

Backlog - Team 50 Cents

Stakeholders

User: Any person that is able to interact with the app and the board(s).

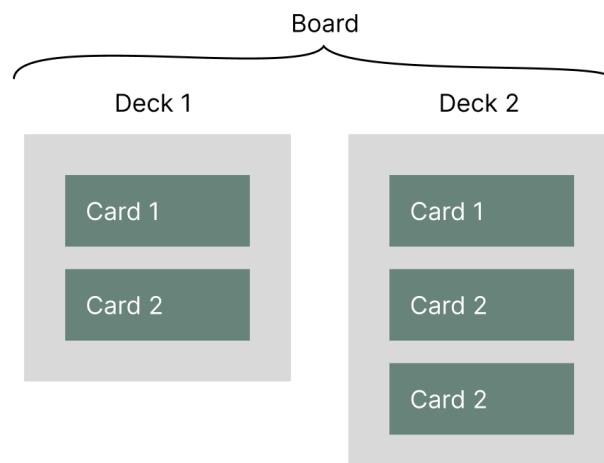
Admin: A person that manages the server, e.g., starts the server locally or is able to edit the boards themselves.

Terminology

Card: A draggable container with a title and other optional fields/properties.

Deck: A list containing multiple cards.

Board: A horizontal list of decks.



Epics

Minimum Viable Product

This epic needs to be completed to provide minimal usable functionality for a collaborative task management application. This would provide basic functionality, such as creating and managing boards with cards. The user must be able to add and edit cards, move them between decks. The application should be user-friendly and accessible, even for users with little technical expertise.

As a **User**, I want to ...

- to log in to the application with a name without having to register, so that I can quickly access my board(s) while also making sure that I am recognizable by the other collaborators.
 - Acceptance criterion: The feature is considered finished whenever it is implemented as shown in **[Figure 6]**
- to add decks to the board, so that I can organize my tasks into groups.
- to add a card to a deck in which I can describe my task.
 - Acceptance criterion: The feature is considered finished whenever it is implemented as shown in **[Figure 1]**
- to edit the title of a card, so that I can update the name of my task.
- to drag and drop a card, so that I can reorder it within a deck.
- to drag a card to a different deck, so that I can change the category of my task.
- to delete a card, so that I can remove tasks that are no longer relevant.
- to see an automatically synchronized view of the board, so that I can collaborate with other users in real time.
- to delete a deck, so that I can remove categories that are no longer relevant.

As an **Admin**, I want ...

- to be able to restart the server, so that I can restart it in case of an error.
- to be able to delete all data from the database using the application's interface, so that I can start fresh.
- to be able to remove password protection
- to be able to see all the boards
- to verify myself as an admin with a password without exposing any noticeable marks in the application of the admin panel, so that regular users don't want to try to log in. (This should be achieved using a custom keyboard shortcut while logging in, to show the admin password prompt.)

Non-functional requirements

- the application should use Spring Boot as the backend
- the application should use JavaFX for the front end
- the application should use a database for persistent storage
- adhere to the created mockups where applicable (can be found in appendix)

Multiple Boards

This epic focuses on enhancing the task management system by allowing users to create multiple boards, each with a unique permanent key, enabling separation of unrelated decks and easy sharing among team members.

As a **User**, I want ...

- to be able to create multiple boards, so that I can organize my decks in groups.
- to delete a board when it is no longer needed.
- each board to have a permanent unique key that can be used for sharing the board with others.
 - Acceptance criterion: The feature is considered finished whenever it is implemented as shown in **[Figure 3]**
- to have an overview page where I can see all the boards I have opened recently.
 - Acceptance criterion: The feature is considered finished whenever it is implemented as shown in **[Figure 2]**
- to be able to change the name of the boards when the focus of the board changes.

Non-functional requirements:

- The recently opened boards should only be saved on the client side, so that without the keys, no one could access the boards.

Card Details

This epic focuses on providing customization of cards, with the option to add a description and a list of subtasks with checkboxes. The user should be able to add, remove, reorder, and edit subtasks, as well as modify the card's details whenever necessary. This functionality will allow users to track progress and keep everyone up-to-date on the status of each task.

As a **User**, I want ...

- to add a description to a card, so that I can provide more context for a task.
- to add subtasks to a card, so that I can break down a task into smaller pieces.
- to mark subtasks as completed, so that I can track my progress on a task.

- to edit the details of a card, including the description and subtasks, so that I can update information about a task.
 - Acceptance criterion: The feature is considered finished whenever it is implemented as shown in **[Figure 5]**
- to reorder subtasks within a card, so that I can prioritize the order in which I complete them.
- to be able to pin or highlight the cards, so that I can prioritize and organize the cards.

