



ASHESI UNIVERSITY COLLEGE

**A REAL-TIME PASSENGER INFORMATION SYSTEM FOR
PUBLIC TRANSIT SYSTEMS IN ACCRA**

APPLIED PROJECT

B.Sc. Computer Science

Alex Kwesi Nkonim Adu

2018

ASHESI UNIVERSITY COLLEGE

**A Real-Time Passenger Information System for Public Transit Systems
in Accra**

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi
University College in partial fulfilment of the requirements for the award of
Bachelor of Science degree in Computer Science

Alex Kwesi Nkonim Adu

April 2018

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

.....

Candidate's Name:

.....

Date:

.....

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of Applied Project laid down by Ashesi University College.

Supervisor's Signature:

.....

Supervisor's Name:

.....

Date:

.....

Acknowledgements

I wish to express my sincere gratitude to my supervisor, Dr Nathan Amanquah, for all the guidance and encouragement he provided as I undertook this project. I would also like to thank all family members and friends who supported me during this project.

Abstract

Transportation has played a vital role in the development of societies globally. The quality of services provided by a country's transportation sector directly influences the productivity of a country's citizens. In Ghana, the transportation services offered by public transport agencies are inefficient and unreliable. The lack of transit information about these services often cause a great level of inconvenience before, during and after an intra-city commute. With the development of mobile technology and global positioning systems, technology can immensely improve the services provided by the public transportation sector in Ghana. A mobile application which functions as a passenger information system is proposed in this project to provide relevant and real time information to commuters in Accra. The proposed system allows transport managers and bus conductors to update information that can be viewed by commuters. It also allows commuters to view the estimated time of arrival of a bus operating along a route in real-time.

Table of Contents

Chapter 1: Introduction	1
1.1 Introduction.....	1
1.2 Background	2
1.3 Motivation.....	2
1.4 Problem Statement.....	3
1.5 Objectives of the Project	3
1.6 Related Work	4
1.7 Overview of Chapters.....	5
Chapter 2: Software Requirement Analysis.....	6
2.1 User Requirements.....	6
2.1.1 Users	6
2.1.2 Use Cases	6
2.1.3 Use Case Diagram.....	8
2.2 Functional Requirements	8
2.2.1 User Requirements.....	9
2.2.2 System Requirements.....	10
2.3 Non-Functional Requirements	11
Chapter 3: Architecture and Design	12
3.1 Multi-Tier Client-Server Architecture	12
3.1.1 Client Server Model.....	12
3.1.2 Presentation Layer	13
3.1.3 Data Management and Application Processing Layer.....	13
3.2 Multi-Tier Architecture.....	15

Chapter 4: Implementation.....	16
4.1 Data	16
4.2 Data Preparation.....	17
4.3 Tools and Technologies	18
4.3.1 Firebase	18
4.3.2 Ionic	18
4.3.3 Google APIs.....	19
4.3.4 Geolocation	20
4.4 Implementation	20
4.4.1 User Authentication	20
4.4.2 Functionalities.....	21
Chapter 5: User Testing	28
5.1 Unit Testing	28
5.2 Component Testing.....	29
5.3 System Testing.....	29
Chapter 6: Conclusion.....	30
6.1 Future Works	30
References.....	31

Chapter 1: Introduction

1.1 Introduction

Transportation has played an integral role in the development of societies globally. History has shown that innovation in the transportation sector serves as a catalyst for industrialization, globalization and economic prosperity. Thus, the quality of service delivered by the transportation sector of a country must be optimized as it directly influences the productivity of citizens and ultimately improves the standard of living in the country.

The public transportation sector of a country provides commuters with reliable mobility services such as mass transit systems. In some developing countries, specifically Ghana, these transit systems are inefficient and unreliable. Ghana's public transport sector consists of mini bus taxis named TroTros, commuter busses, taxis and other transit busses. The lack of planning, and coordination of these services negatively impacts the daily commutes of Ghanaians. Though constant improvements are ongoing to enhance Ghanaian transit systems, the problems that persist are pervasive. Studies show that Ghanaians are appalled by the poor service delivered by the public transport sector. A survey conducted by Birago et al reveals that Metro Mass transit commuters in Ghana often complain of the system's frequent delayed arrival times and unavailability of buses at embarking points (Birago, Mensah, & Sharma, 2017).

With the development of mobile technology and global positioning systems, technology can immensely improve the services provided by the public transportation sector in Ghana. The availability of real time passenger information systems, allows commuters to access real time information which conveys a sense of reliability to aid with planning before and during a journey. This will significantly reduce the feeling of anxiety caused by the

uncertainty of a journey. Transit agencies can also use such technologies and the analytical data it provides, to improve operations and guide decision making in transit management.

1.2 Background

Passenger information systems (PIS) provide users with information about transport services. It is used in many countries to enhance passenger experience and improve the usability of transit systems. When PIS provide real time information, they utilize automatic vehicle location technology to track the location of a specific means of transport in real time. This live data about the location of a means of transport can be used as input to forecast the arrival time of a vehicle at a given location. The algorithm used to estimate the arrival time of a vehicle determines the precision of the system.

1.3 Motivation

Vehicular traffic is a persistent problem that affects the reliability of transportation services in Accra. The problem often stems from roads being congested with too many vehicles at a point in time. If public transportation agencies in Accra improve their services to offer more satisfactory and reliable services, commuters would be incentivised to use public transportation as a means of transport rather than privately owned vehicles. This would consequently reduce the number of vehicle on the road and thus reduce traffic congestion. This project seeks to improve the services delivered by the public transportation sector of the Accra. It seeks to provide a service that encourages commuters to use public transportation and thus reduce the occurrence of vehicular traffic on the roads of Accra.

