**REPUBLIC OF CAMERRON**

PEACE-WORK-FATHERLAND

**UNIVERSITY OF BUEA**



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**

DESIGN AND implementation of a passenger positioning system

**By:**

EMADE ROSINE NSAH........................................................,..FE20A032

ETUGE GIDEON.......................................................................FE20A036

NOUGHUE LEMOUPA FRANK..............................................FE20A092

TAJOUEGO DJUIDJA ESTRELA............................................FE20A107

TAMAH JUSTENE....................................................................FE20A108

**Course Instructor**:

Dr Nkemeni Valery

University of Buea

**2022/2023 Academic Year**

# DEDICATION

This project is dedicated to our families and friends for their guidance and supportthroughout the designand development of this project

# ACHKNOWLEDGEMENTS

We would like to thank God Almighty for the strength and inspiration he gave us to realize this project. We feel blessed with the knowledge and skills we acquired during the development of this project

We express our gratitude to our course instructor DR.Nkemeni Valery for the effort and time he gave us to accomplish our task. It has been a pleasure to work with such a comprehensive lecturer who motivates us to give our inner best

And to our brothers,sisters and friends for their unconditional support and encouragements

# ABSTRACT

A taxi positioning application is a mobile application that can solve pickup problem as well as time saving problems for the drivers and the passengers. JETME, a taxi passenger positioning application that utilizes GPS technology to provide real-time information on the location of taxi passengers. The application aims to improve the efficiency of taxi services by allowing taxi drivers to easily locate and pick up passengers, reducing wait times, reduce fuel usage by drivers and increasing customer satisfaction. The system uses a combination of GPS tracking and data analytics to predict passenger demand in different areas, allowing taxi companies and taxi drivers to optimize their resources and reduce empty trips. The application is designed to be user-friendly, with a simple interface that allows passengers to easily request a ride. The proposed system has the potential to revolutionize the taxi industry by improving the quality of service and reducing costs for both passengers and taxi companies and day-to-day drivers.

# List of Figures

Table of Contents

[DEDICATION i](#_Toc137853809)

[ACHKNOWLEDGEMENTS ii](#_Toc137853810)

[ABSTRACT iii](#_Toc137853811)

[List of Figures iv](#_Toc137853812)

[CHAPTER 1. GENERAL INTRODUCTION 1](#_Toc137853813)

[BACKGROUND 1](#_Toc137853814)

[PROBLEM STATEMENT 1](#_Toc137853815)

[OBJECTIVE 1](#_Toc137853816)

[**General objective** 1](#_Toc137853817)

[**Specific objective** 1](#_Toc137853818)

[PROPSED METHODOLOGY 2](#_Toc137853819)

[CHAPTER 2. LITERATURE REVIEW 3](#_Toc137853820)

[Introduction 3](#_Toc137853821)

[General Concepts 3](#_Toc137853822)

[Related Works 4](#_Toc137853823)

[Partial Conclusion 4](#_Toc137853824)

[CHAPTER 3. ANALYSIS AND DESIGN 5](#_Toc137853825)

[Introduction 5](#_Toc137853826)

[Proposed Methodology 5](#_Toc137853827)

[SYSTEM ANALYSIS 5](#_Toc137853828)

[REQUIREMENTS ENGINEERING 5](#_Toc137853829)

[ Software requirements validation 7](#_Toc137853830)

[**SYSTEM REQUIREMENTS** 7](#_Toc137853831)

[Functional Requirements 7](#_Toc137853832)

[Non-functional requirements 8](#_Toc137853833)

[**DOMAIN REQUIREMENTS** 8](#_Toc137853834)

[SYSTEM DESIGN 9](#_Toc137853835)

[ **Dataflow diagram** 10](#_Toc137853836)

[ **Use case diagram** 10](#_Toc137853837)

[ **Sequence diagram** 12](#_Toc137853838)

[ **Class diagram** 16](#_Toc137853839)

[ Activity diagram 17](#_Toc137853840)

[**UI/UX DESIGN** 19](#_Toc137853841)

[Global Architecture of the JETME Application 23](#_Toc137853842)

[**DATABASE DESIGN OF JETME** 24](#_Toc137853843)

[Figure 11: Entity Relationship diagram 26](#_Toc137853844)

[Description of the resolution process 26](#_Toc137853845)

[Partial conclusion 26](#_Toc137853846)

[CHAPTER 4. IMPLEMENTATION AND RESULTS 27](#_Toc137853847)

[Introduction 27](#_Toc137853848)

[Tools and Materials 27](#_Toc137853849)

[Description of the implementation process 27](#_Toc137853850)

[Presentation and interpretation of results 28](#_Toc137853851)

[Evaluation of the solution 28](#_Toc137853852)

[Partial conclusion 28](#_Toc137853853)

[CHAPTER 5. CONCLUSION AND FURTHER WORKS 30](#_Toc137853854)

[Summary of findings 30](#_Toc137853855)

[Contribution to engineering and technology 30](#_Toc137853856)

[Recommendations 31](#_Toc137853857)

[Difficulties encountered 32](#_Toc137853858)

[Further works 32](#_Toc137853859)

[REFERENCES 34](#_Toc137853860)

[APPENDICES 35](#_Toc137853861)

# CHAPTER 1. GENERAL INTRODUCTION

## BACKGROUND

Nowadays, there is a need for security reliability and efficient transport. With the advancement of technology, the taxi industry has witnessed a significant change in recent times. Earlier, passengers had to hail the cab on the street to get ride, but the evolution of ride-sharing applications like Uber and others, passengers can now book a ride through mobile applications. However, even with the ride-sharing applications, the pickup process is often quiet cumbersome. Passengers have to search or the car in crowded areas, and drivers have to navigate through traffic, causing delays and frustration for both parties. This is where the jetme application comes in to play. In built with the GPS tracking technology this application satisfies both taxi industries and day-to-day drivers.

## PROBLEM STATEMENT

The traditional way of hailing taxi involves standing at the road side and waving at passing taxis which can be time consuming, unreliable and frustrating for passengers. Taxi companies also face challenges in dispatching their fleet of taxi to meet the demand of passengers. Freelance taxis as well roam streets with no available passengers for their route. This results in longer wait time for passengers and increased operational cost for the taxi companies and freelance drivers

## OBJECTIVE

### **General objective**

The primary objective of this application is to provide a hassle-free, user-friendly and efficient taxi service experience to passengers. This application aims to reduce the pickup time for drivers in finding the passengers location and make the passengers journey more comfortable by elimination the need for search for the car. The application also hopes to increase the efficiency of the taxi industry by reducing empty trips. Overall, this application is to provide a better travel experience for passengers and bring more transparency to the ride-sharing industry

### **Specific objective**

The main objectives targeted in this application are to create an application that:

* reduce the pickup time for drivers, making the ride-sharing process more efficient and convenient
* improve the communication between passengers and drivers to ensure a safe and comfortable journey for passengers
* will increase the efficiency of the taxi industry by reducing emptytrips and analyzing passenger and driver data to optimize routes

## PROPSED METHODOLOGY

The passenger positioning application will incorporate the following methodology to achieve the application objectives

**Mobile application development:**

The development of the mobile application for android platforms using GPS technology to pinpoint the passenger’s location and track the driver’s location

