Dokumentation der Hausarbeit im Modul Internetanwendungsarchitekturen

Aufgabe: Multiple Choice Test

Dozent: Benedikt Stemmildt

Daniel Fiolka (Matrikelnr. 6444)*
Steven Hens (Matrikelnr. 6666)†
Dieke Morten Lübberstedt (Matrikelnr. 6547)‡
Christopher Ost (Matrikelnr. 6548)§

NORDAKADEMIE
Köllner Chaussee 11
25337 Elmshorn

*daniel.fiolka@nordakademie.de
†steven.hens@nordakademie.de
‡dieke_morten.luebberstedt@nordakademie.de
§christopher.ost@nordakademie.de

Inhaltsverzeichnis

1 Einleitung

Diese Dokumentation beschreibt die Umsetzung des Hausarbeitsthemas *Multiple Choice Test*. Die Umsetzung beinhaltet eine Internetanwendung mit deren Hilfe Prüfungen für Nordakademie-Seminare und -Vorlesungen durch die Dozenten erstellt und von Studenten durchgeführt werden können. Die Anwendungen wird mit einer Schichtenarchitektur realisiert. Für das Backend wird Java verwendet, für die Datenhaltung eine H2-Datenbank und für die Persistierung das Framework Hibernate. Für das Frontend kommt das Framework Struts2 zum Einsatz.

2 Annahmen

Folgende Annahmen wurden für die Bearbeitung der Hausarbeit getroffen:

- Der Aufgabenteil *Fehlerhafte Eingaben können korrigiert werden, solange der Test noch nicht durchgeführt wurde.* wird verstanden als *solange der Anfang des Bearbeitungszeitraum noch nicht erreicht ist*.
- Beim Anlegen von Fragen durch Dozenten wird davon ausgegangen, dass der Dozent darauf hinweist, ob exakte eine oder mehrere Antworten korrekt sind.
- Beim Anlegen von Antworten wird davon ausgegangen, dass Dozenten dies sinnvoll tun. Das heißt, es gibt keine Fragen mit keiner oder nur einer Antwort.
- Für die Kommunikation des Einmalpassworts an die Studenten wird ein anderes Medium als die Internetanwendung benutzt, z. B. E-Mail.
- Die Anwendung ist mit einem System verbunden, aus dem die User (Dozenten und Studenten) ausgelesen werden. Es sind User zum Testen hinterlegt, welche der import.sql entnommen werden und dort auch selbst erweitert werden können. Folgende User sind eingepflegt:
  - Student Hans Petersen mit der Kennung *hans.petersen* und dem Kennwort *password*
  - Student Diecke Luebberstadt mit der Kennung *diecke.luebberstadt* und dem Kennwort *password*
  - Student Schtiewn Hens mit der Kennung *schtiewn.hens* und dem Kennwort *password*
  - Student Chris West mit der Kennung *chris.west* und dem Kennwort *password*
  - Student Daniela Katzenburger mit der Kennung *daniela.katzenburger* und dem Kennwort *password*
  - Dozent Uwe Mayer mit der Kennung *uwe.mayer* und dem Kennwort *nimda*
  - Dozent Karl Karlson mit der Kennung *karl.karlson* und dem Kennwort *admin*

3 Pflichtenheft

In diesem Kapitel sind die Anforderungen an die Anwendung im Sinne eines Pflichtenhefts dokumentiert.

3.1 Muss-Kriterien

- Dozenten und Studenten können sich online über einen Webbrowser am System anmelden.
- Ein Dozent kann einen oder mehrere Tests anlegen.
- Ein Dozent kann den Bearbeitungszeitraum, die möglichen Creditpoints, die Bestehensgrenze und die Bearbeitungsdauer für eine Prüfung festlegen.
- Zu den Tests können eine oder mehrere Fragen angelegt werden.
- Ein Dozent kann die Punkte für korrekte Antworten, falsche Antworten und fehlende Antworten festlegen.
- Zu den Tests können eine oder mehrere Lückentexte angelegt werden.
- Zu den Fragen können eine oder mehrere Antworten angelegt werden
- Die Fragen können

  eine einfache Auswahlmöglichkeit (eine richtige Antwort) aufweisen.

