Aufgabenstellung (alle Gruppen)

1 Aufgabenstellung

Konzipieren Sie eine Webanwendung zur Planung und Durchführung von Messen und **implementieren Sie einen vereinfachten Prototypen**. Mit der Webanwendung soll es möglich sein, dass

- ein Messeveranstalter die Aufteilung und Belegung der Messehallen plant
- Aussteller Standflächen buchen können
- Besucher Hallenpläne abrufen und Aussteller finden können.

1.1 Fachliche Anforderungen

Berücksichtigen Sie ferner folgende fachliche Anforderungen:

- die möglichen Ausstellungsflächen ergeben sich durch ein Raster mit Flächen gleicher Größe, in die die jeweiligen Hallenflächen eingeteilt werden
- dabei sind Verkehrsflächen (Wege, Ein- und Ausgänge, Notausgänge) zu berücksichtigen sowie Flächen für die Halleninfrastruktur (z.B. Toiletten, Restaurants, Büros der Hallenmeister etc.)
- die Mitarbeiter des Messeveranstalters erhalten nach ihrer Anmeldung zunächst eine Übersicht mit dem aktuellen Stand der Buchungen für die einzelnen Hallen
- die Mitarbeiter des Messeveranstalters können
  - die Einteilung der Hallen festlegen
  - einzelne Flächen sperren und wieder freigeben, wenn diese noch nicht gebucht wurden
  - Buchungen bearbeiten
- die Aussteller
  - erhalten nach ihrer Anmeldung eine Zusammenstellung ihrer Buchungen
  - können freie Flächen buchen oder erfolgte Buchungen stornieren (bis zum Beginn der Messe)
- die Besucher können für jede Halle einen Übersichtsplan abrufen und durch Auswahl einer Ausstellungsfläche weitere Informationen zum Aussteller abrufen
- die Besucher können Aussteller und ihre Ausstellungsflächen gezielt suchen.

Ergänzen Sie weitere Anforderungen, wenn Ihnen das fachlich geboten erscheint.

1.2 Nichtfachliche Anforderungen

Berücksichtigen Sie folgende nichtfachliche Anforderung:

- die Anwendung muss sowohl auf Desktop-Systemen als auch auf mobilen Endgeräten genutzt werden können.

2 Konzeption

2.1 Vorgehensweise

Gehen Sie folgendermaßen vor:

- analysieren Sie die drei verschiedenen Nutzungsszenarien
  - beschreiben Sie für jedes Szenario eine *Persona*
- definieren Sie die Interaktionen jeder *Persona*
  - geben Sie das Interaktionsdesign mit Zustandsdiagrammen an
  - entwerfen Sie passende Wireframes
  - spezifizieren Sie die bei den Interaktionen verwendeten Daten
- berücksichtigen Sie dabei die unterschiedlichen Anforderungen, die sich aus der Nutzung der Desktop-Varianten und der Varianten für mobile Endgeräte ergeben.

2.2 Anforderungen an die Dokumentation

2.2.1 Gliederung und inhaltliche Anforderungen

Verwenden Sie etwa folgende Gliederung:

```
1. Einleitung: allgemeine Beschreibung der Aufgabenstellung (mit Ihren Worten!)

2. Nutzungsszenario "Messeveranstalter"
   2.1 Allgemeine Beschreibung
   2.2 Benutzergruppe "Messeveranstalter": Beschreibung Persona
   2.3 Interaktionsdesign
       2.3.1 Übersicht Interaktionen
       2.3.2 ... Interaktion 1 ...
          ... Zustandsdiagramm(e) und Wireframe(s) nach Bedarf ...
          ... Erläuterungen ...
          ... Erläuterung Daten nach Bedarf ...
       2.3.x usf.
       ...

3. Nutzungsszenario "Aussteller"
   3.1 Allgemeine Beschreibung
   3.2 Benutzergruppe "Aussteller": Beschreibung Persona
   3.3 Interaktionsdesign
       3.3.1 Übersicht Interaktionen
       3.3.2 ... Interaktion 1 ...
          ... Zustandsdiagramm(e) und Wireframe(s) nach Bedarf ...
          ... Erläuterungen ...
          ... Erläuterung Daten nach Bedarf ...
       3.3.x usf.
       ...

4. Nutzungsszenario "Besucher"
   4.1 Allgemeine Beschreibung
   4.2 Benutzergruppe "Besucher": Beschreibung Persona
   4.3 Interaktionsdesign
       4.3.1 Übersicht Interaktionen
       4.3.2 ... Interaktion 1 ...
          ... Zustandsdiagramm(e) und Wireframe(s) nach Bedarf ...
          ... Erläuterungen ...
          ... Erläuterung Daten nach Bedarf ...
       4.3.x usf.
       ...
```

