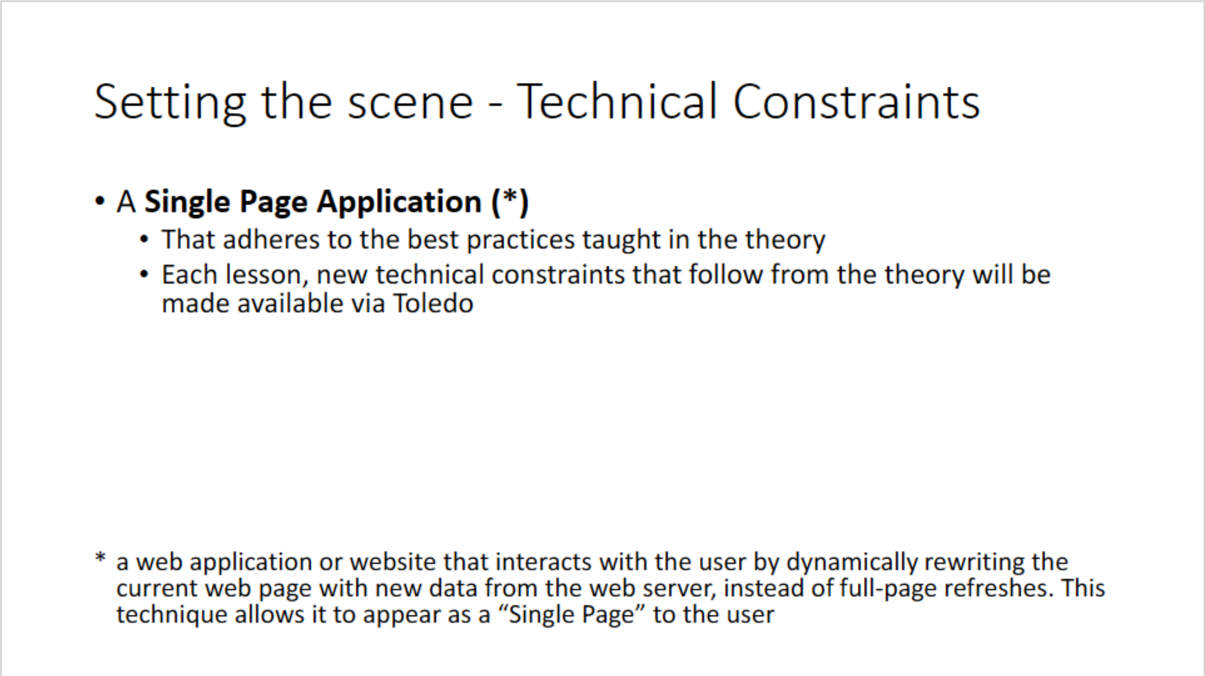
Project Analysis

Demo: https://youtu.be/NtFSYFIAhLo

Regarding Project Requirements & Constraints:

Due to how we approached the project and implemented it to be a complete product on its own, we have gone outside of the expected scope, and hence it is either not possible to meet certain requirements, or too late to change everything to meet certain requirements.

Firstly I would like to bring into attention: In Full-Stack Software Development Toledo, in Introductions. 02-Intro-Project.pdf. Page 5 out of 7.



Setting the scene - Technical Constraints

- A **Single Page Application (*)**
  - That adheres to the best practices taught in the theory
  - Each lesson, new technical constraints that follow from the theory will be made available via Toledo

* a web application or website that interacts with the user by dynamically rewriting the current web page with new data from the web server, instead of full-page refreshes. This technique allows it to appear as a "Single Page" to the user

It is due to this slide that we have worked to have all the content displayed on the index page, with not a single other page file aside from the Discord OAuth2 handler, which immediately redirects back to index after parsing and updating data.

Hence the second Front-end requirement: 2. There is at least 1 dynamic route. Could not be met, unless I were to start all over again.

Furthermore, I have asked during a class if we were allowed to use Discord OAuth2 to handle login, to which you replied yes, but user authentication must still happen.

Hence, considering we get our user data from discord api, we do not have a login form. Nor do we save passwords, meaning we could not meet the 1st requirement of Security.

Besides which, 5th requirement of Security: There exists at least 3 roles in the domain. We use permissions instead of roles, and since these can change in session, using JWT to store permissions would not function, unless we were to generate a new JWT token with each permission change. However, all front-end components vary display based on said permissions.

We have 3 predefined users, discord accounts created using burner emails. There are two created servers owned by different users. They have no additional roles.
You have free reign to create and change roles and assign them to the other users to experiment with the permission based display and functionality changes.

We have decided not to use a seed.ts as we hope the lecturers interact with our application extensively to go over a myriad of possible permission combinations.

Extra notes: I believe that having KanbanCord use discord api to login instead of creating a new account and password is more futureproof, considering many find it a hassle to create new accounts for every new site or application they visit. Instead connecting using an existing account is far easier to manage and more user friendly.

Some interesting stuff that happened:

https://www.youtube.com/watch?v=8gk_L-TrXs8

https://youtu.be/LDkh3gOSZXc

Project Pitch:

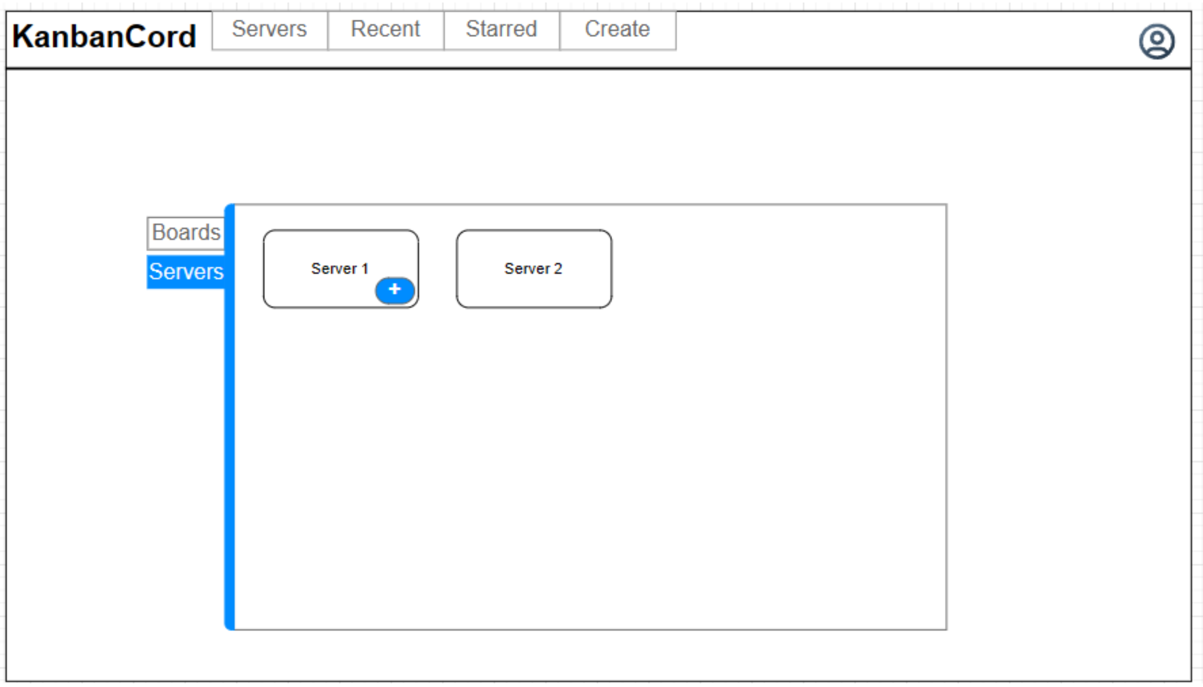
Title: KanbanCord

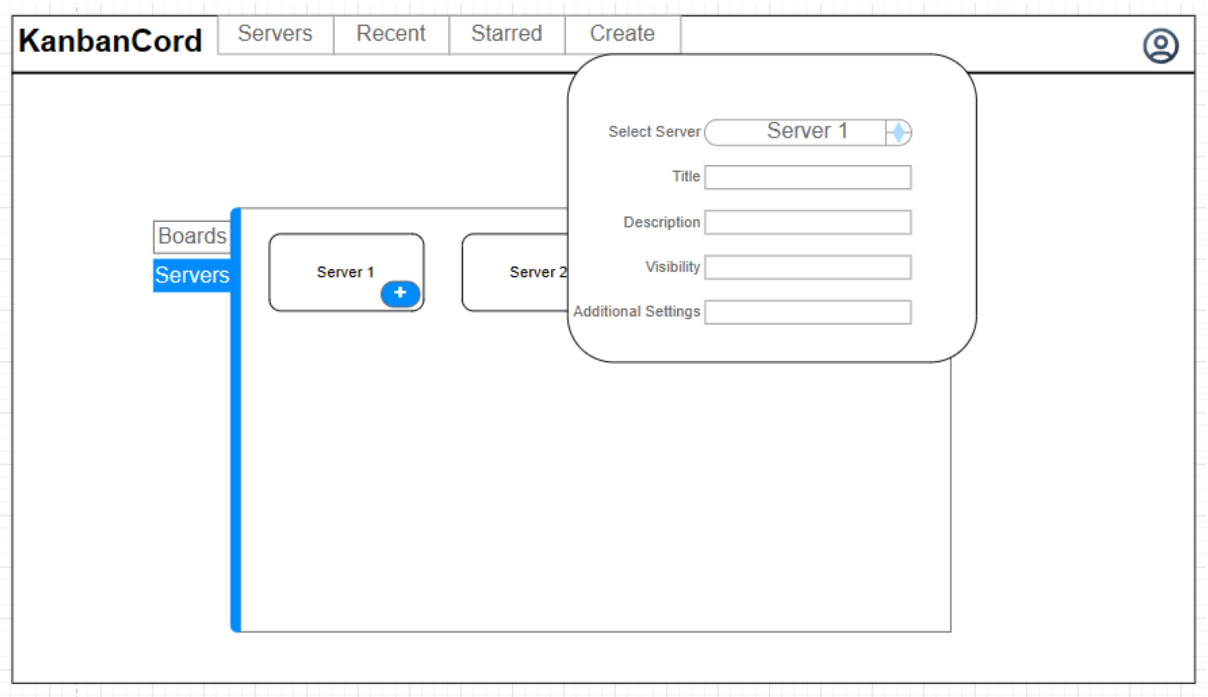
Description: With KanbanCord we want Discord servers and its users to be able to use a Kanban Board that is integrated with Discord, allowing server members to create, view and manage tasks seamlessly within their servers. With role-based access control, users can ensure that only specific roles, members with certain permissions or specified server members can view or edit boards. We aim to make project coordination for discord servers, staff teams or community projects an efficient and intuitive task management experience.

