Simple Student Management System

Projektmitglieder: Ralph Sandmair, Bianca Mauthner

Contents

1 Introduction

1.1 Identifying information

Der Student Manger ist eine Server-Client Architektur, welche es Lehrer sowie Studenten ermöglicht ihre Daten zu verwalten.

1.2 Supplementary information

The web application must be able to handle the following use cases:

- Log in known users (from a database)
- An error message should be displayed for unknown users and wrong passwords
- Display a simple menu for logged in users. Depending on their role, the users can:
- See results for all courses (student)
- Search for a course and display result (student)
- Search for a course and display students [with results] (teacher)
- Search for student and display courses [with results] (teacher)
- Enter course result for a student and save it to DB (teacher [of this course])
- [Enroll a student in a course and save it to DB (teacher [of this course])]
- Log-out

1.3 Other information

Ursprünglich wurde die Verwendung des Glassfishservers angedacht, jedoch im Laufe der Infrastrukturschaffung hat sich herausgestellt, dass dieser nicht mehr gewartet wird. Dementsprechend wurde die von der Community gewartete Alternative Payara Server herangezogen.

Persistenz wird mittels einer MySQL Datenbank gewährleistet. Diese Technologie wurde aufgrund von bereits bestehenden Kentnissen gewählt.

Hibernate wird als Persistence Provider verwendet. Dieses Framework wurde gewählt, da es in dem Interesse der Projektmitglieder war, diese Technologie zu erlernen.

Java Servlets wurden verwendet. Diese Architekturentscheidung wurde getroffen, damit es den Projektmitgliedern ermöglicht wird Erfahrungen mit dieser Technologie zu sammeln.

2 Stakeholders and concerns

2.1 Stakeholders

Lehrenden sind User, welche die Anforderungen, beschrieben in 1.2 fordern.
Studenten sind User, welche die Anforderungen, beschrieben in 1.2 fordern.

2.2 Concerns

1. Übersicht über eigene Kurse
2. Übersicht Zuordnung Studenten zu Kursen
3. Eintragen von Noten
4. Userlogin/logout
5. Eintragen von Studenten in Kurse
6. Suchen von Studenten
7. Suchen von Kursen

2.3 Concern–Stakeholder Traceability

	Studenten	Lehrer
Übersicht über eigene Kurse	x	x
Übersicht Zuordnung Studenten zu Kursen	-	x
Eintragen von Noten	-	x
Userlogin/ logout	x	x
Eintragen von Studenten in Kurse	-	x
Suchen von Studenten	-	x
Suchen von Kursen	x	x

3 Viewpoints

3.1 Student Viewpoint

Für einen Student ist das System eine Möglichkeit für die Einsichtnahme in eigene Kurse sowie in bereits erhaltene Noten.

3.1.1 Concerns

Übersicht über eigene Kurse: Dies stellt die Primäre Aufgabe des Systems, aus dem Blickwinkel eines Studenten dar. Hiermit soll es ihm ermöglicht werden, eigene Noten jener Kurse zu sehen, in denen er eingeschrieben ist.

Userlogin/logout: Ein notwendiger Sicherheitsmechanismus, der sicherstellt, dass eigene Noten nur durch den Studenten selbst und verantwortliches Lehrpersonal eingesehen werden kann.

Suchen von Kursen: Eine Möglichkeit für den Studenten aus vielen Kursen einen einzelnen auszuwählen, um leichter in dessen Note einsicht nehmen zu können.

3.1.2 Typical stakeholders

Dieser Blickwinkel wird von Anwendern des Systems eingenommen, die die Rolle Student einnehmen.

3.1.3 Anti-concerns

Einem Studenten soll es nicht möglich sein, selbst Noten einzutragen oder die Noten eines anderen Studenten sehen zu können.

3.2 Lehrer Viewpoint

3.2.1 Concerns

Übersicht über eigene Kurse: Hiermit soll es einem Lehrenden ermöglicht werden, eine Liste von jenen Kursen zu erhalten, in denen er unterrichtet.

Userlogin/logout: Ein notwendiger Sicherheitsmechanismus, der sicherstellt, dass nur berechtigte Anwender, die Zuordnung von Studenten zu Kursen oder die Noten von Studenten ändern bzw. eintragen können.

