



UE3 – JavaServer Faces (JSF) (25 Punkte)

Ziel dieses Übungsbeispiels ist die Erstellung einer JSF 2.1-basierten Web-Applikation, welche ein *Formel 0* Spiel realisiert. Zusätzlich zur Funktionalität aus Übung 2 soll eine Registrierung und ein Login realisiert werden.

Deadline der Abgabe via TUWEL: **Montag, 13. Mai 2013 23:55 Uhr**
Abgabegespräche: **Mittwoch, 15. Mai 2013**

Abgabemodalität

Beachten Sie die allgemeinen Abgabemodalitäten des TUWEL-Kurses. Zippen Sie Ihre Abgabe, sodass sie folgende Struktur aufweist:

- o src/main/java (Ordner)
 - ... (Java Quellcode)
- o src/main/resources (Ordner)
 - ... (Properties Dateien für Internationalization)
- o src/main/webapp (Ordner)
 - WEB-INF (Ordner)
 - web.xml (Angepasste web.xml)
 - index.xhtml (Login)
 - register.xhtml
 - table.xhtml
 - resources
 - styles
 - o screen.css (CSS-Datei der Musterlösung, aktualisiert)
 - js (Ordner)
 - o jquery.js (jQuery JavaScript Library)
 - img
 - o ... (Benötigte Bilder)
 - mycomponents
 - o login.xhtml (Custom Component)
- o pom.xml

Angabe

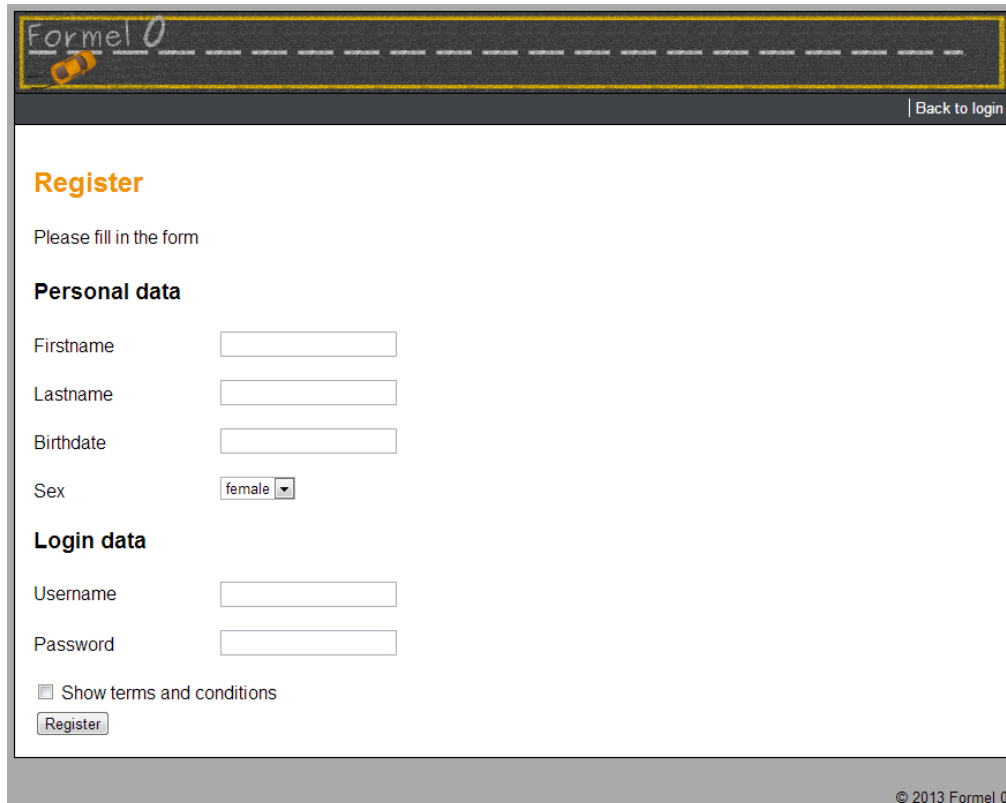
Implementieren Sie eine JSF 2.1-basierte Web-Applikation, welche mit einer bereits eigens in UE2 entwickelnden *Formel 0* API ein webfähiges *Formel 0* Spiel realisiert. Die in Übung 2 erstellte API kann weiterverwendet und erweitert werden. Sie können jedoch auch auf die zur Verfügung gestellte Musterlösung von UE2 aufbauen. Verwenden Sie als User Interface den XHTML- und CSS-Code der von uns zur Verfügung



