Deliverable A – SE D3

Software Project Plan

ProjectPulse

Group Members:

Tauha Imran	22i-1239,
Minahil Ali	22i-0849,
Nabeed Haider	22i-0871

Project Title: **Project Pulse**
Submission Date: 28th April, 2025

---

1. Introduction

This document outlines the **Software Project Plan** for the MERN Stack-based **Task Management System** developed by *Team Pulse Studios*. The project is designed to streamline task delegation, team coordination, and project monitoring through an intuitive and feature-rich platform.

Objectives of the Plan:

- Define a clear roadmap for project execution across multiple sprints
- Establish deliverables and timelines for each development phase
- Identify task dependencies and resource allocation
- Facilitate tracking and reporting using structured breakdowns

Tools Used:

- **Gantt Chart Generation:** Python (Matplotlib)
- **Project Planning and Tracking:** GitHub Projects, Notion
- **Version Control:** Git
- **IDE and Development Tools:** Visual Studio Code, MongoDB Compass, Postman

---

2. Work Breakdown Structure (WBS)

2.1 WBS Overview

The Work Breakdown Structure (WBS) serves as a foundation for project execution by dividing the Task Management System into manageable phases and tasks. It outlines a hierarchical structure that breaks down the entire project scope into defined activities, helping the team manage complexity, monitor progress, and allocate resources efficiently.

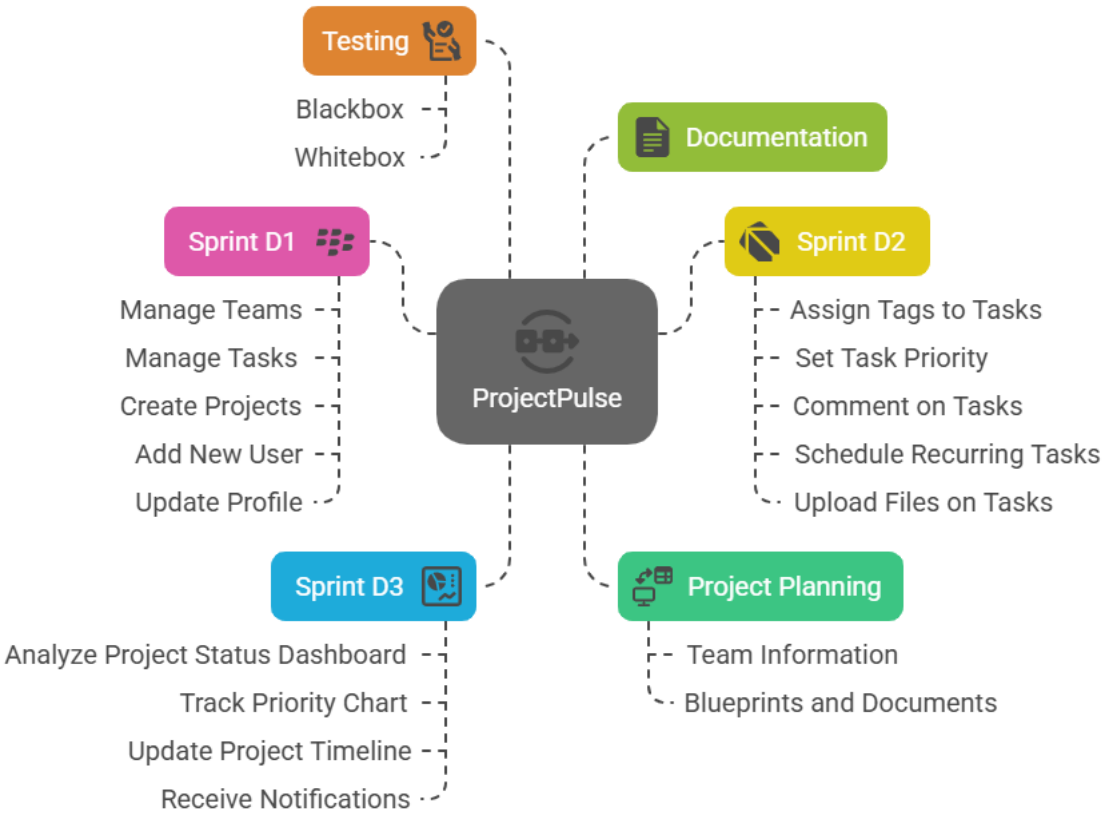
This WBS is designed around six key phases:

1. Initial project planning and setup
2. Three development sprints (D1, D2, D3) aligned with grouped use cases
3. Final testing and deployment
4. Comprehensive documentation

Each sprint focuses on a specific set of use cases to ensure modular development and continuous integration. Dependencies are clearly identified to ensure sequential flow and avoid bottlenecks.

