Project Name: Task Monitoring System

Your Name: Camden Thomson

Date: 3/18/25

Table of Contents

---

1. Problem Statement & System Requirements

Problem Statement

Many individuals and businesses struggle with tracking tasks efficiently, leading to missed deadlines, decreased productivity, and miscommunication. Existing solutions often lack key features like collaboration, deadline notifications, and task categorization.

This project aims to create a **Task Monitoring System** that provides users with an intuitive way to create, track, and complete tasks, improve team collaboration, and ensure timely task completion.

System Requirements

Hardware Requirements

- **Server Hosting**: At least 4GB RAM and 50GB storage
- **Development Machines**: Minimum 8GB RAM
- **Internet Connectivity**: Required for real-time collaboration

Software Requirements

- **Frontend**: React.js, Material UI
- **Backend**: Node.js, Express.js
- **Database**: MongoDB
- **Authentication**: JWT (JSON Web Token)
- **Hosting**: AWS/GCP Cloud Services
- **Development Tools**: GitHub, Notepad++

---

2. Functional Requirements Specification

Task Management

- Users should be able to create tasks with a title, description, status (not started, in progress, completed), priority level, and due date.
- Tasks should be assignable to specific users or teams.
- Tasks must be categorized (e.g., work, personal, etc.).

User Authentication

- Users must securely sign up, log in, and manage their accounts.
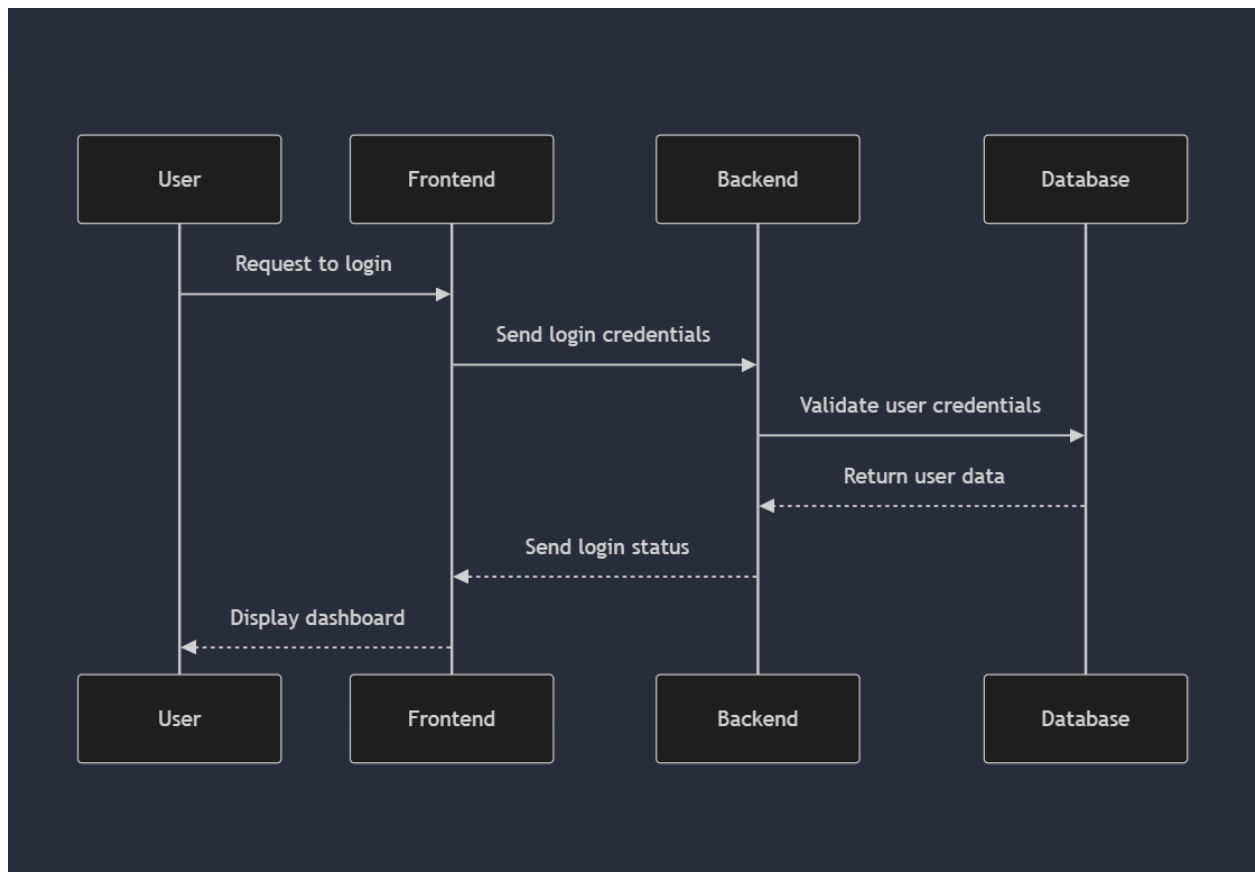- User roles should be supported, with admins and standard users having different permissions.

Collaboration Features

- Team-based task assignment and real-time updates on task progress.
- Task comments and file attachments for team collaboration.