**Real-time communication:**

The application will provide a real-time communication platform to allow passengers to communicate effectively with the drivers

**Software Development method**

The method used to produce a smooth and error free application was the agile method together to create a cohesive and effective user experience focusing on creating an intuitive and visually appealing interfaces

# CHAPTER 2. LITERATURE REVIEW

## Introduction

The advancement of technology has significantly impacted the transportation industry with the emergence of passenger positioning systems playing a crucial role in enhancing travel experience. This section aims to explore the existing research and literature surrounding passenger positioning systems, their applications, benefits and challenges. By reviewing the current state of knowledge in this field, we can gain insight into the potential of these systems and identify areas for further research and development during the development of the JETME application

## General Concepts

A passenger positioning system is a technology that leverages GPS tracking to connect passengers to their available drivers in real-time. The system matches passengers with nearby taxis based on their location and availability, providing a more efficient and reliable way to hail mobile applications for passengers and dispatch software for taxi companies and freelance drivers

Numerous studies have highlighted the positive impact of passenger positioning system on travel efficiency and convenience. This systemenables users to access accurate and up-to-date information about transport schedules,routes and available services. By providing real0time updates, passengers can make informed decisions, reduce waiting time and optimize tier travel routes

Passenger positioning system contribute to enhancing safety and security in transportation. Through the integration of location tracking and monitoring, these systems enable efficient emergency response and provide means of tracking passenger positions. In case of emergencies or incidents, authorities can quickly locate and assist passenger’s ensuring their well-being

## Related Works

Several passenger positioning system have been developed and implemented in different parts of the world. For example, Uber and Lyft are popular hailing applications that use GPS technology to match passengers with nearby drivers. Cities like Douala and Yaoundé have implemented taxi dispatch systems that uses GPS tracking to optimize routes and reduces wait times for passengers. These systems have shown to be effective in improving the efficiency of taxi services and enhancing the overall passenger experience

## Partial Conclusion

The passenger positioning system has the potential to transform the way people navigate and experience transportation. This system has highlighted the benefits of this system in terms of a travel efficiency, safety, personalization and convenience. However, challenges related to privacy, technical issues and user acceptance need to be fully addresses to realize their potential. Future research should focus on advancing the technology, addressing these challenges and exploring innovative applications to further enhance this system

# CHAPTER 3. ANALYSIS AND DESIGN

## Introduction

The traditional method of hailing a taxi has been replaced by technology in recent years. The use of mobile applications that allow passengers to book a ride from the comfort of their homes has revolutionized the taxi industry. However, there is still room for improvement in terms of efficiency and reliability. This paper proposes a solution in the form of a taxi passenger positioning system that uses GPS tracking to connect passengers with nearby taxis in real-time

## Proposed Methodology

As ealier mentioned, the agile method was used throughout the application development phase of this application and most especially in the design phase

**Why use the agile method?**

The agile method is a project management approach that is particularly well-suited to UI/UX design and implementation as it emphasizes flexibility, collaboration, and iterative development. Applying this method to the development of a taxi passenger positioning system, it would involve working closely with stakeholders, including taxi companies and passengers, to identify their needs and preferences.

## SYSTEM ANALYSIS

### REQUIREMENTS ENGINEERING

The process to gather the information from clients and analyze them is known as requirement engineering. The goal of requirement engineering is to develop and maintain sophisticated and descriptive software requirement specifications requirement engineering process. These processes are as follows:

* **Feasibility studies**

This involves getting the ideology of the application from the client. That is, questioning the client to get out the information about the basic functionalities of the system that he requires. These functionalities are then analyzed by the analyst to verify if the idea can be **SMART** that is the idea is to be **S**pecific, **M**easurable, **A**chievable, **R**elevant and **T**ime bound.

Some of the features requested by the client can be viewed below:

* GPS Tracking: The system should be integrated with GPS (Global Positioning System) to track the real-time location of the vehicle.
* Passenger console: The system should have a passenger console for passengers to book their rides, track the location of their ride, and communicate with the driver.
* Driver Console: The system should have a driver console to enable drivers to accept passenger requests, view their trip details, and navigate to their destinations.
* Routing and Navigation: The system should have routing and navigation functionality to offer optimized routes for drivers, reducing travel time and improving vehicle utilization.
* **Requirements gathering**

This involves two things which are the gathering of functional requirements from the client and brainstorming using various engineering techniques. A series of surveys was done to get what is expected of the application from the user and a large number of functionalities were requested by the users.

* **Design and implementation constraints**

The design and implementation of the passenger positioning system face several constraints that can affect its functionality, usability, and effectiveness. The following are some of the design and implementation constraints of the passenger positioning system:

**Technical Constraints:** The technical constraints include the selection of a suitable platform for the system, software and hardware limitations, and network connectivity. The system's performance and functionality can be affected by constraints such as processing power, memory, and storage capacity.

**Scalability constraints:** The road passenger positioning system needs to be scalable to accommodate an increasing number of passengers and drivers. Scalability constraints such as database management, data storage, and server capacity can affect the system's ability to accommodate large volumes of data.

**Security Constraints:** As the system deals with sensitive and personal data such as passenger details, driver information, and location tracking data, security constraints are significant. Constraints such as data encryption, firewalls, and access controls must be considered during the system design and implementation.

**User Interface Constraints:** The user interface design must be intuitive, user-friendly, and accessible across different devices, operating systems, and screens.

**Legal Constraints:** Legal constraints such as data protection laws and privacy regulations, licensing, and intellectual property rights must be adhered to.

## Software requirements validation

After a thorough investigation of the requirements, they were processed and found to be legal, practical and well interpreted following a set of requirement elicitation techniques. These techniques include:

* Task analysis
* Domain analysis
* Brainstorming
* observations

### **SYSTEM REQUIREMENTS**

## Functional Requirements

* Passengers should be able to specify their locations and destinations in the app
* Drivers should be able to view the location of passengers in the app
* The app should provide direction to easiest path in reaching the passengers
* The app should be able to determine the most profitable point for the drivers
* The application should get the drivers location and be able to approximate his arrival time
* Future Passenger Should be able to use the application to book a taxi
* Future Passengers Should be able to cancel a booked taxi
* Application should be able to differentiate between hired taxi and freelance taxi
* Application Should be able to communicate with a GPS tracker
* The Application should be able to run on various platforms

## Non-functional requirements

* **Performance:** The passenger positioning system must be able to perform efficiently and effectively, with minimal lag or delay in response times.
* **Availability:** The passenger positioning system must be available for use at all times, without any downtime or disruptions.
* **Maintainability**: The passenger positioning system must be easy to maintain and update, with minimal disruption to the system’s operation.
* **Compatibility:** The passenger positioning system must be compatible with different operating systems, devices, and software versions.
* **Security:** The passenger positioning system must have robust security measures in place to protect user location and data
* **Reliability**: The passenger positioning system must be reliable and consistent in its operation, without any errors or glitches.
* **Usability:** The passenger positioning system must be easy to use and navigate, with clear instructions and intuitive design.
* **Scalability**: The passenger positioning system must be able to handle increasing amounts of data and users as the system grows over time.
* **Interoperability**: The passenger positioning system must be able to integrate and communicate with other systems and devices.
* **Regulatory Compliance:** The passenger positioning system must comply with relevant regulations and standards for data privacy and security.

