PLANORAMA

Design Document

Team 22:
Sruthi Bhamidipati
Yana Sheoran
Vihaan Khajanchi
Jack Kenny
Sejun Her

Index

Purpose

Productivity tools often fail to strike a balance between simplicity and functionality, leaving users either overwhelmed by complexity or unable to efficiently manage their tasks, goals, and time due to limited features. Existing platforms such as Trello, Notion, and Google Calendar focus on distinct aspects of productivity but lack a comprehensive, integrated solution that combines task management, time tracking, goal setting, and collaboration in an intuitive manner.

Planorama is designed to fill this gap by offering an all-in-one productivity application that streamlines workflow management while keeping users engaged through gamification and collaboration tools. Unlike traditional solutions, Planorama integrates task prioritization, real-time productivity insights, streak tracking, and team-based goal-setting to help users stay organized and motivated.

The application is tailored for individuals and teams—including students managing assignments, professionals handling complex projects, and organizations requiring structured task coordination. Planorama's smart suggestions, customizable interface, and built-in collaboration features ensure that users can personalize their workflow without sacrificing efficiency.

By combining usability, motivation, and teamwork into a single platform, Planorama provides a seamless and engaging way to enhance productivity while making task management more efficient, enjoyable, and adaptable to different working styles.

Functional Requirements

1. Task Management
    - 1.1. As a user, I would like a task page so I can view all my tasks.
    - 1.2. As a user, I would like to be able to create tasks.
    - 1.3. As a user, I would like to be able to edit and reorder tasks.
    - 1.4. As a user, I would like to be able to delete tasks.
    - 1.5. As a user, I would like to be able to create subtasks within a task.
    - 1.6. As a user, I would like to drag and drop tasks so that I can easily visualize my task order.
    - 1.7. As a user, I would like to categorize tasks by priority, tags, or deadlines so that I can make decisions on which order to take the tasks.
    - 1.8. As a user, I would like to color code tasks based on categories so that I can quickly identify them.
    - 1.9. As a user, I would like to set default task templates so that I can quickly create repetitive tasks.

    1.10. As a user, I would like to set different statuses for tasks (in-progress, blocked, waiting for review).
    1.11. As a user, I would like to be able to filter tasks by completion status.
    1.12. As a user, I would like to archive completed tasks so that I can keep my workspace clutter-free.
    1.13. As a user, I would like to be able to roll over incomplete tasks to the following day, either automatically or manually.
    1.14. As a user, I would like to be able to set task dependencies to organize connected tasks better.
    1.15. As a user, I would like to schedule tasks for a future start date so that I can plan ahead without cluttering my current task list.
    1.16. As a user, I would like to receive smart suggestions for breaking up long tasks into smaller subtasks
2. Goal-Setting & Progress Tracking
    2.1. As a user, I would like to jot notes down about tasks that I have or have not completed so that I can reference thoughts for later.
    2.2. As a user, I would like to track undesirable habits/tasks so that I can check progress on if I'm lowering how unproductive I am (if time allows).
3. Time-Tracking & Productivity Insights
    3.1. As a user, I would like to log time spent on tasks so that I can keep track of progress.
    3.2. As a user, I would like to be able to visualize how my time is spent across different categories (if time allows).
    3.3. As a user, I would like to set an estimated time and an actual time for each of my tasks, and see the difference between the two.
    3.4. As a user, I would like to receive a weekly summary of my most and least productive days so that I can adjust my workflow.
    3.5. As a user, I would like to compare my productivity across different weeks/months to track my long-term progress.
    3.6. As a user, I would like to check time spent on breaks so that I can maintain a healthy balance.
    3.7. As a user, I would like to be given gentle reminders through pop-ups or alerts suggesting I take breaks after long periods of continuous work.
    3.8. As a user, I would like to receive productivity tips based on my task history (if time allows).
4. Gamification & Team Features
    4.1. As a user, I would like to see a streak tracker to visualize my consistent task completion.
    4.2. As a user I'd like to be able to navigate to a profile tab where I can see relevant information about my account.

4.3. As a user, I would like to be able to share or display my streak or other object on my profile.
4.4. As a team member, I would like to view a productivity leaderboard to foster a sense of competition and teamwork.
4.5. As a team member, I would like to send and receive kudos for completed tasks to encourage engagement.
4.6. As a team leader, I would like to assign weighted points to tasks to reflect their complexity.
4.7. As a user, I would like to be able to search for other users so that I can use other functionality like teams.
4.8. As a user, I would like to be able to be a contributor (leader or member) for multiple unique teams.
4.9. As a user, I would like to be able to access a unique team page for each of the teams that I am a part of to see team tasks and contributors.
4.10. As a team leader, I would like to create a “team” so that I can manage group tasks.
4.11. As a team leader, I would like to be able to add users to a team.
4.12. As a team leader, I would like to be able to assign tasks.
4.13. As a team leader, I would like to set permissions for different team members so that I can control who edits tasks.
4.14. As a team leader, I would like to be able to view team members’ activity.
4.15. As a team leader, I would like to track the time my members are spending on tasks.
4.16. As a team member, I would like to be able to add tasks.
4.17. As a team member, I would like to be able to claim tasks.
4.18. As a team member, I would like to mark team tasks as completed.
4.19. As a team member, I would like to leave comments on tasks so that I can communicate updates.
4.20. As a team member, I would like to tag people so that they can be notified when a task needs their attention.

5. User Preferences & Accessibility
    5.1. As a user, I would like to create an account with unique log in credentials.
    5.2. As a user, I would like to log in to my account when I return.
    5.3. As a user, I'd like my data to be synced to my account so that signing in somewhere new shows me my tasks.
    5.4. As a user, I would like to be able to use multiple devices so that I can be productive and check my tasks no matter my situation.
    5.5. As a user, I would like to be able to access a settings option that allows for preferred features.

    5.6. As a user, I would like to enable dark mode or adjust the app theme so that I can work in my preferred visual setting.
    5.7. As a user, I would like to change the font size and spacing so that I can improve readability based on my preference.
    5.8. As a user, I would like to choose what view I am in (calendar, timeline, etc.) so that it can be in my most preferred state.
    5.9. As a user, I would like to set focus mode where notifications are muted for deep working sessions.
    5.10. As a user, I would like to receive notifications so that I can be reminded of important tasks and stay productive.
    5.11. As a user, I would like to be able to change my display name for different teams to best suit each team.
    5.12. As a user, I would like to be able to change my profile picture so that I can customize my account.
    5.13. As a user, I would like to be able to delete my account if I no longer want to use the app.
6. Integrations & Data Export
    6.1. As a user, I would like to integrate Planorama with third-party tools like Google Calendar or Slack for streamlined workflow (if time allows).
    6.2. As a user, I would like to export my task data in CSV or PDF format for external analysis (if time allows).

