 # Design

High-level Design (4%)

Describe which architectural pattern you would use to structure the system. Justify your answer.

For the TODO Bot system, the most fitting architectural pattern would be the Layered (also known as Tiered) Architecture, specifically the Model-View-Controller (MVC) variant. This decision is based on the system's need to handle a variety of different tasks, such as user interaction, task management, and real-time updates, which can be neatly divided across the MVC structure.

The Model component would manage the bot's tasks, user profiles, and other data-related operations, ensuring data integrity and providing a clear interface for data access. The View layer would handle the presentation of information to the user, allowing for a flexible and user-friendly interface that could be modified or extended without affecting the underlying business logic. Finally, the Controller layer would serve as the intermediary, processing user commands, invoking model changes, and updating the view accordingly.

This separation of concerns not only allows for an organized codebase that is easier to maintain and evolve but also supports reuse and enhances the bot's ability to incorporate additional features in the future. While the performance may slightly degrade due to the additional layers of abstraction, and maintenance might become more complex, the benefits of this structure—especially in supporting increasing levels of abstraction during design—make it a robust choice for the TODO Bot system.

The MVC architecture also supports the integration of a client-server architecture within its layers, which is beneficial for TODO Bot's need to communicate over a network. This would enable efficient task distribution and workload management between the server (providing services or data) and clients (consuming these services), which could be particularly useful for handling the expected volume of 5000 concurrent users.

In summary, the MVC variant of the Layered Architecture offers a balance of structured organization, ease of maintenance, and potential for future expansion, making it an ideal choice for the TODO Bot system's requirements.

Low-level Design (4%)

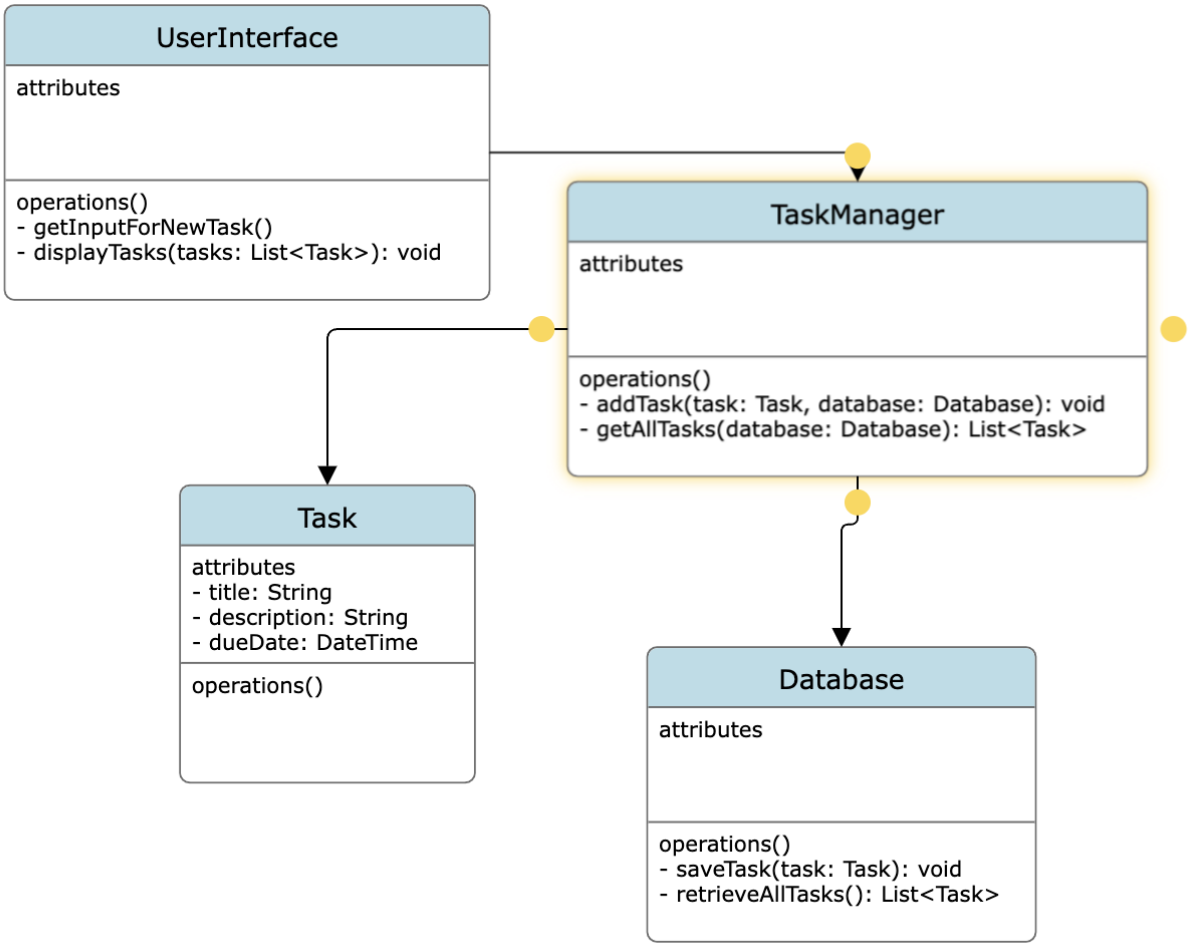
Discuss which design pattern family might be helpful for implementing this project. Justify your answer, providing a code or pseudocode representation _and_ an informal class diagram.

