

# **Rutgers Bus System Manager**

**Thomas Bargoud, Hsinjui Chou, Zeeshan Qureshi, Guramrit Singh**

Our goal in this project was to create a bus system that gave students real-time information regarding bus timings and to make sure students would not spend more than 30 minutes on a bus. Below are the requirements we gathered for this project.

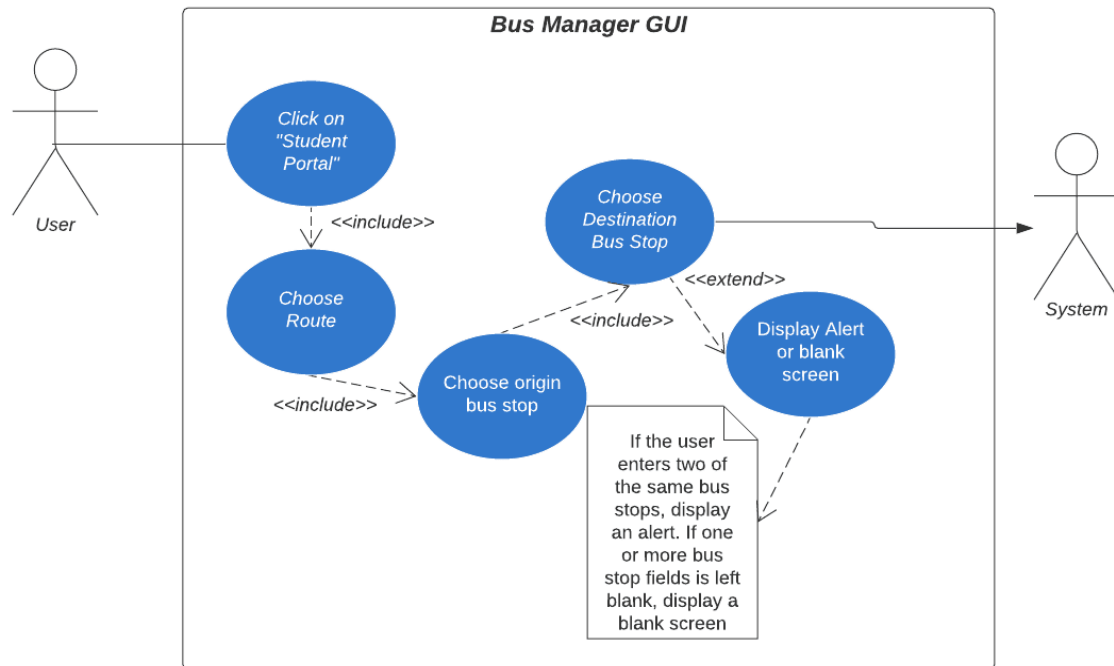
### Requirements Engineering

Functional Requirements	
1.	Users should be able to choose a route and then choose the origin and destination they want to travel to
2.	No passenger should have to wait 30 minutes to get from one stop to another - There can be a route that takes longer than 30 minutes if there is another route that makes the same trip in less than 30 minutes. For example, if the “A” bus gets to Busch Campus Center from College Avenue Student Center in 40 minutes, but the “H” bus gets to Busch Campus Center from College Avenue Student Center in 25 minutes, then the requirement is met.
3.	Once origin and destination are chosen, app should be able to display: -What time the next bus is getting to the origin bus stop -What time the bus will stop at each intermittent stop -What time the bus will get to the destination -How long the trip will take
Non-Functional Requirements	
1.	System should be secure as it is dealing with location information
2.	The number of buses simulated should be reasonably within the budget of the Rutgers DOT
3.	GUI should be user friendly

