CS4632 Model Simulation Project Proposal

Department of Computer Science
College of Computing and Software Engineering
Kennesaw State University

Damien Castro, Dom Evans, Eric Legostaev

Spring 2025

1 Abstract

This project focuses on modeling the pathways, obstacles, construction zones, and vehicle/pedestrian traffic patterns at Kennesaw State University – Marietta Campus. By integrating real-time data from Google Maps API, user inputs, and class schedules, the system will accurately predict traffic volume and patterns. The deliverables, including UML diagrams, initial model designs, and testing frameworks, provide a clear roadmap for the project's development.

The system architecture comprises a front-end developed with HTML, CSS, and JavaScript, and a back-end implemented using Flask, NetworkX, OSRM, and an SQL database. Rigorous testing, including user feedback, direct observations, and validation against real-world data, ensures accuracy. Integrating this module into MobiNav will improve the quality of life for differently-abled individuals by offering safe, efficient, and accessible routes tailored to their needs.

2 Introduction

Differently-abled individuals face significant challenges in navigating everyday environments. Routine tasks, such as finding accessible routes, become complex and time-consuming. This project addresses these challenges by enhancing mobility and independence through an interactive map application, MobiNav.

MobiNav's traffic model and simulation module will focus on Kennesaw State University – Marietta Campus. By modeling pathways, obstacles, construction zones, and traffic patterns, the module integrates with MobiNav to provide accessibility-optimized routing. This approach leverages real-time data, user inputs, and predictive modeling to create personalized, efficient routing solutions.

3 Methodology

3.1 Data Collection

The system will model the campus by:

- Using Google Maps API for real-time obstacle updates.
- Extracting data from satellite and street-view images to identify accessible features.
- Allowing users to report new obstacles.
- Utilizing class schedules to predict building occupancy and simulate traffic patterns.

3.2 Tools and Technologies

Front-End:

- HTML/CSS and JavaScript for a responsive interface.

Back-End:

- Flask framework for server-side logic.
- NetworkX for graph-based traffic modeling.
- OSRM for routing calculations.
- SQL Database for persistent data storage.

3.3 Validation and Testing

- Compare predictions against user feedback and real-world observations.
- Incorporate automated unit and integration testing for robustness.

4 Implementation Details

4.1 UML Diagrams

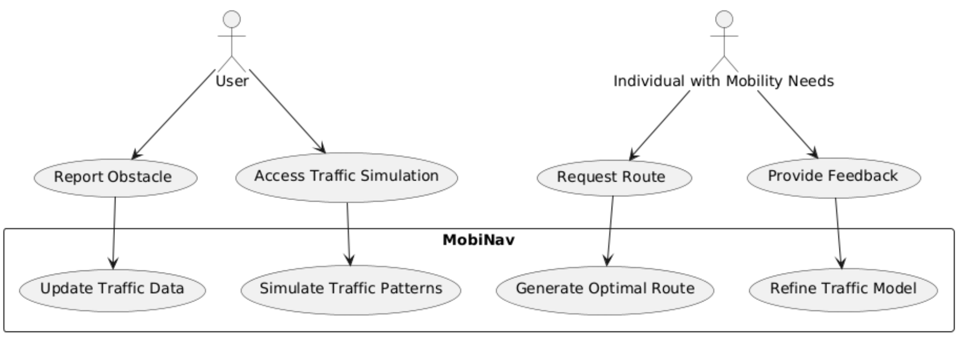

Figure 1: Use Case Diagram showing system interactions.

