Project Title: Productivity Application

Team ID: 050_005_803_803

NOTES

Name: Abhay Prashanth SRN: PES1UG20CS005

Name: Aneesh N A SRN: PES1UG20CS050

TASKS AND REMINDERS

Name: Anil Kumar M C SRN: PES1UG21CS803

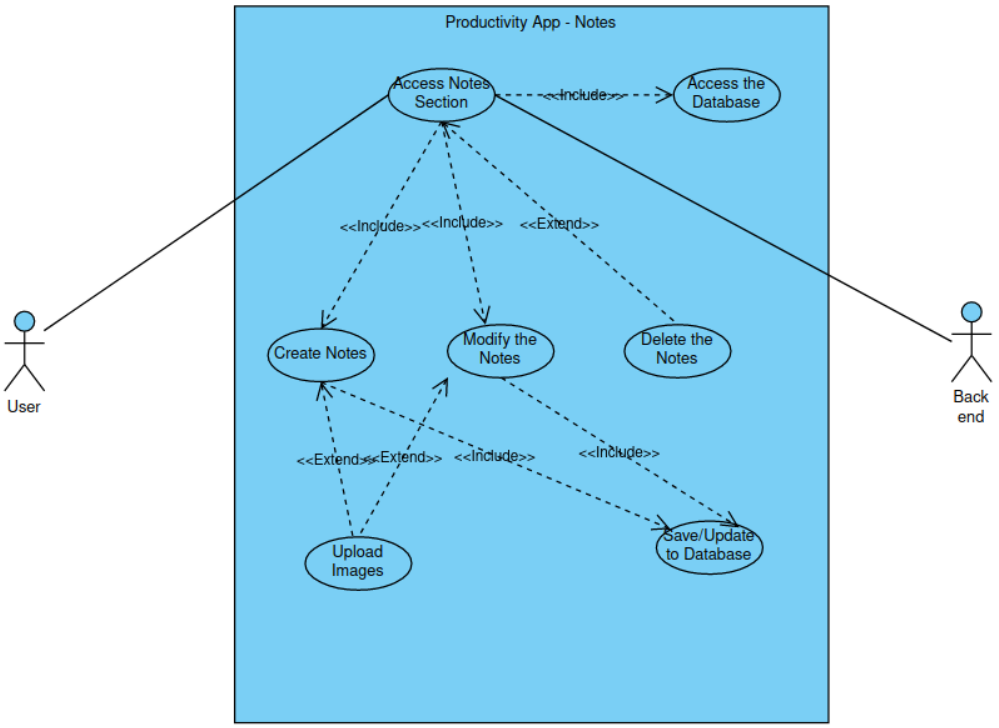
Name: Saurav Dey SRN: PES1UG20CS803

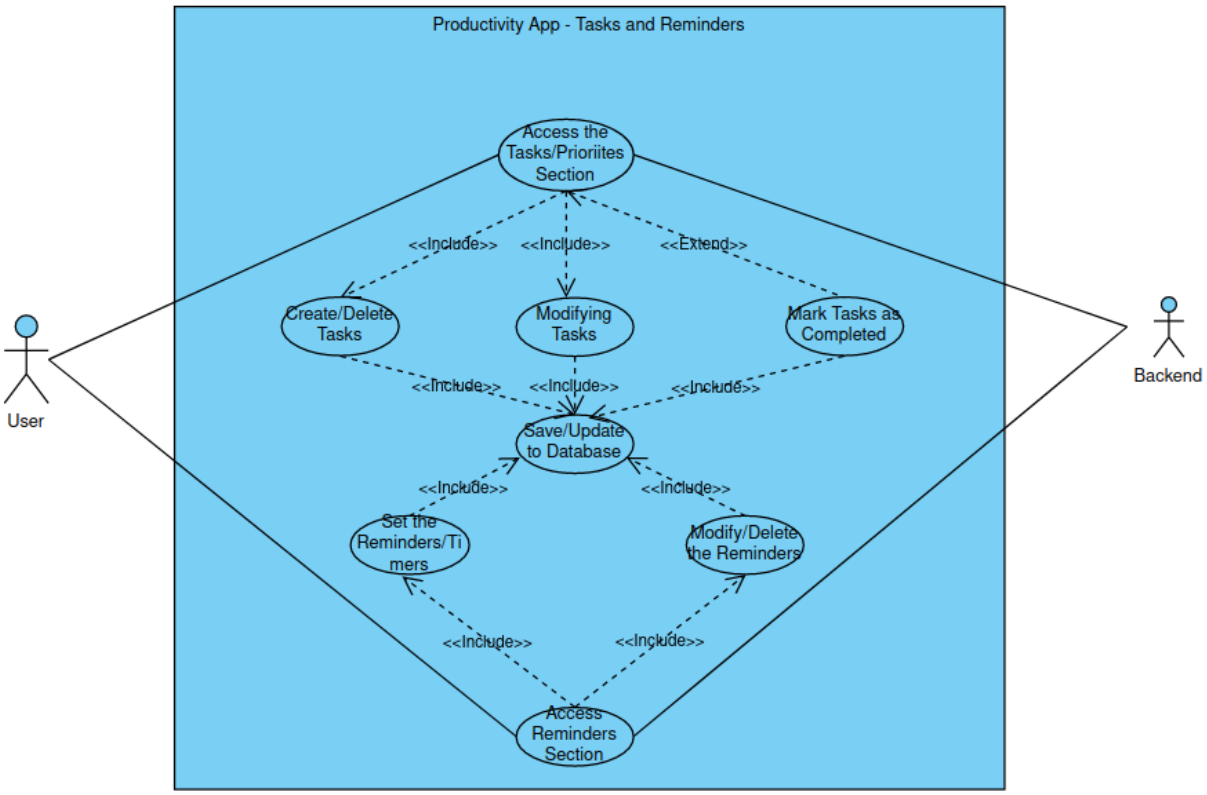
SYNOPSIS

This app offers reminders, tracks goals, organizes personal tasks, stores short notes, and helps manage to-do tasks and daily schedules.

The reminders feature is helpful for individuals who need timely reminders to complete tasks, make appointments, or meet deadlines. Users can set up alerts for specific times or dates, ensuring that they never miss an important event. The app's goal tracking feature allows individuals to set and track their personal goals. They can create multiple goals, assign specific timelines, and monitor their progress towards achieving them. The personal tasks organizer allows users to store and organize personal tasks such as shopping lists, birthday reminders, and other daily errands.The short notes feature enables users to jot down quick notes that can be accessed later, eliminating the need for traditional paper notes. The app's to-do tasks feature helps users stay organized and prioritize their tasks, ensuring they complete them on time. Users can also set up recurring tasks and track their progress towards achieving them.Finally, the scheduler/planner feature helps individuals plan their day, manage their schedules, and allocate time for important tasks.

