**REPUBLIC OF CAMEROON**

PEACE-WORK-FATHERLAND

P.O. Box 63, Buea, South West Region CAMEROON Tel : (237) 3332 21 34/3332 26 90 Fax: (237) 3332 22 72

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER ENGINEERING**



**DEVELOPMENT OF A PASSENGER POSITIONING SYSTEM FOR THE MUNICIPALITY OF BUEA (TRIPTRACK)**

*A dissertation submitted to the Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea, in Partial Fulfilment of the Requirements for the Award of Bachelor of Engineering (B.Eng.) Degree in Computer Engineering*

**By:**

**Group 9: Passenger Positioning System for the municipality of Buea**

**Option**: Software Engineering

**Supervisor**

Dr. NKEMENI VALERY

University of Buea

**2022/2023 Academic Year**

**DEVELOPMENT OF A PASSENGER POSITIONING SYSTEM FOR THE MUNICIPALITY OF BUEA (TRIPTRACK)**

**PRESENTED BY:**

|  |  |
| --- | --- |
| CHI AZAH SHEYRHON | FE20A027 |
| LEGIMA DONAL | FE20A055 |
| NGUESSONG ANIKUI SUZY | FE20A079 |
| NKEMATABONG DIRAN FOLEFAC | FE20A089 |
| TENDONGMOH DELEO A. | FE20A117 |

**2022/2023 Academic Year**

***Dissertation submitted in partial fulfilment of the Requirements for the award of Bachelor of Engineering (B.Eng.) Degree in Computer Engineering.***

**Department of Computer Engineering**

**Faculty of Engineering and Technology**

**University of Buea**

# Certification of Originality

We the undersigned, hereby certify that this dissertation, entitled: “**DEVELOPMENT OF A PASSENGER POSITIONING SYTEM FOR THE MUNICIPALITY OF BUEA(TRIPTRACK)**”

presented by:

|  |  |
| --- | --- |
| CHI AZAH SHEYRHON | FE20A027 |
| LEGIMA DONAL | FE20A055 |
| NGUESSONG ANIKUI SUZY | FE20A079 |
| NKEMATABONG DIRAN FOLEFAC | FE20A089 |
| TENDONGMOH DELEO A. | FE20A117 |

has been carried out by us in the Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea under the supervision of **Dr. NKEMENI VALERY**

This dissertation is authentic and represents the fruits of our own research and efforts.

**Date**  June 18, 2023

Student Supervisor

**Head of Department**

# Dedication

This project report is dedicated to all the passengers who rely on public transportation to reach their destinations.

We developed the passenger positioning system with the aim to enhance the safety, convenience, and overall experience of public transportation for passengers, most importantly to enable drivers minimize fuel consumption. We hope that our work will contribute to making public transportation more accessible and efficient for everyone, especially those who depend on it as their primary means of transportation.

We also dedicate this report to our families and loved ones who provided us with unwavering support, encouragement, and inspiration throughout the project. Their love and belief in our abilities kept us motivated and focused on achieving our goals.

Additionally, we would like to acknowledge and honor all the frontline workers, including taxi drivers, who have worked tirelessly to keep public transportation running despite the tough economic conditions. Their dedication and service are a testament to their commitment to serving their communities.

Finally, we dedicate this report to the future generations of engineers, innovators, and problem-solvers who will continue to push the boundaries of technology and create solutions that benefit society. We hope that our work will inspire and encourage them to pursue their dreams and make a positive impact on the world.

Thank you all for your support and inspiration."

# Acknowledgement

We would like to express our sincere gratitude to all the individuals who have contributed to the development of the passenger positioning system (TripTrack).

First and foremost, we would like to thank our project supervisor Dr Nkemeni Valery for his guidance, valuable insights, and support throughout the project. His expertise and encouragement have been instrumental in the successful completion of this project.

We would also like to thank the Google's search engine for providing us with the necessary resources, and facilities to carry out this project.

We extend our appreciation to our team members who worked tirelessly to develop and test the passenger positioning system. Their dedication, hard work, and collaboration were essential in achieving our project goals.

Finally, we would like to thank all the passengers who participated in the user testing of the system. Their feedback and suggestions played a crucial role in improving the functionality and usability of the system.

Once again, we express our heartfelt appreciation to everyone who has contributed to this project.

# Abstract

The Passenger Positioning System (PPS) is a web-based application designed to track the location of public transportation vehicles in real-time and provide passengers with accurate information about the estimated arrival time of their bus or train. The aim of this project was to develop a reliable and user-friendly PPS that would enhance the safety and convenience of public transportation for passengers.

The project involved several stages, including research, design, development, and testing. The research phase involved an extensive literature review of existing PPS solutions and user requirements analysis to identify the key features and functionalities that passengers would need in a PPS. The design phase involved the creation of wireframes and mockups to visualize the user interface and user experience of the PPS.

The development phase involved the implementation of the PPS using ReactJS for the front-end and Node.js for the back-end. The PPS was integrated with a GPS tracking system to provide real-time vehicle location data to the passengers. The testing phase involved user testing and validation to ensure that the PPS met the user requirements and provided accurate and reliable information to the passengers.

The findings of the project show that the PPS is an effective solution for improving the safety and convenience of public transportation for passengers. The PPS provides accurate and real-time information about the estimated arrival time of the taxi, which can help passengers plan their journey more efficiently and reduce their wait time at the transit stop. The PPS also improves the overall user experience of public transportation, making it more accessible and convenient for all passengers.

In conclusion, the Passenger Positioning System developed in this project is a valuable tool for enhancing the safety and convenience of public transportation for passengers while ensuring the driver spends less fuel and makes maximum income by heading to areas with more passengers.

Table of Contents

[Certification of Originality 3](#_Toc138023095)

[Dedication 5](#_Toc138023096)

[Acknowledgement 5](#_Toc138023097)

[Abstract 6](#_Toc138023098)

[List of Figures 8](#_Toc138023099)

[List of Abbreviations 9](#_Toc138023100)

[CHAPTER 1. GENERAL INTRODUCTION 10](#_Toc138023101)

[Introduction: 10](#_Toc138023102)

[Purpose: 10](#_Toc138023103)

[Problem Statement 10](#_Toc138023104)

[Business and user Requirements: 11](#_Toc138023105)

[Business requirements: 11](#_Toc138023106)

[User requirements: 12](#_Toc138023107)

[Functional requirements: 12](#_Toc138023108)

[Non-functional requirements 13](#_Toc138023109)

[CHAPTER 2. LITERATURE REVIEW 15](#_Toc138023110)

[Existing passenger positioning systems 15](#_Toc138023111)

[Positioning technologies: 15](#_Toc138023112)

[Data storage and processing 15](#_Toc138023113)

[Passenger behavior analysis 16](#_Toc138023114)

[User acceptance and satisfaction 16](#_Toc138023115)

[Privacy and security: 16](#_Toc138023116)

[CHAPTER 3. ANALYSIS AND DESIGN 17](#_Toc138023117)

[System Design 20](#_Toc138023118)

[USE CASE 20](#_Toc138023119)

[Class diagram 22](#_Toc138023120)

[Sequence diagram 25](#_Toc138023121)

[Activity Diagram 29](#_Toc138023122)

[User Interface Design 31](#_Toc138023123)

[Database Design 38](#_Toc138023124)

[CHAPTER 4. IMPLEMENTATION (or REALIZATION) AND RESULTS 43](#_Toc138023125)

[CHAPTER 5. CONCLUSION AND FURTHER WORKS 47](#_Toc138023126)

[Conclusion: 47](#_Toc138023127)

[Further works: 47](#_Toc138023128)

[References 49](#_Toc138023129)

# List of Figures

[Figure 1: system use case diagram 24](#_Toc138021763)

[Figure 2: system class diagram 25](#_Toc138021764)

[Figure 3: sequence diagram for registration 27](#_Toc138021765)

[Figure 4:sequence diagram for login use case 28](#_Toc138021766)

[Figure 5:sequence diagram to view passenger’s location 29](#_Toc138021767)

[*Figure 6:sequence to allow access to user’s location* 30](#_Toc138021768)

[*Figure 7:system activity diagram* 31](#_Toc138021769)

[Figure 8:landing page on a mobile view 33](#_Toc138021770)

[Figure 9: signup page for every user 34](#_Toc138021771)

[Figure 10:login interface 35](#_Toc138021772)

[Figure 11: map page on user's end 36](#_Toc138021773)

[Figure 12: settings page 37](#_Toc138021774)

[Figure 13: verify code for change password 37](#_Toc138021775)

[Figure 14:user class screen shot from database 41](#_Toc138021776)

[Figure 15:driver class screenshot from database 41](#_Toc138021777)

[Figure 16:location class screenshot for database 41](#_Toc138021778)

[Figure 17:destination table screenshot from database 42](#_Toc138021779)

[Figure 18:passenger table screenshot from database 42](#_Toc138021780)

[Figure 19:screenshot of back-end to Django administration 43](#_Toc138021781)

[Figure 20:screenshot of created tables in the database 43](#_Toc138021782)

# List of Abbreviations

1. GUI: Graphical User Interface
2. PPS: Passenger Positioning System
3. GPS: Global Positioning System
4. LCD: Liquid Crystal Display

# CHAPTER 1. GENERAL INTRODUCTION

## Introduction:

Managing transportation is a collaborative activity that is based upon sharing and exchanging information.

