GHANA TECHNOLOGY UNIVERSTY COLLEGE

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

AND

DEPARTMENT OF TELECOMMUNICATION ENGINEERING

TITLE:

DESIGN AND IMPLEMENTATION OF A BUS TICKET RESERVATION PAYMENT SYSTEM USING MOBILE MONEY

A Project Work Submitted in Partial Fulfillment of the

Requirements for BSc. in Computer Engineering and BSc. In Telecommunication Engineering

BY:

EBENEZER ANTWI-BADU (040914012)

KUSI-APPIAH JOSEPH (040914521)

SUPERVISOR

ALFRED ARTHUR

JUNE, 2018

DECLARATION

This project is presented as part of the requirements for BSc in Computer Engineering and BSc in Telecommunication Engineering awarded by Ghana Technology University College. I hereby declare that this project is entirely the result of hard work, research and enquiries. I am confident that this project work is not copied from any other person. All sources of information have however been acknowledged with due respect.

AUTHOR (S): EBENEZER ANTWI-BADU

STUDENT ID: 040914012

SIGNATURES……

DATE: ……

AUTHOR (S): KUSI-APPIAH JOSEPH

STUDENT ID: 040914521

SIGNATURE.....

DATE: ……...

SUPERVISOR: ALFRED ARTHUR

SIGNATURE.....

DATE: .....

HOD: Dr. D.M.O ADJIO

SIGNATURE.....

DATE: .....

HOD: ING. ISAAC HANSON

SIGNATURE.....

DATE: .....

Abstract

The purpose of Bus ticket booking reservation system is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data or information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Bus Ticket Booking System, as described can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on other activities of theirs rather than concentrating on keeping records of their booked ticket. Thus, it will help the organizations in utilizing their resources better. The organization can maintain computerized records without redundant entries. That means there will not be irrelevant information to distract anyone while looking for information.

The aim is to automate the existing manual system by the help of computerized equipments and full-fledged computer system, fulfilling their requirements, so that their valuable data or information can be stored for a longer period with easy accessing and manipulation of the same. More importantly the system allows you to make payments with mobile money which is now a fast and easy way to make payment.

Table of Contents

TABLE OF FIGURES

ACKNOWLEDGEMENT

We would give all thanks and praise to the almighty God for giving us strength and life throughout this project.

We would also use this opportunity to give special thanks to our supervisor Mr. Alfred Arthur for his immense support, guidance and patience. We don't think we would have completed this project successfully if not for you. We are so grateful to you.

Also, we want to thank all other people who supported and contributed one way or the other to a successful completion of the project.

Our overall gratitude goes to all the lecturers of the engineering department and beyond who have imparted great knowledge and offered the best counselling to us during our period of study in this noble institution. We say thank you and hope that you remain our mentors even as we exit the institution.

Of course, we cannot forget those who have been with us all our lives and through this period as well, our families, especially our parents. We thank you so much.

LIST OF ABBREVIATIONS

ACRONYMS	MEANING
HTTP	Hypertext Transfer Protocol
OS	Operating System
WI-FI	Wireless Fidelity
SDLC	Software Development Life Cycle
IDE	Integrated Development Environment
MySQL	My Structured Query Language
XML	Extensible Markup Language
API	Application Programming Interface
JRE	Java runtime Environment
JDK	Java Development Kit
TCP/IP	Transmission Control Protocol/Internet Protocol
ISP	Internet Service Provider
VoIP	Voice over Internet Protocol
IM	Instant message
CPU	Central Processing Unit
DBMS	Database management system
UML	Unified Modeling Language
CSS	Cascading Style Sheet
XHTML	Extensible Hypertext Markup Language
PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
GPS	Global Positioning System
ERD	Entity Relationship Diagram
CRUD	Copy Read Update and Delete
UI	User Interface
RDBMS	Relational Database Management System
SQL	Structured Query Language
MSSQL	Microsoft Structured Query Language
GUI	Graphical User Interface
XAML	Extensible Application Markup Language
I/O	Input/output

CHAPTER ONE: INTRODUCTION

1.1 BACKGROUND OF STUDY

Technology, ever since its introduction, has never seen a setback in its advancement. There has always been major improvement in technology as time goes by. These improvements have become so much that it has changed the world causing humans to encompass it into their lives. The use of devices like smartphones, tablets, and laptop computers for instance has exploded to the point where our whole society is starting to change around it [1].

Today, technological improvement on these devices have given room for advancements in some features and applications which allows the two devices to be used interchangeably. One of these applications is the web browser that has improved the way people gain access to the contents of whatever is been put on the internet. But Smartphones are used by large number of people in the world. According to statista an online webpage associated with statistics, it tells us that in 2014, 1.57 billion people were using smartphones in the world and this value increased to 1.86 billion in 2015 [2].

Again, the Improvement of technology brought about mobile banking, under which the mobile money services can be found. Ever since mobile banking was introduced in the country, mobile money patronage has been increasing rapidly. In August 2016 Mobile money transactions in Ghana reached 679.17 million Ghana Cedis. This represented a 20% growth in the mid-year figure. Notwithstanding, MTN mobile money alone recorded a total of 18.5 million subscribers in 2015 out of which 4.8 million were registered on the service. Most Ghanaians have indeed welcome this mobile banking service [3].

With this great improvement in technology and its acceptance in the country the development of a Bus Ticket Reservation System is appropriate to enable customers of a bus company to buy bus ticket online and pay with their various mobile money accounts. This is an easy method which saves a lot of time, it will enable customers in their comfort zone, to search for available tickets and get their data easily with the help of the browser without any prior experience with queuing at the counter. (Asaad, Ayad and Hayder, 2012) states that an Online

Bus Ticket Reservation System is a web-based application that allows visitors check bus ticket availability, buy bus ticket and pay for the ticket online [4].

E-ticketing is the simplest and fastest way to board a bus. Its main characteristics is to sell and store tickets using electronic devices. Presently, staff at the bus ticket counter are using an internal system to sell ticket at the counter which in a way cost the companies huge sums of money. Customers are unable to buy bus ticket online at this moment and have to go to the counter to buy bus ticket. Sometimes, customer have to join a long queue to buy bus ticket and ask for information. This brings a lot of inconvenience to the customers. In monetary terms, e-ticketing could reduce administrative costs as fewer cashiers are needed, fare processing times could be reduced and a better throughput of passengers could be allowed (Maike, 2014) [5].

1.2 PROBLEM STATEMENT

Passengers waste time at bus stations just to purchase tickets, to enable them embark on a simple trip. Wasting time by going through these long queues can lead to a change in schedule, the person being unproductive and stress.

1.3 OBJECTIVE OF STUDY

The objective of this project is to design and implement a bus ticket reservation system using mobile money. Also, the project will include a chart system that will give the customer a real-time assistance in case of any difficulties.

1.4 SIGNIFICANCE OF STUDY

- It will eliminate long queues at bus terminals (at the counter).
- It will help save time.
- It will help the various telecom operators enhance their mobile money system.
- Customers can check their ticket reservation online and easily make payments.

1.5 METHODOLOGY

The project is going to be developed using the incremental model approach of Software Development Life Cycle (SDLC). This model involves designing implementing and testing the software being developed incrementally until the entire product is finished. The software is decomposed into a number of components (termed as builds) and each of them are then designed, developed and tested.

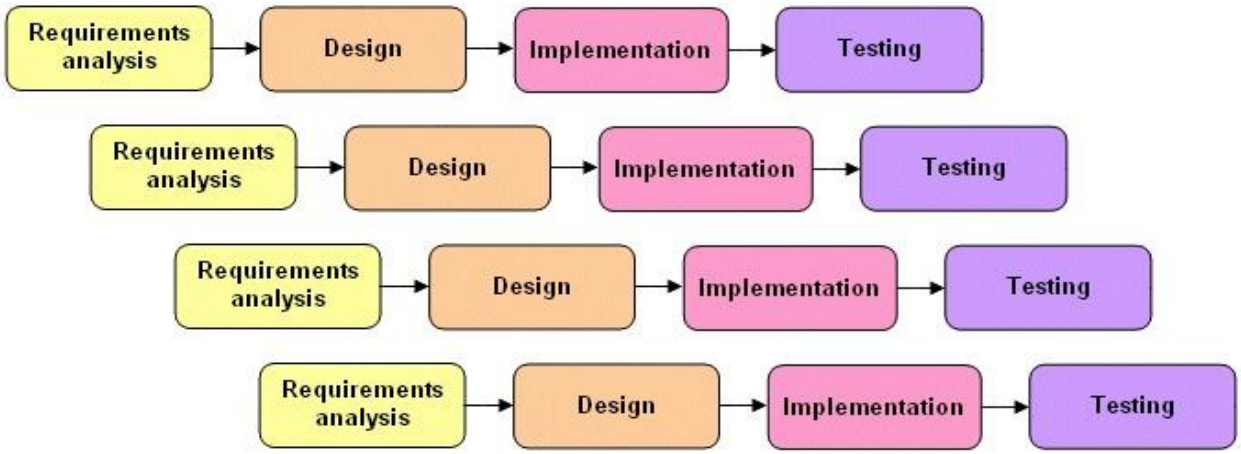


Figure 1 Diagram illustrating the incremental model [6]

1.6 SCOPE OF STUDY

This project is being developed as an android application which will mean without access to your android device and internet connection customers will not be able to gain access to the system.

1.7 ORGANIZATION OF STUDY

Chapter 1 comprises of the background of study, problem statement, objective, significance, methodology and scope of the project work. Chapter 2 presents a literature review on the related project works. Chapter 3 discusses the methodology used in carrying out the project work. Chapter 4 then provides a description of the results, findings and analysis and lastly the chapter 5 talks about conclusions, limitations -and recommendations of the project work.

CHAPTHER TWO: LITERATURE REVIEW

The aim of this project is to design a bus ticket reservation system for customers to be able to make ticket reservations without having to stand in long queues and waste time which will use mobile money as the sole means of payment.

This chapter is concerned with gathering reliable and adequate knowledge pertaining to technologies, methodologies and concepts that will serve as foundation for the project by reviewing a number of works related to the proposed project.

2.1 INTERNET

The internet is at once a world-wide broadcasting capability, a process for information circulation, and a medium for collaboration and communication between individuals and their computers without regards to the location in which they find themselves. The roots of the internet are formed by the interconnection of global independent computers and devices, communications entities and information systems. The global interconnection is made possible by a set of communication standards, procedures and formats that are common among the networks and the numerous devices and computational facilities connected to them. Protocols are the means by which devices are able to communicate with each other. The protocols being used by the internet is the “TCP/IP” (Transmission Control Protocol/Internet Protocol). This protocol is accessed using a modem, broadband, 3G, 4G or a network that is linked through an ISP (Internet Service Provider). For a broadband connection, most computers and devices use WI-FI (Wireless Fidelity) to connect to the router that is connected to the ISP. The internet is filled with lots of information which is meant to be explored, the process of exploring is mostly called surfing and it is made possible by a web browser. Also, the internet allows for communication with others through forums, social networks, e-mails, VoIP and IM (instant message) [7] [8] [9].

2.2 SMARTPHONES

A smartphone is said to be a mobile phone with highly advance features that allows you to do more than just phone calls and sending text messages. Smartphones can browse the internet and run basic software’s just like a computer. A usual smartphone has a high-resolution touchscreen

display for interaction between the user and the device, WI FI connectivity and web browsing capabilities are incorporated in it for accessing the internet and the numerous information that can be found on the web. These devices are just like computers (personal computer or laptop computer), for them to function an operating system is installed on them by the manufactures of the device. There are four main operating systems used by smartphones and they are Android by Google, iOS by Apple, Windows Mobile by Microsoft and Blackberry by Research In Motion.

2.2.1 FEATURES

Smartphones have lots of features, of which one can say is very sophisticated for a small device of its nature. Some of these features will be made mention of.

1. For making and receiving phone calls and text messages.
2. Browsing the internet, sending and receiving e-mails.
3. Making payments online.
4. GPS capability for detecting locations and navigating around.
5. Display of time and date.
6. Display weather and temperature information.
7. Voice control and taking of notes.
8. Virtual assistance (Siri, Cortana, Google talk, Galaxy). Depends on the make of the smartphone.
9. Its ability to run and perform complex tasks. [10] [11]

2.3 COMPUTERS

A computer is an electronic device that manipulates information or data. It can also be defined by others as a programmable machine with two principal characteristics and that is its ability to respond to specific set of instructions in a well-defined manner and being able to execute a prerecorded list of instructions. Computers run on operating systems. The two main types of operating systems are Windows and Mac OS by Microsoft and Apple respectively. In the light of this general purpose computers requires both hardware and software to operate, hardware components such as the memory, for it to store temporal data for programs, storage device to allow for permanent retainment of large amount of data, input and output devices which helps the user

to interact (send information to the computer and receive information in a visual form) with the computer and finally the CPU (central processing unit) which is mostly termed as the brain of the computer, this executes all instructions in the computer. These hardware's spoken of cannot function without a software. With this it can be said a software is a set of instructions that tells the hardware what to do and how to do it. Computers can be classified into various types; these classifications are based on the size and power of the computer. [12] [13]

2.4 MOBILE MONEY IN GHANA

Majority of Ghanaians have embraced the mobile money system rather than the usual banking system. In the year 2010 an article written by Kwami Ahiabenu indicated that an estimated 80 percent of Ghanaians are "unbanked" – meaning they conduct their transactions outside the banking sector with no access to financial services. Products like "mobile money," that enable safe and secure money transfers without the use of a bank account, could have a major impact on this unserved segment of the population. Mobile money gives anyone with a mobile phone the ability to transfer money, make cash payments and conduct other financial transactions over the phone. [14]

In addition, one amazing notice about the mobile money in Ghana is that as time passes more and more people subscribe to it. Patronage of mobile money continues to gain momentum, as the value of transactions for the third year running saw an astronomical jump, from GH¢2.4 billion as of 2013 to about GH¢11.6 billion in 2014, according to the Bank of Ghana (BoG). This means the number of transactions has almost quadrupled since 2012; from 30 million to about 106.4 million in 2014. Registered mobile money customers have also increased from 3,303,837 in 2013 to 5,424,650 in 2014, representing a 64- per cent increase the total number of subscribers also increased from 20,346,016 in 2013 to 21,721,814 in 2014. [15]

Mobile money transactions, given how increasingly people subscribe to it, will be always more than the original banking transaction and so probability of the need of this mobile transaction will always be high in Ghana. It will also go a long way to help the country evolve into a cash free country. The new technologies such as mobile money are laying the foundation for a cashless future for the Ghanaian market (Kwami Ahiabenu,ll, 2015). [16]

2.5 REVIEW OF RELATED PROJECT WORKS

2.5.1 DEVELOPMENT OF AN ONLINE BUS TICKET RESERVATION SYSTEM FOR A TRANSPORTATION SERVICE IN NIGERIA. (OLOYEDE, M.O., ALAYA S.M. ADEWOLE, K.S 2014) [17]

According to Oloyede, M.O., Alaya S.M. and Adewole, K.S; the work is to create a web-based application that allows visitors check bus availability, buy and pay bus tickets online. With this main motive the work is aimed at easing the issues of bus reservation amongst people that stay within the area of study (Nigeria). These issues include time wasting in long queues for tickets as a result of robust and outmoded ticketing system. The proposed work has three models that is; helps enquire the availability of seats in a given bus at a particular date, reserve tickets and lastly cancel a reserved ticket. The system also gives the customers the flexibility to pay for the ticket online.

The architecture of this proposed system supports existing infrastructure requirements and is represented using a three-tier architecture that compromises of user interface process management and Database management system (DBMS) this shows the components of the system, the service they provide and how they communicate to form the system functionality. A UML (Unified Modeling Language) case diagram is used to shoe and give a detailed information of the bus reservation system and the various participants involved in this system. This diagram consists of two participants the user and the administrator. The activities carried out by these two comes together to form the bus reservation system. Some activities of the user are

1. Being able to search for seats.
2. Signing up or registering with the system.
3. A registered user can login to the system.
4. The user can check for available seats.
5. The user can make payments for the seat on the system.
6. Receipts can be printed from the system by the user.