The behavioral design pattern family is the best for implementing our TODO bot. Our AI TODO bot will need to analyze the behavior of its users to accurately and efficiently help developers and engineers without getting in the way. The strategy pattern would be helpful since it allows tasks to be completed at runtime, which is important for an AI-based application. If a developer would like to change how they want to prioritize their tasks, this pattern would help the AI quickly and efficiently reorganize the tasks to the developer's liking.

Pseudocode:

```
class UserInterface {
    // Method to get user input for new task
    method getInputForNewTask() {
        // Take input from user
        // Return input as Task object
    }

    // Method to display tasks to the user
    method displayTasks(tasks) {
        // Loop through tasks and print details
    }
}

class TaskManager {
    // Method to add a new task
    method addTask(task, database) {
        // Add the new task to the database
        database.saveTask(task)
    }

    // Method to get all tasks
    method getAllTasks(database) {
        // Retrieve all tasks from the database
        return database.retrieveAllTasks()
    }
}
```

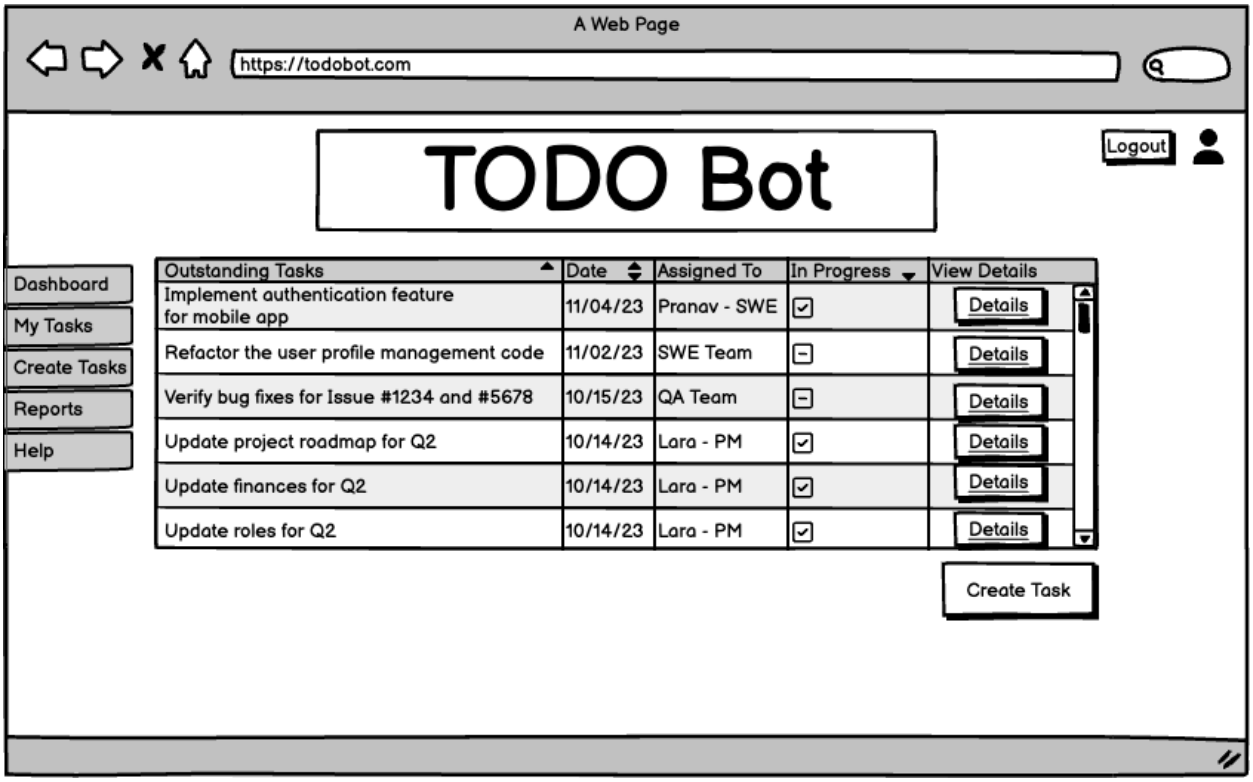
```
class Database {
    // Method to save a task
    method saveTask(task) {

    }

    // Method to retrieve all tasks
    method retrieveAllTasks() {
        // Return a list of Task objects
    }
}

// Task class to represent each task
class Task {
    String title
    String description
    DateTime dueDate
    // Constructor and other necessary methods...
}
```