Non-Functional Requirements

1. The platform must ensure tasks and insights load within 500ms for an optimal user experience while supporting up to 10,000 simultaneous users with minimal latency.
2. The architecture should be scalable with cloud-based solutions to accommodate a growing user base and new features.
3. A responsive design must ensure compatibility across desktop and mobile devices, while an intuitive and consistent user interface minimizes the learning curve.
4. To ensure data security, sensitive user information must be protected with end-to-end encryption, preventing unauthorized access. Additionally, all user actions should be authenticated using OAuth 2.0 standards.
5. The platform must maintain 99.9% uptime, ensuring high availability and reliability for users. Any downtime should be limited to scheduled maintenance periods, minimizing disruptions and optimizing system performance.

Design Outline

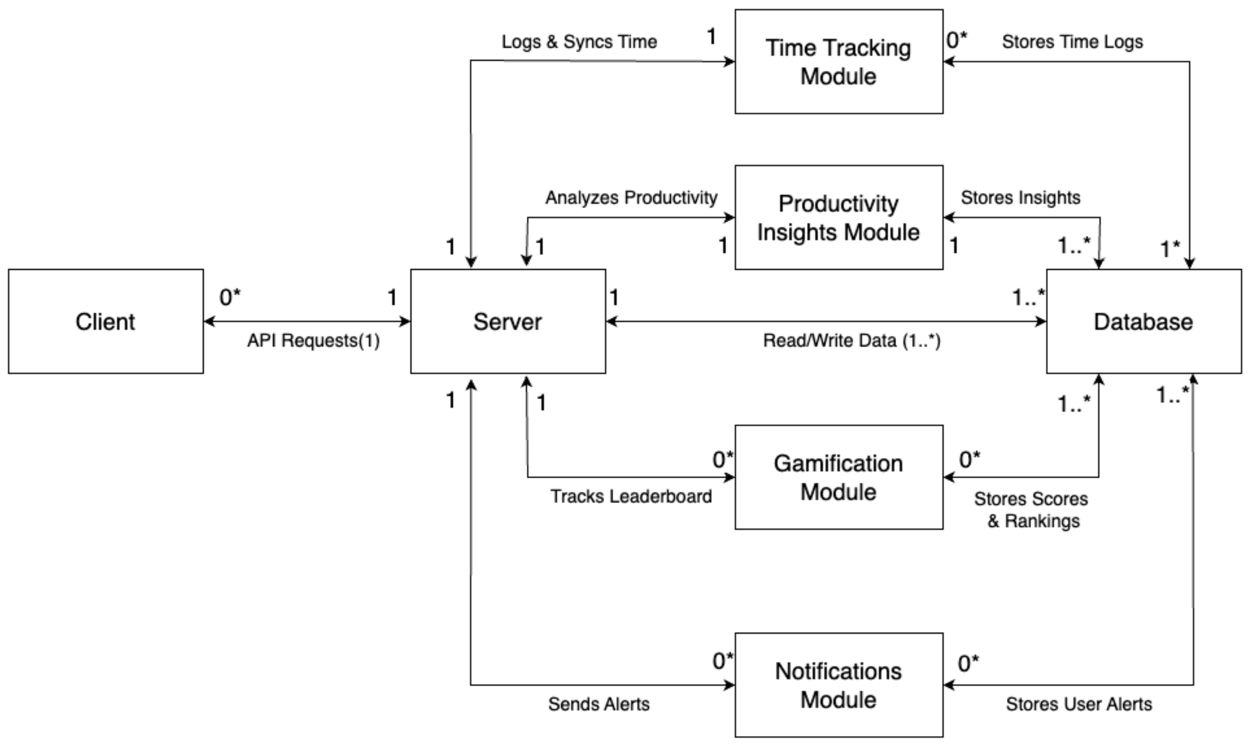
Planorama follows a client-server architecture, where users interact with the web/mobile client, which communicates with the server that processes requests and interacts with the database for persistent storage. The system ensures real-time synchronization and supports multiple concurrent users.

Components

1. **Client (Web Interface)**
   - The Client provides an interactive UI where users manage tasks, track productivity, and collaborate with teams.
   - The Client communicates with the Server via API requests to fetch, update, and sync data in real-time.
   - Users interact with features such as task creation, prioritization, goal setting, streak tracking, and notifications.
2. **Server (API & Business Logic)**
   - The Server processes API requests from multiple clients, handling authentication, user requests, and task operations.
   - It implements role-based permissions for teams and users.
   - It integrates with the Productivity Insight Module and Time Tracking Module to generate analytics.
3. **Database (Persistent Storage)**
   - The Database stores tasks, teams, users, activity logs, and productivity insights.
   - It maintains relationships between users, tasks, teams, and leaderboard rankings.
   - The data is structured using a relational database (PostgreSQL) to ensure efficiency and scalability.
4. **Backend Modules**
   - Time Tracking Module: Logs and analyzes time spent on tasks.
   - Productivity Insights Module: Tracks streaks, task completion trends, and productivity metrics.
   - Gamification Module: Manages leaderboards, rewards, and streak tracking.
   - Notification Module: Sends alerts to users about pending tasks and team updates.

High-Level Overview

Planorama is a **client-server-based productivity platform** designed to support task management, time tracking, goal setting, and team collaboration. The system consists of **three main components**: **Client, Server, and Database**, with additional **backend modules** for analytics and notifications.



Interactions Between Individual System Components

Planorama follows a **client-server model** with several **backend modules** that process data. Each component interacts in a structured way to ensure **smooth task management, productivity tracking, and collaboration**.

1. Client (Web UI) → Server (API & Business Logic)
Interaction Type: Request-Response (REST API)

- The Client is responsible for user interactions and sending API requests to the Server.
- Users create tasks, track progress, join teams, and set preferences via the client UI.
- The Server processes these requests and returns the required data.

2. Server → Database (PostgreSQL)
Interaction Type: Read-Write Queries

- The Server stores and retrieves data from the relational database.
- The Database maintains structured data on Users, Tasks, Teams, Logs, and Notifications.

3. Server → Time Tracking Module
Interaction Type: Log Processing

- The Server sends task duration logs to the Time Tracking Module.
- This module records estimated vs. actual time spent on tasks.
- It helps users monitor productivity trends.

