**System Requirements:**

**General**

**-**The system must be object-oriented.

-Must provide: *safety, liveness and fairness***.**

-Maintain a directory of rooms (each room will have a timeslot)

-The system must handle potentially large amount of users.

-The system must display the status of all the rooms

***Exclusion and concurrency control***

-Only *one user* can write the room at all time

-Timeslots that are booked must have a *waitlist*.

-Upon the cancellation of a reservation, the next person gets the timeslots.

-This action cancels all other reservation (for this user) for this timeslot.

-A waiting list will be implemented as a queue

-Modification of a reservation must be considered as **write** operation.

-Only one writer can be active at a time.

-Displaying the status of the room can be considered as a read operation.

-Access to a room that is being written to must be blocked from the other users.

-Multiple readers are allowed.

**Software requirements**

-A room can be booked multiple times up until a limit (weekly allowance). It can be consecutive (right now 3 1 hour time slots a week).

-There is a maximum an amount of registration per week.

-At the beginning of the week all the queues will be set to 0.

-You can only reserve time slots for the current week.

**Features**

The user needs to be able to:

-Identify and authenticate

-Reserve facilities

-Cancel a reservation

-Modify a reservation

-View created reservations

-If two users arrive at the waiting list at the exact same time the software will flip a coin

-The wait list will have a limit (for now 3)

-Number of reservations is limited to three time slots

-Reminders/notifications (send an email reminder a few hours before)

*Notifications*

The app could notify the users if the room for which they were on a waitlist was liberated.

***Optional***

*Room suggestions*

The app could suggest a room to reserve based on availability or preferences.

*Favorites List*

The users could have the option to add rooms to a list of favorites.

**\*Research the time slots reservations for each room (ex: Library, Grey Nuns etc.): Tentative 1 hour time slots**

**Temp Class Definition**

*Room*

- Attributes

-Location (String)

-Reservation

-Wait List (Data structure tbd)

-*Synchronized* Busy (Boolean)

…

-Methods

…

*Users*

-Attributes

-username (String)

-password (String)

-reservations (Reservation Data structures tbd)

-Methods

…

*Reservation*

- Attributes

-Owner

-

-Methods

-Add

-Cancel

-Modify