



Java framework onderzoek

Project Building Inther-WEB Interface

Inther Logistics Engineering B.V.

Opdrachtgever	Roel Beckers
Opdrachtnemer	Gijs van Belkom
Auteur	Gijs van Belkom
Datum	26-05-2010
Datum gereed	26-05-2010
Versie	1.0 Final
Document	2010-05-26 Java framework onderzoek v1.0.pdf

Versiebeheer

Datum	Versie	Belangrijkste wijzigingen
21-04-2010	0.1	Initiële versie
20-05-2010	0.2	Commentaar doorgevoerd
26-05-2010	1.0	Toegevoegd: JSF 2 - Technologie

Inhoudsopgave

Versiebeheer	2
Inhoudsopgave.....	3
Verklarende woordenlijst	4
1. Inleiding.....	5
1.1. Doel.....	5
1.2. Doelgroep	5
2. Onderzoeksmethode	6
2.1. De methode	6
2.2. Uitvoerbaar stappenplan.....	6
3. Frameworks inventariseren	7
4. Traditionele criteria	8
5. Benodigdheden.....	10
6. Geavanceerde criteria.....	11
7. Afweging	13
8. Compatibiliteit	15
9. Component Library	16
8.1 Component Libraries inventariseren.....	16
8.2 Criteria	16
8.3 Afweging.....	17
10. Conclusie.....	18

Verklarende woordenlijst

Term	Omschrijving
Inter-WEB	Inther webinterface; de grafische website waarin gebruikers informatie kunnen opzoeken en functionaliteit op het INHER systeem kunnen uitvoeren.
Framework	Bibliotheek van softwarecomponenten en standaarden die gebruikt kan worden tijdens het ontwikkelen van nieuwe software.
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 ontwikkel omgeving als NetBeans
Query	Commando welke uitgevoerd kan worden door een database. Hierbij is het de bedoeling dat de database een tabel met gevonden resultaten retourneert.
Servlet	Een servlet is een in Java geschreven programma dat op een web server draait. De servlet zorgt onder andere voor het afhandelen van de communicatie met de cliënt. Servlets kunnen worden gebruikt om bijvoorbeeld invulpagina's op een website te verwerken, grafieken te maken of de toegang tot een website te regelen.
Facelet	Facelets worden globaal voor dezelfde toepassing gebruikt als Servlets alleen is de technische implementatie ervan verschillend.
Stylesheet	Opmaak document; het uiterlijk van grafische componenten wordt hierin bepaald.
Runtime	Een programma in 'runtime' is opgestart / actief
Agile	Ontwikkelmethode welke gebaseerd is op korte en flexibele iteraties

1. Inleiding

1.1. Doel

In het 'Document van Eisen' is een groot aantal requirements vastgelegd. Om deze te kunnen faciliteren in het project "Building Inther-WEB Interface" moet een passend Java framework gekozen worden. Dit document beschrijft de onderzoeksmethode, welke frameworks beschikbaar zijn, welke factoren meespelen en wat uiteindelijk de conclusie van het onderzoek is.

1.2. Doelgroep

Dit document dient als bijlage in de afstudeerscriptie van het project "Building Inther-WEB Interface" waardoor alle betrokken partijen tot de doelgroep behoren. Echter beschrijft dit document de keuzes tussen technische componenten wat resulteert in technische afwegingen. Om dit correct en volledig uiteen te kunnen zetten wordt gebruik gemaakt van bijbehorende technische begrippen en afkortingen wat het lezen door niet technisch onderlegde personen moeilijker maakt. Alle begrippen zullen worden toegelicht in de hoofdstukken en zijn opgenomen in de verklarende woordenlijst met als doel dit document zo toegankelijk mogelijk te maken.

2. Onderzoeksmethode

Vele onderzoeksmethoden zijn beschikbaar om ontwikkelplatformen tegen elkaar uit te zetten, echter zijn methodieken vaak erg theoretisch waardoor de praktische bruikbaarheid afneemt.

2.1. 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 terug te vinden op de Sun website onder nummer: TS-6457, genaamd: “CHOOSING A JAVA WEB FRAMEWORK: A COMPARISON”. Deze methode zal als leidraad gebruikt worden tijdens dit onderzoek. Termen en deelvragen zijn in het Engels gesteld, deze worden letterlijk overgenomen om er zeker van te zijn dat de context onaangetast blijft. Enkele deelvragen hebben zinnige betrekking op het project, deze zijn weggelaten.

