

Erstellung einer Online-Lernplattform mittels Google Web Toolkit

MAGISTERARBEIT

zur Erlangung des akademischen Grades

Magister der Sozial- und Wirtschaftswissenschaften

im Rahmen des Studiums

Informatikmanagement

eingereicht von

Gregor Prokesch

Matrikelnummer 0325505

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Dr. Andreas Ulovec

Wien, 10.08.2011

(Unterschrift Verfasser)

(Unterschrift Betreuer)

Erklärung zur Verfassung der Arbeit

Gregor Prokesch

Fluchtgasse 9/10, A-1090 Wien

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.“

Wien, 10.08.2011

Kurzfassung

Im Rahmen dieser Masterarbeit wurde eine Online-Lernplattform erstellt. Als Basis-Programmiersprache kam hier Java zum Einsatz, welche allerdings innerhalb des Frameworks „Google Web Toolkit“ genutzt wurde.

Dabei wurden die Vorteile einer effizienten Programmierschnittstelle, wie sie das GWT bietet, genutzt. Beispielsweise wurden AJAX-Funktionen angewandt, wie sie für moderne Web-Anwendungen heutzutage typisch sind. Zudem wurden die Vorteile der Java-Programmierung genutzt. Einerseits erfolgte das Design der Applikation auf objektorientierter Basis, um einen modularen Aufbau und somit etwaige spätere Erweiterungen zu ermöglichen. Andererseits wurden auch Java-Fähigkeiten wie zum Beispiel RPCs und jdbc-Datenbankanbindungen eingesetzt.

Die Lernplattform orientiert sich an der bekannten Moodle-Applikation und implementiert einige deren wichtigsten Features. Verschiedene Benutzergruppen werden unterschiedliche Möglichkeiten zur Verfügung gestellt, auf der Lernplattform zu agieren.

Abstract

With this work an online learning platform was developed. The basically programming language which has been used was Java, but particular it was used within the framework "Google Web Toolkit".

In this project the advantages of a modern, efficient programming interface like offered with the GWT was used and thus shown. E.g. AJAX functions were used which are typically for modern web applications. In addition the advantages of programming with Java were utilized. On the one hand the design of the application was done on an object-oriented way for building a modular structure and therefore making it possible to expand the application at a later time. On the other side Java functionalities like RPCs or odbc database connections were used.

The learning platform is oriented on the well known Moodle application and implements some of its most important features. Different user groups have different possibilities for actuating on the learning platform.

Inhalt

1	Aufgabenstellung - Features der Lernplattform.....	6
1.1	Benutzergruppen.....	6
1.2	Grundsätzliche Features	6
2	Methodik	7
2.1	Recherche.....	7
2.2	Modellierungswerkzeug.....	7
2.3	Programmierungsumgebung.....	7
2.4	Datenhaltung.....	7
3	Design der Applikation.....	8
3.1	Namensgebung.....	8
3.2	Konzept.....	8
3.3	Vorgehensweise.....	8
3.4	Use-Cases	9
3.4.1	Use-Case 1: Kurs erstellen.....	9
3.4.2	Use Case 2: Lesson erstellen.....	9
3.4.3	Use-Case 3: Lesson absolvieren.....	11
3.5	Realisierung der Use-Cases.....	12
3.5.1	Umsetzung von Use-Case 1: Kurs erstellen	12
3.5.2	Umsetzung von Use-Case 2: Lesson erstellen	14
3.5.3	Umsetzung von Use-Case 3: Lesson absolvieren	15
3.6	Sitemap.....	17
3.6.1	Sitemap für Studenten.....	17
3.6.2	Sitemap für Kursleiter	18
3.6.3	Sitemap für Administratoren	20
3.7	Design der Datenbank.....	20
3.7.1	Entity-Relationship Diagramm: Kurs.....	20
3.7.2	Entity-Relationship Diagramm: Testfragen	21
3.7.3	Entity-Relationship Diagramm: Forum.....	21
3.7.4	Entity-Relationship Diagramm: Gesamtdesign der Datenbank	22
3.8	Architektur	23

3.9	Klassendesign.....	24
3.9.1	Service- und Logik-Schicht.....	24
3.9.2	Datenschicht	28
3.9.3	Präsentationsschicht	29
4	Implementierung.....	32
4.1	Projektstruktur	32
4.2	Kommunikation	34
4.3	Erstellung des User Interface	36
4.3.1	Verwendete Panels und Widgets.....	38
5	Handbuch.....	41
5.1	Kursleiter – Erstellen eines Kurses.....	41
5.2	Kurseinheit – Präsentation	43
5.3	Kurseinheit Test	46
5.4	Testfrage Multiple Choice.....	46
5.5	Testfrage Zuordnung.....	48
5.6	Testfrage Abgabe.....	49
5.7	Kurseinheit Lesson.....	49
5.8	Student – Kursteilnahme	51
5.9	Kursleiter – Kursbetreuung.....	56
6	Zusammenfassung.....	58
7	Abbildungsverzeichnis.....	59
8	Tabellenverzeichnis.....	60
9	Literaturverzeichnis	61

1 Aufgabenstellung - Features der Lernplattform

Folgend soll dargelegt werden, welche Möglichkeiten mit der Lernplattform realisiert werden sollen.

1.1 Benutzergruppen

Es werden drei Benutzergruppen realisiert: Studenten, Kursleiter und Administratoren. Sie werden unterschiedliche Rechte und Fähigkeiten haben, welche sie auf die nächst höhere Gruppe vererben – ganz im Sinne der Objektorientierung.

1.2 Grundsätzliche Features

Darstellung der Lernmaterialien

Als zentrale Fähigkeit soll auf der Plattform das Erstellen und Anzeigen von Lernmaterialien dienen. Dazu werden den Kursleitern und Studenten zwei Möglichkeiten gegeben. Einerseits wird ein Rich Text – Editor zur Verfügung gestellt, andererseits soll auch das Anzeigen von reinem, selbsterstelltem HTML möglich sein. Weiters soll der Up- und Download von Dateien realisiert werden.

Tests/Übungen erstellen

Zur Leistungsüberprüfung sollen – neben der Abgabe von Hausübungen in digitalem Format – Test- bzw. Übungsfragen dienen. Dabei sollen verschiedene Möglichkeiten zur Auswahl stehen, wie zum Beispiel Multiple Choice, Richtig/Falsch, Zuordnungen, Langfragen und Ergänzungen. Die Test- und Übungsfragen, sowie die Abgabe von Hausübungen können vom Kursleiter zeitabhängig erstellt werden, sodass diese in einem fest definierten Zeitfenster stattfinden können.

Lessons

Lessons sind eine Kombination aus der reinen Darstellung von Lernmaterialien und der Leistungsüberprüfung. Dabei soll es beispielsweise möglich sein, dass Unterrichtsmaterial seitenweise angezeigt und das Gelernte anschließend mit einer Übungsfrage überprüft wird. Auch hier kann das Abarbeiten einer Lesson innerhalb eines vorher bestimmten Zeitraums erfolgen. Durch die Möglichkeit einer Punktevergabe pro gelesene Seite oder beantworteter Übungsfrage kann das Voranschreiten der Lernenden überprüft werden.

Forum

Um Neuigkeiten oder Informationen zu verkünden soll eine Forumsfunktion zur Verfügung gestellt werden. Auch Studenten sollen die Möglichkeit haben, Fragen zu stellen und zu beantworten, wenn der Kursleiter dies zulässt. Dabei wird ein Forumsthread mit einem Beitrag eröffnet, der anschließend beantwortet werden kann. Beantwortet werden können ebenso auch andere Antworten. Die Darstellung des Threads erfolgt der Übersichtlichkeit halber mittels Einrücken.

Emailfunktion

Damit wichtige Informationen sicher die gewünschten Adressaten erreichen, soll die Möglichkeit geschaffen werden, Emails zu verschicken. Dabei soll der Kursleiter die Auswahl haben, ob er die Mail an alle Teilnehmer eines Kurses oder nur an einzelne Studenten verschickt. Weiters sollen Administratoren den Kursleitern Emails schicken können und vice versa.

Verwaltung

Der Verwaltung fällt bei der Online-Lernplattform eine zentrale Rolle zu. Grundsätzlich soll es eine Benutzerverwaltung mit den drei erwähnten Benutzergruppen geben. Studenten sollen von den Kursleitern für Kurse freigeschalten oder – falls nötig – gesperrt werden können. Administratoren verwalten zusätzlich die Rechte der Kursleiter. Die Kursleiter können Kurse erstellen und löschen. Weiters legen sie fest, in welcher Weise Studenten das Forum nutzen dürfen. Sie haben Zugriff auf die Punkteergebnisse von Test- und Übungsfragen sowie von den Lessons.

2 Methodik

2.1 Recherche

Die Recherche für diese Magisterarbeit erfolgt in erster Linie über die Online-Dokumentation des Google Web Toolkits (GWT, siehe [GWT]). Diese ist gut und übersichtlich strukturiert und bietet sowohl intuitive Tutorials als auch Informationen über die Funktionsweise der GWT-Technologie. Weiterführende Fragen werden über entsprechende News-Groups diskutiert und können dort recherchiert werden (siehe [GwtNewsGr]).

Die Funktionsreferenz ist einerseits über die GWT-API ersichtlich (siehe [GwtAPI]), andererseits dienen die Java Standard API (siehe [JavaAPI]) und die API für Java Enterprise Engineering (siehe [JavaEEAPI]) als Referenzen.

2.2 Modellierungswerkzeug

Für die Modellierung wird „Visual Paradigm for UML“ verwendet (siehe [VPUML]). Dieses Tool bietet volle Unterstützung für den Standard UML, die Möglichkeit zur Erstellung von Entity-Relationship-Diagrammen und unterstützt Java-Roundtrip-Engineering.

2.3 Programmierumgebung

Als IDE (Integrated Development Environment) kommt bei der Programmierung die Eclipse-Plattform zum Einsatz (siehe [Eclipse]). Einerseits ist diese für mich die gewohnte Programmierumgebung. Zudem erfolgt die GWT-Dokumentation über den Einsatz von Eclipse. Darüber hinaus existiert mit dem „Google Plugin for Eclipse“ ein Werkzeug zur einfachen Erstellung und Kompilierung von GWT-Projekten (siehe [GPE]).

2.4 Datenhaltung

Zur Datenhaltung dient eine MySQL-Datenbank (siehe [MySQL]). Durch ihren vielfachen Einsatz im professionellen Umfeld hat sie sich in der Vergangenheit bereits als praxisrelevant und durch ihre gute Skalierbarkeit als äußerst performant erwiesen.

3 Design der Applikation

3.1 Namensgebung

Als Name für die Lernplattform wurde „pro_on-learn“ gewählt. „pro“ steht für die Kurzform meines Nachnamens „Prokesch“. „on-learn“ ist eine Mischung aus „online“ und „learn“ und soll die Funktionalität der Plattform widerspiegeln.

3.2 Konzept

Dem Aufbau der Applikation liegt eine 4-Schichten-Architektur, wie bei [Masak10] beschrieben, zugrunde. Die Idee dahinter ist jene, den Datenverkehr zum Client – also dem Internet-Browser des Users – möglichst wenig gering zu halten. Zusätzlich soll die Datenhaltung von der Logik getrennt sein. Das GWT-eigene Verfahren von Remote Procedure Calls ermöglicht diesen Ansatz in konsequenter Weise. Am Server befinden sich die Logik- und Datenschicht. Dementsprechend befinden sich die beiden anderen Schichten – die Service- und Präsentationsschicht – am Client.

Das GWT setzt den Zugriff auf die Logik am Server mittels asynchronen RPC um. Dies erfordert für jedes Service ein Interface welches das RemoteService-Interface erweitert, sowie ein weiteres Interfaces für den asynchronen Methodenaufruf, welches obligatorisch ist. Die Server-seitigen Implementationen dieser Service-Interfaces müssen wiederum das RemoteServiceServlet erweitern.

Dadurch ergibt sich folgender Aufbau:

Logikschicht

Diese befindet sich am Server und beinhaltet jene Klassen, welche Programmlogik enthalten.

Datenschicht

In ihr findet der Zugriff auf die MySQL-Datenbank statt. Ausgehend davon, welcher Benutzer sich einloggt, wird ein jeweils passendes Interface zur Verfügung gestellt, auf das allein die Klassen der Logikschicht Zugriff haben.

Serviceschicht

Die Serviceschicht befindet sich am Client und enthält die Remote-Interfaces welche von den Klassen in der Logikschicht implementiert werden sowie die Interfaces für den asynchronen Zugriff auf den Server.

Präsentationsschicht

Hier befinden sich jene Klassen, die für die Erstellung des User-Interfaces zuständig sind.

3.3 Vorgehensweise

Dem Design liegt die Verwendung von Use Cases zugrunde (siehe [Arlow02], Seite 60). Drei typische Use Cases wurden entworfen: das Erstellen eines Kurses, weiters das Erstellen einer Lesson jeweils durch einen Kursleiter, sowie das Absolvieren einer Lesson durch einen Studenten. Daraufhin wurden Sequenzdiagramme (siehe [Arlow02], Seite 220) gezeichnet, die den zeitlichen Ablauf demonstrieren sollen. Auf Basis dieser Sequenzdiagramme wurden Klassendiagramme (siehe [Oester09], Seite 273) gefertigt und anschließend erweitert. Das Entity-Relationship-Diagramm (siehe [Kemper04]) für

den Datenbankentwurf wurde wiederum auf dieser Grundlage erstellt. Anschließend würden die verschiedenen Diagramme auf einander abgestimmt.

3.4 Use-Cases

Folgend werden die drei erwähnten Use-Cases anhand von Diagrammen sowie natürlich-sprachlich beschrieben.

3.4.1 Use-Case 1: Kurs erstellen

Dieser Use-Case beschreibt den Vorgang wie ein Kursleiter einen Kurs erstellt. Dem Kurs werden Titel und Beschreibung gegeben. Teilnehmende Studenten werden hinzugefügt und Rechte zugewiesen. Weiters wird eine Kurseinheit – in diesem Fall eine Lesson – erstellt:

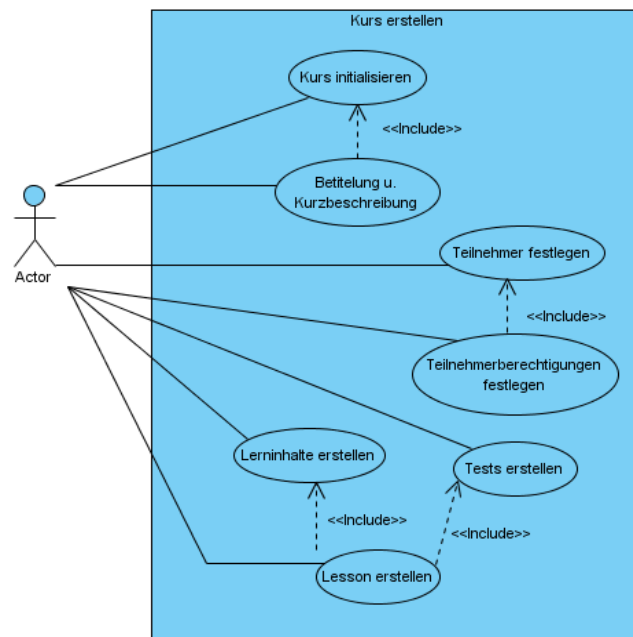


Abbildung 1 - Use-Case Kurs erstellen

Name	Kurs erstellen
Ziel	Erstellung eines Kurses durch Kursleiter
Vorbedingung	Erfolgreicher Login, User ist Kursleiter
Nachbedingung	Kurs erfolgreich erstellt
Beteiligte Akteure	Kursleiter
Beschreibung Ablauf	<ol style="list-style-type: none"> 1. Kurs erstellen 2. Titel und Beschreibung erstellen 3. Studenten festlegen und Rechte erteilen 4. Lesson erstellen

Tabelle 1 - Use Case Kurs erstellen

3.4.2 Use Case 2: Lesson erstellen

Bei diesem Use-Case wird in einen bestehenden Kurs eine Kurseinheit eingebaut. Als Art der Kurseinheit wurde die Lesson gewählt, da diese eine Kombination der beiden

anderen Arten – Präsentation und Test – darstellt. Es werden Lernmaterialien präsentiert, sowie Testfragen erstellt. Weiters wird festgelegt, in welchem Zeitrahmen die Lesson absolviert werden soll und wie viele Punkte die jeweiligen Testfragen wert sind.

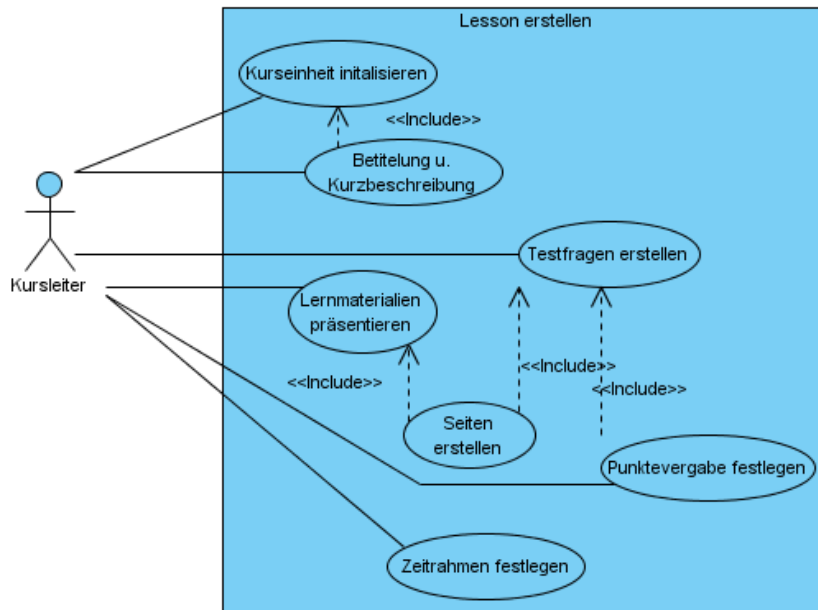


Abbildung 2 - Use-Case Lesson erstellen

Name	Lesson erstellen
Ziel	Erstellung einer Lesson durch Kursleiter
Vorbedingung	Erfolgreicher Login, User ist Kursleiter, Kurs ist erstellt
Nachbedingung	Lesson erfolgreich erstellt
Beteiligte Akteure	Kursleiter
Beschreibung Ablauf	<ol style="list-style-type: none"> 1. Lesson erstellen 2. Titel und Beschreibung zufügen 3. Seiten erstellen 4. Lerninhaltliche Präsentation den Seiten zufügen 5. Testfrage für die Seiten generieren 6. Weitere Seiten erstellen 7. Gegebenenfalls die Seitenreihenfolge ändern

Tabelle 2 - Use-Case Lesson erstellen

3.4.3 Use-Case 3: Lesson absolvieren

Der dritte Use-Case behandelt jenen Fall, dass ein Student eine Lesson absolviert. Dabei liest er sich die Präsentationen der Lerninhalte durch und beantwortet anschließend eine Testfrage. Die hierbei verwendeten Testfragen müssen dergestalt sein, dass die Punktevergabe automatisch erfolgt.

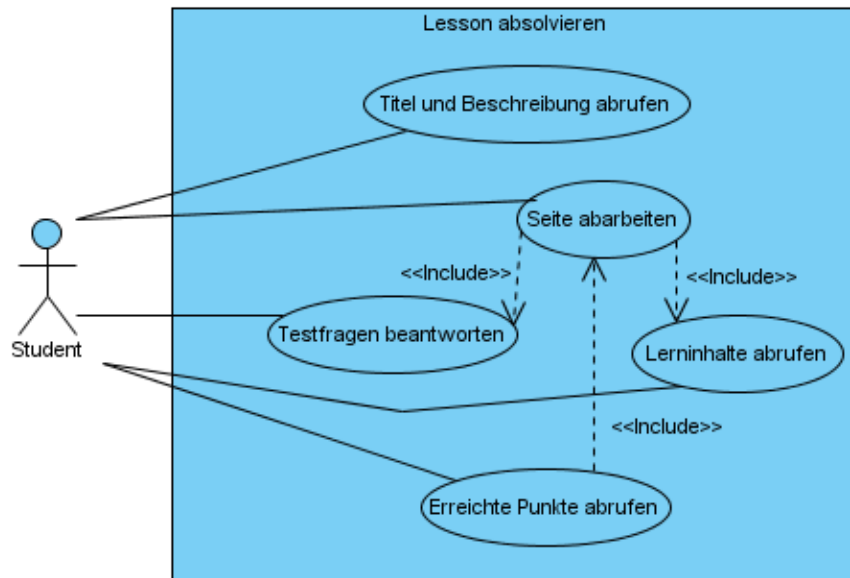


Abbildung 3 - Use-Case Lesson absolvieren

Name	Lesson absolvieren
Ziel	Absolvieren einer Lesson durch Studenten
Vorbedingung	Erfolgreicher Login, User ist Student mit entsprechenden Rechten
Nachbedingung	Lesson erfolgreich absolviert
Beteiligte Akteure	Student
Beschreibung Ablauf	<ol style="list-style-type: none"> 1. Sequenzieller Aufruf der einzelnen Seiten 2. Darstellung der Lerninhalte 3. Darstellung der Testfragen 4. Beantwortung der Testfragen 5. Automatische Punktevergabe

Tabelle 3 - Use-Case Lesson absolvieren

Die Realisierung dieser drei repräsentativen Use-Cases wird in den folgenden Unterkapiteln beschrieben.

3.5 Realisierung der Use-Cases

Zur Umsetzung der Use-Cases werden jeweils Sequenzdiagramme gezeichnet und dazu passende Klassendiagramme erstellt. Zur Erläuterung: die jeweils dunkler gezeichneten Lebenslinien stellen die Klassen für die Bereitstellung des User-Interfaces dar.

