

CS335 GROUP PROJECT

TRANSPORT SYSTEM SOFTWARE

BY

Eimaan Ali-22352376

Hari Venkatesan-22320323

Matthew Greene-21421546

ABSTRACT-Eimaan Ali

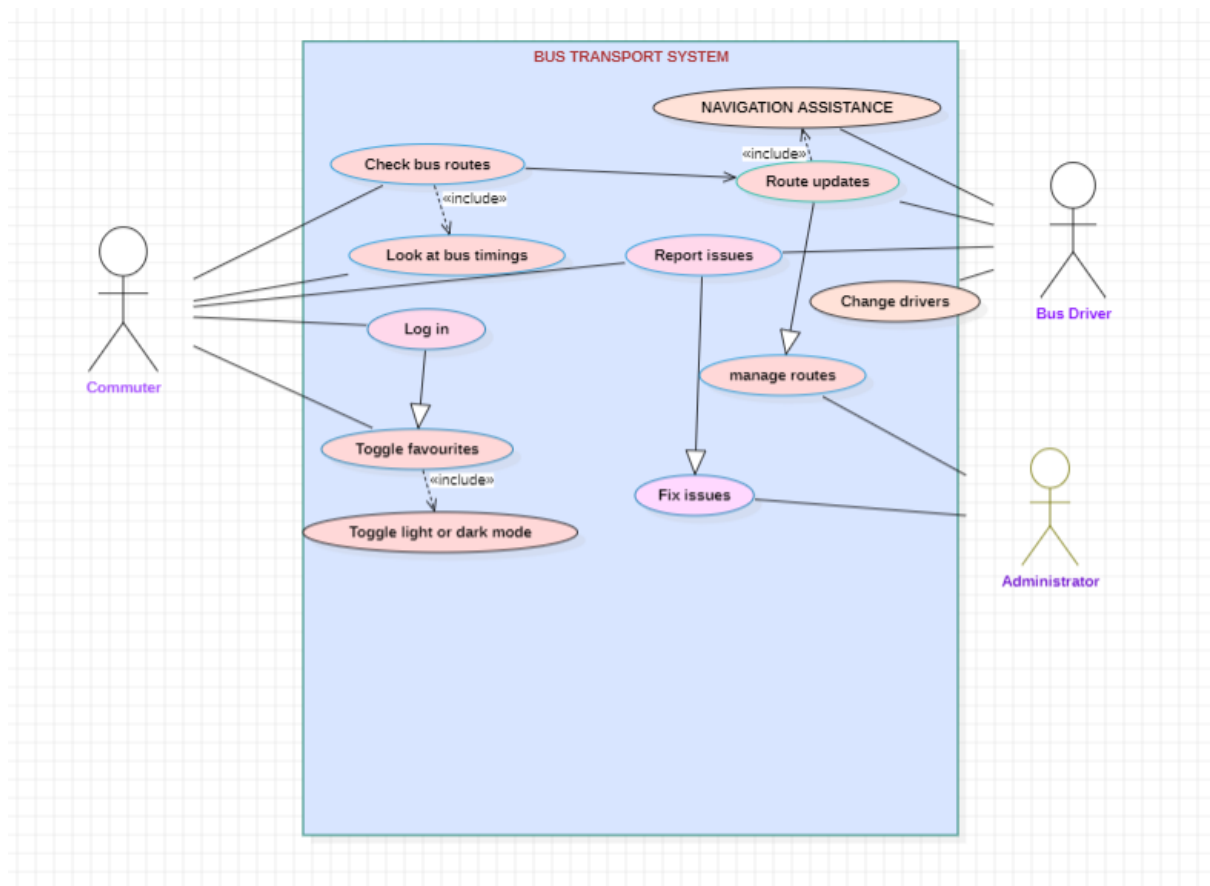
This project develops a Bus Transport System Software designed to enhance operational efficiency and passenger experience. The system supports real-time bus tracking, optimized route management, and effective communication tools, directly benefiting both commuters and bus operators.

User Stories and Requirements: Requirements engineering has been conducted to pinpoint the specific needs of bus system users, from drivers to commuters. This includes key functionalities like real-time bus location updates, efficient ticketing processes, and streamlined route scheduling.

Unified Modeling Language (UML) Diagrams: We utilize detailed UML diagrams, including use cases, sequence, and class diagrams, to accurately depict the system's architecture. These diagrams facilitate a clear understanding of the software's workflow and data interactions.

User Interface Design: The user interface is crafted for simplicity and accessibility, ensuring that all users, particularly commuters, can easily access bus schedules, live tracking information, and ticket services. This approach significantly improves the usability and effectiveness of the bus transport system.

