MOBILE BASED ROUTE PLANNER FOR KLANG VALLEY RAIL TRANSPORTAION

MATTHEW GOH WEE HIEN

A project proposal submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering

Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

August 2013

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name :

ID No. :

Date :

APPROVAL FOR SUBMISSION

I certify that this project proposal entitled **“Mobile Based Route Planner For Klang Valley Rail Transportation”** was prepared by Matthew Goh Wee Hien has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons.) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ___________________________________

Supervisor : Ooi Ean Huat

Date : ___________________________________

Specially dedicated to
my beloved family and those who love me….

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deepest gratitude for those who have helped me throughout the completion of the project. First of all I would like to thank my project supervisor Mr James Ooi Ean Huat for his continuous guidance, support and encouragement given throughout the project execution.

Other than that, I would like to thank my project moderator, Mr Madhavan for his advice and guidance to improve the project.

Last but not least, I would also like to thank all my family and friends that have supported me in this project.

ABSTRACT

This document describes and records the all the processes involved during the development of "**Mobile Based Route Planner for Klang Valley Rail Transportation**". The target audience for this project is citizens that are not familiar with the rail system or backpackers that comes from foreign countries. The purpose of developing such application is to overcome the problem faced by the commuters when travelling using the current rail system. There is a lack of integration between the existing rail systems, although some route planner application is available in the market, but they are not intensive enough and does not offer the required information to the user.

This project aims to deliver an application that will provide correct transit guidance for the commuters. Other than that, the application would be able to provide all necessary transit information such as fare, estimated travel time, station information, nearby attractions and service status to the commuter.

The project is executed by adhering to the best practice of Software Engineering and Software Project Management. Each phase of the project is executed in sequence and documented clearly.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
IDC	International Data Corporation
KL	Kuala Lumpur
OS	Operating System
PHP	Hypertext Preprocessor
UI	User Interface
XML	Extensible Mark Up Language

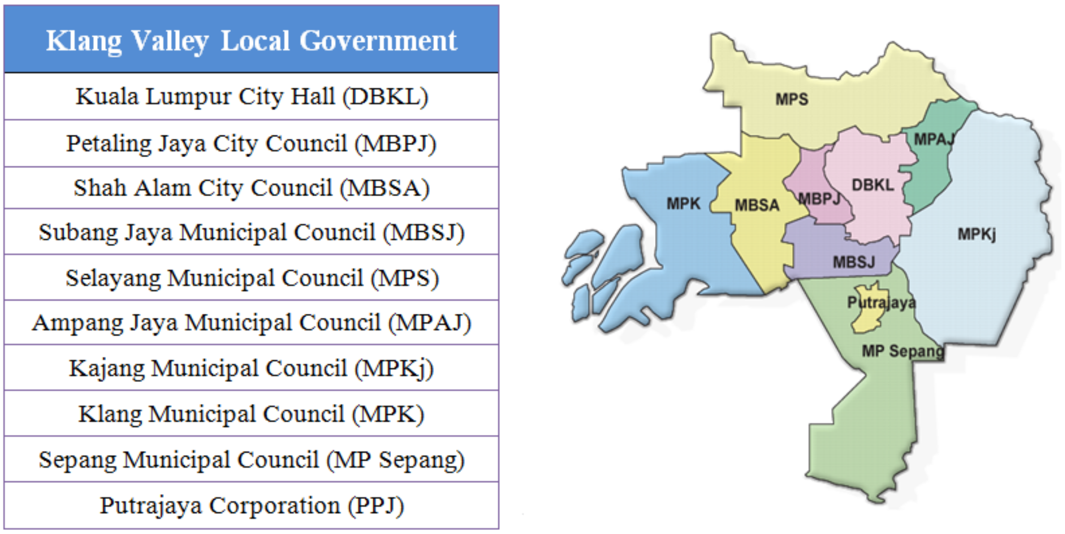
LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Klang Valley

Klang Valley is a rapid growing metropolitan area made up by Kuala Lumpur and several adjoining cities and towns in the state of Selangor. Although no official boundaries are drawn, but it is understood that Klang Valley is an area of approximately 2793km^2 and comprised of 10 municipalities, each with own local authorities.



Klang Valley Local Government
Kuala Lumpur City Hall (DBKL)
Petaling Jaya City Council (MBPJ)
Shah Alam City Council (MBSA)
Subang Jaya Municipal Council (MBSJ)
Selayang Municipal Council (MPS)
Ampang Jaya Municipal Council (MPAJ)
Kajang Municipal Council (MPKj)
Klang Municipal Council (MPK)
Sepang Municipal Council (MP Sepang)
Putrajaya Corporation (PPJ)

Figure 1.1: Klang Valley Local Authorities & Map

Klang Valley is the most important economic, political, education and cultural center for Malaysia and is home to approximately 7.2 million people (The

Star, 2013). In December 2010, Prime Minister Dato’ Seri Najib Tun Razak announced the development blueprint for Greater Klang Valley that aims to transform Klang Valley into top 20 most liveable city in the world and top 20 in economic growth.

1.1.1 Klang Valley Current Rail Transportation System

For a metropolitan with dense population, there must be an effective public transportation system to mobilize millions of commuters from their home to workplace and vice versa. The survival and economic progress of a metropolitan like Klang Valley is tightly coupled to a complex mechanism of public transport that keeps the city functioning at normal pace each day. A glitch on the system will cause a tremendous financial impact. To date, Klang Valley has 8 dedicated rail lines running in parallel to serve the citizens.

Table 1.1: Current Active & Serving Rail Line in Klang Valley

Active & Serving Line		
Operator	**Line**	**Number of Stations**
rapidKL	Kelana Jaya	23
	Sentul Timur – Sri Petaling	18
	Sentul Timur – Ampang	18
	KL Monorail	11
KTM KOMUTER	KTM Batu Caves – Port Klang	27
	KTM Tanjung Malim – Sungai Gadut	30
EXPRESS RAIL LINK	KLIA Ekspres	2
	KLIA Transit	5
	Total	**134**

1.1.2 Klang Valley Rail Expansion

In conjunction to realize the vision of Greater Klang Valley, the government launched the Klang Valley MRT Project which aims to add 3 MRT lines into the current rail transport landscape. In December 2010, the government approved the first 51KM MRT line which span from Sungai Buloh to Kajang with a total of 31 stations (MRTCorp, n.d.). The construction of the first MRT line was officially launched on 8th July 2011 and is expected to be complete on year 2017. Details of the other 2 lines are currently under studies and will be implemented soon. Other than that, works on extending existing Kelana Jaya line and Sentul Timur – Sri Petaling line is on-going and is expected to complete in year 2014 (RapidKL, n.d.).

Table 1.2: Current Active & Confirmed Rail Lines

Active & Confirmed Future Line			
Operator	**Line**	**No. of Stations**	**Status**
rapidKL	Kelana Jaya	23 + 13	23 Stations Active 13 Under Construction
	Sentul Timur – Sri Petaling	18 + 13	18 Stations Active 13 Under Construction
	Sentul Timur – Ampang	18	All Active
	KL Monorail	11	All Active
KTM KOMUTER	KTM Batu Caves – Port Klang	27	All Active
	KTM Tanjung Malim – Sungai Gadut	30	All Active
EXPRESS RAIL LINK	KLIA Ekspres	2	All Active
	KLIA Transit	5	All Active
	Sungai Buloh – Kajang	31	Under

			Construction
	MRT Line 2	22	Planning
	MRT Line 3	24	Planning

1.1.3 Klang Valley Future Integrated Rail Map

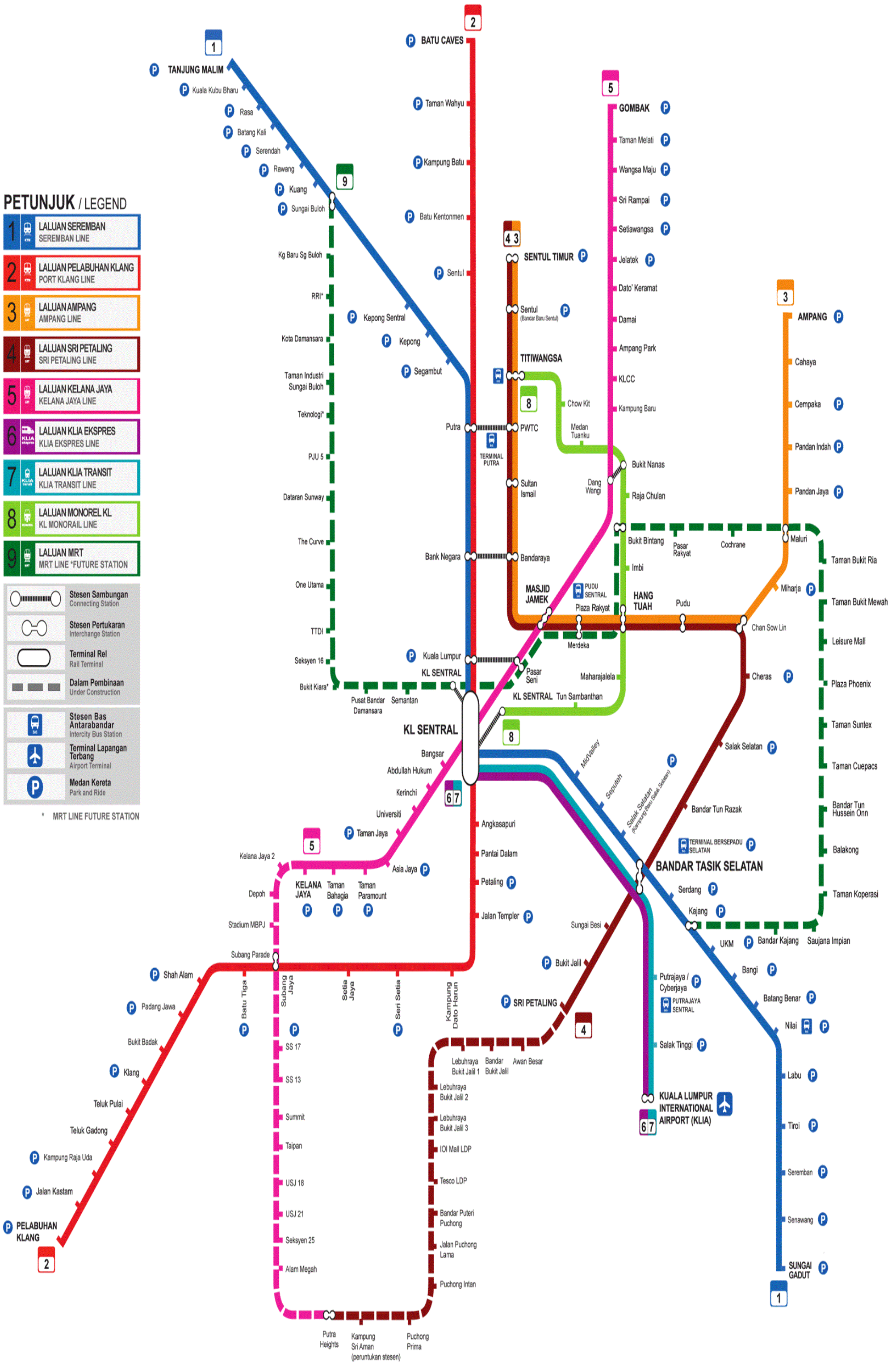


Figure 1.2: Integrated Rail Map For Klang Valley (SPAD, 2012)

1.2 Problem Statement

Despite the growing sophistication of rail network in Klang Valley. It lacks an effective mechanism to convey information about various rail routes to the commuters. Travelling from point A to point B by using rail is a dilemma especially for those who are not familiar with the system. When confirmed rail expansion is completed by year 2017, the total number of interchange node would be up to 17 and at that point of time, even the most frequent user of rail service would not be able to memorize all the interchange.

Existing route planner is available in market, but they are not comprehensive enough because they are riding on the existing web service provided by RapidKL and only display routes operated by RapidKL without integration with other rail operators. Other than that, the User Interface of existing mobile applications is poorly designed and not user friendly. Thus, Klang Valley is in dire to have an effective route planner for its growing sophisticated rail system. Such application would not only effectively solve the above mentioned problem, but in addition, to promote and encourage the use of rail system.

The idea to develop such application on mobile platform is due to the fact that smartphones are getting more common in Malaysia and the application can be easily accessed by users without the need of having an active data connection (except for Live Service Updates).

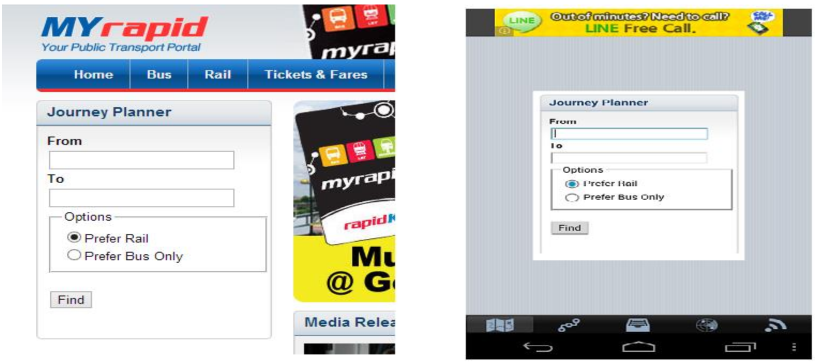


Figure 1.3: RapidKL Portal vs Existing Route Planner Application

1.3 Project Objectives

1.3.1 General Objectives

1. To promote integration between all the rail lines with different operators and provide a common platform that contains information for all the different rail lines.
2. To provide an effective application to assist user to navigate through the complex rail network of Klang Valley and encourages the use of rail system by reducing the uncertainty faced by commuters when utilizing the system.

1.3.2 Specific Objectives

1. To develop a Mobile Based Rail Planning application that integrates all the confirmed rail lines in Klang Valley.
2. To provide a dedicated mobile based rail route planner without the need of having an active data connection (except Live Service Status).
3. To provide accurate transit guidance for commuters to reach their destination seamlessly.
4. To assist commuters on available route options so that they can plan their journey in advance.
5. To provide live service status to commuters so that they can opt for alternative route if glitches happen on their travelled route.

1.4 Project Scope

The project aims to deliver a mobile based transit planner that includes

- Route Planner
  - Able to display correct routes from Point A to Point B.

- Latest Klang Valley Rail Route Map
  - Zoom-able Rail Route Map which includes Sg Buloh – Kajang MRT and KJ & SP Line extension.

- Dedicated UI for Mobile based environment
  - User friendly UI by adhering to Android UI Guidelines

- Savable & Sharable Routes via Email/SMS.
  - Route result can be saved for future use. Route can be shared with friends/families via Email/SMS.

- Travel Fare
  - Shows the correct fare payable from origin to destination.

- Live Service Status
  - Shows the live operating service status from all the rail lines.

- Estimated Travel Time
  - Algorithm to estimate transit time from origin to destination.

- Station Information
  - To display station operating time, facilities available and nearby attractions.

1.5 Technology Used

- Development Tools
  1. Android SDK
  2. Apache Web Server
  3. SQLite Database

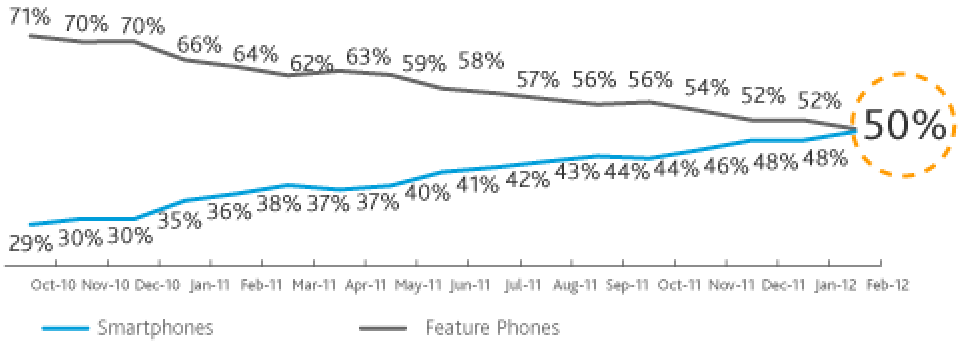
- Programming Language Used
  1. Java
  2. Java Script
  3. HTML

CHAPTER 2

LITERATURE REVIEW

2.1 Global Smartphones Market

Starting from year 2010, the world is experiencing a “Smartphone Boom” (Deutsche Welle, 2012) whereby smartphones are registering a strong growth compared to feature phone. According to research by a renowned global information and measurement firm Nielsen, smartphones penetration is growing steeply while sales of conventional phone ditches. In year 2013, the percentage of people owning smartphones are set to exceed feature phone.



Read as: During February 2012, 50 percent of US mobile subscribers owned a smartphone

Figure 2.1: Smartphones vs Feature Phones Penetration (Neilson, 2012)

Although Malaysia smartphones penetration rate (27% as of Q1 2012) is not as high as those in US (ITU, 2012), but studies by Nielsen has shown that smartphones market for Malaysia in year 2013 is very positive and is expected to expand further. Analysis from International Data Corporation's (IDC) forecasted that smartphones growth will remain strong until year 2017.

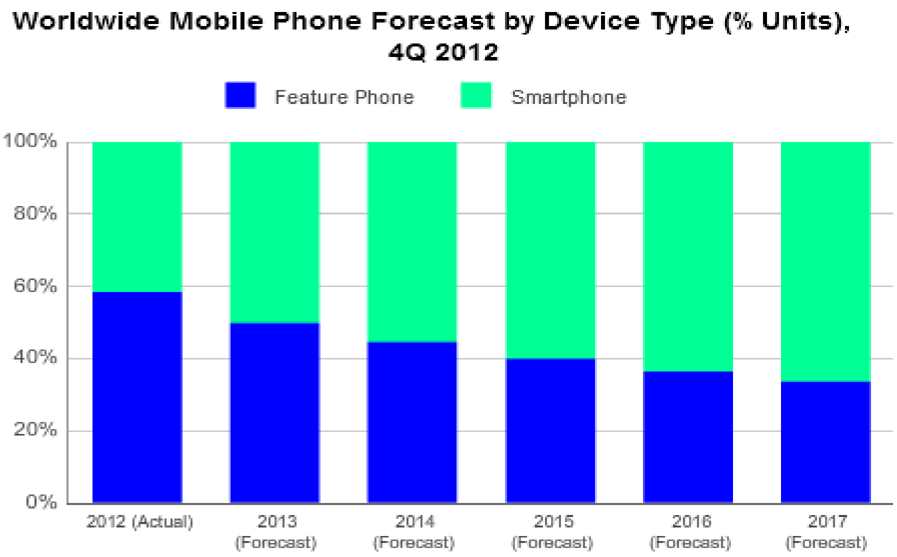


Figure 2.2: Smartphone Growth Forecast (IDC, 2012)

2.2 Mobile Operating System

One of the main selling points of a smartphones other than specification is its mobile operating system. Mobile operating system allows application to be developed under its platform to introduce additional functions. Currently there are 5 major mobile OS player in the world.

Table 2.1: Major Mobile Operating Systems in the World

OS Name	OS Developer	Latest Version	Licence	Development
Android	Google	6.1.3	Open Source	Active
iOS	Apple	4.2.2	Closed Source	Active
Windows Phone	Windows	8	Closed Source	Active
Blackberry	RIM (Research In Motion)	7.1.0.649	Closed Source	Active
Symbian	Nokia	10.1	Open Source	Ceased (2011)

According to studies by IDC, the world mobile operating system is currently dominated by Android and iOS which makes up a total of 84.90% out of the market share. Android is the most popular operating system which constitutes 68% of market share in Q2 2012. Comparison with data from Q2 2011 shown that Android registered the strongest market share growth of 21.2% while other operating system shows decline except Windows Phone.

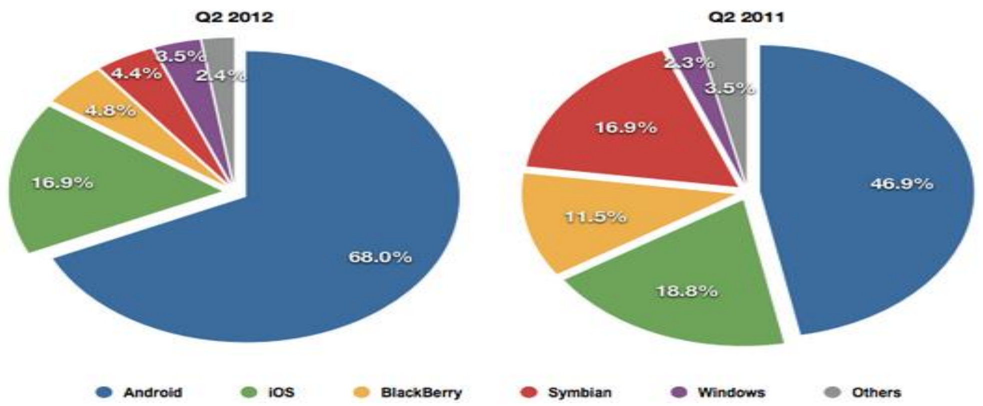


Figure 2.3: Mobile Operating System Market Share on Q2 2011 and Q2 2012 (IDC, 2012)

Research from renowned IT firm Gartner forecasted that Android future growth and outlook will remain strong and positive for many years to come.

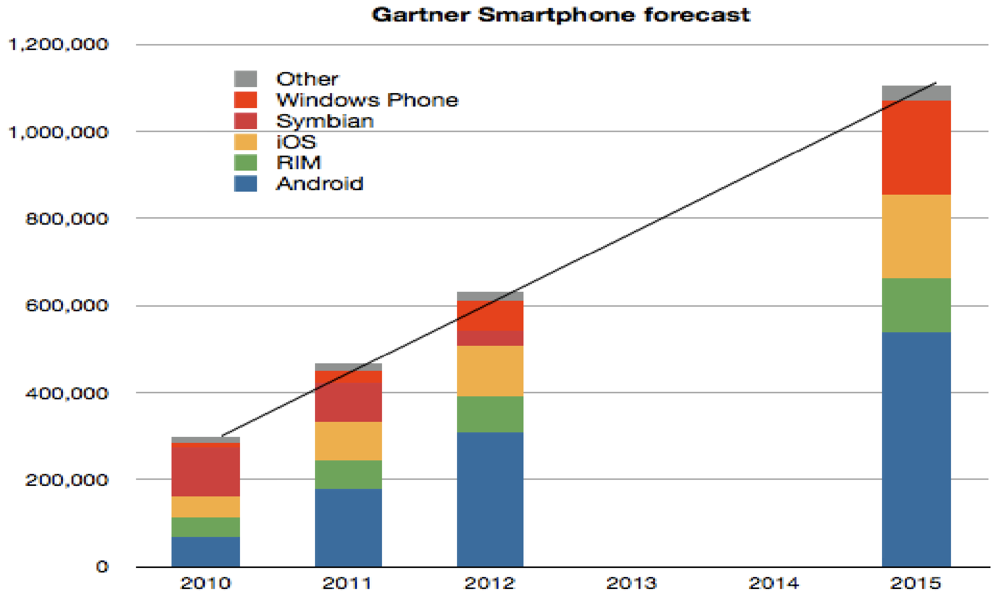


Figure 2.4: Mobile Operating System Forecast (Gartner, 2011)

2.2.1 Android vs iOS Development Cost

Development cost is a great concern for programmers and companies when deciding on development platform. Platform with lower development cost is often preferred due to lower start-up capital. Android and iOS are the most famous mobile operating system in the world. The comparison of development cost between both of them is drawn on the table below.

Table 2.2: Android vs iOS Development Cost

	Android	iOS
Developer	Google	Apple
SDK	Eclipse	XCode
SDK Platform	Any	MAC OS Only
Language	Java	Objective C
Runs on	Any Smartphones & Tablets	Only Apple Products
Developer Fees	RM75 One Time	RM297 / Year
Revenue Sharing	70:30	70:30
Application Store	Google Play	Apple Store

**Based on conversion rate of 1 USD to 3 MYR.*

Based on the comparison of the both, Android is preferred due to lower development cost. Programmers can start developing Android applications simply by downloading the SDK from Google website which bundles all the required libraries and tools. Other than that, Android SDK is platform friendly and is able to run smoothly on a wide range of operating systems. Whereas to develop iOS application, a Mac machine is needed because XCode can only run in MAC OS environment, this will incur additional cost to purchase a Mac machine which costs up to RM3500 per unit.

Application publication fees for Android are also cheaper because only a one-off payment of RM75 is collected and developers can start publishing their applications to Google Play. Meanwhile to develop on iOS platform, developers need to pay RM297 each year to publish their application to Apple Store.

2.3 Development Method

There are 3 ways of developing and delivering functionalities to smartphones each with cons and pros. Selection of development method between native application, web application and hybrid application is very subjective and depends greatly on a few constraints such as:

- Development Cost
  - The lowest investment over return ratio.

- Application Complexity
  - A quick application that only performs basic retrieving of data or complex application that performs complicated computations.

- Performance
  - Fast access needed or latency can be tolerated?

- User Interface Complexity
  - Rich user interface or simple?

- Available Human Capital
  - Man power available for development work and technical knowledge.

- Connectivity
  - Application needs constant connectivity or offline?

2.3.1 Native Application

Application is built and installed on a specific platform. To support both Android and iOS, two set of codes need to be written using the standard SDK provided by each vendor, thus leading to higher development cost (Ngu and Do, 2012). Native application developed using Android SDK cannot be ported or installed on iOS. The user interface of native application is powered by the native graphic rendering libraries which are optimally designed for the particular platform. This ensures a smooth rendering of 2D and 3D graphics. Other than that, the code written using native libraries are all optimized & compiled for the platform, thus processing efficiency and performance is guaranteed. A more diversified application can be created using native approach due to the full privilege granted for hardware access. Application can be more creative by introducing different combinations of available hardware such as GPS, Gyroscopes, Compass and etc.

2.3.2 Web Application

Web application must be accessed via web browser. Nothing will be installed on the device so an active data connection is needed. Web application can be written using any language (C#, Java, PHP, etc.) and can be displayed on different mobile operating system without rewriting the codes. The high portability reduces development cost. The user interface for web application is designed using HTML & Javascript thus web application cannot handle extensive 3D graphic processing (Ngu and Do, 2012). The performance of web application is slow and is greatly dependent on the connection speed. Web application has no access to application store thus programmers and companies can only advertise their application via websites. Other than that, web application often has limited functionalities due to the lack of access to device hardware. Only minimal hardware access is granted to web application such as GPS.

2.3.3 Hybrid Application

Offer the best elements from native and web application. One set of code is written using cross-platform tools and can be deployed into different mobile platform with minimal changes and so development cost is lower compared to native application. Some of the most famous open-sourced cross-platform development tools are shown in the table below.

Table 2.3: Most Famous Cross-platform IDE

IDE	Development Language	Developer
PhoneGap	HTML5, Javascript, CSS	Nitobi Software
Appcelerator Titanium	HTML5, Javascript	Appcelerator Inc
Rhodes	HTML5, Javascript, Ruby	Motorola
MoSync	C/C++, Javascript, HTML, CSS	MoSyncAB

Hybrid application is basically a native app with embedded HTML codes. HTML and Javascript are used to display the user interface, thus hybrid application cannot render 3D graphics smoothly. Hybrid applications are typically slower than native application because the codes are not optimized (Charland and LeRoux, 2011). Although hybrid application enjoys access to device hardware, but it is limited and the supported API is different across developers. The architecture of hybrid application using PhoneGap is shown in the figure below.

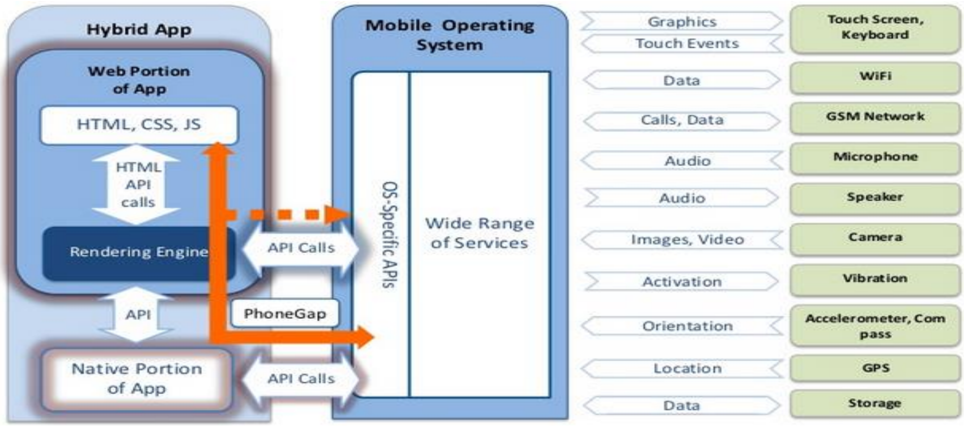


Figure 2.5: Hybrid Application Architecture

2.3.4 Comparison between Native, Web and Hybrid Application

The table below shows the pros and cons of native, web and hybrid application.