For a daily commuter in Accra, the lack of information about transit services often causes a great level of inconvenience before, during and after an intra-city commute. Unfavourable weather conditions, poorly built road infrastructure and vehicular traffic are

factors that immensely contribute to the late arrival of transit busses. The unreliable nature of transit services in the country causes commuters to bare a higher cost for an using alternate means of transport which is more reliable. The implementation of passenger information systems in public transport services will allow commuters to plan trips, save time and patronize more reliable services which assure them of a degree of certainty for travel time.

1.4 Problem Statement

Delays, which are of regular occurrence, cause a high level of uncertainty with regards to waiting times for and arrival times of public transport. This problem of uncertainty evokes a feeling of anxiety and incurs an opportunity cost to the passenger as the time spent in waiting for a bus could have been used in a more productive way.

1.5 Objectives of the Project

Recognizing the pervasive nature of travel time uncertainty in Ghana, this project proposes a solution which utilizes automatic vehicle location systems and arrival time prediction algorithms to build a passenger information system that provides travel time information. As a measure of assessment, the following objectives are essential to the successful completion of the project:

- Designing a mobile application (passenger information system) for commuters, bus conductors and transport managers.
- Developing and implementing a system which calculates and displays the estimated time of arrival for a specific transit bus from a bus stop
- Developing an application which provides a user with real time transit information

1.6 Related Work

In many developed countries, intelligent transportation systems have been implemented to enhance transportation services. Many of such systems include an automated real time bus prediction system. These existing predicting systems have various implementation methods.

Google Maps, developed by Google, is a web-based mapping service that provides users with navigational directions and location based information. It offers a service that allows users to search for and view transit information such as schedules, estimated time of arrivals, and all bus routes in a city ("Google Maps Help", 2018). The Transport for London(TFL) journey planner also provides similar services regarding real-time transit information ("Plan a journey", 2018). Unfortunately, these service are not available for Ghanaian transit services because they are a location based services available for specific countries.

Researchers of the computer department in MAEER'S MIT Academy of Engineering in India, propose a framework for the implementation of a Real-Time Passenger Information System. The implementation utilizes a Global Positioning System (GPS) receiver which periodically obtains GPS coordinates of a bus and transfers the data to a centralised server. Using a route creation algorithm and a computation for the time of arrival at a geographic location, the estimated time of arrival for each bus stop is derived and transmitted through GPRS to the centralized server. The output information can be accessible through a web based portal, an RTPIS rolling display at bus stops or a mobile device application. This implementation is seamless and adaptable for any environment. Though this framework is well designed, its implementation will require the installation of GPS receivers on all transit busses in Accra. This will be costly and would require a long period of time before the system would be utilized.

1.7 Overview of Chapters

This paper is segmented into seven chapters. Chapter 1 provides the reader with background information concerning the field of study and reveals the projects motivation, significance, related works as well as objectives. The requirements of the system, along with a high-level description of the system are presented in Chapter 2. Chapter 3 describes the application's design and architecture specifications. Based on these specifications, Chapter 4 provides a detailed explanation of the tools, techniques and procedures used to implement the system. The 5th Chapter assesses the performance of the implemented passenger information system application. While the last chapter discusses the shortcomings of this project and recommends future works that could address the limitations of this project.

Chapter 2: Software Requirement Analysis

This chapter provides a detailed description of the proposed system's user, functional and non-functional requirements.

2.1 User Requirements

2.1.1 Users

The stakeholders in this proposed system are commuters, transport managers and bus conductors. A commuter can utilize the implemented system while waiting at a transit stop, sitting in a bus, or for planning purposes. They would use the application to view real time information about operating buses and other transit information. A transport manager is responsible for the execution and coordination of all transportation activities within an agency. Thus, the transport manager would utilize the implemented application to add, view and update information regarding the agency's transport services. While the bus is in operation, the bus conductor, who is present on the bus is responsible for updating real time bus information.

2.1.2 Use Cases

As implied, the proposed system seeks to address varying needs of users who utilize or work in the transportation sector of the city of Accra. To ensure that the implemented system caters for such use cases, this project formulated scenarios as a means of reference to envisage the context of the systems usage. The scenarios below aid in the identification of use cases to outline the requirements for each user.

Scenario 1

Kofi Ansah is an entrepreneur who has a business meeting which starts at 2:00pm and ends at 3:30pm. He also has a doctor's appointment at 4:00pm. Kofi wishes to use the Metro Mass Transit system as a means of transportation to arrive at both destinations on time. At 12:30 pm, Kofi opens the proposed application to plan his journey. He checks the estimated time of arrival of buses to be informed of the best time to leave the house for the business meeting as well as the best time to leave the meeting for hospital appointment. The application also suggests the busses that will allow Kofi to arrive in at his destination in the shortest amount of time with the least waiting time.

Scenario 2

It is 4:30pm and raining heavily at the Spanner bus stop. Joan is on her way home, waiting at the bus stop for a bus which usually arrives at 4:35. After waiting for 10 minutes, she realizes the bus is delaying. She is uncertain of the extent of the delay. A taxi driver approaches her and offers to drive her to her destination at three times the cost of the bus fare. Joan uses the mobile application to check the estimated time of arrival the bus. This will guide her decision as to whether to accept the taxi drivers offer or wait for the bus which will cost significantly less.

Scenario 3

Noah, the new Accra manager of the Metro Mass Transit system wishes to improve the services offered to the public. He especially seeks to identify the busses which often delay in their arrivals at bus stops. He utilizes the analytical data which the application provides to view the historical arrival times of busses. With this information, Noah can determine which bus services require changes to prevent frequent delays.

2.1.3 Use Case Diagram

The scenarios above illustrate possible contexts of the application's usage. Through this illustration, it is possible to cater for each user's possible use of the proposed system. The use case diagram in Figure 2.1 serves as a graphical representation of a user's interaction with the system.

