Hochschule
Kaiserslautern
University of
Applied Sciences

Software-Technik Praktikum

Projektaufgabe

Implementierung einer *Take Me Along* - Anwendung

Prof. Dr. Jörg Hettel

24. Mai 2022

Inhaltsverzeichnis

1 Projektbeschreibung **2**

2 Allgemeine Anforderungen **2**
2.1 Funktionsumfang der Anwendung 3
2.2 Technologische Herausforderungen 5
2.2.1 Single Page Anwendung mit REST-Server 6
2.2.2 Benutzer- und Berechtigungsmanagement 6
2.2.3 Sichere Password-Verwaltung 6
2.2.4 Datenbankzugriff mit JPA unter Verwendung einer DataSource 6
2.2.5 Bild-Upload und Persistierung der Bilder in der Datenbank 6

3 Optionale Anforderungen **6**
3.1 Verhinderung von typischen Web-Angriffen 7
3.2 SSL Zugriff auf die Seite 7
3.3 PLZ-Vervollständigung 7
3.4 Containerisierung der kompletten Anwendung 7

4 Technologien und Frameworks **8**

5 Abgaben und Bewertungsgrundlage **8**
5.1 Anwendung 8
5.2 Erklärung zur Ausarbeitung 8

©Prof. Dr. Jörg Hettel

1 Projektbeschreibung

In diesem Projekt soll eine web-basierte Anwendung entwickelt werden, mit der Mitfahrgelegenheiten angeboten bzw. gesucht werden können. Das Fahrtziel ist jeweils immer die Hochschule in Zweibrücken. Ausgehend von den in der bisherigen Vorlesung entstandenen Teilfunktionalitäten, soll die Anwendung folgenden Funktionen implementieren:

Allgemeine Anforderungen:

- Integration aller Funktionalitäten
- Benutzer- und Berechtigungsmanagement
- Sichere Password-Verwaltung
- Datenbankzugriff mit JPA unter Verwendung einer DataSource
- Bild-Upload und Persistierung der Bilder in der Datenbank

Optionale Anforderungen:

- SSL Zugriff auf die Seite
- Containerisierung der kompletten Anwendung (Docker-Compose)

Die einzelnen Anforderungen werden im Folgenden näher beschrieben.

2 Allgemeine Anforderungen

Die folgenden Anforderungen sollten möglichst komplett umgesetzt werden.

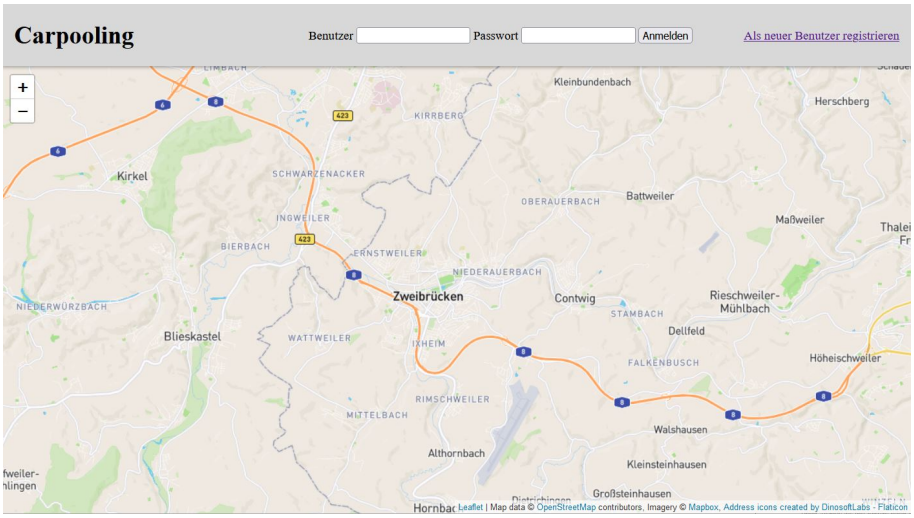

Abbildung 1: *Startseite der Anwendung.*

©Prof. Dr. Jörg Hettel

2.1 Funktionsumfang der Anwendung

Beim Start der Anwendung wird ein Kartenausschnitt von der Umgebung von Zweibrücken gezeigt (vgl. Abb. 1).