4. Server → Productivity Insights Module
Interaction Type: Data Analysis & Insights

- This module analyzes user productivity data.
- Tracks streaks, task completion rates, and work patterns.
- Returns suggestions on workflow improvements.

5. Server → Gamification Module
Interaction Type: Leaderboard & Rewards Management

- Users earn points, badges, and ranks based on productivity.
- Teams have leaderboards to foster motivation.
- The Server interacts with this module to update rankings.

6. Server → Notification Module
Interaction Type: Push Alerts & Reminders

- Users get alerts for deadlines, messages, and team updates.
- The Notification Module processes event triggers to send reminders.

7. Time Tracking, Productivity Insights, Gamification, and Notification Modules → Database

Interaction Type: Data Storage & Retrieval

- Each module stores computed data back into the Database.
- The Database acts as a central repository for all system components.

Design Issues

Functional Issues

1. **How should tasks be categorized?**
    - Option 1: Allow users to create custom categories.
    - Option 2: Provide predefined categories.
    - Option 3: A hybrid approach where users can create custom categories while also selecting from a predefined list.
    - Choice: Option 3
    - Justification: A hybrid approach strikes a balance between structure and personalization. Predefined categories help standardize task organization, making it easier for users to categorize tasks quickly without having to think of new labels. However, allowing users to create custom categories ensures that the system remains adaptable to diverse workflows and industries. This flexibility is crucial for users who may have unique needs that predefined categories do not cover, while also preventing overwhelming users with an excessive number of choices from the outset.

2. **How should team collaboration be handled?**
    - Option 1: Assign team leaders with control over task assignments.
    - Option 2: Allow all team members to create, edit, and assign tasks.
    - Option 3: Role-based permissions where team leaders have additional control.
    - Choice: Option 3
    - Justification: While open collaboration is beneficial, having some level of structured control helps maintain efficiency, especially in larger teams. Role-based permissions ensure that team leaders can oversee the assignment of tasks, preventing mismanagement and ensuring accountability. At the same time, other team members still have the flexibility to contribute, edit, and collaborate within their designated roles. This approach fosters collaboration while preventing chaos, as task ownership and decision-making responsibilities remain clear and well-distributed.

3. **How should task dependencies be managed?**
    - Option 1: Allow users to manually link dependent tasks.
    - Option 2: Automatically suggest dependencies based on task descriptions and deadlines.

- Option 3: Allow manual linking for dependent tasks with smart suggestions.
- Choice: Option 3
- Justification: Dependencies between tasks can be complex, and users need full control over defining them based on their workflow. Manual linking ensures that dependencies reflect the actual relationships between tasks, avoiding incorrect automation that might lead to inefficiencies. However, smart suggestions can enhance productivity by identifying potential dependencies that users may have overlooked. This blend of automation and manual control provides efficiency without sacrificing accuracy, ultimately leading to smoother project execution.

4. **How should recurring tasks be handled?**
   - Option 1: Allow users to set custom recurrence rules.
   - Option 2: Provide predefined recurrence options (daily, weekly, monthly).
   - Option 3: Allow custom recurrence rules in addition to a set of predefined options.
   - Choice: Option 3
   - Justification: Different teams and individuals have varying needs when it comes to recurring tasks. Some may require simple scheduling (e.g., a meeting every Monday), while others might need highly customized recurrence rules (e.g., a task that repeats every third Wednesday). By offering both predefined options and custom rules, the system caters to a broad range of users. Those who prefer simplicity can rely on standard recurrence settings, while users with more complex needs can fine-tune recurrence schedules to fit their specific workflow.

5. **How should collaboration permissions be managed?**
   - Option 1: Give all team members full control over tasks.
   - Option 2: Implement role-based permissions (admin, editor, viewer).
   - Option 3: Assign task control randomly between team members.
   - Choice: Option 2
   - Justification: Giving all team members full control may work for small teams, but in larger or more structured environments, it can lead to confusion, errors, and unintentional modifications. Assigning task control randomly, on the other hand, would create inconsistency, making it difficult to ensure accountability and proper task management. Role-based permissions ensure that tasks are managed efficiently by allowing different levels of access. Admins can oversee and structure task assignments, editors can make necessary changes,

and viewers can stay informed without the risk of accidental edits. This structure maintains a balance between collaboration and security, ensuring smooth workflow management while preventing unauthorized changes.

6. **How should completed tasks be archived?**
    - Option 1: Automatically archive tasks after a set period.
    - Option 2: Allow users to manually archive tasks.
    - Option 3: Automatically archive tasks after a set period, with the option for users to manually archive or override.
    - Choice: Option 3
    - Justification: An automatic archiving system keeps the workspace organized by removing completed tasks after a set period, preventing clutter and making it easier to focus on active tasks. However, some tasks may require extended access before being archived, such as those needed for audits, reports, or future reference. Allowing manual archiving gives users the ability to keep completed tasks accessible when necessary while still benefiting from automated clean-up. This dual approach maintains efficiency without limiting flexibility.

Non-Functional Issues

1. **How should performance be optimized for large teams?**
    - Option 1: Store all data in memory for quick access.
    - Option 2: Use indexing and caching for frequent queries.
    - Option 3: Load data on demand from database, fetching fresh data for every request.
    - Choice: Option 2
    - Justification: While storing all data in memory may provide extremely fast access times, it becomes impractical as team sizes grow, leading to excessive memory usage and potential system instability. While loading data on demand ensures that users always see the most up-to-date information, it significantly increases database load by making redundant queries for frequently accessed data. This leads to slower response times and poor scalability, making it an inefficient choice for a platform designed to support thousands of users simultaneously. Instead, indexing and caching strike the right balance between speed and efficiency. Indexing optimizes database queries by allowing quick lookups, reducing the time needed to retrieve records. Caching frequently accessed data minimizes redundant computations and database load, ensuring high performance even when handling a

