CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Transport refers to the movement of people or goods from one location to another by numerous modes of transportation such as automobiles, trains, aircraft, and even animals such as donkeys and camels. It is classified into three categories according to the surface it travels over. Water (shipping), air, and land (road, pipeline, and train) are among the several modalities. Transportation was accessible prior to the invention of the current modes of transportation, which are conventional modes of transportation. (Adekola et al., 2021)

Traditional modes of transportation include walking and running, which use human physical force. Although some of these methods are still used for short-distance transport today, contemporary technology has significantly increased human power. Human-powered transportation is still popular for leisure and physical activity. Despite the fact that humans may travel about without any infrastructure, modern transportation is also required. (Claudia et al., 2017)

Tricycles are a common means of transportation in most developing countries. Local public transportation in the form of tricycles is available at the Kaduna Polytechnic. Tricycles provide services by carrying commuters to main roadways while using little fuel. The tricycle is renowned for its use as a gap filler. They operate primarily to cover service gaps created by buses and other forms of conventional public transportation. (Litman, 2017)

Every economy relies on transportation. It facilitates the movement of people and things and has a direct impact on service delivery. When choosing a mode of transportation, commuters evaluate safety, availability, cost, dependability, comfortability, and efficiency. Litman (2017)

Every aspect of life has been influenced by technology, including transportation, communication, education, and many more. The safe, efficient transportation of students within the school environment is a challenge faced by higher institutions across the country. The low influx of tricycles and the fixed loading point in school has made student transit unavailable, student has to either go to the loading point or stand and wait for the tricycle to go to the loading point or the student destination. (Turner, 2022)

1.2 Statement of the Problem

The unavailability of tricycles within the school is due to how the manual method is structured, in trying to introduce comfort to the student in acquiring tricycles and to that of tricycle drivers in acquiring passengers (students) some problems were found. Currently, Students must go to the loading point to get on a tricycle, they also have to line up for a long time waiting for the loaded tricycle to go and come back, without prior knowledge of how soon they will be back. Students must pay cash when getting on the tricycle with that aside there is also the problem of price indifferences. Tricycle riders can be sitting idle at the loading point while there is a high demand for their services within the school. The stated problem jogged my interest to embark on the project

1.3 Aim and Objectives of the Study

The aim is to develop a Tricycle Booking Application for students at Kaduna Polytechnic main campus.

Objectives

The objectives of this research work are as follows:

i. Unit and integration testing will be performed to validate the design's effectiveness and efficiency, as well as to guarantee that the functionalities are error-free.

ii. Flutter will be employed in building the user interface and ensuring user experience, while Django rest framework will serve as the Restful API for interacting with Django which is employed as backend technology not forgetting the open-source NoSQL database; firebase which will be employed as the database technology.

1.4 Scope of the Study

This research work is centered on the booking of tricycles, allowing students to check for the availability of the tricycles, and obtaining information about the fares and user payment. It will not cover the booking of other transport means within the school.

1.5 Limitations of the Study

This study's scope has been constrained by several issues, including:

Time - The researcher's busy academic pursuits severely limited the time allotted for research for this study.

Finance - The need for a standard working personal computer unit to execute and debug the application software, hindered the quick and simple progress of the task.

1.6 Significance of Study

This study has a potential impact on the people who will be using the booking system by providing students with an easy and convenient way to book rides on tricycles, the system could potentially improve their mobility and access to transportation, which could in turn improve their ability to attend classes and participate in campus activities on time. Additionally, the system could help to streamline processes within the campus and make it easier for students to get the transportation they need.

1.7 Project Organization

The project is divided into three chapters. The outlines are presented below:

Chapter One: Introduction

Chapter one introduces this project work, the background of the study, the statement of the problem, the aim and objectives, the scope of the study, limitations of the study, the significance of the study, project organization, and the definition of terms.

Chapter Two: Literature review

This chapter focuses on the literature review, and the contributions of other scholars on the subject matter being discussed.

Chapter Three: Methodology and Design

This chapter is concerned with the presentation of the results of system analysis and design. It presents the research methodology used in the development of the system to facilitate an understanding and effective future implementation of the system.

1.8 Definition of Terms

i. **Mobile-based:** Refers to a system that is accessed via a mobile device, such as a smartphone or tablet**.**

ii. **Tricycle (Keke):** A three-wheeled vehicle that is commonly used for transportation in many parts of the world, including Nigeria**.**

iii. **Booking system:** A system that allows users to reserve or book a service or product, such as a ride on a tricycle**.**

iv. **Campus:** The grounds and buildings of a college or university**.**

v. **User interface (UI):** The part of a software application that the user interacts with, including the layout, buttons, and other elements**.**

vi. **User experience (UX):** The overall experience of a user interacting with a product or service, including their emotions, perceptions, and behaviours**.**

vii. **Geolocation:** The process of identifying the geographical location of a device or user, typically using GPS technology**.**

viii. **Payment gateway:** A system that allows users to make payments online, typically using a credit card or other electronic payment method.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter's goal is to demonstrate how the topic under study relates to previous research, current practice, or other areas of knowledge by citing pertinent works by other scholars that have dealt with a similar issue. In addition, this chapter will provide a synthesis of the existing research on the topic, highlighting areas of agreement, disagreement, and gaps in the literature, to establish the importance of the project topic in the field and to identify areas for further research.

2.2 Literature review

Balba et al. (2019). ISAKAY: Android-Based Booking System for Tri-Bike Operators and Drivers Associated with Cloud-Based Data Analytics.

Due to the increasing needs of the commuters in rural areas, there are several issues with scheduling tri-bikes, such as long lines, a low possibility of riding during peak hours, a lack of trikes at unholy hours, the difficulty in securing a ride in a tri-bike becomes an issue and the stress of waiting for booking excursions an automated booking system based on current technologies were created.

Interviews were done to acquire more information to aid in the creation and design of the system. As a final outcome of the project, a system that connects users, drivers, and trike operators utilizing mobile phones and network connectivity was built. The Waterfall Model was chosen as the instrument to achieve the desired result. The suggested mobile application was created with Android Studio and runs on Android 4.4.4 Kitkat and above. GPS technology is the primary tool for mapping and navigating the intended system.

The system for online booking was created to map and locate passengers in a certain location. The system made advantage of the Global Positioning System (GPS) integration, the solution will benefit all commuters, drivers, and operators since trip monitoring is simple, booking is quick, and drivers are easily available because everything is done by mobile phone and network connectivity., while the downside of the system is that it is limited to only Android-based users and it requires internet connectivity. Based on the ISO/IEC 25010:2011 standard, the system received a score of 4.69, indicating that it is capable of resolving commuter and driver difficulties in the booking of trips in Odiongan, Romblon.

Adekola et al. (2021). An Online Road Transport Booking System. Prior to the development of the Road Transport Booking System, road transport never used any framework to preserve records of booking, instead keeping records of booking and storing transport data was done by hand. This technique has resulted in data waste, data insecurity, data delays, and reiteration of consistent data records due to staff incapacity to recover the list of documents and exchanges done in the past. To address the aforementioned gap and so achieve improved data storage and booking, an online Road Transport Booking System that also provides a good user experience and usability is required.

The proposed concept is a two-tier architecture with a backend and a front end. The development technologies employed in the development of this research include PHP, CSS, HTML, JavaScript, MYSQL database, and XAMPP server

The online road transportation ticketing system website was created to make booking easier for passengers, and as a result, it has eliminated many bottlenecks in ticketing. Non-functional needs such as usability and user experience were also met by the system design. Physical booking locations as well as certain online systems were evaluated, and improvements were included in the intended road transportation booking system. In terms of system design, consumers and company owners of road transportation systems can experience easier contact and better deals. The adoption of a web-based technique has several advantages, including the ability to access information from anywhere and at any time of day. Due to the developmental technologies used, the system might lack scalability if there is a high influx of users.

Mohamad et al. (2019). Online Booking Systems for Managing Queues at The Road Transport Department. The purpose of this research is to suggest a unique method for reducing queues or waiting times at public-sector service counters. In other cases, customers wait for many hours to perform a basic transaction service that typically takes less than a minute. In truth, this condition is not seen as time vindication. Furthermore, this condition may cause the citizen to have a negative opinion of the service given, resulting in a decrease in the reputation of the public sector. This research was conducted in parallel with RTD's aim to solve the issue of high waiting times and to assist RTD's mission.

The data and information sources for this study can be classified as either primary or secondary data. Observations, time studies, semi-structured interviews, and questionnaires are used to collect primary data. Secondary data refers to previously collected and compiled historical information or data structures that were utilized to solve problems other than the current one. These statistics were compiled from journals, books, articles, newspapers, and organizational reports. In developing the online system with Adobe Dream Weaver, Oracle SQL Developer software, and Adobe Cold Fusion, and validating and verifying the system with usability studies and questionnaires

This research proposes two types of online systems, appointment booking, and queue booking, with the goal of lowering customer waiting time at the Road Transport Department in Bukit Katil, Melaka. Furthermore, because this research does not just focus on boosting the organization's profit but also on providing high customer satisfaction through outstanding service, it might serve as a reference for other non-profit organizations or public-sector agencies to enhance their service supplied. The downside of the system is that the issues in the system have not been identified and eliminated

Duraisamy et al. (2018). Android Mobile Application for Online Bus Booking System. The existing system has the following flaws. The systems employed by the counter employees are the internal system that is only used to sell bus tickets at the counter, customers must go to the counter to purchase a bus ticket or inquire about the bus timetable, and consumers must pay in cash when purchasing a bus ticket and may have to wait in line for an extended period of time. Furthermore, clients are not permitted to purchase bus tickets over the phone, and the bus company's phone lines are always busy. The bus terminal is only open during office hours for ticket purchases. People must travel to the bus station or terminal to purchase a ticket.