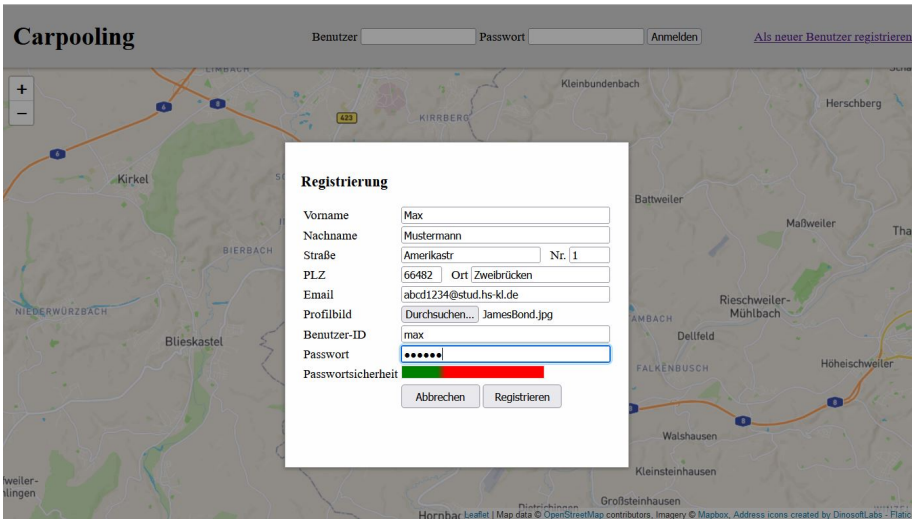

Abbildung 2: *Registrierungsdialog.*

Über eine rechts oben dargestellten Link kann ein Registrierungsfenster geöffnet werden, in dem sich neue Benutzer selbständig registrieren können (vgl. Abb. 2). Bei der Eingabe der Daten, die alle obligatorisch sind, sollen folgende Validierungen durchgeführt werden:

- Vor- und Nachnamen müssen jeweils mit einem Großbuchstaben beginnen und mit einem Kleinbuchstaben enden. Die Namen dürfen keine Ziffern oder eines der folgenden Zeichen enthalten: §,$,%,&,!,?.
- Die eingegebene Postleitzahl muss die Länge fünf (5 Ziffern) haben, wobei die 0 als erste Ziffer erlaubt ist.
- Als Email-Adressen werden nur Studentische Hochschuladressen akzeptiert.
- Die Stärke des Password soll graphisch durch einen Farbgradienten angezeigt werden. Hierbei gibt es folgende Abstufungen

  1. Passwort mit weniger als fünf Zeichen gilt als „nicht sicher” (egal welche Zeichnen benutzt worden sind)
  2. Passwort mit fünf oder mehr Zeichen gilt als „akzeptabel”
  3. Gilt 2. und werden sowohl Groß- als auch Kleinbuchstaben benutzt, gilt das Password als „mittel sicher”
  4. Gilt 3. und werden (neben Groß- und Kleinbuchstaben) zusätzlich Ziffern und Sonderzeichen verwendet, gilt das Password als „sicher”
  5. Gilt 4. und ist das Password länger als 7 Zeichen, gilt das Password als „sehr sicher”

©Prof. Dr. Jörg Hettel

Fehlt eine Angabe oder ist der Benutzername bereits vergeben, soll dem Benutzer eine entsprechende Fehlermeldung angezeigt werden. Bei erfolgreicher Registrierung werden die Benutzerdaten in der Datenbank abgelegt und der Benutzer wird eingeloggt. Später kann sich ein Benutzer direkt mit seinem Login und Password anmelden.
Erfolgreich eingeloggte Benutzer haben die Möglichkeit eine Mitfahrgelegenheit zu suchen (vgl. Abb. 3). Der eigene Standort wird durch einen schwarzen Marker auf der Karte visualisiert. Über einen entsprechenden Button kann sich der Benutzer auch wieder abmelden.

