City University of Hong Kong
Department of Computer Science

BSCCS/BSCS Final Year Project Report 2008-2009

(08CS040)

Project Title:

Bus Arrival Predictor on the platform of Google Android

(Volume 1 of 1)

Student Name : LAM, Ka Ho

Student No. :

Programme Code : BSCCS

For Official Use Only

Supervisor : Mr. LAI, Chi Fai Donny

1st Reader : Dr. WONG, Shek Duncan

2nd Reader : Dr. YU, Yuen Tak

Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

Bus Arrival Predictor on the platform of Google Android

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name:	**Lam Ka Ho**	**Signature:**	
Student ID:		**Date:**	**20th April, 2009**

Extended Abstract

Buses transportation system is one of the busiest systems in Hong Kong, according to Hong Kong Yearbook 2007, franchised buses takes up about 34% of the total daily transport volume, which carries about 4 million passengers per day. Although, it provides service with efficiency to citizens in Hong Kong, people still come across the time when they miss a bus and spend a long time queuing at the bus stops.

Real time passenger information systems have been adopted for a decade in most parts of the Western Europe and Japan for providing bus arrival information. However, actual implementation schedule of those real time information systems in Hong Kong is still remain an uncertainty. Nowadays, a substantial number of smart phones have the multimedia ability and geo-locating ability. Some of the mobile phone users would use 2G technologies to get access to the Internet while they encounter weak connectivity and poor browsing experience. Furthermore, they lack the positioning device, the Global Positioning System (GPS).

This project is aiming to implement a Bus Arrival Predictor on the platform of Google Android™ that can provide relevant bus route information with bus arrival time to users. The project explored the great flexibility and capability of the first Google Android™ powered mobile phone, HTC G1™. Moreover, different development platforms and prevalent bus arrival prediction algorithms are briefly discussed, including Kalman Filter, Automatic Vehicle Location (AVL) and Automatic Passenger Counters (APC).

In completing the project, different technologies like AGPS, location detection, route finding and estimated time of arrival will be examined. What's more, inherited shortage of the evolving Google Android™, such as the lack of walking directions, will be addressed and solution suggested. The ultimate goal of this project is to provide an inspiration of further development on future mobile real time passenger information system, and suggest ways to overcome current limitations.

Acknowledgements

I would like to take this opportunity to acknowledge and extend my heartfelt gratitude to my project supervisor **Mr. Donny LAI** for his yearlong support and guidance during the project development. He has given me many stimulating suggestions and his valuable comments have helped me to solve many of the problems in the project.

I would also like to express my appreciation of **Dr. Becky KWAN** for her guidance on the research for and writing the Interim and Final Reports.

I am thankful to all of my classmates who have given me comments and supports throughout the project.

I must also express my deepest gratitude to my supportive family, my mother Winnie, my father Edmond and my brother Gary for their patience to let me complete this work. I also thank Aman for her imaginative ideas on the graphics used in the project.

Thank you all.

Contents

Table of Figures

1 Introduction

Hong Kong is an international city with sophisticated transportation network. It provides great effectiveness to citizens in Hong Kong. Buses transportation system is one of the busiest systems in Hong Kong, according to Hong Kong Yearbook 2007, licensed buses takes up about *34%* of the total daily transport volume, which carries about 4 million passengers per day (Hong Kong Government, 2007). People living in this fast changing world would likely to spend lesser time to complete more jobs. Therefore, people in Hong Kong will not want to spend extra times on waiting buses and finding buses. As many people may come across the time when they miss a bus and spend a long time queuing at the bus stops. Sometimes people have missed a first choice route and when they kept waiting on the first choice, the second choice arrived at a bus stop far away across a few streets; they would miss the second choice also.

If we have a mobile device that can provide updated bus route information with bus arrival time based on your current location and suggest bus route alternatives to the same destination, it will definitely help us to manage our time efficiently and we can better time our plan. People can decide whether they have to keep waiting the first choice or go across a few streets to wait for the second choice instead. In addition, people can determine whether they have to run or walk to the bus stops when they are near to the potential bus stops.

1.1 Existing Situation

According to the figures from Census and Statistics Department as shown in **Figure 1.1a** (Hong Kong Economic and Trade Office (Canada), 2008), there are *1521* mobile phone subscribers per 1000 population, which means each person owns about 1.5 mobile phones. However, not all of the mobile phones would have the ability of internet accessing. Some of the mobile phone users would use 2G technologies to get access to the Internet while they encounter weak connectivity and poor browsing experience. Furthermore, they lack the positioning device, the Global Positioning System (GPS), which helps tracing the users' current location. As a result, they can only access web pages that provide digital maps but not to locate themselves on the map accordingly. Hence, up to date bus route information cannot be provided.

Hong Kong in Figures (3)				
	2004	2005	2006	2007
Transport, Communications and Tourism				
Port container throughput ('000 TEUs)	21 984	22 602	23 539	**23 998**
Air cargo throughput ('000 tonnes)	3 090	3 402	3 580	**3 742**
Mobile phone subscribers per 1 000 population [7]	662 [1180]	682 [1226]	747 [1 369]	**818 [1 521]**
Visitor arrivals ('000)	21 811	23 359	25 251	**28 169**

Figure 1.1a: Fast Facts about Hong Kong

1.2 Project Objective

The purpose of my project was to implement a Bus Arrival Predictor on the platform of Google Android™ that can provide relevant bus routes information with bus arrival time to users in order to explore the possibility and capability of the Google Android™. The system was run on the HTC's first Android™ based mobile phone, G1™, which has 3G technologies enabled, a built-in GPS receiver and Google Maps™ View display. Users' current position is collected and together with the up to date bus schedules, a bus arrival time is calculated on the server and an alternate bus route was given as a second suggestion. In addition, based on the built-in G-sensor compass mode provided by the G1™, the system can also derive the path to the nearest potential bus stops and the distance is determined.

1.3 Project Scope

The aim of this project is to develop an application on a mobile device running the Google Android™ platform and which it provides relevant bus route information with bus arrival time prediction. The calculation of the prediction is done on the server side and then the processed data are retrieved and presented on the users' mobile device with the help of G1™'s built-in Google Maps™ View display. This application includes the following general features:

- a GPS based system that can determine the distance of the users and the bus stops
- a timing device can tell when the bus should arrive to that bus stop
- a map interface can that shows the potential bus stops
- a bus route adviser can give choices to users
- a bus travelling timer
- a get off notification
- a real time map marker

The detailed descriptions about each feature are as follows:

1.3.1 Determination of the distance between users and bus stops

With the integrated GPS receiver in G1™, users' current location can be identified appropriately. Collected data are then picked up and sent to the server, bus stops locations will be retrieved from the database to the mobile phone. Based on the G-sensor compass mode, distance to the bus stops can be measured.

1.3.2 Prediction of the bus arrival time

Details about the users' current status, such as location, time, are gathered and sent to the server for analysis. Arrival Prediction algorithms are applied and arrival times are then derived from the bus route schedules and the simulation set of data.

1.3.3 Google Maps™ Interface

A Google Maps™ based view will be displayed on the interface. It is the core of this application and basic map control functions are provided, such as zooming, panning and mode changing. Moreover, bus stops details are provided to users on the map.

1.3.4 Bus Route Advisor

Besides the major bus route information are provided, a second preference will be provided as an alternative. Users can decide whether they should take the second option in advance, which is before the next bus has arrived. Also, direction to the potential bus stops will be displayed as a guided route.

1.3.5 Bus Travelling Timer

A timing function that can record the time taken for a particular trip of the user, which the data, such as actual bus arrival time, time taken between stops, can be viewed as a reference for future trips.

1.3.6 Get Off Notification

A get off notification is integrated with the timer, as users can set their get off destination when they get on the bus so that a notification (a ringing sound with vibration) is made to tell the users to get off the bus when they are near, a distance set by users, to their destination.

1.3.7 Real Time Map Marker

Map markers function is that the user can utilize the integrated camera of G1™, so that users can take photos when they are on their trips. Images taken are saved to the SD card and marked on the map according to positions where they were taken. It serves as a memory store to save all you precious moments.

1.4 Significance of the project

With the Android™–powered mobile phone, this application can be utilized more thoroughly.

1.4.1 User-friendly

Google Android™ is a Linux-based mobile phone operating system. It supports finger-touch input method. Users can perform multiple tasks or actions in just a few flips of finger. Every action on the interface can be performed intuitively without costing extra learning time.

1.4.2 Responsive

Since G1™ supports 3G technologies, its internet connectivity can be enhanced when compared to the previous generations. Processed data can be transferred between clients and server with at least 144kbps (kilobits per second) while only 56kbps in the past.

1.4.3 Adaptable

Since Google Android^{TM} provides an applications distribution platform, the Android Market^{TM}, for developers, thus whenever there is an update of the application version, users will be informed immediately and the most updated version is guaranteed to be delivered.

1.4.4 Reliable

Since the calculation of the prediction algorithms will be performed remotely and all related route databases are stored on remote servers, different users will share the same set of route data so as to maintain the conciseness of the dataset and provide an up to date bus route schedule and information to them.

1.4.5 Accurate

With the help of 3G coverage, G1^{TM} can locate the users more accurately with its built-in GPS receiver. Instead of getting positioning signals from satellites directly, G1^{TM} adopts Assisted-GPS that locates a person with the help of telephony signals from traditional telecommunication base stations, which enhances the accuracy to calculate distance between the bus stops.

1.4.6 Accessible

This application is ideal to be run on the G1^{TM} as location determination can be done through 3G coverage and GPS technologies, while with G1^{TM}'s own embedded Google Maps^{TM} View, GPS receiver and the existing 3G foundation in most of the area in Hong Kong, this application will reach a high availability.

2 Literature Review

2.1 Related Applications

2.1.1 In Western Europe

Real time passenger information systems have been adopted for a decade in most parts of the Western Europe and Japan. The systems are seen in many different forms, ranges from traditional static displays to those wireless intelligent information systems providing arrival time which to the nearest minutes or even seconds. **Figure 2.1.1a** (Kole, 2007) shows a regular system architecture for a real time information system.

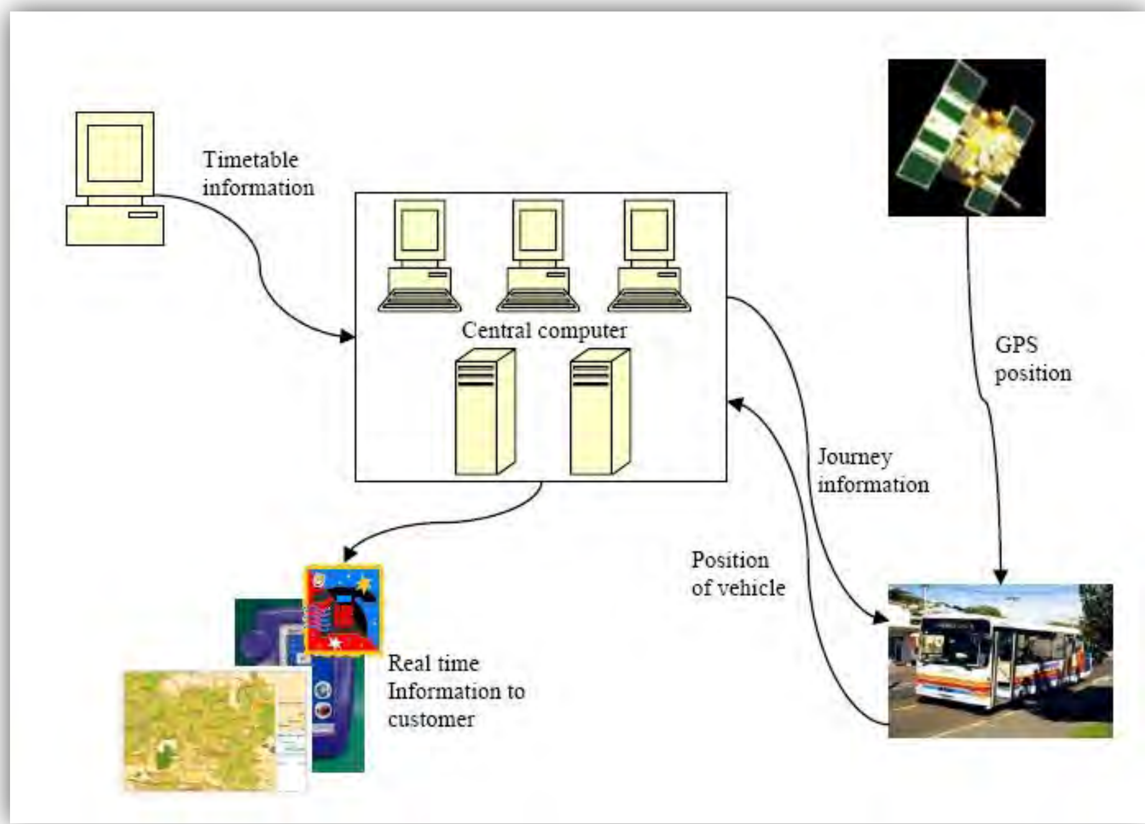


Figure 2.1.1a: Structure of Real Time Information System in Wellington

Dynamic Message Sign (DMS) is the most preferred way to present the transit information, they are commonly known as electronic sign (Transportation Research Board, 2003). **Figure 2.1.1b and 2.1.1c** (Transportation Research Board, 2003) show two LED signs that is used in Portland and London respectively. To increase the mobility of the service, sending SMS to passengers about the route information on demand is a good choice. **Figure 2.1.1d** (Collins et al., 2007) shows a system called "txt bus" utilizing this function. Besides SMS systems can provide mobility, there are systems, which are web-based, provide real time information about the bus route and the bus location on a web site, for those who have a mobile device with internet accessibility, they can visit the web sites and view the current route information. **Figure 2.1.1e to 2.1.1g** (Transportation Research Board, 2003) show some screenshots of those web sites.

Figure 2.1.1b: Portland, Oregon, Tri-Met Transit tracker sign

Figure 2.1.1c: London Buses Countdown sign

Figure 2.1.1d: An instructional sign to be posted at bus stop

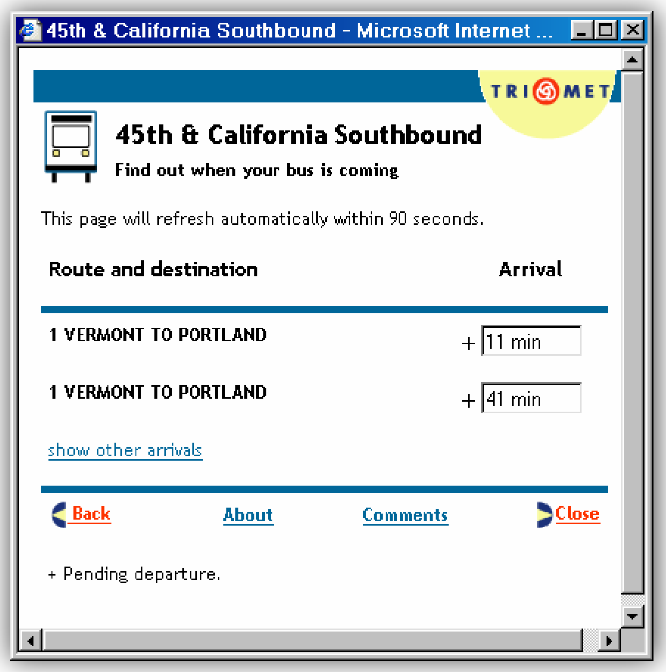


Figure 2.1.1e: Tri-Met Transit Tracker on the Internet

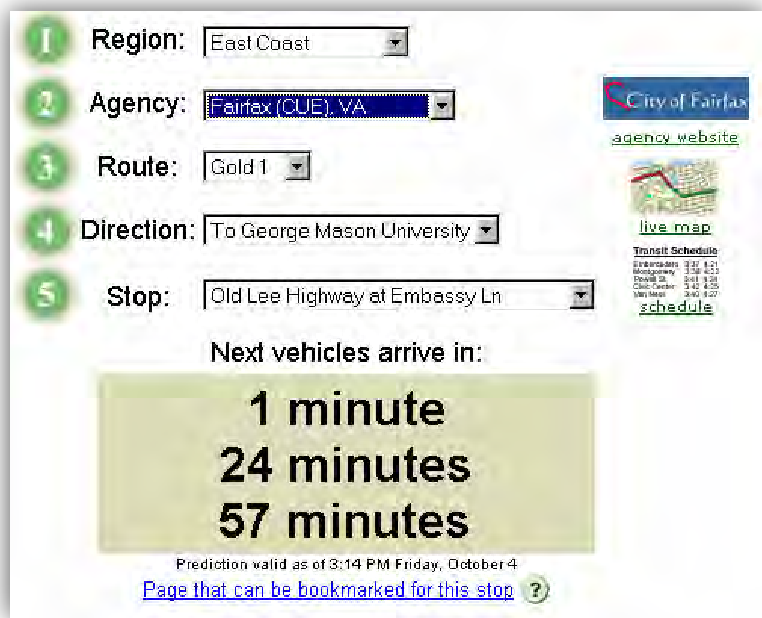
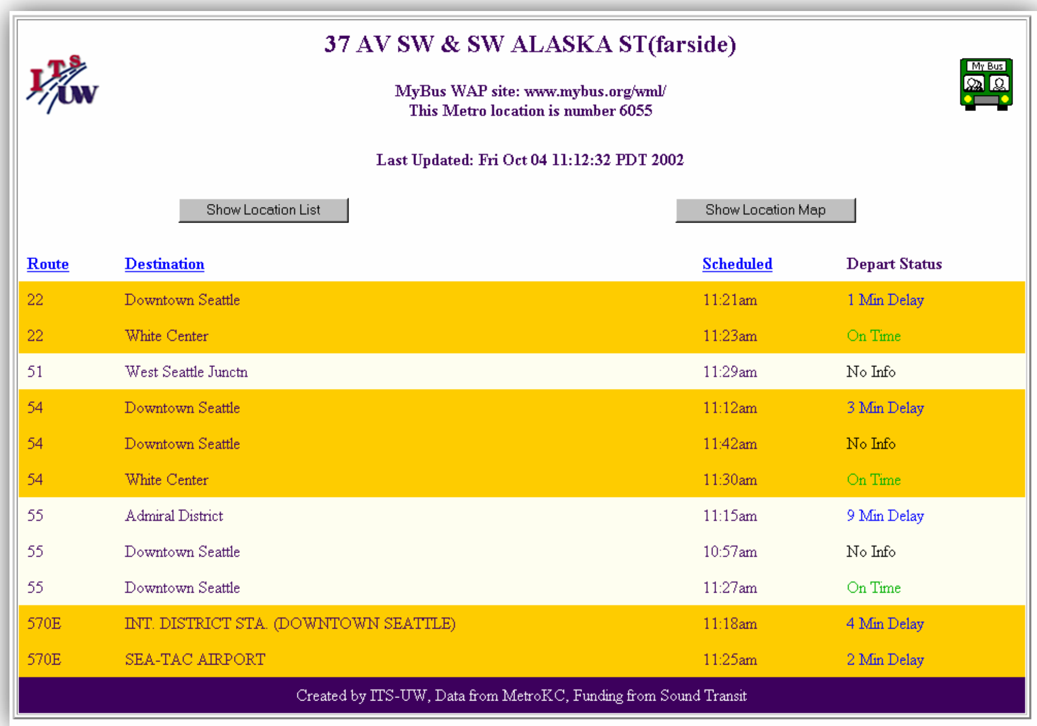


Figure 2.1.1f: Fairfax CUE (City–University–Energysaver) real-time information



Route	Destination	Scheduled	Depart Status
22	Downtown Seattle	11:21am	1 Min Delay
22	White Center	11:23am	On Time
51	West Seattle Junctn	11:29am	No Info
54	Downtown Seattle	11:12am	3 Min Delay
54	Downtown Seattle	11:42am	No Info
54	White Center	11:30am	On Time
55	Admiral District	11:15am	9 Min Delay
55	Downtown Seattle	10:57am	No Info
55	Downtown Seattle	11:27am	On Time
570E	INT. DISTRICT STA. (DOWNTOWN SEATTLE)	11:18am	4 Min Delay
570E	SEA-TAC AIRPORT	11:25am	2 Min Delay

Created by ITS-UW, Data from MetroKC, Funding from Sound Transit

Figure 2.1.1g: King County Metro (Washington State) MyBus information

Another method to provide real time information is to combine graphics and digital maps in order to indicate the exact current location of the vehicle on the track. **Figure 2.1.1h** (Transportation Research Board, 2003) displays a system showing the current location of the bus within the service area. It is much more intuitive but lacks the prediction time when compared to previous ones.

