



Leistungsbeurteilung 151

Modulnummer 151 Version/Release R6 Anbieter Version 3.0
Titel Datenbanken in Web-Applikation einbinden
Autoren Michael Schneider, Alexander Flick (alexander.flick@bbb Baden.ch)

Datum:

Name:

Vorname:

Klasse:

Experte:

Note:

Hilfsmittel

Folgende Hilfsmittel sind erlaubt:

- Alle Unterlagen
- Internet

Prüfungsdauer
Prüfungsinfrastruktur

Ganze Unterrichtsdauer Modul 151
Folgende Infrastruktur steht zur Verfügung:

- Eigenes Notebook

Notenskala

Linear (Note = $1 + 5 \cdot \frac{\text{Erreichte Punktzahl}}{\text{Gesamte Punktzahl}}$), auf halbe Noten gerundet.
1 ist die schlechteste, 6 die beste Note.

| HZ | Handlungsziel | Aufgabe(n) |
|----|---|------------|
| 1 | Anforderungen an eine Web-Applikation und Gegebenheiten der einzubindenden Datenbank bzw. des Datenbestandes analysieren, Anbindungstechnik definieren und dokumentieren. | |
| 2 | Schutz- und sicherheitswürdige Informationen unter Beachtung des Datenschutzes identifizieren und Massnahmen definieren. | |
| 3 | Anbindung der Web-Applikation an die Datenbank bzw. an den Datenbestand realisieren und dabei Transaktionen, Datensicherheit und Datenschutz beachten. | |
| 4 | Änderungswunsch gemäss vorgegebenem Änderungsvorgehen umsetzen. | |
| 5 | Test- und Abnahmeverfahren definieren, durchführen und im Testprotokoll dokumentieren. Allenfalls erforderliche Korrekturen vornehmen. | |

Beurteilungsgesichtspunkte:

| Aufgabe | Gewichtung | Bewertung |
|---------|------------|---|
| 1 | 1 | Die Abgabe des Projektes erfolgte nach Vorgabe. |
| 2 | 1 | Die Tiers und die Aufgaben wurden korrekt und konkret aufgelistet. |
| 3 | 1 | Die Technologien wurden pro Tier korrekt aufgelistet. |
| 4 | 1 | Die Datenbanksteuerung wurde richtig beschrieben und kategorisiert. |
| 5 | 1 | Die GUI Prototypen berücksichtigen die Vorgaben. |
| 6 | 3 | Das Administrationsinterface wurde nach Vorgaben realisiert |
| | 3 | Die Webapplikation ist professionell implementiert |
| | 2 | Die Passwortabfrage wurde implementiert und dokumentiert. |
| | 3 | Die Applikation (Quiz) wurde nach Vorgaben realisiert. |
| | 1 | Sicherheitsaspekte umgesetzt |
| 7 | 1 | Die Form und der Inhalt des Abnahmeprotokolls entsprechen den Vorgaben. |
| 8 | 2 | Die Tests im Abnahmeprotokoll wurden professionell durchgeführt und dokumentiert. |
| 8 | 1 | Unit Test erstellt und durchgeführt, Silenium Test dokumentiert und durchgeführt. |
| 9 | | Die abgesprochenen Erweiterungen entsprechen dem definierten Umfang. |

Punkte:

| Aufgabe Nr.: | Gewichtung: | Punktzahl: | Erreicht: |
|----------------------|-------------|------------|-----------|
| 1 | 1 | | |
| 2 | 1 | | |
| 3 | 1 | | |
| 4 | 1 | | |
| 5 | 1 | | |
| 6 | 3 | | |
| | 3 | | |
| | 2 | | |
| | 3 | | |
| | 1 | | |
| 7 | 1 | | |
| 8 | 2 | | |
| 8 | 1 | | |
| 9 | | | |
| Total Punkte | 63 | | |
| Note ungerundet | | | |
| Note gerundet | | | |

Hinweise:

- Diese Prüfung ist eine Einzelarbeit.
- Geben Sie diese Aufgabenstellung und Lösungsmaterial beschriftet mit Namen, Vornamen und Klasse ab.
- Bezeichnen Sie Lösungen und Korrekturen klar und deutlich.
- Verwenden Sie einen dokumentenechten Stift (Tinte/Kugelschreiber).
- Wenn Sie Lösungen elektronisch oder auf Zusatzblättern abgeben, schreiben Sie auf dem Aufgabenblatt einen Verweis auf die entsprechenden Aufgaben.
- Alle Daten dieser LB (Aufgabendateien und Lösungen) müssen sofort nach dem Abschluss der LB und dem Hochladen gelöscht werden.
- Es dürfen keine alten Prüfungen und/oder Daten von alten Prüfungen verwendet werden.
- Elektronisch abgegebene Lösungen werden mit Vorteil gezippt. Andere Kompressionsverfahren sind auf eigene Gefahr hin zu benutzen. Es liegt in Ihrer Verantwortung, dass alle Dateien abgegeben werden.
- Die elektronischen Lösungen werden am Ende der Prüfungszeit auf Moodle hochgeladen.
- Alle Dateien müssen in einem Ordner abgelegt werden, dessen Name dem Format VornameNachname_LB_M151_V232 entspricht. (z.B. HansMuster_LB_M151_V232.zip).
 - Die gepackte Datei soll gleich benannt werden wie der Ordner.
- Folgende Formate sind bei den elektronischen Lösungen erlaubt:
 - Programmcode: Netbeans-Projekt mit Maven-Webapplikation
 - Bilder: .png, .jpg, .gif
 - Diagramme: siehe Bilder.
 - Texte: ASCII-Texte (*.sql)
 - Anderes: Microsoft Word
- Vor Beginn der Prüfung abgegebene Daten:
 - M151_Daten.zip