Table 2.4: Comparison between Different Development Method

	Native	Web	Hybrid
User Interface	Native Graphic Rendering API	HTML5 & Javascript	HTML5 & Javascript
Performance	Fast	Slow	Slow
Portability	Low	High	High
Distribution	Application Store	Website	Application Store
Device Hardware Access	High	Low	Moderate
Connectivity	Offline / Online	Online	Offline / Online
Development Cost	Highest	Low	Moderate

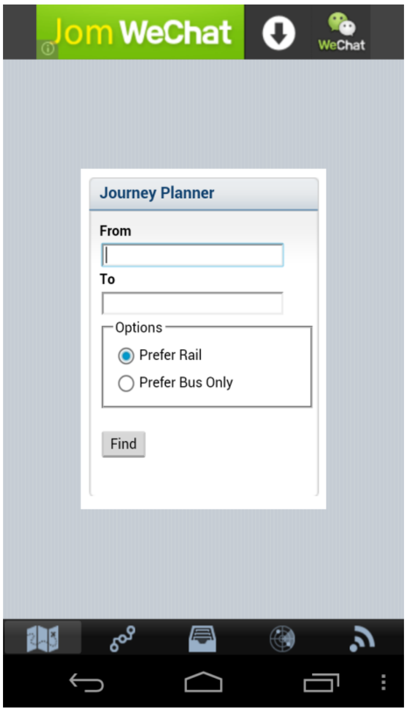
After evaluating 3 different development methods, native application development is still preferred for the implementation of this project over web application and hybrid application. The main evaluation criterion is the concern on performance. Complex data type and computations will be performed to find the shortest possible route between origin and destination, thus performance of the application is very important. Web application and hybrid application are slow in performance, thus more time will need to be taken to compute the shortest route and this will give a bad user experience to the customer. The core functionalities of this project such as route planner is designed to be available offline after considering the fact that not every smartphone users subscribe to a data plan and a quality data connection is not always guaranteed especially when travelling underground using the rail system. So the application is not suitable to be implemented as web application because web application needs a consistent and stable data connection. Although hybrid application can work in offline state, but the lack of established standard over hybrid development platform is refraining user from using it (Singh and Palmieri, 2012).

2.4 Existing Application In The Market

Currently there are 3 Android based route planning for Malaysia rail transportation is available in the market. All the application will be studied, analysed and compared based on the functions that will be delivered under the current project scope. Other than that, 2 overseas route planning application will be analysed too to act as a benchmark for the project.

2.4.1 KL Transport Planner

The user interface for the application is bad because the interface does not scale well with the device screen. Besides that, the shortcut icons provided at the bottom of application lacks description and is not informative. The figure below shows the main screen of the application.



Journey Planner

From

To

Options

● Prefer Rail

○ Prefer Bus Only

Find

Figure 2.6: Screenshot of KL Transport Planner

This application provides the following function,

- Route Planner

  Although there is a route planner function in the application, but the route planner is relying on RapidKL website as the backend and an active data connection is needed to compute the route. The main problem with RapidKL route planner is that it lacks integration to other rail operators. The route planner is only able to compute the route for the 4 rail lines (KL Monorail, Kelana Jaya Line, Ampang Line and Sri Petaling Line) serving under the company. Other rail line such as KTM Komuter & ERL is not included in the rail planner.

- Rail Route Map

  The application does provide a zoom-able rail route map for Klang Valley.

- Savable & Sharable Route

  The route result can be saved for future reference. It can be shared to other application that supports JPEG uploading.

- Fare Display

  This application can show the correct fare needed for travel.

- Live Service Status

  This application can show live service status, but it is not properly formatted. Service status from other rail operators such as KTM is harvested using Twitter. Thus some useless information such as re-tweets and morning greetings are also extracted which will cause confusions to user.

The following functions are not supported in the application.

- Estimated Travel Time
- Station Information

2.4.2 KL Transit Planner

The user interface of this application is considered good due to the simplicity of the application. The main screen of the application is shown in the figure below.

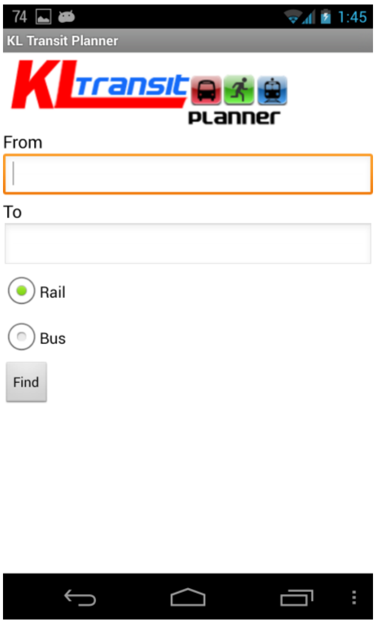


Figure 2.7: Screenshot of KL Transit Planner

This application provides the following function,

- Route Planner

  This application provides a route planner, but it is riding on RapidKL web service as the backend. As discussed earlier, RapidKL route planner lacks integration to other rail operators.

- Fare Display

  This application can show the correct fare needed for travel.

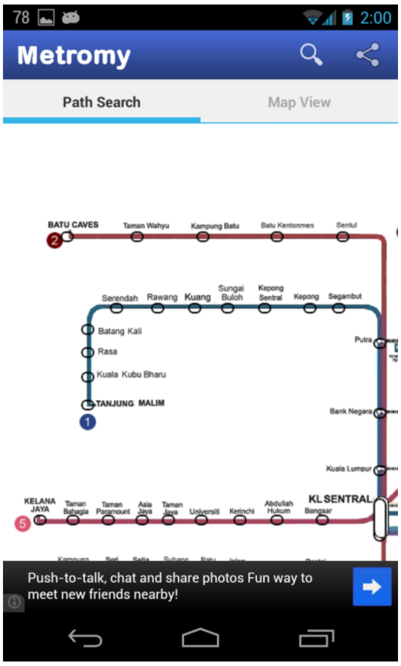
The following functions are not supported in the application.

- Rail Route Map
- Savabe & Sharable Route
- Estimated Arrival Time

- Live Service Status
- Station Information

2.4.3 Metromy

The user interface for this application is considered good because it scales well to the device screen. The main screen of the application is shown in the figure below.



Push-to-talk, chat and share photos Fun way to meet new friends nearby!

Figure 2.8: Screenshot of Metromy

The application provides the following function,

- Route Planner

  This application provides an offline route planner which integrates all the rail route lines in Klang Valley.

- Rail Route Map

  The application does provide a zoom-able rail route map for Klang Valley.

- Fare Display

  This application can show the correct fare needed for travel.

The following functions are not supported in the application.

- Savable & Sharable Route
- Estimated Travel Time
- Live Service Status
- Station Information

2.4.4 Hong Kong MTR Mobile

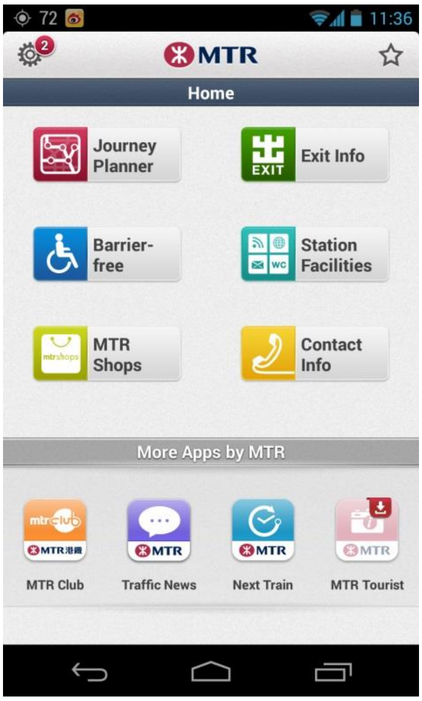


Figure 2.9: Screenshot of Hong Kong MTR Mobile

The application provides the following function,

- Route Planner
  This application provides an offline route planner for the Hong Kong MTR System.

- Rail Route Map
  The application does provide a zoom-able rail route map for Hong Kong MTR System.

- Fare Display
  This application can show the correct fare needed for travel.

- Estimated Travel Time
  This application can show the estimated travel time.

- Savable & Sharable Route
  The route result can be saved and shared.

- Station Information
  This application provides functionality to view station information.

The following functions are not supported in the application.

- Live Service Status

2.5 Shortest Path Problem in Graph Theory

Shortest Path problem is the problem of finding the most optimal shortest path from a specific source node to the destination node using graph. A graph is made up of nodes/vertices and edges connecting them. It can be directed or undirected. Directed graph has direction associated with edges while undirected graph does not have direction. Other than that, a graph can also be weighted or unweighted. Weighted graph is associated with label on the edges, usually a real number where the number can be the distance between two nodes or time taken to travel between two nodes. Unweighted graph does not have any label on the edges. The figure below shows the different element of a graph.

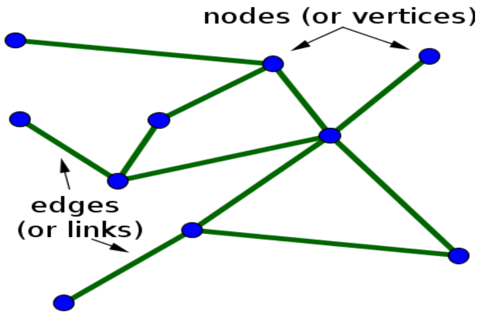


Figure 2.10: Nodes and Edges of a Graph

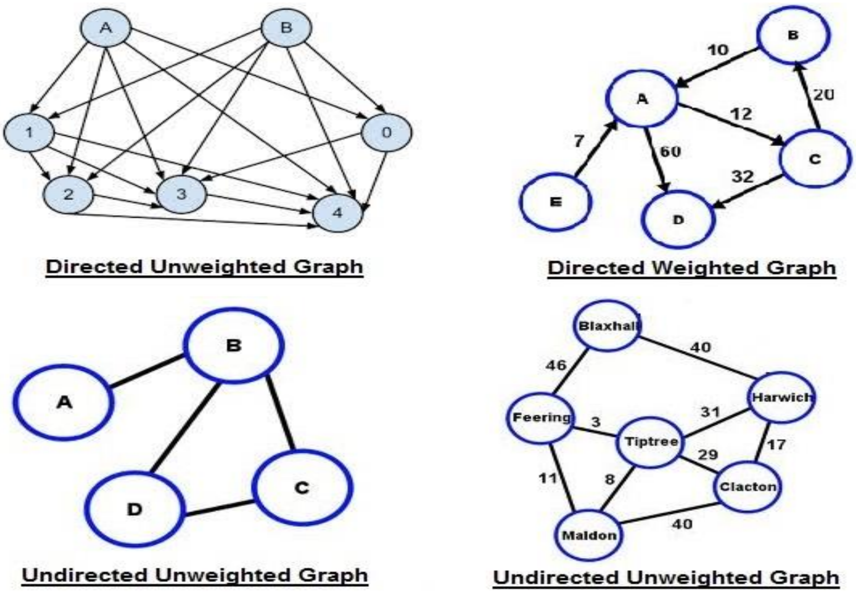


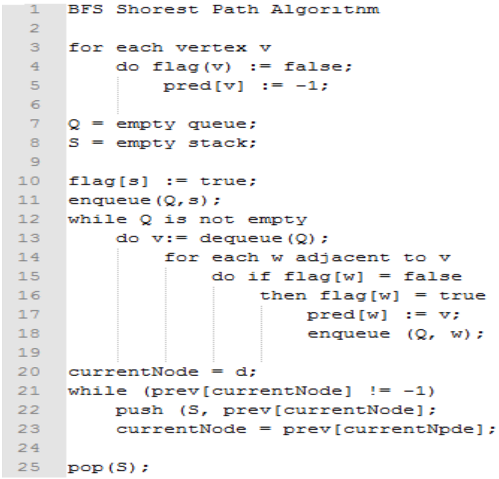
Figure 2.11: Different Type of Graph

2.5.1 Breadth First Search

Breadth First Search is a classical graph traversal algorithm. It can be extended to find and report a shortest path between 2 nodes. The precondition for Breadth First Search to work is that the graph must be unweighted. The weighted version of Breadth First Search is known as Dijkstra Algorithm (Howland, Lewis, Hicks and Pitts, 2003) which will be discussed later. The steps below describe the Breadth First Search algorithm to obtain shortest path.

1) Initialized an empty Visited Table and set last visited node to -1 for all nodes and Adjacency List that contains all the neighbours for a particular node. Initialize an empty Queue.
2) Place the source node into the queue and mark it as current.
3) Traverse through the adjacency list and enqueue all the neighbours belonging to current node. Mark all neighbour as visited and update the current node as their last visited node.
4) Dequeue the current node. The first in queue will be chosen as current node. Step 3 – 4 will be repeated until all nodes are marked as visited.
5) To print the shortest path between 2 nodes, loop the visited table and print the last visited node in reverse order using a stack.

The pseudo code for Breadth First Search is shown in the figure below.



```
BFS Shorest Path Algorithm

for each vertex v
    do flag(v) := false;
        pred[v] := -1;

Q = empty queue;
S = empty stack;

flag[s] := true;
enqueue(Q,s);
while Q is not empty
    do v:= dequeue(Q);
        for each w adjacent to v
            do if flag[w] = false
                then flag[w] = true
                    pred[w] := v;
                    enqueue (Q, w);

currentNode = d;
while (prev[currentNode] != -1)
    push (S, prev[currentNode];
    currentNode = prev[currentNpde];

pop(S);
```

Figure 2.12: Breadth First Search Pseudo Code

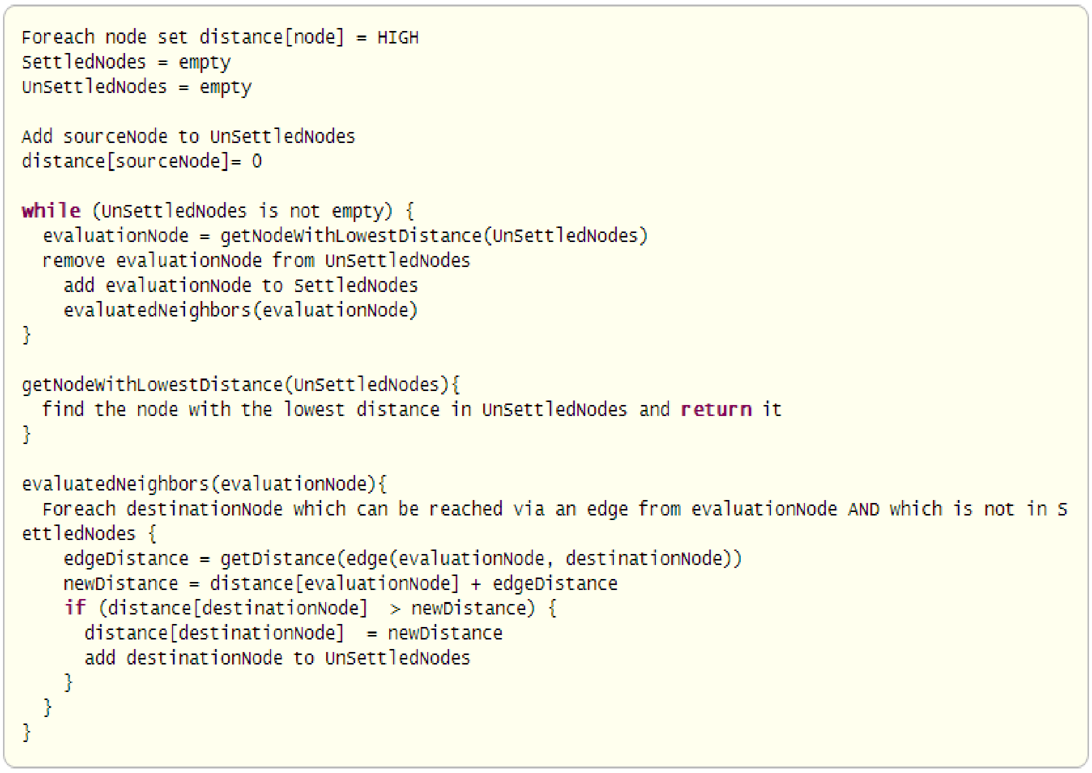
2.5.2 Dijkstra Algorithm

Dijkstra Algorithm is one of the best known and most widely implemented single-source shortest path algorithms. It is introduced by a famous Dutch computer scientist Edsger Dijkstra in year 1959. When applied, Dijikstra Algorithm can compute the shortest path from a chosen origin to the destination. It is known as single source shortest path algorithm because it finds the entire shortest path from a source vertex to all the other vertices in the graph. Dijkstra Algorithm is mostly implemented to solve routing problem.

The pre-condition for Dijkstra Algorithm to work correctly is that the weightage on edges should not be negative. Dijkstra Algorithm will not work correctly if the weightage is a negative number. The steps below describe the Dijkstra Algorithm to obtain shortest path.

1) Initialize the cost of initial node to 0 and all other node to infinity.
2) Mark the initial node as current and mark all the other nodes as unvisited. Create a record to store all the unvisited nodes.
3) Calculate the distance between the current node with all the unvisited neighbours. If the distance is smaller than previously examine distance, the shortest result distance will be overwritten.
4) When all possible neighbours of current node have been computed, the current node will be marked as visited and removed from unvisited record. Node marked as visited will not be checked anymore.
5) Select the next unvisited node with shortest distance and set it to current node. Repeat step 3 – 5 until the destination node is marked as visited.

The pseudo code for Dijkstra Algorithm is shown in the figure below.



```
Foreach node set distance[node] = HIGH
SettledNodes = empty
UnSettledNodes = empty

Add sourceNode to UnSettledNodes
distance[sourceNode]= 0

while (UnSettledNodes is not empty) {
  evaluationNode = getNodeWithLowestDistance(UnSettledNodes)
  remove evaluationNode from UnSettledNodes
    add evaluationNode to SettledNodes
    evaluatedNeighbors(evaluationNode)
}

getNodeWithLowestDistance(UnSettledNodes){
  find the node with the lowest distance in UnSettledNodes and return it
}

evaluatedNeighbors(evaluationNode){
  Foreach destinationNode which can be reached via an edge from evaluationNode AND which is not in S
ettledNodes {
    edgeDistance = getDistance(edge(evaluationNode, destinationNode))
    newDistance = distance[evaluationNode] + edgeDistance
    if (distance[destinationNode]  > newDistance) {
      distance[destinationNode]  = newDistance
      add destinationNode to UnSettledNodes
    }
  }
}
```

Figure 2.13: Dijkstra Algorithm Pseudo Code

2.5.3 Comparison between Breadth First Search & Dijkstra Algorithm

Table 2.5: Comparison Between BFS & Dijkstra Algorithm

	Breadth First Search	Dijkstra
Weighted Graph	Not Supported	Supported
Efficiency	Good	Good
Complete & Optimal Solution	Yes	Yes

After evaluating both algorithms, Dijkstra Algorithm is found to be more suitable for the implementation of this project. This is because Breadth First Search does not support a weighted graph. Weightage on the edges is very important in this project because it can represent the distance between two rail stations. Total distance from origin to destination should be taken account because it is useful in deducing the estimated arrival time for the whole journey. In terms of efficiency, Breadth First

Search may perform slightly faster than Dijkstra because it does not take the weightage into consideration, but the performance difference should not be significant because the graph size for the route planner is not very big. The main advantage of both algorithms is that both of them promised a complete and optimal solution should there exists a path between the origin and destination.

CHAPTER 3

METHODOLOGY

3.1 Evaluation of Development Methodology

3.1.1 Spiral Model

The spiral software development model is first defined by Barry Boehm in year 1986. The model combines the iterative nature of prototyping model and the systemic aspect of conventional waterfall model. The spiral model is shown in the figure below.

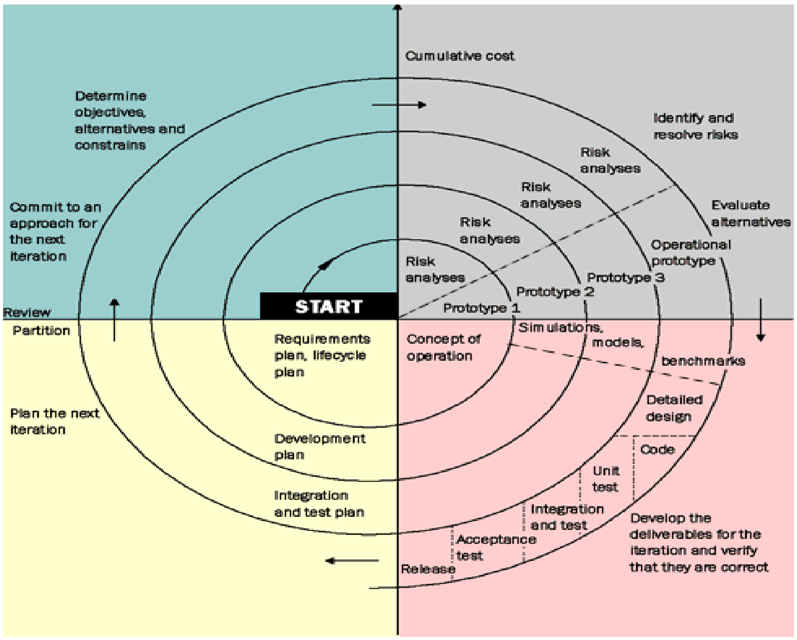


Figure 3.1: Spiral Development Model

Software developed at the end of each loop is evaluated by customers. Feedbacks from customer will be taken into consideration and is incorporated into the planning and design phase of next iteration which results in a more complete and user friendly software. The iteration continues throughout the lifeline of software development. One example of Spiral model development is the evolution of Android mobile operating system. Android 1.0 is regarded as the product of first iteration. After the launching of Android 1.0, Google enhances the features of Android 2.0 in the second iteration based on feedback collected from users.

Each loop contains 4 sections

1) Determine Objectives, Alternatives and Constraints.
2) Evaluate Alternatives, Identify, Resolve Risks.
3) Develop, Verify Next Level Product
4) Plan Next Phase

The risks of the project can be better predicted because risk analysis will be carried out at each loop. This allows project manager to foresee on-going risks and take necessary adjustment to rectify them.

Spiral model is usually suitable for projects that are,

- Large in scale and mission critical.
- High risk projects.

The disadvantages of Spiral Model are

- High cost.
- Requires a lot of risk assessments expertise.
- Amount of documentation increases with each iteration.
- New and not widely used.

3.1.2 Prototyping Model

Prototyping is a type of software development method which works best when requirements and objectives cannot be formulated clearly at initial stage. Brief requirements are collected and a rough prototype will be build based on the drafted requirements. Based on the prototype build, requirements are further elicited and prototype will be enhanced based on information collected. The prototype refining and re-evaluation process will repeat and stop only when clear requirements are sought. The different phases of prototyping model are shown in the figure below.

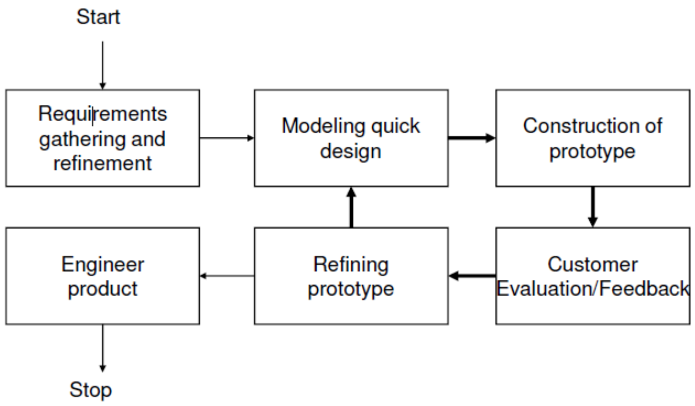

Start
Requirements gathering and refinement
Modeling quick design
Construction of prototype
Customer Evaluation/Feedback
Refining prototype
Engineer product
Stop

Figure 3.2: Prototyping Model

There are 2 types of prototype.

1. Closed-ended ("Throw Away")
   a. The prototype serves solely as demonstration for requirements collection. Will be abandoned once clear requirements are defined.
2. Open-ended
   a. Prototype will be developed into final product according once requirements are defined.

Prototyping Model is best suited for project that,

- Initial requirements cannot be clearly defined.
- Requires active user participation
- Programs that requires a lot of user interaction.

The disadvantages of using prototyping model are:

- Slow
- Customer might assume prototype as working model and expect fast completion.
- Active involvement of user increases change to system thus impacting development progress.

3.1.3 Rapid Application Development (RAD)

Rapid Application Development is an iterative software development methodology that stressed on rapid delivery of high quality software in a short interval of time by reducing the time spent on pre-planning. Software is broken into modules where each module prototype is built, evaluated and refined iteratively. Breaking the software into modules provides more flexibility as any changes can be adapted quickly. This method encourages the use of CASE tools and existing software frameworks to shorten time taken for analysis, design and implementation to ensure a swift delivery of a usable end product.

There are 4 phases in RAD,

1. Requirement & Planning Phase
    - Requirements Elicitation and Analysis.
2. User Design Phase
    - Develop models and prototypes.
3. Construction Phase
    - Coding & Debugging.
4. Cutover Phase
    - Testing, user training.

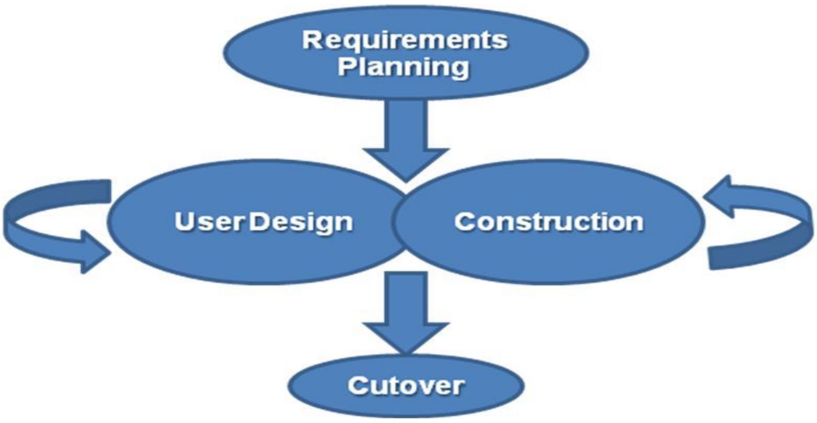


Figure 3.3: Rapid Application Development

RAD is suitable for,

- Project scale is small.
- Tight deadline.
- Project requirements and scope can be clearly defined.
- Developers are skilled.
- Low risk program.

The disadvantages of RAD is,

- Fast development speed might compromise quality.
- Difficulties to modularize software.
- Requires strong commitment from team member.

3.1.4 Adopted Project Development Methodology

After evaluating all the 3 development methodologies, Rapid Application Development is found to be the best methodology that is suited to the nature of current project.

- Project Scale: Small
- Development Duration: Approximately 4 months
- Requirements have been clearly defined and outlined.
- The project does not carry any risks.
- The project can be modularized.

