

Tasks

Sopra-WS15/16

Gruppe 10

„Science Party“

1. Science-party-play

1.1. Backend

1.1.1. Startzeige anzeigen

Die Startseite soll unter “/” erreichbar sein, wenn kein User eingeloggt ist.

1.1.2. Registrieren anzeigen

Die Seite zum Erstellen neuer Accounts soll unter “/register” erreichbar sein.

1.1.3. Neuen User erstellen

Die vom Formular erzeugten Daten werden per POST an /register gesendet und dort verarbeitet. Es muss ein neuer User erzeugt und in der Datenbank abgespeichert werden. Ist die Registrierung erfolgreich, wird der User eingeloggt und automatisch zum Userbereich weitergeleitet. War die Registrierung nicht erfolgreich, wird der entsprechende Fehler z.B. mit 403 zurück gemeldet.

1.1.4. LogIn-Seite anzeigen

Die LogIn-Seite soll unter “/login” erreichbar sein, wenn kein User eingeloggt ist. Ist bereits ein User angemeldet, wird dieser zum Userbereich weitergeleitet.

1.1.5. Login eines Users verarbeiten

Die LogIn-Daten werden per POST an den Server gesendet. Wurde für die entsprechende E-Mail Adresse das richtige Passwort eingegeben, wird der entsprechende User durch das Erstellen einer Session eingeloggt und direkt in den Userbereich weitergeleitet. Wurde das falsche Passwort eingegeben oder ist ein anderer Fehler vorhanden, wird dieser z.B. mit 401 und einer entsprechenden Nachricht zurück gemeldet.

1.1.6. Anzeigen des Userbereichs

Ist bereits ein User angemeldet, ist der Userbereich unter “/” oder “/login” erreichbar. Die Startseite für den Userbereich ist das eigene Profil “/profil”.

1.1.7. Anzeigen eines Profils

Das eigene Profil soll unter “/profil” erreichbar sein und das Profil anderer User unter “/profil/<id>”. Beim Öffnen eines Profils müssen Folgende Daten an den Client übertragen werden:

- Informationen zum User wie z.B. Vorname, Nachname
- Speicherort des aktuellen Bildes/Avatares
- Die Spielstatistik mit Anzahl an Spielen, Ranglistenplatz etc.
- Bereits erworbene Fähigkeiten

1.1.8. Profil bearbeiten

Das eigene Profil soll bearbeitet werden können, die aktualisierten Daten für den eigenen Account werden per POST an /profil/edit gesendet. Die Seite zum Bearbeiten des Profils kann unter /profil/edit mit GET abgefragt werden.

1.1.9. Profilbild verwalten

Es soll möglich sein, sein eigenes Profilbild hoch zu laden. Die Seite zum Hochladen des Bildes ist unter /profil/picture verfügbar.

1.1.10. Hinzufügen einer Fähigkeit

Fähigkeiten können mittels QR Code eingelesen werden. Die Seite zum einlesen eines QR Codes ist unter “/qr” erreichbar. Hinter dem QR Code ist ein String hinterlegt welcher nach dem Auslesen per POST an “/qr” gesendet wird. Kann dieser String mit einer Fähigkeit verbunden werden, wird die Fähigkeit freigeschalten und eine Erfolgsmeldung an den Browser gesendet.

1.1.11. Profil löschen

Der Prozess um ein Profil zu löschen soll mit aufrufen von “/profile/delete” gestartet werden. Um versehentliches löschen von Profilen zu vermeiden wird eine Seite zurückgegeben, welche zur Bestätigung der Aktion mit Passwort aufruft. Das Passwort wird per POST an “/profile/delete” übergeben und wenn es richtig eingegeben wurde, wird der User ausgeloggt und der Account gelöscht.

1.1.12. Spiel erstellen

Ein User soll ein neues Spiel erstellen können. Dafür sind folgende Schritte Informationen:

1. Themenbereich auswählen (ID)
2. Anzahl der Spieler (INT)
3. Mitspieler (Liste mit IDs der Mitspieler)

Das Spiel wird in der Tabelle “games” erzeugt, der Status des Spieles wird auf “PENDING” gesetzt. In der Tabelle “player” werden die Mitspieler erzeugt und der Status der Mitspieler auf “INVITED” gesetzt.

1.1.13. Liste von Freunden

Auf die Anfrage GET Anfrage /friends/get/list soll eine Liste der Freunde zurückgegeben werden.

1.1.14. Startende Spiele

Auf die Anfrage GET /games/pending werden zwei Listen zurückgegeben.