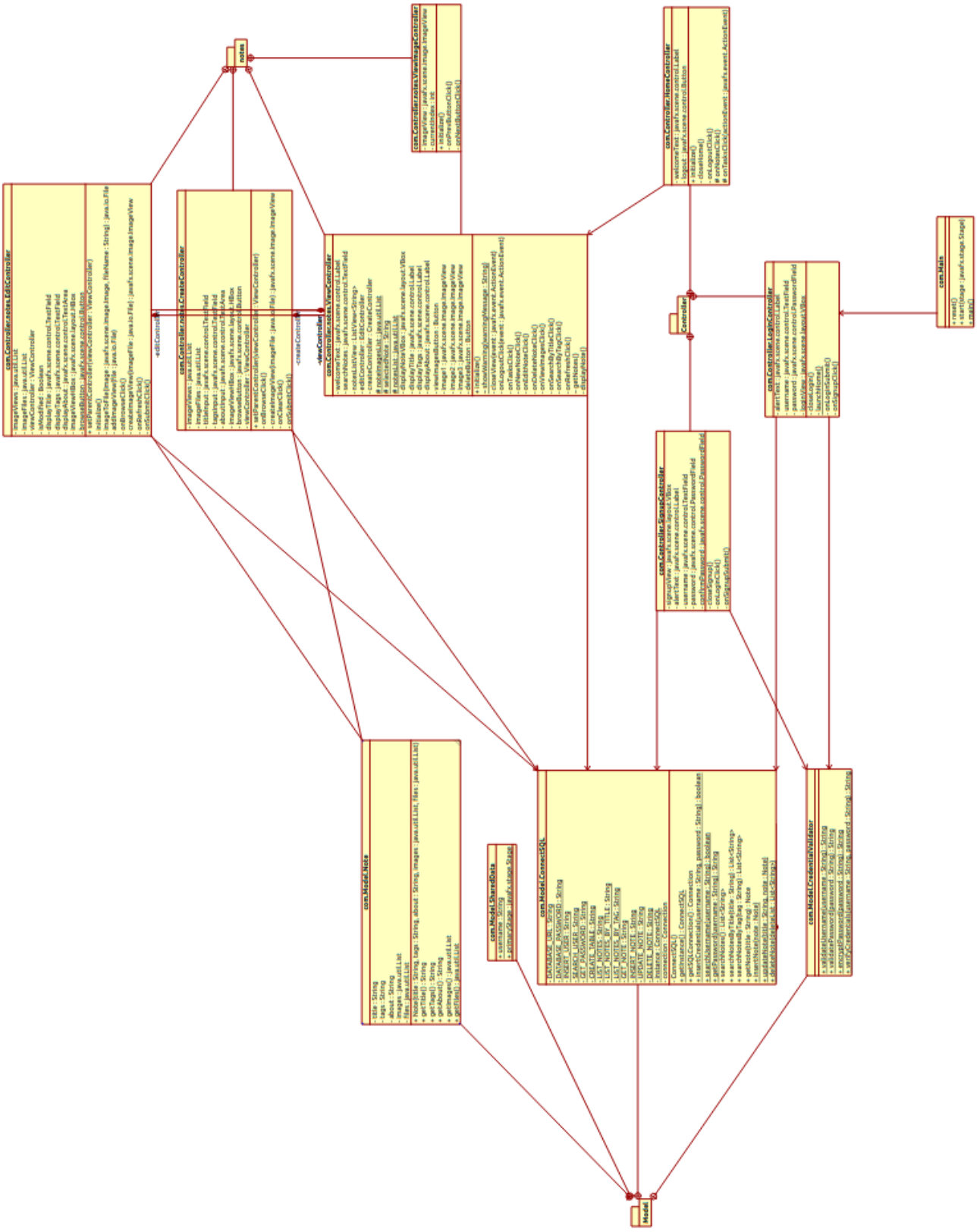
UCD


Productivity App - Notes
Access Notes Section
<<Include>>
Access the Database
<<Include>>
<<Include>>
<<Extend>>
User
Create Notes
Modify the Notes
Delete the Notes
Back end
<<Extend>>
<<Extend>>
<<Include>>
<<Include>>
Upload Images
Save/Update to Database