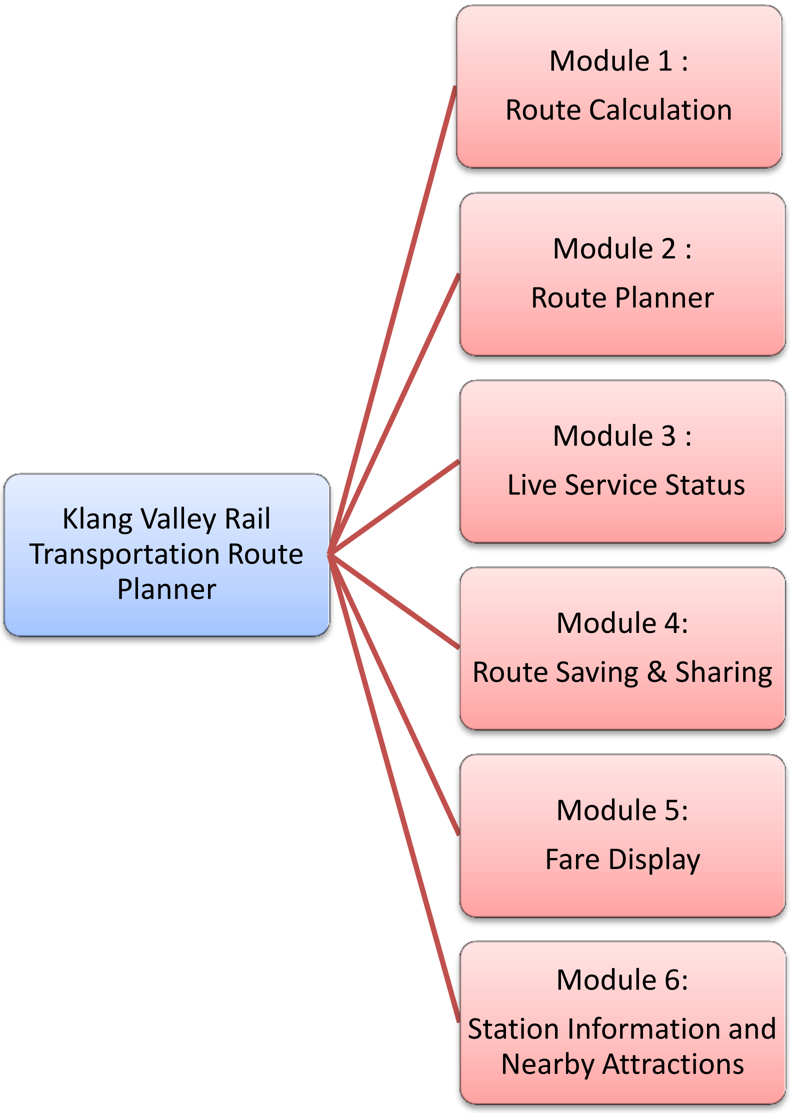

Klang Valley Rail Transportation Route Planner
Module 1 : Route Calculation
Module 2 : Route Planner
Module 3 : Live Service Status
Module 4: Route Saving & Sharing
Module 5: Fare Display
Module 6: Station Information and Nearby Attractions

Figure 3.4: Modules of the Project.

3.2 Development Tools

- Android SDK

  Android Software Development Kit comprises of a comprehensive set of tools for development under the Android platform. It includes the debugger, libraries, Android Emulator, documentation, sample code and tutorials. The official integrated IDE is Eclipse using the Android Development Tools plugin. Android is choosen because it is the world most widely used mobile operating system and the development cost is lowest.

- Apache Web Server

  Apache is the most popular open sourced web server in the world. Apache can be extended by installing plug-in modules. Apache supports a wide range of features such as CGI, SSL and virtual domain. Apache is free for download and is available in most OS. It is distributed freely by Apache Software Foundation.

- SQLite Database

  SQLite is a light weight database developed to support mobile operating system. It is a stripped down version of RDBMS and often resides in the system architecture of the implementing environment. It is open sourced and is ACID compliance.

- Notepad++

  Notepad++ is a free source code editor. It supports a wide range of programming and scripting language such as C, C++, C#, Java, PHP, Javascript and etc. Notepad++ is extensible by installing plugins. It is very popular because it is able to support a wide range of programming languages. Notepad++ will be used to write PHP codes for the web server in this project.

3.3 Project Plan

3.3.1 Work Breakdown Structure

The project will be divided into 6 phases.

1. Planning & Analysis
2. Design
3. Development
4. Testing
5. Deployment
6. Documentation

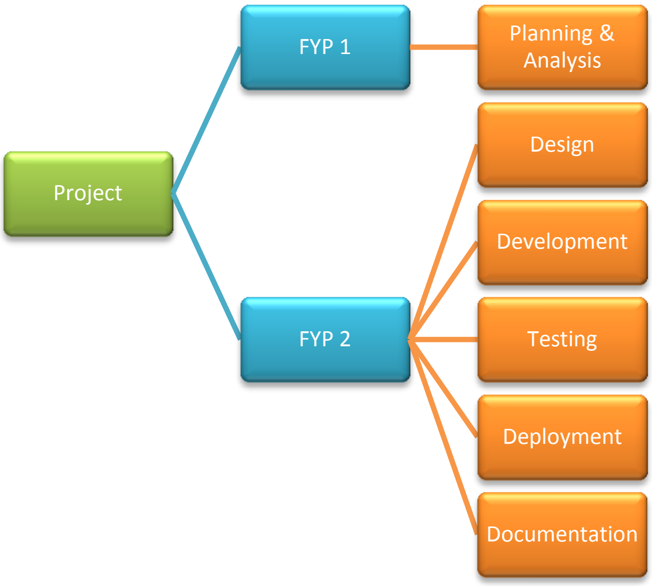
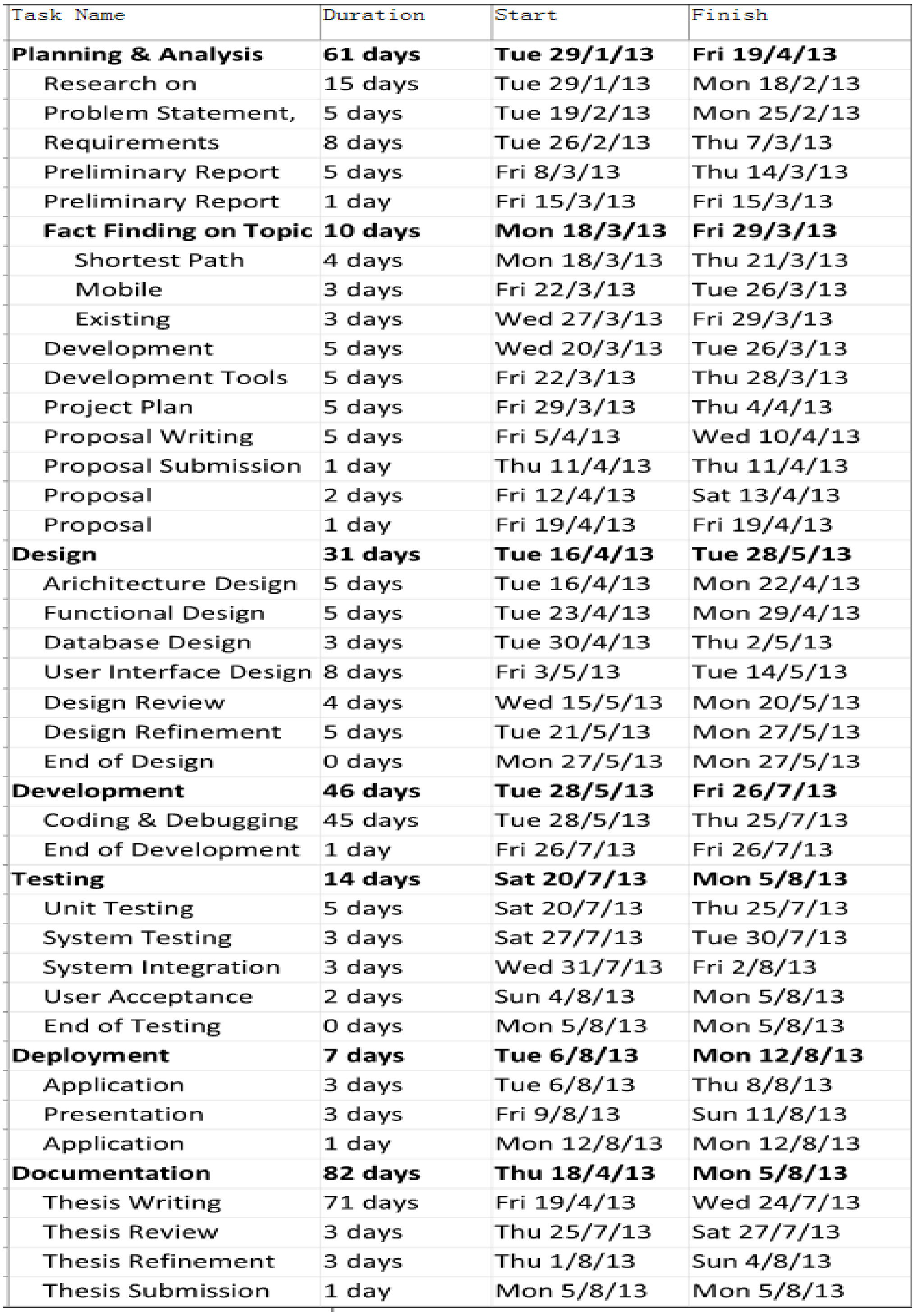


Figure 3.5: WBS Chart

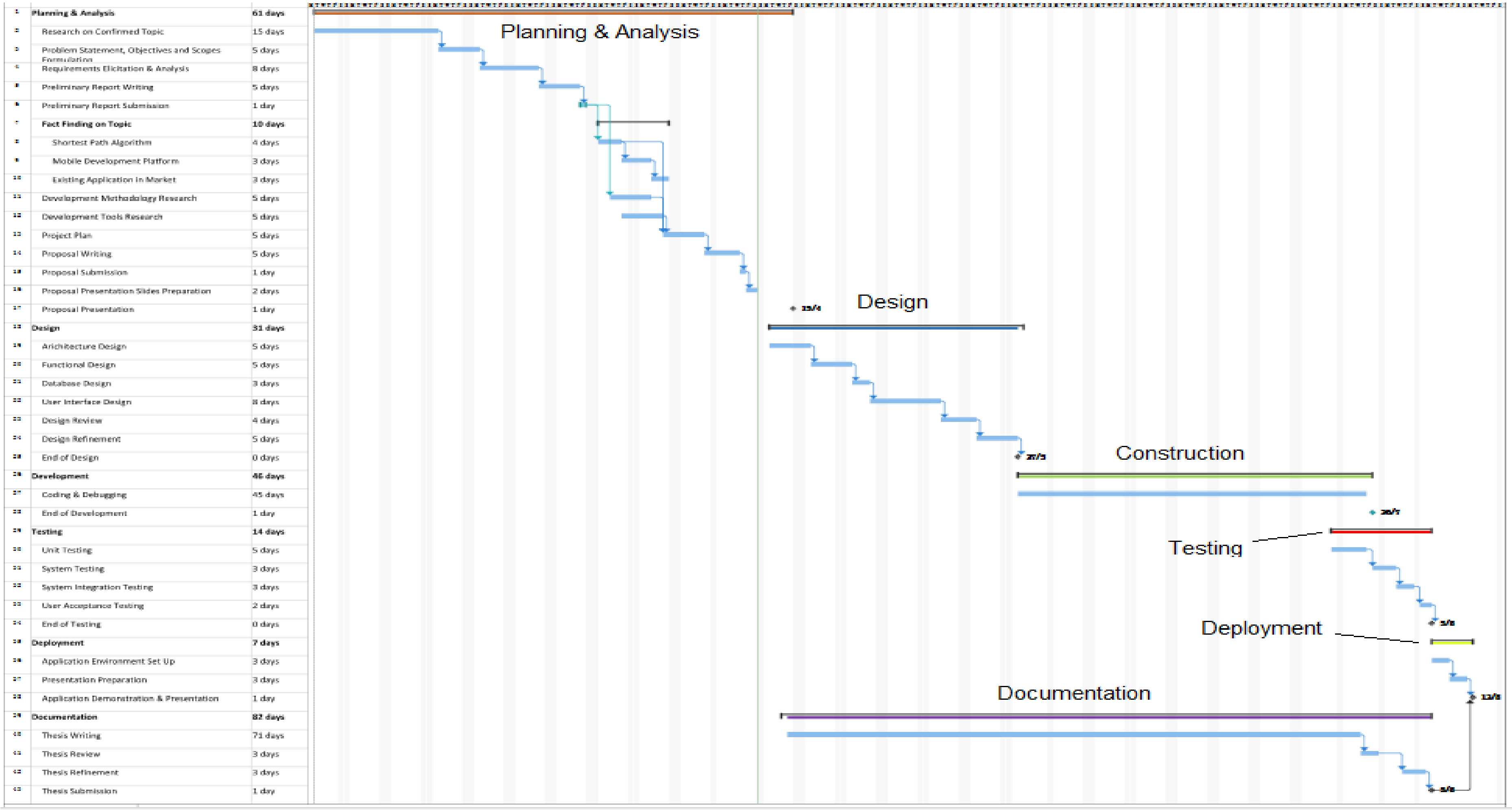
3.3.2 Gantt Chart

The figure below shows the tasks performed their duration, start date and finish date.



Task Name	Duration	Start	Finish
Planning & Analysis	**61 days**	**Tue 29/1/13**	**Fri 19/4/13**
Research on	15 days	Tue 29/1/13	Mon 18/2/13
Problem Statement,	5 days	Tue 19/2/13	Mon 25/2/13
Requirements	8 days	Tue 26/2/13	Thu 7/3/13
Preliminary Report	5 days	Fri 8/3/13	Thu 14/3/13
Preliminary Report	1 day	Fri 15/3/13	Fri 15/3/13
Fact Finding on Topic	**10 days**	**Mon 18/3/13**	**Fri 29/3/13**
Shortest Path	4 days	Mon 18/3/13	Thu 21/3/13
Mobile	3 days	Fri 22/3/13	Tue 26/3/13
Existing	3 days	Wed 27/3/13	Fri 29/3/13
Development	5 days	Wed 20/3/13	Tue 26/3/13
Development Tools	5 days	Fri 22/3/13	Thu 28/3/13
Project Plan	5 days	Fri 29/3/13	Thu 4/4/13
Proposal Writing	5 days	Fri 5/4/13	Wed 10/4/13
Proposal Submission	1 day	Thu 11/4/13	Thu 11/4/13
Proposal	2 days	Fri 12/4/13	Sat 13/4/13
Proposal	1 day	Fri 19/4/13	Fri 19/4/13
Design	**31 days**	**Tue 16/4/13**	**Tue 28/5/13**
Arichitecture Design	5 days	Tue 16/4/13	Mon 22/4/13
Functional Design	5 days	Tue 23/4/13	Mon 29/4/13
Database Design	3 days	Tue 30/4/13	Thu 2/5/13
User Interface Design	8 days	Fri 3/5/13	Tue 14/5/13
Design Review	4 days	Wed 15/5/13	Mon 20/5/13
Design Refinement	5 days	Tue 21/5/13	Mon 27/5/13
End of Design	0 days	Mon 27/5/13	Mon 27/5/13
Development	**46 days**	**Tue 28/5/13**	**Fri 26/7/13**
Coding & Debugging	45 days	Tue 28/5/13	Thu 25/7/13
End of Development	1 day	Fri 26/7/13	Fri 26/7/13
Testing	**14 days**	**Sat 20/7/13**	**Mon 5/8/13**
Unit Testing	5 days	Sat 20/7/13	Thu 25/7/13
System Testing	3 days	Sat 27/7/13	Tue 30/7/13
System Integration	3 days	Wed 31/7/13	Fri 2/8/13
User Acceptance	2 days	Sun 4/8/13	Mon 5/8/13
End of Testing	0 days	Mon 5/8/13	Mon 5/8/13
Deployment	**7 days**	**Tue 6/8/13**	**Mon 12/8/13**
Application	3 days	Tue 6/8/13	Thu 8/8/13
Presentation	3 days	Fri 9/8/13	Sun 11/8/13
Application	1 day	Mon 12/8/13	Mon 12/8/13
Documentation	**82 days**	**Thu 18/4/13**	**Mon 5/8/13**
Thesis Writing	71 days	Fri 19/4/13	Wed 24/7/13
Thesis Review	3 days	Thu 25/7/13	Sat 27/7/13
Thesis Refinement	3 days	Thu 1/8/13	Sun 4/8/13
Thesis Submission	1 day	Mon 5/8/13	Mon 5/8/13

Figure 3.6: Tasks Breakdown for Gantt Chart



1	**Planning & Analysis**	**61 days**
2	Research on Confirmed Topic	15 days
3	Problem Statement, Objectives and Scopes Formulation	5 days
4	Requirements Elicitation & Analysis	8 days
5	Preliminary Report Writing	5 days
6	Preliminary Report Submission	1 day
7	**Fact Finding on Topic**	**10 days**
8	Shortest Path Algorithm	4 days
9	Mobile Development Platform	3 days
10	Existing Application In Market	3 days
11	Development Methodology Research	5 days
12	Development Tools Research	5 days
13	Project Plan	5 days
14	Proposal Writing	5 days
15	Proposal Submission	1 day
16	Proposal Presentation Slides Preparation	2 days
17	Proposal Presentation	1 day
18	**Design**	**31 days**
19	Architecture Design	5 days
20	Functional Design	5 days
21	Database Design	3 days
22	User Interface Design	8 days
23	Design Review	4 days
24	Design Refinement	5 days
25	End of Design	0 days
26	**Development**	**46 days**
27	Coding & Debugging	45 days
28	End of Development	1 day
29	**Testing**	**14 days**
30	Unit Testing	5 days
31	System Testing	3 days
32	System Integration Testing	3 days
33	User Acceptance Testing	2 days
34	End of Testing	0 days
35	**Deployment**	**7 days**
36	Application Environment Set Up	3 days
37	Presentation Preparation	3 days
38	Application Demonstration & Presentation	1 day
39	**Documentation**	**82 days**
40	Thesis Writing	71 days
41	Thesis Review	3 days
42	Thesis Refinement	3 days
43	Thesis Submission	1 day

Figure 3.7: Gantt Chart

CHAPTER 4

PROJECT SPECIFICATION

4.1 Requirements Gathering & Analysis

Requirements gathering and analysis is a basic and crucial process in software project management. Requirements gathering are the process of collecting the correct input from various stakeholders using the most suitable methods and identify the problems they faced. Requirements analysis involves deep analysis of problems identified in requirements gathering phase and transform them into correct specification. Both steps are very important because it identifies what the system should do in order to satisfy all the stakeholders. It serves as a foundation for the execution of project. If mistake made on requirements gathering and analysis process is not fixed, the size of mistake will expand as the porject progress and will eventually cause the whole project to fail.

There are various techniques available to elicit requirements for different project with different context. In this project, 2 techniques below are used to harvest the correct requirements.

- Study of Existing System
  Similar route planning application available in the market will be downloaded, installed, analysed and compared.

- Document Analysis

Document analysis involves the study of existing documents that are relevant for the project. In this project, documents such as the train arrival time, frequency table, fare table, station information and station facilities are studied thoroughly and valuable information is extracted for the use of project.

4.1.1 Outcome of Study of Existing System

4 applications are downloaded from Google Play Market. 3 of the applications are designed specifically for the use in Malaysia. Another application is specially designed for Hong Kong MTR. The functionality of all the applications is summarized in the table below.

Table 4.1: Functions of various rail planner application.

	Route Planner	Rail Map	Good UI	Savable & Sharable Route	Fare Display	Live Service Status	Estimated Travel Time	Station Information
KL Transport Planner	√	√	×	√	√	√	×	×
KL Transit Planner	√	×	×	×	√	×	×	×
Metromy	√	√	√	×	√	×	×	×
Hong Kong MTR	√	√	√	√	√	×	√	√

4.1.2 Outcome of Document Analysis

Studies on the document provided by various rail operators found that the frequency of train arrival is different in different time. This will affect the estimated travel time of the commuter and should be taken into consideration when building the application. The frequency of train is summarized in the table below.

Table 4.2: Rail Lines and Average Train Frequency

Rail Lines	Average Frequency (mins)
Seremban Line	15
Port Klang Line	15
Ampang Line	7
Sri Petaling Line	7
Kelana Jaya Line	5
KLIA Ekspres Line	25
KLIA Transit Line	25
MRT Line	5

Other than that, it is discovered that the speed of train varies with distance because of acceleration and deceleration. Stations that have short distance in between have a lower train speed. The average speed on different distance is summarized in the table below.

Table 4.3: Speed of train vs Station Distance.

Rail Lines	Station Distance	Speed
Seremban Line & Port Klang Line	<1 KM	40km/h
	<1.5 KM	45km/h
	<2 KM	50km/h
	<2.5 KM	55km/h
	<4.0 KM	60km/h
	<5.5 KM	65km/h
	>8.0 KM	70km/h
Ampang Line & Sri	<1 KM	40km/h

Petaling Line	<1.5 KM	45km/h
	<2 KM	50km/h
	<2.5 KM	55km/h
	>3 KM	60km/h
Kelana Jaya Line & MRT Line	<1 KM	40km/h
	<1.5 KM	45km/h
	<2 KM	50km/h
	>2.5 KM	55km/h
KLIA Ekspres Line	-	140km/h
KLIA Transit Line	-	100km/h
Walking Speed	-	5km/h

4.2 Feasibility Study

Feasibility study is a preliminary process undertaken before the start of any project to analyse the viability of a project. The study helps to answer question such as "Should the propose project idea continue? Or it is not realistic to continue with the project."

4.2.1 Economic Feasibility

Economic feasibility is carried out to gauge the economic viability of a project by performing cost/benefit analysis. The overall cost for this project is considered very low because

- Hardware Cost
    - Only requires a laptop and an Android phone to develop.
    - Requires no hardware customization thus lowering the cost.
- Software Cost
    - Most of the software used is open source and requires no license.
    - For software that needs to be purchased, trial version will be used.

- Human Cost
  - The project is fully develop by the student, no hiring of external expertise needed thus no human cost is involved because no salary needs to be paid.

4.2.2 Technical Feasibility

Technical feasibility is conducted to determine the technology needed for the project, and whether the current technology exists and the assessment of technical competency of the developer. The technology needed for this project is Java programming and Android programming, both the technology is already well established in the market and is being supported actively by a wide base of developer. Other than that, both the subjects have been taught in the campus, thus the developer already possesses enough technical knowledge to undertake the project.

4.2.3 Schedule Feasibility

Schedule feasibility is carried out to determine the duration of project and whether the duration is realistic or not. The total duration for this project is 8 months and is within reasonable range to complete the project of such scale by one person. The progress of the project is actively traced by a series of milestones defined in the project plan, and the systemic approach in software project management will ensure the project to be completed within the stipulated time.

4.2.4 Operational Feasibility

Operational Feasibility is conducted to determine the usefulness of the developed application, whether the application can solve the problems identified during scope definition and how well it can satisfy all the requirements identified. The application is built to help citizens or tourists that are not familiar with Klang Valley rail system to get to their destination correctly. Other than providing the correct route, the system provides extra functionalities to display the service status, station information and nearby attractions. The application is developed on a mobile platform to provide ease of access to the users. The user can download the application and install in their smartphone and use it at anytime and anywhere.

4.3 Functional Requirements

a. Route Calculation Module

i. The application must be able to retrieve vertex and edge data stored in the database.
ii. The application must be able to link the correct vertex with edge to construct a fully linked map.
iii. The application must be able to find the shortest path from a source station to a destination by using travel time as the weight.
iv. The application must be able to calculate the estimated travel time for a given path.
v. The application must be able to export the correct path into a CSV file with correct formatting.

b. Route Planner Module

i. The application must be able to show a user friendly interface that incorporates the latest rail route map for Klang Valley.
ii. The rail route map must be zoom-able.
iii. The application must allow the user to select their origin and destination station.

iv. The application must allow the user to reset origin and destination station.

v. The application must allow the user to swap the origin and destination station.

vi. The application must be able to show a legend for the rail route map.

vii. The application must allow user to click and view station information.

c. Live Service Status Module

i. The application must be able to grab the latest XML file containing service status from the web server.

ii. The application must be able to show the correct service status to the user.

iii. The application must allow the user to refresh the live service status.

iv. The application must be able to show an error message when internet connection is not active.

d. Route Saving & Sharing Module.

i. The application must allow route result to be saved for future reference.

ii. The application must allow user to view the saved route.

iii. The application must allow user to delete the saved route.

iv. The application must allow user to share the route result via SMS/Email.

v. The application must be able to allow user to share station information via SMS/Email.

e. Fare Display Module

   i. The application must be able to display the correct fare payable from origin to destination.

f. Station Information & Nearby Attractions Module.

   i. The application must be able to show the station opening and closing time, and facilities available to the user.
   ii. The application must be able to show a list of attractions nearby the selected station.
   iii. The application must be able to show the details of attractions that have been selected.

4.4 Non Functional Requirements

a. Performance

   i. The application must be able to display the correct route result within 5 seconds of search button click.
   ii. The application must be able to list out the station information for a station within 3 seconds when selected.
   iii. The application must be able to list out the nearby attractions for station within 3 seconds when selected.
   iv. The application must be able to display the correct service status within 8 seconds.

b. Usability

   i. The user interface of the application must be self-descriptive and easy to use.
   ii. The user interface must have a consistent navigation menu throughout the whole application.

CHAPTER 5

SYSTEM DESIGN

5.1 Entity Relationship Diagram

5.1.1 ER Diagram 1 (Route Calculation)

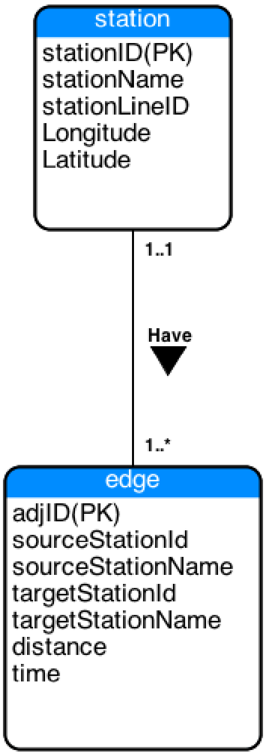


Figure 5.1: ER Diagram for Route Calculation.

As shown on the figure above, the ERD diagram for route calculation involves 2 tables. The station table contains the station information which will be retrieved and stored as a Vertex object in the application. Whereas the edge table contains the links between stations. The edge will be retrieved and stored as Edge object in application. Each station must have 1 or more edge.

5.1.2 ER Diagram 2 (Android Application)

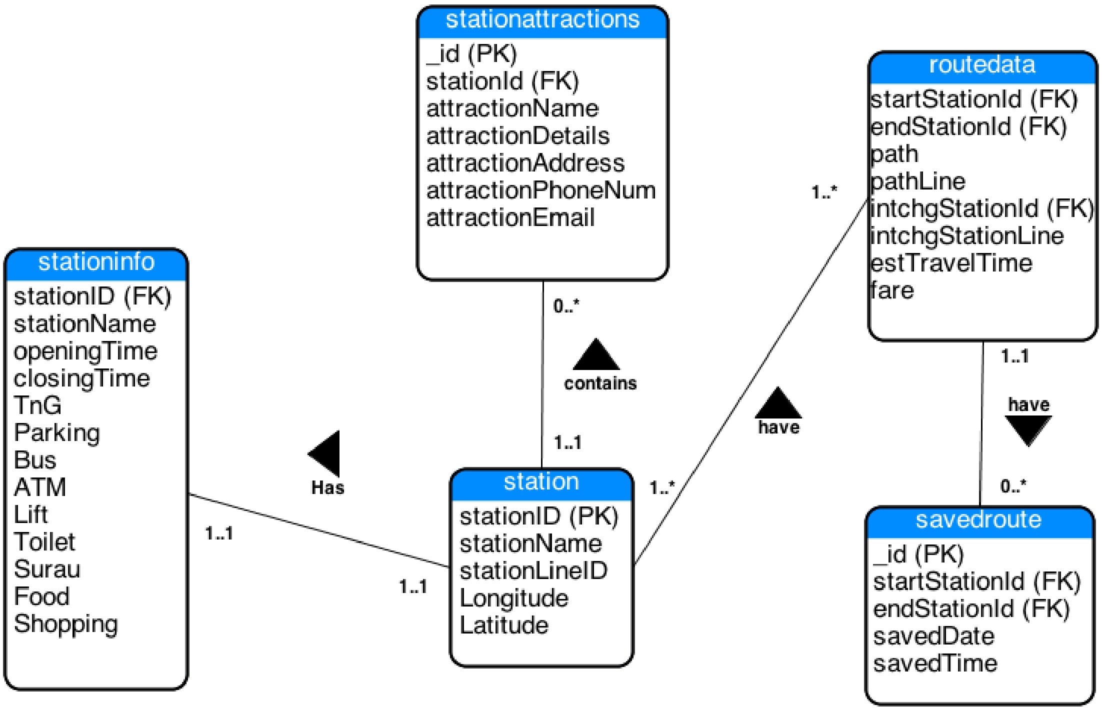


Figure 5.2: ER Diagram for Android Application.

The ERD diagram for the Android application contains 5 tables as shown in the figure above. The function and relationship of each table is described in the table below.

Table 5.1: Description for ER Diagram 2.

Table Name	Functions
station	Contains the station information for every station. Referenced by stationinfo, stationattractions,routedata and savedroute tables.
stationinfo	Contains the station information such as opening time, closing time and facilities available. Each station must have one station information.

stationattractions	Contains nearby attractions for a station. A station can have zero or more attractions. The attraction belonging to a station is identified by the stationId attribute.
routedata	Contains all the pre-calculated route, the interchange station, estimated travel time and fare.
savedroute	Contains the information about saved route.