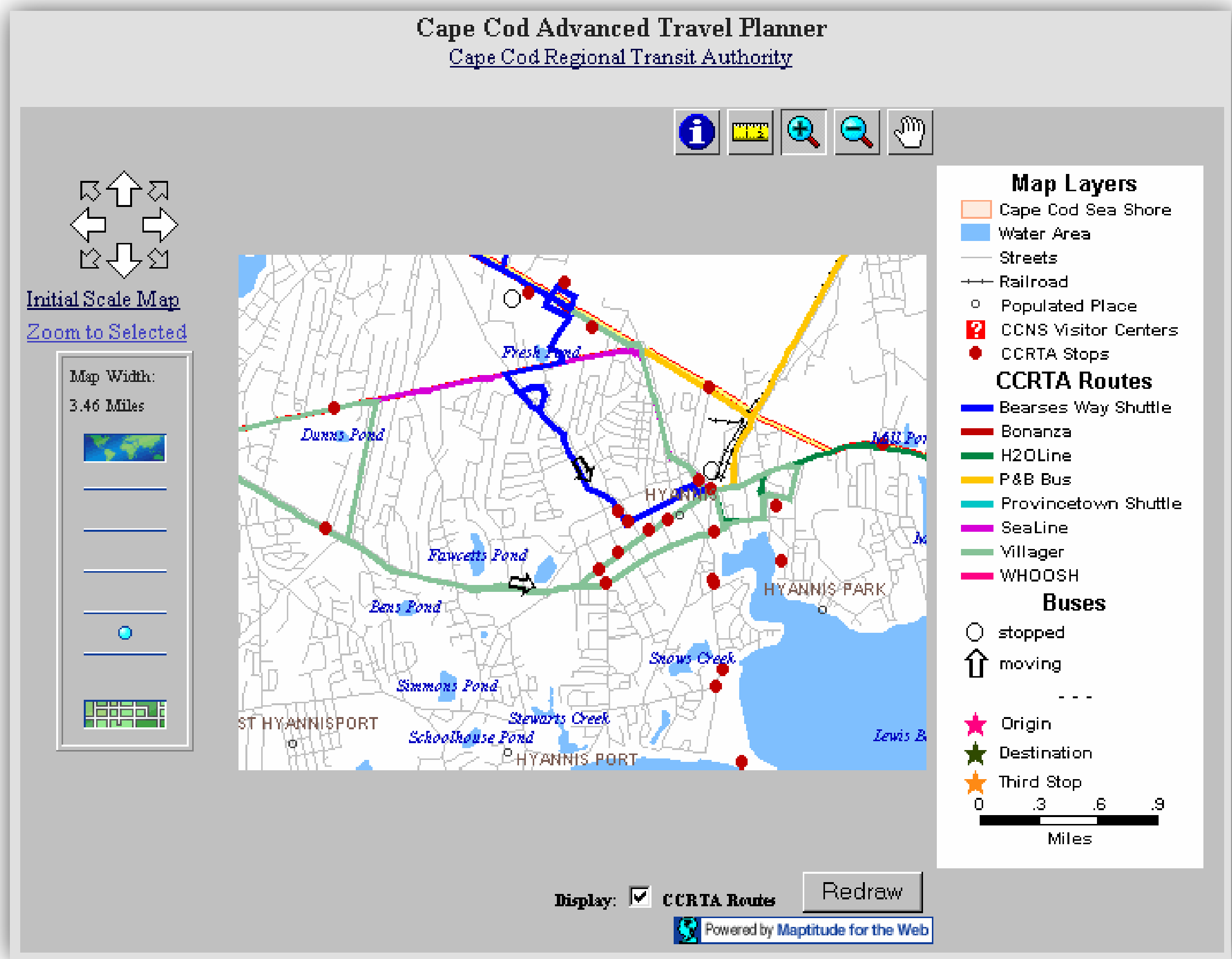


Figure 2.1.1h: Cape Cod Regional Transit Authority (CCRTA) real-time bus location map

2.1.2 In Hong Kong

Unlike the Western Europe, utilizing the bus arrival prediction system is not prevalent in Hong Kong. Recently, the Kowloon Motor Bus Co. Ltd. and the Citybus Ltd. have also developed their own bus arrival time prediction system. Kowloon Motor Bus Co. Ltd. was the first bus company in Hong Kong to provide a Bus Stop Announcement System on their buses, so that speakers on the bus would tell the information of the upcoming bus stops to passengers. The system was controlled by the driver only and is not relying on GPS receivers, therefore, if the bus captain forgets to press the button, no stops information would be announced. **Figure 2.1.2a** shows a LED display of the Bus Stop Announcement System on a bus.

Kowloon Motor Bus Co. Ltd. has also adopted the Electronic Terminus Management System (Kowloon Motor Bus Co. Limited., 2009) with full network coverage recently. When bus captains have arrived the terminus, he/she can just swipe their smart cards on a card reader, so that their arrival time is recorded, departure time is calculated and displayed to passengers through their LED display equipped at the terminus. **Figure 2.1.2b** shows the screen of the Electronic Terminus Management System terminal while **Figure 2.1.2c** shows the LED screen displaying buses departure time.

Citybus Ltd. has started to provide the prediction service trial since 1st August, 2007 only on the Route A10, an airport express route. It has a similar mechanism as with the "txt bus", passengers can get real time bus arrival information through SMS messages. Real time information of the bus can be provided is because each of the Route A10 bus has equipped with a GPS receiver, so that location and timing information can be delivered to passengers through control terminal. **Figure 2.1.2d** (Citybus Co. Limited., 2007) shows the procedures to acquire the prediction time information.

Besides arrival prediction system developed by bus enterprises, there are also a few small scale mobile software providing bus routes information querying without the arrival time prediction and the integration of the GPS location service for potential bus stops. **Figure 2.1.2e** (Studio KUMA BusInfo software, 2008) shows a mobile route searching application developed on Windows Mobile platform, called the "Studio KUMA BusInfo software". **Figure 2.1.2f** (MiniBus, 2009) shows another route searching application run on the iPhone platform.

Figure 2.1.2a: LED display of Bus Stop Announcement System

Figure 2.1.2b: Screen of the Electronic Terminus Management System

Figure 2.1.2c: LED screen displaying bus departure times

Figure 2.1.2d: Procedures to acquire bus arrival time information

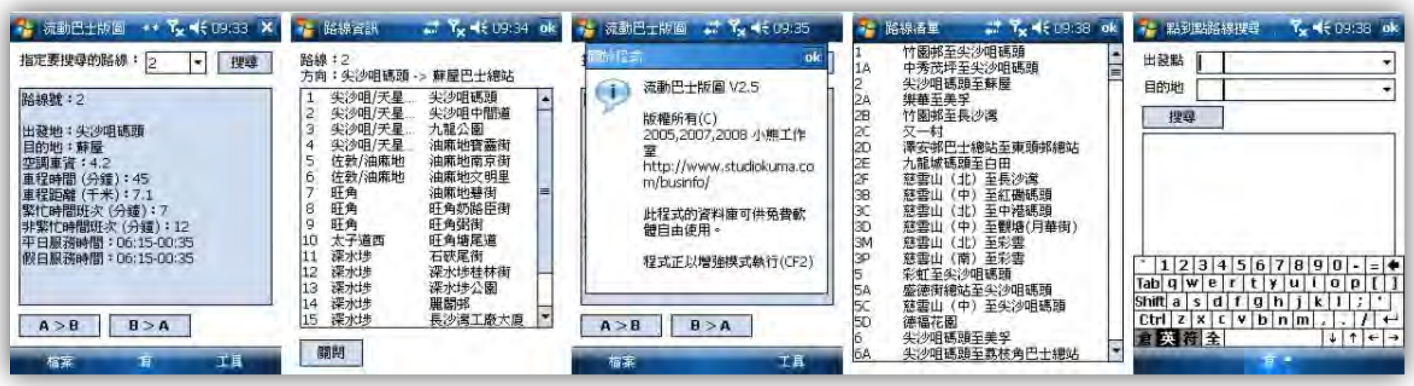


Figure 2.1.2e: Screenshots of the "Studio KUMA BusInfo software"

Figure 2.1.2f: Screenshot of the "MiniBus"

2.2 Related Research

Hong Kong has its own well-developed transportation network. Unlike the existing railway system, passengers have to make decision on the choice of bus route based on the specific bus stops (Chau, 2002). Due to this constriction, choosing buses as the mode of transportation has its own advantages. There are many researches on the prediction model of arrival time and travel time in the field. According to Shalaby et al (2001), various researchers have made different progression on this topic, such as developed four algorithms to predict the bus arrival time with the provision of the vehicles' current location and time (Lin and Zeng), modeled the vehicles arrival times based on historical statistical data and Automatic Vehicle Location (AVL) Data (Wall and Dailey. 1999) and invented methodology to evaluate the real time information system on three real life cases (Mishalani, Lee and McCord, 2000). Maclean et al. (2001) and Chau mentioned that AVL is the most commonly used technology in the prediction model while some of them used Kalman Filtering Algorithms together to forecast bus arrival time when there is insufficient physical location data provided.

Several researchers have proofed that the implementation of real time bus information does contribute to the increase of patronage of bus companies. Actually, these systems became popular among Western European countries and Japan a decade ago. Here are some examples that show the use of real time passenger information (Chau, 2002):

- Metro Rapid in Los Angeles, USA
- The SuperRoute 66 in Ipswitch, UK
- SMART Route in Liverpool, UK
- Gothic in City of Gothenburg, Sweden

Surveys have also been conducted on the preference and effect of having real time information system. Kole (2007) stated that there is a 22% patronage increase in Christchurch while about 50% among the increase are contributed to the real time information system and a 30% increase in Auckland Airbus companies within a year with an 80% satisfaction rate. The surveys also unveiled that the integration of the system has other both tangible and intangible benefits. Passengers do have lesser complaints on the schedules and drivers (Kole, 2007). From the passengers' point of view, delays are reduced and arrival time can be timed accurately (Dessouky et al., 1999). Chau added from a survey carried out in Belgium (1999-2000) that the waiting is more acceptable, time seems going more quickly when the arrival time is a certain, also, passengers observe a more reliable bus service as accurate information are provided so that waiting bus at night can be much safer. Moreover, they may have the possibility to carry out some last minute shopping. Seat availability, comfort, promotion of bus companies' image and the certainty of bus arrival time are those intangible benefits for sure (Chau, 2002).

In Hong Kong, real time information systems are used intensively in railway systems, however, there were limited similar projects carried out by bus companies. New World First Bus Services Ltd. has conducted a trial on GPS in 1999, the system has collected data from three satellites and sent them to the central terminal for analyzing the location of the buses. Nevertheless, due to the physical environment of Hong Kong, tall buildings are closely scattered, satellites signals are adversely blocked, and to solve the problem, New World First Bus Ltd. has introduced the installation of odometer and electronic gyro on buses when weak signals are detected. Citybus Ltd. has suspended its research since a pilot run and started another trial run on the SMS Bus Arrival Prediction Trial system solely on a particular bus route lately. The Kowloon Motor Bus Co. Ltd. has run a preliminary test and trial on buses with GPS several years ago (Chau, 2002) and has developed the Electronic Terminus Management System for schedules and terminus management but not for real time arrival predictions. Although, the three main franchised bus companies has conducted related researches and tests, the actual implementation schedule of those real time information systems in Hong Kong is still remain an uncertainty.

2.3 Background Studies

2.3.1 Location Technologies

Nowadays, a substantial number of smart phones have the multimedia ability and geo-locating ability. While some people may get confused with GPS and AGPS, here provides a brief background study about them.

2.3.1.1 Global Positioning System (GPS)

Global Positioning System is composed of satellites and GPS receivers. GPS receivers receive signals from the satellites orbiting in space in 6 different planes 20 kilometers away from Earth (Porcino, 2001). There are 24 satellites orbiting in space at present originally owned by United States government for military purposes and are now opened for commercial use. **Figure 2.3.1.1a** (Jeong, 2004) illustrates the satellites orbiting in space. The GPS receiver installed in the mobile handsets will receive radio signals from satellites and compare with the local duplication of geo data to calculate its actual location on Earth. To increase the accuracy, data received from three satellites can perform the calculation of two-dimensional location, including the longitude and latitude. For three-dimensional location information, consisting longitude, latitude and altitude, data from at least 4 satellites are required (Chadil et al., 2008).

Figure 2.3.1.1a: Twenty-Four Satellites of GPS

2.3.1.2 Assisted Global Positioning System (AGPS)

AGPS is sometimes known as Aided Global Positioning System. As it stated, the positioning mechanism behind it requires assistance data together with the satellites. **Figure 2.3.1.2a** (Karunanayake et al., 2007) illustrates an AGPS system in brief. As mentioned by Karunanayake et al., assistance data helps to increase sensitivity, so that allowing the device to function well in non-line-of-sight (NLOS) signal environment, such as urban canyons and indoors. Traditional GPS receivers are designed to function under line-of-sight (LOS) environment, therefore when there are inadequate signals detected, AGPS can get help from assistance data. However, when there is no assistance data from network, normally data are came from cellular network nowadays, AGPS architecture allows GPS receivers to work on solely so as to increase system flexibility. Most of the recent mobile handsets comprising location detection employ the AGPS approach as it has greater adaptability, including the HTC G1™, which is chosen in this project.

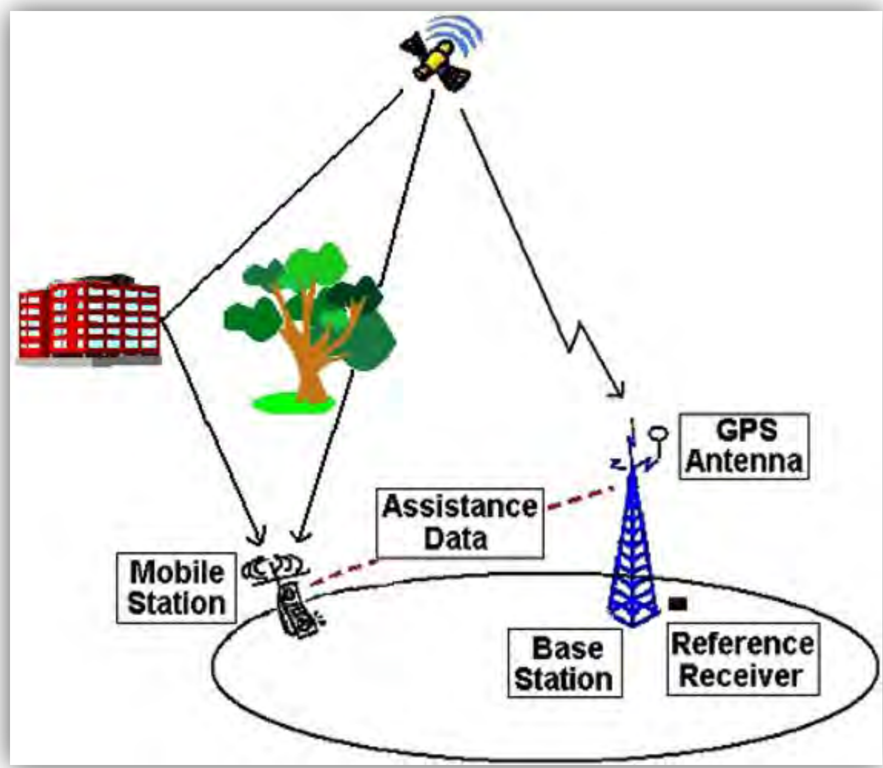


Figure 2.3.1.2a: An AGPS system

2.3.2 Development Platforms

In this section, different development platforms, both software and hardware parts, are briefly discussed. Google Android™ is the core development environment in this project while Google Maps™ contributes to most of the functions relating to geographic location. As we did not have a real physical phone, the HTC G1™, as will be available only after April in 2009, all of our program development was done and tested within a G1™ Emulator, which was installed to Eclipse, a program development environment.

2.3.2.1 Google Android™ and Eclipse plug-in

Android™ is also a stack of software comprising an operating system, middleware and key applications for portable devices (Google, 2008). **Figure 2.3.2.1a** (Google, 2008) presents the architecture of an Android™ operating system. Linux kernel, version 2.6, is the bottom layer of the architecture and it provides services like memory and process management. It also forms the abstract layer between the hardware and software stack. Moving upwards are the libraries, Android™ provides C/C++ libraries, such as SQLite, for developers in order to help the development of various system components. Application framework is to layer to provide system flexibility and reusability as components in this layer are designed to be reused so that different applications can share the same set of core capabilities. On the topmost layer lays the application layer where different factory made and users' functions are present, including Phone, Browser, Contacts, etc. (Google, 2008).

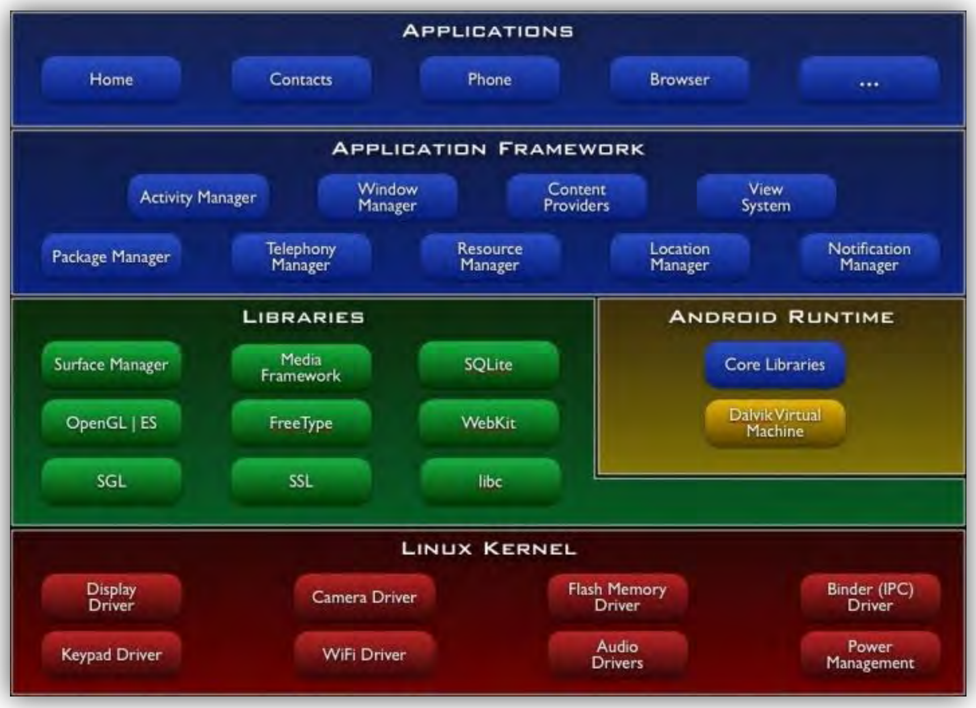


Figure 2.3.2.1a: Major components of Android™ operating system

As HTC G1^{TM} is not available in Hong Kong until the first quarter in 2009, most of our program development relies on the G1^{TMv} Emulator installed in Eclipse as a plug-in called Android Development Tools (ADT). The integrated plug-in helps developers in various way, as it provides powerful debugging tools to speed up debugging and revising, so that development of Android^{TM} applications can be faster and easier (Google, 2008).

2.3.2.2 Google Maps^{TM}

Google has offered many types of services; one of them is Google Maps^{TM}. Google Maps^{TM} is a powerful service; it offers user-friendly interface with Google's leading mapping technology and provides local business information, such as addresses, contacts and driving directions, on the map. Android^{TM} has adopted a mobile version of Google Maps^{TM} in it, developers can integrate most of the functions provided as of the original version has to their applications. Users can view the map in satellite, traffic, hybrid and street mode. Maps are draggable and zoomable with a few finger taps on the touch screen in G1^{TM} (Google, 2008). Google Maps^{TM} experiences in Android^{TM} are expected to be the same as in the original web-based version.

2.3.2.3 HTC G1^{TM}

HTC G1^{TM} is the first smart phone that is Android^{TM} powered. Android^{TM} will be the operating system of the phone. It has Google Gmail^{TM}, YouTube^{TM}, Google Talk^{TM}, Google Calendar^{TM} and Google Maps^{TM} preinstalled. Those powerful services make G1^{TM} to be an all-in-one communication tool so that let users to keep track on things surround them (HTC, 2008). This project is going to be implemented with Android^{TM}, however, the physical phone was not available until late April. Therefore testing and running could not be carried out on this G1^{TM}. It was undesirable as emulator did not simulate all conditions when a physical phone interact with the real life environment, such as testing location detection accuracy on an actual street. With its 3G and AGPS capabilities, our application should perform satisfactorily. **Figure 2.3.2.3a** (HTC, 2008) shows a screenshot of the real G1^{TM}.

