

The World's Greatest TODO App

Team MJ's

Moses Hughes, Matthew Moore

Traditional to-do apps often lack the necessary flexibility and customization for software development teams, leading to inefficient software development plans and inadequate organization.

Research

“A REVIEW OF DAILY PRODUCTIVITY GROWTH USING TODO MANAGER” IRJMEFS, vol. 2, no. 12, Dec. 2020

TODO lists can be improved by adding several key features: partial task completion, task efficiency, and a target completion indicator.

Partial indicator: allow users to be able to complete smaller tasks, making management more efficient.

Task efficiency visualization: allow users to see how efficient they are by seeing completed and pending tasks, keeping accountability.

Target completion indicator: allow users to set a goal for themselves, compared with daily task efficiency, keeping users motivated

Solution: The World's Greatest TODO APP

Recognizing the pivotal role of clear requirements in software engineering success, The World's Greatest TODO App addresses this challenge by enabling managers to assign roles and groups, fostering improved task organization and collaboration.

The app further enhances the software development process by seamlessly integrating with popular calendar apps, offering developers a visual timeline for task completion.



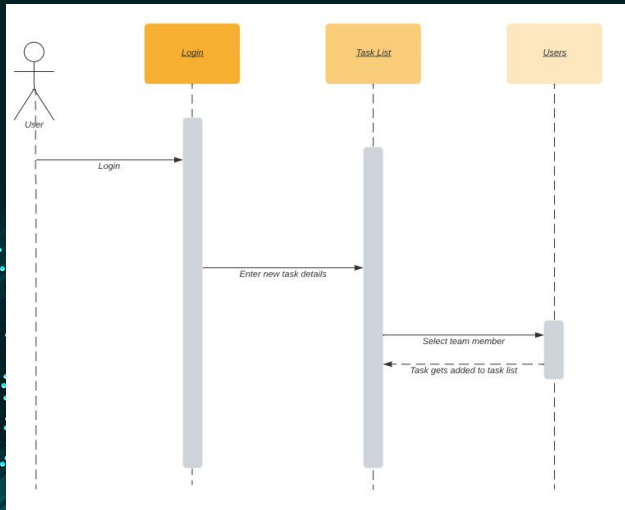
Solution: The World's Greatest TODO APP

By simplifying task assignment and deadline setting, the TODO app contributes to more successful projects, aligning with the crucial factors identified in The CHAOS Report for project success.

- Teamwide TODO lists with claimable tasks
- Personal tasks with timelines and subtasks
- Integration with Google Calendar
- Projects that are made up of tasks
- Managers assign tasks to individuals within the team



Use Case 1: Create a New Task



Preconditions:

- User must be logged into the system.
- User has the necessary project access permissions.

Main Flow:

- User initiates the task creation process.
- User provides task details, including title, description, priority, and due date [S1].
- User assigns the task to a team member [S2].
- System validates the input and creates the task [S3].

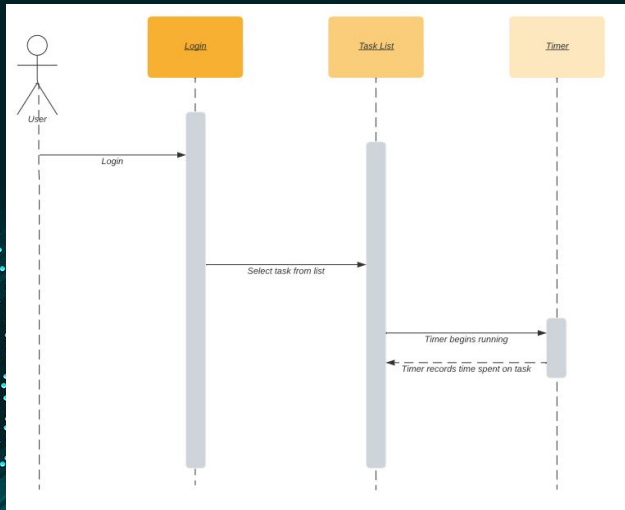
Subflows:

- [S1] User enters task details through the task creation form.
- [S2] User selects a team member from the list of available team members.
- [S3] System creates the task and adds it to the project's task list.

Alternative Flows:

- [E1] The user cancels the task creation process.

Use Case 2: Track the Time Spent on a Task



Preconditions:

User must be logged into the system.

User is assigned to a task.

Main Flow:

User selects a task for which they want to track time.

User starts a timer for the task [S1].

User stops the timer when work on the task is complete [S2].

System records the time spent on the task [S3].

Subflows:

[S1] User starts a timer by clicking a "Start Timer" button.

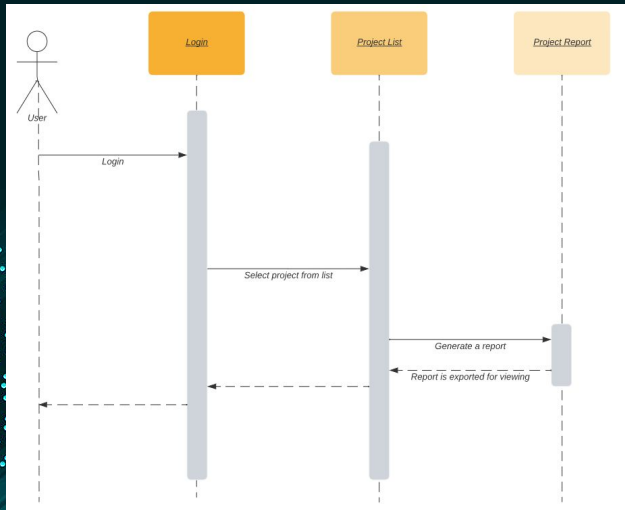
[S2] User stops the timer by clicking a "Stop Timer" button.