5.2 Data Dictionary

5.2.1 Data Dictionary for ER Diagram 1

Entity Name: station

Attributes	Description	Data Type	PK/FK	Nulls
stationID	ID that uniquely identifies a station	Integer	PK	N
stationName	Station's Name	String		N
stationLineID	Station's Operating Line	Integer		N
Longitude	Station's Longitude	Double		N
Latitude	Station's Latitude	Double		N

Entity Name: edge

Attributes	Description	Data Type	PK/FK	Nulls
adjID	ID that uniquely identifies an edge	Integer	PK	N
sourceStationId	ID for source station.	Integer	FK	N
sourceStationName	Name for source station.	String		N
targetStationId	ID for target station.	Integer	FK	N
targetStationName	Name for target station	String		N
distance	Distance from source to target station.	Double		N
Time	Time taken to travel from source to target station.	Double		N

5.2.2 Data Dictionary for ER Diagram 2

Entity Name: station

Attributes	Description	Data Type	PK/FK	Nulls
stationID	ID that uniquely identifies a station	Integer	PK	N
stationName	Station’s Name	String		N
stationLineID	Station’s Operating Line	Integer		N
Longitude	Station’s Longitude	Double		N
Latitude	Station’s Latitude	Double		N

Entity Name: routedata

Attributes	Description	Data Type	PK/FK	Nulls
startStationId	ID for origin station.	Integer	FK	N
endStationId	ID for destination station	Integer	FK	N
path	Path data from start to end station.	String		N
pathLine	Line data from start to end station	String		N
intchgStationId	ID for interchange station.	Integer	FK	N
intchgStationLine	Line for interchange station.	Integer		N
estTravelTime	Time taken to travel from source to target station.	Double		N
fare	Fare payable to travel from source to target station.	Double		N

Entity Name: savedroute

Attributes	Description	Data Type	PK/FK	Nulls
startStationId	ID for origin station.	Integer	FK	N
endStationId	ID for destination station	Integer	FK	N
path	Path data from start to end station.	String		N
pathLine	Line data from start to end station	String		N
intchgStationId	ID for interchange station.	Integer	FK	Y

intchgStationLine	Line for interchange station.	Integer		Y
estTravelTime	Time taken to travel from source to target station.	Double		N
fare	Fare payable to travel from source to target station.	Double		N

Entity Name: stationinfo

Attributes	Description	Data Type	PK/FK	Nulls
stationID	ID that uniquely identifies a station	Integer	FK	N
stationName	Name for Station	String		N
openingTime	Opening Time of station	String		N
closingTime	Closing Time of station	String		N
TnG	Touch N Go facility	Integer		Y
Parking	Parking facility	Integer		Y
Bus	Feeder bus facility	Integer		Y
ATM	ATM facility	Integer		Y
Lift	Lift facility	Integer		Y
Toilet	Toilet facility	Integer		Y
Surau	Surau facility	Integer		Y
Food	Food facility	Integer		Y
Shopping	Shopping facility	Integer		Y

Entity Name: stationattractions

Attributes	Description	Data Type	PK/FK	Nulls
_id	ID that uniquely identifies an attraction.	Integer	PK	N
stationId	ID for station.	Integer	FK	N
attractionName	Attraction’s Name.	String		N
attractionDetails	Detail for Attraction.	String		N
attractionAddress	Address for attraction.	String		Y
attractionPhoneN	Phone number for attraction.	String		Y

um				
attractionEmail	Email for attraction.	String		Y

5.3 Use Case Diagram

The diagram below shows the action that can be performed by the application user and administrator.

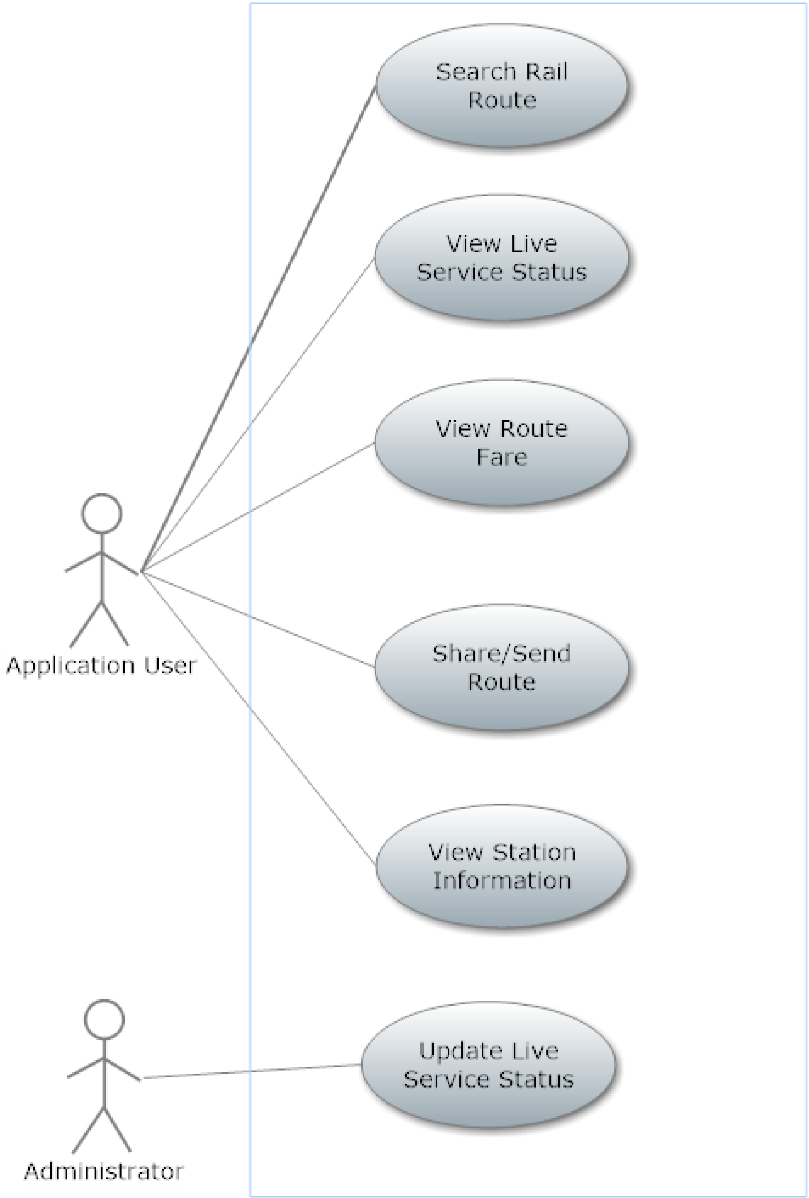


Figure 5.3: Use Case Diagram

5.4 Class Diagram

5.4.1 Class Diagram 1 (Route Calculation)

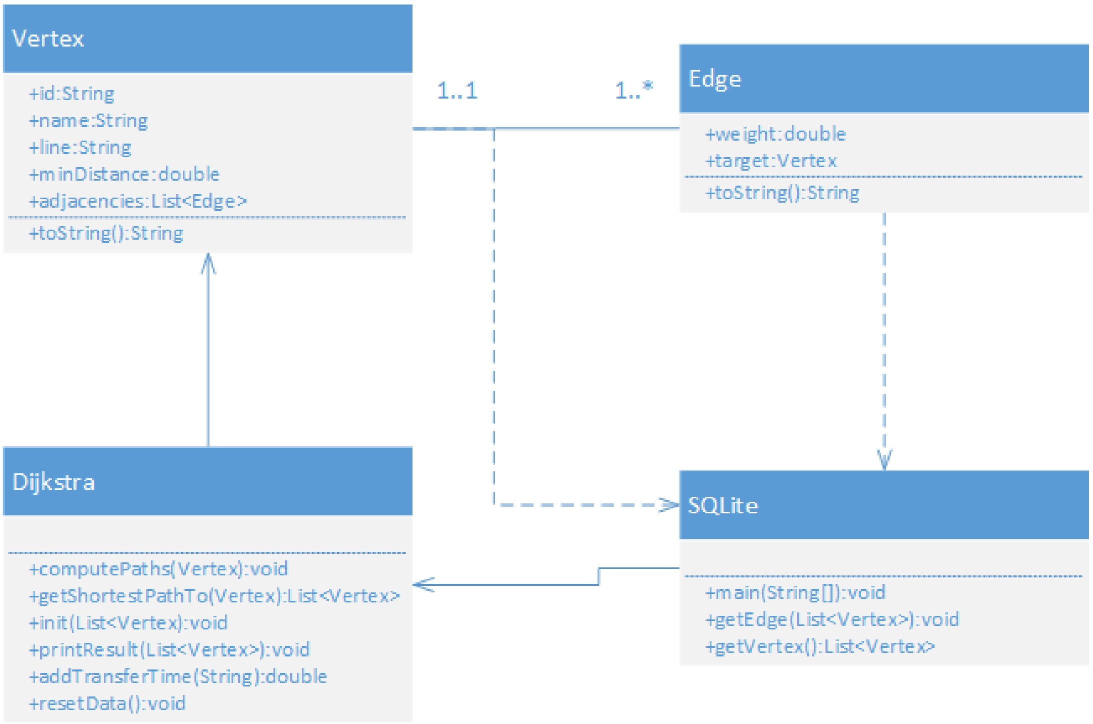


Figure 5.4: Class Diagram for Route Calculation.

As shown in the figure above, the route calculation module contains 4 classes. The vertex class is used to store the station information. Each vertex contains one more edge. Edge is the links between the stations. Class SQLite is used to retrieve station information and edge information from the database and store them into as Vertex and Edge object. Dijkstra class contains the core logic to compute and print the shortest route calculated based on travel time between stations.

5.4.2 Class Diagram 2 (Android Application)

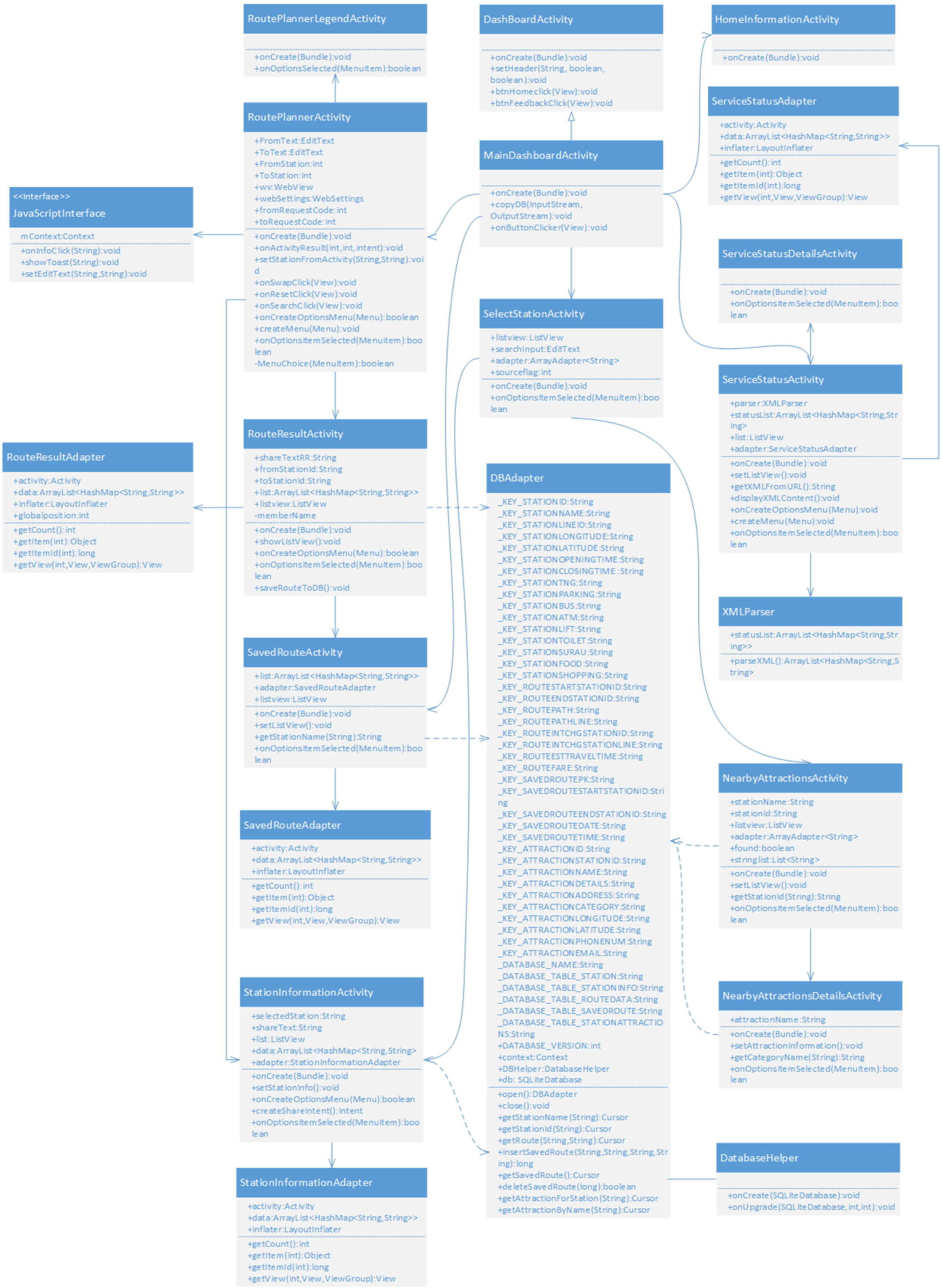

RoutePlannerLegendActivity
+onCreate(Bundle):void
+onOptionsSelected(MenuItem):boolean
DashBoardActivity
+onCreate(Bundle):void
+setHeader(String, boolean, boolean):void
+btnHomeclick(View):void
+btnFeedbackClick(View):void
HomeInformationActivity
+onCreate(Bundle):void
ServiceStatusAdapter
+activity:Activity
+data:ArrayList<HashMap<String,String>>
+inflater:LayoutInflater
+getCount():int
+getItem(int):Object
+getItemId(int):long
+getView(int,View,ViewGroup):View
RoutePlannerActivity
+FromText:EditText
+ToText:EditText
+FromStation:int
+ToStation:int
+wv:WebView
+webSettings:WebSettings
+fromRequestCode:int
+toRequestCode:int
+onCreate(Bundle):void
+onActivityResult(int,int,intent):void
+setStationFromActivity(String,String):void
+onSwapClick(View):void
+onResetClick(View):void
+onSearchClick(View):void
+onCreateOptionsMenu(Menu):boolean
+createMenu(Menu):void
+onOptionsItemSelected(Menuitem):boolean
-MenuChoice(MenuItem):boolean
MainDashboardActivity
+onCreate(Bundle):void
+copyDB(InputStream, OutputStream):void
+onButtonClicker(View):void
<<Interface>>
JavaScriptInterface
mContext:Context
+onInfoClick(String):void
+showToast(String):void
+setEditText(String,String):void
ServiceStatusDetailsActivity
+onCreate(Bundle):void
+onOptionsItemSelected(MenuItem):boolean
SelectStationActivity
+listview:ListView
+searchInput:EditText
+adapter:ArrayAdapter<String>
+sourceflag:int
+onCreate(Bundle):void
+onOptionsItemSelected(MenuItem):boolean
ServiceStatusActivity
+parser:XMLParser
+statusList:ArrayList<HashMap<String,String>
+list:ListView
+adapter:ServiceStatusAdapter
+onCreate(Bundle):void
+setListView():void
+getXMLFromURL():String
+displayXMLContent():void
+onCreateOptionsMenu(Menu):void
+createMenu(Menu):void
+onOptionsItemSelected(MenuItem):boolean
RouteResultActivity
+shareTextRR:String
+fromStationId:String
+toStationId:String
+list:ArrayList<HashMap<String,String>>
+listview:ListView
-memberName
+onCreate(Bundle):void
+showListView():void
+onCreateOptionsMenu(Menu):boolean
+onOptionsItemSelected(MenuItem):boolean
+saveRouteToDB():void
RouteResultAdapter
+activity:Activity
+data:ArrayList<HashMap<String,String>>
+inflater:LayoutInflater
+globalposition:int
+getCount():int
+getItem(int):Object
+getItemId(int):long
+getView(int,View,ViewGroup):View
DBAdapter
_KEY_STATIONID:String
_KEY_STATIONNAME:String
_KEY_STATIONLINEID:String
_KEY_STATIONLONGITUDE:String
_KEY_STATIONLATITUDE:String
_KEY_STATIONOPENINGTIME:String
_KEY_STATIONCLOSINGTIME :String
_KEY_STATIONTNG:String
_KEY_STATIONPARKING:String
_KEY_STATIONBUS:String
_KEY_STATIONATM:String
_KEY_STATIONLIFT:String
_KEY_STATIONTOILET:String
_KEY_STATIONSURAU:String
_KEY_STATIONFOOD:String
_KEY_STATIONSHOPPING:String
_KEY_ROUTESTARTSTATIONID:String
_KEY_ROUTEENDSTATIONID:String
_KEY_ROUTEPATH:String
_KEY_ROUTEPATHLINE:String
_KEY_ROUTEINTCHGSTATIONID:String
_KEY_ROUTEINTCHGSTATIONLINE:String
_KEY_ROUTEESTTRAVELTIME:String
_KEY_ROUTEFARE:String
_KEY_SAVEDROUTEPK:String
_KEY_SAVEDROUTESTARTSTATIONID:String
_KEY_SAVEDROUTEENDSTATIONID:String
_KEY_SAVEDROUTEDATE:String
_KEY_SAVEDROUTETIME:String
_KEY_ATTRACTIONID:String
_KEY_ATTRACTIONSTATIONID:String
_KEY_ATTRACTIONNAME:String
_KEY_ATTRACTIONDETAILS:String
_KEY_ATTRACTIONADDRESS:String
_KEY_ATTRACTIONCATEGORY:String
_KEY_ATTRACTIONLONGITUDE:String
_KEY_ATTRACTIONLATITUDE:String
_KEY_ATTRACTIONPHONENUM:String
_KEY_ATTRACTIONEMAIL:String
_DATABASE_NAME:String
_DATABASE_TABLE_STATION:String
_DATABASE_TABLE_STATIONINFO:String
_DATABASE_TABLE_ROUTEDATA:String
_DATABASE_TABLE_SAVEDROUTE:String
_DATABASE_TABLE_STATIONATTRACTIONS:String
+DATABASE_VERSION:int
+context:Context
+DBHelper:DatabaseHelper
+db: SQLiteDatabase
+open():DBAdapter
+close():void
+getStationName(String):Cursor
+getStationId(String):Cursor
+getRoute(String,String):Cursor
+insertSavedRoute(String,String,String,String):long
+getSavedRoute():Cursor
+deleteSavedRoute(long):boolean
+getAttractionForStation(String):Cursor
+getAttractionByName(String):Cursor
XMLParser
+statusList:ArrayList<HashMap<String,String>>
+parseXML():ArrayList<HashMap<String,String>
SavedRouteActivity
+list:ArrayList<HashMap<String,String>>
+adapter:SavedRouteAdapter
+listview:ListView
+onCreate(Bundle):void
+setListView():void
+getStationName(String):String
+onOptionsItemSelected(MenuItem):boolean
NearbyAttractionsActivity
+stationName:String
+stationId:String
+listview:ListView
+adapter:ArrayAdapter<String>
+found:boolean
+stringlist:List<String>
+onCreate(Bundle):void
+setListView():void
+getStationId(String):String
+onOptionsItemSelected(MenuItem):boolean
SavedRouteAdapter
+activity:Activity
+data:ArrayList<HashMap<String,String>>
+inflater:LayoutInflater
+getCount():int
+getItem(int):Object
+getItemId(int):long
+getView(int,View,ViewGroup):View
StationInformationActivity
+selectedStation:String
+shareText:String
+list:ListView
+data:ArrayList<HashMap<String,String>
+adapter:StationInformationAdapter
+onCreate(Bundle):void
+setStationInfo():void
+onCreateOptionsMenu(Menu):boolean
+createShareIntent():Intent
+onOptionsItemSelected(MenuItem):boolean
NearbyAttractionsDetailsActivity
+attractionName:String
+onCreate(Bundle):void
+setAttractionInformation():void
+getCategoryName(String):String
+onOptionsItemSelected(MenuItem):boolean
DatabaseHelper
+onCreate(SQLiteDatabase):void
+onUpgrade(SQLiteDatabase,int,int):void
StationInformationAdapter
+activity:Activity
+data:ArrayList<HashMap<String,String>>
+inflater:LayoutInflater
+getCount():int
+getItem(int):Object
+getItemId(int):long
+getView(int,View,ViewGroup):View

Figure 5.5: Class Diagram for Android Application.

As shown in the figure above, the Android application contains 21 classes. The main page of the application is represented by MainDashBoardActivity class. The main page contains icon which can redirect the user to different interface. Classes name ending with word Adapter is a helper class to help to populate the user interface in correct order. The DBAdapter is a utility class used to communicate with the embedded SQLite database. The XMLParser class is used to parse the service status XML file obtained from web server.

5.5 Activity Diagram

5.5.1 Route Planner

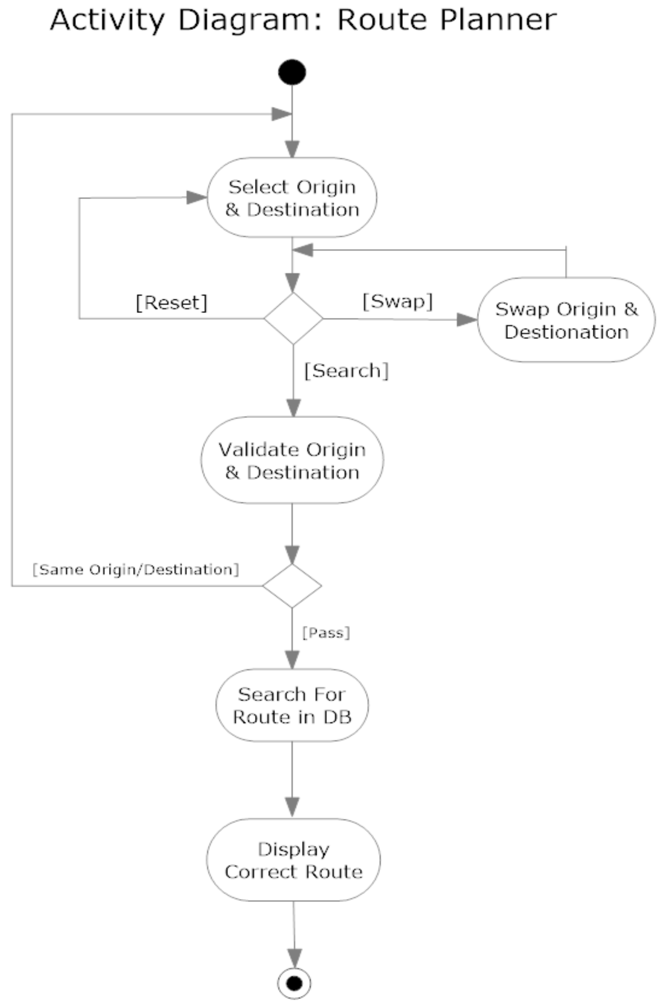

Activity Diagram: Route Planner
Select Origin & Destination
[Reset]
[Swap]
Swap Origin & Destionation
[Search]
Validate Origin & Destination
[Same Origin/Destination]
[Pass]
Search For Route in DB
Display Correct Route

Figure 5.6: Activity Diagram for Route Planner.

5.5.2 Service Status

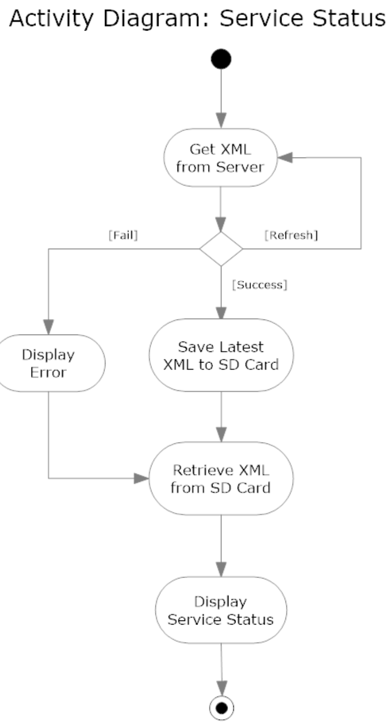

Activity Diagram: Service Status
Get XML from Server
[Fail]
[Refresh]
[Success]
Display Error
Save Latest XML to SD Card
Retrieve XML from SD Card
Display Service Status

Figure 5.7: Activity Diagram for Service Status.

5.5.2.1 Station Information

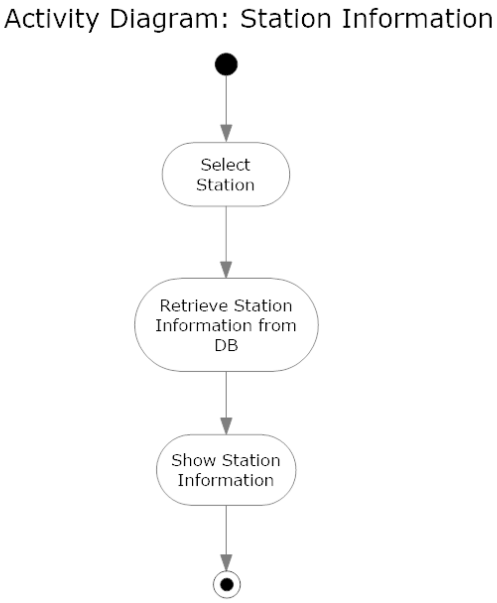

Activity Diagram: Station Information
Select Station
Retrieve Station Information from DB
Show Station Information

Figure 5.8: Activity Diagram for Station Information.

5.5.3 Nearby Attractions

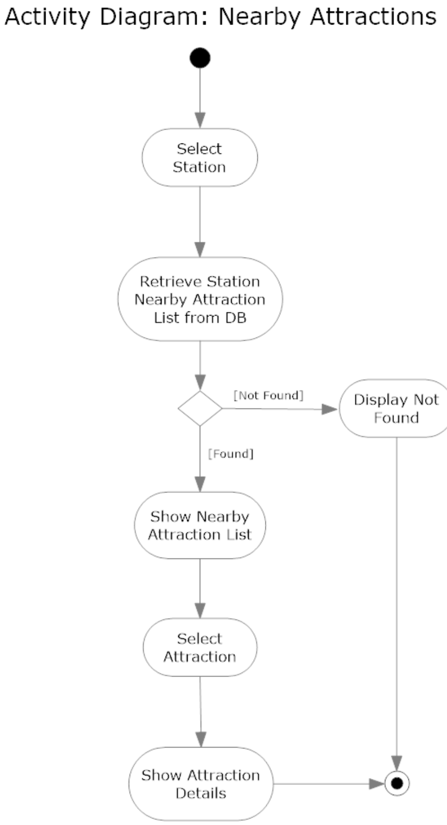

Activity Diagram: Nearby Attractions
Select Station
Retrieve Station Nearby Attraction List from DB
[Not Found]
Display Not Found
[Found]
Show Nearby Attraction List
Select Attraction
Show Attraction Details

Figure 5.9: Activity Diagram for Nearby Attractions.

5.5.4 Saved Route

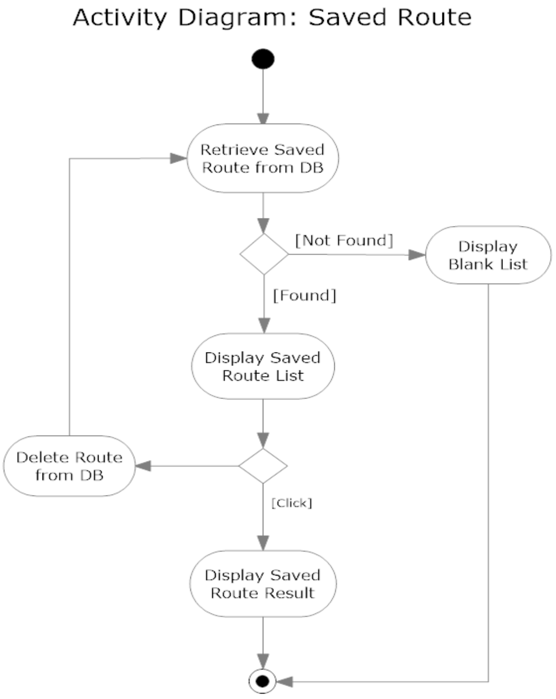


Figure 5.10: Activity Diagram for Saved Route.