I Administrative Vorgaben

Diese Leistungsbeurteilung (LB) ist eine individuelle Projektarbeit. Einzelne Teile dieser LB können während dem Unterricht bearbeitet werden, nachdem die entsprechende Theorie vermittelt worden ist. Die restliche Arbeit muss zu Hause geleistet werden. Eigene Erweiterungen sind willkommen und werden mit Punkten belohnt. Sie müssen aber mit der Lehrperson abgesprochen und dokumentiert werden (siehe Aufgabe 9 auf Seite 8).

Zur Realisierung dürfen sie eine beliebige geeignete Programmiersprache verwenden.

Erstellen Sie ein Lösungsdokument (MusterHans_LB_M151_V232.doc), indem Sie die geforderten Teilaufgaben ablegen.

II Fallbeschreibung

Ziel des Projektes ist es, eine abgewandelte Version des Quiz zu implementieren, das in der Fernsehshow «Wer wird Millionär» durchgeführt wird. Dieses Quiz soll zu Lernzwecken und zur Unterhaltung gebraucht werden können.



Abbildung 1: Ein Kandidat in der Quizshow

Wer wird Millionär im Fernsehen

In jeder Spielrunde werden eine Frage und vier Antworten präsentiert, von denen nur eine korrekt ist. Für jede richtige Antwort erhält der Kandidat einen Geldbetrag, bei einer falschen Antwort verliert er alles, was er sich erspielt hat und scheidet aus. Der Kandidat kann bei jeder Frage bevor er die Antwort gegeben hat das Spiel selbstständig beenden und den Gewinn kassieren. In der Sendung gibt es drei Joker, die jeweils nur einmal eingelöst werden dürfen: Der Kandidat darf jemanden anrufen, das Publikum befragen oder zwei falsche Antworten ausblenden.

Wer wird Millionär als Webapplikation

In der Webapplikation soll die Frageform - eine Frage mit vier Antwortmöglichkeiten, von denen nur eine korrekt ist - beibehalten werden. Für jede korrekt beantwortete Frage erhält der Spieler 30 Punkte. Damit die Recherchemöglichkeit eingeschränkt wird, soll die Zeit zwischen dem Start des Quiz und dem Abschluss gemessen werden. Der 50:50-Joker soll dem Spieler einmal zur Verfügung stehen. Benutzt er ihn, werden zwei falsche Antworten ausgeblendet. Die Fragen werden von einem Administrator eingepflegt und unterhalten. Als Ansporn gibt es eine Highscoreliste, die einen Vergleich der Resultate ermöglicht.

Anforderungen:

- Administration
 - Der Administrator muss sich durch Username und Passwort authentifizieren.
 - Der Administrator muss Fragen mit Antworten anlegen, ändern und löschen können.
 - Der Administrator muss Kategorien anlegen und jede Frage einer Kategorie zuordnen können.
 - Der Administrator kann einzelne Einträge der Highscoreliste löschen.
- GUI
 - Als Client dient ein Webbrowser.
 - Zu jeder Frage wird gespeichert, wie oft sie richtig und wie oft sie falsch beantwortet worden ist.
 - Zu jeder Frage wird beim Quiz prozentual angezeigt, wie oft sie richtig beantwortet wurde.
 - Der Spieler sieht zu jeder Zeit seine aktuelle Punktzahl.
 - Der Spieler sieht zu jeder Zeit, ob er den 50:50-Joker noch einsetzen kann.
 - Der 50:50 Joker markiert zwei falsche Antworten und macht sie unauswählbar.
 - War die vom Spieler gewählte Antwort richtig, so wird dies dem Spieler mitgeteilt und weiter zur nächsten Frage gegangen.
 - War die vom Spieler gewählte Antwort falsch, so wird dies dem Spieler mitgeteilt und die richtige Antwort dargestellt.
 - Hat der Spieler eine falsche Antwort eingegeben, so bricht das Quiz ab, der Versuch wird mit 0 Punkten gewertet und erscheint nicht auf der Highscoreliste, ansonsten schon.
 - Der Spieler muss seinen Namen eingeben können, mit dem er auf der Highscoreliste erscheint.
 - In der Highscoreliste werden folgende Daten aufgeführt:
 - Rang
 - Gewichtete Punkte: Punktzahl geteilt durch die Dauer des Quiz in Sekunden
 - Name des Spielers
 - Zeitpunkt des Spiels
 - Anzahl Punkte
 - Dauer des Quiz
 - Gewählte Kategorien
 - Die Highscoreliste wird nach Rang, der durch die gewichteten Punkte bestimmt wird, aufsteigend sortiert.
- Spiel
 - Der Spieler kann die Kategorien wählen, aus denen die Fragen zufällig ausgewählt werden.
 - Die Fragen mit den Antworten werden dem Spieler nacheinander präsentiert.
 - Keine Frage soll während eines Spieles mehr als einmal gestellt werden.
 - Zu jeder Frage gibt es vier Antworten: Eine korrekte und drei falsche.
 - Der Spieler kann entweder eine Antwort auswählen oder aufhören und seinen Gewinn realisieren.
 - Jede korrekt gewählte Antwort gibt 30 Punkte auf das Spielerkonto.
 - Das Spiel soll mit einer spielbaren Anzahl Fragen gefüllt werden.
 - Die Zeit zwischen dem Start des Quiz und dem Aufhören soll gemessen werden.

