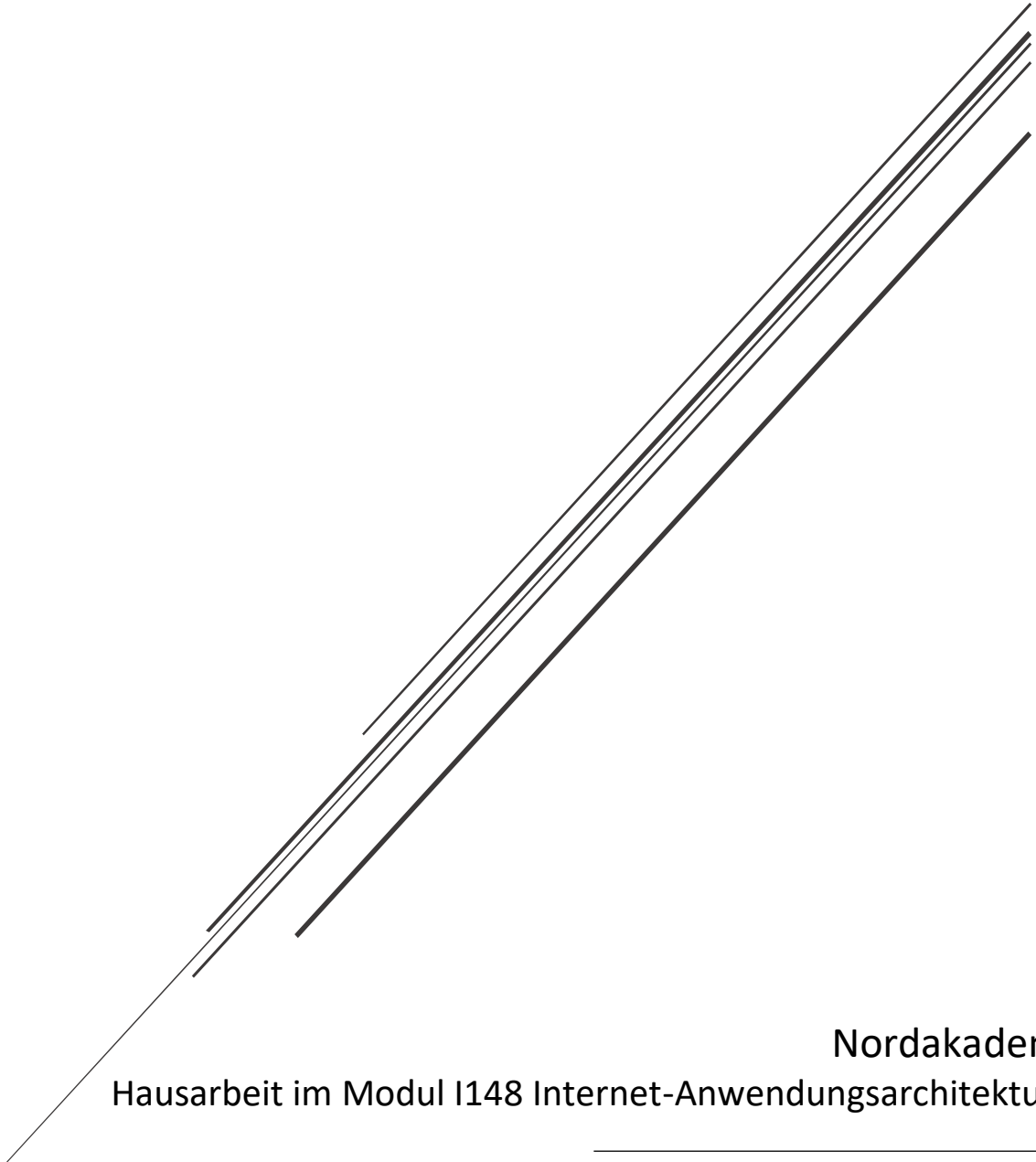


PFLICHTENHEFT

Multiple Choice Tests



Nordakademie
Hausarbeit im Modul I148 Internet-Anwendungsarchitekturen

Ferenc Beutel
Max Hort
Hendrik Peters
Melanie Beckmann

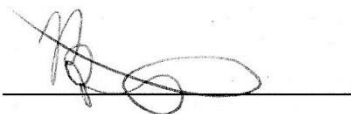
Inhaltsverzeichnis

Erklärung der Eigenleistung.....	II
Installations- und Nutzungsanleitung (Readme)	III
DV-Konzept	- 1 -
1. Einleitung	- 1 -
1.1. Zielsetzung.....	- 1 -
1.2. Produktziele.....	- 1 -
2. Produktübersicht.....	- 1 -
2.1. Produkt-Einsatz	- 1 -
2.2. Benutzer-Eigenschaften.....	- 1 -
2.3. Produkt-Umgebung	- 2 -
2.4. Produkt-Funktionen.....	- 2 -
2.5. Vereinbarte Erweiterungen und Annahmen.....	- 4 -
2.6. Produkt-Daten	- 5 -
3. Testfälle	- 6 -
4. Softwarearchitektur	- 9 -
4.1. Softwarearchitektur	- 9 -
4.2. Klassenmodell.....	- 9 -
4.3. ER-Modell	- 10 -
4.4. Dialogmodell	- 11 -
4.4.1. Sicht des Dozenten	- 12 -
4.4.2. Sicht des Studenten	- 12 -
4.5. Entwurf des Usability Design	- 13 -
Hilfsmittel und Quellenverzeichnis.....	IV
Anhang	V
4.6. Anhang 1: Entwurfszeichnungen	V
4.7. Anhang 2: Layout.....	V


Erklärung der Eigenleistung

Hiermit versichern wir, dass diese Hausarbeit im Modul I148 Internetanwendungsarchitekturen von uns selbstständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt wurde. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungskommission vorgelegen.