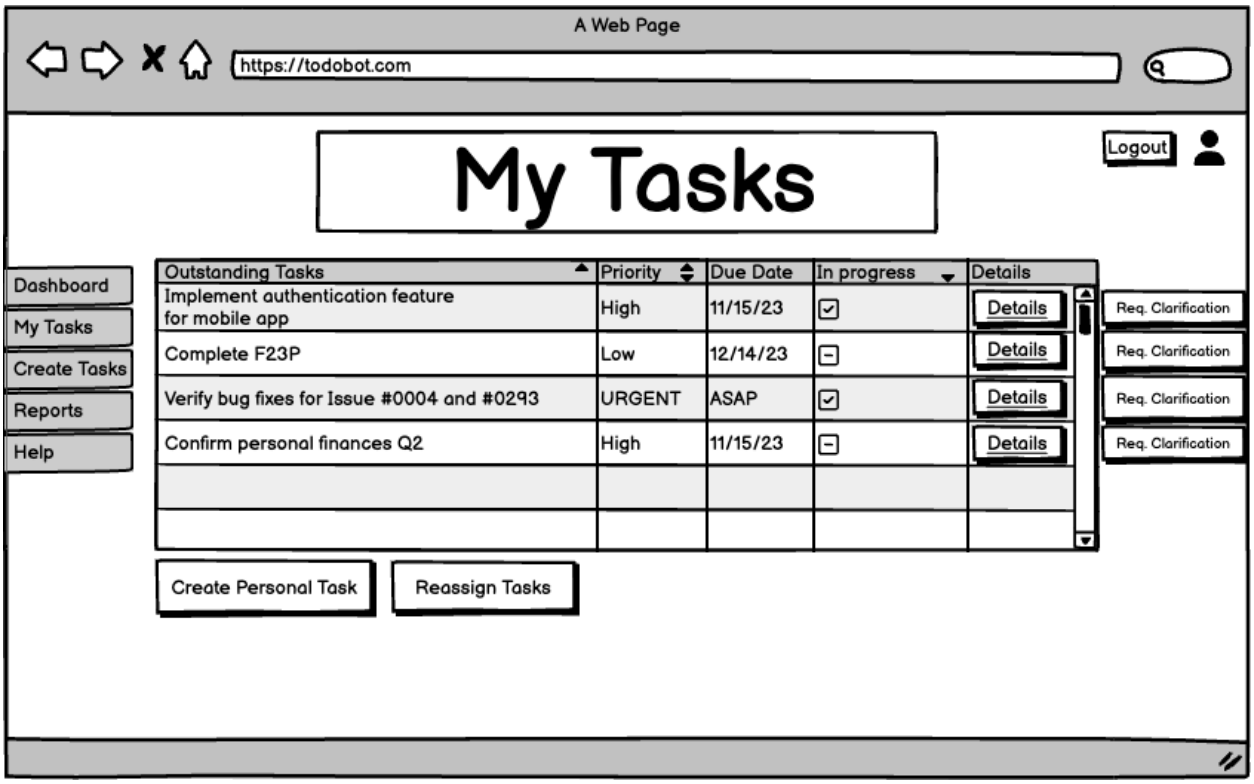

UserInterface
attributes
operations()
- getInputForNewTask()
- displayTasks(tasks: List<Task>): void
TaskManager
attributes
operations()
- addTask(task: Task, database: Database): void
- getAllTasks(database: Database): List<Task>