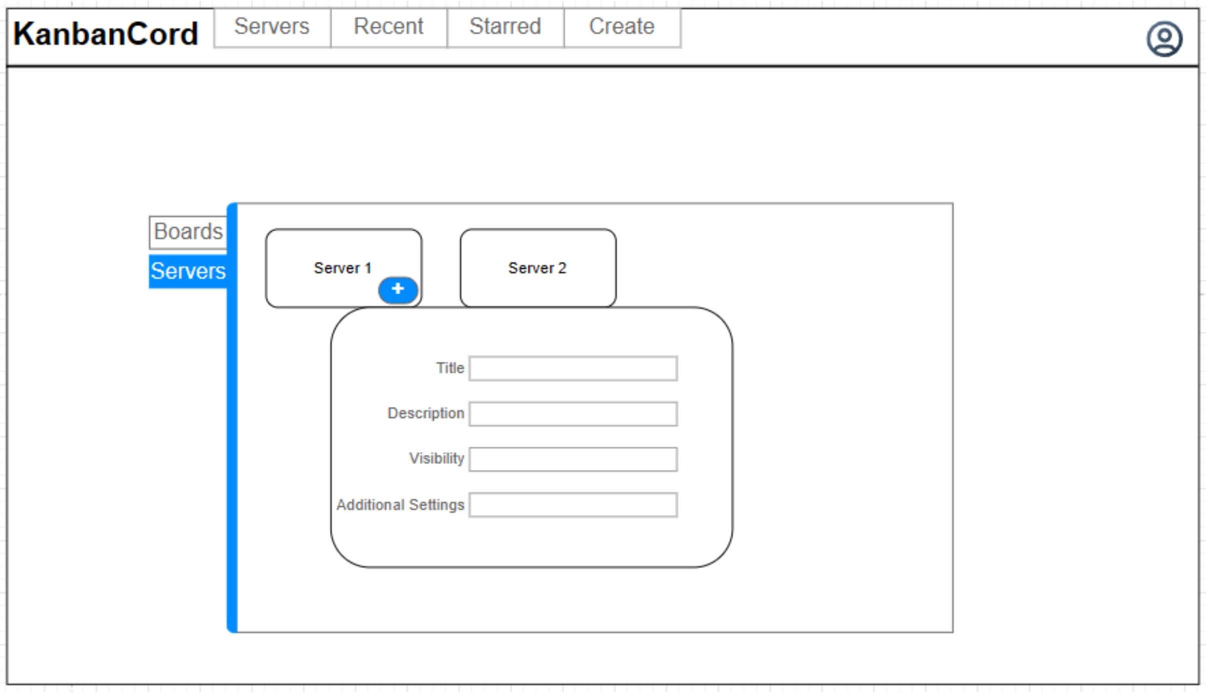
User Stories:

User Story 1:

- As a server member with the necessary permissions, I am able to create a new kanban board for a specific server, so that I can organize tasks and projects for the server members
- Wireframes:



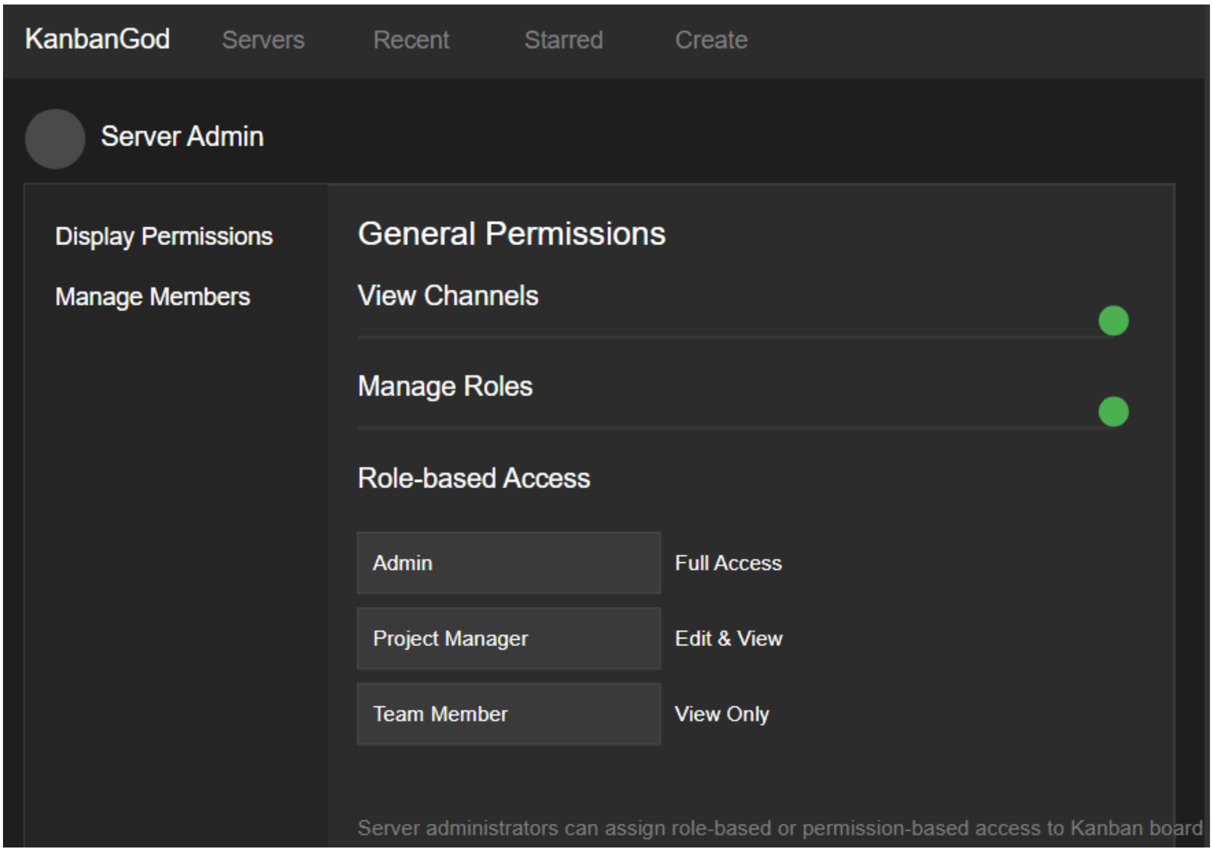




- Acceptance Criteria:
  - The user can create a kanban board by selecting a Discord server they belong to.
  - The board creation form allows the user to set the board’s name and modify the columns
  - The board is saved and linked to the selected Discord server.