### **DOMAIN REQUIREMENTS**

Domain requirements reflect the environment in which the system operates, so when we talk about an application domain we mean environments for operating and building the application.

## SYSTEM DESIGN

This section describes the UML diagrams which were derived from the analysis of the system

From the analysis of the system, the drivers were split into two categories. We have the freelance and the hired drivers

**Freelance drivers:**

These are drivers who function independently of the municipal council. That is, they are the sole decision makers of their routes. The decide on their payment fees and decide when and how to operate

**Hired driver:**

These are drivers that work under the council. The taxis are owned by the council and the drivers work daily. They have no say in the price exchange as they receive their payments solely at the end of the month. These drivers can be hired to drive a particular passenger the whole day depending on the number of hours requested by the passenger

Below are some of the diagrams used to described the system in design view

### **Dataflow diagram**

This section provides the flow information from one object to another and how they inter relate within the JETME application

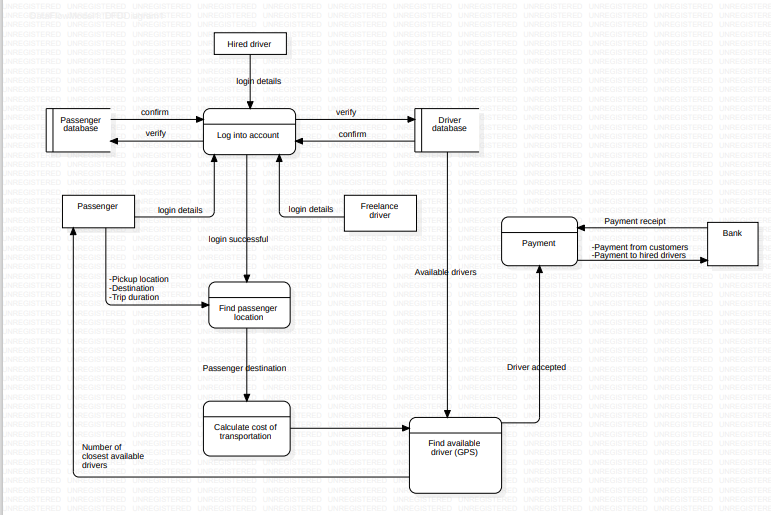


Figure 1: Dataflow Diagram

### **Use case diagram**

The use case diagram provides a high level view of the interactions between the actors which could be the users of the external systems and the systems being designed. It shows the various use cases (tasks) that the system will perform to meet the needs of the drivers and the passengers. With this diagram, we can identify the actors involved, the goals the each want to achieve and the interactions between them. With this information, we get to identify the system requirements, prioritize the development efforts and ensure that the system meets the needs of the users. Below is a sample use case diagram for the passenger positioning system

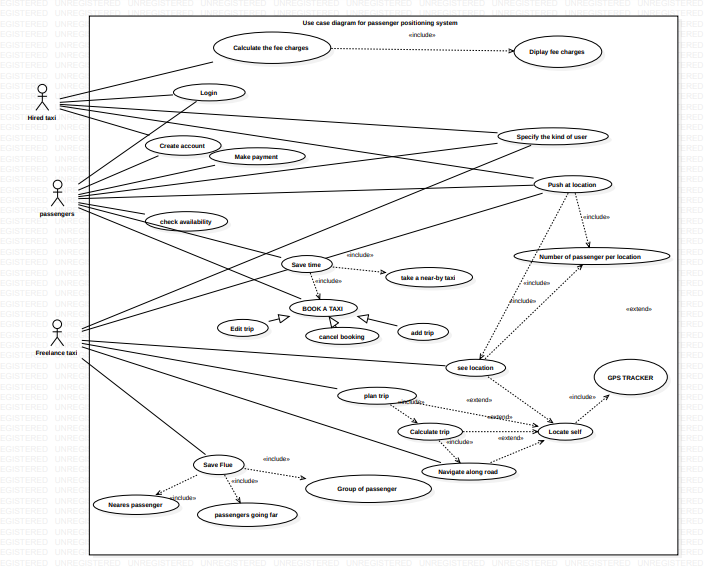


Figure 2: Use case Diagram

**Diagram description**

Actors of the system include:

* **Hired Taxi:** A Taxi employed by a municipal council or some other organization which acts only on demand

His interface is both visual and vocal permitting him to get and input sensitive data like location and Destination

**Free Lance Taxi:** A Taxi that acts by itself placing profit above all

His interface is mainly vocal, allowing to get sensitive data without being distracted.

* **Passenger:** A visual interface allowing him to accurately enter needed information to get a taxi

A detailed explanation of the different use cases involved with the system can be demonstrated in the series of sequence diagrams below

### **Sequence diagram**

These diagrams provide information about the interactions between the different objects in the system. It portrays the flow of messages between these objects in a particular scenario of a use case. With this information, we are able to understand the flow of control and data through the passenger positioning system, identify potential performance issues and validate the correctness of the systems behavior. Below are a set of diagrams that have been designed per use case

1. **Create account and login sequence diagram:**

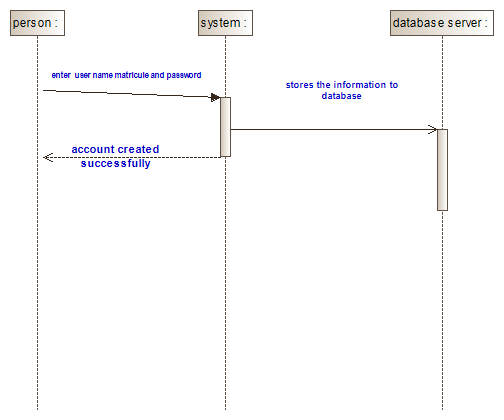


Figure 3: Create Account and login

**Diagram description**

Create account

This functionality involves just the hired taxi and the passengers. Only passengers who notify the driver that they want to hire a taxi are required to have an account for security purposes. Before they reach to create an account page, they must first specify the kind of user they are. They enter their information, and the information is checked on the server for correctness and existence. If the information is correct but does not yet exist in the database, the account will be created.  
 Login  
The login page come after an account has been created. The users involves enters their credentials and the system checks in the database for existence and if they are for consistence. If the information are corrects, the user then enter into the dashboard

