

Ashesi University College

“Catch The Bus:” Facilitating Transportation to and from the Ashesi campus in Berekuso

Ivan Tetteh Digber

2012

Applied Project

Ashesi University College

“Catch The Bus:” Facilitating Transportation to and
from the Ashesi campus in Berekuso

By

Ivan Tetteh Digber

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.

April 2012

Applied Project

Declaration Page

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

Supervisor's Signature:.....

Supervisor's Name:.....

Date:

Acknowledgement Page

Without the immense support of my father, Francis Digber for the resources provided. Thanks to my family and good friends; Bernard Ghartey, Kofi Manful and Edem Kumodzi, this project would not have been a success.

Special thanks go to my project supervisor, Dr Astrid T Larssen for the guidance, feedback and support during the execution of this project.

Abstract

This report presents a transit information system which provides information on the Web, via SMS and to Android based mobiles. The system relies on a web hosting service as a backbone, and it aims to facilitate transportation in all locations in which this transit information system is being used by making transit information available to commuters. This system was developed and tested with a network of privately owned vehicles in Ashesi University College in Ghana. A custom designed GPS application which logs its GPS coordinates to a web hosted service was developed for an Android based phone and placed on a vehicle.

Users can query information from this web hosted service by installing an application on their Android-based phone, via SMS with any non-GPS-enabled cell phone or via the web. Details on Requirements Gathering, System Design, Implementation, Testing and an outline of future work are reported. This system uses the available technology at disposal to provide a solution to the persistent lack of transit information in developing countries.

Table of Contents

Declaration Page.....	ii
Acknowledgement Page.....	iii
Abstract.....	iv
Table of Contents	v
List of Tables	vii
List of Figures	viii
Chapter 1.....	1
1.1 Introduction	1
1.2 Background, Purpose, Significance	2
1.3 Objectives.....	5
1.4 Implications	6
1.5 Approach.....	6
1.6 Motivation	7
1.7 Literature Review.....	8
1.8 Conclusions on Literature Review	13
Chapter 2.....	15
2.1 User Groups.....	15
2.2 Requirements Gathering.....	15
2.3 Functional Requirements.....	16
2.4 Non-Functional Requirements	16
2.5 Design - System Architecture.....	17
2.5.1 Data Storage & Retrieval Layer	17
2.5.2 Dynamic Data Processing.....	18
2.5.3 Data Rendering and Visualization Layer.....	18
2.5.4 User Interfaces	18
2.5.5 Database Design.....	19
2.5.6 Application Transition and Dataflow.....	20
2.6 Start Phase.....	21
2.6.1 Retrieving Bus in Segment	22
2.6.2 Updating Bus Seats/ View Seats and Request via SMS.....	23
2.6.3 Requesting Map.....	23
2.6.4 Post Message/ Request Message.....	24

2.6.5	Log Bus Location.....	24
2.6.7	User Interfaces.....	25
Chapter 3:	Implementation	30
3.1	Development Tools.....	31
3.2	Database Technology	31
3.3	Client and Server Technologies.....	32
3.4	Challenges	33
3.5	Testing & Results.....	34
3.5.1	Unit Testing.....	34
3.5.2	Component Testing.....	35
3.5.3	System Testing.....	35
Chapter 4:	Conclusion and Future Work.....	37
Works Cited	39
Appendix.....		40
Appendix A:	Database Architecture.....	40
Appendix B:	Use Cases.....	40
Observation from Use Case 1.....		41
Observation from Use Case 2.....		42
Observation from Use Case 3.....		43
Observation from Use Case 4.....		43

List of Tables

TABLE 1 COMPARISON OF KEY FEATURES OF TRANSIT SYSTEMS	12
TABLE 2 DIAGRAM OF SYSTEM ARCHITECTURE	17
TABLE 3 TABLE OF DATABASE TABLES WHICH WILL STORE INFORMATION	19
TABLE 4: TABLE SHOWING USE CASES AND THE PROCEDURE FOR EACH USE CASE	40

List of Figures

FIGURE 1: DIAGRAM SHOWING PHASES OF DATAFLOW	21
FIGURE 2: DIAGRAM SHOWING HOW THE SEGMENT WILL BE RETRIEVED FROM A SEGMENT	22
FIGURE 3: DIAGRAM SHOWING HOW SEATING INFORMATION WILL BE RETRIEVED.....	23
FIGURE 4: DIAGRAM SHOWING HOW MAP INFORMATION IS REQUESTED	23
FIGURE 5: DIAGRAM SHOWING HOW MESSAGES ARE POSTED AND REQUESTED	24
FIGURE 6: WEB INTERFACE SHOWING A MAP WITH BUS INFORMATION, CTB MESSAGES BY USERS AND TRAFFIC TWEETS.	25
FIGURE 7: WEB INTERFACE WHICH ALLOWS ALL USER GROUPS TO POST CTB MESSAGES.....	26
FIGURE 8: MOBILE INTERFACE OF BUS DRIVER OR PRIVATE VEHICLE OWNER OR CONDUCTOR.....	27
FIGURE 9: MOBILE INTERFACE OF BUS RIDER	28
FIGURE 10: VIEWING AND POSTING MESSAGES VIA THE APPLICATION..	29
FIGURE 11: SYSTEM OVERVIEW.....	30
FIGURE 12: DATABASE ARCHITECTURE OF CATCH THE BUS.....	40

Chapter 1

1.1 Introduction

Transportation from one location to another requires the right kind of transportation at the right time. Every day people use various transportation services for different purposes. Access to these transportation services leads to mobility of people for work and recreational purposes, as well as mobility of goods and services. Access to transportation is also critical to timely and affordable delivery of services such as health care and education. This can empower vulnerable groups by increasing their independence, and is key to maintaining social networks (Anderson et al., 2010).

Aside from the access to transport services, commuters need to have access to information about its availability in order to make informed transit decisions. In the developed world, commuters can often access information about bus and train schedules via web pages, SMS/text message, phone or printed schedules which are maintained by transportation authorities. Information which can be accessed includes the real time location of buses, commute cost, traffic and road updates, transit alerts and expected arrival time (Metro Transit, 2012). Most of these solutions rely on some form of central infrastructure to provide the servers the data needed to respond to queries from commuters. Commuters may find transportation information insufficient if such an infrastructure does not exist in a community.

This report discusses the design and implementation of a transit information system which makes use of current available technology to provide information to commuters. "Catch the Bus" is a locally developed transit information system which makes use of GPS, SMS and webhosting technologies. While this project did not involve a full scale deployment, the design, implementation and system tests provide a framework for how to create technological solutions that can be introduced in communities. This report can also serve as reference for other developing regions with limited resources, resource-constrained transportation companies and ad-hoc transportation resources for both inter-city and intra-city transport.

1.2 Background, Purpose, Significance

Public intercity transportation in Ghana is primarily made up of taxis, "trotros" and the Metro Mass buses. Trotros are the most frequently used means of transport because it is a cheaper option. The trotro is operated by a driver and a conductor or 'mate'. To board a trotro, one has to wait at a designated bus stop for one to come by. The conductor usually signals to passengers by waving a hand in a specific direction. This hand gesture gives passengers an indication of the route which the trotro will use. The conductor also shouts the destination or the major bus stops where the trotro will stop on its route to a destination, for example "Circle, Circle, Circle" indicating that the trotro is heading to the Kwame Nkrumah circle. In other cases, passengers wait at lorry parks and form queues waiting for the next available trotro to come by or maybe stop a taxi.

These transport modes do not rely on a set schedule which implies that commuters have to plan and be hopeful that they will get a trotro to their various destinations. Planning their schedules involves monitoring traffic and peak travel times over periods of time to discover transit patterns on the routes they frequent. These patterns deal with knowing times when there is available seating on trotros or determining how long a trotro in transit will take to arrive at a destination. During peak hours, commuters have to wait for long periods in queues to board a vehicle from one location to another. Also, these public transportation modes do not always use a fixed route which sometimes makes access to them difficult. This in turn makes planning difficult for the commuter since there no access to route information.

