



TECHNISCHE UNIVERSITÄT BERLIN

Software Engineering for Embedded Systems Group – Prof. Dr. Sabine Glesner

www.pes.tu-berlin.de Secr. TEL 12-4 Ernst-Reuter-Platz 7 10587 Berlin



Softwaretechnik und Programmierparadigmen WiSe 2014/2015

Prof. Dr. Sabine Glesner

Joachim Fellmuth

Dr. Thomas Göthel

Lydia Mattick

Tutoren

joachim.fellmuth@tu-berlin.de

thomas.goethel@tu-berlin.de

lydia.mattick@tu-berlin.de

Hausaufgabenblatt 3

Ausgabe: 23.01.15

Abgabe: 15.02.15

Die von euch entworfene Lösung für das Konferenzportal wurde vom Auftraggeber abgesegnet und nun soll es an die Implementierung gehen. Zur Erinnerung: Die ermittelten Anforderungen waren die Folgenden:

Ein Nutzer kann sich unter Angabe seines Namens und Kontaktdaten in der Software registrieren. Er muss dabei ID und Passwort wählen, mit denen er sich von da an am System anmelden kann. Nutzer dürfen Vorsitzende von Konferenzen, Autoren und/oder Reviewer von Papers sein.

Die Vorsitzende einer Konferenz kann diese im System registrieren. Dazu werden mehrere Daten benötigt:

- der Name der Konferenz,
- ID bzw. Kurzname,
- der Termin und Ort für die Tagung der Konferenz,
- der Beschreibungstext,
- die erste Deadline - für die Einreichung von Papers,
- die zweite Deadline - für die Reviews der eingereichten Papers und
- die maximale Seitenanzahl für Papers.

Die ID(Kurzname) darf nicht mit der einer anderen Konferenz übereinstimmen. Bei den Terminen ist sicherzustellen, dass die zweite Deadline mindestens vier Wochen nach der ersten Deadline ist, und der Konferenztermin selbst wiederum mindestens vier Wochen nach der zweiten Deadline. Spätere Verschiebungen der drei Termine kann die Vorsitzende unter Einhaltung dieser Bedingungen durchführen.

Autoren können ihre Papers in Textform hochladen. Dazu muss zusätzlich der Titel, das Verfassungsjahr und das Abstract eingegeben werden. Die Seitenzahl wird für den Inhaltstext automatisch bestimmt und gespeichert. Der Inhalt des Papers darf nachträglich angepasst werden. Autoren können außerdem alle registrierten Konferenzen einsehen. Ein noch nicht veröffentlichtes Paper darf bei nur einer Konferenz bis zur ersten Deadline eingereicht werden, wenn es die vorgesehene maximale Seitenzahl nicht übersteigt. Ein eingereichtes Paper darf zwischen der ersten und zweiten Deadline und, falls es akzeptiert wurde, nach Abschluss der Konferenz nicht angepasst werden.

Nach Ablauf der ersten Deadline kann die Vorsitzende den Papers Nutzer als Reviewer zuweisen. Ein Reviewer darf nicht Autor desselben Papers sein. Die Auswahl der Reviewer und ihre Anzahl liegt im Ermessen der Vorsitzenden und wird vom System nicht modelliert. Bis zur zweiten Deadline hat ein Reviewer Zeit, ein Review für ein ihm zugewiesenes Paper einzureichen. Es gibt außerdem die Möglichkeit, ein Review abzulehnen, um der Vorsitzenden zu signalisieren, dass ein neuer Reviewer zugewiesen werden sollte. Ein Review beinhaltet einen Text und zusätzlich die Bewertung *accepted*, *borderline* oder *rejected*.