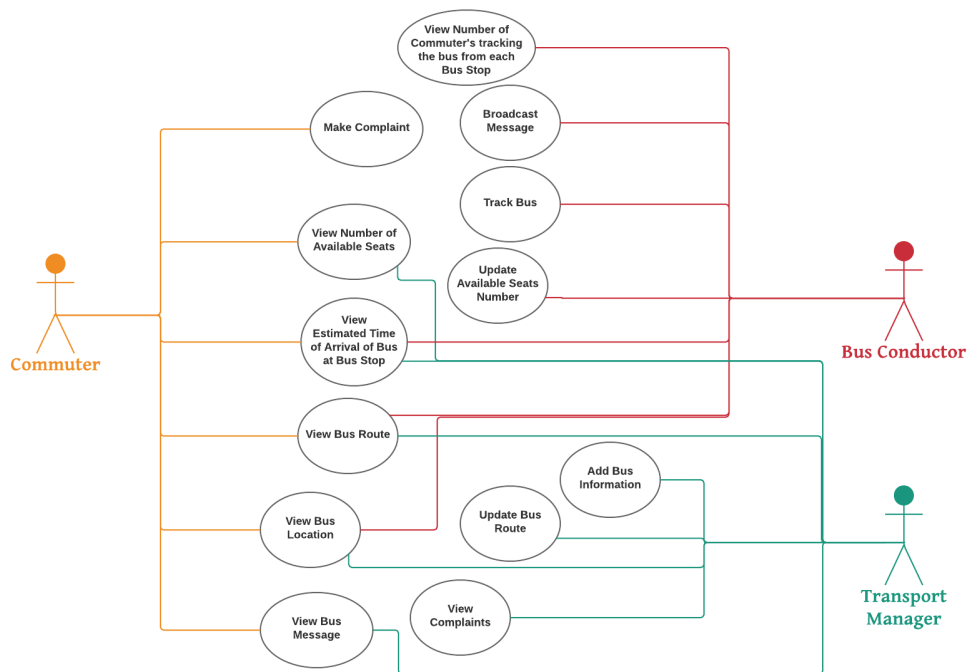


Figure 2.1 Use Case Diagram

2.2 Functional Requirements

The functional requirements of the system defines the services provided by the system and the behaviour of the system. This project categorises the functional requirement specifications into user requirements and system requirements.

2.2.1 User Requirements

1. A commuter using the system should be able to select a bus route and view all bus stops along that route.
2. A commuter should be able to view all the buses that can transport them to their destination.
3. A commuter should be able to view in real time, the estimated times of arrival of any bus at a bus stop.
4. A commuter should be able to view in real time, the number of seats available on a bus.
5. A commuter should be able to view in real time, the location of a bus.
6. A commuter should be able to view messages concerning specific buses
7. A commuter should be able to make a complaint about the quality of a bus's service.
8. A commuter should be able to view all buses and bus stops within walking distance of their location.
9. A transport manager should be able to add new bus information.
10. A transport manager should be able to view in real time, the location of an agency owned bus.
11. A transport manager should be able to view messages concerning specific agency owned buses
12. A transport manager should be able to number of available seats on an agency owned bus
13. A transport manager should be able to create a new bus route.
14. A transport manager should be able to verify the identity of a bus conductor.
15. A transport manager should be able to view complaints made by commuters about agency owned buses.

16. A transport manager should be able to view past data about bus arrival times.
17. A bus conductor should be able to record in real time, the location of a bus in operation.
18. A bus conductor should be able to update the number of seats on a bus in operation.
19. A bus conductor should be able to view in real time, the location of the bus and its proximity to bus stops.
20. A bus conductor should be able to view in real time, the number of commuters tracking the bus.
21. A bus conductor should be able to broadcast messages concerning a bus in operation.
22. A bus conductor should be able to view in real time, the estimated time of arrival of a bus in operation at a bus stop.
23. A bus conductor should be able to view the stops on each bus route.

2.2.2 System Requirements

1. The system must provide information about all bus stops and routes in Accra.
2. The system must provide frequently updated transit information about the location of busses operating in Accra.
3. The system must compute and provide users with the estimated time of arrival of a bus at any bus stop along its operating route.
4. The system must provide information about the number of seats available on a bus.
5. The system must provide information about the number of computes tracking a specific bus from a particular bus stop.

2.3 Non-Functional Requirements

1. The system must be reliable.
2. The system must be available at all times.
3. The system must provide accurate information.
4. The system must provide up-to-date information.
5. The system must be secure and require user authentication to control user permissions.
6. The system must be accessible on Android, iOS and Windows mobile devices.

Chapter 3: Architecture and Design

This chapter describes the system's architecture and design.

A system's architecture is the conceptual representation of the overall organisational structure of the system, the principal components and their relationships (Sommerville, 2016). In designing a distributed system's architecture, it is important to take into consideration manageability, dependability, performance and the system's non-functional requirements (Sommerville, 2016). Considering these factors, the implemented PIS is built on an architectural style known as a multi-tier client-server architecture.

3.1 Multi-Tier Client-Server Architecture

3.1.1 Client Server Model

The implemented PIS is a client-server system; a system where users interact with a device and access services provided by a remote computer. The client in this PIS is the mobile device running the application on either an Android, iOS, or a Windows platform. The client requests for data computed and stored on remote servers.

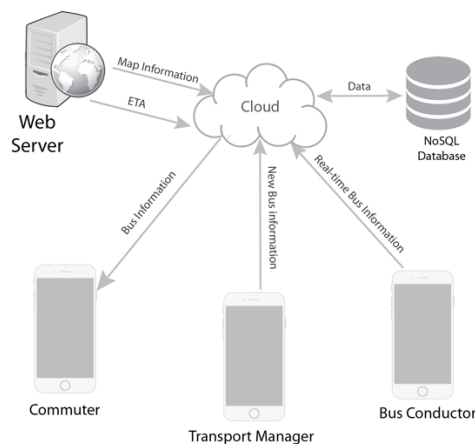


Figure 3.1- Context-Level Data-Flow Diagram Overview

This system's client-server architecture separates the presentation, computation and processing of information into four logical layers. These layers are labelled as the presentation layer, data management layer, application processing layer and database layer.