The administrator also carries out these activities

1. Verifying all registered users and granting them access to login to the system.
2. To acknowledge any payment made by the user on the system.
3. Also add vehicles, drivers and generate reports.

All this was made possible in the project because it was developed with JavaScript, XHTML, PHP, CSS, and MySQL.

Advantages

- It saves time and prevents long queues at the bus station.
- Payments are made easily with credit cards online.

Disadvantages

- Only registered customers can make reservations and payment with the system.
- Internet connection is constantly needed.
- Payment methods is only restricted to credit card

2.5.2 DESCRIPTION WITH UML HOTEL RESERVATION SYSTEM. (LOUW, DOOR JANNE, 2006, MAY 10,2006) [18]

Louw and Door Janne, indicated that the project has been developed as a hotel management system that can be used online. This system allows guests to book rooms online. Some of the tasks the system can do is providing a query for arriving date and the length of staying, providing the number of On rooms, view all available rooms and provides user the ability to choose one or more of them, recording the number of On rooms, recording kind of guests and how many are going to be in the room, providing the cost of booking, asking the users if they want additional service; such as, dinner or breakfast and also storing the guests detail; like, name, address and telephone, asking the user for confirmation, final confirmation views with the detail of booking and the guests can review or cancel the booking.

For more understanding to the system the author provided some figures with explanation. The first figure shows the relationship between the end user and the web server and how the users

interface stats and the application is done step by step. The second figure shows the relationship between the user and the screen. The screen transfers HTML codes to interface and when the user interacts with it some processes are done then, the screen shows another page. While the third figure shows how each page is related to each other.

The user can go to the next page by three ways. The first one is the user after inserting his/her information goes through pages in sequence way. The second is to navigate, and this way provide the user the ability to go the previous pages or to the login page. The final one is, once the user enters unaccepted or wrong details, the current page is maintained.

This article is almost a UML design. JAVA is one of the suggested tools for building this system. This system common with our system in with some of reservation process; such as storing the customer information and choosing the payment method. Also, this system used JAVA as our system but it differs in the kind of user, this system is used online by the guest while our system is used locally by the employee of the hotel.

Advantages

- It saves time and prevents long queues.
- Payments are made easily online.

Disadvantages

- Only registered customers can make reservations and payment with the system.
- Internet connection is constantly needed.
- Payment methods is only restricted to credit card.

2.5.3 ONLINE BUS BOOKING SYSTEM (BAROT ALOK T., DAYANI YASH K. 2014) [19]

An online bus booking system is to be developed in order to aid customers in making bookings of their bus tickets, cancel the reservation they have made and also providing them with the necessary information about the bus. This is what Barot Alok T. and Dayani Yash K. intends to make their project to do. This project is a web application that will be developed with Asp.net with C# Framework 3.5 and SQL server 2005. Asp .net with C# Framework will be used for the Front end and SQL server 2005 will be used for the back end. The work made it known to us that the current

system being used is tedious; in the sense that one has to come to the office of the bus company in order to be able to get a ticket causing long queues at the office and also wasting time. It goes on further to make us understand the workers of the company also have to stand or sit for long hours because the system has to be operated manually.

With this the project emphasizes strongly on putting a stop to the old system and also bring on board a login and password service for the employee and the regular customer to use this system. It has two of specification and these are the functional and module specification. The functional specification talks about what the user and the administrator will be able to do whenever they gain access to the web application. The module specification serves as the backbone of the application. This is because it provides the user and the administrator with the necessary information and the guide lines on how to go about their various functions when they are using the web application. The project provides a flow chart and a user case diagram which is used to show and breakdown the methodology used in developing the work. It takes every process and gives a diagrammatic representation of it which makes the it easier to understand how the work will go about.

Barot Alok T. and Dayani Yash K. indicated that the work could be improve further by making a mobile application of it, adding a GPS locator to get an advanced mapping system for the route during the journey.

Advantages

- It saves time and prevents long queues at the bus station.
- Payments are made easily with credit cards online.

Disadvantages

- Only registered customers can make reservations and payment with the system.
- Internet connection is constantly needed.
- Payment methods is only restricted to credit card.

2.6 SUMMARY

Upon reviewing the related works, it has brought to light that online reservation and payment systems have made it easier for making any type of online purchases and payment which ends up saving lots of time. But limits the mode of payment to only credit cards.

The review further went on to make us know that both web-based applications and mobile applications constantly need connection to the internet to be able to make, check and pay for the reserved ticket.

Additionally, we got to know from the review that Java, SQL, php and HTML were the best environments or IDE to be used in designing a web-based application while other environments or IDE like Asp.net with C# Framework are the best for building mobile applications

However, the review shows that the project has two user interfaces one for the customer and visitor of the application and the other for the administrator. Both sides provide information to help assist each other, so any false information provided by these users makes the application inefficient by given wrong information.

The review also pointed out that the requirements for developing an application is very necessary as it forms the foundation for every application.

Lastly, there are issues always regarding the use of either database or shared preference technology to share data. But knowing the amount of data the application is going to deal with will help select the most appropriate technology needed.

CHAPTHER THREE: METHODOLOGY

This chapter will elaborate on the stages through which the mobile application had to go through before been developed. We will discuss the software development method used in the creation of the project which is the incremental model. The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- It will ensure data accuracy.
- Records will be efficiently maintained by DBMS.
- Availability of seats can be enquired easily.
- Passengers can also cancel their tickets easily.
- Minimum time needed for the various processing.
- It will provide better Service.

Shuchi, G. (2008) stated that system design is to create a technical solution that satisfies the functional requirements for the system. [20] At this point in the project life cycle there should be a Functional Specification, written primarily in business terminology, containing a complete description of the operational needs of the various organizational entities that will use the new system. The challenge is to translate all of this information into Technical Specifications that accurately describe the design of the system, and that can be used as input to System Construction. To be able to get a clear picture of the development process we will divide this chapter into four main parts and they are as follows;

- An overview of the Incremental Model
- Requirement Analysis
- Design Application
- Implementation

3.1 INCREMENTAL MODEL

Incremental build model is a process in the software development where the product is designed, implemented and tested incrementally (a little more is added at each step) until the product is finished. Development and maintenance is needed.

This model of system development life cycle was used to develop the application. Its emphasis is placed on the documentation in detail as compared to the recent software development methodologies. The model allows someone to review the system's documentation and get a very good understanding of the steps taken to complete the development of the project. This model is simple to use and allows for easy fault detection during development.

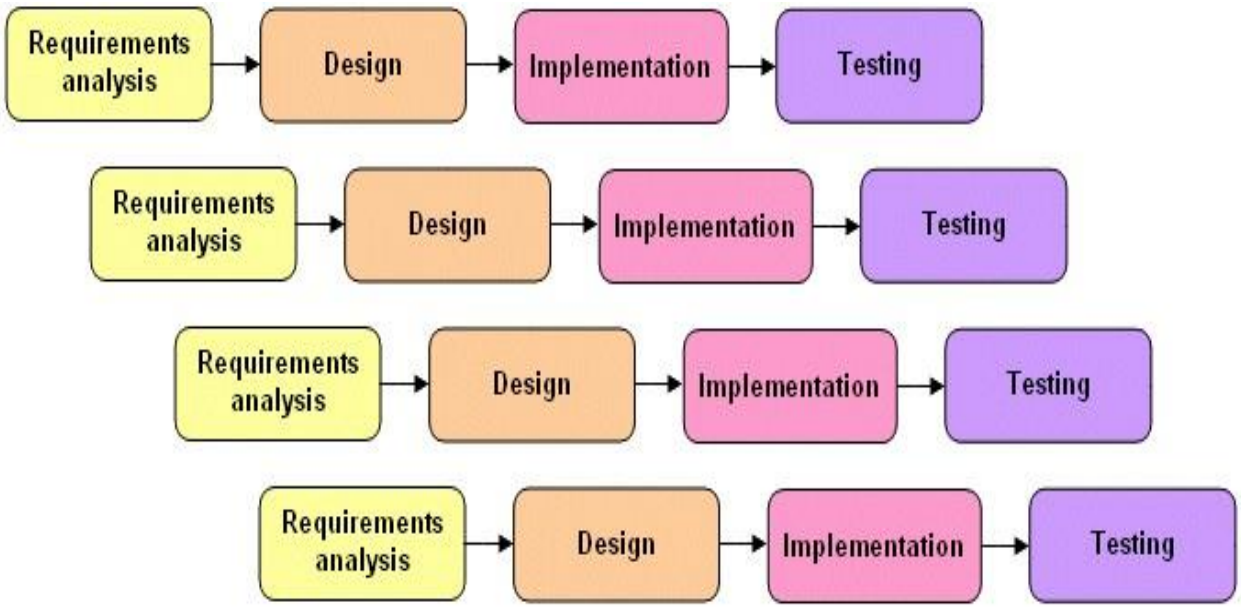


Figure 2 Diagram illustrating the incremental model [21]

3.2 REQUIREMENTS ANALYSIS

Requirements analysis can also be known as requirements engineering. And it is the process of determining the user expectations for a new or modified product. These requirements must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.

The critical analysis and examination of both functional and non-functional requirements is termed as requirement analysis. Functional requirements describe the services expected of the system or its components. Non-functional requirements are those that are not directly concerned with the deliverables of the system.

Functional Requirements of the System

Functional requirements define the specific functions that the system performs, along with the data operated on by the functions. The functional requirements are presented in scenarios that depict an operational system from the perspective of its end users. Included are one or more examples of all system features and an enumeration of all the specific requirements associated with these features.

- The system shall incorporate a mechanism to authenticate its users.
- The system shall verify and validate all user input and should notify in case of error detection and should help the user in error correction.
- The system shall allow sharing of files in the system.
- The system shall allow quick messages to be exchanged without face to face interaction.

Non-Functional Requirements

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services [22].

Non-functional requirements state how well the system does what it is supposed to do. That is, performance is an attribute of function.

These requirements are those that specify criteria that can be used to judge the operation of a system as instead of specific behaviors. These are the qualities the system has:

- Platform Compatibility: The application is designed to run on any android smartphone platform making it android machine independent.

- Availability: This is how available the application is to the user. The user end of the application requires internet connection before it can be accessed. Without internet connection, the application is rendered useless.

- Performance: The application will undergo performance-testing programs that will be analyzed to see if the application can get users to book and pay for their tickets successfully. Good hardware and software specifications coupled with a working Internet connection and good working data will also enhance the performance of the application.

3.2.1 HARDWARE REQUIREMENTS

Computer

The development computer must have the basic Input/output components and the following as well.

- Processing speed of 1 GHz or faster.
- RAM of 1 GB or faster.
- 20GB disk space free space.

Smartphone

The phone that will run the application should have the following features.

- RAM of 128MB or faster.
- Free storage of about 25MB and above.
- Android operating system.

3.2.2 SOFTWARE REQUIREMENTS FOR SERVER-END

- The computer to be used should run on Windows XP, Vista, 7, 8,8.1 or 10, Linux or Mac OS.
- The computer must also have a web browser installed with an Internet connection.
- The web browser should have HTML capability.

3.3 APPLICATION DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.

This section of the methodology determines and documents how we intend developing the project. We define the components, interfaces, data, modules and architecture for the system to the specified requirements. The section is divided into the following phases:

- System architecture
- Flow Chart diagram
- Entity Relationship Diagram (ERD)

3.3.1 SYSTEM ARCHITECTURE

The diagram below shows the architecture of the project. The database stores the details of the users and administrators on the platform.

The ASP.NET web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. It is an ideal platform for building restful applications on the .NET Framework and serves as the entry point of all requests. Whenever it receives a request, it validates it to ensure that it is of the required format before responding to it. All CRUD (Copy Read Update and Delete) processes of the database is handled by a module called the database layer. All implementation of the logic (Login, saving and retrieving of information from the database) of the server is done by another module called the controller.

The application communicates with the database through the ASP.NET using the Jason Language and the method of information transfer is by HTTP. Once the user launches the application, a request is sent to the java server to retrieve the user details from the database. The server then fetches the requested information and stores it in a session. The application can now be used by the user. Also, when an administrator logs in, the server communicates with the database using the same procedure.

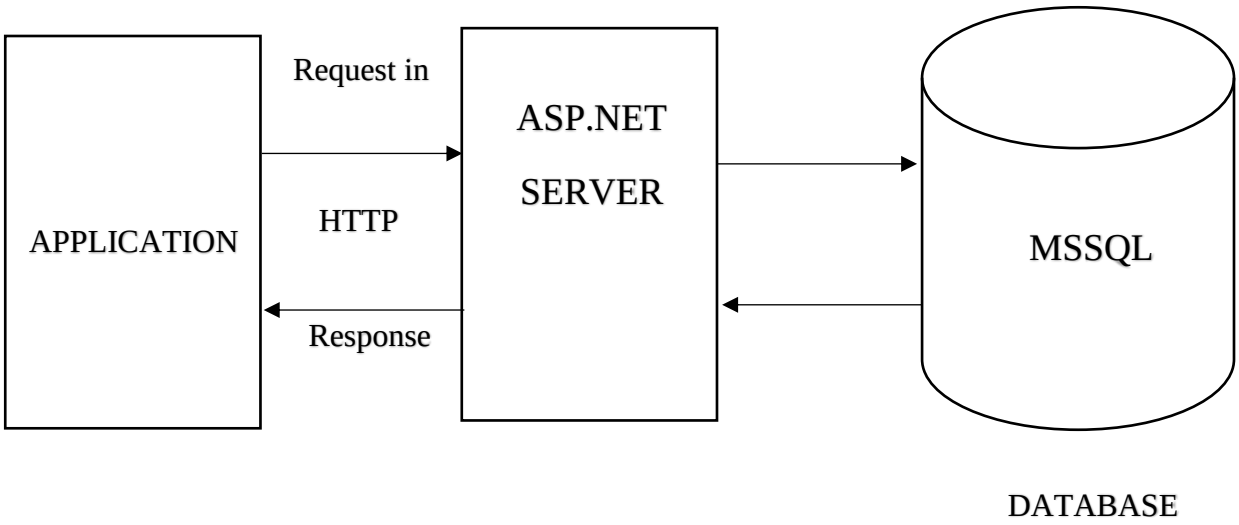
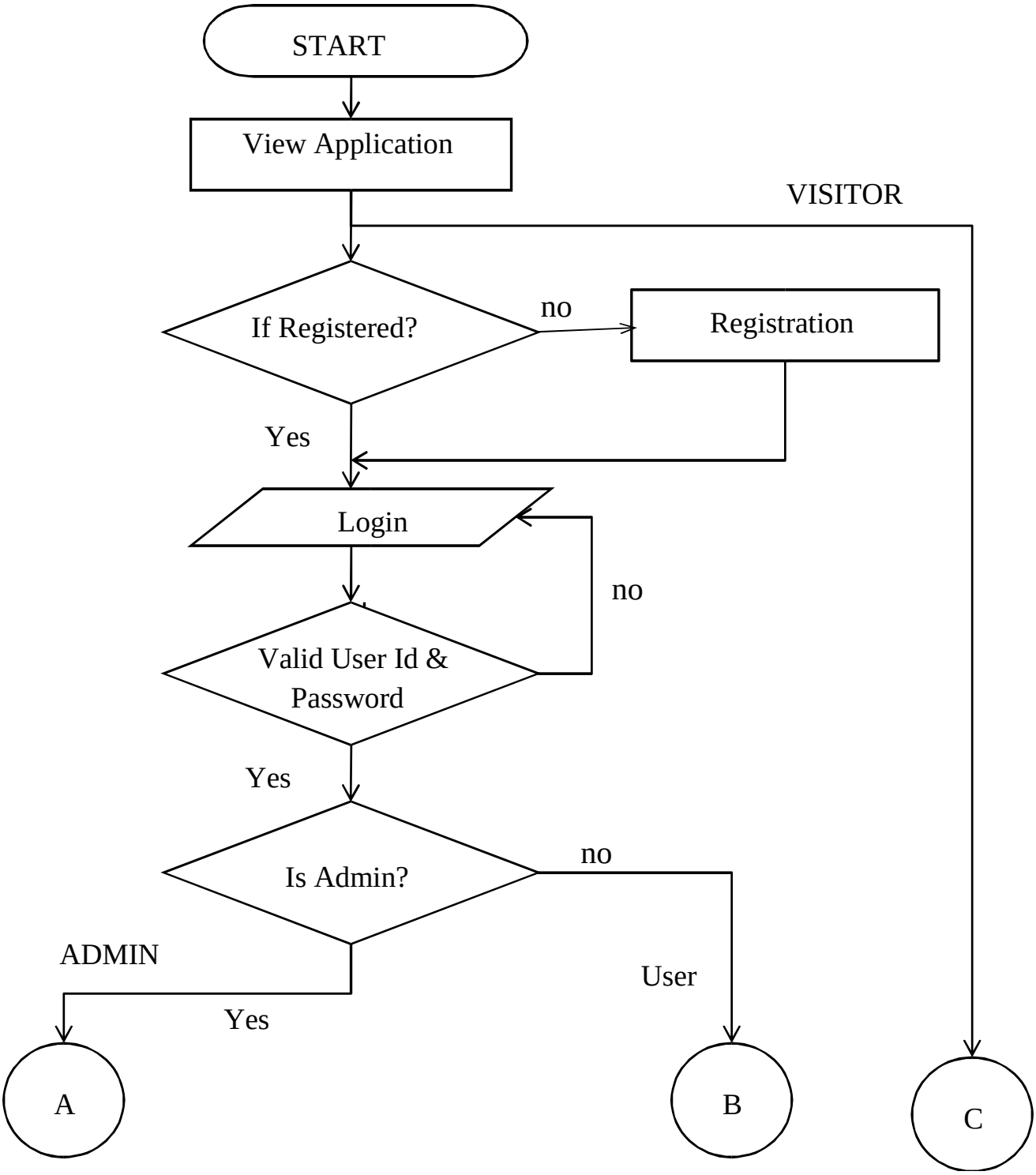


