besloten om de functionele test op het einde van de uitloophase uit te voeren. In paragraaf 8.2 wordt omschreven waarom deze keuze gemaakt is.

**Acceptatietest**

Ter goedkeuring van een applicatie zullen eindgebruikers een acceptatietest doorlopen in de overdrachtsfase. Voor het Inther-WEB project is nog geen volledig acceptatie testplan opgesteld omdat na de afstudeerperiode niet alle functionaliteit geïmplementeerd is. Hierdoor bestaat nog geen duidelijk uitzicht op een go-life van het project, wat acceptatietesten momenteel overbodig maakt.

Normaal gesproken zijn in een acceptatietest de volgende zaken opgenomen:

- De functionele test wordt herhaald aangezien eindgebruikers anders met een applicatie omgaan dan collega's, zelfs als deze geen technische achtergrond hebben.
- Usability test waarbij getest wordt of het ontwerp voldoet aan de verwachtingen.
- Bij de performance test wordt de reactiesnelheid, geheugenverbruik en processortijd verbruik gemeten bij normaal gebruik. Ook wordt gecontroleerd of andere processen negatief beïnvloed worden.
- Een stresstest heeft dezelfde meetpunten als een performance test maar nu zal het breekpunt van de applicatie opgezocht worden, met andere woorden: Hoeveel belasting kan het systeem hebben voordat het onhandelbaar traag wordt?
- Tijdens de volumetest zal gecontroleerd worden of het systeem de gewenste capaciteit behaalt. Voor Inther-WEB is deze test minder van belang omdat de productiecapaciteit niet van de webinterface afhangt.
- Als laatste vindt een monkeytest plaats. Deze test zal welbewust misbruik maken van het systeem. Door onvoorziene situaties en omstandigheden te veroorzaken wordt gecontroleerd of het systeem voldoende stabiel is.

## 8. Overdrachtsfase

De overdrachtsfase dient gedeeltelijk als uitloop waardoor een detailplanning niet volledig in te vullen was in het Plan van Aanpak. Ook staat voor deze fase het schrijven van een technische handleiding in de planning.

### 8.1. Technische handleiding

Een technische handleiding dient als naslagwerk voor de ontwikkelaars. Deze handleiding is verdeeld in drie onderdelen:

1. Toelichting op het databaseontwerp. Hierdoor krijgt de lezer een beeld van de werking van de applicatie.
2. Toelichting op de Java code aan de hand van de ontwerpspecificatie zodat een sluitend technisch verhaal ontstaat.
3. Stappenplan waarin omschreven wordt, welke acties nodig zijn om een overzicht te configureren in de database.

Vanwege de internationale bedrijfscultuur is de technische handleiding volledig in het Engels opgeleverd. Inther maakt gebruik van een interne Wiki omgeving om documentatie centraal voor alle medewerkers beschikbaar te stellen. Wiki is een webapplicatie waarin gebruikers gezamenlijk informatie kunnen delen. Doordat alle medewerkers deze wiki omgeving goed bijhouden ontstaat een encyclopedie aan informatie.

### 8.2. Uitloop

De uitloop is gebruikt voor het afronden van de realisatie van het project. Door een optimistische planning en enkele problemen is de implementatie van enkele componenten achtergebleven. Zoals in paragraaf 5.2 omschreven staat, bleek de integratie van Trinidad moeilijker dan verwacht. Hierdoor heeft de onderzoeksfase langer geduurd dan oorspronkelijk gepland. Tijdens de Realisatie & Testfase veroorzaakte het compatibiliteitsprobleem met Inther-LC ook voor vertraging. Door de opgelopen vertraging was het niet zinvol om het Testplan uit te voeren tijdens de Realisatie & Testfase. Achteraf zal het testplan wel uitgevoerd worden om zeker te zijn van een succesvol eindresultaat.