USE CASE DIAGRAM(Hari Venkatesan)



DESCRIPTION(Hari Venkatesan):

The purpose of this Use-Case Diagram was to create scenarios for buses and bus users. The commuter, the administrator and the bus drivers were the 3 types of users we used to describe the potential necessities that they may need to solve

USE-CASE DIAGRAM CREATION(Hari Venkatesan):

The commuter was the Primary Actor for the application as they would be much more involved with the app than the bus driver or administrator. The bus driver and administrator are the secondary actors because they would use the app based on feedback from the commuter and react accordingly.

All my actors would work independently so I didn't group any of them together. Creating this diagram helped me better understand the roles of those using the app, the requirements the commuter predominantly needs, and how to combat issues when and if they arise. Also, the layout clearly depicts how the application reacts depending on how the user interacts with it.

User Stories for Transport System Software

(Eimaan Ali)

User stories drive the development of our transport system software, focusing on tailored functionalities for distinct user roles and system requirements that ensure robust, reliable, and user-friendly service.

USER STORIES

- As a [commuter \(passenger\)](#), I want to be able to view the real-time locations of buses on a map on my smartphone so that I can reduce my waiting time at the bus stop.
- As a [commuter \(passenger\)](#), I want to be able to set up alerts for my favourite routes so that I can receive notifications about any disruptions or changes directly on my smartphone and live tracking of my buses, ensuring I always have the latest information.
- As a [tourist](#), I want to search for public transport routes and schedules on my smartphone that connect specific tourist attractions so that I can plan my visits more effectively.
- As a [tourist](#), I want to be able to purchase tickets through my smartphone for various public transport options so that I can seamlessly travel across the city without needing to physically buy tickets at each station.
- As a [bus driver](#), I want to receive updates about my next trips and any changes to them directly on my in-vehicle display system so that I can manage my route and schedule more efficiently.
- As a [bus driver](#), I want to have access to an automated fare collection system integrated into my in-vehicle display that allows passengers to pay their fares digitally, so that I can ensure efficient, contactless transactions and focus more on driving safely.
- As an [administrator](#) of the transport system, I want to be able to track all vehicles in real-time from a central dashboard so that I can optimize routes and respond to any incidents quickly.
- As an [administrator](#) of the transport system, I want the ability to send mass notifications to all drivers regarding critical updates or emergencies so that I can ensure all staff are informed simultaneously, enhancing communication and response times.

- As a [transport planner](#), I want to generate reports on the usage patterns of bus routes so that I can make data-driven decisions about where to add new routes or enhance existing ones.
- As a [transport planner](#), I want to access predictive analytics tools in the software that use historical data to forecast future passenger loads on different routes, allowing me to proactively adjust resources and schedules to meet anticipated demand.

FUNCTIONAL REQUIREMENTS (Eimaan Ali)

- **Real-time Tracking:** The system must provide real-time tracking of all public transportation vehicles in the network.
- **Route Planning:** The system should allow users to input a starting point and destination and receive the best possible routes using public transportation.
- **Schedule Management:** The system must enable drivers to view and manage their schedules and any updates to their routes.
- **Central Management Dashboard:** There must be a dashboard for administrators to monitor and manage the fleet, including vehicle locations and operational status.
- **Reporting Tools:** The system should provide tools for generating various operational and usage reports.

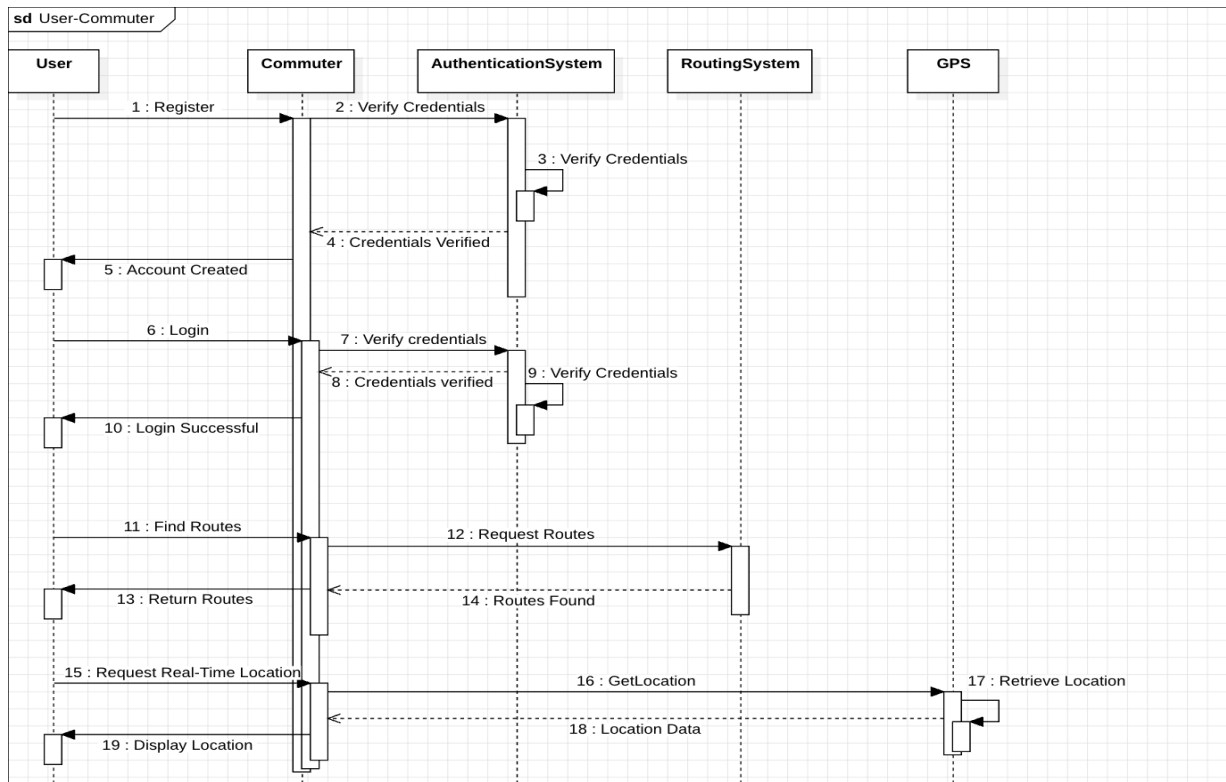
NON-FUNCTIONAL REQUIREMENTS (Eimaan Ali)