2.2 Hierarchical WBS Diagram

A visual format showing project decomposition



2.3 Tabular WBS with Details

WBS ID	Task Name	Description	Estimated Duration
1.0	Project Planning	Requirement gathering, initial repo setup	21 days
2.0	Sprint D1 – Core Features	Team mgmt, task creation, project setup	10 days
3.0	Sprint D2 – Intermediate	Time tracking, recurring tasks, file uploads	10 days
4.0	Sprint D3 – Advanced	Gantt charts, timeline updates, document versioning	18 days
5.0	Testing & Deployment	Test cases, bug fixing, deployment	8 days
6.0	Documentation	SRS, API docs, user manuals	8 days

---

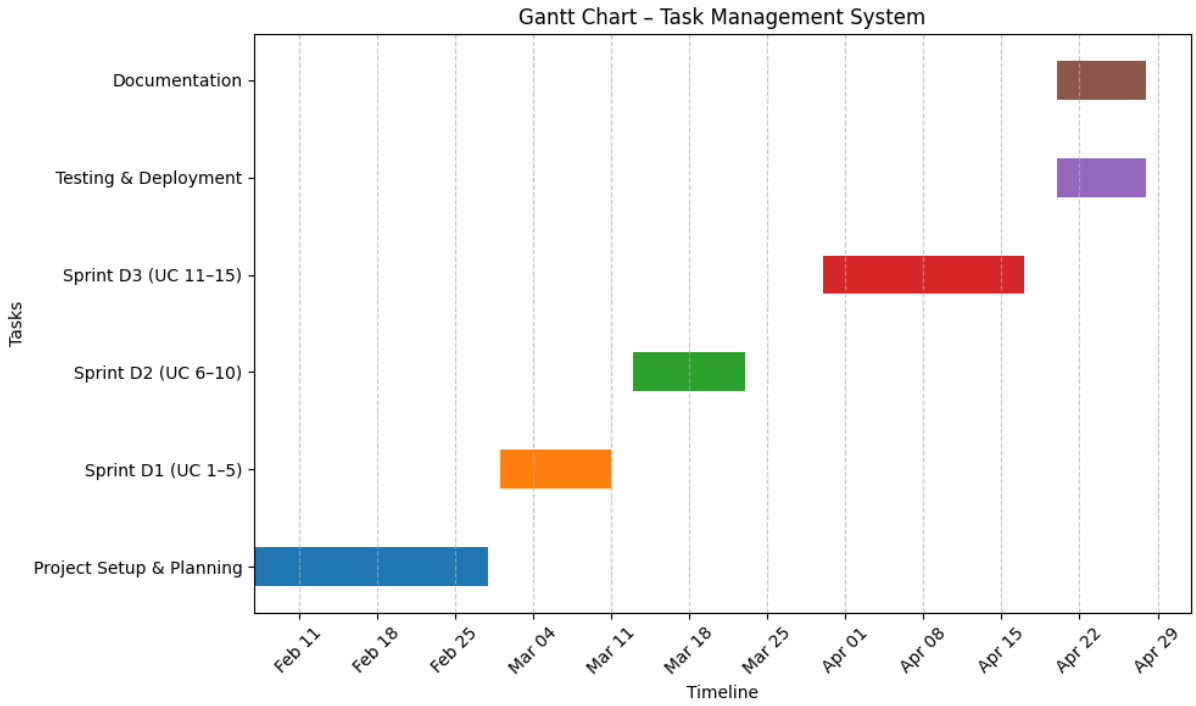
3. Project Gantt Chart

3.1 Gantt Chart Overview

The Gantt chart illustrates the overall timeline and task distribution for developing the Task Management System. The project spans from early February to the end of April 2025, segmented into six primary phases.

3.2 Gantt Chart Diagram

Below is the Gantt chart for the development of this project.



Phase Breakdown

1. **Project Setup & Planning**
    - **Duration:** February 5 – February 28
    - Activities included requirement gathering, initial planning, project repository setup, and design discussions.
2. **Sprint D1 (Use Cases 1–5)**
    - **Duration:** February 28 – March 11
    - Developing core features such as team management, task creation, project setup, and priority assignment.
3. **Sprint D2 (Use Cases 6–10)**
    - **Duration:** March 11 – March 23
    - Implementation of intermediate-level features including time tracking, recurring tasks, notifications, and document uploads.
4. **Sprint D3 (Use Cases 11–15)**
    - **Duration:** March 29 – April 17
    - Focused on advanced functionalities like Gantt chart visualization, document versioning, project timeline updates, and progress reporting.
5. **Testing & Deployment**
    - **Duration:** April 19 – April 28

- Comprehensive testing, bug fixing, and final deployment to a production environment.

6. **Documentation**
    - **Duration:** April 20 – April 28
    - Preparation of final project documentation, including Software Requirements Specification (SRS), user manuals, API references, and deployment guides.

Overlapping Activities

The final phase includes parallel execution of **Testing & Deployment** and **Documentation**, enabling efficient project closure and timely delivery.

---

4. Conclusion

The Software Project Plan for **Project Pulse** lays out a detailed roadmap that ensures systematic development, efficient task allocation, and timely delivery. By decomposing the work into well-defined phases and incorporating tools like Gantt charts and WBS, the team is equipped to tackle each sprint with clarity and coordination.

Structured planning helps in minimizing risks, tracking dependencies, and adapting to evolving requirements. The plan not only facilitates smoother collaboration among team members but also provides transparency for stakeholders through measurable milestones. Overall, this planning approach is expected to enhance productivity, ensure quality, and deliver a robust task management system by the end of April 2025.