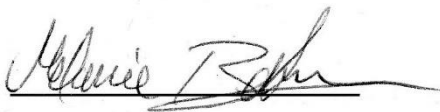
Die anteilige Leistung und Zuordnung der Aktivitäten der vier Gruppenmitglieder kann durch die in jeder Klasse angegebenen Autoren nachvollzogen werden. Die Arbeit fand größtenteils in Pair Programming in der Uni statt. Außerdem kann im Tasklog und Gitlog nachvollzogen werden, wie die Aufgaben verteilt waren.



Ferenc Beutel



Max Hort



Melanie Beckmann



Hendrik Peters

Installations- und Nutzungsanleitung (Readme)

Die Nutzungsanleitung für unser Projekt findet sich in der Readme Datei, die direkt im Github zu finden ist.

Generell können über unsere Java Anwendung Testdaten generiert werden. Hierzu wird die Seite */application-administration* geöffnet. Es ist auswählbar, wie viele Studenten, Dozenten und Seminare initial angelegt werden sollen. Ein Login ist hierzu noch nicht erforderlich.

Außerdem wird von hier über den Button das Test-Token an die Studenten verschickt.

DV-Konzept

1. Einleitung

1.1. Zielsetzung

Innerhalb des Projektes „Multiple Choice Tests“ soll eine Möglichkeit geschaffen werden, das Wissen der Studenten in einem internetgestützten Prüfungssystem abzufragen und entsprechend nach ECTS zu kreditieren. Der Prozess soll damit für Studenten und Dozenten vereinfacht werden.

1.2. Produktziele

Mit dem Prüfungssystem für die „Multiple Choice Tests“ wird es den Dozenten ermöglicht, zu dem in ihrem Seminar vermittelten Wissen einen Test mit entsprechenden Fragen und Antworten zu erstellen. Die Studenten können diesen Test zur Erlangung ihrer Credit Points im internetgestützten Prüfungssystem ablegen. Dafür wird Ihnen per E-Mail ein Einmal-Passwort zugeschickt, wodurch eine Authentifizierung sichergestellt wird.

Der Student hat die Möglichkeit, Seminare aus einer Liste zu wählen und sich einzuschreiben. Zusätzlich erhält der Student auch eine Übersicht zu seinen abzulegenden Prüfungen zu den verschiedenen Seminaren, an denen er teilgenommen hat. Sobald ein Test abgeschlossen ist, wird dem Studenten auf der Home-Seite angezeigt, ob er bestanden (grün) oder nicht bestanden (rot) hat.

Außerdem bietet das Prüfungssystem für den Dozenten eine Übersicht über die angelegten Tests für alle Seminare, die er hält.

2. Produktübersicht

2.1. Produkt-Einsatz

Das Produkt wird speziell zur Anwendung für die Nordakademie zur Verfügung gestellt. Die Zielgruppe besteht aus den Dozenten, die Seminare leiten und Tests erstellen, und den Studenten, die die Prüfungen ablegen sollen.

2.2. Benutzer-Eigenschaften

Die Nutzung erfolgt über eine Webapplikation. Das Prüfungssystem ist als Webdienst erreichbar und verhindert unbefugten Zugang durch eine vorgeschaltete Registrierung der Studenten und einen notwendigen, User spezifischen Login.

Die Dozenten können nur über die /application-administration Seite von dem System-Administrator angelegt werden, um Missbrauch zu verhindern.

2.3. Produkt-Umgebung

Das Prüfungssystem wird in Java und JavaScript realisiert. Die Technische Produktumgebung besteht aus den folgenden Komponenten:

- Struts2 für das Frontend
- Nutzung von Freemarker, Sass und Bootstrap
- Hibernate als Datenbank Technologie
- Spring
- Javamail

Die Verwendung des Spring Frameworks ermöglicht die problemlose Verbindung zwischen den Komponenten. Details gehen aus der beigefügten Softwarearchitektur hervor.

2.4. Produkt-Funktionen

/F10/ Geschäftsprozess: Test anlegen

Akteur: Dozent

Beschreibung: Der Dozent wählt ein Seminar aus, zu dem er einen Test anlegen möchte. Nun gibt er allgemeine Daten zum Test an: in welchem Zeitraum findet der Test statt, wie viele Credit Points gibt der Test, welche Prozentzahl ist zum Bestehen nötig, wie viel Zeit ist zur Durchführung gegeben, wie werden Falsche oder fehlende Antworten bewertet.

Ergebnis: Ein Testobjekt, zu dem im nächsten Schritt Fragen und Antworten angelegt werden.

Fehler: Der Dozent hat kein Seminar, dass er zurzeit gibt.

/F20/ Geschäftsprozess: Fragen und Antworten anlegen

Akteur: Dozent

Beschreibung: zu dem eben erstellten Test können nun Fragen hinzugefügt werden. Hierzu muss der „Typ“ (einfache Auswahlmöglichkeit, mehrfache Auswahlmöglichkeit, Lückentext) ausgewählt werden. Nun können die Fragen eingegeben werden und mit den zugehörigen Antworten direkt gespeichert werden. Die Antworten müssen für die Fragen direkt mit einer Aussage über die Richtigkeit angelegt werden. Bei dem Lückentext werden Antworten automatisch als Richtig markiert.