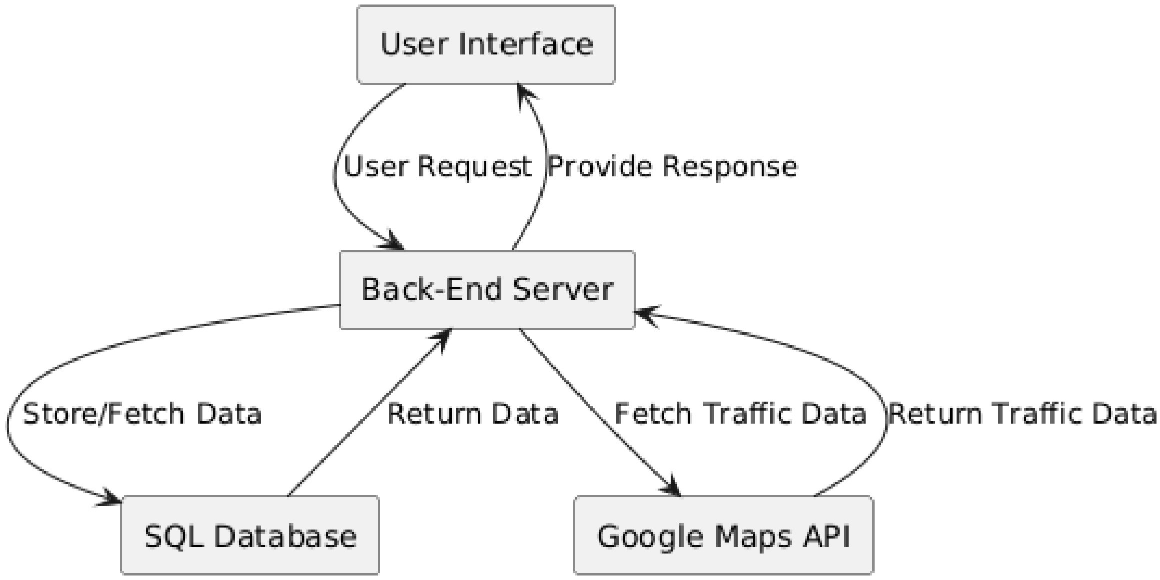

Figure 2: Data Flow Diagram visualizing system processes.

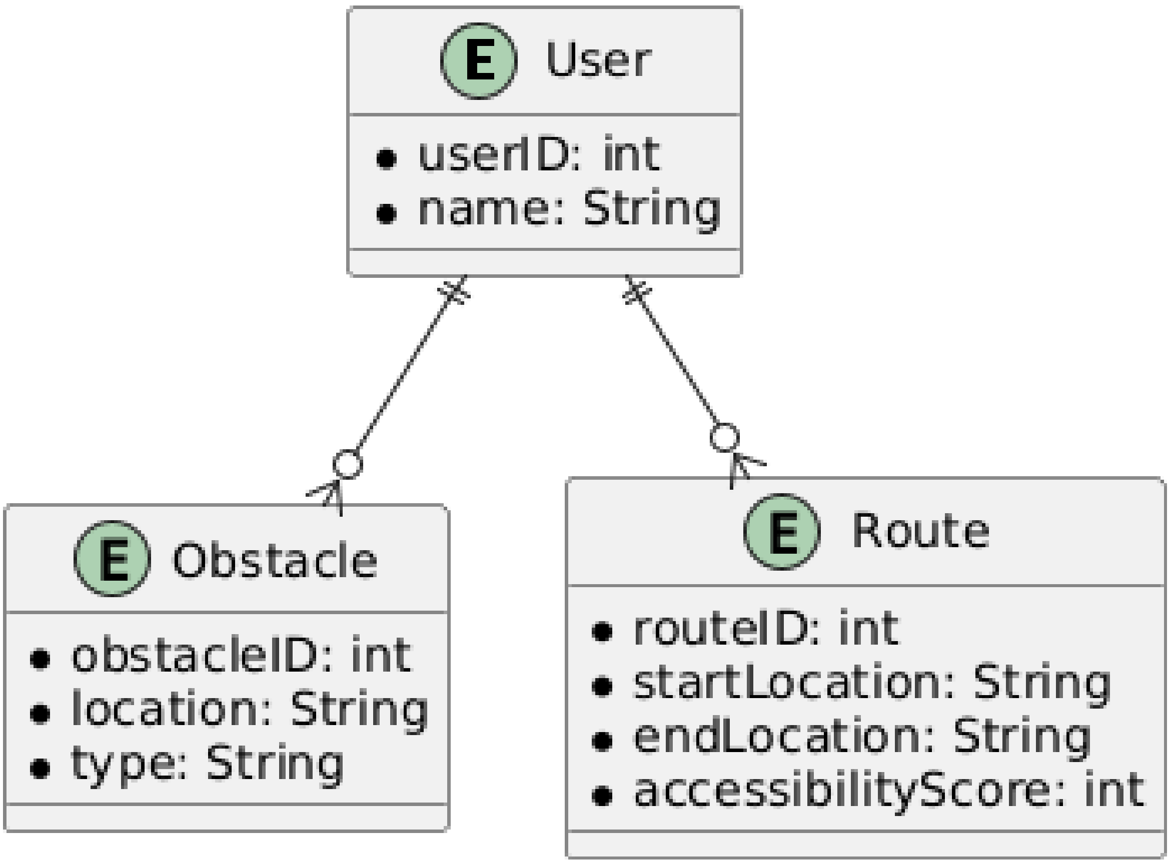

Figure 3: Entity-Relationship Diagram defining database structure.

