

Homework 3

Exercise 1:

1. The room booking service shall provide an overview of all available rooms in the selected university building.

Functional

No description on how the overview should look like. Specify more what the service shall look like (e.g. list, calendar, per room view etc.)

2. The overview shall mark the room that are occupied based on the date and time selected by the user.

Functional

How should it mark those rooms? Coloured highlighting? How does the User select the date and time? Drop-down menu or similar?

3. Room bookings shall be updated in real-time, that is: if a user is looking at the overview and another user at the same time books a room, the overview shall be updated as fast as possible.

Quality (Performance)

“as fast as possible” and “real-time” is to inaccurate. Instant updates are almost never possible. So, what are “acceptable” timings, 5 minutes, 5 seconds etc?

What should happen if two users book the room at the same time? Should it already be displayed when another user is currently selecting the room?

4. Only logged-in users shall be able to book rooms.

Quality (Security)

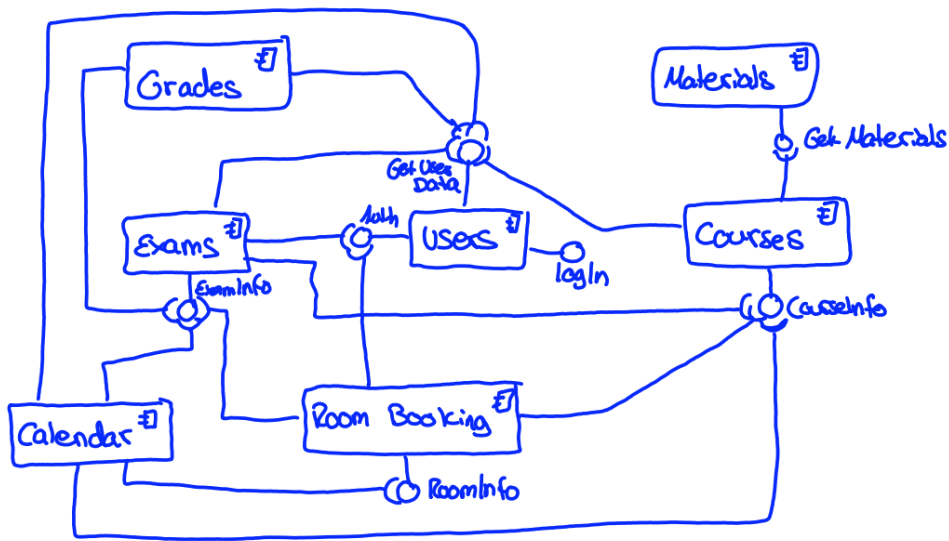
Should not logged in users be able to see room bookings and only logged-in users are able to book rooms? Or should not logged-in users also not see booked rooms? How does the user Log-In to the service?

5. An initial version of the service should be ready after 5 months and consist of a simple text-based approach that can be tested by a small group of people.

Project Requirement

A precise date would be nice. Maybe a little bit clearer what should be exactly included in the initial Version.

Exercise 2:



Get User Data : Gets the saved user data

Auth. : Perform authentication

ExamInfo : Gets the Exam informations

Room Info : Gets the information of booked Rooms

Course Info : Gets course information

Get Materials : Gets materials

Login : Performs the login

Exercise 3:

Task a)

TC 1: grade1: 1.0, grade2: 2.0, grade3: 0.0, return: 1.5, exception: -

TC 2: grade1: 3.0, grade2: 1.0, grade3: 2.0, return: 1.5, exception: -

TC 3: grade1: 1.0, grade2: 3.0, grade3: 0.0, return: -, exception: ThirdExaminerNeedException

TC 4: grade1: 1.1, grade2: 2.0, grade3: 0.0, return: -, exception: NoSuchGradeException

TC5: grade1: 1.0, grade2: 0.0, grade3: 0.0, return: -, exception: NoSuchGradeException

TC 6: grade1: 2.0, grade2: 5.0, grade3: 5.0, return: 5.0, exception: - (this Test fails)