3.5.1 Umsetzung von Use-Case 1: Kurs erstellen

Sequenzdiagramm

- Dem User Kursleiter wird eine Lebenslinie zugeteilt.
- Um einen Kurs zu erstellen verwendet er die ihm eigene Methode „erstelleKurs()“.
- Er besucht die Site „Kurs“ und lässt sich seine Optionen anzeigen um danach den Kurs via Remote-Interface „KursEdit“ zu editieren.
- Er erstellt Titel und Beschreibung und fügt Teilnehmer hinzu, deren Rechte er anschließend festlegt.
- Danach erstellt er eine Kurseinheit und gelangt zum Editieren dieser auf eine neue Site „Kurseinheit“.
- Mittels dem Remote-Interface „KurseinheitEdit“ kann eine Kurseinheit bearbeitet werden, indem z.B. Seiten hinzugefügt werden, etc.
- Abschließend werden der Kurs, bzw. die Kurseinheiten freigegeben und die Studenten können per Email benachrichtigt werden.

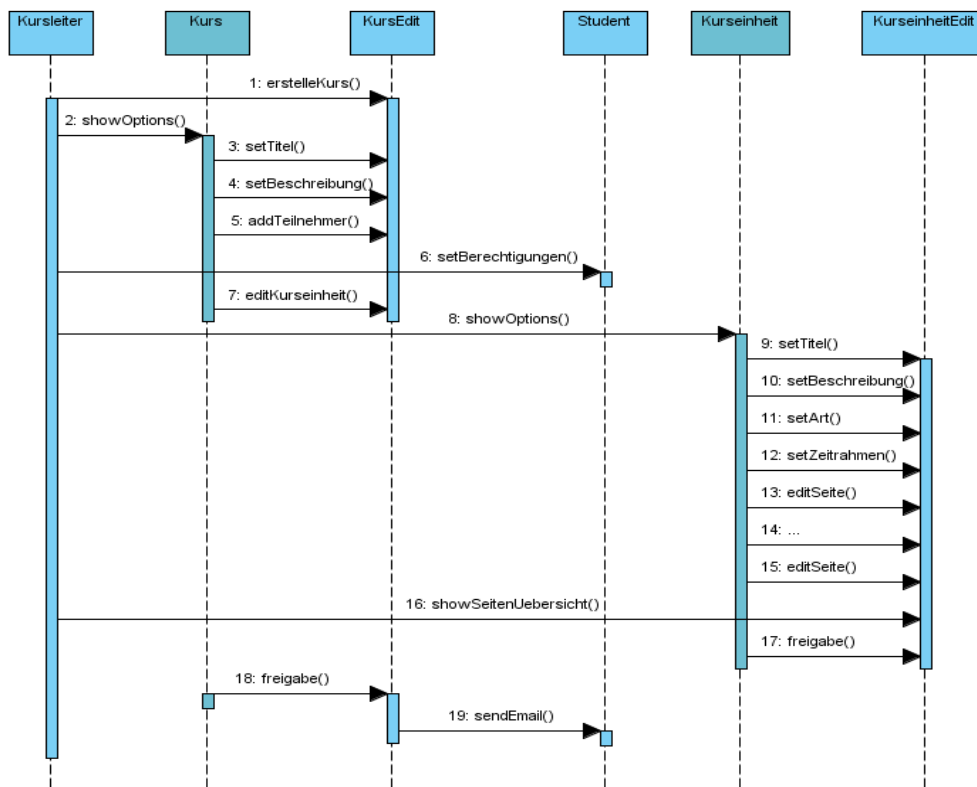


Abbildung 4 - Sequenzdiagramm Kurs erstellen

Klassendiagramm

Das nachfolgende Diagramm zeigt die Umsetzung des Sequenzdiagramms in Klassen. Sichtbar sind hier auch bereits architektonische Entscheidungen wie die Unterteilung in die vier Schichten sowie die Trennung von Client und Server.

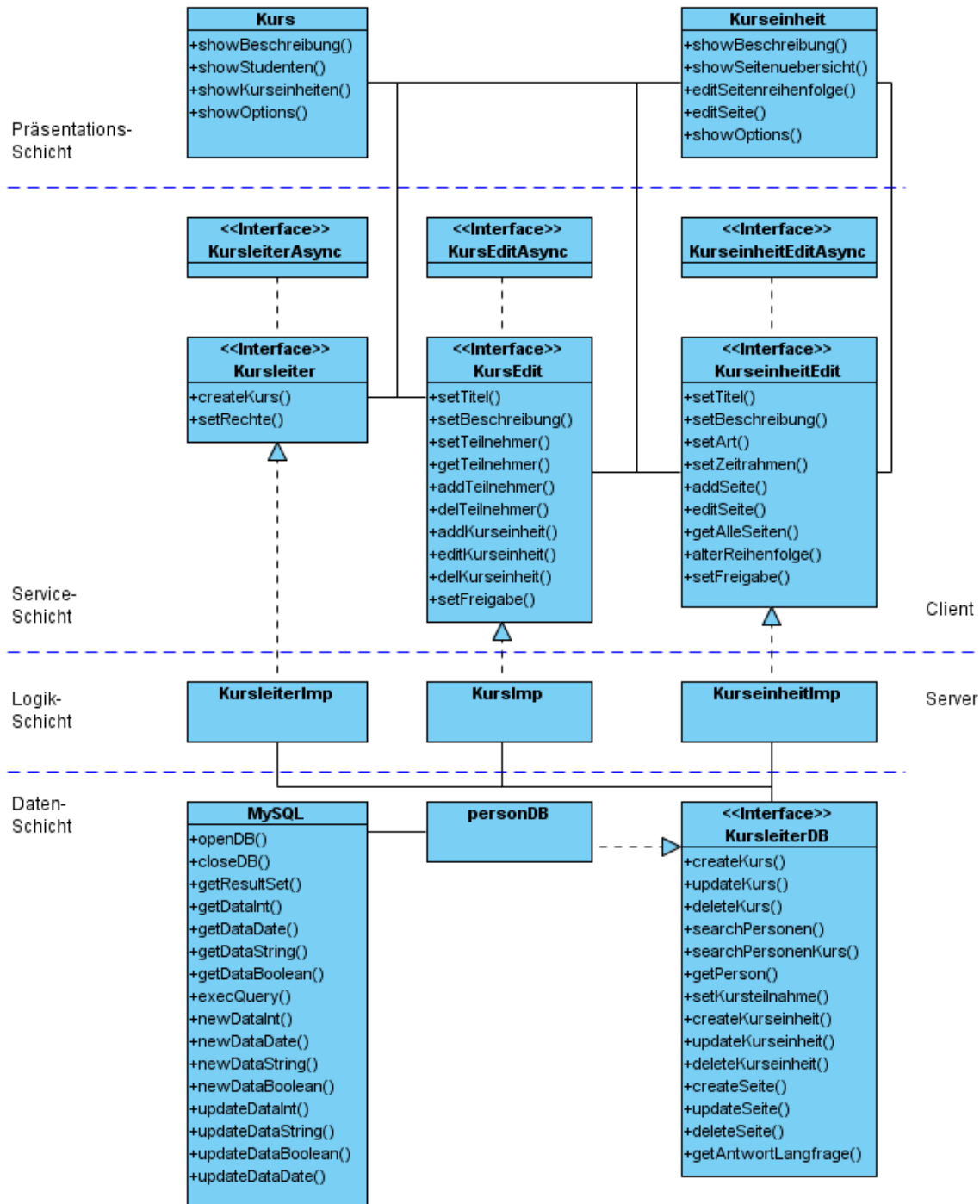


Abbildung 5 - Klassendiagramm Kurs erstellen

3.5.2 Umsetzung von Use-Case 2: Lesson erstellen

Sequenzdiagramm

- Dem User „Kursleiter“ wird wiederum eine Lebenslinie zugeteilt.
- Auf der Site „Kurs“ lässt er sich seine Optionen anzeigen um über die Klasse „KursEdit“ eine neue Kurseinheit zu erstellen.
- Er gelangt auf die Site „Kurseinheit“ und kann die Kurseinheit über die Klasse „KurseinheitEdit“ als Lesson definieren und Titel sowie Beschreibung und Zeitrahmen hinzufügen. Weiters kann er hier eine neue Seite für die Lesson erstellen und bearbeiten.
- Dies geschieht auf der Site „Seite“. Via Klasse „SeiteEdit“ kann er der Seite eine lerninhaltliche Präsentation sowie eine Testfrage hinzufügen.
- Zur Bearbeitung der Testfrage gelangt er auf die Site „Testfrage“. Als Beispiel wurde exemplarisch eine Wahr-Falsch-Testfrage gewählt. Deren Bearbeitung findet über die Klasse „WahrFalschEdit“ statt.
- Anschließend können der Lesson weitere Seiten hinzugefügt werden.
- Danach kann sich der Kursleiter auf der Site „Kurseinheit“ eine Gesamtübersicht seiner erstellten Seiten anzeigen lassen. Mittels Drag'n'Drop kann die Reihenfolge der Seiten verändert werden.
- Abschließend kann die Kurseinheit für die Studenten freigegeben werden.

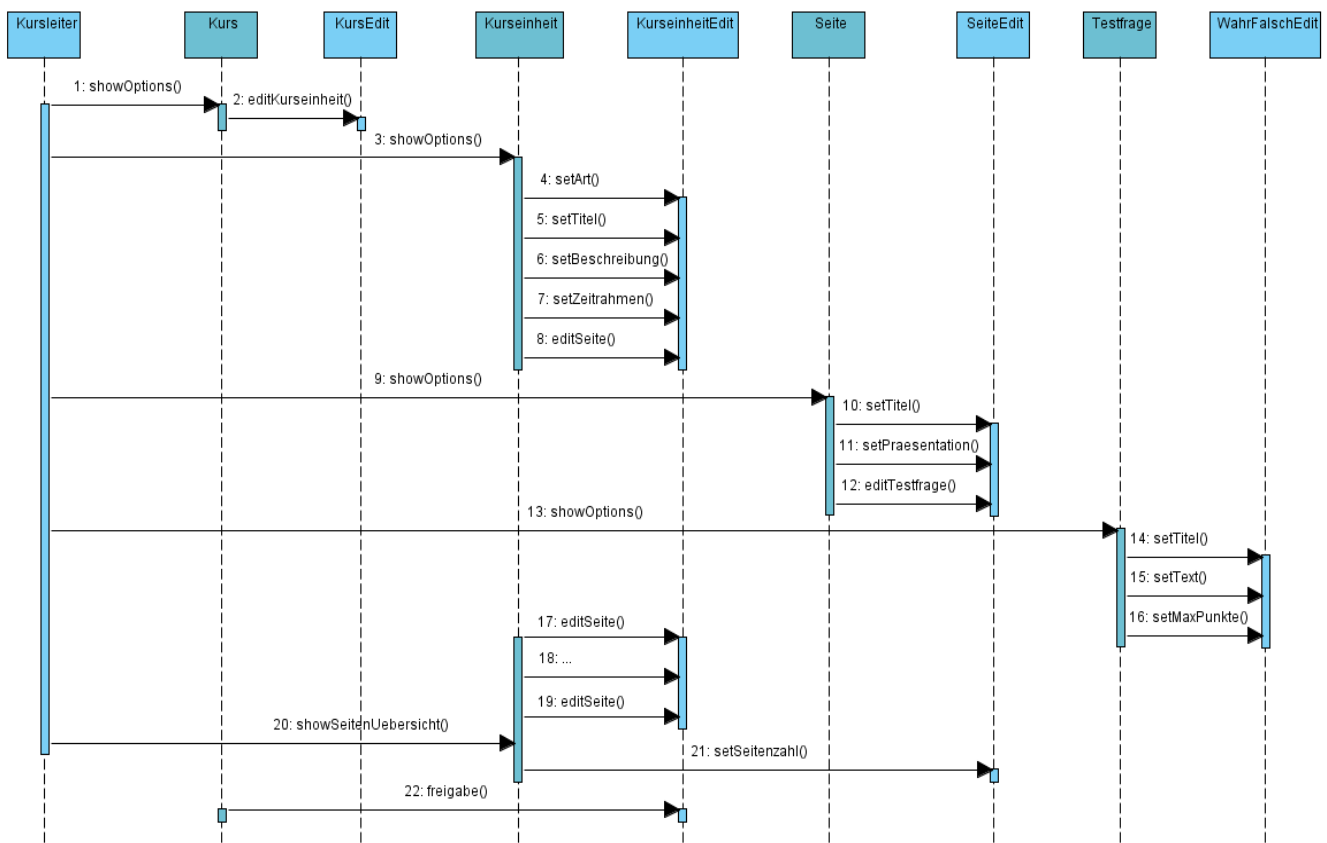


Abbildung 6 - Sequenzdiagramm Lesson erstellen

Klassendiagramm

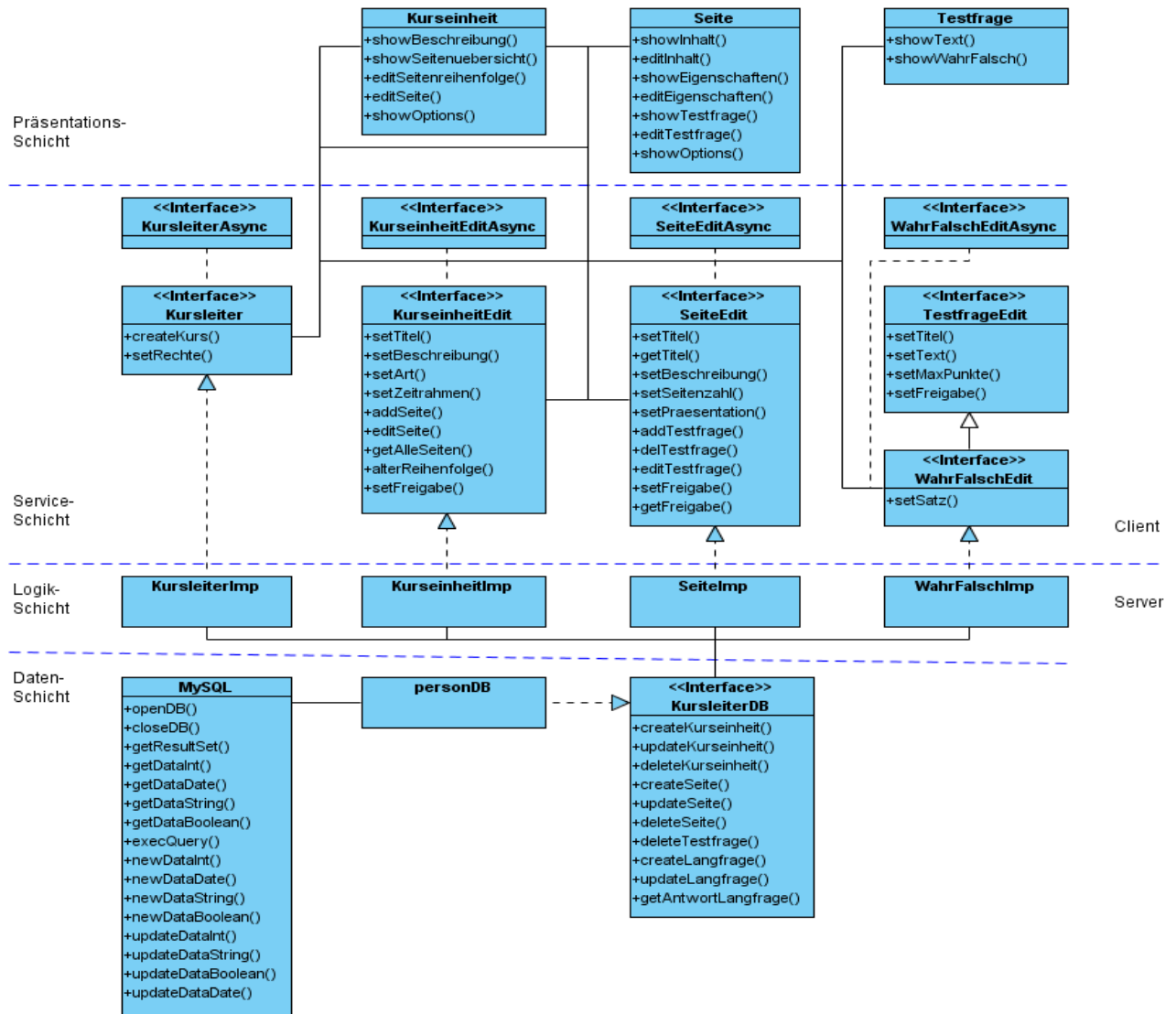


Abbildung 7 - Klassendiagramm Lesson erstellen

3.5.3 Umsetzung von Use-Case 3: Lesson absolvieren

Sequenzdiagramm

- Hier erhält der User „Student“ eine Lebenslinie
- Via Site „Kurs“ und der Klasse „KursView“ lässt er sich die Kurseinheit anzeigen.
- Auf der Site „Kurseinheit“ kann er deren Beschreibung abrufen und die, in diesem Fall Lesson, starten
- Er gelangt auf die Site „Seite“, liest dort die Präsentation und beantwortet anschließend eine dazu passende Testfrage.
- Dies geschieht über die eingebettete Site „Testfrage“ und eine der passenden Klassen. In diesem Fall wurde wieder exemplarisch eine Wahr-Falsch-Frage gewählt, weshalb die Klasse „WahrFalschView“ benutzt wird.
- Hat der Student die Frage beantwortet gelangt er zur nächsten Seite und somit wiederum zur Präsentation und einer Testfrage.

- Wurden alle Seiten der Lesson abgearbeitet, gelangt er zurück zur Site „KurseinheitView“ und kann dort nun seine erreichte Punktezahl abrufen.

