



## Web and Data Engineering, SS 2020

### „Projektarbeit: Eventplanungssystem“

#### 1. Allgemeine Aufgabenstellung

Implementieren Sie ein Eventplanungssystem. Das System soll genutzt werden um Nutzerinteresse für potentielle Veranstaltungen zu sondieren. Ihr System benötigt keine Nutzerauthentikation und soll es ermöglichen Informationen über ein geplantes Event online zu stellen, zu welchem Nutzer anonym Interesse oder Desinteresse anmelden können. Diese sollen öffentlich einsehbar sein. Ihr System kann hier Ähnlichkeiten zu vereinfachten Umfragesystemen mit binären Antworten haben.

Folgende Funktionalitäten soll Ihr System dazu für einen Nutzer anbieten:

- Erstellen eines neuen Events. Das Event muss hierbei einen Veranstaltungsort (Name oder Länge und Breitengrad), ein Veranstaltungsdatum, einen einzigartigen Namen, eine Beschreibung und einen Veranstaltungstypen haben. Der Veranstaltungsort soll hier frei gewählt werden können, das Veranstaltungsdatum darf nicht in der Vergangenheit liegen und der Veranstaltungstyp soll aus einer vorgefertigten Liste von Typen gewählt werden können.
- Falls ein Event weniger als eine Woche entfernt ist soll auf der Seite wenn möglich automatisch ein Wetterbericht zu dem voraussichtlichen Wetter angezeigt werden. Sie können dafür folgende API verwenden: <https://www.metaweather.com/api>. Falls keine Wetterdaten im nähen Umfeld bzw. für den Ort gefunden werden konnten soll stattdessen eine entsprechender Informationstext angezeigt werden.
- Ein Event soll entweder mit Interesse oder Desinteresse markiert werden können. Verwenden Sie eine geeignete Möglichkeit um Sessions zu managen. Solange das Browserfenster offen ist soll ein Nutzer seine Wahl ändern können. Sie brauchen keine zusätzlichen Sicherheitsvorkehrungen gegen Mehrfachabstimmung zu implementieren.
- Die Startseite des Systems soll die letzten 20 eingestellten Events anzeigen. Jedes Event soll hier reduziert, also zum Beispiel nur mit Namen und Datum und Beschreibungsauszug angezeigt werden. Zeigen Sie hierbei nur Events die in der Zukunft liegen. Von dieser Darstellung soll zu einer Vollansicht in eigener Seite oder Modal weitergeleitet werden können.
- **Teamgröße von 3:** Die Startseite soll außerdem ermöglichen die letzten 20 Events auf einen Typen zu filtern.
- **Teamgröße von 3:** Behandeln Sie die drei Events mit dem größten Interesse anders als normale Events. Jedes der drei Events soll entsprechend seines Ranges hervorgehoben werden (z.B. über Styling oder Icon) und außerdem in einer Top 3 Liste in der Sidebar angezeigt werden die auch vergangene Events beinhaltet.
- **Teamgröße von 3:** Das System muss eine einfache Suche nach Teilen eines Eventnamens oder nach einem Veranstaltungsort anbieten. Sie können bei der Ergebnisdarstellung die Schnelldarstellung von der Startseite wiederverwenden.

Bitte beachten Sie, dass einige der oben genannten Funktionalitäten durch Details in der Umsetzung (siehe unten) weiter spezifiziert werden.

Ihre Implementierung soll im **Spring Stack**<sup>1</sup> geschehen. Sie können eine In-Memory Datenbank (z.B. H2<sup>2</sup>) oder eine **standalone Datenbank** als Persistentschicht für die Events verwenden. Das Frontend soll in **Thymeleaf** umgesetzt werden und darf **Javascript** Funktionalität beinhalten.