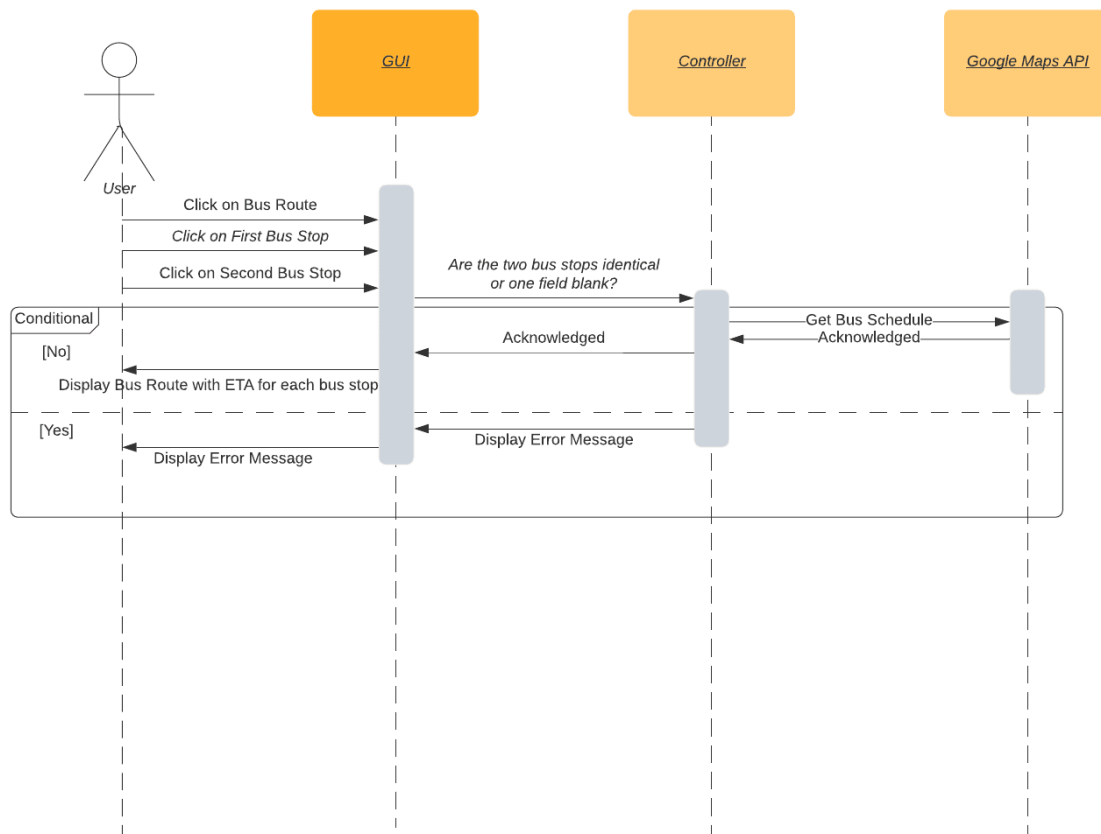
To achieve this goal, we designed our system as shown below:

