Backlog - Requirements Engineering

1 Stakeholders

- *User* - All users of our application, they are able to create boards using an id/key and modify it.
- *Admin* - A user that can unlock password protected boards, remove or add boards, and restart the server.

2 Terminology

Board Refers to the main screen of our application, contains lists which in turn contain entries.. 1

Entry Refers to a task added to the board, organised on the vertical axis under a specific list.. 1

List Are on the horizontal axis, encapsulate entries.. 1

3 The objective

We are building, as an overview, a personal task list organizer. The main page is a board that contains lists and these lists, in turn, contain cards that can be added, removed and moved between the lists.

INTENTIONALLY LEFT BLANK

4 Requirements

4.1 Mandatory for passing

1. Create a Client-Server Application (Using Spring Boot and JavaFX)
2. The boards must persist in a database, so the server should be connected to a database.
3. The ability to create a new board.
4. The ability to join a board created by another user.
5. The ability to add/remove/edit cards and lists.
6. The ability to pick a username before entering a project (no registration needed).
7. An intuitive interface, options and buttons (summarize titles + extra simple description where needed).
8. The ability to have the tasks arbitrarily ordered.
9. The ability to have all progress synchronized between users of the same board.
10. The ability to drag and drop tasks.
11. The ability to have the tasks easily recognizable based on their title.

4.2 Should be implemented

1. The ability to join multiple boards using a key and a password and have them show up on the main page.
2. The ability to specify the server I want to connect to.
3. The ability to have password protected boards (will need to implement some hashing).
4. The ability to change a board from a public one to a password protected one.
5. Generation of SHA-1 identifiers for boards.
6. The ability to have details on cards such as a description.
7. A nested task list.
8. The ability to add different tags to an entry to classify the task.
9. The ability to customize boards and cards (Add a background, colors for tags, choose font color and size).
10. The ability to use easy keyboard shortcuts to facilitate use of the application.
11. The ability to add sub-tasks for tasks.
12. The ability to add media to cards (Pictures, icons and attachments).
13. The ability to edit tasks and sub-tasks directly from the board and to have them highlighted when editing.
14. The ability to easily cycle through tasks while editing them by virtue of the ”tab” key.
15. The ability to set a board to a different mode (read-only, read-write).

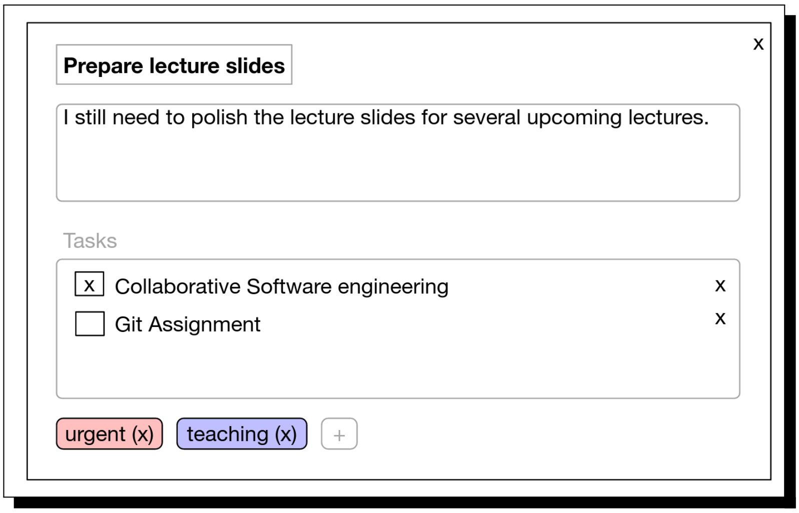
4.3 Could be implemented

1. A list of all boards.
2. Board analytics such as: cards created, activity per week, users connected.
3. Optional deadlines and progress bars on tasks.
4. Filtering the tasks based on names and/or tags (optionally using fuzzy searching).
5. Board history to see how a board has changed over time, with the authors of these changes.

4.4 Do not implement

1. User management.
2. Race condition remedies.

5 Mocks



Prepare lecture slides

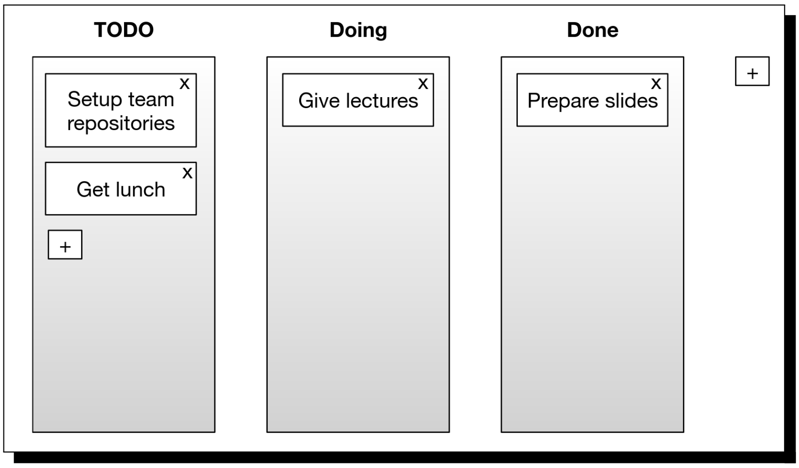
I still need to polish the lecture slides for several upcoming lectures.

Tasks

☒ Collaborative Software engineering

☐ Git Assignment

Figure 1: A mock task



TODO

Setup team repositories

Get lunch

Doing

Give lectures

Done

Prepare slides

Figure 2: A mock of the board