Mobile phones are becoming smaller with modern GUIs, and are very powerful. In addition, they are now used everywhere and at any time (ubiquitous). The advancements and enhancement of the hardware went through have made mobiles smaller in size and more effective and efficient. Furthermore, it allows us to include many types of peripherals that are not limited with a specific number. Nowadays, there are three different methodologies to allow smart mobile phones to be capable of utilizing the location and positioning services:

 1) The utilization of Satellite Positioning,

2) The utilization of Wi-Fi Positioning and

 3) The utilization of Cellular Positioning.

## Purpose:

This is a website application to be used by both the passenger and the taxi drivers, where the passengers use it to specify their location or position at a given time, whereas the drivers use it to locate the exact location of more passengers.

This can help both drivers and passengers in the following ways:

* It reduces the amount of time passengers have to wait for a taxi
* Drivers can optimize fuel consumption as the app will guide them to move to locations where more potential passengers are available.

## Problem Statement

1. **Fuel:**

With the recent increase in the prize of fuel, fuel preservation is sort out by all drivers and transportation companies

1. **Lack of real-time information:**

Many transportation companies face the problem of not being able to provide their passengers with real-time information about vehicle locations and estimated arrival times, which can lead to frustration and dissatisfaction.

1. **Routing:**

Inefficient routing and time delay due to traffic. Drivers turn to get stuck in traffic due to inefficient information about routes with dense population

1. **Management:**

Difficulty in company vehicles distributions across the municipality: Transportation companies with many vehicles struggle with managing and partitioning all of their vehicles across the city.

# Business and user Requirements:

### Business requirements:

After analyzing the system, we were able to come out with the following business requirements

1. **ETA Calculation:** The system should be able to calculate accurate ETA for passengers and provide updates in real-time.
2. **Integration with Transportation Providers**: The system should be able to integrate with transportation provider’s systems to provide seamless communication and data exchange.
3. **Performance and Reliability:** The system should be able to handle a large number of users and provide reliable performance without downtime or outages.
4. **Scalability and Flexibility:** The system should be able to handle large numbers of vehicles and passengers, and be flexible enough to adapt to changing business needs and requirements.
5. **Improved Customer Experience:** The system should enhance the overall customer experience by providing accurate and timely information, easy-to-use interfaces, and efficient transportation services.
6. **Cost-Effectiveness**: The system should be cost-effective, with a reasonable return pricing
7. **Passenger Safety and Security:** The system should have features to ensure passenger safety and security, such as real-time monitoring.
8. **Emergency Response Capabilities:** the system should have ways to indicate passengers under undergoing and emergency so as to get the attention of the nearest vehicle
9. **Fuel:** Reduces fuel consumption since the drivers can avoid unnecessary rides and safe money.

### User requirements:

1. **Real-Time Updates**: The app should provide real-time updates on passenger locations and ETA.
2. **Multi-Platform Support:** The app should be available on multiple platforms, including iOS and Android.
3. **Personalization**: The app should be able to provide personalized information and settings for each user.
4. **Security And Privacy**: The app should have secure authentication and data protection mechanisms to ensure user privacy and prevent unauthorized access.
5. **Notifications**: The app should be able to send notifications to users regarding updates, alerts, or other relevant information.
6. **Easy-To-Use Interface:** The system should have a user-friendly interface that is easy to navigate and understand, with clear and concise information about vehicle locations, routes, and estimated arrival times.
7. **Mobile Accessibility:** The system should be accessible through mobile devices, such as smartphones and tablets, including integration with vehicle GPS LCD display
8. **Feedback and Ratings:** The system should allow passengers to provide feedback and ratings about their experience

### Functional requirements:

From analysis of the system we can deduce the following functional requirements.

* **User registration and login:** The system should allow users to create an account and login securely.
* **Location tracking:** The system should be able to track the location of passengers using GPS or other location-based technologies.
* **ETA calculation:** The system should be able to calculate accurate ETA for passengers based on their location and the transportation route.
* **Real-time updates:** The system should provide real-time updates on passenger locations and ETA to both users and transportation providers.
* **Route optimization:** The system should be able to optimize transportation routes based on passenger demand and traffic conditions.
* **Push notifications:** The system should be able to send push notifications to users regarding updates, alerts, or other relevant information.
* **Analytic and reporting:** The system should be able to generate reports and analytic on passenger behavior, route performance, and other relevant metrics.
* **Integration with transportation providers:** The system should be able to integrate with transportation providers' systems to provide seamless communication and data exchange.

### Non-functional requirements

* **Payment integration:** The system should be able to integrate with payment systems to allow users to pay for transportation services directly through the app.
* **Personalization:** The system should be able to provide personalized information and settings for each user, such as preferred routes or modes of transportation.
* **Performance**: The system should be able to handle a large number of users and provide reliable performance without downtime or outages.
* **Scalability**: The system should be able to scale up or down depending on the number of users and the demand for transportation services.
* **Security:** The system should have secure authentication and data protection mechanisms to ensure user privacy and prevent unauthorized access.
* **Usability:** The app should have an intuitive user interface that is easy to use, even for non-technical users.
* **Accessibility**: The app should be accessible to users with disabilities and provide support for assistive technologies.
* **Reliability:** The system should be highly available and provide reliable performance even under heavy load or adverse conditions.
* **Availability**: The system should be available 24/7 to meet the needs of users and transportation providers.
* **Maintainability:** The system should be easy to maintain and upgrade over time, with minimal disruption to users.
* **Compatibility:** The app should be compatible with a wide range of devices, operating systems, and browsers.
* **Data management:** The system should be able to manage and store large amounts of data, such as passenger locations and transportation routes, in a secure and efficient manner.

# CHAPTER 2. LITERATURE REVIEW

## Existing passenger positioning systems

There are a variety of existing passenger positioning systems in use today which include Africab, Cameroon Transit, Waya Waya and My bus, each with its own strengths and weaknesses.

One common type of passenger positioning system is based on GPS technology. This allows for accurate location tracking, but can be subject to signal interference in certain environments. Another type of system uses Bluetooth beacons to determine a passenger's location, but this can be less accurate than GPS and is limited by the range of the beacons.

Other systems use a combination of technologies, such as RFID and Wi-Fi, to track passenger locations. These systems can provide more accurate location data in certain environments, but may require additional hardware and infrastructure.

## Positioning technologies:

In several countries several positioning technologies such as GPS-based systems, Wi-Fi-based systems, Bluetooth-based systems and RFID-based systems are used to develop such a system.

* GPS-based systems use GPS technology to track the location of vehicles and passengers in real-time. They are commonly used in transportation systems such as buses, taxis, and ride-sharing services
* Wi-Fi-based systems use Wi-Fi signals to track the location of passengers within a specific area, such as an airport or shopping mall. They can provide more precise location information than GPS-based systems in certain indoor environments.
* Bluetooth-based systems use Bluetooth technology to track the location of passengers within a specific area. They are commonly used in airports and other transportation hubs to track passenger movements and provide real-time information to travelers.
* RFID-based systems use radio frequency identification (RFID) technology to track the location of passengers and their luggage within a specific area, such as an airport or train station. They can provide real-time information on passenger where abouts and help to improve security and efficiency in transportation systems.

## Data storage and processing

Data storage and processing are critical components of a passenger positioning system, as they enable the collection, storage, and analysis of vast amounts of location data. GPS data collected from passengers and drivers can provide valuable insights into passenger behavior, traffic patterns, and transportation system efficiency. In order to effectively utilize this data, a robust data storage and processing infrastructure must be in place.

Other approaches have been to use a distributed system, where location data is stored and processed across multiple devices or servers. This can provide greater scalability and flexibility, but may be more difficult to manage and secure.

## Passenger behavior analysis

Passenger behavior analysis is an important area of research for passenger positioning systems, as it can provide insights into passenger preferences, habits, and needs. This information can be used to improve the overall passenger experience and optimize transportation services.

One common approach to passenger behavior analysis is to use data mining techniques to extract patterns and trends from location data. This can help identify popular routes, peak travel times, and areas of congestion, which can inform service planning and scheduling.

## User acceptance and satisfaction

A review of studies on user acceptance and satisfaction with passenger positioning systems can help to identify factors that contribute to user adoption and satisfaction, as well as potential barriers to adoption.

## Privacy and security:

Privacy concerns: The collection and use of personal data in a passenger positioning system can raise privacy concerns, particularly if the system is tracking individuals in real-time. It is important to ensure that the system is designed in a way that protects the privacy of passengers and complies with relevant privacy laws and regulations.

Data protection: Passenger positioning systems may collect and store sensitive personal data, such as biometric data or location data. It is important to ensure that this data is protected from unauthorized access, use, or disclosure, and that appropriate data protection measures are in place.

Cybersecurity risks: Passenger positioning systems may be vulnerable to cybersecurity risks, such as hacking or data breaches. It is important to ensure that appropriate cybersecurity measures are in place to protect the system and the data it collects.

Transparency and consent: It is important to ensure that passengers are informed about the collection and use of their personal data and are given the opportunity to provide informed consent. Transparent communication about the purpose and scope of the passenger positioning system can help to build trust and improve user acceptance.

Ethical considerations: The use of passenger positioning systems raises ethical considerations, such as the potential for discrimination or stigmatization based on passenger movements or behavior. It is important to ensure that the system is designed and implemented in an ethical and responsible manner, and that potential risks and impacts are carefully considered. By conducting a comprehensive literature review on these and other related topics, researchers can gain a better understanding of the state of the art in passenger positioning systems, identify gaps in the existing literature, and inform the design and implementation of a new passenger positioning system.