Figure 2.3.2.3a: Front and side view of HTC G1™

2.3.3 Bus Arrival Prediction Algorithms

There are several commonly used bus arrival prediction algorithms in the market. Although this project is not focusing on the evaluation of each algorithm, a brief investigation of some of the famous ones can be imparted.

2.3.3.1 Kalman Filter

As described by Welch and Bishop (2007):

> Kalman filter is a set of mathematical equations that provide an efficient computational (recursive) solution of the least-square method. The filter is very powerful in several aspects: it supports estimation of past, present, and even future states, and it can do so even when the precise nature of the method system is unknown. (Welch and Bishop, 2007, p.1)

The Kalman filter woks in a form of feedback control to estimate a process state. **Figure 2.3.3.1a** (Welch and Bishop, 2007) shows a lifecycle of a Kalman filter. Time update process is responsible for predicting forwards while the measurement update process is responsible for giving a feedback, which is to make correction and produce a new measurement according to the previous prediction. Jeong (2004) has also stated that Kalman filter has the ability to adapt situation with traffic fluctuation over time.

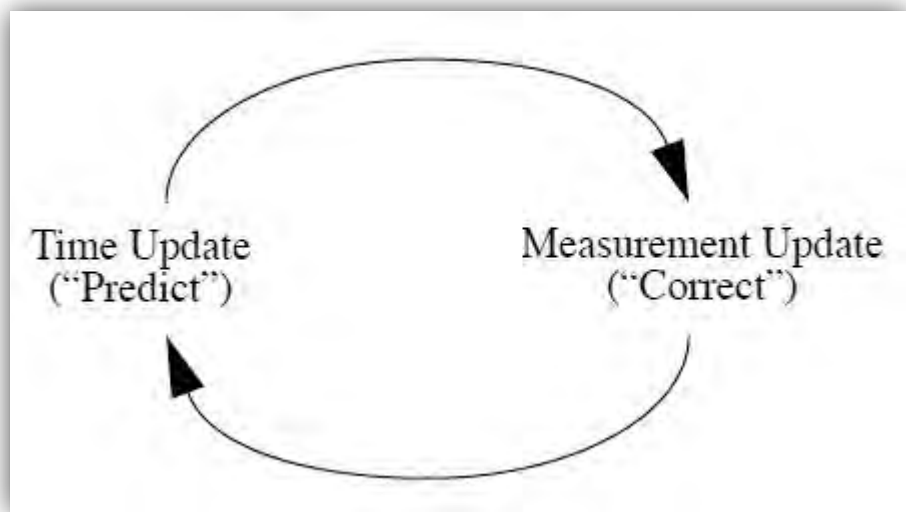


Figure 2.3.3.1a: The ongoing discrete Kalman filter cycle.
The time update projects the current state estimate ahead in time.
The measurement update adjusts the projected estimate by
an actual measurement at that time.

2.3.3.2 Automatic Vehicles Location (AVL)

Automatic Vehicle Location is the essence of most of the Intelligent Transportation Systems nowadays. It forms an important part of the transits arrival prediction. To buses that have installed AVL devices, it provides real time information of that specific bus on the track, including location, speed and direction. As mentioned by Dessouky et al. (1999), data collection for further analysis can be done with the following technologies, (a) Signpost and Odometer (SO), (b) Global Positioning System and (c) Radio Navigation. As discussed before, Global Positioning System is the choice that most commonly used (Chau, 2002). Information collected and analyzed is treated as historical data for future use, such as system planning, bus scheduling and running times prediction. However, AVL does not guarantee a 100% data coverage; communication faults cause problematic missing data (McLeod, 2007). When there is insufficient traffic data, historical data are retrieved and assumption data are made to fill in the missing ones. Besides vehicle locations, there are types of data, which can affect the prediction of arrival, such as current traffic conditions and historical traffic conditions, current bus operating data and historical bus operating data (Transportation Research Board, 2003).

2.3.3.3 Automatic Passenger Counters (APC)

Automatic Passenger Counters account for an increase of quantity and quality of real time information about the bus operation (Chen et al., 2004) and they usually work with AVL system to enhance accuracy. The information collected will consist of a newer version of vehicle location, speeds, travel time, occupancies, etc. Information collection is an automated process and done when passengers boarding and alighting at different time and locations. Two common technologies are used for data compilation, (a) infrared beam and (b) treadle mats. Infrared beam are placed along the passengers' path, interruption of the beam activate the APC when passengers aboard or alight. Similarly, stepping on the treadle mats placed at doorways induces pressure and activates the APC (Dessouky et al., 1999). Among the two of them, infrared beam is the more popular approach. APC is favorable to arrival prediction as it reduces data collection, time for data processing and enhances operation efficiency. Yet, APC and AVL have the same shortage, every tracked vehicles have to installed the extra APC and AVL devices and replacement rate is high when the vehicles is renewed and it is costly. Furthermore, APC only records data when the bus has stopped at scheduled bus stops, therefore, if skipping bus stops happens, missing data will occur.

3 System Requirement and Design

In this project, both the server side and the client Android^{TM} side were considered. Calculation of the bus arrival prediction was done on the server side programs to reduce the load on CPU utilization of the handset. Location and desired route information were sent to server for computation; data with bus arrival prediction result was then be retrieved and illustrated by the application interface.

Tourists and generic public are the target users. The system can attract citizens who choose buses not to be choice of public transportation mode. The system can also encourage last minute shopping, which will be welcomed by shopkeepers.

3.1 Functional Requirements

No.	Descriptions	Urgency	Category
RQ1	User can get current location.	High	Core
RQ2	User can perform zooming on the map.	High	Core
RQ3	User can choose map modes.	Medium	Core
RQ4	User can search bus routes.	High	Core
RQ5	User can get the path to the bus stops.	High	Core
RQ6	User can get the bus arrival prediction.	High	Core
RQ7	User can record the travelling time.	High	Core
RQ8	User can snap pictures when travelling.	Medium	Core

3.2 Use Case Diagram

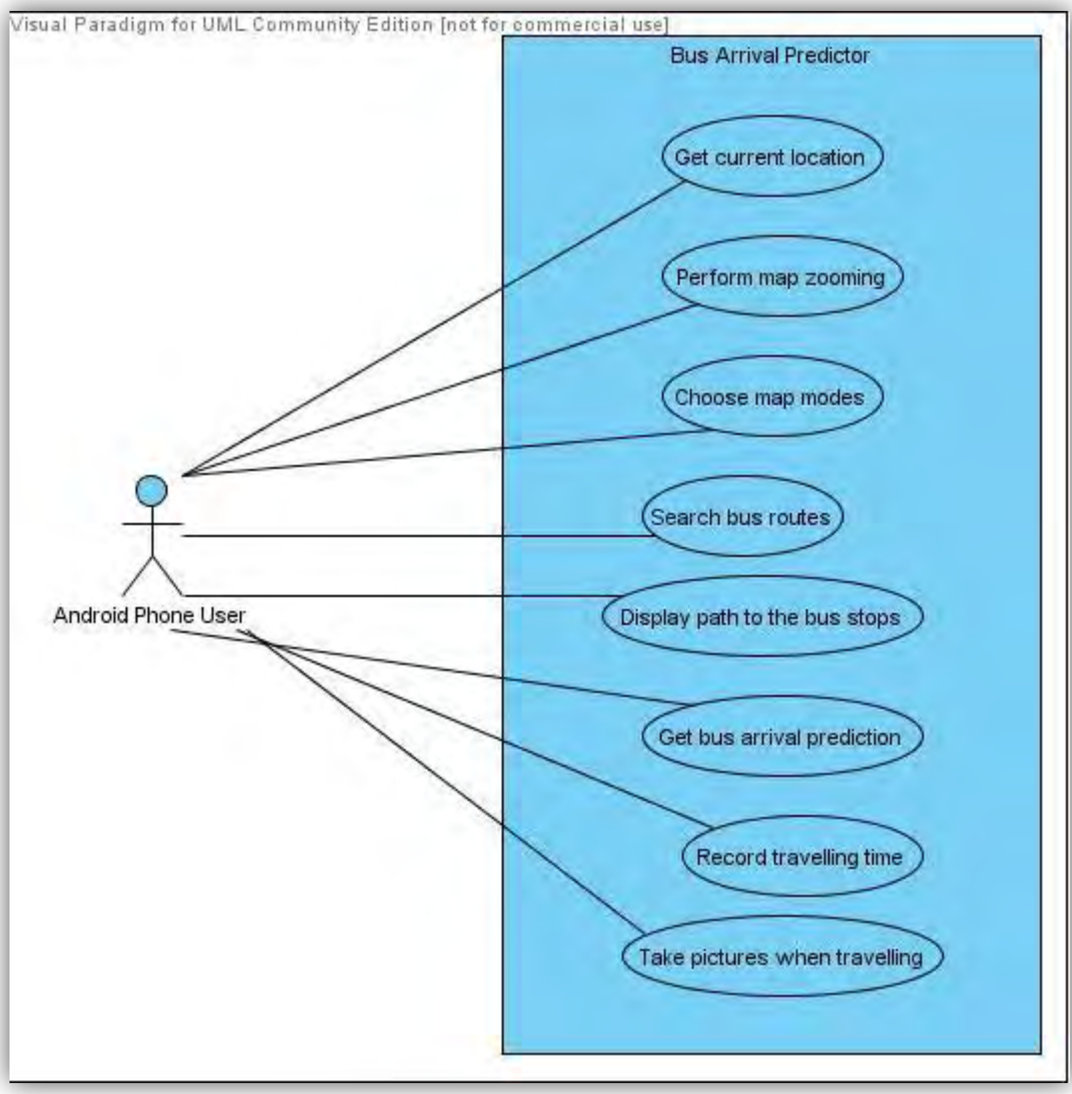


Figure 3.2a: Bus arrival predictor Use Case Diagram

Figure 3.2a shows the use case diagram of the bus arrival predictor, it illustrates the overall interactions between the user and the system. The detailed specifications of each of the use case are clarified as follows.

3.3 Use Case Specifications

Use Case ID:	UC01	
Use Case Name:	Get current location	
Actor(s):	Android™ phone user	
Description:	This user case describes the process of locating the user using AGPS. On completion, the system displays the position of the user by an indicator on the map.	
Reference ID:	RQ01	
Precondition:	The user has to turn on positioning device.	
Trigger:	none	
Typical Course of Event:	**Actor Action(s)**	**System Response(s)**
	Step 1: This use case is initiated when the user started this application. Step 4: This use case is concluded when the users see their current location on the map.	Step 2: The system call the Location Manager for location update. Step 3: The system gets AGPS data from network.
Alternative Courses:	none	
Post-condition or Results:	The located position is stored as last fix for next positioning.	
Implementation Constraints and Specifications:	none	

Use Case ID:	UC02	
Use Case Name:	Perform map zooming	
Actor(s):	Android™ phone user	
Description:	This user case describes the process when the user zoom in or out on the map. On completion, the system displays the map according to the zooming level.	
Reference ID:	RQ02	
Precondition:	The bus stops viewer tab is selected.	
Trigger:	none	
Typical Course of Event:	**Actor Action(s)**	**System Response(s)**
	Step 1: This use case is initiated when the user opens the zooming panel. Step 4: This use case is concluded when the users see the map reloaded with new map data.	Step 2: The system call the Map View Controller to perform zomming. Step 3: The system gets more map data of the zoomed map from the Google Maps™ data provider.
Alternative Courses:	none	
Post-condition or Results:	none	
Implementation Constraints and Specifications:	none	

<table>
<tr><td>Use Case ID:</td><td colspan="2">UC03</td></tr>
<tr><td>Use Case Name:</td><td colspan="2">Choose map modes</td></tr>
<tr><td>Actor(s):</td><td colspan="2">Android™ phone user</td></tr>
<tr><td>Description:</td><td colspan="2">This user case describes the process when the user chooses different map modes of the map. On completion, the system displays the map according to different map modes.</td></tr>
<tr><td>Reference ID:</td><td colspan="2">RQ03</td></tr>
<tr><td>Precondition:</td><td colspan="2">The bus stops viewer tab is selected.</td></tr>
<tr><td>Trigger:</td><td colspan="2">none</td></tr>
<tr><td>Typical Course of Event:</td><td>Actor Action(s)</td><td>System Response(s)</td></tr>
<tr><td></td><td>Step 1: This use case is initiated when the user presses the “Map Mode” button.

Step 4: This use case is concluded when the users see the map reloaded with new map data.</td><td>Step 2: The system call the Map View Controller perform map mode switching.

Step 3: The system gets more map data of the selected mode from the Google Maps™ data provider.</td></tr>
<tr><td>Alternative Courses:</td><td colspan="2">none</td></tr>
<tr><td>Post-condition or Results:</td><td colspan="2">none</td></tr>
<tr><td>Implementation Constraints and Specifications:</td><td colspan="2">none</td></tr>
</table>

Use Case ID:	UC04	
Use Case Name:	Search bus routes	
Actor(s):	Android™ phone user	
Description:	This user case describes the process when the user search a bus route with desired destination. On completion, the system shows desired bus stops location.	
Reference ID:	RQ04	
Precondition:	The bus stops viewer tab is selected.	
Trigger:	none	
Typical Course of Event:	**Actor Action(s)**	**System Response(s)**
	Step 1: This use case is initiated when the user presses the "Search" button. Step 2: The user key in the desired destination or bus route id. Step 5: This use case is concluded when the user sees the closet bus stop that will reach the desired destination is displayed.	Step 3: The system get the current location and desired destination. Step 4: The system call the search route handler on server to look for desired bus stop.
Alternative Courses:	Step 5a: This use case is concluded when a list of bus routes that will reach the same destination is displayed. The closet bus stops that will reach the desired destination are displayed.	
Post-condition or Results:	Selected bus route information is stored at handset local database for arrival prediction.	
Implementation Constraints and Specifications:	none	

Use Case ID:	UC05	
Use Case Name:	Display path to the bus stops	
Actor(s):	Android™ phone user	
Description:	This user case describes the process when users look for the route to the desired bus stop. On completion, the system shows a path to the desired bus stops on the map.	
Reference ID:	RQ05	
Precondition:	The bus stops viewer tab is selected.	
Trigger:	none	
Typical Course of Event:	**Actor Action(s)**	**System Response(s)**
	Step 1: This use case is initiated when the user presses on a desired bus stop and set it as destination. Step 2: The user presses the "Show the direction" button. Step 5: This use case is concluded when the user sees a path to the bus stop is displayed on the map and distance is told.	Step 3: The system get the current location and desired destination. Step 4: The system calls the Google Maps™ Direction controller to look for a path.
Alternative Courses:	Step 2a: The user presses the "Show by radar" button. Step 4a: The system calls the radar application to look for a path. Step5a: This use case is concluded when the distance and direction to the desired bus stop is shown on a radar.	
Post-condition or Results:	none	
Implementation Constraints and Specifications:	none	

Use Case ID:	UC06	
Use Case Name:	Get bus arrival prediction	
Actor(s):	Android^{TM} phone user	
Description:	This user case describes the process when the user look for the predicted bus arrival time. On completion, the system shows the predicted bus arrival time.	
Reference ID:	RQ06	
Precondition:	The timer tab is selected.	
Trigger:	none	
Typical Course of Event:	**Actor Action(s)**	**System Response(s)**
	Step 1: This use case is initiated when the user presses the "Predictor" button. Step 2: The user select the desired bus route at the nearest bus stop on the map.	Step 3: The system get the current location and desired route information. Step 4: The system calls the predictor handler on server to calculate the arrival prediction. Step 5: This use case is concluded when the system shows the predicted arrival time and suggests alternative routes to the destination (UC04).
Alternative Courses:	none	
Post-condition or Results:	none	
Implementation Constraints and Specifications:	none	

Use Case ID:	UC07	
Use Case Name:	Record travelling time	
Actor(s):	Android™ phone user	
Description:	This user case describes the process when the user records the travelling time. On completion, the system stores the journey data to the database as prediction data references.	
Reference ID:	RQ07	
Precondition:	The timer tab is selected.	
Trigger:	none	
Typical Course of Event:	**Actor Action(s)**	**System Response(s)**
	Step 1: This use case is initiated when the user presses the "Start Timer" button. Step 2: The user select the desired bus route for recording.	Step 3: The system get the current location and desired route information. Step 4: The system records the time when the user reaches each bus stop by AGPS location detection. Step 5: This use case is concluded when the system records down the time when the bus arrives at each bus stop.
Alternative Courses:	none	
Post-condition or Results:	The recorded time data are stored at remote database as historical AVL data for future predictions references.	
Implementation Constraints and Specifications:	none	

Use Case ID:	UC08	
Use Case Name:	Take pictures when travelling	
Actor(s):	Android™ phone user	
Description:	This user case describes the process when the user takes pictures when travelling on the bus. On completion, the system stores the photos at SD card and shows the photos on the map.	
Reference ID:	RQ08	
Precondition:	The camera tab is selected.	
Trigger:	none	
Typical Course of Event:	**Actor Action(s)**	**System Response(s)**
	Step 1: This use case is initiated when the user presses the "Take picture" button.	Step 2: The system get the current location. Step 3: The system takes a photo. Step 4: This use case is concluded when the system stored the photo is stored at SD card and photo is displayed on the map according to the location when it is taken.
Alternative Courses:	none	
Post-condition or Results:	The photos taken are stored in a SD card.	
Implementation Constraints and Specifications:	none	

3.4 Database Structure

route

+ROUTE_ID	**varchar(10)**	**Nullable = false**
+BUS_STOP_ID	**integer(10)**	**Nullable = false**
BUS_STOP_NAME	varchar(255)	Nullable = true
BUS_STOP_AREA	varchar(255)	Nullable = true
NORMAL_FARE	decimal(5, 2)	Nullable = true
AIR_CON_FARE	decimal(5, 2)	Nullable = true
+DIRECTION	**char(1)**	**Nullable = false**
MAX_STOP	integer(10)	Nullable = true
BUS_STOP_NAME_UTF8	varchar(255)	Nullable = true
BUS_STOP_AREA_UTF8	varchar(255)	Nullable = true

historical

+ROUTE_ID	**varchar(10)**	**Nullable = false**
+BUS_STOP_ID	**integer(10)**	**Nullable = false**
LAT	decimal(10, 10)	Nullable = false
LONG	decimal(10, 10)	Nullable = false
TIME	time(10)	Nullable = false
Date	date	Nullable = false

stops_position

+ROUTE_ID	**varchar(10)**	**Nullable = false**
+BUS_STOP_ID	**integer(10)**	**Nullable = false**
LAT	decimal(10, 6)	Nullable = false
LONG	decimal(10, 6)	Nullable = false
+DIRECTION	**integer(5)**	**Nullable = false**

avl

+ROUTE_ID	**varchar(10)**	**Nullable = false**
+BUS_STOP_ID	**integer(10)**	**Nullable = false**
+BUS_ID	**integer(10)**	**Nullable = false**
LAT	decimal(10, 10)	Nullable = false
LONG	decimal(10, 10)	Nullable = false
DWELL_TIME	integer(10)	Nullable = false
ARRIVE_AT	time(10)	Nullable = false
LEAVE_AT	time(10)	Nullable = false
DATE	date	Nullable = false
TIME	time(10)	Nullable = false

schedule

+ID	**integer(10)**	**Nullable = false**
+ROUTE_ID	**varchar(10)**	**Nullable = false**
+DIRECTION	**char(1)**	**Nullable = false**
+WEEK_DAY	**integer(1)**	**Nullable = false**
+PERIOD_START	**time(10)**	**Nullable = false**
+PERIOD_END	**time(10)**	**Nullable = false**
TIME_BETWEEN_MIN	integer(10)	Nullable = true
TIME_BETWEEN_MAX	integer(10)	Nullable = true

district

+ID	**integer(11)**	**Nullable = false**
BUS_STOP_DISTRICT	varchar(255)	Nullable = true

district_area

+DISTRICT_ID	**integer(11)**	**Nullable = false**
BUS_STOP_AREA	varchar(255)	Nullable = true

direction_from_to

+ROUTE_ID	**varchar(10)**	**Nullable = false**
+DIRECTION	**char(1)**	**Nullable = false**
FROM	varchar(255)	Nullable = true
TO	varchar(255)	Nullable = true
FROM_UTF8	varchar(255)	Nullable = true
TO_UTF8	varchar(255)	Nullable = true

Figure 3.4a: Database design of bus arrival predictor

This application consists of five tables in the remote database for handling arrival prediction as shown in **Figure 3.4a**:

- ***route***: Table contains all bus routes for the arrival prediction.
- ***stops_position***: Table contains all bus stops locations of all routes in Route table.
- ***schedule***: Table contains all schedules of bus routes in Route table.
- ***historical***: Table contains all historical data of previous travelling data by the application user.
- ***avl***: Table contains the assumption set of automatic vehicle location data and is used for arrival prediction with historical data.
- **district**: Table contains all the 18 districts names and the corresponding ids.
- **district_area**: Table contains all the bus stop area names and it has a foreign on the “ID” of table “district”.