Ergebnis: Eine Frage mit zugehörigen Antworten.

Fehler: es existiert noch kein Test

/F30/ Geschäftsprozess: Anmelden am System

Akteur: Dozent oder Student, also User

Beschreibung: Der User sieht die Anmeldemaske vor sich und gibt zur Identifizierung seine Email und sein Passwort ein.

Ergebnis: Anzeigen der Startseite zur Bestätigung der erfolgreichen Anmeldung des Nutzers am System

Fehler: Der User ist bereits angemeldet oder noch nicht registriert.

/F40/ Geschäftsprozess: Teilnehmer zur Prüfung festlegen

Akteur: System, indirekt Student

Beschreibung: Dadurch, dass sich der Student zu angebotenen Seminaren einschreibt, wird er vom System automatisch zur Testteilnahme hinzugefügt, sobald der Dozent den Test zum Seminar zur Verfügung stellt und der Testzeitraum startet.

Ergebnis: Die Studenten erhalten eine E-Mail mit einem Einmal-Passwort zur Teilnahme am Test. Der Button zur Testteilnahme auf der Home-Seite des Studenten ist aktiv.

/F50/ Geschäftsprozess: Test anzeigen lassen

Akteur: Student

Beschreibung: Der Student sieht eine Liste aller Prüfungen, die er abzulegen hat in Abhängigkeit der Seminare, die er belegt hat. Außerdem kann er sehen, ob bereits abgelegte Tests bestanden oder nicht bestanden sind.

Ergebnis: Eine Liste der Tests zugeordnet zu den Seminaren

Fehler: Der Student hat weder abgelegte Tests noch ist er zur Teilnahme an einem Test registriert.

/F60/ Geschäftsprozess: Test ablegen

Akteur: Student

Beschreibung: Der Test wird durch ein Einmal-Passwort freigeschaltet. Nun kann der Student alle Fragen durch Ankreuzen oder Befüllen der Lücken beantworten. Er kann Fragen zurückstellen und sich am Ende noch einmal alle Fragen ansehen und ggf. Korrekturen vornehmen. Der Test ist auf eine Bearbeitungszeit beschränkt und wird automatisch abgeschickt sobald die Zeit abgelaufen ist.

Ergebnis: der ausgefüllte Test wird gespeichert und abgeschickt und der Student erhält sein Ergebnis.

/F70/ Geschäftsprozess: Test bearbeiten

Akteur: Dozent

Beschreibung: Der Dozent kann seine Tests mit Fragen und Antworten beliebig oft bearbeiten. Hierzu öffnet er den angelegten Test und kann Testinformationen oder bereits angelegte Fragen und Antworten editieren oder auch falsche Fragen löschen.

Ergebnis: Ein aktualisiertes Testobjekt.

Fehler: Der Test Bearbeitungszeitraum hat schon begonnen.

2.5. Vereinbarte Erweiterungen und Annahmen

Die ursprünglichen Anforderungen von dem Auftraggeber wurden in Rücksprache um folgende Dinge ergänzt, um einen angenehmen Fluss durch die Anwendung und eine möglichst produktive und langfristige Nutzung zu ermöglichen:

/F80/ Geschäftsprozess: Seminar anlegen

Akteur: Dozent

Beschreibung: Der Dozent legt seine Seminare, die er an der Nordakademie hält eigenständig an. Hierzu gibt er in dem dafür vorgesehenen Formular Namen, Beschreibung, Start- und Enddatum sowie maximale Teilnehmerzahl an. Dies ist Voraussetzung für den Geschäftsprozess /F10/.

Ergebnis: Ein Seminarobjekt, welches auf der Startseite des Dozenten erscheint und für die Studenten in der Seminarliste auswählbar ist.

/F90/ Geschäftsprozess: Seminar bearbeiten

Akteur: Dozent

Beschreibung: Der Dozent kann seine Seminare bearbeiten. Hierzu öffnet er das angelegte Seminar und kann alle Informationen bearbeiten.

Ergebnis: Ein Seminarobjekt, mit aktualisierten Informationen.

Fehler: Das Seminar hat schon begonnen.

/F100/ Geschäftsprozess: In ein Seminar einschreiben

Akteur: Student

Beschreibung: Der Student wählt in der Seminarliste ein Seminar aus. Auf der Detailseite des Seminars schreibt er sich durch das Betätigen des Buttons „einschreiben“ für das Seminar ein.

Ergebnis: Der Student wird als Teilnehmer des Seminars gespeichert und bekommt dieses auf seiner Startseite angezeigt.

Fehler: Das Seminar ist voll, das Seminar hat schon begonnen oder der Student ist bereits eingeschrieben.

F/110/ Geschäftsprozess: Test Ergebnis anzeigen

Akteur: System, indirekt Student

Beschreibung: Das System berechnet automatisch die Testauswertung.

Ergebnis: Der Student bekommt nach dem abschicken des Tests direkt die erreichte Punktzahl angezeigt. Später werden dem Studenten seine Ergebnisse auf der Home Seite angezeigt. Durch grüne oder rote Hinterlegung wird angezeigt ob der Test zu dem Seminar bestanden oder nicht bestanden wurde. Auch die Punktzahl ist hier festgehalten.

2.6. Produkt-Daten

/D10/

Es kann maximal 1 Test pro Seminar angelegt werden.