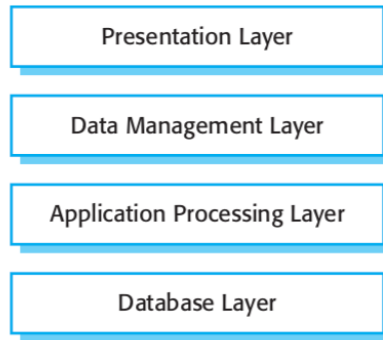


Figure 3.2 Layered Architecture Diagram
(Sommerville, 2016)

3.1.2 Presentation Layer

The presentation layer of a client-server system is the aspect of the application concerned with managing user interaction. It essentially defines the system's user interfaces and user experience. The implemented application's interface is designed in an intuitive manner that enhances usability, reduces cognitive load, ensures consistency in design and follows platform conventions. The typography for the application enhances user experience by communicating textual information in a manner that establishes information hierarchy. The application also presents location based information on a map to allow users to easily perceive such information.

3.1.3 Data Management and Application Processing Layer

The application layer of the system implements the logic of the application by providing functionality to meet end user's needs (Sommerville, 2016). This layer in the

system, accepts inputs from the user and executes functions such as computing the estimated time of arrival of a bus. A data-intensive application such as the implemented system requires that application processing is handled on servers which are non-volatile, thus cloud-based servers are used for processing. The data transferred between the server and the client is managed by the data management layer (Sommerville, 2016).

3.1.4 Database Layer

The database layer stores the data required for the presentation layer and application processing. Based on the non-functional requirements, this system uses a highly secure cloud-based NoSQL database which is accessible from any location, has elastic scaling capabilities, and real time delivery of data. The database stores all real-time information about busses, routes, stops and users.

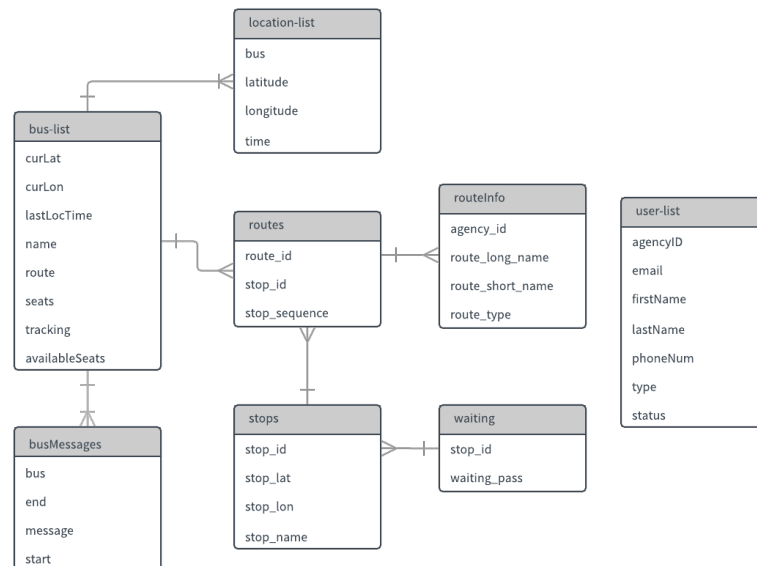


Figure 3.3 Entity Relation Diagram

Figure 3.3 shows an entity relation diagram that represents all the tables in the database and their relations. The bus-list table contains of all information about a bus. This

includes the name of the bus, the current location, the route of the bus, its seating capacity, the tracking status and the last time the busses location was recorded. The location-list table permanently stores all location information about a bus. The busMessages table stores all messages about a bus. The table stores the message as well as the time the message is to be broadcasted and the time of expiry for the message. The routeInfo table an agency's route information, this includes the route's names and route type. The user-list table stores all user information. The waiting table stores the number of commuters tracking the bus from a particular bus stop.

3.2 Multi-Tier Architecture

The multi-tier client-server architecture style is used to build systems, such as this PIS, which integrates data from multiple sources (Sommerville, 2016). In this architecture, the presentation of data, data management, application processing, and database functions are executed on different processors (Sommerville, 2016). This means that the implemented application, which is built on this architecture, does not require much processing power from the client to process data as most of the data processing is be handled by servers. This is beneficial because the application can run on devices that have slow processor speeds.

Chapter 4: Implementation

The mobile application was built using various tools and technologies in order to create a complete PIS. For it to function as a PIS for Accra, it stores and uses transit data about bus stops and routes in Accra in a database. This chapter describes in detail, the techniques, procedures and tools used to implement the proposed system.

4.1 Data

In building a passenger information system, it is imperative that transportation information is acquired and stored in a database accessible through the system. In May 2016, the Meltwater Entrepreneurial School of Technology in partnership with the Accra Metropolitan Assembly (AMA) and the French Agency for Development (AFD) made public collated data on over 300 TroTro routes and about 2500 bus stops in Accra in 2015 ("The TroTro Apps Challenge", 2018). The data retrieved from the online source is categorized into two folders, "GTFS" and "Trips Recorded". The "Trips Recorded" folder contains comma separated files while the "GTFS" folder contains subfolders which contain shapefiles and text files. Each folder includes a text file that describes the data enclosed in the folder.