3.5 Class Structure

Classes relationships under the platform of Google Android™ were different from normal Java projects which consists of lots of dependencies, inheritances and and associations. As the framework of Android™ was components based, it was suggested that classes (java objects) to be reusable independently as a whole, therefore classes were classified into four main components which will be discussed in section 5.2.1.

Figure 3.5a described the nested package structure of the Android™ project. Basically, the project was divided into several packages:

1. **constant**: It contains all the shared constants between classes such as server ip and port.
2. **adapter**: It contains all the list view controllers and database helpers.
3. **view**: It contains all the customized views.
4. **app**: It contains all the core activities of the application and also a system-gen class called R.java is placed there to store all the resources id.
5. **utils**: It contains all the helper class providing various utility functions.
6. **items**: It contains all the classes for data transfer, similar to Data Transfer Object.

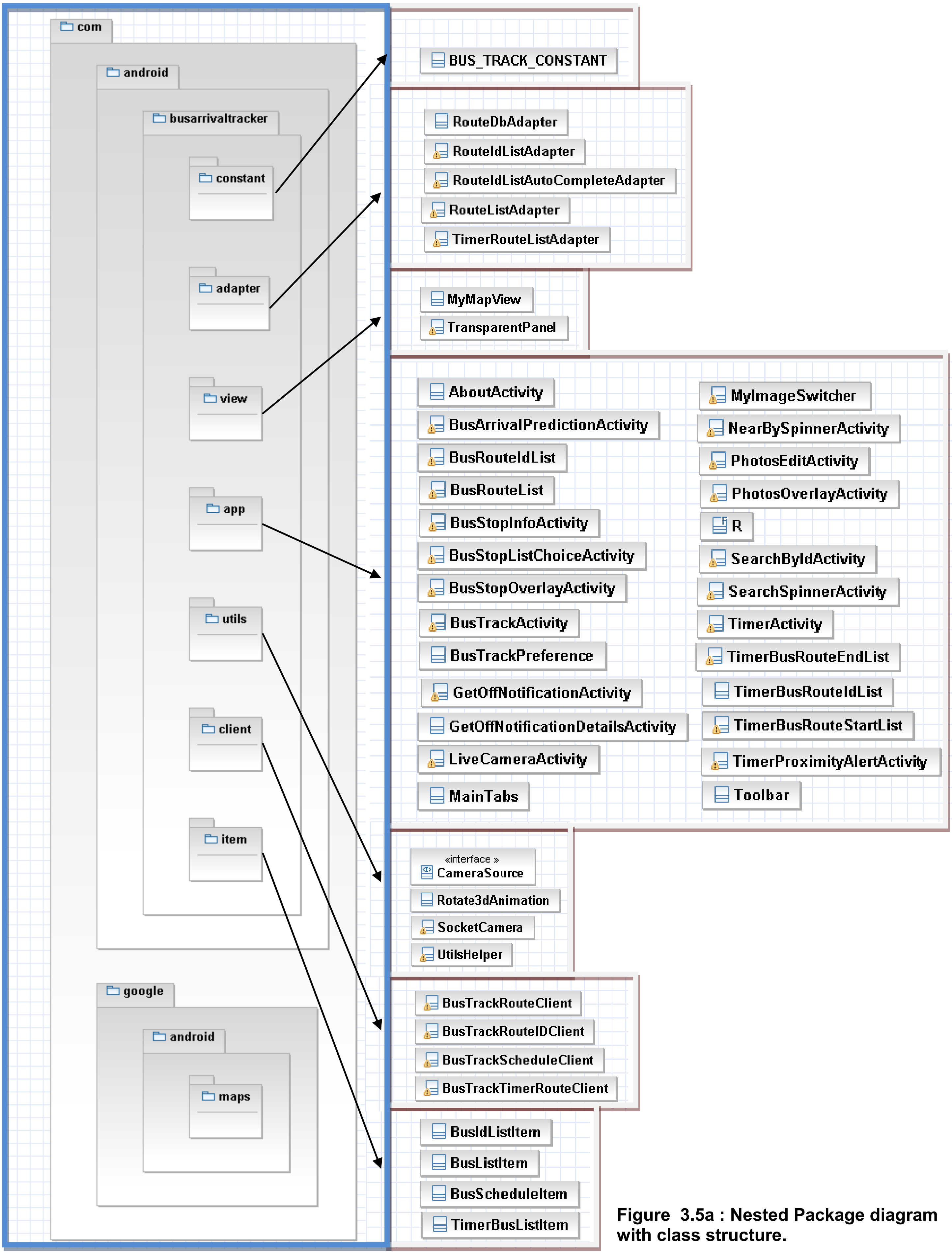


Figure 3.5a : Nested Package diagram with class structure.

3.6 Sequence Diagrams

In this section, **Figure 3.5a - 3.5h** shows the sequence of how the actions are called and how the data flow are made between different interfaces, handlers and data sources of each use case described above.

- **Figure 3.6a**: The sequence of getting AGPS data from the network and the procession of data from location manager to map interface.
- **Figure 3.6b**: The sequence of setting map modes. Map data are retrieved from the Google Maps™ data provider for map view display through map view controller.
- **Figure 3.6c**: The sequence of zooming map and zoomed map data are retrieved from Google Maps™ data provider for map view display.
- **Figure 3.6d**: The sequence of searching bus routes. Bus route information are calculated and retrieved from remote databases, route and stops_position, based on users' current location.
- **Figure 3.6e**: The sequence of getting a path of desired bus stop. Direction data are retrieved from Google Maps™ data provider. Alternatively, user can get direction data from G-sensor compass as in radar mode.
- **Figure 3.6f**: The sequence of getting bus arrival prediction and the prediction is based on server side computation using remote databases, schedule, avl and historical, as data sources. Afterwards, suggestions are made.
- **Figure 3.6g**: The sequence of recording travelling time. Current user location and retrieved bus stops location are compared, when user (bus) stops at each bus stop, times will be gathered to update historical records.
- **Figure 3.6h**: The sequence of taking photos when travelling and photos will be saved at SD card; which will be displayed accordingly on map based on user's current location.