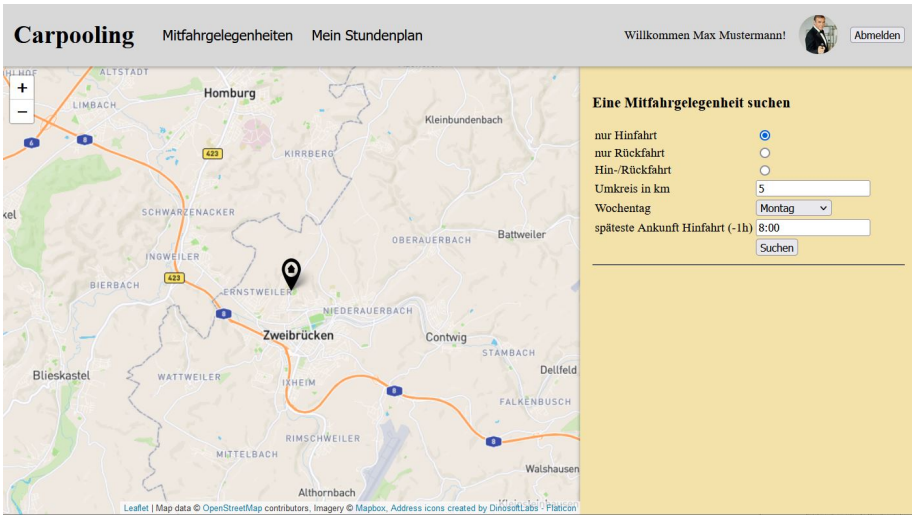

Abbildung 3: *Suche einer Mitfahrgelegenheit.*

Entsprechende Treffer einer Suche werden auf der Karte und rechts in einer Liste angezeigt (vgl. Abb. 4).

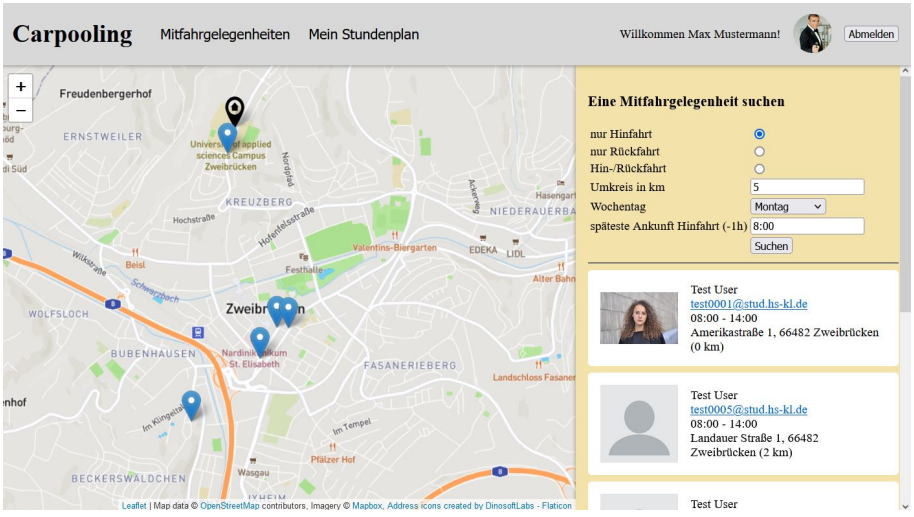

Abbildung 4: *Ergebnis einer Suche.*

Wird rechte eine Person ausgewählt, wird die Auswahl grau hinterlegt und die zugehörige

©Prof. Dr. Jörg Hettel

Position auf der Karte durch einen roten Marker angezeigt. Umgekehrt wird bei einer Auswahl eines Markers, dieser rot gefärbt und der Eintrag rechts grau unterlegt (vgl. Abb. 5).

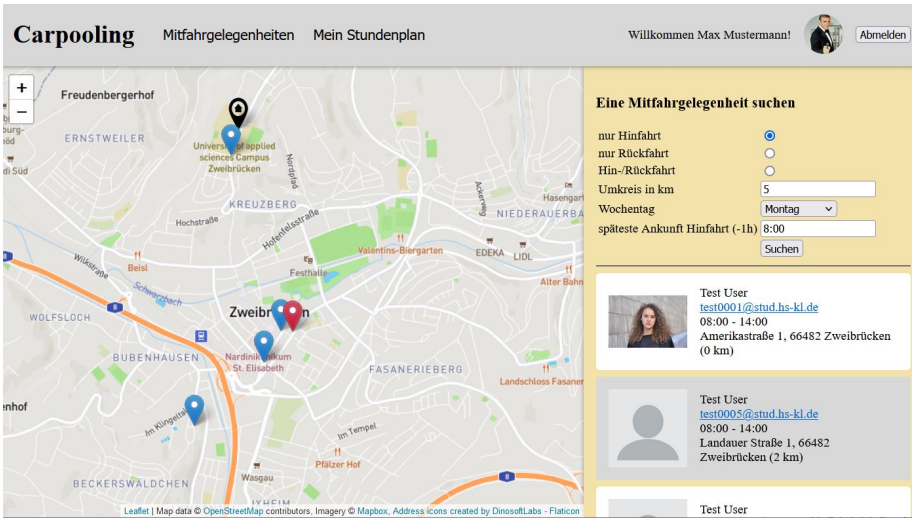

Abbildung 5: *Auswahl eines Suchergebnisses.*

Eigene Angebote können ebenfalls hinterlegt werden, z.B. auf einer eigenen Tab-Seite. Hierzu kann ein entsprechendes Wochenschema ausgefüllt werde (vgl. Abb. 6). Die Angaben können auch wieder jeder Zeit geändert werden.

