

REQUIREMENTS ROWING - Team 31c

CSE2115 Software Engineering Methods of the Computer Science curriculum at the Delft University of Technology.

CONTENTS

1. Functional Requirements

1.1. Must Haves

1.2. Should Haves

1.3. Could Haves

1.4. Would/Won't Haves

2. Non-functional Requirements

1. FUNCTIONAL REQUIREMENTS

For the scenario rowing, the requirements regarding functionality and service are grouped under the Functional Requirements. Within these functional requirements, four categories can be identified using the MoSCoW model for prioritizing requirements:

1.1 MUST HAVES

- All users of the system need to be able to authenticate themselves.
- Users must be able to create an account with a unique ID and password.
- Users must be able to specify the time slots they are available.
- Users must be able to see what activities are available in their specified slots and enroll in them.
- Users are able to choose one or multiple of the 5 roles in a rowing team (cox, coach, port side rower, starboard side rower, sculling rower) they can fill. If the user chooses the position “cox”, then they should also provide their certifications.
- Users who have the cox role can only enroll in activities which require boats their certification allows them to steer.
 - A user with with a 8+ certificate can steer a C4, a 4+, or an 8+ boat
 - A user with with a 4+ certificate can only steer a C4, or a 4+ boat
 - A user with with a C4 certificate can only steer a C4 boat
- When a user is an owner of the activity, they must be able to create, edit and cancel the activity.
- Activity owner has to accept/decline if somebody enrolls.

1.2. SHOULD HAVES

- When a user wants to participate in a competition they must indicate their gender, organization and whether they are a competitive or amateur rower in their account.

- The system should also be functioning for competitions.
- Certain competitions should be available only for competitive rowers – should have level for rowers
- Users should not be able to match with a training if the training starts within a half hour or the competition starts within a day.
- Activity owner should be able to participate in their own activity.
- Users must be notified if they were accepted for the activity.

1.3. COULD HAVES

- A user that has enrolled in an activity could be able to cancel after they were accepted.
- A user could be notified in case they get declined to join an activity.
- More boats (other than C4, 4+ and 8+) could be available.
- A user should not be able to enroll in activities during times that they are already accepted/enrolled for another activity

1.4. WOULD/WON'T HAVES

- The system shall have a graphical user interface (GUI)
- A user shall be able to edit or cancel any activity that they did not create
- A user shall be able to add new certificates other than C4, 4+ or 8+

2. NON-FUNCTIONAL REQUIREMENTS

The system should be written in the Java programming language (version 11).

The system must be built with Spring Boot (Spring framework) and Gradle.

The system should provide security using Spring.