Figure 3 System architecture of application

3.3.2 FLOW CHART DIAGRAM

At this stage the flow chart diagram will give us a summarized idea of what this application will be doing. The application first takes you to the login page. If you have already registered you can login to the system else you will be made to register before logging in. During the login process, the system verifies all usernames and passwords to know if they can be found in the system's database. If there are no valid matchings within the database, it prompts you to reenter your user id(username) and password again. After logging in with a valid username and password the system determines if the credentials are that of a user(customer) or that of an administrator. If it is that of an administrator the system allows to update bus information, view booking details, receive payments information, manage comments and generates reports. After it has successfully performed all these functions it allows the administrator to logout and close the application. For the user, the system allows to view bus information, book the ticket, cancel the booking ticket and post comment. The user can then logout after all these functions are successfully performed and can close the application.


START
View Application
VISITOR
If Registered?
no
Registration
Yes
Login
no
Valid User Id &
Password
Yes
Is Admin?
no
ADMIN
User
Yes
A
B
C

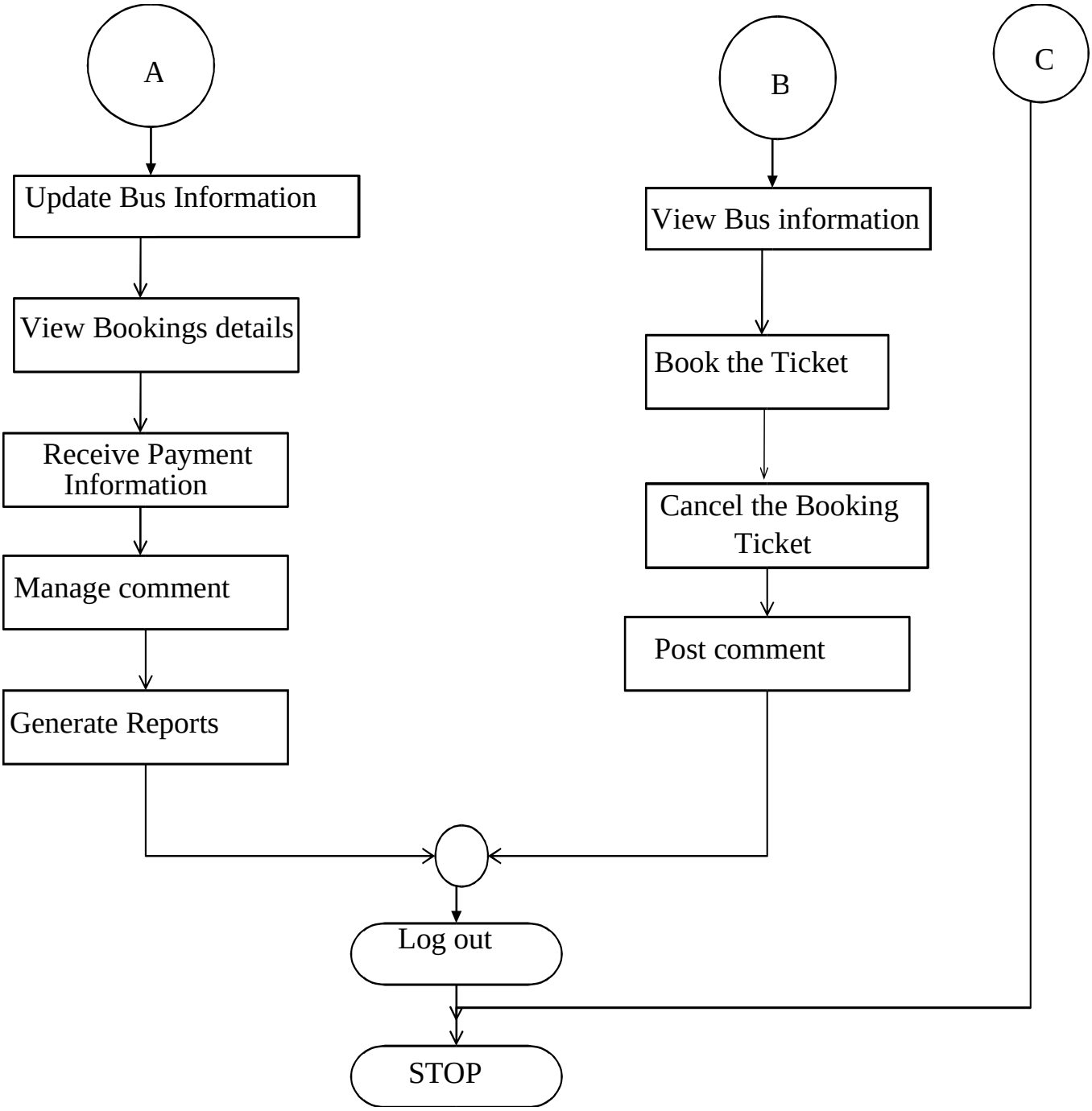


Figure 4 Flow chart diagram of the application

3.3.3 ENTITY RELATIONSHIP DIAGRAM

An Entity Relationship(ER) model defines the conceptual view of a database. Visually the ER-model is the best method for designing a database. The E-R model describes the database as entity, attribute and relationships.

An entity represents a real-world object or concept, such as a user, as shown in the database. They are represented as the rectangle in an E-R model. All entities have attributes.

An attribute represents some property of interest that further describes an entity, such as the user's name or e-mail, age, phone number and so on. In an E-R model they are identified as oval shapes. There are lines that connects attributes to the corresponding entity. Some attributes are unique and so they are used to identify entities. These attributes are called the primary keys and, in an ER-model they are identified as oval shapes but with underlined text.

A relationship among two or more entities represents an interaction among these entities. This relationship may be to update, to add and so on. The diamond shape represents relationships in an E-R model. The diagram below shows an E-R model of the database of the application.

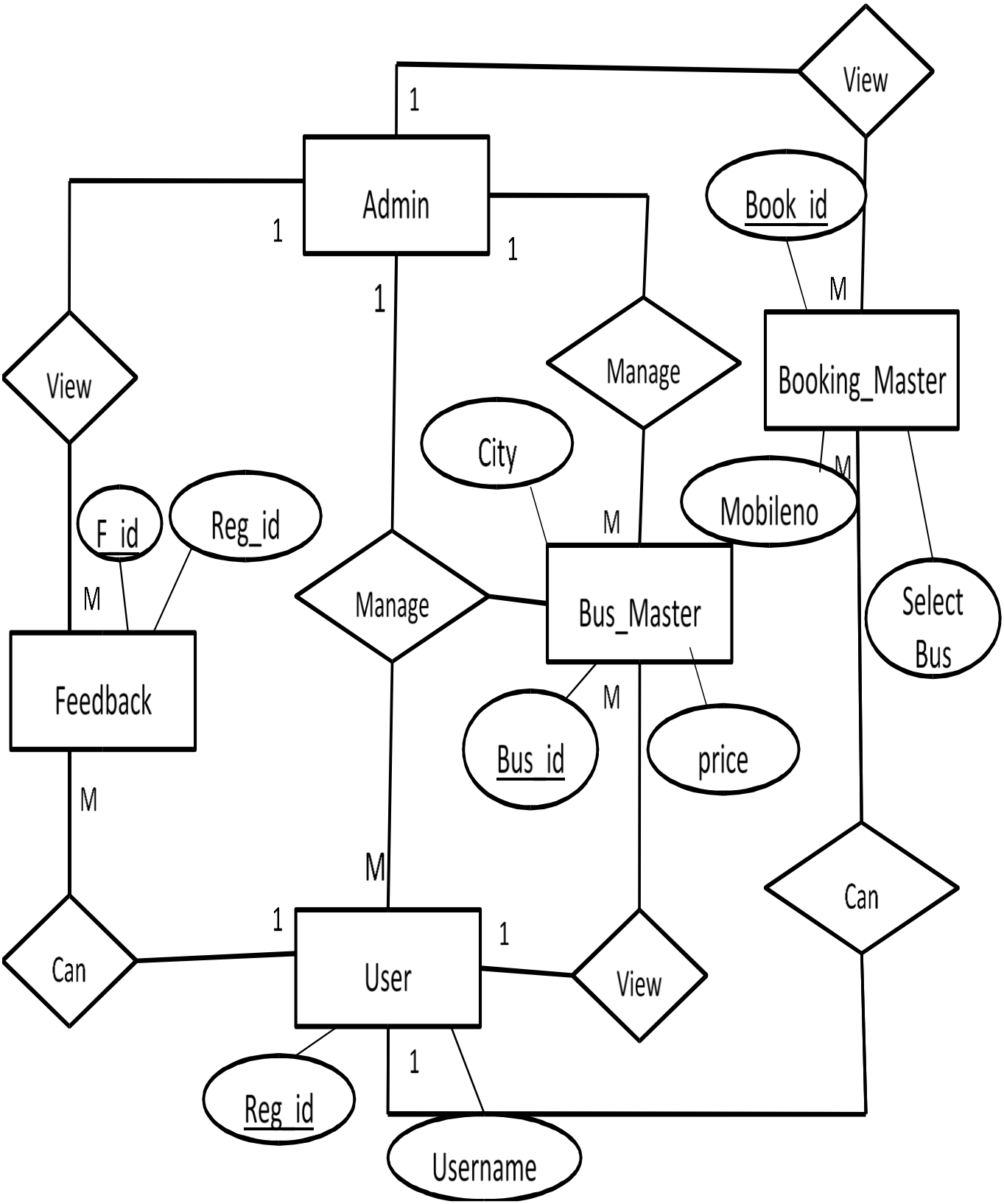


Figure 5 Entity Relationship Diagram of database

3.4 IMPLEMENTATION

3.4.1 AN ACTIVITY

“Activity” is the name given to a single application window with which the user can interact at any given time [23] For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched - the main activity [24].

Any time a new activity starts, the previous one is stopped, and preserved in a stack (the "back stack").

When a new activity starts it pushes onto the back stack and has user focus. The back stack abides to the basic "last in, first out" stack mechanism, so, when the user is done with the current activity and presses the Back button, it is popped from the stack (and destroyed) and the previous activity resumes.

When an activity is stopped because a new one starts, it is notified of this change in state through the activity's lifecycle callback methods. There are several callback methods that an activity receives due to a change in its state. The callback method includes onCreate (), onPause (), onStop (), onResume (). Each callback provides you the opportunity to perform specific work that's appropriate to that state change. For instance, when stopped, your activity should release any large objects, such as network or database connections.

A resumed activity can reacquire the necessary resources and continue with the interrupted actions. The following figure shows the lifecycle of an activity of a typical android application from when an activity is invoked by a user to when it is paused, when it goes into the background, some decisions made by the android operating system which assigns priorities to applications in the background depending on their memory requirements among other factors and eventually the destruction of the activity which can signify the end of the android application if there are no other activities to be run.

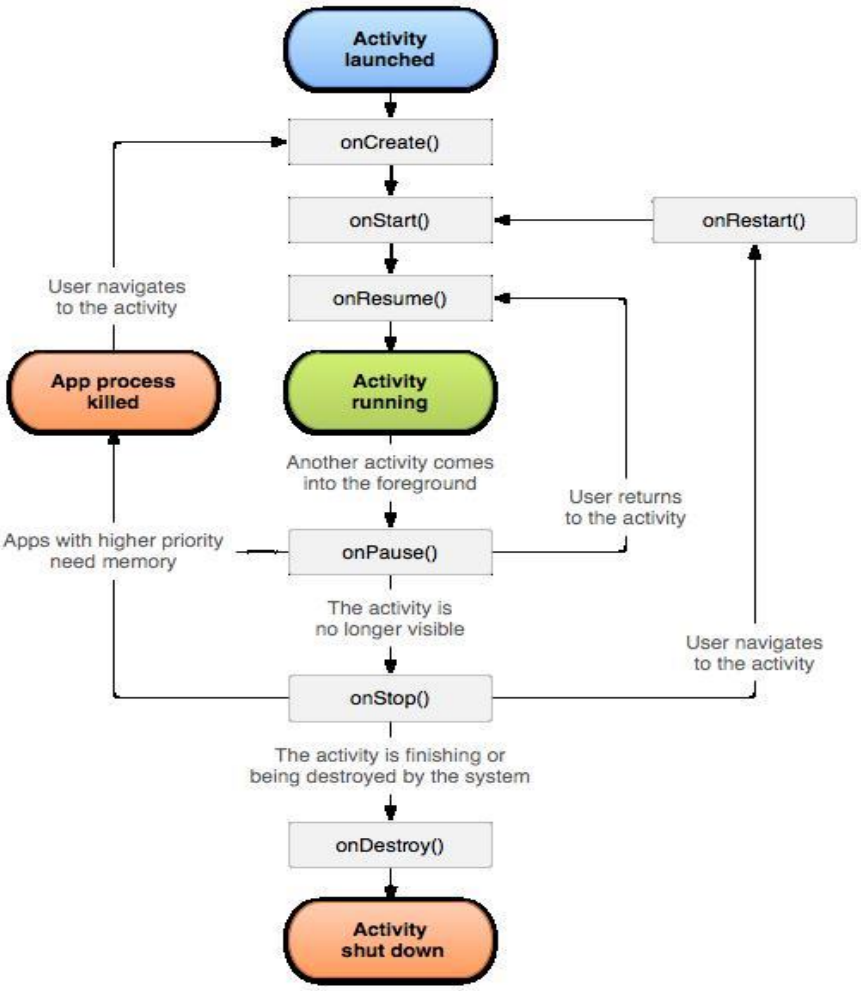


Figure 6 Life Cycle of an Activity [18]

The following describes the callback methods received when there is a change of state.

onCreate ()

This activity is called when the system creates an activity. This is where you should do all of your normal static set up: For example, your app should create views and bind data to lists here. Most importantly, this is where you must call setContentView () to define the layout for the activity's user interface. When onCreate () finishes, the next callback is always onStart ().

onRestart()

Called after your activity has been stopped, prior to it being started again. Always followed by onStart ()

onStart()

As onCreate() exits, the activity enters the Started state, and the activity becomes visible to the user. This callback contains what amounts to the activity's final preparations for coming to the foreground and becoming interactive.

onResume()

The system invokes this callback just before the activity starts interacting with the user. At this point, the activity is at the top of the activity stack and captures all user input. Most of an app's core functionality is implemented in the onResume() method. The onPause() callback always follows onResume().

onPause()

Called as part of the activity lifecycle when an activity is going into the background but has not (yet) been killed. When the system calls onPause() for an activity, it technically means the activity is still partially visible, but most often is an indication that the user is leaving the activity, and the activity will soon enter the Stopped or Resumed state. An activity in the Paused state may continue to update the UI if the user is expecting the UI to update.
Once onPause() finishes executing, the next callback is either onStop() or onResume(), depending on what happens after the activity enters the Paused state.

onStop()

The system calls onStop() when the activity is no longer visible to the user. This may happen because the activity is being destroyed, a new activity is starting, or an existing activity is entering a Resumed state and is covering the stopped activity. In all of these cases, the stopped activity is no longer visible at all. The next callback that the system calls is either onRestart(), if the activity is coming back to interact with the user, or onDestroy() if this activity is completely terminating.

onDestroy()

The system invokes this callback before an activity is destroyed. This callback is the final one that the activity receives. onDestroy() is usually implemented to ensure that all of an activity's resources are released when the activity, or the process containing it, is destroyed.