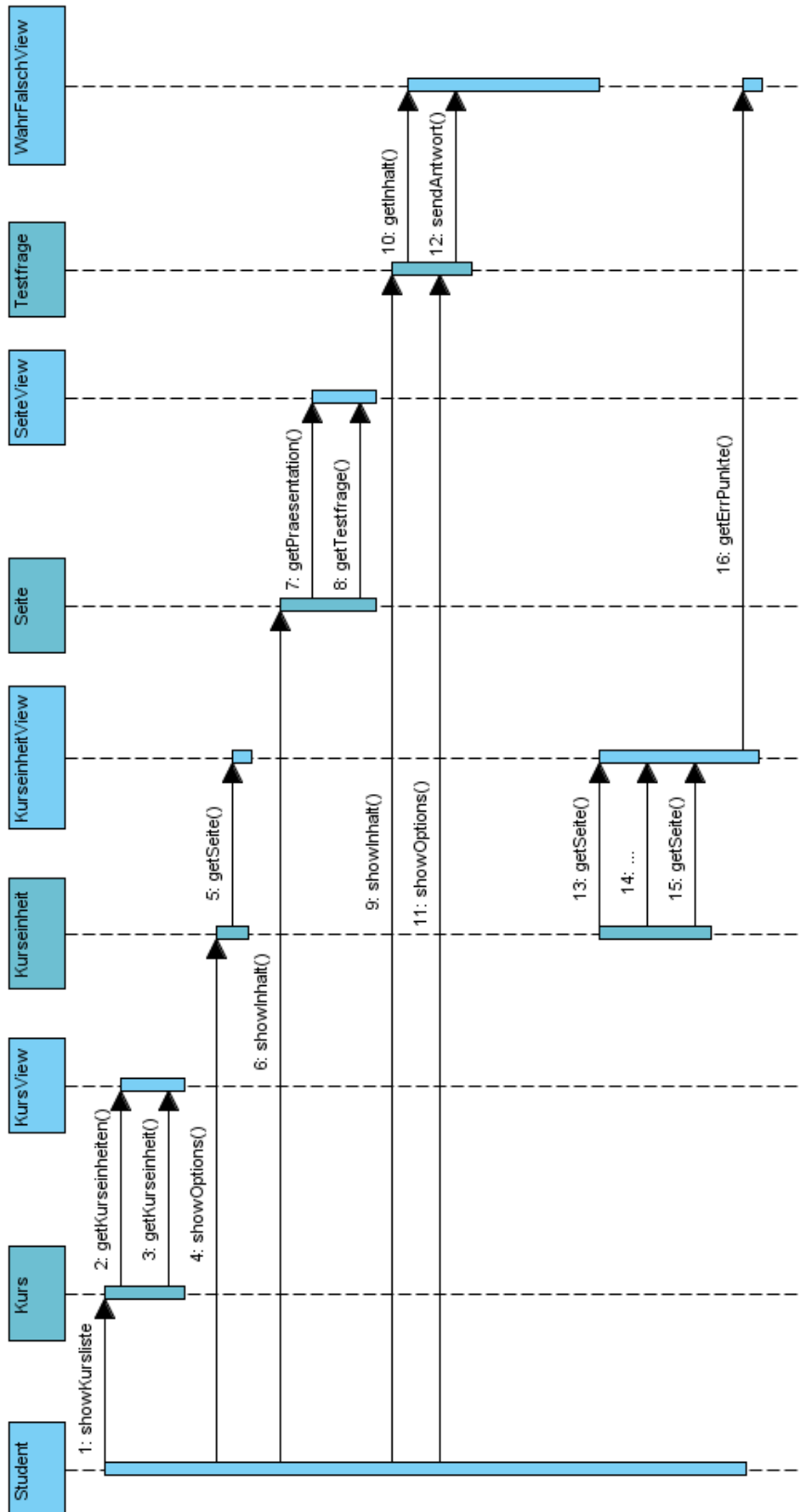


Abbildung 8 - Sequenzdiagramm Lesson absolvieren

Klassendiagramm

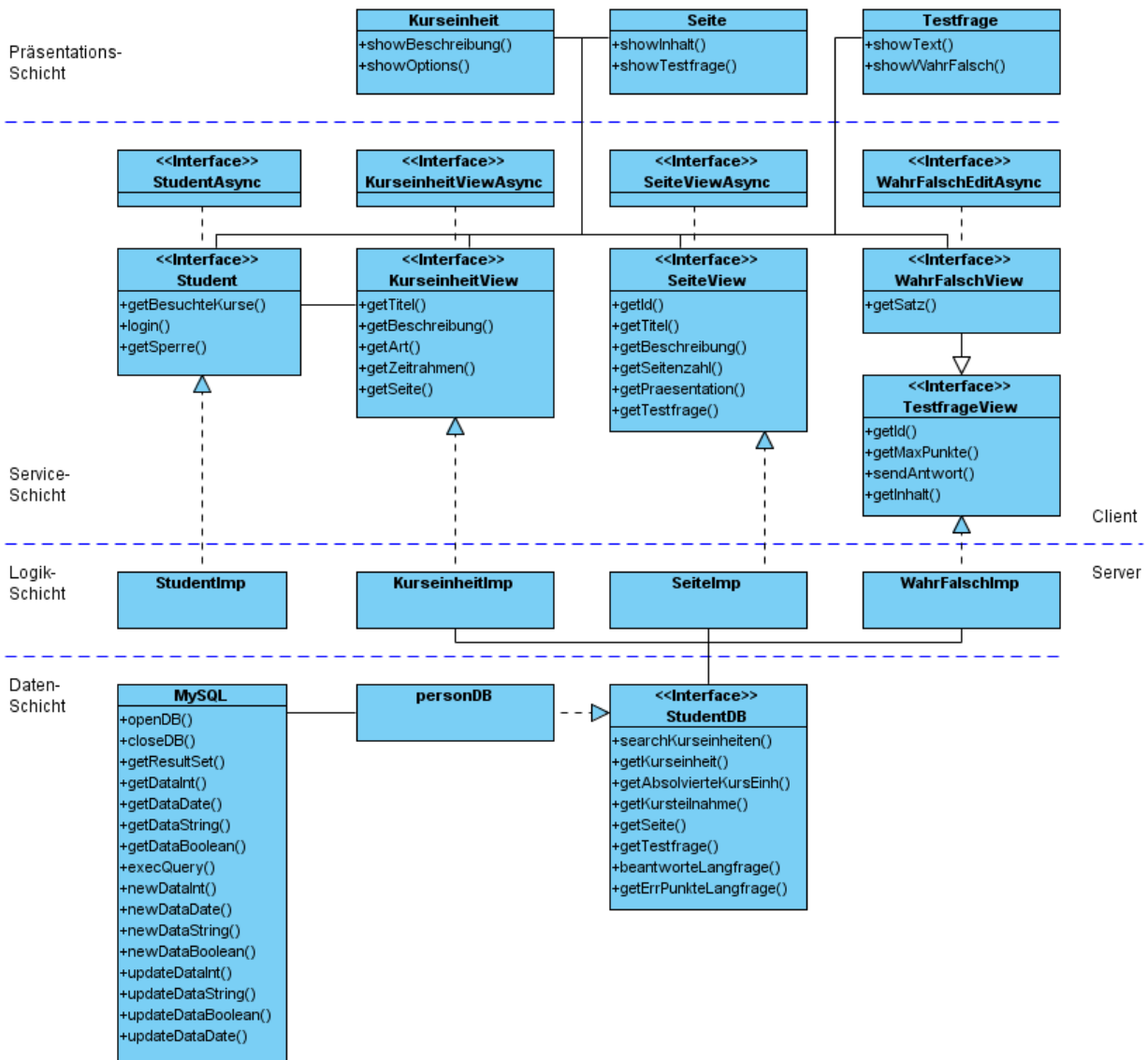


Abbildung 9 - Klassendiagramm Lesson absolvieren

3.6 Sitemap

Zur Überlegung der benötigten Funktionalität wurde aufbauend auf die vorhandenen Grundlagen eine Sitemap, wie bei [Judmaier05]) beschrieben, erstellt. Hier wurden einerseits die Use-Cases als Aufbau der Website umgesetzt. Außerdem wurde auf weitere geplante Funktionalitäten, welche bisher nicht abgedeckt wurden, Rücksicht genommen. Dazu zählen beispielsweise die Forumsfunktion oder die Verwaltungsaufgaben der Administratoren. Zur besseren Übersicht wurden den Benutzergruppen entsprechend drei Sitemaps angefertigt.

3.6.1 Sitemap für Studenten

Nach dem erfolgreichen Login gelangt der Student zu seiner WelcomePage auf der er unter anderem eine Liste mit seinen Kursen abrufen kann. Via Hyperlink gelangt er dann zur Startseite des jeweiligen Kurses und kann von dort aus seine Aktivitäten starten. Diese sind das Abrufen einer Kurseinheit oder des Kurs-Forums. Betrachtet er das

Erstellung einer Online-Lernplattform mittels GWT

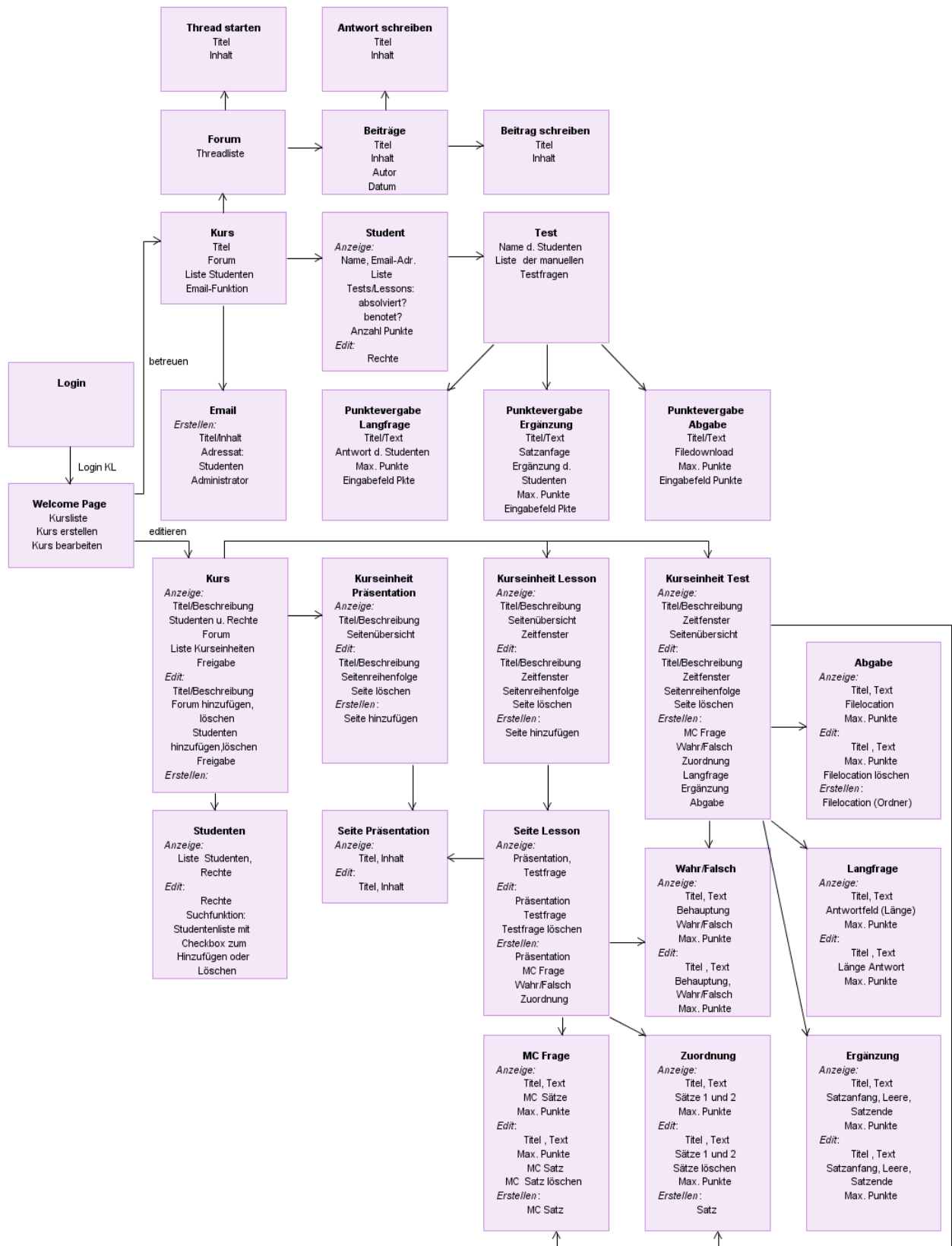


Abbildung 11 - Sitemap Kursleiter

3.6.3 Sitemap für Administratoren

Dem Administrator fällt die Verwaltungstätigkeit zu. Er kann Personen im System aufnehmen, ihre Daten editieren und sie bei Bedarf sperren. Auch für Kurse hat er die Möglichkeit, sie zu sperren.

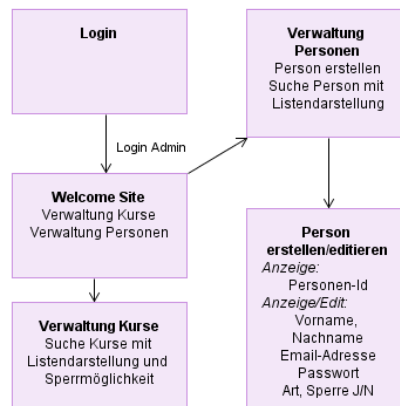


Abbildung 12 - Sitemap Administratoren

3.7 Design der Datenbank

Auf dieser Grundlage konnten alle benötigten Informationen gesammelt werden, um das Datenbankdesign durchführen zu können. Dieses erfolgte mittels Entity-Relationship-Diagramm. Zur besseren Übersicht zeigen die nächsten drei Abbildungen Teildiagramme.

3.7.1 Entity-Relationship Diagramm: Kurs

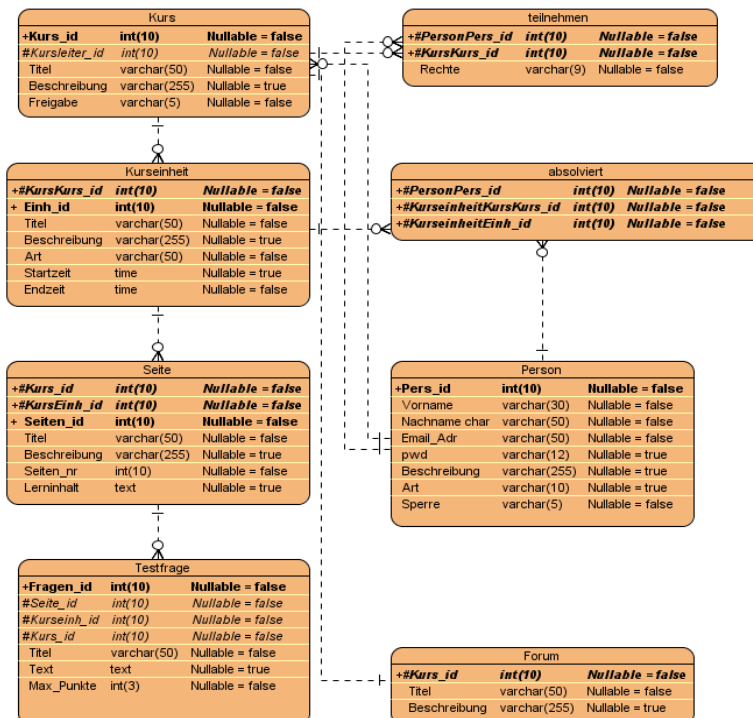


Abbildung 13 - ER Diagramm Kurs

3.7.2 Entity-Relationship Diagramm: Testfragen

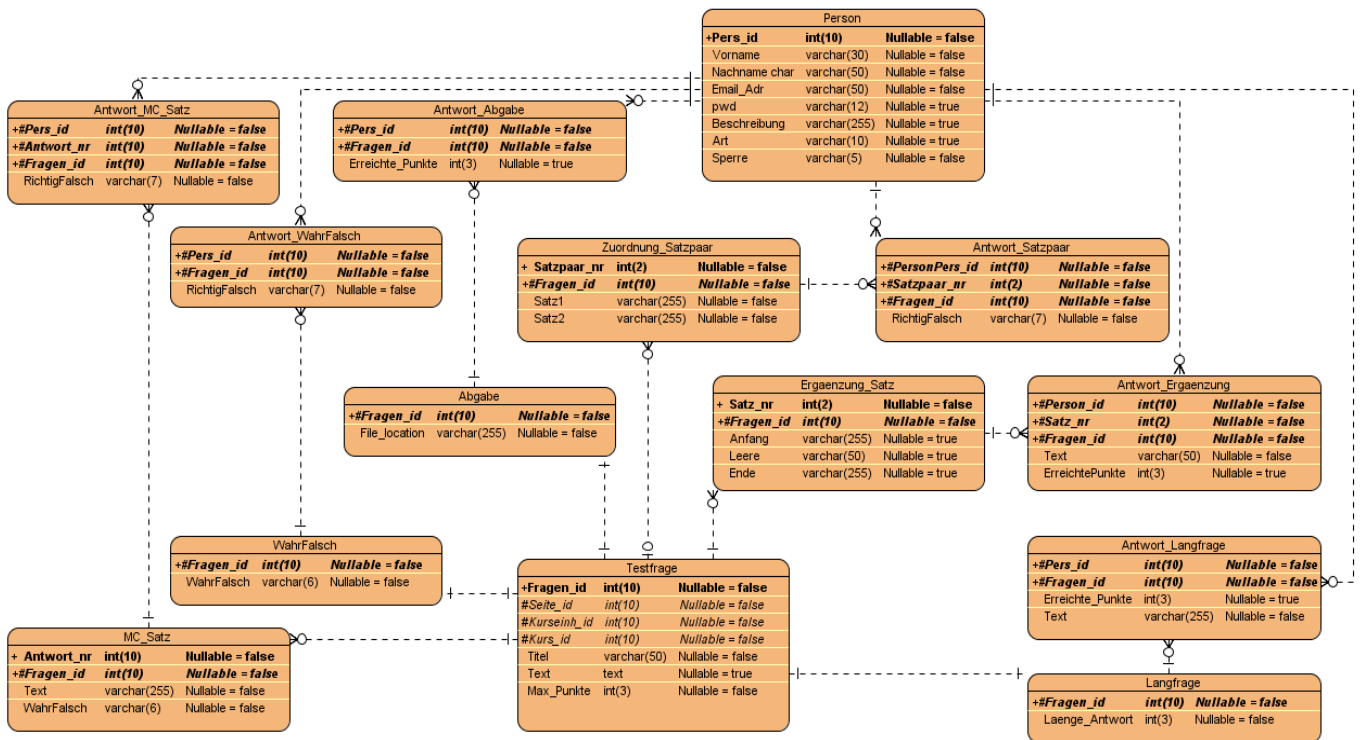


Abbildung 14 - ER Diagramm Testfragen

3.7.3 Entity-Relationship Diagramm: Forum

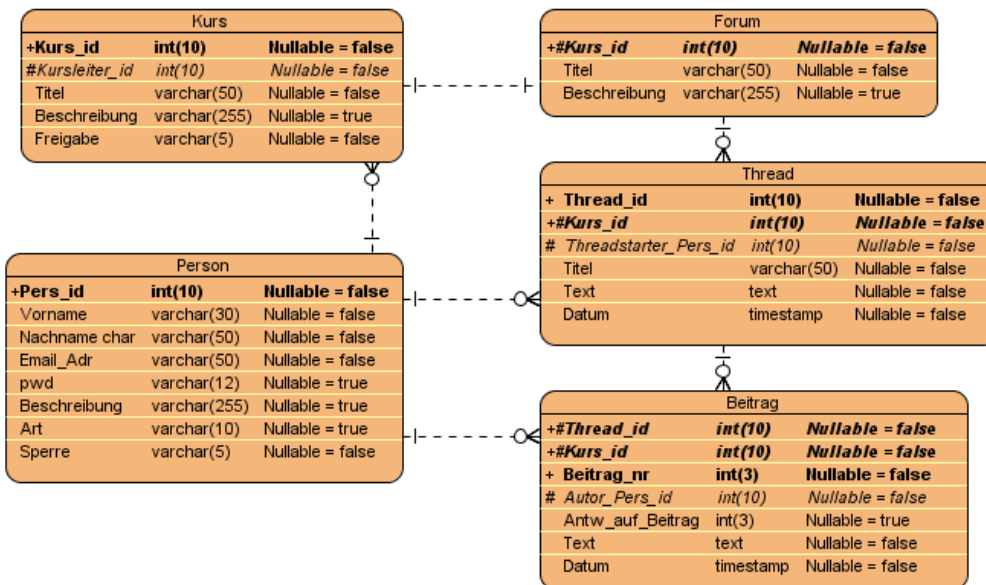


Abbildung 15 - ER Diagramm Forum

3.7.4 Entity-Relationship Diagramm: Gesamtdesign der Datenbank

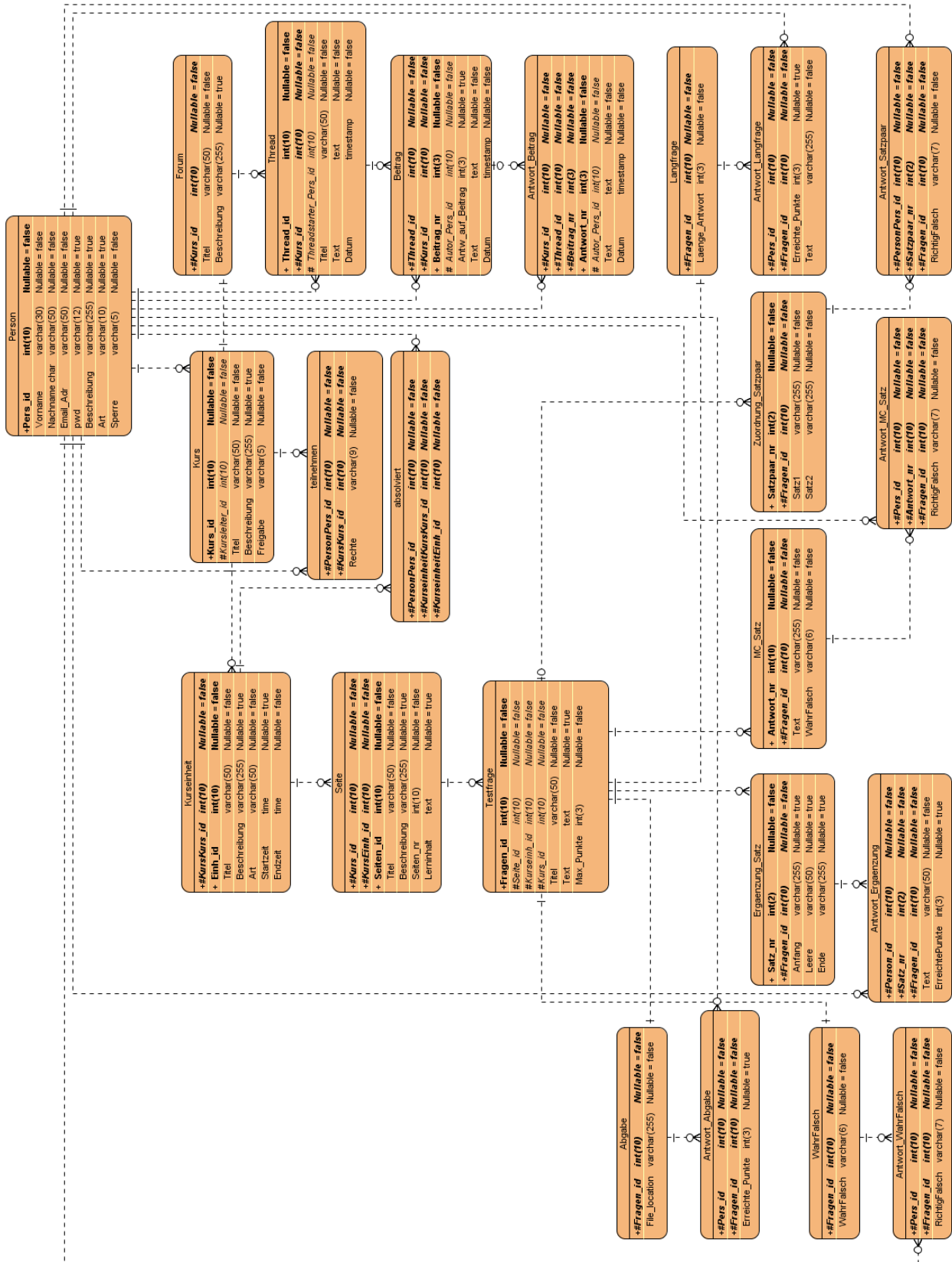


Abbildung 16 - ER Diagramm Gesamtdesign

3.8 Architektur

Das nachfolgende Komponentendiagramm zeigt die 4-Tier-Architektur, sowie die Paketstruktur innerhalb der jeweiligen Schichten. Die Trennung zwischen Präsentations- und Service-Layer, sowie zwischen Logik- und Daten-Layer erfolgt rein durch die Paketstruktur und die Anbindung durch die möglichst lose gehaltenen Methodenaufrufe. Trotzdem erachte ich diese Betrachtungsweise als sinnvoll, da die Präsentations- und die Daten-Schicht bei Bedarf mit nur geringem Aufwand ausgetauscht werden können. Zudem ist das GWT, wie eingangs erwähnt, äußerst konsequent betreffend der Trennung zwischen Programmlogik und Darstellung des User-Interfaces.