In Berekuso the main public modes of transport include a few trotros, a lot of taxis and one Metro Mass transit bus. Ashesi's move to Berekuso from Labone has had transportation issues. Initially, the most common mode of transportation from Accra to Berekuso and vice versa was via the school bus. The bus had a fixed route, stopped at specific bus stops and followed a schedule. This was until it broke down in early February 2012. This breakdown caused other private transport services to spring up, the most notable is GLIDE which stopped operations only after a few weeks. For most students, faculty and staff who do not have their own cars, transportation to the Berekuso campus is challenging. However, it is even more the difficulty getting a ride back to Accra from Berekuso.

Getting back to Accra involves calling up taxi drivers to find their locations, availability and their schedules. Based on this information, members of the community make decisions whether to board an available taxi from Berekuso to their destination. Also, there are cases where members of community are willing to give rides to other members headed in their direction. Unfortunately, there is no platform which allows riders to communicate with the bus driver or owner of a vehicle.

Though commuters of the Ashesi community currently find their way to campus and back to their homes one way or another, putting in place a tool that could help facilitate transportation, would improve the commuting for members of the Ashesi community. As a result, the amount of effort committed to organizing transportation will be reduced. Vital information that has to do with available seating on the bus and real time location of the bus will be provided to users of the tool. The full purpose and the potential of tool will greatly depend on the accuracy of the information provided to the users.

With the full implementation of this tool, users will have more control over their trips and schedules since they will be able to access necessary information. Also, people who are new to the Ashesi campus will have less difficulty moving to the campus and back. Users will be provided with a variety of transit schedules of vehicles either heading to or coming from a location depending on the availability of the vehicles.

1.3 Objectives

Since specific information such as the location of the bus and available seating are necessary to guide bus riders make transit decisions, the objectives of this project will be streamlined towards providing users with relevant information. The information will also need to be visually represented to make comprehension easy. Below are the objectives of the project.

Log the geographic coordinates of the bus.

Logging the coordinates of a vehicle is an essential component of this project. Information obtained from logging the coordinates as well as time and speed can help with predicting the arrival time. This will be done by building an Android based application which will sit on the bus logging the required data as the bus moves.

Present Information to Users

This involves building a website and a mobile application. The website will provide visual information about the bus stops, the bus location and arrival time of the bus on a map. All of these will be supported by some web service of the application. Google Maps® will be used as the mapping tool in this case. The mobile phone application, on the other hand, will be used by the bus driver as well as the bus rider. The bus driver will use the mobile phone application to communicate with bus riders by providing them with the necessary information. The bus rider will use the mobile phone application to view seating and real time location information and

also communicate with the bus driver. This mobile phone application will be built specifically for the Android platform.

Provide Information to Users with Low End Phones

For users who do not have Android based smartphones, bus information will be provided to them. This will involve sending a text message with a key word to a phone number. A response with the bus information will sent back to the bus riders.

1.4 Implications

A number of implications arise from the successful implementation of this project. First of all, to make effective decisions, one needs to have information about the situation at hand. This tool will provide its users the relevant information needed to make transit decisions. Second, transport services provided to the Ashesi community in Berekuso can be better coordinated to improve transit satisfaction. Finally, the success of this project will provide an essential component to a model which can be used to improve intercity transit in Ghana.

1.5 Approach

For this project, I intend to build transit information system which will provide relevant information to bus riders. First of all, I will find out the potential user groups of this system and interact with them to obtain the different requirements. These requirements will help obtain a focus for the project and identify the various components which will make up the

system and the different ways in which the components will interact. Information on the location of the bus requires a GPS enabled device which will log the coordinates of the bus. This GPS device will obtain the location of the bus from a GPS satellite and log them to a central server. Available technologies capable of providing location based services will be considered. These services can be customized depending on the location of the user (Manful, 2009). Examples of these include showing the location of the bus and the various bus stops on the map based on an identified location. Testing will be carried out on each of the identified components of the system to ensure they function as expected.

1.6 Motivation

Two factors motivated me to take up this topic. First, Ashesi's move to Berekuso from Labone has had transportation issues. One of them is catching the bus on its way to Berekuso from different locations. Being able to access a system that would accurately provide users with information on the bus schedule will reduce the amount of effort put into catching the bus. Also, new bus riders who are not conversant with the bus schedule at the various bus stops can easily have access to this information in order to plan their schedule.

Second, this project could be expanded to cover the public transport sector of Ghana to improve the public transit system. Most vehicles involved in public transport do not have a schedule which they operate, therefore making it difficult for riders to plan their routes. This tool should in the long run increase the rider transit satisfaction.

1.7 Literature Review

Transit information solutions have been developed to different levels based on the technologies and contexts in which they exist. These transit information systems also depend on the availability of relevant data required in order to provide accurate transit information. These solutions vary from country to country depending on the level of technology and the availability of data relevant to providing close to accurate transit information. Below is a summary of related work aimed at providing transit information solutions;

OneBusAway®

OneBusAway® was developed by Brian Ferris, Kari Watkins and Alan Borning at the University of Washington, Seattle. OneBusAway® is made up of a set of tools, providing access to real-time transit information for Seattle bus riders through a variety of interfaces, including web, phone, SMS, and mobile devices (Ferris et al, 2009). The OneBusAway® system is subdivided into Route Maps and Timetables, Real-Time Tracker, Service Alerts, Trip Planner and Interface Modalities.

The relevant features to my project include the real-time tracker, service alert, various interfaces to real-time tracking data. These interfaces include telephone numbers which users can call to have arrival information read to them, an SMS interface for receiving arrival information as text messages, a standard web interface and a website optimized for internet-enabled mobile phones.

For OneBusAway®, it was discovered that most traveller information systems do not use the full complement of communication modalities to notify their users in cases of change in schedules (Ferris et al, 2009). Therefore a service alert system which is integrated with route timetables, trip planner and time tracker components will provide notifications to their users on route changes and diversions. OneBusAway® is also considering the use of crowd-sourced information which would allow riders to share information about service changes as they happen.

OneBusAway® is written in Java and uses a variety of standard open source development libraries and frameworks in its implementation. Other tools and technologies used include the Apache Struts MVC web framework, AJAX, Google Web Toolkit, Asterisk PBX server, FastAGI and a text-to-speech engine which is licensed from Cepstral.

Current and future research by OneBusAway® includes Automatic Learning of User Travel Patterns, Automatic Notification, Value Sensitive Design, Transit Travel Behaviour and Real-time Arrival Prediction Accuracy.

Tiramisu® – The Real-time Bus Tracker

Tiramisu is a crowd powered application aimed at helping people make better use of the public transit. It empowers people by allowing them to track buses in real time. This is done by authorizing a GPS enabled phone to log GPS coordinates to a server until it is stopped. This will also provide

other users with dynamic and accurate information about the location of a bus. Information about the fullness of the bus is reported by its users also. When using Tiramisu, the user can see bus information through a map, lists of nearby stops, and favourites that you pick. Tiramisu also lets the user rate bus fullness and report suggestions, problems, and positive experiences to the transit agency, with or without pictures. This application is associated with research at the RERC on Accessible Public Transportation at Carnegie Mellon University (Tiramisu Transit LLC, 2011).

Star-Bus (*Bus)

Star-Bus® was developed by Ruth E. Anderson, Waylon Brunette, Erica Johnson, Caitlin Lustig, Anthony Poon, Cynthia Putnam, Odina Salihbaeva, Beth E. Kolko and Gaetano Borriello in Kirgizstan. Kirgizstan is a developing country which has many challenges including access to information. This team provided a grassroots solution to the persistent lack of transport information by using the technology which could be leveraged for small scale, institutionally-independent transportation information systems (Anderson et al, 2010).