3.4.2 DEVELOPMENT TOOLS

MSSQL

An open source relational database management system (RDBMS) based on Structured Query Language (SQL) – one of the most popular languages for adding, accessing and processing data in a database, and runs as a server providing a multi-user access to a number of databases.

MSSQL is developed, distributed, and supported by Microsoft Corporation. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MSSQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications. A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and ―pointers‖ between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of ―MSSQL stands for ―Structured Query Language. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, ―SQL-92 refers to the standard released in

MSSQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MSSQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MSSQL can also scale up to clusters of machines, networked together.

MSSQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MSSQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MSSQL Server highly suited for accessing databases on the Internet.

Android Studio

There are so many Integrated Development Environments (IDE) for the development of an android application but we chose to use the android studio as it is recommended by google for the development of android applications. One should have installed Java Development Kit (JDK) or Java Runtime Environment (JRE) before he or she can start using the android studio. Android studio is an intelligent code editor capable of refactoring,

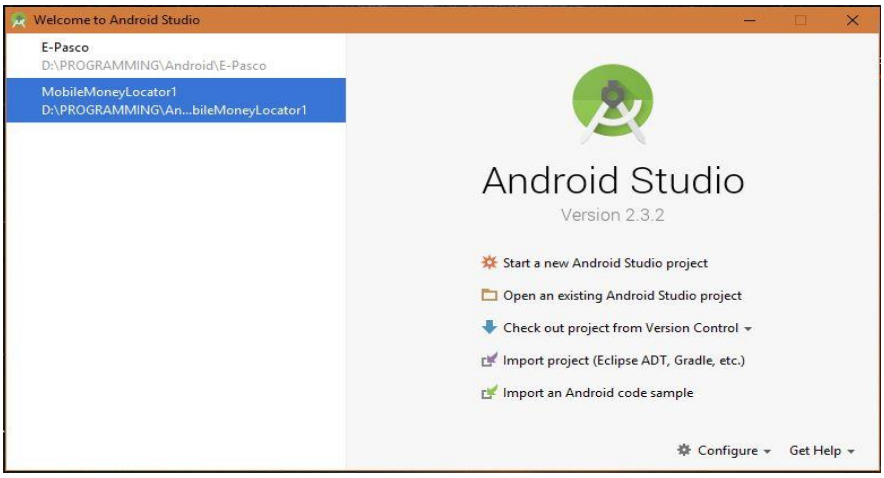


Figure 7 The Android Studio IDE [self-captured]

Microsoft Visual Studio

It is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

The most basic edition of Visual Studio, the Community edition, is available free of charge.

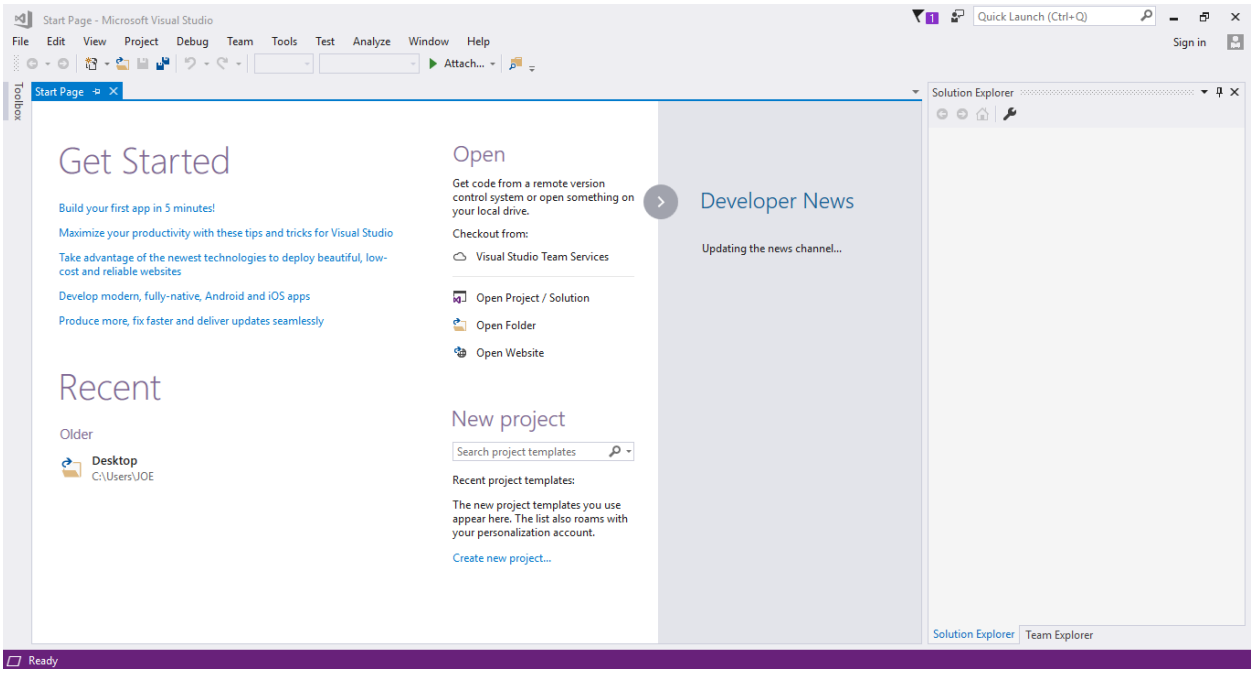


Figure 8 Microsoft Visual Studio[Self-Captured]

ASP.NET

The ASP.NET web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. It is an ideal platform for building restful applications on the .NET Framework and serves as the entry point of all requests. Whenever it receives a request, it validates it to ensure that it is of the required format before responding to it. All CRUD (Copy Read Update and Delete) processes of the database is handled by a module called the database layer. All implementation of the logic (Login, saving and retrieving of information from the database) of the server is done by another module called the controller.

Figure 9 ASP.NET

Xamarin

Xamarin is a cross-platform development tool. It solves dilemmas many developers face when developing cross-platform apps: separate coding languages and UI paradigms. With Xamarin, you can use C# to develop iOS, Android, *and* Universal Windows apps. And with Xamarin Forms, interface design for all three platforms can be accomplished within its XAML-based framework

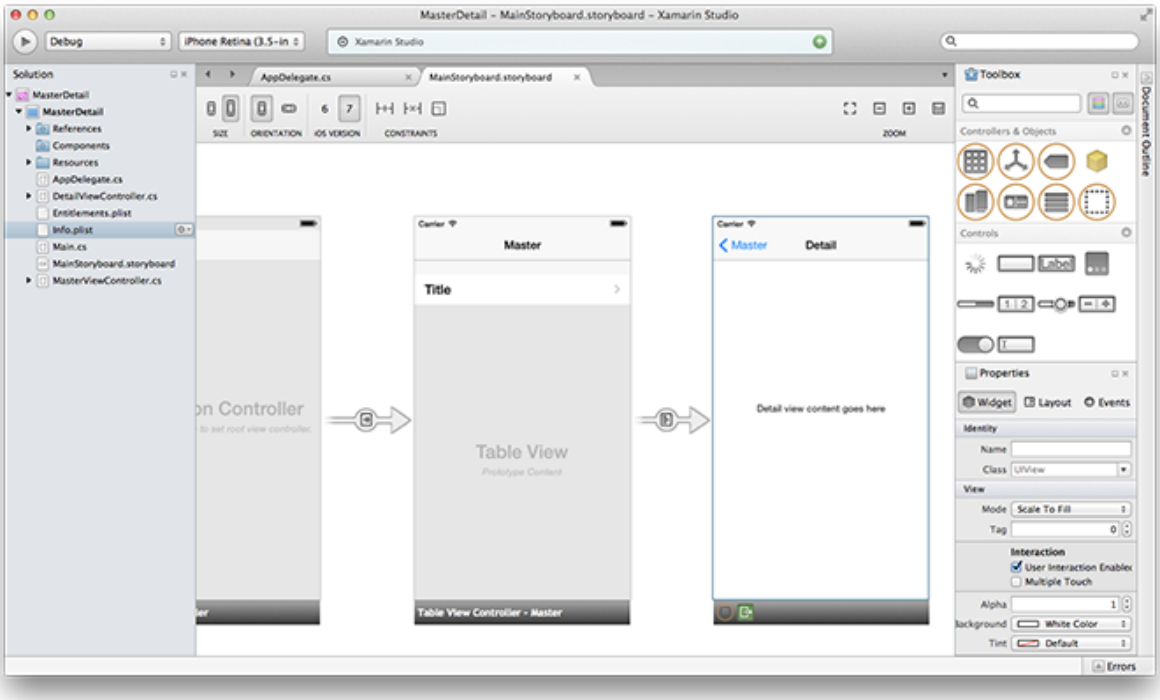


Figure 10 Xamarin [Self-Captured]

CHAPTER FOUR: TEST RESULTS AND ANALYSIS

4.1 OVERVIEW

This section describes the implementation and testing the bus ticket reservation application on a real android system. We also conducted performance experiments in terms of accuracy when it comes to making payments across various telecommunication networks on an Android mobile phone.

4.2 TESTING RESULTS AND ANALYSIS

The application is made up of two main modules or platforms,

- Back End: This contains all the codes that were used in building the application and the database that contains all the information of all the customers registered. The Administrator adds buses, routes and also deletes buses that are not in function anymore from the database.
- The User End: This is the interface that the normal user will be able to view. Once the application is being installed and opened, the user will view a login page. If the user is registered on the system, login is successful and he or she is redirected to the user homepage. On the other hand, if the user is new to the system he or she clicks on the registration link to sign up.

4.3 DESCRIPTION OF KEY FEATURES

4.3.1 FEATURES OF APPLICATION

When the application is installed and launched, it opens a login page where the Administrator and the user can login. The user can also reset his or her password using the "**Forgot Password**" link. It opens a page where the user will have to enter his or her email address for verification and the password will be sent straight to the email address entered.

On this same page there is a registration link captioned "**Haven't registered yet?**" which redirects a new user to the registration page. Here, the application demands of the user to enter a Username, First Name, Last Name, Email, Phone, Password, Confirm Password to be able to register.

Therefore, the application has got both Administrator and User access.

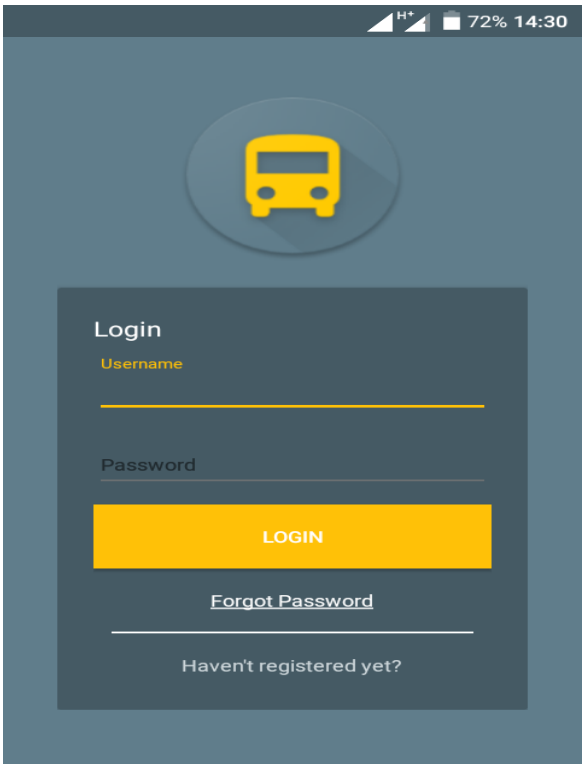


Figure 11 Login page for both user and admin

Admin Access

Upon logging in successfully as the Administrator, a page is displayed where there is a menu button that shows four options and they are as follows:

- Home
- Buses
- Reservation/Tickets
- Feedback

There is a **Signout** button which allows the Administrator to logout from the application. The application permits the Administrator to perform changes in profile and also read feedbacks.

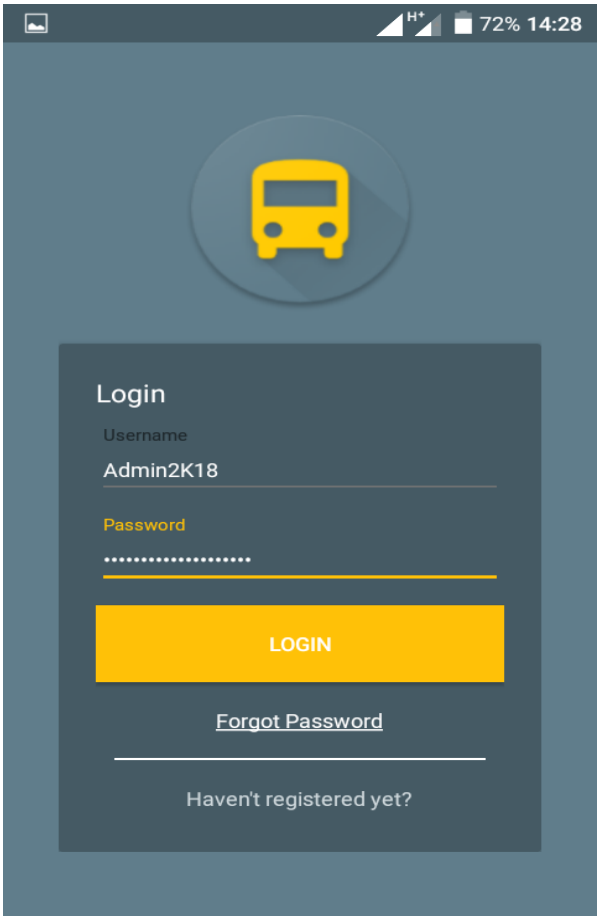


Figure 12 Login details for admin

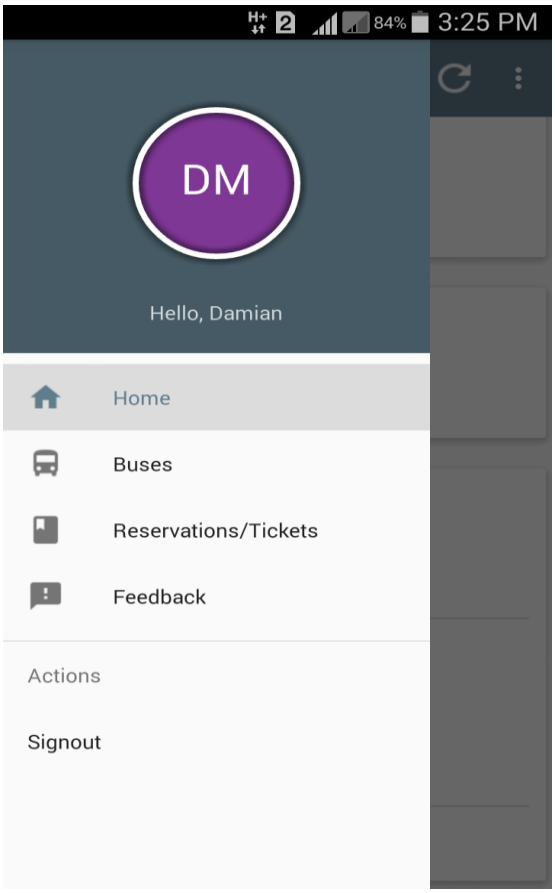


Figure 13 Home screen of the admin with drop down menu

Home View

The Home View shows the total amount of money received based on daily and weekly reservations, the number of buses available, the journey routes and the number of users on the system.