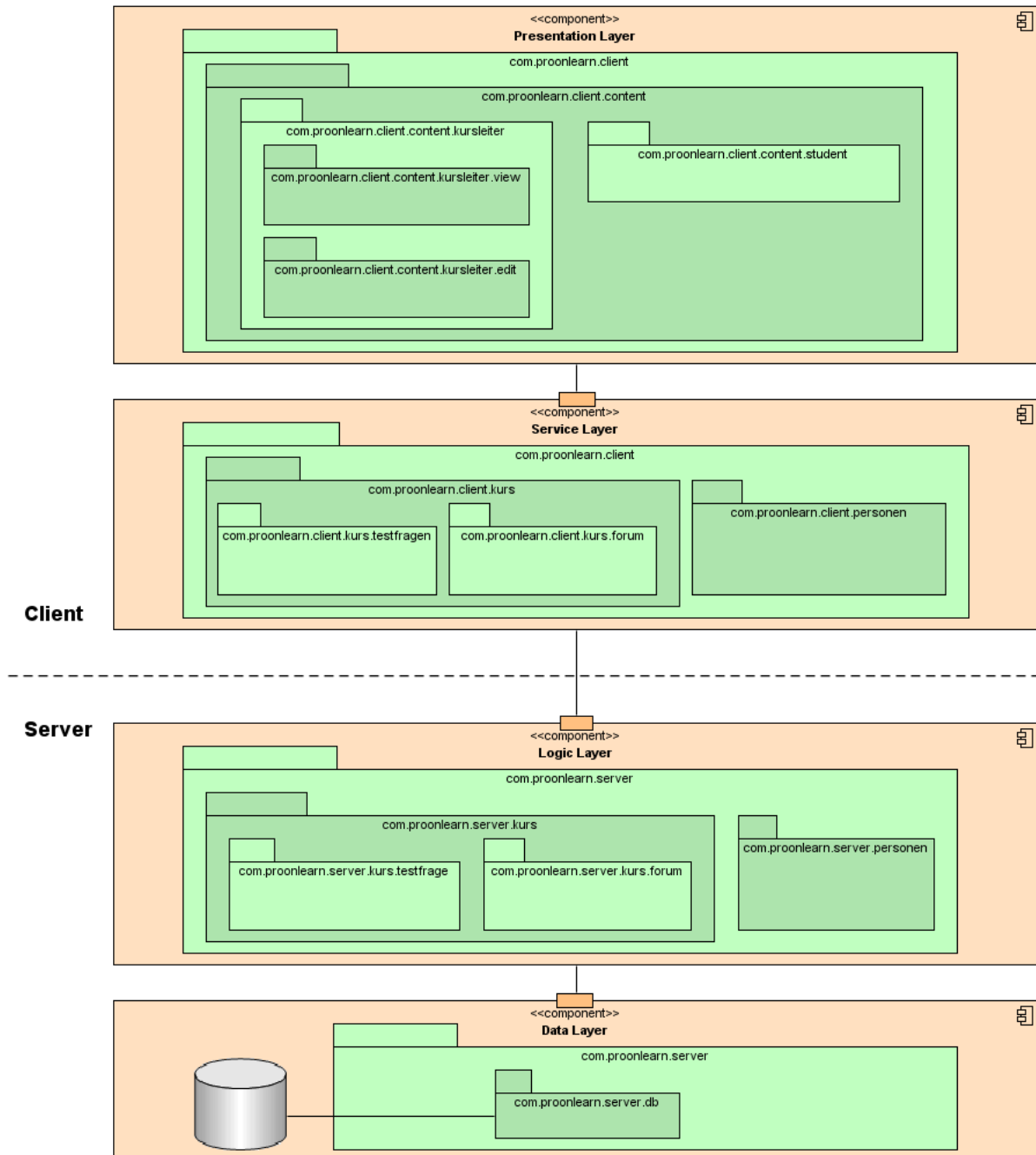


Abbildung 17 - Architektur

3.9 Klassendesign

Die bisherigen Designschritte – Use-Cases, Sequenzdiagramme mit den zugehörigen Klassen und deren wichtigsten Funktionen, Sitemaps, Entity-Relationship Diagramme und das Schichtenmodell mit der Paketstruktur – bilden nun die Grundlage für das genaue Klassendesign. Dies soll in den folgenden Unterkapiteln auf Basis der vier Schichten erläutert werden.

3.9.1 Service- und Logik-Schicht

Die Service- und Logik-Schicht werden zusammen betrachtet, da deren Aufbau analog zueinander erfolgt. Die zur Verfügung gestellten Services sind auf die jeweilige Benutzergruppe bzw. den Bedarf abgestimmt. Beispielsweise existieren für eine Kurseinheit drei Service-Interfaces: Eines zur Erstellung und zum Editieren durch den Kursleiter, ein weiteres zum Betrachten und Abarbeiten für Studenten und ein drittes zur Betreuung wiederum durch den Kursleiter. Diese Service-Interfaces befinden sich – gemeinsam mit den obligatorischen Interfaces zum asynchronen Methodenaufwurf am Client. Die Implementation erfolgt in einer gemeinsamen Klasse am Server.

Klassendiagramm: Personen

In der nächsten Abbildung wird ersichtlich, dass die Service-Interfaces am Client das RemoteService-Interface des GWT erweitern, sowie asynchrone Interfaces zugeordnet sind. Zusätzlich implementieren die zugehörigen Klassen am Server das RemoteServiceServlet. Zudem wird dargestellt, wie die einzelnen Usergruppen ihre Eigenschaften und Fähigkeiten der nächst höheren Gruppe weiter vererben.

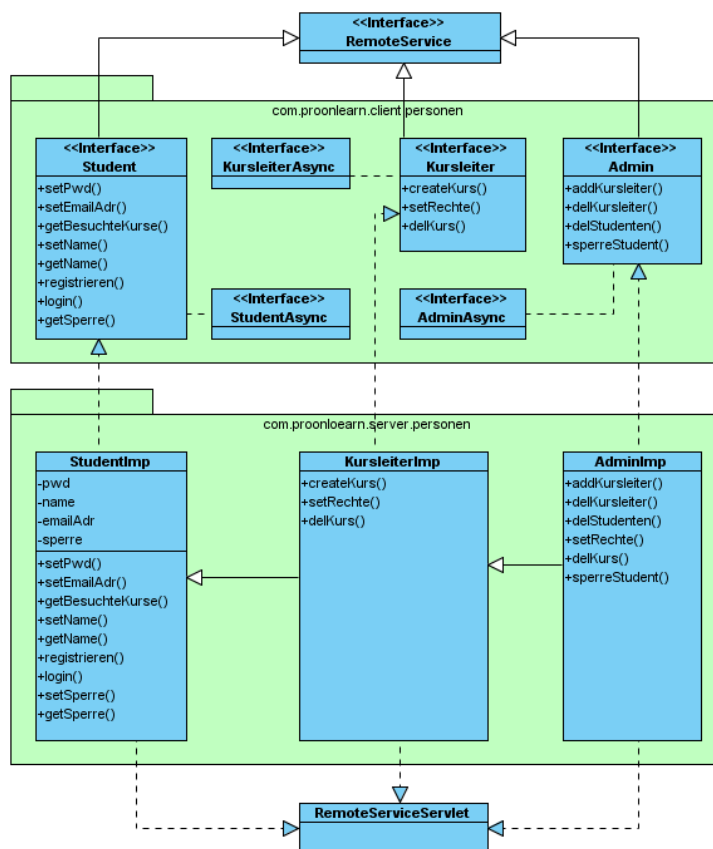


Abbildung 18 - Klassendiagramm Personen

Logik-Schicht als komplex. Sie beerben alle die Superklasse „Testfrage“ und unterteilen sich in Fragen deren Punktevergabe automatisch erfolgt und solche, bei denen der Kursleiter die Punkte selbst vergibt.

Automatische Fragen

Für sie existieren jeweils Service-Interfaces zum Erstellen durch den Kursleiter sowie zum Betrachten und Beantworten der Fragen durch die Studenten.

Multiple Choice:

Bei Multiple Choice–Fragen können vom Kursleiter mehrere Sätze mit jeweiligem Wahrheitsgehalt generiert werden. Die Studenten markieren dann mittels Checkbox, ob der Satz der Wahrheit entspricht oder nicht.

Wahr – Falsch:

Eine Wahr–Falsch – Frage entspricht der einfachsten Form einer Multiple Choice – Frage. Hier existiert nur ein einzelner Satz.

Zuordnungen:

Bei Zuordnungen werden vom Kursleiter mehrere zusammengehörige Satzpaare erstellt. Der Student kann diese mittels Drag’n’Drop passend zusammen führen.

Manuelle Fragen

Zusätzlich zu den Service-Interfaces die auch bei den automatischen Fragen gebraucht werden, werden hier noch Interfaces benötigt die es dem Kursleiter ermöglichen die Antworten der Studenten zu betrachten und Punkte zu vergeben.

Langfragen:

Bei Langfragen erstellt der Kursleiter eine Frage und gibt vor, wieviele Zeichen eine Antwort maximal haben darf.

Ergänzungen:

Ergänzungen bestehen aus einem Satzanfang, einer leeren Textbox und einem Satzende. Auch hier gibt der Kursleiter die maximale Zeichenanzahl vor.

Abgaben:

Abgaben sehen File – Uploads vor. Der Student erstellt ein Dokument und ladet dieses hoch. Der Kursleiter betrachtet dieses und vergibt anschließend eine entsprechende Punktezahl.

Das nachfolgende Diagramm in Abbildung 20 zeigt den Aufbau der Testfragen. Die Service-Interfaces am Client besitzen asynchrone Interfaces und erweitern das RemoteService-Interface des GWT. Die zugehörigen Klassen am Server implementieren je nach Zusammengehörigkeit wieder mehrere dieser Interfaces und erweitern das RemoteServiceServlet des GWT.

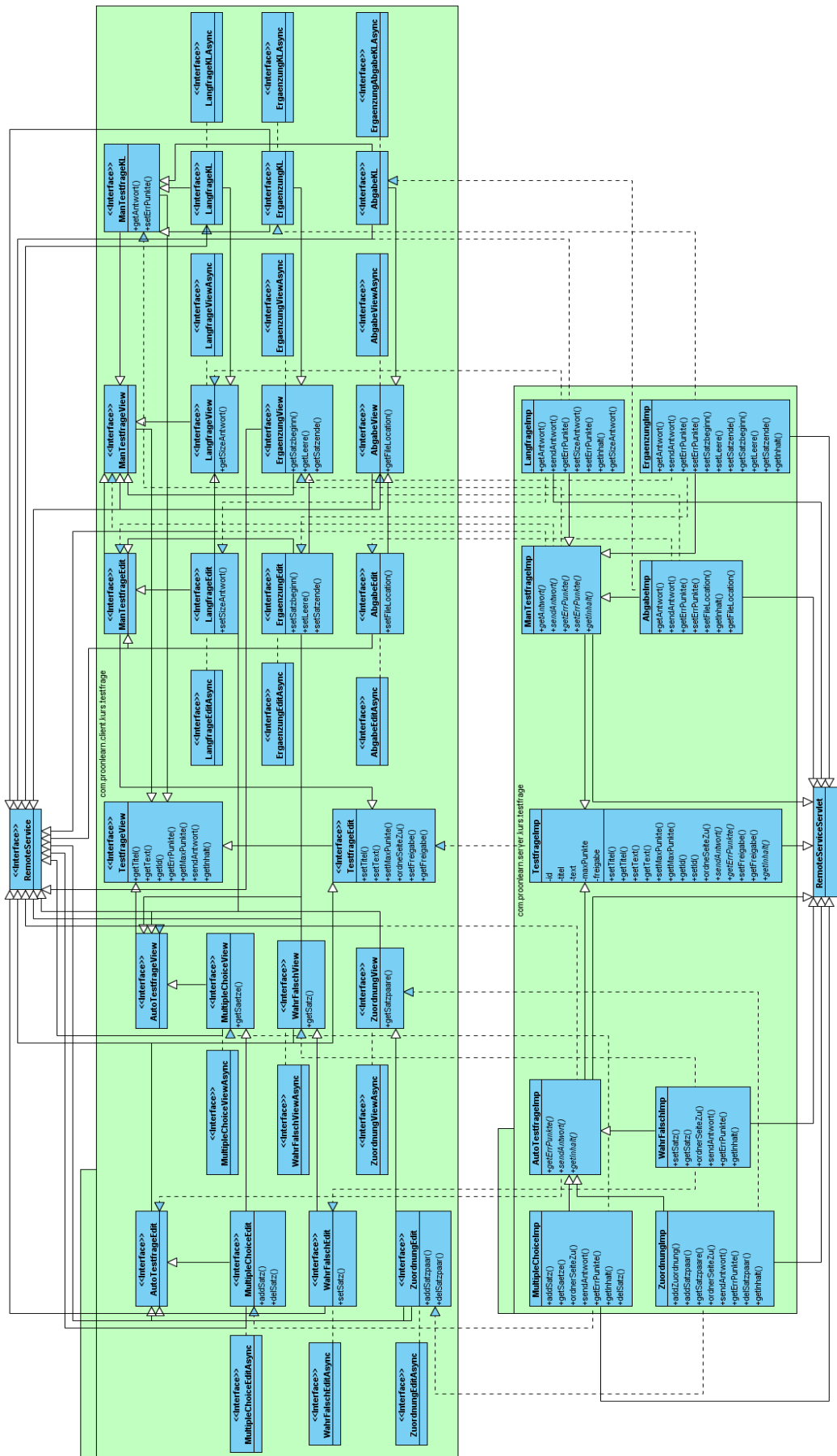


Abbildung 20 - Klassendiagramm Testfragen

Klassendiagramm: Forum

Nachstehende Abbildung zeigt das Klassendiagramm welches sich für die Forumsfunktion verantwortlich zeigt:

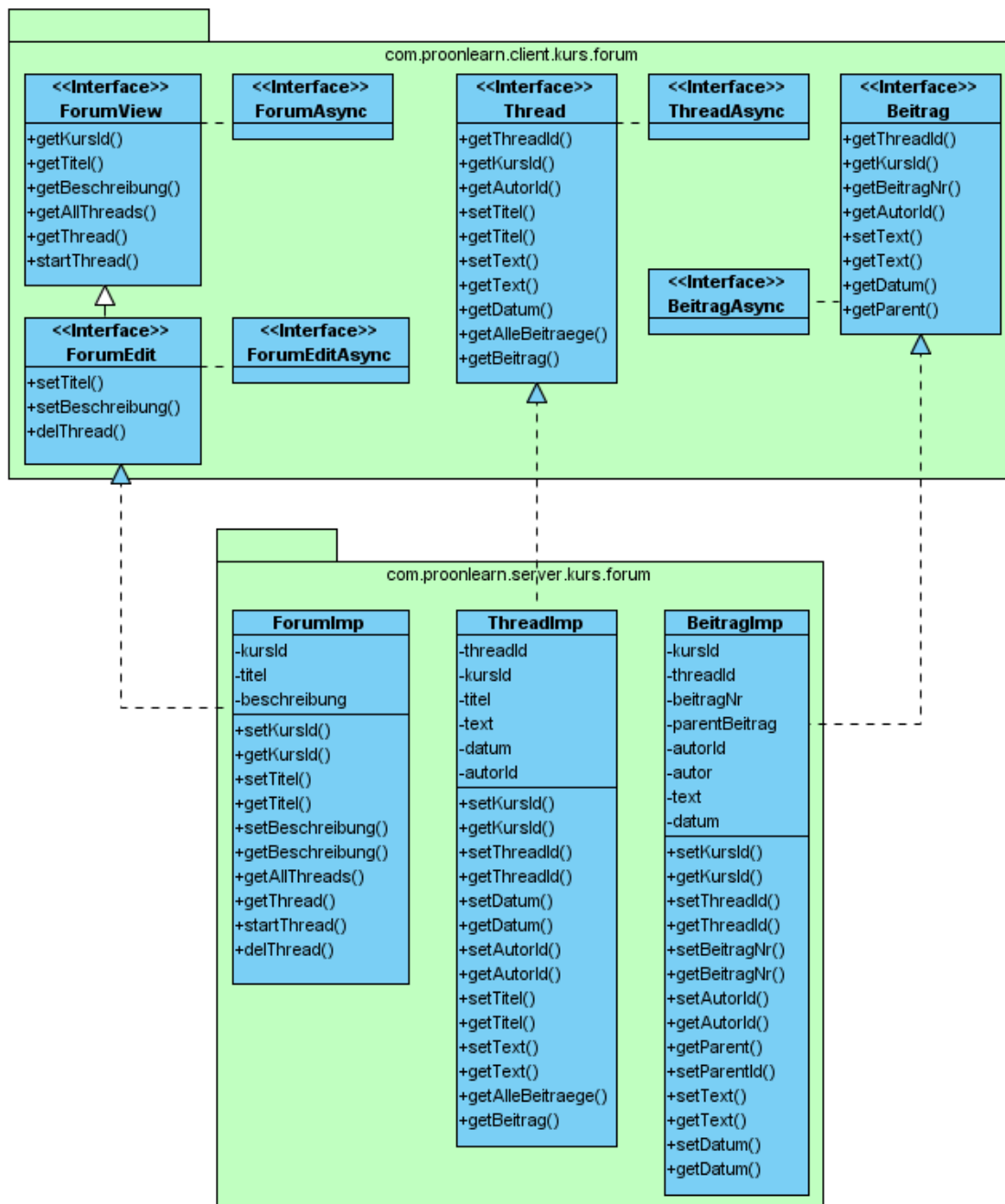


Abbildung 21 - Klassendiagramm Forum

Allein jene Klassen aus der Logikschicht greifen auf die Datenschicht zu. Im folgenden Kapitel werde ich auf diesen Layer eingehen.

3.9.2 Datenschicht

In der Datenschicht wird mit der Klasse „MySQL“ die Verbindung zur MySQL – Datenbank hergestellt. Wie oben erwähnt, steht den Klassen der Logikschicht je nach

Benutzer eines von drei Interfaces zur Verfügung das die benötigten Funktionen beinhaltet. Die Klasse „PersonDB“ implementiert diese drei Interfaces. Die folgende Abbildung zeigt das zugehörige Klassendiagramm:

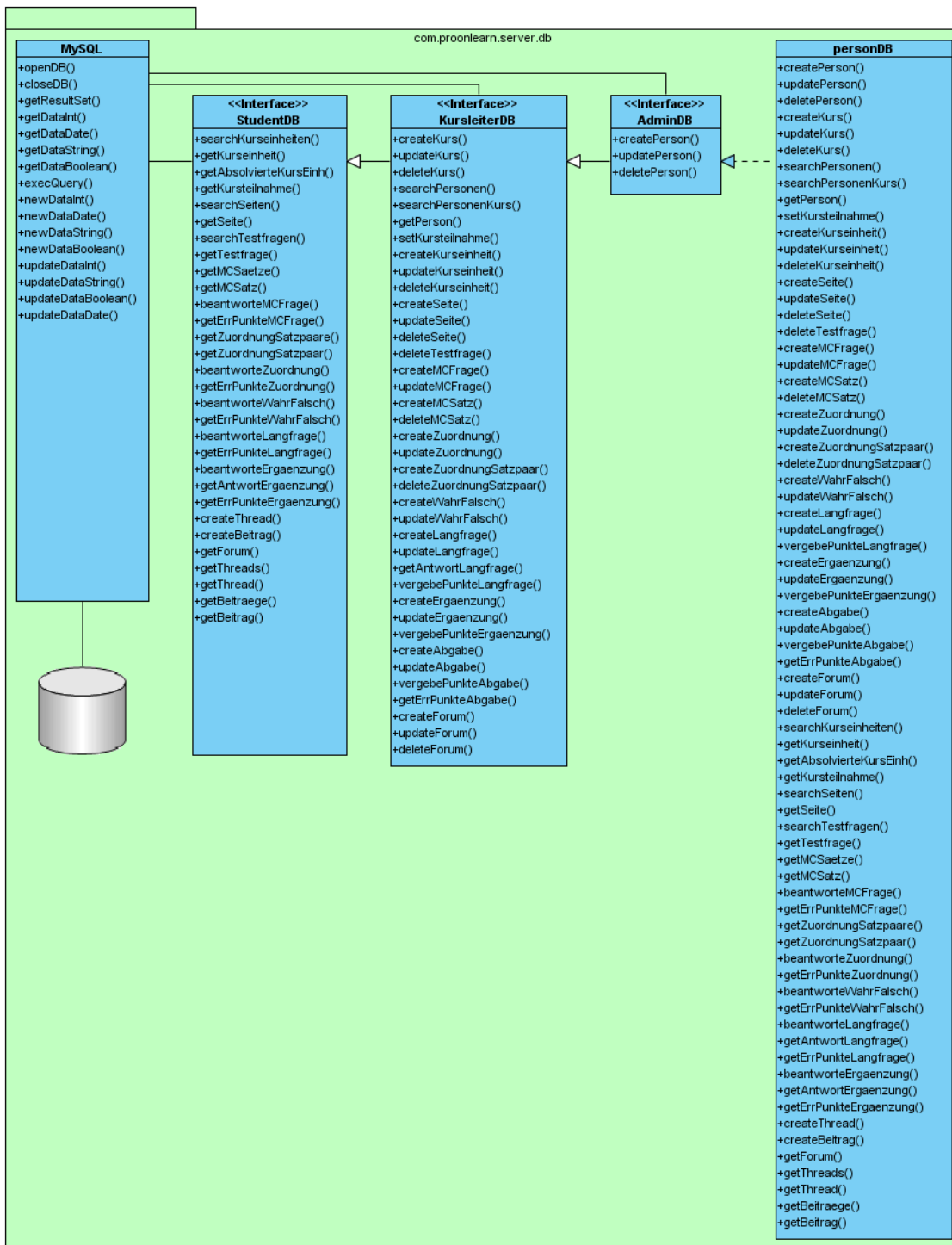


Abbildung 22 - Klassendiagramm Datenschicht

3.9.3 Präsentationsschicht

Das nächste Diagramm zeigt den Paketaufbau am Client. Als Einstiegspunkt für die Applikation dient die Klasse „Proonlearn“, welche das vom GWT benötigte Interface

„EntryPoint“ implementiert. Diese Klasse ist also jenes Modul, auf das beim Aufruf der Lernplattform zugegriffen wird. Von ihr aus werden weitere Klassen aufgerufen, die Inhalte bereitstellen sollen. Die Klasse „HeaderWidget“ stellt einen Header zur Verfügung. Ein Footer wird durch „FooterWidget“ realisiert. „MenuWidget“ generiert ein Menü, das durch die Methode „showOptions()“ in den jeweiligen Klassen der Präsentations-Schicht aufgerufen wird und sich nach den Bedürfnissen des Users richtet. Die Klasse „ContentWidget“ dient als Leinwand für den eigentlichen Inhalt der Plattform. Sie stellt die Darstellung des Logins zur Verfügung und greift danach auf die Klassen in den Unterpaketen des Packages „content“ zu. Diese bedienen sich wiederum an den Interfaces zum asynchronen Methodenaufruf aus dem Paket „kurs“.