- **Usability:** The user interface should be intuitive and easy to use, accommodating users with various levels of technical proficiency.
- **Accessibility:** The system should be accessible on multiple devices, including smartphones, tablets, and desktop computers, and support accessibility features for disabled users.
- **Reliability:** The system should be operational 24/7, with a downtime of less than 0.1% annually.

SEQUENCE DIAGRAMS (Eimaan Ali)

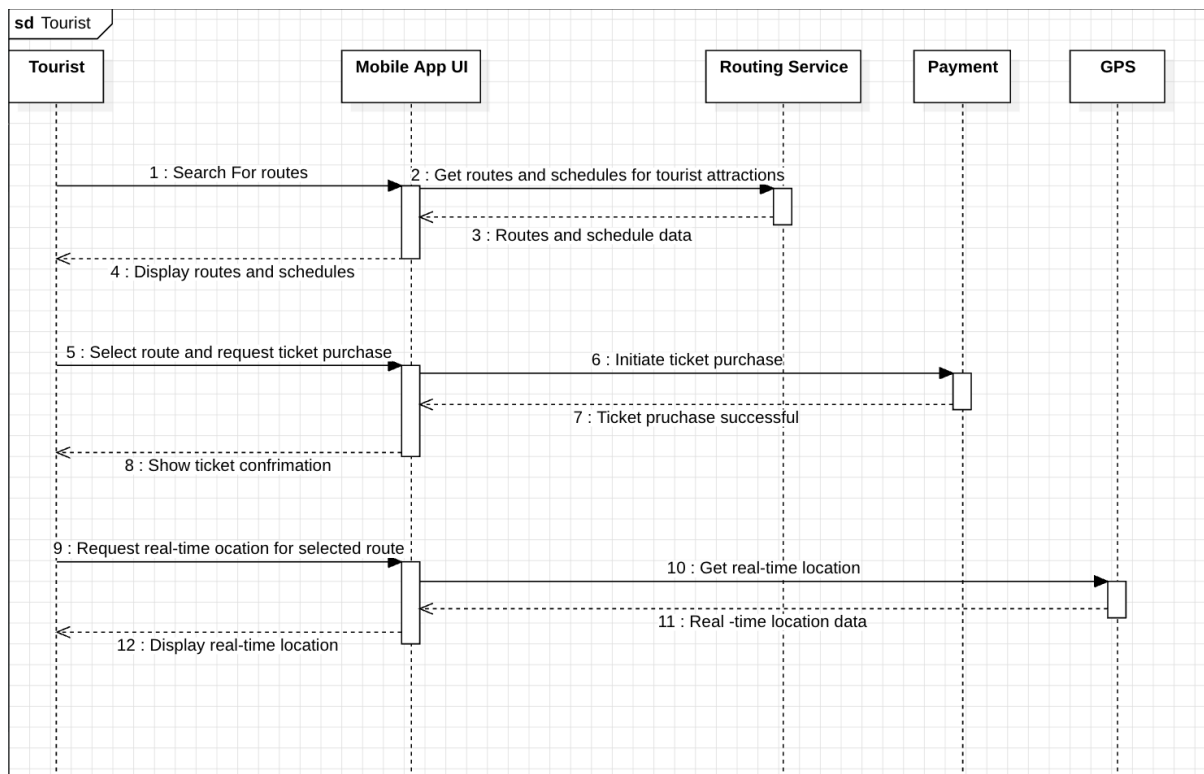
Sequence diagrams are essential for understanding and improving our transportation systems. By mapping out the interactions within the system, we can enhance our operational efficiencies, respond more effectively to incidents, and improve overall