/D20/

Es können nur User mit einer Nordakademie E-Mail-Adresse registriert werden

/D30/

Ein Seminar beinhaltet immer genau einen Dozenten.

/D40/

Fragen können nur angelegt werden, wenn schon ein Test existiert.

/D50/

Es können sich nicht mehr Studenten einschreiben als die max. Teilnehmerzahl des Seminars.

/D60/

Ein Student kann einen Test zum Seminar nur genau 1x ablegen.

/D70/

Ein Seminar kann nur bearbeitet werden, wenn heute < Startdatum.

/D80/

Ein Test kann nicht vor dem Startdatum des Seminars starten.

/D90/

Ein Test und seine Fragen können nur bearbeitet werden, wenn Heute < Startdatum des Tests.

/D100/

Ein Student kann sich nicht mehrfach zu einem Seminar einschreiben.

3. Testfälle

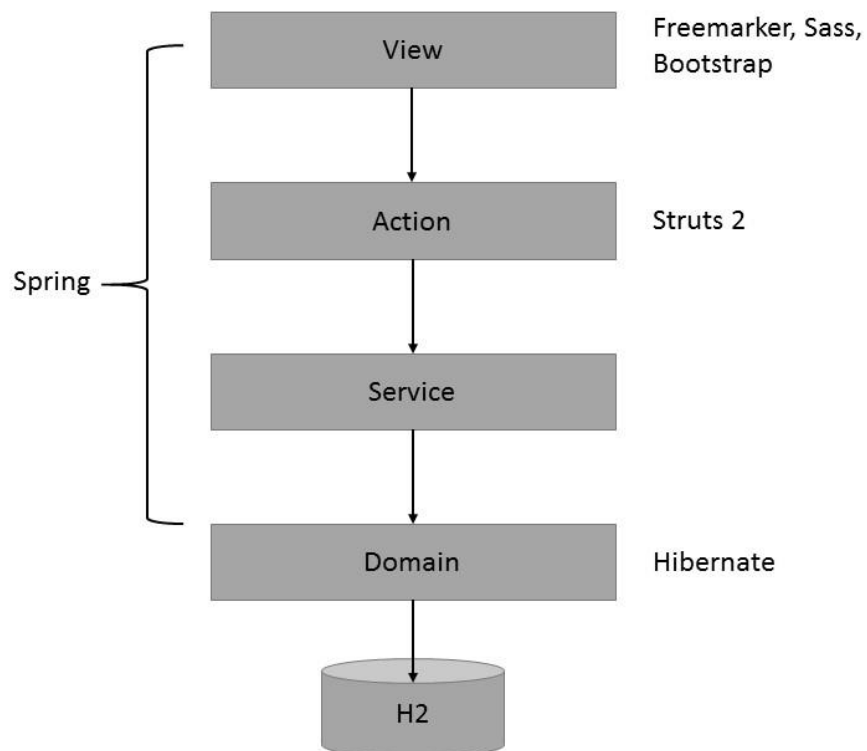
Testfälle				
Nr	Name	Vorbedingung	Aktionen	Ergebnis
1	Registrierung Registrierung Teil 2	-Student an der Nordakademie mit E-Mail-Adresse	1. Öffnen der Seite /home. 2. Auswählen „Registrierung“ in der Navigationszeile 3. Ausfüllen des Formulars auf der Seite /registration-form 4. Abschicken des Formulars. 5. Email öffnen 6. Link in Email klicken zur Bestätigung der Registrierung	Eine E-Mail zur Bestätigung bzw. zum Abschluss der Registrierung Der Link leitet auf die Seite registration-step2-complete weiter.
2	Anmeldung	-Student an der Nordakademie -Registrierung als User	1. Öffnen der Seite /login 2. Eingabe Email und Passwort zur Identifizierung 3. Betätigen des Anmeldebuttons	Begrüßung des Nutzers auf der Startseite /home, die Navigationsleiste zeigt nun die User abhängigen Links an (Seminar anlegen bzw. Seminar auswählen Logout statt Login und Registrierung)
3	Seminar anlegen	-Als Dozent angemeldet	1. anmelden am System (/login) 2. In der Navigationsleiste Seminar anlegen wählen (/add-seminar-form) 3. Seminarformular ausfüllen. 4. Hinzufügen drücken	Das Seminar wird nun auf der Home Seite (/home) in der Tabelle angezeigt. Hier ist ein Button, um einen Test zum Seminar hinzuzufügen.
4	Seminar bearbeiten	-Als Dozent angemeldet -Seminarobjekt existiert	1. Seite /home öffnen 2. In der Tabelle auf der /home das zu bearbeitende Seminar anklicken 3. Weiterleitung zur Bearbeitungsseite /edit-seminar-form 4. gewünschte Daten im Formular ändern. 5. Speichern	Das Seminar wird mit den Daten gespeichert bzw. aktualisiert. Der Nutzer wird zurück auf die Home Seite (/home) geleitet.
5	Test einstellen	-Seminar existiert -Als Dozent angemeldet sein	1. /home öffnen 2. Seminar wählen und Test anlegen auswählen 3. auf Seite /add-test-form grundlegende Infos zum Test festlegen (Dauer,	Angelegter Test, zu dem Fragen angelegt werden können. Der Test ist in der Datenbank persistiert. Der Nutzer kann nun den Test des Seminars bearbeiten und er wird auf /home in der