The author has opted to employ a blend of both methodologies for the research methodology (quantitative and qualitative). This is because, in addition to strong statistical analysis, he requires in-depth information from particular members of the public for this attempt. A collection of programming languages is required for the effective creation of any system. The languages employed in this system are Java, Android Studio, Dreamweaver, PHP, and MySQL

As the number of travellers increases over time, so does the demand for a well-equipped bus service. Our responsive bus booking system with mobile applications (Android) is uniquely designed to cater to worldwide clients for travel management organizations, destination management companies, travel aggregators, business-to-business (B2B), business-to-customer (B2C) travel agencies, and tour operators. Furthermore, the system is limited to only Android-based users and requires internet connectivity

Wibawa et al. (2021). Bus Ticket Booking Information System. This article was written during the covid pandemic, and mobility is limited during a pandemic like this one. As a result, it has a significant impact on commonly performed tasks, such as transportation. In this circumstance, people choose to stay inside their houses and avoid crowds in order to limit their exposure to the Covid-19 virus. This has an impact on the sale of vehicle tickets in a variety of ways, including air, land, and sea travel. A bus ticket booking website was designed to address this issue.

In this information system, we perform a survey on pre-existing websites connected to ticketing in terms of purchasing tickets online, both in terms of observing and strengthening current features that will be built later. Furthermore, we performed inspections and interviews at the use case to determine the needs, operations, and data required so that the system we designed is helpful and in agreement with what is now in the firm. A set of programming languages is essential for the efficient development of any system. HTML, CSS, PHP, and MySQL are the languages used in this system.

The system is anticipated to make it easier for customers to book transport buses at the same time they purchase tickets to their desired locations. This information technology can also help travel bus business owners run their operations more effectively. This information system not only makes it easy for customers and company owners, but it also has a good influence on the destination because it may concurrently advertise these areas.

2.3 Summary of Related Literature Reviews

Author & Year	**Title & Description**	**Merit and Demerits**
Balba et al. (2019)	ISAKAY: Android-based booking system for tri-bike operators and drivers associated with cloud-based data analytics. The researchers devised the concept of building and developing an Android-based booking system that centralizes the transportation management of tricycles in rural locations around the Philippines.	The booking system can mass-produce a dashboard by displaying data analytics for future tri-bike management system upgrades. The system is limited to only Android-based users.
Adekola et al. (2021).	An Online Road Transport Booking System. This research focuses on automating the Road Transportation Booking System so that customers and employees can buy and sell tickets online.	customers and employees can buy and sell tickets online. The system lack scalability, high influx of user makes the site slow.
Mohamad et al. (2019).	Online Booking Systems for Managing Queues at The Road Transport Department.	This system informs clients of the length of their wait so that they can arrive at the

	The purpose of this research is to suggest an online system for managing queues during service and, as a result, reducing waiting time.	counter in time for their allotted service. The system is limited to only Android-based users and requires internet connectivity
Duraisamy et al. (2018).	Android Mobile Application for Online Bus Booking System The system enables consumers to book tickets at the touch of a button without having to wait in line or go to the counter	Well-designed payment platform. The system is limited to only Android-based users.
Wibawa et al. (2021).	Bus Ticket Booking Information System Tickets can be purchased using this bus ticket booking website instead of going straight to the ticket sales shop.	Buyers can purchase tickets using their smartphones or laptops. The cost of the service is expensive.

CHAPTER THREE

METHODOLOGY AND DESIGN

3.1 Introduction

A methodology is an approach to rigorous study or investigation, particularly to uncover new facts or information; hence, research methodology should be good enough to make the attainment of the established objectives attainable with certain components, such as methods of data collecting and design. This chapter includes the input/output specifications and system requirements for the development of a tricycle booking application, as well as the system modeling (use case, activity, and class diagrams).

3.2 Methods of Data Collection

It is crucial to acquire data and facts about the current system before implementing any system since one has to understand what is happening. Two techniques were used to conduct this study.

i. Observation of the Work Environment
ii. Documentation

3.2.1 Observation of the Work Environment

This strategy was used to collect information and data for this study by observing how the manual system functioned. Through close observation, the most obvious problems with the current system were found. The setting in which the observation is made can be changed in a variety of ways by using the observational method.

3.2.2 Documentation

Secondary data gathering includes documentation. Journals, manuals, previous work, publications, and other sources are used in this manner. This data-gathering strategy is chosen because it allows for comparison with previous research. This includes the internet, which is a tool for data collecting. The internet was utilized to research complex or unclear problems.

3.3 System Modeling

A system model is a conceptual model that describes and represents a system. Any interaction between a group of components that work together to accomplish a single goal is referred to as a system. Visual models of the object-oriented software-intensive systems can be made using a set of graphic notation techniques that are part of the Unified Modeling Language, which is employed in this modern system design. Use case diagrams, class diagrams, and activity diagrams are among the UML diagrams used in this new design.

3.3.1 Use Case Diagrams

Use cases are collections of interactions between systems and users. Use case diagrams are used to visually summarize a system's functionality in terms of its actors, its goals (represented as use cases), and any dependencies between those use cases.

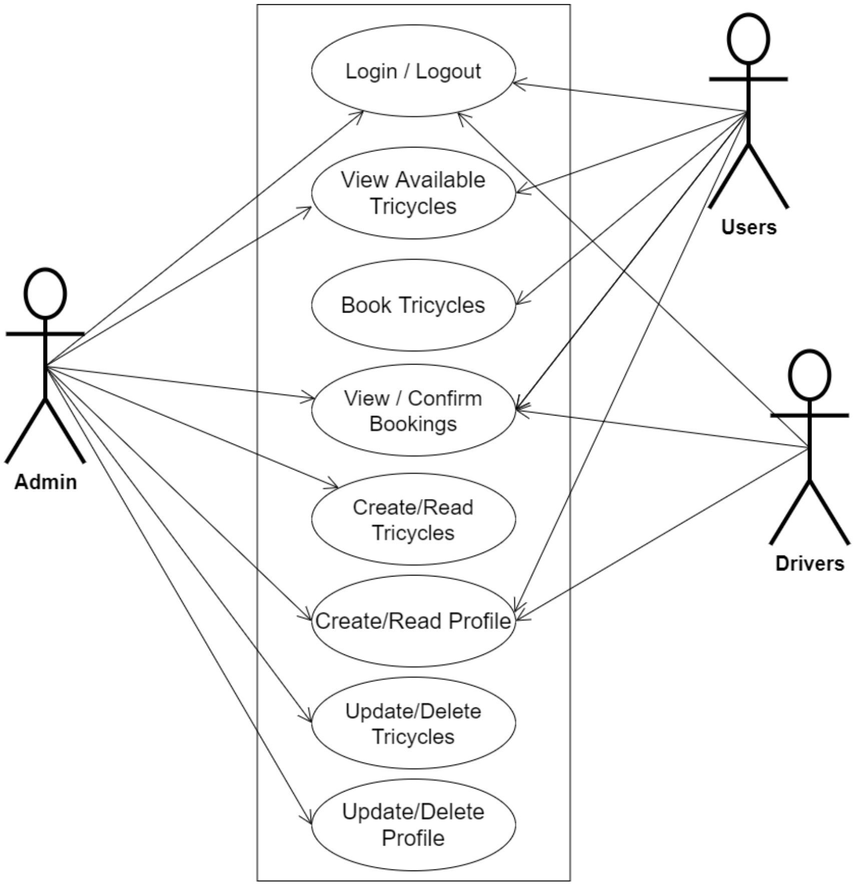


Fig 3.1 System Use Case Diagram

3.3.2 Class Diagrams

The Unified Modeling Language (UML) class diagram is an implementation of an independent view of how the system interface will be, with each class having its own properties and illustrating how they interact with one another. Class diagrams use the rules established by the Unified Modeling Language to visually depict the static structure and composition of a specific system (UML).