COMMUTER



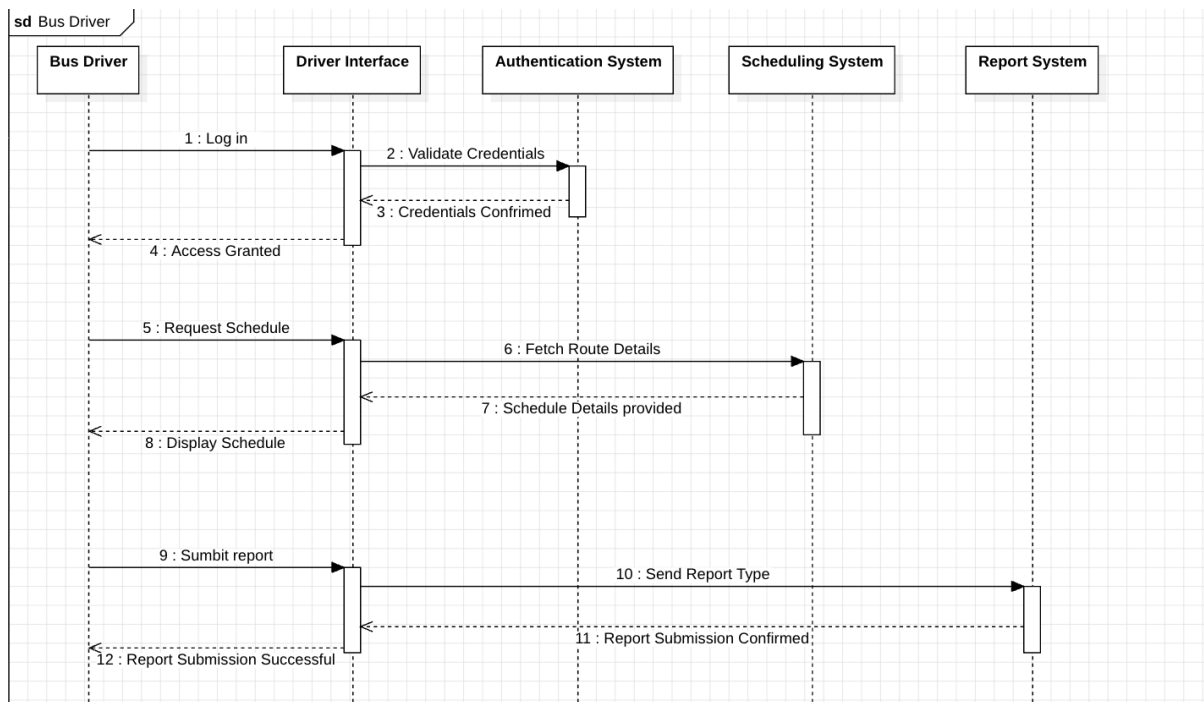
- **User Registration and Login:** A User registers via the Commuter interface, which verifies their details with the Authentication System. On success, the User is notified of account creation.
- For login, the User's credentials are verified similarly, and a successful login message is displayed.
- **Route Finding:** The logged-in User searches for routes. The Commuter interface requests this from the Routing System, which then returns the options back to be displayed to the User.
- **Real-Time Location Tracking:** The User can request real-time location updates. The Commuter interface retrieves this from the GPS system and presents the live data to the User.

TOURIST (Eimaan Ali)