Legen Sie für Ihre Projektarbeit ein **FIM-Gitlab Repository** an. Alle Komponenten der Projektarbeit können im selben Repository programmiert werden. **Das Repository muss innerhalb der ersten Woche für den Übungsleiter zugänglich gemacht werden. Schreiben Sie dafür eine Email an den Übungsleiter und laden Sie den Übungsleiter (@weissger) über GitLab als Reporter ein.** Achten Sie darauf ihren Code über ihren Account einzuchecken. Versuchen Sie **viele einzelne Commits durchzuführen** um Ihre Arbeit zu dokumentieren. Benutzen Sie aussagekräftige **Commitnachrichten** und versuchen Sie eine Funktionalität nach der anderen zu bearbeiten. **Ihr Beitrag am Projekt wird unter anderem an Git-Statistiken und Ihren Umgang mit Git festgemacht.**

## 2. Umsetzung

- (a) **Umsetzungsdetails zum Frontend** Verwenden Sie zur Frontend-Implementierung Thymeleaf<sup>3</sup> und Javascript. **Javascript Code darf direkt in den HTMLs eingebettet werden.** Definieren Sie sich Template-Fragmente zur Modularisierung sofern sinnvoll (Navigation). Folgende Punkte sind dabei zu beachten:

**Übersicht über die zuletzt erstellten Events** Die Events in den Übersichten sollen als **Teaser** angezeigt werden. Legen Sie hierbei Schwellwerte für die maximale Länge des angezeigten Inhalts fest (z.B. mittels Javascript JQuery<sup>4</sup> o. serverseitig).

**Access/Navigation Management** Auf der Detailseite soll der Benutzer angezeigt bekommen ob er in der **Session** bereits Interesse oder Desinteresse angemeldet hat. Dies kann z.B. dadurch passieren das der jeweilige Button/Link **deaktiviert und optisch verändert wird.**

**Input Validierung** Fehleingaben (z.B. leere Eingabefelder bei Titel eines Events) sollen bereits client-seitig mit Javascript abgefangen werden. Dem Benutzer wird ein **Hinweis** gegeben, wie die **Eingabe korrigiert** werden kann. Analog sollen **Hinweise** auch bei Interaktionen gegeben werden, die erst **server-seitig** fehlschlagen (z.B. Eventname bereits benutzt).

**Bonuspunkt: Footer** Inkludieren Sie einen Footer der eine **Copyright Notiz** enthält. Dieser soll auf **mobilen** Endgeräten **ausgeblendet** werden und der Content soll stattdessen über die Navigation erreichbar sein. Verwenden Sie ein Overlay (Modal) das bei erneuter Interaktion geschlossen werden soll.

Die Verwendung zusätzlicher Frameworks ist erlaubt und empfohlen (siehe **JQuery**). Das Design des Frontends bleibt Ihrer Kreativität überlassen. Versuchen Sie soweit möglich die Umsetzung nach den Responsive Design Paradigmas umzusetzen. Auch hier wird der Einsatz von Frameworks empfohlen (z.B. **Bootstrap**<sup>5</sup>).

- (b) **Umsetzungsdetails zum Backend** Das Backend soll auf Spring Framework basieren und mittels einer ausführbaren JAR bzw. WAR Datei deployed werden. Sie können hierfür den embedded Tomcat Server von Spring Boot<sup>6</sup> verwenden. Entwickeln Sie das

---

<sup>1</sup><https://spring.io/>

<sup>2</sup><http://www.h2database.com/html/main.html>

<sup>3</sup><https://www.thymeleaf.org/documentation.html>

<sup>4</sup><https://jquery.com/>

<sup>5</sup><http://getbootstrap.com>

<sup>6</sup><https://projects.spring.io/spring-boot/>

Projekt als ein **Maven**<sup>7</sup> Projekt.

*Anmerkung:* Alle für das Projekt benötigten Abhängigkeiten können über Maven eingebunden werden.

**Separation of Concerns** Implementieren Sie das Backend unter Beachtung des Separation of Concerns Designprinzips. Das Programm soll durch Entwicklung von **Service-, Controller-, Datentransferobjekts-** (DTO) und **Persistenzobjektsschicht** modularisiert werden.

**Teamgröße von 3: Veranstaltungstypen** Der Veranstaltungstyp des Events soll aus einer Liste vordefinierter Typen auf dem Server gewählt werden können. Die Liste der Typen soll beim Neustart des Servers verändert werden können ohne den Server Code zu modifizieren.