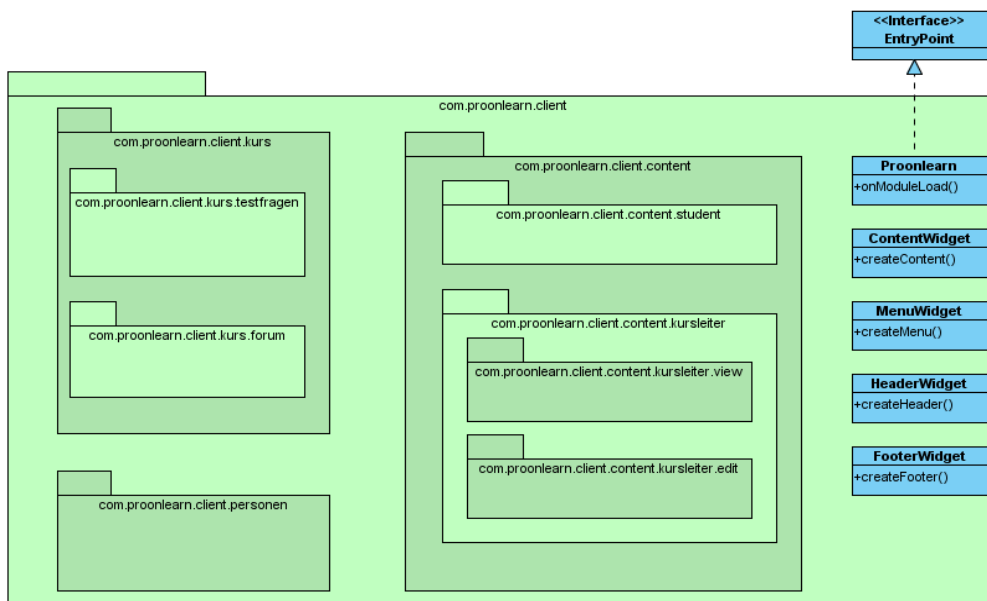


Abbildung 23 - Klassendiagramm Präsentationsschicht

Content: Student

Folgend wird die Klassenstruktur zur Darstellung der Online-Lernplattform für Studenten gezeigt.

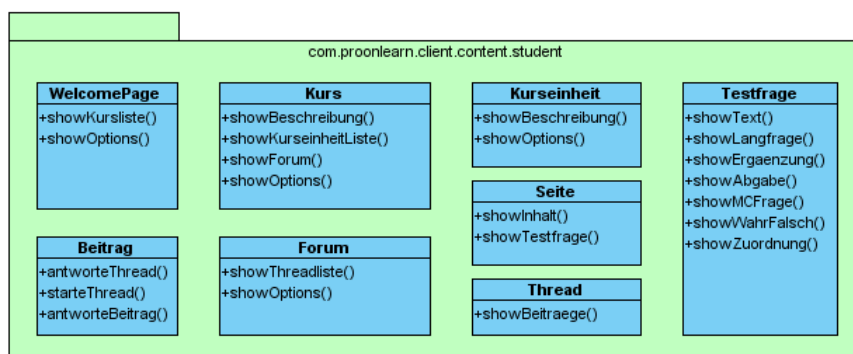


Abbildung 24 - Klassendiagramm Content Student

Content: Kursleiter

Wie erwähnt hat der User „Kursleiter“ zwei Sichtweisen auf die Applikation: Einerseits um Kurse, Kurseinheiten, Seiten usw. zu erstellen. Andererseits um Kurse zu betreuen. Daher ist das Unterpaket „kursleiter“ wiederum in zwei Unterpakete „view“ und „edit“

unterteilt. Die nachfolgende Abbildung zeigt diese Pakete sowie die enthaltenen Klassen. Da der Kursleiter auch am Forum teilnehmen kann, wird hier auf die Klassen im Package „student“ zugegriffen. Einzig die Klasse „Forum“ wird erweitert um das Löschen von Threads zu ermöglichen.

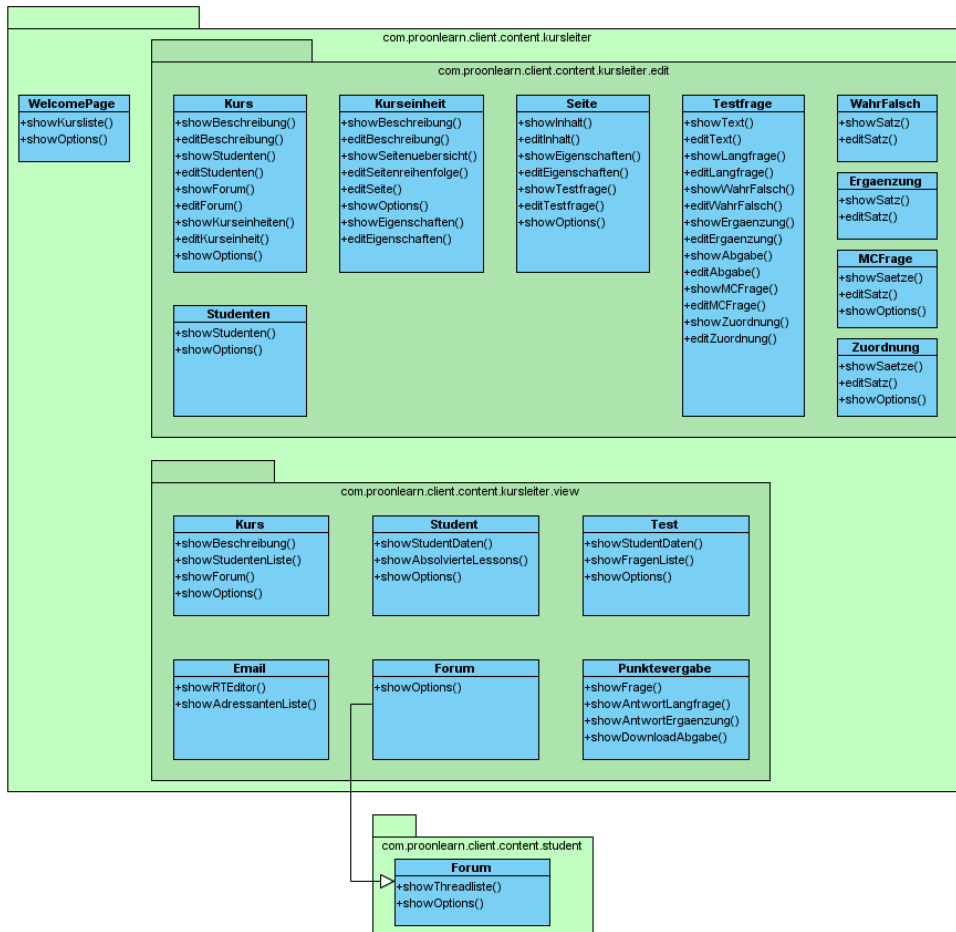


Abbildung 25 - Klassendiagramm Content Kursleiter

4 Implementierung

Der Fokus dieser Magisterarbeit liegt sicher in der praktischen Umsetzung und der Implementierung der vorgestellten Funktionen. Der Charakter der Arbeit entspricht daher jener von professioneller Softwareentwicklung in der Praxis. Dementsprechend soll hier auch nicht die genaue Funktionsweise des Google Web Toolkit erläutert werden. Trotzdem soll kurz umrissen werden, wie das GWT hier für die vorgestellten Erfordernisse herangezogen wurde.

4.1 Projektstruktur

Entsprechend der GWT Dokumentation (siehe [GwtDir]) besteht das Projekt nach der Generierung durch das Google Plugin auf höchstem Niveau aus dem src- und dem war-Verzeichnis sowie aus den eingebundenen Bibliotheken:

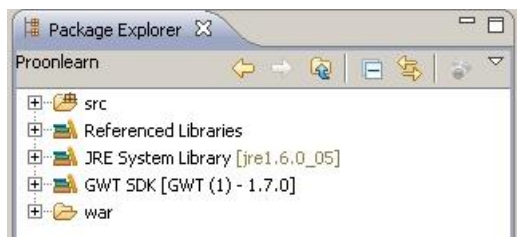


Abbildung 26 - Projektstruktur

src-Verzeichnis:

GWT-Anwendungen werden in Modulen konfiguriert. Ein XML-File dient dabei – wie bei [GwtXML] beschrieben – als zentrale Konfigurationsdatei. In ihr wird festgelegt, welche Module in der Applikation genutzt werden und welche Bibliotheken dazu eingebunden werden. Weiters wird bestimmt welche Java-Klasse als Einstieg dient:



Abbildung 27 - Moduldefinition

Die XML-Datei befindet sich im Root-Verzeichnis der Applikation – im diesem Falle „com.proonlearn“. Die Java-Klasse für den Applikationseinstieg muss das GWT-Interface „EntryPoint“ implementieren und besitzt die Methode „onModuleLoad()“. In dieser Methode befindet sich jener Code der beim Start ausgeführt wird. Von hier aus werden

auf die restlichen Klassen zugegriffen welche sich – ebenfalls im src-Verzeichnis – in der Paketstruktur entsprechend den Klassendiagrammen aus dem Kapitel "Design der Applikation" dieser Arbeit befinden:

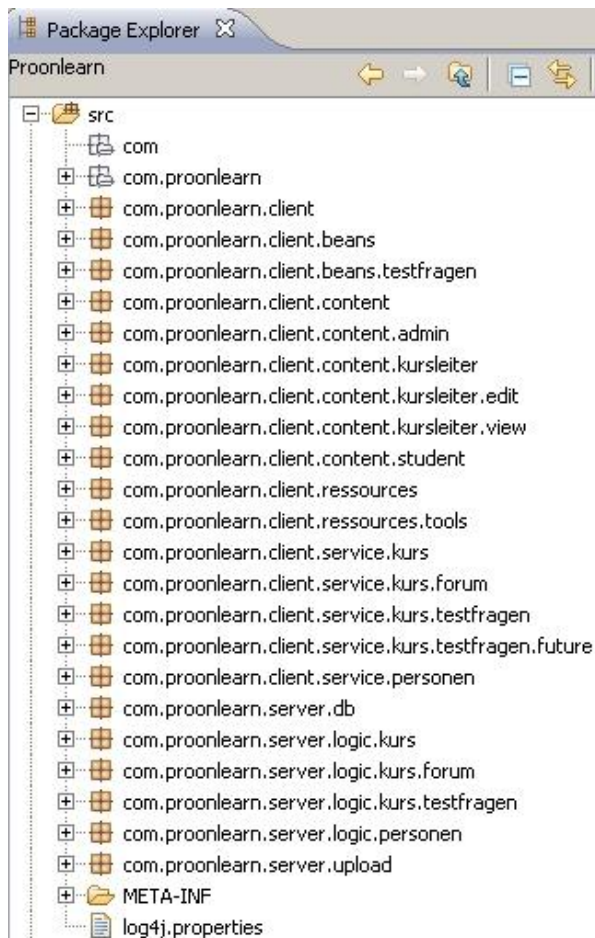


Abbildung 28 - Paketstruktur

war-Verzeichnis:

Dieses enthält die kompilierten JavaScript Klassen sowie statischen Inhalt wie beispielsweise:

- die Host-Page welche die gesamte Webapplikation beheimatet
- die CSS Datei welche für die Optik der grafische Darstellung zuständig ist
- das Favicon

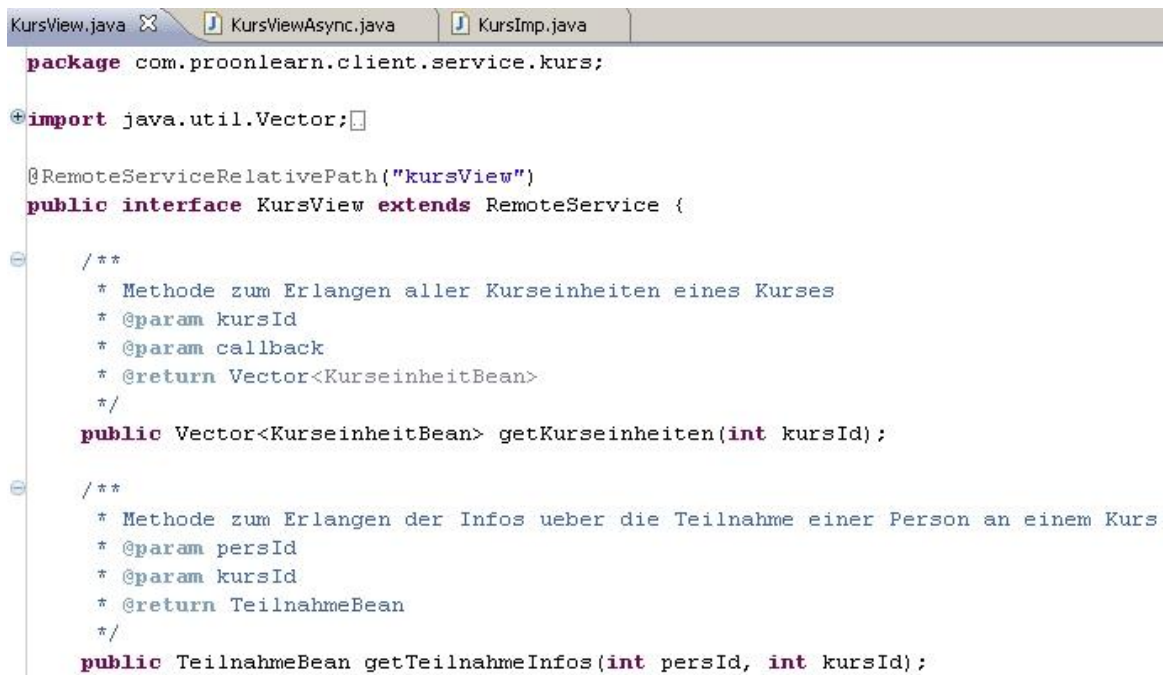
Weiters befinden sich im war-Verzeichnis die durch die Webapplikation automatisch generierten Ordner für den Up- und Download von Dateien.

4.2 Kommunikation

Die Kommunikation zwischen Client und Server wird im GWT mittels „Asynchronous Callback“ umgesetzt – siehe dazu [GwtRPC]. Dabei werden serialisierte Objekte wie beispielsweise primitive Datentypen aber auch JavaBeans über das Netzwerk ausgetauscht.

Am Client müssen dabei synchrone Interfaces erstellt werden, welche die benötigten Methoden definieren und das GWT Interface „RemoteService“ erweitern. Diese Interfaces müssen auf Serverseite von Klassen implementiert werden welche das GWT „RemoteServiceServlet“ erweitern. Damit diese Methoden vom Client aus aufgerufen werden können müssen hier noch asynchrone Interfaces erstellt werden. Die GWT Namenskonvention gibt hier den gleichen Namen wie bei den entsprechenden synchronen Interfaces vor, jedoch mit dem Zusatz „Async“.

Ein Beispiel aus dem Proonlearn Projekt zeigt folgend die entsprechenden Dateien anhand eines Ausschnittes des Interfaces für das Objekt „KursView“ welches die Funktionen eines Kurses für die Studenten-View bereitstellt. Das Interface „KursView“:



```

KursView.java X  KursViewAsync.java  KursImp.java
package com.proonlearn.client.service.kurs;

import java.util.Vector;

@RemoteServiceRelativePath("kursView")
public interface KursView extends RemoteService {

    /**
     * Methode zum Erlangen aller Kurseinheiten eines Kurses
     * @param kursId
     * @param callback
     * @return Vector<KurseinheitBean>
     */
    public Vector<KurseinheitBean> getKurseinheiten(int kursId);

    /**
     * Methode zum Erlangen der Infos ueber die Teilnahme einer Person an einem Kurs
     * @param persId
     * @param kursId
     * @return TeilnahmeBean
     */
    public TeilnahmeBean getTeilnahmeInfos(int persId, int kursId);
  
```

Abbildung 29 - Interface KursView

Die Methoden aus „KursView“ werden serverseitig in „KursImp“ implementiert, wie die folgende Abbildung 30 zeigt.

```

KursView.java | KursViewAsync.java | KursImp.java X
package com.proonlearn.server.logic.kurs;

import java.util.Calendar;

public class KursImp extends RemoteServiceServlet implements KursEdit, KursView, KursViewKL {

    private static final long serialVersionUID = 1L;

    /**
     * Methode zum Erlangen aller Kurseinheiten eines Kurses
     * @param kursId
     * @param callback
     * @return Vector<KurseinheitBean>
     */
    public TeilnahmeBean getTeilnahmeInfos(int persId, int kursId) {
        StudentDB db = new PersonDB();
        return db.getTeilnahmeInfos(persId, kursId);
    }

    /**
     * Methode zum Speichern des Titels eines Kurses
     */
    public boolean setTitel(int kursId, String titel) {
        KursleiterDB db = new PersonDB();
        KursBean kurs = db.getKurs(kursId);
        return db.updateKurs(kursId, kurs.getKursleiterId(), titel, kurs.getBeschreibung(), k
    }
}

```

Abbildung 30 - Implementation KursImp

Das asynchrone Interface „KursViewAsync“ ermöglicht den Methodenaufruf durch den Client:

```

KursView.java | KursViewAsync.java X | KursImp.java
package com.proonlearn.client.service.kurs;

import java.util.Vector;

public interface KursViewAsync {

    void getKurseinheiten(int kursId, AsyncCallback<Vector<KurseinheitBean>> callback);

    void getTeilnahmeInfos(int persId, int kursId, AsyncCallback<TeilnahmeBean> callback);

    void getKursId(AsyncCallback<Integer> callback);
}

```

Abbildung 31 - Interface KursViewAsync

Mit dem asynchronem Aufruf der serverseitig implementierten Methoden mittels „AsyncCallback<>()“ werden Daten zum Client übertragen und können hier dargestellt werden ohne die gesamte Seite neu zu laden:

```

private void showKurseinheiten() {
    panel.add(new HTML("Folgende Kurseinheiten sind zu betreuen:"));
    kurs.getKurseinheiten(kursBean.getKursId(), new AsyncCallback<Vector<KurseinheitBean>>() {

        public void onFailure(Throwable caught) {
            panel.add(new HTML("Serverfehler beim Erlangen der Kurseinheiten aufgetreten"));
            System.out.println("KursKL getKurseinheiten Exception:\n"+caught.getMessage());
        }

        public void onSuccess(Vector<KurseinheitBean> result) {
            if(result==null || result.isEmpty()) {
                panel.add(new HTML("Derzeit hat dieser Kurs (noch) keine Kurseinheiten"));
                return;
            }
        }
    });
}

```

Abbildung 32 - Methode AsyncCallback<>()

4.3 Erstellung des User Interface

Die grafische Benutzerschnittstelle wird innerhalb in eine HTML-Seite, die sogenannte „HTML Host Page“ (siehe [GwtHost]) welche sich im war-Verzeichnis befindet, eingebettet und mittels Panels und Widgets erzeugt. GWT Panels sind im wie bei HTML als Container zu betrachten. Sie können verschiedene Layouts aufweisen und GUI Komponenten beinhalten wie weitere Panels oder eben Widgets. Bei Widgets handelt es sich um Komponenten wie Buttons, Tabellen oder Ähnliches. Zur Erstellung der grafischen Schnittstelle siehe auch die GWT-Dokumentation unter [GwtGUI]. Bei beiden Arten von GUI steht bereits eine Vielzahl von Standardkomponenten von Haus aus zur Verfügung. Eigene Widgets können jedoch selbst geschaffen werden, indem gegebene Widget-Klassen erweitert werden.

Der grundsätzliche Aufbau der grafischen Darstellung im vorliegenden Projekt geht folgendermaßen vor:

- Die HTML Host-Page „Proonlearn.html“ ermöglicht die grafische Darstellung mit folgender – hier aus Platzgründen umgebrochener – Zeile:

```

<script type="text/javascript" language="javascript"
src="proonlearn/proonlearn.nocache.js"></script>

```

Abbildung 33 - Einbindung der grafischen Darstellung

- Durch <div> Container werden sogenannte „RootPanels“ definiert, wodurch die Darstellung räumlich platziert wird.
- Die Klasse „Proonlearn“ die sich im Root-Verzeichnis der Client-Seite des Projektes befindet greift mit der statischen Methode RootPanel.get() auf solch einen Container zu:

```

private final Panel cPanel = RootPanel.get("contentContainer");
private final Panel hPanel = RootPanel.get("headerContainer");
private final Panel fPanel = RootPanel.get("footerContainer");
private final Panel mPanel = RootPanel.get("menuContainer");

```

Abbildung 34 - Methode RootPanel.get()

- Jede Entität wie „Kurs“, „Kurseinheit“, etc. welche eine grafische Repräsentation in einem dieser erlangten Panels benötigt wird in Proonlearn durch eine Klasse realisiert welche die abstrakte Klasse „ContentWidget“ implementiert, die das

Interface „LazyPanel“ erweitert und bereits diverse Seitenmaße definiert. Nachfolgend zu sehen ist die Abbildung dieser abstrakten Klasse:

```

package com.proonlearn.client;

import com.google.gwt.user.client.ui.LazyPanel;

public abstract class ContentWidget extends LazyPanel {

    protected String width = "800px";
    protected String widthM10 = "790px";
    protected String widthM20 = "780px";
    protected String widthM100 = "700px";
    protected String widthD2 = "400px";

    protected int widthInt = 800;

    protected abstract Widget createWidget();
}

```

Abbildung 35 - ContentWidget

Folgend die Umsetzung anhand des Beispiels einer Kursbeschreibung aus der Sicht eines Kursleiters:

```

package com.proonlearn.client.content.kursleiter.view;

import java.util.Iterator;

public class KursKL extends ContentWidget {

    private KursViewKLAsync kurs = GWT.create(KursViewKL.class);
    private KursleiterApplication app;
    private KursBean kursBean;
    private Vector<KurseinheitBean> kurseinheiten;

    private MyImageBundle myImageBundle = GWT.create(MyImageBundle.class);
    private VerticalPanel panel = new VerticalPanel();

    public KursKL(KursleiterApplication app) {
        this.app = app;
    }

    protected Widget createWidget() {
        panel.setSpacing(5);
        panel.setWidth(width);
        return panel;
    }

    public void showContent() {
        kursBean = app.getKursBean();
        showKurseinheiten();
    }
}

```

Abbildung 36 - ContentWidget Kursleiter

- Die Seite wird nun aufgebaut, indem das erweiterte LazyPanel der ContentWidget-Klasse von der Applikation im RootPanel platziert wird, welches in der Proonlearn-Klasse erlangt wurde. In dieses Panel können je nach Bedarf weitere Panels gelegt werden um ein gewünschtes Layout zu erlangen. Nicht zuletzt erfolgt das Hinzufügen der einzelnen Widgets.