# CHAPTER 3. ANALYSIS AND DESIGN

The analysis and design chapter of this report focuses on the development of a passenger positioning system. This chapter presents a detailed overview of the analysis and design process that was undertaken to create a system that accurately and efficiently tracks the position of passengers within a given space. The chapter begins by a discussion of the analysis and design methodologies that were used and concludes with a description of the final design of the system, including the technologies and algorithms utilized to achieve accurate passenger positioning. Overall, this chapter provides a comprehensive analysis of the design process for the passenger positioning system, and can serve as a valuable resource for others interested in developing similar positioning systems.

Analysis and Design methodology

For this system we made use of the agile model and a modular design system.

**Agile model**: The agile model is a software development methodology that is based on iterative and incremental development. It is often used in complex projects where requirements are likely to change, and there is a need for flexibility and adaptability. In the case of a passenger positioning system, there are several reasons why the agile model was used:

1. Dynamic requirements: The requirements for a passenger positioning system may be complex and constantly evolving. An agile approach allows for the flexibility to adapt to changing requirements throughout the development process, enabling the system to better meet the needs of the end-users.

2. Iterative development: The agile methodology involves breaking down the development process into smaller, more manageable iterations. This approach allows for a quicker response to user feedback and helps to ensure that the final product meets the needs of the users.

3. Collaborative approach: Agile development involves close collaboration between the development team and the end-users. This approach ensures that the system is developed with the end-users in mind, and that their needs and feedback are taken into account throughout the development process.

4. Faster time-to-market: The agile methodology emphasizes the delivery of working software in shorter timeframes, which can help to reduce time-to-market and ensure that the system is delivered on time and within budget.

5. Quality assurance: Agile development includes frequent testing and quality assurance measures, which can help to identify and address issues early in the development process, resulting in a more reliable and robust system.

Overall, the agile model was beneficial for developing the passenger positioning system, as it allowed adaptability, collaboration, while ensuring that the final product met the needs of the end-users and in high quality. The following points are some reasons why the agile methodology was used:

* Dynamic requirements: The requirements for a passenger positioning are complex and constantly evolving. An agile approach allows for the flexibility to adapt to changing requirements throughout the development process, enabling the system to better meet the needs of the end-users.
* Iterative development: The agile methodology involves breaking down the development process into smaller, more manageable iterations. This approach allows for a quicker response to user feedback and helps to ensure that the final product meets the needs of the users.
* Collaborative approach: Agile development involves close collaboration between the development team and the end-users. This approach ensures that the system is developed with the end-users in mind, and that their needs and feedback are taken into account throughout the development process.
* Faster time-to-market: The agile methodology emphasizes the delivery of working software in shorter timeframes, which can help to reduce time-to-market and ensure that the system is delivered on time and within budget.
* Quality assurance: Agile development includes frequent testing and quality assurance measures, which can help to identify and address issues early in the development process, resulting in a more reliable and robust system.

**Modular design**: A modular design is an approach to system design where the system is broken down into smaller, more manageable modules or components. Each module or component is designed to perform a specific function or set of functions, and can be developed and tested independently of the other modules. In the case of the passenger positioning system, there are several reasons why we used the modular design:

1. Scalability: A modular design allows for the system to be easily scalable, by adding or removing modules as needed. This approach helps to ensure that the system can adapt to changing requirements and can be easily expanded to accommodate future needs.

2. Flexibility: This design provides greater flexibility in system development, as each module can be developed and tested independently. This approach allows for greater flexibility in the development process and can help to reduce the risk of delays or errors caused by interdependencies between modules.

3. Maintainability: A modular design can make the system easier to maintain and update over time. If a module needs to be updated or replaced, it can be done without affecting the other modules, which can help to reduce downtime and increase system availability.

4. Reusability: A modular design can make it easier to reuse modules or components in other systems or applications. This approach can help to reduce development time and costs, and can lead to greater consistency and efficiency in system development.

## System Design

The system designs were developed using the unified modelling language (UML). UML is a standardized visual modeling language used in software engineering for designing and documenting software systems. It provides a common language and notation to communicate software design ideas and concepts between stakeholders such as developers, analysts, and end-users. The UML includes a wide range of diagrams, including use case diagrams, class diagrams, sequence diagrams, and activity diagrams, which our system made use of. The UML is widely used in software development projects to communicate design ideas and requirements, identify potential issues or design flaws, and support planning and development activities.

### USE CASE

A use case is a description of a specific interaction between a user (or actor) and the system to achieve a specific goal. A use case must yield an observable result that is of value to the user of the system. An actor represents a role of a user that interacts with the system that you are modeling

As mentioned, it consists of actors of different priorities to the system. The use cases in this diagram solve the problem of the functional requirements which was mention in the system requirement document done earlier

Use cases have three main type of actors but for our mobile application, we have two main actors which are the primary and secondary actors, which are located at the left and right of the use case diagram respectively

For a passenger positioning system, the actors in the system are.

**Actors**

* Passenger(Primary)
* Transportation Agency(Primary)
* System(Secondary)
* GPS(Secondary)

#### Various Actors and their Use Cases

1) Passenger

- Register

- Allow Access To their Location

- View Others passengers Location

- Login

2) Transportation Agency

- Register

- Login

- View Passengers Location

3) System

- Send Notification

- Regard Update

- Regard Alert

- Provide Relevant Info

- Calculate ETA

- Generate Report

- Provide Analysis

4) GPS

- Takes Passenger Location

- Provide Passenger Location

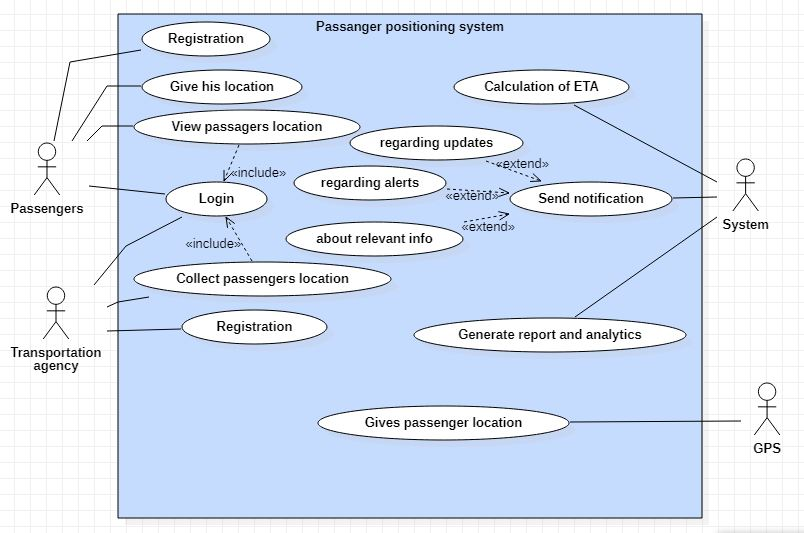


Figure 1: system use case diagram

### Class diagram

This is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

For the system we were required to model, we identified the following classes:

Implementing a class diagram for Passenger Positioning system:

After analyzing the system, we were able to come out with the various classes for our class diagram:

Main classes for class diagram

1. **Passenger**: Represents a passenger who is being tracked by the system.
2. **Location**: Represents a location within the transportation vehicle or facility.
3. **Sensor**: Represents a device that is used to detect the location of passengers.
4. **Tracking System**: Represents the main system that coordinates the tracking of passengers and communicates with the sensors.
5. **Transport Agency**: They are drivers or companies who locate passengers on the map and get to them thereby minimizing fuel consumption

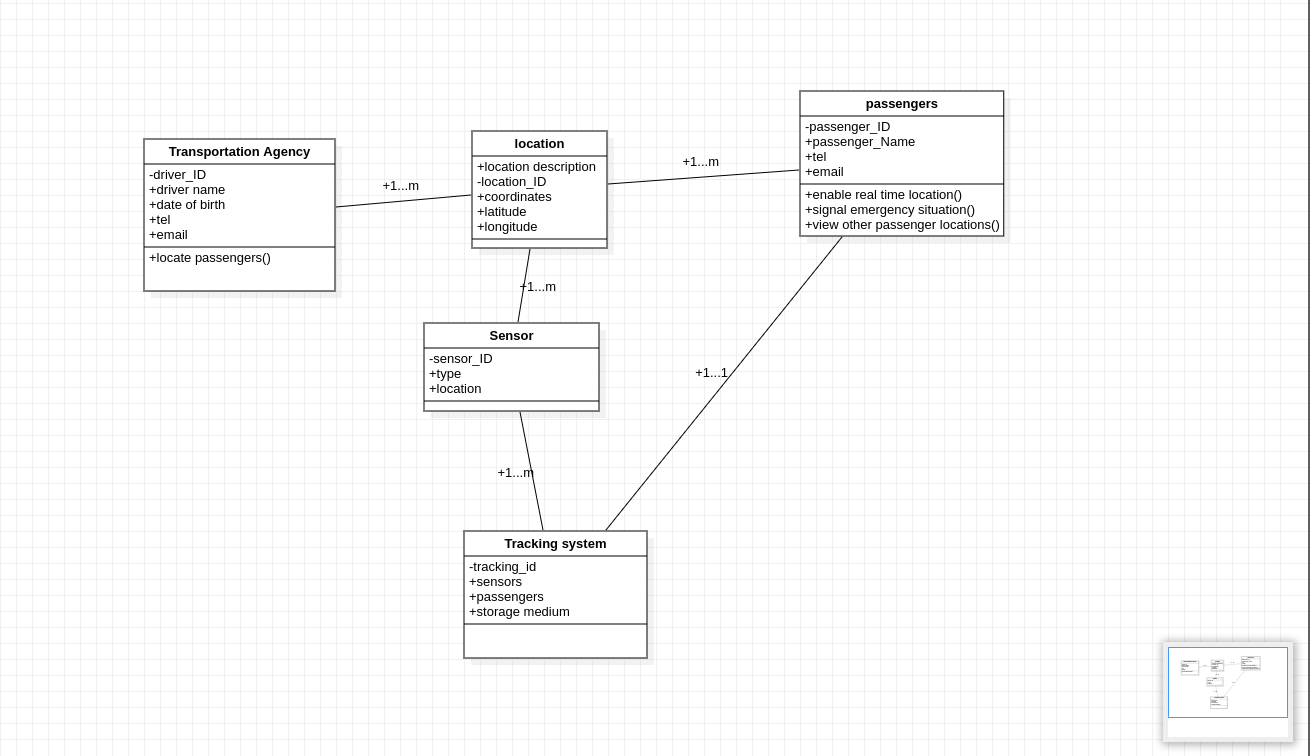


Figure 2: system class diagram

From the diagram above, we can view the various relationships among the classes:

* The location class is linked to the transport agency by a one to many relationships that is a transportation agency may have one or many locations viewed on the map.
* Passenger class too share a common relationship with the location class which is also in the sense that passenger can have one or multiple locations on the map.
* For each passenger in the system they can have one and only one tracking system per passenger for their device.
* Sensors class share a one to many relationships with both the location class and the tracking system class.
* The tracking class makes sure it gets details for each passenger based on the sensor on their various devices.