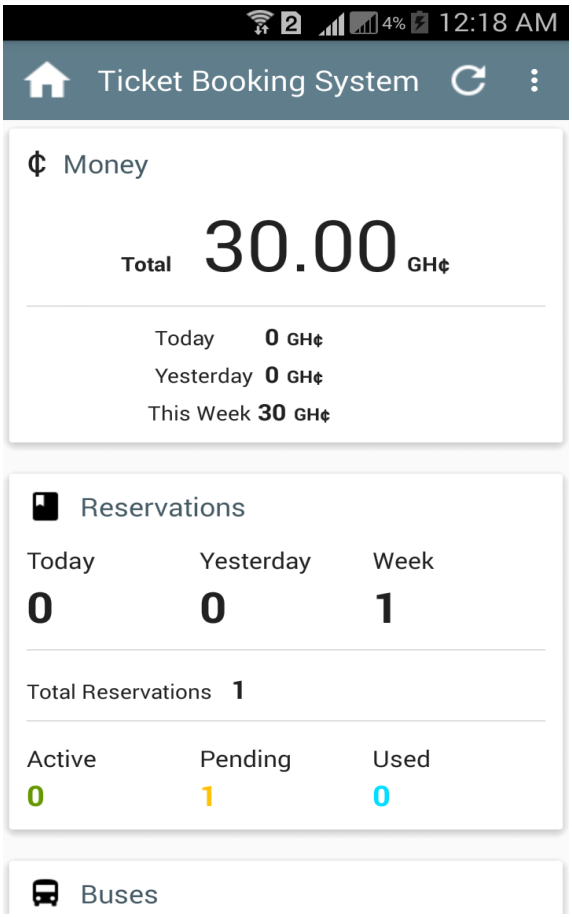


Figure 14 Home screen of the admin

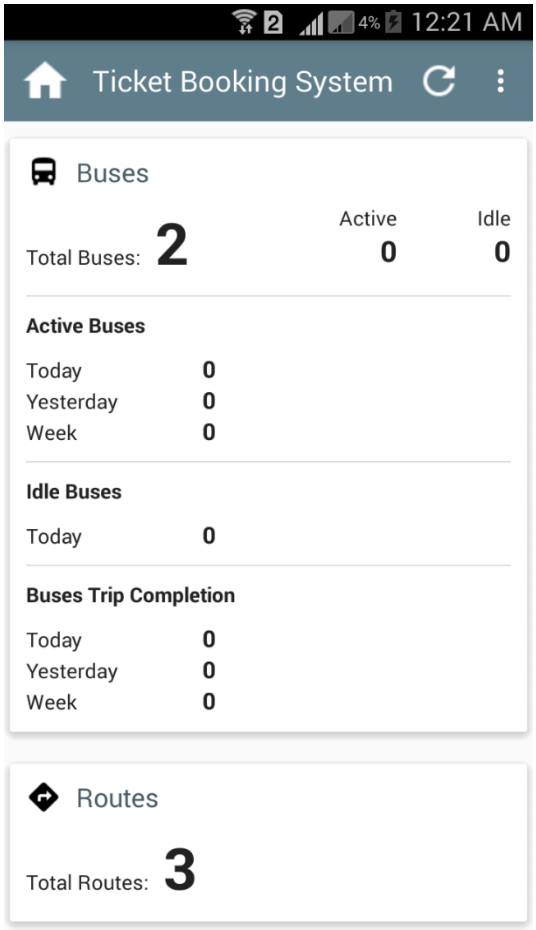


Figure 15 Home screen of the admin

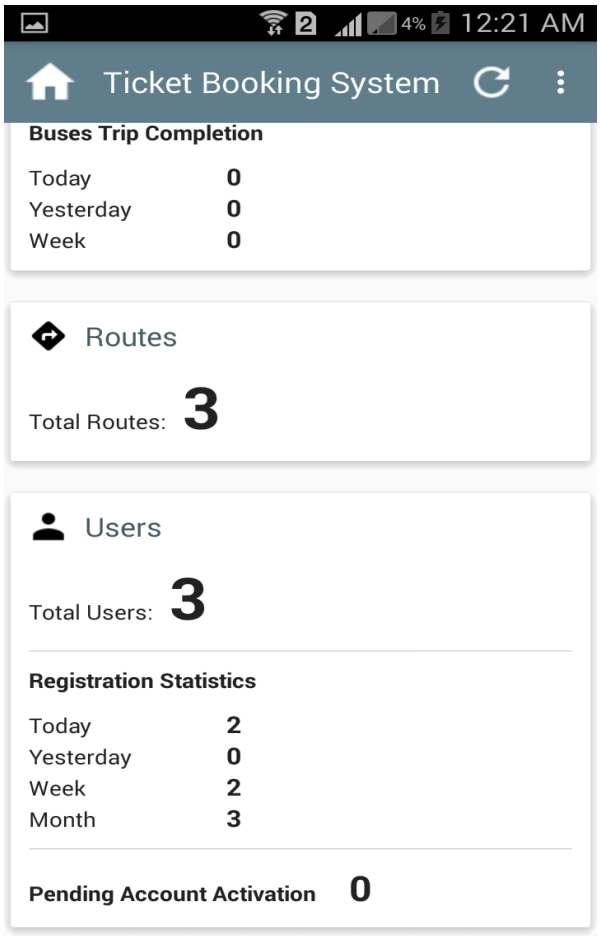


Figure 16 Home screen of the admin

Buses

This section allows the Administrator to add buses and journey details. The Administrator can also delete inactive bus from the list.

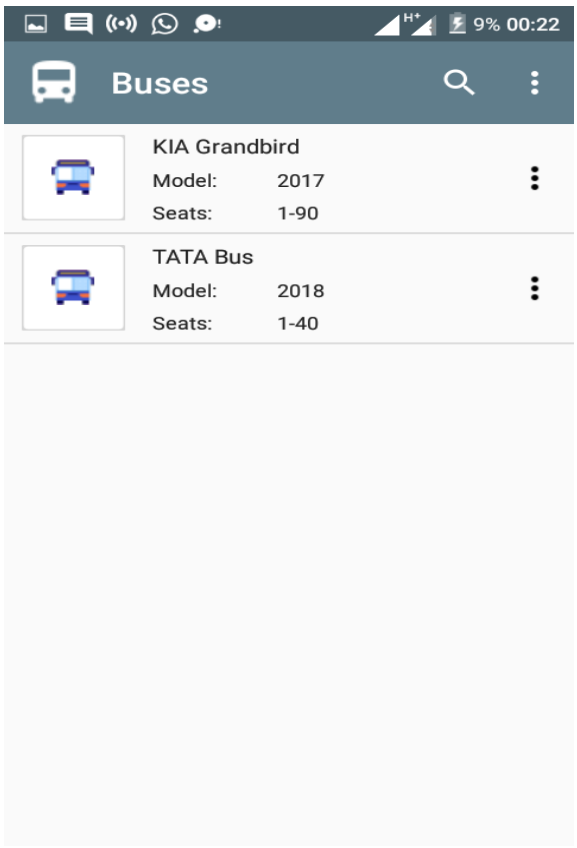


Figure 17 Home screen of buses on the admin page

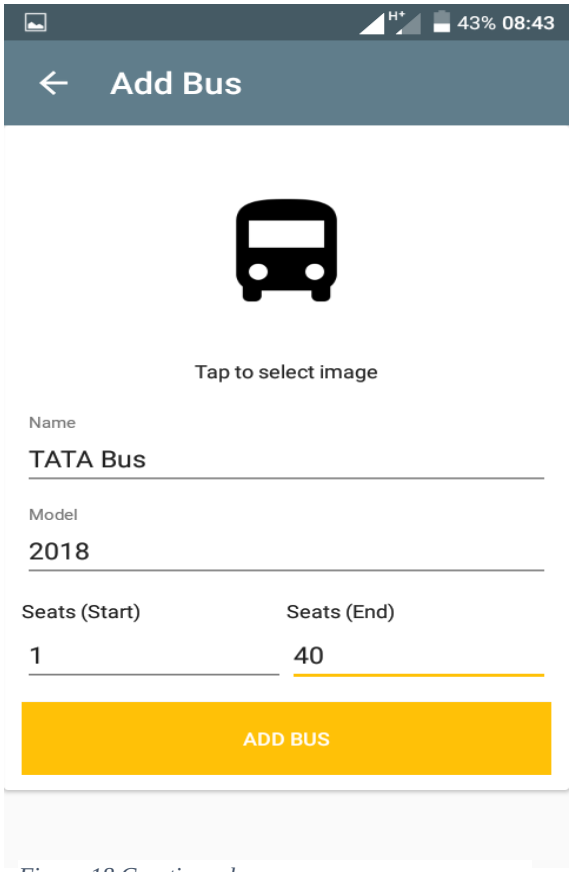


Figure 18 Creating a bus

Reservation

The Administrator oversees all bookings made by users

The application sends an SMS notification to the Administrator whenever a user books a reservation.

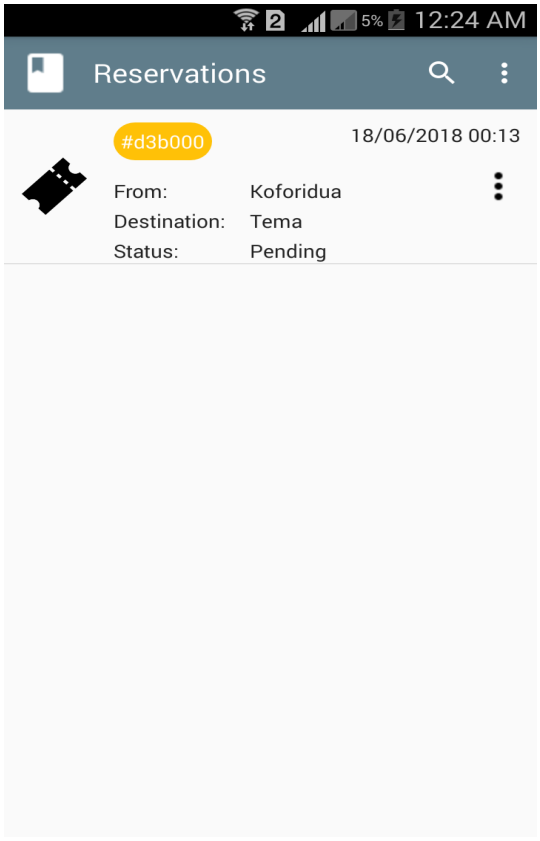


Figure 19 Home screen of reservation on the admin page

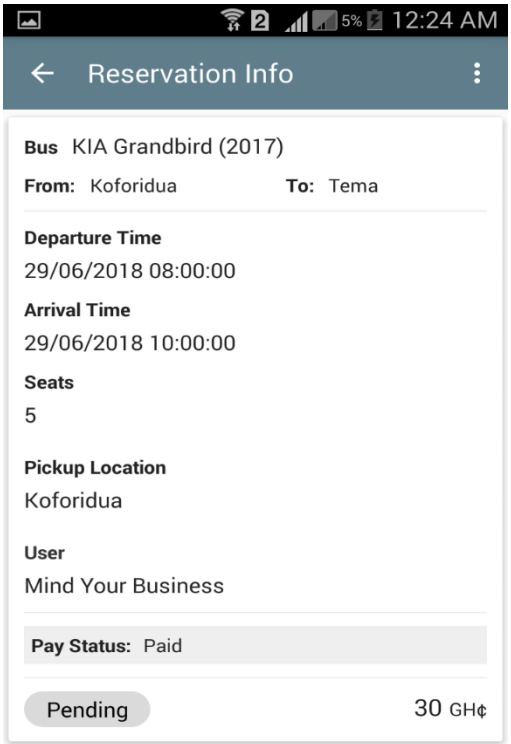


Figure 20 Reservation Information

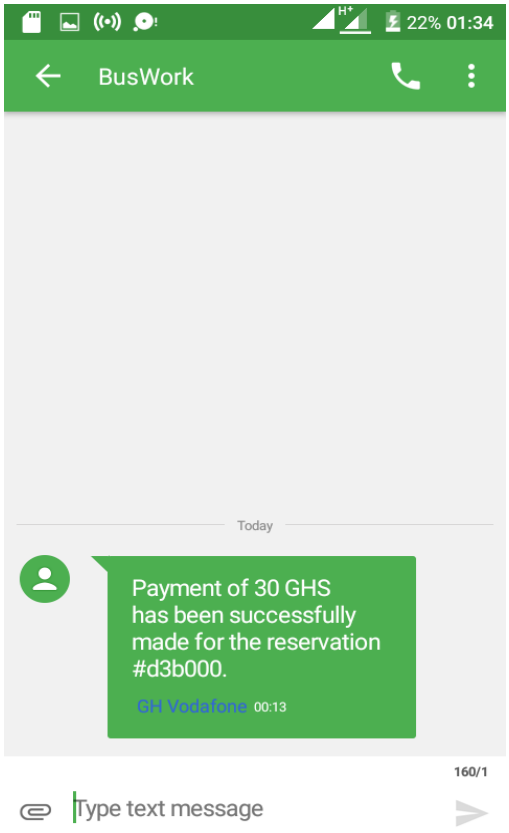


Figure 21 SMS notification received for payment made from user

Feedback

Comments made by Users concerning the application is viewed by the Administrator in this section.

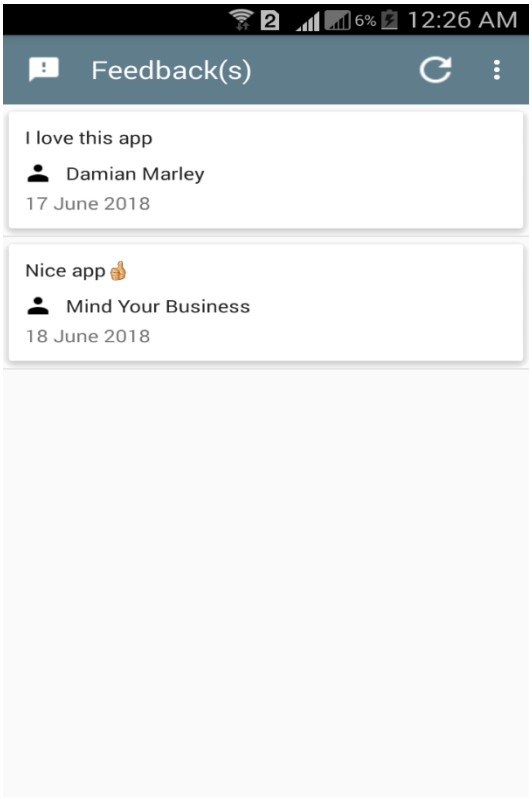


Figure 22 Feedback received from users

Administrator's Profile

This is where the administrator views his or her profile. The fields Username, Full Name, Email and Phone are subject to change.