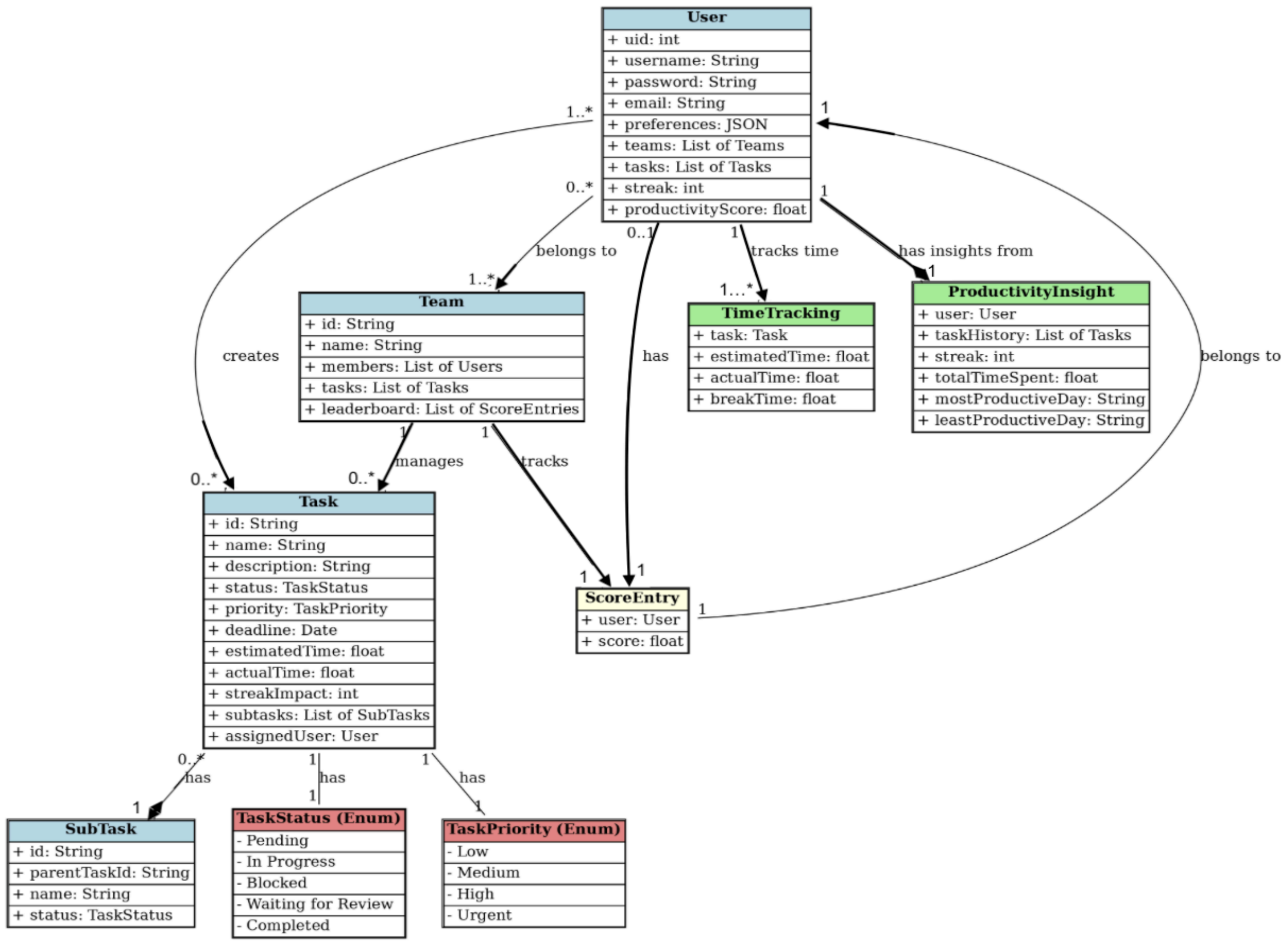
large number of users and tasks. This approach scales well, keeping response times low while avoiding unnecessary resource consumption.

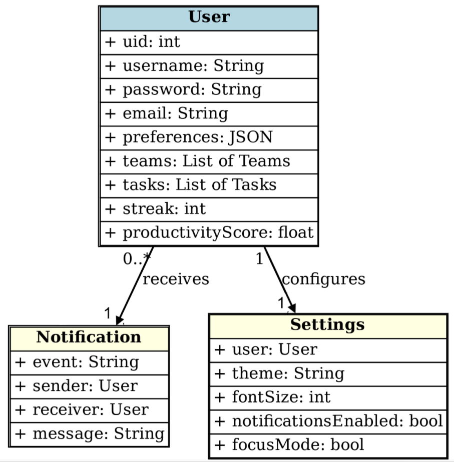
2. **How should data privacy be ensured?**
    - Option 1: Encrypt all user-related data.
    - Option 2: Encrypt only sensitive data (passwords, personal details).
    - Option 3: Allow users to choose which data to encrypt.
    - Choice: Option 1
    - Justification: Encrypting only sensitive data leaves other user-related information vulnerable to unauthorized access in case of a breach. Allowing users to choose which data to encrypt risks inconsistent security, as users may leave sensitive information unprotected, increasing the likelihood of data breaches. Encrypting all user-related data ensures maximum security, protecting not just passwords and personal details but also communications, task data, and collaboration history. This comprehensive approach fosters trust among users by ensuring their information remains confidential, preventing unauthorized access, and complying with data protection regulations. Additionally, full encryption minimizes risks associated with insider threats and accidental leaks, creating a more secure and privacy-conscious system.

3. **How should user activity be logged?**
    - Option 1: Store only recent actions (last 30 days).
    - Option 2: Keep a full history of all activities.
    - Option 3: Store recent actions but allow users to export history.
    - Choice: Option 3
    - Justification: Keeping a full history of all activities indefinitely can lead to significant storage overhead, slowing down the system and increasing maintenance costs. On the other hand, limiting logs to only the last 30 days may not be sufficient for auditing, tracking progress, or resolving disputes. A hybrid approach ensures efficiency while maintaining user control—recent actions are stored for quick access, while older logs can be exported if needed. This way, users can retrieve historical data when necessary without overloading the system, making it both practical and scalable.

Design Details

Class Design





Descriptions of Classes and Interactions

User

- Represents an individual using the application.
- Stores credentials, preferences, assigned tasks, teams, and productivity statistics.
- Users belong to one or multiple teams and can create, modify, and manage tasks.
- Tracks streaks and productivity scores based on task completion.

Task

- Represents an action item created by a user.
- Includes details such as description, priority, deadline, and assigned user.
- Has relationships with SubTasks, TaskStatus, and TaskPriority.
- Can have estimated and actual time tracked for productivity analysis.
- Contributes to streak tracking and leaderboard rankings.

SubTask

- Represents a smaller, dependent action item linked to a parent task.
- Used for breaking larger tasks into manageable steps.

Team

- Represents a collaborative group of users working on shared tasks.
- Contains a list of members, assigned tasks, and a leaderboard tracking productivity scores.
- Allows team leaders to manage tasks, permissions, and track members' activity.