### Sequence diagram

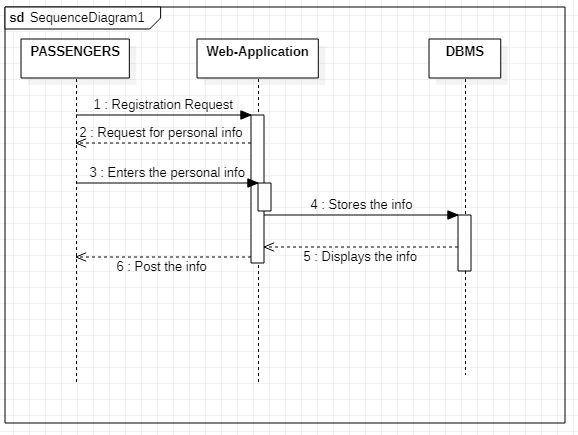
A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects.

Sequence diagrams always come up as a result of use cases of an application.

Even though not all of the use cases were used to provide the sequence diagram of the entire application, below are a series of sequence diagrams for some of the major use cases of the system.

Sequence Diagram for Register

* Firstly, we have a sequence diagram which came to existence from use case called Register

Figure 3: sequence diagram for registration

* Here Both primary users, which are the passengers and transport agency register their personal information on the web application which in turn processes the data and sends it in the database which is being hashed and stored since it’s their confidential information.
* Then a request is sent back to the web app signaling the users that the account has been successfully created.

Sequence Diagram for login Use case

* Secondly this sequence diagram is as a result of the Login use case which was assigned to the two primary actors that is the Passenger and the Agency.
* The scheme of flow is rather straightforward, in that before a passenger or Agency login their credentials are to be verified on the database.
* The Database checks if the information entered is correct and if correct returns a positive feedback else an error message to the system.

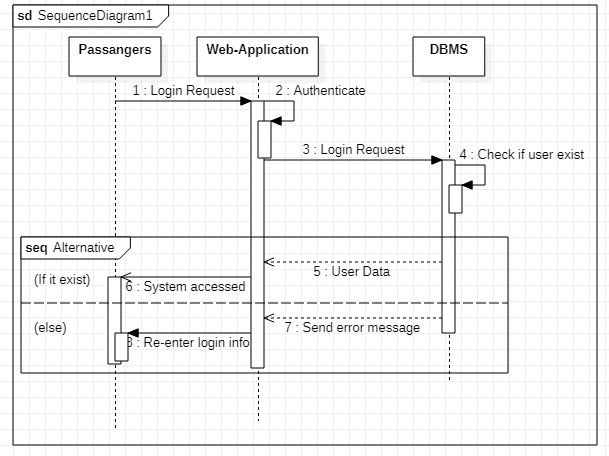


Figure 4:sequence diagram for login use case

Sequence Diagram for View Passengers Location

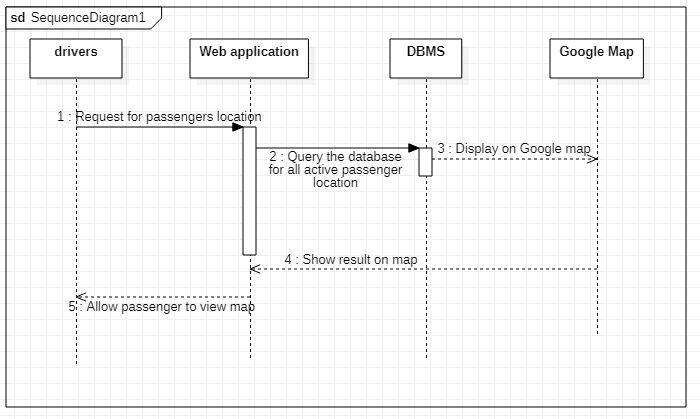
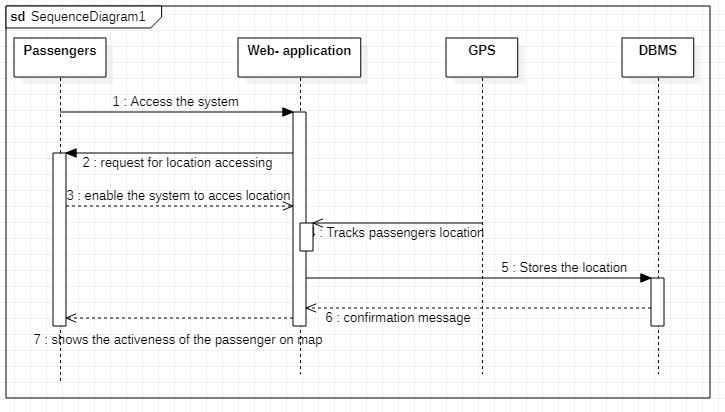
* Thirdly this sequence diagram is as a result of the View Passengers Location use case which was assigned to just one of the actors called the Transportation Agency though bought of the primary actors which are the passengers and the agency are able to view passenger’s location.
* This sequence represents the flow of the transport agency locating users on the map via geolocation.
* Here travel agencies requesting for passenger location is queried on the web app and the details are sent to the data base which retrieves the information and sends the location details to the map API which in turns return the output on the web application.
* Also, passengers will be able to see other passenger’s location which may drive them to go where there are many passengers in other to have more chances of having a taxi.

Figure 5:sequence diagram to view passenger’s location

Sequence Diagram for Allow Access to Location

The following sequence diagram gives us a brief overview of the passenger enabling real time location on the web application.

* Here, the passengers enable their location immediately after accessing the web app for real time monitoring of the passenger’s location.

*Figure 6:sequence to allow access to user’s location*

### Activity Diagram

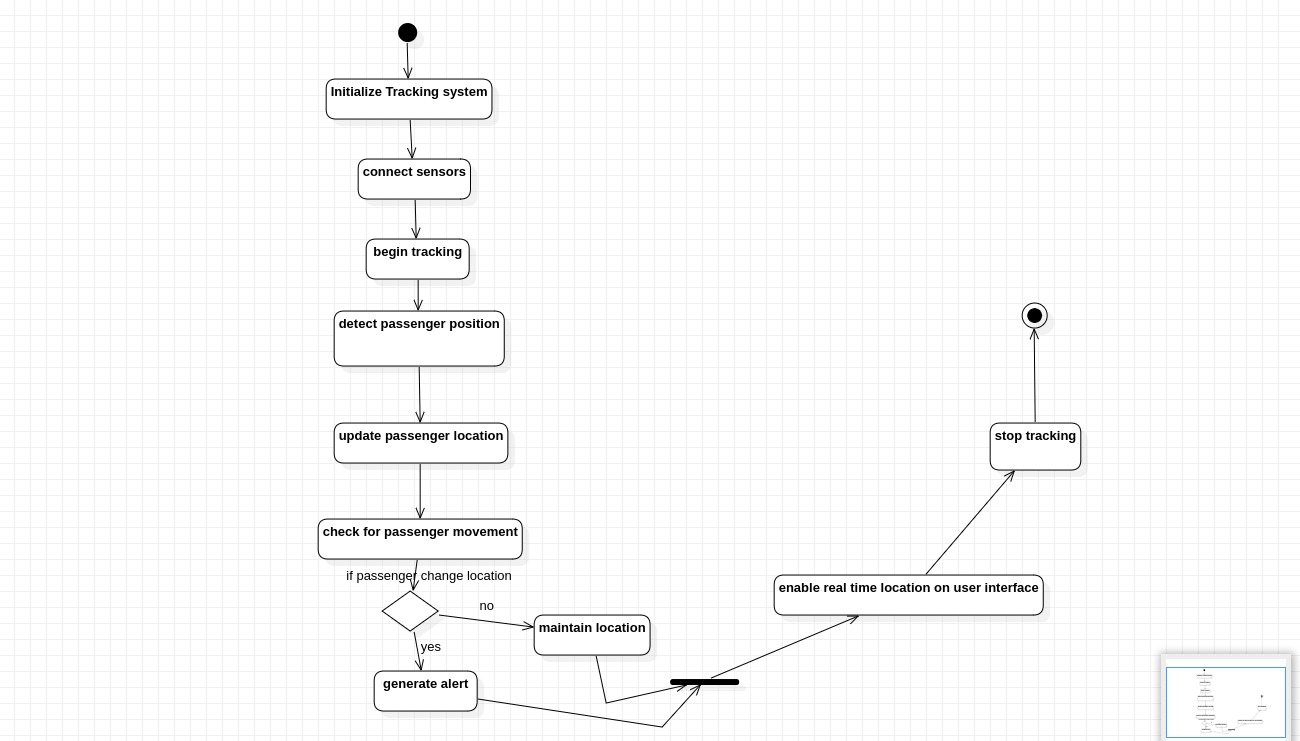
This visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

The activity diagram illustrates the basic flow of events in a passenger positioning system. The system starts by initializing the tracking system and connecting the sensors. It then begins tracking the passengers, detecting their location and updating it as necessary. The system checks for any unexpected movement by the passengers and generates an alert if necessary. The system also stores the passenger location data and provides real-time location data to the user interface. Reports on passenger movement can be generated as needed. Finally, the system stops tracking when the user disables on his phone and the process ends.