  eine mehrfache Auswahlmöglichkeit (mehrere richtige Antworten) aufweisen.
- Ein Lückentext kann eine oder mehrere richtige Antworten enthalten.
- Änderungen am Test sind noch möglich, wenn sein Startzeitpunkt noch nicht erreicht wurde.
- Ein Dozent kann Studenten mit einem Einmalpasswort die Teilnahme am Test ermöglichen.
- Ein Student kann eine nichtbestandene Prüfung nicht wiederholen.
- Einem Studenten werden die zu absolvierenden Tests angezeigt.
- Ein Student kann sich zum Ablegen einer Prüfung entscheiden und durchläuft alle dazugehörigen Fragen.
- Wenn der Bearbeitungszeitraum der Prüfung abgelaufen ist, kann der Student keine Antworten mehr abgeben. Die Prüfung ist damit autmatisch beendet.

3.2 Kann-Kriterien

- Ein Student hat die Möglichkeit, Fragen zurückzustellen und sich diese nach der letzten Frage nochmals mit ggf. vorgenommenen Teilantworten anzeigen zu lassen.
- Vor Ende der Prüfung kann sich ein Student noch einmal alle Antworten zu den Fragen anzeigen lassen und ggf. korrigieren.

3.3 Abgrenzungskriterien

- Die Entität Seminar muss nicht berücksichtigt werden.
- Ein Dozent muss sich nicht online registrieren können.
- Ein Student muss sich nicht online registrieren können.

3.4 Zielgruppen

Das Produkt dient den Seminarvortragenden zur möglichst einfachen Überprüfung der im Seminar vermittelten Lehrinhalte an der NORDAKADEMIE. Den Studenten wird diese Überprüfung in Form eines Multiple-Choice-Tests angeboten. Zielgruppe dieser Anwendung sind vortragende Dozenten und die Studenten der NORDAKADEMIE.

`Dozenten`: Seminarvortragende Dozenten haben die Möglichkeit ihren vermittelten Lehrinhalt anhand eines selbst erstellten Multiple-Choice-Tests zu prüfen.

`Studenten`: Die Studenten der NORDAKADEMIE haben die Möglichkeit an einer oder mehreren Prüfungen in Form von Multiple Choice Tests teilzunehmen.

4 Installationsanleitung

Für das Ausführen der Anwendung wird Java in Version 8 vorausgesetzt.

4.1 Vorbereitung in IntelliJ

Um die Anwendung ausführen zu können, muss diese zunächst in IntelliJ importiert werden. Der Import findet hierbei aus GitHub statt. Eine Zugriffsberechtigung auf das Projekt in GitHub ist hierzu notwendig. Um das Projet in IntelliJ zu importieren wird unter *File -> New -> Project from Version Control -> Git* der Import initiiert. In dem sich öffnendem Fenster wird unter

Git Repositorx URL folgender Link eingetragen: **https://github.com/NakTNHenSte/multiple-choice**. Unter *Parent Dictionary* wird das Verzeichnis gewählt, in dem das Projekt gecloned bzw. abgespeichert werden soll. Der *Directory Name* muss nicht weiter angepasst werden. Durch Klicken auf *clone* wird dann das Projekt heruntergeladen und in das gewählte *Parent Dictionary* abgespeichert. Anschließend sollte IntelliJ das neue Projekt öffnen. Für die weiteren Einstellungen ist es von Nöten, dass das Projekt als Maven-Projekt vorliegt. Dazu wird im Projekt-Explorer mit Rechtsklick der Punkt *Add Framework Support...* ausgewählt. Im sich öffnendem Fenster wird *Maven* angehakt und mit *OK* bestätigt.