2.2. Uitvoerbaar stappenplan

Volgens de gekozen methodiek zijn onderstaande stappen noodzakelijk om een goed onderzoek te kunnen uitvoeren. In het volgende hoofdstuk wordt dit concreet uitgewerkt.

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

3. Frameworks inventariseren

Momenteel zijn er honderden Java frameworks beschikbaar waardoor een globale inventarisatie nodig is om frameworks uit te sluiten die bij voorbaat niet voldoen, zo wordt de focus gelegd op mogelijk bruikbare componenten. Tijdens dit project zal voor de inventarisatieslag gekozen worden voor de meest populaire frameworks van dit moment gebaseerd op het aanbod van de NetBeans community. Dit aanbod omvat platformen om alle mogelijke varianten van websites te implementeren; van een statische pagina tot volledig interactieve gebruikersapplicatie.

- Google Web Toolkit (GWT)
- Spring Web MVC
- Struts 2
- Groovy/Grails
- Wicket
- Vaadin
- JavaServer Faces 2 (JSF2)

4. Traditionele criteria

Traditionele criteria worden vastgesteld door onderstaande vragen te beantwoorden. Niet alle criteria zijn van doorslaggevend belang binnen dit project aangezien de volledige applicatie opnieuw gebouwd moet worden en daardoor nog bijna geen technische specificaties buiten de requirements vastgelegd zijn.

The Hello World test: What does it take to write Hello World?

Een proof of concept van het gekozen platform zou minimaal een overviewquery uit de Inther-LC database user moeten laden, executeren en de resultaten in een dynamisch gegenereerde tabel tonen. Queries zijn database commando's waarmee informatie uit tabellen kan worden opgevraagd, een overviewquery wordt door INThER gebruikt om voor een specifiek overzicht in de webinterface gegevens op te kunnen halen.

View decoration style: Does it use Templates or JavaServer Pages software/Taglibs, Java technology control of blocks?

Per gebruiker moet de style aangepast kunnen worden. Hiervoor is het gewenst om eenvoudig stylesheets te kunnen toevoegen. Verdere technieken zijn voor deze vraag nog niet gespecificeerd.

Speed of Deployment: What does it take to ship it?

De webinterface moet hardwarematig zonder uitbreidingen in gebruik te nemen zijn. Softwarematig is een deployment script noodzakelijk om nieuwe releases snel te kunnen installeren. Eenmalig moet de webserver service geïnstalleerd worden.

Speed of Development: New – What is the learning curve?

Momenteel wordt binnen Inther nog geen gebruik gemaakt van technieken welke in paragraaf 3.1 genoemd zijn. Wel is een interne registratietool gebouwd met JavaServer Faces de logische voorloper van JavaServerFaces 2. Bekende techniek of gedeeltelijk bekende techniek heeft de voorkeur aangezien hierover al enige mate van kennis aanwezig is.

Speed of Development: Agile – Is changing features easy?

Voordat gestart kan worden met programmeren worden alle features opgenomen in het technisch ontwerp. Hierdoor zou aanpassen van de feature set niet regelmatig voorkomen. Dit maakt het mogelijk om nieuwe features via een nieuwe release door te voeren.

Validation: Does it have lots of build-in validators?

Gebruikersinput van datum zoekvelden zal gevalideerd moeten worden door een datumvalidator class. Van overige zoekvelden hoeft de invoer niet gevalideerd te worden, bij foutieve invoer zal automatisch de tekst verschijnen: "Geen gegevens gevonden" waardoor de gebruiker automatisch controleert waarom deze melding optreedt.

Internationalization / Multilingualization: What style of properties? Varying scopes?

De webinterface wordt op internationaal niveau ingezet waardoor deze alle mogelijke talen moet kunnen ondersteunen. Per gebruiker moet dit configureerbaar zijn. Hiervoor is binnen Inther een techniek beschikbaar waardoor het gekozen framework zelf geen feature hoeft te bieden om meerdere talen op basis van configuratie te ondersteunen.

Scalability

Schaalbaarheid speelt een minimale rol. Met oog op de toekomst zal rekening gehouden moeten worden met minimaal 50 tot mogelijk 100 sessies. In verhouding met webserver op het internet valt dit in het niet.