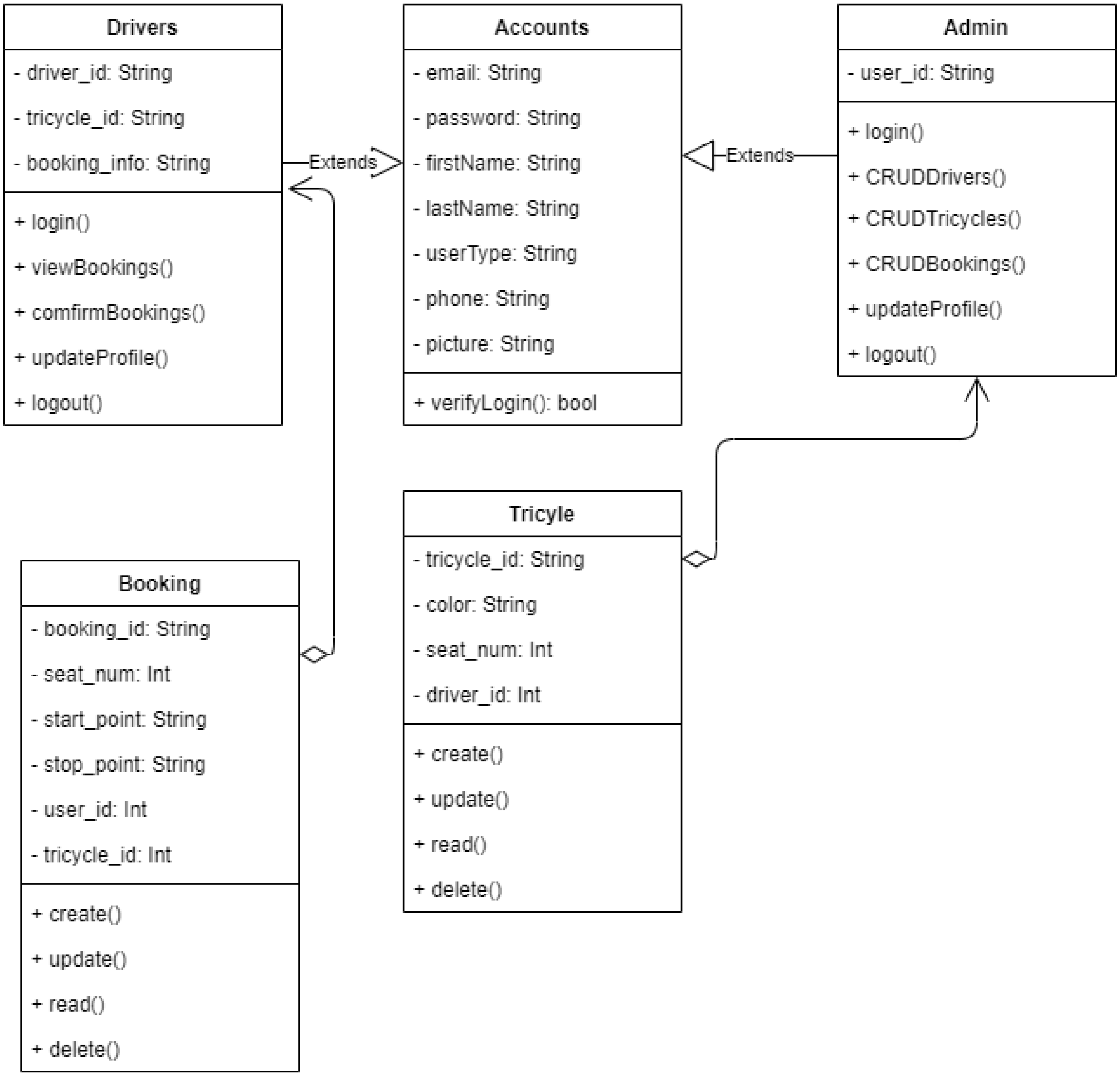


Fig 3.2 System Class Diagram

3.3.3 Activity Diagrams

Similar to a flowchart or a data flow diagram, an activity diagram visually depicts a sequence of events or the flow of control in a system, but it functions more like an advanced version of both.

Login

The process for gaining access to the system is depicted in the diagram below; in order to gain access, the email address and password must be accurate.

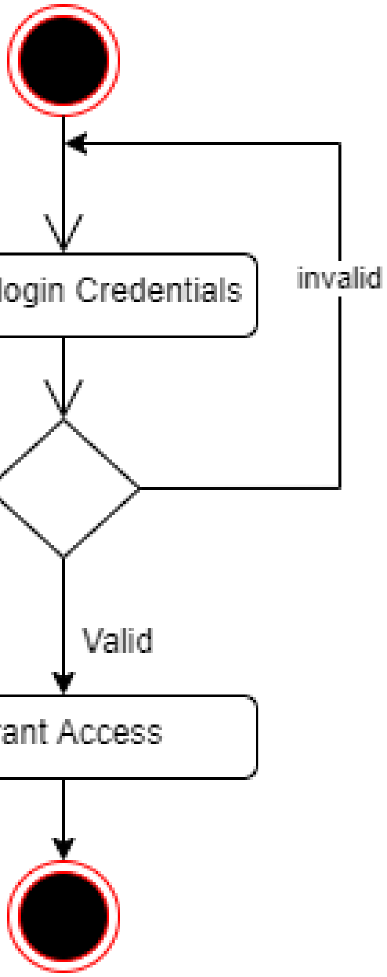


Fig 3.3 Login Activity Diagram

Booking Tricycle

The process for booking a tricycle is depicted below, to book one has to check for the availability of the tricycle.

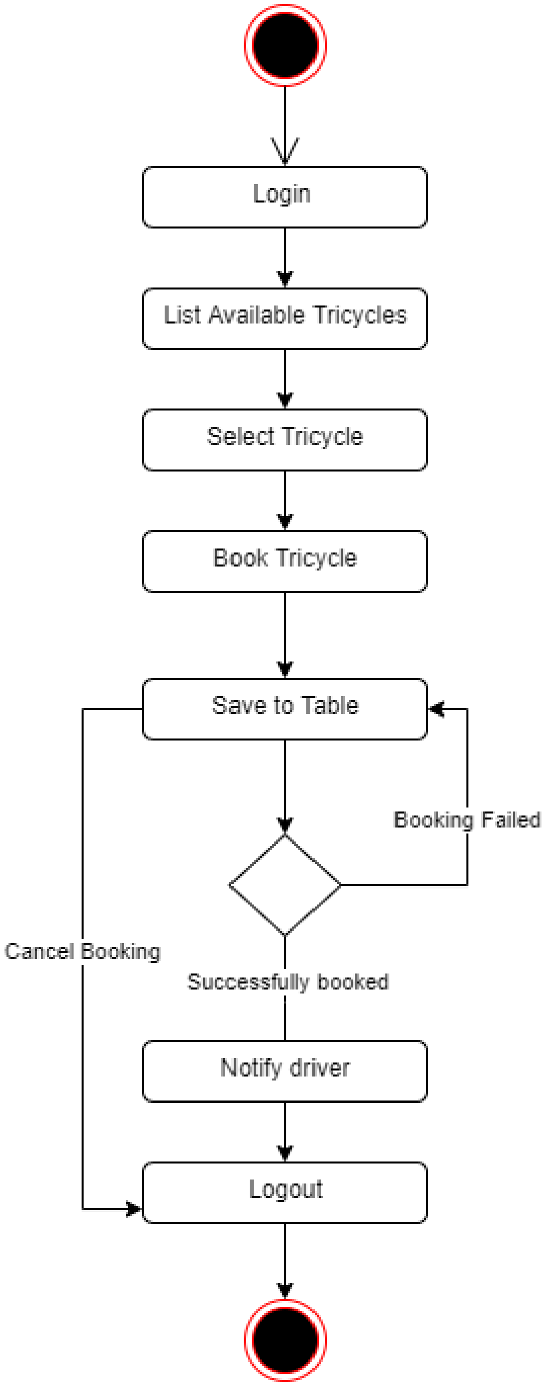


Fig 3.4 Booking Activity Diagram

3.4 Database Design

The logical explanation of how data is kept in the computer's memory is called input specification. The freedom experienced in using the system, as well as the convenience of retrieving and reading the data and assuring applicability across the internet, make SQL standards essential for ensuring that structured data is uniform and independent of applications. Some of the input specifications employed in this project work are presented below.

i. Accounts Table: contains basic information about all system users.
ii. Booking Table: contains every system booking information.

Table 3.1 Account Input Specification Table

FIELD NAME	DATA TYPE	LENGTH	DESCRIPTION
Email	String	150	Email for login (case sensitive)
Password	String	150	Access Code (case sensitive)
Firstname	String	150	User first name
Lastname	String	150	User last name
Phone	String	150	User phone number
Picture	String	-	User profile picture
acct_id	String	64	A unique string for identifying users

Primary key: acct_id

Table 3.2 Booking Input Specification Table

FIELD NAME	DATA TYPE	LENGTH	DESCRIPTION
Tricycle_id	String	150	A unique string for identifying tricycles
Seat_num	Integer	10	Passenger seat number
Starting_point	String	150	Passenger starting location
Stopping_point	String	150	Passenger stopping location
User_id	String	150	A unique string for identifying user
Payment_type	String	20	Cash or online payment
book_id	String	64	A unique string for identifying users

Primary key: book_id

3.5 Output Design

This declares and displays the outcome of the given input. The automated system's output is dependent on its input. The output specification is listed below.

Table 3.3 Account Output Design Table

Email	Password	Firstname	Lastname	Phone	Picture	Acct_id
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

Table 3.4 Booking Output Design Table

Tricycle_id	Seat_num	Starting_point	Stopping_point	User_id	Payment_type	Book_id
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX

3.6 Input & User Interface Design

This shows a visual representation of the system interface; it will be made to be intuitive to use, quick to respond, and visually appealing. Additionally, it will be properly protected, so signing in will be necessary to view some levels of the contents. A mid-fidelity wireframing application named Draw.io is used to assist with the designs.