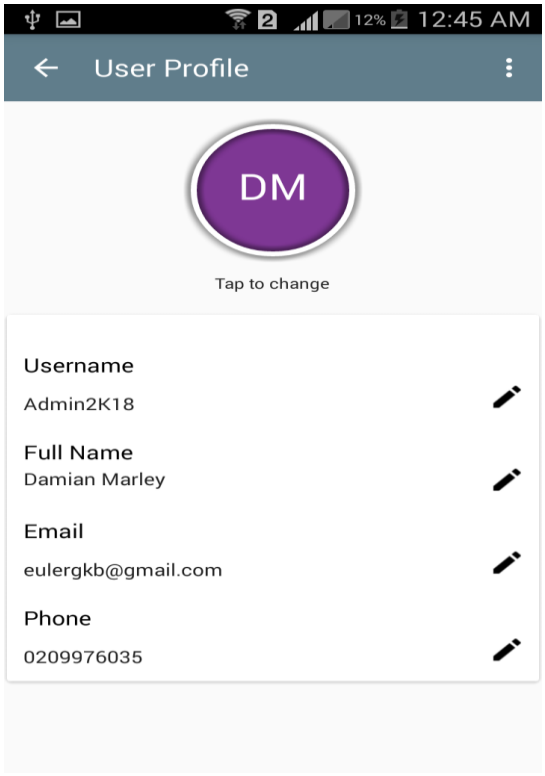


Figure 23 Admin's Profile

User Access

Upon registering successfully as a User, a confirmation code is sent via SMS to the User's phone for it to be used in activating his or account.

When the User login is successful, a page is displayed where there is a menu button that shows three options and they are as follows:

- Home
- Buses
- Reservation/Tickets

There is a **Signout** button which allows the User to logout from the application. The application permits the User to perform changes in profile and also send feedbacks.

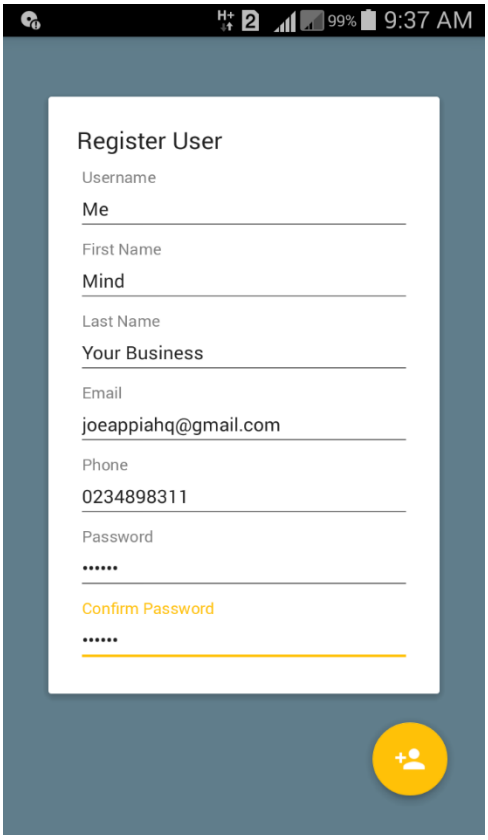


Figure 24 User Registration

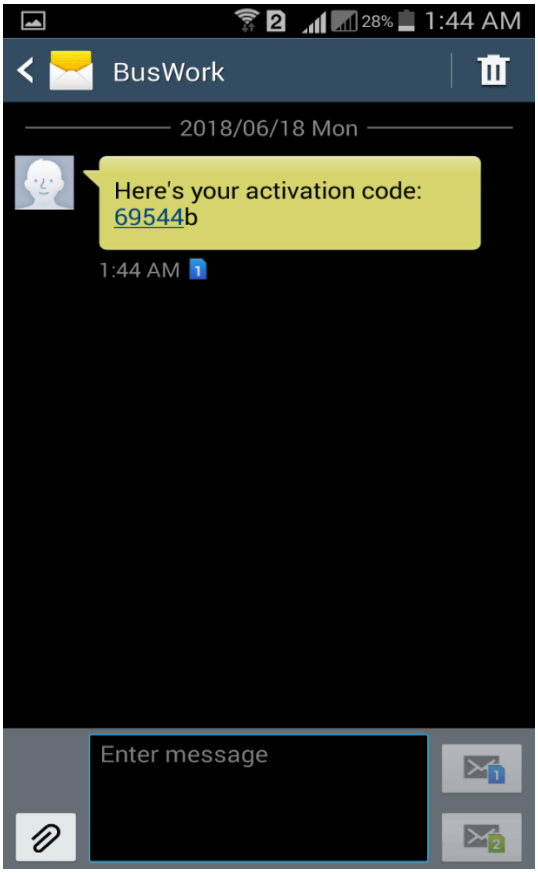


Figure 25 Activation code received by text

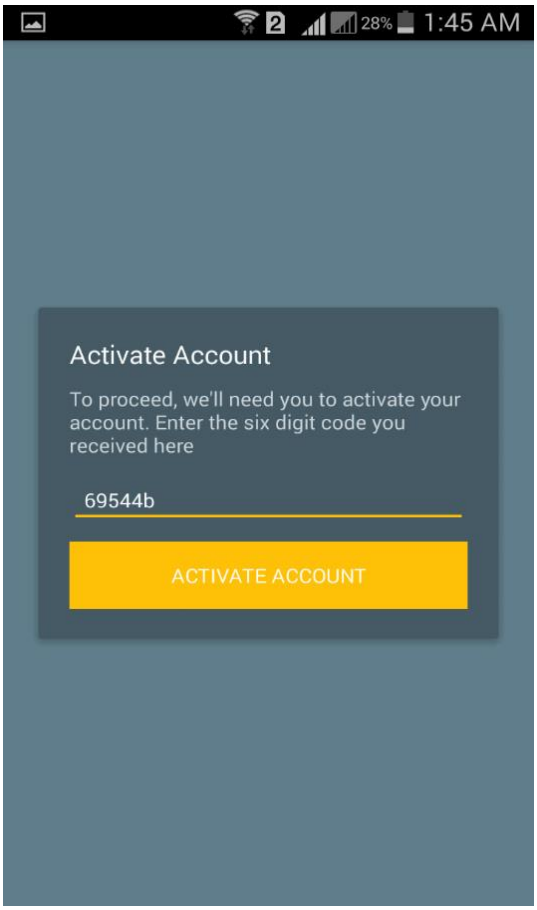


Figure 26 Account Activation page

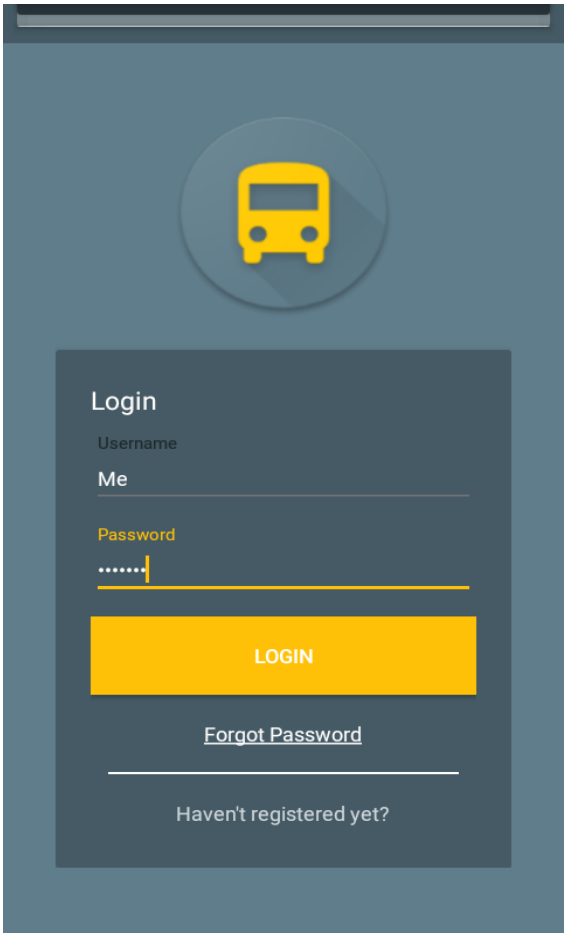


Figure 27 User login page

Home View

The Home View gives a summary of reservation made by the User on daily or weekly basis. There is also button that allows the User to book a reservation.

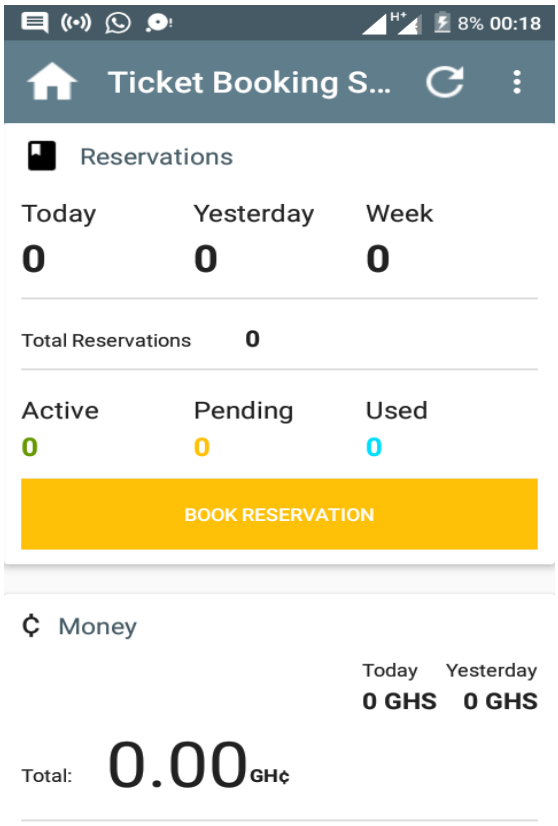


Figure 28 User home page

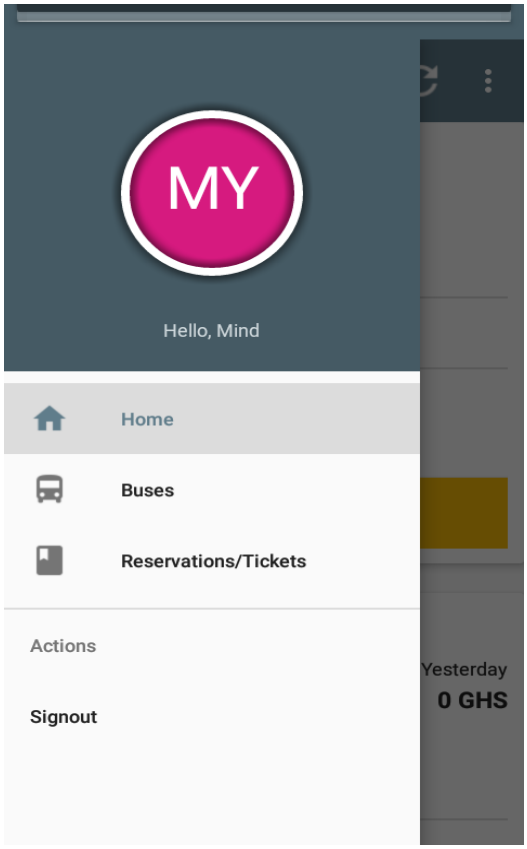


Figure 29 User home page with drop down menu

Buses

This section allows the User to view buses and journey details and also check for the available buses for his or her preferred journey.

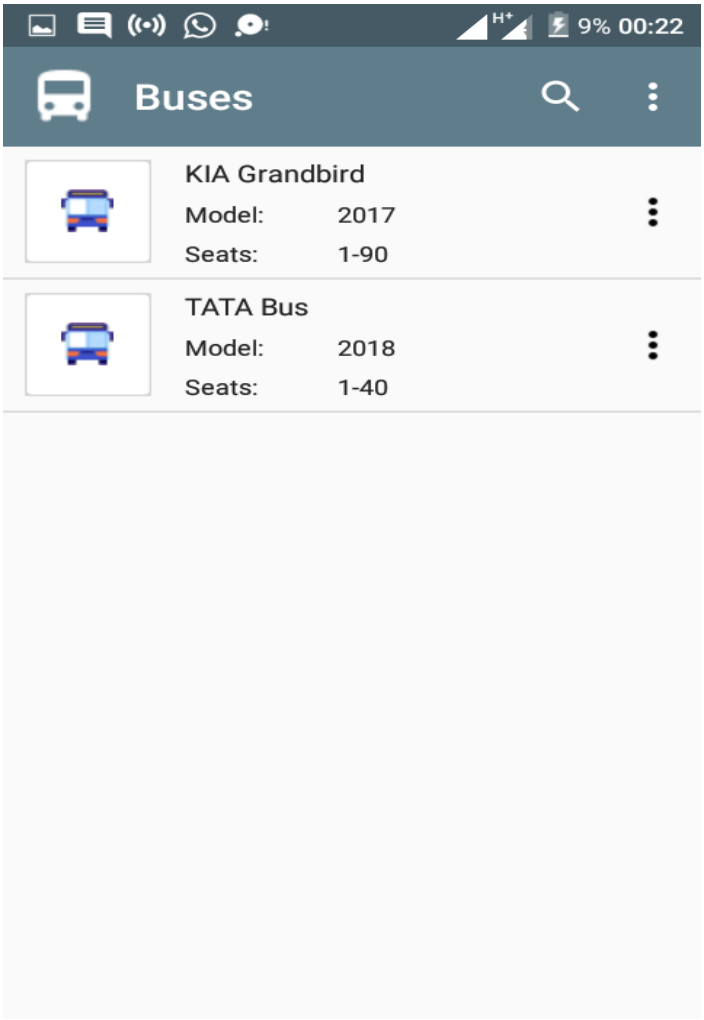


Figure 30 List of buses

Reservation

This is where the user gets access to book reservation for a particular bus. The User is allowed to select a bus which comes along with its routes. Details for the journey which takes into consideration the time of departure and arrival are clearly stated. The application also gives the User the opportunity to select any seat of his or her choice. If a particular seat has already been booked by another User for that same journey, it doesn't show in the list when the new User is selecting a seat. This is because, each and every seat has been numbered.

Once booking is successful, an SMS is sent to the Administrator with the reservation's ticket number. The application then redirects the User to the homepage where he or she can view the ticket number.

