

Reservation System M223

Dokumentation

Modul 223: Multiuser-Applikationen
objektorientiert realisieren

Finale Version / 20.06.2025

Team: Vladimir Zivanovic und Christian Hübner

Table of Contents

1. Einleitung

2. Projektbeschreibung

3. Technische Umsetzung

3.1 Entity-Relationship-Diagramm (ERD)

3.2 UML-Zustandsdiagramm

3.3 UML-Klassendiagramm

4. Testdaten

5. Reflexion

1. Einleitung

Das Projekt „Terminkalender“ ist eine Webanwendung, die im Rahmen des Moduls 223 entwickelt wurde, um die Verwaltung von Raumreservierungen in einem Unternehmensgebäude zu unterstützen. Die Anwendung ermöglicht es Benutzern, Räume für Sitzungen und Veranstaltungen zu reservieren, ohne ein Benutzerkonto zu benötigen. Sie bietet Funktionen zur Erfassung, Bearbeitung und Einsicht von Reservationen, die in einer MySQL-Datenbank gespeichert werden.

2. Projektbeschreibung

Die Webanwendung „Terminkalender“ ermöglicht die Verwaltung von Raumreservierungen mit folgenden Hauptfunktionen:

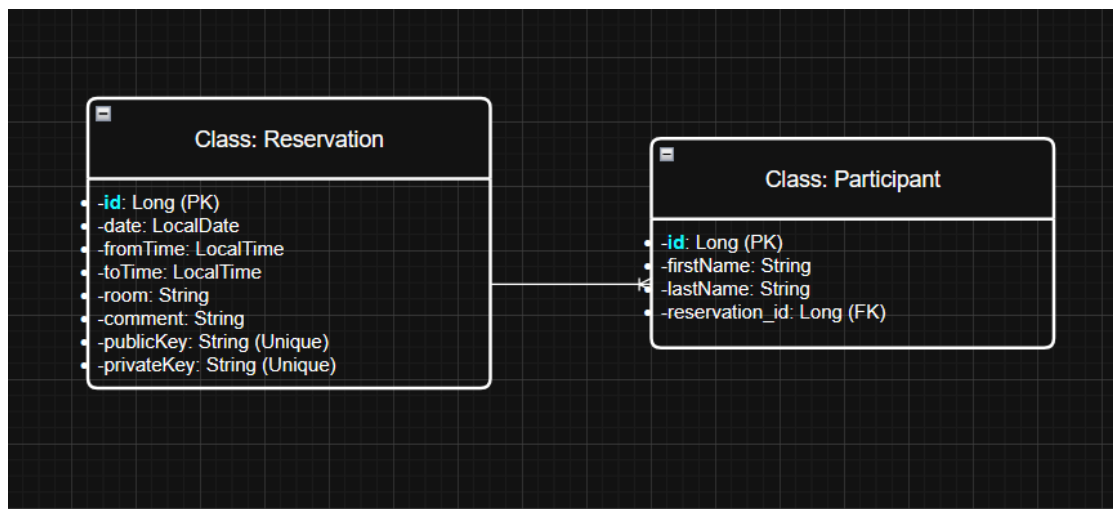
- Erstellung von Reservationen mit Angaben zu Datum, Uhrzeit, Raum, Bemerkung und Teilnehmern.
- Überprüfung der Raumverfügbarkeit vor Erstellung einer Reservation.
- Generierung von öffentlichen und privaten Schlüsseln für die Einsicht und Bearbeitung von Reservationen.
- Anzeige aller Reservationen in einer Liste.

3. Technische Umsetzung

Die Anwendung wurde mit Spring Boot und Thymeleaf entwickelt. Die Datenbank ist eine MySQL-Datenbank, die über Spring Data JPA angesprochen wird. Das Projekt verwendet Maven als Build-Tool und ist auf GitHub versioniert.