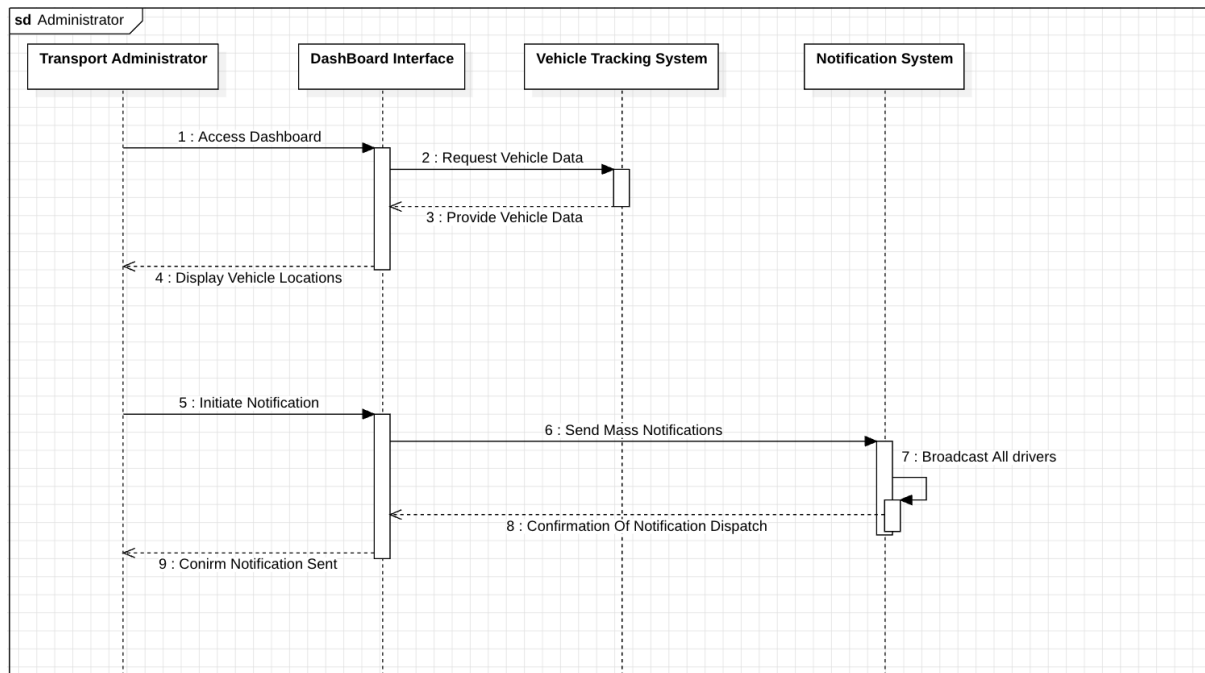
- **Registration and Access:** A Tourist creates an account through the transport app's interface, which is validated by the Authentication System. Once registered, the tourist logs in with their credentials, receiving confirmation upon successful entry.
- **Exploring Routes:** The authenticated Tourist uses the app to look for public transport routes that connect to major tourist attractions. The search request is processed by the Routing System, which delivers various travel options. These are then displayed to the Tourist through the app's interface.
- **Live Location Services:** For more efficient planning, the Tourist can also request real-time updates on the location of transport vehicles. The app interfaces with the GPS system to fetch this data, providing the Tourist with current location information to aid in travel decisions.

BUS DRIVER (Eimaan Ali)



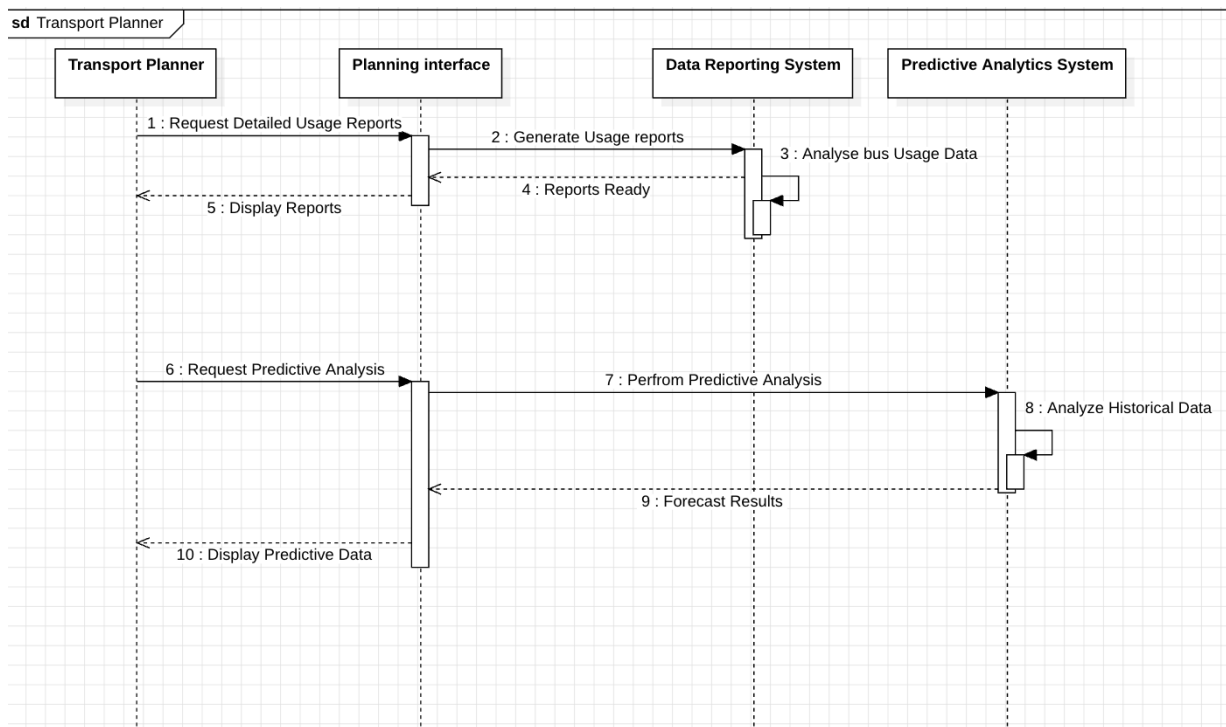
- **Security and Authentication:** The system ensures that only authorized drivers can access route information and submit reports, maintaining operational security.
- **Efficiency:** The automated process for retrieving route schedules and submitting reports saves time for the driver, allowing them to focus more on driving and less on administrative tasks.
- **Communication:** The sequence diagram illustrates how communication flows between the bus driver and various backend systems, ensuring that all actions are tracked, and responses are timely.