4.2 Vorbereitung in IntelliJ - TomCat

Für die Ausführung der Anwendung wird der TomCat-WebServer benötigt. Dieser muss unter *Run -> Edit Configurations* zunächst konfiguriert werden. Durch das Klicken auf das + Symbol wird ein Menü geöffnet, in dem der Tomcat Server ausgewählt wird. Dies geschieht durch Klicken auf *Tomcat Server -> Local*. In dem sich öffnendem Fenster wird dann der Name des Servers eingetragen (z.B. TomCat). Im Reiter *Server* muss sichergestellt sein, dass unter dem Punkt JRE: die richtige JRE (1.8) angegeben ist. Die Ports können beibehalten werden. Es kann darüber hinaus auch sein, dass etwas überhalb des OK-Buttons eine Warnung mit der Meldung *Warning: No artifacts configured* angezeigt wird. In diesem Fall wird auf den Button *Fix* geklickt und mit anschließendem Klick unter *Problems* und dem Klick auf das Projekt (hier: multiple-choice)geklickt. Unter dem Punkt *Project SDK* muss die aktuelle JRE (1.8) ausgewählt und mit Klick auf *OK* bestätigt werden. Weitere Einstellungen sind dann unter dem Reiter *Deployment* vorzunehmen. Dort wird durch das + Symbol rechts vom Fenster mit “Nothing to deploy” ein neues Artifakt hinzugefügt. Im sich öffnendem Fenster wird der Typ *war exploded* ausgewählt und die Konfiguration mit dem OK-Button abgeschlossen.

4.3 Vorbereitung in IntelliJ - Datenbankpfad

Es muss der Pfad der Datenbank noch angepasst werden. Dieser ist im Projekt in der *spring.xml* anzupassen. Zu finden ist die Datei unter *multiple-choice -> src -> main -> resources*. Der dort hinterlegte Pfad *jdbc:h2:C:/Users/Steven/git/multiple-choice/db.mv.db/* muss mit dem richtigen Pfad des auszuführenden Computers angepasst werden. Die Verwendung eines relativen Pfades kann unter Umständen nicht zuverlässig funktionieren. Dies müsste ggf. ausprobiert werden.

4.4 Anwendung ausführen

Um die Anwendung auszuführen muss lediglich oben rechts in der Zeile neben dem inzwischen zu sehendem TomCat-Server ein grünes Play-Sysmbnol zu sehen sein. Beim Klick auf dieses

wird der Server gestartet und der Browser mit der Anwendung geöffnet.

5 Zurücksetzen der Datenbank

Die Aufgabenstellung besagt, dass die die Datenbank der Anwendung in einen initialen Stand versetzt werden können muss, der ein Testen der Anwendung ermöglicht. Dies wurde folgendermaßen umgesetzt:

- Die Datenbank wird bei jedem Start der Anwendungen mit Daten aus der import.sql gefüllt.
- Das Handling der Datenbank durch Hibernate ist in der spring.xml konfiguriert. In den `jpaProperties` wird eine `property` als `create-drop` gesetzt. Bei jedem Start der Anwendung wird die verbundene H2-Datenbank verworfen und neu initialisiert.

6 Sonstige Anmerkungen

Unter diesem Abschnitt sind weitere Anmerkungen im Bezug zur Hausarbeit vermerkt.

6.1 Verwendung von Mockups

In der beginnenden Phase der Hausarbeit haben wir uns in unseren Besprechungen über ein mögliches Design der Anwendung ausgetauscht. Dieses haben wir in Mockups festgehalten. Im weiteren Verlauf der Hausarbeit haben wir jedoch festgestellt, dass das komplette Einhalten des gewählten Designs zu viel Zeit in Anspruch nimmt.

7 Tools und Hilfsmittel

https://struts.apache.org

https://www.mkyong.com/struts2/struts-2-action-tag-example/

http://www.codejava.net/frameworks/struts/working-with-httpsession-in-struts2-a-login-example

https://www.javatpoint.com/struts-2-login-and-logout-example

https://stackoverflow.com/questions/5509606/sessions-in-struts2-application

https://www.mkyong.com/struts2/struts-2-actionerror-actionmessage-example/

http://www.simplecodestuffs.com/login-interceptor-struts-2/

https://mail-archives.apache.org/mod_mbox/struts-user/200811.mbox/%3C20402372.post@talk.nabble.com%3E

https://stackoverflow.com/questions/4583285/change-jsp-on-button-click

https://regexr.com/

https://balsamiq.com/

https://github.com

https://erdplus.com/#/

https://www.jetbrains.com/idea/

https://getbootstrap.com/

https://jquery.com/

https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.3/umd/popper.min.js