- Spiele zu denen ich eingeladen und noch antworten muss
- Spiele denen ich beigetreten bin, die aber noch nicht begonnen haben

1.1.15. Spieleinladung annehmen

Per POST Anfrage an /games/accept/<id> kann ein Spieler einem Spiel betreten, jedoch nur wenn er auch für dieses eingeladen wurde. Wenn die Einladung angenommen wurde, soll geprüft werden, ob dies der letzte Spieler war und bei Bedarf das Spiel automatisch gestartet werden.

1.1.16. Spieleinladung ablehnen

Per POST Anfrage an /games/decline/<id> kann ein Spieler eine Spieleinladung ablehnen. Der Spieler wird für das Spiel gelöscht.

1.1.17. Spiel starten

Das Spiel soll gestartet werden, sobald alle Spieler zu oder abgesagt haben. Der Spielstatus wird auf "PLAYING" geändert. Es wird zufällig entschieden, welcher Spieler an der Reihe ist.

1.1.18. Spielstatus abfragen

Sobald das Spiel läuft, soll der Spielstatus als GET unter /game/status/<id> abgefragt werden können. Es wird nur ein Spielstatus zurückgegeben, wenn der Spieler auch Teilnehmer des Spieles ist. Die Abfrage muss folgende Informationen enthalten:

- Anzahl der Spieler
- Position der Spieler
- Aktiver Spieler

Wurde der Status vom aktiven Spieler abgefragt, wird zusätzlich noch folgendes übermittelt:

- Eine zufällige Frage.
- Die Fähigkeiten, welche er einsetzen kann.

1.1.19. Frage beantworten

Die aktuelle Frage kann mit einer POST Anfrage an /games/question/answer/<gameid> übermittelt werden. Es wird zurück gemeldet, ob die Frage richtig oder falsch war und was die richtige Antwort war. Nach dem Beantworten der Frage wird dem Spieler die Leistung verbucht und der nächste Spieler ist an der Reihe. Es wird überprüft ob der aktive Spieler Feld 40 erreicht hat. Wenn Feld 40 erreicht wurde, wird das Spiel beendet.

1.1.20. Fähigkeit einsetzen

Ein Spieler, der am Zug ist, kann eine Fähigkeit einsetzen. Die Fähigkeit, welche eingesetzt wird, kann per POST an /games/perk/<gameid> übermittelt werden. Der Spielstatus wird entsprechend der eingesetzten Fähigkeit aktualisiert.

1.1.21. Spiel beenden

Beim Beenden des Spieles soll der Ranglisten-Punktestand jedes Spielers aktualisiert werden. Das Spiel wird auf "FINISHED" gesetzt. Die Spieler werden auf "FINISHED" gesetzt.

1.1.22. Spiel verlassen

Das Spiel kann mit /games/leave/<id> verlassen werden.

1.1.23. LogOut

Man soll sich mit der GET Abfrage /logout abmelden können

1.1.24. Allgemeine Top10 Rangliste anzeigen

Es soll unter /ranking eine Liste mit den 10 besten Spielern zurückgegeben werden. Die Liste soll Name, Rang, Status und Anzahl der Spiele insgesamt anzeigen.

1.1.25. Themenbezogene Top10 Rangliste anzeigen

Es soll unter /ranking/<themenid> eine Liste mit den 10 besten Spielern pro Themenbereich zurückgegeben werden. Die Liste soll Name, Rang, Status und Anzahl der Spiele insgesamt anzeigen.

1.1.26. Gespräche anzeigen

Unter der GET Anfrage /messages soll einer Übersicht aller Gespräche angezeigt werden.

1.1.27. Gespräch anzeigen

Es soll ein Gespräch mit einer Person/Gruppe angezeigt werden können. Dieses kann via GET unter /messages/show/<id> angezeigt werden können.

1.1.28. Nachricht senden

Es kann eine Nachricht in einen Chatverlauf gesendet werden. Diese wird per POST an /messages/send/<id> gesendet werden

1.1.29. Gespräch löschen

Es soll möglich sein ein Gespräch zu löschen bzw. aus einem Chat auszutreten. Dies wird per POST an /messages/delete/<id> angefordert.

1.1.30. Neues Gespräch erzeugen

Es soll ein neues Gespräch mit beliebig vielen Teilnehmern erzeugt werden können. Das wird ein neuer Chat in der Datenbank erstellt, welchem die entsprechenden Teilnehmer hinzugefügt werden. Die Teilnehmer und die erste Nachricht werden per POST an /message/new/send übergeben.

