Namal University, Mianwali

Department of Computer Science

Course Name:
CSC-225 – Software Engineering

System Design Document

AI-Based Task Manager & Smart Scheduler

Version 1.0
Approved

Prepared By

Mishaal Aqdas (NUM-BSCS-2024-35)
Muhammad Imran (NUM-BSCS-2024-50)
Khadija Tul Kubra (NUM-BSCS-2024-30)

Submitted To

Mam Asiya Batool

Date: 18 January 2026

Contents

1 Introduction

The purpose of this design report is to present the system design for the **AI-Based Task Manager & Smart Scheduler** as part of Project Milestone 3. This document translates the approved Software Requirements Specification (SRS) into a structured design that defines system behavior, constraints, and major design decisions prior to implementation. The design focuses on clarity, modularity, and feasibility to ensure smooth development and future scalability.

2 Design Assumptions and Constraints

2.1 Design Assumptions

- Users will have access to smartphones with basic internet connectivity.
- The system will be used primarily by individual users for personal task management.
- AI-based scheduling decisions are based on historical task data and user preferences.
- External services such as notification and calendar APIs will be available and reliable.

2.2 Design Constraints

- Limited computational resources on mobile devices.
- Dependency on third-party APIs for notifications and calendar integration.
- Offline functionality is restricted to cached data only.
- Security and privacy constraints for storing user task data.

3 Key Design Decisions

Several key design decisions were made to ensure system efficiency and maintainability:

- A layered architecture was selected to separate concerns and simplify maintenance.
- AI logic was isolated into a dedicated layer to allow independent enhancement of intelligent features.
- Modular services were used to support scalability and future feature expansion.
- Cloud-based storage was chosen to enable data backup and synchronization across devices.

3.1 Figma Prototyping

Figma was used to design and prototype the system diagrams, including Data Flow Diagrams (DFDs). The interactive prototype helps visualize system processes and validate design decisions before implementation.
Prototype Link:
Figma System Design Prototype

4 Use Case Design Description

Use case design identifies system actors and major functionalities. Registered users manage tasks, schedules, notifications, and progress. System administrators manage users and monitor system operations.

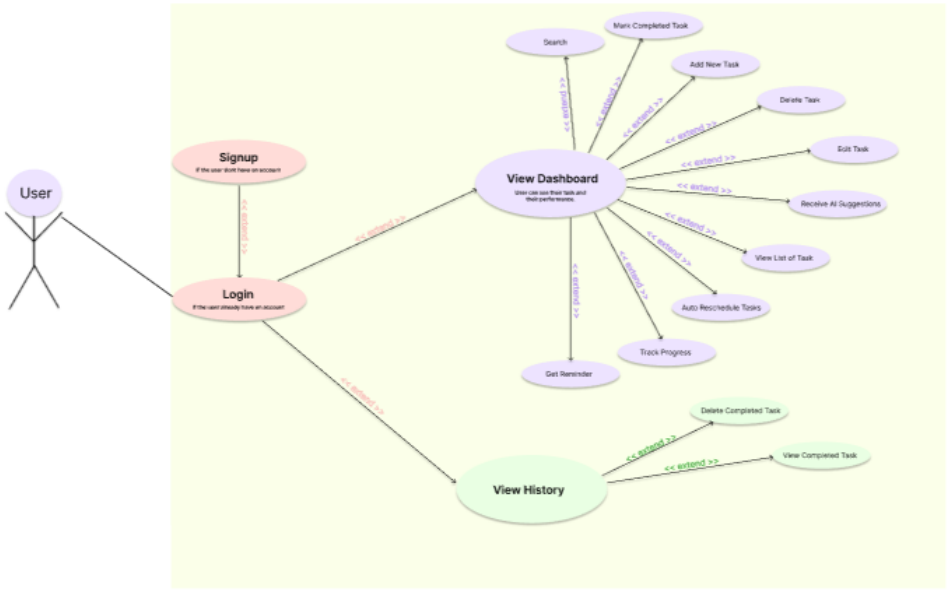


Figure 1: Use Case Diagram

Link: Use Case Diagram Link
Diagram Instruction: Design a UML Use Case Diagram showing:

- Actor: User, Admin
- Use cases: Create Task, Schedule Task, Receive Reminder, AI Reschedule

5 Data Flow Design Description

Data Flow Diagrams (DFDs) represent how data moves through the system.