Tags

This epic aims to enrich the task management system by implementing a comprehensive tagging system, enabling users to categorize their tasks and prioritize their workflow. The system will allow for multiple tags to be added to a card, with options to remove and view all tags at a glance.

As a **User**, I want ...

- to add tags to a card, so that I can categorize and group similar tasks together.
- to add multiple tags to a card, so that I can assign different categories to a task.
 - Acceptance criterion: The feature is considered finished whenever it is implemented as shown in **[Figure 5]**
- to see the tags displayed on the board view, so that I can quickly identify the category of a task.
- to remove tags from a card, so that I can update the categorization of a task.
- to prevent adding the same tag twice to a card, so that I can avoid duplicates.
- to rename a tag, so that I can update the name of a category for consistency and clarity.
- to delete a tag, so that I can remove a category that is no longer needed.
- to create a new tag, so that I can assign a new category to a task that does not fit any existing tags.

Non-functional Requirements:

- The tags should be separate entities in the database, so that we can later rename them or add a color (preparing for epic: Customization).

Customization

This epic focuses on improving user engagement and satisfaction by providing a personalized experience. The system will allow users to set a custom background image, choose from a selection of default images, or upload their own, enhancing the overall aesthetic and appeal of the interface.

As a **User**, I want ...

- to select a background image for a board, so that I can personalize my experience.
- to choose from a list of default images for my account background, so that I can easily select a pre-designed option.
- As a user, I want to upload a custom image for my account background, so that I can use an image that I like or that represents me.

Keyboard Shortcuts / Accessibility

This epic entails the implementation of efficient keyboard shortcuts, improving user productivity and accessibility. The system will provide smooth navigation between items using tab and arrow keys, alongside quick actions such as creating a new empty board, selecting items, and moving them around.

As a **User**, I want ...

- to be able to navigate the application using the keyboard, so that I can easily move between items without using the mouse.
- to use the tab key to navigate between different items on the page, so that I can quickly move to the next focusable element.
- to use the arrow keys to navigate between different items on the page, so that I can easily move up, down, left, or right within a section.
- to be able to open any menu without clicking, so that I can quickly access the options that I need.
- to use the Ctrl+N keyboard shortcut to create a new empty board, so that I can quickly start a new project.
- to select items on the page using either the mouse or the arrow keys, so that I can use the navigation method that suits me best.
- to move the selected item using the alt + arrow key shortcut, so that I can quickly reposition items on the page.

Password Protected Boards

This epic is designed to enhance the security and privacy of the task management system by implementing password-protected boards. This will enable read-only access for unauthorized users, requiring a password for write access, ensuring that sensitive information remains confidential.

As a **User**, I want ...

- to be able to set a board to be password-protected, so that others can only view them in read-only mode unless they enter the password too.
 - Acceptance criterion: The feature is considered finished whenever it is implemented as shown in **[Figure 4]**
- to be prompted for a password when I try to make changes to a password-protected board, so that I can verify that I have the correct permissions.

Awareness (bonus)

This epic focuses on making visual feedback for collaborating users of each other's actions, so that they can naturally resolve conflicts and don't try to make parallel changes on a specific property.

As a **User**, I want ...

- to see whether a card is selected by someone, so that I won't try to edit the card.
- to see when a card is being dragged by another user, so that I won't drag it at the same time too. (I also shouldn't be able to do so.)
- to see the names of the active users on the board, so that I can identify them.

Mockups

Additional acceptance criteria: The final product needs to adhere to the following mockups.

Figma prototype:

<https://www.figma.com/file/8aQj1IL0iKCsJvph3CbiYx/OOPP-Todo-list?node-id=59%3A16&t=4V6Tgzcl5iM6ZeLm-1>

