**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 access the room at all time

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

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

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

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

-Only one writer can be active at a time. (Similar to previous point about rooms)

-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.

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

**Features**

The user needs to be able to:

-Identify and authenticate

-Reserve facilities

-Cancel a reservation

-Modify a reservation

-View created reservations

*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.

**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