Notifications & Alerts

- Automated email and in-app notifications for approaching deadlines.
- Alerts should be triggered for overdue tasks or task status changes.
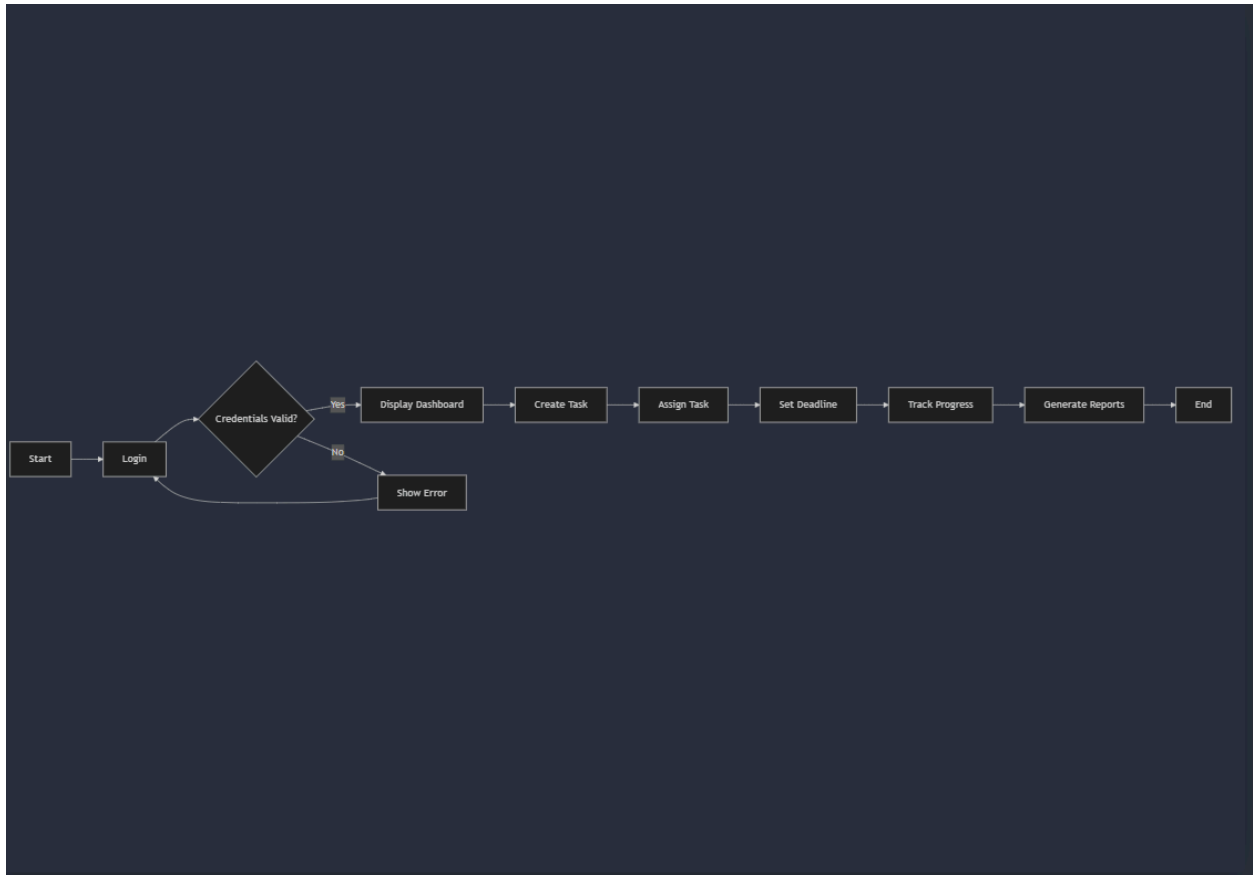
---

3. System Sequence Diagram

This **System Sequence Diagram** shows the basic login process in the system, from the user interacting with the frontend to backend validation and data retrieval from the database.

---

4. Activity Diagram

This **Activity Diagram** outlines the flow of activities from user login to generating reports and ending the process.

---

5. User Interface Specification

Dashboard UI

- **Components**:
    - Task List: Displays tasks with titles, statuses, due dates, and priorities.
    - Filters: Users can filter tasks by status, priority, or due date.
    - Navigation Menu: Links to other sections like Task Creation and Reports.

Task Creation Form UI

- **Components**:
    - Input Fields: Task title, description, due date, priority.
    - Status Dropdown: Allows users to set task status (Not Started, In Progress, Completed).

○　Submit Button: Allows users to create a new task.

---

6. Project Plan

Phase	Tasks	Duration
Planning	Define requirements, tech stack, UI mockups	1 week
UI/UX Design	Create wireframes, user journeys, and UI design	2 weeks
Backend Setup	Implement API endpoints, authentication, DB models	3 weeks
Frontend Development	Build UI components, integrate backend APIs	3 weeks
Testing	Unit tests, integration tests, user feedback	2 weeks
Deployment	Deploy to cloud, optimize performance, and launch	1 week
Maintenance	Monitor performance, gather feedback, and iterate	Ongoing

7. References

- React.js Documentation
- Material UI Documentation
- Node.js Documentation
- MongoDB Documentation

---

8. Highlighted Changes

- **Section on Functional Requirements**: Modified to include **notifications** and **task categorization** features, as per feedback.
- **Activity Diagram**: Revised the flow of tasks based on the **collaboration features** feedback.