

Project Problem #6: You've been asked to help the Rutgers Department of Transportation (DoT) to design a student bus route management system. The DoT wants a system that meets certain performance criteria, e.g., frequency, cost, the maximum time to destination etc. You will need to make sensible necessary assumptions of the budget, the time between campuses, frequency etc., but no student should have to wait more than 30 minutes to get to a class on any campus from any point on the New Brunswick campus.

Specification and Requirements Engineering:

- **Specification and User Requirements:**

- This student bus route management system is designed to display the time of arrival and path between two bus stops at Rutgers University. The user of this app should not have to wait more than 30 minutes to travel between two bus stops for class, events, etc.

- **System Requirements:**

- System will hold constant access to the Rutgers Bus network to access buses' locations throughout all the campuses
- An estimated time of arrival and path will be provided to the user from 2 locations (to and from) where they choose.
- An option to set an alert notification when the desired bus is within 5 minutes to the bus stop
- A map of the buses on route to see their physical location in the event of heavy traffic and I misinterpreted ETA. (We are measuring ETA by distance and time without considering heavy traffic that could affect ETA)

- **Functional Requirements:**

- The user should be able to view all buses in route chosen via Google API map
- The user should be able to view the ETA and route of the buses in route chosen
- The user should be able to input their starting point location and the destination bus stop they choose

- **Non-functional Requirements:**

- **Organizational Requirements:**

- Users need to authenticate themselves by signing in with netid and password or RUID number.

- **Performance Requirements:**

- The system should be able to withstand immense throughput from the users and execute in a timely manner.
- The system should regulate bus scheduling to be at most 30 min apart per bus per route

- **Safety Requirements:**

- In the event of an accident or anything that causes route blockage or danger to any drivers or passengers, the system should notify them.

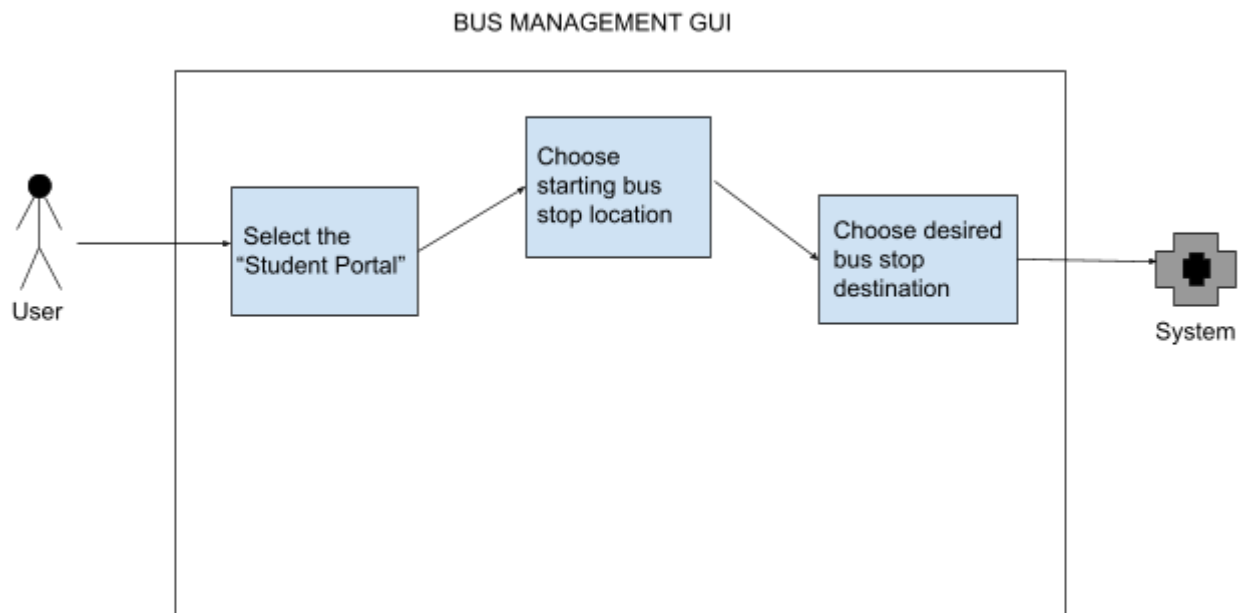
- **Security Requirements:**

- To prevent any hacking into the system, authorization via netid/password or RUID is required before using system

