

Entwurfsbeschreibung

1 Allgemeines

Unsere Softwarestudie ist eine serverseitige Webanwendung, die verschiedene Funktionen zum Prüfungsmanagement zur Verfügung stellt. Die Anwendung besteht aus JavaServer Pages, Servlets und weiteren Java-Klassen. Diese werden als Servlet-Spezifikation in eine Webcontainer-Umgebung (z.B. Tomcat) integriert und über diese ausgeführt. Dazu reicht auf Client-Seite ein normaler Web-Browser.

Die Speicherung von persistenten Daten (die Nutzer- und Prüfungsdaten) erfolgt in XML-Dateien.

Wir haben eine Trennung der Funktionalität, im Sinne des MVC-Modells, genutzt. Wir unterscheiden ein Model (die Java-Klassen), ein View (die JSP's) und einen Controller (die Servlets). Der View ist dynamisch und passt sich dem jeweiligen Benutzer an.

2 Produktübersicht

Für das Prüfungsmanagement unterscheiden wir verschiedene Rollen: Student, Prüfer und Prüfungsamt. Ein Nutzer registriert sich, kann sich einer dieser Rollen zuordnen und sich anschließend anmelden.

Je nach Rollen unterscheiden sich die Funktionen, die dem Nutzer zur Verfügung stehen.

Student:

- Übersicht über eingetragene Prüfungen einsehen
- Ein- oder Austragen in veröffentlichte Prüfungstermine
- Übersicht über aktuelle Prüfungsanmeldungen einsehen

Prüfer:

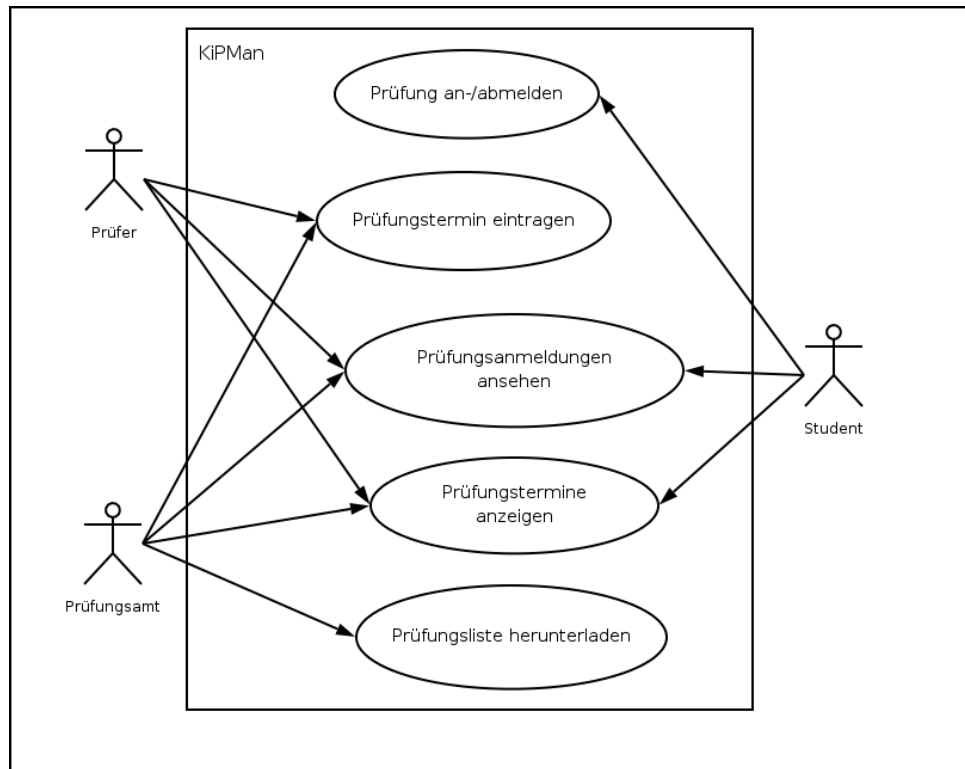
- Übersicht über eingetragene Prüfungen einsehen
- Prüfungstermine veröffentlichen (Unterscheidung zwischen mündlicher oder schriftlicher Prüfung)
- Hinzufügen weiterer Termine zu einer mündlichen Prüfung
- Übersicht über aktuelle Prüfungsanmeldungen einsehen

Prüfungsamt:

- Übersicht über eingetragene Prüfungen einsehen
- Prüfungstermine veröffentlichen (Unterscheidung zwischen mündlicher oder schriftlicher Prüfung)
- Hinzufügen weiterer Termine zu einer mündlichen Prüfung
- Übersicht über aktuelle Prüfungsanmeldungen einsehen

- Liste der aktuellen Prüfungstermine als XML-Datei anzeigen und abspeichern

Hier eine kleine Übersicht über die wichtigsten Funktionen in Form eines Use-Case-Diagramms.