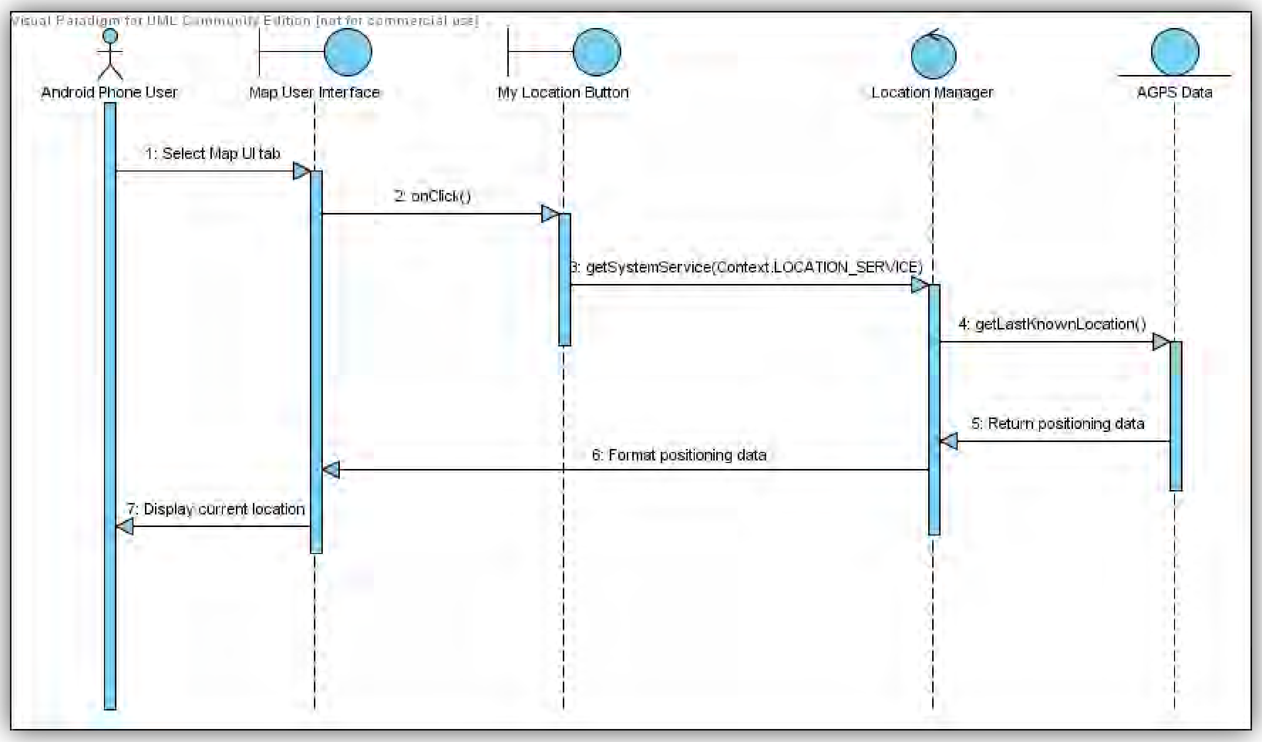


Figure 3.6a: Sequence diagram of getting current location

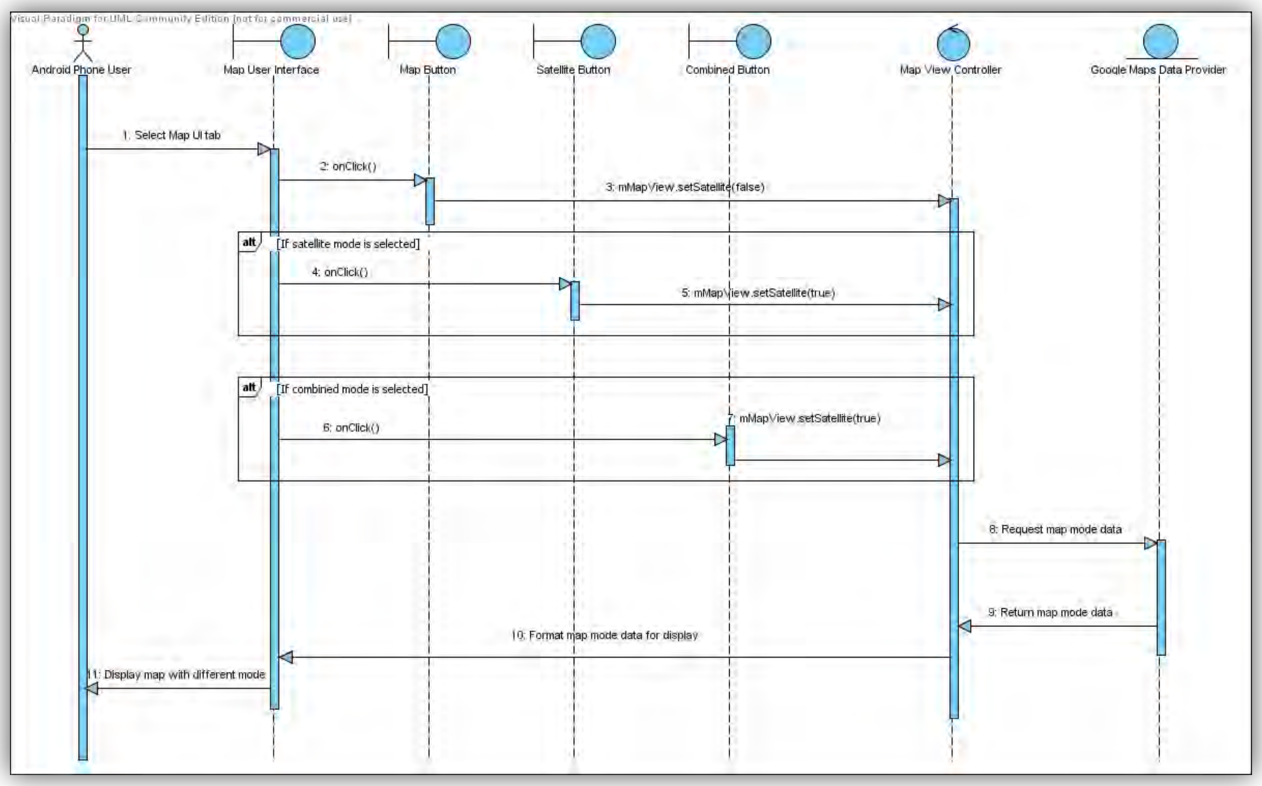


Figure 3.6b: Sequence diagram of choosing map modes

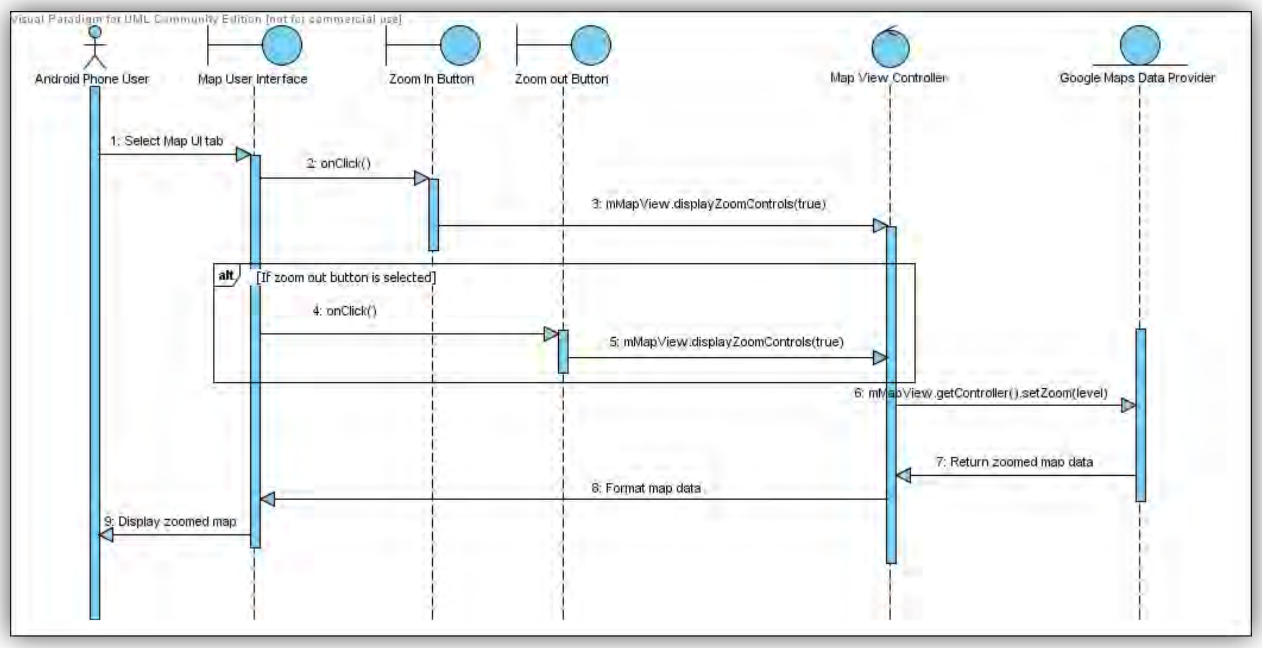


Figure 3.6c: Sequence diagram of map zooming

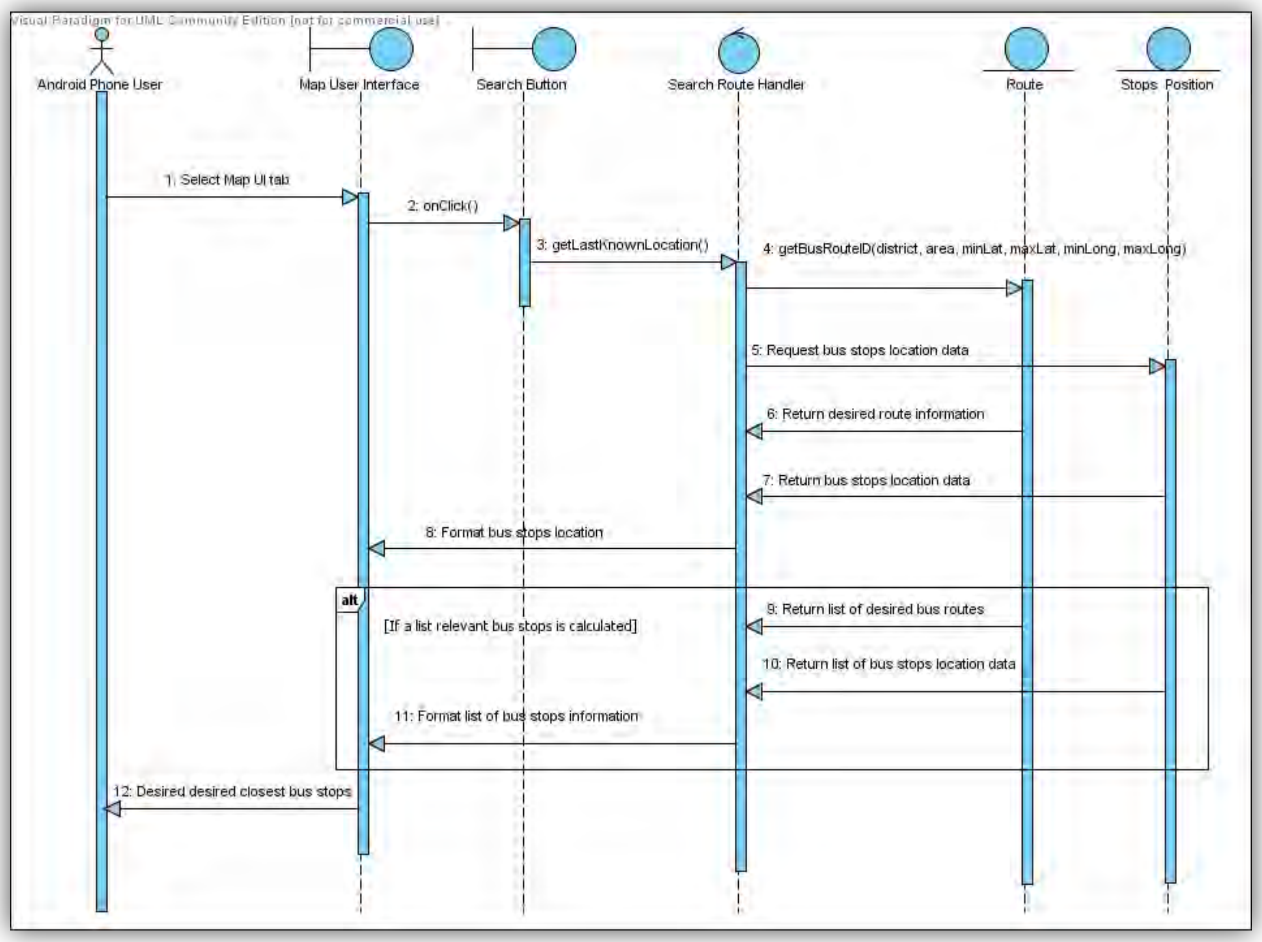


Figure 3.6d: Sequence diagram of searching bus routes

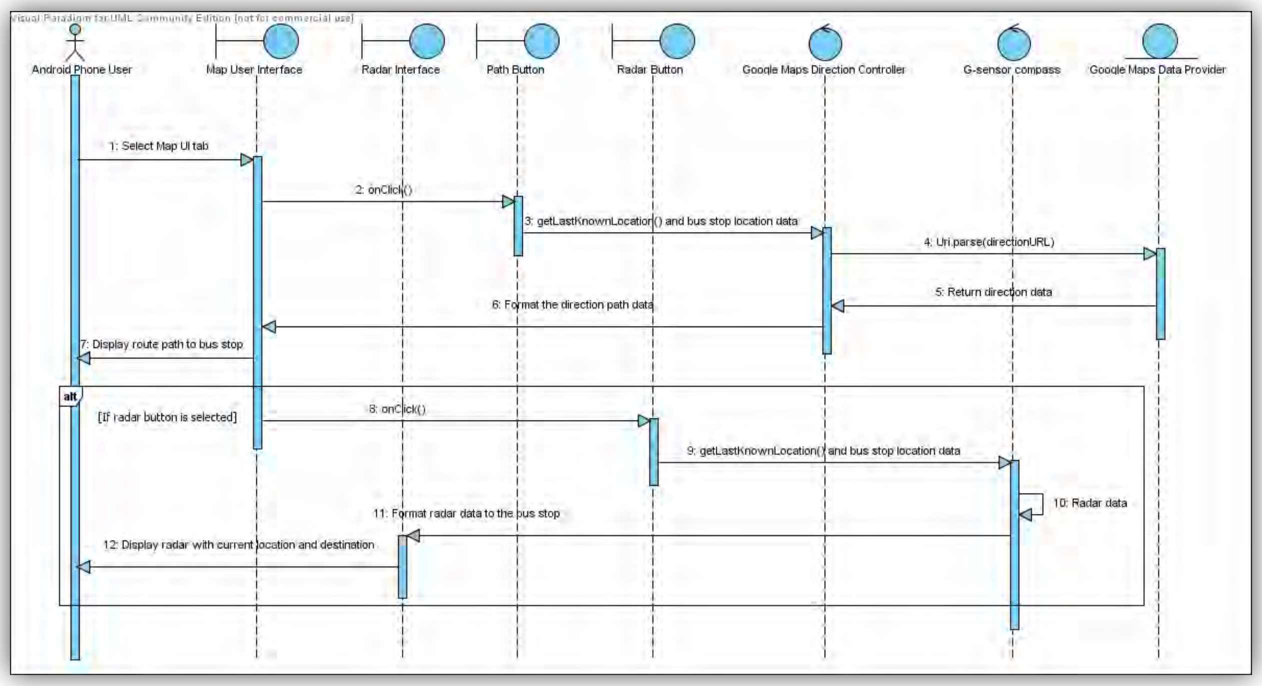


Figure 3.6e: Sequence diagram of getting path to bus stop

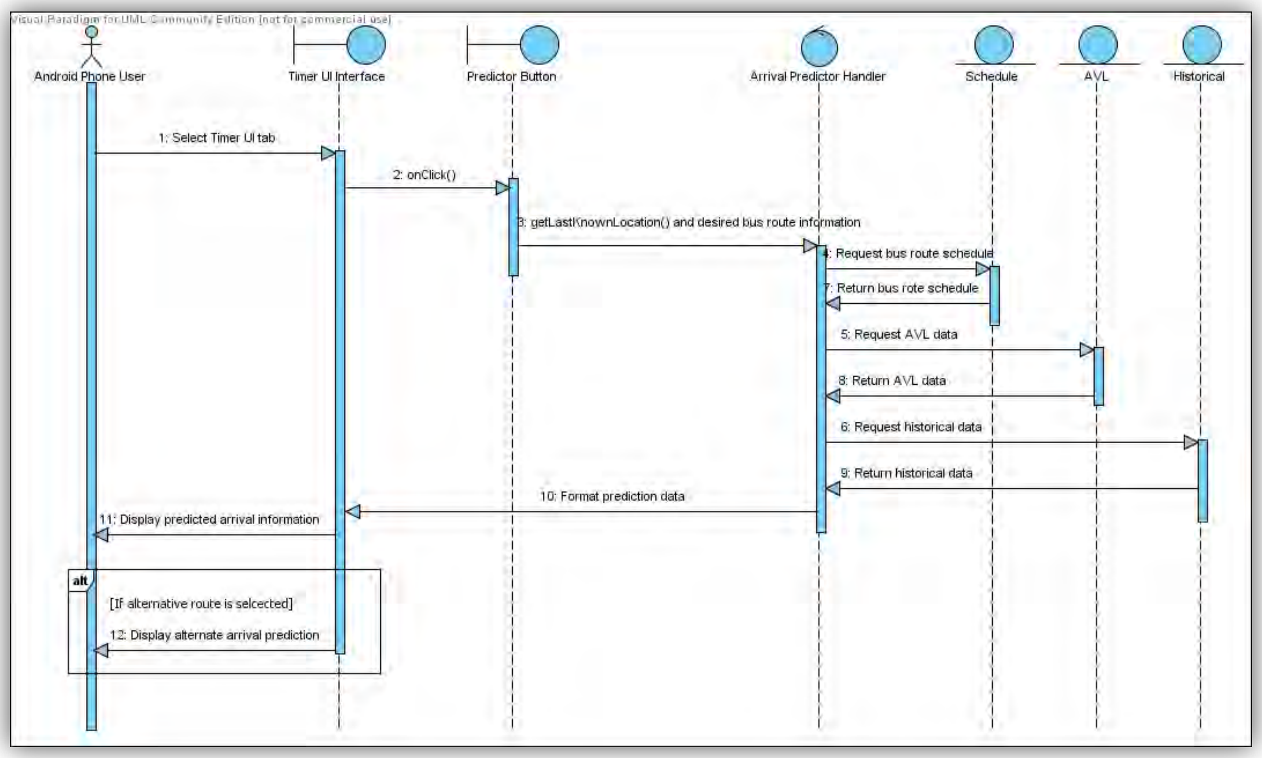


Figure 3.6f: Sequence diagram of getting bus arrival prediction

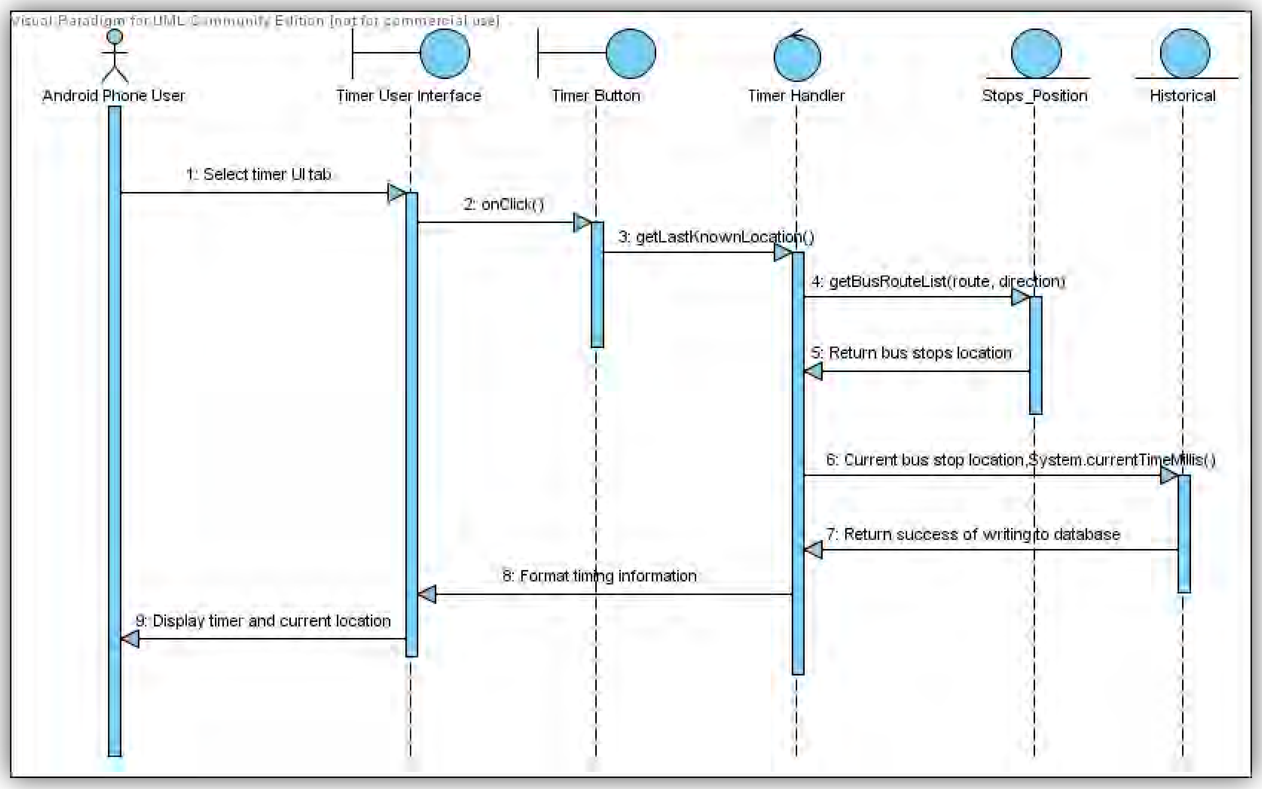


Figure 3.6g: Sequence diagram of recording travelling time

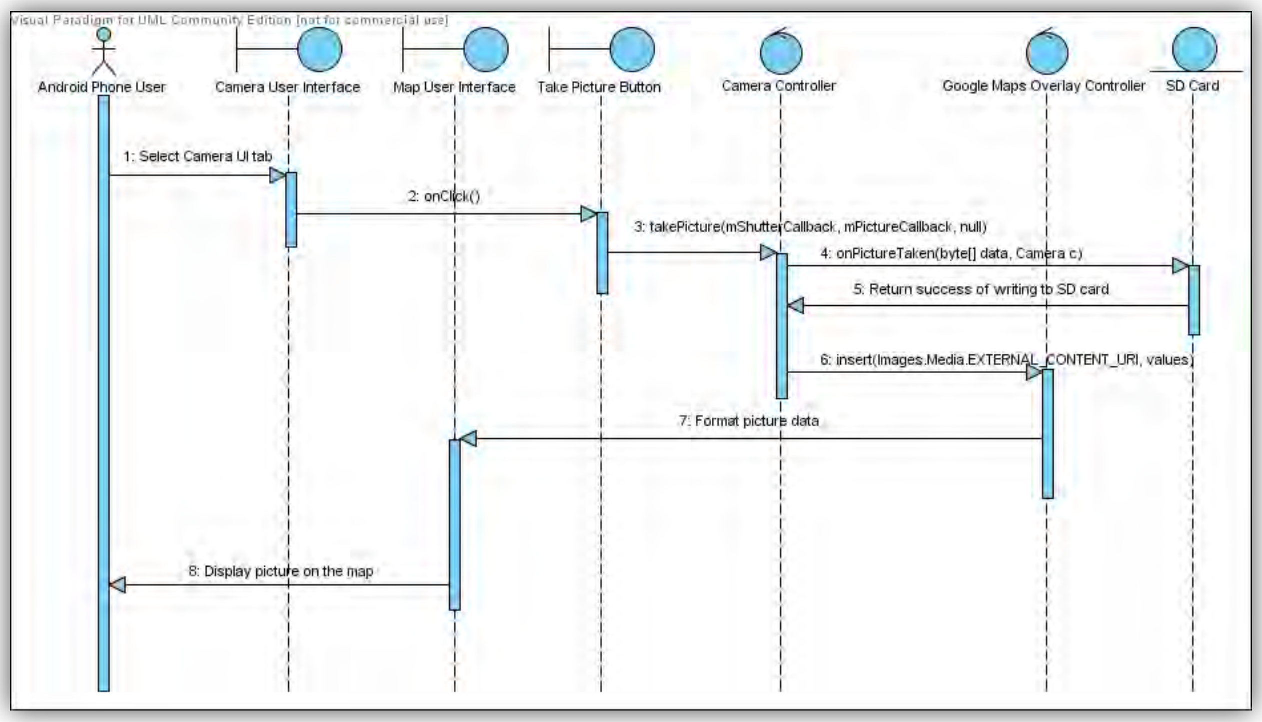


Figure 3.6h: Sequence diagram of taking picture when travelling

3.7 System Architecture

The system was implemented as a 3-tier application. **Figure 3.7a** shows the overall structure of the application. It includes a client application on the Android^{TM} phone, a Bus Arrival Predictor Application Server (BAPAS) and a remote database for managing routes related information and users data.

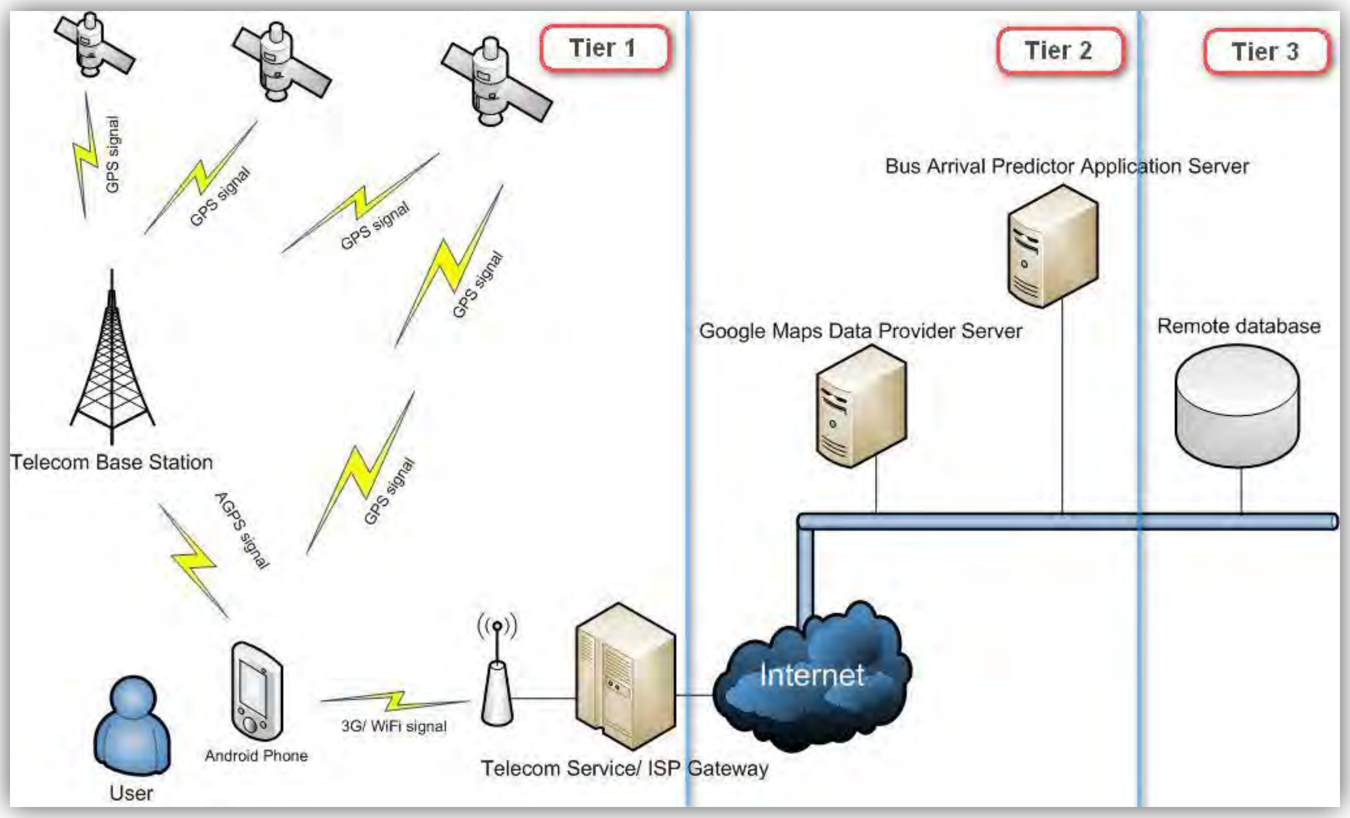


Figure 3.7a: System Architecture Design

A 3-tier application adopts the client/ server computing model. All of the three individual layers can be implemented with different languages and concurrently. Therefore, when there are changes on the business logic tier, reallocation of resources and implementations will not affect the other two. It enhances the flexibility of this application and eases a lot more programming on future system extension.

4 Methodology and Resources

4.1 Estimated Time of Arrival

As discussed above, there are several bus arrival prediction algorithms available. However, they share a common physical requirement that it is not feasible to work out on this project, the acquisition of real time location information. As in Hong Kong, there are no AVL or APC systems running by bus providers or they are under development and are not opened to public. In addition, it is not possible to ask for an installation of APC and AVL devices on each bus in Hong Kong at this stage; therefore obtaining a real time location of the buses causes a difficulty. Although this project is not focusing on evaluating the efficiency of each prediction algorithm, we still have to get the location of the buses to simulate and calculate the bus arrival time. Hence, this project was insisting on a set of simulation data to simulate the real time location of the buses.

The set of simulation data did most of the location tracking information part. The idea was that to simulate the actual position of the buses on track and the idea was based on AVL and APC. The estimation was mainly based on a set of assumed historical data of the operating buses, including:

- buses schedules,
- travelling time between bus stops,
- difference in travelling at peak and non-peak hours,
- difference in travelling on weekdays and holidays,
- travelling date and time,
- latitude,
- longitude,
- dwell time, etc.

The calculation of the estimated time was based on the server side, while the client side, Google Android^{TM} application, would do the request, retrieve the result for display, and map manipulations. Besides the assumption set of data, users can get a much more precise time of arrival based on their previous journeys on that bus route, as discussed in section 4.4.

The estimation was done according to the selected bus routes at the current bus stop, that was different from the Electronic Terminus Management System provided by Kowloon Motor Bus Company Limited, which provided departure times at bus terminals only.

4.2 Map Manipulations

Digital maps are becoming prevalent nowadays; everyone should have came across with CentaMap™ and YelloPage™ when they are looking for information about their destinations. Google™ has become one of the most popular informative digital maps provider in the world recently and released Google Maps™ with positive feedbacks. This technology is evolving and takes a significant place in the market in our time.

As an integrated feature of Google Android™, Google Maps™ provides a great flexibility of map manipulations on HTC G1™. This project was to focus mainly on the functions with maps. Functions such as locating current user's location, bus stops locations and finding routes to the bus stops can be implemented easily with the powerful Google Maps™ APIs (Application Programming Interface). It eases the development for map manipulations on the handsets and strengthens the advantage of mobility of the application. Hence, that is the main reason, Goolge Android™ is chosen as the platform of this application.

4.3 Routing to bus stops

Google Maps™ supports suggestions on direction to the destination. The original web-based Google Maps™ would provide driving and walking directions with detailed distance and time information. However, in Google Android™, there only supports driving directions to the destination and is not open sourced anymore to developers other than authors. Therefore, to a user that is usually walking on the street far away from the main roads, driving direction to the bus stops does not work sometimes.

To overcome the problem, this project explored a groundbreaking technology of Google Android™, the integrated G-sensor compass inside G1™. As with ordinary

GPS device, facing direction cannot be determined, that is traditional applications can provide users a path, but which direction is the users facing, the system cannot resolve. Therefore, users will be stuck on which way they should go in order to find and follow the suggested path. With the compass, we facilitated the route finding by implementing a radar application, which can show the distance and direction to the bus stops in a more perceptive way, as like as reading on compass.

4.4 Bus Travelling Timer

As mentioned before, the real time information of the bus location forms an important part of predicting arrival times. Yet, we lack the real set of AVL and APC data, we insisted on the assumption set of data. Bus Travelling Timer helped the prediction in a way that it recorded down the travelling time of users in a particular route. These stored travelling times were sent to the server and updates to the database will be made. These records were considered as historical data on the prediction. When a request on the arrival time is made, historical data will be taken into account for calculation, so that an average arrival time of the assumption set and historical set is produced to increase accuracy.

Travelling time records are stored when the users have selected a desired route and activated the timer on the start. The application will then note down the time when the bus arrives at each bus stop, this is going to be done with the proximity alert in Android™, as the phone is approaching to a chosen destination (bus stop), specific functions (notificaitons) can be called.

4.5 Google Android™

Google Android™ is chosen as the platform of this application. The reason is that Google Android™ provides a suite of Google™ registered applications that make it more competitive as an all-in-one communication device. Users can perform all the works with just a single sign-on and emails, messages, chats, contacts, etc. will be synchronized remotely with their desktops and Gmail™ accounts. Therefore, applications, which are built on Android™ has the advantage of interrelating all the registered products of Google™. Also, modular programs can be easily created by the system nature of Android™ that it is originally structured. Hence, reusable and extensible codes can be developed, it enriches, and smoothes the progress of development experience.

One shortage and is also the strength, Android^{TM} is still on the evolving stage, several things, such as documentation and APIs are not fixed yet and is still changing overtime, therefore applications based on previous versions may not work with new versions. While bearing this, Android^{TM} has the ability to be extensible and refined so that greater possibilities will the applications have.

5 System Implementation

After the discussion of the main functions of the Bus Arrival Predictor, a comprehensive description of the predictor should be provided in order to build the application. In this section, the interface components, steps of use and the detailed implantation issues of this Bus Arrival Predictor were introduced and discussed.

5.1 Interface Design

In this section, the application is introduced. As this application was developed under the Android™ emulator, the screens were all captured from the emulator and this should be the same as seen on the real HTC G1™.