Abbildung 6: *Eingabe des eigenen Angebots.*

2.2 Technologische Herausforderungen

Bei der Fertigstellung der Anwendung müssen noch verschiedene technologische Probleme gelöst bzw. integriert werden. Für manche Anforderungen muss das initiale Datenbankschema geändert bzw. erweitert werden.

©Prof. Dr. Jörg Hettel

2.2.1 Single Page Anwendung mit REST-Server

Als Architekturstil soll eine Single Page Anwendung entwickelt werden. Die komplette Kommunikation läuft nach dem Laden der Einstiegsseite über REST-Aufrufe an den Server mit Hilfe von AJAX

2.2.2 Benutzer- und Berechtigungsmanagement

Stellen Sie sicher, dass nur angemeldete Benutzer eine Suchanfrage stellen können. Hierzu müssen Sie selbst ein rudimentäres Benutzer- und Zugriffsmanagement implementieren. (In der Praxis wird man dies eher nicht tun, da es für die Microprofile-Architektur hierfür bereits fertige Lösungsbausteine gibt.)

2.2.3 Sichere Password-Verwaltung

Passwörter sollten nie im Klartext in einer Datenbank abgelegt werden. Ein übliches Verfahren ist, dass von dem Password ein sicherer Hashwert berechnet und dieser in der Datenbank abgelegt wird. Dieses Verfahren hat aber noch den Nachteil, dass gleiche Passwörter immer denselben Hashwert ergeben. Ist somit einmal ein Passwort geknackt, d.h. Password und zugehöriger Hashwert sind bekannt, kann diese Information dazu benutzt werden, bei anderen Angriffen ein Passwort leicht zu identifizieren. Dieser Angriff ist sehr vielversprechend, da Benutzer oft immer das gleiche Passwort benutzen. (Siehe hierzu auch Attacken mit Hilfe von Rainbow-Tabellen: `https://de.wikipedia.org/wiki/Rainbow_Table`).
Um solche Attacken zu erschweren (zu versalzen), wird das Hashing „randomisiert", indem noch ein zufällig erzeugter Wert mit einbezogen wird. So ergibt das Hashing eines Passwords immer einen unterschiedlichen Hashwert. Dieser zufällig erzeugte Wert (salt) muss dann mit dem Passwort zusammen in der Datenbank hinterlegt werden.
Implementieren Sie eine entsprechende Password-Verwaltung.

2.2.4 Datenbankzugriff mit JPA unter Verwendung einer DataSource

Datenbankverbindungen sollten im Server-Umfeld immer über eine Datasource verwaltet werden. Achten Sie darauf, dass die Anwendung eine beim Server eingerichtete Datasource benutzt. Benutzen Sie als JPA-Provider EclipseLink. Beachten Sie, dass für die Geo-Daten in der Datenbank benutzerdefinierte Konverter geschrieben müssen, da es kein Standard-Mapping hierfür gibt. Die Konverter müssen dann bei den Mapping-Annotations angegeben werden. Beteiligte Klassen/Interfaces sind `AttributeConverter` und `Converter`, beide aus dem Paket `javax.persistence`.

2.2.5 Bild-Upload und Persistierung der Bilder in der Datenbank

Die Porträtbilder sollen auch in der Datenbank mit verwaltet werden.

3 Optionale Anforderungen

Die folgenden Anforderungen sind optional und gehen positiv in die Bewertung des Projekts ein.

©Prof. Dr. Jörg Hettel

3.1 Verhinderung von typischen Web-Angriffen

Sichern Sie die Anwendung so ab, dass folgende Angriffe nicht mehr möglich bzw. nur noch erschwert möglich sind:

- Reflexiver Cross-Side-Scritpting Angriff, d.h. eingegeben er JavaScript-Code, muss „unschädlich" gemacht werden.
- SQL-Injection-Angriff, z.B. bei der Eingabe des Benutzernamens und Passwords

3.2 SSL Zugriff auf die Seite

Die aktuelle Verbindung zwischen Client und Server erfolgt bis jetzt ohne Verschlüsselung. Stellen Sie das Kommunikationsprotokoll so um, dass generell `https` (SSL/TLS) benutzt wird. Die notwendigen Zertifikate können Sie vorerst selbst erstellen. In einer produktiven Umgebung müssten natürlich offizielle Zertifikate verwendet werden, die von allen Browsern ohne Nachfrage akzeptiert werden.