Suchen von Kursen: Hiermit soll es einem Lehrenden ermöglicht werden im gesamten Kurspool nach einem bestimmten Kurs zu suchen und die in diesem Kurs eingeschriebenen Studenten einzusehen.

Suchen von Studenten: Dies soll es einem Studenten ermöglichen, einzelne Studenten zu suchen und deren Kurse zu sehen.

Übersicht Zuordnung Studenten zu Kursen: Mittels diesem Punkt soll es einem Lehrenden ermöglicht werden, eine Liste aller Studenten und deren Zuordnungen zu den einzelnen Kursen zu erhalten.

Eintragen von Noten: Dies soll einem Lehrer ermöglichen, die Noten von Studenten seiner Kurse zu setzen.

Eintragen von Studenten in Kurse: Aufbauen auf der Funktionalität: Übersicht Zuordnung Studenten zu Kursen, soll es einem Lehrenden möglich sein, Studenten in Kurse einzuschreiben.

3.2.2 Typical stakeholders

Dieser Blickwinkel wird von Anwendern des Systems eingenommen, die die Rolle Lehrer einnehmen.

3.4 Model kinds

Entity-Relation Diagram: Wird zur Darstellung des Datenbank-Modells verwendet.

Schichten-Diagram: Wird zur Darsellung des Gesamtsystems verwendet.

3.4.1 Entity-Relation Diagram

3.4.1.1 Entity-Relation Diagram conventions

Dieses Modell wirde mittels des JPA-Modeller Plug-Ins der Netbeans IDE erstellt und stellt die Tabellenstruktur der im System verwendeten Datenbank dar. Resultierende Diagramme werden mittels der Crow‘s Foot Notation dargestellt.

3.4.2 Schichten-Diagram

In dieser eigens erstellten Notation werden Blöcke mittels Farben einem Systembereich zugeordnet. Blau werden jene Komponenten dargestellt, die für einen Anwender des Systems sichtbar sind. Gelb sind jene Elemente, die die Geschäftslogik des Systems abbilden. Grün sind die Elemente, die zur Sicherung von Userdaten verwendet werden.

Übereinander stehende Elemente stellen eine Abhängigkeit dar (Z.B. ist es notwendig Hibernate zu verwenden, um JPA verwenden zu können). Nebeneinander stehende Elemente stellen eine Interaktionsbeziehung dar (z.B. Servlets kommunizieren mit EJB und vice versa).

3.4.3 Modul - Diagram

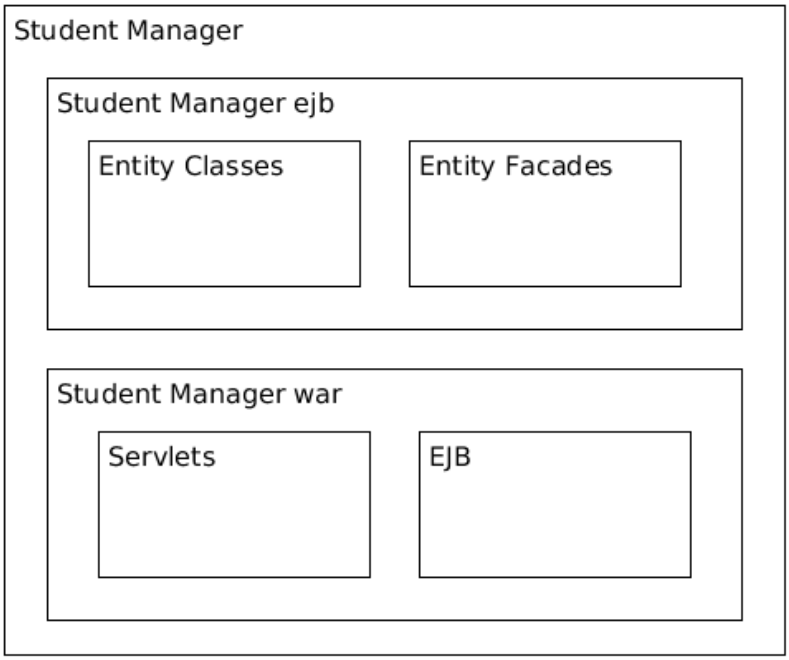
In solchen Diagrammen werden die logischen Module eines Projektes dargestellt. Dafür werden verschachtelte Rechtecke verwendet. Die höchste Verschachtelungsebene ist dabei das äußerste Rechteck.

