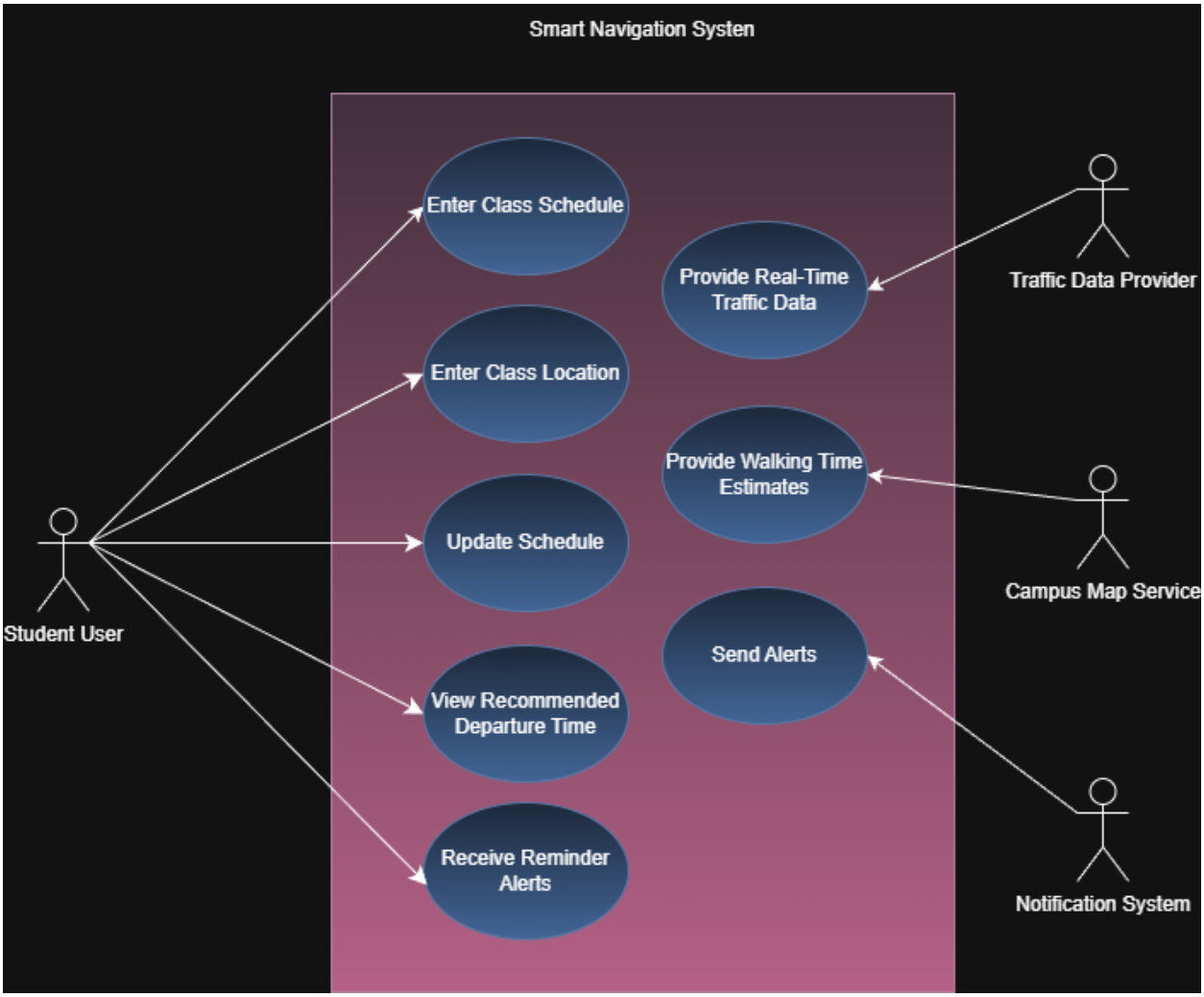
Software Requirements Specification

Use Case Diagram:



Functional Requirements:

FR1

Title: Schedule Editor

Description: The user should be able to input their class schedule directly into the program. They should also be allowed to manually edit their schedule.

FR2

Title: Starting location setup

Description: The program should allow the user to input their current or home address. (starting location)

FR3

Title: Routing

Description: The program should be able to calculate the optimal route from the user's starting address to their college.

FR4

Title: Traffic analysis

Description: The program should be able to analyze traffic patterns using databases and be able to predict traffic conditions for specific days.

FR5

Title: Journey calculation

Description: The program should be able to use traffic patterns and the user's schedule to give them a recommended departure time.

FR6

Title: Notifications

Description: There must be a notification/alert function to notify the user for when to leave.

FR7

Title: Storage

Description: The program should store all data in a database (user schedule, locations, traffic).

Non-functional Requirements:

NFR1

Title: Performance
Description: The system shall calculate departure times within a reasonable response time (for example: under 2 seconds).

NFR2

Title: Reliability
Description: The system should provide accurate departure time recommendations based on available traffic data.

NFR3

Title: Usability
Description: The system should have a simple user interface suitable for student users.

NFR4

Title: Availability
Description: The system should always be available for use.

NFR5

Title: Scalability
Description: The system should be able to support multiple concurrent users without any noticeable performance degradation.

NFR6

Title: Security
Description: The system shall securely store user data and protect it from unauthorized access.

NFR7

Title: Maintainability
Description: The system shall be designed in a modular manner to allow easy updates and maintenance.

NFR8

Title: Compatibility
Description: The system shall be compatible with commonly used operating systems and devices.

NFR9

Title: Data Accuracy
Description: The system shall ensure that stored traffic and schedule data remains consistent and up to date.

NFR10

Title: Fault Tolerance
Description: The system shall handle failures appropriately - without crashing.

Front end, Back end and Database?

Front End: The front end of the Smart Commute System is implemented using HTML and CSS because these technologies provide a simple, lightweight, and universally supported way to build the user interface. They are ideal for creating the forms and display pages needed for entering class schedules and viewing recommended departure times.

Back End: The backend is implemented using either Java and/or Python, both of which are well-suited for handling the system's business logic, integrating external APIs, and interacting with the SQL database.

Database: The project uses DBeaver and MySQL to design and populate a relational database that stores schedule data, traffic metrics, and calculated travel times. The database is connected to Visual Studio Code, where application code reads time-based records, selects the shortest travel duration through mathematical comparison, and updates traffic charts automatically at regular intervals