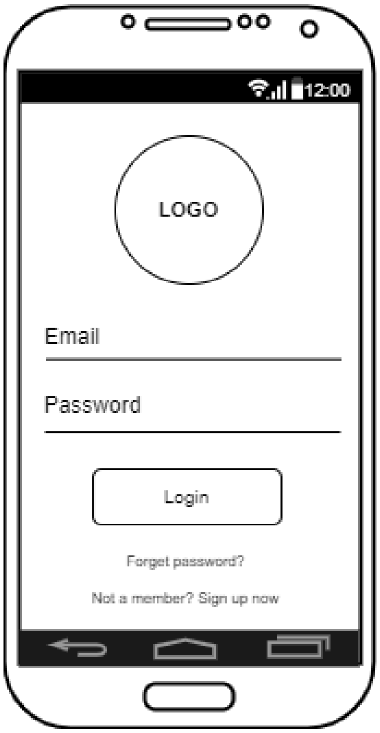


Fig 3.5 User Login Screen

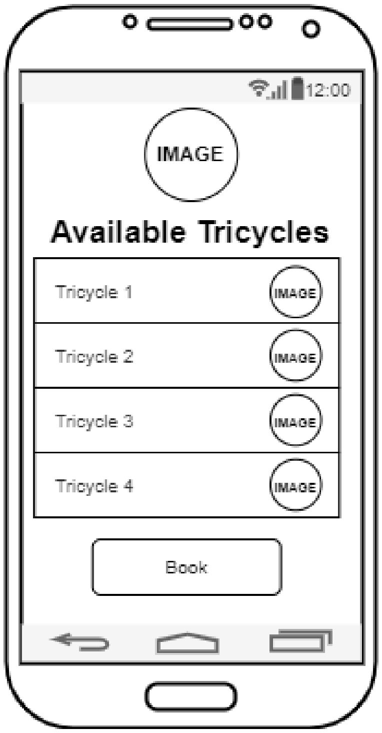


Fig 3.6 Available Tricycles Screen

3.7 System Requirement

Every piece of software that is created has preset system requirements that it must meet in order to run at its best. However, the system requirements are the bare minimum hardware and software needed for the system's intended operation.

3.7.1 Hardware Requirement

System Hardware Requirement Include:

a. Minimum of 8 GB of RAM (Random Access Memory) installed.

b. Minimum of intel core i3 processor.

c. Minimum of 250GB HDD (Hard Disk Drive).

3.7.1 Software Requirement

The software requirements include:

a. At least windows 10 OS (Operating System).

b. Flutter Installation.

c. Vs. Code / Android studio installation.

d. Emulator installation.

3.8 Choice of Programming Language

The proposed design will be implemented using flutter for its user interface (frontend) while Django will be used for the backend, firebase will be used for its database due to its portability, and Django REST Framework will be employed for its REST-full APIs, the combination of the above modern technology forms the technology for this research work.

CHAPTER FOUR

SYSTEM IMPLEMENTATION EVALUATION

4.1 Introduction

This section describes in concise detail how the new system is implemented for effective operation. It shows samples of the working (new) system designed and how the system is to be installed.

4.2 System Testing and Evaluation

There are many reasons to conduct the testing for the developed system because is only through testing that we can be able to analyze any problem in the new system and provide solutions to these problems This project employed both unit and integration testing to ensure the effectiveness and efficiency of design and to ensure that the new system meets its required functionalities and is error-free.

Unit Testing

In this section, individual units or single components of the system are tested independently to ensure that individual phases are working effectively without errors.

Integration Testing

Testing of the program was implemented using integration testing all the units were put together as one so they function as one. The link between the various units was tested to be sure that they are correctly integrated, and also to be sure that the units can function correctly together as one.

4.3 System Installation

In order to use the proposed application on any computer system, the following steps need to be taken:

1. Make sure, android studio, JDK, and Android emulator are installed on the system.
2. Copy your project folder to any location of your choice.
3. Open the project folder in Visual Studio Code

4. In the terminal run “`flutter pub get`” to get all the dependencies in the pubspec.yaml file
5. Select the Android emulator as the device to be used.
6. Locate the main.dart file and run the file in debug mode.

4.4 Security Measures

Since the scope of the application is public, some of the information and screens are restricted to either the tricycle driver or the user (student). But some other information and functionalities are restricted to some and not all who use the application. The restrictions are carried out by the use of passwords which gives different levels of access to users.

4.5 Sample Outputs

These describe and give the pictorial representation of the program or software; it shows and gives a clear understanding of the design, and displays all the interfaces

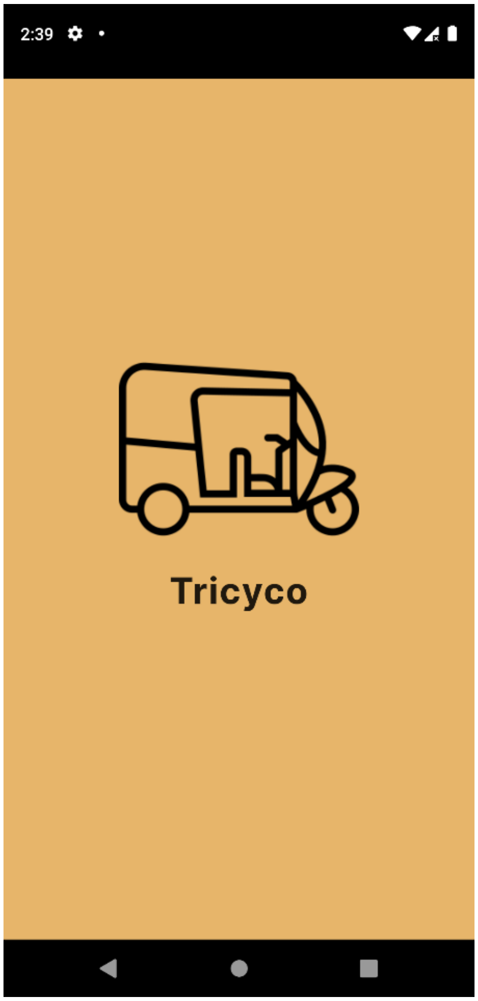


Fig 4.1 Splash Screen

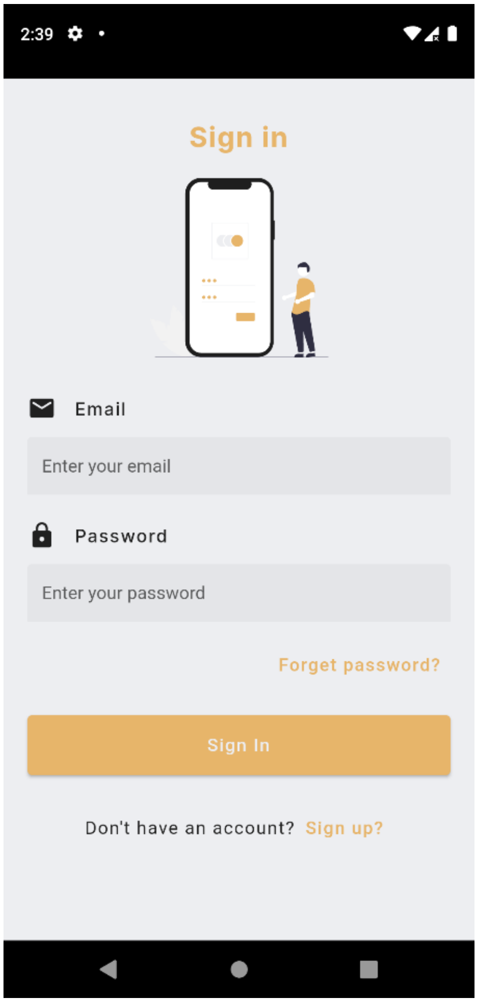
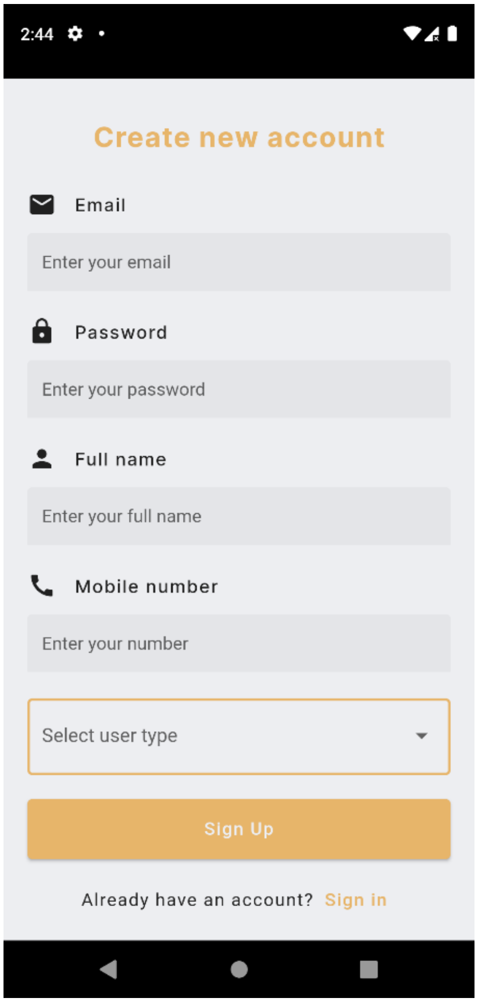


Fig 4.2 Sign-in Screen

Fig 4.1 Splash Screen: This is the first screen displayed to every user that wishes to make use of the application.

Fig 4.2 Sign-in Screen: The screen grants users to access (students, drivers) to the application only if the correct credentials are provided.