1. **Upload location sequence diagram(plan trip):**

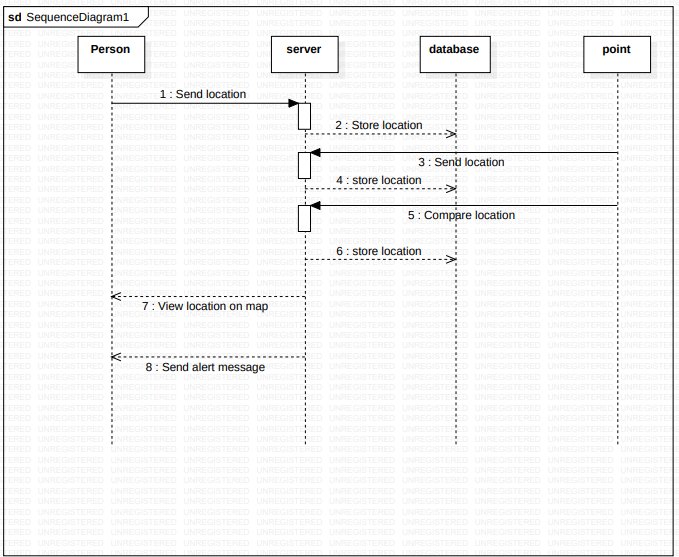


Figure 4:Upload Location

1. **Booking sequence diagram:**

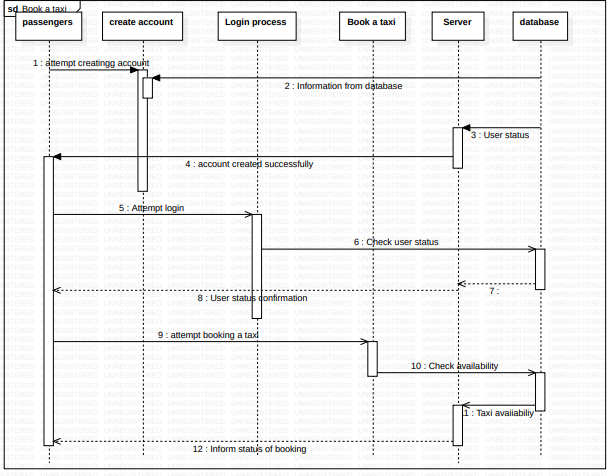


Figure 5:Book a Ride

**Diagram description:**It is used by the passenger. When the passenger enters the system, the system asks what kind of user he is. After the kind of user is specified by the person, the person enters the dash board. The passenger then sends a request to book a taxi. The system then asks the user to either create an account or login. For new users, they will create an account and then log in. For old users, they will login to their account. After logging in, they select make payment, and the system asks them the number of hours they will want the taxi to work with them. After specifying the number of hours, the system calculates the price and sends it. The user then sends a request to the server, and the server requests the card information. After input and validation, the user receipt is displayed and ready for download.

1. **Location sequence diagram**

This use case is used by all the users of the system. The person involved sends a request to the server for the location. The server then sends the information to the database. The Google API now sends the point to the server, and the server sends the information to the database. The API now compares the position and sends it back to the server. The server then stores it in the database. The server also sends the information to the person on Google Maps, and an alert message is given so the person can check.

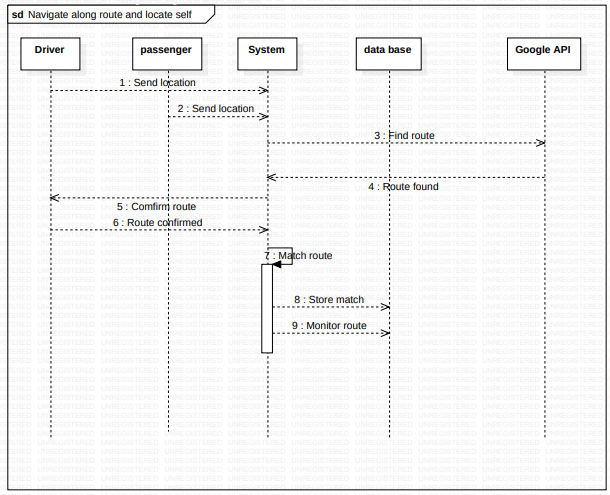


Figure 6:location sequence diagram

1. **Payment sequence diagram:**

Diagram description:

This is done by the passenger. The passenger enters and clicks on book a taxi, the system then ask for the number of hours he will want to work with the taxi. The system then calculates the amount that the passenger is supposed to pay, and the price is display to the passenger. The passenger now chooses a payment method of payment, enter his/her credentials and validate the transaction. The system after receiving the payment then gives a receipt to the passenger. The system then updates the database

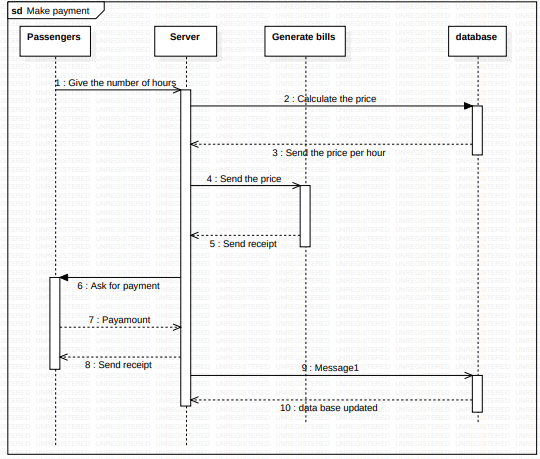


Figure 7: Payment Sequence diagram

### **Class diagram**

This is a visual representation of the classes, interfaces and relationships in the system. We are provided with the detailed view of the systems structure, including the attributes and methods of each class and the relationships between them. Below you are provided with the different objects that we use in the passenger positioning system, their properties and their behaviors which is used to design the systems architecture, define the systems data model and ensure that the system is scalable and maintainable

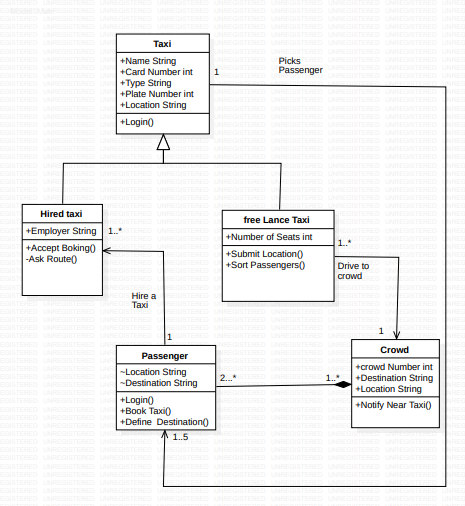


Figure 8: Class diagram

## Activity diagram

In this section, we communicate the information about the flow of activities in the system including the decisions, loops and concurrencies of the passenger positioning system. Here you get the sequence of actions and their dependencies, as well as the conditions and constraints that govern them

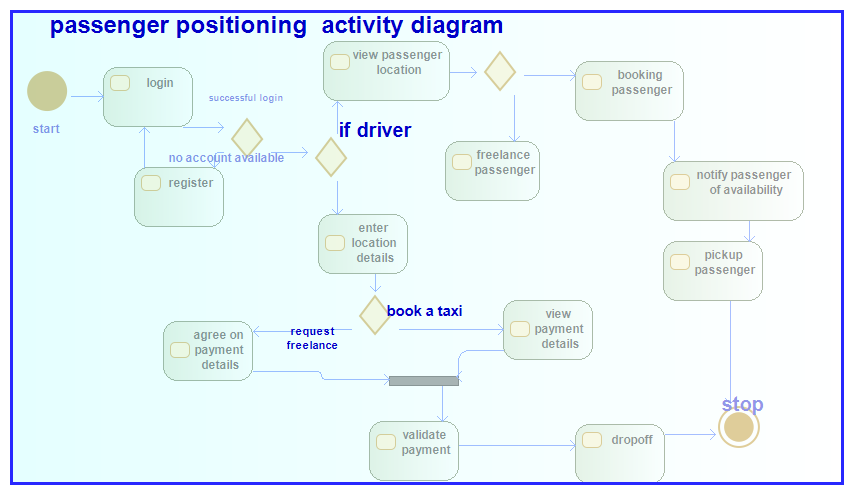


Figure 9: Activity Diagram

### **UI/UX DESIGN**

The different interface designs that were implemented with the system can be view below