User Story 2:

- As a server administrator, I am able to assign role or permission based access to kanban boards, so that I can control what criteria allows members to view or edit the board.
- Wireframes:



General Permissions

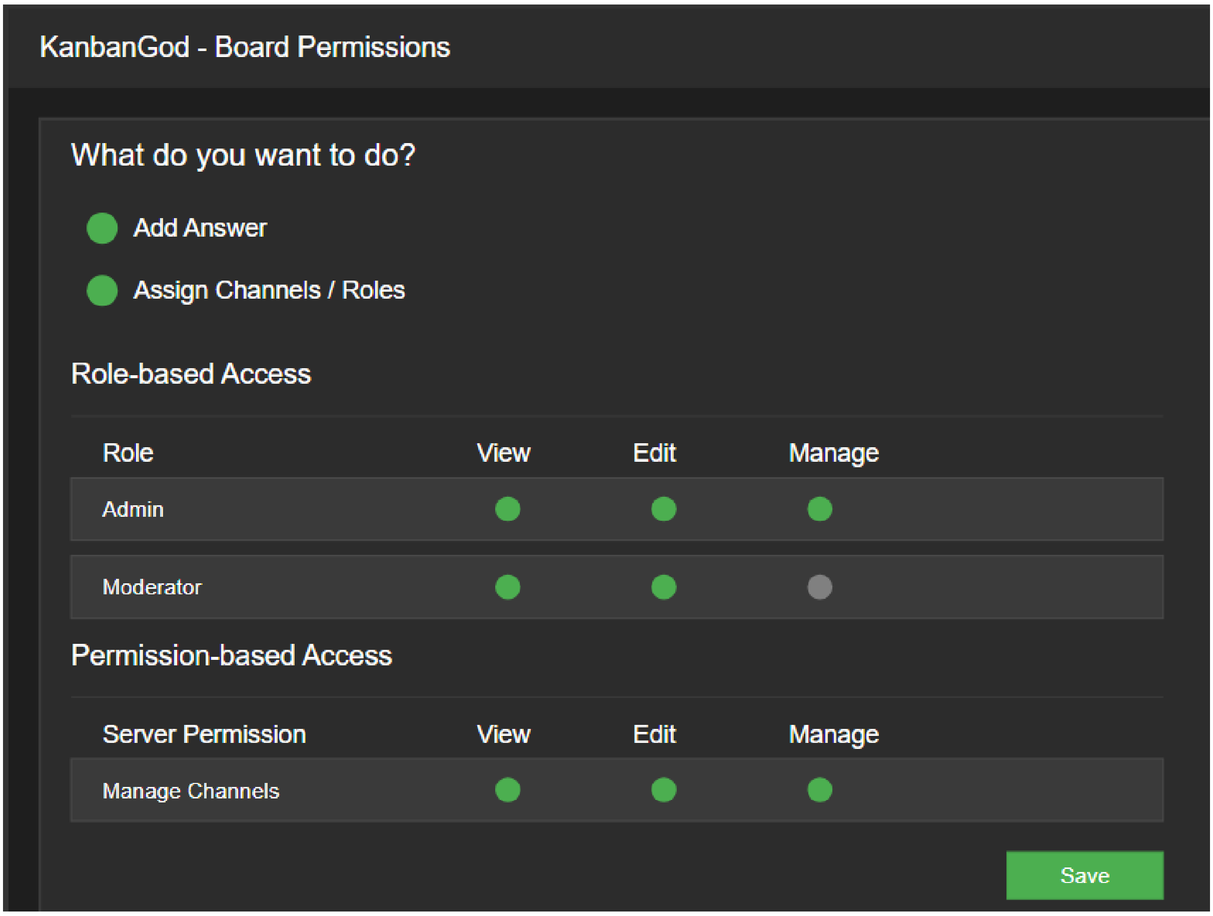
View Channels ●

Manage Roles ●

Role-based Access

Admin	Full Access
Project Manager	Edit & View
Team Member	View Only

Server administrators can assign role-based or permission-based access to Kanban board



KanbanGod - Board Permissions

What do you want to do?