4 Views

4.1 View: Module View

In dieser Sicht werden die Module des Student Managers dargestellt. Diese werden mittels des Modul-Diagrams dargestellt.

4.1.2 Student Manager Module View

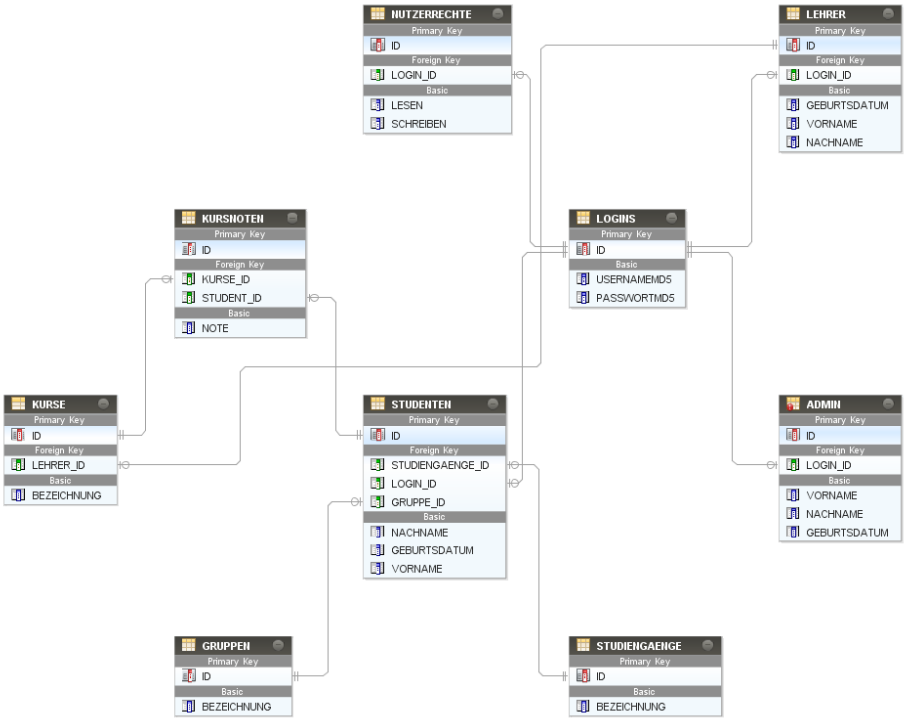


4.1.3 Known Issues with View

Die Unterscheidung der Module Entity Classes und Entity Facades wird in der Netbeans-Projektstruktur nicht mittels eigenen Java-Packages abgebildet. Die Classen befinden sich in einem gemeinsamen Packages.

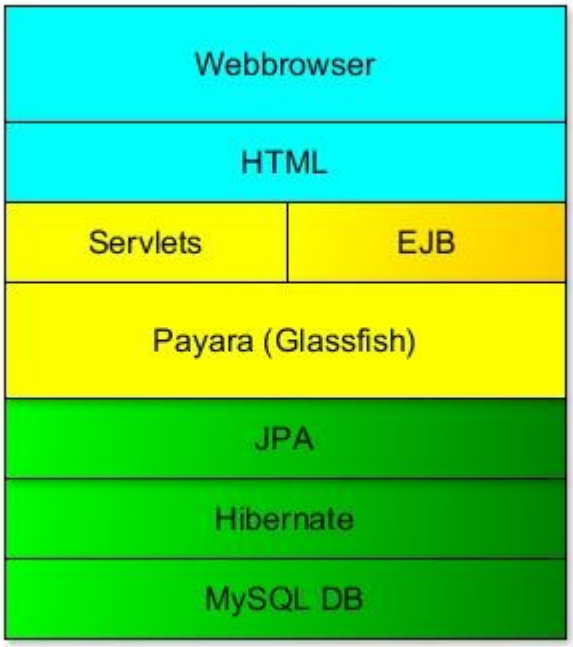
4.2 View: Database

In dieser Sicht wird das Datenbankmodell des Student Managers dargestellt. Dafür wird das Entity-Relation Diagram verwendet.



4.3 View: Layer View

In dieser Sicht wird eine Gesamtsicht auf das System dargestellt. Hierfür wird das Schichten-Diagramm verwendet.



4.3 View: Use Cases

Hier werden Use Cases des Systems nach UML dargestellt.