1. Splash screen

This is the welcome page that is seen by the user upon launching the application

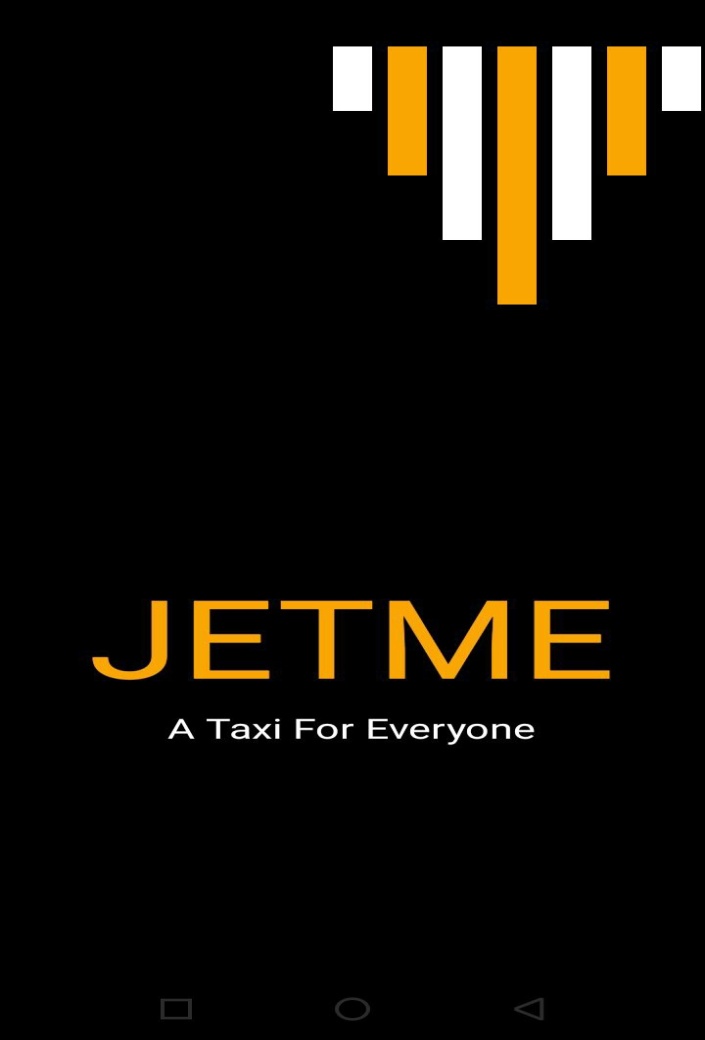
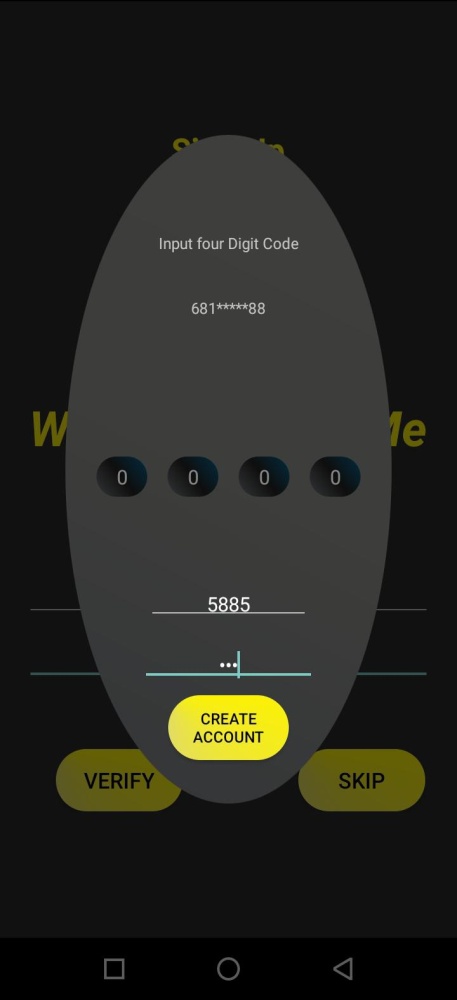
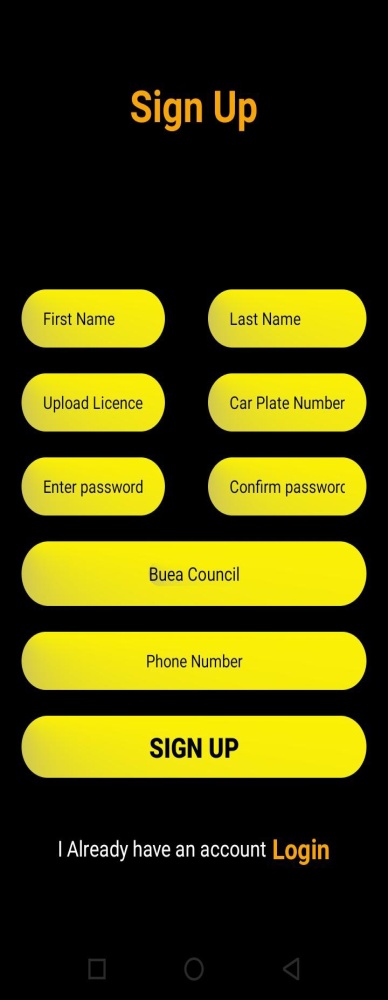


Figure 10: Splash screen

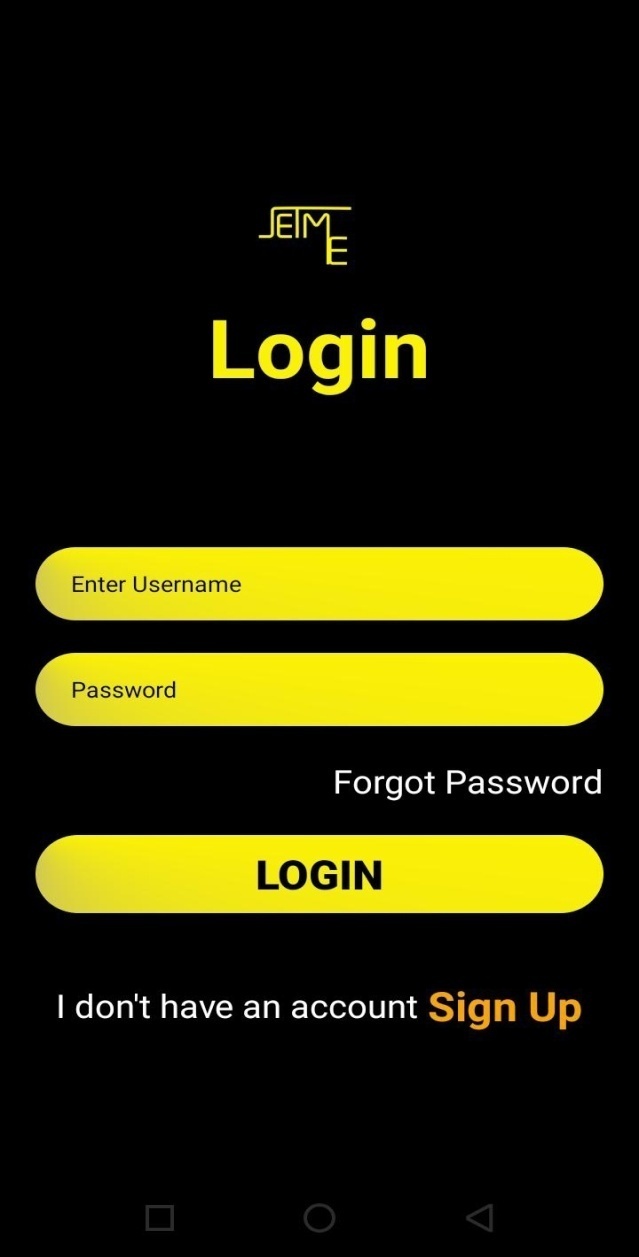
1. Sign up

This page is to get the user details to create an account. Not every user is entitled to have an account in the application. Therefore, this aspect is necessary to just the drivers and the users that will be involved in the booking process. Upon creating an account a verification page is viewed which is the finally stage to validate account creation. The driver, signup, verification and passenger signup can be seen below respectively