1.1.31. Personen suchen

Es soll möglich sein nach einer Person zu suchen. Auf POST Anfrage mit Vorname, Nachname /oder E-Mail an /profil/search/ wird ein Liste mit gefundenen Personen zurückgegeben.

1.1.32. Freunde anzeigen

Es soll eine Liste mit Freunden angezeigt werden, welche per GET /friends abgerufen werden kann.

1.1.33. Freundschaftsanfragen anzeigen

Es sollen alle offenen Freundschaftsanfragen in einer Liste angezeigt werden. Diese sollen per GET /friends/requests angefordert werden.

1.1.34. Freundschaftsanfrage beantworten

Eine Freundschaftsanfrage soll per POST an /friends/request beantwortet werden können.

1.1.35. Freund entfernen

Ein Freund kann per POST an /friends/delete/<id> gelöscht werden.

1.1.36. Userdaten von allen Usern über AuthorConnect abrufen

Per HTTP Post Abfrage an /ac/user/list soll eine Liste mit allen Usern im JSON Format abgefragt werden können.

1.1.37. Userdaten für einen User über AuthorConnect abrufen

Per HTTP Post Abfrage an /ac/user/<id> sollen die Userdaten des entsprechenden Users abgefragt werden können.

1.1.38. Liste von Themen inkl. Fragen und Antworten abrufen

Per HTTP Post Abfrage an /ac/topic/list soll eine Liste mit allen Topics und zugehörigen Fragen/Antworten im JSON Format abgefragt werden können.

1.2 Front-End

1.2.1. Aussehen: Startseite

Horizontales allgemeines Menü, für nicht registrierte User/Gäste.

Logo (Groß; top, center)

Auf der Startseite sollen Logo, Video und Loginbutton angezeigt werden.

1.2.2. Aussehen: Registrieren-Seite

Horizontales allgemeines Menü, für nicht registrierte User/Gäste.

Logo (top, center)

Formularblock zur Erfassung der Daten.

1.2.3. Registrieren

Auf der Registrierseite soll ein Formular zur Erfassung von E-Mail, Geburtsdatum, Vor- und Nachname eingefügt werden. Diese werden als JSON via AJAX an die Datenbank weiter geleitet.

HTTP Messages: -E-Mail bereits vorhanden
-Verbotener Input

1.2.4. Aussehen: LogIn-Seite

Horizontales, allgemeines Menü, für nicht registrierte User/Gäste.

Logo (top, center)

Formularblock zur Eingabe der LogIn-Daten

1.2.5. LogIn

Auf der Loginseite soll ein Formular zur Erfassung der Logindaten (E-Mail, Passwort)

eingefügt werden. Diese werden als JSON via AJAX an den Server gesendet und abgeglichen

HTTP Messages: Fehlerhafte Eingabe: E-Mail oder Passwort falsch.

1.2.6. Aussehen: Userbereich (Start)

Horizontales, authorized Usermenü, für eingeloggte, registrierte User.

Logo (top, left, menü)

Logout button (top, right)

1.2.7. Userbereich (Start)

Startseite des Userbereichs stellt das eigene Profil dar, das unter “/profile” erreicht wird.

Dabei werden dem Nutzer Profilbild, Nutzerdaten, Ranking, Statistiken und Fähigkeiten präsentiert. Diese Parameter können im Gegensatz zu Freudprofilen im eigenen Profil modifiziert werden.

1.2.8. Aussehen: Profil Anzeigen

Besteht aus Blöcken (Template), die via Serverabfrage mit Inhalt gefüllt werden.

1.2.9. Profil Interface

Profil kann bearbeitet werden.

Mittels Buttons können einzelne Elemente bearbeitet werden.

1.2.10. Profilbild verwalten

Das Profilbild soll mittels eines Buttons bearbeitet werden können.

Auf der der Seite /profile/picture kann das Bild hochgeladen oder gelöscht werden.

1.2.11. Aussehen: Fähigkeit hinzufügen (QR-Code Scan)

Horizontales, authorized Usermenü, für eingeloggte, registrierte User.

Logo (top, left, menü)

LogOut-Button (top, right)

Scannfenster (middle, center)

Auf der “Fähigkeiten hinzufügen”-Seite kann mittels eines QR-Code Scanners ein String eingelesen werden, dieser wird als JSON via AJAX an den Server gesendet und abgeglichen.

1.2.12. Aussehen: Profil löschen

Horizontales, authorized Usermenü, für eingeloggte, registrierte User.