3 Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

Alle Dateien liegen, als Web-Archiv in dem Servlet-Container "Tomcat" integriert, auf dem Server. Das System ist funktional nach dem MVC-Model aufgeteilt. Der View bildet sich aus den JavaServer Pages. Diese werden mit Hilfe des JSP-Compilers Jasper in Servlets übersetzt und dann im Container ausgeführt. Die Servlets nehmen die Anfragen des Clients entgegen und senden eine Antwort. Der Inhalt der Antwort kann dabei dynamisch erstellt werden. Der Nutzer auf der Client-Seite erhält html-Code, der in seinem Browser dargestellt werden kann. Über die JavaServer Pages können auch direkt Servlets aufgerufen werden, welche den Controller bilden und auch im Servlet-Container ausgeführt werden.

Ein Servlet ist eine von `javax.servlet.http.HttpServlet` abgeleitete Java-Klasse. Die beiden Methoden `doGet` und `doPost` der Superklasse werden überschrieben um die beiden wichtigsten http-Methoden GET und POST verarbeiten zu können. So können die Servlets z.B. über eine Formulareingabe aufgerufen werden und diese leiten dann die Verarbeitung der übergebenen Daten. Außerdem kann ein Servlet auch über einen Link aufgerufen werden.

Die Metadaten der Servlets, z.B. über welchen Namen diese aufgerufen werden, werden in dem Deployment Descriptor (`web.xml`) gespeichert.

Zur Verarbeitung der Daten greifen die Servlets auf das Model zurück, welches aus weiteren Java-

Klassen besteht. Das Speichern der Daten erfolgt über Klassen im Model unseres Systemes. Dazu nutzen wir die JDOM-Bibliothek (www.jdom.org) und speichern die Daten in XML-Dateien.

4 Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

4.1 Das LoginServlet

Das LoginServlet ist für alle Funktionen zuständig, die auf die Daten der Nutzer zugreifen. Die Startseite (`index.jsp`) bietet dem Nutzer die Möglichkeit sich anzumelden oder sich zu registrieren (`registration.jsp`). Beide Funktionen werden durch das LoginServlet bearbeitet. Dieses wird also jeweils über den Submit-Button eines Formulars aufgerufen und die eingegebenen Daten werden dem Servlet übergeben.

Das LoginServlet ruft nun Methoden der `UserEdit.java` auf, welche entweder einen neuen Benutzer anlegt (bei einer Registrierung) oder nachprüft, ob der sich anmeldende Benutzer schon registriert ist. Dies geschieht immer über die `XMLEdit.java`, welche für das Schreiben in und das Lesen aus den XML-Dateien zuständig ist.

Nach erfolgreichem Anmelden gelangt der Nutzer zur Prüfungsübersicht (`overview.jsp`).

4.2 Das ExamServlet

Das ExamServlet ist für die Funktionen zuständig, die auf die Daten der Prüfungen zugreifen. Je nach Rolle des Benutzers stehen ihm unterschiedliche Funktionen nach der erfolgreichen Anmeldung zur Verfügung und je nachdem ist auch die Prüfungsübersicht unterschiedlich aufgebaut. Jedem Benutzer erscheint die Liste mit den veröffentlichten Prüfungsterminen. Die dynamische Generierung dieser Übersicht wird durch die JavaServer Page `overview.jsp` über den Servlet-Container realisiert.

Die weiteren Funktionen laufen über das ExamServlet. Dazu werden Methoden der `ExamEdit`-Klasse aufgerufen, welches wieder über die `XMLEdit.java` die XML-Dateien modifiziert.

`Examview.jsp` generiert sich ebenfalls selbst und zeigt alle Informationen zur gewählten Prüfung an.

`Addexam.jsp` und `addappointment.jsp` stellen jeweils ein Formular zur Verfügung und über einen Submit-Button wird das ExamServlet aufgerufen, welches diese Eingaben verarbeitet.