3.1 Entity-Relationship-Diagramm (ERD)

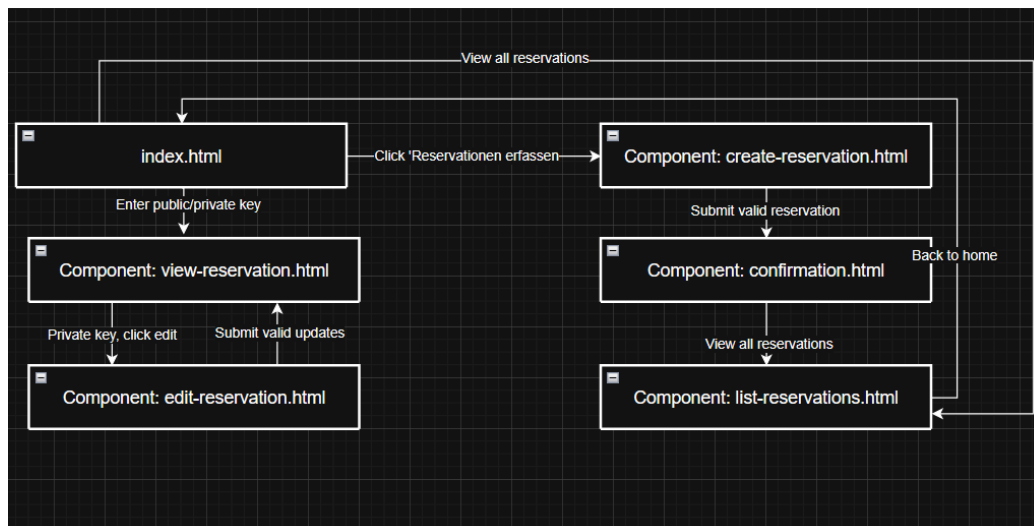
Das ERD beschreibt die Datenbankstruktur mit zwei Entitäten: **Reservation** und **Participants**.



3.2 UML-Zustandsdiagramm

Das Zustandsdiagramm zeigt den Navigationsfluss:

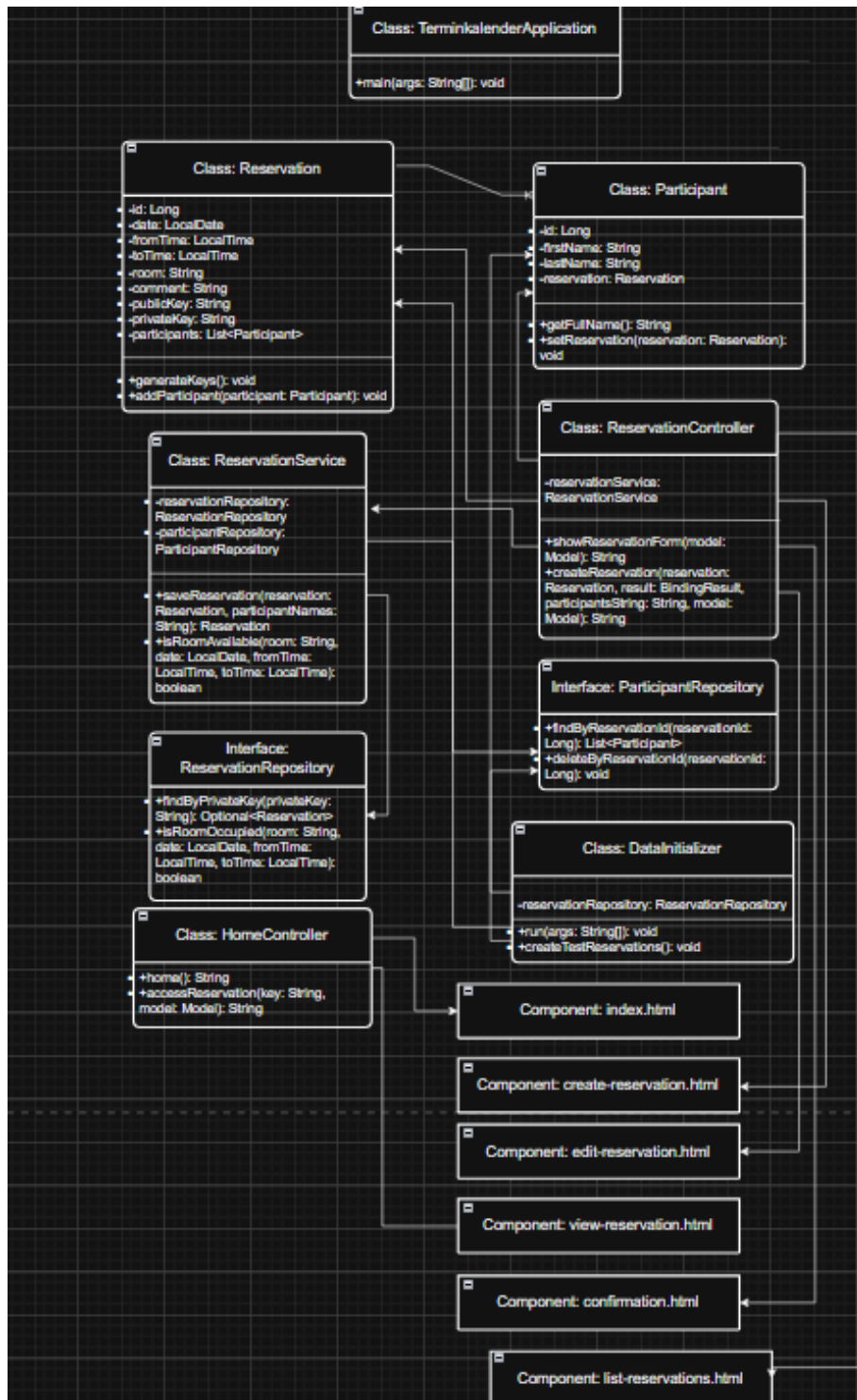
- Seiten: Home (index.html, initial), Create Reservation, Confirmation, View Reservation, Edit Reservation, List Reservations.