So, these are the various stages or activities in our system:

**Main scheme or flow**

1. Start
2. Initialize Tracking System
3. Connect Sensors
4. Begin Tracking
5. Detect Passenger Location
6. Update Passenger Location
7. Check for Passenger Movement
8. Generate Alert if Passenger Moves from Expected Location
9. Store Passenger Location Data
10. Provide Real-Time Location Data to User Interface
11. Generate Reports on Passenger Movement
12. Stop Tracking
13. End

*Figure 7:system activity diagram*

So that is the basic systematic flow of events relative to the flow of our system.

This summarizes the basic diagrams required for our system.

# User Interface Design

The purpose of this section is to outline the design of a passenger positioning system, which is intended to improve the overall experience of passengers using public transportation. The system is designed to provide real-time information about the location of taxis and passengers, as well as the expected arrival times at various stops based on various approximations. This will enable passengers to better plan their journeys and reduce the amount of time spent waiting for transportation while drivers on their part will be able to minimize their fuel consumption and make more profits.

This design was done basically with two key features in mind:

Accessibility features: The system was designed with accessibility in mind particularly with the mobile phone view.

Map-based interface: The system will display a map of the transportation network, with icons representing the locations of taxis and passengers in real-time.

This design had as goal to provide users with a clear and intuitive interface that is easy to use. The system should be accessible to a wide range of users, including those with disabilities or limited technological experience. Additionally, the system should be designed to accommodate a large amount of data, including real-time location data for multiple vehicles and expected arrival times at many different stops.

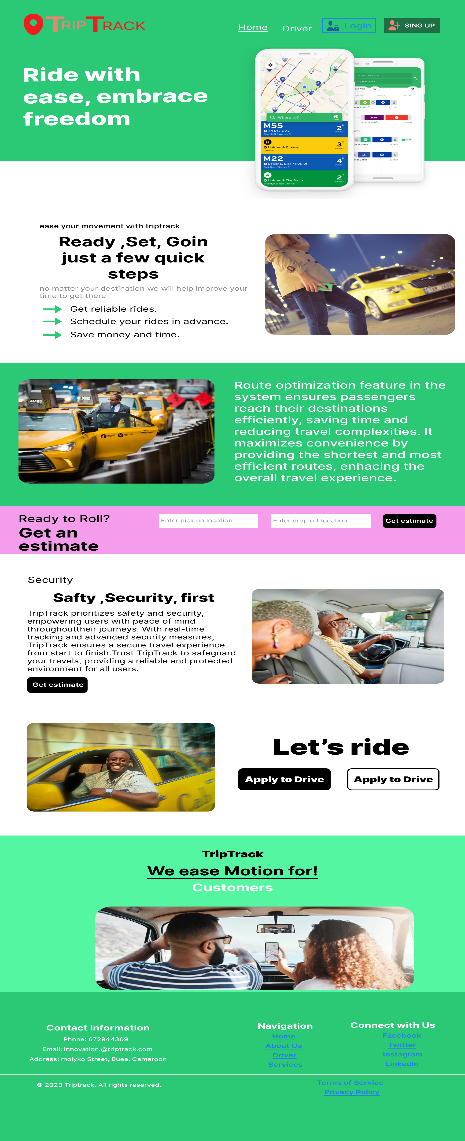
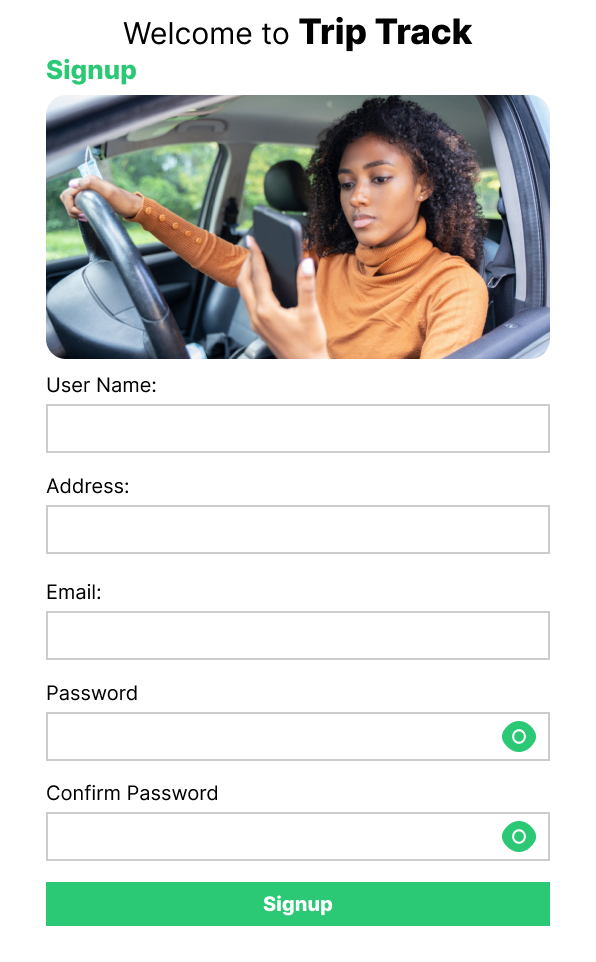
In the following pages we will present you some screenshots of the user interface.

Figure 8:landing page on a mobile view

This page is acts as a welcome page to all users of the system, it displays the singup for first timers, login for persons who already have an account, the about us, contact and some animations with vital text for the users to ensure safety.

Figure 9: signup page for every user

This page demands the user’s basic information such as his name, address, email and password which are vital for registration process to be conclude

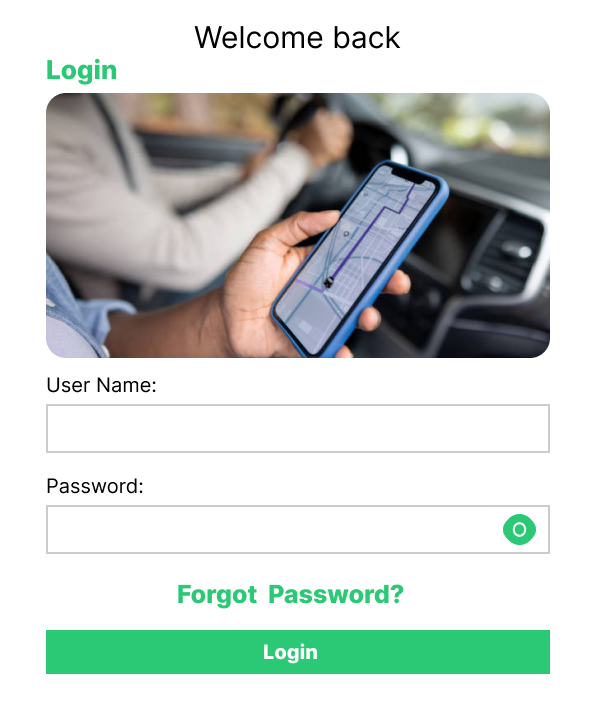
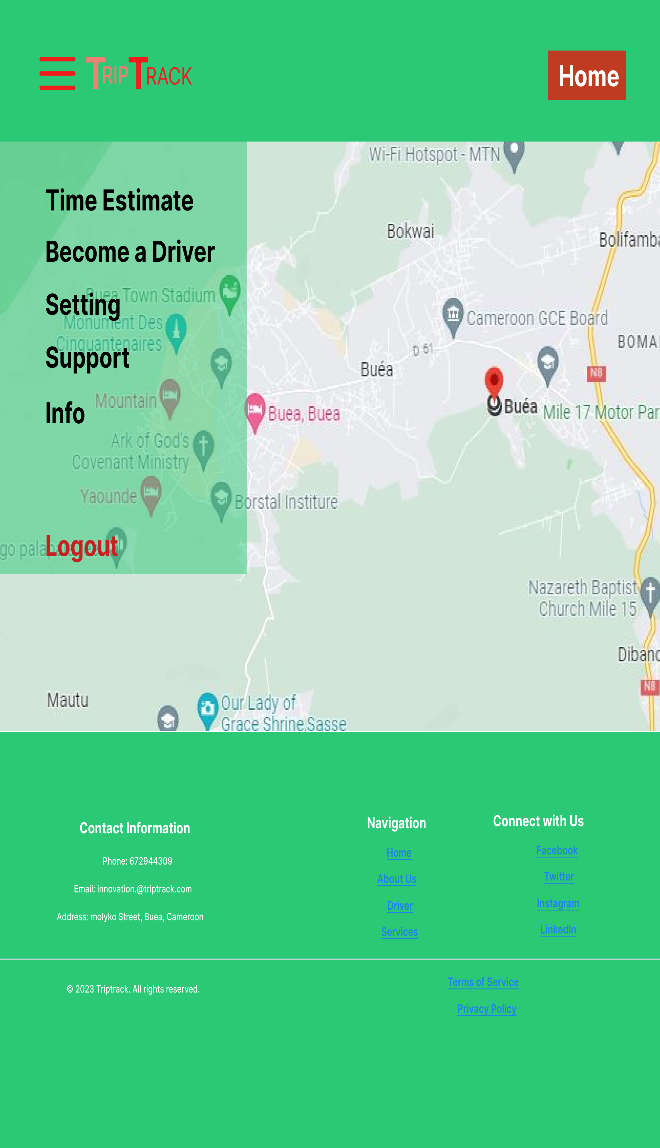


Figure 10:login interface

Upon successfully signing up, the user only needs to login whenever he needs to use the system.