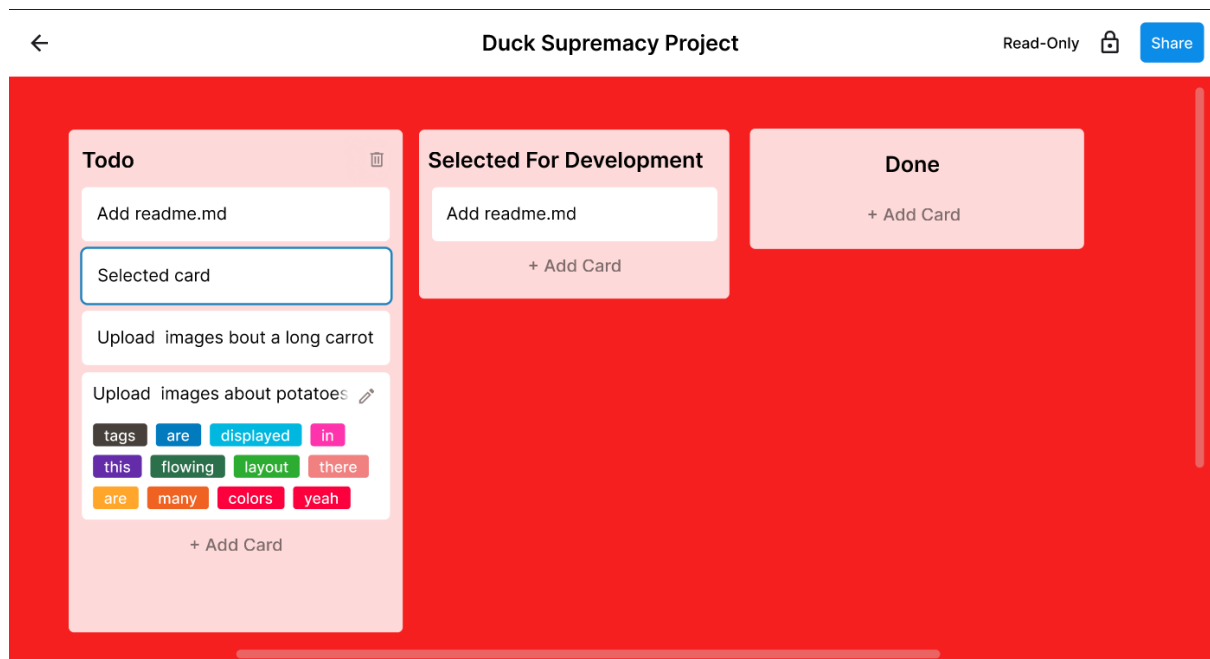


Figure 1. Board view

Boards

Join With Key...