- ● Add Answer
- ● Assign Channels / Roles

Role-based Access

Role	View	Edit	Manage
Admin	●	●	●
Moderator	●	●	○

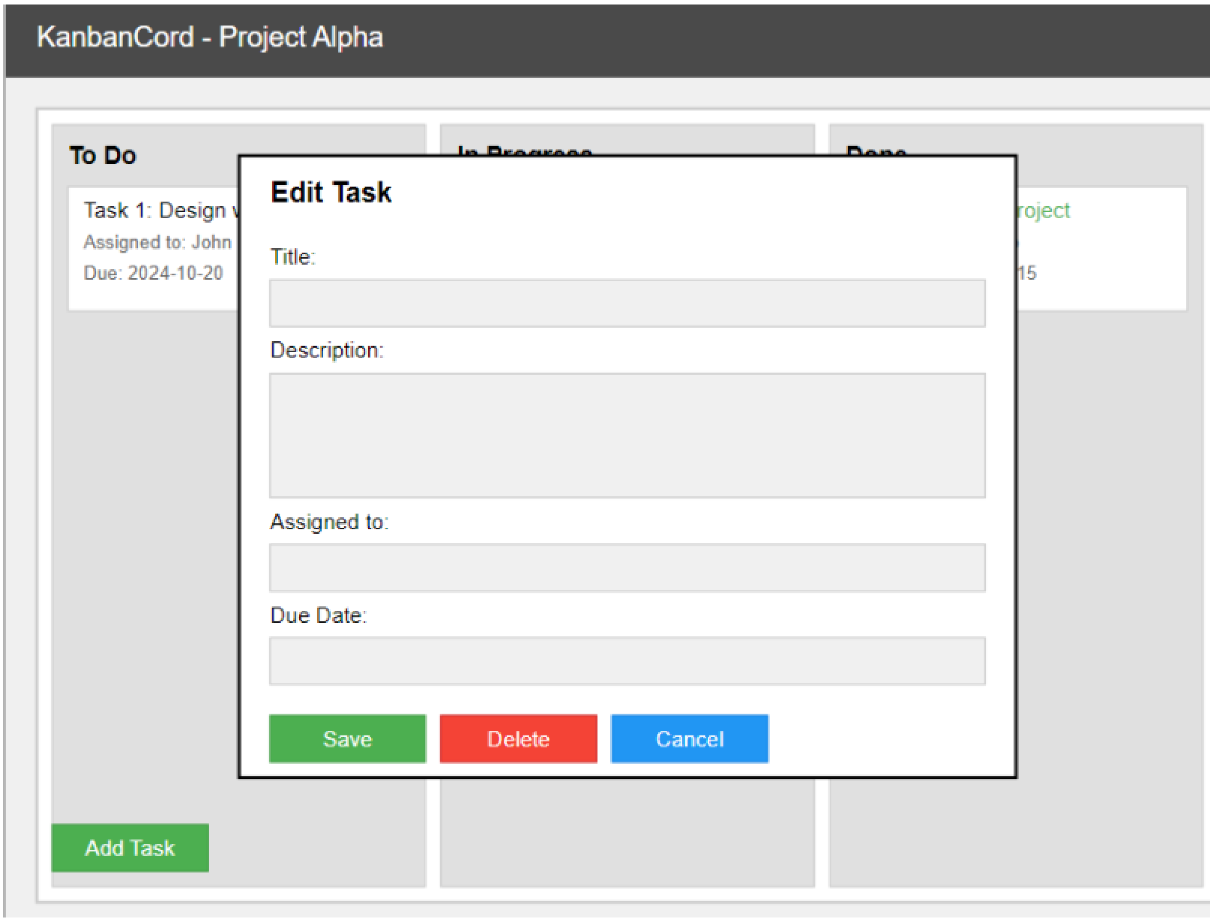
Permission-based Access

Server Permission	View	Edit	Manage
Manage Channels	●	●	●

- Acceptance Criteria:
  - The user can set permissions for roles or discord permissions when creating or editing a kanban board.
  - Role-based access should allow specific roles to either view, edit or manage tasks on the board.
  - Permission-based access should allow members who have certain server permissions to either view, edit or manage tasks on the board.
  - The system restricts access based on the assigned criteria.

User Story 3:

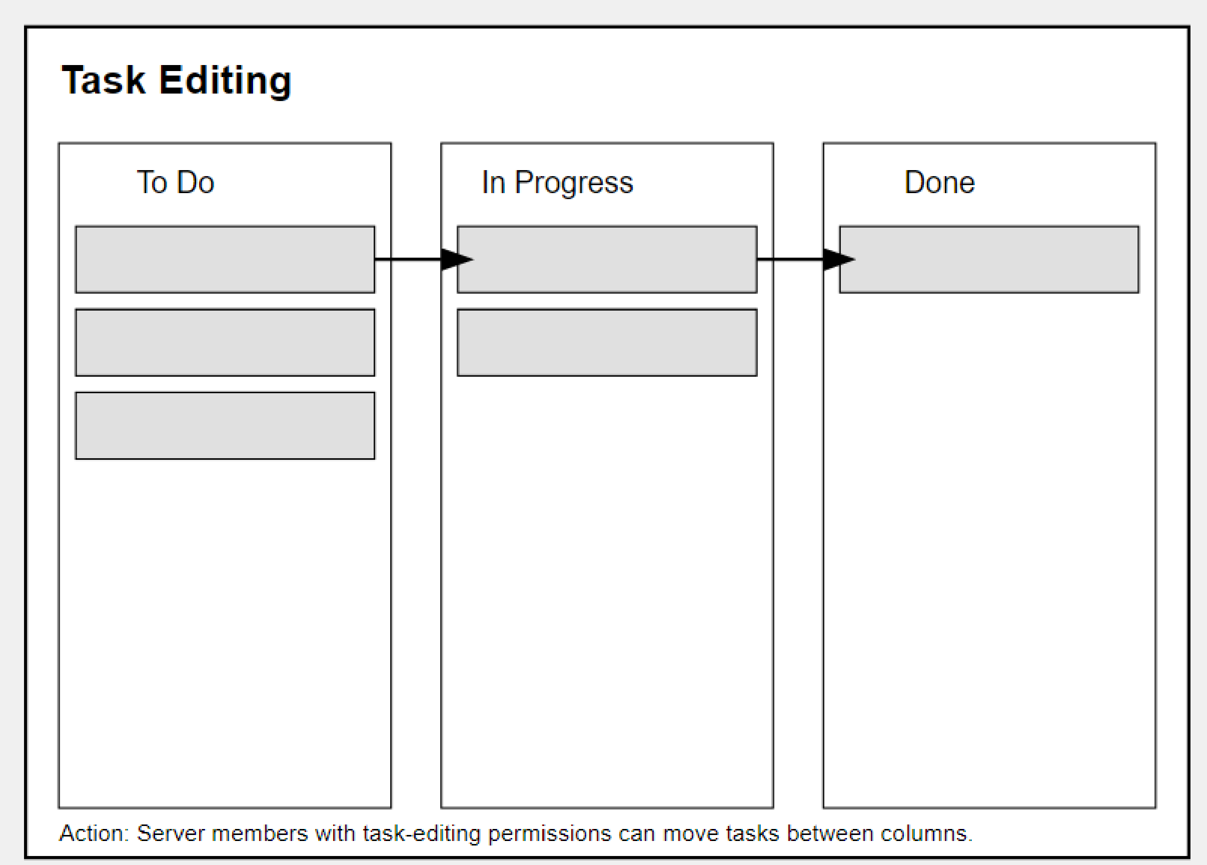
- As a server member with the proper permissions, I am able to add, edit or delete tasks on the kanban board, so that I can manage and update the tasks related to the server's projects.
- Wireframe:



- Acceptance Criteria:
  - The user can add new tasks to any column on the kanban board.
  - The user can edit task details (title, description, due date, etc).
  - The user can delete tasks from the board.
  - Changes are reflected in real-time on the board for all users with access.