Programmierung

- Einfache Formulareingaben, wie leere Textfelder etc., sollen auf Client- und Serverseite geprüft werden.
- Die Wahl der Datenbank steht Ihnen frei (objektorientierte DB, NoSQL-DB, relationale DB mit stored Procedures und referentielle Integrität). Ein relationales Datenbankschema liegt als Empfehlung (Moodle M151_Daten.zip) bei. Es steht Ihnen frei dieses zu nutzen.
- Es soll eine Datenbankanbindung verwendet werden, die möglichst unabhängig vom tatsächlich eingesetzten Produkt ist.
- Transaktionsmanagement ist einzusetzen.
- Sicherheitsaspekte werden umgesetzt.
- Die Applikation soll als eine sessionbasierte professionelle 4-Tier Architektur implementiert werden. Dies bedeutet:
Webserver Layer: Templating System muss eingesetzt werden. Ein externes Framework bzw. eine Komponentenbibliothek werden eingebunden. Einsatz von HTML5 und CSS.
Business Logic: Einziger Tier, der eine direkte Verbindung zur Datenbank hat.

III Aufgaben

Sie bekommen nun den Auftrag, die vorgegebene Fallbeschreibung (Situation) zu analysieren, einen Prototyp zu planen und diesen als Web-Applikation mit JSF umzusetzen. Notieren Sie Ihre Antworten im Lösungsdokument.

- 1) Setzen Sie alle Hinweise (S.4) und administrativen Vorgaben (S.5) um.
- 2) Beschreiben Sie, auf welchem Tier Sie die dynamischen Elemente der Anwendung unterbringen möchten:
 - a. Tier 1 (Presentation):
 - b. Tier 2 (Webserver):
 - c. Tier 3 (Application Server):
 - d. Tier 4 (Datasever):
- 3) Beschreiben Sie, welche Programmiersprache bzw. Technologie Sie verwenden möchten für:
 - a. Tier 1 (Presentation):
 - b. Tier 2 (Webserver):
 - c. Tier 3 (Application Server):
 - d. Tier 4 (Datasever):
- 4) Wie steuern Sie Ihre Datenbank an? Wie ist das Interface aufgebaut und zu welcher Kategorie gehört es?
- 5) Erstellen Sie Prototypen für das GUI (Admininterface und Quiz-Seite).
- 6) Implementieren Sie die Webapplikation mit Datenbankanbindung nach Vorgaben:
 - a. Administrationsbereich
 - i. Kreieren Sie ein Administrationsinterface für den Fragen-Administrator. Das Interface soll das Handling von Kategorien, Fragen und Antworten ermöglichen. Folgende Ansichten und Funktionen sollen verfügbar sein: Liste, neue Entität hinzufügen, Entität verändern, Entität löschen
 - ii. In der Highscoreliste sollen einzelne Einträge gelöscht werden können.

- iii. Implementieren Sie die Eingabeprüfung für das Admininterface. Nach Anforderung soll die Prüfung auf der Clientseite und **in weiteren Tiers stattfinden.** Zeigen Sie im Lösungsdokument, wie Sie in welchem Tier welche Überprüfungen vornehmen.
- b. Spiel
 - i. **Das Quiz soll nach Vorgabe durchgeführt werden können.**
- 7) Bereiten Sie eine Abnahme Ihrer Applikation vor. Erstellen Sie alle dafür notwendigen Dokumente.
- 8) Testen Sie Ihr Projekt und Dokumentieren Sie anhand Ihres Abnahmeprotokolls.
 - a. Funktionstest in Form von Unit-Test (Abgabe mit dem Projekt)
 - b. Integrationstest mit Selenium (Dokumentation im Lösungsdokument)
- 9) Fakultativ können Sie eigene Erweiterungen codieren. Erstellen Sie Anforderungen und sprechen Sie sich vorgängig mit der Lehrperson ab. In diesem Gespräch wird die mit den Erweiterungen erreichbare Punktzahl vereinbart.