ADMINISTRATOR (Eimaan Ali)



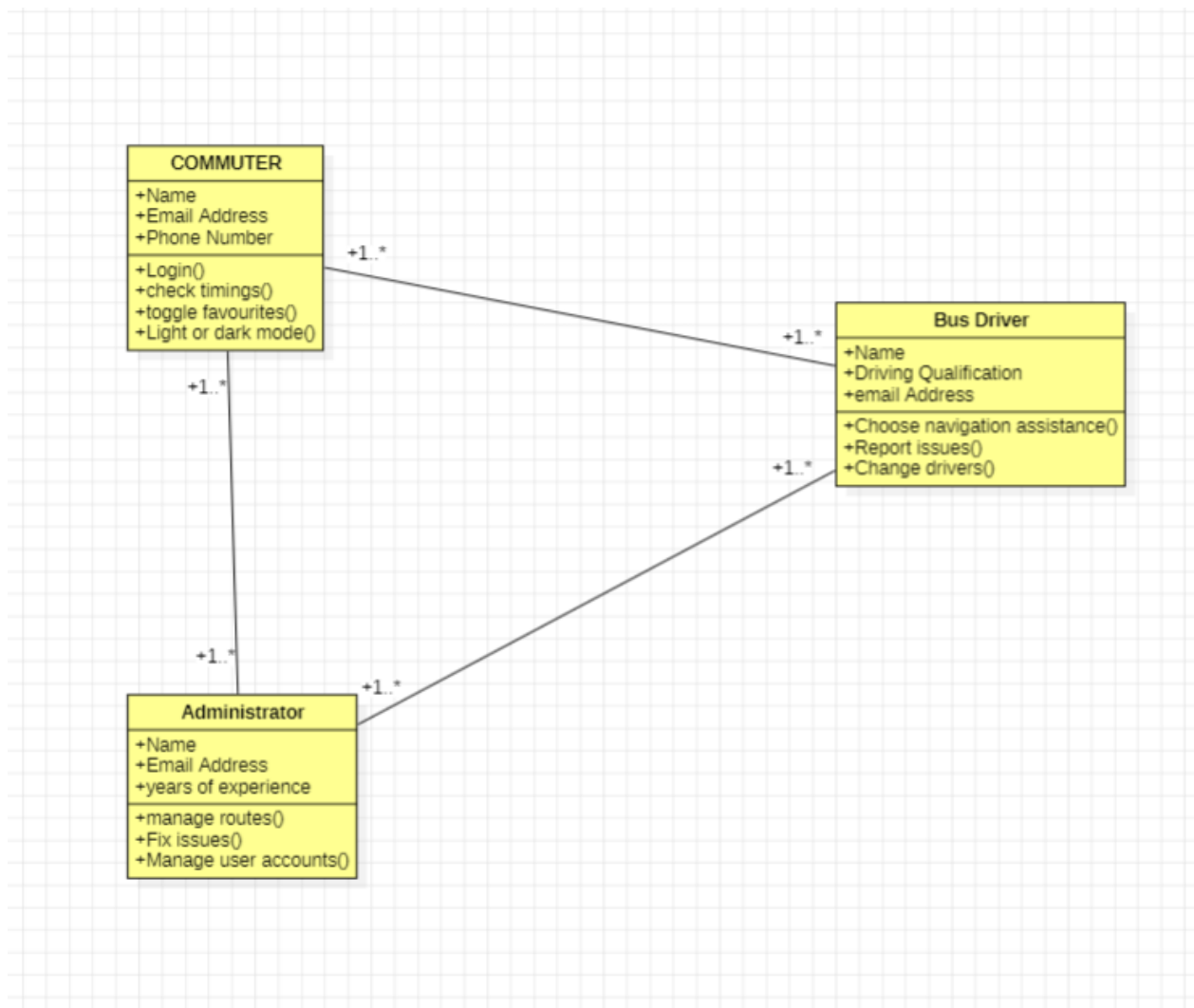
- **Efficiency and Responsiveness:** The ability to track vehicles in real-time allows the administrator to make informed decisions that enhance operational efficiency and safety.
- **Enhanced Communication:** The mass notification system ensures that all drivers receive crucial information simultaneously, reducing delays in communication and increasing the overall responsiveness of the transport system.
- **Centralized Control:** The central dashboard serves as a critical tool for the administrator, providing a single point of control for monitoring, communication, and management of the transport system.