Research by the team on mobile phone usage indicated that it was growing at a remarkable rate, including in rural areas. SMS was used regularly, while internet usage was not the primary mode of ICT access. This guided the development process and emphasized the need to build an SMS-based transit information system. This is because SMS-based solutions have also proven robust, flexible and valuable in multiple contexts (Anderson et al, 2010). Interviews were also conducted with bus

riders to provide more specific feedback on marshrutka (privately run buses) and SMS use in order to give insight into the proposed solution. Interviews with bus drivers were also conducted to inform the development team of Star-Bus® of their understanding of the marshrutka system and thus how their system might fit economically and logistically within that system.

The solution proposed by Star-Bus® System factored in the technology usage patterns and technology infrastructure available in resource-constrained environments. The system requires each bus driver in the transportation network to have a star-box device installed on their bus. This will automate information gathering. The star-box sends location update messages to a star-bus server. The star-box can be turned off and on. To minimize driver interaction and improve maintainability of the system, an interface which allows remote control of key controls via text messages sent to the star-box. Other methods of sending location updates were considered due to the high cost of sending SMS. Some of these methods include USSD messages, poll methods and reporting only when the marshrutka deviated from its arrival time from a normal predictive model.

The Star-bus® server is purposed to receive and process updates from the star-boxes and queries from riders. It therefore does not require the use of internet, facilitating its deployment in areas with low internet connectivity. The server was implemented using MySMS, an application framework built on top of SMSLib. Initial testing was done in English but

the server has been designed to accept queries and send messages in alternate character sets. The server currently handles *location update messages, queries about bus arrival time and user geo-coding requests* all via SMS.

Star-Bus® was tested by deploying the star-boxes on multiple bus routes and multiple runs of the same route. Usability testing was also done. Some of the issues that came up include SIM Card and Mobile Provider Network Differences, SIM Message Latency and Drop Rates and Need for Feedback on State of Hardware, Timing similarities and prediction accuracy, geo-coding locations with star-boxes (Anderson et al, 2010). For future work, Star-bus hopes to focus on developing a location based business model in preparation for large-scale deployment of the service. Below is a table comparing the various transit information systems that exist.

Table 1 Comparison of key features of transit systems

	OneBusAway®	Star-Bus®	Tiramisu®
Real Time GPS Tracking	✓	✓	✓
Mobile Application	✓	—	✓
Web Application	✓	—	✓
SMS Application	✓	✓	—
Predict Arrival Time	✓	✓	✓
GPS Equipped Bus	✓	✓	—

Rider Input	—	—	✓
Driver Input	—	✓	—
Service Alerts	✓	—	✓
Seating Information	—	—	✓

1.8 Conclusions on Literature Review

The advanced nature of technology which include fast internet and GPS enabled buses, coupled with the availability of traffic and transit information, makes it relatively convenient for OneBusAway® to provide real-time transit information. This can be seen in the future work that has been laid out, as they can be described as very technical and sophisticated.

Tiramisu®, which is crowd sourced based, relies heavily on users to provide other users with transit information. This approach is subject to errors and misleading information could be provided by users to other users of the application. It can be used only by users who have smartphones, since the application has been built for Android and Apple IOS platforms. Alerts can be sent to users of the application based on what other users have reported.

Star-Bus®, on the other hand, provides a solution which would better suit the context of Ghana. One major feature which was of interest was the use of text messages to inform bus riders on the status of the bus if they sent a query. The approach by Star-Bus® will be adopted while adding

some features from OneBusAway® and Tiramisu® to implement CatchTheBus®.

In the following chapters of this report, details of the requirements gathered from the respective stakeholder and information on the design choices – Chapter 2; methods of implementation and testing and results – Chapter 3 as well as the conclusion – Chapter 4. Each of these topics will be broken down into their various sub categories where necessary in order to provide a detailed description of the components of the project.

Chapter 2

The core functionalities to be provided by the application is providing real time information on the location of the bus on a map and indicating the available seating on the bus. Others functionalities include receiving and sending broadcast messages and predicting the arrival time of a bus at a bus stop. There are other services that can be built on this core. One of them has to do with reserving seats by making advanced electronic payments and basic search for buses or bus stops. However, while they will be mentioned, not all of them will be implemented since they are outside the scope and time restrictions of this project. Details on these services are provided in the section titled Further Work.

2.1 User Groups

In this application, there are four main user groups.

1. Bus Rider: This user seeks to board the bus from one location to another.
2. Conductor: This user assists the bus driver on every bus trip.
3. Bus Driver: This user drives the bus from one location to another.
4. Private Driver: This user uses his or her private vehicle to commute from home to Berekuso and back.

2.2 Requirements Gathering

Requirements for this project were gathered mainly through interacting with all four user groups of the application.

2.3 Functional Requirements

The application is expected to perform the following functions listed below;

- I. The application must be able to display a map showing the major bus stops.
- II. The application must be able to display a map showing real time location of the bus.
- III. Bus riders must be able to receive broadcast messages sent from the bus.
- IV. Users of the application must be able to send messages to a central database where its users can access.
- V. The application must be able to log the real time location of the bus.
- VI. Conductor must be able to tally people on the bus and indicate the capacity state of the bus.
- VII. The application must be able to predict the bus arrival time at a bus stop.
- VIII. The bus driver can indicate the available seating on the bus as well as update it where necessary.

2.4 Non-Functional Requirements

These are requirements that specify how the application should perform the functions stated above.

- I. The application should not burden mobile phones with excessive processing that could deplete battery power.

- II. The application should adopt a prediction algorithm that would best use of available data.

All of these requirements have been stated to help meet the design goals that have been set for this project.

2.5 Design - System Architecture

The system architecture of Catch The Bus will be made up of a four-tier client server design with one layer dedicated to serve at the user end. This architecture will be able to support cross client platforms such as the web and different mobile interfaces. The burden of processing and storing data will be dedicated to specific layers in order to meet the requirement of not burdening the mobile phones with excessive processing. Also since data that are stored in the database is used by cross platform users, data sharing becomes relatively simple. This is illustrated in the diagram below.

Table 2 Diagram of System Architecture

Web Interface Mobile Interface - Smartphone (Android)
Data Rendering & Visualization
Bus Segment Management Map Management Location Management (Data Storage & Retrieval)
CTB Database - Data Storage & Retrieval

2.5.1 Data Storage & Retrieval Layer

This layer will manage interactions with the database. The layer will be responsible for loading all data needed from the database & storing any

data in it. Data to be stored include bus details; *GPS coordinates, speed of bus, segment location, real time messages and bus stop details*. This is to ensure that there is a single point of entry to the database. This will provide means to secure this layer and monitor any activity that interacts with the database. The database itself will form one tier upon which the DSR layer will be established.

2.5.2 Dynamic Data Processing

Data loaded from the DSR layer and sent to this layer for processing. This layer will process requests for real time bus arrival information, updating the map with real time location of the bus, updating bus riders with available seating, messages, granting permission to bus drivers who will provide information on seating. Drivers who use the application will have their locations logged and sent to the DSR layer for storage.

2.5.3 Data Rendering and Visualization Layer

This layer will make available to the various interfaces the results of the Dynamic Data Processing layer. The results will be rendered in a manner that will be efficient to the interfaces. This layer will also receive requests from the user interfaces and pass them to the DDP layer for them to be processed.

2.5.4 User Interfaces

This layer will display a map with bus stops, real time bus location, available seating and other bus travel details in a web browser. Also, a

mobile interface will be developed to make the above features available. The Data Rendering and Visualization layer will be structured so that development for other different technologies could be easily implemented.