+ New Board

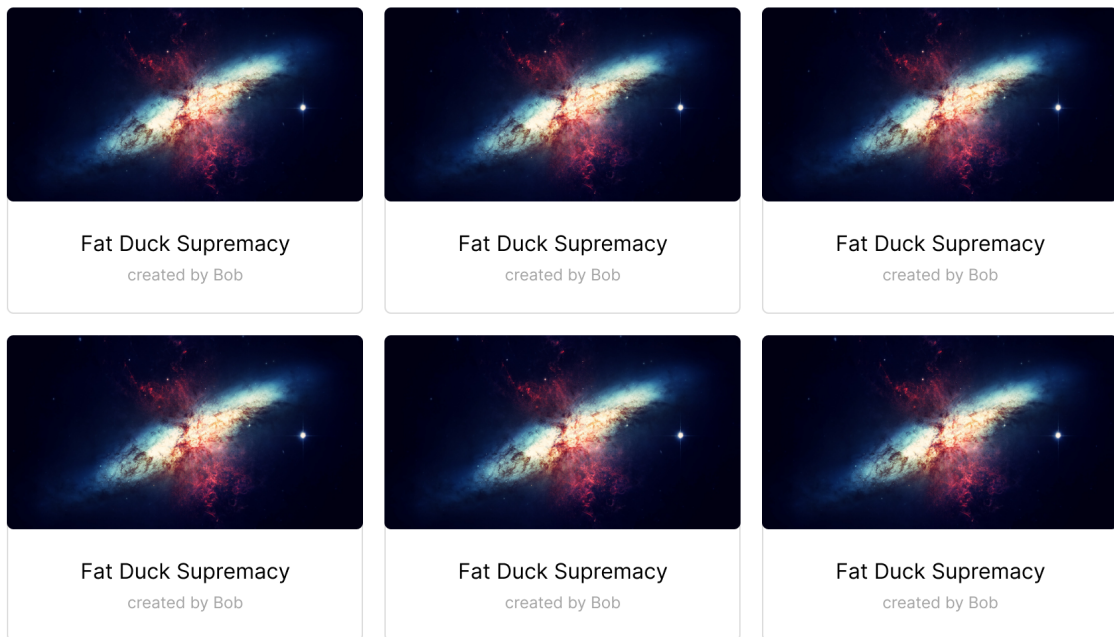


Figure 2. Overview

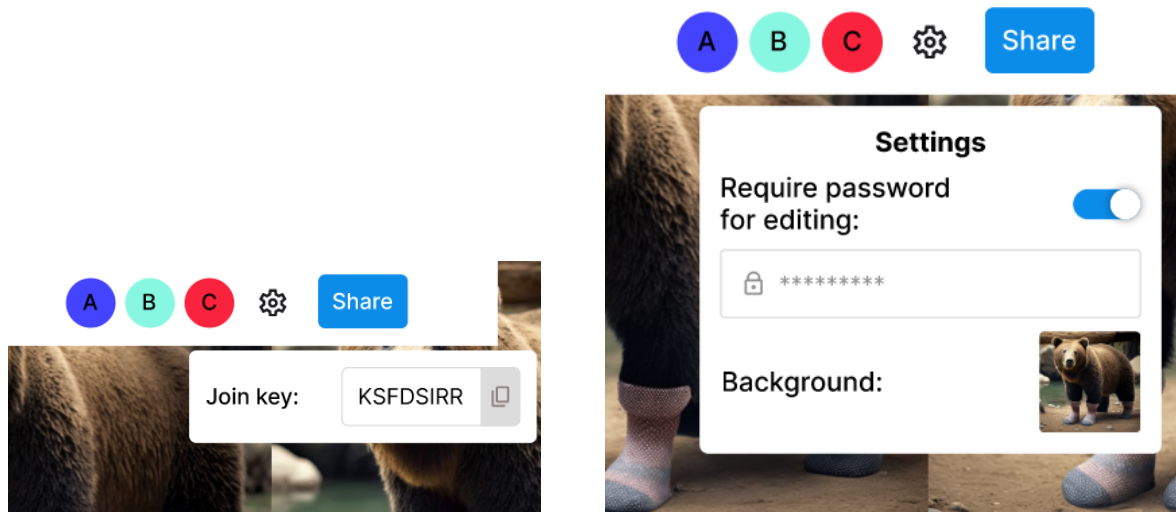


Figure 3. Sharing with key

Figure 4. Board settings

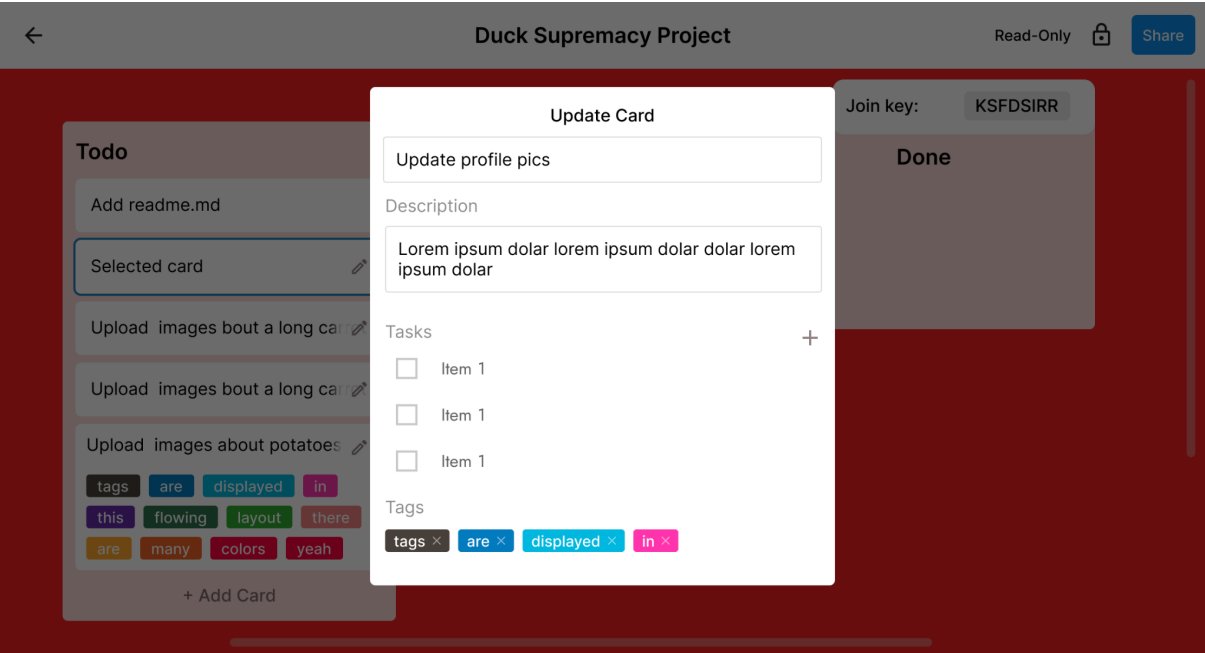


Figure 5. Card details

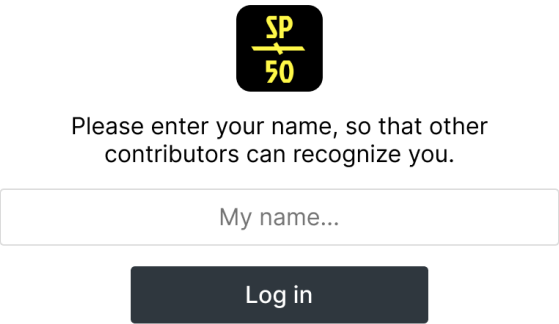


Figure 6. Login screen