User Story 4:

- As a server member with task-editing permissions, I am able to move tasks between columns, so that I can reflect the current status of the tasks.
- Wireframe:

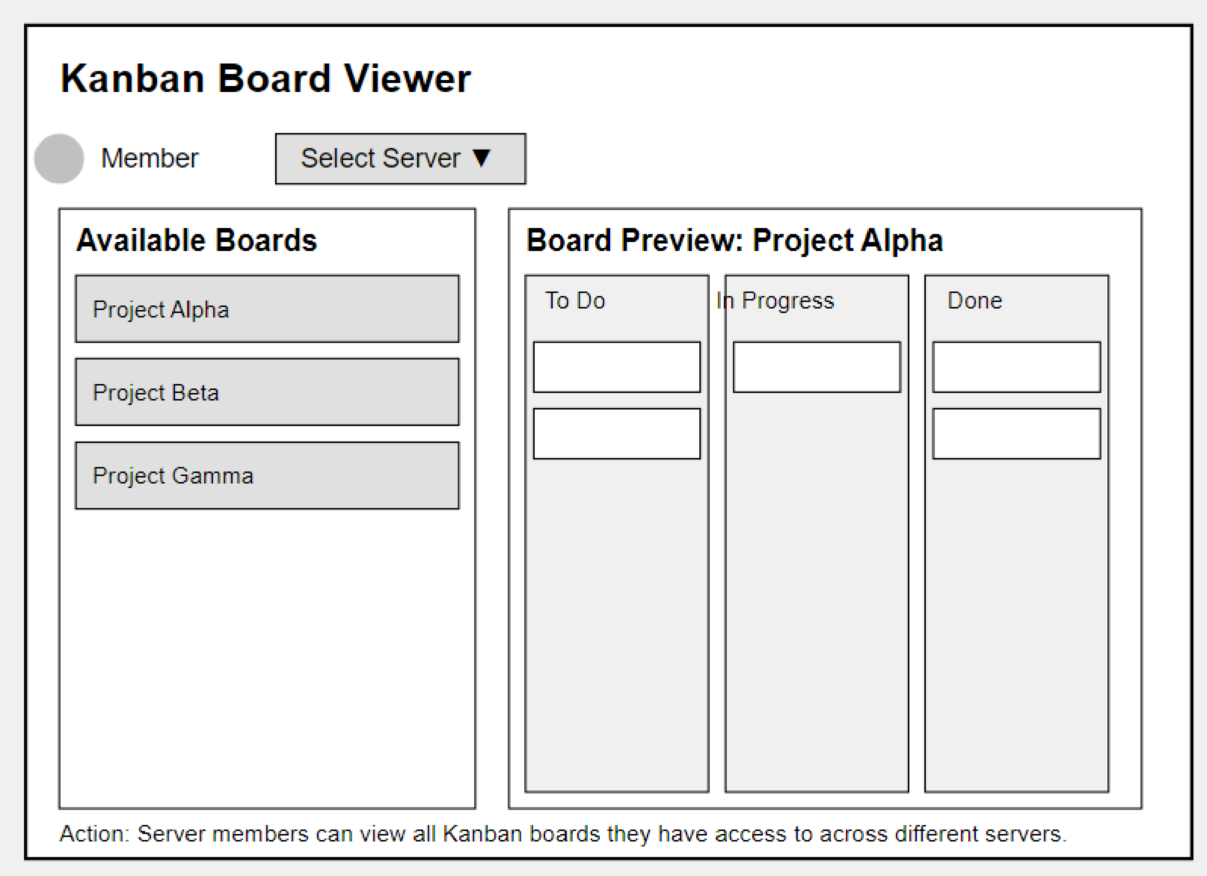


Action: Server members with task-editing permissions can move tasks between columns.

- Acceptance Criteria:
    - The user can drag and drop tasks between columns.
    - Task status changes are updated in real-time.
    - Other users with access to the board can see the task movement immediately.

User Story 5:

- As a server member, I am able to view all the kanban boards I have access to, so that I can easily keep track of multiple projects across different servers.
- Wireframe:

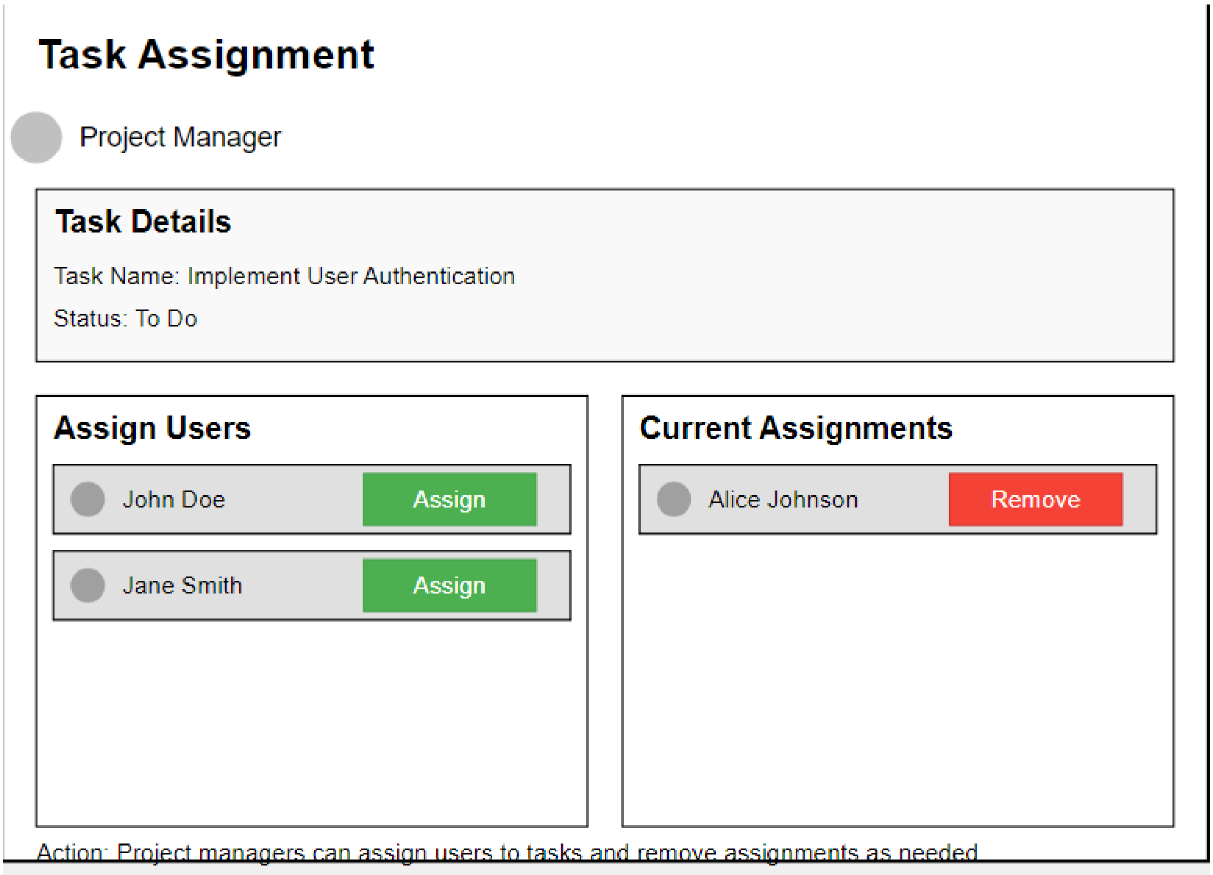


Action: Server members can view all Kanban boards they have access to across different servers.

- Acceptance Criteria:
  - The user can see a dashboard displaying all the boards they have access to across servers.
  - The dashboard allows the user to quickly switch between different boards.
  - Only boards the user has permission to access are visible.

User Story 6:

- As a project manager, I am able to assign user(s) to a task on the board, so that I can ensure every task has been assigned to someone.
- Wireframe:



Task Assignment

Project Manager

Task Details

Task Name: Implement User Authentication

Status: To Do

Assign Users

John Doe

Jane Smith

Current Assignments

Alice Johnson

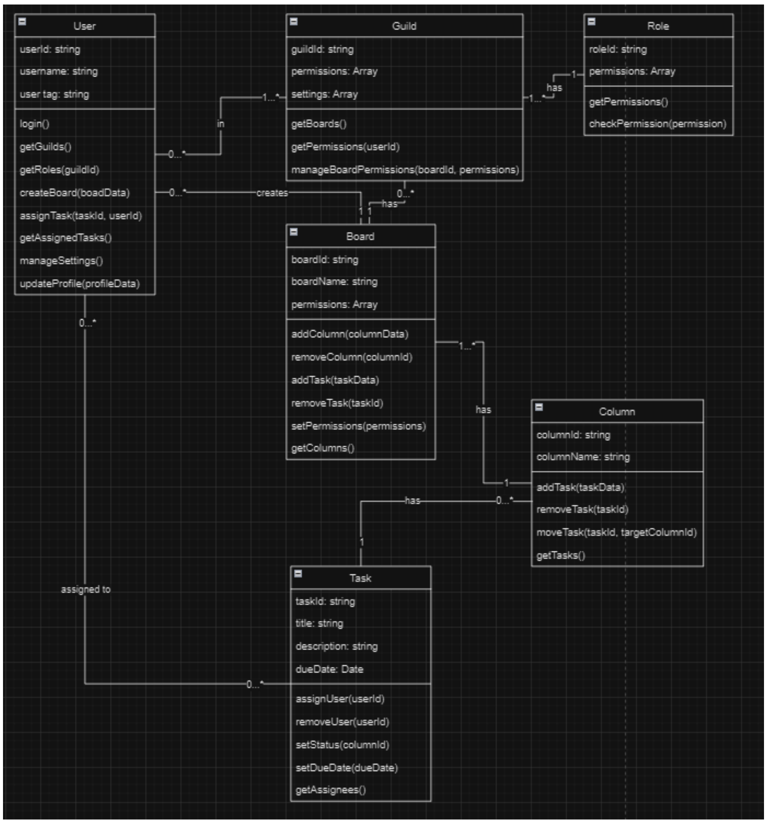
Action: Project managers can assign users to tasks and remove assignments as needed

- Acceptance Criteria:
  - A task can be assigned to more than one user.
  - Only people with the ability to assign users can assign other users to a task.