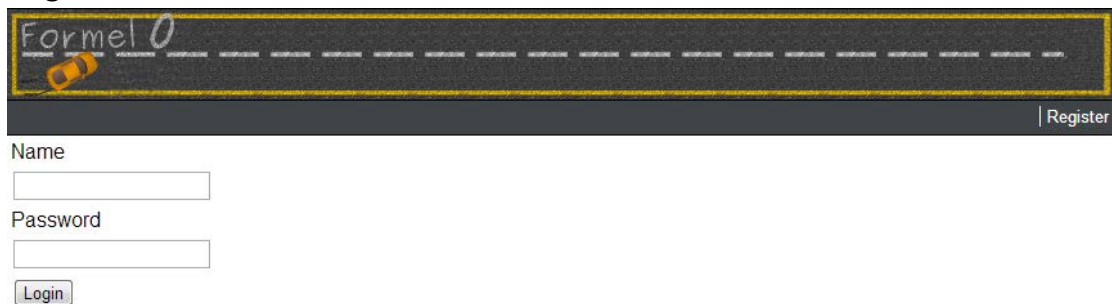
gestellten Musterlösung für Übung 1, die in den Angaberessourcen dieser Übung zu finden ist (verändern Sie keine ID Attributwerte - dies ist für die automatisierte Validierung Ihrer Lösung erforderlich). Nachfolgend werden für gewisse Elemente bestimmte Ids verlangt welche kein Vorschlag sondern Voraussetzung dafür sind, dass Ihre Lösung positiv bewertet werden kann! Bitte beachten Sie, dass durch die Verwendung von NamingContainer(z.B. h:form) die Ids deren Kinder mit der eigenen Id geprefixt werden. Wenn in den Anforderungen zu Ids der Text "respektive 'form:XXX'" steht, dann ist damit gemeint, dass die Element mit der jeweiligen Id XXX in dem NamingContainer mit der Id "form" liegen.

Anforderungen an Ihre Implementierung:

- Es muss ein *Formel 0* Spiel realisiert werden, bei dem ein Spieler gegen einen computergesteuerten Gegner spielt.
- Es muss jedoch möglich sein, dass mehrere Spieler je ein eigenes Spiel am selben Server starten können. Diese dürfen sich nicht gegenseitig beeinflussen. Testen Sie dies mit Kollegen oder zwei unterschiedlichen Browsern. (Hinweis: Mehrere Tabs innerhalb eines Browsers benutzen die selbe Session)
- Die Spielinformationen (momentan führender Spieler, aktuell gespielte Runde, bisher gespielte Zeit) müssen dynamisch neben dem Spielbrett (siehe `table.html` in den Angabe-Ressourcen) ausgegeben werden.
- Alle Dateien müssen UTF-8 codiert sein!
- Das User Interface muss natürlich soweit wie mit JSF 2.1 möglich den Anforderungen von XHTML 1.1 sowie WCAG-AA gerecht werden.
- Implementieren Sie eine Registrierung (siehe `register.html` und Abbildung) und ein Login (siehe Abbildung). Es muss keine Datenbankbindung realisiert werden. Die Registrierung aller User muss jedoch für die Dauer der Applikation (bis zu einem neuen Deploy oder Restart des Servers) erhalten bleiben. Der Benutzer soll zu Beginn auf die Login Seite kommen. Falls er noch keine Login Daten besitzt, kann der User mittels Link auf die Registrierungsseite gelangen. Nach der erfolgreichen Registrierung erscheint ein Link zur Login Seite. Bei erfolgreichem Login kann der Spieler das Spiel beginnen.



- Login-Seite:



- Alle Felder der Registrierung (Benutzername, Passwort, Vorname, Nachname, Geburtsdatum, Geschlecht) sind Pflichtfelder und müssen **serverseitig** (in JSF) validiert werden: Nur Buchstaben in Loginnamen, Vornamen, Nachnamen; Passwort muss mindestens 2 Zeichen lang sein und zumindest **einen** Buchstaben und **eine** Ziffer enthalten. Für das Geburtsdatum erstellen sie einen **Custom Validator** im Format TT.MM.JJJJ.
- Die Ids der Fehlermeldungen müssen dem Format "INPUTID_messages" respektive "form:INPUTID_messages" entsprechen wobei INPUTID durch die Id des Inputs für welche die Messages sind ersetzt wird. Wenn Sie nun ein Input mit der Id "firstname" respektive "form:firstname" haben, so müssen sich die Fehlermeldungen welche sich auf dieses Feld beziehen in einem Container(siehe h:messages) befinden welche die Id "firstname_messages" respektive "form:firstname_messages" hat.
- Formulare(h:form Tags) müssen alle die Id "form" haben, im Navigationsbereich(Kind Element des Div-Containers mit der Id

"navigation") müssen etwaige Formulare die Id "navForm" haben. Sie müssen mit maximal 2 Formularen je Seite auskommen.