5. Benodigdheden

Wat is werkelijk nodig om de webinterface te implementeren? Een algemene vraag die kan worden uitgesplitst in verschillende deelvragen. Deze deelvragen zullen niet direct meegenomen worden in de eindafweging, maar zijn een voorbereiding op de geavanceerde criteria om een zo realistisch mogelijk beeld te scheppen.

Am I building an application?

De webinterface is een applicatie welke dynamisch opgebouwd wordt wanneer de gebruiker navigeert, zoekt of functionaliteit uitvoert. Het betreft zeker geen statische website. Inther software wordt standaard ontwikkeld met een database back-end waardoor Inther-WEB uitstekend opgezet kan worden middels het drie-lagen (three-tier) architectuur model zoals hieronder staat opgesomd:

1. Grafische weergave laag op de client
2. Logische laag op de webserver
3. Database back-end

How big is my application?

De hoeveelheid beschikbare pagina's is niet van invloed op de gekozen techniek, er zijn slechts een aantal pagina's beschikbaar:

- Inlogvenster
- Hoofdpagina
- Overview
- Scada
- Dashboard

What is my usage scenario?

Momenteel zijn maximaal ongeveer 20 sessies tegelijkertijd actief. Dit kan variëren van een intensief gebruikt workstation tot het incidenteel opzoeken van informatie.

Is this likely to change?

Tijdens dit project zullen alle losse applicaties geïntegreerd worden in de webinterface, gevolg hiervan is een stijging in het aantal actieve sessies op de webserver. Hierdoor zal mogelijk rekening gehouden moeten worden met minimaal 50 actieve sessies.

Do I really need a three-tier architecture?

Aangezien Inther software bijna altijd ontwikkeld wordt met een database back-end leent Inther-WEB zich perfect voor dit model. Sowieso is een database-backend noodzakelijk aangezien daaruit de weer te geven resultaten opgevraagd moet worden.

What is the size and competencies of my team?

Het project "Building Inther-WEB Interface" is een afstudeerproject en wordt door één persoon doorlopen. Competenties en vooral de grootte van het team zijn beperkt in verhouding met normale projecten. Meer hierover is terug te vinden in het 'Plan van Aanpak'.

6. Geavanceerde criteria

Geavanceerde criteria gaan technologisch dieper in op de gestelde requirements om zo te bepalen welke technieken gebruikt gaan worden tijdens het project.

AJAX support

Ajax staat voor Asynchronous Javascript And XML en is een techniek om pagina's gedeeltelijk te gedeeltelijke requests en posts over het netwerk te hoeven versturen wat resulteert in gedeeltelijk updaten van een website zonder deze volledig te hoeven verversen. Deze techniek is een must voor de nieuwe Inter-WEB applicatie.

View Language – Java programming Language, EL, OGNL, Multiple

Weergave taal of in het Engels View Language is een containerbegrip voor alle programmeertalen die de grafische weergave van componenten regelt. Bijna alle componenten op de website zullen aangemaakt worden vanuit Java code, belangrijk is dat de view language hiervandaan gebruikt kan worden.

JavaServer Pages (JSP) en Facelets zijn een ideale oplossing omdat deze view language via een Application Programming Interface, afgekort met API aangeroepen kan worden vanuit Java code. Een API wordt gebruikt als koppeling tussen verschillende lagen software. In dit geval hoeft de Java engineer geen kennis te hebben van de exacte werking van Facelets, het aanroepen van enkele beschikbare volstaat.

Community

Aangezien de webinterface een commercieel product is moet de continuïteit gewaarborgd zijn. Zekerheid in het voortbestaan van het framework is erg wenselijk. Bij voorkeur gaat het om een framework project dat vanuit SUN opgezet is en waarvan een langdurige ontwikkelplanning bekend is.

Maturity

Ook hier is continuïteit van het eindproduct belangrijk. Aangezien dit project als Research & Development beschouwd wordt mag het framework vooruitstrevend en nieuw zijn, maar moet wel als stabiele versie zijn uitgebracht.