**Datenbank** Verwenden Sie eine In-Memory Datenbank mittels Spring Boot Integration. Das Schema der Datenbank soll aus Javaklassen mittels der Java Persistence API<sup>8</sup> generiert werden oder mittels einem Initialisierungsskript geladen werden.

*Anmerkung:* Sie können auch eine Postgresql oder **Mysql Datenbank** verwenden. Allgemein ist jedoch die Verwendung einer In-memory Datenbank empfohlen.

**Datenbankinitialisierung** Schaffen Sie eine Möglichkeit die Applikation bei Serverstart mit einigen (2-3) **Events zu initialisieren**. Dieser Vorgang soll auf Wunsch deaktivierbar sein.

**Datenbankanbindung** Die Daten der Applikation sollen mittels **Spring Data**<sup>9</sup> **Repositories die benötigte CRUD Funktionalität** der Datenbankanbindung umsetzen. Schreiben Sie eigene SQL Queries wo benötigt.

**Bonuspunkt: REST** Bieten Sie zusätzlich zu dem Frontend Seiten einen Endpunkt an der auf Anfrage **GET /events?n=20** ein JSON Objekt mit den Daten der letzten *n* Events anzeigt.

### (c) **Dokumentation, Setup Guide**

Fügen Sie neben Ihrer Implementierung eine **kurze Dokumentation** bzw. einen **Setup Guide** für Ihr Projekt in das Repository hinzu. Erklären Sie darin kurz, **wie Ihr Projekt zu starten und zu handhaben ist**. Beschreiben Sie dies selbst wenn Ihr Projekt bereits durch ein Starten der Spring Applikationen und dem simplen Aufruf der Index Seite funktioniert. Erklären Sie Besonderheiten in der Dokumentation. **Der Unterpunkt zur Initialisierung der Datenbank sollte hier ebenfalls beschrieben werden.**

## 3. **Abgabe und Organisatorisches**

Erarbeiten Sie zum Start des Projektes **eine Persona und ein passendes Szenario** für das System in **1-2 Seiten**. Das Dokument muss bis Montag den 08. Juni 2020 um 0:00 MESZ in das Repository hinzugefügt werden. Der Inhalt des Dokuments dient der Strukturierung Ihrer Bemühungen und fließt nicht in die Bewertung des Projekts ein. Die Abgabe ist jedoch **verpflichtend**. **Vermerken Sie ebenfalls die Teammitglieder (Name + Matrikelnummer) in diesem Dokument und laden Sie das Dokument innerhalb der ersten Woche in ihr GIT Repository.**

- Gruppengröße: 2-3 Personen
- **Abgabedatum:** Montag den 01. Juli 2020 um 0:00 MESZ.

<sup>7</sup><https://maven.apache.org/index.html>

<sup>8</sup><http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>

<sup>9</sup><http://projects.spring.io/spring-data/>

- **Abgabe:** Die Aufgabe müssen Sie mit einem FIM-GitLab<sup>10</sup> **Repository** verwalten. Sollten Besonderheiten beachtet werden, um Ihr Projekt zu bedienen, muss dies in Ihrer Dokumentation vermerkt sein (siehe Aufgabenstellung zur Dokumentation)!

#### 4. **FIM Gitlab**

Für den Zugang auf das FIM Gitlab benötigen Sie einen FIM Account. Sie können die FIM Webseite<sup>11</sup> besuchen um mehr zum Accountmanagement des FIM Accounts zu erfahren. Falls nötig können Sie OpenVPN<sup>12</sup> verwenden um sich in das Universitätsnetzwerk einzuwählen.

---

<sup>10</sup><https://git.fim.uni-passau.de/>

<sup>11</sup><https://www.fim.uni-passau.de/en/it-services/login-und-kennung/fim-accounts/>

<sup>12</sup><https://www.zim.uni-passau.de/dienstleistungen/netzwerk-und-server/netzwerkzugang/openvpn/>