Task
attributes
- title: String
- description: String
- dueDate: DateTime
operations()
Database
attributes
operations()
- saveTask(task: Task): void
- retrieveAllTasks(): List<Task>

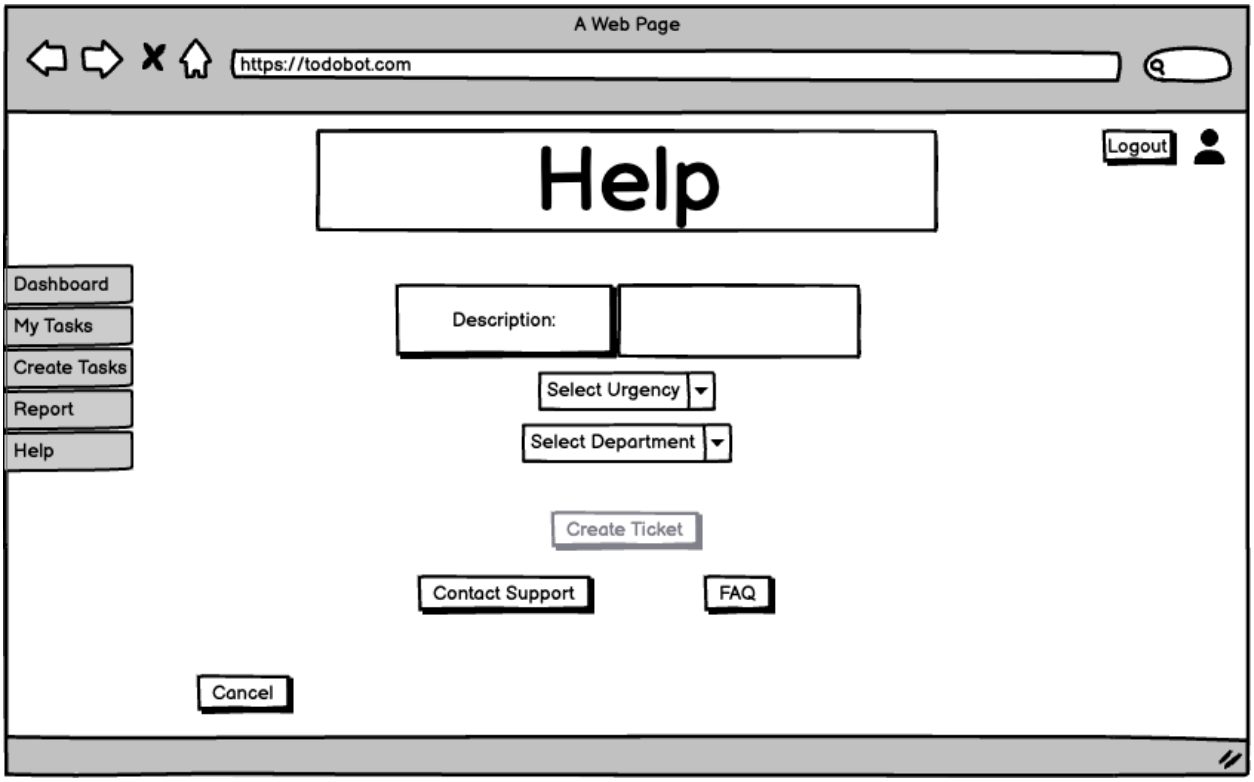
Design Sketch (4%)