TaskStatus (Enumeration)

- Defines different states of a task.
- Possible statuses: Pending, In Progress, Blocked, Waiting for Review, Completed.

TaskPriority (Enumeration)

- Categorizes task urgency levels.
- Possible levels: Low, Medium, High, Urgent.

ProductivityInsight

- Analyzes user productivity trends based on task history.
- Stores streaks, time spent, and identifies most and least productive days.
- Helps users adjust their workflow and optimize task completion efficiency.

TimeTracking

- Records estimated vs. actual time spent on tasks.
- Includes break time tracking to encourage healthy work habits.

Settings

- Allows users to customize their experience, including theme, font size, notifications, and focus mode.
- Ensures accessibility and personalization of the application.

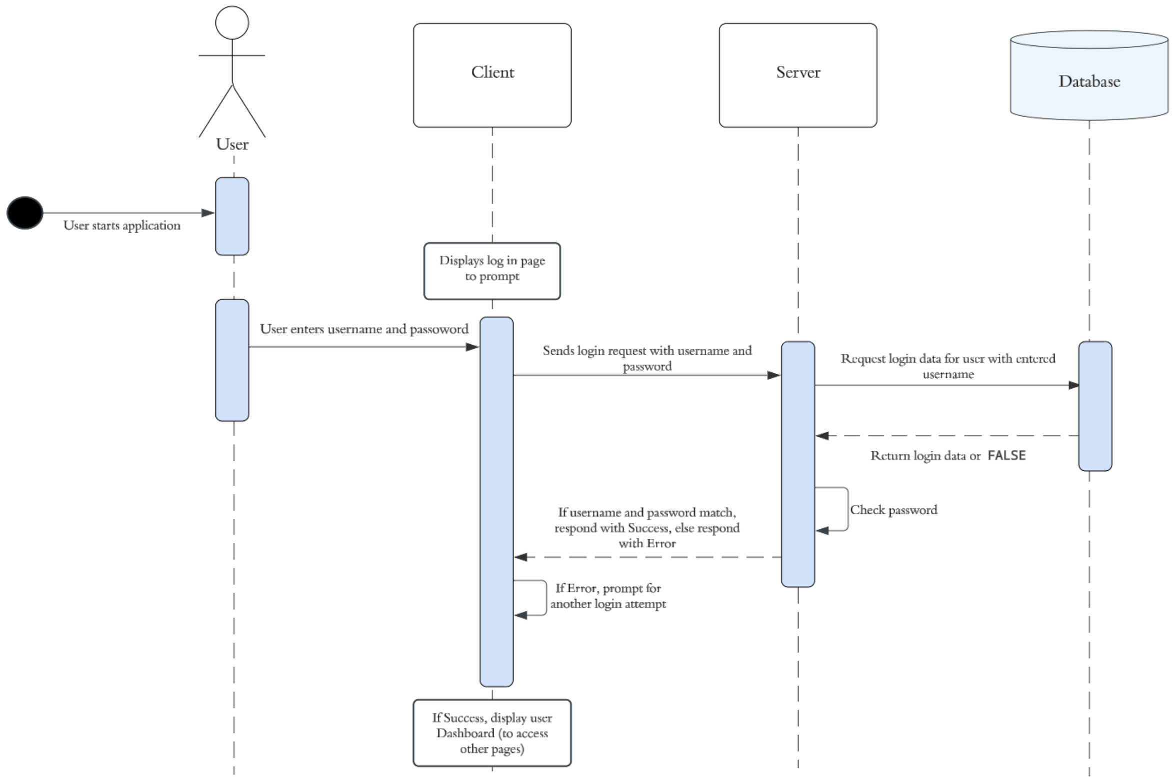
Notification

- Sends alerts for events such as task updates, due dates, and messages from team members.
- Ensures users stay informed of important activities.

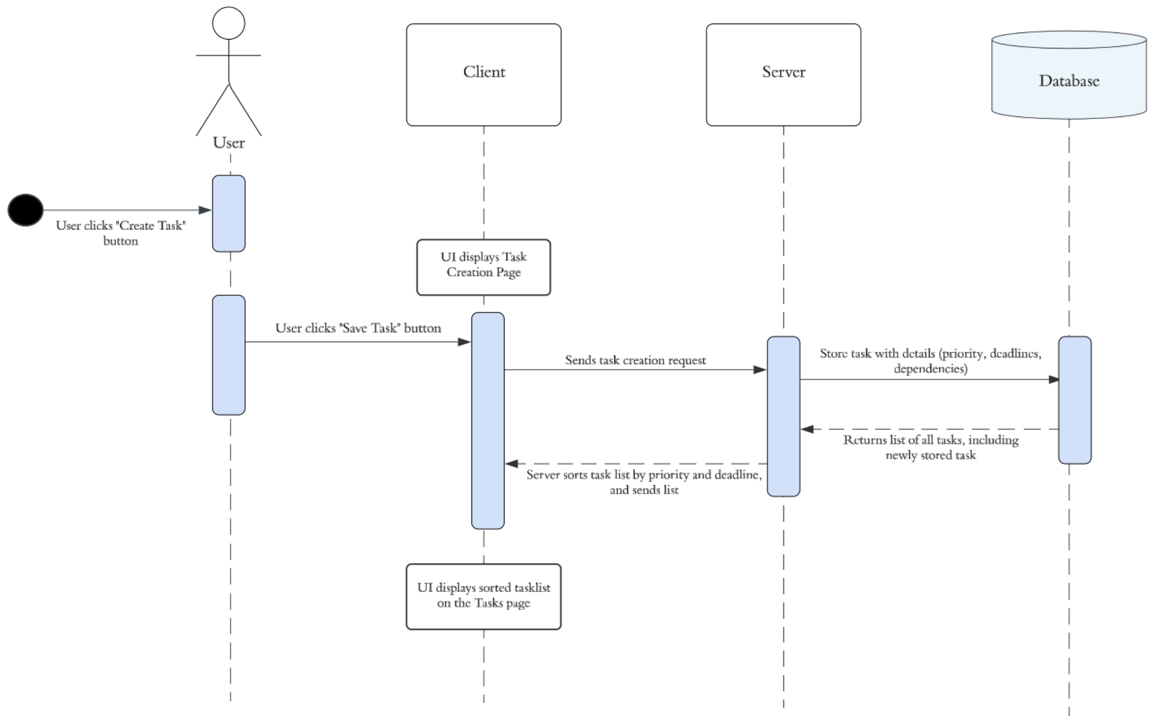
ScoreEntry

- Represents a user's ranking within a team.
- Stores the productivity score and updates the team leaderboard dynamically.