Figure 11: map page on user's end

Once a user has successfully signed up, he is redirected to the map where he can input his location and destination

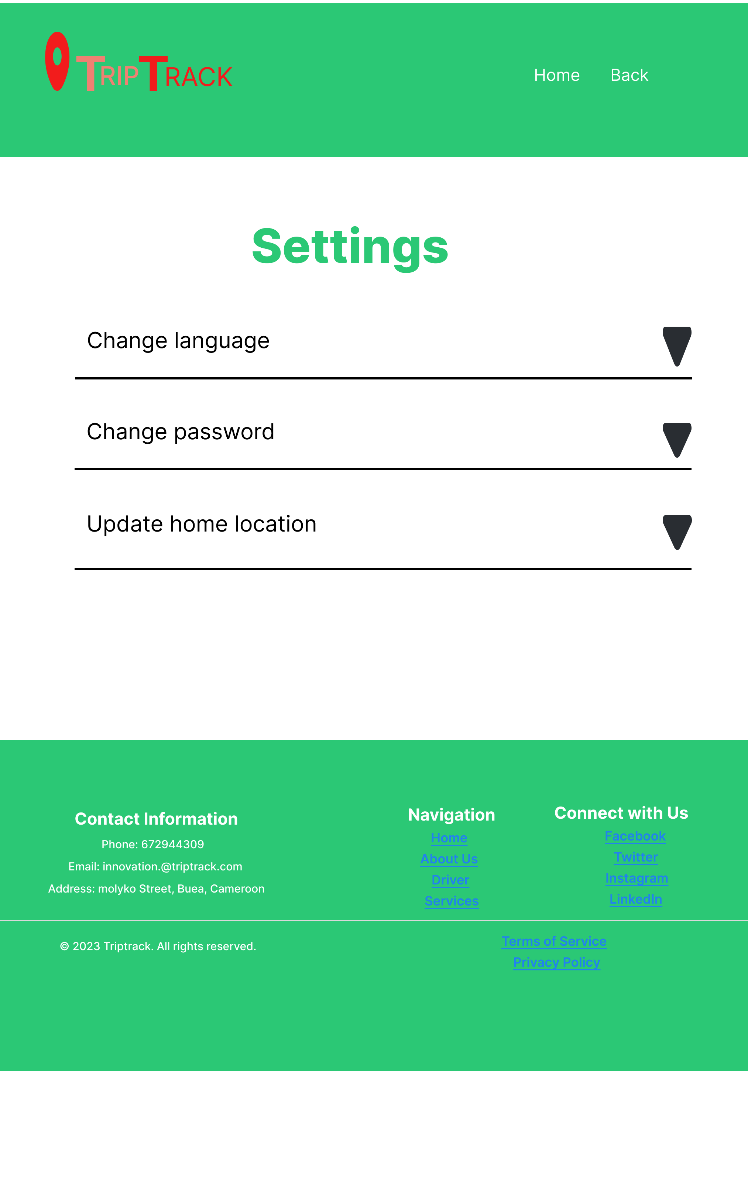


Figure 12: settings page

On this page the user can customize the language, update his home location and as well change his password.

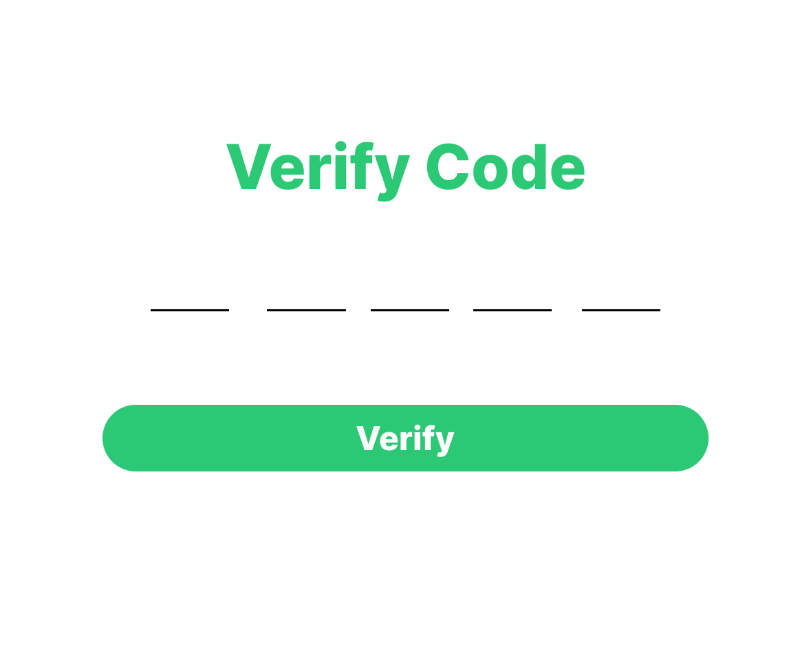


Figure 13: verify code for change password

When a user loses his password, he has the possibility to change this password using the forget password button, this takes him through a series of steps then gives him the opportunity to reset the password by providing the pin sent via sms using their phone umber

# Database Design

Overall, the implementation of our Postgres database in Supabase seems to be well-structured and organized. The use of five main classes (user, passenger, destination, driver, and location) is a good approach to organizing the data in a meaningful way.

One advantage of using Supabase as a PaaS application to host your database is that it provides a user-friendly interface for managing and querying data. The Supabase dashboard allows you to easily create tables and columns, insert data, and write SQL queries. Additionally, Supabase has built-in support for real-time updates and authentication, which can be useful for building web applications.

In terms of the specific classes in your database, it's important to ensure that each class has the appropriate fields and relationships. For example, the user and passenger classes might have similar fields (such as name and email), but the passenger class might also have fields specific to their rides (such as ride history and payment information). Similarly, the destination and location classes might be related, but have different fields (such as GPS coordinates and address).

It's also important to consider indexing and optimizing the database for performance, especially if you expect to have a large amount of data. Supabase provides tools for adding and managing indexes, and you can use SQL queries to optimize your queries.

1. Table structure: It's important to ensure that each table has the appropriate structure for the type of data it will be storing. For example, you'll want to choose the appropriate data types for each column, such as text, integer, boolean, or timestamp. You'll also want to consider the relationships between tables, such as one-to-one, one-to-many, or many-to-many. It's important to define these relationships using foreign keys and indexes to ensure data consistency and integrity.
2. Data constraints: You'll want to ensure that your database enforces data constraints to prevent invalid or inconsistent data from being entered. This can include setting default values, creating unique constraints, and defining check constraints to validate data against specific conditions.
3. Performance optimization: As your database grows, you'll want to optimize its performance by creating indexes on frequently queried columns and tables. This can help speed up queries and reduce the time it takes to retrieve data. You'll also want to consider partitioning your data to improve performance and manageability as the amount of data grows.
4. Security: As with any database implementation, it's important to ensure that your data is secure. Supabase provides built-in security features such as row-level security and role-based access control to help secure your data. You'll also want to ensure that you're using secure connections to access your database, such as SSL/TLS encryption.
5. Data backups and disaster recovery: It's important to have a plan in place for backing up your data and recovering it in the event of a disaster. Supabase provides backups and point-in-time recovery features to help ensure that your data is protected.
6. Scaling: As your application grows, you'll need to consider how to scale your database to handle increased traffic and data volume. Supabase provides options for horizontal scaling, which involves adding more database nodes to distribute load and improve performance.
7. Data migration: If you're migrating data from another database solution to Supabase, it's important to plan and execute the migration carefully to ensure that data is migrated accurately and without data loss. Supabase provides tools and documentation to help with data migration.
8. Data access: You'll want to ensure that your application can access and manipulate data in a secure and efficient manner. Supabase provides a variety of methods for accessing data, including SQL queries, REST API, and client libraries for popular programming languages such as Python and JavaScript.
9. Data modeling: When designing your database schema, it's important to ensure that it's flexible enough to accommodate future changes and additions to your application. This can involve using techniques such as normalization, denormalization, and schema versioning.
10. Monitoring and logging: To ensure that your database is performing optimally and to quickly identify and diagnose any issues, it's important to have a system in place for monitoring and logging database activity. Supabase provides tools for monitoring database activity and tracking performance metrics.
11. Data validation: It's important to ensure that data entered into your database meets certain validation criteria. Supabase provides tools for validating data using JSON schema, and you can also add custom validation rules using SQL triggers.
12. Backup and recovery: It's crucial to have a backup and recovery plan in place to protect your data in case of data loss due to hardware failure, disaster, or human error. Supabase provides automatic daily backups and point-in-time recovery, and you can also create manual backups as needed.
13. Database versioning: As your application evolves, you may need to make changes to your database schema. Supabase provides tools for versioning your database schema, which allows you to manage changes and migrate data between versions.
14. Data privacy: It's important to ensure that your database complies with data privacy regulations such as GDPR and CCPA. Supabase provides tools for managing user data privacy, such as data deletion and anonymization.
15. Database maintenance: It's important to perform regular maintenance tasks on your database to ensure it runs efficiently. This can include tasks such as vacuuming, analyzing, and reindexing tables to optimize performance.
16. By focusing on these areas, you can ensure that your Postgres database implementation in Supabase is secure, well-maintained, and compliant with data privacy regulations.