Figure 5.1a: Interface of the Android environment

Figure 5.1b: Interface of the Bus Arrival Predictor

5.1.1 Three main tabs for three main features

This application was mainly developed into three main tabs and each of them represented a main feature of the Bus Arrival Predictor, they were as follows:

- **Predictor**: The core of this application as most of the functions about bus route information were implemented inside this tab. (**Figure 5.1.1a**)
- **Camera**: The tab contained a camera interface allowing users to take photos while they are on their journey. (**Figure 5.1.1b**)
- **Timer**: The tab allowed the users to time their journey and set notification on their way to get off the bus. (**Figure 5.1.1c**)

Figure 5.1.1a: Screen of the Predictor Tab

Figure 5.1.1b: Screen of the Camera Tab

Figure 5.1.1c: Screen of the Timer Tab

5.1.2 Main functions of Predictor

In this section, the operational screens of the main functions of the Predictor were presented. Also, the operational steps were shown with brief descriptions. Functions menu was activated through the press of the "Menu" button on the phone and **Figure 5.1.2a and 5.1.2b** shows the screens when functions menu popped up.

Figure 5.1.2a: Function menu popped up

Figure 5.1.2b: Function menu when “More” is pressed.

5.1.2.1 Map Mode

In this function, a submenu popped up and users can select their desired map mode for the map view by clicking one of them. There were altogether three different modes for selection, the Plain Map mode, the Satellite mode and the Traffic mode. **Figure 5.1.2.1a** displays the screen with submenu.

Figure 5.1.2.1a: Submenu for map mode and Satellite mode is selected.

5.1.2.2 Map Zooming

This function did the normal zoom-in and zoom-out of the map. It can be activated by pressing the “Zoom” button on the function menu or by a single tap on the map view. Figure 5.1.2.2a shows a zooming control panel appeared at the bottom of the screen

Figure 5.1.2.2a: A zooming control panel appeared.

5.1.2.3 View Photos on Map

Photos taken during a trip or a ride on the bus were saved on the SD card in the phone. Each photo would be saved together with a pair of latitude and longitude coordinates so that they could be shown on the map view with markers. **Figure 5.1.2.3a** shows photos markers on different locations on the map. When users single tap on the marker, a preview of the photo popped up and users can choose to view the details of the photo or to let the application guide them back to where the photo was taken.

Figure 5.1.2.3a: Photos markers on the map.

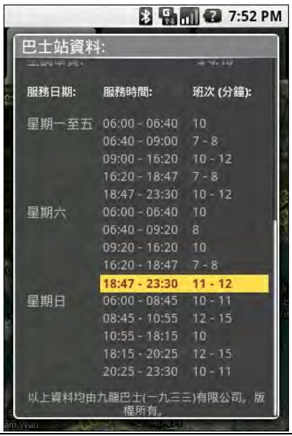
Steps for getting direction back to where the photo was taken		
Steps	**Description**	**Screens**
1	Users tap on the "Take me there" button when the photo preview popped up.	
2	A route was generated with detailed description of turns and distance.	
3	The guiding route was shown on the map and let users to follow.	

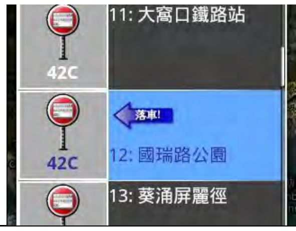
5.1.2.4 Search Bus Routes

This function let the users to search about bus route information such as the destination, bus stops names, bus stops photos, schedule and fares. It was implemented as two search categories, search by bus route number and search by origin and destination. **Figure 5.1.2.4a** illustrated the selection of the two search types.

Figure 5.1.2.4a: Screen of searching bus route information by two types.

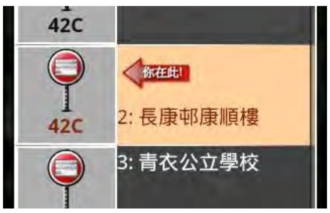
Steps for searching bus route information by route number		
Steps	**Description**	**Screens**
1	Enter route number into the search field at the center.	
2	Auto-complete was provided to reduce the number of typing.	
3	Select the direction of the route.	

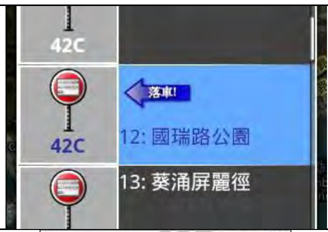
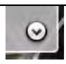
4	A list view was shown to indicate the number and name of each stop.	
5	Detailed information of a particular bus stop was shown with picture, fares and schedules. A highlighted timeslot indicated the real time schedule of the bus route.	 

Steps for searching bus route information by destination		
Steps	**Description**	**Screens**
1	Select the district and area of your origin and destination.	
2	Selection was implemented with spinner so that no typing was needed to increase the efficiency.	
3	Select the direction of the route.	*Same as search by route number.*
4	A list view was shown to indicate the number and name of each stop. In the list, a "Get on" and a "Get off" indicator showed the bus stops to get on the bus and get off the bus so that users could arrive their destination.	
5	Detailed information of a particular bus stop was shown with picture, fares and schedules. A highlighted timeslot indicated the real time schedule of the bus route.	*Same as search by route number.*

5.1.2.5 Search Nearby Bus Stops

Based on the users current location, retrieved from GPS receiver, users can look for the nearest bus stops that provide suitable bus routes and reach their destination. In this function, alternative bus routes were suggested, distance and direction to the bus routes were calculated and provided and bus arrival time was predicted.

Steps for searching nearby bus stops		
Steps	**Description**	**Screens**
1	Select the district and area of your destination.	
2	A window popped up. It indicated the number of suitable bus routes near you and asked if you would like to view their positions on map.	
3a (i)	If you choose not to view bus stops on map, a list of suitable bus routes was shown with a button "Show on map" at the top of the list.	
3a (ii)	A list view was shown to indicate the number and name of each stop. In the list, a "You are here" and a "Get off" indicator showed the nearest bus stops to get on and bus stop to get off the bus so that users could arrive their	

	destination.	
3b	If you choose to view bus stops on map, bus stops markers were shown on the map to indicate their positions and your current location.	
4	A single tap on the marker would trigger a panel showing the corresponding bus route numbers in that bus stop on the top of the screen.	 
5	A single tap on  would pop up a window of a picture of that bus stop, showing its name, your distance to that stop and three buttons to show bus arrival time(6a), direction to the bus stop(6b) and a radar mode direction to the bus stop(6c).	 

6a	A list of predicted bus arrival time was shown.	
6b	A direction showing how to get to the bus stop from your current location.	
6c	Radar mode showing the direction and distance between you and the bus stop and it works just like a compass by integrating the built-in G-sensor compass in the phone.	

5.1.2.6 Clear Map

This function clears all the markers on the map view and reset the map mode back to the Plain map mode.

Steps for Clearing Map		
Steps	**Description**	**Screens**
1	Tap on “Clear Map” on the pop up function menu and Plain map mode would be shown.	

5.1.2.7 Settings

Settings stored all the shared preferences used by this application, including the followings:

- **Language preference**: Setting the language used. (*English and Traditional Chinese*)
- **Reachable preference**: Setting the default searching distance used to look for the nearest bus stops. (*200 meters and 400 meters*)
- **Get Off Notification preference**: Setting the default distance used to notify the users to get off the bus before the destination. (*30 – 100 meters*)

Figure 5.1.2.7a: Screen of the preferences.

5.1.3 Main functions of Camera

In this section, the operational screens of the main functions of the Camera were presented. Also, the operational steps were shown with brief descriptions. Functions menu was activated through the press of the "Menu" button on the phone and **Figure 5.1.3a** shows the screens when functions menu popped up. As shown from the figure, the background of the camera tab should be the real time image captured from the built-in camera on the phone.

Figure 5.1.3a: Function menu popped up.

5.1.3.1 Take Pictures during your journey

In addition to the phone's built-in camera function, this function would also retrieve and save the users' current location into the photo through GPS receiver. Photos taken by this application would be put together in the same album called the "Bus Arrival Predictor" and saved on the SD card.

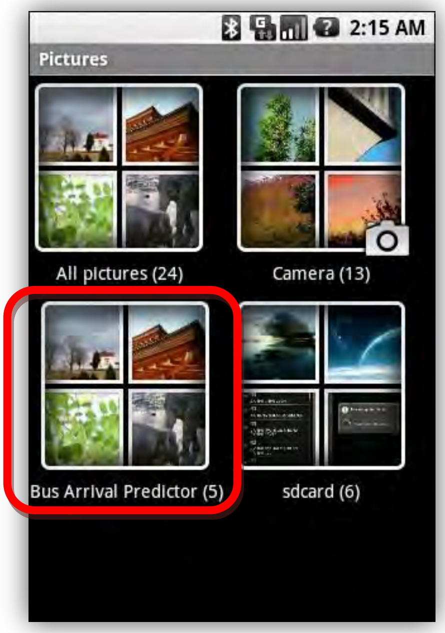


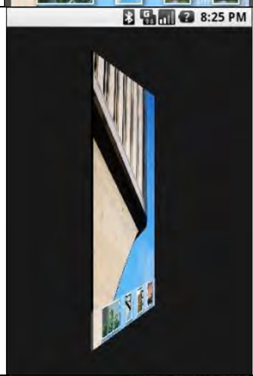
Figure 5.1.3.1a: Photos taken were saved in the album "Bus Arrival Predictor".

5.1.3.2 View Pictures in phone

Users can view the photos taken and pictures saved on the phone's local memory or on the SD card. It provides a shortcut to the phone's picture directory and is the same as viewing pictures from the built-in Picture function.

5.1.3.3 View/Edit My Gallery

Besides viewing all your photos as stated in section 5.1.3.2, users can particularly view the "Bus Arrival Predictor" album through this function. Users can even edit the information stored in the photos, such as the photo's name, title, bus route taken when taking this picture and description of this photo. You can type in whatever information in description.

<table>
<tr><th colspan="3">Steps for Viewing/ Editing my gallery</th></tr>
<tr><th>Steps</th><th>Description</th><th>Screens</th></tr>
<tr><td>1</td><td>Slide left or right to cycle through on the thumbnails at the bottom. A full screen preview is shown.</td><td>
</td></tr>
<tr><td>2</td><td>Single tap on the photo to trigger an animation of flipping to simulate turning to the back of the photo. It enhances users' experience.</td><td>
</td></tr>
<tr><td>3</td><td>Edit the information as shown and then save or cancel the changes.</td><td>
</td></tr>
</table>

		
4	Tap on “Take me there” would show a route direction bringing you back to where the photo was taken.	*Same as getting direction back to where the photo was taken.*

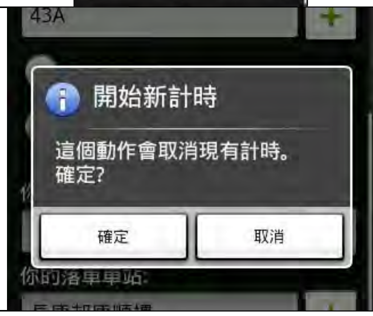
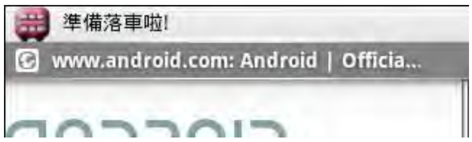
5.1.4 Main functions of Timer

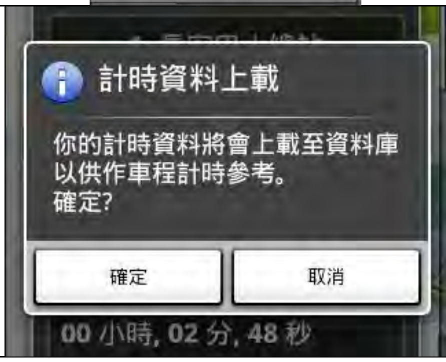
In this section, the operational screens of the main functions of the Camera were presented. Also, the operational steps were shown with brief descriptions.

5.1.4.1 Setting Timer

Users can setup a timer to time their journey on bus and work as a preference for arrival prediction. In addition, a get-off notification would be setup to notify users to get-off the bus when they are near to the destination.

Steps for Setting Timer		
Steps	**Description**	**Screens**
1	Enter your bus route number that you are taking and select the direction from where you are heading to.	
2	Select the bus stop where you get on the bus by clicking the “+” button..	

3	Select the bus stop where you will get off the bus. Only bus stops after the bus stops you have selected in step 2 would be shown.	 
4	Start the timer.	
5	When you are near to the destination, a sound would be played and the phone would vibrate for seconds. A notification message is shown on the screen also.	 

6	View the notification by pulling down the notification bar at the top of the screen.	
7	Tap on the notification and a window would pop up showing a summary of your ride including total time taken.	
8	Upload the timing information to the server as a preference for arrival prediction as you wish.	

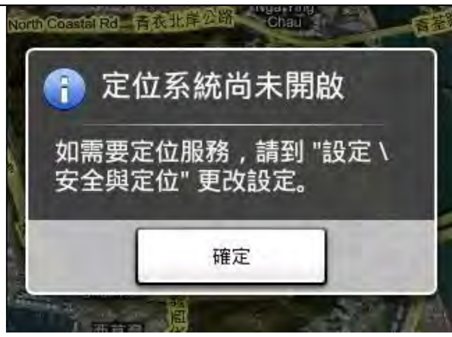
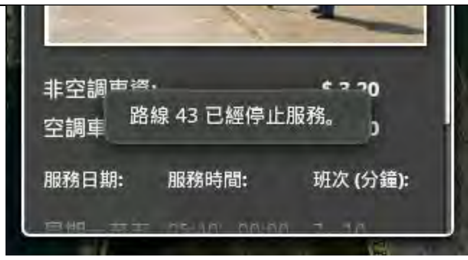
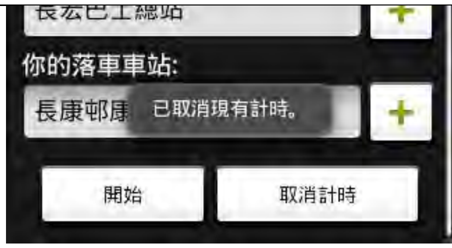
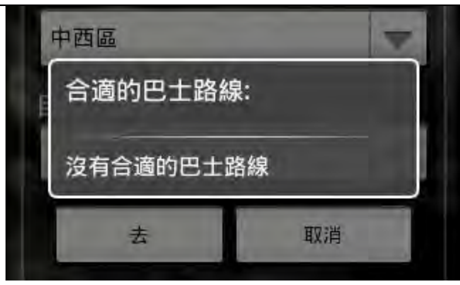
5.1.4.2 Clear Timer

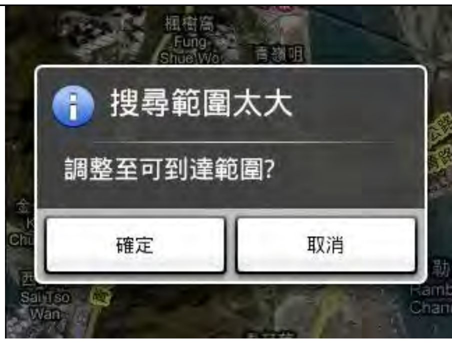
This function clears all the timers that were set and saved into the database before.

Steps for Clearing Timer		
Steps	**Description**	**Screens**
1	Tap on "Cancel Timer" at the bottom and a short message will be shown to indicate the timer was cancelled.	

5.1.5 Notification and Error Detection

In this section, different kinds of the interface for short notification and error detection of this application would be illustrated. Notifications were shown to get the users' attention to inform them something important have happened and maybe some actions were advised to be taken to tackle the problem.

Types of notification and error detection		
Type	**Examples**	**Screen**
Error detection (Dialog)	GPS receiver is not turned on.	
Notification (Short Message)	A particular bus route has stopped service, i.e. out of service hour.	
Notification (Short Message)	Timer has been cancelled.	
Notification (Dialog)	No suitable bus routes to your destination.	

Notification (Prompt Dialog)	To set to the reachable distance or not.	
Notification (Sound and Vibration)	The get off notification is triggered.	 

5.2 Implementation Issues

As the platform of Google Android™ is a newly developed operating system for mobile phones and it is evolving in the state that the current release version is the version 1.1_r1, several concepts and terms were discussed and introduced briefly before considering the detailed implementation issues.

5.2.1 Concepts and Terms

All applications that run on the platform of Google Android™ are written in Java programming language. Each application, including compiled codes, resources (e.g. images) and any required data are all bundled into an **Android Package**, an archive file with extension *"**.apk**".* This package file will be the file downloaded through **Android Market™**, a place where developers can publish their applications for users to download, to the users' phone and installed.

Before an application to start, the system will check the existence of a file called **AndroidManifest.xml** in the package**,** which is a structured XML file declaring all the properties of the application, such as naming all the libraries needed to be included, stating permissions to be granted to the application and describing the composing components of the application. **Figure 5.2.1a** shows an example of the XML file.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.android.busarrivaltracker.app" android:versionName="1.0.0"
        android:versionCode="1">
        <uses-permission android:name="android.permission.INTERNET" />
        …
        <application android:icon="@drawable/bus_track_icon"
                android:label="@string/app_name">
                <uses-library android:name="com.google.android.maps" />
                <activity android:name=".BusTrackActivity"
                        android:label="@string/app_name"
                        android:theme="@style/Theme.NoBackground">
                </activity>
                …
        </application>
</manifest>
```

Figure 5.2.1a: Example AndroidManifest.xml from the Bus Arrival Predictor.

Android applications are all composed of different components and each component can be reusable for different applications. Unlike most applications on other systems, Android applications do not have a single entry point for everything, i.e. the main() function. Instead, all components, i.e. Java objects, can be instantiated and run when as needed. There are altogether four main types of components:

1. **Activities**: The visual part of the application and an interactive user interface is provided to users, such as the MapActivity and ListActivity used in this application.
2. **Services**: Invisible to user interfaces and run as a background job.
3. **Broadcast Receivers**: Handles all the broadcast messages, such as battery is low.
4. **Content Providers**: Making a set of application data, such as contacts and photos are available to the applications.

In the Bus Arrival Predictor, all of the four components were utilized to handle all the operations needed and discussed in section 5.1.

5.2.2 Detailed Discussion

5.2.2.1 Time of Arrival Predicting

As mentioned before in section 4.1, this project was not evaluating the efficiency of each of the arrival prediction algorithms. Therefore, all of data consisting the bus arrival prediction are based on an assumption of data, which holds the assumed real time location of the buses and the time when they have arrived at particular bus stops in brief.

The aim to implement this function is to show the idea of bus arrival predictions can be incorporated into mobile applications and to inspire future developments on this idea. **Figure 5.2.2.1a** shows a flow chart describing the idea of getting the predictions from the assumption set of data.

The flow chart shows that getting the prediction data is convenient as all of the data are preset. Before querying on the prediction data, the system would check whether there are historical data or AVL data present in the database for prediction, if not, no prediction would be provided.

Besides, getting prediction data passively from the preset data, users can actively upload their travel time to the database so that more prediction would be available to other users. Users' contribution is important in this system as those data uploaded are really reflecting the actual situation when the users travel on a bus. Those uploaded data would even include the factor of traffic congestions and other factors affecting the schedule. Although those data are valuable, users have their own choice on whether to perform the upload in order to protect their privacy.

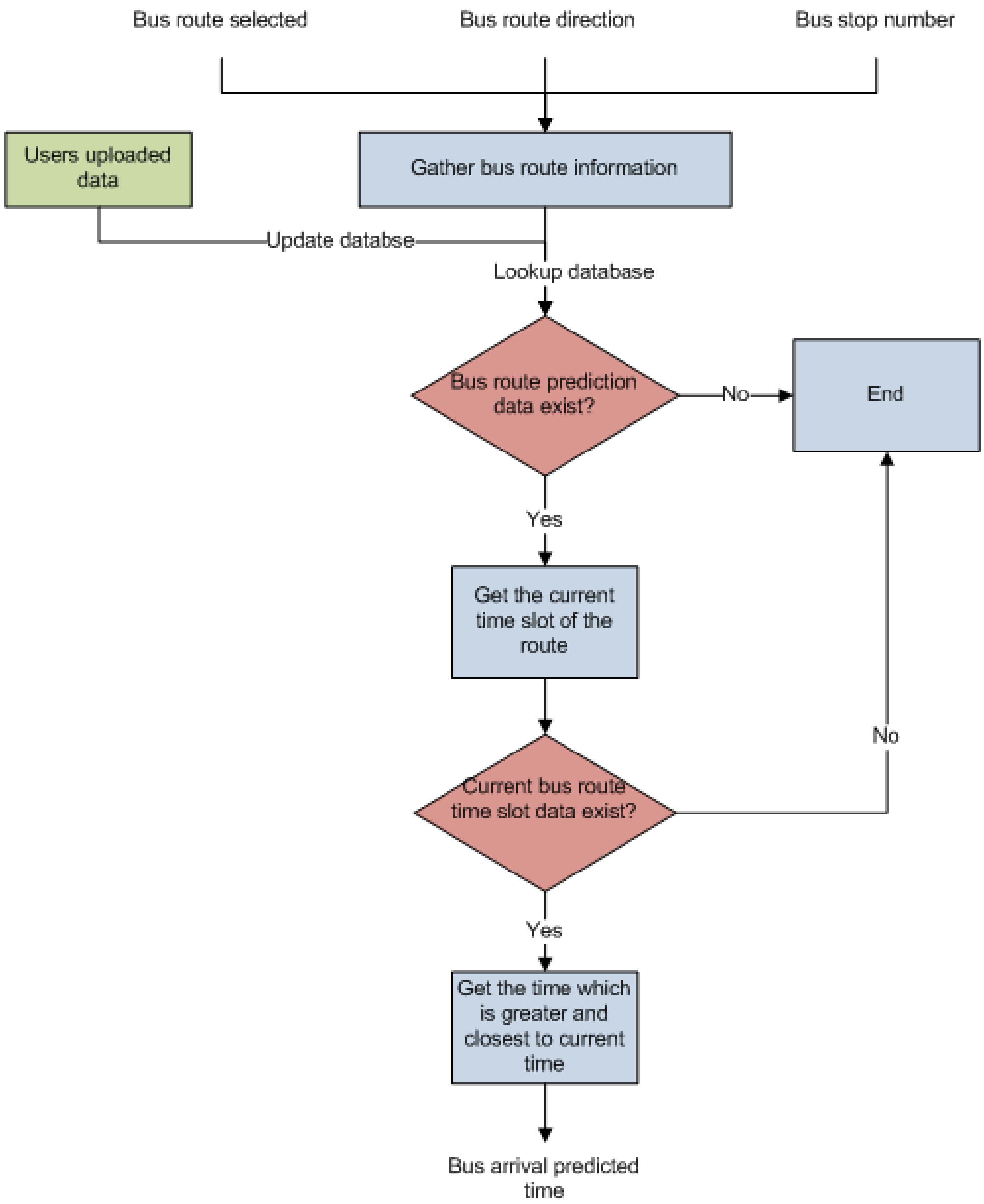


Figure 5.2.2.1a: The flows to obtain bus arrival prediction time.

5.2.2.2 Map Manipulating

For the various platforms of mobile phones, there are several ways to obtain maps data or location information. One of the ways was to view the maps online with the built-in mobile phone web browsers, such as the Opera Mini and the Safari, to obtain digital maps data, but the ability of map manipulations would be bound by the layout engine of the browsers and this may fail the users' expectations. Another way was to apply maps data from the Land Department or getting a specific portion of the map from some digital map providers as a still image, such as the Centamap, both are inefficient as acquiring those data are through a middleman.

In the platform of Google Android^{TM}, map manipulations can be done in a convenient way. Since the Google Android^{TM} platform has an embedded lite-weight version of the original Google Maps^{TM}, application would like to implemented a map view could include the package of the Google Maps^{TM} so that the application like the Bus Arrival Predictor could display and control a Google Map interface.

To create a MapView layout in the application, developers can follow some simple steps:

Step 1. Extend your activity with ***MapActivity***.

Step 2. Include the following into the ***AndroidManifest.xml*** file:

```
<uses-library android:name="com.google.android.maps" />
```

Step 3. Include a ***MapView*** into the layout.

After embedding a map view into the application, users can simply call methods provided by the MapView class to control the map interface, including map modes changing, map zooming and current location displaying.

5.2.2.3 Nearest Bus Stops Searching

Detecting the nearest bus stops can be further divided into the following two main steps:

1. Locating the nearest bus stops on the map.
2. Determining the bus stops located containing suitable bus routes to reach the destination. This step will be discussed in section 5.2.2.4.

In this part, the discussion would be focused on the step 1. To locate the nearest bus stops, the system will have to determine the searching area of the detection, which is called the **Reachable Distanc**e in this application. It defined the maximum distance between the users and potential bus stops, i.e. within the reachable distance, locate all the nearby bus stops.

Two approaches can be applied to locate the bus stops as illustrated in **Figure 5.2.2.3a and 5.2.2.3b**.

Figure 5.2.2.3a: Rectangular Search Area within the screen.

Figure 5.2.2.3b: Circular Search Area within the screen.

As shown on the above figures, circular search area approach is not applicable. There were roads missed by the search as there may be potential bus stops containing bus routes to reach the destinations.

To perform the rectangular search, we need to obtain the coordinates on the screen, as they will be mapped to the real latitude and longitude pairs on the Earth. Android SDK has provided some convenient methods to obtain those data. The **getLatitudeSpan()** and **getLongitudeSpan()** are used to get the horizontal and vertical span of the coordinates from the center point (users current location) on the screen. Therefore, minimum and maximum of latitude and longitude pairs can be obtained by the following codes:

```
// Get the search area
int latHalfSpan = mMapView.getLatitudeSpan() >> 1;
int longHalfSpan = mMapView.getLongitudeSpan() >> 1;

int latitudeE6 = center.getLatitudeE6();
int longitudeE6 = center.getLongitudeE6();

minLong = ((float) (longitudeE6 - longHalfSpan)) / MILLION;
maxLong = ((float) (longitudeE6 + longHalfSpan)) / MILLION;

minLat = ((float) (latitudeE6 - latHalfSpan)) / MILLION;
maxLat = ((float) (latitudeE6 + latHalfSpan)) / MILLION;
```

Therefore, the two corners' coordinates would be mapped to the rectangular search area as shown in **Figure 5.2.2.3c**.

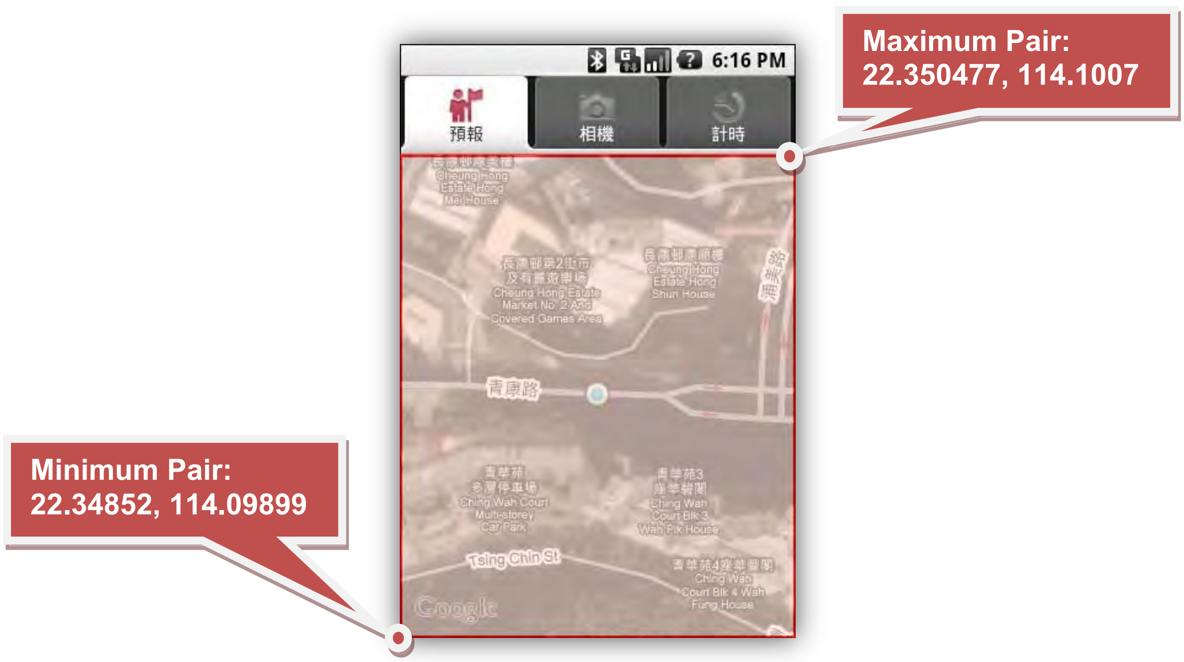


Figure 5.2.2.3c: The two pair of screen coordinates mapped to real latitudes and longitudes.

With the minimum and maximum pair, we can obtain all the bus stops in the search area easily with simple geometry calculations as all the locations are represented as a point on the map. And the sample results of the searching has been shown on the section 5.1.2.5, steps 3b.

5.2.2.4 Suitable Bus Routes Querying

After attaining the nearest bus stops on the map, we need to determine whether the bus stops we have found containing suitable bus routes. All of the calculations are done on the application server, the business layer, to reduce the computation load on the phone. The computation can be summarized into the flow chart on the next page, **Figure 5.2.2.4a**.

The flow charts showed the brief idea of determining the suitable routes. As facilitated by the database table structure, retrieved data can be informative with enough bus route details, such as bus stop number, bus stop name and direction. The chart shows that the returned bus routes are grouped by the latitudes and longitudes of bus stops, it is because there may be multiple bus stops around the users and providing suitable bus routes to reach the same destination, which are called the **alternative bus routes** in this application. Those alternative bus routes are important to users when they have missed their first choice of bus route, users can choose to take the alternatives so that time can be saved on travelling.

Another issue on the querying suitable bus routes is that, direction of the bus route should be defined when designing the data set in the database. Without the direction, querying on the dataset is pointless as multiple bus stops may return on the same screen, as sometimes bus stops are close when they are on the opposite side of a road, containing the same bus route heading to different destinations in opposite direction.

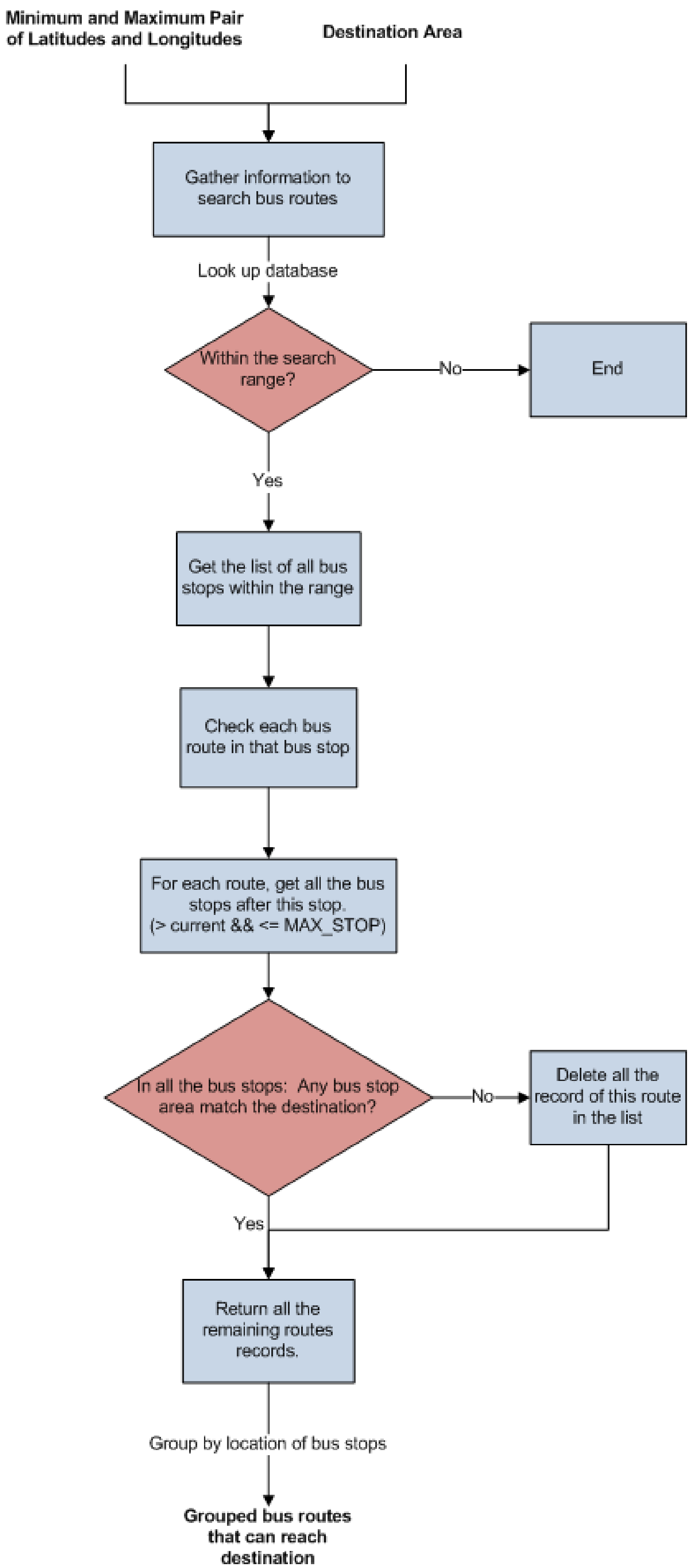


Figure 5.2.2.4a: The flows to obtain suitable bus routes.

5.2.2.5 Location Routing

Finding the direction to a particular location from users' current position can be implemented using the *Google DrivingDirections API* provided in the SDK version m5 but it has been removed from the API since version 1.0. Developers can no longer get access to the methods to implement and incorporate the driving directions feature inside their application.

In order to get the direction, developers have to make use of another method below to start a new intent. Intent is an asynchronous message that activated the components in the application. It got the name of the activity being demanded and specified the URI of the data being processed.

```
//Show the direction to the bus stop
ExampleActivity.this.startActivity(
      new Intent(Intent.ACTION_VIEW,
            Uri.parse(
                  "http://maps.google.com/maps?f=d" +
                  "&saddr=" + mCurrentLocation.getLatitude() +
                  "," + mCurrentLocation.getLongitude() +
                  "&daddr=" + mBusStop.getLatitude() +
                  "," + mBusStop.getLongitude() + "&hl=en"
            )
      )
);
```

A popup window would be displayed when you launch an activity through this intent as shown in **Figure 5.2.2.5a**. It was there to ask you to choose the method to view the direction, either by web browser or on the map, users should choose the latter one and direction information would be provided by Google with details, in terms of turns and stops.

Figure 5.2.2.5a: Popup window questioning about the method used to view direction.

One of the shortages of using the original Google direction is that, it provided only driving directions. It is sometimes meaningless for users to follow the driving directions to get to the nearest bus stops. See Figure **5.2.2.5b.**

Figure 5.2.2.5b: Differences between suggested and actual direction path.

In order to solve the above problem, the Predictor has incorporated a component called the Radar. It gives an alternative way for users to get to the bus stops. It utilizes the phone's built-in G-sensor compass, users can get their heading direction as if using real radar, i.e. users just need to walk to the direction the marker shown on the screen. **Figure 5.2.2.5c** illustrates an example situation. Using the radar as alternative has two main advantages:

1. No extra distance has to be travelled as following the driving directions.
2. Helpful to poor map readers as they would know which direction they should head to while that information are not provided on the map.

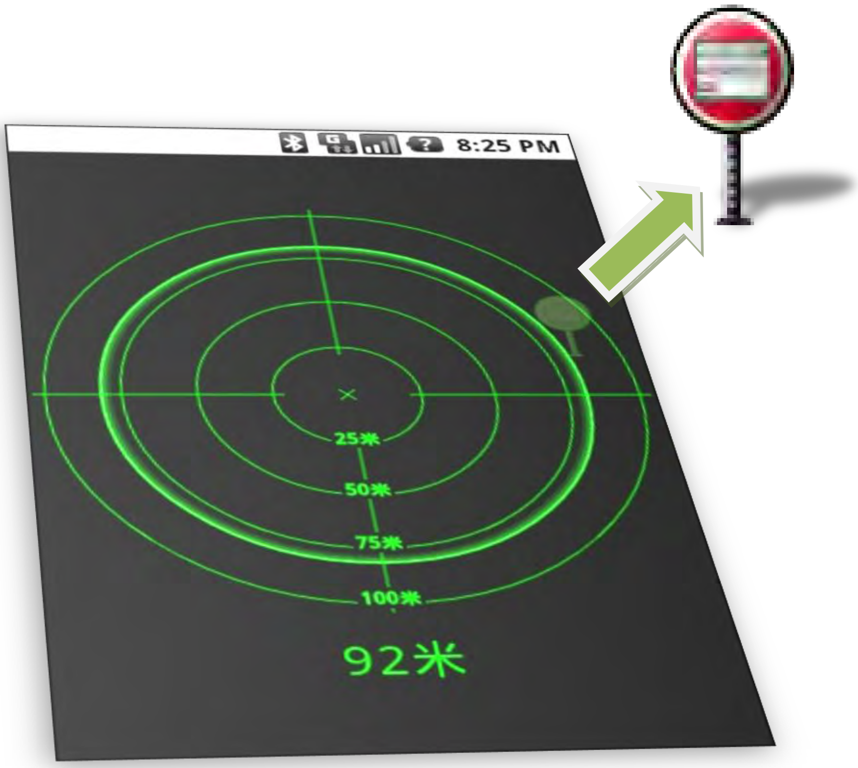


Figure 5.2.2.5c: Example situation when using the radar.

5.2.2.6 Get Off Notification Handling

One of the main features of the Predictor is that it provides a get off notification for users while timing their journey. The timer ran as a background job, which we called a **service**. It is essential to run at the background as users can continue to start and run other applications on the screen, i.e. do not hold and lock the user interface.

After starting the timer, users can even quit the Predictor and do whatever they want, such as starting the media player to listen to their favorite songs or web browser to go web surfing during their trip.

Once the phone has detected the users were getting near to (within a distance chosen by users in *Settings*, 30m – 100m) their destination, the Predictor would notify the **NotificationManager**, which is a member of the application framework in the platform, through a broadcast message to trigger the get off notification. **Figure 5.2.2.6a and 5.2.2.6b** show the screens of the notification is triggered when the user is browsing the web and listening to a song. The notification would consist of the followings to get the users' attention:

1. a short message on the screen telling the name of the next bus top (their destination)
2. a short message on the status bar with a little bus icon
3. a sound chosen by the users
4. two short vibrations

A summary window would be popped up showing the time taken for the trip when the users has tapped on the message in the pull down list of the status bar. See section 5.1.4.1, step 7.

Background jobs handling can be done in a convenient way in the platform of Google Android™, with background processes developers can have greater flexibility on designing the application.

Figure 5.2.2.6a: Notification is triggered.

Figure 5.2.2.6b: Screen when status bar is pulled down.

5.2.2.7 Communication Structure Handling

To facilitate the consistence of the bus route data between users, all of the bus route details were stored on a remote database. Users can get access to the database data through the application server.

For the data transmission between the client and the server, JSON was chosen to be the data structure. JSON is well known for data serializing and unserializing. Firstly, unlike traditional XML, JSON does not need to establish a known document type definition (DTD) between senders and recipients; which each DTD must contain a larger number of extra padding and means larger XML files. Secondly, JSON structure echoes the standard structure of programming data types, which means when encoding the data, there would be only a limit amount of characters added to define the structure and the value of data. Therefore, the speed of encoding mechanism is fast and data is compact enough to reduce the size of JSON strings transmitted.

Since this is a mobile application, the smaller the amount of the data transmitted, the better the responsiveness of the application would be and the carrier would charge lesser money on the transmission fee. The following shows a sample JSONArray string returned from server.