2.5.5 Database Design

The database must be capable of providing the bus rider with the necessary information about transit. In order to do this it must be able to store and retrieve data associated with the following entities; bus stops, predicted arrival time, available seating on the bus, location of the bus on a map and broadcast messages. These entities were obtained from the application requirements and functions. The tables derived from the attributes of the objects were then normalized. The resultant structure of the tables in the database that will hold the data is shown in the Appendix.

Table 3 Table of database tables which will store information

Table Name	Description
Bus	This holds data relating to bus details. It includes the bus name, bus driver, bus latitude and longitude, bus speed and available seats.
Bus_schedule	This holds data relating to the schedule of the bus. The predicted travel time and destinations of each bus are stored in this table.
Bus_stop	This table contains information about all the major bus stops through which the bus goes through. This information is plotted on a map powered by Google Maps®
Gps_device	This table contains information about the GPS

	devices that track the location of the bus.
Location	This table logs the GPS coordinates, speed of the bus and time when the information was logged with respect to the GPS device. Information obtained from this table will be used to update the location of the bus and display it on the map.
Messages	This table will store broadcast messages that are sent to bus riders.
Segment	This table will store information about the segmentation which is used to determine the segment in which the user or bus is located.
Segment_has_bus	This table is updated with the bus if it enters a segment.
Users	This table stores data on users who are authorized provide information to bus riders.

2.5.6 Application Transition and Dataflow

The following diagrams show the different paths that can be taken in the application and how data is transferred between application stages.

2.6 Start Phase

Diagram showing all the possible phases the application will go through.

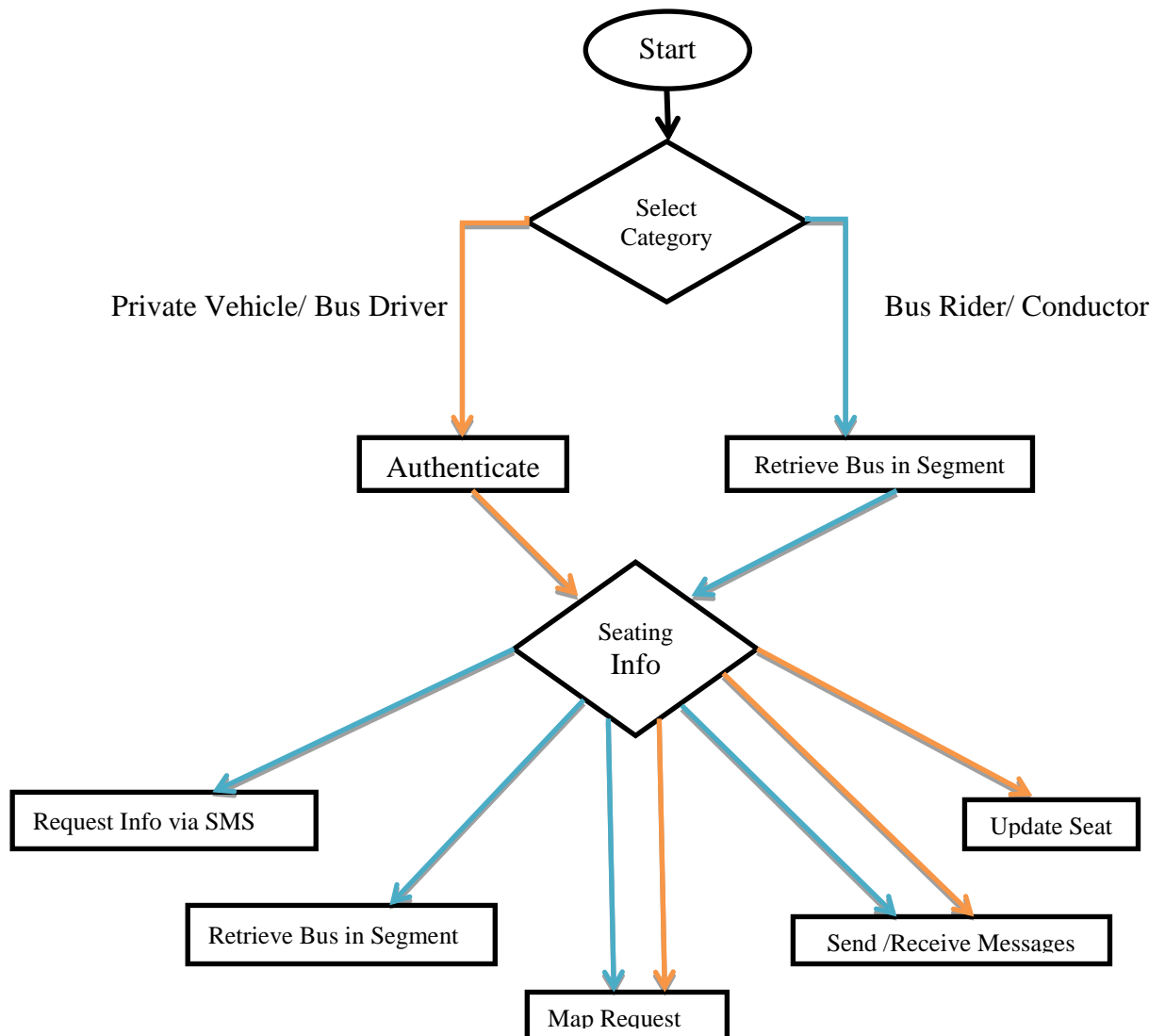


Figure 1: Diagram showing phases of dataflow

When the application starts, the user is prompted to choose a category. If the user is a bus rider, the GPS location of the rider is compared to the available segments in the database to determine which buses are in the rider's segment. An interface showing seating information on a selected bus is presented. The bus rider can then choose to request the buses that are available from a list of predefined segments, receive broadcast

messages, indicate his or her location on a map and request a map showing his or location as well as that of the bus. The bus rider can query the location of the bus via SMS and get response via SMS also.

If the user is a bus driver, he supplies his credentials, and provides information about the bus or vehicle he or she is driving as well as available seats. Once all this information is provided, the driver can update seat information and request a map showing his or location on the map, as well as riders requesting available information about the available buses.

2.6.1 Retrieving Bus in Segment

Phase to retrieve the bus in a defined segment.

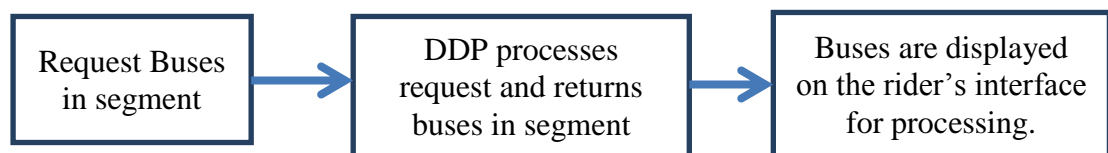


Figure 2: Diagram showing how the segment will be retrieved from a segment

Information about buses that are in the same segment of the user is presented to the user. This is done by retrieving the coordinates of the user and comparing it to the predefined segments that have been setup in the segments table. With the identified segment of the user, buses located within the identified segments are presented to the user for further processing.

2.6.2 Updating Bus Seats/ View Seats and Request via SMS

Phase to update and view seating information.