4.3.1 Verwendete Panels und Widgets

Nachstehend eine Liste der GWT Panel-Klassen die bei der Gestaltung der GUI zur Verwendungen kamen:

- **Panel:**
Panels können technisch nicht getrennt von Widgets betrachtet werden. Im eigentlichen Sinne sind Panels spezielle Widgets welche andere Widgets beinhalten können. „Panel“ stellt die abstrakte Oberklasse für diese speziellen Widgets dar.
- **RootPanel:**
RootPanel ist jenes Top-Level-Panel welches mittels der statischen Methode RootPanel.get() erlangt wird und sämtliche anderen Widgets enthält.
- **FlowPanel:**
FlowPanel stellt die einfachste Art eines Panels dar. HTML-seitig stellt es einen <div> Container dar, in welchem der Reihe nach Elemente platziert werden können ohne ein besonderes Layout zur Verfügung zu stellen.
- **SimplePanel:**
SimplePanel stellt ein Panel dar, welches genau ein Widget beinhalten kann. Es wird im Projekt beispielsweise bei der Darstellung von Statusmeldungen genutzt.
- **LazyPanel:**
Die Klasse „LazyPanel“ dient dazu ein Panel kontrolliert zum genau gewünschten Zeitpunkt mittels den Methoden setVisible(true) oder ensureWidget() zu instanziiieren. Bei Proonlearn werden alle Top-Level Panels welche Elemente wie beispielsweise „Kurs“, „Kurseinheit“ oder „Testfrage“ enthalten von einer Klasse abgeleitet welche „LazyPanel“ erweitert.
- **AbsolutePanel:**
Die Klasse AbsolutePanel wird dann eingesetzt, wenn es notwendig ist, die enthaltenen Widgets absolut zu positionieren, bzw. wenn sich diese überlappen können müssen. Im Projekt wurden diese Eigenschaften bei der Realisierung von Drag'n'Drop benötigt.
- **VerticalPanel:**
In einem VerticalPanel werden enthaltene Kindselemente in einer vertikalen Spalte angeordnet.
- **HorizontalPanel:**
Analog zum VerticalPanel ordnet das HorizontalPanel alle enthaltenen Elemente in einer horizontalen Reihe an.
- **DeckPanel:**
Das DeckPanel stellt einen Stapel von Panels dar bei welchem immer nur ein Einziges sichtbar gemacht werden kann. Im Projekt kommt es öfter zum Einsatz, beispielsweise wenn Inhalte wahlweise dargestellt oder bearbeitet werden sollen.
- **Grid:**
Dieses Panel stellt eine Erweiterung einer HTML-Tabelle dar und kann Text, HTML oder wiederum weitere Widgets in seinen Zellen beinhalten.
- **FocusPanel:**
Stellt für ein Panel die Eigenschaft zur Verfügung, dass der Inhalt des Panels auswählbar wird. Dies wird bei Proonlearn genutzt, wenn nach einer Drag'n'Drop-Aktion die Neuordnung von GUI-Elementen ausgewertet und gespeichert werden muss.

- DecoratorPanel:
Das DecoratorPanel stellt ein SimplePanel dar welches eine grafische Umrandung aufweist und damit GUI-Elemente optisch aufbessert.

Folgende Widgets, welche nativ vom GWT zur Verfügung stehen, wurden genutzt:

- Widget:
Widget stellt jene Oberklasse dar, von der die große Mehrheit aller GUI Elemente abgeleitet wird. Sie stellt die beiden essentiellen Fähigkeiten von Widgets zur Verfügung. Nämlich erstens die Fähigkeit zu einem Panel als Kindselement hinzugefügt zu werden und zweitens Browser-Events entgegen zu nehmen.
- HTML:
Das "HTML"-Widget kann beliebiges HTML darstellen. Im Proonlearn-Projekt wird vielfach genutzt um Text-Meldungen wiederzugeben:
- TextBox:
Vergleichbar mit HTML-Textboxen wird dieses einzeilige Text-Widget beispielsweise beim Login oder bei diversen Dialogen verwendet.
- PasswordTextBox:
Identisch mit dem TextBox-Widget mit dem Unterschied, dass Eingaben in Punkt-Form maskiert dargestellt werden.
- Hyperlink:
Das Hyperlink-Widget erzeugt einen Querverweis innerhalb eines GWT-Projektes. Dabei dient es zur Navigation im History-Stack der laufenden GWT-Applikation – ähnlich den Back- und Forward-Buttons eines Webbrowsers – siehe dazu auch [GwtHistory]. Es dient jedoch nicht – wie vielleicht vermuten ließe – zur Erzeugung eines beliebigen Verweises im World Wide Web.
- Button:
Ein Widget das einen Standard-Button erzeugt.
- RadioButton:
Dieses Widget stellt einen HTML-Radio Button zur Verfügung. Radio Buttons wurden beispielsweise bei der Gestaltung von Multiple Choice – Fragen benutzt.
- ListBox:
Mit dem ListBox-Widget kann dem User eine begrenzte Auswahl zur Verfügung gestellt werden. Als Darstellungsmöglichkeiten sind hier sowohl eine Dropdown-Liste als auch ein Listenfeld gegeben.
- Image:
Das Image-Widget erlaubt die Darstellung eines Bildes.
- Tree:
Mit dem Tree-Widget lassen sich dynamische Baumstrukturen erzeugen, deren Elemente vom Benutzer geöffnet, geschlossen und ausgewählt werden können. Es wurde bei der Darstellung des Menüs zur Navigation genutzt.

Eine Übersicht von Widget und Panels wird unter [WidgetGal] zur Verfügung gestellt.

Nachstehend eine Liste von Klassen welche als erweiterte Widgets grafische Elemente darstellen. Zur Erweiterung von GWT Widgets siehe auch [CustGUI].

- TitelEditor, MyDialogBox, ConfirmDialog

Dabei handelt es sich um Widgets welche Benutzereingaben per Maus und Keyboard entgegennehmen und verarbeiten können. Dabei wurden Standard-Widgets wie TextBox, PasswordTextBox, Button, etc. kombiniert.

- RichTextEditor
Der RichTextEditor wird sowohl bei der Erstellung von Beschreibungen für Kurse, Kurseinheiten und ähnliches als auch bei der Erstellung von Lerninhalten benötigt. Zum Einsatz kommt hier das GWT Widget RichTextArea welches komplexe Textformatierung ermöglicht. Dieses wird mit der Klasse RichTextToolbar kombiniert in welche im Wesentlichen eine ListBox erzeugt welche aus Bildern und Buttons zusammen gesetzt ist und Benutzereingaben wie Mausklick handhabt.
- StartEndZeitEditor
Dieses selbst erstellte Widget dient zum Festlegen des Zeitrahmens von Kurseinheiten wie Lessons oder Tests. Dabei wurde zweimal (je Anfangs- und Endzeitpunkt) das GWT Widget DatePicker benutzt, welches dem Benutzer erlaubt, per Mausklick ein bestimmtes Datum und eine Uhrzeit auszuwählen.
- Drag'n'Drop
Drag'n'Drop wurde im Projekt für zweierlei Zwecke eingesetzt: Einerseits kann beim Editieren eines Kurses auf diese Weise die Reihenfolge der Seiten einer Kurseinheit bequem verändert werden. Andererseits wird Drag'n'Drop genutzt wenn Student bei Testfragen Zuordnungen zu lösen haben, bei der die passenden Sätze mit der Maus aneinander gereiht werden müssen. Dazu wurde die externe Library „gwt-dnd-2.6.5.jar“ von Fred Sauer eingebunden (siehe [SauerDND]). Folgende Objekte dieser Library wurden genutzt:
 - PickupDragController
Der PickupDragController erlaubt es, ein Widget welches diesen Controller implementiert mit der Maus auszuwählen und zu verschieben.
 - FlowPanelDropController
Mit dem FlowPanelDropController kann ein verschobenes Widget, welches Kindselement eines FlowPanels ist, mit der Maus wieder „losgelassen“ werden. Dabei wird die Reihenfolge der im FlowPanel enthaltenen Elemente verändert.

5 Handbuch

Dieses Kapitel stellt das Handbuch für die Applikation dar. Da dieses Kapitel auch dafür dienen soll, die praktische Umsetzung der Webapplikation zu zeigen wird für das Handbuch eine Anzahl von Screenshots genutzt. Die Funktionen werden dabei aus folgenden drei Sichten beschrieben:

- Es soll gezeigt werden, wie ein Kursleiter einen neuen Kurs erstellen und editieren kann. Dabei wird auch die Generierung von Testfragen erläutert.
- Ein fiktiver Student Max Mustermann soll diesen erstellten Kurs anschließend besuchen und die Kurseinheiten absolvieren. Dies wird hier erklärend behandelt.
- Nun soll der Kursleiter den Kurs betreuen. Dabei wird dargestellt wie erreichte Punkte abgefragt und manuell Punkte vergeben werden können.

5.1 Kursleiter – Erstellen eines Kurses

Voraussetzung für das Generieren, Bearbeiten und Betreuen von Kursen ist das Vorhandensein eines entsprechenden Users mit der Rolle „Kursleiter“. Dies wird durch einen Login anhand von Email-Adresse und Passwort abgefragt:



Abbildung 37 - Login Kursleiter

Anhand der Benutzerdaten stellt das System diese Rolle fest – Student, Kursleiter oder Administrator. In unserem Falle wurde der User als Kursleiter erkannt. Auf seiner Welcome-Seite werden etwaig vorhandene Kurse dargestellt, bzw. wird die Möglichkeit zur Verfügung gestellt einen neuen Kurs zu generieren. Dabei wird zu allererst nach einem Namen für diesen Kurs gefragt:



Abbildung 38 - Kursleiter Welcome Page

Nach dem erfolgreichen Anlegen des Kurses gelangt der Benutzer auf die Übersichtsseite des Kurses. Hier kann der Titel nachträglich geändert und eine Kursbeschreibung verfasst werden:

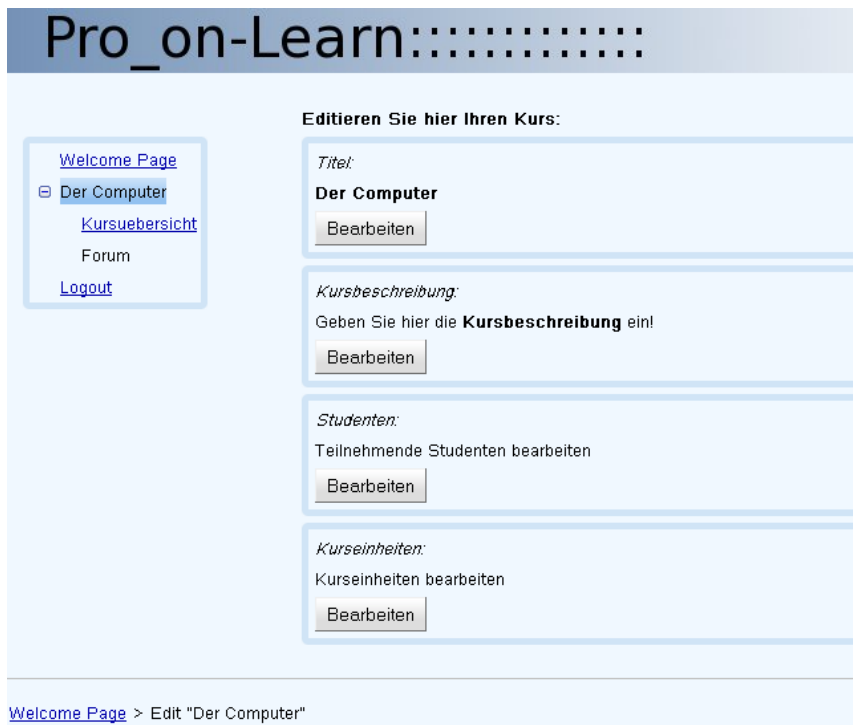


Abbildung 39 - Edit Kurs Übersicht

Zum Hinzufügen von Teilnehmenden erweitert man per Mausclick das Feld „Teilnehmende Studenten bearbeiten“. Durch Eingabe der eindeutig identifizierenden Email-Adresse und Festlegen der Rolle – Student oder Gast – per Radio-Button kann ein durch den Administrator registrierter Student als Kursteilnehmer definiert werden:

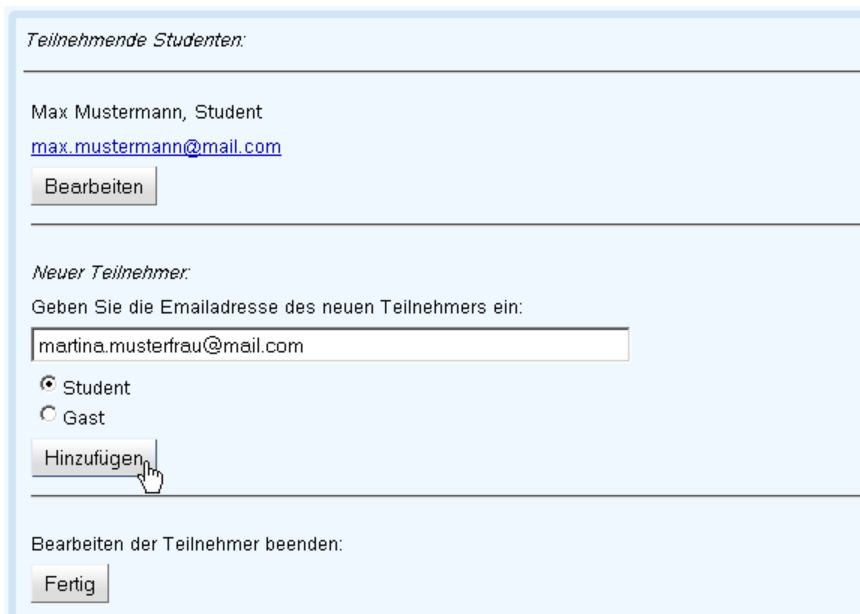


Abbildung 40 - Teilnehmerverwaltung

Nach dem alle Teilnehmer registriert sind kann das Feld durch Klicken auf den Button „Fertig“ wieder minimiert werden.

5.2 Kurseinheit – Präsentation

Nun können Kurseinheiten erstellt werden. Exemplarisch wird hier das Erstellen je einer Präsentation, einer Lesson sowie eines Tests gezeigt. Durch Klicken auf den entsprechenden Button „Bearbeiten“ wird das Feld erweitert. Wie in der nächsten Abbildung zu sehen ist, hat unser Beispielskurs noch keine Kurseinheiten. Wir erstellen zu Beginn eine erste Präsentation:

Kurseinheiten:

Derzeit hat dieser Kurs keine Kurseinheiten

Neue Kurseinheit anlegen:
Wählen Sie aus: Präsentation, Test oder Lesson

Bearbeiten der Kurseinheiten beenden:

Abbildung 41 - Präsentation erstellen

Analog zur Erstellung eines Kurses gelangt man ebenso beim Anlegen einer neuen Kurseinheit zu einer Übersichtsseite auf der Titel und Beschreibung bearbeitet werden können:

Pro_on-Learn:.....

[Welcome Page](#)
☐ [Der Computer](#)
 [Kursuebersicht](#)
 [Forum](#)
 [Logout](#)

Editieren Sie hier Ihre Präsentation:

Titel:
Der Computer - Grundlagen

Beschreibung der Kurseinheit:
Geben Sie hier die **Beschreibung der Kurseinheit** ein!

Seiten:
Seiten der Kurseinheit bearbeiten

[Welcome Page](#) > [Der Computer](#) > Edit "Der Computer - Grundlagen"

Abbildung 42 - Präsentation editieren

Anschließend können der Präsentation Seiten hinzugefügt werden:

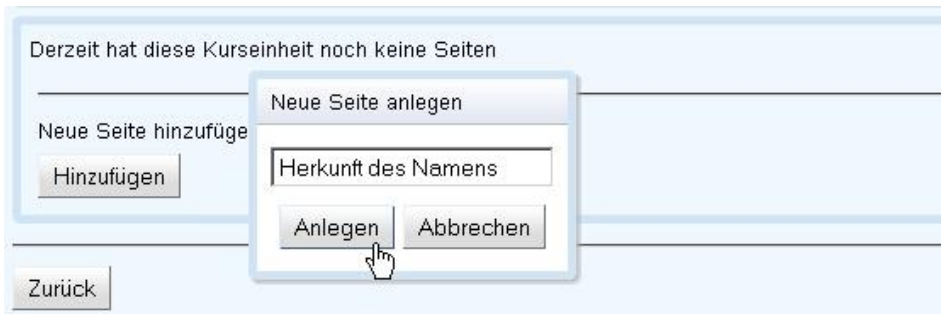


Abbildung 43 - Seite hinzufügen

Mittels Richtext-Editor kann der Inhalt für jede Seite erstellt und entsprechend formatiert werden. Zudem können Hyperlinks und Abbildungen hinzugefügt werden:

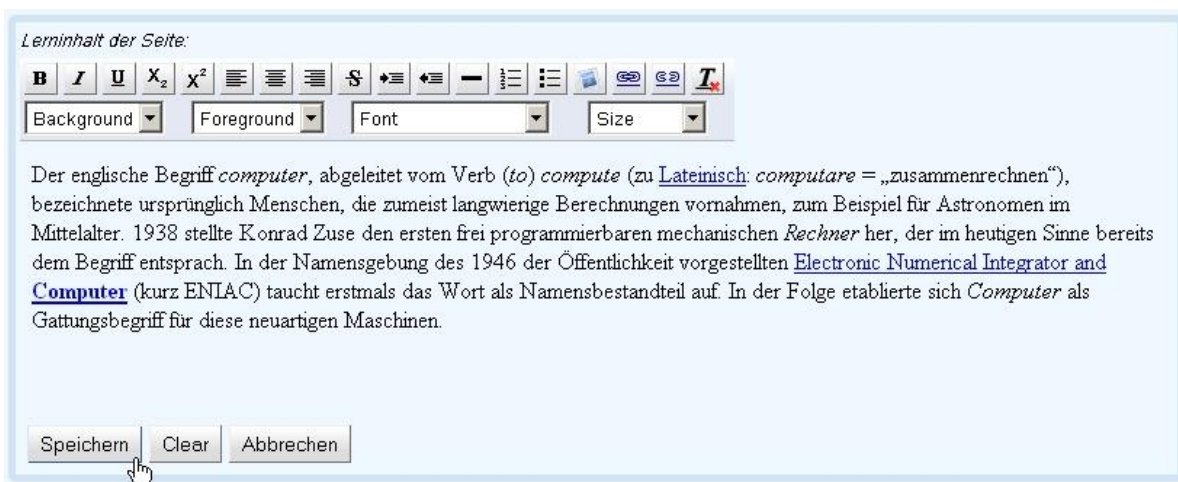


Abbildung 44 - Seite formatieren

Das Prinzip des Aufbaus von Seiten erfolgt nach immer gleichem Schema. Einer Seite kann grundsätzlich sowohl Lerninhalt als auch eine Testfrage hinzugefügt werden. Die Zugehörigkeit zu einer bestimmten Art von Kurseinheit schränkt dies ein. D.h., allein die Tatsache, dass eine Seite im Rahmen einer Präsentation erstellt wird legt fest, dass sich auf ihr nur Lerninhalt befinden kann, bzw. darf. Analog dazu kann auf den Seiten eines Tests jeweils nur eine Testfrage platziert werden. Nur Seiten von Lessons erlauben beides gleichzeitig.

Die erstellten Seiten werden als Liste dargestellt, wie auf Abbildung 45 zu sehen ist. Dabei können Seiten nachträglich bearbeitet oder gegebenenfalls auch gelöscht werden.

Herkunft des Namens

Der englische Begriff computer, abgeleitet vom Verb (to) compute (zu Lateinisch: computare = „zusammenrechnen“), bezeich...

Grundlagen

Grundsätzlich unterscheiden sich zwei Bauweisen: Ein Computer ist ein Digitalcomputer, wenn er mit digitalen Geräteeinhe...

Softwarearchitektur

Die Von-Neumann-Architektur ist gewissermaßen die unterste Ebene des Funktionsprinzips eines Computers oberhalb der elek...

Hardwarearchitektur

Das heute allgemein angewandte Prinzip, das nach seiner Beschreibung durch John von Neumann von 1946 als „Von-Neumann-Ar...

Einleitung

Ein Computer respektive Rechner ist ein Apparat, der Daten mithilfe einer programmierbaren Rechenvorschrift verarbeiten ...

Ändern sie die Reihenfolge per Drag'n'Drop:

Neue Seite hinzufügen:

Abbildung 45 - Seitenliste

Die Reihenfolge der Seiten kann per Drag'n'Drop verändert werden:

✕ **Einleitung**

Ein Computer respektive Rechner ist ein Apparat, der Daten mithilfe einer programmierbaren Rechenvorschrift verarbeiten ...

Der englische Begriff computer, abgeleitet vom Verb (to) compute (zu Lateinisch: computare = „zusammenrechnen“), bezeich...

Grundlagen

Grundsätzlich unterscheiden sich zwei Bauweisen: Ein Computer ist ein Digitalcomputer, wenn er mit digitalen Geräteeinhe...