Transport Planner (Eimaan Ali)



- **Strategic Planning Support:** The sequence diagram demonstrates how the transport planner leverages technology to make data-driven decisions that enhance operational efficiency and passenger satisfaction.
- **Integration of Advanced Analytics:** Showcases the use of both retrospective data analysis for reporting and predictive modelling to forecast future conditions, highlighting the system's advanced capabilities.
- **Impact on Service Improvement:** Discuss how these insights enable the transport planner to proactively manage resources, plan better routes, adjust schedules, and potentially introduce new services based on anticipated passenger behaviour and demand.

CLASS DIAGRAM (Hari Venkatesan)



DESCRIPTION (Hari Venkatesan)

The class diagram involves creating various objects that the app would contain that the users may encounter. Each class that we created has their own attributes, uniquely identifying and describing each type of user that uses the app, and operations that describes the users' roles whilst using the app.

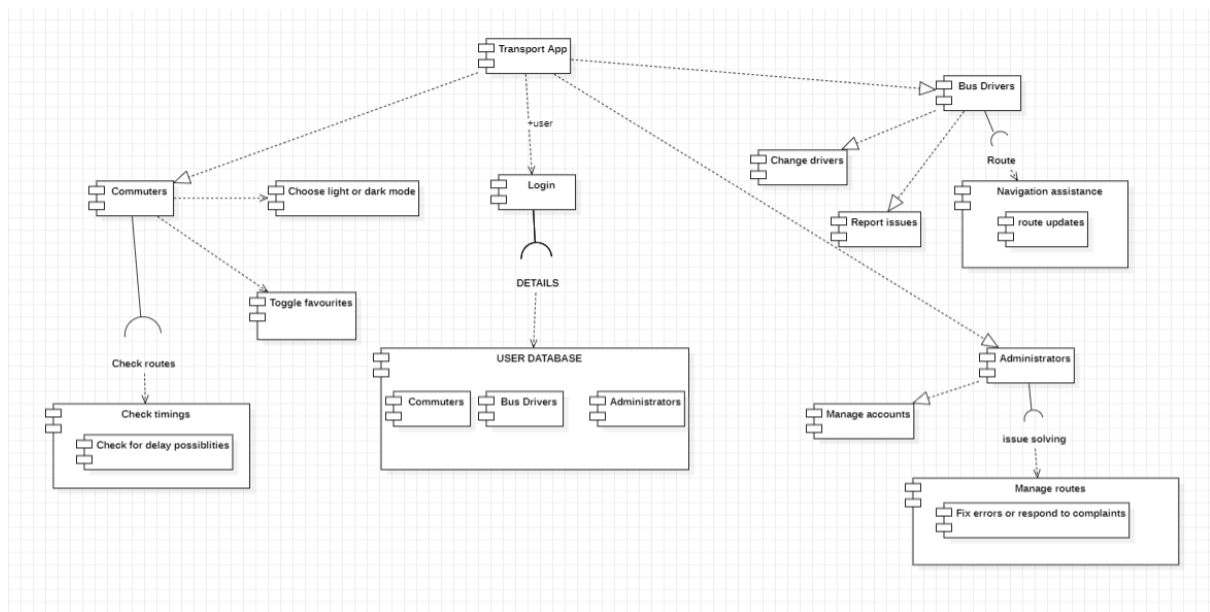
CLASSES (Hari Venkatesan)

Each class has their own attributes and operations which contains specific types of information for each user, both describing them and their tasks. For example, only the administrator can manage bus routes and/or fix potential issues, and only the bus driver can change the bus driver. The relationships between the users are very important as well. It depicts how each user works off each other. The administrator and bus driver would tend to respond to the commuters' feedback.

UNDERSTANDING(Hari Venkatesan)

Constructing this class diagram really helped me understand how classes and their respective objects work and interact with each other. I believe that it made describing this overall system far easier rather than just typing a long block of text. Anyone creating a similar system after reading the class diagram should have a better understanding of what the role of every user of this app is. It should also help make creating the user interface later.

ARCHITECTURE DIAGRAM(Hari Venkatesan)