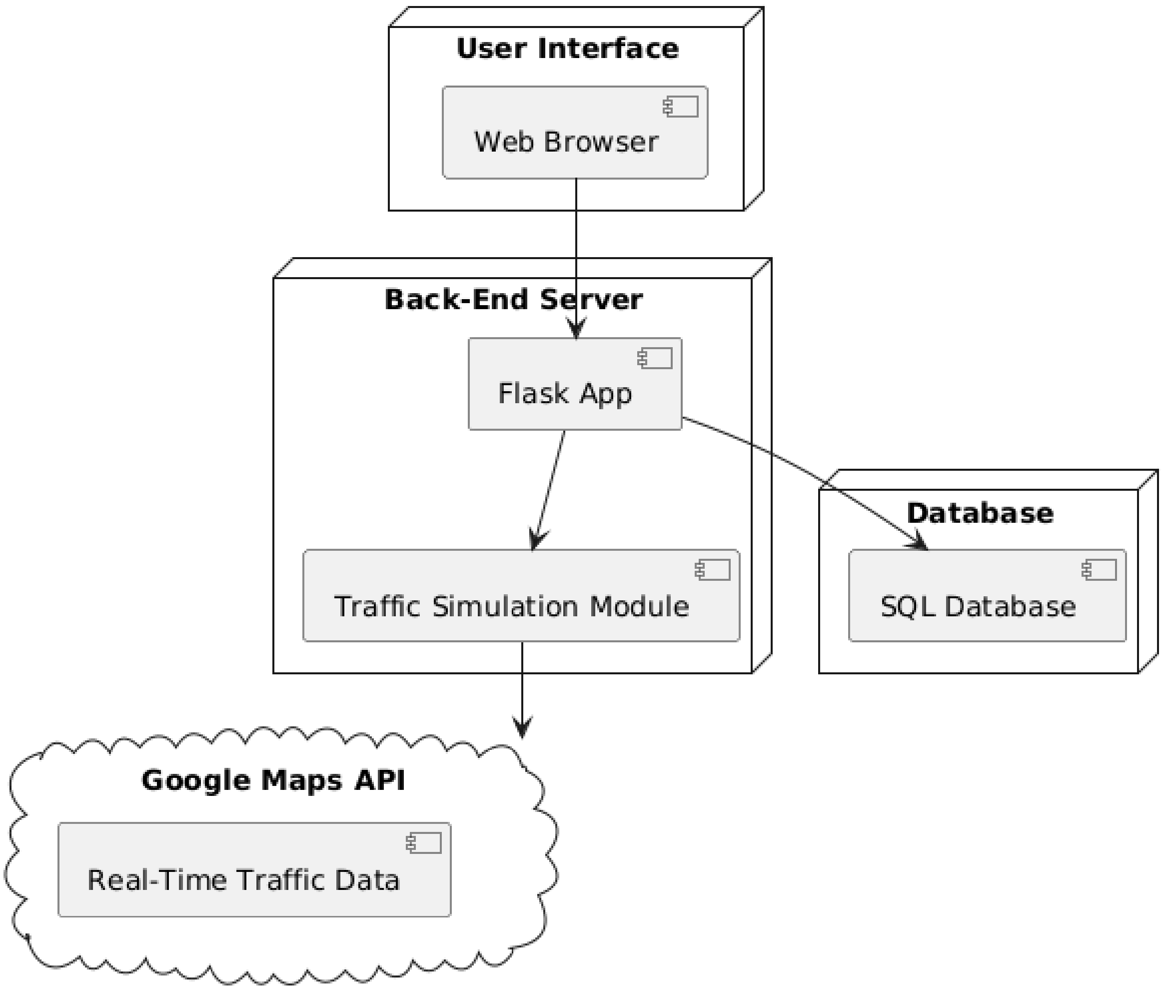

Figure 4: System Architecture Diagram illustrating integration.

4.2 Initial Model Design

4.3 Development Plan

- Front-End: Develop an accessible, intuitive interface for users to interact with the system.
- Back-End: Implement logic to process traffic data, run simulations, and manage user inputs.
- Database: Design and deploy an SQL database based on the Entity-Relationship Diagram.

4.4 Integration Plan

- API Integration: Incorporate Google Maps API, NetworkX, and OSRM for real-time data and routing.
- User Input: Enable reporting of obstacles through the interface and update the database dynamically.

5 Experimentation and Simulation Runs

Simulations will evaluate traffic patterns under diverse conditions:

- Core Scenarios: Test typical student routes during peak and off-peak hours.
- Edge Cases: Assess system performance during blockages or extreme conditions.

Key metrics include:

- Route accuracy.
- Computational efficiency.
- Scalability under high user loads.

6 Validation and Verification

Technical Validation:

- Verify the accuracy of routing and simulation algorithms.
- Ensure consistent performance of API integrations.

User Validation:

- Conduct focus group testing, including individuals with mobility needs.
- Collect feedback via surveys and analyze user interactions.

Performance benchmarks will confirm system reliability, responsiveness, and compliance with accessibility standards.

7 Conclusion

This proposal outlines a comprehensive plan for developing and integrating a traffic model and simulation module into MobiNav. By leveraging advanced tools and methodologies, the project will enhance accessibility and independence for differently-abled individuals. Through rigorous validation and testing, the system will deliver accurate, efficient, and inclusive routing solutions, making a meaningful impact on campus navigation and beyond.