Abbildung 46 - Drag'n'Drop Seitenreihenfolge

5.3 Kurseinheit Test

Nun soll hier das Anlegen eines Testes gezeigt werden:

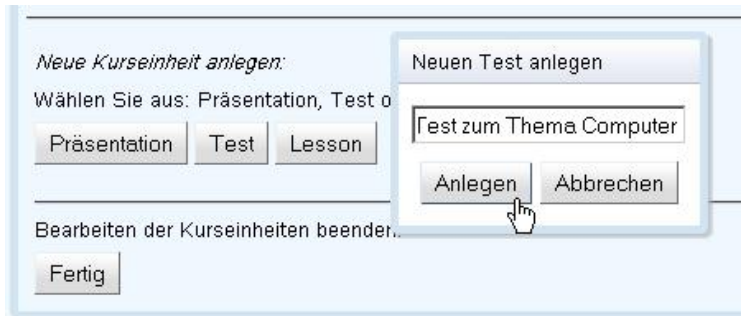


Abbildung 47 - Test erstellen

Wie bei Tests im Allgemeinen üblich, wird bei Bedarf das Festlegen eines Zeitrahmens ermöglicht in welchem der Test von den Studenten absolviert werden kann bzw. muss. In unserem Beispiel haben die Teilnehmer vom 29. Juni 8:30 bis 30. Juni, 8:30 Zeit, den Test zu absolvieren:



Abbildung 48 - Zeitrahmen festlegen

Das Hinzufügen von einzelnen Seiten des Testes erfolgt analog zur Erstellung einer Präsentation:

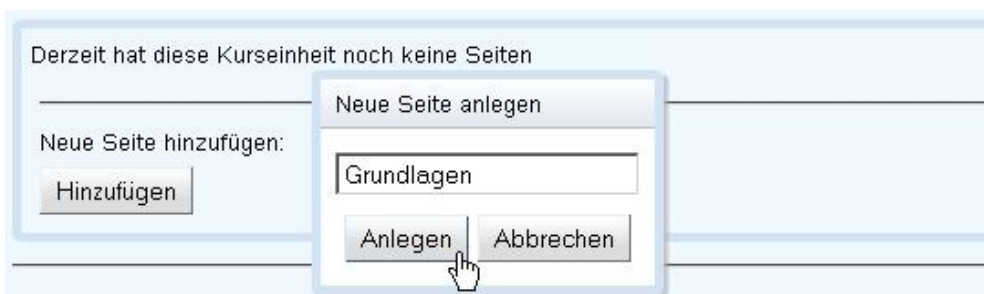


Abbildung 49 - Testseite hinzufügen

5.4 Testfrage Multiple Choice

Nun kann der Seite eine Testfrage hinzugefügt werden. Folgendes Beispiel zeigt das Erstellen einer Multiple Choice Frage:

Editieren Sie hier Ihre Seite:

Titel:
Grundlagen
Bearbeiten

Testfrage hinzufügen:
Multiple Choice Zuordnungen Upload

Abbildung 50 - Multiple Choice Frage erstellen

Titel und Fragetext der Testfrage können anschließend erstellt werden:

Editieren Sie hier Ihre Multiple Choice-Frage:

Titel:
MC Frage zu Grundlagen des Computers
Bearbeiten

Text der Testfrage:

B **I** **U** x_2 x^2

Background Foreground Font Size

Geben Sie an, ob die jeweiligen Sätze wahr oder falsch sind!

Speichern Clear Abbrechen

Abbildung 51 - Multiple Choice Frage Texteditierung

Multiple Choice Fragen bestehen aus einer beliebigen Anzahl von Sätzen mit unterschiedlichem Wahrheitsgehalt. Ein Student muss bei Beantwortung der Frage jeweils durch Anklicken von Radio-Buttons angeben, ob er einen Satz für richtig oder falsch hält. Daher werden bei der Erstellung nun Multiple Choice Sätze hinzugefügt und jeweils deren Wahrheitsgehalt festgelegt:



Abbildung 52 - Multiple Choice Satz hinzufügen

Nachdem alle MC-Sätze angelegt sind, können diese angezeigt bzw. nachträglich bearbeitet werden:

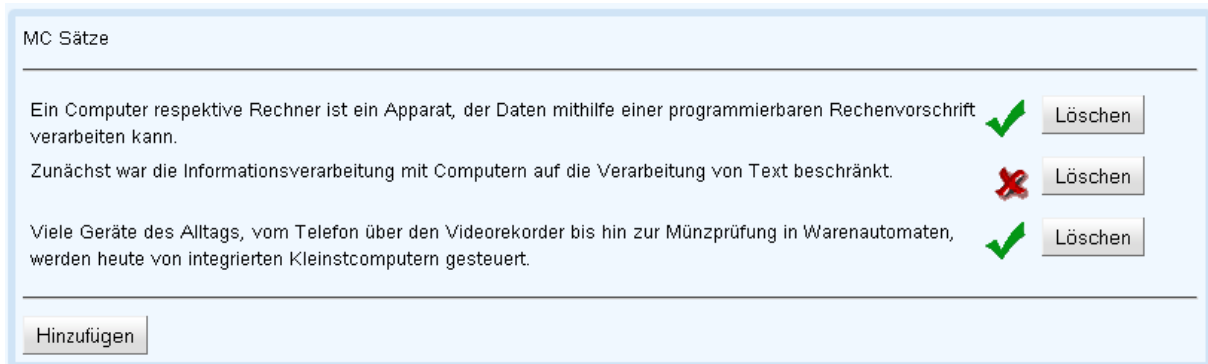


Abbildung 53 - Übersicht Multiple Choice Sätze

5.5 Testfrage Zuordnung

Zuordnungen bestehen aus mehreren Satzpaaren welche beim Absolvieren der Testfrage permutiert dargestellt werden und vom Studenten per Drag'n'Drop passend zugeordnet werden müssen. Nach dem Anlegen der Testfrage inklusive Titel und Fragentext werden mehrere Paare von jeweils zwei zusammengehörigen Sätzen erstellt:

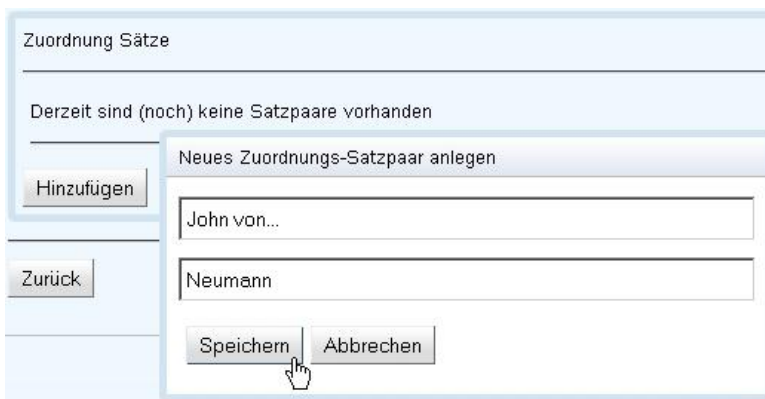


Abbildung 54 - Testfrage Zuordnung erstellen

Diese werden anschließend in einer Liste dargestellt und können nachträglich bearbeitet, bzw. gelöscht werden:

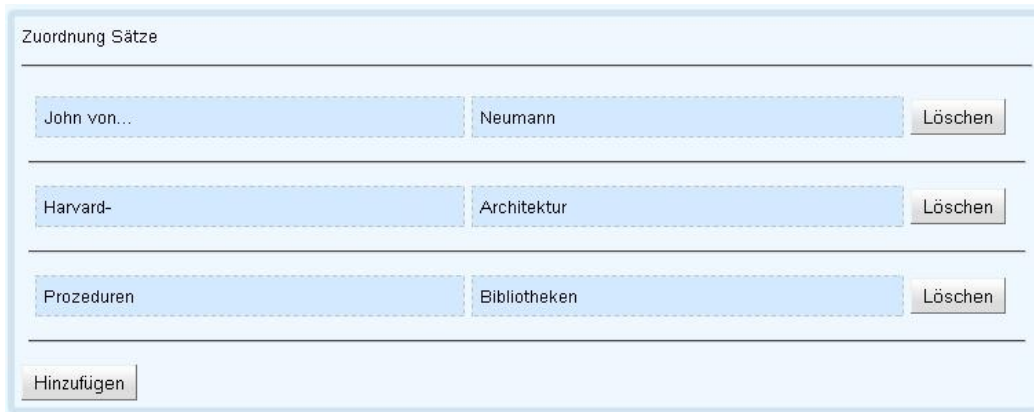


Abbildung 55 - Darstellung Zuordnung Sätze

5.6 Testfrage Abgabe

Bei der Abgabe handelt es sich um eine einfache Aufforderung an den Studenten, eine Datei hochzuladen, damit diese anschließend vom Kursleiter beurteilt werden kann. Das Erstellen einer Abgabe erfolgt durch einfaches Anlegen einer Seite und Hinzufügen einer Testfrage mit dementsprechendem Fragetext. Der Text kann dabei wie gewohnt mittels Richtext-Editors erstellt und bearbeitet werden:

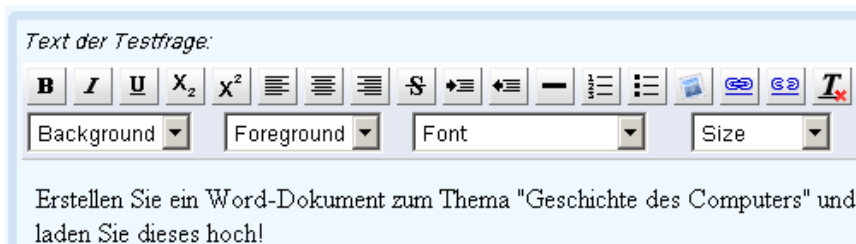


Abbildung 56 - Abgabe Texteditierung

5.7 Kurseinheit Lesson

Als letzte Art von Kurseinheit soll noch kurz das Anlegen einer Lesson gezeigt werden:

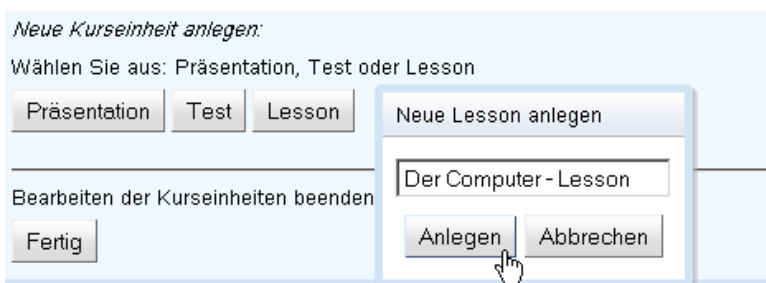


Abbildung 57 - Erstellung Lesson

Analog zu Präsentationen und Tests haben auch Lessons einen Titel, eine Beschreibung und bei Bedarf einen Zeitrahmen. Diese Merkmale können wie gewohnt jederzeit bearbeitet werden. Die Seiten einer Lesson bestehen jeweils aus Lerninhalt, welcher mittels Richtext-Editor grafisch formatiert werden kann und einer Testfrage:

Editieren Sie hier Ihre Seite:

Titel:
Geschichte
Bearbeiten

Lerninhalt der Seite:
Die Computertechnologie entwickelte sich im Vergleich zu anderen Elektrogeräten sehr schnell. Die Geschichte der Entwicklung des Computers reicht zurück bis in die Antike und ist damit wesentlich länger als die Geschichte der modernen Computertechnologien und mechanischen resp. elektrischen Hilfsmitteln (Rechenmaschinen oder [Hardware](#)). Sie umfasst dabei auch die Entwicklung von Rechenmethoden, die etwa für einfache Schreibgeräte auf Papier und Tafeln entwickelt wurden. Im Folgenden wird entsprechend versucht, einen Überblick über diese Entwicklungen zu geben.
Bearbeiten

Multiple Choice Frage


Multiple Choice Frage
Geben Sie jeweils an, ob der Satz wahr ist oder nicht!


Bearbeiten Löschen


Abbildung 58 - Seitenaufbau Lesson


In unserem Beispiel wurde für die Testfrage eine Multiple Choice Frage gewählt:

MC Sätze

Software ist seit jeher Bestandteil von Computern.  Löschen

Die Geschichte des Computers umfasst auch die Entwicklung von Rechenmethoden.  Löschen

Die Geschichte des Computers geht zurück bis in die Antike.  Löschen

"Software" beschreibt alle physischen Komponenten eines Computers.  Löschen

Hinzufügen

Abbildung 59 - Multiple Choice Frage Lesson

Die erstellten Kurseinheiten werden in einer Liste übersichtlich dargestellt. Durch die Symbole „P“, „L“ und „T“ wird die Art der jeweiligen Einheit dargestellt (Präsentation, Lesson, Test). Weiters werden Titel und Kurzbeschreibung angezeigt. Von dieser Liste aus können die Kurseinheiten weiter bearbeitet werden. Zudem besteht die Möglichkeit, Kurseinheiten zu löschen wie Abbildung 60 zeigt:

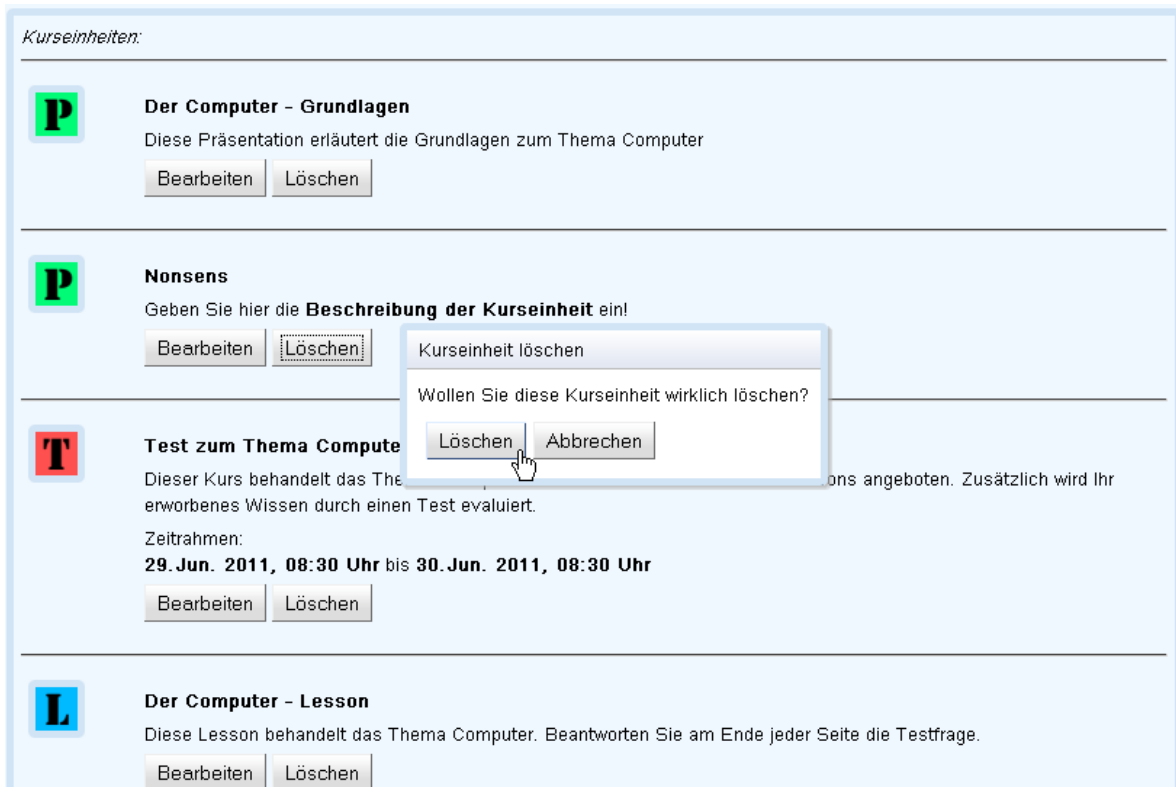


Abbildung 60 - Übersicht Kurseinheiten

5.8 Student – Kursteilnahme

Nun soll anhand des Studenten Max Mustermanns das Teilnehmen an einem Kurs gezeigt werden:

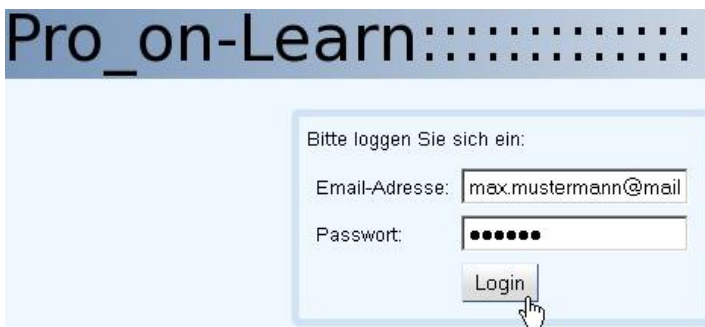


Abbildung 61 - Login Student

Anhand seiner Login-Daten erkennt ihn das System als Teilnehmer und listet auf seiner Startseite alle Kurse auf, in die er inskribiert ist:

Guten Tag Max Mustermann!

Schön, dass Sie wieder hier sind!

Ihre Kurse:

[Didaktik in der Informatik](#)

Dieser Kurs hat den Zweck, Ihnen die Didaktik der Informatik näher zu bringen.

Viel Erfolg dabei,

Ihr Kursleiter

[Heimautomatisierung](#)

Das Thema Heimautomatisierung ist ein Spezialgebiet der Gebäudeautomatisierung und befasst sich im Weitesten mit dem Begriff der 'Systemintegration' im privaten Haushaltsbereich.

[Der Computer](#)

Dieser Kurs behandelt das Thema "Computer". Es werden Präsentation und Lessons angeboten. Zusätzlich wird Ihr erworbenes Wissen durch einen Test evaluiert.

Abbildung 62 - Welcome Page Student

Da unser fiktiver Kursleiter im vorangegangenen Kapitel den Kurs „Der Computer“ erstellt hat, wird Max Mustermann an diesem Kurs teilnehmen und dessen Kurseinheiten absolvieren. Diese werden auf der Kursübersicht-Seite in Listenform angezeigt:

Der Computer

Dieser Kurs behandelt das Thema "Computer". Es werden Präsentation und Lessons angeboten. Zusätzlich wird Ihr erworbenes Wissen durch einen Test evaluiert.

Folgende Kurseinheiten sind verfügbar:



[Der Computer - Grundlagen](#)

Diese Präsentation erläutert die Grundlagen zum Thema Computer



[Test zum Thema Computer](#)

Dieser Kurs behandelt das Thema "Computer". Es werden Präsentation und Lessons angeboten. Zusätzlich wird Ihr erworbenes Wissen durch einen Test evaluiert.

Zeitraumen:

29. Jun. 2011, 08:30 Uhr bis 30. Jun. 2011, 08:30 Uhr



[Der Computer - Lesson](#)

Diese Lesson behandelt das Thema Computer. Beantworten Sie am Ende jeder Seite die Testfrage.

Abbildung 63 - Kursübersicht

Die Präsentation zum Thema Grundlagen des Computers wird gestartet:

Der Computer - Grundlagen

Diese Präsentation erläutert die Grundlagen zum Thema Computer

Starten

Zurück




Abbildung 64 - Start Präsentation

Seite für Seite kann sich der Teilnehmer durch die Präsentation arbeiten:

Einleitung

Ein **Computer** respektive **Rechner** ist ein Apparat, der *Daten* mithilfe einer programmierbaren [Rechenvorschrift verarbeiten](#) kann.



Zunächst war die Informationsverarbeitung mit Computern auf die Verarbeitung von Zahlen beschränkt. Mit zunehmender Leistungsfähigkeit eröffneten sich neue Einsatzbereiche. Computer sind heute in allen Bereichen des täglichen Lebens vorzufinden: Sie dienen der Verarbeitung und Ausgabe von Informationen in Wirtschaft und Behörden, der Berechnung der Statik von Bauwerken bis hin zur [Steuerung](#) von Waschmaschinen und Automobilen. Die leistungsfähigsten Computer werden eingesetzt, um komplexe Vorgänge zu [simulieren](#): Beispiele sind die Klimaforschung, thermodynamische Fragestellungen, medizinische Berechnungen – bis hin zu militärischen Aufgaben, bspw. der Simulation des Einsatzes von nuklearen Waffen. Viele Geräte des Alltags, vom [Telefon](#) über den [Videorekorder](#) bis hin zur Münzprüfung in Warenautomaten, werden heute von integrierten Kleinstcomputern gesteuert ([eingebettetes System](#)).

Weiter >>

Abbildung 65 - Absolvieren Präsentation

Nun will Max Mustermann den angebotenen Test absolvieren um seinen Wissenstand zu überprüfen. Leider muss er feststellen, dass er das gegebene Zeitfenster bereits überschritten hat:

Test zum Thema Computer

Dieser Kurs behandelt das Thema "Computer". Es werden Präsentation und Lessons angeboten. Zusätzlich wird Ihr erworbenes Wissen durch einen Test evaluiert.

Zeitraumen:
29. Jun. 2011, 08:30 Uhr bis 30. Jun. 2011, 08:30 Uhr

Der Zeitraumen zum Absolvieren dieser Kurseinheit ist entweder noch nicht erreicht oder bereits überschritten

Starten Zurück Refresh

Abbildung 66 - Test Zeitraumen überschritten

Nach Intervention beim Kursleiter lässt sich dieser erweichen und erweitert den Zeitraum. Nun kann Max Mustermann den Test absolvieren:

Test zum Thema Computer

Dieser Kurs behandelt das Thema "Computer". Es werden Präsentation und Lessons angeboten. Zusätzlich wird Ihr erworbenes Wissen durch einen Test evaluiert.

Zeitraumen:
29. Jun. 2011, 08:30 Uhr bis 30. Jun. 2011, 08:30 Uhr

Sie haben diesen Test noch nicht absolviert. Jetzt haben Sie die Gelegenheit dazu!

