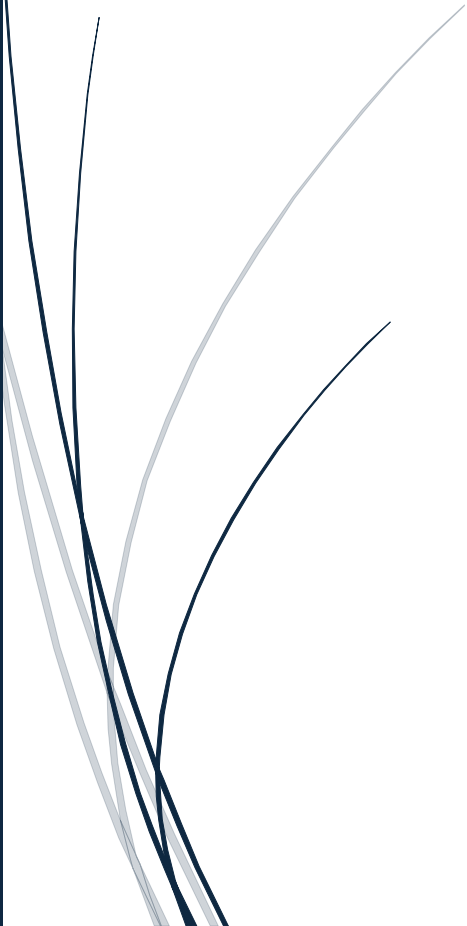


18/06/2025

Projektarbeit M223

Dokumentation



Falivene Dario, Satapathy Chandra, Brys Natalia
BZRA IMST

Änderungshistorie

Version	Datum	Beschreibung
0.1	14.05.2025	Erstellung des Dokuments und des Aktivitätsdiagramms
0.2	28.05.2025	Erstellung des UML-Klassendiagramms, der Projektbeschreibung und ERD
0.3	18.06.2025	Erstellen des Fazits
1.0	18.06.2025	Vorbereitung des Dokuments für das Review

Inhaltsverzeichnis

1. Projektbeschreibung	3
2. Aktivitätsdiagramm	3
3. UML-Klassendiagramm	5
4. Entity Relationship Diagram (ERD)	6
5. Fazit.....	6

1. Projektbeschreibung

Dieses Dokument enthält die Dokumentation zum Projekt OrariApereti für das Modul 223. Das Projekt beinhaltet die Möglichkeit, Zimmer zu reservieren, sowie Reservierungen anhand der vorhandenen Schlüssel auszulesen und zu ändern. OrariApereti ist ein Mehrbenutzer-Raumreservierungssystem, das mit Spring Boot (Java) für das Backend und React (TypeScript, Vite) für das Frontend entwickelt wurde.

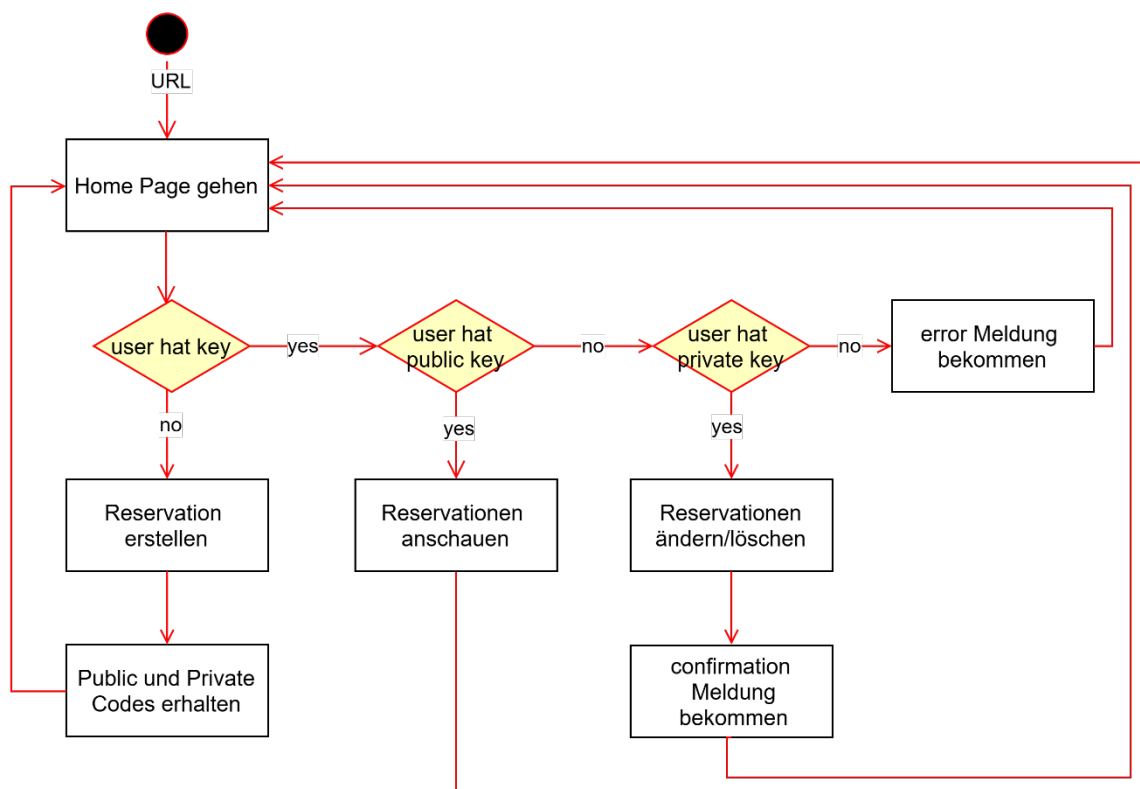
Noch kurz zur Code Struktur:

Im [Root-Verzeichnis](#) liegt eine [README.md](#), die im Wesentlichen die gleichen Informationen enthält wie diese Beschreibung (Die Lizenz unter welchem diese Projekt publiziert wird, wird dort noch erwähnt).

Im [Unterordner OrariApereti](#) befinden sich zwei weitere Ordner: [Backend](#) und [Frontend](#). In jedem dieser Ordner liegt eine eigene README.md ([Backend](#) & [Frontend](#)), die erklärt, wie man den jeweiligen Teil ausführt und worauf man achten sollte (application.properties und .env Datei).

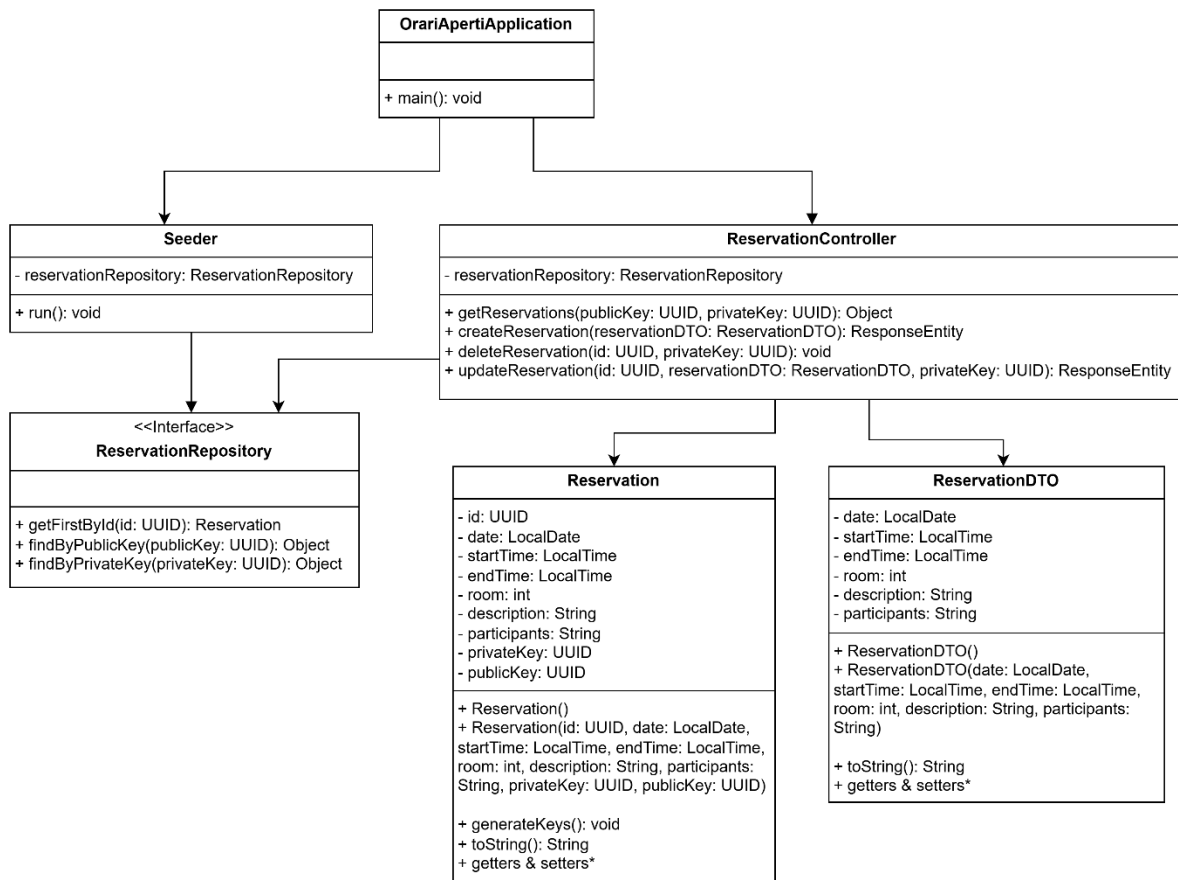
Alle Dokumentationsdateien sowie die meisten «.drawio» Dateien für die Diagramme befinden sich im [Doku Ordner](#), wie es im Projektbescrieb vorgeschrieben ist.