			Zeit, Kreditpunkte, Bestehensgrenze & Bewertung 4. Test speichern	Tabelle angezeigt als Verknüpfung.
6	Test bearbeiten	-Seminar existiert -Als Dozent angemeldet sein	1./home öffnen 2.Seminar wählen und Test bearbeiten auswählen 3. auf Seite /edit-test- form gewünschte Infos zum Test ändern 4. Test speichern	Aktualisierte Test Infos werden gespeichert am Testobjekt. Der Nutzer wird auf die /home Seite geleitet
7	Fragen anlegen	-Seminar und Test existiert -Test hat noch nicht begonnen -Als Dozent angemeldet sein	1./home öffnen und in der Tabelle Test bearbeiten (/edit-test- form) auswählen 2. hier den Button Frage Hinzufügen wählen 3.Das Formular(/add- question-form) ausfüllen. (Fragetyp wählen und Anweisungen folgen) 4.Antworten festlegen 5. Frage speichern	Eine Frage, die zum Test gehört. Die Frage ist in der Datenbank mit den entsprechenden Antworten persistiert. Der Nutzer wird zurück auf die /edit-test-form Seite geleitet und sieht in der Tabelle die soeben erstellte Frage. Diese kann über einen Button gelöscht oder bearbeitet werden.
8	Frage bearbeiten	-Frage existiert -Test Zeitraum hat noch nicht begonnen -Als Dozent angemeldet sein	1./home öffnen und in der Tabelle Test bearbeiten auswählen (/edit-test-form). 2. über Tabelle der Fragen zur Frage navigieren 3. über den Button die Frage öffnen (/edit-question-form) 4. Daten ändern 5. Speichern	Die Frage ist gespeichert und wird in der Tabelle auf der Test bearbeiten Seite (/edit- test-form) aktualisiert angezeigt.
9	Frage löschen	-Frage existiert -Test Zeitraum hat noch nicht begonnen -Als Dozent angemeldet sein	1./edit-test-form öffnen 2. In der Tabelle der Fragen den „-“ Button neben der zu löschenden Frage drücken	Die Seite wird aktualisiert und die Frage verschwindet.
10	Zum Seminar einschreiben	-Als Student angemeldet sein	1.Auf der /home Seite im Navigationsmenü Seminar auswählen anklicken 2. auf /seminar-list In der Liste Seminar suchen und anklicken 3. Auf der Seminar Detail (/seminar- detail) Seite den Button „einschreiben“ wählen	Der Student wird zurück auf die Home Seite (/home) geleitet. Hier sieht er in der Tabelle nun, dass er zum Seminar eingeschrieben ist, da das Seminar in seiner Tabelle angezeigt wird.

11	Prüfungsstart anzeigen	-Als Student angemeldet sein und in mind. Ein Seminar eingeschrieben -Prüfungs-/ Test-Zeitraum hat begonnen	1. auf Startseite navigieren (/home) 2. prüfen, ob eine E-Mail mit einem Zulassungs-Token erhalten wurde	Der Student sieht, dass der Starte Test Button in der Tabelle nun aktiv ist.
12	Prüfung ablegen	-Als Student angemeldet -Test und Fragen existieren -Startdatum des Tests <= heute -Enddatum >= heute	1./home Seite öffnen 2.Test starten Button drücken 2.auf /start-test Token (Email) zur Freischaltung eingeben und submit drücken 3./perform-test mit erster Frage öffnet sich 4.verbleibende Durchführungszeit wird über der Frage angezeigt. 3.Fragen beantworten indem Checkbox ausgewählt wird bzw. Lücke gefüllt wird. 4. Weiter zur nächsten Frage über Pfeilbutton 5. Test abschicken über Button	Der Test ist abgeschickt und der Student bekommt auf /submit-test.action direkt seine erreichte Punktzahl angezeigt. Auf der /home Seite ist das Seminar je nach Ergebnis grün(bestanden) oder rot (durchgefallen)hinterlegt und die erreichte Punktzahl wird gezeigt.
13	Frage zurückstellen	-Als Student angemeldet -Test starten	1./home Seite öffnen 2.Test starten Button drücken 2.auf /start-test Token (Email) zur Freischaltung eingeben und submit drücken 3./perform-test mit erster Frage öffnet sich 4. „postpone question“ Button drücken. 5. Weiter mit der nächsten Frage	Die Frage erscheint am Ende des Testes erneut und kann beantwortet werden.
14	Abmeldung	-Login	1.In der Navigationsleiste den Link „Logout“ drücken	Der User wird auf die Logout-Seite /logout geleitet und sein Logout wird bestätigt. Die User bezogenen Links in der Navigationsleiste ändern sich. (Home, Login und Registrierung ist jetzt da)