Table 4.1 Description of files in GTFS Folder ("The TroTro Apps Challenge", 2018)

File Name	Description	GTFS Specifications	Accra Data
agency.txt	Information on different transit operators	Required	Available
calendar.txt	List of days included in assignment period	Required	Available
routes.txt	List of bus lines	Required	Available

stop_times.txt	Arrival times at bus stops during data collection period	Required	Available
stops.txt	List of bus stops	Required	Available
trips.txt	List of trips	Required	Available
shapes.txt	Rules for drawing the linear shape of a bus line itinerary on a map	Optional	Available
fare_attributes.txt	Fare information.	Optional	Available
fare_rules.txt	Rules for applying fare information to trotro routes.	Optional	Available
File Name	Description		
stops.csv	List of bus stops.		
routes.csv	List of sequence of stops along each route.		

4.2 Data Preparation

The files used in the implemented system were stops.txt, routes.txt, and routes.csv. These files contained the list of bus stops, bus routes and sequence of stops along each bus route respectively. These files were used because the transit information they contain are necessary to build a system that meets the functional requirements of this project. All files used were converted into comma separated values file formats. Irrelevant and outdated data such as the timestamps, speed of travelling TroTros, fare and number of passengers boarding at each stop were removed from the original routes.csv file. All converted and original CSV files were converted into keyed JSON files using an online tool available on convertcsv.com

("CSV To JSON Converter", 2018). These JSON files were then imported into a NoSQL database.

4.3 Tools and Technologies

4.3.1 Firebase

The implemented system uses Firebase Realtime Database because it satisfies the need for a real-time database as stated in the architecture and design specifications. Firebase Realtime Database is a low latency cloud-based NoSQL database service which allows clients to store, retrieve and query data in real time (Firebase, 2018). It provides developers with multi-platform support as well as an API to store data on the Firebase cloud and synchronize this data across clients in real-time. The implemented system accesses Firebase Realtime Database through AngularFire2, an AngularJS library for Firebase. This library also provides access to Firebase Authentication; a group of user authentication services comprising of email, phone number and social media authentication. The database stores all real-time information, about the bus, routes, stops, users and messages, in a JSON format.

4.3.2 Ionic

As per the requirements, the implemented system must run on Android, iOS and Windows devices. Considering these requirements and the constraints on development time, Ionic was selected as the best suited development kit to develop the application.

Ionic is an open source mobile application development kit which uses web technologies such as HTML, Sassy CSS and Typescript to develop interactive native and progressive web applications ("Ionic Native Geolocation", 2018).

HTML is the markup language used to define the structure of pages in the application developed with Ionic. Sass CSS is the style sheet scripting language compiled into Cascading Style Sheets used to design the presentation of HTML elements on a page. Typescript is a superset of JavaScript, used to dynamically control content on pages and interact with servers. The Ionic front-end framework allows developers to create hybrid applications with native mobile components, tools and functionalities. Its applications use Angular JS to support core functionality and Cordova as a native wrapper, to run the web application as a native application ("Ionic Native Geolocation", 2018).

4.3.3 Google APIs

Google Maps JavaScript API is an open source application programming interface that allows a developer to display and customize maps in an application. The map information is based on accurate and real-time data stored on Google Maps servers. Through an API call, a developer can embed an interactive and customizable map into an application, display location-based data such as traffic, draw a route on the map, and anchor icons at coordinates on the map ("Google Maps Android API", 2018).

Google Directions Service API is also an open source API provided by Google. It allows developers to make requests then receive real-time and relevant information about directions between locations. The API accepts as input, mode of transport, starting location, destination, stopovers and other directional information. It then calculates the most efficient route based on travelling time, distance, number of turns and other traveling information then returns direction results. The direction results returned includes the distance, duration in traffic, step by step directions as well as the estimated time of arrival based on historical and current traffic conditions. The API also outputs routing information that can be rendered

on the map to visually display directions ("Google Maps Directions API", 2018).

4.3.4 Geolocation

The implemented system requires a user's device location information to execute certain functions. In order to acquire the GPS coordinates of the device running the application, the application utilizes the Ionic Native Geolocation plugin. The plugin which is based on W3C Geolocation API Specification, returns the latitude and longitude of the device obtained from the device's GPS and network signals such as IP address, RFID, WiFi and Bluetooth MAC addresses, and GSM/CDMA cell IDs_ ("Ionic Native Geolocation", 2018).

4.4 Implementation

4.4.1 User Authentication

The implemented system was built for three main users; a commuter, a bus conductor, and a transport manager. Hence, the application provides a login portal for each user to sign in. It also allows new commuters to register for an account. Each users account is authenticated with Firebase Authentication which verifies the identity of users, enforces user permissions and ensures account validation with security constraints. Transport managers are not permitted to create accounts for themselves because they are required to verify their licencing by a superior authority. Thus, the super administrator creates an account for transport managers after ensuring they have an authentic licence to manage a public transport agency. Bus conductors can register for an account on the implemented system but can only access the application after their account has been authenticated for use by their agency's transport manager.