Die Bewertung eines Papers wird aus dessen Reviews berechnet. Jede Reviewbewertung kann in Punkte umgerechnet werden. Die Vorsitzende kann den Punkteschlüssel und die Zusagekriterien anpassen:

- die Punktzahl für ein *accepted* Review, voreingestellt auf 3 Punkte,
- die Punktzahl für ein *borderline* Review, voreingestellt auf 1 Punkt,
- die Punktzahl für ein *rejected* Review, voreingestellt auf -2 Punkte,
- die minimale Gesamtpunktzahl, voreingestellt auf 3 Punkte,
- der minimale Durchschnitt der Punktzahlen, voreingestellt auf 1 Punkte und
- die Option, nur Papers von Autoren zu akzeptieren, welche bereits eine Mindestanzahl von Papers veröffentlicht haben. Diese Einstellung ist voreingestellt auf 0 Papers.

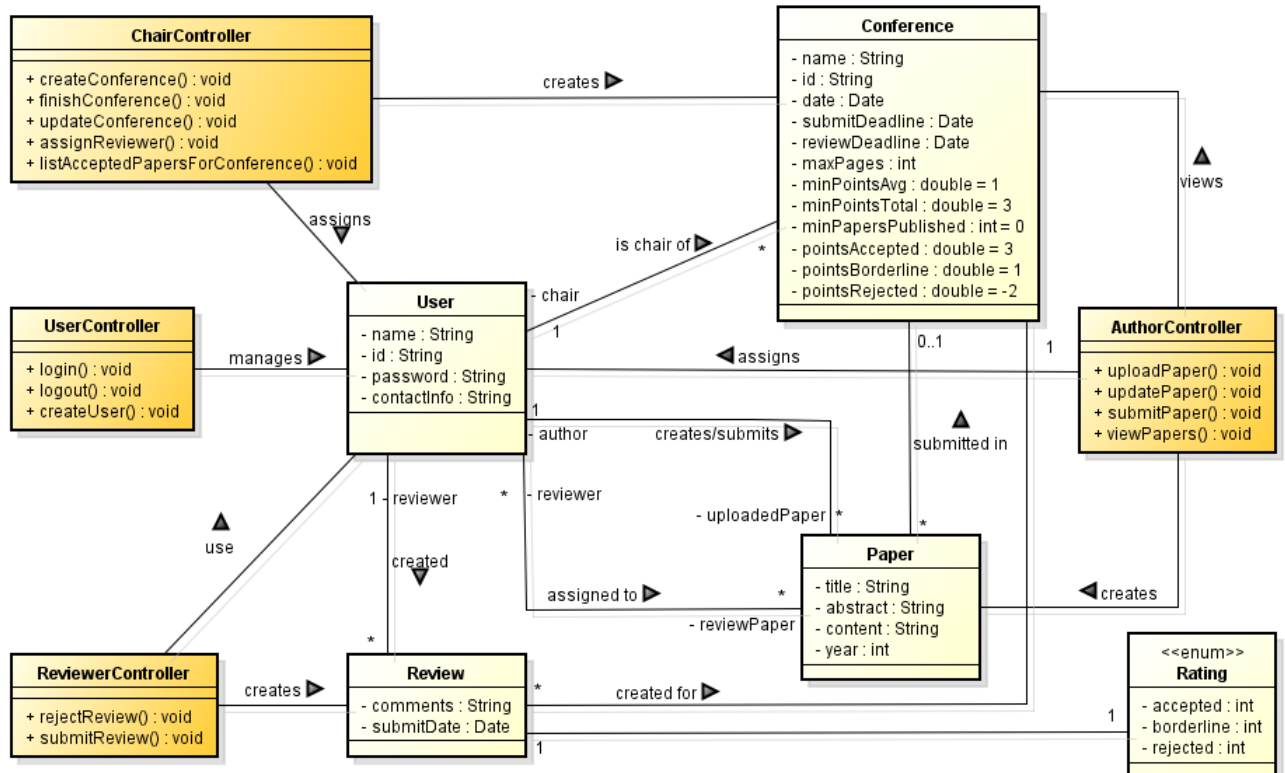
Die Vorsitzende kann nach der zweiten Deadline die Konferenz abschließen. Solange die Konferenz noch nicht abgeschlossen wurde, können sowohl der Punkteschlüssel als auch die Zusagekriterien geändert werden. Die Vorsitzende kann sich alle Papers anzeigen lassen, welche momentan die Zusagekriterien erfüllen: Dadurch kann sie die Anzahl der akzeptierten Papers indirekt beeinflussen, um der Konferenz gerecht zu werden.