Sequence Diagram for When a User Starts the Application

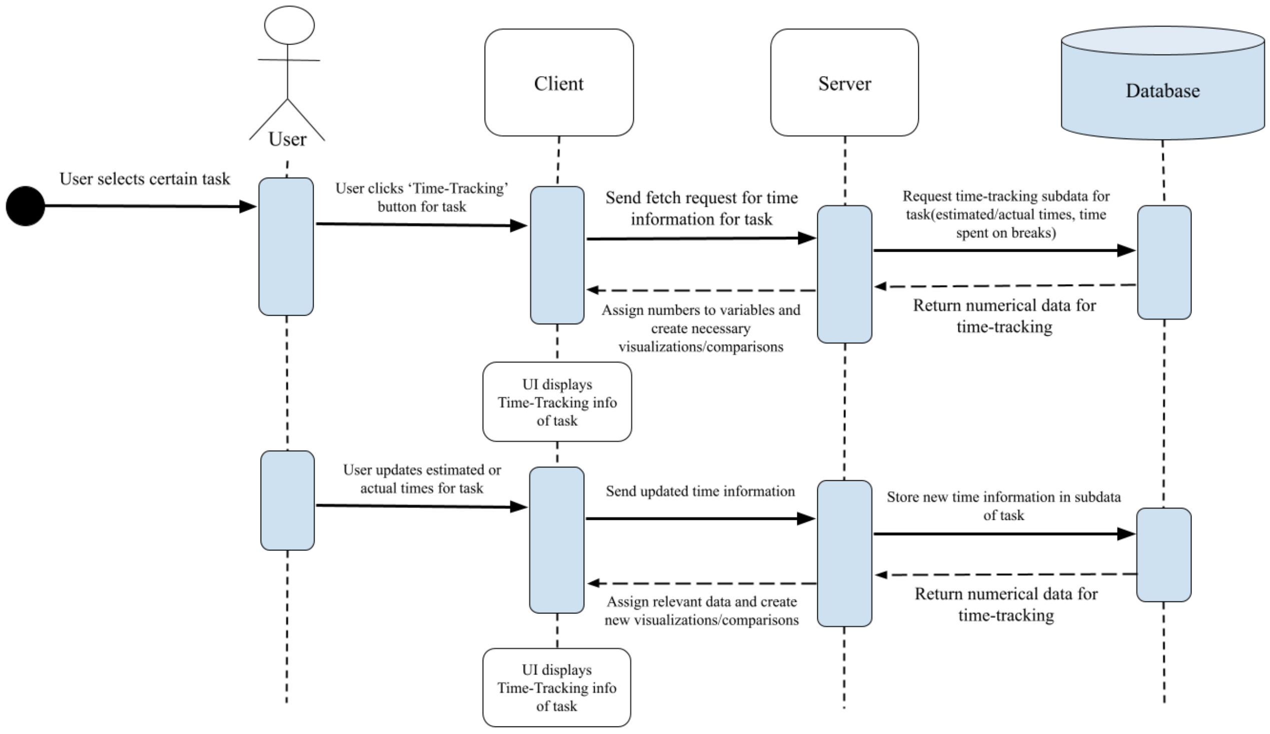

User
Client
Server
Database
User starts application
Displays log in page to prompt
User enters username and passoword
Sends login request with username and password
Request login data for user with entered username
Return login data or FALSE
Check password
If username and password match, respond with Success, else respond with Error
If Error, prompt for another login attempt
If Success, display user Dashboard (to access other pages)

Sequence Diagram for Task Creation



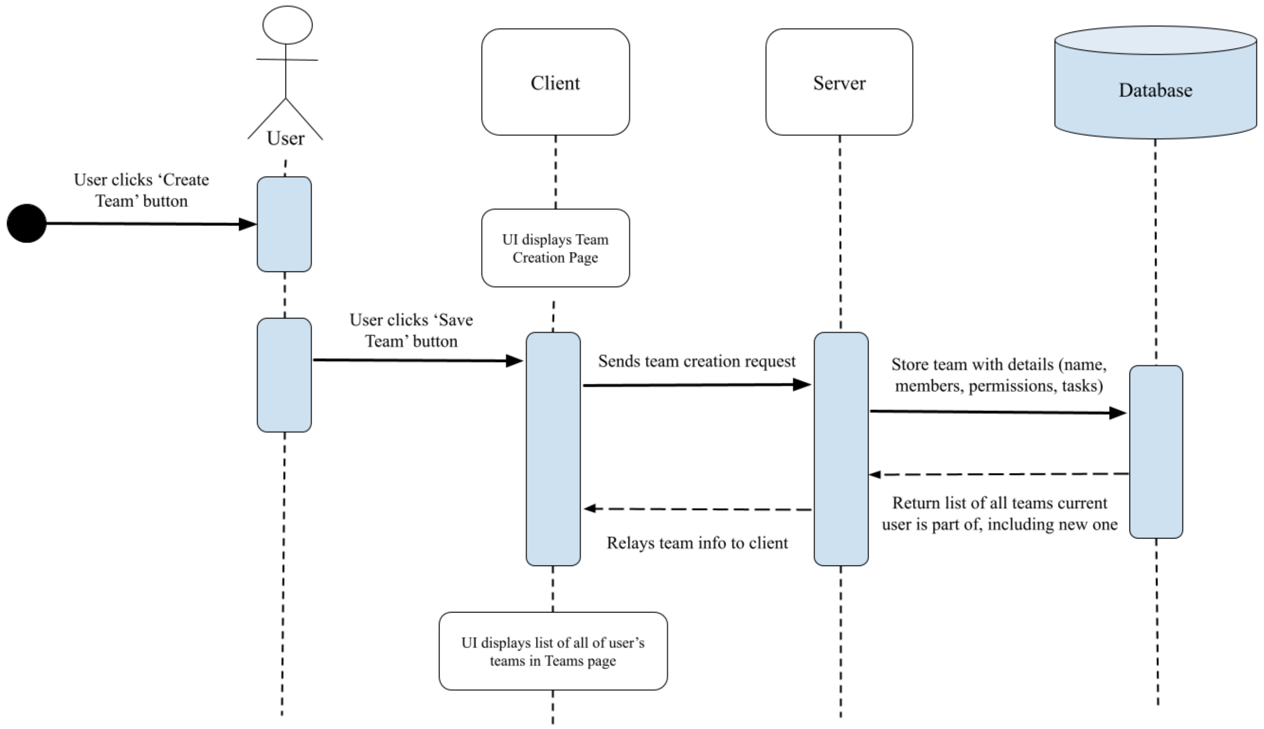
User
Client
Server
Database
User clicks "Create Task" button
UI displays Task Creation Page
User clicks "Save Task" button
Sends task creation request
Store task with details (priority, deadlines, dependencies)
Returns list of all tasks, including newly stored task
Server sorts task list by priority and deadline, and sends list
UI displays sorted tasklist on the Tasks page