```
returnStr: [
{"airConFare":4.1,"busStopId":1,"normalFare":3.2,"direction":"F","busStopArea":"CHEUNG CHING/CHEUNG
HONG/MAYFAIR GARDEN","busStopName":"CHEUNG HONG BUS TERMINUS","routeId":"43"},
{"airConFare":4.1,"busStopId":2,"normalFare":3.2,"direction":"F","busStopArea":"CHEUNG CHING/CHEUNG
HONG/MAYFAIR GARDEN","busStopName":"CHING SHING COURT","routeId":"43"},
{"airConFare":4.1,"busStopId":3,"normalFare":3.2,"direction":"F","busStopArea":"CHEUNG CHING/CHEUNG
HONG/MAYFAIR GARDEN","busStopName":"CHEUNG CHING BUS TERMINUS","routeId":"43"},
...
{"airConFare":4.1,"busStopId":17,"normalFare":3.2,"direction":"F","busStopArea":"TAI WO HAU/KWAI
CHUNG ESTATE","busStopName":"FU ON HOUSE TAI WO HAU ESTATE","routeId":"43"},
{"airConFare":4.1,"busStopId":18,"normalFare":3.2,"direction":"F","busStopArea":"TSUEN WAN
CENTRE","busStopName":"TAI WO HAU FACTORY BUILDING","routeId":"43"},
{"airConFare":4.1,"busStopId":19,"normalFare":3.2,"direction":"F","busStopArea":"TSUEN WAN
CENTRE","busStopName":"BO SHEK MANSION","routeId":"43"},
{"airConFare":4.1,"busStopId":20,"normalFare":3.2,"direction":"F","busStopArea":"TSUEN WAN
CENTRE","busStopName":"LADY TRENCH POLYCLINIC","routeId":"43"},
{"airConFare":0,"busStopId":21,"normalFare":0,"direction":"F","busStopArea":"TSUEN WAN
CENTRE","busStopName":"TSUEN WAN WEST RAILWAY STATION BUS TERMINUS","routeId":"43"}
]
```

5.3 Testing and Results

Testing with physical phone is not possible at the current stage, as HTC G1™ is still not available in Hong Kong yet. It was considered as a limitation of this project as discussed in section 6.2.3. Thus, all tests conducted were based on the emulator to test for the feasibility of functions.

Positioning accuracy and impact on battery life cannot be tested without physical phone. Also, bus arrival prediction time accuracy cannot be tested as all predictions are based on assumption and simulation data to illustrate the feature only.

Cases	Description	Results
1	Test for different map modes, zooming and clear map..	3 map modes are set successfully. Zooming and clear map are performed successfully.
2	Test for showing pictures taken on the map.	1. Photos are shown on the map with different location. 2. Photos locations can be retrieved back from photos details. 3. Directions are provided through Google directions and is shown correctly to guide users back to where the photo was taken.
3	Test for searching bus routes with route number.	1. Routes are shown correctly with different directions provided. 2. Routes are shown correctly with circular routes. 3. Route details are shown correctly, including current time slot, fares and schedules.
4	Test for searching bus routes with origin and destination.	1. Routes are shown correctly with different directions provided. 2. Routes are shown correctly with circular routes. 3. Route details are shown correctly, including current time slot, fares and schedules.

		4. Markers are shown at correct stops that indicate the stops for get on and get off.
5	Test for searching nearby bus routes.	1. Routes are shown correctly with different directions provided. 2. Routes are shown correctly with circular routes. 3. Route details are shown correctly, including current time slot, fares and schedules. 4. Markers are shown at correct stops that indicate the stops for get on and get off. 5. Closet bus stops are shown on the map with map markers within the screen. 6. Closet bus stops contain correct bus route information to reach destination. 7. Distance between users and bus stops are calculated in meters and shown correctly. 8. Directions are provided through Google directions and is shown correctly to guide users to get to the bus stops. 9. Radar can be activated. Distance is shown on the radar and bus stop icon is moved accordingly on the screen to indication the direction to it.
6	Test for setting preferences.	1. Language preferences can be set accordingly, results returned from servers are shown with the language

		selected. But interface language does not change unless the application is recompiled. 2. 2 different reachable distance can be detected by nearby bus route search accordingly. 3. 8 different get off distance can be detected by the proximity alert to activate the get off notification successfully.
7	Test for taking photos	1. Photos can be captured from web cam server successfully. 2. Photos can be saved successfully into SD card and within the same album called “Bus Arrival Predictor” 3. Location details can be saved into the photo. 4. Picture name, title, bus route taken and description can be modified and saved successfully. 5. Photos taken can be viewed correctly in the Gallery.
8	Test for setting timer.	1. Bus routes can be searched successfully. 2. Starting bus stop can be selected and set correctly. 3. Stops after “starting bus stop” were shown successfully. Ending bus stop can be selected and set correctly. 4. Timer can be set and cancelled successfully. 5. Get Off Notification was activated successfully when user was within the get off distance set in the

		Settings. 6. Get Off Notification can be activated successfully when Bus Arrival Predictor was quit. 7. Get Off Notification can be activated successfully when users was playing music with media player and browsing web pages with web browser.

6 Evaluation

In this section, several challenges were discussed and solutions were introduced to overcome them. Besides challenges, some of limitations of this project were described here.

6.1 Challenges

6.1.1 No walking direction provided

Determining the direction to a location in the platform of Google Android^{TM} is all handled by the Google Maps^{TM} embedded in the phone; developers cannot get access to the core APIs. And for now, only driving directions are provided, walking direction is still unavailable. To this Predictor, providing driving direction to users is sometimes meaningless as discussed in section 5.2.2.5. Users walk on the streets but not on the roads.

To deal with the problem, a Radar application was introduced so that an intuitive way can be provided to users to know their direction. This is helpful to poor map readers as normally when people look at a map, they seldom know the orientation of the map so that it maps to the physical environment. With the help of radar, users would probably figure out the direction towards their bus stops in no time.

6.1.2 Camera simulation

In the SDK of Google Android^{TM}, it was stated that there would be no support for external camera on the emulator. To illustrate the functions in the Camera tab of this application, a web camera server was setup in order to simulate the present of a camera on the phone.

With the web cam server running, real time still images can be retrieved from the web camera as if a real camera is taking photos. Web cam server settings were included in the Settings, server IP and port can be altered so that different server connections can be made. **Figure 6.1.2a** shows the screens of web cam server settings.

Figure 6.1.2a: Users can set the server IP and port.

6.1.3 Location simulation

Locating the user's current location was a core part in this application, as having the current location; the application can determine the distance between users and bus stops and the direction to the bus stops.

In order to simulate the user's current location on the emulator, location files are needed, in this application "*.GPX" files were used to store the mock location points, those testing points are all retrieved from the application Google Earth^{TM}. By pinpointing a list of locations on the Earth^{TM}, it can generate those points as a track or waypoints in GPX or KML format. Passing the GPX files into the Dalvik Debug Monitor Service (DDMS), a debugging tool, location-aware operations can be performed.

6.1.4 Chinese Language Support

On the current version of the platform, only German, English (UK) and English (US) can be set as system locale; which means, there is no support for Chinese as the system language and region. Although setting locale is not possible, Android™ do support for alternate resources (for alternate language and configuration) using the following file structure.

```
MyApp/
        res/
                values-en/
                        strings.xml   //Strings for English locale setting (a)
                values/
                        strings.xml   //Strings for default setting (b)
```

Resources like strings, themes, colors and arrays in Android™ are all in XML format, with UTF-8 encoding, Chinese characters can be entered so that they were displayed on the user interface. The system can automatically determine which "strings.xml" have to be used according to the locale setting. If all Chinese characters strings were entered in (a) above instead of English characters, then whenever the system detect the locale settings, (a) is always chosen as default (since English (US) is set as locale) and so Chinese interface language was used as shown from previous system screenshots.

6.2 Limitations

6.2.1 No real time bus location data

As mentioned, the bus arrival predictions were based on an assumption set of data, which are inaccurate to predict the arrival of buses as they are not real AVL or APC data. No such data can be obtained from any of the bus companies in Hong Kong as those data were not opened to public or actually, there were no GPS devices installed on every bus to obtain those location data.

6.2.2 Data collection

Data of bus route information consist of bus route number, bus stop number, bus stop name, bus stop image, bus stop area, district, schedules, fares and coordinates of each bus stop. Since there is no data provider to serve the bundle of data, each type of the data had to be input into the database manually. Names, fares, schedules and images were all collected from Kowloon Motor Bus Company Limited web site with both the English and the Chinese set. For bus stops positions, each of them was gathered from pinpointing on the Google Earth™ and was double-checked with the positions on Centamap.

The data collection process was therefore time-consuming. Thus, it is not possible to include all the bus routes available in Hong Kong at this stage. Consequently, here were only limited amount of bus route data for demonstrating the application.

6.2.3 No physical phone available

Google Android™ was made to be run on any Linux based mobile phones on the market. The first phone available was the HTC G1™, which was available in 18 international countries, but Hong Kong is still not on the list. The second generation of the Android phone called HTC Magic™, claimed to launch in Asia this April. Therefore, there is still no physical phone available for a real testing to be carried out. Impact on the battery life, GPS receiving accuracy and responsiveness on real 3G or EDGE network cannot be tested until the launch of the phone in Hong Kong.

7 Conclusion

7.1 Critical Review

Throughout this year, I have come across many challenges that I have never met before. This was my first time that I have to be in charge of the whole project, and this was a yearlong project. In addition, it was my first time to deal with programming on mobile phones. Thanks to the one-year placement, I have learnt a lot on Java programming, which was really valuable and beneficial to my final year project. At the beginning, I was new to mobile application programming. Actually, I have to spent massive time on studying the specifications of the platform and trial running the tutorials and sample codes before proposing my project idea to my supervisor. However, those spent time were worthwhile.

Technically, the challenge would the platform of Google Android™. This was a newly developed operating system for mobile phones first announced since November 2007. The challenge was that the platform is evolving (the latest version is beta version 1.5 SDK released on 13th April) and APIs are changing throughout the time. When there was new version published, some of the methods would be no longer usable or accessible. I have to keep my coding up-to-date so as to ensure my application does not have any compilation errors. Moreover, as this is a fresh platform, there were only very limited books, articles or even comprehensive APIs talking about this. Therefore, when I was stuck on some methods, surfing forums and discussion groups would be my best choice. Although that was time-consuming, I found it satisfied when the problem was solved.

Personally, I have to be more self-discipline as besides completing my final year project; I have also courses to take in the semesters. Time management was the most precious thing I have learnt in this project. What’s more, friendship among classmates can be enhanced as we have spent more time together in the project laboratory or outside class discussing the projects. All in all, the final year project has given me a lot of memorable moments that I will treasure them for my life time.

7.2 Achievements

This project has introduced an innovative way to incorporate location detection with bus routes searching and arrival prediction. In this section, several major achievements were discussed.

7.2.1 Nearby bus stops searching

The searching for nearby bus stops has enhancing the traditional bus route searching approaches. Not only bus routes information are provided, bus also the bus stops locations are presented on an interactive map interface with directions supplied.

7.2.2 Travel recorder

When travelling, users can take pictures within the application. Their memories can be kept in the photo taken, together with words of description. To recall memories, the application can guide the users back to where the photos were taken with the built-in direction feature.

7.2.3 Get off notification

Not all of the buses in Hong Kong have installed the Bus Stop Announcement System or similar as mentioned in section 2.1.2. Users can set their own get off notification so that when they are near to the destination, the phone would ring and vibrate to inform the user to be ready to get off the bus. This feature is also suitable for those who would sleep, read books or play electronic games on buses so that they would not miss the bus stops.

7.2.4 Extensible 3-tier structure

The 3-tier system architecture is so flexible that many more of the bus routes can be entered into the database without changing the interface of the Predictor and business logic in application server so that a more comprehensive route searching can be done.

7.3 Future Extension

As mentioned, there were still limitations in this application. A main purpose to develop this application was to inspire future extension on the same field so that more real time passenger information can be provided.

7.3.1 Implement the real time location detection of buses

One of the main limitations of this application is that, there is no real time location information on buses. That is, no accurate bus arrival prediction can be provided. If in the future, bus companies can provide their vehicles location data to public, then real time bus arrival prediction could be done with ease.

7.3.2 Implement interchange information

As schedule is tight, only trips information without interchange are provided and implemented. However, it is always true that sometime people would take more than one bus routes to travel from one point to another point. Implementing interchange information should be done to fulfill users' expectations.

7.3.3 Implement caching mechanism

As this is mobile application, all of the transmitted data through carrier's network would cost money. To reduce the cost, a caching mechanism should be developed so that large files such as bus stop images could be cached into phone's local memory on the first time, so that the same image would not be downloaded again.

8 Cited References

Chadil, N., Russameesawang, A., and Keeratiwintakorn, P. (2008). Real-time tracking management system using GPS, GPRS and Google earth. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008* (pp. 393-396). Krabi: King Mongkut's University of Technology.

Chau, S. Y. (2002). The Value of Real Time Information at Bus Stop in Hong Kong. *University of Hong Kong*. Available: http://sunzi.lib.hku.hk/hkuto/record/B3194565X [2008, November 03].

Chen, M., Liu, X., Xia, J., and Chien, S. I. (2004). A dynamic bus-arrival time prediction model based on APC data. *Computer-Aided Civil and Infrastructure Engineering*, 19(5), 364-376.

Citybus Co. Limited. (2007, August 01) Cityflyer Route. *Citybus Co. Limited.* Available: http://www.nwstbus.com.hk/_common/images/photo/A10_SMS_poster.pdf [2009, April 01]

Collins, C., Grude, A., Scholl, M., and Thompson, R. (2007). txt bus: wait time information on demand. In *Conference on Human Factors in Computing Systems* (pp. 2049-2054). New York: ACM.

Dessouky, M., Hall, R., Nowroozi, A., and Mourikas, K. (1999). Bus dispatching at timed transfer transit stations using bus tracking technology. *Transportation Research Part C: Emerging Technologies,* 7(4), 187-208.

Google. (2008). Android - An Open Handset Alliance Project. *Google.* Available: http://code.google.com/android/what-is-android.html [2008, October 26].

Google. (2008). Maps Help. *Google.* Available: http://local.google.com/support/bin/answer.py?hl=en&answer=7060 [2008, October 26].

Hong Kong Economic & Trade Office (Canada). (2008, September). Hong Kong Fast Facts. *Hong Kong Economic & Trade Office (Canada).* Available: http://www.hketo.ca/abouthk/figures.html [2008, October 10].

Hong Kong Government. (2007). Hong Kong Yearbook 2007. *Hong Kong Government*. Available: http://www.yearbook.gov.hk/2007/en/pdf/E13.pdf [2008, October 09].

HTC. (2008). HTC – Touch Phone, PDA Phone, Smartphone, Mobile Computer: Products – T-Mobile G1: Overview. *HTC.* Available: http://www.htc.com/www/product/g1/overview.html [2008, October 26].

Jeong, R. H. (2004, December). The Prediction of Bus Arrival Time Using Automatic Vehicle Location Systems Data. *Texas A&M University* . Available: https://txspace.tamu.edu/bitstream/handle/1969.1/1458/etd-tamu-2004C-CVEN-Jeong.pdf;jsessionid=63296EB16250B5A00B9BC6451D50B812?sequence=1 [2008, November 01].

Karunanayake, M._D., Cannon, M._E., and Lachapelle, G. (2007). Analysis of assistance data on AGPS performance. *Measurement Science and Technology,* 18(7), 1908-1916.

Kowloon Motor Bus Co. Limited. (2009, March 29). NewsRoom – Press Release. Kowloon Motor Bus Co. Limited. Available: http://www.kmb.hk/english.php?page=next/news/service&file=news1q09/news20090325.html [2009, April 02].

Kole, A. (2007, May 01). Real Time Information. *Greater Wellington Regional Council*. Available: http://www.gw.govt.nz/council-reports/pdfs%5Creportdocs%5C2007_280_1_Report.pdf [2008, October 15].

Lin, W. H., and Zeng, J. *An Experimental Study On Real-Time Bus Arrival Time Prediction With GPS Data*. Center for Transportation Research and Department of Civil and Environmental Engineering. Virginia Polytechnic Institute and State University.

Maclean, S. D., and Dailey, D. J. (2001). Real-time Bus Information on Mobile Devices. In *2001 IEEE Intelligent Transportation Systems Proceedings* (pp. 988-993). Oakland: ITSC, IEEE.

McLeod, F. (2007). Estimating bus passenger waiting times from incomplete bus arrivals data. *Journal of the Operational Research Society*, 58(11), 1518-1525.

MiniBus. (2009). Hong Kong, China: *Hutchison Telephone Company Limited © 3HK & Mobilesoft*

Mishalani, R. G., Lee, S., and McCord, M. R. (2000). *Evaluation of Real-Time Bus Arrival Information Systems*. Transportation Research Board Annual Meeting. Washington D.C.

Porcino, D. (2001). Location of Third Generation Mobile Devices: A Comparison Between Terrestrial and Satellite Positioning Systems. In *IEEE Vehicular Technology Conference* (pp. 2970-2974). Rhodes: Institute of Electrical and Electronics Engineers Inc.

Shalaby, A., Lyon, C., and Sayed, T. (2001). Transferability of travel time models and provision of real-time arrival time information. In *2001 IEEE Intelligent Transportation Systems Proceedings* (pp. 982-987). Oakland: ITSC, IEEE.

Studio KUMA BusInfo software. (2008). Hong Kong, China: *Studio KUMA*.

Transportation Research Board. (2003). *Real-Time Bus Arrival Information Systems: A Synthesis of Transit Practice*. United States of America: Transportation Research Board.

Wall, Z., and Dailey, D.J. (1999). An Algorithm for Prcdicting the Arrival Time of Mass Transit Vehicles Using Automatic Vehicle Location Data. Transportation Research Board Annual Meeting. Washington D.C.

Welch, G., and Bishop, G. (2007, March 08). An Introduction to the Kalman Filter. *University of North Carolina at Chapel Hill*. Available: http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf [2008, October 30].

9 Appendices

9.1 Monthly Logs

Project Month	Progress Log
Oct, 2008	Works Achieved: -Basic Android API Control I. Layout II. Button III. Tabs IV. Intents -Basic MapView API Control I. Apply a Google Map API Key II. Zoom in, zoom out III. Change map mode IV. Handle Button events V. Get current location VI. Change Current location VII. Import/ Mock GPS Data -GPX -KML -Google Map to GPX -Mock a route -UI Construction Cycle -Debug Skill I. Logcat II. DDMS tool -Mock SD Card I. media II. photo III. songs -Basic Local Database Control I. SQLite (Manager) -Database Create -Basic Query and Update -push/pull data -Set local current Time of the device Works under studying: -Learning the difference between versions of the new and old SDK -MapView API I. Add overlay to the map view II. Show info./picture on the overlay III. Show a complete bus route on map IV. calculate distance between destinations

	-Database I. Schema design II. Remote database connection and data retrieval III. Design simulation data -Studying different bus arrival prediction algorithms I. AVL II. APC III. Real Time VS non Real Time Data IV. Reading on relevant Journals and Papers for reference -Android API I. Driving direction II. Compass mode III. Radar Method VS Straight Line Path Estimation

Nov, 2008	Works Achieved: -Study on different prediction algorithms: I. AVL II. APC III. Real Time VS non Real Time Data IV. Reading on relevant Journals and Papers for reference -Completed Final Year Project Interim Report -Figure out solution to compensate driving direction with compass mode -Database I. Schema design Works under studying: -Learning the difference between versions of the new and old SDK -MapView API I. Add overlay to the map view II. Show info./picture on the overlay III. Show a complete bus route on map IV. calculate distance between destinations -Database I. Remote database connection and data retrieval II. Design simulation data -Android API I. Radar Method VS Straight Line Path Estimation

Dec, 2008	Works Achieved: -Database I. Schema design II. Remote Database Setup for assumption set of arrival data III. Data Retrieval -Web Server I. Setup II. Connection with database III. Communication test with Android -Android API I. Radar Mode Invocation Works under studying: -Learning the difference between versions of the new and old SDK -MapView API I. Add overlay to the map view II. Show info./picture on the overlay III. Show a complete bus route on map IV. calculate distance between destinations -Database I. Design simulation data -Android API I. Straight Line Path Estimation

Jan, 2009	Works Achieved: -Database I. Schema design II. Remote Database Setup for assumption set of arrival data III. Data Retrieval -Web Server I. Bus Stops Image Retrieval from server II. Tested the setup with web cam server -Android API I. Straight Line Path Estimation II. Camera Api testing III. Image retrieval from web cam server to simulate camera capturing process I V. Bus routes searching V. Bus routes list information querying Works under studying: -Learning the difference between versions of the new and old SDK

	-MapView API I. Add overlay to the map view II. Show info./picture on the overlay III. Show a complete bus route on map IV. calculate distance between destinations V. show the directions to the bus stops -Database I. Design simulation data -Android API I. Customized layout II. Customized list adapter

Feb, 2009	Works Achieved: -Database I. Retrieve the schedule data II. Retrieve the route details III. Design simulation data -Android API I. Camera Implementation II. Bus routes searching based on the user current location (within the user's reachable range) III. Added the indicators to users about which bus stops to get on and off so that to reach their destination IV. Added the detection on the ON/OFF status of the GPS V. Storing and retrieving photos(with location data) on device's SD card VI. Implementation of Tabs with selector API -MapView API I. calculate distance between destinations II. show the directions to the bus stops Works under studying: -Learning the difference between versions of the new r1.1 and old SDK -MapView API I. Add overlay to the map view II. Show info./picture on the overlay III. Show all potential bus stops -Android API I. Time arrival precise estimation

Mar, 2009	Works Achieved: -Implemented the Traditional Chinese Version -Database I. Timer Data -Android API I. Added Timer for timing travel (Chronometer) II. Added alert notifications for arriving your destination III. Camera Implementation with both web cam server and server images approaches IV. Added Settings for languge, reachable distance and web cam preferences V. Added About of this application -MapView API I. Add overlay to the map view II. Show info./picture on the overlay III. Show all potential bus stops Finished Development and Writing Final Report.