5.5.5 Saving Favourite Route

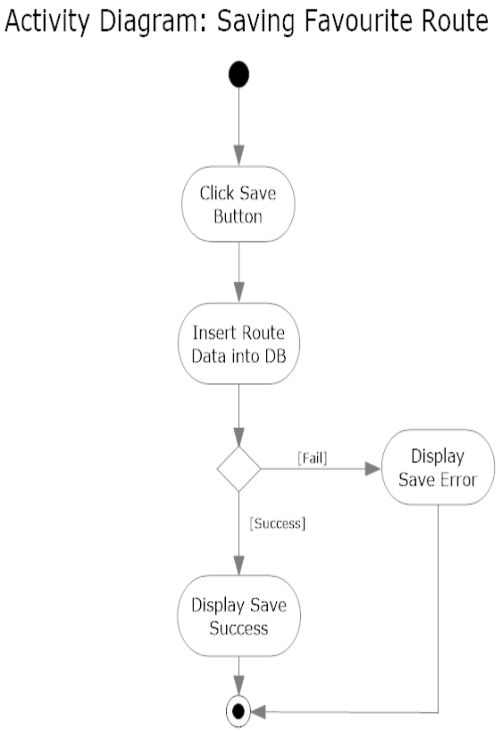


Figure 5.11: Activity Diagram for Saving Favourite Route.

5.5.6 Share Route

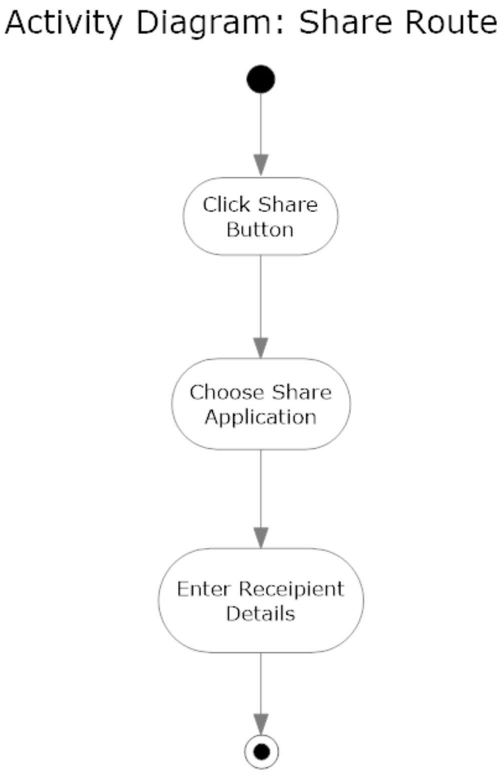


Figure 5.12: Activity Diagram for Share Route.

5.6 Deployment Diagram

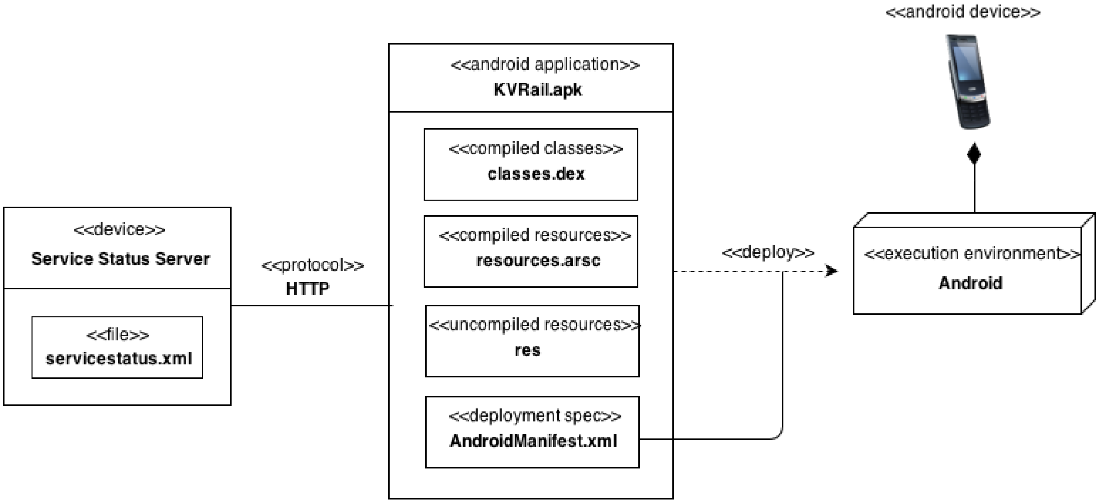


Figure 5.13: Deployment Diagram for the Project.

5.7 Overall System Architecture

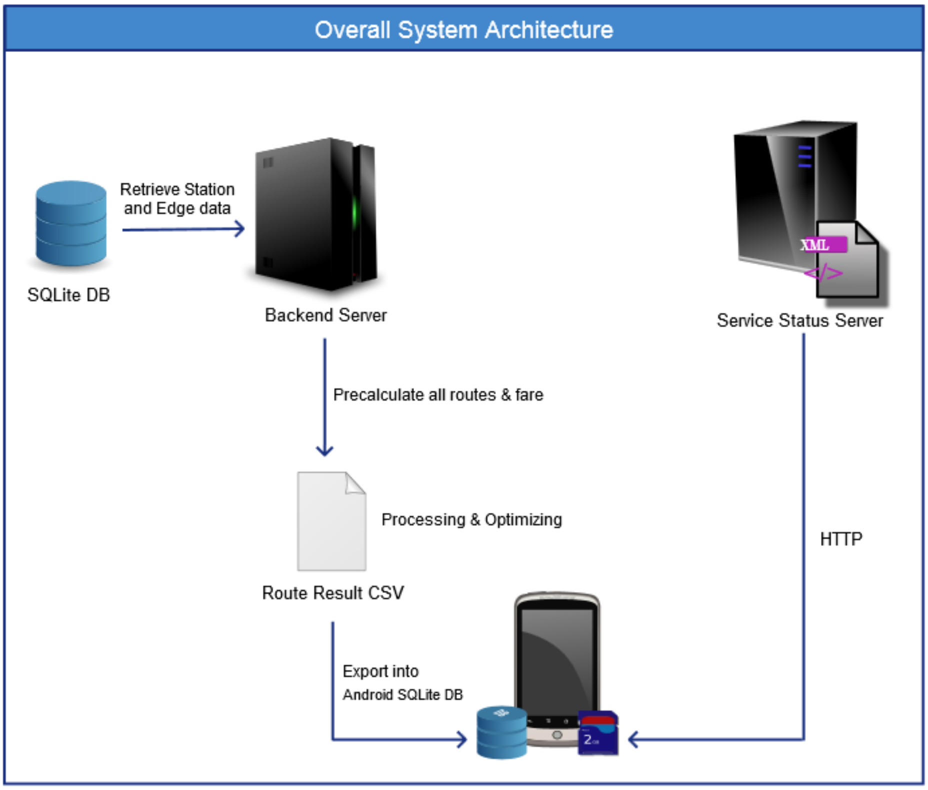

Overall System Architecture
Retrieve Station and Edge data
SQLite DB
Backend Server
Service Status Server
XML
</>
Precalculate all routes & fare
Processing & Optimizing
Route Result CSV
HTTP
Export into Android SQLite DB

Figure 5.14: Overall System Architecture of Project.

The figure above describes the overall system architecture of the application. The route result is pre-calculated by a backend server and will be stored into a CSV file to facilitate processing. The fare data will be added to the Route Result CSV file. Once processing and optimization of route is completed, the CSV file will be exported into the SQLite database located in the device. The shortest route algorithm is not implemented inside the android application due to the lack of processing power and memory on mobile devices. The XML file containing service status will be stored inside a web server. The application will grab the XML file from the web server and store it into SD Card. The saving of XML into SD Card is to enable user to view the latest-updated service status if internet connection is not present.

5.8 User Interface Design

5.8.1 Main Page

The figure below shows the Main landing interface of the application. It is organized according to Android Dashboard Design Pattern.

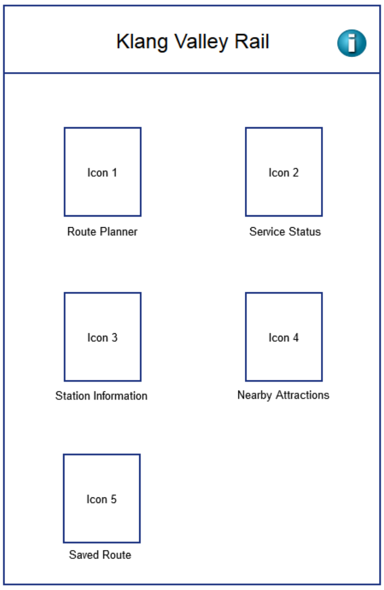


Figure 5.15: UI Design for Main Page.

5.8.2 Route Planner

The figure below shows the route planner page. It allows the user to select from and to station. Other than selecting from a list of station, user can select stations by tapping on their desired station in the route map. The route map can be pinched and zoomed according to user preference. Clicking the information icon will redirect the user to the Legend page.

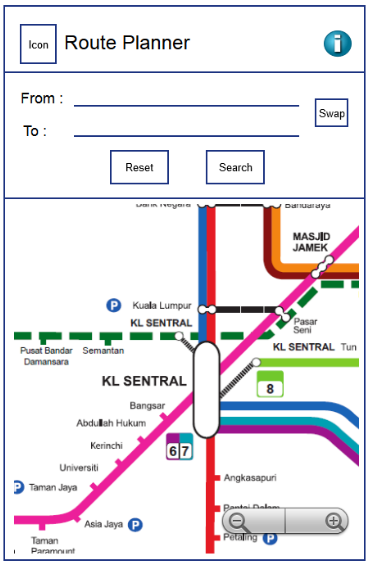
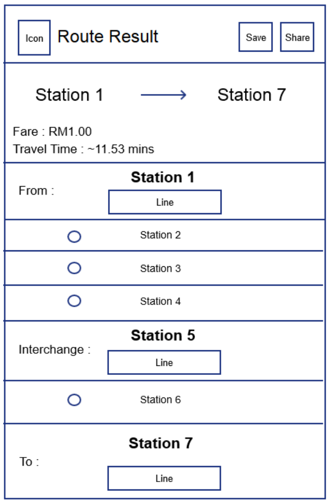


Figure 5.16: UI Design for Route Planner.

5.8.3 Route Result

The figure below shows the user interface of route result. The route result can be saved or shared by clicking on the button on top right corner. The fare and estimated travel time will be shown to the user too.



Route Result

Station 1 → Station 7

Fare : RM1.00
Travel Time : ~11.53 mins

From : **Station 1**
Line

○ Station 2

○ Station 3

○ Station 4

Interchange : **Station 5**
Line

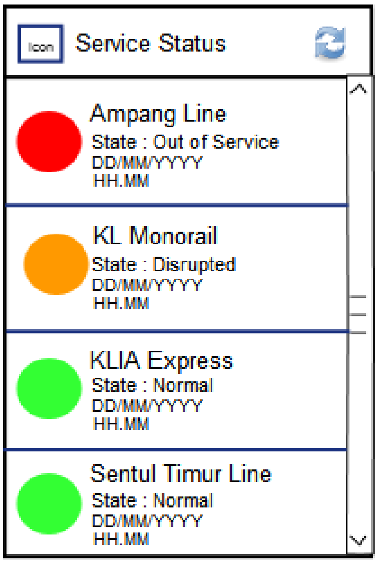
○ Station 6

To : **Station 7**
Line

Figure 5.17: UI Design for Route Result.

5.8.4 Service Status

The figure below shows the service status user interface.



Service Status

Ampang Line
State : Out of Service
DD/MM/YYYY
HH.MM

KL Monorail
State : Disrupted
DD/MM/YYYY
HH.MM

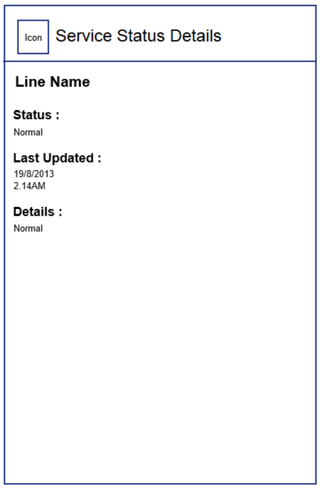
KLIA Express
State : Normal
DD/MM/YYYY
HH.MM

Sentul Timur Line
State : Normal
DD/MM/YYYY
HH.MM

Figure 5.18: UI Design for Service Status.

5.8.5 Service Status Details

The figure below shows the user interface of service status details.



Line Name

Status :
Normal

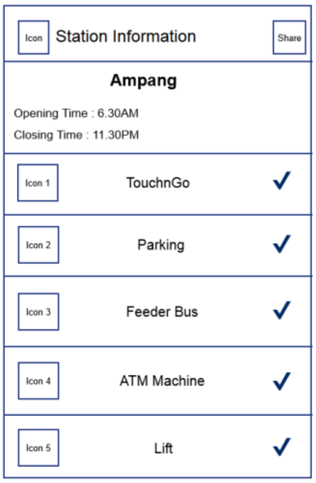
Last Updated :
19/8/2013
2.14AM

Details :
Normal

Figure 5.19: UI Design for Service Status Details.

5.8.6 Station Information

The figure below shows the station information user interface. All the facilities available in the station will be listed out along with opening and closing time. The station information can be shared via SMS/Email by clicking on the icon on top right corner.



Ampang

Opening Time : 6.30AM
Closing Time : 11.30PM

Icon 1 TouchnGo ✓

Icon 2 Parking ✓

Icon 3 Feeder Bus ✓

Icon 4 ATM Machine ✓

Icon 5 Lift ✓

Figure 5.20: UI Design for Station Information.

5.8.7 Nearby Attractions

The figure below shows the nearby attractions list for a station.

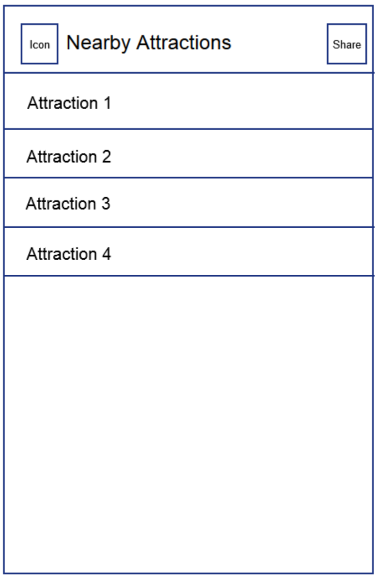
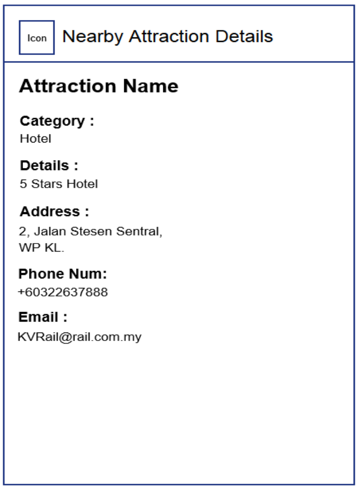


Figure 5.21: UI Design for Nearby Attractions.

5.8.8 Nearby Attraction Details

The figure below shows the nearby attraction details for an attraction. It will display the details, address, phone number and email of the attractions.



Attraction Name

Category :
Hotel

Details :
5 Stars Hotel

Address :
2, Jalan Stesen Sentral,
WP KL.

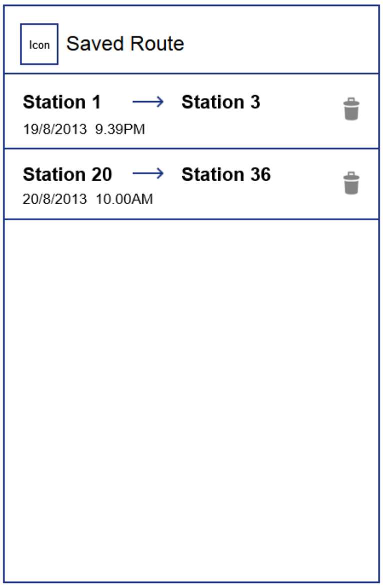
Phone Num:
+60322637888

Email :
KVRail@rail.com.my

Figure 5.22: UI Design for Nearby Attraction Details.

5.8.9 Saved Route

The figure below shows the saved route user interface. All saved route will be shown in the page together with the route station, date and time. The delete button can be clicked to remove saved route.



Station 1 ⟶ Station 3
19/8/2013 9.39PM

Station 20 ⟶ Station 36
20/8/2013 10.00AM

Figure 5.23: UI Design for Saved Route.

CHAPTER 6

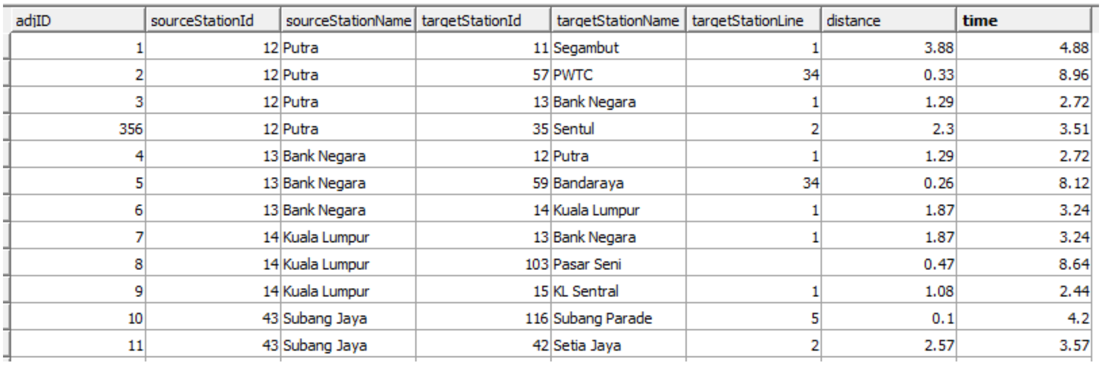
IMPLEMENTATION & TESTING

6.1 Implementation of Application

6.1.1 Route Calculation Module

In this project, the route calculation module is separated from the Android application. The reason being so is that the route calculation algorithm requires a lot of resources and processing power, this is not suitable to be executed on the mobile platform due to limited processing power and memory available on mobile devices. All possible routes from one station to another will be pre-calculated with a laptop computer and results will be exported into a CSV file to add in fare details. The route calculation is done using Dijkstra Algorithm with time of travel between stations as the weight. Time is choosen as the weight because it is more realistic compared to number of stations. The lowest number of stations would not mean it is the best route because some of it fails to take into account the time of switching trains.

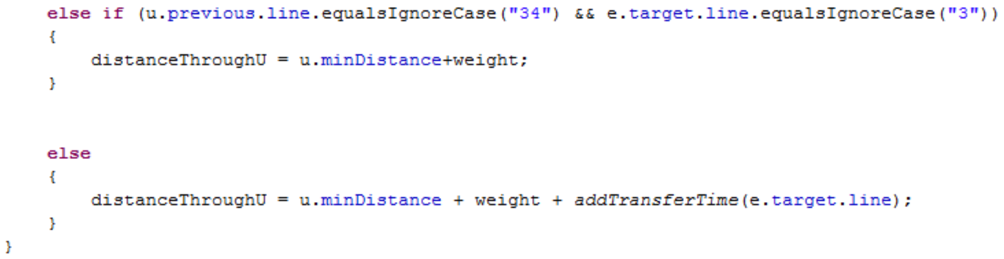
The time of needed to travel between stations are calculated based on the information harvested from the requirements gathering and analysis phase earlier. The figure below shows the edge database that contains link information for every station and the time needed to travel on the links.



adjID	sourceStationId	sourceStationName	targetStationId	targetStationName	targetStationLine	distance	time
1	12	Putra	11	Segambut	1	3.88	4.88
2	12	Putra	57	PWTC	34	0.33	8.96
3	12	Putra	13	Bank Negara	1	1.29	2.72
356	12	Putra	35	Sentul	2	2.3	3.51
4	13	Bank Negara	12	Putra	1	1.29	2.72
5	13	Bank Negara	59	Bandaraya	34	0.26	8.12
6	13	Bank Negara	14	Kuala Lumpur	1	1.87	3.24
7	14	Kuala Lumpur	13	Bank Negara	1	1.87	3.24
8	14	Kuala Lumpur	103	Pasar Seni		0.47	8.64
9	14	Kuala Lumpur	15	KL Sentral	1	1.08	2.44
10	43	Subang Jaya	116	Subang Parade	5	0.1	4.2
11	43	Subang Jaya	42	Setia Jaya	2	2.57	3.57

Figure 6.1: Structure for edge table.

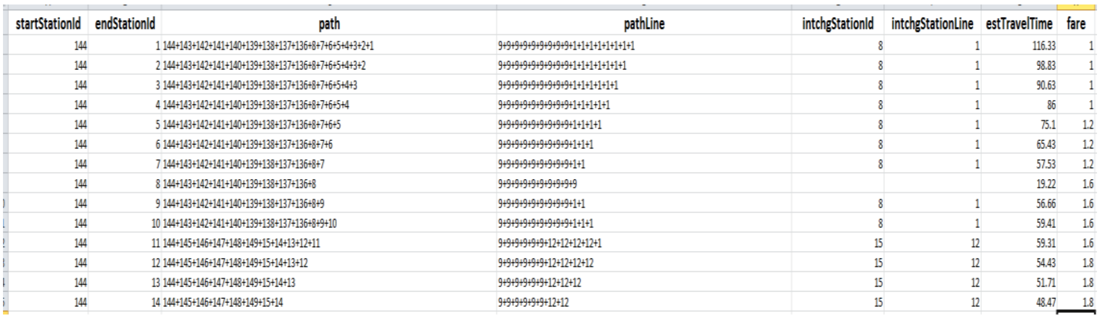
The Dijkstra Algorithm used to calculate the shortest route is modified to add in transfer time dynamically during execution to obtain a more realistic path. The figure below shows a portion of the code that adds the transfer time when the application comes across an interchange node.



```
        else if (u.previous.line.equalsIgnoreCase("34") && e.target.line.equalsIgnoreCase("3"))
        {
            distanceThroughU = u.minDistance+weight;
        }

        else
        {
            distanceThroughU = u.minDistance + weight + addTransferTime(e.target.line);
        }
    }
```

Figure 6.2: Section of Code for Adding Transfer Time.

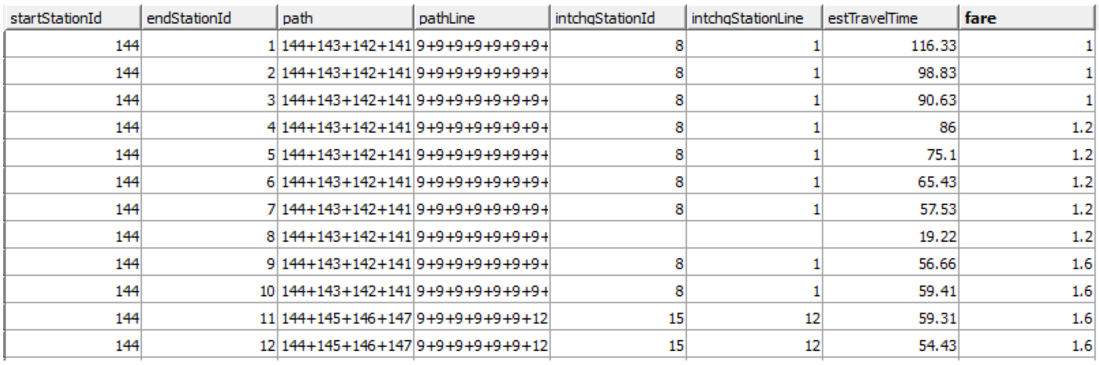
Each station will contain 162 routes to other stations, and the total number of pre-calculated route for all stations will be *163 stations x 162 routes = 26406 routes*. After the route has been calculated, it will be printed into a CSV file for easier processing. The CSV file will be imported into Microsoft Excel and fare details will be added to it. Fare details needs to be updated manually because there is no payment integration between KTM, KLIA Ekpress, KLIA Transit and RapidKL. Other than that, there are no observable patterns in the payment calculation which can be automated. The sample of CSV result file is shown in the figure below.



startStationId	endStationId	path	pathLine	intchgStationId	intchgStationLine	estTravelTime	fare
144	1	144+143+142+141+140+139+138+137+136+8+7+6+5+4+3+2+1	9+9+9+9+9+9+9+9+9+1+1+1+1+1+1+1+1	8	1	116.33	1
144	2	144+143+142+141+140+139+138+137+136+8+7+6+5+4+3+2	9+9+9+9+9+9+9+9+9+1+1+1+1+1+1+1	8	1	98.83	1
144	3	144+143+142+141+140+139+138+137+136+8+7+6+5+4+3	9+9+9+9+9+9+9+9+9+1+1+1+1+1+1	8	1	90.63	1
144	4	144+143+142+141+140+139+138+137+136+8+7+6+5+4	9+9+9+9+9+9+9+9+9+1+1+1+1+1	8	1	86	1
144	5	144+143+142+141+140+139+138+137+136+8+7+6+5	9+9+9+9+9+9+9+9+9+1+1+1+1	8	1	75.1	1.2
144	6	144+143+142+141+140+139+138+137+136+8+7+6	9+9+9+9+9+9+9+9+9+1+1+1	8	1	65.43	1.2
144	7	144+143+142+141+140+139+138+137+136+8+7	9+9+9+9+9+9+9+9+9+1+1	8	1	57.53	1.2
144	8	144+143+142+141+140+139+138+137+136+8	9+9+9+9+9+9+9+9+9+9			19.22	1.6
144	9	144+143+142+141+140+139+138+137+136+8+9	9+9+9+9+9+9+9+9+9+1+1	8	1	56.66	1.6
144	10	144+143+142+141+140+139+138+137+136+8+9+10	9+9+9+9+9+9+9+9+9+1+1+1	8	1	59.41	1.6
144	11	144+145+146+147+148+149+15+14+13+12+11	9+9+9+9+9+9+12+12+12+12+1	15	12	59.31	1.6
144	12	144+145+146+147+148+149+15+14+13+12	9+9+9+9+9+9+12+12+12+12	15	12	54.43	1.8
144	13	144+145+146+147+148+149+15+14+13	9+9+9+9+9+9+12+12+12	15	12	51.71	1.8
144	14	144+145+146+147+148+149+15+14	9+9+9+9+9+9+12+12	15	12	48.47	1.8

Figure 6.3: CSV Structure of Pre-calculated Route Data.

After adding fare details on the route result data, the Microsoft Excel file will be converted to a CSV file and exported into routedata SQLite file. The routedata SQLite file will be used by the android application to show the correct routes. The figure below shows the structure of routedata SQLite table.



startStationId	endStationId	path	pathLine	intchgStationId	intchgStationLine	estTravelTime	fare
144	1	144+143+142+141	9+9+9+9+9+9+9+	8	1	116.33	1
144	2	144+143+142+141	9+9+9+9+9+9+9+	8	1	98.83	1
144	3	144+143+142+141	9+9+9+9+9+9+9+	8	1	90.63	1
144	4	144+143+142+141	9+9+9+9+9+9+9+	8	1	86	1.2
144	5	144+143+142+141	9+9+9+9+9+9+9+	8	1	75.1	1.2
144	6	144+143+142+141	9+9+9+9+9+9+9+	8	1	65.43	1.2
144	7	144+143+142+141	9+9+9+9+9+9+9+	8	1	57.53	1.2
144	8	144+143+142+141	9+9+9+9+9+9+9+			19.22	1.2
144	9	144+143+142+141	9+9+9+9+9+9+9+	8	1	56.66	1.6
144	10	144+143+142+141	9+9+9+9+9+9+9+	8	1	59.41	1.6
144	11	144+145+146+147	9+9+9+9+9+9+12	15	12	59.31	1.6
144	12	144+145+146+147	9+9+9+9+9+9+12	15	12	54.43	1.6

Figure 6.4: Structure of routedata table.

6.1.2 Route Planner Module

The route planner module is implemented on the Android mobile operating platform. The module contains a screen to select the origin and destination station and another interface to show the route result to the user. The user interface for station selection is shown in the figure below.

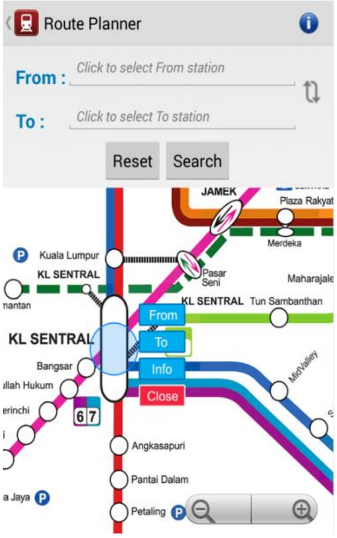


Figure 6.5: Screenshot of Route Planner Interface.

Other than selecting station by tapping beside the From/To text, user can pinch and zoom the rail map below and select their stations or view information by pressing on the submenu. The From station will be highlighted in green colour and the To station in red. When the points are clicked, it will show a blue circle with a submenu with operations. The figure below shows the different operations that can be performed by clicking on the map.