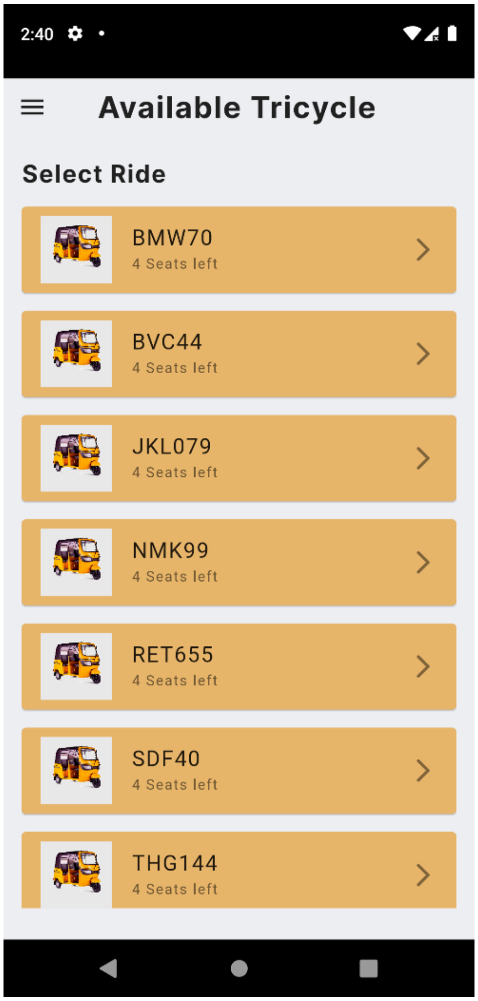
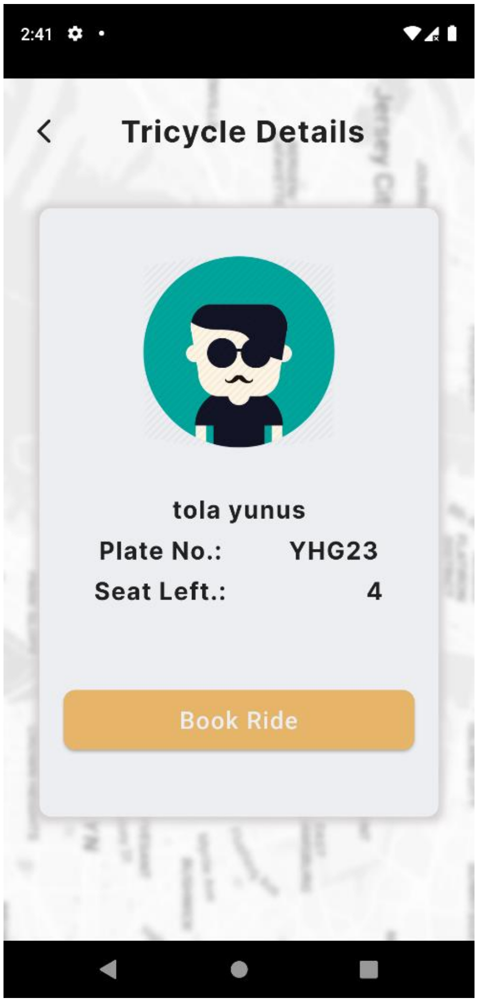


Fig 4.3 Sign-up Screen

Fig 4.4 Available Tricycle

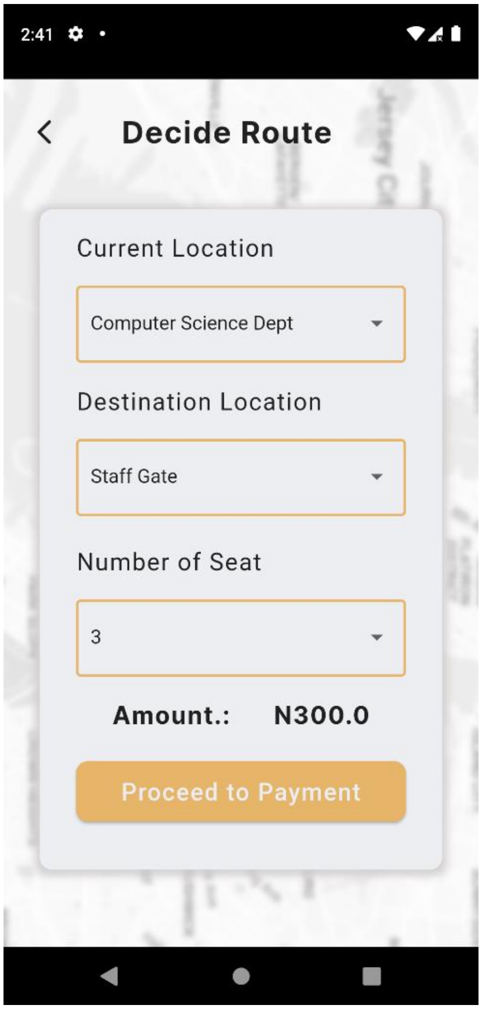
Fig 4.3 **Sign-up Screen**: The application ensures that only registered users can have access to the system, there the screen enables the creation of accounts for new users (students and drivers)

Fig 4.4 **Available Tricycle**: Authenticated students can see a list of the available tricycles, and for each tricycle, they can see the available seats left



tola yunus
Plate No.: YHG23
Seat Left.: 4

Fig 4.5 Tricycle Details

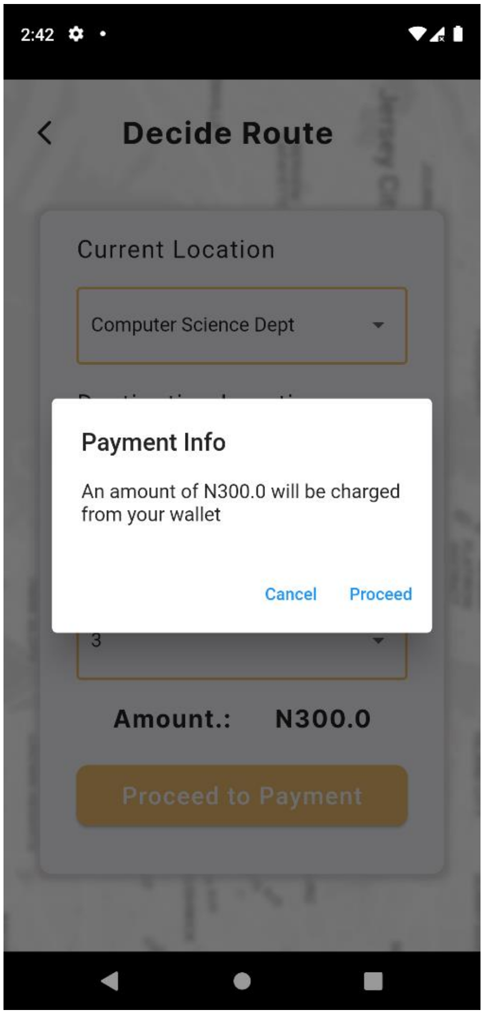


Current Location: Computer Science Dept
Destination Location: Staff Gate
Number of Seat: 3
Amount.: N300.0

Fig 4.6 Decide Route Screen

Fig 4.5 **Tricyle Details**: The screen displays the details of the tricycle, the details contain information such as the driver image, driver plate number, and the number of seats left

Fig 4.6 **Decide Route Screen**: The page handles the navigation of the student, where the student selects the number of seats he or she is booking for, the current and destination location of the student



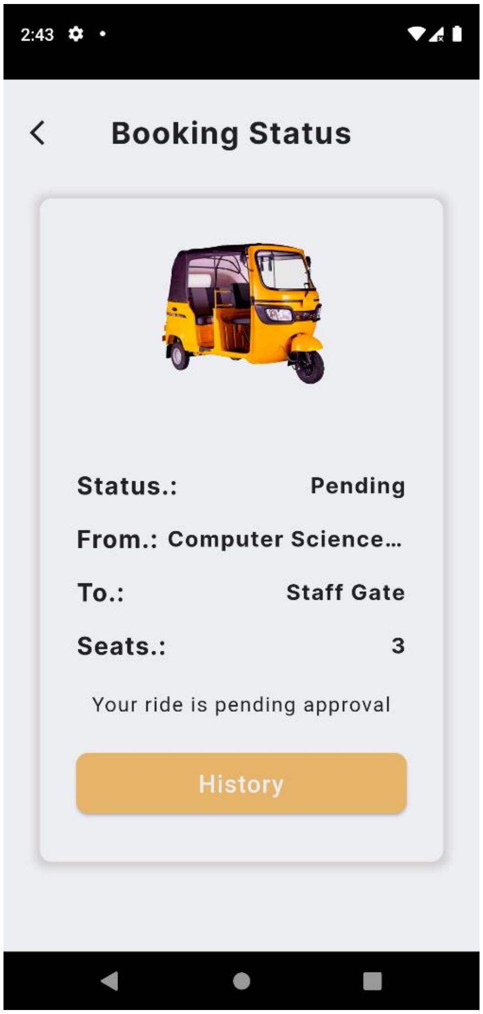
Current Location

Computer Science Dept

Payment Info

An amount of N300.0 will be charged from your wallet

Amount.: N300.0



Status.: Pending

From.: Computer Science...

To.: Staff Gate

Seats.: 3

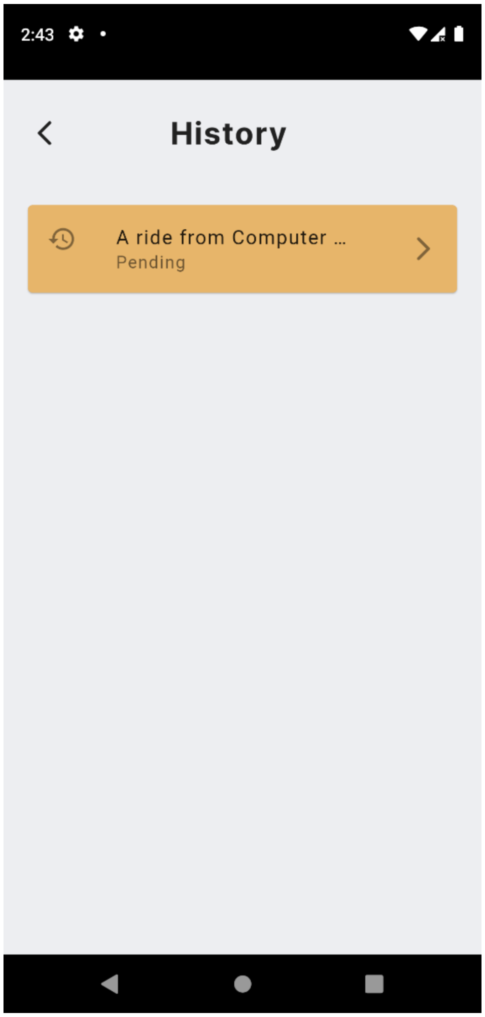
Your ride is pending approval

Fig 4.7 Payment Confirmation

Fig 4.8 Booking Status

Fig 4.7 **Payment Confirmation**: The page ensures student confirms that a certain amount of money will be charged from their account, possible exceptions have been handled gracefully.