5.1 Level 0 DFD

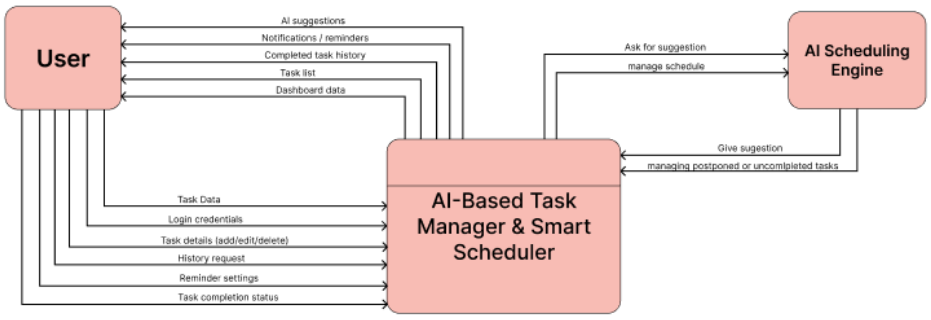


Figure 2: DFD Level 0

Link: DFD 0 Diagram Link

5.2 Level 1 and Level 2 DFD

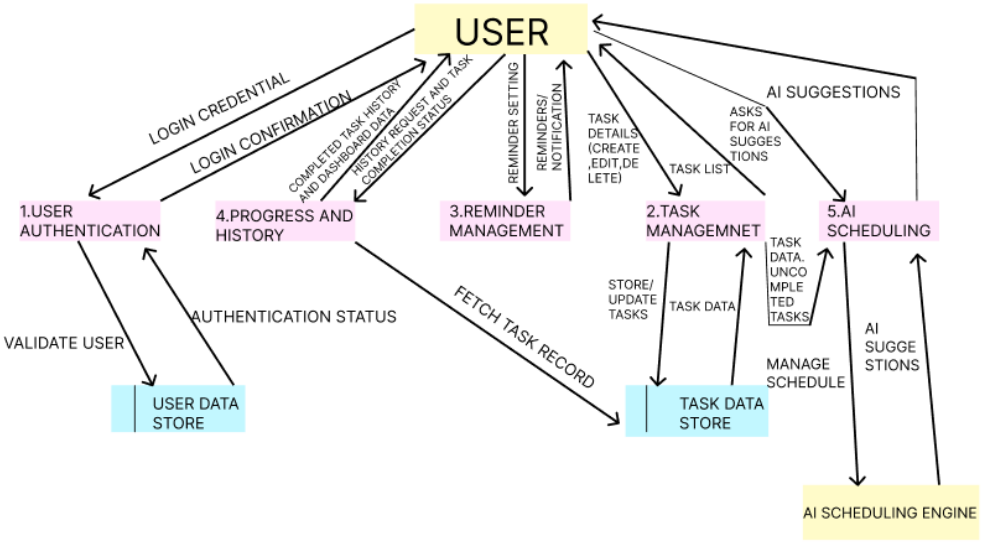


Figure 3: DFD Level 1

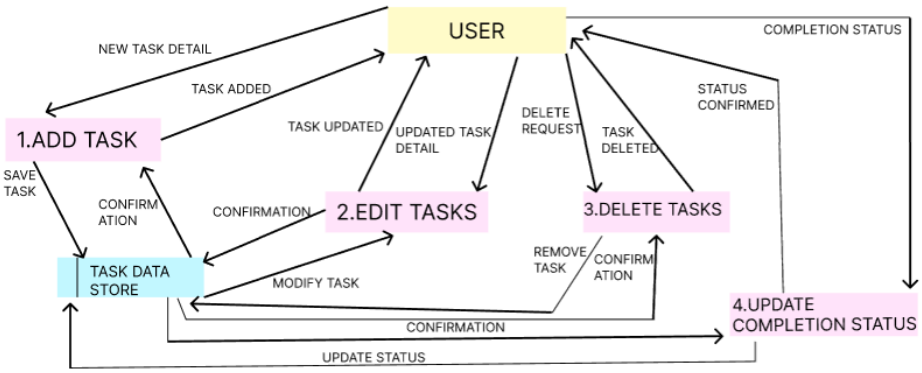


Figure 4: DFD Level 2

Diagram Instruction:
DFD is created using:

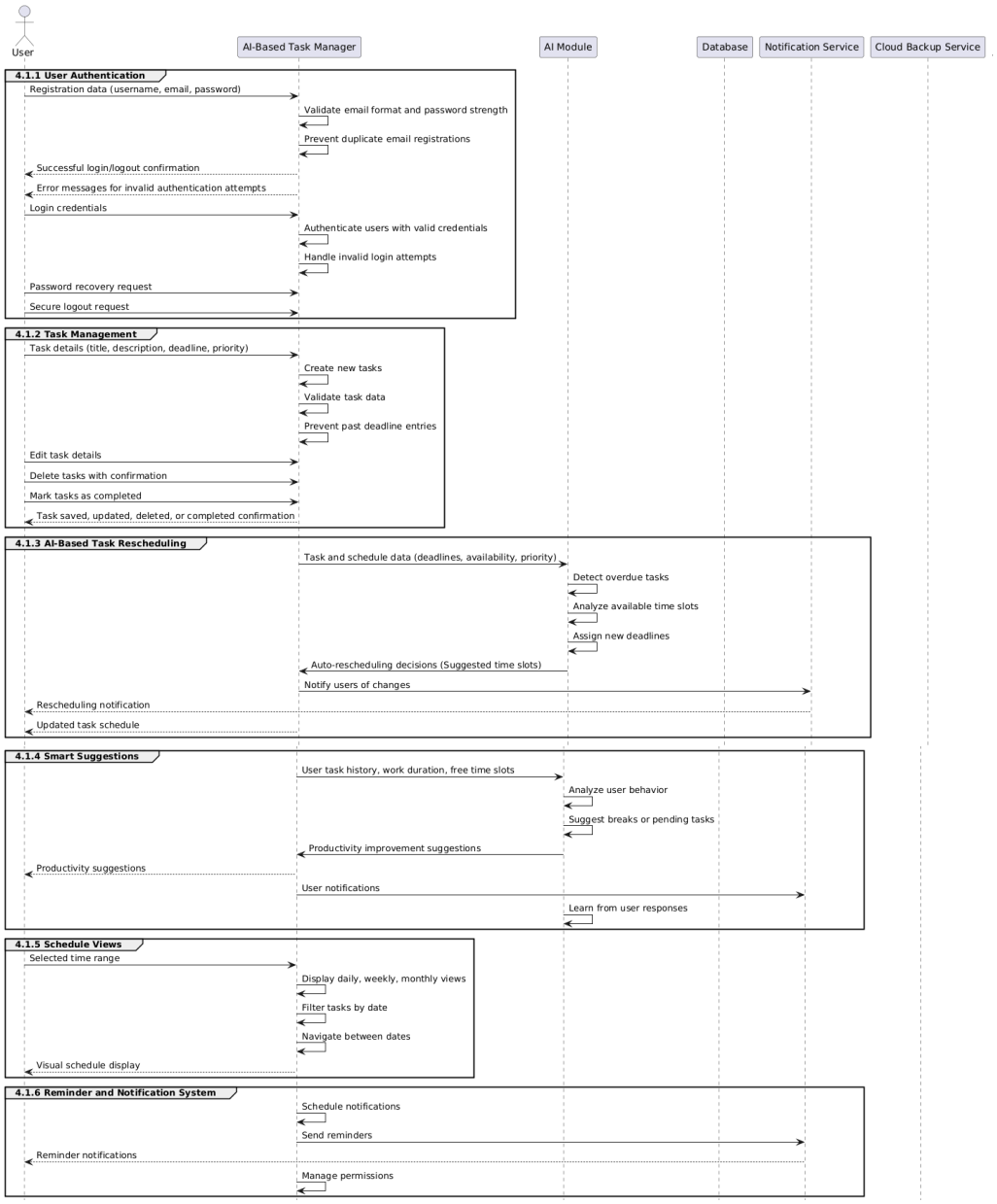
- figma

DFD 1 and 2 Diagram Link:

`Figma Design - DFD1 Diagrams`

6 Sequence Diagram

Sequence diagrams model interactions between users, system components, databases, and external services.