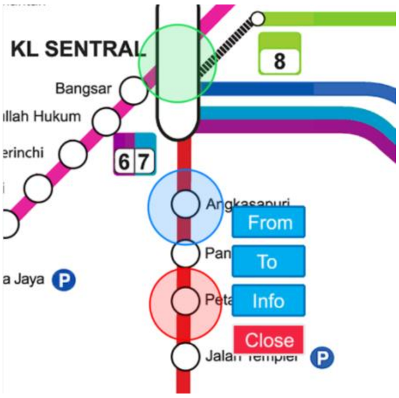
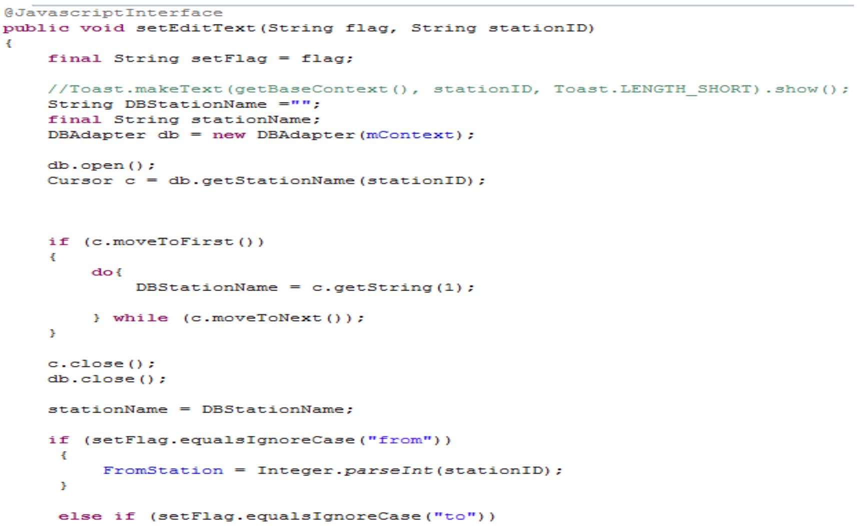


Figure 6.6: Screenshot of Map Highlight & Sub Menu.

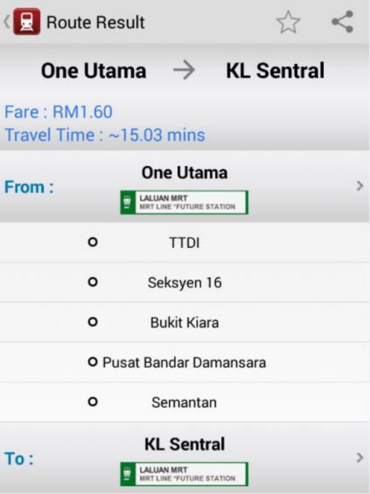
The resolution of the rail map is set to 3080 x 3060 pixels to minimize the deterioration in quality due to zoom. In order to speed up the image rendering process, the rail map is divided into 96 tiles, each measuring 255 x 385 pixels. The image is hold in place using a HTML file and the communication of the map and android is handled by a JavaScript interface. Section of the JavaScript interface code is shown in the figure below.



```
@JavascriptInterface
public void setEditText(String flag, String stationID)
{
    final String setFlag = flag;

    //Toast.makeText(getBaseContext(), stationID, Toast.LENGTH_SHORT).show();
    String DBStationName ="";
    final String stationName;
    DBAdapter db = new DBAdapter(mContext);

    db.open();
    Cursor c = db.getStationName(stationID);


    if (c.moveToFirst())
    {
        do{
            DBStationName = c.getString(1);

        } while (c.moveToNext());
    }

    c.close();
    db.close();

    stationName = DBStationName;

    if (setFlag.equalsIgnoreCase("from"))
     {
          FromStation = Integer.parseInt(stationID);
     }

     else if (setFlag.equalsIgnoreCase("to"))
```

Figure 6.7: Section of Code for JavaScript Interface.

After the correct stations have been selected, the search button will invoke the RouteResult page to display the correct route information. The toStationId and fromStationId will be used to query the routedata table for route information. The route information will then be extracted from the database, processed and displayed to the user in the correct format. The figure below shows the route result page.



Fare : RM1.60

Travel Time : ~15.03 mins

From : One Utama

TTDI

Seksyen 16

Bukit Kiara

Pusat Bandar Damansara

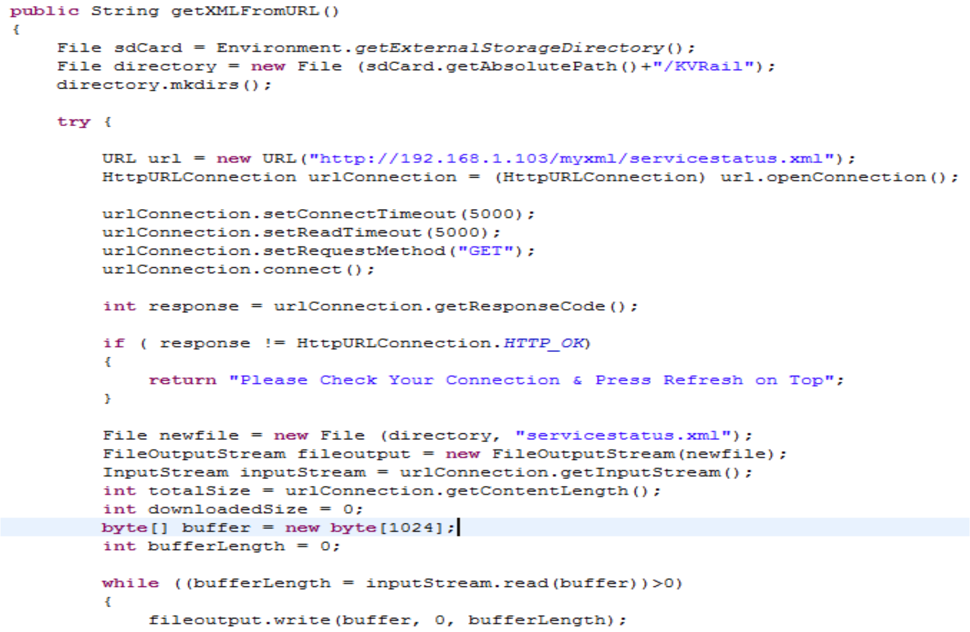
Semantan

To : KL Sentral

Figure 6.8: Screenshot of Route Result Interface.

6.1.3 Live Service Status Module

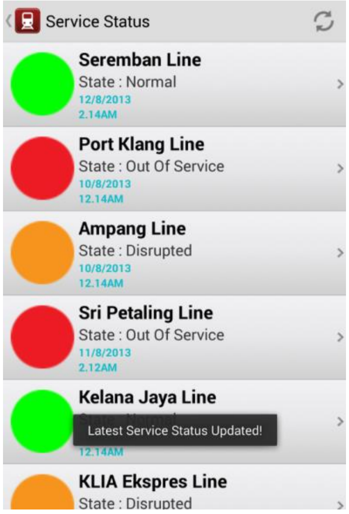
The function of live service status module is to update and display the latest service status to the user. It is implemented by downloading; saving and parsing XML file from a web server. When internet connection is active, the XML file will be downloaded and saved in the SD Card of the phone. The content of XML will be parsed to show the correct data to the user. If internet connection is not available, the application will show the last-updated service status saved in the SD Card. The figure below shows the section of code implemented to grab the latest XML file from the web server.



```
public String getXMLFromURL()
{
    File sdCard = Environment.getExternalStorageDirectory();
    File directory = new File (sdCard.getAbsolutePath()+"/KVRail");
    directory.mkdirs();

    try {

        URL url = new URL("http://192.168.1.103/myxml/servicestatus.xml");
        HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();

        urlConnection.setConnectTimeout(5000);
        urlConnection.setReadTimeout(5000);
        urlConnection.setRequestMethod("GET");
        urlConnection.connect();

        int response = urlConnection.getResponseCode();

        if ( response != HttpURLConnection.HTTP_OK)
        {
            return "Please Check Your Connection & Press Refresh on Top";
        }

        File newfile = new File (directory, "servicestatus.xml");
        FileOutputStream fileoutput = new FileOutputStream(newfile);
        InputStream inputStream = urlConnection.getInputStream();
        int totalSize = urlConnection.getContentLength();
        int downloadedSize = 0;
        byte[] buffer = new byte[1024];
        int bufferLength = 0;

        while ((bufferLength = inputStream.read(buffer))>0)
        {
            fileoutput.write(buffer, 0, bufferLength);
```

Figure 6.9: Section of Code to Grab XML From Web Server.

There will be a refresh button on the top right corner to update the live service status. The service status interface is shown in the diagram below.



Service Status

Seremban Line
State : Normal
12/8/2013
2.14AM

Port Klang Line
State : Out Of Service
10/8/2013
12.14AM

Ampang Line
State : Disrupted
10/8/2013
12.14AM

Sri Petaling Line
State : Out Of Service
11/8/2013
2.12AM

Kelana Jaya Line
State : Normal
12.14AM

Latest Service Status Updated!

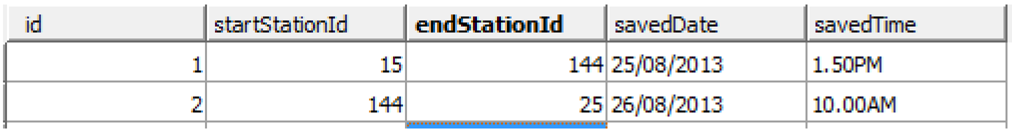
KLIA Ekspres Line
State : Disrupted

Figure 6.10: Screenshot of Service Status Interface.

6.1.4 Route Saving & Sharing Module

The route saving function allows user to save their preferred route for reference in future and the sharing function allows user to share the route result formatted into text version to their friends/family via SMS/Email. Other than sharing route result, station information can be shared too.

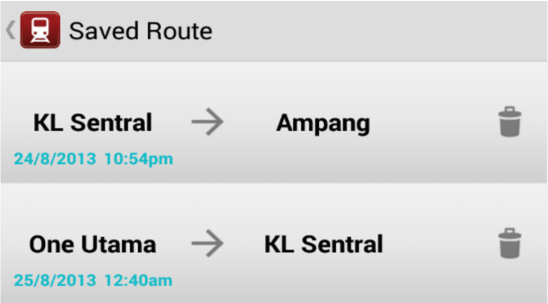
The pre-condition for route saving is that the user must be have performed route selection process. Once the user is in the route result page, it can click on the save button on top right corner to save the route. Once clicked, the button will insert the toStationId and fromStationId, date and time to the savedroute database. If the database transaction is successful, a notification message will show “Successfully Saved!” The table structure of savedroute in shown in the figure below.



id	startStationId	endStationId	savedDate	savedTime
1	15	144	25/08/2013	1.50PM
2	144	25	26/08/2013	10.00AM

Figure 6.11: Structure of savedroute table.

The saved route can be retrieved from the main page by clicking on “Saved Route” icon. Once clicked, all the saved route will be retrieved from the savedroute table and displayed to the user. User can choose to click on the route to view the result or delete button to erase the route from database. The figure below shows the saved route list.

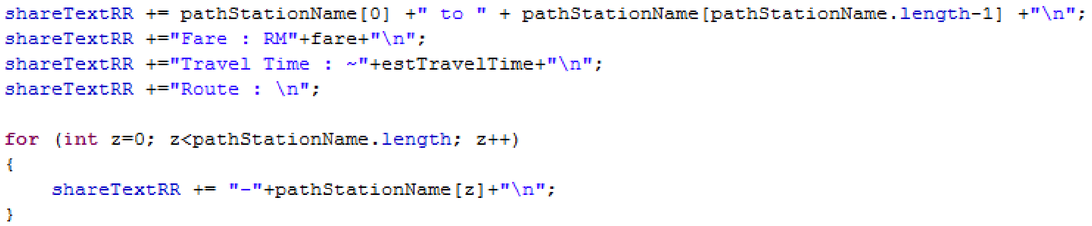


KL Sentral → Ampang
24/8/2013 10:54pm

One Utama → KL Sentral
25/8/2013 12:40am

Figure 6.12: Screenshot of Saved Route Interface.

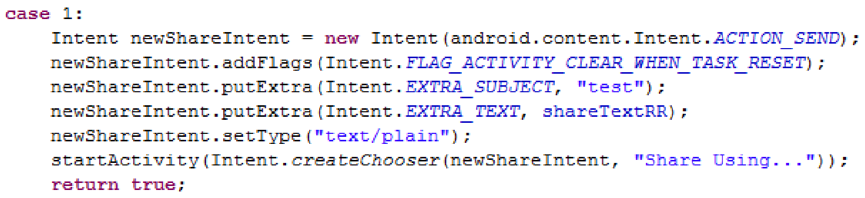
The pre-condition of sharing function is same as above; user must have landed on route result page before he/she can share the route result. The sharing function is achieved by inserting a String variable to store and format the route result. Section of code below shows the shareText variable being manipulated to show the correct format of text.



```
shareTextRR += pathStationName[0] +" to " + pathStationName[pathStationName.length-1] +"\n";
shareTextRR +="Fare : RM"+fare+"\n";
shareTextRR +="Travel Time : ~"+estTravelTime+"\n";
shareTextRR +="Route : \n";

for (int z=0; z<pathStationName.length; z++)
{
    shareTextRR += "-"+pathStationName[z]+"\n";
}
```

Figure 6.13: Section of Code for Share Text.

User can share by clicking on the share button on the top right corner of the application. Once clicked, the application will send Intent to all the other application along with shareText content. Section of code below shows the starting of new Intent and embedding of shareText into intent.



```
case 1:
    Intent newShareIntent = new Intent(android.content.Intent.ACTION_SEND);
    newShareIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_WHEN_TASK_RESET);
    newShareIntent.putExtra(Intent.EXTRA_SUBJECT, "test");
    newShareIntent.putExtra(Intent.EXTRA_TEXT, shareTextRR);
    newShareIntent.setType("text/plain");
    startActivity(Intent.createChooser(newShareIntent, "Share Using..."));
    return true;
```

Figure 6.14: Section of Code for Starting New Share Intent.

The information will then be retrieved by the other application for sharing. The figure below shows the share action button, the share provider application list and the SMS on route result that has been received.

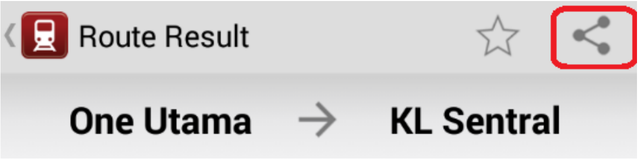
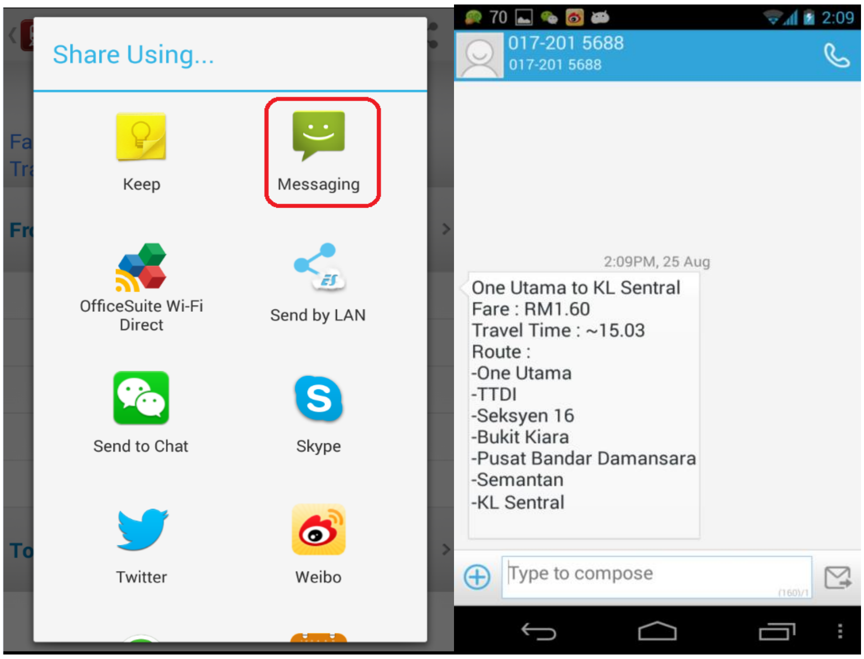


Figure 6.15: Screenshot of Share Button.

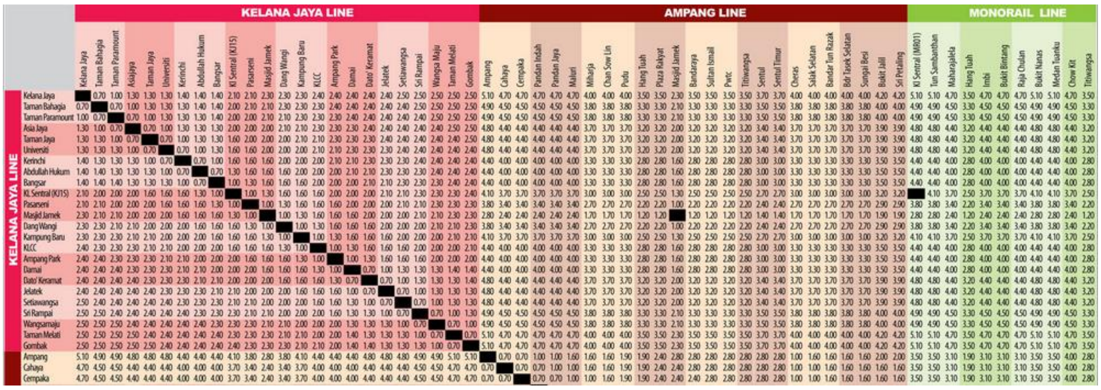


One Utama to KL Sentral
Fare : RM1.60
Travel Time : ~15.03
Route :
-One Utama
-TTDI
-Seksyen 16
-Bukit Kiara
-Pusat Bandar Damansara
-Semantan
-KL Sentral

Figure 6.16: Screenshot of Share Action Provider & SMS Received.

6.1.5 Fare Display Module

The fare data of the route must be inserted manually because there is no integration between KTM, KLIA Ekspres, KLIA Transit with RapidKL operated route. Other than that, the fare for MRT line is currently unavailable. The calculation of fare varies by operator and there is no standardized algorithm to generate them. Thus it involves manual work of checking the route against the fare table and insert into the database. The figure below shows the fare table of the RapidKL operated route.



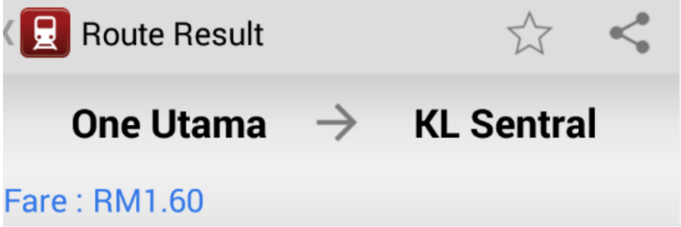
		KELANA JAYA LINE																							
		Kelana Jaya	Taman Bahagia	Taman Paramount	Asiajaya	Taman Jaya	Universiti	Kerinchi	Abdullah Hukum	Bangsar	Kl Sentral (KJ15)	Pasarseni	Masjid Jamek	Dang Wangi	Kampung Baru	KLCC	Ampang Park	Damai	Dato' Keramat	Jelatek	Setiawangsa	Sri Rampai	Wangsa Maju	Taman Melati	Gombak
KELANA JAYA LINE	Kelana Jaya	■	0.70	1.00	1.30	1.30	1.30	1.40	1.40	1.40	2.10	2.10	2.30	2.30	2.30	2.40	2.40	2.40	2.40	2.40	2.50	2.50	2.50	2.50	2.50
	Taman Bahagia	0.70	■	0.70	1.00	1.30	1.30	1.30	1.40	1.40	2.00	2.10	2.10	2.30	2.30	2.30	2.40	2.40	2.40	2.40	2.40	2.50	2.50	2.50	2.50
	Taman Paramount	1.00	0.70	■	0.70	1.00	1.30	1.30	1.30	1.40	2.00	2.00	2.10	2.10	2.30	2.30	2.30	2.40	2.40	2.40	2.40	2.40	2.50	2.50	2.50
	Asia Jaya	1.30	1.00	0.70	■	0.70	1.00	1.30	1.30	1.30	2.00	2.00	2.00	2.10	2.10	2.30	2.30	2.30	2.40	2.40	2.40	2.40	2.40	2.50	2.50
	Taman Jaya	1.30	1.30	1.00	0.70	■	0.70	1.00	1.30	1.30	1.60	2.00	2.00	2.00	2.10	2.10	2.30	2.30	2.30	2.40	2.40	2.40	2.40	2.40	2.50
	Universiti	1.30	1.30	1.30	1.00	0.70	■	0.70	1.00	1.30	1.60	1.60	2.00	2.00	2.00	2.10	2.10	2.30	2.30	2.30	2.40	2.40	2.40	2.40	2.40
	Kerinchi	1.40	1.30	1.30	1.30	1.00	0.70	■	0.70	1.00	1.60	1.60	1.60	2.00	2.00	2.00	2.10	2.10	2.30	2.30	2.30	2.40	2.40	2.40	2.40
	Abdullah Hukum	1.40	1.40	1.30	1.30	1.30	1.00	0.70	■	0.70	1.30	1.60	1.60	1.60	2.00	2.00	2.00	2.10	2.10	2.30	2.30	2.30	2.40	2.40	2.40
	Bangsar	1.40	1.40	1.40	1.30	1.30	1.30	1.00	0.70	■	1.00	1.30	1.60	1.60	1.60	2.00	2.00	2.00	2.10	2.10	2.30	2.30	2.30	2.40	2.40
	KL Sentral (KJ15)	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	1.00	■	1.00	1.30	1.60	1.60	1.60	2.00	2.00	2.00	2.10	2.10	2.30	2.30	2.30	2.40
	Pasarseni	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	1.00	■	1.00	1.30	1.60	1.60	1.60	2.00	2.00	2.00	2.10	2.10	2.30	2.30	2.30
	Masjid Jamek	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	1.00	■	1.00	1.30	1.60	1.60	1.60	2.00	2.00	2.00	2.10	2.10	2.30	2.30
	Dang Wangi	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	1.00	■	1.00	1.30	1.60	1.60	1.60	2.00	2.00	2.00	2.10	2.10	2.30
	Kampung Baru	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	1.00	■	1.00	1.30	1.60	1.60	1.60	2.00	2.00	2.00	2.10	2.10
	KLCC	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	1.00	■	1.00	1.30	1.60	1.60	1.60	2.00	2.00	2.00	2.10
	Ampang Park	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	1.00	■	1.00	1.30	1.60	1.60	1.60	2.00	2.00	2.00
	Damai	2.40	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	1.00	■	0.70	1.00	1.30	1.30	1.30	1.40	1.40
	Dato' Keramat	2.40	2.40	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.30	0.70	■	0.70	1.00	1.30	1.30	1.30	1.40
	Jelatek	2.40	2.40	2.40	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.60	1.00	0.70	■	0.70	1.00	1.30	1.30	1.30
	Setiawangsa	2.50	2.40	2.40	2.40	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.60	1.30	1.00	0.70	■	0.70	1.00	1.30	1.30
	Sri Rampai	2.50	2.50	2.40	2.40	2.40	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.60	1.30	1.30	1.00	0.70	■	0.70	1.00	1.30
	Wangsamaju	2.50	2.50	2.50	2.40	2.40	2.40	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	2.00	1.30	1.30	1.30	1.00	0.70	■	0.70	1.00
	Taman Melati	2.50	2.50	2.50	2.50	2.40	2.40	2.40	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	2.00	1.40	1.30	1.30	1.30	1.00	0.70	■	0.70
	Gombak	2.50	2.50	2.50	2.50	2.50	2.40	2.40	2.40	2.40	2.40	2.30	2.30	2.30	2.10	2.10	2.00	1.40	1.40	1.30	1.30	1.30	1.00	0.70	■
	Ampang	5.10	4.90	4.90	4.80	4.80	4.80	4.40	4.40	4.40	4.10	3.80	2.80	3.80	4.10	4.40	4.40	4.40	4.80	4.80	4.80	4.90	4.90	5.10	5.10
	Cahaya	4.70	4.50	4.50	4.40	4.40	4.40	4.00	4.00	4.00	3.70	3.40	2.40	3.40	3.70	4.00	4.00	4.00	4.40	4.40	4.40	4.50	4.50	4.70	4.70
	Cempaka	4.70	4.50	4.50	4.40	4.40	4.40	4.00	4.00	4.00	3.70	3.40	2.40	3.40	3.70	4.00	4.00	4.00	4.40	4.40	4.40	4.50	4.50	4.70	4.70

		AMPANG LINE																								
		Ampang	Cahaya	Cempaka	Pandan Indah	Pandan Jaya	Maluri	Miharja	Chan Sow Lin	Pudu	Hang Tuah	Plaza Rakyat	Masjid Jamek	Bandaraya	Sultan Ismail	Pwtc	Titiwangsa	Sentul	Sentul Timur	Cheras	Salak Selatan	Bandar Tun Razak	Bdr Tasek Selatan	Sungai Besi	Bukit Jalil	Sri Petaling
KELANA JAYA LINE	Kelana Jaya	5.10	4.70	4.70	4.70	4.70	4.70	4.00	4.00	4.00	3.50	3.50	2.30	3.50	3.50	3.50	3.50	3.70	3.70	4.00	4.00	4.00	4.00	4.00	4.20	4.20
	Taman Bahagia	4.90	4.50	4.50	4.50	4.50	4.50	3.80	3.80	3.80	3.30	3.30	2.10	3.30	3.30	3.30	3.30	3.50	3.50	3.80	3.80	3.80	3.80	3.80	4.00	4.00
	Taman Paramount	4.90	4.50	4.50	4.50	4.50	4.50	3.80	3.80	3.80	3.30	3.30	2.10	3.30	3.30	3.30	3.30	3.50	3.50	3.80	3.80	3.80	3.80	3.80	4.00	4.00
	Asia Jaya	4.80	4.40	4.40	4.40	4.40	4.40	3.70	3.70	3.70	3.20	3.20	2.00	3.20	3.20	3.20	3.20	3.40	3.40	3.70	3.70	3.70	3.70	3.70	3.90	3.90
	Taman Jaya	4.80	4.40	4.40	4.40	4.40	4.40	3.70	3.70	3.70	3.20	3.20	2.00	3.20	3.20	3.20	3.20	3.40	3.40	3.70	3.70	3.70	3.70	3.70	3.90	3.90
	Universiti	4.80	4.40	4.40	4.40	4.40	4.40	3.70	3.70	3.70	3.20	3.20	2.00	3.20	3.20	3.20	3.20	3.40	3.40	3.70	3.70	3.70	3.70	3.70	3.90	3.90
	Kerinchi	4.40	4.00	4.00	4.00	4.00	4.00	3.30	3.30	3.30	2.80	2.80	1.60	2.80	2.80	2.80	2.80	3.00	3.00	3.30	3.30	3.30	3.30	3.30	3.50	3.50
	Abdullah Hukum	4.40	4.00	4.00	4.00	4.00	4.00	3.30	3.30	3.30	2.80	2.80	1.60	2.80	2.80	2.80	2.80	3.00	3.00	3.30	3.30	3.30	3.30	3.30	3.50	3.50
	Bangsar	4.40	4.00	4.00	4.00	4.00	4.00	3.30	3.30	3.30	2.80	2.80	1.60	2.80	2.80	2.80	2.80	3.00	3.00	3.30	3.30	3.30	3.30	3.30	3.50	3.50
	KL Sentral (KJ15)	4.10	3.70	3.70	3.70	3.70	3.70	3.00	3.00	3.00	2.50	2.50	1.30	2.50	2.50	2.50	2.50	2.70	2.70	3.00	3.00	3.00	3.00	3.00	3.20	3.20
	Pasarseni	3.80	3.40	3.40	3.40	3.40	3.40	2.70	2.70	2.70	2.20	2.20	1.00	2.20	2.20	2.20	2.20	2.40	2.40	2.70	2.70	2.70	2.70	2.70	2.90	2.90
	Masjid Jamek	2.80	2.40	2.40	2.40	2.40	2.40	1.70	1.70	1.70	1.20	1.20	■	1.20	1.20	1.20	1.20	1.40	1.40	1.70	1.70	1.70	1.70	1.70	1.90	1.90
	Dang Wangi	3.80	3.40	3.40	3.40	3.40	3.40	2.70	2.70	2.70	2.20	2.20	1.00	2.20	2.20	2.20	2.20	2.40	2.40	2.70	2.70	2.70	2.70	2.70	2.90	2.90
	Kampung Baru	4.10	3.70	3.70	3.70	3.70	3.70	3.00	3.00	3.00	2.50	2.50	1.30	2.50	2.50	2.50	2.50	2.70	2.70	3.00	3.00	3.00	3.00	3.00	3.20	3.20
	KLCC	4.40	4.00	4.00	4.00	4.00	4.00	3.30	3.30	3.30	2.80	2.80	1.60	2.80	2.80	2.80	2.80	3.00	3.00	3.30	3.30	3.30	3.30	3.30	3.50	3.50
	Ampang Park	4.40	4.00	4.00	4.00	4.00	4.00	3.30	3.30	3.30	2.80	2.80	1.60	2.80	2.80	2.80	2.80	3.00	3.00	3.30	3.30	3.30	3.30	3.30	3.50	3.50
	Damai	4.40	4.00	4.00	4.00	4.00	4.00	3.30	3.30	3.30	2.80	2.80	1.60	2.80	2.80	2.80	2.80	3.00	3.00	3.30	3.30	3.30	3.30	3.30	3.50	3.50
	Dato' Keramat	4.80	4.40	4.40	4.40	4.40	4.40	3.70	3.70	3.70	3.20	3.20	2.00	3.20	3.20	3.20	3.20	3.40	3.40	3.70	3.70	3.70	3.70	3.70	3.90	3.90
	Jelatek	4.80	4.40	4.40	4.40	4.40	4.40	3.70	3.70	3.70	3.20	3.20	2.00	3.20	3.20	3.20	3.20	3.40	3.40	3.70	3.70	3.70	3.70	3.70	3.90	3.90
	Setiawangsa	4.80	4.40	4.40	4.40	4.40	4.40	3.70	3.70	3.70	3.20	3.20	2.00	3.20	3.20	3.20	3.20	3.40	3.40	3.70	3.70	3.70	3.70	3.70	3.90	3.90
	Sri Rampai	4.90	4.50	4.50	4.50	4.50	4.50	3.80	3.80	3.80	3.30	3.30	2.10	3.30	3.30	3.30	3.30	3.50	3.50	3.80	3.80	3.80	3.80	3.80	4.00	4.00
	Wangsamaju	4.90	4.50	4.50	4.50	4.50	4.50	3.80	3.80	3.80	3.30	3.30	2.10	3.30	3.30	3.30	3.30	3.50	3.50	3.80	3.80	3.80	3.80	3.80	4.00	4.00
	Taman Melati	5.10	4.70	4.70	4.70	4.70	4.70	4.00	4.00	4.00	3.50	3.50	2.30	3.50	3.50	3.50	3.50	3.70	3.70	4.00	4.00	4.00	4.00	4.00	4.20	4.20
	Gombak	5.10	4.70	4.70	4.70	4.70	4.70	4.00	4.00	4.00	3.50	3.50	2.30	3.50	3.50	3.50	3.50	3.70	3.70	4.00	4.00	4.00	4.00	4.00	4.20	4.20
	Ampang	■	0.70	0.70	1.00	1.00	1.60	1.60	1.60	1.90	1.90	2.40	2.80	2.80	2.80	2.80	2.80	2.80	2.80	1.00	1.60	1.60	1.60	1.60	2.00	2.00
	Cahaya	0.70	■	0.70	0.70	1.00	1.00	1.60	1.60	1.90	1.90	2.40	2.40	2.80	2.80	2.80	2.80	2.80	2.80	1.00	1.00	1.60	1.60	1.60	1.60	1.60
	Cempaka	0.70	0.70	■	0.70	0.70	1.00	1.00	1.60	1.90	1.90	2.40	2.40	2.40	2.80	2.80	2.80	2.80	2.80	1.00	1.00	1.60	1.60	1.60	1.60	1.60