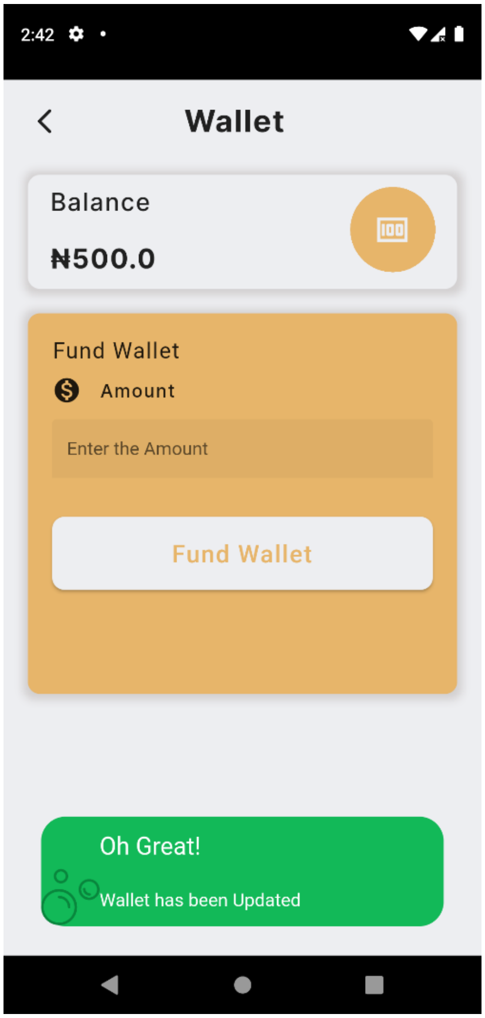
Fig 4.8 **Booking Status**: The page displays the status of any booked ride, here student can confirm if the ride has been approved by the driver.



History

A ride from Computer ...
Pending

Fig 4.9 History Page



Wallet

Balance
₦500.0

Fund Wallet
Amount
Enter the Amount

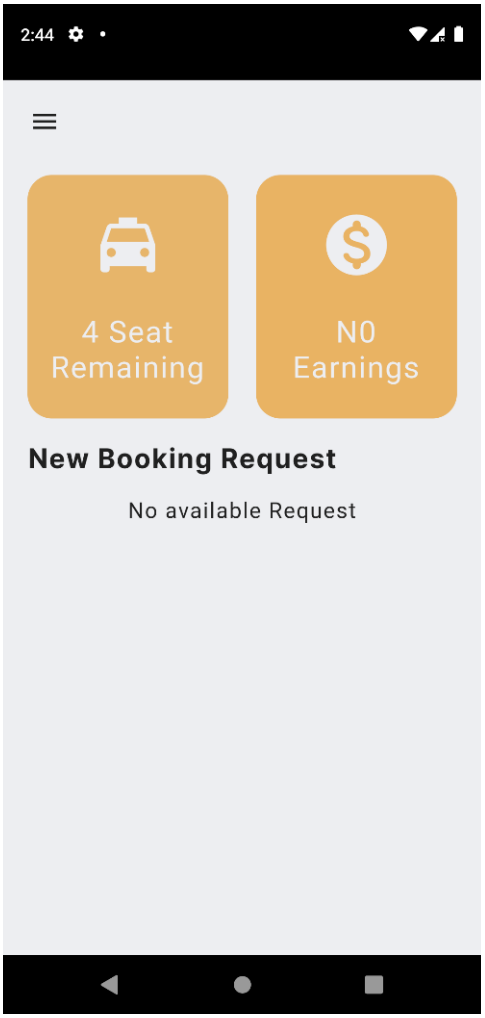
Oh Great!
Wallet has been Updated

Fig 4.10 Fund Wallet

Fig 4.9 **History Page**: A comprehensive list of all booked rides can be seen from this page, the list attaches a quick preview of the status of each booked ride.

Fig 4.10 **Fund Wallet**: Since the student has to make payments from their wallet, the page is a simulation of the funding of the wallet.

Driver Screens



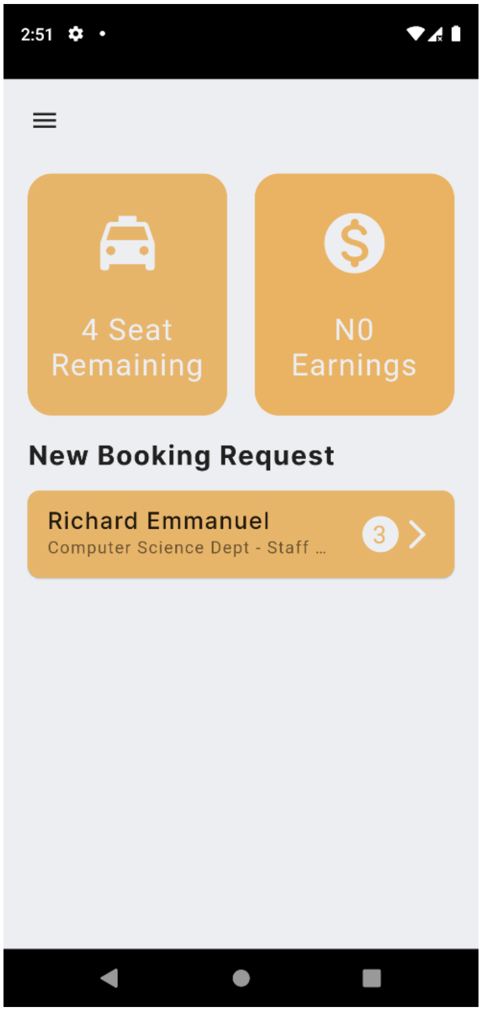
4 Seat Remaining

N0 Earnings

New Booking Request

No available Request

Fig 4.11 Driver Home Screen



4 Seat Remaining

N0 Earnings

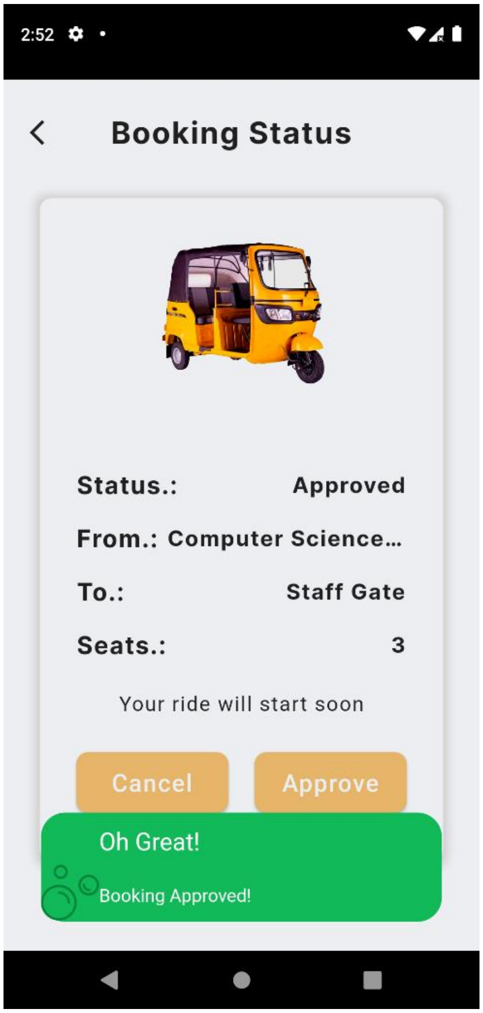
New Booking Request

Richard Emmanuel
Computer Science Dept - Staff ...

Fig 4.12 Booking Request

Fig 4.11 **Driver Home Screen**: The screen is the driver home page where the driver can see the amount he has earned, the number of seats left, and the booking request.

Fig 4.12 **Booking Request**: The application implements a real-time feature so booking requests can come in without refreshing the screen.



Booking Status

Status.: Approved

From.: Computer Science…

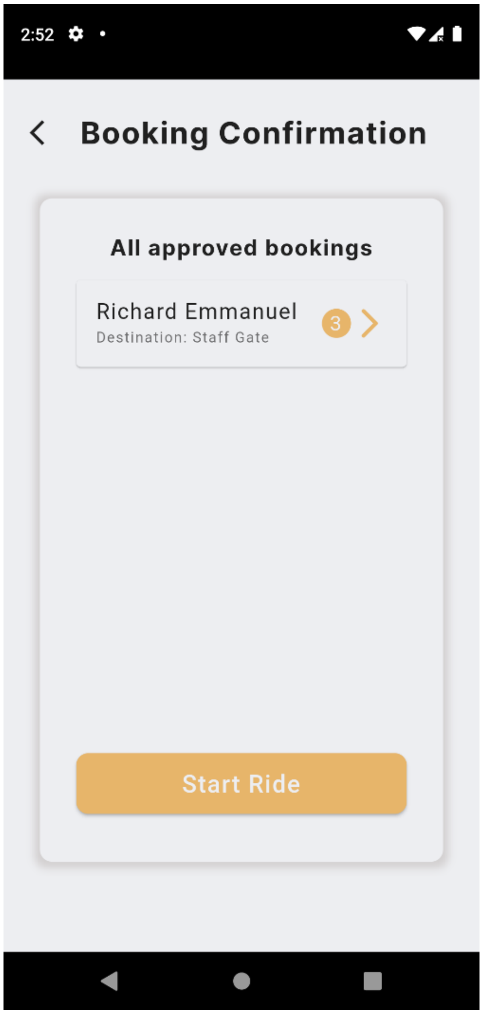
To.: Staff Gate

Seats.: 3

Your ride will start soon

Oh Great!