Pull based or Push based

Acties die door gebruikers uitgevoerd worden, ook wel event-driven genoemd, triggeren de onderliggende code welke indien nodig gegevens uit de database haalt. Het ophalen van gegevens uit de database gebeurt hoofdzakelijk alleen wanneer de gebruiker een actie uitvoert dus is er sprake van een pull based ontwerp.

Momenteel is hierop één uitzondering van toepassing; zodra een hardware component zoals een lopende band of inpakmachine een storing veroorzaakt zal een Event-Notifier bericht naar de webinterface gepushed worden.

Three-tier support – EJB/Spring and hibernate

Ondersteunen van meerdere lagen levert een goed gescheiden applicatie, een pluspunt voor onderhoudsvriendelijkheid. De implementatie hiervan kan echter nadelige gevolgen hebben aangezien sommige technieken een behoorlijk complexe implementatie en configuratie behoeven zoals Enterprise Java Beans (EJB) welke verplicht interfaces implementeert per Bean. Spring en Hibernate frameworks hebben dit nadeel niet. Ondersteuning van standaard classes, oftewel Plain Old Java Objects (POJO) heeft een hoge prioriteit.

Friendly URLs: URL mapping

URL's zijn de toegangspaden naar verschillende pagina's binnen de applicatie. Met behulp van URL's wordt navigatie mogelijk vanuit het tabmenu en detail overview links. De term Friendly URL's geeft aan dat URL's beter leesbaar zijn voor de eindgebruiker, de URL tekst wordt logischer weer te geven. Gebruikers hebben niets te maken met URL's wanneer ze de applicatie gebruiken, dus is deze techniek niet van belang.

Client/Server JavaScript programming language

JavaScript (JS) is een programmeertaal die veel gebruikt wordt om interactieve websites te ontwikkelen. Door hedendaagse technieken wordt het zichtbare gebruik van JS code voor applicatieontwikkelaars steeds verder uitgedund. Op de achtergrond is JS niet weg te denken, bijvoorbeeld AJAX technologie zoals eerder beschreven. Gewenst is een framework waarbij geen JS zelf geschreven hoeft te worden.

7. Afweging

Om een correcte en objectieve afweging te kunnen maken zullen de geavanceerde criteria uitgezet worden tegen de geïnventariseerde frameworks. De kolom maturity geeft het jaartal weer sinds de eerste release van het framework. Versienummers zijn hierin buiten beschouwing gelaten.

	AJAX support	View Language	Maturity	Community	Push or Pull	Three Tier support	Friendly URLs
Google Web Toolkit	Ja	JavaScript	2006	Klein	Pull	Nee	Mogelijk
Spring Web MVC	Ja	JSP	2002	Groot	Beide	Ja	Ja
Struts 2	Ja	JSP, OGNL	2000	Groot	Push	Ja	Mogelijk
Groovy / Grails	Nee	Java, Groovy	2006	Groeiende	Push	Ja	Nee
Wicket	Ja	OGNL	2004	Gemiddeld	Pull	Ja	Ja
Vaadin	Ja	JavaScript, JSP	2003	Gemiddeld	Pull	Ja	Ja
JavaServer Faces 2	Ja	Facelets	2004	Gemiddeld	Beide	Ja	Ja

Bovenstaande uiteenzetting bied de mogelijkheid om frameworks uit te sluiten die bij voorbaat niet voldoen. Uit de geavanceerde criteria blijkt dat de selectie minimaal moet beschikken over:

- AJAX Support
- Three Tier Support
- Push & Pull

Na het uitsorteren van de criteria blijven twee mogelijke frameworks over; Spring Web MVC en JavaServer Faces 2. Uit de traditionele criteria, benodigdheden en geavanceerde criteria blijkt dat eenvoudige configuratie en view language belangrijke factoren zijn voor onderhoud, beheersbaarheid en uitbreidingsmogelijkheden.

Spring Web MVC

Spring Web MVC maakt net als JavaServer Faces 1 gebruik van JavaServer Pages (JSP) als View Language. Dit levert als voordeel op dat standaard Java code gebruikt kan worden en er is geen JavaScript noodzakelijk om grafische componenten weer te geven. Nadeel aan het gebruik van JSP is dat elke webpagina, genaamd Servlet, gekoppeld dient te worden aan een Java class. Het koppelen heet officieel servlet-mapping gebeurt statisch in een configuratiebestand genaamd web.xml. Zodra het webproject groeit zullen vele koppelingen ontstaan wat een onoverzichtelijke configuratie oplevert.