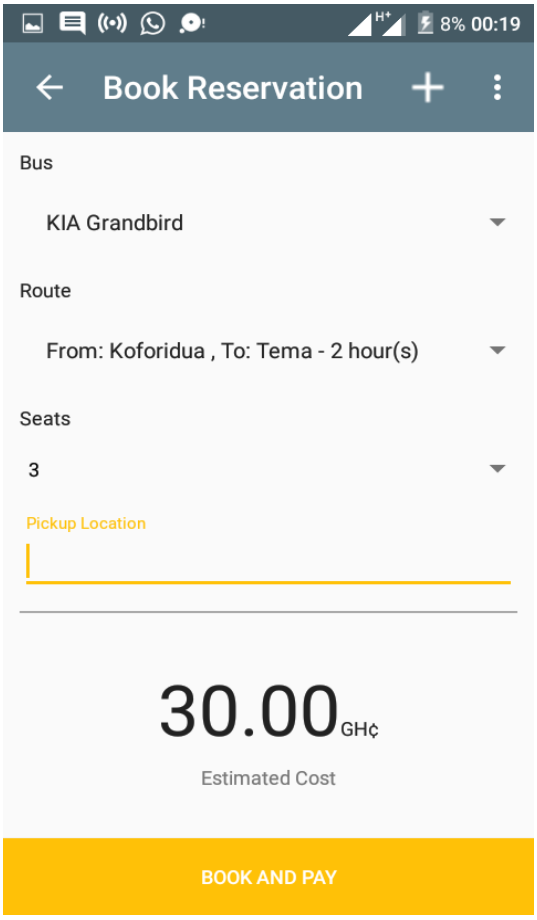


Figure 31 Book reservation

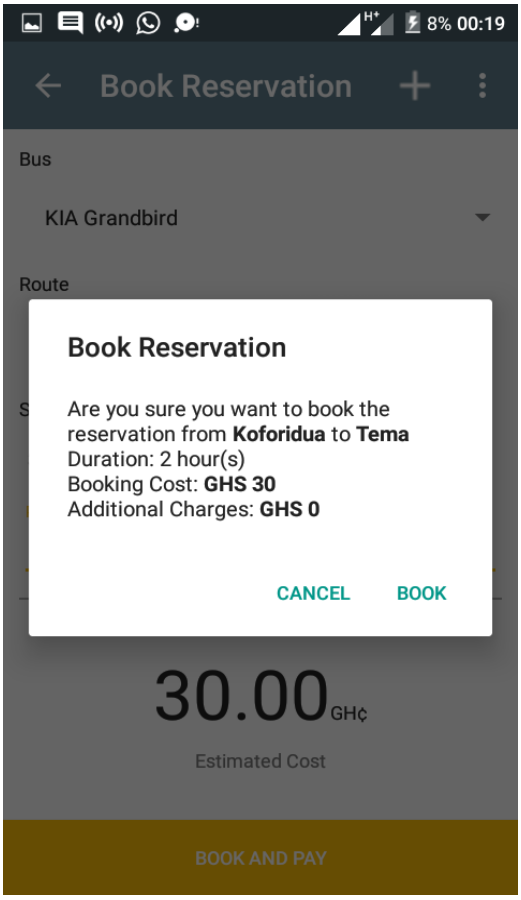


Figure 32 Confirm reservation

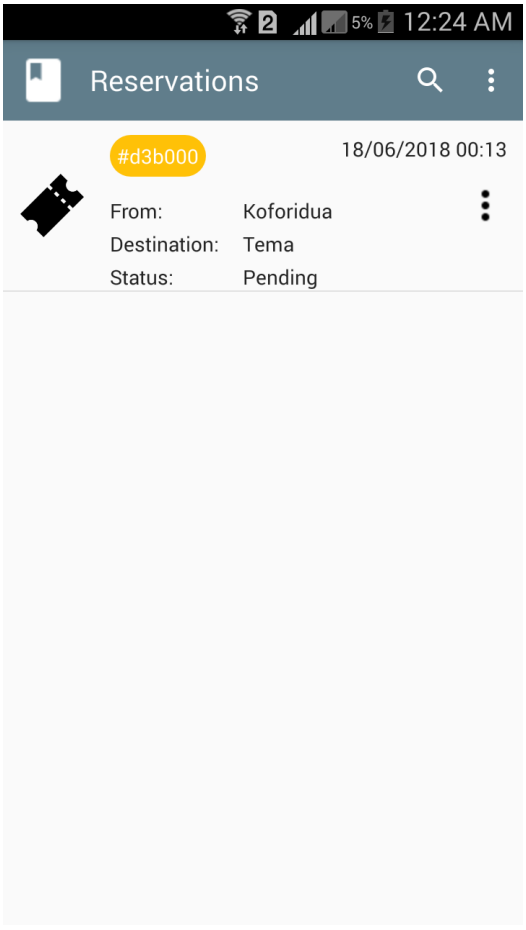


Figure 33 Reservations

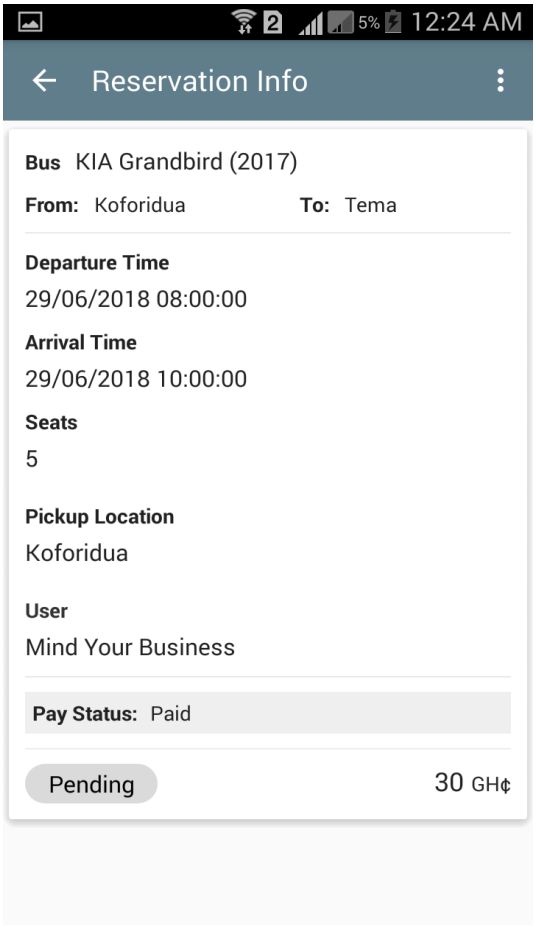


Figure 34 Reservation Information

User's User Profile

This is where the User views his or her profile. The fields Username, Full Name, Email and Phone are subject to change. There is a button that allows the user to manage his or her wallet. That is, the User can add other mobile money wallets.

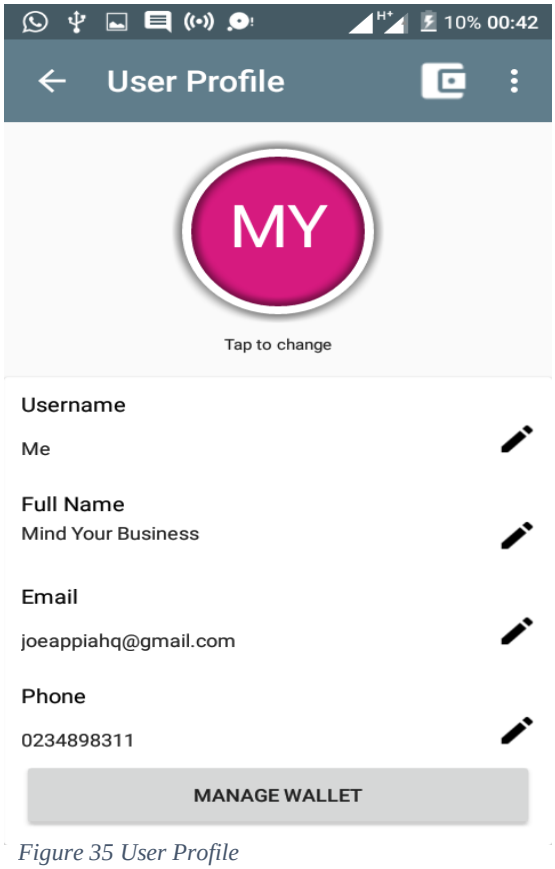


Figure 35 User Profile

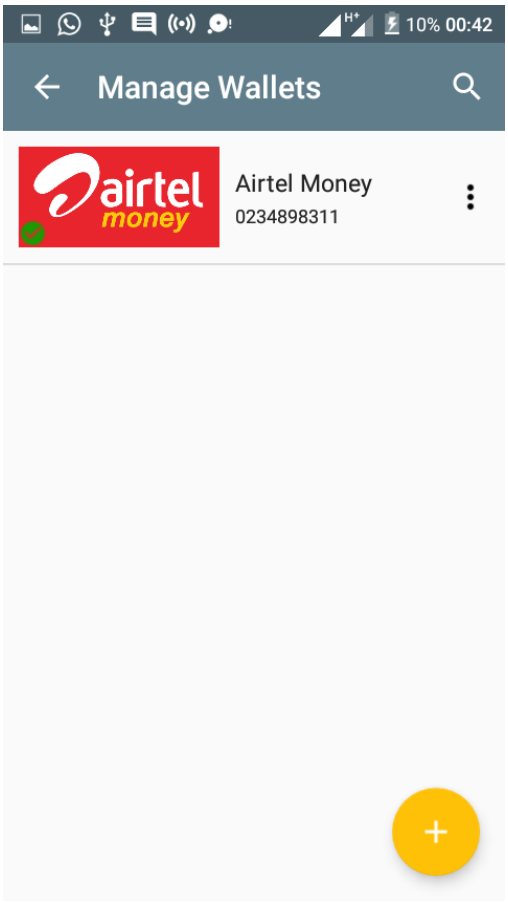


Figure 36 User Wallet

4.4 THE ENDS OF THE APPLICATION

There are two forms of ends in this project. These ends depend on what can be done and by whom. The ends of the application are;

- User End
- Back End

User End

After the development and marketing of the application completely, the individual that uses the product is termed as the end user. This is useful in that it differentiates between the user that needs a bug-free product and those that use the product for development purposes. The term end-user thus distinguishes the user for which the product is designed from other users who are making the product possible for the end user. Often, the term user is appropriate.

This application makes it possible for the user to book and pay for bus reservations. The user pays for the reservation from a mobile money wallet.

The application also allows Users to cancel reservations made.

The Back End

A back-end application or program serves indirectly in support of the front-end services, usually by being closer to the required resource or having the capability to communicate with the required resource. The back-end application may interact directly with the front-end or, perhaps more typically, is a program called from an intermediate program that mediates front-end and back-end activities.

The backend consists of two primary components: application business logic and data processing or management. Separating the backend into these parts significantly helps in reaching the understanding of what backend-as-a-service is.

In this application the Administrator has a page that shows various reservations made by Users, the number of buses available and also feedbacks sent from the Users. Every action performed is stored in a database which is only accessible by the Administrator. This is what we term as the Database Development page.

Max Rows: 1000

Id	Name	Model	Seats	DateCreated	LastUpdated	IsSoftDeleted	ProfileImage_Id
1	Tatra	Default	1-50	2/25/2018 9:12:...	2/25/2018 9:12:...	False	5
2	Ayalolo	Default	1-90	2/25/2018 9:12:...	2/25/2018 9:12:...	False	8
3	Puller	V29	1-90	3/24/2018 3:20:...	3/24/2018 3:24:...	True	NULL
4	KANTANKA AIR	Bee	1-90	3/24/2018 3:55:...	3/24/2018 5:53:...	True	7
5	Papparazi	2K19	1-30	3/24/2018 4:00:...	3/24/2018 5:53:...	True	6
6	Motor Sport	VK12	1-90	3/24/2018 5:45:...	3/24/2018 5:53:...	True	9
7	Pee	Rover123	1-90	3/24/2018 5:55:...	3/24/2018 5:55:...	False	10
8	Mongo	HREF12	1-90	3/24/2018 5:56:...	3/24/2018 5:56:...	False	NULL
9	Tango	Orange Lime - ...	1-90	3/24/2018 5:56:...	3/24/2018 5:56:...	False	11
10	My Bus2010	2011	1-90	3/24/2018 5:57:...	3/24/2018 5:57:...	False	NULL
11	Deever	ASTRA	1-30	4/28/2018 5:29:...	4/28/2018 5:29:...	False	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 37 List of Buses

Max Rows: 1000

Id	RefNo	Cancelled	RouteId	Seats	DateCreated	LastUpdated	User_Id	PickupLocation
1	257000	False	1	14	5/7/2018 3:52:5...	5/7/2018 3:57:5...	3e46a838-ee8a-...	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 38 List of Reservation

Max Rows: 1000

Id	From	DepartureTime	ArrivalTime	FromLat	FromLng	Cost	Comments	Destination	DestinationLat	DestinationLng	DateCreated	LastUpdated	IsSoftDeleted	Bus_Id
1	Accra	4/4/2018 12:00:...	4/5/2018 12:00:...	NULL	NULL	20		Kumasi	NULL	NULL	2/25/2018 9:12:...	2/25/2018 9:12:...	False	1
2	Kaneshie	2/16/2018 12:00...	3/20/2018 12:00...	NULL	NULL	32		Nsawam	NULL	NULL	2/25/2018 9:12:...	2/25/2018 9:12:...	False	1
3	Tamale	4/4/2018 12:00:...	5/5/2018 12:00:...	NULL	NULL	16		East Tafo	NULL	NULL	2/25/2018 9:12:...	2/25/2018 9:12:...	False	1
4	Taifa	2/16/2018 12:00...	7/20/2018 12:00...	NULL	NULL	12		Dome	NULL	NULL	2/25/2018 9:12:...	2/25/2018 9:12:...	False	2
5	Brong Ahafo	4/4/2018 12:00:...	6/5/2018 12:00:...	NULL	NULL	20		Ho	NULL	NULL	2/25/2018 9:12:...	2/25/2018 9:12:...	False	2
6	Bee	4/20/2018 12:00...	5/22/2018 12:00...	NULL	NULL	10		Nsawam	NULL	NULL	2/25/2018 9:12:...	2/25/2018 9:12:...	False	2
7	Harna	3/27/2018 3:00:...	3/26/2018 12:00...	NULL	NULL	20		Depue	NULL	NULL	3/24/2018 1:20:...	3/24/2018 1:20:...	False	1
8	Kaneshie	3/1/2022 12:00:...	3/30/2022 12:00...	NULL	NULL	23		Teshie	NULL	NULL	3/24/2018 1:47:...	3/24/2018 1:47:...	False	2
9	Kokofu	3/3/2019 6:30:0...	3/30/2019 2:28:...	NULL	NULL	32		Mangani	NULL	NULL	3/24/2018 2:29:...	3/24/2018 2:29:...	False	2
10	Jake	3/24/2020 6:29:...	3/30/2022 2:29:...	NULL	NULL	90		Mountain Everest	NULL	NULL	3/24/2018 2:30:...	3/24/2018 2:30:...	False	2
11	Kakora	3/26/2018 4:05:...	3/27/2018 4:05:...	NULL	NULL	10		Dumsi	NULL	NULL	3/24/2018 4:05:...	3/24/2018 4:05:...	False	5
12	Tamale	3/24/2018 8:58:...	3/28/2018 5:58:...	NULL	NULL	10		US Oregon	NULL	NULL	3/24/2018 5:58:...	3/24/2018 5:58:...	False	1
13	Mongo	3/27/2018 6:00:...	3/30/2018 6:00:...	NULL	NULL	12		Orgen US	NULL	NULL	3/24/2018 6:00:...	3/24/2018 6:00:...	False	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 39 List of Routes

Max Rows: 1000

Id	PasswordHash	PasswordSalt	Username	Email	Phone	FullName	LockedOut	DateCreated	LastUpdated	EmailConfirmed	IsSoftDeleted	ProfileImage_Id
79d-cc00a368c3f8	z/M7jQYteohc6...	HQgrVxMVfLRE...	Emma	emma@trotro....	0501583979	Emmanuel Ans...	False	3/6/2018 7:48:5...	3/6/2018 7:49:4...	True	False	NULL
11504136-eceb-...	pJYSJEZlB+VcY...	zuGswJccaR9H...	Monkey	monkey@powe...	0233412234	Monkey Powers	False	3/4/2018 11:28:...	3/4/2018 11:28:...	True	False	NULL
129392ad-6fff-4...	ogqjjogl27hmD...	Fu/6xlK8ivwd25...	Pack	mongo@db.com	0243341232	Part Mongoo	False	3/4/2018 11:44:...	3/4/2018 11:44:...	True	False	NULL
13b99221-cc13-...	UIC6ffRP+D6Hx...	+LKR+aCSxv6Lt...	Hitler	ekb2@live.com	0501583923	Kofi Hitler	False	3/4/2018 8:58:5...	3/4/2018 8:58:5...	False	False	NULL
3e46a838-ee8a-...	Rda2SvBwjtnlKl...	loDeKHR9p0dj...	kamal	kamal@tro.com	0501583978	Kamal Yakub	False	3/6/2018 7:53:5...	3/6/2018 7:54:1...	True	False	NULL
460d1662-ad21...	GljPbHesP7l7M...	wv7XW0CzkW...	iwan	iwan@live.com	0201231239	Malam Iwan	False	3/24/2018 6:08:...	3/24/2018 6:09:...	True	False	NULL
585ce4bd-5ae6...	JgXXjHUKGaTI+...	wYbr4I9Ht34Sz...	Shatta	boahene@hide...	0201231231	Kofi Waler	False	3/4/2018 11:13:...	3/4/2018 11:15:...	True	False	NULL
64304908-64b8-...	5MSo+1suZ9T...	h6WNrLd6146w...	Admin2K18	eulergkb@gma...	0209153453	Damian Marley	False	2/25/2018 9:12:...	2/25/2018 9:12:...	True	False	4
6e808f5b-b757-...	lLgcwPr4tLCh+...	vQZxwW9TOiD...	Styles	mile@gmail.com	0201654321	Mile Styles	False	3/5/2018 12:45:...	3/5/2018 12:45:...	True	False	NULL
7e9fe1cc-9fbb-...	r5QCpqMdEU...	q8v96yMh9AMl...	Dude	henry@dude.io	0555456342	Dude Henry	False	3/6/2018 7:46:2...	3/6/2018 7:46:5...	True	False	NULL
7f0afcc9-afe1-4...	Fo973U9eqbU2...	2Hj5lqNn+P7X...	KofiTROTRO	kofi@trotrotrac...	0201583982	Kofi Boahene	False	3/22/2018 10:18...	3/24/2018 10:25...	True	False	3
7f50c8f1-d322-...	NyK3DYityMdy...	19V3wC7z3132...	bazaarContext	euler-kb@live.c...	0501583982	Kwame Kofi Ny...	False	2/25/2018 9:12:...	4/28/2018 5:50:...	True	False	2
8ba25c3c-4ebf-...	YQd2bFRMPM...	Slh8PF5F86umv...	Popoo	shoo@gmail.co...	0201231237	Poo Shoo	False	3/4/2018 11:25:...	3/6/2018 7:38:3...	True	False	NULL
8cf6d58d-bea0-...	ejXbDBiV5yv6E...	t39HS+ASEyDn...	Demo	demo@live.com	0203243211	Demo Barker	False	3/6/2018 7:41:0...	3/6/2018 7:41:4...	True	False	NULL
b8161d06-b785...	Nq/EmQGNTV...	Dr3KVd1h6leqh...	TNana	nana@tro.com	0501583981	Nana Darko	False	3/6/2018 7:51:3...	3/6/2018 7:51:5...	True	False	NULL
d3856e1a-9470-...	nHfT2Tpnem/h...	sbyMbMKv7wy...	TTKofi	kofi@trotrotrac...	0501583980	Kofi Boahene	False	3/4/2018 10:36:...	3/6/2018 7:32:0...	True	True	NULL

Figure 40 List of users

CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

Having gone through all the necessary processes, we were able to design and develop an application that is able to help users make bus reservation with ease and most importantly pay with their mobile money account and also cancel the reservation if not relevant anymore.