Booking Approved!



Booking Confirmation

All approved bookings

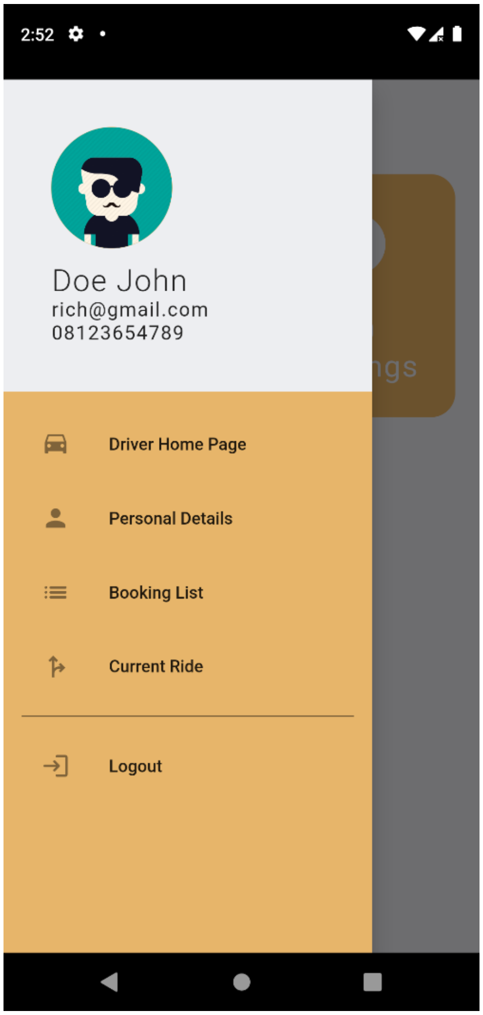
Richard Emmanuel 3
Destination: Staff Gate

Fig 4.13 Approve Booking

Fig 4.14 Start Ride

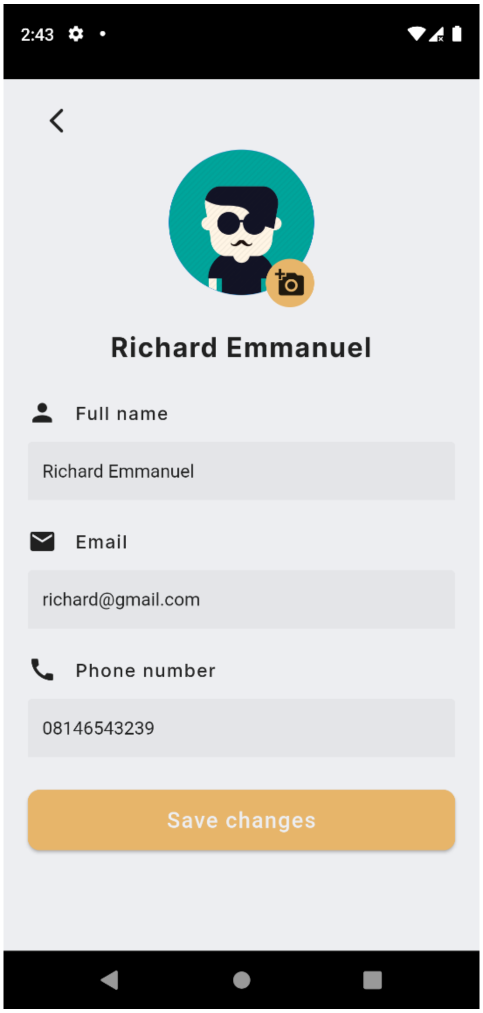
Fig 4.13 **Approve Booking**: On the booking status page, the driver can approve or cancel a ride.

Fig 4.14 **Start Ride**: On this page, the driver can see all the approved rides and can decide to start the ride



2:52

Doe John
rich@gmail.com
08123654789



2:43

Richard Emmanuel

Full name: Richard Emmanuel

Email: richard@gmail.com

Phone number: 08146543239

Fig 4.15 Sidebar

Fig 4.16 User profile

Fig 4.15 **Sidebar**: This is a navigation drawer that shows the functionality available for the authenticated user.

Fig 4.16 **User Profile**: On this page, the authenticated user can update his/her account profile

CHAPTER FIVE

SUMMARY CONCLUSION AND RECOMMENDATION

5.1 Summary

This project aims to develop an efficient mobile application that allows students at Kaduna Polytechnic to book tricycles for transportation within the main campus. The application will address the challenges faced by students in accessing reliable and convenient transportation services on campus. By providing a user-friendly platform, the project seeks to enhance students' mobility, reduce waiting times, and improve overall transportation efficiency. This project aims to enhance the student experience and promote a more convenient and accessible campus environment.

5.2 Conclusion

In conclusion, this research project has made significant strides toward improving the transportation system for students at Kaduna Polytechnic's main campus through the development of the Tricycle Booking Application. By providing a convenient and reliable platform for tricycle bookings, this application addresses the challenges faced by students in accessing transportation services on campus. The project successfully implemented a proof of concept, demonstrating the feasibility and effectiveness of the proposed prototype. Through this application, students can easily book tricycles and enjoy improved mobility within the campus, reducing waiting times and enhancing overall transportation efficiency. This project contributes to enhancing the student experience and fostering a more accessible and convenient campus environment.

5.2 Recommendation

Based on the findings and outcomes of the project the following recommendations are proposed:

i. User Feedback and Continuous Improvement: It is crucial to gather feedback from students who use the tricycle booking application regularly. This feedback will help identify any issues, challenges, or areas for improvement. Continuous updates and enhancements should be made based on user feedback to ensure the application remains user-friendly and meets the evolving needs of the students.

ii. Scalability and Expansion: As the application gains popularity and the user base expands, it is essential to ensure the scalability and reliability of the system. The infrastructure should be capable of handling increased user traffic and be scalable to accommodate future growth. Regular monitoring and performance optimization should be carried out to maintain a seamless user experience.

iii. Collaboration with Tricycle Operators: Establish a partnership or collaboration with tricycle operators within the campus to streamline the booking process and ensure a sufficient number of tricycles are available for students. This can include incentives for tricycle operators to participate in the application and maintain a high level of service.

iv. Promote Awareness and Adoption: Conduct awareness campaigns and provide training sessions to familiarize students with the tricycle booking application and its benefits. Encourage widespread adoption of the application among the student community to maximize its effectiveness and ensure its long-term sustainability.

By implementing these recommendations, the Tricycle Booking Application can further enhance the transportation experience for students at Kaduna Polytechnic, promoting efficiency, convenience, and accessibility within the main campus.

REFERENCES

Adekola, O.D., Mensah, Y.M., Maitanmi, S.O., Akande, O., Kasali, F.A., Omotunde, A.A., Ajayi, W., & Ajiboye, O.A. (2021). “An Online Road Transport Booking System”. *Asian Journal of Computer Science and Technology.*

Balba, N. P., Rosas, N. M. F., Drilon, P. J. Z., Fallaria, S. J. F., & Fallaria, J. F. (2019). “ISAKAY: Android-based booking system for tri-bike operators and drivers associated with cloud-based data analytics”. *International Journal of Innovative Technology and Exploring Engineering*, *8(*10), 1265–1269. https://doi.org/10.35940/ijitee.I7535.0881019

Bari, Md & Efroymson, Debra. (2022). “Efficient Use of Road Space and Maximisation of Door-to-Door Mobility: Suggestions for Improvements in Dhaka.”

Duraisamy, S., & Abuhuraira, J. U. (2018). “Android Mobile Application for Online Bus Booking System”. *International Journal of Information System and Engineering, 6*(July), 34–56.

Mohamad, E., Jiga, I. A., Rahmat, R., Azlan, A., Rahman, M. S. A., & Saptari, A. (2019). “Online Booking Systems for Managing Queues at The Road Transport Department”. *JIE Scientific Journal on Research and Application of Industrial System*, *4*(1), 21. https://doi.org/10.33021/jie.v4i1.745

Para, P. (2022). Importance of Transportation. John K Philips. https://www.johnkphilips.co.uk/blog/importance-of-transportation/

Wibawa, S. C., Fatah, D. A., Huda Firmansyah, Moch. M., Mustafit, N. H., A., D. L., Putri, E. R., & Laksono, C. S. (2021). “BUS TICKET BOOKING INFORMATION SYSTEM.” *International Journal of Science, Engineering and Information Technology, 6*(1), 266–269. https://doi.org/10.21107/ijseit.v6i1.13155

Agustin, C. P., & Costales, N. A. (2018). “Booking System”. *International Journal of Advanced Research in Management and Social Sciences*, 10(6), 211-236.