2.2.2 Weitere Verarbeitung

Die Dokumentation wird als utf-8 kodierter Text mit der einfachen Auszeichnungssprache *markdown* erstellt. Mit Hilfe des Werkzeugs *pandoc* (siehe Hilfsmittel) erfolgt die Umsetzung in eine HTML-Datei:

```
pandoc -f markdown -t html5 -s -c iasp1.css --toc <IhreDatei> -o <IhreHTML5Datei>
```

Die Datei `iasp1.css` enthält zusätzliche CSS-Stilregeln und wird Ihnen zur Verfügung gestellt.

2.3 Hilfsmittel

2.3.1 Erstellung Wireframes

Erstellen Sie die Wireframes mit dem Werkzeug *pencil* (siehe https://pencil.evolus.vn/). Erzeugen Sie *png*- oder *svg*-Dateien und referenzieren Sie diese im Markdown-Text. Verwenden Sie Bildunterschriften.

2.3.2 Erstellung Dokumentation

Zustandsdiagramme erstellen Sie mit dem Werkzeug *umlet* (siehe http://www.umlet.com). Erzeugen Sie *png*-Dateien und referenzieren Sie diese im Markdown-Text. Verwenden Sie Bildunterschriften.

Verwenden Sie das Werkzeug *pandoc* (siehe http://pandoc.org/) zur Konvertierung der Markdown-Datei in eine HTML5-Datei.

3 Implementierung Prototyp

Implementieren Sie die Webanwendung in vereinfachter Form serverseitig mit `Python` und clientseitig mit `HTML5`, `CSS` und `JavaScript`. Legen Sie Wert auf eine hohe Gebrauchstauglichkeit der Benutzungsschnittstelle.

3.1 Vereinfachungen

Berücksichtigen Sie folgende Vereinfachungen:

- es ist nicht erforderlich, eine spezielle Benutzerverwaltung zu implementieren; ermöglichen Sie die Unterscheidung der verschiedenen Benutzergruppen z.B. durch eine Auswahl mit einem Drop-Down-Menü im Navigationsbar
- die nichtfachlichen Anforderungen werden nicht berücksichtigt, d.h. der Prototyp muss nur den Anforderungen von Desktop-Systemen gerecht werden
- die Angaben zu Hallen und deren Rastereinteilung erfolgt mit Hilfe von Konfigurationsdateien im JSON-Format, die serverseitig vorgehalten werden.
- eine Dokumentation ist für den Prototypen nicht erforderlich
- die Datenhaltung kann serverseitig mit Dateien im JSON-Format erfolgen.

3.2 Vorgehensweise

Definieren Sie zunächst das REST-Interface. Implementieren Sie den Webserver und testen Sie das REST-Interface mit `curl`-Skripten. Erstellen Sie dann die clientseitigen Strukturen und Skripte.

Versuchen Sie, moderne Layout-Konzepte wie `Flex` und `Grid` zu verwenden.

3.3 Hilfsmittel

Verwenden Sie zur Implementierung die Hilfsmittel, die Ihnen aus der Veranstaltung WEB bekannt sind, insbesondere

- `cherrypy` zur Implementierung des Webservers mit REST-Interface
- `te/tm` zur clientseitigen Verwendung von templates
- `es` zur clientseitigen Verwendung des Publish-Subcriber-Musters
- `fetch`-API zum ansynchronen Datenverkehr zwischen Client und Server.

4 Testat

Zum Testat müssen Sie

- Ihr Konzept vorstellen und erläutern können
- Ihre Dokumentation vorlegen und deren Vollständigkeit nachweisen
- Ihren Prototypen vorführen und erläutern können.