Software Design Document

Mini Project Management Software

Akhilesh M K

Abhinav R Bharadwaj

Abhiram H A

Akash Vijayakumar Angadi

Table of Contents

1. Introduction

1.1 Purpose

The Mini Project Management System Design Document outlines the architecture and system design for the development of a compact project management software solution. This system, is engineered to streamline and enhance project planning, monitoring, and execution for small to medium-sized projects. This document is created for the benefit of Project Managers, Software Engineers, and all stakeholders involved in the realization of this software application.

1.2 Scope

The Project Management System addresses the management of software projects. It provides the framework for organizing and managing resources in such a way that these resources deliver all the work required to complete a software project within defined scope, time and cost constraints. The Project Management System serves as a central hub for planning, executing, and tracking software projects. It fosters collaboration among project stakeholders, enabling effective communication and decision-making. Additionally, it facilitates the efficient allocation of resources, ensuring that tasks are assigned to the right team members with the necessary skills and expertise. The Project Management System not only streamlines project execution but also supports comprehensive project documentation and reporting. It maintains a detailed project history, including task progress, milestones achieved, and any changes made to the project scope, ensuring transparency and accountability throughout the project's lifecycle.

1.3 Definitions, Acronyms and Abbreviations

Acronym	Meaning
MPMS	Mini Project Management System
UI	User Interface
API	Application Programming Interface
GUI	Graphical User Interface
JSON	JavaScript Object Notation

2. System Overview

- Backend

  The backend of the MPMS is mainly responsible for data management and communication with the front-end. Python has been selected as the primary programming language to power the backend due to its rich ecosystem of libraries, and suitability for web development. The backend is further enhanced by the use of JSON files for data storage. This storage method ensures data persistence and allows for quick and easy retrieval of project-related information.

- Front End

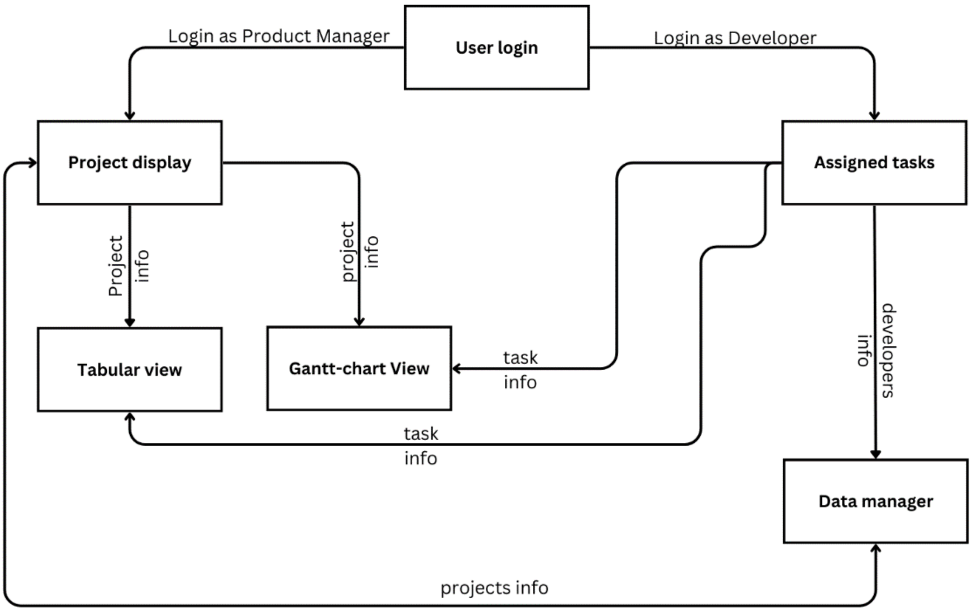
  Streamlit, a popular web application framework for Python, serves as the front-end component of the MPMS. Streamlit provides a simple and easy-to-use interface that makes it easy for users to interact with the system without any difficulties. Its ease of use allows project managers and team members to access and manipulate project data without requiring extensive technical knowledge.

- Data Storage

  The MPMS leverages JSON files for data storage. JSON provides a lightweight and human-readable format for storing project-related data. This approach simplifies data management and retrieval, making it a practical choice for a compact project management system. JSON files store essential information, such as project details, task assignments, user profiles, and system configurations.