Tulio, J. (2017). “How Important are Grab and Uber to Filipinos”. *Top Gear*. Retrieved from https://www.topgear.com.ph/features/feature-articles/uber-grabstories-filipinos-a00187-20170721?ref=article_tag

Ahmed, K., Khan, A., Siddiqui, O. A., Iftikhar, S. A., Das, B., & Khurrum, M. (2018). “Proposed Efficient Method for Ticket Booking (Pemtb)”. *Journal of Fundamental and Applied Sciences*, 10(6S), 322.

Doni, A. F., Mufarroha, F. A., Negera, Y. D. P., & Sugiarti, E. (2021). “Development of Web Based Information System for Branding Services”. *OP Conference Series: Materials Science and Engineering*, *1125*, (1), 012051. doi:10.1088/1757-899X/1125/1/012051.

Younas, M., Jawawi, D. N. A., Ghani, I., & Kazmi, R. (2017). “Non-Functional Requirements Elicitation Guideline for Agile Methods”. *Journal of Telecommunications, Electronic and Computer Engineering*, 9(3-4 Special Issue), 137–142.

APPENDIX

Home.dart

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:tricycle/components/delegatedText.dart';
import 'package:tricycle/components/navigationDrawer.dart';
import 'package:tricycle/controllers/tricycleDetailsController.dart';
import 'package:tricycle/models/tricycle_data.dart';
import 'package:tricycle/services/database.dart';
import 'package:tricycle/utils/constant.dart';

class HomePage extends StatefulWidget {
  HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  final scaffoldKey = GlobalKey<ScaffoldState>();

  DatabaseService databaseService = Get.put(DatabaseService());

  TricycleDetailsController tricycleDetailsController =
      Get.put(TricycleDetailsController());

  @override
  Widget build(BuildContext context) {
    final size = MediaQuery.of(context).size;
    return SafeArea(
child: Scaffold(
  key: scaffoldKey,
  backgroundColor: Constants.secondaryColor,
  drawer: const DelegatedNavigationDrawer(),
  body: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Row(
        children: [
          IconButton(
            onPressed: () => scaffoldKey.currentState!.openDrawer(),
            icon: const Icon(
              Icons.menu,
              size: 25,
            ),
          ),
          const SizedBox(width: 30),
          DelegatedText(
            text: "Available Tricycle",
            fontSize: 25,
            fontName: 'InterBold',
          ),
        ],
      ),
      Padding(
        padding: const EdgeInsets.symmetric(
          horizontal: 15,
          vertical: 20,
        ),
        child: SizedBox(
          height: size.height * .8,
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    DelegatedText(
      text: 'Select Ride',
      fontSize: 20,
      fontName: 'InterBold',
    ),
    Expanded(
      child: SingleChildScrollView(
        child: StreamBuilder<List<TricycleData>>(
          stream: databaseService.readTricycleData(),
          builder: (context, snapshot) {
            if (snapshot.hasError) {
              return Text(
                  "Something went wrong! ${snapshot.error}");
            } else if (snapshot.hasData) {
              final tricycleDataList = snapshot.data!;
              if (tricycleDataList.isNotEmpty) {
                return ListView.builder(
                  shrinkWrap: true,
                  physics: const NeverScrollableScrollPhysics(),
                  itemCount: tricycleDataList.length,
                  itemBuilder: (context, index) {
                    final tricycleData =
                        tricycleDataList[index];
                    return InkWell(
                      onTap: () => {
                        tricycleDetailsController.driverID =
                            tricycleData.id,
                        tricycleDetailsController
              .getTricycleDetails()
        },
        child: Card(
          margin: const EdgeInsets.only(top: 15),
          color: Constants.primaryColor,
          child: ListTile(
            leading:
                Image.asset("assets/keke.jpeg"),
            title: DelegatedText(
              text: tricycleData.plateNumber,
              fontSize: 18,
            ),
            subtitle: DelegatedText(
                text:
                    "${tricycleData.pass} Seats left",
                fontSize: 12),
            trailing: const Icon(
                Icons.arrow_forward_ios_rounded),
          ),
        ),
      );
    },
  );
} else {
  return Center(
    child: Padding(
      padding: const EdgeInsets.only(top: 18.0),
      child: DelegatedText(
          text: "No available Tricycle",
          fontSize: 20),
    ),
                );
               }
              } else {
               return const Center(
                 child: CircularProgressIndicator());
              }
             },
            ),
           ),
          ),
         ],
        ),
       ),
      ),
     ],
    ),
   ),
  );
 }
}
```

Services.dart

```
import 'dart:ffi';
import 'dart:io';
import 'dart:typed_data';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_storage/firebase_storage.dart';
import 'package:get/get.dart';
import 'package:tricycle/models/bookingList_data.dart';
import 'package:tricycle/models/tricycle_data.dart';
import 'package:tricycle/models/user_data.dart';
import 'package:tricycle/models/wallet_data.dart';
import 'package:flutter/services.dart';
import 'package:tricycle/utils/constant.dart';

class DatabaseService extends GetxController {
  String? uid;
  DatabaseService({this.uid});

  UserData? userData;

  WalletData? walletData;

  // collection reference
  var usersCollection = FirebaseFirestore.instance.collection("users");
  var bookingCollection = FirebaseFirestore.instance.collection("BookingList");
  var tricycleCollection =
      FirebaseFirestore.instance.collection("tricycleData");
  var walletCollection = FirebaseFirestore.instance.collection("wallet");
  var filesCollection = FirebaseStorage.instance.ref();

  //Create user
  Future createUserData(
      String name, String email, String phone, String userType) async {
    walletCollection.doc(uid).set({'balance': 0});
    await setImage(uid);
    return await usersCollection.doc(uid).set(
      {
        'name': name,
        'email': email,
        'phone': phone,
        'userType': userType,
      },
    );
  }

  //Determine userType
  Future<UserData?> getUser(String uid, String? userType) async {
    // Query database to get user type
    final snapshot = await usersCollection.doc(uid).get();
    // Return user type as string
    if (snapshot.exists) {
      if (userType != 'Driver') {
        userData = UserData.fromJson(snapshot.data()!);
      }
      return UserData.fromJson(snapshot.data()!);
    }
    return null;
  }

  //Check if the driver profile is updated
  Future<bool> checkIfProfileUpdated(String uid) async {
    // Query database to check if user has updated profile
    final snapshot = await tricycleCollection.doc(uid).get();
    // Return true or false
    if (snapshot.exists) {
      final data = TricycleData.fromMap(snapshot.data()!, snapshot.id);
      if (data.plateNumber != "") {
        return true;
      }
    }
    return false;
  }

  //Update tricycle data
  Future<bool> updateTricycleData(String plateNumber, String color) async {
    await tricycleCollection.doc(uid).update({
      "plateNumber": plateNumber,
      "color": color,
    });
    return true;
  }

  Stream<List<TricycleData>> readTricycleData() {
    return tricycleCollection
        .where('status', isEqualTo: true)
        .where("plateNumber", isNotEqualTo: "")
        .snapshots()
        .map((snapshot) =>
            snapshot.docs.map((doc) => TricycleData.fromJson(doc)).toList());
  }

  Future<TricycleData?> getTricycleData(String driveID) async {
    final snapshot = await tricycleCollection.doc(driveID).get();
    if (snapshot.exists) {
      return TricycleData.fromMap(snapshot.data()!, driveID);
    }
    return null;
  }

  Future<String> bookTricycle(String? userID, String? driverID, String to,
      String from, int seats) async {
```

```
  final docRef = await bookingCollection.add({
    'userID': userID,
    'driverID': driverID,
    'to': to,
    'from': from,
    'status': false,
    'seats': seats,
    'hasCompleted': false,
    'disapprove': false,
    'created': FieldValue.serverTimestamp()
  });
  return docRef.id;
}

Stream<Booking?> getBookingStatus(String? uid) {
  return bookingCollection.doc(uid).snapshots().map(
    (snapshot) {
      if (snapshot.exists) {
        return Booking.fromJson(snapshot.data()!);
      }
      return null;
    },
  );
}

Stream<List<BookingList>> getUserBookings(String? uid) {
  return bookingCollection
      .where('userID', isEqualTo: uid)
      .orderBy('created', descending: true)
      .snapshots()
      .map(
        (snapshot) =>
            snapshot.docs.map((doc) => BookingList.fromJson(doc)).toList(),
      );
}

Future<QuerySnapshot<Map<String, dynamic>>> getUserPendingBookings(
    String? uid) {
  return bookingCollection
      .where('userID', isEqualTo: uid)
      .where('status', isEqualTo: false)
      .where('disapprove', isEqualTo: false)
      .get();
}

Stream<WalletData?> getBalance(String uid) {
  return walletCollection.doc(uid).snapshots().map((snapshot) {
    if (snapshot.exists) {
      walletData = WalletData.fromJson(snapshot.data()!);
      return walletData;
    }
    return null;
  });
}

Future<bool> setBalance(String uid, String balance) async {
  walletCollection.doc(uid).update({
    "balance": walletData!.balance + double.parse(balance),
  });
  return true;
}
```

```
Future<bool> disburseFunds(String uid, String amount) async {
  walletCollection.doc(uid).update({
    "balance": walletData!.balance - double.parse(amount),
  });
  return true;
}

Stream<UserData?> getUserProfile(String uid) {
  return usersCollection.doc(uid).snapshots().map((snapshot) {
    if (snapshot.exists) {
      return UserData.fromJson(snapshot.data()!);
    }
    return null;
  });
}

Future<bool> updateProfile(
  String uid,
  String name,
  String email,
  String phone,
) async {
  usersCollection.doc(uid).update({
    "email": email,
    "phone": phone,
    "name": name,
  });
  return true;
}
```