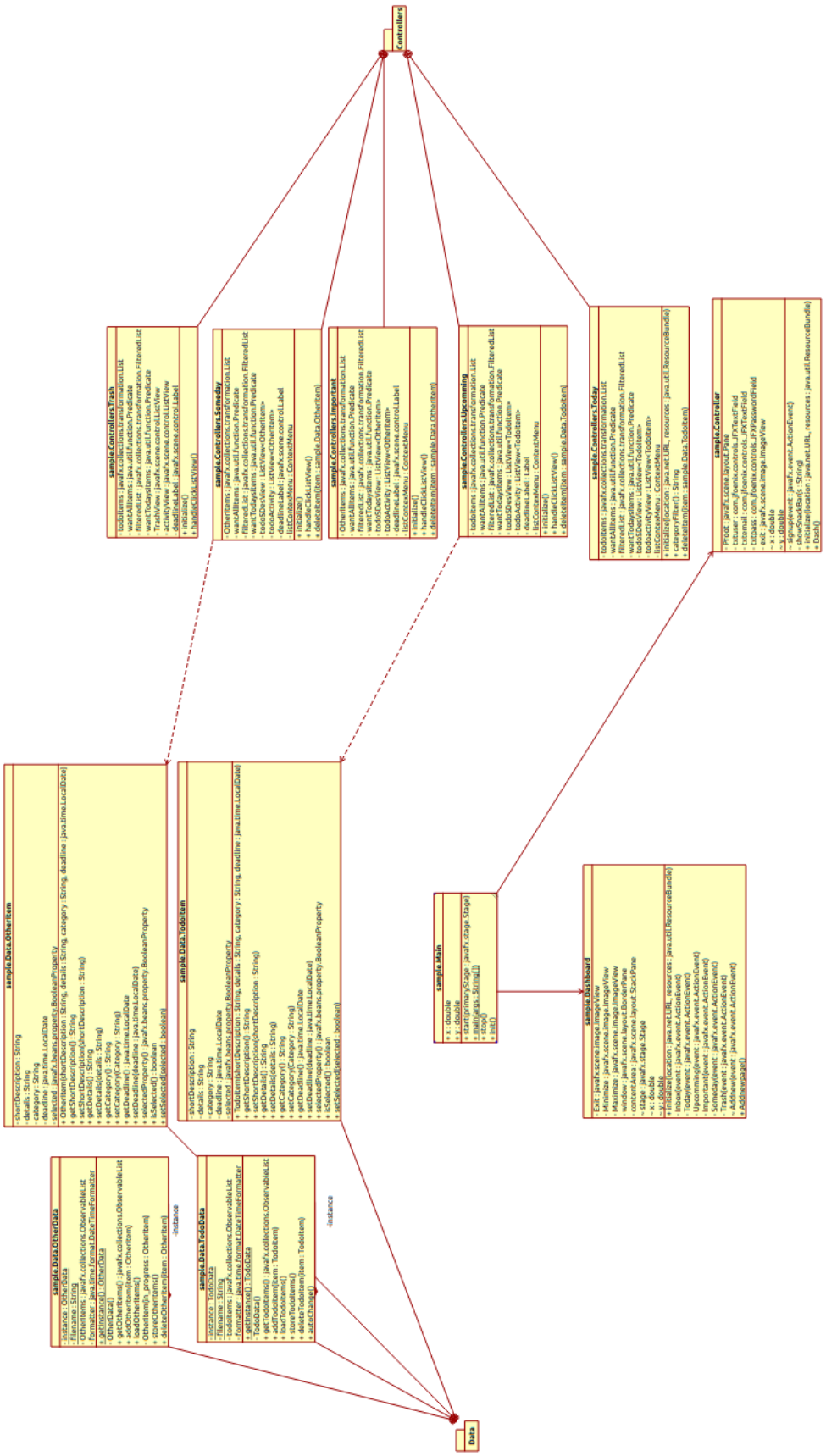

Productivity App - Tasks and Reminders
Access the Tasks/Prioriites Section
<<Include>>
<<Include>>
<<Extend>>
User
Create/Delete Tasks
Modifying Tasks
Mark Tasks as Completed
Backend
<<Include>>
<<Include>>
<<Include>>
Save/Update to Database
<<Include>>
<<Include>>
Set the Reminders/Timers
Modify/Delete the Reminders
<<Include>>
<<Include>>
Access Reminders Section