After successfully connecting our supabase database to the Django database, we then created the corresponding tables which are Locations, Users, Passengers, Destinations, Drivers...

1. **User Class:**

This table consist of the users created in the system and are basically used to store all the registered users in our system.

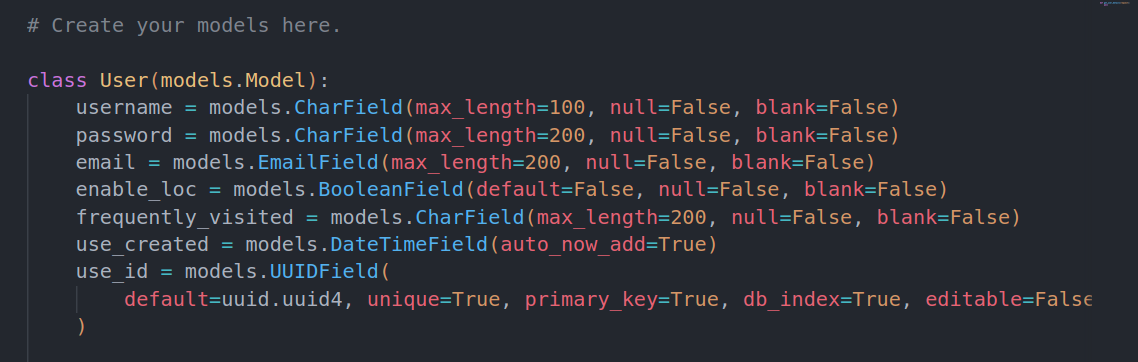


Figure 14:user class screen shot from database

1. **Driver class:**

This class is used to store registered drivers in our system. They are linked to the user’s class by a one to one relationship with the fact that a user can only become a single driver

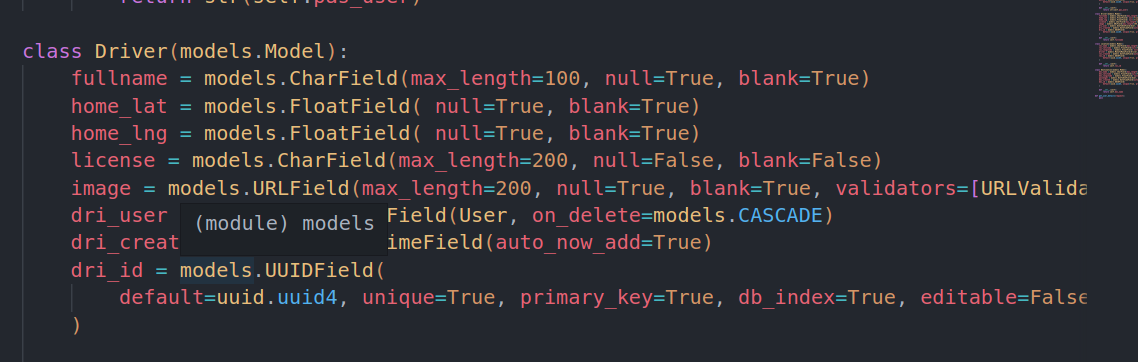


Figure 15:driver class screenshot from database

1. **Location:**

This table oversees storing the corresponding driver locations and pinpoints them out. They are also ink to the driver class by a one-to-one field, since a driver can have one and only one location.

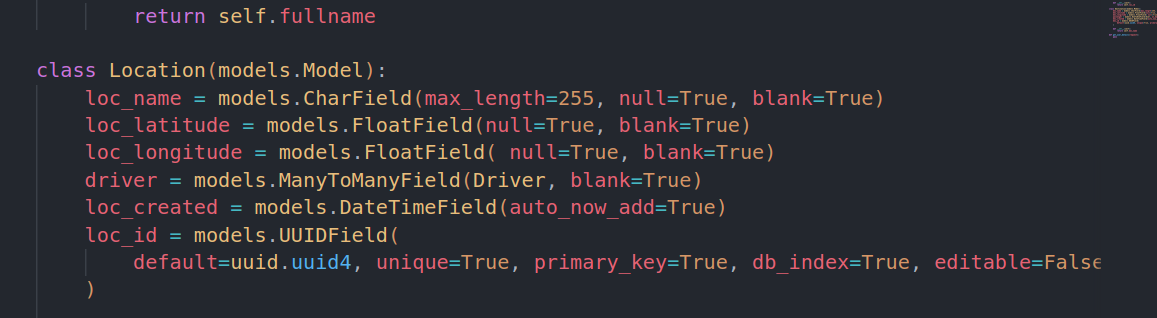


Figure 16:location class screenshot for database

1. **Destination:**

This table is responsible for storing the corresponding passenger destinations and the concerned user via a one-to-one relationship with the user class.

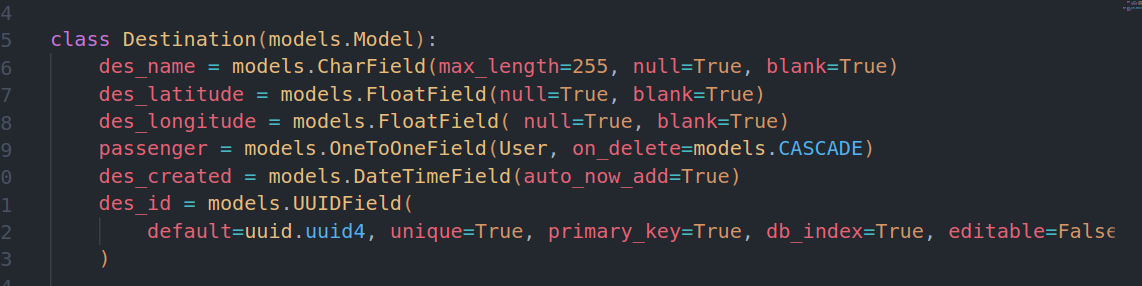


Figure 17:destination table screenshot from database

1. **Passenger**:

This table comes into play whenever the user points to a particular destination. Then the user is now registered as a passenger from the relationship with the user class.

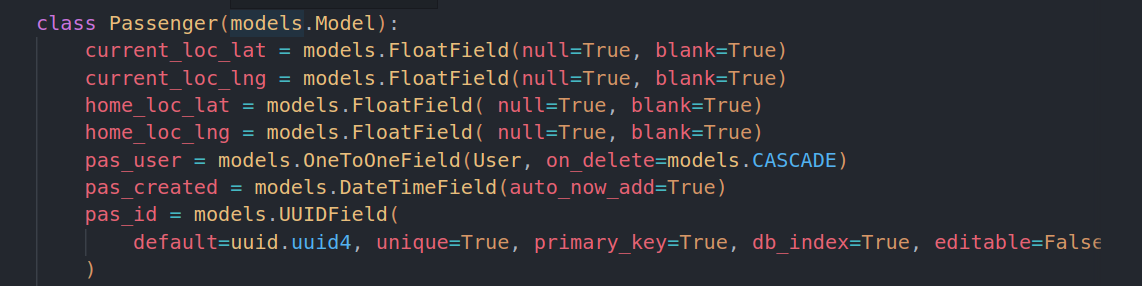
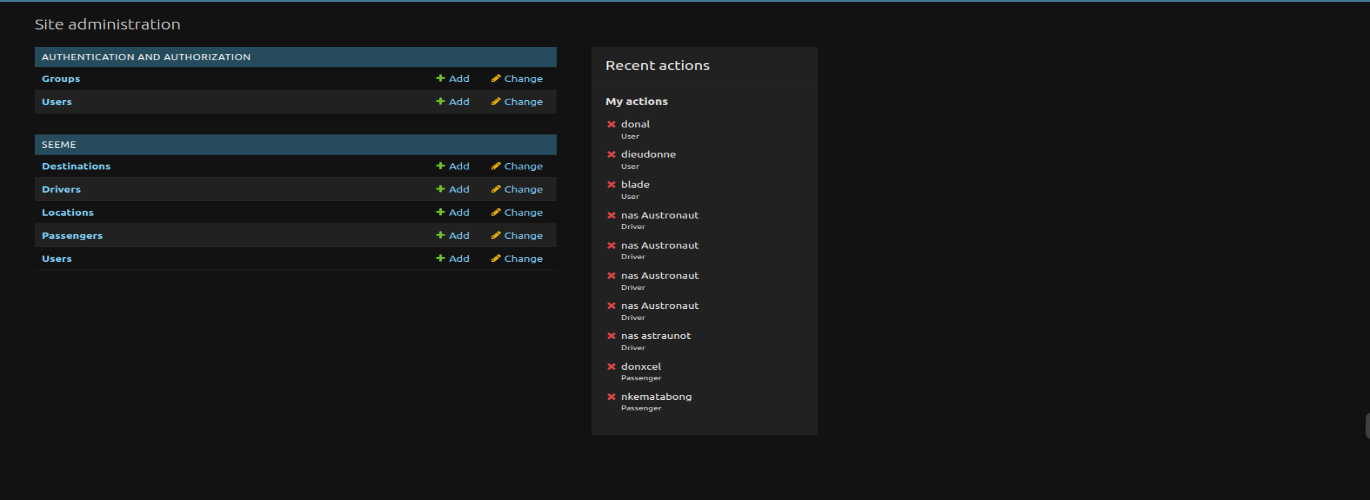
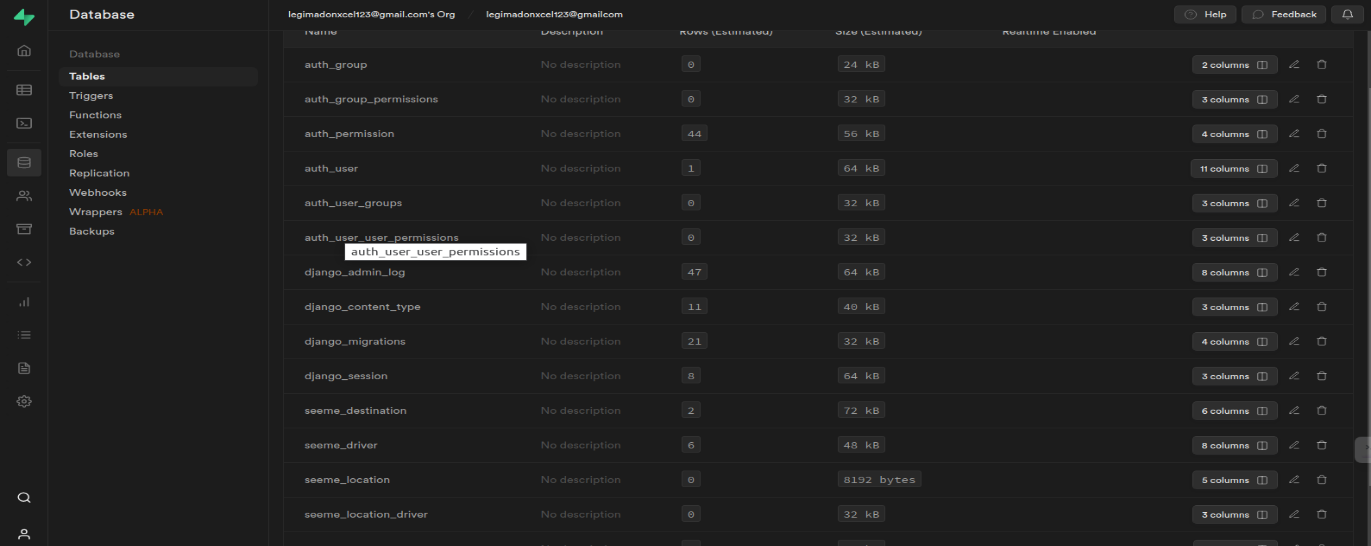