4. Softwarearchitektur

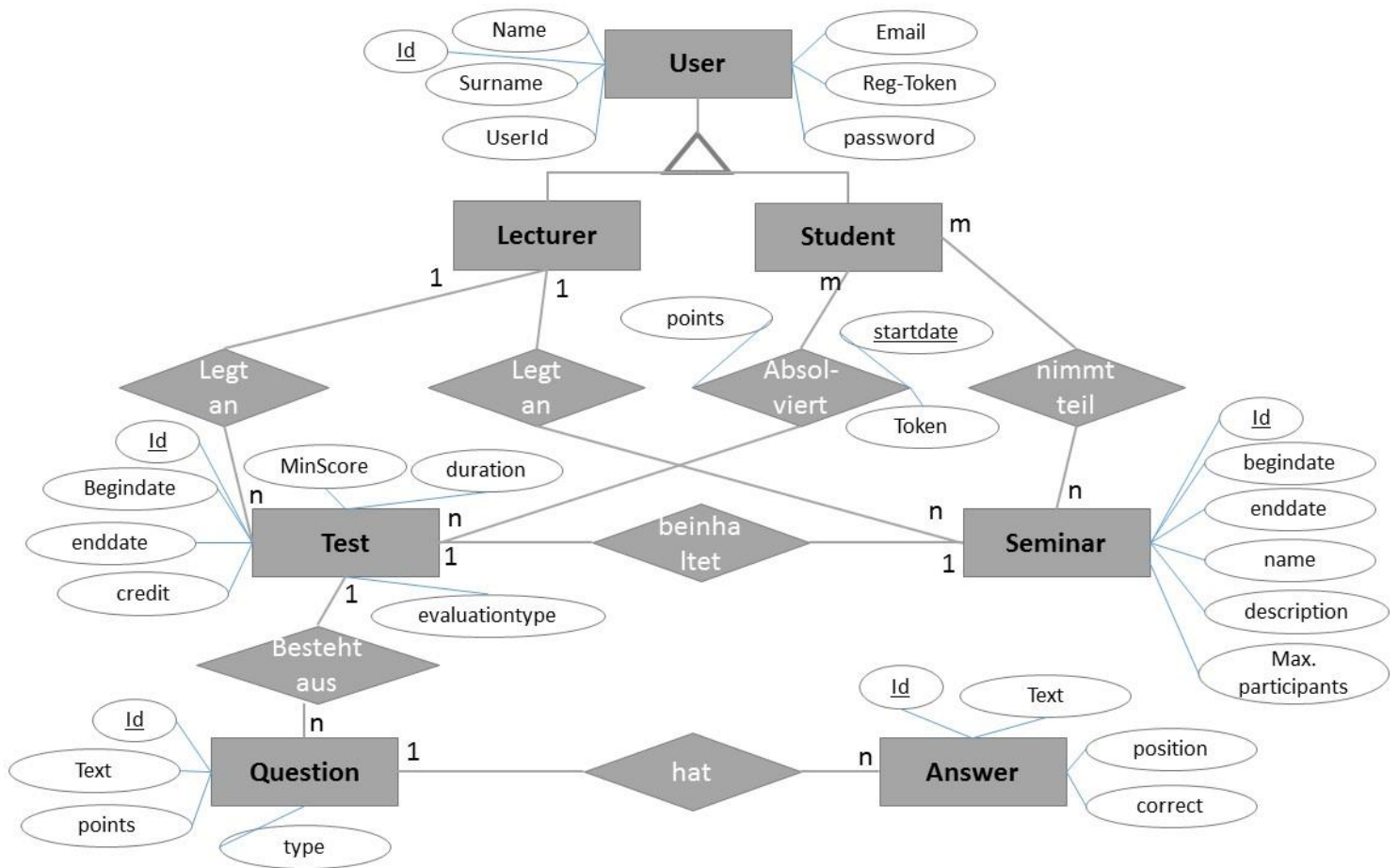
4.1. Softwarearchitektur



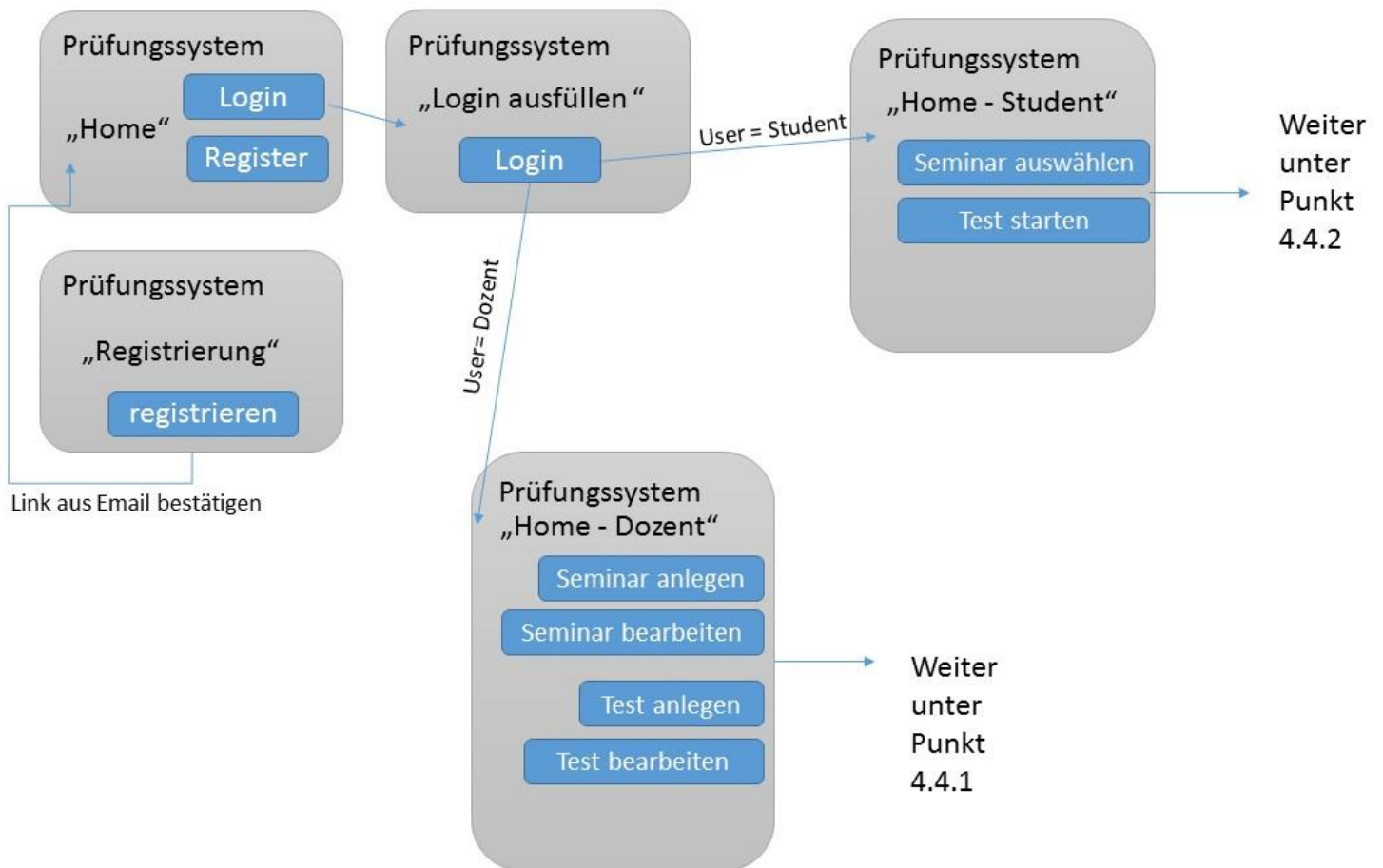
4.2. Klassenmodell

Aufgrund der Größe ist es als Klassendiagramm.png direkt im Projekt gespeichert

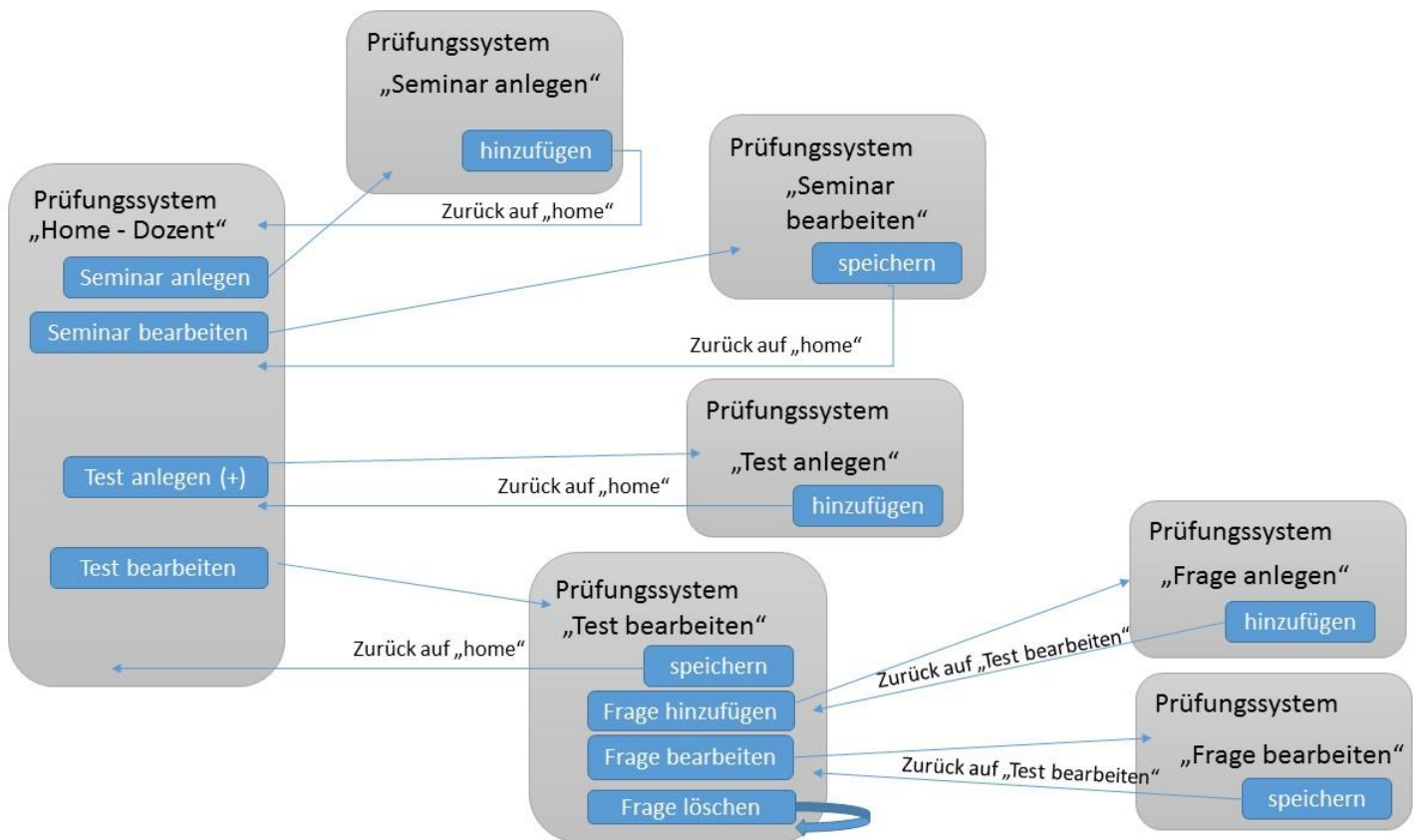
4.3. ER-Modell



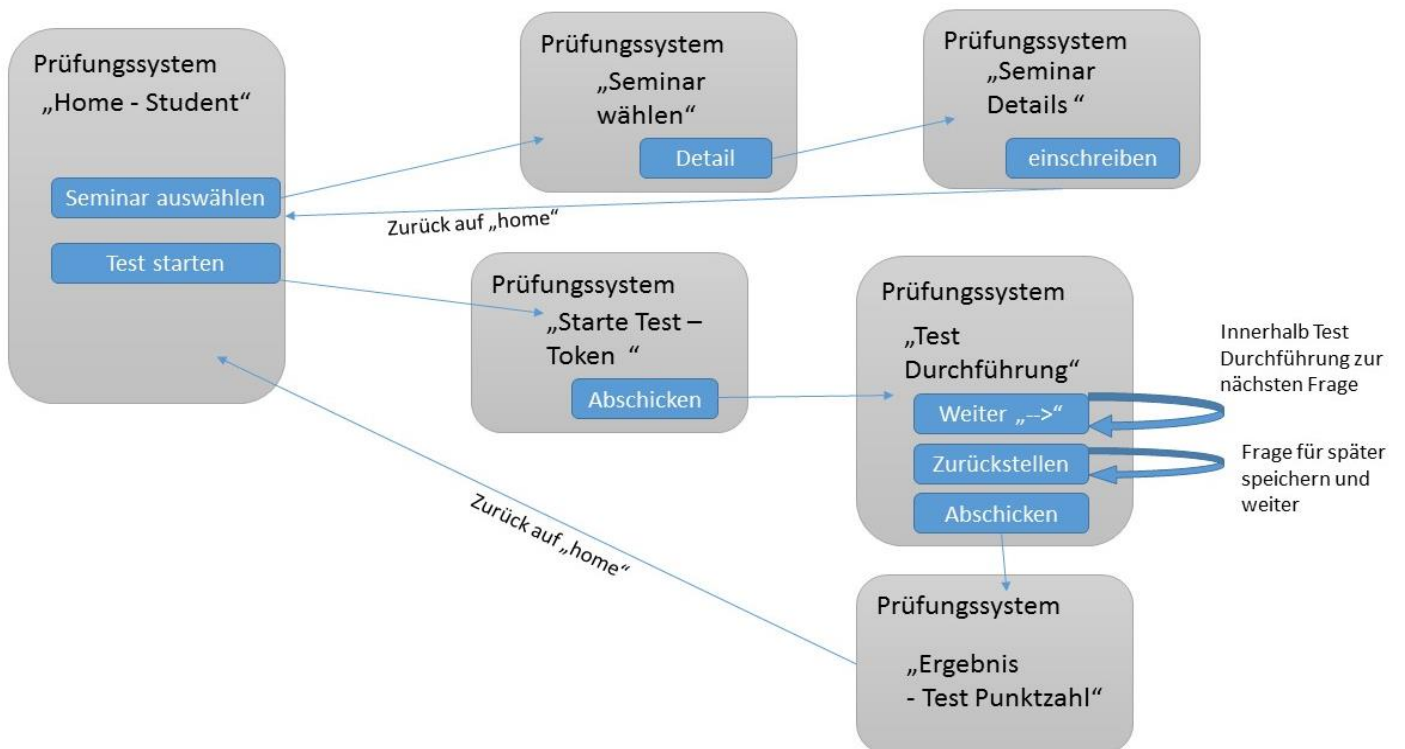
4.4. Dialogmodell



4.4.1. Sicht des Dozenten



4.4.2. Sicht des Studenten



4.5. Entwurf des Usability Design

Wir haben uns nicht nur wie im Dialogmodell dargestellt konzeptionelle Gedanken gemacht, wie der Anwendungsfluss gestaltet werden soll, sondern uns auch intensiv damit auseinandergesetzt, wie die einzelnen Masken im Frontend aussehen sollen.