		MONORAIL LINE										
		Kl Sentral (MR01)	Tun Sambanthan	Maharajalela	Hang Tuah	Imbi	Bukit Bintang	Raja Chulan	Bukit Nanas	Medan Tuanku	Chow Kit	Titiwangsa
KELANA JAYA LINE	Kelana Jaya	5.10	5.10	4.70	3.50	4.70	4.70	4.70	5.10	5.10	4.70	3.50
	Taman Bahagia	4.90	4.90	4.50	3.30	4.50	4.50	4.50	4.90	4.90	4.50	3.30
	Taman Paramount	4.90	4.90	4.50	3.30	4.50	4.50	4.50	4.90	4.90	4.50	3.30
	Asia Jaya	4.80	4.80	4.40	3.20	4.40	4.40	4.40	4.80	4.80	4.40	3.20
	Taman Jaya	4.80	4.80	4.40	3.20	4.40	4.40	4.40	4.80	4.80	4.40	3.20
	Universiti	4.80	4.80	4.40	3.20	4.40	4.40	4.40	4.80	4.80	4.40	3.20
	Kerinchi	4.40	4.40	4.00	2.80	4.00	4.00	4.00	4.40	4.40	4.00	2.80
	Abdullah Hukum	4.40	4.40	4.00	2.80	4.00	4.00	4.00	4.40	4.40	4.00	2.80
	Bangsar	4.40	4.40	4.00	2.80	4.00	4.00	4.00	4.40	4.40	4.00	2.80
	KL Sentral (KJ15)	■	4.10	3.70	2.50	3.70	3.70	3.70	4.10	4.10	3.70	2.50
	Pasarseni	3.80	3.80	3.40	2.20	3.40	3.40	3.40	3.80	3.80	3.40	2.20
	Masjid Jamek	2.80	2.80	2.40	1.20	2.40	2.40	2.40	2.80	2.80	2.40	1.20
	Dang Wangi	3.80	3.80	3.40	2.20	3.40	3.40	3.40	3.80	3.80	3.40	2.20
	Kampung Baru	4.10	4.10	3.70	2.50	3.70	3.70	3.70	4.10	4.10	3.70	2.50
	KLCC	4.40	4.40	4.00	2.80	4.00	4.00	4.00	4.40	4.40	4.00	2.80
	Ampang Park	4.40	4.40	4.00	2.80	4.00	4.00	4.00	4.40	4.40	4.00	2.80
	Damai	4.40	4.40	4.00	2.80	4.00	4.00	4.00	4.40	4.40	4.00	2.80
	Dato' Keramat	4.80	4.80	4.40	3.20	4.40	4.40	4.40	4.80	4.80	4.40	3.20
	Jelatek	4.80	4.80	4.40	3.20	4.40	4.40	4.40	4.80	4.80	4.40	3.20
	Setiawangsa	4.80	4.80	4.40	3.20	4.40	4.40	4.40	4.80	4.80	4.40	3.20
	Sri Rampai	4.90	4.90	4.50	3.30	4.50	4.50	4.50	4.90	4.90	4.50	3.30
	Wangsamaju	4.90	4.90	4.50	3.30	4.50	4.50	4.50	4.90	4.90	4.50	3.30
	Taman Melati	5.10	5.10	4.70	3.50	4.70	4.70	4.70	5.10	5.10	4.70	3.50
	Gombak	5.10	5.10	4.70	3.50	4.70	4.70	4.70	5.10	5.10	4.70	3.50
	Ampang	3.50	3.50	3.10	1.90	3.10	3.10	3.10	3.50	3.50	4.00	2.80
	Cahaya	3.50	3.50	3.10	1.90	3.10	3.10	3.10	3.50	3.50	4.00	2.80
	Cempaka	3.50	3.50	3.10	1.90	3.10	3.10	3.10	3.50	3.50	4.00	2.80

Figure 6.17: RapidKL Fare Table.

The fare table will be inserted into the fare column of routedata table. The fare will be retrieved and shown to the user when the user searches for a route. The figure below shows the fare display on route result page.



One Utama → KL Sentral

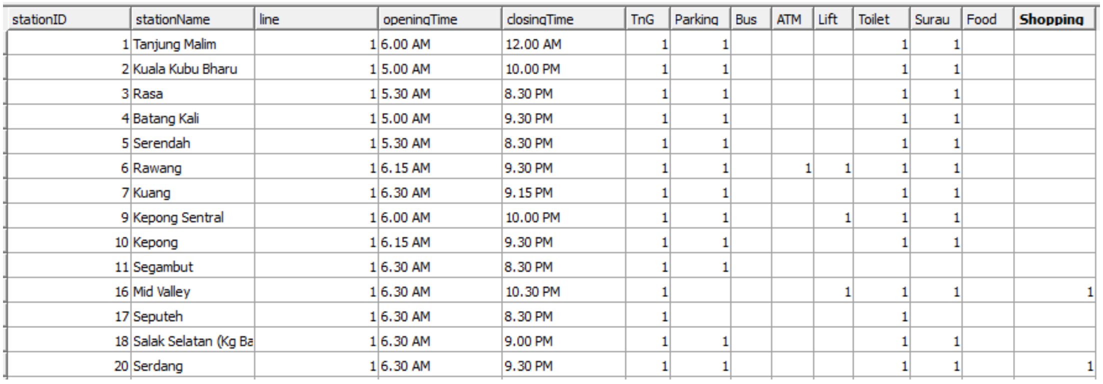
Fare : RM1.60

Figure 6.18: Screenshot of Fare Display.

6.1.6 Station Information & Nearby Attractions Module

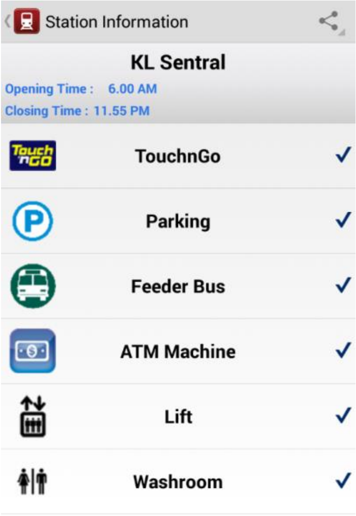
The station information function shows the opening time, closing time and the facilities available in a station and the nearby attractions function shows a list of attractions around a station.

The station information is implemented by storing the relevant data into the stationinfo database table. When the user selects a station, the station information of that particular station will be queried from database and displayed to the user. The figure below shows the stationinfo table and the station information page.



stationID	stationName	line	openingTime	closingTime	TnG	Parking	Bus	ATM	Lift	Toilet	Surau	Food	Shopping
1	Tanjung Malim	1	6.00 AM	12.00 AM	1	1				1	1		
2	Kuala Kubu Bharu	1	5.00 AM	10.00 PM	1	1				1	1		
3	Rasa	1	5.30 AM	8.30 PM	1	1				1	1		
4	Batang Kali	1	5.00 AM	9.30 PM	1	1				1	1		
5	Serendah	1	5.30 AM	8.30 PM	1	1				1	1		
6	Rawang	1	6.15 AM	9.30 PM	1	1		1	1	1	1		
7	Kuang	1	6.30 AM	9.15 PM	1	1				1	1		
9	Kepong Sentral	1	6.00 AM	10.00 PM	1	1			1	1	1		
10	Kepong	1	6.15 AM	9.30 PM	1	1				1	1		
11	Segambut	1	6.30 AM	8.30 PM	1	1							
16	Mid Valley	1	6.30 AM	10.30 PM	1				1	1	1		1
17	Seputeh	1	6.30 AM	8.30 PM	1					1			
18	Salak Selatan (Kg Ba	1	6.30 AM	9.00 PM	1	1				1	1		
20	Serdang	1	6.30 AM	9.30 PM	1	1				1	1		1

Figure 6.19: Structure of stationinfo table.



KL Sentral

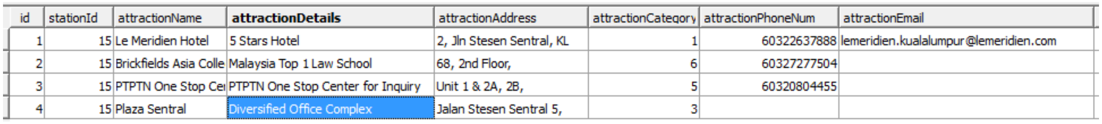
Opening Time : 6.00 AM

Closing Time : 11.55 PM

	TouchnGo	✓
	Parking	✓
	Feeder Bus	✓
	ATM Machine	✓
	Lift	✓
	Washroom	✓

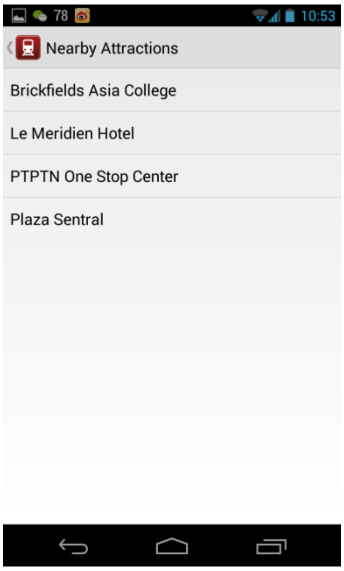
Figure 6.20: Screenshot of Station Information Interface.

The nearby attractions function is implemented in a similar way. The nearby attraction of a station and their details are inserted into stationattractions table. When the user clicks on a station, the stationId will be used to query the correct station attraction. The result will be displayed to the user. The figure below shows the stationattractions table and the station attraction list for a station.



id	stationId	attractionName	attractionDetails	attractionAddress	attractionCategory	attractionPhoneNum	attractionEmail
1	15	Le Meridien Hotel	5 Stars Hotel	2, Jln Stesen Sentral, KL	1	60322637888	lemeridien.kualalumpur@lemeridien.com
2	15	Brickfields Asia Colle	Malaysia Top 1 Law School	68, 2nd Floor,	6	60327277504	
3	15	PTPTN One Stop Cei	PTPTN One Stop Center for Inquiry	Unit 1 & 2A, 2B,	5	60320804455	
4	15	Plaza Sentral	Diversified Office Complex	Jalan Stesen Sentral 5,	3		

Figure 6.21: Structure of stationattractions table.



Nearby Attractions

Brickfields Asia College

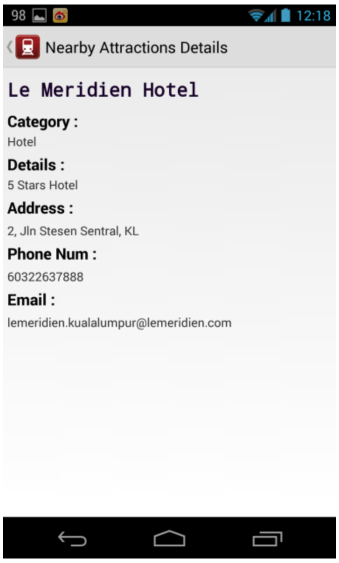
Le Meridien Hotel

PTPTN One Stop Center

Plaza Sentral

Figure 6.22: Nearby Attractions List Interface.

When the user clicks on the attraction name, the details of the attraction will be shown to the user. The interface of attraction details page is shown in the figure below.



Nearby Attractions Details

Le Meridien Hotel

Category :
Hotel

Details :
5 Stars Hotel

Address :
2, Jln Stesen Sentral, KL

Phone Num :
60322637888

Email :
lemeridien.kualalumpur@lemeridien.com

Figure 6.23: Screenshot of Nearby Attraction Details Interface.

6.2 Unit Testing

During the development phase, unit testing will be performed to discover defects and solve them. Unit testing is carried out to test individual program units or methods. It is focused on testing the functionality of an objects or method to ensure they performed as intended. Unit testing will often be carried out after the completion of one or more methods. It will be performed for many times during the course of development to ensure everything is working as planned. If the test fails, the problem causing it will be identified, solved and retested until success. The unit testing that has been completed will be shown in the section below.

6.2.1.1 Route Calculation Module

Test Case	Test Execution Steps	Expected Result	Pass/Fail
Database Connection	1. Run the application. 2. Look for error message in console window.	Connection to database should be successful without error.	Pass
Store station as vertex object.	1. Run the application.	Station information retrieved from database and stored as vertex object and displayed.	Pass
Import edges for vertex object.	1. Run the application.	Edges for respective vertex correctly stored and displayed.	Pass
Finding Shortest Path	1. Input the source station. 2. Input the destination station. 3. Run the application.	The correct path should be displayed to user.	Pass
Fare Display	1. Input source station. 2. Input destination station. 3. Run the application. 4. Validate fare against actual fare table.	The correct fare for path should be displayed correctly to user.	Pass

Estimated Travel Time	1. Input source station. 2. Input destination station. 3. Run the application. 4. Validate the estimated travel time against the pre-calculated travel time according to travel time table.	The correct estimated travel time should be displayed correctly to the user.	Pass
Path Line	1. Run the application. 2. Validate the line for each path and interchanges according to route map.	The rail line for each station and interchange station must be displayed correctly.	Pass
Export Result to CSV	1. Run the application. 2. Open result.csv located at root directory.	All the route information must be correctly stored and formatted.	Pass

6.2.2 Main Screen

Test Case	Test Execution Steps	Expected Result	Pass/Fail
Route Planner Button	1. Click on the button.	Link to the route planner page.	Pass
Service Status Button	1. Click on the button.	Link to service status page.	Pass
Station Information Button	1. Click on the button.	Link to select station page.	Pass
Nearby Attractions Button	1. Click on the button.	Link to select station page.	Pass
Saved Route Button	1. Click on the button.	Link to saved route page.	Pass

6.2.3 Route Planner Module

Test Case	Test Execution Steps	Expected Result	Pass/Fail
From Text Box	1. Click on the button.	Link to select station page.	Pass
To Text Box	1. Click on the button.	Link to select station page.	Pass
Reset Button	1. Select from and to station. 2. Click on the reset button.	From and To station should be defaulted to empty, the highlight icon on map should be hidden.	Pass
Search Button	1. Click on the button.	Show no stations selected error.	Pass
Search Button	1. Select an origin station. 2. Click on Search Button.	Shows choose destination station error.	Pass
Search Button	1. Select a destination station. 2. Click on Search Button.	Shows choose origin station error.	Pass
Search Button	1. Select an origin station. 2. Select a destination station. 3. Click on Search Button.	Link to route result page.	Pass
Information Button	1. Click on the button.	Link to Route Planner Legend page.	Pass
Swap Button	1. Select from and to station. 2. Click on swap button.	From and To station should be swapped.	Pass
Station Information Button	1. Click on the desired station on map. 2. Click Info button from the sub menu.	Link to the correct station information page.	Pass
Close Button	1. Click on the desired station on map. 2. Click the close button on sub menu.	The submenu will be closed.	Pass
Map Zoom In & Out	1. Pinch/double tap on the map.	Map should be resized according to gesture.	Pass

Share Button	1. Click on the share button. 2. Select application to share.	Route result shared to other application	Pass
Save Button	1. Click on Save Button.	A toast message will appear to inform successful/failed saving.	Pass
Route Result	1. Select from station. 2. Select to station. 3. Click Search.	The route result should be correctly listed.	Pass
From/Interchange/To station click.	1. Click on from/interchange/to station.	Link to station information page.	Pass

6.2.4 Service Status Module

Test Case	Test Execution Steps	Expected Result	Pass/Fail
Get XML from Web Server	1. Click on “Service Status” button.	Latest XML downloaded and saved in SD Card.	Pass
Parse XML	1. Click on “Service Status” button.	XML content should be extracted and displayed correctly.	Pass
Line Item Click.	1. Click on the line information displayed on screen.	Link to line details page.	Pass
Refresh Button	1. Click on refresh button on top right corner.	Activity will be restart. Latest XML will be downloaded and displayed.	Pass

6.2.5 Station Information & Nearby Attractions Module

Test Case	Test Execution Steps	Expected Result	Pass/Fail
Database Connection	1. Run the application.	Database connection must be successful without error.	Pass
Get station information from database.	1. Set the station name. 2. Run the application.	Database should return cursor containing data for selected station.	Pass
Display station information.	1. Set the station name. 2. Run the application.	Application should display the correct information in correct column.	Pass
Get station attractions from database.	1. Set the station name. 2. Run the application.	Database should return cursor containing nearby attractions data for selected station.	Pass
Display station attractions.	1. Set the station name. 2. Run the application.	Application should display all the correct attractions in correct layout.	Pass

6.2.6 Saved Route Module

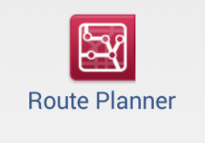
Test Case	Test Execution Steps	Expected Result	Pass/Fail
Database Connection	1. Run the application.	Database connection must be successful without error.	Pass
Delete Button	1. Click on delete button.	Application must be able to show “button clicked” message.	Pass

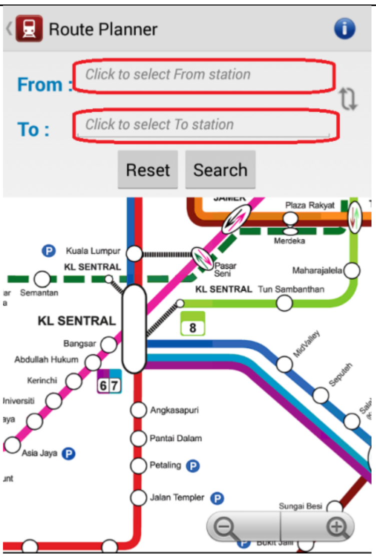
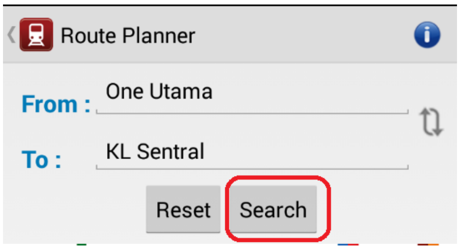
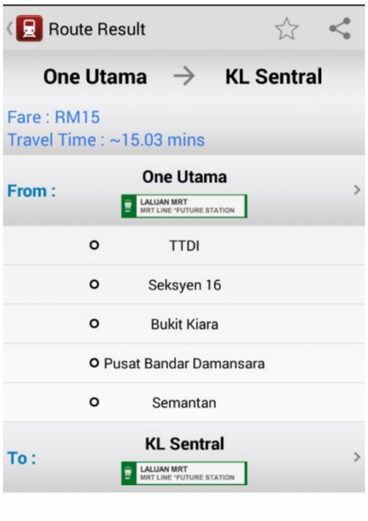
Get Saved Route from Database.	1. Run the application.	Application should display all the saved route from database.	Pass
Delete saved route from Database.	1. Run the application.	Application should display error in deleting saved route.	Pass
Delete saved route from Database.	1. Set the id of saved route to be deleted. 2. Run the application.	Database entry for the saved route should be deleted successfully.	Pass

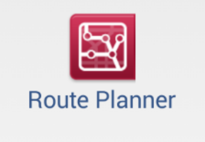
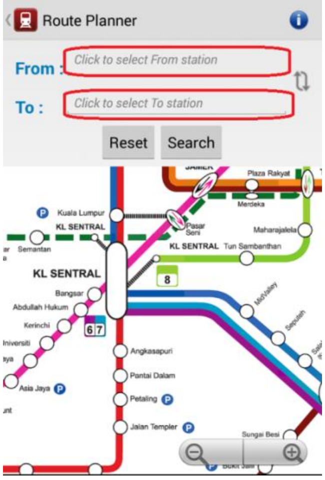
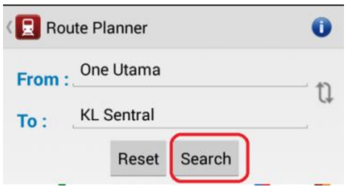
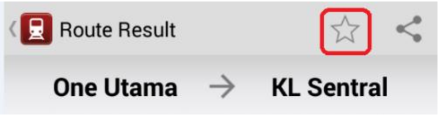
6.3 Integration Testing

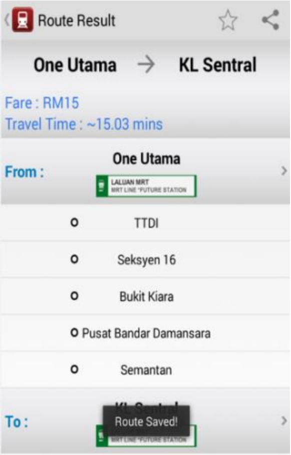
Integration testing is performed after unit testing. It is performed when individual modules are combined as a group. Integration testing is crucial to discover error when different modules are integrated to perform functionalities.

6.3.1 Test Cases

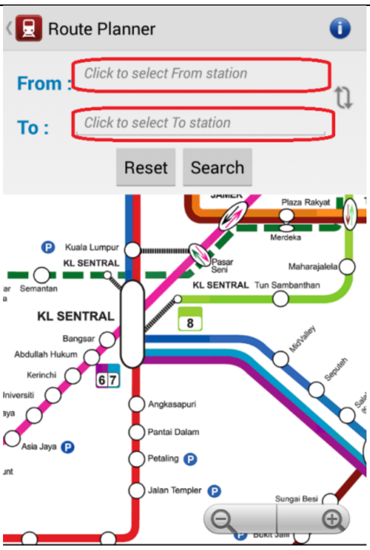
Test Case	To search for a route from Origin station to Destination station and display the route.
Test **Performed**	Steps 1. Click the “Route Planner” icon from the main screen.  2. Select the To & From station by tapping the empty column beside text. Or tap on the empty circle beside station name in the map.

	 Route Planner From : Click to select From station To : Click to select To station Reset Search Plaza Rakyat Merdeka Kuala Lumpur KL SENTRAL Pasar Seni Maharajalela Semantan KL SENTRAL Tun Sambanthan 8 Bangsar Abdullah Hukum Kerinchi 6 7 Universiti Asia Jaya Angkasapuri Pantai Dalam Petaling Jalan Templer MidValley Seputeh Sungai Besi Bukit Jalil 3. After selecting both the To & From station, click on the Search Button.  Route Planner From : One Utama To : KL Sentral Reset Search
Expected Results	When both the From & To station has been selected, the clicking of search button will link to route result page that will display the correct route information, fare and estimated travel time.  Route Result One Utama → KL Sentral Fare : RM15 Travel Time : ~15.03 mins From : One Utama LALUAN MRT MRT LINE *FUTURE STATION TTDI Seksyen 16 Bukit Kiara Pusat Bandar Damansara Semantan To : KL Sentral LALUAN MRT MRT LINE *FUTURE STATION
Actual Results	The application is able to show the correct route information, fare and estimated travel time for the selected station.
Pass / Fail	Pass.

Test Case	Save route result
Test Performed	Steps 1. Click the “Route Planner” icon from the main screen.  Route Planner 2. Select the To & From station by tapping the empty column beside text. Or tap on the empty circle beside station name in the map.  Route Planner From : Click to select From station To : Click to select To station Reset Search Plaza Rakyat Merdeka Kuala Lumpur KL SENTRAL Pasar Seni Maharajalela KL SENTRAL Tun Sambanthan Semantan KL SENTRAL 8 Bangsar Abdullah Hukum Kerinchi 6 7 Mid Valley Seputeh Angkasapuri Pantai Dalam Petaling Asia Jaya Jalan Templer Sungai Besi 3. After selecting both the To & From station, click on the Search Button.  Route Planner From : One Utama To : KL Sentral Reset Search 4. When the route result page is shown, click on the save button on the top right corner of the screen.  Route Result One Utama → KL Sentral
Expected	

Results	When the save button is clicked, if the route saving is successful, it will display a notification "Route Saved!"  Route Result One Utama → KL Sentral Fare : RM15 Travel Time : ~15.03 mins From : One Utama LALUAN MRT MRT LINE *FUTURE STATION TTDI Seksyen 16 Bukit Kiara Pusat Bandar Damansara Semantan To : KL Sentral Route Saved!
Actual Results	The application is able to save the route successfully and display the notification "Route Saved!"
Pass / Fail	Pass.

Test Case	Share route result by SMS
Test Performed	<u>Steps</u> 1. Click the "Route Planner" icon from the main screen. 2. Select the To & From station by tapping the empty column beside text. Or tap on the empty circle beside station name in the map.



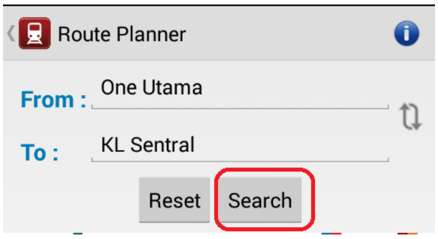
Route Planner

From : Click to select From station

To : Click to select To station

Reset Search

3. After selecting both the To & From station, click on the Search Button.



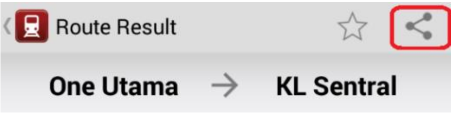
Route Planner

From : One Utama

To : KL Sentral

Reset Search

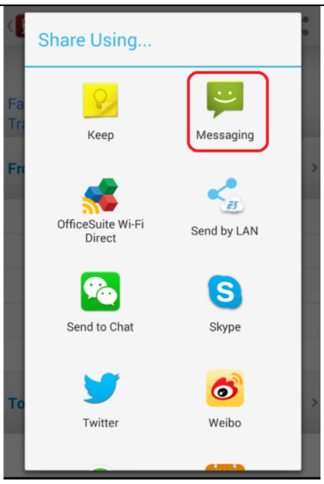
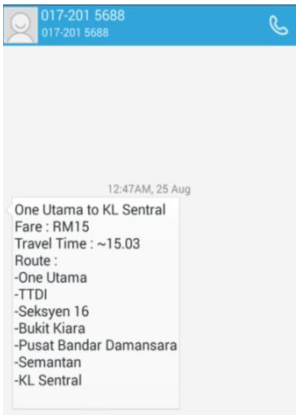
4. When the route result page is shown, click on the share button on the top right corner of the screen.

