

Terminkalender

JSF-Projekt

Lukas Wigger & Elin de Wit

Aufgabenstellung/Ziel

Das Ziel dieses Projektes ist es eine Objektorientierte Web-Applikation gemäss der Dokumentation (siehe Kap. Ausgangslage & Erweiterungen) zu realisieren.

Das Projekt muss bis Zum 30.05.2023 fertiggestellt werden.

Ausgangslage:

In einem Gebäude eines Unternehmens sind verschiedene Räume verfügbar. Diese Räume werden für Sitzungen und Veranstaltungen genutzt. Eine Webapplikation soll die Verwaltung der Räume und der Termine bzw. Reservationen unterstützen.

Die Anwendung bietet mit einem Link oder einem Knopf auf die Hauptseite (Facelet index.xhtml) die Möglichkeit einen Raum zu reservieren. Dazu wird kein Benutzerkonto benötigt. Klick man auf "Reservationen erfassen", wird ein Formular angezeigt, in das der Organisator folgenden Informationen eintragen muss. Kein Feld darf leer geblieben.

| Datenfeld | Datentyp | Bemerkung | XHTML-Element |
|---|---------------|--|--------------------|
| Datum | Date | TT.MM.JJJJ Datum in der Zukunft | |
| Von | Time/String | HH:MM | |
| Bis | Time/String | HH:MM | |
| Zimmer | int oder byte | 101, 102, 103, 104, 105 | h:selectOneListbox |
| Bemerkung | String | 10 - 200 Alphanumerische Zeichen | h:inputTextarea |
| Liste von Teilnehmern: Eine oder mehrere Vor- und Nachnamen | String | Nur kleine und grosse Buchstaben [A-Z][Aa-Zz] Trennzeichen: Komma | h:inputTextarea |

Nachdem der Termin angelegt ist, erhält man eine Bestätigung und zwei Schlüssel bzw. Reservationscodes. Mit einem Code (*Private*) kann der Benutzer die Reservation ändern oder löschen. Mit dem anderen Schlüssel (*Public*) kann der Benutzer die anderen Teilnehmer einladen, z.B. per E-Mail mit dem Schlüssel und der Adresse der Seite.

Den private bzw. public Schlüssel kann man auf der Startseite in einem Textfeld eingeben und wird dann zu einer entsprechenden Seite weitergeleitet, die entweder zur Terminbearbeitung (private) oder zur Einsicht (public) führt.

Es gibt eine Liste mit allen erzeugten Terminen, die mittels eines Arraylist implementiert ist und nicht permanent (z.B. in einer Datenbank) gespeichert wird.

Erweiterungen:

Termin check 🛑 - Nicht Implementiert

Bevor ein neuer Termin erstellt und in die Webapplikation eingetragen wird, sollte eine Überprüfung stattfinden, um sicherzustellen, dass die Reservation möglich ist. Dabei wird überprüft, ob das Zimmer noch verfügbar ist und ob die angegebenen Datum- und Zeitangaben ("Von" und "Bis") sinnvoll sind, d.h. ob sie im gültigen Zeitrahmen liegen und logisch sind.