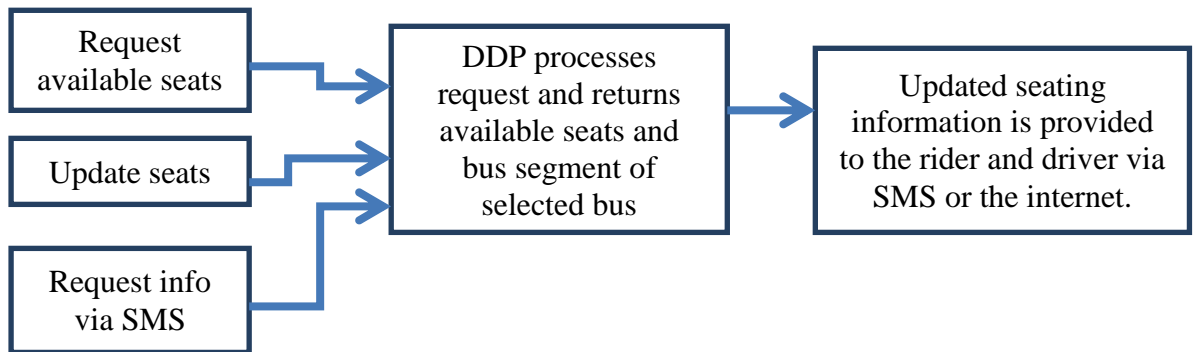


Figure 3: Diagram showing how seating information will be retrieved

Whenever a rider boards the bus or hops off the bus, the number of available seats is also updated, either by a decrease in seats or an increase in seats. The bus rider is able to retrieve the latest available seating on the bus by sending a request to the server.

2.6.3 Requesting Map

Phase to request map information

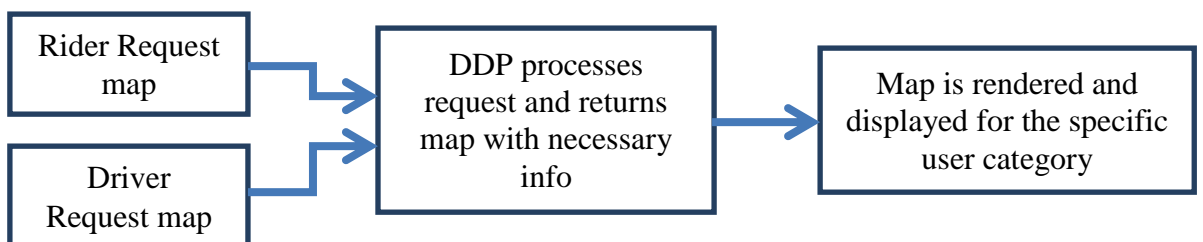


Figure 4: Diagram showing how map information is requested

Bus riders and bus drivers can request for a map that will show the rider's current location as well as the real-time location of the bus. First of all, the GPS coordinate of the rider is sent to the DDP layer while the GPS

coordinates of the bus will be retrieved from the DSR layer. Data from the bus rider and the DSR layer will be processed and a map will be returned indicating a marker of the rider and the location of the bus.

2.6.4 Post Message/ Request Message

Phase to post and retrieve messages

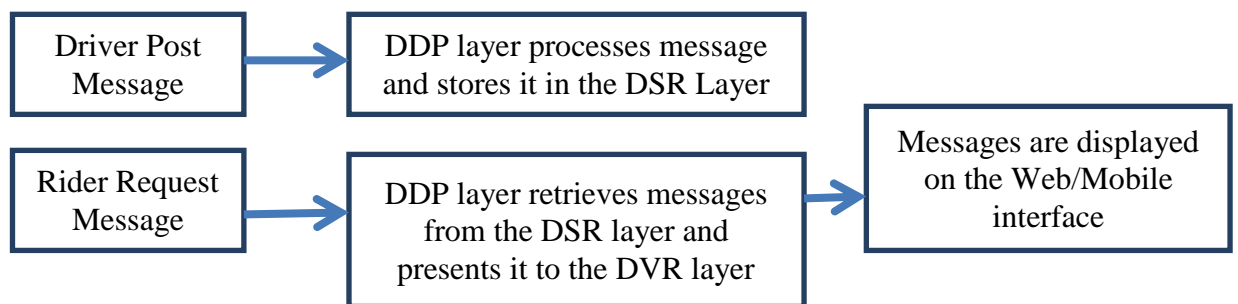


Figure 5: Diagram showing how messages are posted and requested

Bus drivers can post messages to the messages table about situations on the routes, traffic information and so on. Messages posted by bus drivers are accessed by bus riders. At the moment, bus riders can only view messages. Messages displayed to bus riders will be restricted to a particular day. This will be done to provide relevant information to bus riders.

2.6.5 Log Bus Location

A GPS enabled device which has the capability of logging the real time locations is put on the bus. It retrieves its GPS coordinates after every 40 seconds and logs the latest coordinates to a database. Information stored to the database includes the longitude, latitude, bus speed as well as the time of logging the coordinates.

2.6.6 Update Bus (Location, Segment)

After every thirty seconds, the most recent coordinates logged by the GPS logger update the coordinates of the bus. While this is being done, the segment of the bus will be determined and updated if necessary.

2.6.7 User Interfaces

The interfaces, which will be presented to users of the application, are detailed below. They include both mobile and browser interfaces.

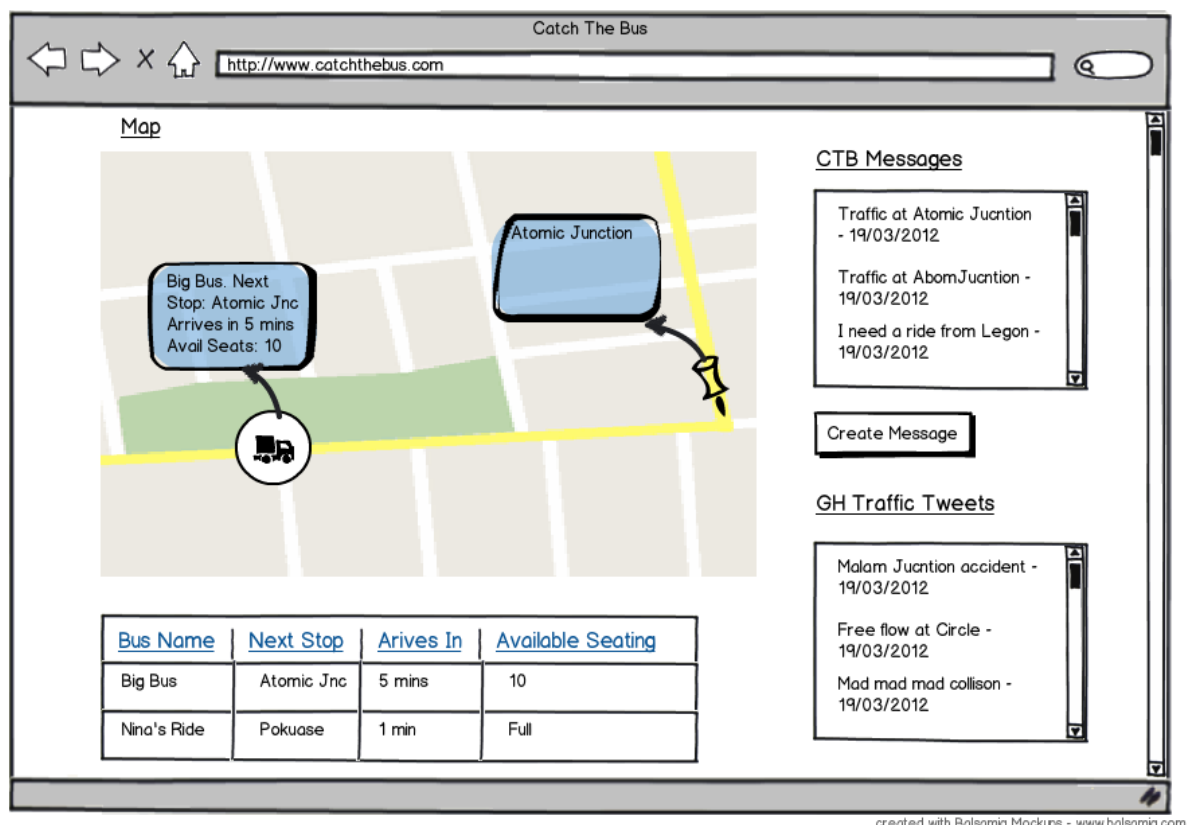


Figure 6: Web interface showing a map with bus information, CTB messages by users and traffic tweets.