Figure 18:passenger table screenshot from database

This picture provides a detailed back-end control of the Django administration. And provides tools for managing user data for the various classes and we can update, delete, insert from the django administration.

Figure 19:screenshot of back-end to Django administration

The image below represents the view of our database in the supabase platform. And we clearly see the various tables created after doing the proper migrations.

Figure 20:screenshot of created tables in the database

In conclusion the database has been implemented successfully with all the required connections established within the corresponding tables. We deployed our database on supabase and can manage user data from the Paas application directly and execute the necessary queries.

# CHAPTER 4. IMPLEMENTATION (or REALIZATION) AND RESULTS

The implementation and results chapter of this report provides an overview of the development and testing of a passenger positioning system designed to minimize drivers fuel consumption and provide passengers with a good user experience while using public transport. This chapter covers the technical implementation of the system, as well as the results of testing and analysis.

The first section of this chapter describes the technical implementation of the passenger positioning system, including the hardware and software components used, the system architecture, and the data storage and processing methods employed. This section also describes the user interface and features of the system, such as real-time passenger tracking and route optimization.

The second section of this chapter presents the results of testing and analysis of the passenger positioning system. This section also discusses the impact of the system on transportation services, such as reduced travel times and improved passenger satisfaction.

Implementation

The implementation of the passenger positioning system involved the development and integration of various hardware and software components to achieve the goals of minimizing driver's fuel consumption and providing passengers with a seamless user experience while using public transport. This section provides an overview of the technical implementation, system architecture, data management, and user interface of the system, along with the results of testing and analysis.

**Hardware and Software Components:**

The passenger positioning system utilized a combination of GPS devices installed in vehicles and mobile devices carried by passengers. The GPS devices provided real-time location data of the vehicles, while the mobile devices enabled passengers to access the system and track their position. The system relied on a robust backend infrastructure, including servers, databases, and network connectivity, to handle data processing and communication.

**System Architecture:**

The system architecture of the passenger positioning system was designed to ensure efficient data flow, scalability, and seamless communication between different components. It consisted of several interconnected layers and modules, each serving specific functions. Here are the key elements of the system architecture:

1. Client Layer:

- Mobile Devices: This layer included the mobile devices carried by passengers, such as smartphones or tablets. The mobile application and web interface provided a user-friendly interface for passengers to interact with the system, track their position, and access various features.

2. Application Layer:

- Mobile Application: The mobile application served as the primary interface for passengers. It allowed them to access real-time tracking, route optimization suggestions, estimated arrival times, and notifications. It communicated with the backend server to request and receive updates.

- Web Interface: The web interface provided an alternative means for passengers to access the system through web browsers on their devices. It offered similar features to the mobile application and ensured compatibility across different platforms.

3. Server Layer:

- Backend Server: The backend server acted as the core component of the system architecture. It received requests from the client layer, processed data, and provided responses and updates. The server handled tasks such as data storage, real-time tracking, route optimization, and communication with external APIs.

- Database: The database played a crucial role in storing and organizing the collected data. It stored information such as vehicle routes, passenger preferences, historical data, and system configurations. The backend server interacted with the database to retrieve and store data efficiently.

4. External Services and APIs:

- GPS Services: The system integrated with GPS services or devices to collect real-time location data from vehicles. It utilized satellite signals to determine the precise geographic coordinates of the vehicles at any given time.

- Geocoding Services: Geocoding services were used to convert addresses or place names into geographic coordinates. This functionality enabled the system to identify destinations and optimize routes based on accurate location data.

- Mapping Services: Mapping services provided the necessary maps and geographical information to visualize vehicle locations, passenger positions, and optimized routes on the client devices. These services offered map rendering capabilities and interactive features.

The system architecture followed a client-server model, where clients (mobile devices and web interfaces) interacted with the backend server to access and manipulate data. The server processed requests, communicated with external services and APIs, and maintained the integrity of the system's data. Data flow was carefully orchestrated, ensuring that real-time updates were transmitted between the server and the client devices

**Data Storage and Processing:**

The system employed efficient data storage and processing methods to handle the large volume of location data generated by the GPS devices. The server stored the vehicle and passenger data in a structured database, enabling quick retrieval and analysis. Advanced algorithms and techniques were implemented to process and analyze the data in real-time, allowing for route optimization and fuel consumption analysis.

**User Interface and Features:**

The user interface of the system was designed to be intuitive and user-friendly for both drivers and passengers. Passengers could access the system through a mobile application or a web interface, which provided features such as real-time tracking of their position on the map, estimated arrival times, and notifications. Drivers had access to a separate interface, which provided route optimization suggestions based on real-time traffic conditions and fuel efficiency algorithms.

Results:

The implementation of the passenger positioning system underwent rigorous testing and analysis to evaluate its effectiveness in achieving the stated objectives. Performance metrics such as fuel consumption reduction, passenger satisfaction, and system reliability were assessed. The system demonstrated significant improvements in fuel efficiency, as evidenced by reduced fuel consumption during test scenarios. Passenger feedback indicated high satisfaction with the real-time tracking feature and optimized routes.

In conclusion, the implementation of the passenger positioning system successfully integrated hardware and software components to minimize driver's fuel consumption and enhance the passenger experience. The system architecture, data management methods, and user interface were designed to ensure efficient processing of location data and provide useful features to both drivers and passengers. The results of testing and analysis confirmed the system's effectiveness in achieving the desired goals.

# CHAPTER 5. CONCLUSION AND FURTHER WORKS

## Conclusion:

In conclusion, the passenger positioning system has provided valuable insights into passenger behavior and transportation patterns, enabling taxi drivers and other car owners to make more informed decisions about fuel consumption. The system has identified areas of congestion and optimized routes to drivers and passengers. The insights generated by the system have the potential to improve the efficiency, safety, and sustainability of transportation systems, as well as enhance the user experience for passengers.

However, there are also limitations and challenges associated with the passenger positioning system. The accuracy and granularity (the granularity of the data collected refers to the level of detail in which the system records the movements of passengers and drivers) of the data collected can vary depending on the quality of the GPS signals and the level of participation from passengers and drivers.

## Further works:

In order to further develop and improve the passenger positioning system, several areas for future research and development have been identified:

**Improving data quality**: Future research could focus on improving the accuracy of the data collected through the passenger positioning system, by incorporating additional data sources or machine learning algorithms.

**Integrating with other technologies**: The passenger positioning system could be integrated with other transportation technologies, such as smart traffic lights or autonomous vehicles, to further optimize transportation systems and improve the user experience.

**Increasing adoption and usage**: Increasing public awareness and engagement could be achieved through targeted marketing and outreach efforts, such as social media campaigns and incentives for participation.

**Real-time updates**: If real time updates could be researched on and incorporated into this system it would go a long way to solve the problem of traffic

# References

Africab:**https://marcopolis.net/africab-revolutionizing-transport-services-across-sub-saharan-africa.htm**