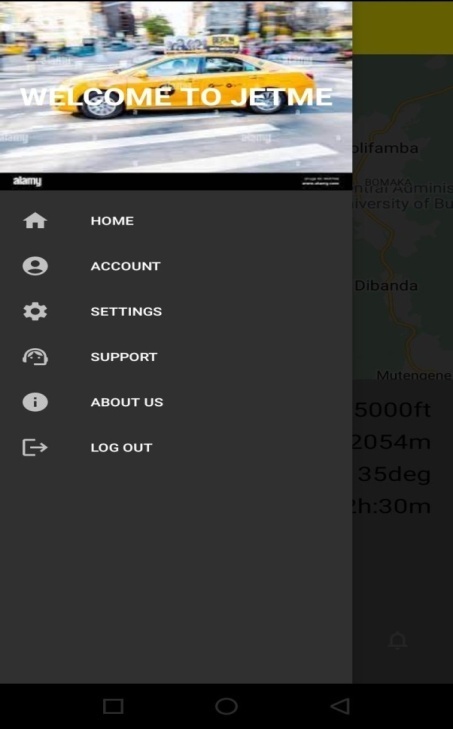
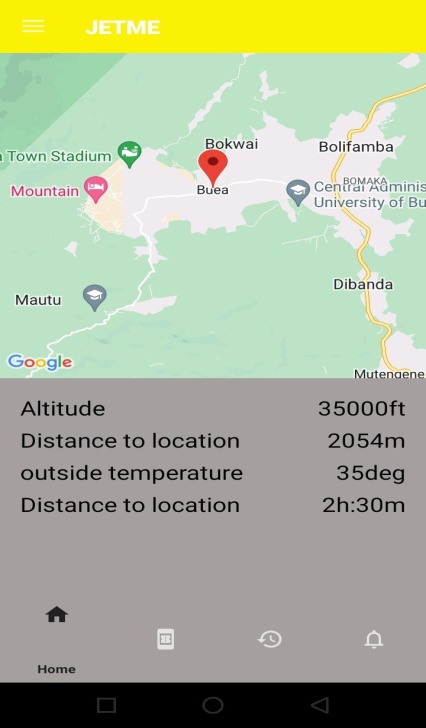
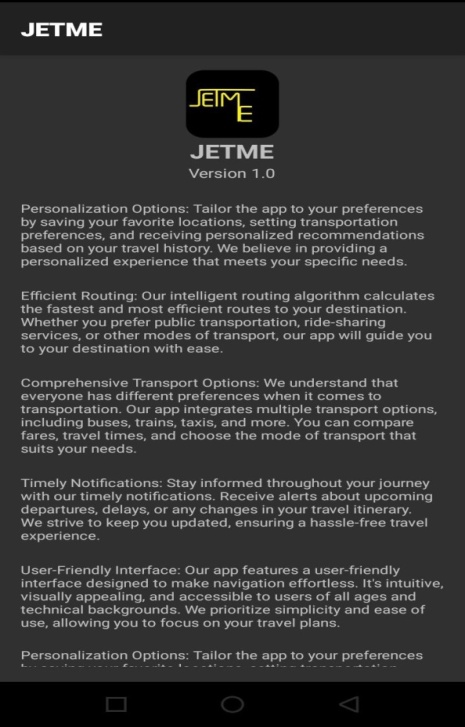
1. **Login**

This page is useful just to passengers who have accounts in the application and mandatory for all drivers registered to the system. Not every user is obliged to have one.



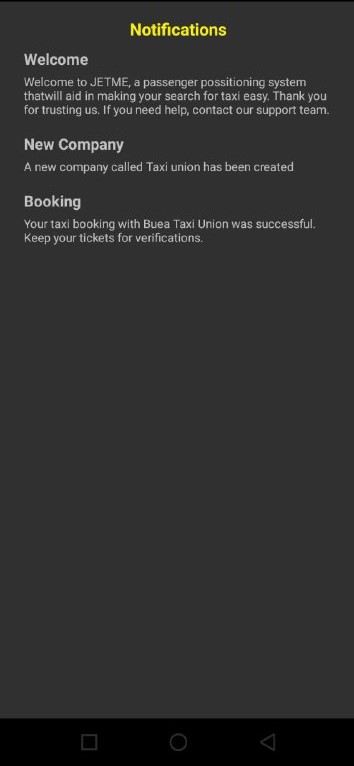
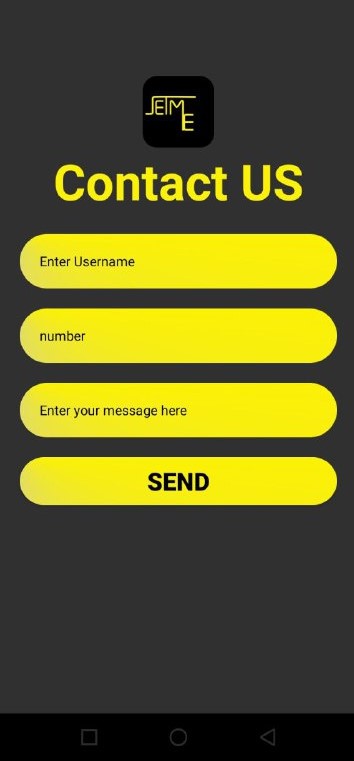
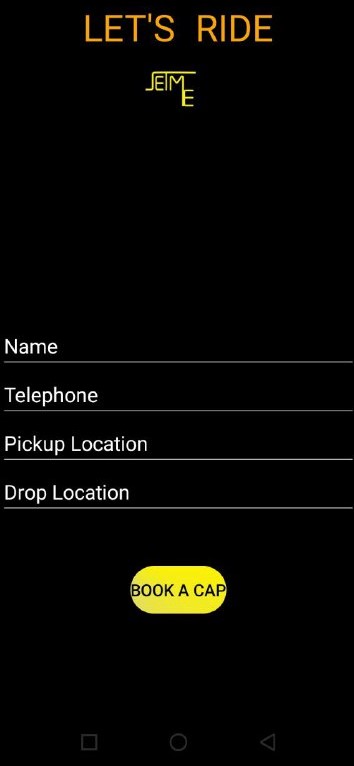
1. **Dashboard**

This page contains links to all the necessary actions that can be performed with the application. It is accessible by everyone depending on the type of user. From this page we have the link to the menu page, the notification page, book ride which is a basic functionality and the about page as well. All this links are basic functionalities of the JETME application.