From the web interface, users can access real time bus information in a table. Messages that have been sent by users of Catch the Bus can be viewed from the website. The messages displayed are limited to a

particular day in order to present relevant information to the user. General traffic information from the @Gh_traffic Twitter account is also displayed on the interface. Users can also send messages by clicking on the create message button. They are required to fill in the necessary fields before their messages are posted. This is shown in Fig.7.

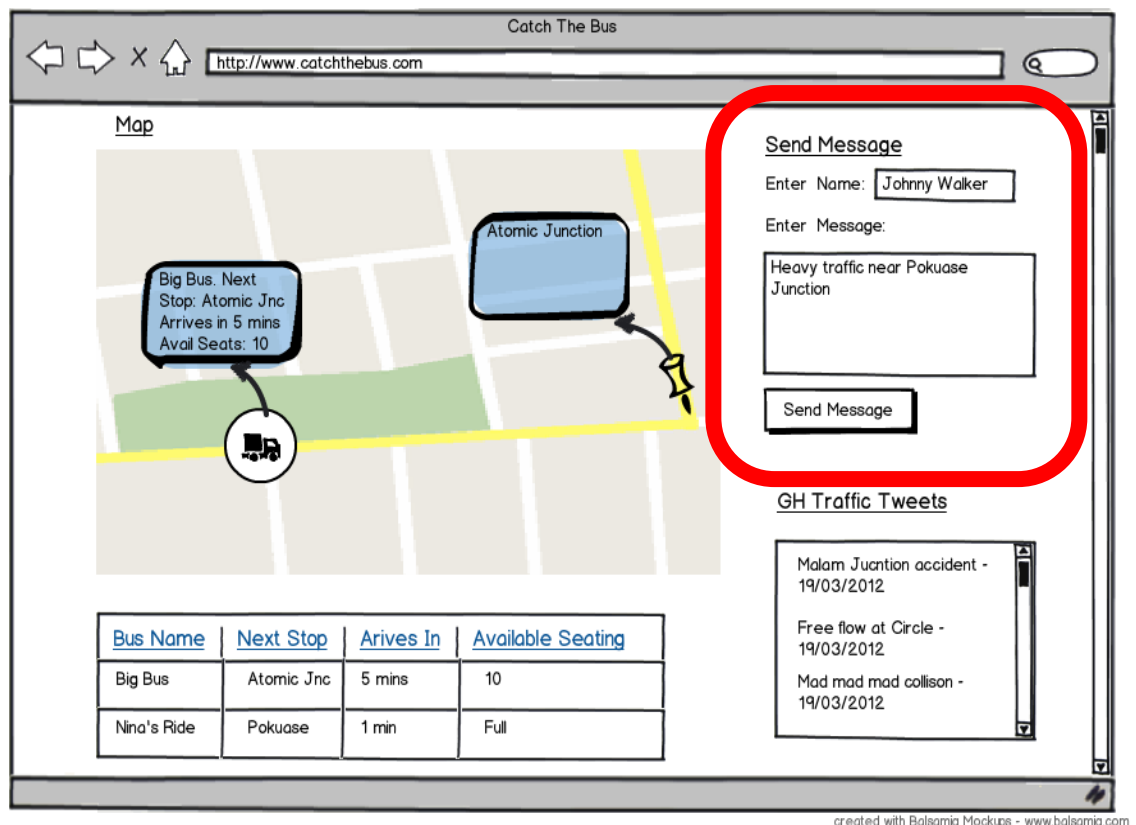


Figure 7: Web interface which allows all user groups to post CTB messages



created with Balsamiq Mockups - www.balsamiq.com

Figure 8: Mobile interface of bus driver or private vehicle owner or conductor

The interface in Fig.8 is presented to the driver/conductor when the application is accessed on the Android phone. The driver/conductor selects the bus driver option, which then requires that credentials are entered for authentication. The vehicle number, as well as the available seating, is entered into the application. Once this is set up, the driver/conductor can update the available seating by increasing and decreasing the seat count. The driver/conductor can also view the location of the vehicle on a Google Maps map. The driver/conductor can send messages or view messages by users of the application. One major challenge with this design is how the

driver will update the available seating while driving. Ideally, the driver would not want to be distracted when driving. A distraction could lead to an accident which must be avoided at all cost.



created with Balsamiq Mockups - www.balsamiq.com

Figure 9: Mobile interface of bus rider

The interface in Fig.9 is presented to the bus rider when that option is selected. The user then requests a list of available buses or vehicles that exist with the segment of the user. Once any of the buses is accessed, information on the available seating and the arrival time is shown to the user.



Figure 10: Viewing and posting messages via the application

The user can also access messages that have been sent via the application or post messages as shown in Fig.10. Users who wish to post messages will have to supply their names and the message. Their names will be necessary to tag every message that is being sent via the application.

The above interfaces will be served with information from the Data Rendering and Visualization layer. All processing is done by the server, therefore limiting the amount of processing done by the mobile phone and the web interface.

Chapter 3: Implementation

The figure below is an overview of the application briefly showing how the different components will interact.

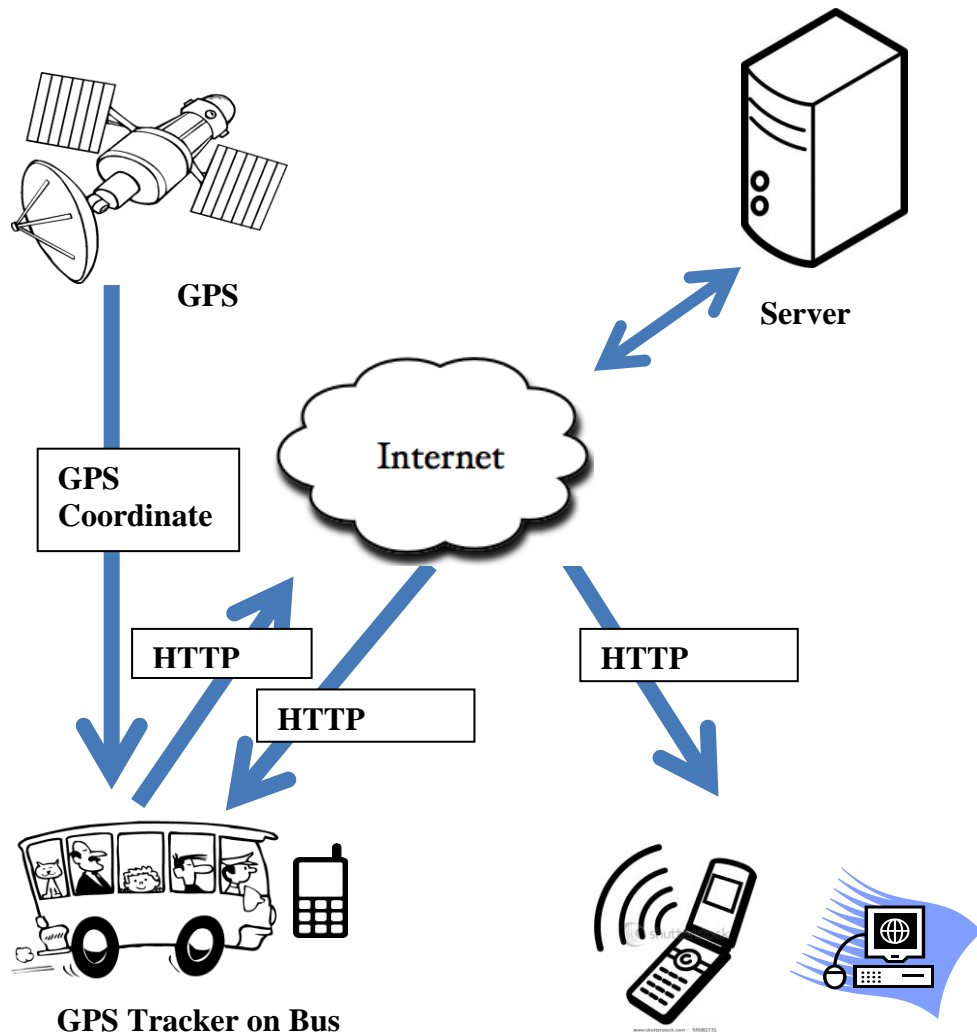


Figure 11: System Overview

GPS coordinates are obtained from a GPS satellite and logged to a server over the internet. Information about seating and messages are also sent to the server via the internet. The bus driver can also request information about his location on a map and messages that have been posted by bus riders. Bus riders will need to fetch information about available seating,

bus location and messages from the server over the internet. This will be done via the mobile phone or web browser.