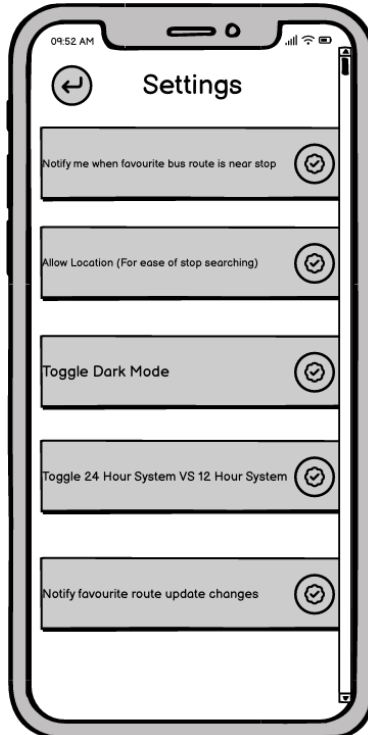
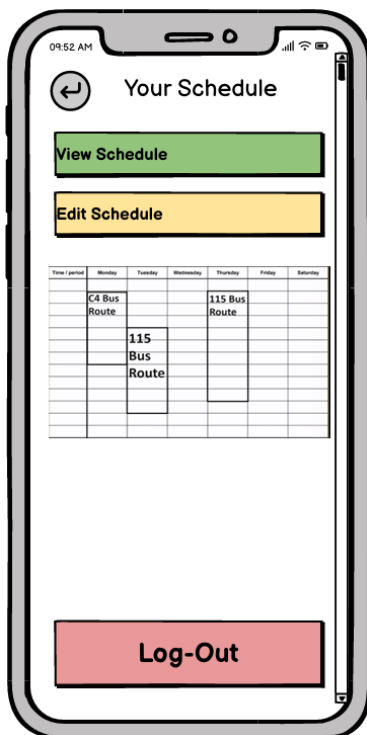
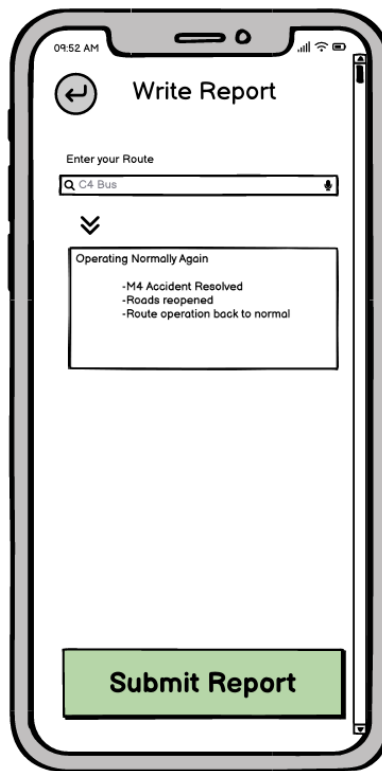
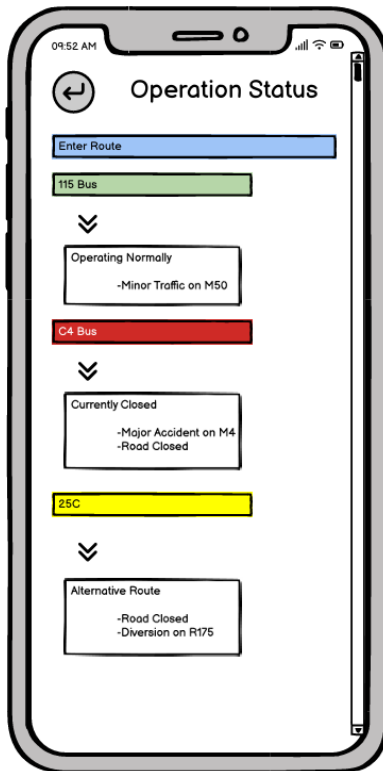


DESCRIPTION(Hari Venkatesan)

The architecture diagram is a simple diagram describing how classes can be packaged together so that it can be reused later. It breaks down the idea of this system into smaller parts so that it can be clearly presented to the end-user. From the diagram, it is easy to understand what the role of each user is and how this can be implemented into the app.

User Interface Model (Matthew Greene)





Description

This is the interface for our purposed app idea. There are two portions, the user/bus rider interface and the admin/Bus Operator side. The concept for the interface of this bus app was to make it both simplistic and multifunctional. Users tend to react more positively to apps with a more friendly appearance. The simplicity also synergizes well with it's purpose to a bus driver. The interface of the app encourage users to open the app to give it a quick check, maybe send a small notice about a bus route delay. We don't want to take too much time off the bus driver as they have duties of their own

System Testing (Matthew Greene)

No.	Test Cases	Test Data	Expected Results	Actual Result
1	Can commuter access Live Times	Email Address, Check timings()	Returns Live Timings for buses	Success!
2	Can a tourist use route finder service	Manage routes()	Returns Optimal routes	Success!
3	Can commuters favourite their bus stops	toggle favourites()	Sorts bus stops by favourites and unfavourites	Success!
4	Can Bus Drivers check for updates on routes	Report issues()	Returns list of reports for requested routes	Success!
5	Can Bus Drivers/Admins check/manage routes	manage routes()	Returns updated listing of live times	Success!
6	Can any Users toggle app appearance	Light/dark mode()	Updates appearance of app	Success!
7	Can Bus Drivers log-in correctly	Login() +Email Addresss	Returns dashboard access for Bus driver	Success!
8	Can Administrators manage accounts	Manage user accounts()	Returns a list of registered accounts	Success!
9	Can Administrators manage routes	manage routes()	Returns of concurrent bus routes	Success!
10	Can tourist purchase ticket	purchase()	returns purchased ticket	Success!

System testing serves as the final assurance that the software meets both functional and non-functional requirements, this gives stakeholders confidence in its reliability and performance before deployment. Through meticulous testing, errors are identified and rectified, guaranteeing a user-friendly end product.