CSS  - Erledigt

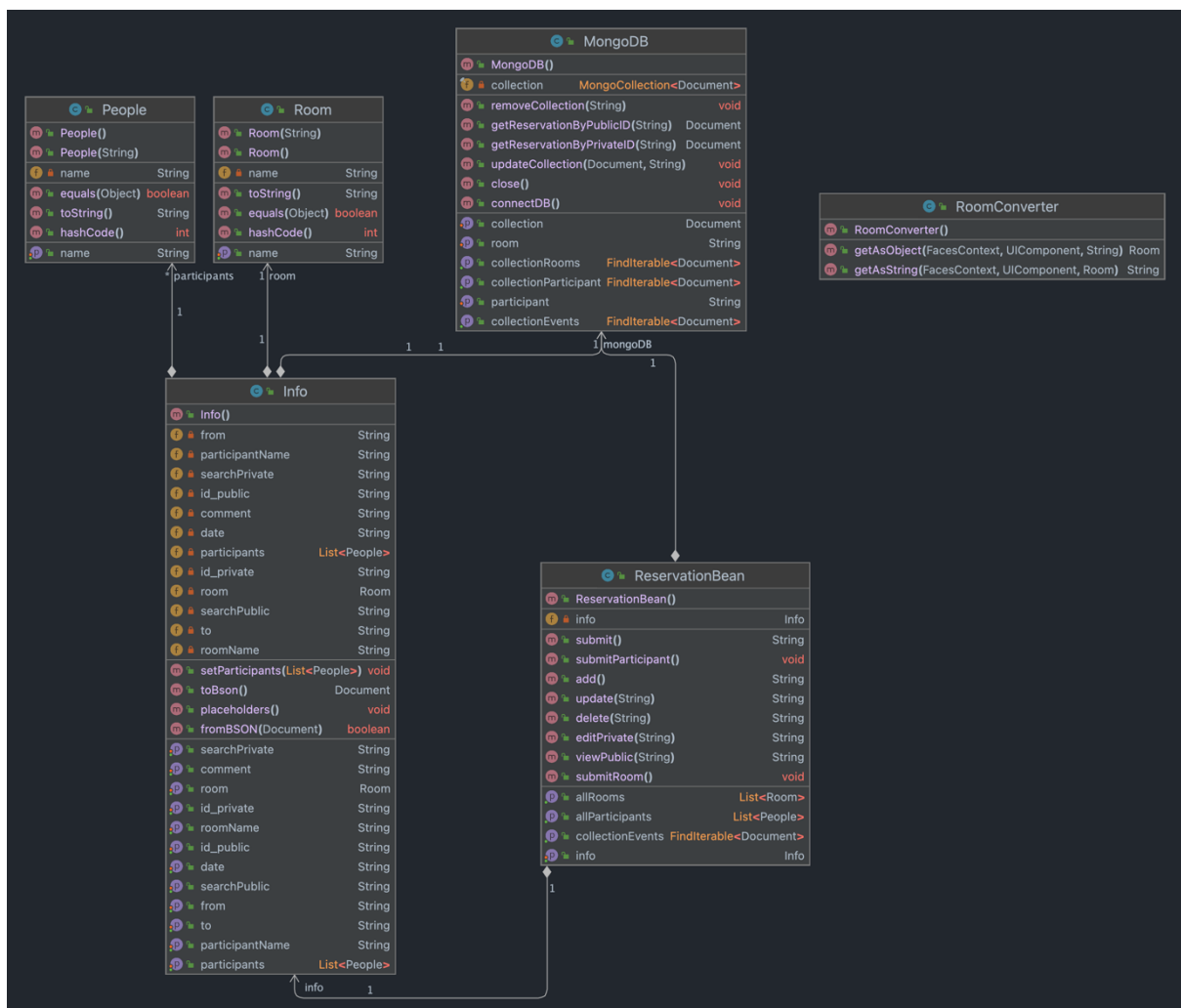
Zusätzlich zu diesen funktionalen Aspekten sollte die Webapplikation auch über CSS-Styling verfügen.