3.1 Development Tools

Since development was being done for the Android platform, the tools and API's provided by the Software Development Kit for development requires the use of the Java programming language. The Java programming language is ideal because of the extensive documentation and examples that exist. Also, my familiarity with the Java programming language made it comfortable to work with. Eclipse's IDE for Java Developers was used in compiling the Java code. This required downloading version 10 of the Android SDK. The Notepad++ text editor was used to create webpages and server side scripts. The webpages and server side scripts were created using HTML, Javascript, AJAX and PHP. Plugins from GoogleMaps®, JQuery and Twitter were also used.

3.2 Database Technology

The database management system used for implementation was MYSQL. The main reasons for this choice were the relative cost in comparison to other DBMS applications such as ORACLE. My familiarity working with the MYSQL DBMS, its high performance, high reliability and ease of use are other reasons for selecting it. This satisfies the DSR layer specified in section 2.5.1 of the design of the application. To access the information in the database from the Android Platform, the Apache HTTP package was used to access a server side script which connected to the database and

retrieved the necessary information. AJAX was used in the web browser to make similar calls to access server scripts to fetch data from the database.

3.3 Client and Server Technologies

Two interfaces were provided for the client side of the application, the web and mobile. For the web interface, a combination of HTML, Javascript and CSS was used for development. HTML was mainly used to present the information to the user in an organised manner. Javascript added interactivity to the webpages since HTML pages are mostly static. The interactivity had to do with sending messages, updating the location of the bus on a map and viewing Twitter feeds. AJAX was used to send requests the server side scripts which when executed returned results to the webpages. For the Android platform, the Java programming language and the necessary packages were used to implement the logic.

PHP – Hypertext Preprocessor, a server-side scripting language executed on the server. This was used since it supported interacting with the MSQl DBMS. One of the requirements was to reduce the burden on mobile devices by restricting computation to the server. PHP will be used to achieve this requirement. All computations will be done in the Dynamic Data Processing layer as described in section 2.2.3.2 of the design of the application.

3.4 Challenges

During the implementation of the design, a couple of challenges were encountered. They were either resolved or bypassed, which required changing the technology used to implement the solution. The challenges involved determining the segment of the bus and displaying the map on the Android phone.

3.4.1 Segmentation/Determining segment of Bus

Google's Fusion Tables was to be used to store segment information which could be queried to determine the segment in which the bus was located. This technology is fairly new and after several failed attempts to setup, this technology was abandoned. As a result, segment information was stored in MSQl database and simple method of querying the segment information was used. Two major bus stops were used to define a segment. A rectangle using coordinates which specified bottom left-corner, bottom right-corner and top right-corner, top left-corner was drawn with the two bus stops located inside it. The GPS-coordinates of the bus of the bus which is made up of a longitude and latitude was compared with the coordinates of the rectangle to find out if it fell within the range. This method worked and the challenge was dealt with.

3.4.2 Maps Display on Android

Ideally the position of the bus that shows on the Android map should change with respect to the latest GPS coordinates logged. Achieving this objective proved to be quite challenging. Any time a new location of the

bus was acquired and inserted on the map, the old position could not be removed. A number of reasons accounted for this. First of all, there was a mix up with the coordinates that were returned to the database. Second, there were a number of functions from the Java Android Maps Overlays package which could be used to remove an item on the map. The file which returned the coordinates of the bus was fixed and the right coordinates were returned. A lot of reading online and trial and error were used as a method to fix the display problem on the map.

3.5 Testing & Results

Development Testing was carried out on the system. This method is a defect testing process which aims at discovering bugs in the application and fixing them or changing the program to fix any problems. Unit testing, Component testing and System testing collectively make up development testing. Full testing could not be completed due to time constraints but a full detail on testing that was carried out is listed in the Appendix.

3.5.1 Unit Testing

This involved testing the functionality of the functions, objects and classes that were used in the application. For example, there is a function that performs an HTTP connection to a web server to fetch information on the location of the bus. This function was tested by executing it to make sure the right kind of data was obtained. It failed to handle bad data that were returned to it. This required that the function be debugged and fixed. The

data that were returned from the webserver were checked to make sure it conformed to what would suite the function. If it did not suite the function, an error message was sent out to the user. This approach was taken to test all the other functions, objects and classes that were used in developing the application.

3.5.2 Component Testing

This method of testing is made up of testing several individual interacting units of the application. A combination of these objects forms different kinds of interfaces. The Parameter interface and the Message passing interfaces were identified after the units were combined. The parameter interface involved passing data from one component to another while the message passing interface dealt with components requesting a service from another component. After testing, some of the issues that came up involved delays with responding to the client from the server. This was mainly due to the poor network coverage by the telecom companies. Details of trials have been added to the appendix.

3.5.3 System Testing

The different components of the application were integrated to create a version of the application which was tested as a system. Use cases were developed in order to test the interaction between the various components. Use cases were a preferred method because each use case is implemented by several components of the system. One of the use cases was made up of users seeing the real time location of the bus on a map as it moved from one point to another. This involved using a component that

displays the information on the phone and another that fetches the information from the web server. Another use case also involved users querying the location of the bus via SMS and getting a response of the segment in which the bus was located and information on available seating. This involved a component that logs the GPS coordinates of the bus to a web server and another that queries the web server for the bus location via SMS. Test trials and their results have been attached to the appendix of the document.

Chapter 4: Conclusion and Future Work

The above report presents a solution which aims to combine data from cellular GPS satellites, SMS and the internet to provide a transit information system. The aim is to make transit information available to commuters who are members of the Ashesi community. The above solution borrows from existing transit information system, including Starbus, Tiramiusu and OneBusAway. The solution was developed by making use of the available technology. The "Catch the Bus" solution aims at reducing cost of using multiple GPS devices. Since the same device used by the driver to update seating information is also used as GPS tracker, no extra cost is incurred to obtain GPS device. Also, the method of obtaining the location of the vehicle works well for this context, since there is little or no information on street names and other locations on Google Maps®.

There still remains some future work on the different user interfaces and some of the functionality supporting the Android maps interface. With the design of the system architecture, other features could easily be built on the current system. Examples of these features include:

- Implementing an algorithm that will predict the arrival time of the bus from one location to another.
- Incorporate a ticketing system where users can make payments for seats in advance.
- Adding an Interactive Voice Response (IVR) system which will allow a computer to interact with bus riders through the use of voice to provide them with transit information.

- Adding search features which will allow users to search for bus stops and buses.
- Using the system to generate information needed to implement a traffic information system.

Also, the application needs be ported to the other mobile phone platforms such as Windows Phone 8, Apple's IOS, Blackberry's RIM and the Java 2 Platform Micro Edition platform (J2ME). This will increase the reach of the application to a wider group of potential users.