4.4.2 Functionalities

4.4.2.1 Adding a New Route

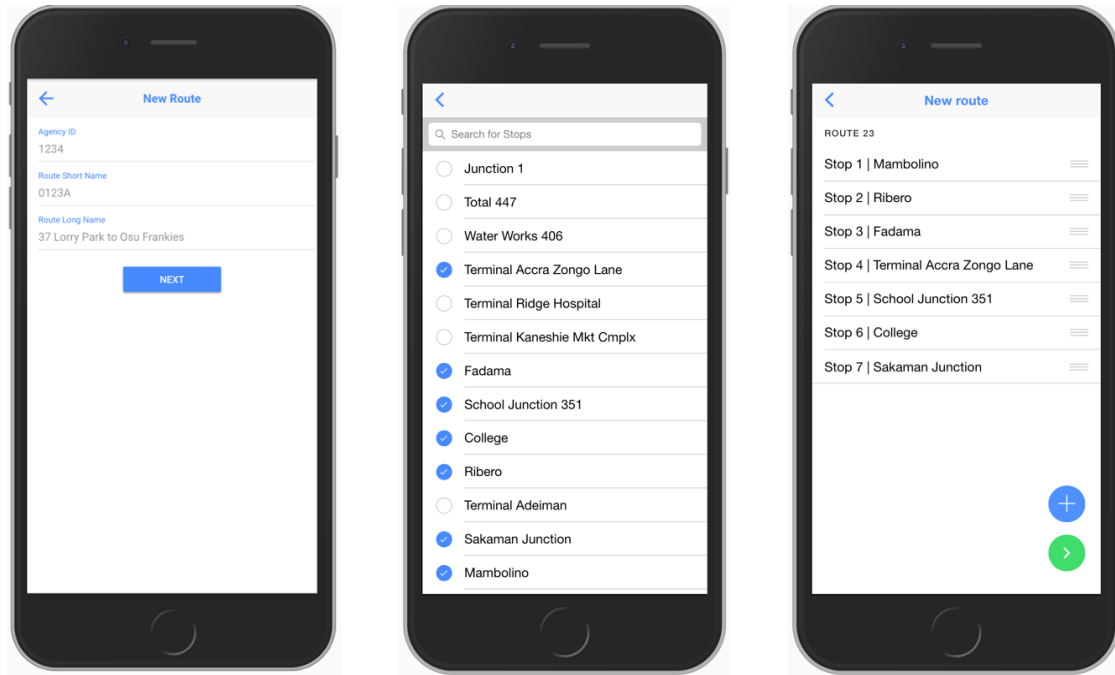


Figure 4.1 Add New Route Pages

The pages, shown above in Figure 4.1, allow a transport manager to create a new route which consists of all the stops that a bus will visit during its operation. The manager begins the process by inputting an agency id, the route id, and the name of a route then proceeds to the next page. The application then retrieves all bus stops using the **getStops()** function then displays all the stops on a page. When the manager selects a bus stop on the list displayed **selectStop()** function adds the selected stop to the list of stops on the new route. The application allows the manager to easily reorder the sequence of stops on the new route. This is done by selecting the area where the stop name is displayed and dragging the stop on the list from one position to another. When the manager reorders the stops on the list, the stop number is automatically updated. After adding all the stops on the route, the manager can then add the route to the database. The application uses the **addRoute()**

function to store all new route information in the route table and routeInfo table stored in the database.

4.4.2.2 Adding New Bus

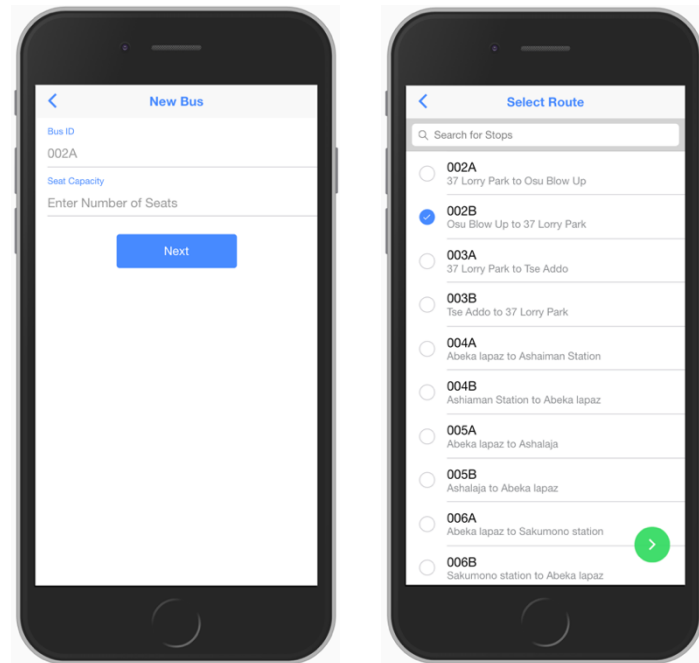


Figure 4.2 Add a New Bus

The pages in Figure 4.2 allow the transport manager to add new bus information to the database. The transport manager inputs the bus name and the number of seats on the bus then selects the route for the new bus. When the transport manager completes the steps to add a new bus, the data is added to the bus table in the Firebase Real-time database. The **addBus()** function stores the new bus information into the bus-list table and seats table in the database.

4.4.2.3 Verify Bus Conductor

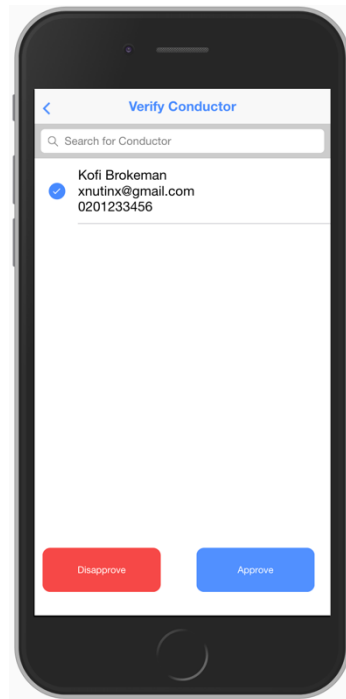


Figure 4.3 Verify Bus Conductor

The page in Figure 4.3 allows a manager to verify the identity of a bus conductor. The manager can view all unverified bus conductors accounts then select an action for one or more accounts. When the manager verifies the identity of a bus conductor, the database updates the user table in the database to reflect the changes in the conductor's information. The **approve()** and **disapprove()** functions update the status of a bus conductor in the user-list table in the database. After being verified, the conductor is granted permission to login, track and update bus information.

4.4.2.4 Track Bus



Figure 4.4 Track Bus Page

When the bus begins operation, the bus conductor uses the application to track the bus and this allows commuter's to view the bus location. The conductor selects a bus and has the option to start tracking. When the conductor begins tracking, their mobile device acts as a tracker for the bus. Every 60 seconds application calls the **updateBusLoc()** function. This function records and updates the current location of the bus in the bus-list table in the database. Every 5 minutes, the application calls the **addBusLocation()** function which records the current location of the bus and permanently stores it in the location-list table.