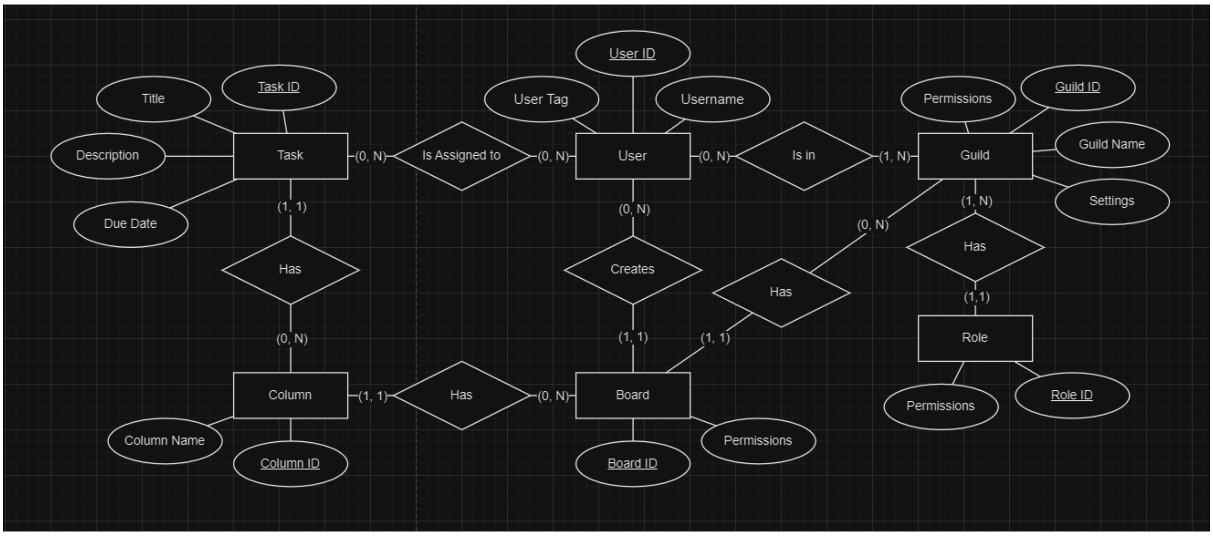
Models:

(As most of the models are the same as the actual database aside from adding a few properties such as taskIndex, columnIndex, userAvatar etc. I decided not to update the models due to lack of time)

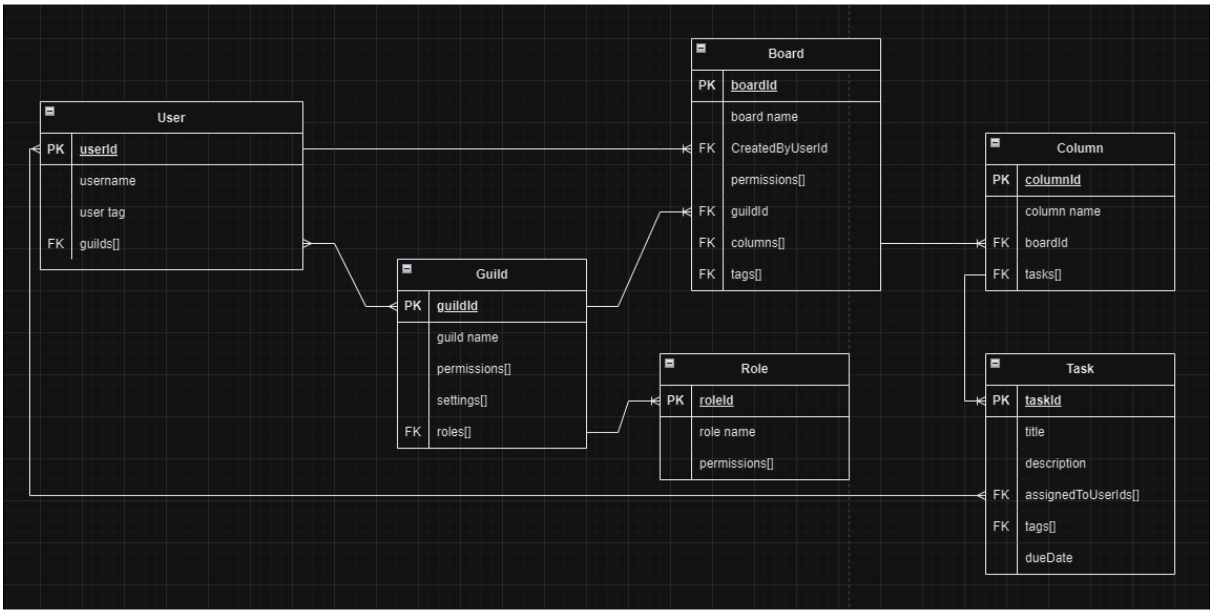
Domain Model:



Conceptual Model:



Logical Model:



Add-ons:

All Permissions:

Discord Permissions:

- View Channels
- Manage Channels
- Manage Roles
- Create Expressions
- Manage Expressions
- View Audit Log
- View Server Insights
- Manage Webhooks
- Manage Server
- Create Invite
- Change Nickname
- Manage Nicknames
- Kick Members

- Ban Members
- Timeout Members
- Send Messages
- Send Messages in Threads
- Create Public Threads
- Create Private Threads
- Embed Links
- Attach Files
- Add Reactions
- Use External Emoji
- Use External Stickers
- Mention @everyone, @here, and All Roles
- Manage Messages
- Manage Threads
- Send Text-to-Speech Messages
- Send Voice Messages
- Create Polls
- Connect
- Speak
- Video
- Use Soundboard
- Use External Sounds
- Use Voice Activity
- Priority Speaker
- Mute Members
- Deafen Members
- Move Members
- Set Voice Channel Status

- Use Application Commands
- Use Activities
- Use External Apps
- Request to Speak
- Create Events
- Manage Events
- Administrator

Kanban Board Permissions:

- View Board
- Create Board
- Edit Board (Name)
- Delete Board
- Manage Board Permissions
- Manage Guild Settings
- Create Columns
- Delete Columns
- Edit Columns(Rename, move columns)
- Create Tasks
- Edit Tasks(Modify details, title, description, due date)
- Delete Tasks
- Assign Tasks (Only allows you to assign or remove yourself from tasks)
- Change Task Status(Move task across columns)
- Manage Task Assignees (Allows adding or removing any user from tasks)
- View Activity Log
- Administrator