**UML Diagrams for Personal Task Manager Application**

This document contains UML diagrams illustrating various aspects of the Personal Task Manager Application. These diagrams provide a visual representation of system functionalities, architecture, and interactions, aiding in better understanding and implementation.

**1. Use Case Diagram**

The Use Case Diagram represents the interactions between users and the system's functionalities.

**PlantUML Code:**

@startuml usecase

actor User

usecase "Login" as UC1

usecase "Register" as UC2

usecase "Add Task" as UC3

usecase "Edit Task" as UC4

usecase "Delete Task" as UC5

usecase "View Tasks" as UC6

usecase "Set Tags" as UC7

usecase "Set Priorities" as UC8

User --> UC1

User --> UC2

User --> UC3

User --> UC4

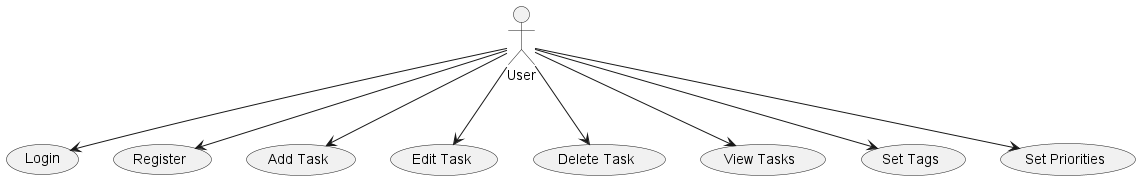
User --> UC5

User --> UC6

User --> UC7

User --> UC8

@enduml



**2. Class Diagram**

The Class Diagram depicts the structure of the application, including classes, attributes, and relationships.

**PlantUML Code:**

@startuml class

class User {

+id: int

+email: string

+password: string

}

class Task {

+id: int

+user\_id: int

+title: string

+description: string

+due\_date: Date

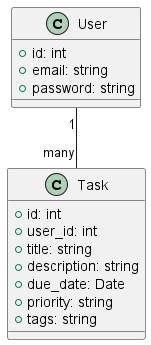
+priority: string

+tags: string

}

User "1" -- "many" Task

@enduml



**3. Sequence Diagram**

The Sequence Diagram illustrates the interaction between components when a user adds a task.

**PlantUML Code:**

@startuml sequence

actor User

participant "Web Interface" as Web

participant "Task Manager" as TaskMgr

participant "Database" as DB

User -> Web: Open Task Manager

Web -> User: Display Login Page

User -> Web: Login

Web -> TaskMgr: Authenticate User

TaskMgr -> DB: Validate Credentials

DB --> TaskMgr: Success/Failure

TaskMgr --> Web: Login Result

Web --> User: Login Successful

User -> Web: Add New Task

Web -> TaskMgr: Save Task

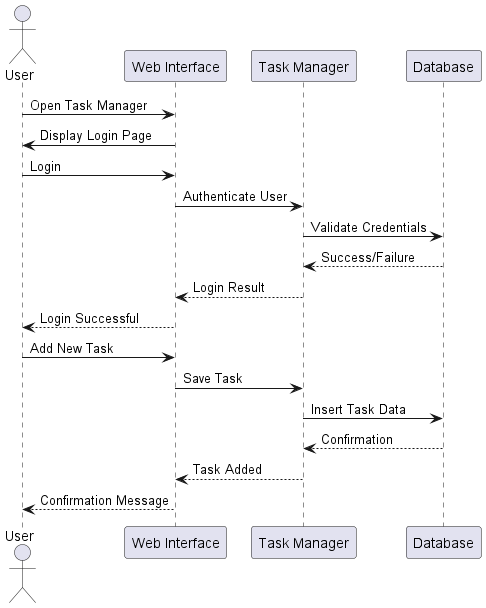
TaskMgr -> DB: Insert Task Data

DB --> TaskMgr: Confirmation

TaskMgr --> Web: Task Added

Web --> User: Confirmation Message

@enduml



**4. Activity Diagram**

The Activity Diagram outlines the flow of actions when editing a task.

**PlantUML Code:**

@startuml

start

:Login to System;

if (Is user authenticated?) then (Yes)

:Display Task List;

:Select a Task;

:Edit Task Details;

:Save Changes;

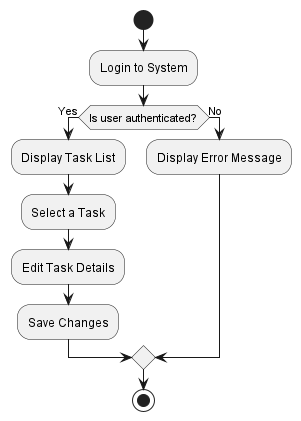
else (No)

:Display Error Message;

endif

stop

@enduml



**5. Entity-Relationship Diagram (ERD)**

The ERD illustrates the database schema and the relationships between entities.

**PlantUML Code:**

@startuml

entity User {

\* id: int

\* email: string

\* password: string

}

entity Task {

\* id: int

\* user\_id: int

\* title: string

\* description: string

\* due\_date: Date

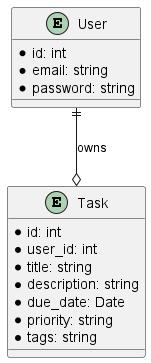
\* priority: string

\* tags: string

}

User ||--o Task: owns

@enduml



**6. Deployment Diagram**

The Deployment Diagram represents the deployment architecture of the application on Streamlit Cloud.

**PlantUML Code:**

@startuml

node "User Device" {

[Web Browser]

}

node "Streamlit Cloud" {

component "Personal Task Manager"

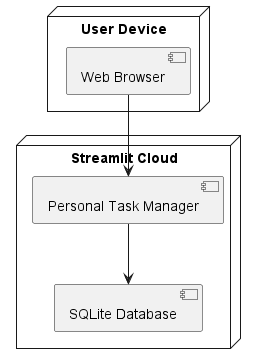
component "SQLite Database"

}

[Web Browser] --> "Personal Task Manager"

"Personal Task Manager" --> "SQLite Database"

@enduml



This document provides the essential UML diagrams for the Personal Task Manager Application. These diagrams facilitate a clear understanding of the system’s design and interactions.