2. Aktivitätsdiagramm



Das Diagramm zeigt den Ablauf, mit dem ein Benutzer je nach seinem Schlüssel (Public Key oder Private Key) Reservierungen auf der Startseite erstellen, anzeigen, ändern oder löschen kann. Sollte der Nutzer noch keinen Schlüssel besitzen, kann er eine Reservierung vornehmen und erhält beide Schlüssel.

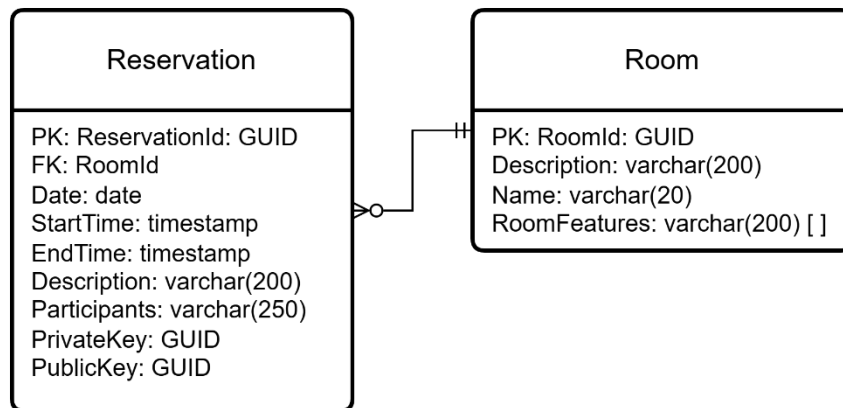
3. UML-Klassendiagramm



* Get- und Set-Methoden für alle Attribute der Klasse

Während des Entwicklungsprozesses hat sich das Klassendiagramm ein wenig verändert. Jetzt ist "Room" eine Klasse. In der Klasse "Reservation" beziehen wir uns daher auf das "Room" als Klasse und nicht als int. In der Klasse „Room“ sind die Attribute: id (UUID), roomNumber (String) und roomFeatures (array). Room Features ist ein Enum.

4. Entity Relationship Diagram (ERD)



Wir haben zwei Tabellen in der Datenbank: eine für Reservierungen und eine für Zimmer. Beim Anlegen einer Reservierung muss ein Zimmer, das sich in der Datenbank befindet, ausgewählt werden. Ein Zimmer kann mehrere oder gar keine Reservierungen haben.

5. Fazit

Wir konnten das Wissen, das wir im Node.js-Modul 248 erworben haben, erfolgreich mit den Inhalten dieses Moduls verknüpfen und anwenden. Im Rahmen des Projekts haben wir ein Java-Spring-Backend (REST API) entwickelt und dieses erfolgreich mit einer modernen Frontend-Webanwendung auf Basis von React, TypeScript, Vite, Tailwind und Shadcn verbunden.

Als mögliche Weiterentwicklungen des Projekts wären unter anderem eine Login-/Signup-Funktion mit Benutzerkonten beziehungsweise könnte man Reservierungen mit Konten verknüpfen. Auch die Integration eines Kalenders zum Reservierungen verwalten wäre eine sinnvolle Erweiterung (Zum Beispiel den Export eines Events im ICAL/ICS Format).