Starten Zurück

Abbildung 67 - Test Absolvieren möglich

Der folgende Screenshot zeigt die Beantwortung der Multiple Choice – Frage, wobei absichtlich eine Antwort falsch gesetzt wird:

Grundlagen

Geben Sie an, ob die jeweiligen Sätze wahr oder falsch sind!

Ein Computer respektive Rechner ist ein Apparat, der Daten mithilfe einer programmierbaren Rechenvorschrift verarbeiten kann. Wahr Falsch

Zunächst war die Informationsverarbeitung mit Computern auf die Verarbeitung von Text beschränkt. Wahr Falsch

Viele Geräte des Alltags, vom Telefon über den Videorekorder bis hin zur Münzprüfung in Warenautomaten, werden heute von integrierten Kleincomputern gesteuert. Wahr Falsch

Absenden

Abbildung 68 - Beantwortung Multiple Choice Frage

Daher erhält der Student auch nur zwei der drei möglichen Punkte:

Proonlearn sagt:

Erreichte Punkte: 2

OK

Abbildung 69 - Punktevergabe Multiple Choice Frage

Nun muss Max Mustermann die Zuordnung richtig lösen, indem er die passenden Begriffe per Drag'n'Drop zusammen fügt wie auf Abbildung 70 zusehen ist.

John von...

Architektur




Abbildung 70 - Drag'n'Drop Zuordnung

Nachdem alle passenden Zuordnungen vorgenommen wurden, kann die Antwort abgesendet werden:

Prozeduren

Bibliotheken



Absenden

Abbildung 71 - Absenden Antwort Zuordnung

Nun soll ein selbst erstelltes Dokument hochgeladen werden. Dabei wird ein Auswahl-Dialog verwendet:

Abgabe

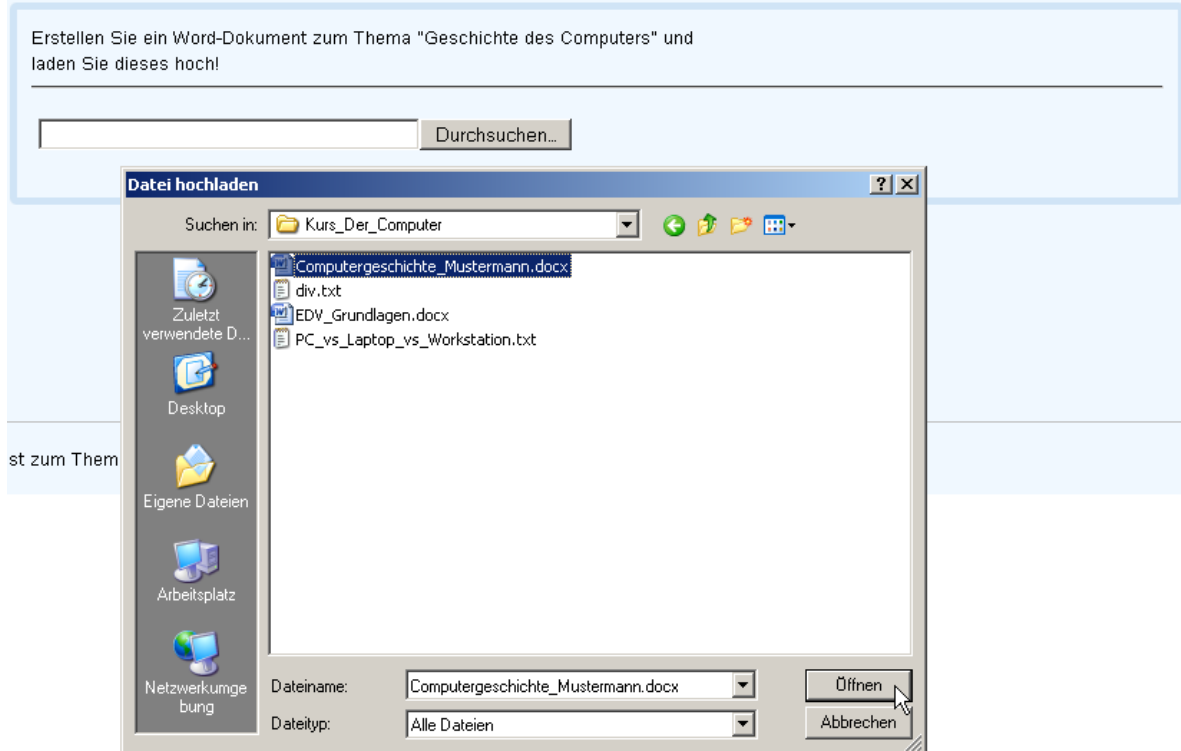


Abbildung 72 - Datei Upload

Nach dem Upload erscheinen zwei wichtige Meldungen. Einerseits wird mittels Dialogfenster der erfolgreiche Upload kommuniziert und außerdem mitgeteilt, dass die Punktevergabe dafür vom Kursleiter erst erfolgen muss. Andererseits wird – da es sich beim Upload in unserem Beispiel um die letzte Testfrage handelte – angezeigt, dass der Test nun absolviert ist und dass bisher eine gewisse Punktezahl erreicht wurde:

Test zum Thema Computer

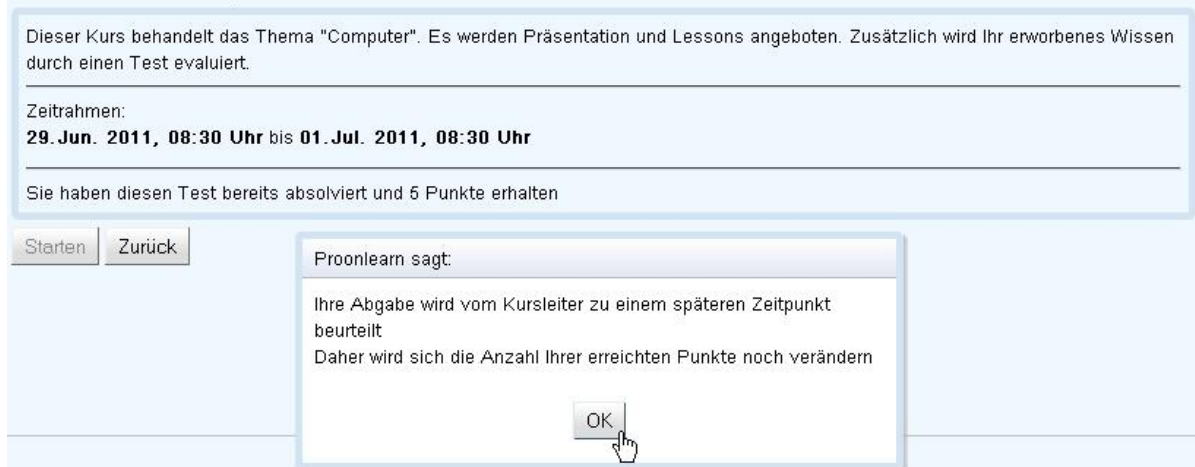


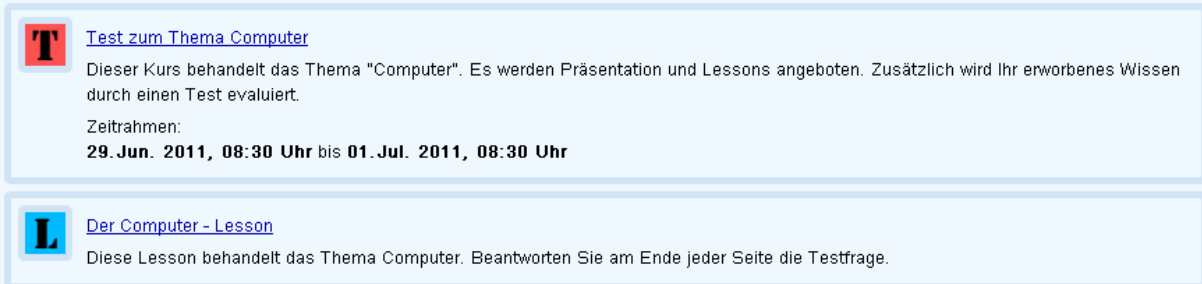
Abbildung 73 - Information nach absolviertem Test

Das Absolvieren einer Lesson erfolgt analog zu Präsentation und Test da es sich hierbei um eine Mischung aus diesen beiden Kurseinheiten handelt.

5.9 Kursleiter – Kursbetreuung

Nach dem Login hat der Kursleiter die Möglichkeit seine laufenden Kurse zu betreuen. Dabei kann er feststellen, ob die Teilnehmer Kurseinheiten absolviert haben und wie viele Punkte sie jeweils erlangt haben. Zudem kann der Kursleiter selbst Punkte vergeben, wie es beispielsweise bei Abgaben notwendig ist.

Folgende Kurseinheiten sind zu betreuen:




The screenshot shows two course units in a list. The first unit is titled 'Test zum Thema Computer' and includes a description, a time frame from June 29 to July 1, 2011, and a red 'T' icon. The second unit is titled 'Der Computer - Lesson' and includes a description and a blue 'L' icon.

T [Test zum Thema Computer](#)
Dieser Kurs behandelt das Thema "Computer". Es werden Präsentation und Lessons angeboten. Zusätzlich wird Ihr erworbenes Wissen durch einen Test evaluiert.
Zeitraumen:
29. Jun. 2011, 08:30 Uhr bis 01. Jul. 2011, 08:30 Uhr

L [Der Computer - Lesson](#)
Diese Lesson behandelt das Thema Computer. Beantworten Sie am Ende jeder Seite die Testfrage.

Abbildung 74 - Betreuung Kurseinheiten

In Listenform wird dargestellt wie die teilnehmenden Studenten bei den jeweiligen Kurseinheiten abgeschnitten haben. In unserem Beispiel ist zudem noch die von den Studenten hochgeladene Abgabe zu beurteilen:



The screenshot displays a list of test takers for the 'Test zum Thema Computer'. Each entry shows the student's name, email address, and score. Below each entry is a link to the submission and a note that some test questions are not yet graded. A 'Zurück' button is located at the bottom left.

Absolventen für Test zum Thema Computer

Max Mustermann
max.mustermann@mail.com
Erreichte Punkte: 5 von 16

Folgende Testfragen sind noch nicht beurteilt:
[Abgabe zum Thema Computer](#)

Martina Musterfrau
martina.musterfrau@mail.com
Erreichte Punkte: 3 von 16

Folgende Testfragen sind noch nicht beurteilt:
[Abgabe zum Thema Computer](#)

Zurück

Abbildung 75 - Testabsolventen

Dabei stellt der Link „Abgabe“ die Datei als Download zur Verfügung. Nach Begutachtung der hochgeladenen Datei kann eine Punktevergabe vorgenommen werden:

Abgabe zum Thema Computer

Erstellen Sie ein Word-Dokument zum Thema "Geschichte des Computers" und laden Sie dieses hoch!

[Abgabe](#)

Vergeben Sie Punkte für diese Testaufgabe:

Abbildung 76 - Punktevergabe

Insgesamt hat sich unser fiktiver Student Mustermann mit 12 von 16 möglichen Punkten als mäßig erfolgreicher Kursteilnehmer erwiesen:

Absolventen für Test zum Thema Computer

Max Mustermann

max.mustermann@mail.com

Erreichte Punkte: 12 von 16

Abbildung 77 - Erreichte Punkte

6 Zusammenfassung

Bei der Erstellung der Webapplikation wurde so vorgegangen, dass nach Festlegen der Features welche eine Lernplattform der Meinung des Autors nach mindestens vorweisen sollte die Recherchephase erfolgte. Dabei wurde die passende Datenbank, verwendete Tools beispielsweise für die vorgestellten Diagramme sowie die DIE ausgewählt. Zudem wurde die Dokumentation des Google Web Toolkits ausgiebig studiert. Danach folgte die Designphase. Dabei wurde von Use-Cases ausgegangen und über Sequenzdiagramme die jeweiligen Klassendiagramme ermittelt welche als Basis für die Implementation dienten. Die entsprechende Implementation erfolgte anschließend. Hier lag der arbeitsintensivste Bereich in der Erstellung des grafischen User-Interfaces welche jedoch durch die Verwendung der in Kapitel 4.3 vorgestellten GWT-Widgets erheblich erleichtert und beschleunigt wurde. Den Abschluss dieser Magisterarbeit stellte die konkrete Schilderung eben dieser praktischen Umsetzung sowie das Erstellen eines Benutzerhandbuches in Kapitel 5 dar.

Als Fazit dieser Arbeit lässt sich festhalten, dass das GWT Webentwicklern ein Werkzeug in die Hand gibt, welches einerseits Effizienz und Flexibilität durch die einerseits angebotenen aber auch erweiterbaren grafischen Elemente und andererseits durch Technologien wie beispielsweise die eigenen asynchronen Methodenaufrufe bietet. Andererseits wird durch die Verwendung von Java als Basissprache Objektorientierung auf professionelle Weise geboten wodurch Stabilität und modulares Vorgehen ermöglicht wird.

7 Abbildungsverzeichnis

Abbildung 1 - Use-Case Kurs erstellen	9
Abbildung 2 - Use-Case Lesson erstellen.....	10
Abbildung 3 - Use-Case Lesson absolvieren	11
Abbildung 4 - Sequenzdiagramm Kurs erstellen	12
Abbildung 5 - Klassendiagramm Kurs erstellen	13
Abbildung 6 - Sequenzdiagramm Lesson erstellen.....	14
Abbildung 7 - Klassendiagramm Lesson erstellen	15
Abbildung 8 - Sequenzdiagramm Lesson absolvieren	16
Abbildung 9 - Klassendiagramm Lesson absolvieren	17
Abbildung 10 - Sitemap Studenten	18
Abbildung 11 - Sitemap Kursleiter	19
Abbildung 12 - Sitemap Administratoren.....	20
Abbildung 13 - ER Diagramm Kurs.....	20
Abbildung 14 - ER Diagramm Testfragen	21
Abbildung 15 - ER Diagramm Forum.....	21
Abbildung 16 - ER Diagramm Gesamtdesign	22
Abbildung 17 - Architektur	23
Abbildung 18 - Klassendiagramm Personen.....	24
Abbildung 19 - Klassendiagramm Kurs.....	25
Abbildung 20 - Klassendiagramm Testfragen	27
Abbildung 21 - Klassendiagramm Forum.....	28
Abbildung 22 - Klassendiagramm Datenschicht.....	29
Abbildung 23 - Klassendiagramm Präsentationsschicht	30
Abbildung 24 - Klassendiagramm Content Student	30
Abbildung 25 - Klassendiagramm Content Kursleiter.....	31
Abbildung 26 - Projektstruktur	32
Abbildung 27 - Moduldefinition	32
Abbildung 28 - Paketstruktur	33
Abbildung 29 - Interface KursView.....	34
Abbildung 30 - Implementation KursImp.....	35
Abbildung 31 - Interface KursViewAsync.....	35
Abbildung 32 - Methode AsyncCallback<>().....	36
Abbildung 33 - Einbindung der grafischen Darstellung	36
Abbildung 34 - Methode RootPanel.get().....	36
Abbildung 35 - ContentWidget.....	37
Abbildung 36 - ContentWidget Kursleiter.....	37
Abbildung 37 - Login Kursleiter.....	41
Abbildung 38 - Kursleiter Welcome Page	41
Abbildung 39 - Edit Kurs Übersicht.....	42
Abbildung 40 - Teilnehmerverwaltung	42
Abbildung 41 - Präsentation erstellen	43
Abbildung 42 - Präsentation editieren.....	43
Abbildung 43 - Seite hinzufügen.....	44
Abbildung 44 - Seite formatieren	44
Abbildung 45 - Seitenliste.....	45
Abbildung 46 - Drag'n'Drop Seitenreihenfolge	45

Abbildung 47 - Test erstellen	46
Abbildung 48 - Zeitrahmen festlegen	46
Abbildung 49 - Testseite hinzufügen.....	46
Abbildung 50 - Multiple Choice Frage erstellen.....	47
Abbildung 51 - Multiple Choice Frage Texteditierung	47
Abbildung 52 - Multiple Choice Satz hinzufügen.....	48
Abbildung 53 - Übersicht Multiple Choice Sätze	48
Abbildung 54 - Testfrage Zuordnung erstellen	48
Abbildung 55 - Darstellung Zuordnung Sätze	49
Abbildung 56 - Abgabe Texteditierung.....	49
Abbildung 57 - Erstellung Lesson	49
Abbildung 58 - Seitenaufbau Lesson	50
Abbildung 59 - Multiple Choice Frage Lesson	50
Abbildung 60 - Übersicht Kurseinheiten.....	51
Abbildung 61 - Login Student	51
Abbildung 62 - Welcome Page Student	52
Abbildung 63 - Kursübersicht.....	52
Abbildung 64 - Start Präsentation	52
Abbildung 65 - Absolvieren Präsentation.....	53
Abbildung 66 - Test Zeitrahmen überschritten	53
Abbildung 67 - Test Absolvieren möglich.....	53
Abbildung 68 - Beantwortung Multiple Choice Frage	54
Abbildung 69 - Punktevergabe Multiple Choice Frage	54
Abbildung 70 - Drag'n'Drop Zuordnung	54
Abbildung 71 - Absenden Antwort Zuordnung	54
Abbildung 72 - Datei Upload.....	55
Abbildung 73 - Information nach absolviertem Test	55
Abbildung 74 - Betreuung Kurseinheiten	56
Abbildung 75 - Testabsolventen	56
Abbildung 76 - Punktevergabe	57
Abbildung 77 - Erreichte Punkte	57

8 Tabellenverzeichnis

Tabelle 1 - Use Case Kurs erstellen	9
Tabelle 2 - Use-Case Lesson erstellen.....	10
Tabelle 3 - Use-Case Lesson absolvieren	11

9 Literaturverzeichnis

- [Arlow02] Arlow, Jim : UML and the unified process. - London [u.a.] : Addison-Wesley, 2002
- [CustGUI] GWT – Creating Custom Widgets [online]. Verfügbar bei: <http://code.google.com/intl/de/webtoolkit/doc/1.6/DevGuideUserInterface.html#DevGuideCreatingCustomWidgets> [Zugang 25.5.2011]
- [Eclipse] Eclipse [online]. Verfügbar bei <http://www.eclipse.org/> [Zugang 7.3.2009]
- [GPE] GPE – Google Plugin for Eclipse [online]. Verfügbar bei: <http://code.google.com/intl/de/eclipse/> [Zugang 12.4.2009]
- [GWT] Google Web Toolkit - Productivity for developers, performance for users [online]. Verfügbar bei <http://code.google.com/intl/de-DE/webtoolkit/> [Zugang 19.3.2009]
- [GwtAPI] Google API Libraries for GWT [online]. Verfügbar bei <http://code.google.com/intl/de-DE/webtoolkit/googleapilibraries.html> [Zugang 15.4.2009]
- [GwtDir] GWT – Standard Directory and Package Layout [online]. Verfügbar bei: <http://code.google.com/intl/de/webtoolkit/doc/1.6/DevGuideOrganizingProjects.html#DevGuideDirectoriesPackageConventions> [Zugang 2.6.2011]
- [GwtGUI] GWT – Building User Interfaces [online]. Verfügbar bei: <http://code.google.com/intl/de/webtoolkit/doc/1.6/DevGuideUserInterface.html> [Zugang 8.6.2011]
- [GwtHistory] GWT – History [online]. Verfügbar bei: <http://code.google.com/intl/de/webtoolkit/doc/1.6/DevGuideCodingBasics.html#DevGuideHistory> [Zugang 18.5.2011]
- [GwtHost] GWT – HTML Host Pages [online]. Verfügbar bei: <http://code.google.com/intl/de/webtoolkit/doc/1.6/DevGuideOrganizingProjects.html#DevGuideHostPage> [Zugang 3.6.2011]
- [GwtNewsGr] Google Groups – Google Web Toolkit [online]. Verfügbar bei <http://groups.google.com/group/Google-Web-Toolkit> [Zugang 5.2.2009]
- [GwtRPC] GWT – Remote Procedure Calls [online]. Verfügbar bei: <http://code.google.com/intl/de/webtoolkit/doc/1.6/DevGuideServerCommunication.html#DevGuideRemoteProcedureCalls> [Zugang 19.5.2011]
- [GwtXML] GWT – Defining a module: format of module XML files [online]. Verfügbar bei: <http://code.google.com/intl/de/webtoolkit/doc/1.6/DevGuideOrganizingProjects.html#DevGuideModuleXml> [Zugang 23.5.2011]
- [JavaAPI] Java 2 Platform Standard Edition 5.0 API Specification [online].

Verfügbar bei <http://java.sun.com/j2se/1.5.0/docs/api/> [Zugang 5.4.2009]

- [JavaEEAPI] Java 2 Platform Enterprise Edition, v 5.0 API Specifications [online]. Verfügbar bei <http://download.oracle.com/javaee/5/api/> [Zugang 5.4.2009]
- [Judmaier05] Judmaier, Peter: Konzept und Umsetzung eines eLearning-Kurses für die berufsbegleitende Erwachsenenbildung. – Wien : TU Dissertation 2005
- [Kemper04] Kemper, Alfons : Datenbanksysteme. - München ; Wien : Oldenbourg, 2004
- [Masak10] Masak, Dieter : Der Architekturreview. - Berlin [u.a.] : Springer, 2010
- [MySQL] MySQL.com [online]. Verfügbar bei <http://mysql.com/> [Zugang 30.3.2009]
- [Oester09] Oestereich, Bernd : Analyse und Design mit UML 2.3. - München : Oldenbourg, 2009
- [SauerDND] gwt-dnd – Drang-and-Drop library for Google Web Toolkit [online]. Verfügbar bei: <http://code.google.com/p/gwt-dnd/> [Zugang 1.7.2011]
- [VPUML] Visual Paradigm for UML [online]. Verfügbar bei <http://www.visual-paradigm.com/product/vpuml/> [Zugang 5.2.2009]
- [WidgetGal] GWT – Widget Gallery [online]. Verfügbar bei: <http://code.google.com/intl/de/webtoolkit/doc/1.6/RefWidgetGallery.html> [Zugang 22.5.2011]