[S3] System records the elapsed time and updates the task with the time spent.

Alternative Flows:

[E1] The user forgets to stop the timer and needs to adjust the time manually.

Use Case 3: Generate Task Reports



Preconditions:

User must be logged into the system.

User has access to project and task data.

Main Flow:

User selects a project or a specific time frame for the report.

User requests the system to generate a task report [S1].

System compiles a report containing task details, time spent, and progress [S2].

User views and exports the report [S3].

Subflows:

[S1] User specifies report parameters (e.g., project, time frame).

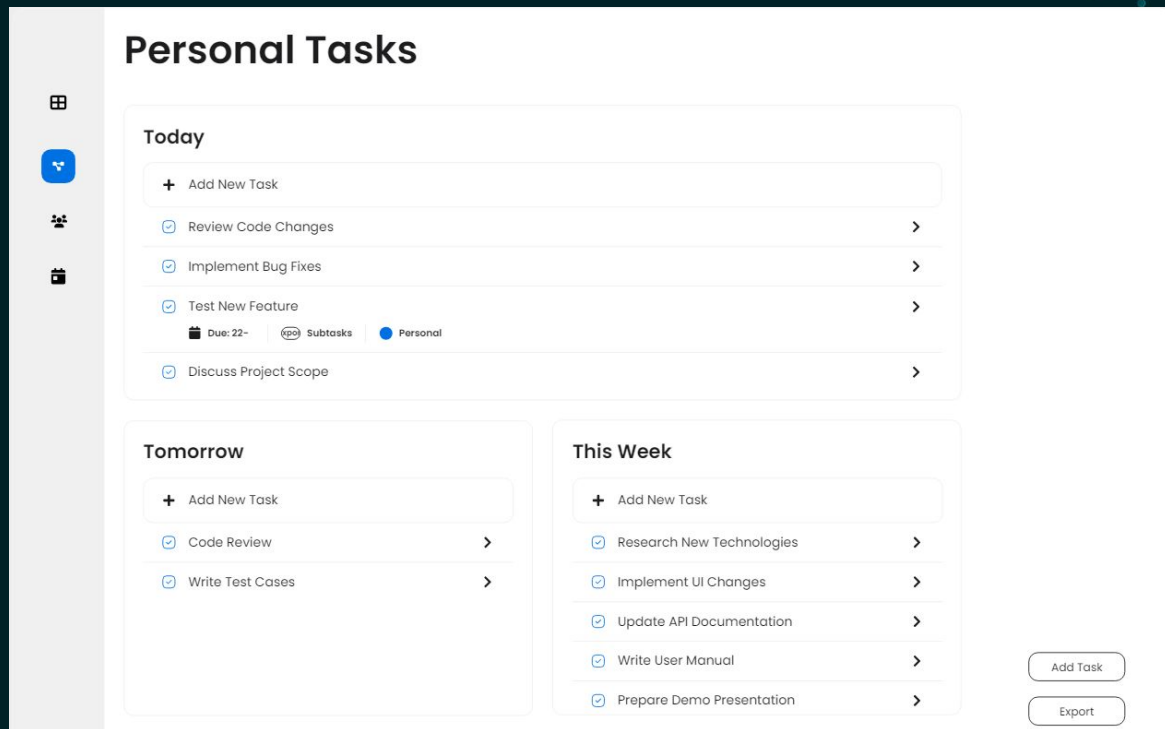
[S2] System generates the report with relevant task data.

[S3] User can view the report in the app and export it if needed.

Alternative Flows:

[E1] The user cancels the report generation.

Mockup



Mockup

☰

▼

👤

📅

🗑️

✕

Task name

Status

Start Task

Complete Task

Project

Project Name 🔍

Due Date

11/11/1111 📅

Priority

Set Priority

Description

Subtasks

☒ Subtask

☒ Subtask

☒ Subtask

Notes

No Comments Yet.

1

Processes Used

Our T.W.G.T.A. project makes use of the agile software process. This was chosen because it provides great flexibility for the team to respond to any issues that arise during development.

Being an iterative process, it will also allow us to easily respond to user feedback and develop new features following an initial deployment of the app.



Group Requirements Elicitation

Plan to utilize group elicitation techniques to generate feedback

- Focus groups, primarily of software developers
 - This will be extra helpful in our circumstance because as software developers they will have an understanding of how the app works and what is realistic to implement.
- Surveys
 - Through the data collected through surveys, we can understand the priorities of user needs, and then improve software functions and features.



Future work and Limitations

- Designing a robust formal UI.
 - Lack a graphic designer or anyone proficient in UI/UX development
- Implementing login and authentication and other security services
- Group Elicitation



What we've learned

- Importance of scrums and process models in the development of real-world projects.
- How user stories help in interpreting how a specific function operates.
- How use cases help in creating a proper flow sequence for common actions in the program, and any outlying flows that must be accounted for

