

	Datum	07.12.2022
Informatik Web Application Engineering	Übungsblatt 9	
Thomas Feilhauer	Zu lösen bis 14.12.2022, 8:00 Uhr	

Vorbereitungsaufgabe (in der Übungsstunde zu lösen):

Entwickeln Sie eine Web-Applikation mit Java auf Basis des MVC-Prinzips, mit der sich eine Überweisung in einem Online-Banking-System simulieren lässt. Über eine allgemeine Online Banking Startseite soll eine Überweisung ausgewählt werden können. Im Überweisungsformular sollen Kontonummer, Bankleitzahl und Betrag eingegeben werden können. Es soll eine Server-seitige Überprüfung stattfinden, ob die Bankleitzahl genau acht Stellen hat und der zu überweisende Betrag größer als 1000 € ist. Im Falle einer korrekten Eingabe soll die Transaktion in eine Datei geschrieben werden und eine entsprechende Bestätigung an den User zurückgegeben werden. Im Fehlerfall sollen geeignete Fehlerseiten das Problem beschreiben.

Folgende Web-Seiten können beim Lösen der Aufgabe(n) nützlich sein:

<https://netbeans.org/kb/docs/web/mysql-webapp.html>

<http://www.torsten-horn.de/techdocs/jee-tomcat-eclipse.htm>

Aufgabe 14: User -Authentifizierung (10 Punkte)

Erweitern Sie Ihre *Bibliotheksverwaltungs-Web-Site* um die Möglichkeit, neue User in das System aufzunehmen und einen User-Logon.

Die Einträge des User-Antrag-Formulars sollen nun nicht mehr per E-Mail an Sie geschickt werden, sondern automatisch in einer entsprechenden Datenstruktur abgelegt werden, die über eine Variable im **Application-Scope** bereitgestellt wird. Dabei muss überprüft werden, ob es einen User mit der angegebenen User-ID bereits gibt. Gibt es die User-ID noch nicht, wird ein neuer Datensatz in der Datenstruktur angelegt (es genügt Vorname, Nachname, UserID und Passwort in die Datenstruktur einzutragen), ansonsten erfolgt eine Rückmeldung an den User, dass eine andere User-ID gewählt werden muss. Kann die neue User-ID angelegt werden, soll dem User zur Bestätigung eine kurze Begrüßungsseite mit seiner User-ID angezeigt werden.

Legen Sie einen Test-User an (auslesbar aus einem Config-File oder hardcoded) mit folgenden Credentials – User-ID: user, Passwort: pass.

Durch die User-Authentisierung sollen die Benutzer:Innen erst nach erfolgter Authentisierung Zugriff haben auf die Unterseiten mit Ausnahme der Unterseite mit dem Account-Antrag-Formular. Dazu muss ein noch nicht eingeloggter User beim Aufrufen der Unterseiten auf eine Login-Seite umgeleitet werden. Überprüfen Sie dazu die vom User eingegebenen User-ID und Passwort mit den in der Applikation registrierten User-Einträgen. Nur bei korrekter Eingabe von gültigen User-ID und Passwort darf die Unterseite angezeigt werden. Der Login soll nur einmal zu Beginn der Session erforderlich sein, danach kann jederzeit ohne erneuten Login auf sämtliche Unterseiten zugegriffen werden.

Implementieren Sie die vorzunehmenden Auswertungen und Operationen in Java (**Servlets, JSPs, JavaBeans**) auf Basis einer geeigneten **MVC-Architektur**. Verwenden Sie in den JSPs nur Tags und Ausdrücke, ggf. mit EL, keine Scriptlets. Greifen Sie aus den JSPs heraus nur lesend auf JavaBeans zu und rufen Sie keine

Methoden auf. Alle eingehenden Requests sollten vom Controller-Servlet entgegengenommen und verarbeitet werden. Jeder Response an den User soll aus einer entsprechenden JSP heraus erzeugt werden.

Verwenden Sie für die Implementierung **JakartaEE 9+** mit dem **jakarta**-Namespace (nicht die Klassen im javax-Namespace).

Dokumentieren Sie, welchen Web-Container Sie einsetzen sowie das Deployment ihrer Web-Applikation darauf.

Testen Sie das Deployment der WAR außerhalb der IDE und dokumentieren Sie die resultierende URL, unter der die Einstiegsseite Ihrer Webapplikation vom localhost im Web-Browser aufgerufen werden kann.

Neben dem Source-Code und der Dokumentation reichen Sie bitte ein deploybares WAR-File zusammen mit Ihrer Lösung ein.

Falls Sie JSTL für die Übungsaufgabe einsetzen wollen, hier noch ein Hinweis für das Deployment von WebApps mit JSTL unter Tomcat 10:

Download jakarta.servlet.jsp.jstl-api-3.0.0.jar & jakarta.servlet.jsp.jstl-3.0.1.jar:

<https://repo.maven.apache.org/maven2/org/glassfish/web/jakarta.servlet.jsp.jstl/3.0.1/jakarta.servlet.jsp.jstl-3.0.1.jar>

<https://repo.maven.apache.org/maven2/jakarta/servlet/jsp/jstl/jakarta.servlet.jsp.jstl-api/3.0.0/jakarta.servlet.jsp.jstl-api-3.0.0.jar>

Stellen Sie jakarta.servlet.jsp.jstl-api-3.0.0.jar & jakarta.servlet.jsp.jstl-3.0.1.jar für das Deployment in das /WEB-INF/lib Verzeichnis der zu deployenden Web-Applikation ein.

Die Aufgabe wird in den folgenden Übungsblättern fortgesetzt.