## Use Case Diagram

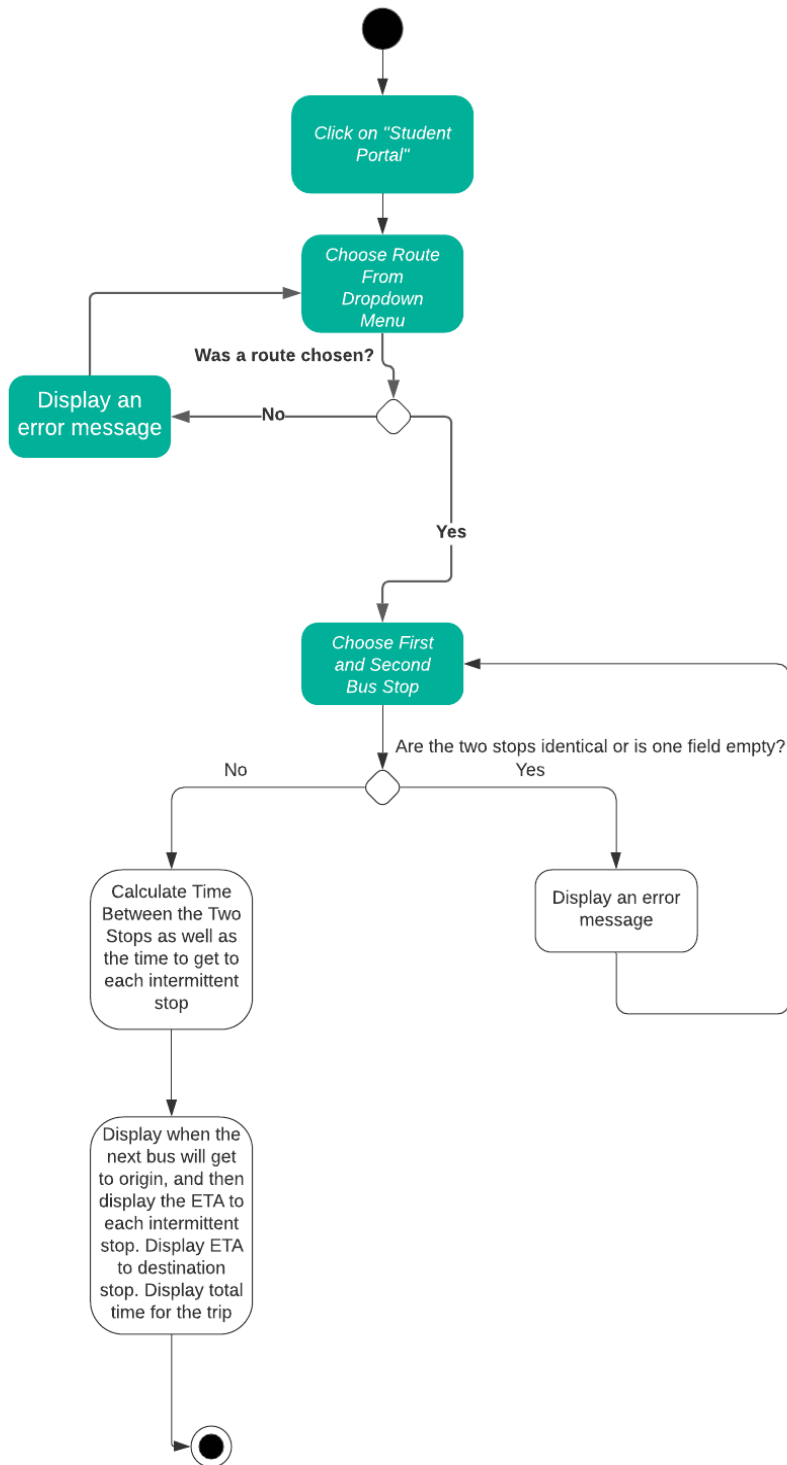
Zeesnan Qureshi | December 15, 2021



## Sequence Diagram

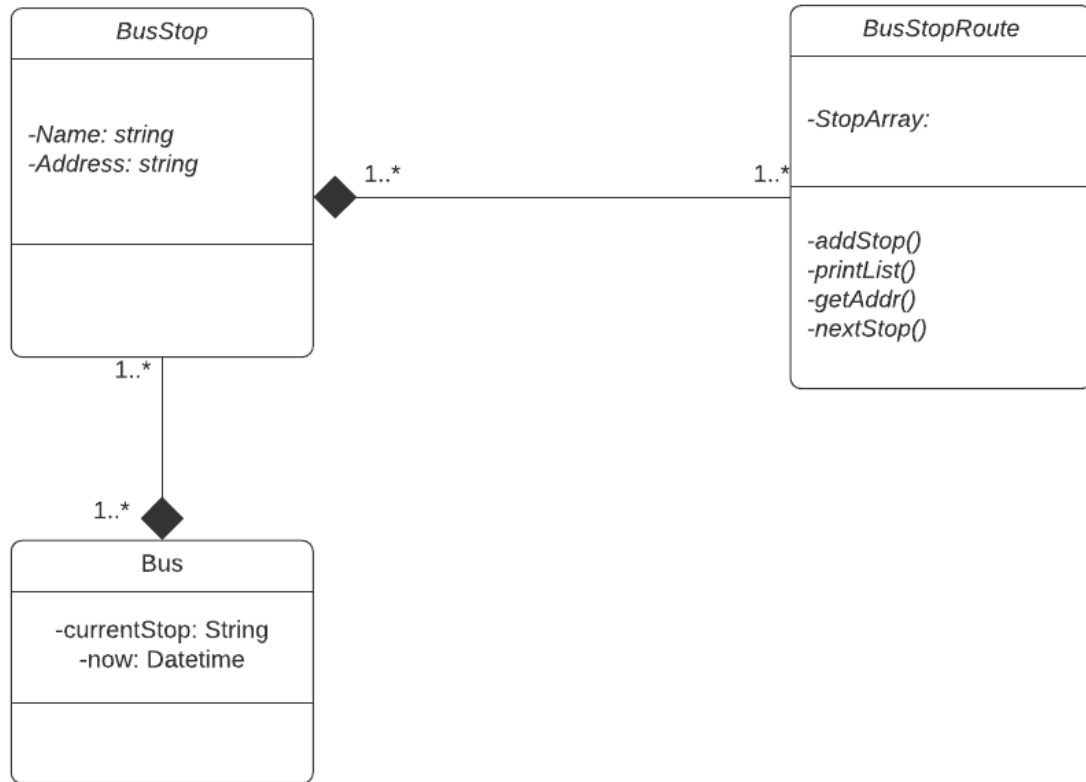


## Activity Diagram

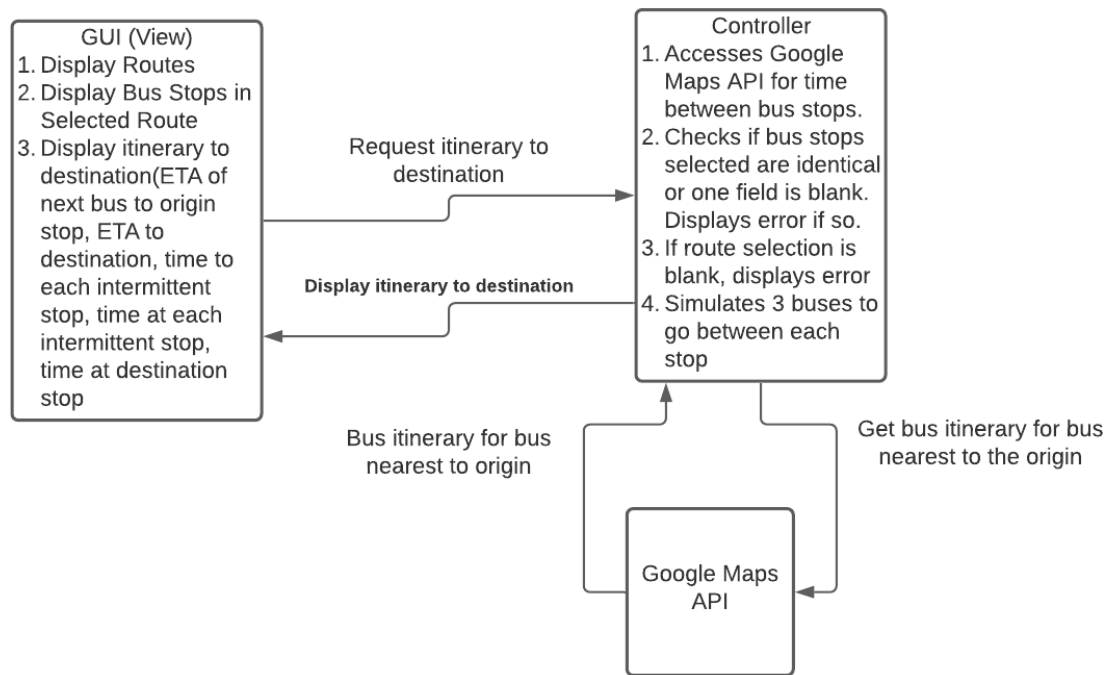


## Class Diagram

Zeeshan Qureshi | December 16, 2018



## Architectural Design – MVC



## Specification and Implementation

The GUI of the system was developed using the tkinter library from Python. In the backend, we had 3 classes: “bus”, “busStop”, and “busStopRoute”. The “bus” class was made to simulate buses in our application. The “busStop” class’ function was to add bus stops to our routes and function of the “busStopRoute” class was to add the different bus routes to the application. In the software, we simulated 3 buses per route and we found that no unique route took longer than 30 minutes. Please find more on this in our testing section. On the user’s side of things, the user can select the route they want to take, and their origin and destination stops. Then the system accesses the Google Maps API which gives what time the next bus is getting to the origin bus stop, what time the bus will stop at each intermittent stop, what time the bus will get to the destination, and how long the total trip will take.

## Testing and Evaluation

A few bugs in the GUI needed to be fixed throughout the sprints. Our implementation of the Google Maps API needed to be tested as well. Furthermore, we manually tested each route if any route was over 30 minutes. In the EE route, 3 routes were over 30 minutes, however on the F bus, they were under 30 minutes. These routes on the EE bus were “Gibbons-Henderson”, “Biel Road-Lipman Hall”, and “Henderson-Biel Road”. The reason this test passed was because a student can board the F bus instead of the EE bus and still get to their classes on time under 30 minutes from the same origin to the same destination. Three buses per route were simulated in our system, which is a reasonable amount and likely in the budget of the Rutgers DOT as there are at least an average of three buses per route currently in the Rutgers Bus System.