ScheDool

Classroom Reservation Application

Members: Poon, Beam, May, Win **Date: 01 Sep 2025**

Problem / Pain Point

CMKL University employs a huge set of competencies providing a wide range of subjects for a diverse set of students. In implementing these competencies into the real world, several challenges arise, one of them is classroom scheduling. CMKL staff are tasked with creating a schedule for a semester, they need to find appropriate rooms for most, if not all, competencies. Given that there can be as many as fifty competencies per semester, this can be a serious challenge. In addition to avoiding conflicts, such as, scheduling the same room for two different competencies at the same time, this task may require finding a room with specific attributes, for instance, the attribute of capacity, that is, how many students a room can accommodate or availability of special equipment.

Capabilities

- The application database must include details of all available rooms at CMKL. s
- Add a new room to the database as a candidate location, with capacity and other attributes.
- Mark an existing room as unavailable from a start date to an end date (for instance, if it is under renovation).
- Maintain information about all room reservations (competency/#students/day of week/startdate/time/repetition)
- Edit a reservation - detect and avoid conflicts
- Delete a reservation
- Create a new reservation and select a room. The system must detect conflicts due to time or group size.
- Create a new reservation and ask the system to recommend a room based on time and group size.
- Search for a particular competency and display its schedule.
- Display the schedule for a specific week - might have multiple classes in the same time slot, which should be in different rooms.
- Display the schedule for a specific room for a specific week
- Export the weekly schedule or room-specific schedule to a PDF or text file.

Requirements

Must Have

1. The application **database** must store details of all rooms available at CMKL.
2. Ability to add new rooms to the database as potential locations, including capacity and other relevant attributes.
3. Reservation Management:
   - ○ 3.1 Keep track of all room reservations, including competency, number of students, day_of_the_week, start_date, time, and repetition.
   - ○ 3.2 Allow **creation** of new reservations and selection of an appropriate room.
   - ○ 3.3 Detects and prevents scheduling **conflicts**.
4. Modify Reservations:
   - ○ 4.1 **Delete** reservations.
   - ○ 4.2 **Edit** reservation details.
5. Display the weekly schedule, showing multiple classes in the same time slot if they are in different rooms.

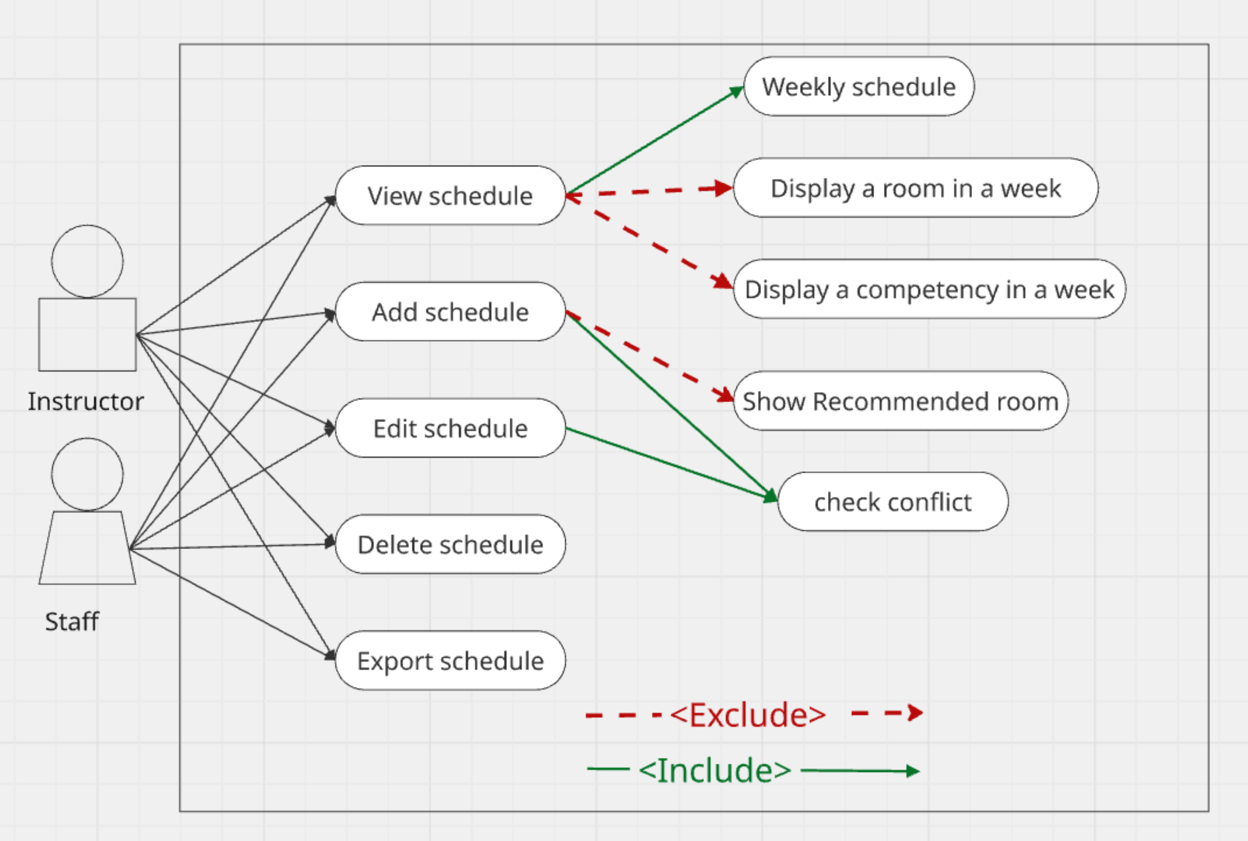
Nice to have

6. Schedule Search & Display (Display with Constraints):
   - ○ 6.1 Search for a specific **competency** and display its schedule.
   - ○ 6.2 Display the schedule for a specific **room** for a specific week.
7. **Export** weekly schedules or room-specific schedules as PDF or text files.

Blue Sky

8. When creating a reservation, provide a room **recommendation** based on availability and group size.

Use Case Diagram



Use Case Narrative

Use Case Name: Add schedule
Actor: Instructor
Goal: Adding new room reservation to schedule
Precondition: The room must existed in database
Main success scenario:

1. System displays a weekly schedule.
2. System asks for schedule details and constraints (competency, group size, timeslot and repetition).
3. User enters the schedule details and constraints (competency, group size, timeslot and repetition).
4. System check for existing competency in schedule
5. System checks for available rooms from the user timeslot.
6. System filters available rooms based on user constraints and display them.
7. User selects an available room.
8. System marks the selected room and timeslot as unavailable for the scheduled period.
9. System displays new user schedule

Extensions (a):

6a. System filters available rooms based on user constraints and display them.
7a. System recommends a room for the user.
8a. User selects a recommended room.
9a. System marks the room unavailable for the scheduled period.

Extensions (b):

4b. System checks for existing competency in the schedule.
5b. System finds existing competency in the schedule.
6b. System displays the existing competency in the schedule.
7b.1. User cancels the **Add schedule** user case.
7b.2. Return to step 2 of the **Main success scenario** (re-enter schedule details and constraints).

Postconditions: The system must mark the selected room and timeslot as unavailable.