Na afloop van de Overdrachtsfase zullen onderstaande componenten geïmplementeerd zijn:

- Nested datagrid
- Zoekvelden
- Sorteren
- Menu
- Standaard menu functionaliteit
- Login
- Sessie
- Filter



## Conclusie en aanbevelingen

### Conclusie

Na afloop van de afstudeerperiode blijkt dat het project “Building Inther-WEB Interface” nog niet volledig afgerond is. Al bij het vaststellen van de requirements in de Initiatief & Definitiefase bleek dat het niet mogelijk was om alle wensen en eisen te implementeren binnen de gestelde afstudeerperiode. Dit komt omdat er tijdens de brainstorm sessies zoveel requirements vastgesteld zijn dat dit onmogelijk haalbaar is. Ondanks dat het project niet volledig afgerond is, kan wel geconcludeerd worden dat deze afstudeerperiode succesvol was. Het technische ontwerp is erg uitgebreid en generiek opgezet, zodat dit bruikbaar blijft. Ook zijn er al diverse verbeteringen gerealiseerd:

- Indeling van de grafische componenten
- Detailweergave nested in de datagrid weergeven
- Op alle kolomen kunnen zoeken
- Sortering over meerdere kolomen
- Overviewquersys parsen met daarin de zoekvelden, sortering en paginator
- Dynamisch opbouwen van overzichten

### Aanbevelingen

Om er zeker van te zijn dat de nieuwe webinterface uitgroeit tot een bruikbare applicatie waarin klanten alle informatie en tools centraal beschikbaar hebben is het ten eerste aan te bevelen om door te gaan met de implementatie van het ontwerp. Tijdens dit project is een aantal moeilijkheden naar boven gekomen die succesvol zijn opgelost. De verwachting is dan ook dat de grootste hindernissen tijdens de afstudeerperiode genomen zijn met betrekking tot de realisatie van het volledige project.

Look and Feel van een applicatie is een veel bepalende factor tot succes. Niet-technische gebruikers beoordelen de kwaliteit van de achterliggende software bijna altijd op basis van de grafische weergave. Streven naar een zo professioneel en modern mogelijke weergave is daarom erg belangrijk. Zodra de webinterface voldoende ontwikkeld is, is het raadzaam om deze door een externe grafische ontwerper te laten modelleren. Als leidraad zouden de schetsen uit het GUI ontwerp meegenomen moeten worden zodat de indeling niet verloren gaat.

In het nieuwe ontwerp is gekozen om de standaardoverzichten te scheiden van dashboard diagrammen en tabellen. Met de nieuwe webinterface is het mogelijk om een pagina te bouwen waarin een datagrid gecombineerd wordt met een dashboard. Combineren van overzichten kan in de toekomst efficiëntere overzichten opleveren. Bijvoorbeeld een tabel waarin de huidige waardes staan met daaronder een performance meter. Hierdoor wordt in één oogopslag duidelijk wat er gebeurt op de werkvloer; ideaal voor werklasteroverzichten of behaalde capaciteit per gebruiker en machine.

Tijdens de Initialisatie & Definitiefase is er geopperd om een tool te ontwikkelen waarmee de standaardoverzichten automatisch geconfigureerd worden. Dit heeft als voordeel dat niet alleen ontwikkelaars maar alle overige collega's overzichten kunnen bouwen voor een klantproject. Vooral de Realisation afdeling heeft regelmatig het verzoek om overzichten aan te maken of wijzigen. Aangezien veel van de configuratie eenvoudiger is geworden kan een dergelijke tool sneller ontwikkeld worden.

## Evaluatie

Terugkijkend op de afstudeerperiode ben ik tevreden over het opgeleverde werk. Het doel van de opdracht is voor een groot gedeelte bereikt en het onderzoek is degelijk uitgevoerd. Hopelijk word ik zelf in staat gesteld om het project volledig af te ronden en aanwezig te zijn bij de eerste live gang op een productieomgeving.