Works Cited

- Anderson, R. E., Brunette, W., Johnson, E., Lustig, C., Poon, A., Puntam, C., et al. (2010). Retrieved March 13, 2012, from University of Washington, Computer Science & Engineering:
<https://www.cs.washington.edu/homes/rea/papers/StarBus-ICTD2009.pdf>
- Balsamiq. (2012). *Mock Ups Web Demo*. Retrieved March 2012, from <http://builds.balsamiq.com/b/mockups-web-demo/>
- Ferris, B., Watkins, K. E., & Borning, D. A. (2010). *OneBusAway: Behavioral and Satisfaction Changes Resulting from Providing Real-Time Arrival Information for Public Transit*. Atlanta, GA.
- Ferris, B., Watkins, K., & Borning, A. (2009). Retrieved November 18, 2011, from One Bus Away web site:
wiki.onebusaway.org/bin/download/Main/.../mobicase-2009.pdf
- Google. (2012). *Google Maps*. Retrieved 2012, from Google Maps:
<http://www.maps.google.com>
- LLC, T. T. (2011, Novmber 4). *Tiramisu* . Retrieved March 14, 2012, from Google Play Website:
<https://play.google.com/store/apps/details?id=edu.cmu.dv1.androidprod>
- Manful, K. (2009). *Location Based Services for 2D/3D Maps*. Accra: Ashesi University College.
- MySQL. (2012). *Why MySQL*. Retrieved April 1, 2012, from MySQL Website: <http://www.mysql.com/why-mysql/>
- Transit, M. (2012). *Rider Tools*. Retrieved April 3, 2012, from Metro Transit Website.

Appendix

Appendix A: Database Architecture

Diagram of Database Architecture

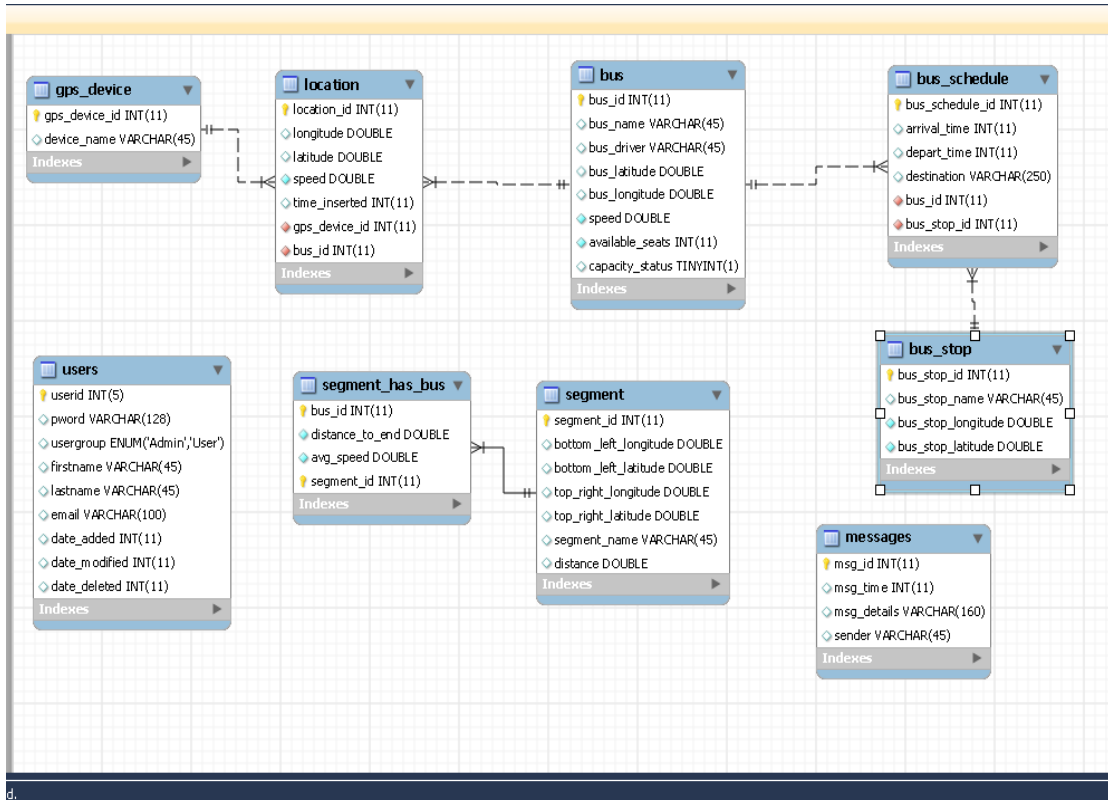


Figure 12: Database Architecture of Catch the Bus

Appendix B: Use Cases

Table of Use Cases

Table 4: Table Showing Use cases and the procedure for each use case

	Use Case	Procedure
1	GPS Logging	Application to log GPS was deployed on an Android phone and handed to Dr L for testing. The GPS coordinates were logged anytime the location of the phone changed or if it was idle for 2 minutes. A Google maps image was supposed to be automatically rendered with the location

		of the bus anytime a new location was logged to the database.
2	Driver Login, Select Bus/ Vehicle and seating	Application was deployed on two android phones. The driver drove from the Berekuso pitch up the Ashesi campus whiles another user was on the campus.
3	Send Message/ Receive Broadcast Messages	Application was deployed on two android phones. User of Android phone A posted a message then user of android phone B queried the database for the new message and vice versa.
4	Query location of Bus via SMS	Application was deployed on Android phone and user drove away in a vehicle from Berekuso to the Atomic Junction. Another user queried the location of the bus via SMS.

Observation from Use Case 1

Trial 1 – Results

- Phone got very hot on the suspicion that it was processing a lot of data in a short period of time.
- GPS coordinates logged were fairly accurate.
- The Google maps image failed to automatically render displaying the location of the vehicle on the map. For the location of the bus to be displayed on the map, the Google maps page had to be manually refreshed. The image indicating the location of the vehicle was described as too small.

Things to consider

- Reduce the rate at which GPS tracker logs the coordinates. This will reduce the amount of processing done by phone. This has been set to a default of 50 seconds.
- Review code which renders the location of the vehicle on the map.
- Increase the size of the image displaying the location of the bus.

Trial 2 – Results

- The phone got less hot compared to the first trial.
- Map was rendered with an image of the location of the bus at about every ten second interval. Internet connection was an issue which made it difficult to obtain the map with the rendered image of the bus.

Observation from Use Case 2

Trial 1- Results

- The driver was able to successfully login and entered seating details. The driver was able to update seating information while driving.
- The user on campus was able to see the changes to seating.
- The user failed to see the real time location of the bus on a map on the phone.
- The user was also able to get the vehicles that existed within his segmented region but failed to indicate when the vehicle went outside the segmented area.

Things to consider in next trial

- Focus on getting the map to be updated with the real time location of the bus.

Trial 2 – Results

- The map was rendered with the location of the bus as it moved from location to another.

Observation from Use Case 3

Trial 1 - Results

- User was able to type and send message to a database provided by webhosting service.
- User was able to view messages that were logged.
- Timestamps on messages were not accurate. This could be due to the time zone of the webhosting service.
- The focus of the textbox to enter the message was cantered which is far from ideal.

Things to consider in next trial

- Adjust time which is logged to the database of webhosting server or method of logging the time.
- Set the focus of the textbox to enter the message to the top left corner.

Observation from Use Case 4

Trial 1 - Result

- User was unable to obtain the correct segment in which the vehicle was located. On the other hand, seating information and other vehicle information were obtained via SMS.
- Receiving the SMS took a while to process after the SMS query was sent from a phone.

Things to consider in next trial

- Review code which obtains the segment in which the bus is located.