Sequence Diagram for Time-Tracking

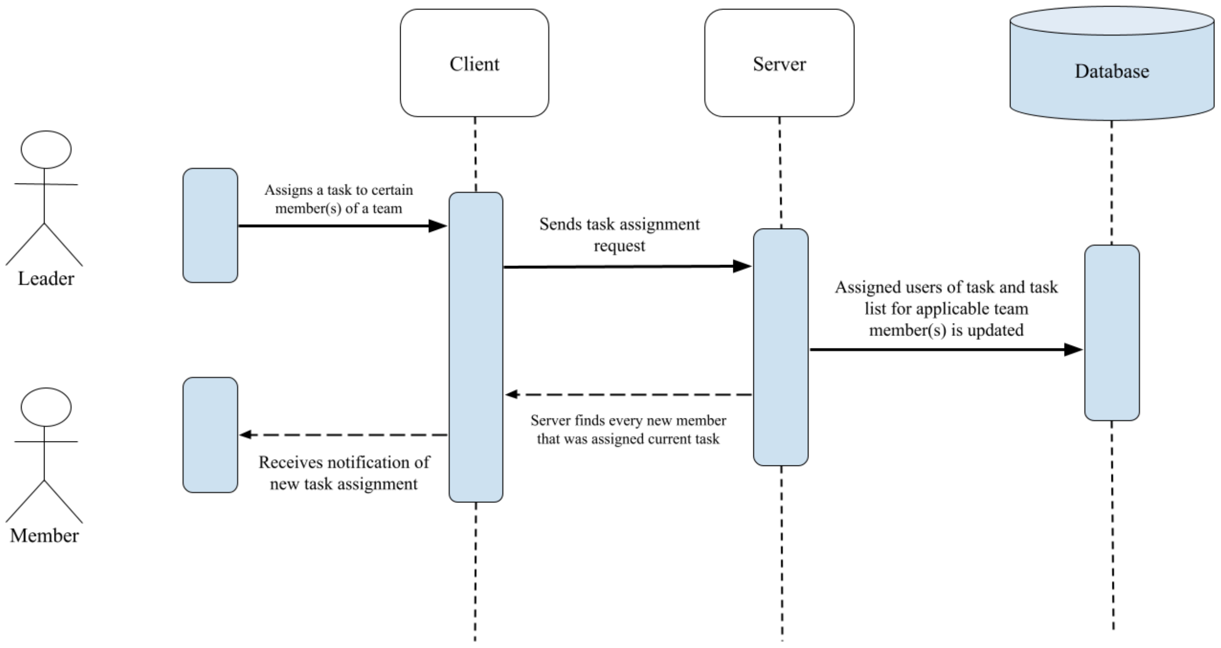


User
Client
Server
Database
User selects certain task
User clicks 'Time-Tracking' button for task
Send fetch request for time information for task
Request time-tracking subdata for task(estimated/actual times, time spent on breaks)
Return numerical data for time-tracking
Assign numbers to variables and create necessary visualizations/comparisons
UI displays Time-Tracking info of task
User updates estimated or actual times for task
Send updated time information
Store new time information in subdata of task
Return numerical data for time-tracking
Assign relevant data and create new visualizations/comparisons
UI displays Time-Tracking info of task

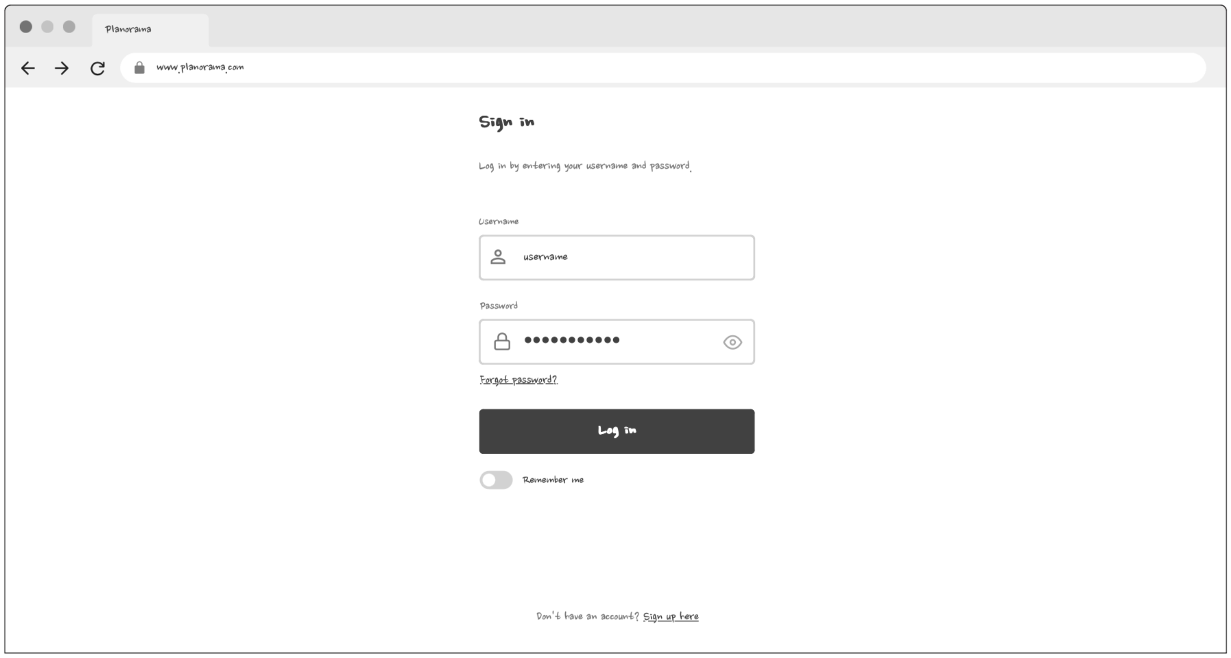
Sequence Diagram for Team Creation


User
Client
Server
Database
User clicks 'Create Team' button
UI displays Team Creation Page
User clicks 'Save Team' button
Sends team creation request
Store team with details (name, members, permissions, tasks)
Return list of all teams current user is part of, including new one
Relays team info to client
UI displays list of all of user's teams in Teams page