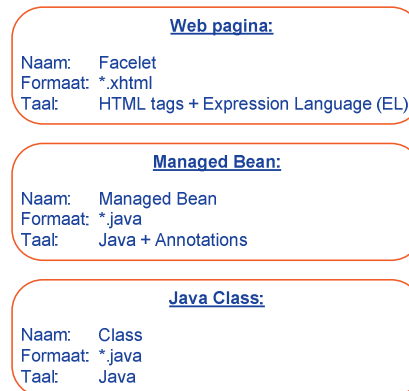
JavaServer Faces 2

JavaServer Faces 2 maakt geen gebruik van Servlets maar gebruikt een nieuwe versie webpagina's genaamd Facelets. Net als Spring Web MVC worden de achterliggende classes met standaard Java code geïmplementeerd. Bijkomend voordeel is dat JSF 2 tijdens het opstarten van de applicatie automatisch met behulp van annotations facelet-mapping toepast. Annotations worden toegevoegd in de Java code als detectievlag voor het framework. Deze nieuwe techniek zorgt ervoor dat de statische configuratie

voor een groot gedeelte komt te vervallen, dat maakt JSF 2 een bijzonder populaire techniek bij het implementeren van uitgebreide three-tier applicaties. Wanneer alle afwegingen bekeken worden blijkt dat JavaServer Faces 2 de beste keuze is voor het project "Building Inther-WEB Interface".

JavaServer Faces 2 - Technologie

Om een duidelijker beeld te scheppen over JSF 2 techniek en gebruikte termen is onderstaande schematische afbeelding opgenomen welke de opbouw van een JSF 2 project weergeeft.



Facelets bevatten de grafisch zichtbare en eventueel onzichtbare componenten. Componenten kunnen aangemaakt worden met behulp van HTML tags. Binnen de tag van een component is het mogelijk een koppeling te maken met Java code. De koppeling wordt 'binding' genoemd en verwijst altijd naar een achterliggende Managed Bean class. Het binden gebeurt met de Expression Language taal (EL).

Managed Beans worden geprogrammeerd met standaard Java code. Daarom worden deze ook wel Plain Old Java Objects (POJOs) genoemd. Deze term is niet volledig terecht omdat managed beans met behulp van annotations extra toevoegingen krijgen. De drie belangrijkste toevoegingen zijn:

- Aanduiding dat de class een managed bean is
- Naam van de bean welke in de webpagina gebruikt wordt om component binding toe te passen.
- Scope van de bean welke aangeeft hoelang de bean in stand gehouden wordt, bijvoorbeeld: session of request.

Met de release van Facelets is het mogelijk gemaakt om vanuit een managed bean direct EL functionaliteit te benaderen middels een Application programming interface (API). Een API is geschreven in standaard Java code en zorgt ervoor dat de software ontwikkelaar geen kennis hoeft te hebben van de onderliggende techniek, in dit geval EL, maar kan deze gebruiken door beschikbare functies aan te roepen in de API. Hierdoor kan een grafisch component aangemaakt worden zonder deze op de webpagina met behulp van tags toe te voegen. Dit heet het dynamisch creëren van componenten.

Managed beans kunnen gebruik maken van Java classes. Een Java class is een logische verzameling van functies en variabelen welke als bouwstenen voor een applicatie dienen. Bijvoorbeeld een verbinding opzetten met een database of gegevens van een datagrid faciliteren.

8. Compatibiliteit

Implementatie met behulp van JSF 2 vereist logischerwijs ondersteuning vanuit de ontwikkelomgeving, ook wel bekend als Integrated Development Environment (IDE). INHER 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 ondermeer 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. Binnen INHER wordt tot nu toe hooguit gebruik gemaakt van GlassFish v3 Prelude, de voorganger van GlassFish v3.

Binnen NetBeans 6.8 wordt de mogelijkheid geboden om zowel voor J2EE als GlassFish te werken met een vorige release versie. In het kader van research & development en het oog op de toekomst is het wenselijk om gebruik te maken van de meest recente techniek.