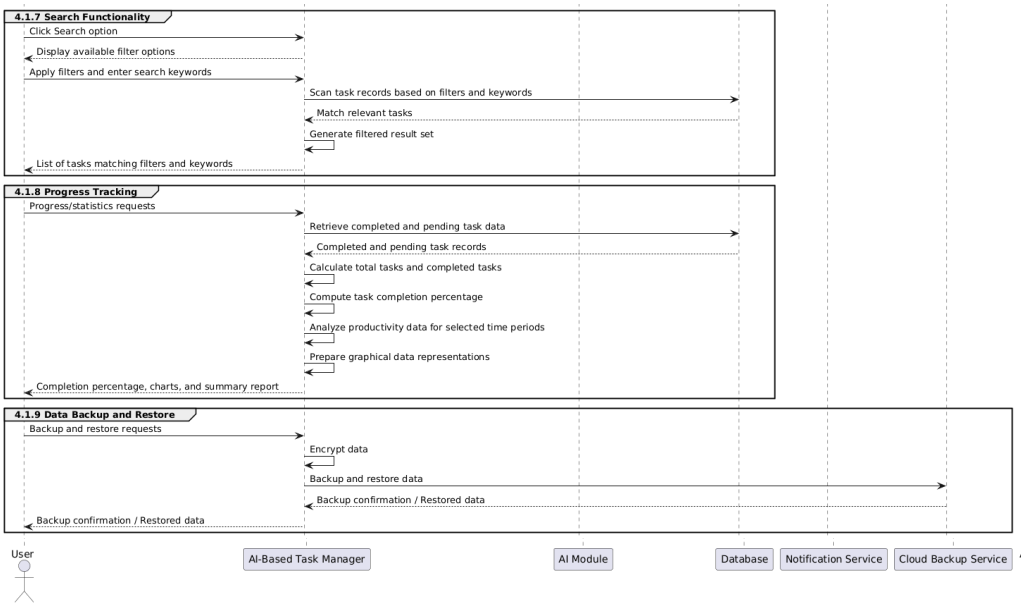


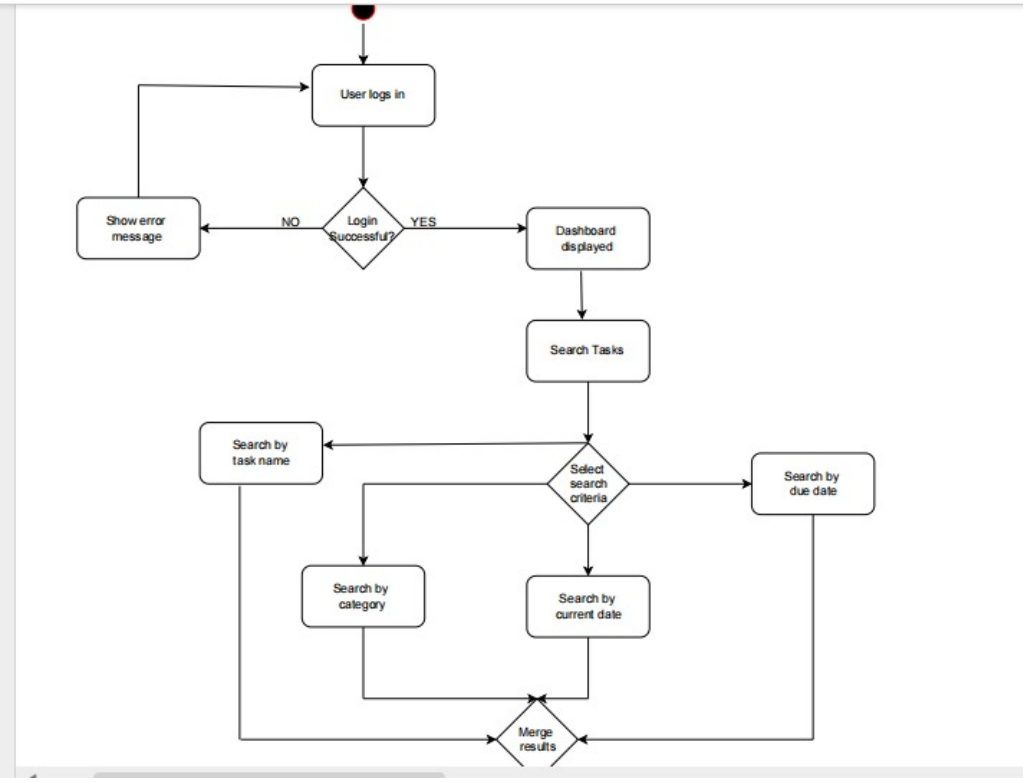
Figure 5: Sequence Diagram

Diagram Instruction: Include lifelines for:

- User
- Mobile App
- AI Engine
- Database
- Notification Service

7 Workflow Design Description

Activity diagrams illustrate workflows such as task creation, AI rescheduling, notifications, and backup.