The page in Figure 4.4 displays the location of the bus on a bus route and allows the conductor to update the number of available seats. It increments and decrements the number of seats available on the bus when the conductor presses the respective buttons. The **addSeat()** and **reduceSeat()** functions update the number of seats available on the bus

stored in the seats table in the database.

The styling of the map, displaying bus information, is customized based on the time of day to improve user experience. From 5AM to 7PM, the map is styled with colours that emit high levels of blue wavelengths. According to a Harvard Medical School Publication, these blue wavelengths enhance user productivity because they boost attention, reaction times and mood ("Blue light has a dark side", 2012). At night, blue wavelengths negatively affect levels of melatonin, a sleep inducing hormone in the body ("Blue light has a dark side", 2012). Based on this information, the application activates the night mode of the map between 7PM and 5AM. The night mode of the map also improves readability and reduces the eye strain of the user at night. All maps in the application display live traffic information obtained by google.

4.4.2.5 View Closest Bus and Stops

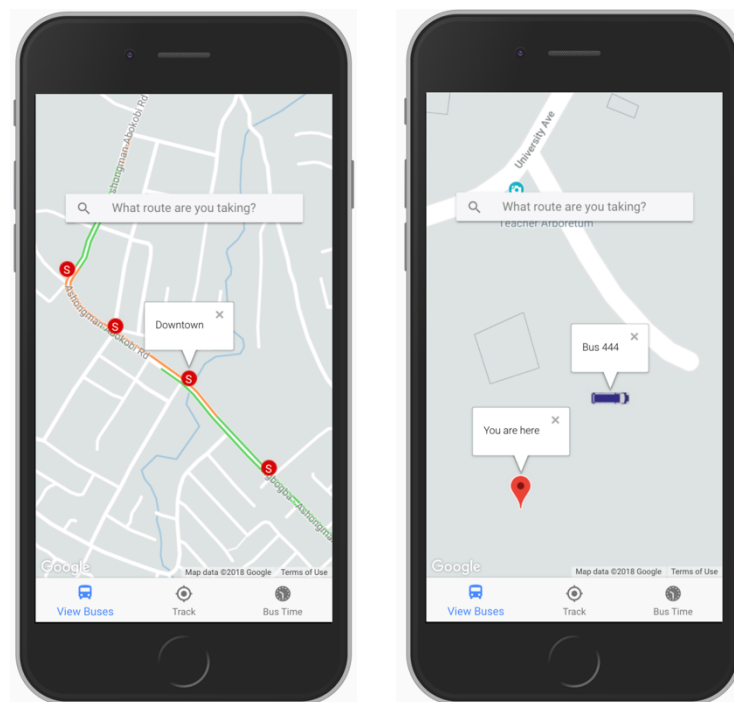


Figure 4.5 Commuter Home Page

The commuter, upon first opening the application, is presented with a map that displays relevant information based on the commuter's location. A commuter's location is acquired from the device's GPS. The map utilizes icons to mark the commuter's current location as well as bus stops and busses nearby.

In order to display the bus stops and buses within walking distance of a commuter, the application uses an algorithm which applies the haversine formula, a mathematical formula used to calculate the shortest distance between two pairs of coordinates on a sphere. The formula is expressed below in Equation 1.1,

$$d = 2r \arcsin \sqrt{\sin^2 \left(\frac{\theta_2 - \theta_1}{2} \right) + \cos(\theta_1) \cos(\theta_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)}$$

Equation 1.1 Haversine Formula (Mahmoud & Akkari, 2018)

Where r is the radius of the chosen sphere. θ and λ are the latitude and longitude respectively (Mahmoud & Akkari, 2018).

Though the earth is not perfectly spherical, the haversine formula is generally used to calculate the approximate distance between two locations on the earth's surface. The algorithm uses the formula to compute the distance between a user's location and all operating busses as well as bus stops. The algorithm then stores and displays on the map all bus stops and busses within a 2-kilometre radius of the commuter's location. The **getClosestBusses()** and **getClosestStops()** functions implement the Haversine formula to return all buses and bus stops respectively. The **addCloseStopMarkers()** and **addCloseBusMarkers()** functions place markers on the map to represent the closest bus stops and bus markers based on the user's location.

4.4.2.6 View ETA of Bus

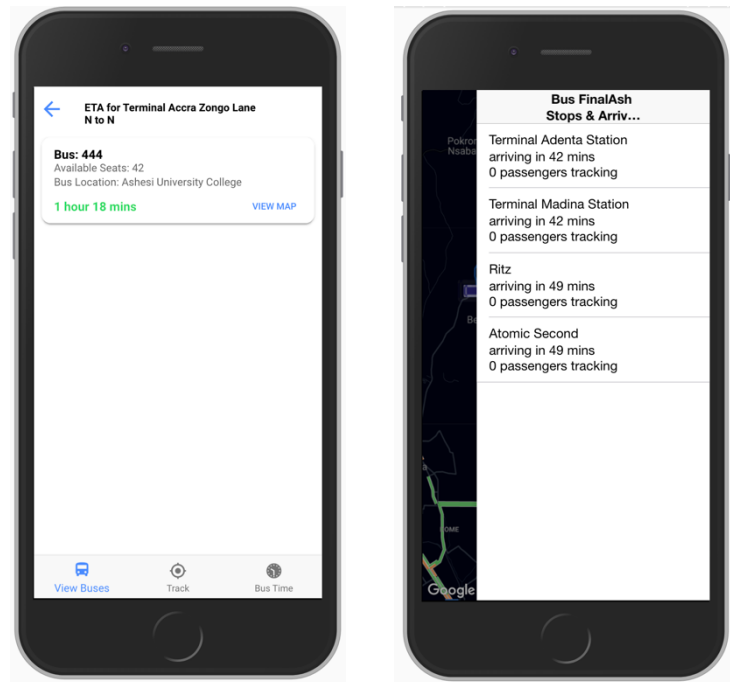


Figure 4.6 ETA Pages

All users of the application can view the estimated time of arrival of a selected bus. The function **getETA()** retrieves the estimate time of arrival between the bus location and a selected bus stop. The function requests and stores the information acquired through Google's Directions Service API.