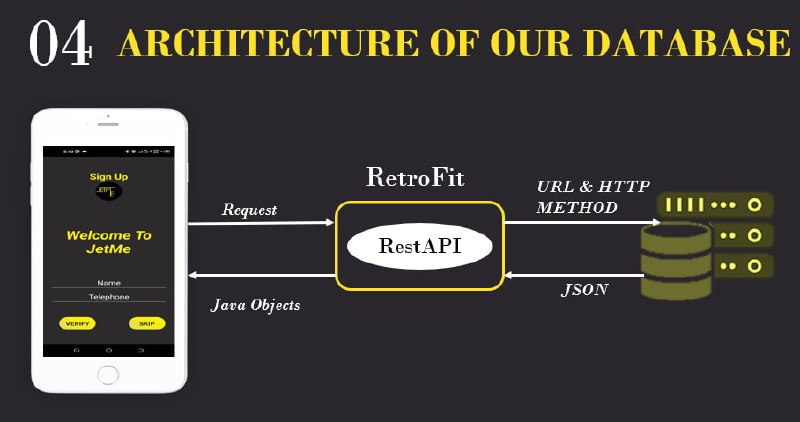
 ****

**Book ride**

This is functionality with the system that permits users willing to hire a taxi to gain access to one.



## Global Architecture of the JETME Application



The global architecture of JETME passenger positioning system typically consists of three main components:

1. **Onboard Unit (OBU)** - This is the device installed inside the taxi that communicates with the central system and gathers relevant data such as the taxi's location, speed, and status.
2. **Central Dispatching System (CDS)** - This is the nerve center of the system, which receives real-time data from the onboard unit and performs various functions such as route optimization, fare calculation, and passenger matching.
3. **Mobile Application** - This is the user-facing interface of the system, which enables passengers to request rides, track their ride in real-time, and make payments.

Finally, the mobile application provides an easy-to-use interface for passengers. They can request a ride, track the taxi on a map, and pay for the ride. The app also allows passengers to rate the driver and provide feedback.

The architecture of the database for a taxi passenger positioning system typically involves the following components:

* **Storage:** The database stores the data about passengers and taxis in a structured way, using tables and relationships between them.
* **Query processing:** The DBMS provides tools for querying and retrieving data from the database using SQL (Structured Query Language).
* **Access control:** The DBMS controls access to the data in the database, using security features such as user authentication and permission management.
* **Recovery:** The DBMS provides features for managing transaction logging and recovery, ensuring that the database can be restored to a consistent state in case of data corruption or other issues.
* **Performance tuning:** The DBMS provides tools for optimizing the performance of the database, including indexing and other optimization techniques.

### **DATABASE DESIGN OF JETME**

The design of this system is intended to provide real-time information on the location of taxis and enable passengers to book a taxi from their current location.

CONCEPTUAL DESIGNS

* Entities:

Passenger

Drivers

Admin

Free Lance Taxi

* Attributes

Passenger: Passenger\_ld, passenger\_Location, Telephone, Names, Destination, Password

Admin: Names, AdminID, AdminPassword

Free Lance Taxi: Names, Plate\_Number, Freelance\_Id, First Name, Last\_NamesFreelance\_password, Driver\_license

Hired Taxi: First Name, Last Name, Plate Number, Driver license, Employer, password, hired\_id

* Relationships
* One passenger can book only one taxi at a time
* One hired taxi can accept only one passenger at a time
* Many hire taxi can request for validation at a time
* An admin can validate only one hired taxi at a time
* An admin can validate only one free-lance taxi at a time
* Many free-lance taxi can be sent for validation at a time
* One free-lance taxi can carry many passenger at a time
* One passenger can take only one taxi at a time
* Relational schema

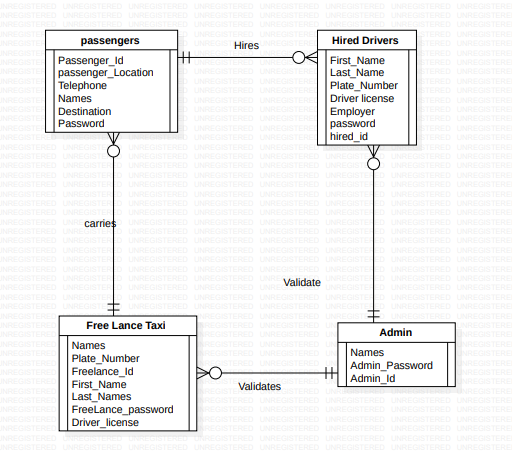
Admin( admin-Id, name, password-password)

Passenger(Passenger-ID, name, phone number, location, destination, password,\*freelance-id)

Hired drivers (hired-ID, name, phone number, driver license, plate-number, employer , Hpassword, f-name, l-name, \*Admin-id)

Freelance Taxis (freelance-id, license, firstname, lastname plate number, password, username, \*admin-id)

The above information can be illustrated on the following diagrams



### Figure 11: Entity Relationship diagram

## Description of the resolution process

## Partial conclusion

# CHAPTER 4. IMPLEMENTATION AND RESULTS

## Introduction

## Tools and Materials

These are software applications that we used to create bring forth the system to live. To ensure a user friendly system we made use of the following software tool.

* **Figma:**This tool was used to bring out the designs of passenger positioning system
* **Android Studio:** This tool was used to implement the designs that were created using the figma software
* **XAMPP:** Tool used for the database implementation

## Description of the implementation process

The network architecture of the system is that of a client-sever, this governed the entire flow of information an data within the system, the necessary Requirement engineering was done to ensure the architecture meets requirements.

1. **Server.**

Either windows or Linux with preference for the second. The two service used within the server are mysql and apache2 chosen for being secured and robust the handle HTTPS and HTTP request only from php scripts. This request is passed to the database processed and passed back to php which in turns sends if needed results to the client. A single file is used to connected the php scripts to add another layer of security the file is closed at the end of every server script this terminates the client-server communication when needed

* Services

Apache and myqsl

* Database type

Mysql

* Platform

Windows or Linux

1. **Client**

The client is a mobile based application which submit HTTPS and HTTP request to the server all the request are handle by a library “retrofit” able to support asynchronous calls to the server particularly known for its fast response time. The clients were modeled from real life systems together with their relations. The client application was built using Java thus native app this comes from the need of getting the coarse location and fine location for real time data implying access to hardware and network provider a task best done in native mode.

* Application Type

Native application

* Programming Language

Java

* Platform

Android

## Presentation and interpretation of results

## Evaluation of the solution

## Partial conclusion

# CHAPTER 5. CONCLUSION AND FURTHER WORKS

## Summary of findings

After conducting extensive analysis and development, we could build a passenger positioning system that provides real-time updates to both passengers and drivers, resulting in improved pick-up times and correct drop-off locations, thus saving fuel. For this achievement, we implemented the GPS technology and wireless communication to track passengers’ locations and relay information to drivers.

With this, passengers can have access to a user-friendly application which allows them to input their destination and track their ride in real-time. Drivers also have access to the system through a dashboard interface, which provides them with the location of all passengers, enabling them to optimize their pick-up and drop-off times. Additionally, the system helps to improve safety by providing drivers with accurate information about the location of their passengers, reducing the risk of missed or incorrect drop-off locations.