CLASS DIAGRAM

NOTES

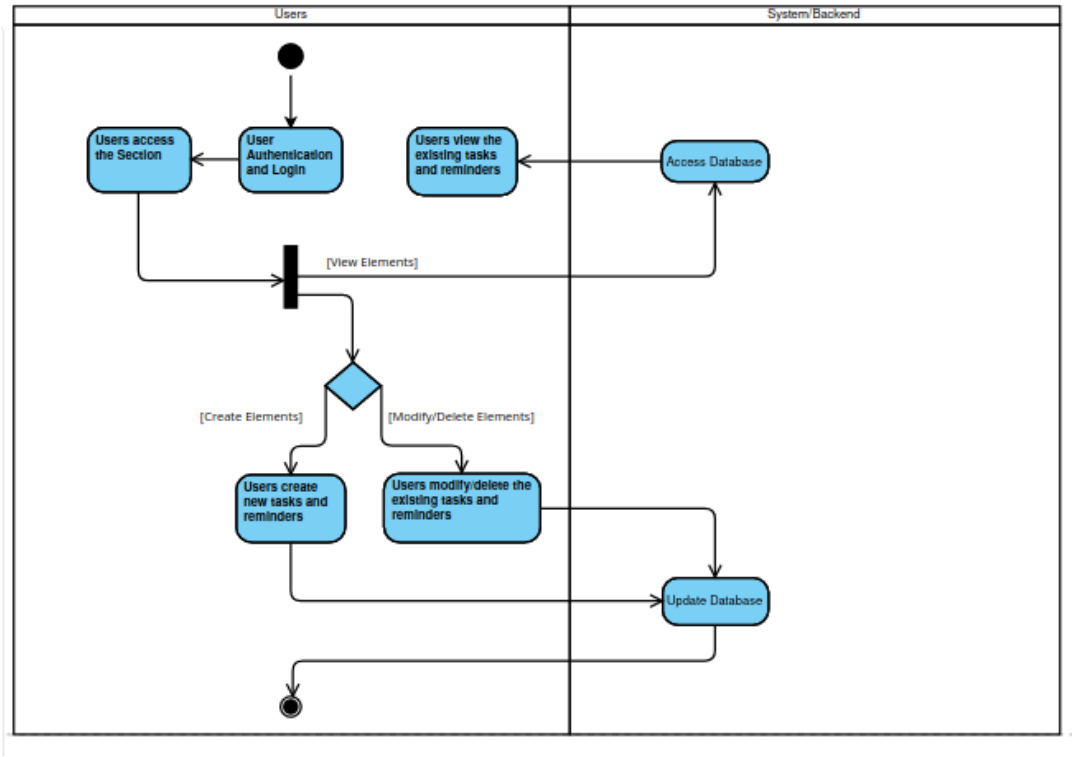


TASKS AND REMINDERS

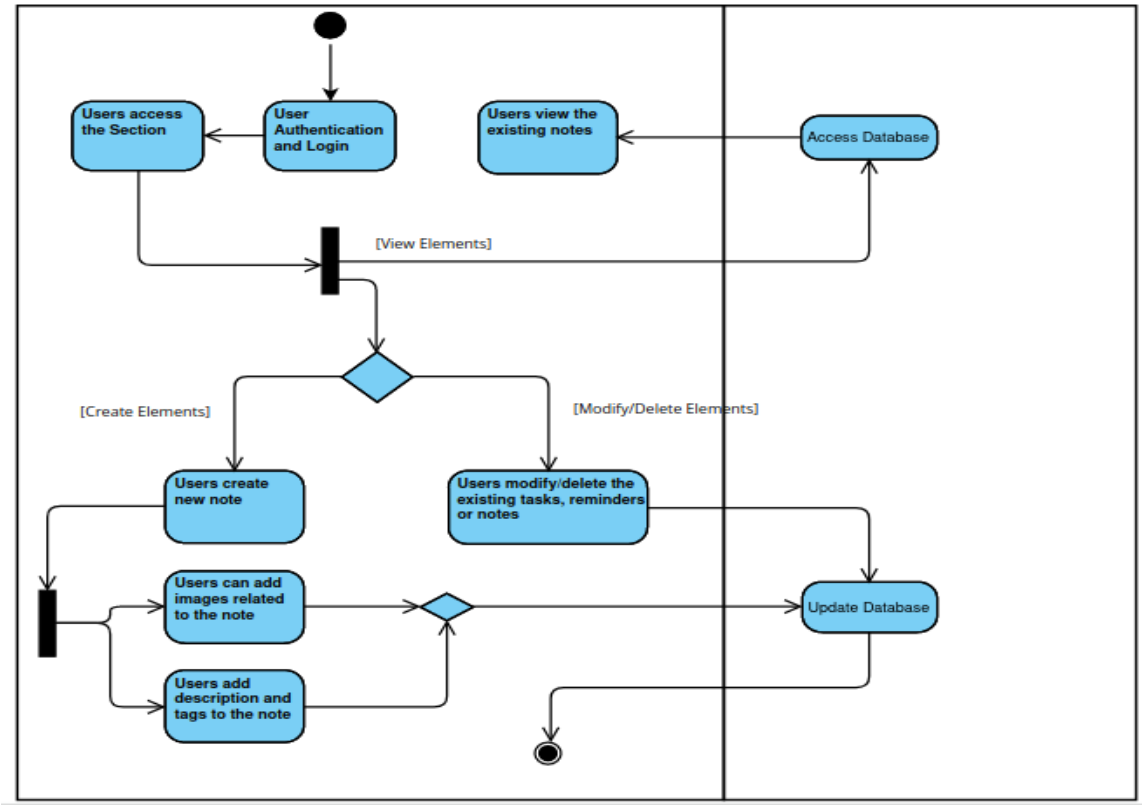


ACTIVITY DIAGRAM

Tasks and Reminders

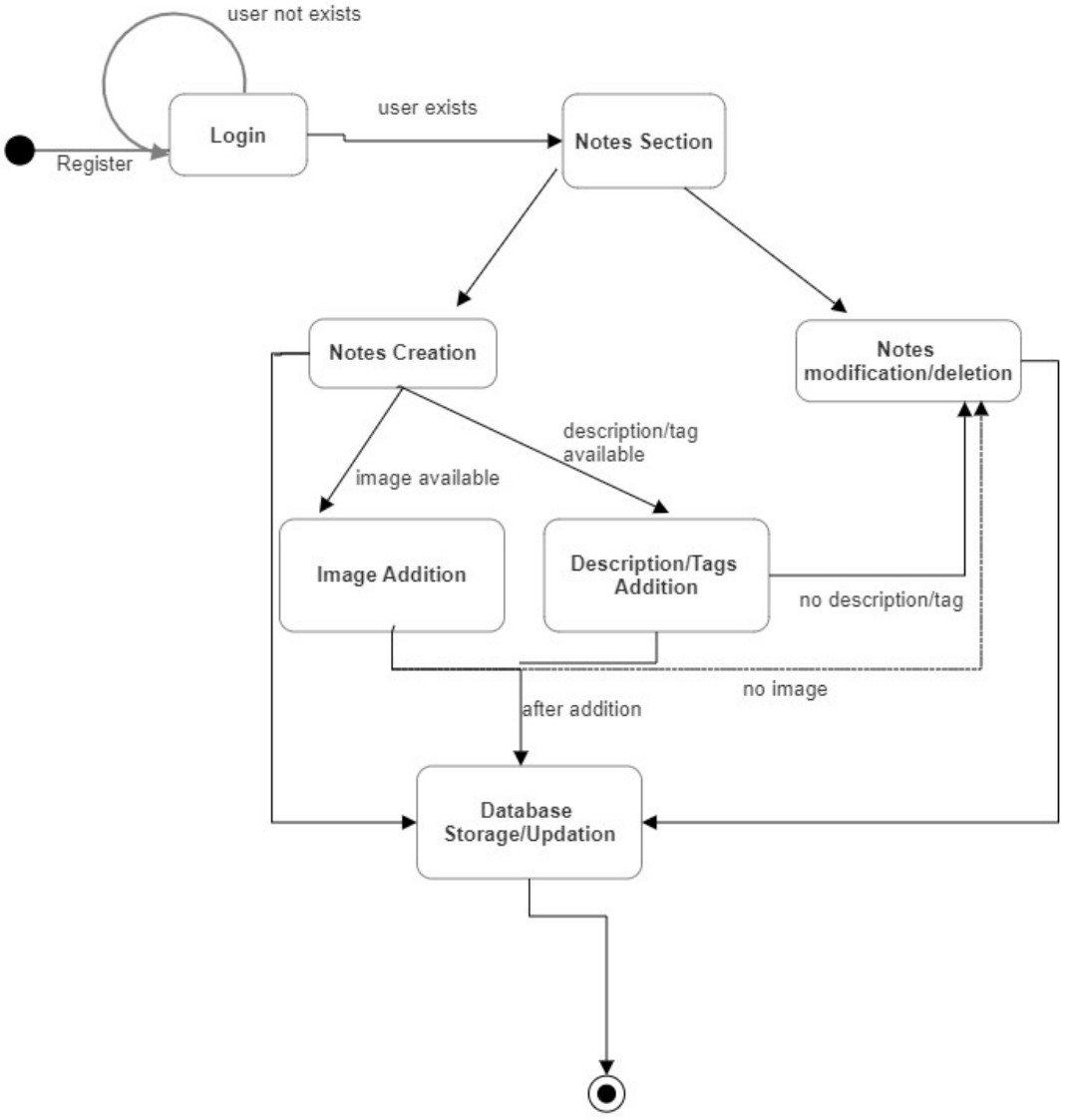


Notes



STATE DIAGRAMS

NOTES



DESIGN PRINCIPLES AND PATTERNS

Single Responsibility Principle (SRP): Each class or component in the application should have only one responsibility or concern. For example, a class responsible for managing notes should not also be responsible for managing tasks or reminders. This can help keep the code modular and easier to maintain.

Open-Closed Principle (OCP): The application's code should be open for extension but closed for modification. This means that new features or functionality can be added to the application without needing to modify existing code. This can be achieved by using interfaces, abstract classes, and dependency injection**.**

Dependency Inversion Principle (DIP): High-level modules should not depend on low-level modules, but both should depend on abstractions. This means that the code should be structured so that classes or components are loosely coupled and can be easily replaced or extended. This can help improve the flexibility and maintainability of the code.

Singleton Pattern: This pattern can be used to ensure that there is only one instance of a class in the application. For example, a database manager class could be implemented as a singleton, to ensure that there is only one connection to the database.

Facade Pattern: This pattern can be used to simplify complex systems or APIs. For example, a facade could be used to simplify the interaction between the user interface and the application's data storage, providing a simpler and more intuitive interface for the user.