The application is very important to anyone who finds him or herself in a position where going to the bus station will be a problem and needs to make a trip in a short possible time. This application helps users know the available buses, it's time of departure and arrival and even gives you an option for you to be picked at a destination of your choices along it's (the routes of the bus) way.

With the help of our Supervisor, we were able to design and develop an Android application that can help users book bus reservations.

5.1.1 LIMITATIONS

Although the application works successfully, it has some limitations. The application was successful in the testing and analysis stage, we had some limitations which we think if worked on in the subsequent years will generally improve the application. Some of these limitations are;

- The application works only on Android Operating System. Operating systems besides Android will not support the application.
- Besides it being an android application it doesn't work well on all android operation system because of the various upgrades in the operating system.
- The application needs constant connection to the internet to function and so when the user runs out of bundle the application is rendered useless.

This means the application will not run on any operating system except Android operating system. Other operating systems such as iOS (Apple devices operate on this operating system), Symbian (Some Nokia devices operate on this operating system), Windows (Microsoft devices and some smartphones operate on this operating system) and other operating systems will not support this application.

5.1.2 CHALLENGES

The development of this project was not an easy thing to come by. We faced a lot of difficulties in developing this project. Some of the challenges we faced were:

- We not being able to get the mobile money API to integrate it with our application because the companies to provide us with the service were adamant on we getting a physical company which has been properly registered and besides they cannot give out their API for school projects. In their defense, it is to prevent fraudsters from gaining access to their system.

5.1.3 ACHIEVEMENTS

- We began the project with the aim to develop an application that will allow users make bus reservations and pay using mobile money. We were not able to achieve our aim.
- This whole project has given us in depth knowledge about Android application Development. As a result, we would be able to create and manage Android applications in future.

5.1.4 LESSONS LEARNT

We discovered that, in the development of applications that make use of mobile money for financial transactions there is the need for mobile money API and SMS API integration.

5.2 RECOMMENDATIONS

Day in day out new technologies appear in the technological sphere, and upon completion of this project, it was found out that new technologies have sprung up.

In this section, we recommend that further research be performed in developing the application for other operating systems besides Android, that is, iOS, Symbian, Windows and others. This project has laid foundations for further project to be developed and eventually port the system to more mobile systems including iOS and Windows Phone OS.

Additionally, since this application cannot be used by non-smartphone users we recommend a similar of this nature is made for users of such phones. With the help of USSD which provides location-based content services for low end phones this could be achieved.

Finally, we recommend that APIs for projects be made available to students by the School.

REFERENCES

1. J. Martin. [Online]. Available: http://blog.martin.edu.au/index.php/informtion-technology/technology-advanced-time/ [Accessed 25th December 2017].
2. "Smartphone users worldwide: Statista," [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worlwide/[Accessed 25th December 2017]
3. "mobile-money-in-ghana-mobile-money-transactions-see-20-growth-in-mid-2016: Pulse Ghana," [Online]. Available: https://www.pulse.com.gh/telecom/mobile-money-in-ghana-mobile-money-transactions-see-20-growth-in-mid-2016-id5366002.html[Accessed 25th December 2017].
4. Asad, A.A., Ayad, M.J. and Hayder, N.K. (2012). Design and Developing Online Iraqi Bus Reservation System Using Unified Modeling Language. International Journal of Scientific knowledge Available at: http://www.ijsk.org/uploads/3/1/1/7/3117743/v3i103_information_technology.pdf [Accessed 26th December 2017]
5. Maike, J.P. (2014). Train, bus and museum - Interrelations of diverse actors within integrated E - ticketing schemes. Available at: http://www.mobil-tum.vt.bgu.tum.de/fileadmin/w00bqi/www/Session_Poster/Puhe.pdf [Accessed 26th December,2017]
6. "incremental model: Technologyuk," [Online]. Available: https://www.technologyuk.net/computing/sad/images/incremental_model.gif [Accessed 26th December,2017]
7. “Brief History of the Internet internetsociety” [Online]. Available: https://www.internetsociety.org/internet/history-internet/brief-history-internet/ [Accessed 28th December, 2017]

8. What Is The Internet (And What Makes It Work) https://www.cnri.reston.va.us/what_is_internet.html [Online]. Available: [Accessed 28th December, 2017]

9. "Internet computerhope" [Online]. Available: https://www.computerhope.com/jargon/i/internet.htm [Accessed 28th December 2017]

10. Smartphone https://www.computerhope.com/jargon/s/smartphone.htm [Accessed 6th January, 2018]

11. Smartphone https://www.techopedia.com/definition/2977/smartphone [Accessed 6th January, 2018]

12. computer https://www.webopedia.com/TERM/C/computer.html [Accessed 6th January, 2018]

13. What is a computer? https://www.gcflearnfree.org/computerbasics/what-is-a-computer/1/ [Accessed 6th January, 2018]

14. "mobile-money-in-ghana-mobile-money-transactions-see-20-growth-in-mid-2016: Pulse Ghana," [Online]. Available: https://www.pulse.com.gh/telecom/mobile-money-in-ghana-mobile-money-transactions-see-20-growth-in-mid-2016-id5366002.html [Accessed 25th December 2017].

15. "Mobile-money-holds-huge-potential-for-Ghana-391851: Ghanaweb," [Online]. Available: http://www.ghanaweb.com/GhanaHomePage/NewsArchive/Mobile-money-holds-huge-potential-for-Ghana-391851 [Accessed 25th December 2017].

16. “6th-accra-tech-salon-to-discuss-the-impact-of-mobile-money-and-technology-in-ghana Penplusbytes” [Online]. Available: penplusbytes.org/site/index.php/6th-accra-tech-salon-to-discuss-the-impact-of-mobile-money-and-technology-in-ghana/ [Accessed 10th January, 2018]
17. http://www.iiste.org/journals/ Computer Engineering and Intelligent Systems Vol.5, No.12, 2014 [Accessed 10th January, 2018]
18. http://www.liacs.nl/assets/Bachelorscripties/2006-08JanneLouw.pdf[Accessed 10th January, 2018]
19. https://www.coursehero.com/file/21742621/ONLINE-BUS-BOOKING-SYSTEM/[Accessed 10th January, 2018]
20. Shuchi, G. (2008). Explain the Phased Waterfall Model. Available at: http://www.careerride.com/testingphased-waterfall-model.aspx [Accessed 25th May, 2018]
21. "incremental model: Technologyuk," [Online]. Available: https://www.technologyuk.net/computing/sad/images/incremental_model.gif [Accessed 26th December,2017]
22. S. Ian, "Functional and Non-Functional requirements," in Software Engineering, Pearson
23. O. Cinar, "Application Components," in *Android Quick APIs Reference*, Apress, 2015, p. 31.
24. "incremental model: Technologyuk," [Online]. Available: https://www.technologyuk.net/computing/sad/images/incremental_model.gif [Accessed 26th December 2017].

APPENDIX

APPENDIX I: FULL CODE OF MAIN ACTIVITY

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
```

```
using Android.Widget;
using BookingSystem.Android.Factory;
using BookingSystem.API.Models;
using System.Threading.Tasks;
using BookingSystem.Android.Helpers;

namespace BookingSystem.Android
{
    [Activity(MainLauncher = true, Theme = "@android:style/Theme.NoDisplay")]
    public class MainActivity : Activity
    {
        protected async void InitializeStateAsync()
        {
            try
            {
                //  Determine user state here
             var proxy = ProxyFactory.GetProxyInstace();

                if (AuthenticationManager.HasSession)
                {
                    var authInfo = AuthenticationManager.CurrentSession;

                    if (await proxy.RestoreAsync(authInfo, true))
                    {
                        NavigationHelper.NavigateUserHome(this);
                        Finish();
                        return;
                    }

                }
            }
            catch
            {
                global::Android.Util.Log.Debug("com.booking.system.Startup", "Failed
restoring previous session");
            }

            //
            var intent = new Intent(this, typeof(LoginActivity));

            //  Navigate to other activity
            intent.SetFlags(ActivityFlags.ClearTask | ActivityFlags.NewTask |
ActivityFlags.SingleTop);
            StartActivity(intent);

            //
            Finish();
        }

        protected override void OnCreate(Bundle savedInstanceState)
        {
            base.OnCreate(savedInstanceState);

            //  Initialize state
            InitializeStateAsync();

        }
    }
```

APPENDIX II: FULL CODE OF ADMIN ACTIVITY

```
using Android.App;
using Android.Widget;
using Android.OS;
using Android.Views;
using System;

namespace BookingSystem.Android
{
    [Activity(Label = "@string/AppName", Icon = "@drawable/icon")]
    public class AdminMainActivity : BaseDrawerActivity
    {
        static MenuLayoutBinding[] Bindings = new MenuLayoutBinding[]
        {
           new MenuLayoutBinding()
            {
                Title =
Application.Context.Resources.GetString(Resource.String.AppName),
                Layout = Resource.Layout.admin_home_layout,
                MenuItemId = Resource.Id.nav_action_home,
                PageType = typeof(Pages.AdminHomePage),
                Icon = Resource.Drawable.ic_home_black_18dp,
                ToolbarIcon = Resource.Drawable.ic_home_white_18dp,
                Flags = MenuFlag.DefaultPage | MenuFlag.Refresh,
            },

            new MenuLayoutBinding()
            {
                Title = "Buses",
                Layout = Resource.Layout.buses_layout,
                MenuItemId = Resource.Id.nav_action_buses,
                PageType = typeof(Pages.BusesPage),
                Icon = Resource.Drawable.ic_bus_black_18dp,
                ToolbarIcon = Resource.Drawable.ic_bus_white_18dp,
                Flags = MenuFlag.Search | MenuFlag.AddButton
            },

            new MenuLayoutBinding()
            {
                Title = "Reservations",
                Layout = Resource.Layout.reservations_layout,
                PageType = typeof(Pages.ReservationsPage),
                MenuItemId = Resource.Id.nav_action_reservations,
                Icon = Resource.Drawable.ic_book_black_18dp,
                ToolbarIcon = Resource.Drawable.ic_book_white_18dp,
                Flags = MenuFlag.Search
            },

            new MenuLayoutBinding()
            {
                Title = "Feedback(s)",
                Layout = Resource.Layout.feedbacks_layout,
                MenuItemId = Resource.Id.nav_action_feedback,
```

```
                PageType = typeof(Pages.FeedbackPage),
                Icon = Resource.Drawable.ic_feedback_black_24dp,
                ToolbarIcon = Resource.Drawable.ic_feedback_white_24dp,
                Flags = MenuFlag.Refresh
            }
        };

        public AdminMainActivity()
        {

        }


     protected override int GetHeaderLayout() => Resource.Layout.nav_user_info;

        protected override int GetNavigationMenu() =>
Resource.Menu.navigation_menu_admin;

        protected override int? GetMenuResource() => Resource.Menu.actions_main;

        /// <summary>
        /// Gets binding for menu items
        /// </summary>
        /// <returns></returns>
        protected override MenuLayoutBinding[] GetMenuLayoutBindings() => Bindings;

        protected override bool OnNavigationItemClicked(IMenuItem menu)
        {
            if (menu.ItemId == Resource.Id.nav_signout)
            {
                ConfirmSignout();
                return true;
            }

            //  Wasn't handled
            return false;
        }

        public override bool OnOptionsItemSelected(IMenuItem item)
        {
            switch (item.ItemId)
            {
                case Resource.Id.action_user_profile:
                    StartActivity(typeof(UserProfileActivity));
                    break;
                case Resource.Id.action_about:
                    StartActivity(typeof(AboutActivity));
                    break;
                case Resource.Id.action_signout:
                    ConfirmSignout();
                    break;
            }

            return base.OnOptionsItemSelected(item);
        }

    }
}
```

APPENDIX III: FULL CODE OF USER ACTIVITY

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;

namespace BookingSystem.Android
{
    [Activity(Label = "@string/AppName")]
    public class UserMainActivity : BaseDrawerActivity
    {
        private readonly MenuLayoutBinding[] Bindings = new MenuLayoutBinding[]
        {
            new MenuLayoutBinding()
            {
                Title =
Application.Context.Resources.GetString(Resource.String.AppName),
                Layout = Resource.Layout.user_home_layout,
                MenuItemId = Resource.Id.nav_action_home,
                PageType = typeof(Pages.UserHomePage),
                Icon = Resource.Drawable.ic_home_black_18dp,
                ToolbarIcon = Resource.Drawable.ic_home_white_18dp,
                Flags = MenuFlag.DefaultPage | MenuFlag.Refresh,
            },

            new MenuLayoutBinding()
            {
                Title = "Buses",
                Layout = Resource.Layout.buses_layout,
                MenuItemId = Resource.Id.nav_action_buses,
                PageType = typeof(Pages.BusesPage),
                Icon = Resource.Drawable.ic_bus_black_18dp,
                ToolbarIcon = Resource.Drawable.ic_bus_white_18dp,
                Flags = MenuFlag.Search
            },

            new MenuLayoutBinding()
            {
                Title = "Reservations",
                Layout = Resource.Layout.reservations_layout,
                PageType = typeof(Pages.ReservationsPage),
                MenuItemId = Resource.Id.nav_action_reservations,
                Icon = Resource.Drawable.ic_book_black_18dp,
                ToolbarIcon = Resource.Drawable.ic_book_white_18dp,
                Flags = MenuFlag.AddButton | MenuFlag.Search
            }
```

```
        };

        public UserMainActivity()
        {

        }

        protected override int GetHeaderLayout() => Resource.Layout.nav_user_info;

        protected override int GetNavigationMenu() =>
Resource.Menu.navigation_menu_user;

        protected override int? GetMenuResource() => Resource.Menu.actions_main;

        protected override MenuLayoutBinding[] GetMenuLayoutBindings() => Bindings;

        protected override bool OnNavigationItemClicked(IMenuItem menu)
        {
            if (menu.ItemId == Resource.Id.nav_signout)
            {
                ConfirmSignout();
                return true;
            }

            //  Wasn't handled
            return false;
        }

        public override bool OnOptionsItemSelected(IMenuItem item)
        {
            switch (item.ItemId)
            {
                case Resource.Id.action_user_profile:
                    StartActivity(new Intent(this, typeof(UserProfileActivity)));
                    return true;
                case Resource.Id.action_about:
                    StartActivity(new Intent(this, typeof(AboutActivity)));
                    return true;
                case Resource.Id.action_signout:
                    ConfirmSignout();
                    return true;
            }

            return base.OnOptionsItemSelected(item);
        }
    }
}
```