**Summary about the project:**

We need to construct a system which would allow users (e.x: students) to reserve a room at different time slots. In case the room is already reserved, all incoming reservation requests will be placed in a waiting list. Once the requested room becomes available for a specific time slot, the system would allocate the room to the user who is been waiting (e.x: queue, linkedlist) and remove the corresponding entry from the waiting list.

Using the project description, I highlighted the conceptual classes in yellow and the actions or system events in green.

The system **maintains** a directory of rooms. Each room may be **reserved** at a number of time slots. The users of the system are already registered with the facility. Multiple users can access the system in order to reserve the facilities. Alternatively, users can access the system in order to **cancel** an existing reservation, **modify** an existing reservation, or merely to **view** reservations. The system **maintains** a registry of all reservations. Before performing any task, users must be **identified** and **authenticated** by the system.

The system allows multiple users to access its registry in order to view its status (all availabilities and reservations), but only one user may access a room at any given time in order to create, modify or cancel a reservation. A user who wants to reserve a time slot for a room that is already booked at that time can be **placed** on a waiting list and be able to obtain the room upon cancellation of the current reservation. Upon obtaining such reservation, the user is **removed** from any and all other waiting lists on any other room that has been reserved over the same time slot.