3.3 UML-Klassendiagramm

Das Klassendiagramm beinhaltet:

- Alle Klassen
- Alle Interfaces
- Alle Components
- Relations zwischen Components, Interfaces, Klassen.



4. Testdaten

Die Anwendung initialisiert zwei Testreservierungen.

Eingaben:

Reservation 1	Reservation 2
Raum 101	Raum 102
Datum in 7 Tagen	Datum in 7 Tagen
09:00-11:00	09:00-11:00
Max Mustermann, Anna Schmidt	John Doe, Jane Smith

Ausgaben:

Reservation 1	Reservation 2
Reservation Confirmed!	Room is not available at the specified time.
Datum in 7 Tagen	
Public Key: 6c237c72-615b-4939-b08e-ee0c0f027588	
Private Key: 74058869-f78a-401f-a51a-0787a316c474 (Keep this secure!)	

5. Reflexion

-Vladimir Zivanovic (Teamkoordinator): Als Teamkoordinator war ich für die Koordination unseres Schulprojekts „Terminkalender“ verantwortlich. Ein Problem war, dass wir anfangs Schwierigkeiten hatten, uns auf einen einheitlichen Zeitplan zu einigen, was die Zusammenarbeit verzögerte. Wir haben das gelöst, indem wir eine klare Aufgabenverteilung erstellt und wöchentliche Treffen abgehalten haben. Ich bin richtig glücklich mit dem Ergebnis – wir haben eine funktionierende Anwendung entwickelt, wo ich sehr stolz darüber bin!

-Christian Hübner (Bug Tester): Als Bug Tester habe ich das Terminkalender-Projekt getestet und bei Problemen geholfen. Ein Challenge war, dass wir anfangs viele Fehler in der Raumverfügbarkeitsprüfung entdeckt haben, was uns Zeit kostete. Ich habe das mit gezielten Tests und enger Zusammenarbeit mit dem Team gelöst, um die Bugs zu beheben. Ich bin superstolz auf uns – die Anwendung läuft jetzt stabil, und ich habe viel über Fehlerbehebung gelernt!