Sequence Diagram to Team Task Assignment


Client
Server
Database
Leader
Member
Assigns a task to certain member(s) of a team
Sends task assignment request
Assigned users of task and task list for applicable team member(s) is updated
Server finds every new member that was assigned current task
Receives notification of new task assignment

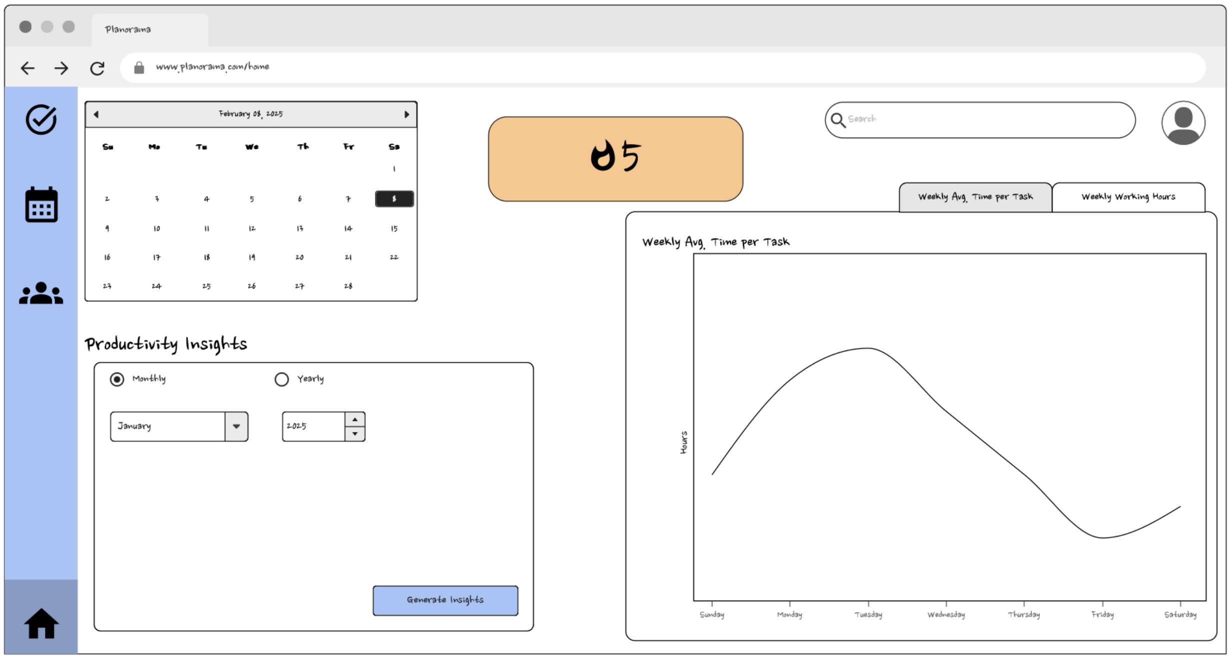
UI Mockups



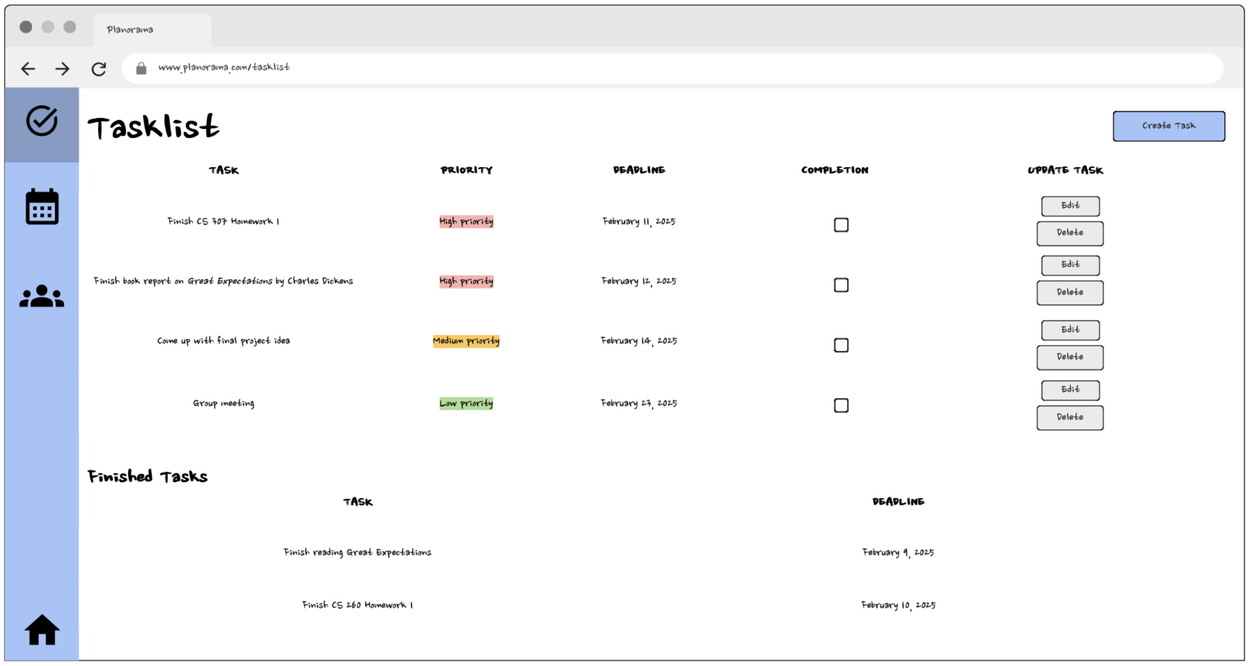
Log in by entering your username and password.

Don't have an account? Sign up here

Login page



User home dashboard



Tasklist

TASK	PRIORITY	DEADLINE	COMPLETION	UPDATE TASK
Finish CS 307 Homework I	High priority	February 11, 2025	☐	Edit Delete
Finish book report on *Great Expectations* by Charles Dickens	High priority	February 12, 2025	☐	Edit Delete
Come up with final project idea	Medium priority	February 14, 2025	☐	Edit Delete
Group meeting	Low priority	February 23, 2025	☐	Edit Delete

Finished Tasks

TASK	DEADLINE
Finish reading Great Expectations	February 9, 2025
Finish CS 260 Homework I	February 10, 2025

Tasklist page