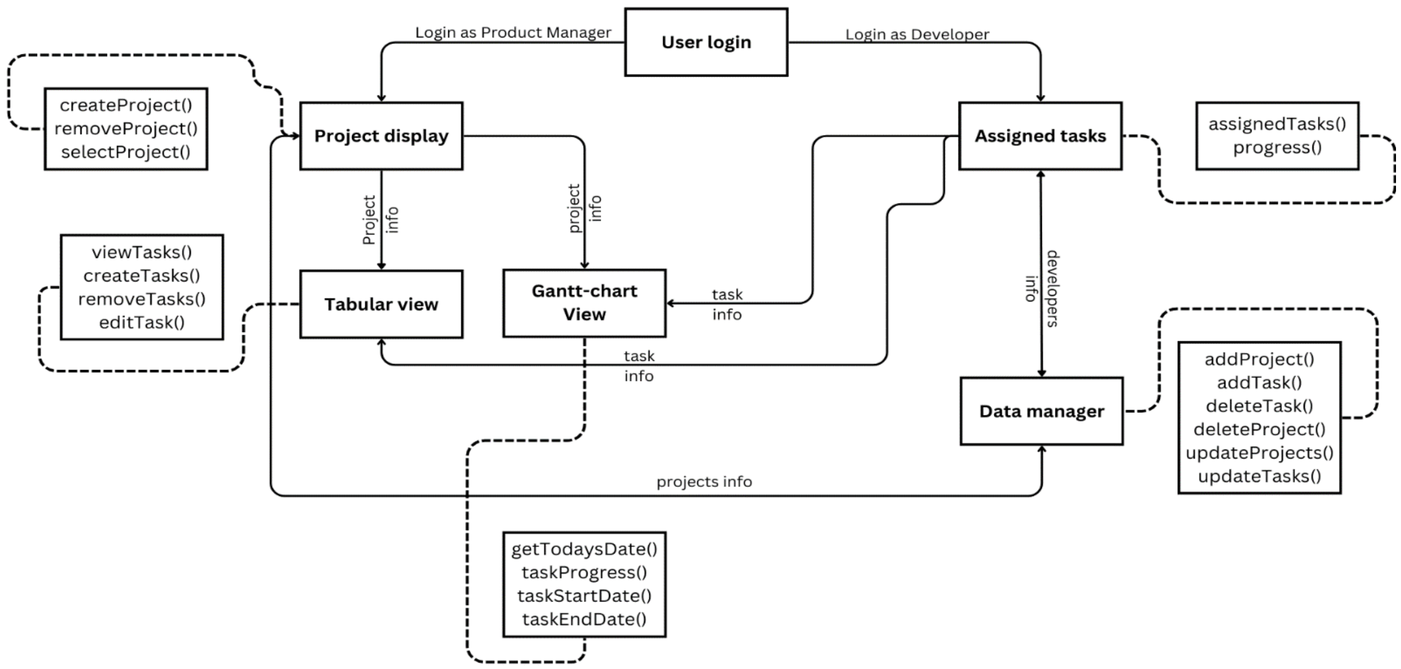
3. System Components

3.1 Decomposition Description



The diagram above provides a top-down overview of the anticipated functionality of a web application and how its various components will interact. The "user login" feature enables both PMs and developers to securely log in to the portal. The "project display" feature presents a comprehensive list of all the projects managed by the Product Manager. The "Tabular view" offers an organized presentation of the subtasks associated with one of the PM's projects. The "Gantt-chart view" offers a visual representation of project timelines, progress, and resource allocation. Lastly, the "assigned tasks" feature displays a developer's tasks in both tabular and Gantt-chart formats for easy reference.

3.2 Dependency Description



The diagram above illustrates the component structure of the MPMS (Multi-Project Management System) and the interdependencies between its modules. Dotted lines are utilized to denote functions integrated into these modules. Users, including Project Managers (PMs) and developers, access the system through the "User Login" module.

When a PM logs in, the system redirects them to the "Project Display" module. Within this module, PMs can perform actions such as creating new projects, removing completed or decommissioned projects, or selecting projects under their supervision from a list. Upon selecting a project, the system further redirects to the "Tabular View" and "Gantt-Chart View." Here, PMs can manipulate tasks within the selected project by adding, editing, or deleting them. A responsive Gantt chart dynamically reflects these changes, displaying progress percentages and calculating remaining days for each task.

In the case of a developer logging in, they are directed to the "Display Tasks" module. Within this module, developers can view tasks assigned to them, along with associated project deadlines. Developers can also update task progress.

All changes, whether initiated by PMs or developers, are recorded and managed by the "Data Manager" module, which ensures that edits, deletions, and additions are accurately reflected in the database.

4. Detailed Design

4.1 RTM

Requirement-ID	Requirement Description	Design Component
4.1	User roles	User login
4.2	User profile	User login
4.3	Manage Project List	Project display
4.4	Manage Project Leader	Tabular view
4.5	Manage task	Assigned task