Contribution to engineering and technology

The passenger positioning system makes a significant contribution to the field of engineering and technology. It provides an innovative solution to a common problem in the transportation industry.

The system demonstrates the potential of technology to improve the efficiency and safety of transportation services. By providing real-time tracking of passengers, it enables drivers to optimize their routes and reduce the time and fuel required for each trip. This does not only benefit the drivers but also reduces the overall carbon footprint of transportation services.

The passenger positioning system has also demonstrated the potential of mobile applications to enhance the user experience in transportation services. By providing passengers with a simple and intuitive interface, it has made it easier for them to book rides and track their progress in real-time. This has helped to reduce confusion and improve overall satisfaction with transportation services.

Overall, the passenger positioning system demonstrate the potential of engineering and technology to solve complex problems and improve the quality of life for people around the world. Its success serves as an inspiration for future innovations in transportation and other industries.

## Recommendations

Some recommendations of the passenger positioning system include;

1. **User-Friendly:** The passenger positioning system should be easy to use and intuitive for both passengers and drivers. This could include a simple interface, clear instructions, and easy-to-follow navigation.

2. **High-Quality Hardware and Software:** The accuracy and reliability of the system will depend on the quality of the hardware and software used. Investing in high-quality components will ensure that the system functions properly and provides accurate information.

3. **Safety and Security:** Passenger safety should be a top priority when designing the system. This could include features such as emergency alerts, driver background checks, and real-time tracking to ensure that passengers are safe at all times.

4. **Real-Time Updates:** Passengers should be kept informed of any changes or delays in their travel plans. Providing real-time updates and notifications through the system can help reduce frustration and improve the overall passenger experience.

5. **Integration with Other Transportation Systems:** Integrating the system with other transportation systems, such as public transit or ride-sharing services, can provide a more seamless travel experience for passengers.

6. **Regular Maintenance and Updates:** Regular maintenance and updates are essential to ensure that the system remains up-to-date and functioning properly. This could include software updates, hardware replacements, and regular testing to identify any issues.

7. **Privacy Concerns:** Passengers' personal information should be protected at all times. This could include encryption of data, strict access controls, and policies around data retention and deletion.

## Difficulties encountered

* Poor connection which lead to the following:

1. Slow Download and Upload Speeds: Slow internet speeds which significantly slowed down the process of downloading or uploading large files.

2. Limited Access to Resources: Online resources such as Documentation, tutorials, and forums to help solve problems were limited due Poor internet connection can limit access making it harder to resolve issues quickly.

3. Collaboration and Communication: The project required our team to work remotely. Poor internet connection made it difficult for team members to collaborate effectively, leading to delays and miscommunication.

* Technical Challenges: Developing a system that is accurate, reliable, and scalable can be a complex technical challenge. The system must be able to handle large amounts of data, process it quickly, and provide accurate information in real-time.

## Further works

The passenger positioning system makes significant contributions to the field of transportation, but there is still room for further development and improvement. Some potential areas for future work include:

1. **Integration with other transportation systems:** The passenger positioning system could be integrated with other transportation systems, such as public transit, to provide a more seamless and efficient travel experience for passengers.

2. **Enhanced safety features:** The system could be further developed to include additional safety features, such as emergency alerts and driver monitoring, to ensure the safety of both passengers and drivers.

3. **Improved accuracy and reliability:** While the system is already highly accurate and reliable, there is always room for improvement. Future work could focus on improving the accuracy and reliability of the system even further.

4. **Expansion to new markets:** The passenger positioning system has already been widely adopted in many parts of the world, but there are still many markets where it has not yet been introduced. Future work could focus on expanding the system to new markets and regions.

# REFERENCES

[1] Android Developers. "Android Studio." Official Website. [Online]. Available: https://developer.android.com/studio. [Accessed: Day Month Year].

[2] Android Developers. "Android Documentation." Official Website. [Online]. Available: https://developer.android.com/docs. [Accessed: Day Month Year].

[3] Google Maps Platform. "Maps SDK for Android." Official Documentation. [Online]. Available: https://developers.google.com/maps/documentation/android-sdk/overview. [Accessed: Day Month Year].

[4] Retrofit. "Type-Safe HTTP Client for Android and Java." Official Website. [Online]. Available: https://square.github.io/retrofit/. [Accessed: Day Month Year].

[5] MySQL. "MySQL Documentation." Official Website. [Online]. Available: https://dev.mysql.com/doc/. [Accessed: Day Month Year].

[6] PHP. "PHP Documentation." Official Website. [Online]. Available: https://www.php.net/docs.php. [Accessed: Day Month Year].

# APPENDICES

Appendix A: Data Structures

A.1 Passenger Class

- id: int

- name: String

- phone: String

- location: String

- lastDestinations: List<String>

A.2 Driver Class

- id: int

- name: String

- phone: String

- location: String

- available: boolean

A.3 Ride Class

- id: int

- passengerId: int

- driverId: int

- pickupLocation: String

- destination: String

- status: String

Appendix B: API Endpoints

B.1 Passenger Endpoints

- GET /passengers: Retrieve a list of all passengers

- GET /passengers/{id}: Retrieve a specific passenger by ID

- POST /passengers: Create a new passenger

- PUT /passengers/{id}: Update an existing passenger

- DELETE /passengers/{id}: Delete a passenger

B.2 Driver Endpoints

- GET /drivers: Retrieve a list of all drivers

- GET /drivers/{id}: Retrieve a specific driver by ID

- POST /drivers: Create a new driver

- PUT /drivers/{id}: Update an existing driver

- DELETE /drivers/{id}: Delete a driver

B.3 Ride Endpoints

- GET /rides: Retrieve a list of all rides

- GET /rides/{id}: Retrieve a specific ride by ID

- POST /rides: Create a new ride

- PUT /rides/{id}: Update an existing ride

- DELETE /rides/{id}: Delete a ride

Appendix C: Database Schema

C.1 Passenger Table

- id: int (Primary Key)

- name: varchar(255)

- phone: varchar(20)

- location: varchar(255)

- last\_destinations: text

C.2 Driver Table

- id: int (Primary Key)

- name: varchar(255)

- phone: varchar(20)

- location: varchar(255)

- available: boolean

C.3 Ride Table

- id: int (Primary Key)

- passenger\_id: int (Foreign Key referencing Passenger Table)

- driver\_id: int (Foreign Key referencing Driver Table)

- pickup\_location: varchar(255)

- destination: varchar(255)

- status: varchar(20)

Appendix D: Additional Features

D.1 Real-Time Tracking

Implement real-time tracking of passenger and driver locations using GPS coordinates.

D.2 Payment Integration

Integrate a payment system to allow passengers to make payments for their rides securely.

D.3 Reviews and Ratings

Implement a feature for passengers to rate and review drivers after completing a ride.

D.4 Push Notifications

Enable push notifications to keep passengers informed about ride updates and driver availability.

D.5 Advanced Routing

Incorporate advanced routing algorithms to optimize the selection of drivers for passenger requests.

Appendix E: Glossary

- GPS: Global Positioning System

- API: Application Programming Interface

- Endpoint: A specific URL that an API exposes for performing actions or retrieving data.