3.3 PLZ-Vervollständigung

Auf unserem Rechner `Escher` http://escher.informatik.hs-kl.de:8080/PlzService/ wird ein PLZ-Service bereitgestellt, der für eine übergebene PLZ den oder die Ortsnamen zurückliefert. Je nach Anfrage als Liste oder im JSON-Format. Beispiel:
http://escher.informatik.hs-kl.de:8080/PlzService/ort?plz=66484
liefert:

```
66484 Battweiler;
66484 Kleinsteinhausen;
66484 Dietrichingen;
66484 Großsteinhausen;
66484 Winterbach;
66484 Riedelberg;
66484 Schmitshausen;
66484 Walshausen;
66484 Althornbach;
```

Sie können den Service nutzen, um zu überprüfen ob PLZ und Ortsnamen korrespondieren bzw.um den Ortsnamen automatisch auszufüllen.

3.4 Containerisierung der kompletten Anwendung

Neben der mySQL-Datenbank kann auch OpenLiberty in einem Docker-Container betrieben werden. Analog zur mySQL-Datenbank müssen dann auch hier Zugriffsports und Verzeichnisse für die Anwendungen (war-Dateien) „externalisiert" werden. Damit man jetzt nicht alles in einzelnen Docker-Containern starten und stoppen muss, kann die komplette Anwendungsinfrastruktur (mySQL, OpernLiberty, etc.) mit einer Docker-Compose-Datei gemanaged werden. Folgende Ausbaustufen sind möglich:

1. Betrieb von OpernLiberty in einem Docker-Container
2. OpenLiberty- und mySQL-Management über ein Docker-Compose

©Prof. Dr. Jörg Hettel

4 Technologien und Frameworks

Es dürfen keine weiteren Frameworks wie z.B. jQuery oder Bootstrap benutzt werden. Auf Client-Seite ist rudimentäres HTML 5, CSS 3 und JavaScript zu verwenden. Auf Server-Seite Microprofile ohne Zusatzkomponenten und OpenLiberty als Laufzeitumgebung. Weiter kann davon ausgegangen werden, dass die Anwendung nur mit „modernen" Browsern bedient wird, bei denen die Ausführung von JavaScript erlaubt ist.
Als Datenbank soll mySQL verwendet und mit einem Docker-Container betrieben werden.

5 Abgaben und Bewertungsgrundlage

Die Abgabe besteht der eigentlichen Anwendung (Sourcen).

5.1 Anwendung

Abzugeben ist eine lauffähige Anwendung und die entsprechenden Sourcen. Die Anwendung muss als komplette war-Datei abgegeben werden. Weiter müssen, ja nach Bedarf auch notwendige Skripte und eine kleine Anleitung (readme-Datei) abgegeben werden.
Folgende Kriterien werden für die Bewertung herangezogen:

- Funktionsumfang und Korrektheit der Umsetzung. Für das Erreichen eine sehr guten Note müssen auch optionale Anforderungen umgesetzt werden.
- Trennung von Layout und HTML und sinnvoller Einsatz von CSS. Die HTML-Seite sollte möglichst keinen JavaScript-Code enthalten.
- Klassendesign und Codestrukturierung. Der Klassenentwurf sollte gängige Designprinzipien berücksichtigen.
- Verständlicher Code (insbesondere bei JavaScript und Java). Beachten Sie die Regeln von *Clean Code*.
- Präsentation. Am Ende muss jeder Teilnehmer seine fertig gestellte Anwendung vorstellen und gegebenenfalls Fragen dazu beantworten. (Termine werden auf der OLAT-Seite veröffentlicht.)

Abzugeben ist der komplette Projektcode und alle entstandenen Artefakte.

5.2 Erklärung zur Ausarbeitung

Weiter ist eine Erklärung abzugeben, die folgenden Wortlaut hat:

©Prof. Dr. Jörg Hettel

Hiermit erkläre ich, *Vorname Nachname (Matrikel)*, dass ich die von mir abgegebenen Artefakte im Modul Software-Technik Projekt selbstständig und ohne fremde Hilfe angefertigt habe und keine anderen als in der Abhandlung oder im Source angegebenen Hilfen benutzt habe; dass ich die Übernahme von Source-Code, wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der angegebenen Artefakte gekennzeichnet habe. Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

Unterschrift

Die Abgabe der unterschriebenen Erklärung kann in Form einer Scann-Kopie erfolgen.

©Prof. Dr. Jörg Hettel