Persistente Datenspeicherung  - Erledigt

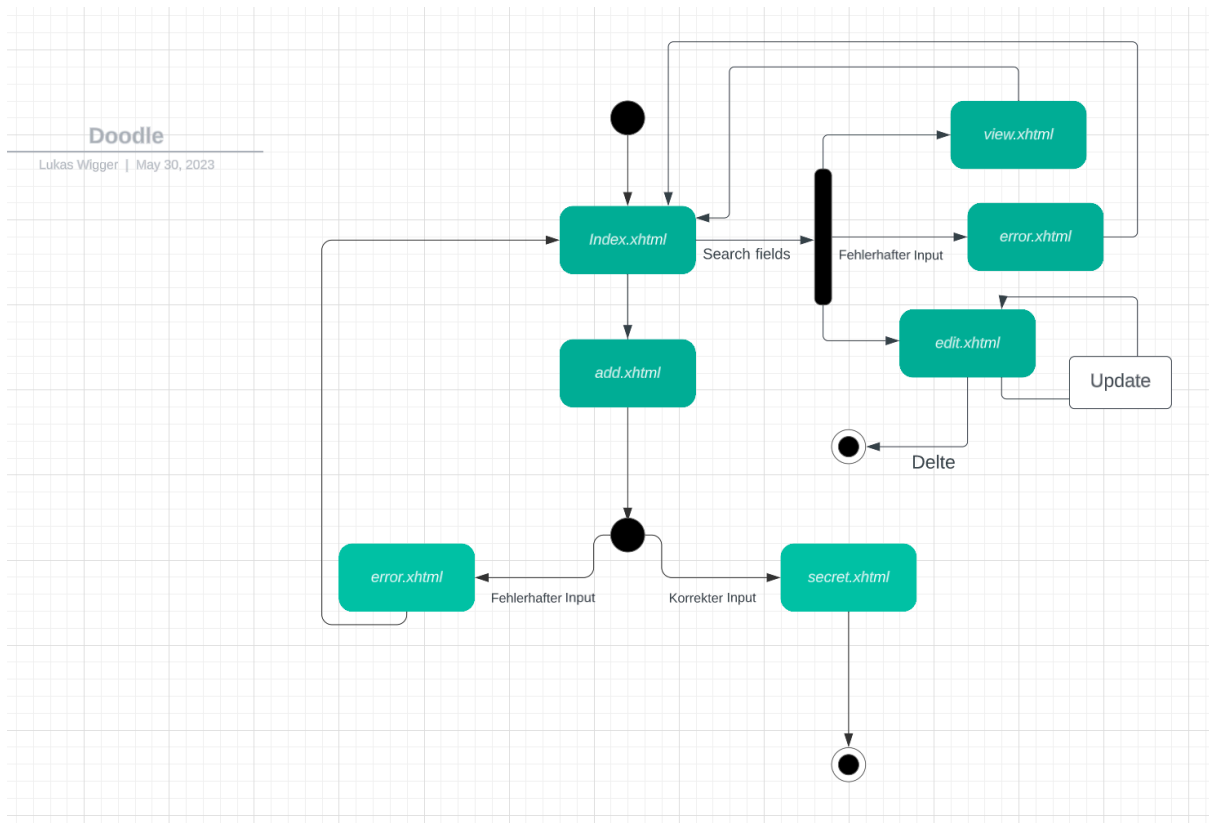
Um die Funktionalität der Webapplikation zu erweitern, sollte auch eine persistente Datenspeicherung implementiert werden. Dies bedeutet, dass die eingegebenen Informationen und Reservierungsdaten dauerhaft gespeichert werden, um sicherzustellen, dass sie auch nach dem Schließen der Anwendung oder einem Neustart des Systems verfügbar bleiben.

Diagramme:

UML-Klassendiagramm:



Zustandsdiagramm – Tutorial



Technologies/Ressourcen:

In diesem Projekt nutzen wir folgende Software bzw. Ressourcen:

JSF: Jakarta Faces ist ein Framework-Standard zur Entwicklung von grafischen Benutzeroberflächen für Webanwendungen. Basierend auf Servlets und JSP-Technik, gehört JSF zu den Webtechnologien der Jakarta EE.

Wildfly: Den Applikationsserver für die Servlets

MongoDB: Die Datenbanktechnologie, welche wir verwenden, um die Web-App zu entwickeln.

Tailwind CSS: Das CSS-Framework welche wir verwenden, um die Web-App zu entwickeln.

IntelliJ IDEA: Die Entwicklungsumgebung, die wir nutzen, um das Projekt zu realisieren.

Code:

Den Code kann man unter dem folgenden Link abrufen: <https://gitlab.com/LukasW01/m223-terminkalender-doodle>