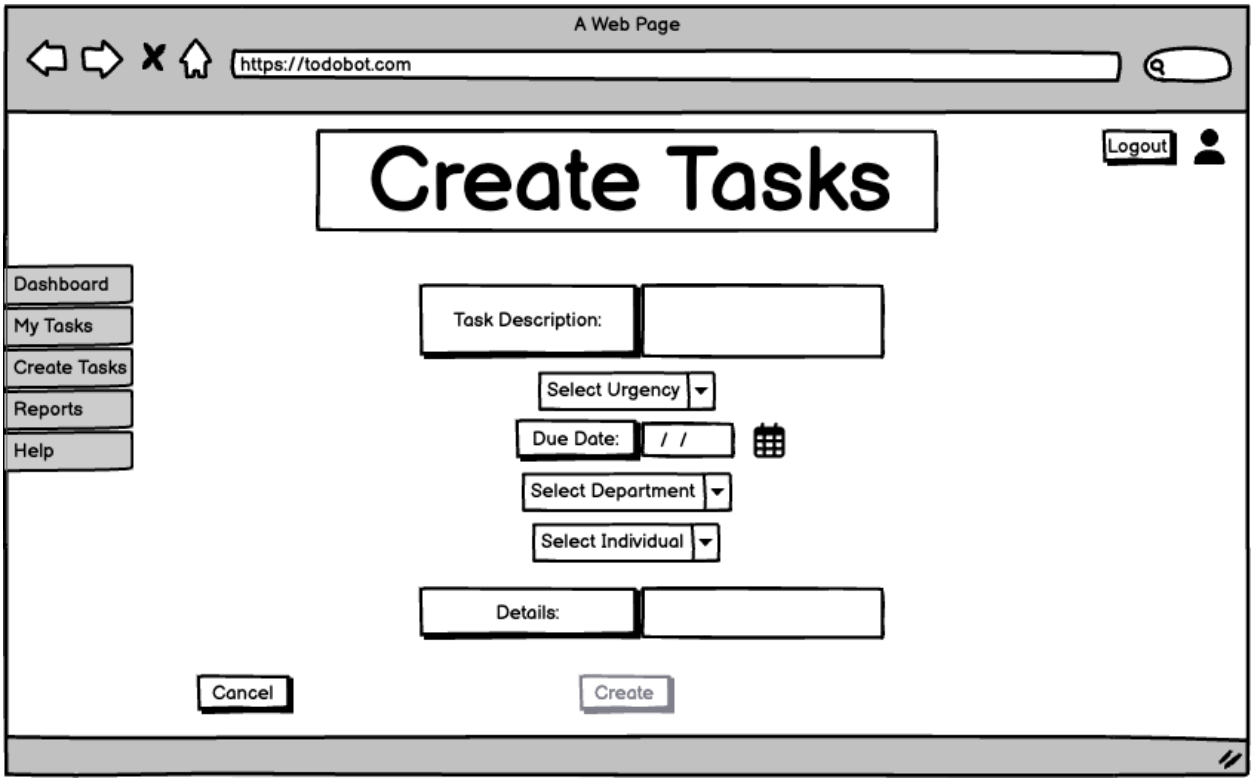
Design sketches provide a visual overview of the look and feel of your project. This may include but is not limited to one of the following:

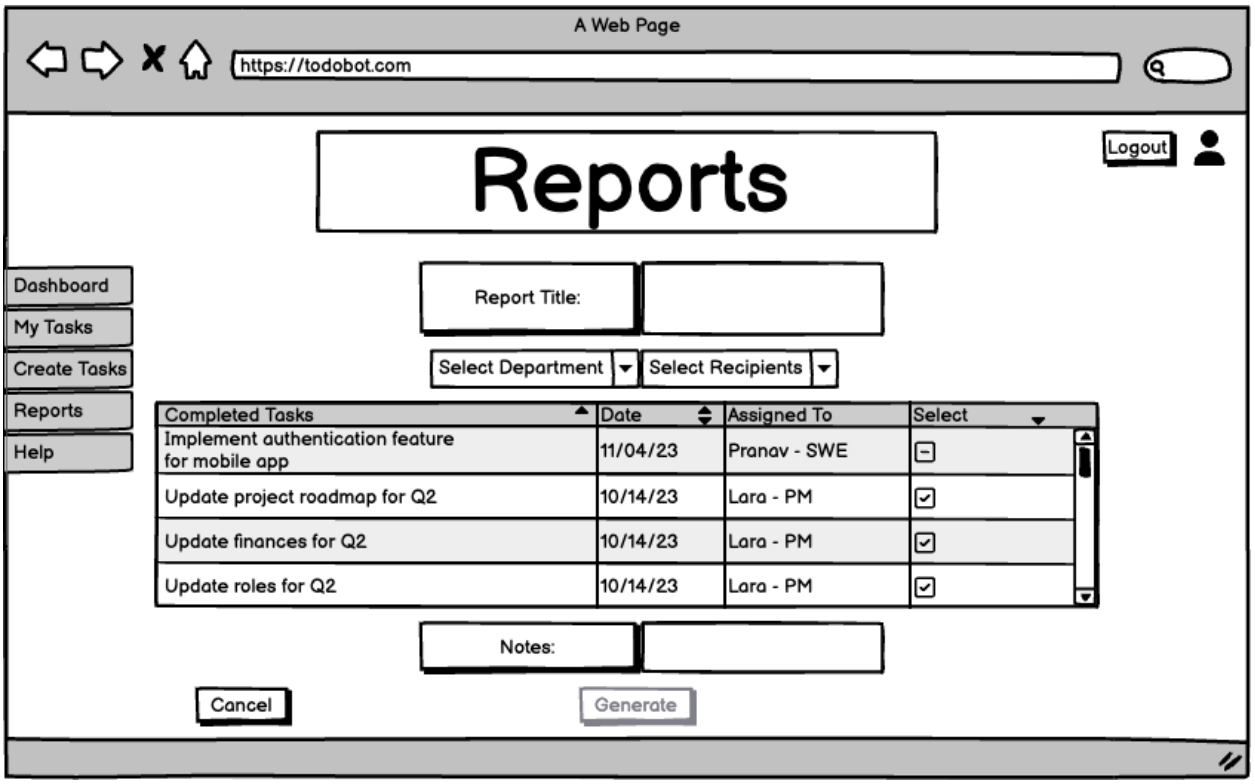
* Creating a wireframe mockup of your project user interface in action.
* Creating a storyboard that illustrates a primary task that a user would complete with your project.


A Web Page
https://todobot.com
TODO Bot
Logout
Dashboard
My Tasks
Create Tasks
Reports
Help
Outstanding Tasks
Date
Assigned To
In Progress
View Details
Implement authentication feature for mobile app
11/04/23
Pranav - SWE
Details
Refactor the user profile management code
11/02/23
SWE Team
Details
Verify bug fixes for Issue #1234 and #5678
10/15/23
QA Team
Details
Update project roadmap for Q2
10/14/23
Lara - PM
Details
Update finances for Q2
10/14/23
Lara - PM
Details
Update roles for Q2
10/14/23
Lara - PM
Details
Create Task


A Web Page
https://todobot.com
My Tasks
Logout
Dashboard
My Tasks
Create Tasks
Reports
Help
Outstanding Tasks
Priority
Due Date
In progress
Details
Implement authentication feature for mobile app
High
11/15/23
Details
Req. Clarification
Complete F23P
Low
12/14/23
Details
Req. Clarification
Verify bug fixes for Issue #0004 and #0293
URGENT
ASAP
Details
Req. Clarification
Confirm personal finances Q2
High
11/15/23
Details
Req. Clarification
Create Personal Task
Reassign Tasks


A Web Page
https://todobot.com
Help
Logout
Dashboard
My Tasks
Create Tasks
Report
Help
Description:
Select Urgency
Select Department
Create Ticket
Contact Support
FAQ
Cancel


A Web Page
https://todobot.com
Create Tasks
Logout
Dashboard
My Tasks
Create Tasks
Reports
Help
Task Description:
Select Urgency
Due Date:
/ /
Select Department
Select Individual
Details:
Cancel
Create


A Web Page
https://todobot.com
Reports
Logout
Dashboard
My Tasks
Create Tasks
Reports
Help
Report Title:
Select Department
Select Recipients
Completed Tasks
Date
Assigned To
Select
Implement authentication feature for mobile app
11/04/23
Pranav - SWE
Update project roadmap for Q2
10/14/23
Lara - PM
Update finances for Q2
10/14/23
Lara - PM
Update roles for Q2
10/14/23
Lara - PM
Notes:
Cancel
Generate

