Issues (with descriptions)

When writing issues, think:

- what each role can do, and what they can’t do (constraints)

- what happens before and after an event/action/story is completed

- how an user action affects others / other roles

1. Book sports equipment -- MUST

* Only available when capacity is not yet reached
* Only available between 16:00-23:00
* Sports equipment for rent are volleyball nets, soccer balls, hockey sticks, basketballs and more
* Every equipment has a name and a sport related to it
* Booking sports equipment is free if you already have a timeslot
* Target audience: User

Register for lessons in pre-allocated time slots -- MUST

* Time slots between 09:00-16:00
* Before: A certain time slot is available to register for. After: A time slot cannot be registered for anymore when capacity is reached.
* Target audience: User

1. Reserve the sport fields/halls -- MUST
   * Time slots 16:00-23:00
   * Each time slot must be one hour. They start every full hour, so the first time slot will be from 16:00-17:00 and so onwards until the sport center closes.
   * Sport halls/fields all have a unique name, a minimum and a maximum capacity, so you need to have at least the min capacity and not more than the max capacity to reserve a sport hall/field.
   * There are three sport halls named: X1, X2 and X3
   * Every sport field is related to a certain sport
   * Target audience: User
2. Subscribe to a basic subscription -- MUST
   * Only one reservation per day
   * Target audience: User
3. Subscribe to a premium subscription -- MUST
   * Up to 3 reservations per day
   * Target audience: User
4. Login -- MUST
   * Username must be a unique string ID
   * Unique ID string comes with password
   * Target audience: User
5. Create an account -- MUST
   * Have a unique string ID
   * Make own password
   * Target audience: User
6. Monitor who used the material/equipment last -- SHOULD
   * show history of people that used the equipment
   * Target audience: Administrator
7. Make a reservation for an entire team -- SHOULD
   * Team size should be between min and max capacity of field to reserve
   * Only a team can make a reservation for a team sport (e.g football)
   * Target audience: User
8. Create a team -- SHOULD
   * Create a group consisting multiple members
   * Booking will be limited to lowest member’s subscription. Since group reservations count towards the maximum amount of reservations per day for every team member. Therefore if one of the team members has reached its max number of reservations for that day, the entire team would not be able to reserve anymore.
   * Target audience: User
9. Cancel reservation -- COULD
   * Users are able to cancel their own reservation
   * Target audience: User
10. Add a sports field/hall -- COULD
    * Admins are able to create a new sports field/hall
    * Target audience: Administrator
11. Remove a sports field/hall -- COULD
    * Admins are able to remove an existing sports field/hall
    * Target audience: Administrator
12. Modify field/hall capacity -- COULD
    * Admins are able to change the capacity of any existing fields/halls
    * Target audience: Administrator
13. Add equipment -- COULD
    * Admins are able to add new equipment
    * Target audience: Administrator
14. Remove equipment -- COULD
    * Admins are able to remove an existing equipment
    * Target audience: Administrator

**Non-Functional Requirements**

1. System should be built modular
2. System should allow for easy integrations with other systems
3. System’s individual components need to be scalable
4. System should be written in Java version 11
5. All interactions with the systems are handled through the APIs
6. System must be built with Spring Boot and Gradle
7. Security should be implemented by using Spring Security
8. User’s password must be stored safely in the database
9. System should be built using a microservice architecture