Logo (top, left, menü)

LogOut-Button (top, right)

Bestätigungsblock, Warnung vor Löschen, Bestätigung mittels Passworteingabe und "OK", Abbruch durch "Abbrechen", Passwort wird als JSON via AJAX an den Server gesendet und abgeglichen, falls Korrekt ⇒ Nutzerdaten löschen

1.2.13. Spielen-Seite

Auf der Seite /game soll ausgewählt werden ob ein neues Spiel erstellt werden soll oder ein bestehendes geladen werden soll. Des Weiteren sollen die Einladungen aufgerufen werden.

1.2.14. Aussehen: Spiel erstellen

Es soll ein Dialog implementiert werden, in welchem ein neues Spiel erzeugt werden kann.

Folgende Informationen sollen dabei erfasst werden:

- Themenbereich
- Anzahl der Mitspieler
- Mitspieler (Einladung senden) -> Freundesliste aufrufen und einladen

1.2.15. Aussehen: Einladungen

In diesem Dialog soll angezeigt werden, welche Spiele gerade in der Startphase sind.

Es sollen alle aktuellen Einladungen angezeigt werden.

Einladungen können "Accepted" und "Declined" werden.

1.2.16. Spieler-Status anzeigen

Der Spielerstatus wird im Profil und in der Freundesliste neben den Namen angezeigt.

Bsp.: Name (Status) → Christian Weingärtner (Status: PLAYING)

1.2.17. LogOut

Logout-Button in Menüleiste rechts außen.

1.2.18. Ranking

Tabelle mit Top 10. (Default: Rang insgesamt)

1.2.19. Ranking pro Thema

Themenbezogen (Select-Option-Form)

1.2.20. Alle Gespräche

Liste von bestehenden Chats

1.2.21. Ausgewähltes Gespräch

Die Liste von Nachrichten soll zeitlich sortiert sein.

1.2.22. Nachricht senden

Formular am Fuß eines ausgewählten Gesprächs.

Hier kann ein Textfeld ausgefüllt werden und mittels eines Buttons kann der eingegebene Inhalt gesendet werden. Dabei wird ein Zeitstempel mit erfasst um die Nachrichten der Reihe nach darzustellen.

1.2.23. Gespräch löschen

In der Liste aller Gespräche befindet sich am Ende der Zeile des jeweiligen Gesprächs ein Button zum Löschen des Gesprächs. Das Löschen eines Gesprächs muss bestätigt werden.

1.2.24. Gespräch erstellen

1. In Freundesliste soll für jede Person eine Zeile mit einem Button angezeigt werden. Falls ein Gespräch mit einer Person noch nicht vorhanden ist, soll ein neues Gespräch erstellt werden können.
2. In der obersten Zeile der Gesprächsliste soll ein Link zum Erstellen eines Gesprächs dargestellt werden

1.2.25. Person suchen

Erste Zeile in Freundesliste → Suchleiste → Datenbank-Abgleich mit allen Personen

1.2.26. Freunde anzeigen

Der Menüpunkt „Freunde“ soll die Freundesliste anzeigen

1.2.27. Freundschaftsanfragen

Freundschaftsanfragen sollen in der Freundesliste unter Searchbar und über bestehenden Freunden angezeigt werden, falls welche existieren.

1.2.28. Freundschaftsanfrage beantworten

Es soll ein Button in Listenzeile für Annehmen/Ignorieren angezeigt werden

1.2.29. Freund entfernen

In der letzten Zeile der Liste soll ein Button zur Bestätigung des Entfernens implementiert sein.

1.2.30. Spielfeld anzeigen

Das Spielfeld soll unter /games/play/<id> angezeigt werden. Das Spielfeld besteht aus Progress Bars. Jeder Spieler bekommt seine eigene und die der Gegner angezeigt.

1.2.31. Zug machen (Multiplayer-Modus)

Im Mehrspieler-Modus soll zu Beginn des Zuges eines Users ein Dialogfenster aufpoppen: “Du bist am Zug!” Dieses wird mittels eines Buttons “Weiter” geschlossen werden.

1.2.32. Quizfrage des Zugs

Nach dem Erreichen des neuen Feldes soll sich ein Dialogfenster mit Quizfrage öffnen, welche zufällig aus der Liste der zum Feld passenden Quizfragen ausgewählt wurde. Die Frage soll in Form einer Single-Choice-Frage erscheinen und der User soll via eines Mausklicks die passende Antwort auswählen können und mit eines “Antworten”-Buttons absenden können.

1.2.33. Ende eines Zugs

Am Ende eines Zugs soll, nach Beantworten der Quizfrage, ein weiteres Dialogfenster aufpoppen, welches anzeigt, ob die zuvor angegebene Antwort richtig war, und falls nicht, welche die richtige Antwort gewesen wäre.