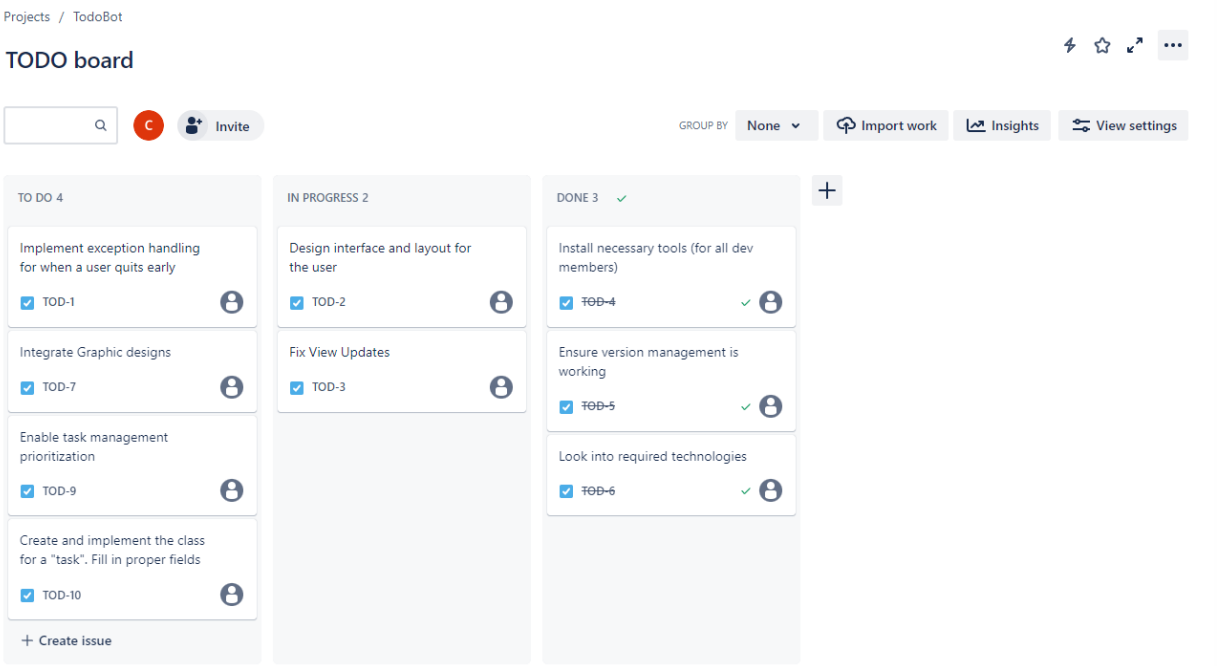
In designing this program, we focused on creating a simple, user-friendly interface that ensures straightforward navigation and accessibility. We incorporated a consistent side navigation bar across various screens to allow users to move between tasks easily. The grid-based layout of the interface was used to look simple and organized. The minimalist design approach also aids in focusing the user's attention on the task at hand, without unnecessary distractions. Overall, the goal was to create an environment where users could achieve their objectives with efficiency and ease.

Project Check-In

Complete [this survey](https://forms.gle/XwNUaj4xjuH6grsq8) to provide an update on your team progress on the project for this semester. Only one team member needs to complete this for the group.

Process Deliverable (3%)

The submission for this deliverable will depend on the specific SE process model your team plans to use to complete the group project (as described in your project proposal). Example submissions for different processes include:
* Prototyping: submit a prototype of your system
* Scrum: submit the notes (including each teammate) from your most recent scrum meeting
* Kanban: submit a list of prioritized tasks from your task management system
* Waterfall: submit supplementary planning documentation
* Extreme programming: submit acceptance test criteria
* Spiral: submit risk analysis
* ...
For other processes not listed above, the instructor will contact you with the exact submission requirements for this task.



Above we have our Kanban Board For our TODO bot posted, highlighting developer tasks to do, ones in progress, and ones completed

Due: November 10 at 11:59pm
- [ ] High-level design (done)
- [ ] Low-level design (done)
- [ ] Mock user interface (done)
- [ ] Process II deliverable (done)
- [ ] Project Check-in survey (done)