Tijdens projecten op school heb ik al vaker de rol van projectleider gehad waardoor de fases van een dergelijk project niet nieuw zijn. Wat dit project vooral anders maakt, is dat ik persoonlijk verantwoordelijk ben voor alle deliverables. Naast de tijdrovende implementatie moest ook veel tijd gestoken worden in documentatie. Op tijd schakelen tussen de verschillende onderdelen ging gedurende het project steeds beter.

Opzetten van een sluitende planning bleek lastig. Tijdens de opstart van het project was ik enigszins overenthousiast over de vastgestelde requirements en beschikbare tijd waardoor ik teveel wilde doen. De afstudeerperiode lijkt bij aanvang lang te duren, maar is zo ten einde. Implementatie van een complex probleem duurt al snel enkele weken totdat de code bug vrij is. In de loop van het project zijn een aantal keer taken verschoven omdat onvoorziene omstandigheden voorrang behoeften. Ik heb ondervonden dat Research and Development projecten moeilijker beheersbaar zijn dan reguliere projecten. Uiteindelijk is het enthousiasme niet afgenomen over het eindresultaat alleen is er nog ruimte voor vervolg.

Persoonlijk was de grootste technische uitdaging het werken met de nieuwe techniek JavaServer Faces 2 en Trinidad omdat ik hier nog geen kennis van had. Door veel zelfstudie en begeleiding van mijn stagebegeleider heb ik veel kennis opgedaan tijdens dit project. Vooral de uitgebreide en zeer technische documentatie over JSF 2 en Trinidad is soms lastig te interpreteren.

De colleges Software Engineering kwamen goed van pas binnen dit project. Omdat tijdens de colleges en practica gewerkt is met de programmeertaal Java was dit niet nieuw voor mij. Vooral waren de colleges van kwartaal 2 nuttig omdat toen veel aandacht is besteed aan de verschillende beschikbare datamodellen en de onderlinge verschillen. Deze kennis heb ik zinvol gebruikt om de meest optimale keuzes te maken tijdens het ontwerp en implementatie. Mede hierdoor is het idee om de overviewquery te parsen haalbaar gebleken.

Gedurende mijn opleiding heb ik geen colleges gevolgd met betrekking tot databases. Sinds ik in het bedrijfsleven actief ben, zie ik dit als een gemis van de opleiding zoals ik die gevolgd heb omdat veel van de huidige beschikbare software is gekoppeld aan een database. Ik raad dan ook alle ICT studenten aan om dit keuzevak zeker te volgen.

---

## Literatuurlijst

### ***Gedrukte literatuur***

- Burns, E. en Schalk, C. *JavaServer Faces 2.0: The complete Reference*. New York: The McGraw-Hill Companies, 2010
- Laan, G. *Softwareontwikkeling met UML en Java*. Benelux: Pearson Education, 2007
- Put, van der W. *Website usability*. Benelux: Pearson Education, 2006

### ***Digitale literatuur***

- Sun NetBeans  
<http://netbeans.org/>
- JavaOne Technical Sessions TS-6457  
<http://developers.sun.com/learning/javaoneonline/2008/pdf/TS-6457.pdf>
- CodeRange  
<http://www.coderanch.com/t/486936/JSF/java/Trinidad-TableRenderingContext-nullpointer#2188046>
- Myfaces Trinidad  
<http://myfaces.apache.org/trinidad/index.html>
- IceFaces  
<http://www.icefaces.org/main/home/>
- JBoss Community RichFaces  
<http://www.jboss.org/richfaces>
- PrimeFaces  
<http://www.primefaces.org/>
- Sun Java  
<http://java.sun.com/javaee/>

---

## Bijlagen

**Bijlage I: Plan van Aanpak**

**Bijlage II: Document van Eisen**

**Bijlage III: Ontwerp Grafische User Interface**

**Bijlage IV: Java Framework onderzoek**

**Bijlage V: Ontwerpspecificatie**

**Bijlage VI: Testplan**