Sobald die Vorsitzende eine Konferenz abschließt, werden vom System mehrere Aufgaben erledigt: Alle Papers, welche die Zusagekriterien erfüllen, bleiben mit der Konferenz verknüpft. Die Autoren dieser Papers werden über die Zusage benachrichtigt. Alle übrigen Papers werden von der Konferenz getrennt. Betroffene Autoren erhalten dann die Nachricht, dass ihr Paper bei einer anderen Konferenz eingereicht werden darf. Die von getrennten Konferenzen erhaltenen Reviews werden nicht vom Paper getrennt, beeinflussen aber nicht die Bewertung bei der momentan verknüpften Konferenz. Nach dem Konferenztermin werden alle akzeptierten Papers veröffentlicht.

1. Implementierung

Zur Umsetzung der Anforderungen ist eine Web-Anwendung zu implementieren, die in einem Apache Tomcat Server ausgeführt werden kann. Die technischen Anforderungen sind in den Hinweisen zu finden.

Ihr könnt für die Implementierung das folgende Klassenmodell verwenden. In diesem sind allerdings Rückgabetypen und Parameter der Operationen bisher nicht berücksichtigt, außerdem wurden bisher nur die Attribute aufgeführt, die im Text direkt erwähnt werden. Eure Implementierung kann auch vom Klassenmodell abweichen. In dem Fall muss die Abgabe mit einem Klassenmodell dokumentiert werden.



Für den Anfang haben wir Euch ein Eclipse Servlet/JSP-Projekt zusammengestellt, in dem schon einige Use-Cases exemplarisch implementiert sind. Im Projekt existiert auch eine README-Datei, die einige Informationen zur Orientierung enthält. In den schon vorhandenen Klassen markieren TODO-Kommentare die Stellen an denen Ihr weitermachen könnt. Selbstverständlich fehlen auch noch Klassen/JSP-Dateien.

Da es nicht Inhalt dieser Veranstaltung ist, die konkret verwendeten Technologien zu vermitteln sondern einmal die bisher erarbeiteten Modelle umzusetzen, gibt es folgende Vereinfachungen und Hinweise:

- Es gibt keine Punkte für die GUI. Sie dient ausschließlich dazu, dem Benutzer die Anwendungsfälle zur Verfügung zu stellen und die nötigen Ausgaben anzuzeigen. Das einfache Layout aus dem Beispiel kann also auch für die restliche Implementierung verwendet werden.
- Wir testen nur die Funktionalität der Anwendung entsprechend der Aufgabenstellung. Es müssen keinen Sicherheits-Aspekte implementiert werden. Es ist also ok dass das Passwort der Benutzer nicht verschlüsselt übertragen wird.
- Die Anwendung muss die Daten nicht permanent speichern können, also keine Datenbanken etc.
- Die Seitenzahl darf vereinfacht als die Anzahl der Zeichen des Texts durch 3000 ermittelt werden.
- Die Benutzer können über Ereignisse informiert werden, indem die Methode `informUser(...)` der Klasse `User` mit einer entsprechend aussagekräftigen Nachricht aufgerufen wird.
- Für Use-Cases, die der Benutzer ausführt, sollen Vorbedingungen unter Verwendung der Systemzeit geprüft werden. Wir müssen aber trotzdem in der Lage sein z.B. Funktionalitäten zu testen die mit Konferenzen nach ihren Deadlines durchgeführt werden. Dafür sind DEBUG-Methoden zu erstellen mit denen das System Test-Objekte unabhängig der Deadlines anlegen kann. Ein Beispiel findet ihr in `ConferenceServlet.createDebuggingData()` in Form von `createConference()` und `createConferenceDEBUG()`. In letzterer sollen die Deadlines beim Erstellen nicht geprüft werden. Der Kunde soll die DEBUG-Varianten natürlich nicht erreichen können.