2. Science-Party-FX

2.1. Autoren-Backend

2.1.1. Datenbankverknüpfung herstellen

Die Datenbank soll mittels der EbeanORM-Bibliothek hergestellt werden.

2.1.2. Methoden definieren

- Interface erstellen mit folgenden (leeren) Methoden:
- get: lädt entsprechende Objekte aus der Datenbank
- new: Erstellen neuer Objekte für die Datenbank
- delete: Löschen von Objekten aus der Datenbank
- edit: Bearbeiten der Daten aus der Datenbank
- save: Abspeichern von Werten in die Datenbank

2.1.3. Methoden bereitstellen und Werte in der Datenbank anpassen

Es sollen Methoden bereitgestellt werden und Werte in der Datenbank angepasst werden

2.1.4. Die Daten aus der Datenbank holen , Geburtsdatum in Alter umwandeln

Die Daten sollen aus der Datenbank geholt werden, das Geburtsdatum soll in Alter umgerechnet werden.

2.1.5. Boolean in der Datenbank aktualisieren

Booleans in der Datenbank sollen aktualisiert werden.

2.1.6. User aus Datenbank entfernen

Es sollen User aus der Datenbank entfernt werden können.

2.1.7. Proberly-Listen durchsuchen

Die Proberly-Listen sollen durchsucht werden.

2.1.8. Liste der QR-Codes aus der Datenbank ziehen

Die Liste der QR-Codes sollen aus der Datenbank gezogen werden.

2.1.9. Perk-Namen aus der Datenbank ziehen

Die Perk-Namen sollen aus der Datenbank gezogen werden.

2.1.10. QR-Generator aus dem Internet ziehen

Ein QR-Generator soll aus dem Internet gezogen werden.

2.1.11. Code aus Internet-Bibliothek ziehen

Codes sollen aus der Internet-Bibliothek gezogen werden können.

2.1.12. Datenbank abfragen

Geburtsdatum, Summe der Nutzer ID, Summe der Spiele ID
Summe ID der Spiel/Wochentag

2.2. Autoren-User Interface

2.2.1. Fragen hinzufügen, bearbeiten, löschen

Dies soll mittels eines Buttons realisiert werden; dazu muss eine Frage angeklickt werden, worauf sich ein Dropdown-Menü mit den Antworten aufklappt.

- neue .fxml Quiz-Datei

2.2.2. Themen hinzufügen, bearbeiten, löschen mittels Buttons

Es sollen Buttons implementiert sein, mit denen Themen hinzugefügt, bearbeitet und gelöscht werden können.

2.2.3. Bei Klick auf ein Thema werden rechts die dazugehörigen Fragen angezeigt

Es soll eine fxml Datei erstellt werden für die Liste(Dropdown) rechts und links anklickbare Themenfelder

2.2.4. Userdaten anzeigen

Vorname, Nachname, Alter, Email, Bild

- neue User .fxml Datei: Links Liste aller User, rechts Profil des Users, Bildfeld

2.2.5. User-Account Ent-/Sperren mittels Checkbox

Es soll eine Checkbox implementiert sein, mit der man User-Account sperren bzw. entsperren kann

2.2.6. Button für „User löschen“

Es soll ein Button implementiert sein, mit dem User gelöscht werden können.

2.2.7. User suchen und Ergebnisse anzeigen

Es soll ein Suchfeld mit Suchbutton implementiert sein. Es soll möglich sein, nach Vor-, Nachname und ID zu suchen.

2.2.8. Bestehende QR-Codes anzeigen

Die QR-Codes sollen angezeigt werden, links soll die Liste der Fähigkeiten und rechts der QR-Code angezeigt werden.

2.2.9. QR-Code erstellen, abspeichern und einer Fähigkeit zuordnen

neue QR-Code .fxml Datei: links Name der Fähigkeit, rechts:
QR-Code, Texteingabefeld, Druckbutton

DB:

- Abspeichern

in der GUI:

- Bereich für QR-Code zentriert auf der rechten Seite
- darunter Eingabefeld für Text
- Button "QR-Code erstellen"

2.2.10. QR-Code drucken

Ein Button beim QR-Code soll den Druckvorgang auslösen

2.2.11. Statistiken anzeigen

neue Statistik .fxml Datei

Tabelle anzeigen:

- Anzahl aller gespielten Spiele
- Anzahl aller registrierten Nutzer
- Durchschnittsalter

Grafik anzeigen:

- Spiele pro Tag im Wochenverlauf