Unsere Ideen haben wir zeichnerisch festgehalten. Den ersten Entwurf haben wir intensiv diskutiert und schnell konnten wir uns auf einige Verbesserungsmöglichkeiten einigen. Diese haben wir in unserem finalen Layout festgehalten. Anschließend konnte die Entwicklung der Seitengestaltung starten.

Die Zeichnungen können aus den im Anhang verwiesenen Dateien entnommen werden.

Hilfsmittel und Quellenverzeichnis

Grundsätzlich diene uns unser Wissen aus der Vorlesung als Quelle und somit konnte der Code eigenständig entwickelt werden.

Apache Maven

<https://maven.apache.org/plugins/maven-compiler-plugin/dependency-info.html>

Freemarker

<http://freemarker.org/freemarkerdownload.html>

Hibernate & H2

<http://hibernate.org/orm/downloads/>

<http://www.h2database.com/html/main.html>

JavaMail API

<http://www.oracle.com/technetwork/java/javamail/index.html>

Lombok

<https://projectlombok.org/mavenrepo/>

Mockito

<https://mvnrepository.com/artifact/org.mockito/mockito-all/1.9.5>

Spring

<https://mvnrepository.com/artifact/org.springframework/spring-webmvc/4.3.3.RELEASE>

Struts 2

<http://struts.apache.org/downloads.html>

Anhang

4.6. Anhang 1: Entwurfszeichnungen

Datei Entwurf.pdf im Projekt hinterlegt

4.7. Anhang 2: Layout

Datei Layout_v2.pdf im Projekt hinterlegt