Chapter 5: User Testing

While developing the application, a series of test cases were conducted to ensure that the final application has optimum performance and functions as a user-friendly application. The test cases and results are presented in this chapter.

5.1 Unit Testing

Individual programming units, specifically functions, were tested with different input parameters to ensure that they functioned as intended. The functions for email and password authentication were tested by inputting both correct and wrong authentication details. The result of this was a success, as the functions only accepted the valid information. The function to sign up a user was tested with cases which had invalid and non-existent email addresses. This test case was also successful as the user is unable to sign up with invalid information. When testing the function which calculates and retrieves the estimated time of arrival, different input parameters were used to test the accuracy of the result. This test case was also successful because the output of the function was verified with known locations and rough estimates for travelling distance. The function for calculating the estimated time of arrival for multiple locations along a route was tested with various number of routes. Test cases with greater than 23 locations failed to return an output. This failure was due to the limitation of the free version of Google's Direction Service API as it only accepts up to 23 waypoints to be inputted as parameters in each request. Another limitation of the system discovered through testing, revealed that Google's APIs had a limitation on the number of queries that can be made in 24 hours.

5.2 Component Testing

The component interfaces, which consists of integrated units, were tested to discover and fix bugs that occurred from interactions between components. In testing whether a transport manager can complete the steps of adding a new route, the module was tested to ensure that the correct information was passed between pages. This tests was successful. Another component test was undergone to ensure that a conductor can select, track and update bus information of a specific bus. The test was successful as the correct information was passed and shown on the respective pages

5. 3 System Testing

The application was also tested to ensure that all components of the system were compatible, interacted in the intended manner and transferred the correct data across components (Sommerville, 2016). To perform test cases for this aspect of testing, the application was deployed on two mobile devices. One device was used by a user who acted as a bus conductor. This user, while travelling in a vehicle tracked and update bus information on the application. The second device was used by a user that acted as a commuter requesting bus information on the application. This user was able to successfully view accurate and up-to-date information about the bus in operation.

Chapter 6: Conclusion

This project sought to use mobile technology to provide real-time transit information to commuters in Accra. The implemented application successfully meets all the requirements of the project. It allows commuters to select any public bus route in Accra and view all relevant transit information concerning that route. This application provides users with a level of certainty about the arrival time of public transport buses utilizing the system.

6.1 Future Works

Though this system meets the needs of each user, future works can be undertaken to address the limitations of the implemented system. The application can be further designed and developed to provide functionalities such as:

1. Finding the most efficient route based on a user's start point and destination.
2. Providing cashless payment functionalities to allow commuters pay for bus rides

References

- Blue light has a dark side. (2012). *Harvard Health Letter*. Retrieved from <https://www.health.harvard.edu/staying-healthy/blue-light-has-a-dark-side>
- Birago, D., Mensah, S. O., & Sharma, S. (2017). Level of service delivery of public transport and mode choice in Accra, Ghana. *Transportation Research Part F: Traffic Psychology and Behaviour*, 46(Part B), 284 – 300.
<https://doi.org/https://doi.org/10.1016/j.trf.2016.09.033>
- Chandurkar, S., Mugade, S., Sinha, S., Misal, M. and Borekar, P. (2013). Implementation of Real Time Bus Monitoring and Passenger Information System. *International Journal of Scientific and Research Publications*, [online] 3(5). Available at: <https://pdfs.semanticscholar.org/1899/f811428dceda59f41a8d59dc0978744577e5.pdf> [Accessed 5 Oct. 2017].
- CSV To JSON Converter. (2018). *Convertcsv.com*. Retrieved 18 April 2018, from <http://www.convertcsv.com/csv-to-json.htm>
- Firebase. (2018). *Firebase*. Retrieved 18 April 2018, from <https://firebase.google.com>
- Google Maps Android API. (2018). *Google Developers*. Retrieved 18 April 2018, from <https://developers.google.com/maps/documentation/android-api/intro>
- Google Maps Directions API. (2018). *Google Developers*. Retrieved 18 April 2018, from <https://developers.google.com/maps/documentation/directions/intro>
- Google Maps Help. (2018). *Support.google.com*. Retrieved 19 April 2018, from https://support.google.com/maps/answer/3092439?api_client=help_panel&authuser=0&co=GENIE.Platform%3DAndroid&oco=1&hl=en

Ionic Framework Guide. (2018). *Ionicframework.com*. Retrieved 18 April 2018, from <https://ionicframework.com/docs/v1/guide/preface.html>

Ionic Native Geolocation. (2018). *Ionic Framework*. Retrieved 18 April 2018, from <https://ionicframework.com/docs/native/geolocation/>

Mahmoud, H., & Akkari, N. (2018). Shortest Path Calculation: A Comparative Study for Location-Based Recommender System. Retrieved from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7791971&tag=1>

Plan a journey. (2018). *Transport for London*. Retrieved 19 April 2018, from <https://tfl.gov.uk/plan-a-journey/>

Sommerville, I. (2016). *Software engineering*. Harlow, England: Pearson Education Limited.

The TroTro Apps Challenge. (2018). MEST Africa. Retrieved 18 April 2018, from <https://meltwater.org/the-trotro-apps-challenge-mest-may-27-28/>