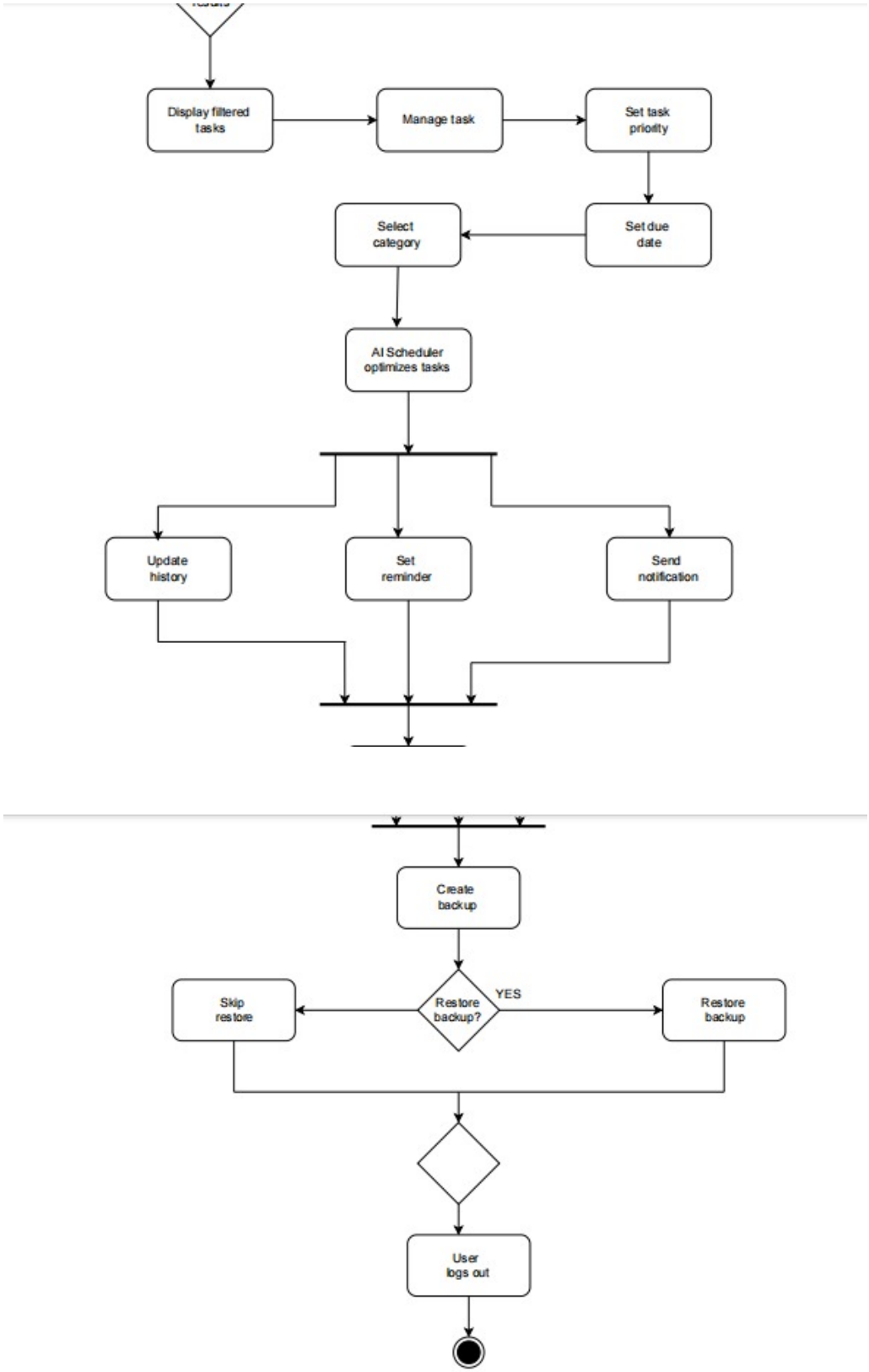


Figure 6: Activity Diagram

8 Structural and Component Design Description

8.1 Class Diagram

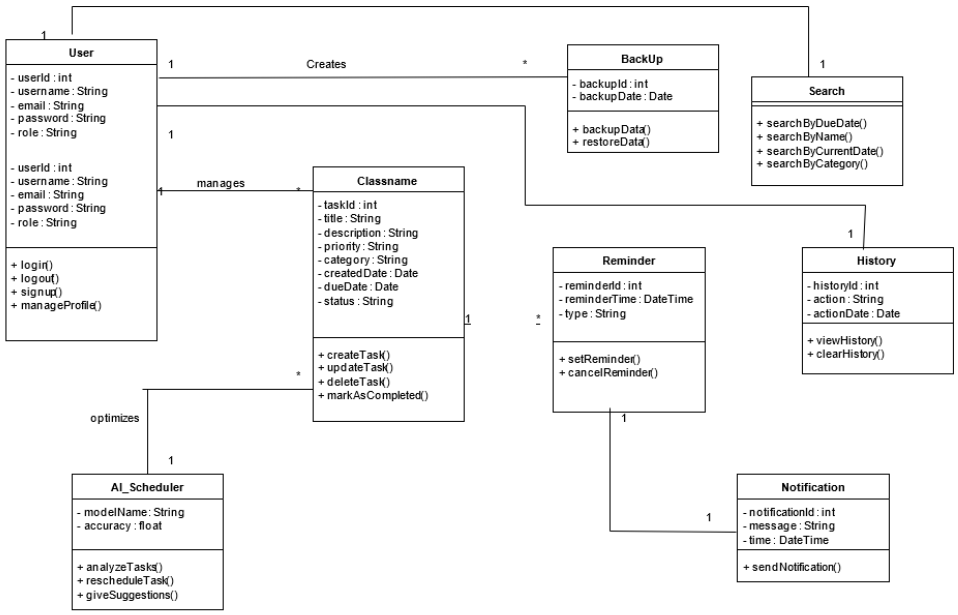


Figure 7: Class Diagram

8.2 Component Diagram

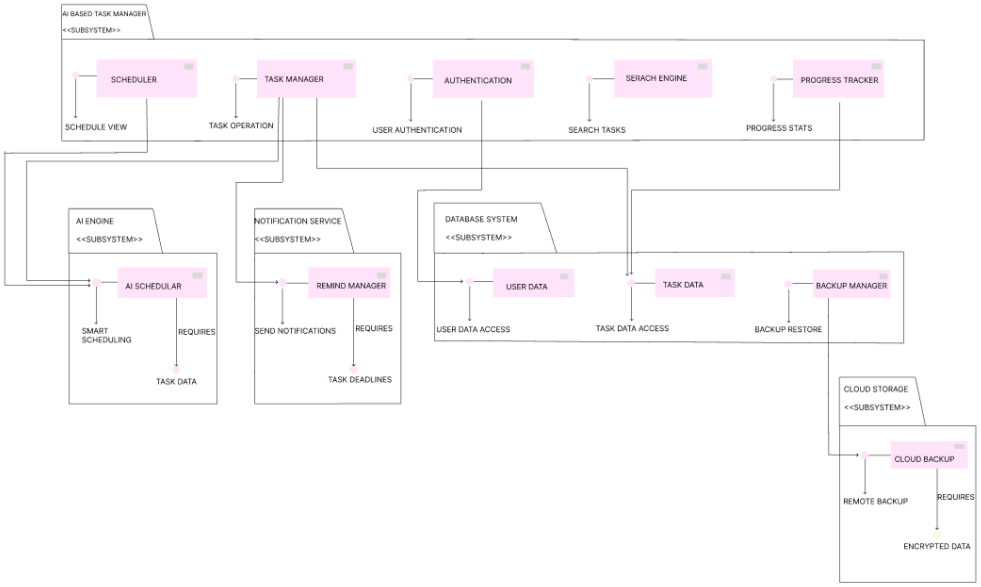


Figure 8: Component Diagram

Link:

`Component Diagram Link`

9 GitHub Link

The system design artifacts and prototypes are maintained using modern collaboration tools:

- **GitHub** is used for version control and managing project documentation.

Link:
AI-Based Task Manager – GitHub repository Link

10 Conclusion

This system design provides a structured blueprint for implementing the AI-Based Task Manager & Smart Scheduler. By aligning design artifacts with requirements and validating them through prototypes, the system is well-prepared for the implementation phase.