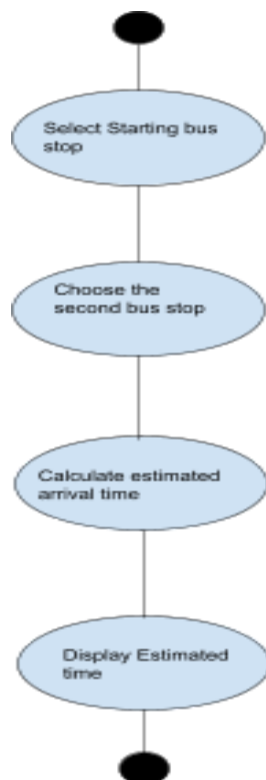
- The system should remain available for the operating hours for buses (7 am - 3 am)
- **Domain Requirements:**
 - Utilizes the Google API
- **Usability Requirements:**
 - Should be easy to use and easy to understand without a learning curve (User simplicity)

Systems Modeling

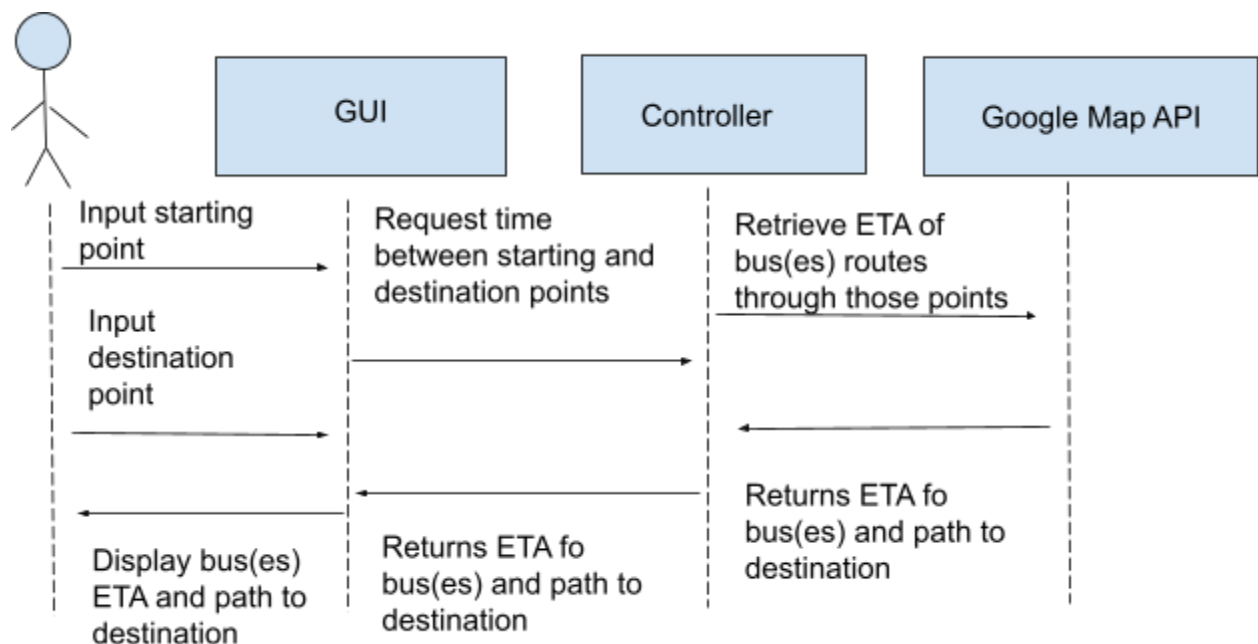
Use Case Diagram:



Activity Diagram:

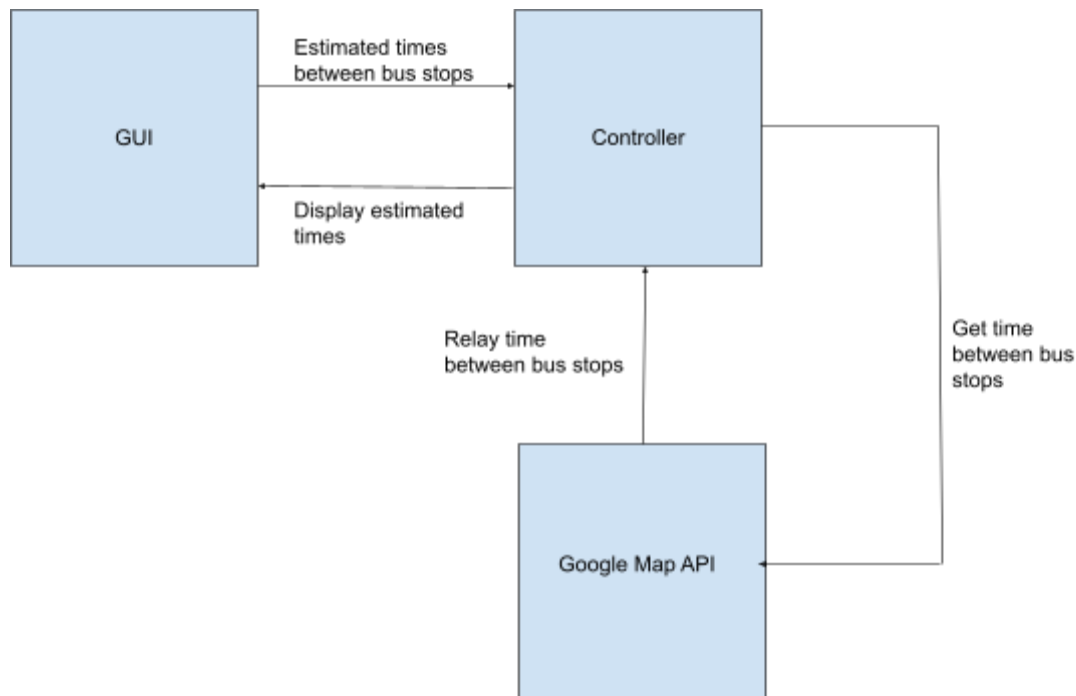


Sequence Diagram

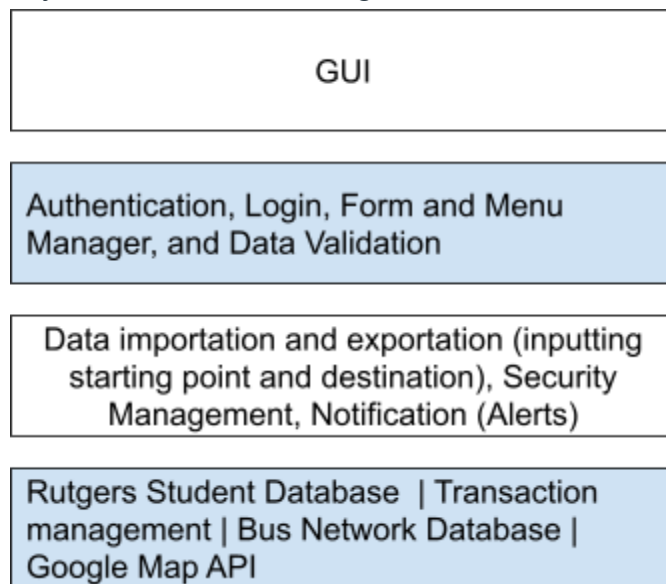


Architectural Design:

MVC Diagram



Layered Architecture Diagram:

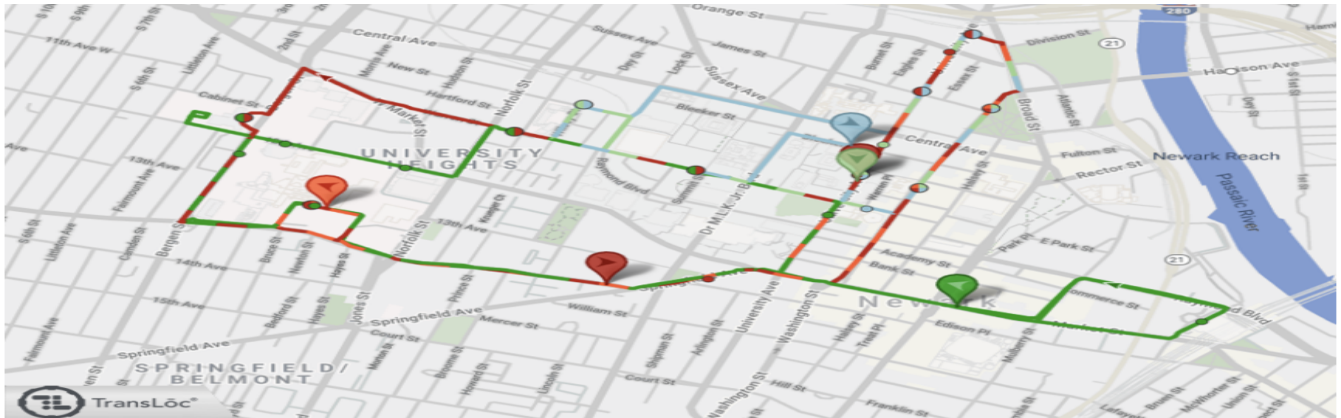


Design and Implementation:

Design Ideas

- User will sign-in to be authenticated via netid/password or RUID
- User will input their starting point
- User will input their destination
- User has the option to be alerted 5 min before the bus reaches their starting point

- System should display ETA of bus(es) arriving to user starting point
- System should display a map of the buses on route



Software Testing (across all levels):

No code was developed.

Evaluation:

- Investigated how to implement users requirements
- Through multiple diagrams we determined the ideal and step my step systems the code should follow
- Clear plan of what the desired code should accomplish
- Explored limitations of code and Google API distance matrix

Project Management:

Risk Management

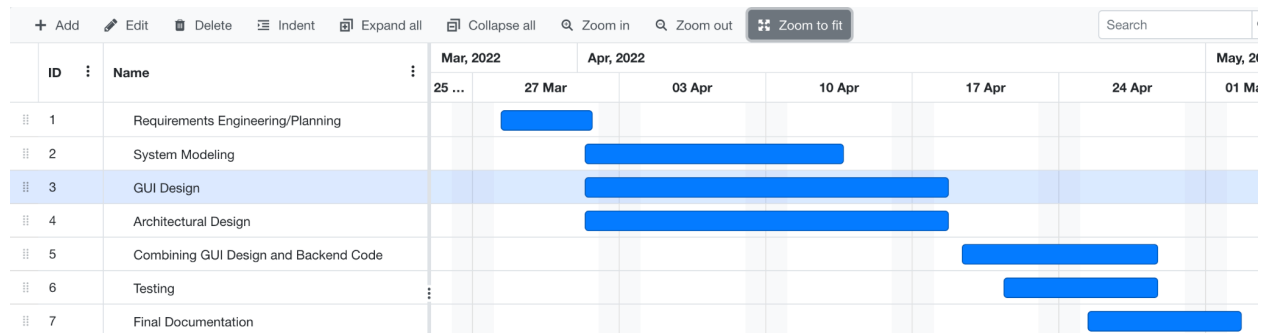
Description	Impact	Probability	Severity	Mitigation
Team members may not submit code on time or are unresponsive	Deadlines for sprints will have to be pushed. Other team members may need to take up extra work to finish the project on time	Moderate	Serious	Check up periodically before sprint deadline on how everyone is progressing with their assignment
The estimated	This would	Moderate	Tolerable	Checking up

time for this sprint is underestimated	affect the quality of the sprint or it would change the schedule for production of the software			periodically on the progress of each task will help the group adjust the deadlines faster if necessary
Group members do not have the requisite experience to work on their tasks	Team members would have to learn skills before starting to code their task	Moderate	Tolerable	Tasks should be started early so that team members can learn the skills required to complete the task

Effort to make system dependable (Availability)

By making sure that our system is constantly available during the operating hours of buses we make our system more dependable, providing service to users.

Gantt Chart



Plan

Moving forward to our next sprint we want to familiarize ourselves with the Google API distance matrix and begin implementing code before our next sprint. We plan on implementing the security requirement (student authentication), requesting starting and destination points, classes needed for the system, and GUI interface.