Hinweis: Abzugeben ist das Projekt als exportiertes eclipse-Projekt im .zip-Format. Bitte benennt die Pakete um: von example nach tutXX_grX. Benennt auch das Projekt um so dass es den Gruppennamen enthält.

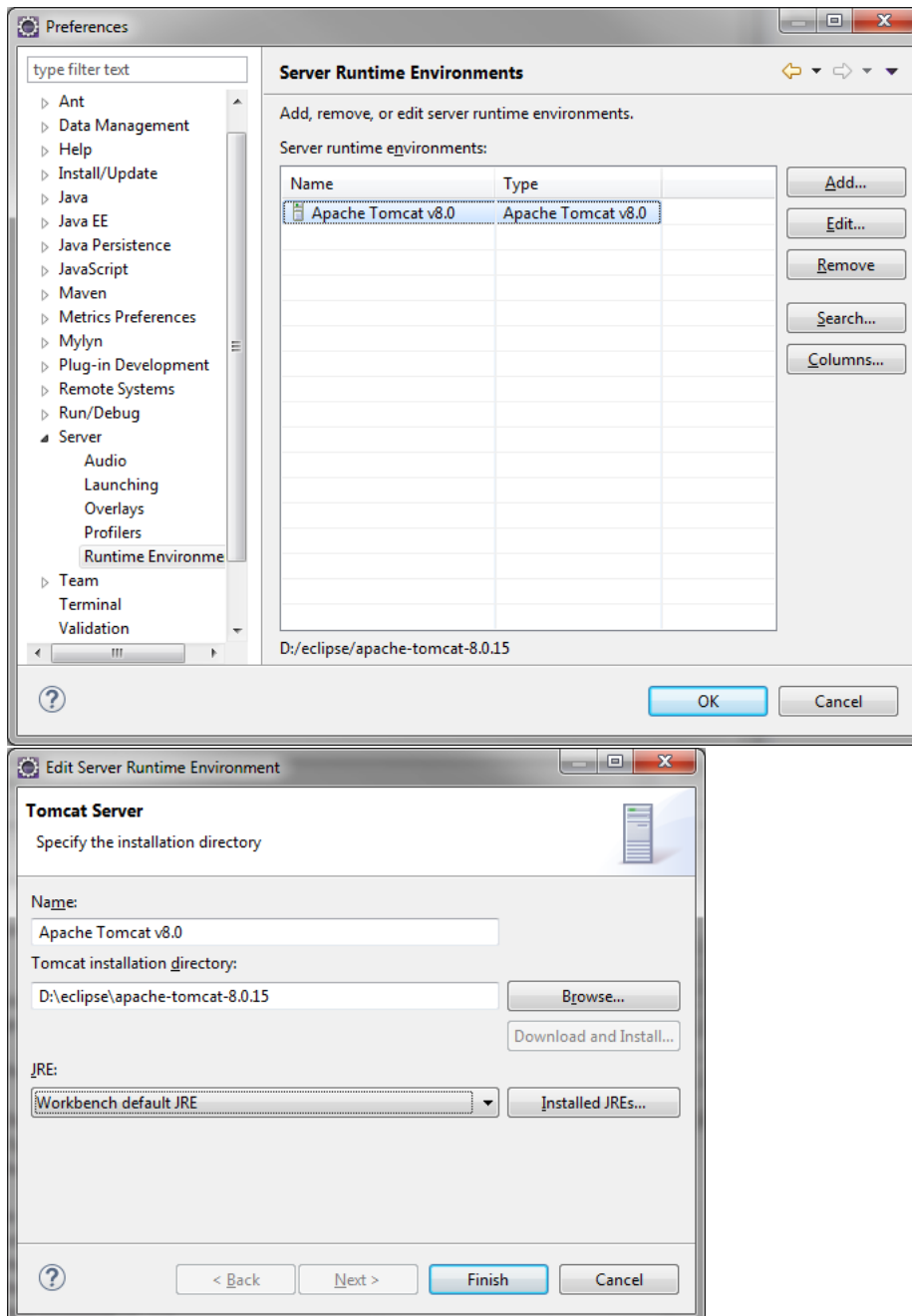
Hinweise zur Einrichtung des Systems:

Ihr benötigt

- JDK 7 (JRE reicht für Tomcat nicht aus)
- Eclipse Luna in der J2EE-Version, <http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/lunasr1a>
- Tomcat 8 <http://tomcat.apache.org/download-80.cgi>

Anleitung:

- Eclipse und Tomcat herunterladen und entpacken
- Linux: .bashrc editieren: `export CATALINA_HOME=TOMCAT-ROOT-ORDNER` hinzufügen, wobei TOMCAT-ROOT-ORDNER der Ort ist an dem Tomcat entpackt wurde
- In eclipse muss dann unter Window → Preferences → Server → Runtime Environment Apache Tomcat v8.0 über den Add-Button hinzugefügt werden. Nach dem Klicken des Next-Buttons muss der Tomcat-Installationspfad angegeben werden. (Siehe Bilder unten).
- Um das Projekt in eclipse zu importieren muss die zip-Datei zunächst entpackt werden und dann unter File → Import → General → Existing Projects into Workspace, nach dem Klicken von Next unter Select root directory der entpackte Ordner ConferencePortal ausgewählt werden.
- Vor dem Ausführen muss mit Rechtsklick auf das eclipse-Projekt unter Properties → Java Build Path unter dem Reiter Libraries die JRE System Library die JDK 7 hinzugefügt werden (Button Add Library).
- Bevor das Projekt ausgeführt werden kann, muss unter Run Configurations → Apache Tomcat der Tomcat v8.0 Server at localhost ausgewählt werden.
- Das Projekt wird auf dem Server dann mit Rechtsklick auf `ConferenceServlet.java` → Run as... → Run on Server ausgeführt.



2. Metriken

Erstellt für die beiden Funktionen mit der höchsten Komplexität nach McCabe eurer Implementierung aus Aufgabe 1 den Kontrollflussgraphen und stellt dar, wie die jeweilige Komplexität berechnet wird.

3. Testen

Weist eure Qualitätssicherung für die Implementierung aus Aufgabe 1 nach, indem ihr JUnit-Testfälle für das Programm erstellt. Das Beispielprojekt enthält in dem Paket Test schon ein paar beispielhafte Test-Fälle. Die Test-Fälle sollen nur für die Controller-Klassen (Die Use-Cases!) des Klassendiagramms erstellt werden und Zweigüberdeckung erreichen. Natürlich sollte bei Abgabe von den Testfällen keiner fehlschlagen.

Generelle Hinweise und Abgabeformalitäten

- Wir vergleichen eure Abgabe **nicht** mit einer Musterlösung, da es meist mehrere gültige Lösungsmöglichkeiten gibt.
- Fügt eurer Abgabe eine Titelseite mit euren Namen, dem Namen eures Tutors und eurem (offiziellen) Tutoriumstermin hinzu.
- Wiederholer von „MPGI 3 - Softwaretechnik“, welche ihre Hausaufgabennote behalten wollen, schreiben ihren Namen / ihre Matrikelnummer **nicht** auf die Abgabe.
- Diese Hausaufgabe ist eine Prüfungsleistung! Wenn ein (gruppenübergreifendes) Plagiat entdeckt wird, resultiert dies in einem Fehlversuch.
- Die Abgabe erfolgt pünktlich über Isis, als .pdf bzw. .zip (Eclipse-Projekt).