- Die Eingabe Elemente "Name" und "Password"(siehe Abbildung Login-Seite) müssen jeweils die Ids "name" und "password" respektive "login:form:name" und "login:form:password" haben. Der Login Button muss die Id "submit" respektive "login:form:submit" haben. Außerdem muss die Composite Component die Id "login" verwenden. Bei einem Login-Fehler muss die Fehlermeldung in einem Container(siehe h:messages) mit der Id "login_messages" angezeigt werden.
- Die Checkbox für "Show Terms and Conditions"(siehe Abbildung Registrierung-Seite) muss die Id "showterms" respektive "form:showterms" haben und der Container in welchem die "Terms and Conditions" angezeigt werden muss die Id "terms" respektive "form:terms" haben.
- Der Link "Register"(siehe Abbildung Login-Seite) muss die Id "register" respektive "navForm:register" haben, wobei Sie das Formular in der Navigation für diese Seite nicht unbedingt benötigen.
- Die in Übung 1 clientseitige erstellte JavaScript Validierung müssen Sie aus dem Code **entfernen**.
- Klickt der User bei der Registrierung auf die Checkbox „Show terms and conditions“, so wird eine Zeile mit „Play at least once a day!“ bzw. der Deutsche Text "Spiele zumindest einmal am Tag!" angezeigt.
- Realisieren Sie den Login mittels einer **Composite Component**. Das Composite Interface muss einen Namen, ein Passwort und eine Action übergeben werden können.
- Verwenden Sie zur Erstellung der Seiten **Facelet Templates (ui:composition template)** und **Composition (ui:include)** - Achtung Compositions sind etwas anderes als Composite Components.
- Gestalten Sie das Spiel so, dass es sowohl in Englisch als auch in Deutsch angezeigt werden kann (Internationalization I18n)
- Beim Würfeln soll nicht die gesamte Seite neu geladen werden sondern nur der Spielbereich und die Spielinformationen neu übertragen werden (AJAX Component)



Angabe

Die Angaberessourcen können in TUWEL heruntergeladen werden. Diese beinhalten die folgenden Dateien:

- UE3-Angabe.zip
 - UE3.pdf (Dieses Dokument)
 - src/main/webapp
 - WEB-INF (Ordner)
 - web.xml
 - register.html
 - table.html
 - resources
 - styles
 - screen.css (CSS-Datei der Musterlösung, aktualisiert)
 - js (Ordner)
 - jquery.js (jQuery JavaScript Library)
 - img
 - ... (Benötigte Bilder)
 - src/main/java/formel0api (Ordner)
 - *.java (formel0 api java Dateien)
 - pom.xml (Maven Projekt Deskriptor)

Hinweis

Testen Sie ihre Abgabe in den Browsern Firefox und Chrome.

Das JSF2.0 Sample Store Beispiel, welches Sie als ZIP auch über TUWEL runterladen können, zeigt Ihnen die Verwendung von JSF. Es wird daher empfohlen, sich dieses Beispiel und den Source Code genau anzusehen. Zusätzlich zu den Folien des JSF Gastvortrages empfehlen wir Ihnen, die BIG-internen Folien (vor allem zusätzlich Templates [S 27], Compositions [S 28], Composite Components [S 29] und Internationalisierung I18N [S 46]) und das Beispiel JSF2.0 Sample Store durchzusehen. Wir sind uns im Klaren, dass einige dieser Features in der Vorlesung aus Zeitgründen nur angeschnitten wurden, sie aber dennoch einen wichtigen Teil von JSF 2.1 darstellen und mit den angegebenen Ressourcen leicht zu verwenden sind. Es wird Ihnen viel zur Lösung der Übung beitragen. Des Weiteren ist <http://jsfatwork.irian.at/> sehr zu empfehlen.

Das Maven Projekt können Sie mithilfe von Netbeans öffnen (File Project) und ausführen. Die .war Datei finden Sie im Ordner „target“ des Projekts.

→ Open

Sie können Ihre Entwicklungsumgebung (IDE) frei wählen, wir empfehlen jedoch NetBeans 7.3 Java EE (<https://netbeans.org/downloads/>) mit Tomcat 7 und JDK 7 zu verwenden. Bedenken Sie, dass Projekt auf Tomcat 7 laufen muss.



Verwenden Sie zur Validierung Ihrer XHTML-Seiten das vom W3C zur Verfügung gestellte Validation Service unter <http://validator.w3.org/>. Zur Überprüfung der WAI-Tauglichkeit steht Ihnen eine Vielzahl von Services im Internet zur Verfügung. Einige Überprüfungsmethoden und Hilfsmittel wurden in der Vorlesung vorgestellt (siehe Vorlesungsfolien).