9. Component Library

Bovenop het gekozen framework kan een component library gebruikt worden. Door het toevoegen van een Component library aan JSF 2 zijn meer grafische componenten beschikbaar en/of meer functionaliteiten beschikbaar per component. Bijvoorbeeld de standaard tabel van JSF 2 kan rijen en kolommen weergeven, door het gebruik van een tabel uit een component library kan mogelijk standaard gesorteerd worden op meerdere kolommen of scrollbars zijn beschikbaar indien noodzakelijk. Hierdoor wordt het implementeren van functionaliteit die door gebruikers als 'standaard' gezien wordt eenvoudiger. Ook voor het kiezen van een Component Library zal de onderzoeksmethode uit hoofdstuk 2 gebruikt worden.

8.1 Component Libraries inventariseren

JavaServer Faces versie 2 is uitgekomen in juni 2009 met als gevolg dat er weinig component libraries beschikbaar zijn volledig gebaseerd op JSF 2. Component libraries welke op JSF 1 gebaseerd zijn met JSF 2 compatibiliteit zijn niet bruikbaar. Dit wordt veroorzaakt door het doordat JSF 1 gebruik maakt van Servlets versus Facelets van JSF 2. Het verschil hiertussen is uitgewerkt in hoofdstuk 7.

- Myfaces Trinidad 2.0.0 Alpha
- ICEfaces 2.0 Alpha 2
- RichFaces 4.0 Alpha 1
- PrimeFaces 2.0.1

8.2 Criteria

Criteria zijn gebaseerd op het "Document van Eisen" en "GUI Design". Tijdens de selectie van requirements is rekening gehouden met de standaard functionaliteit van JSF 2, alleen criteria die extra toevoegingen vereisen zijn opgenomen. Ook is J2EE 6 toegevoegd als criteria met als reden dat JSF 2 en J2EE6 nieuwe technologieën zijn waardoor niet alle component libraries compatible zijn.

- Multi level menu
- Datum picker
- Table grid met scrollbars
- Table grid met nested details
- Table grid resultaten selecteren
- Rechtermuisknop menu, ook bekend als content menu
- J2EE 6

8.3 Afweging

	Multi level menu	Datum picker	Grid met scrollbars	Grid met nested details	Grid resultaten selecteren	Content menu	J2EE6
Myfaces Trinidad	Ja	Ja	Nee	Ja	Ja	Nee	Ja
ICEfaces	Nee	Ja	Nee	Ja	Ja	Ja	Ja
RichFaces	Nee	Ja	Ja	Nee	Ja	Ja	Nee
Primefaces	Ja	Ja	Ja	Nee	Ja	Nee	Ja

Volgens het “GUI design” zijn twee componenten van doorslaggevend belang tijdens de Component Library afweging:

- Multi level menu om navigatie mogelijk te maken
- Nested weergave van detailoverzicht

Zodra deze twee vereisten worden toegepast op bovenstaande uiteenzetting blijkt dat Myfaces Trinidad de beste keuze is als uitbreidingsset op JSF 2. Een belangrijk minpunt in de functionaliteit is het ontbreken van een content menu. Verder zal moeten blijken in de nabije toekomst of dit pakket volledig uitgebouwd gaat worden tot officiële stabiele release, zoals omschreven in paragraaf 8.2 betreft de meest recente versie een ALPHA release.

10. Conclusie

Framework

Uit de afweging van het te kiezen Java framework beschreven in hoofdstuk 7 blijkt dat een ruim voldoende passende techniek gekozen kan worden. JavaServer Faces 2 biedt redelijk eenvoudige en generieke implementatie door het gebruik van standaard Java code samen met een beperkte configuratie zonder concessies te doen aan functionaliteit.

Component Library

Ter uitbreiding van de standaard functionaliteit van JSF 2 is gekozen voor Myfaces Trinidad 2.0.0 Component Library. Hier bleek een ideale situatie minder haalbaar aangezien niet alle gewenste functionaliteit voor inther-web samen aanwezig is in een van de libraries. Indien noodzakelijk bestaat de mogelijkheid om meerdere component libraries te gebruiken binnen het inther-web project. Tijdens de implementatie zal blijken of dit noodzakelijk is. Helaas wordt dan de afhankelijkheid van externe partijen vergroot. Ook kan integratie van twee verschillende libraries tot onverwachte compatibiliteitsproblemen leiden; problemen die op voorhand moeilijk te onderzoeken zijn.