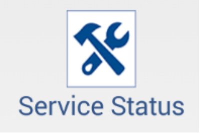


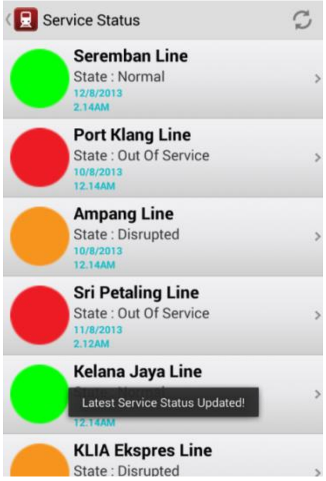
Route Result

One Utama → KL Sentral

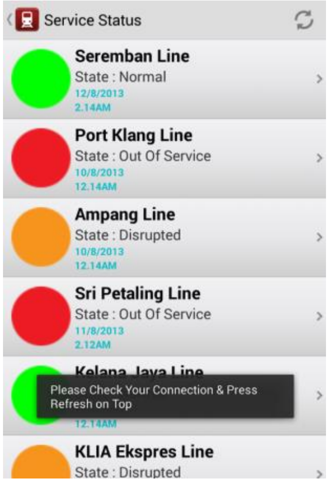
5. A selection menu will pop up and select SMS from the menu.

	 Share Using... Keep Messaging OfficeSuite Wi-Fi Direct Send by LAN Send to Chat Skype Twitter Weibo 6. Enter the recipient number and click send.
Expected Results	The application will format the route result into text and send it to the recipient via SMS.  017-201 5688 017-201 5688 12:47AM, 25 Aug One Utama to KL Sentral Fare : RM15 Travel Time : ~15.03 Route : -One Utama -TTDI -Seksyen 16 -Bukit Kiara -Pusat Bandar Damansara -Semantan -KL Sentral
Actual Results	The application is able to send the route result in text format to the recipient number via SMS
Pass / Fail	Pass.

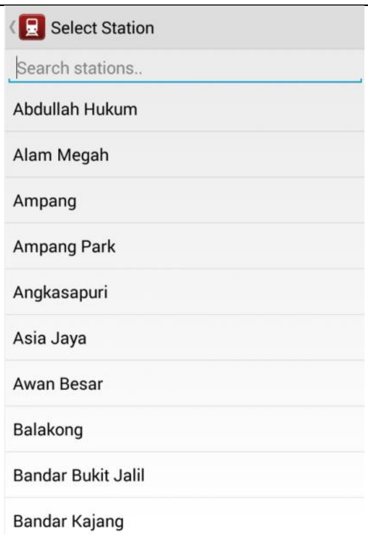
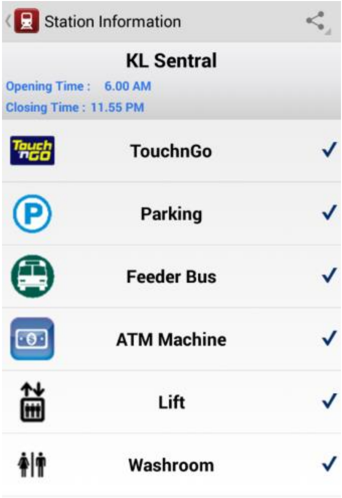
Test Case	To obtain the latest service status XML from the webserver and display the service information to user.
Test Performed	<u>Steps</u> 1. Turn On Data/WiFi Connection. 2. Click the "Service Status" icon from the main screen. 

	3. Wait for the service status list to appear.
Expected Results	When the application successfully grabs the latest service status XML file from the webserver, it will display a notification indicating “Latest Service Status Updated”. The correct service status will then be displayed to the user.  Service Status Seremban Line State : Normal 12/8/2013 2.14AM Port Klang Line State : Out Of Service 10/8/2013 12.14AM Ampang Line State : Disrupted 10/8/2013 12.14AM Sri Petaling Line State : Out Of Service 11/8/2013 2.12AM Kelana Jaya Line Latest Service Status Updated! 12.14AM KLIA Ekspres Line State : Disrupted
Actual Results	The application is able to grab the latest service status xml file from webserver and display the correct notification. The service status for all line is displayed correctly.
Pass / Fail	Pass.

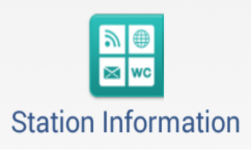
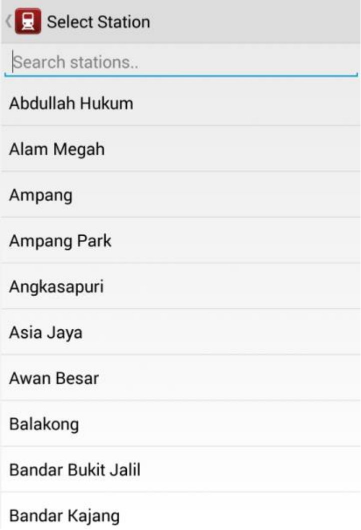
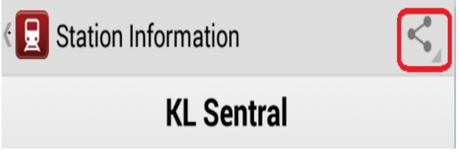
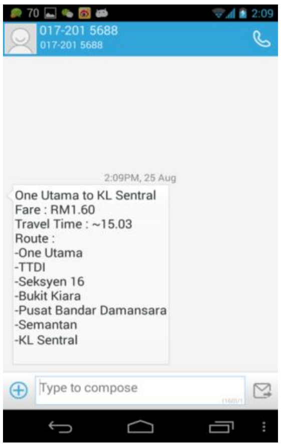
Test Case	Fails to obtain latest service status XML from server, but able to display service status based on last updated information.
Test Performed	<u>Steps</u> 1. Turn Off Data/WiFi Connection 2. Click the “Service Status” icon from the main screen. 3. Wait for the service status list to appear.
Expected Results	When the application fails to grab the latest service status XML file from the webserver, it will display a notification “Please Check Your Connection & Press Refresh on Top”. The service

	status will be displayed to user based on the last updated service status file.  Service Status **Seremban Line** State : Normal 12/8/2013 2.14AM **Port Klang Line** State : Out Of Service 10/8/2013 12.14AM **Ampang Line** State : Disrupted 10/8/2013 12.14AM **Sri Petaling Line** State : Out Of Service 11/8/2013 2.12AM **Kelana Jaya Line** Please Check Your Connection & Press Refresh on Top 12.14AM **KLIA Ekspres Line** State : Disrupted
Actual Results	The application shows there is a network error in obtaining the latest service status XML from the webserver. The last updated service status is displayed to the user.
Pass / Fail	Pass.

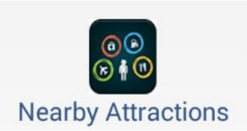
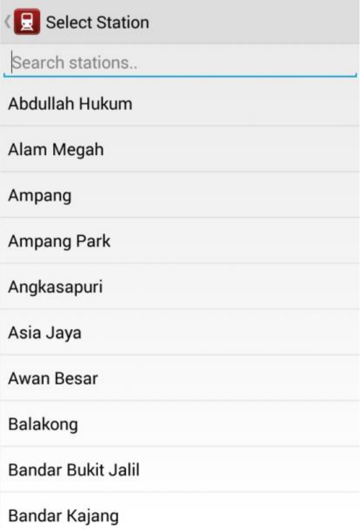
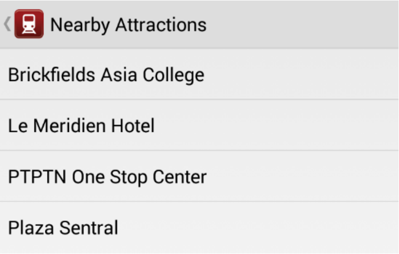
Test Case	Show the correct station information.
Test Performed	<u>Steps</u> 1. Click the “Station Information” icon from the main screen. 2. Select station from list or search for station name from list.

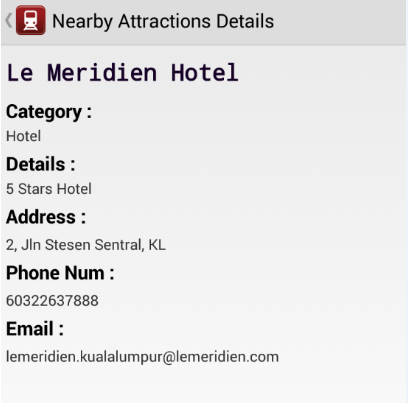
	 Select Station Search stations.. Abdullah Hukum Alam Megah Ampang Ampang Park Angkasapuri Asia Jaya Awan Besar Balakong Bandar Bukit Jalil Bandar Kajang 3. Click on the station that you want to display information.
Expected Results	When the station has been choosen, the opening time, closing time and facilities of the station will be displayed to user.  Station Information KL Sentral Opening Time : 6.00 AM Closing Time : 11.55 PM TouchnGo ✓ Parking ✓ Feeder Bus ✓ ATM Machine ✓ Lift ✓ Washroom ✓
Actual Results	The application is able to show the correct opening time, closing time and facilities for the choosen station.
Pass / Fail	Pass.

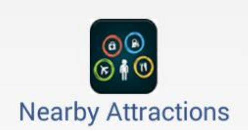
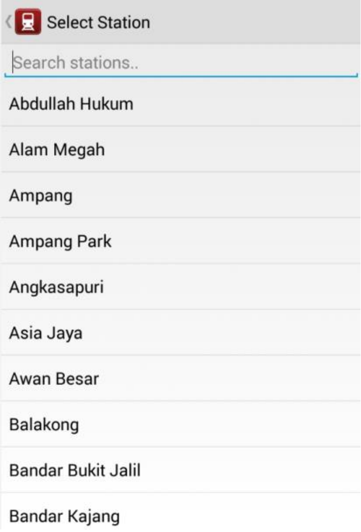
Test Case	Share station information by SMS.
Test Performed	<u>Steps</u> 1. Click the “Station Information” icon from the main screen.

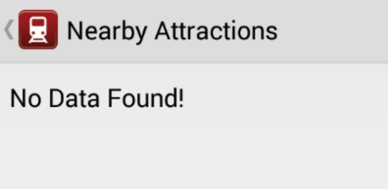
	 2. Select station from list or search for station name from list.  Select Station Search stations.. Abdullah Hukum Alam Megah Ampang Ampang Park Angkasapuri Asia Jaya Awan Besar Balakong Bandar Bukit Jalil Bandar Kajang 3. Click on the station that you want to display information. 4. When the station information page is shown, click on the share button.  5. Select SMS from the share application list. 6. Enter the recipient number.
Expected Results	The recipient should be able to receive the station information formatted in text via SMS.  017-201 5688 017-201 5688 2:09PM, 25 Aug One Utama to KL Sentral Fare : RM1.60 Travel Time : ~15.03 Route : -One Utama -TTDI -Seksyen 16 -Bukit Kiara -Pusat Bandar Damansara -Semantan -KL Sentral Type to compose
Actual	The application is able to send station information in text format to

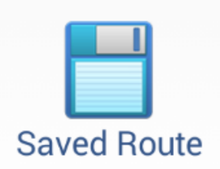
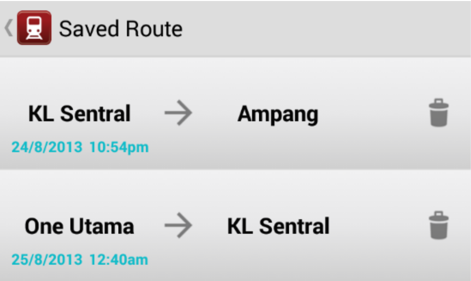
Results	the correct recipient.
Pass / Fail	Pass.

Test Case	Show the nearby attractions for a station.
Test Performed	Steps 1. Click the “Nearby Attractions” icon from the main screen.  2. Select station from list or search for station name from list.  Search stations.. Abdullah Hukum Alam Megah Ampang Ampang Park Angkasapuri Asia Jaya Awan Besar Balakong Bandar Bukit Jalil Bandar Kajang 3. Click on the station that you want to display nearby attractions. 4. Click on the attraction name to display more information.
Expected Results	When the station has been choosen, the list of nearby attractions will be shown.  Brickfields Asia College Le Meridien Hotel PTPTN One Stop Center Plaza Sentral When the user click on the attraction name, the details of attraction

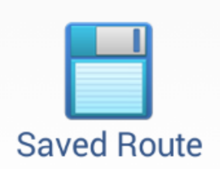
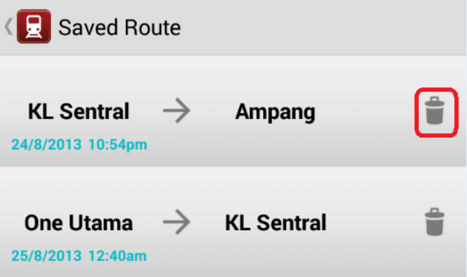
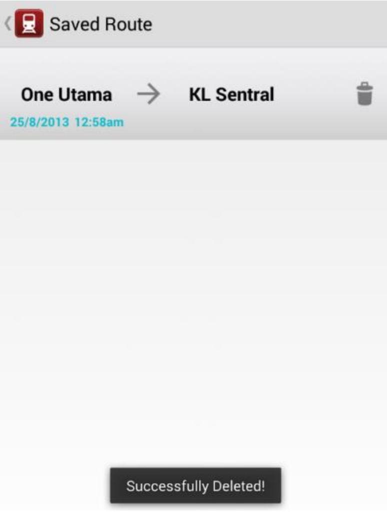
	will be shown.  Nearby Attractions Details Le Meridien Hotel **Category :** Hotel **Details :** 5 Stars Hotel **Address :** 2, Jln Stesen Sentral, KL **Phone Num :** 60322637888 **Email :** lemeridien.kualalumpur@lemeridien.com
Actual Results	The application is able to show the correct list of nearby attractions for a station. When the attraction name is clicked, the details of attraction are shown.
Pass / Fail	Pass.

Test Case	No nearby attractions for a station.
Test Performed	<u>Steps</u> 1. Click the "Nearby Attractions" icon from the main screen.  2. Select station from list or search for station name from list.  Select Station Search stations.. Abdullah Hukum Alam Megah Ampang Ampang Park Angkasapuri Asia Jaya Awan Besar Balakong Bandar Bukit Jalil Bandar Kajang 3. Click on the station that you want to display nearby attractions.

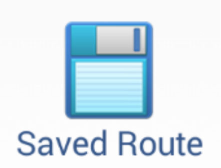
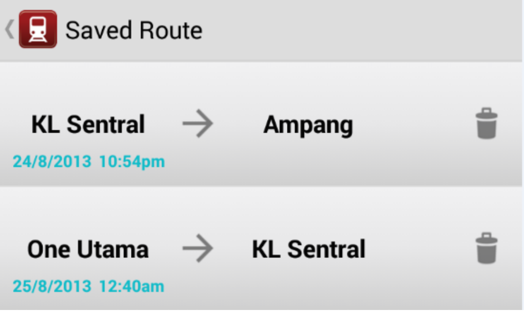
Expected Results	When the station has been choosen, if nearby attractions data is not found, a “No Data Found!” text will be displayed to the user.  Nearby Attractions No Data Found!
Actual Results	The application is able to show “No Data Found!” text for station without nearby attractions.
Pass / Fail	Pass.

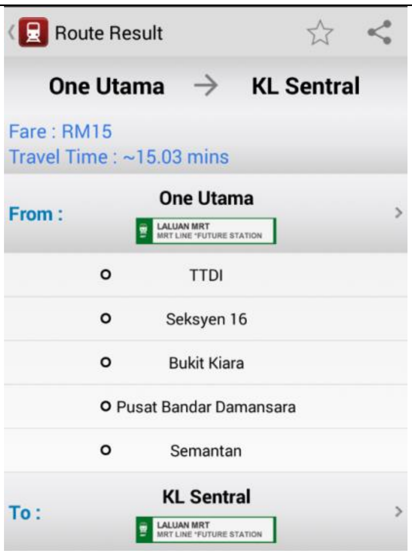
Test Case	Display Saved Route
Test Performed	<u>Steps</u> 1. Click the “Saved Route” icon from the main screen.  2. Wait for the saved route list to be displayed.
Expected Results	All the saved route saved in savedroute table will be retrieved and display to the user.  Saved Route KL Sentral → Ampang 24/8/2013 10:54pm One Utama → KL Sentral 25/8/2013 12:40am
Actual Results	The application is able to show all the saved route listed in savedroute table.

Pass / Fail	Pass.

Test Case	Delete Saved Route
Test Performed	<u>Steps</u> 1. Click the “Saved Route” icon from the main screen.  Saved Route 2. Wait for the saved route list to be displayed 3. Click on the Delete Icon beside the route you want to delete.  Saved Route KL Sentral → Ampang 24/8/2013 10:54pm One Utama → KL Sentral 25/8/2013 12:40am 4. A pop up dialog will appear. Click “Confirm” to delete.
Expected Results	When the delete operation is confirmed, the saved route will be erased from the savedroute table. A “Successfully Deleted” message will appear if it is deleted successfully. The list will be updated with the remaining saved route.  Saved Route One Utama → KL Sentral 25/8/2013 12:58am Successfully Deleted!

Actual Results	The application is able to delete the saved route successfully and list is updated with remaining saved route.
Pass / Fail	Pass.

Test Case	Select Saved Route
Test Performed	<u>Steps</u> 1. Click the "Saved Route" icon from the main screen.  2. Wait for the saved route list to be displayed 3. Click on the saved route that you would like to display.  KL Sentral → Ampang 24/8/2013 10:54pm One Utama → KL Sentral 25/8/2013 12:40am
Expected Results	When the saved route item is clicked, the saved route result will be shown to the user.

	 Route Result One Utama → KL Sentral Fare : RM15 Travel Time : ~15.03 mins From : One Utama LALUAN MRT MRT LINE *FUTURE STATION TTDI Seksyen 16 Bukit Kiara Pusat Bandar Damansara Semantan To : KL Sentral LALUAN MRT MRT LINE *FUTURE STATION
Actual Results	The application is able to show the correct saved route result.
Pass / Fail	Pass.

6.4 System Testing

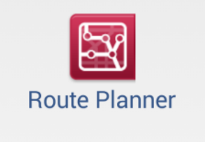
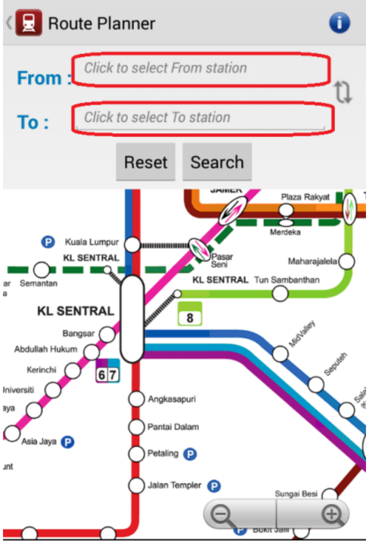
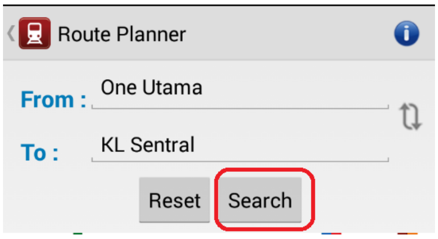
System testing is carried out after integration testing to evaluate whether the application is in compliance with the requirements. System testing is conducted on fully integrated software.

6.4.1 Testing Equipment Specification

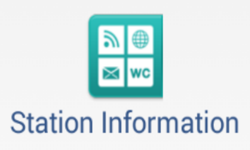
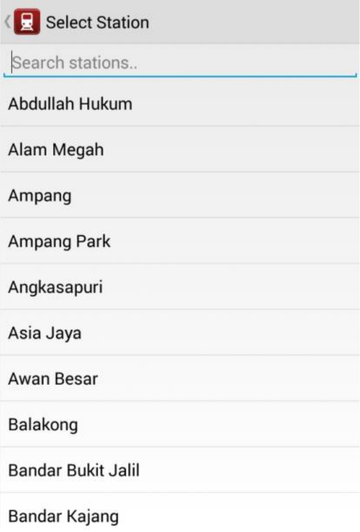
Device Name: LG Nexus 4

- Processor: 1.5Ghz Quad Core Krait
- Memory: 2GB
- OS : Android 4.3 (Jelly Bean)

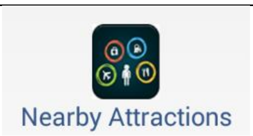
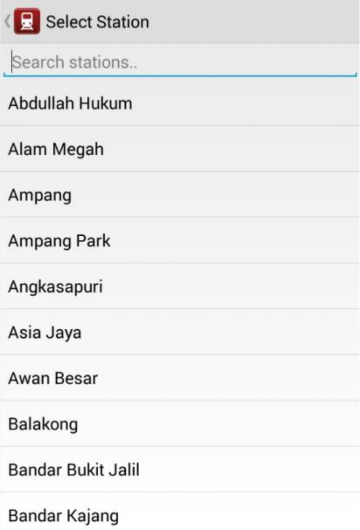
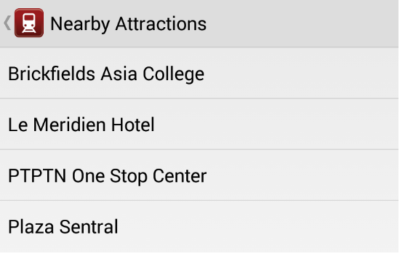
6.4.2 Test Cases

Test Case	Performance Testing: Route result page landing time.
Test Performed	Steps 1. Click the “Route Planner” icon from the main screen.  2. Select the To & From station by tapping the empty column beside text. Or tap on the empty circle beside station name in the map.  Route Planner From : Click to select From station To : Click to select To station Reset Search Plaza Rakyat Merdeka Kuala Lumpur KL SENTRAL Pasar Seni Maharajalela Semantan KL SENTRAL Tun Sambanthan KL SENTRAL 8 Bangsar Abdullah Hukum Kerinchi 6 7 MidValley Seputeh Angkasapuri Pantai Dalam Petaling Jalan Templer Asia Jaya Sungai Besi 3. After selecting both the To & From station, click on the Search Button.  Route Planner From : One Utama To : KL Sentral Reset Search
Expected Results	Route result will be shown to user within 5 seconds after search button click.
Actual Results	The application is able to show the route result in less than 2 seconds after search button click.

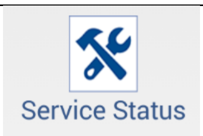
Pass / Fail	Pass.

Test Case	Performance Testing : Station information page landing time.
Test Performed	<u>Steps</u> 1. Click the “Station Information” icon from the main screen.  2. Select station from list or search for station name from list.  Select Station Search stations.. Abdullah Hukum Alam Megah Ampang Ampang Park Angkasapuri Asia Jaya Awan Besar Balakong Bandar Bukit Jalil Bandar Kajang 3. Click on the station that you want to display information.
Expected Results	The station information will be shown within 3 seconds after station click.
Actual Results	The station information is shown within 1 second after station click.
Pass / Fail	Pass.

Test Case	Performance Testing: Nearby attractions page landing time.
Test Performed	<u>Steps</u> 1. Click the “Nearby Attractions” icon from the main screen.

	 Nearby Attractions 2. Select station from list or search for station name from list.  Select Station Search stations.. Abdullah Hukum Alam Megah Ampang Ampang Park Angkasapuri Asia Jaya Awan Besar Balakong Bandar Bukit Jalil Bandar Kajang
Expected Results	Nearby attractions page will be shown to the user within 3 seconds of station click.  Nearby Attractions Brickfields Asia College Le Meridien Hotel PTPTN One Stop Center Plaza Sentral
Actual Results	Nearby attractions for station is shown within 1 second after station click.
Pass / Fail	Pass.

Test Case	Performance Testing: Service status page landing time.
Test Performed	<u>Steps</u> 1. Click the “Service Status” icon from the main screen.

	 2. Wait for the service status list to appear.
Expected Results	Service status page will be shown to the user within 8 seconds after button click.
Actual Results	Service Status for various rail lines is shown within 6 seconds after button click.
Pass / Fail	Pass.

CHAPTER 7

CONCLUSION & RECOMMENDATION

7.1 Contribution of the Application

This application helps citizens and tourists by providing accurate transit guidance on the complex Klang Valley rail network. The application is very comprehensive as it not only assists in providing correct travel route, but is able to display the estimated travel time and fare. Other than that, users can check on the station facilities and nearby attractions. User does not need to waste their time and go online to check for information because the application acts as a one stop centre that provides all the necessary information for using the rail system. The save route function allows user to plan their journey in advance and user can assists their friends or family by sharing route and station information to them via SMS/Email. The core functionalities of the application is designed to be available in offline state, meaning user does not need to have internet connection to access the route planner, station information, nearby attractions and saved route function. This provides a great convenience to user as not every user have access to internet or connection may be limited due to bad network signals. The application helps to reduce the uncertainty of user when using the rail network and encourages more people to utilize the rail system.

7.2 Limitations of Application

The route planning algorithm of the application is not optimized enough because it is unable to take in all the variables such as train frequency in different period of day, transit time, and train speed into consideration. Thus some of the result is not optimized and require tuning. The application is not able to offer user with different option of route available.

7.3 Future Enhancement

- More research has to be done on the route planning algorithm to find out the most accurate and optimized route.
- The application should offer user different preference on route search, such as least fare, shortest time, shortest distance, least interchange station.
- Integrate the application with a bus planning application to provide a more comprehensive route planning application that can guide user to navigate using the bus and rail system.

7.4 Conclusion

The project has been developed over a period of 8 months, starting from the submission of proposal, to requirements gathering, design, implementation and testing. The project execution is done in accordance to the Gantt chart and follows the best practice of Software Engineering.

The requirements are first elicited using Study on Existing System and Document Analysis techniques. The output collected from requirements gathering is further analysed to come out with the functional and non-functional requirements. The application database structure, class structure and flows are then designed based on the specification. The implementation phase started after designing phase. The route calculation part of the application is coded using Java programming, whereas the user interface and other functionalities are implemented using Android SDK.

When implementation phase is on-going, testing phase is conducted in parallel. The application developed has demonstrated the ability to satisfy all the requirements after undergoing a series of test. All the objectives and requirements for the project have been successfully fulfilled at the end of the project.

REFERENCES

Accenture, 2011. Nokia and Accenture Close Symbian Software Development and Support Services Outsourcing Agreement. *Accenture Newsroom*, [Online] 30 Sep. Available at: http://newsroom.accenture.com/news/nokia-and-accenture-close-symbian-software-development-and-support-services-outsourcing-agreement.htm [Accessed 15 March 2013].

Charland, A. and LeRoux B., 2011. Mobile Application Development: Web vs. Native. Queue – Data, [Online] 1 April. Available at: http://dl.acm.org/citation.cfm?id=1966989.1968203&coll=DL&dl=ACM [Accessed 20 March 2013].

Deutsche Welle, 2012. Global smartphone boom expected, report shows. *Deutsche Welle*, [Online] 21 Sep. Available at: http://www.dw.de/global-smartphone-boom-expected-report-shows/a-16395292 [Accessed 16 March 2013].

Gartner, 2011. Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year End 2012. *Gartner*, [Online] 7 April. Available at: https://www.gartner.com/it/page.jsp?id=1622614 [Accessed 24 March 2013].

Google, 2013. *User Interface Guidelines | Android Developers*. [Online]Available at: http://developer.android.com/guide/practices/ui_guidelines/index.html. [Accessed 20 March 13].

Google, 2013. *App Components | Android Developers*. [Online] Available at: http://developer.android.com/guide/components/index.html. [Accessed 21 March 13].

Howland, J.E., Lewis, M., Hicks, T. and Pitts, G., 2003. A breadth-first companion for the CSI course. *Journal of Computing Sciences in Colleges*, [E-journal] 18,4, 11-15, Available through: ACM Digital Library http://dl.acm.org/citation.cfm?id=767598.767602&coll=DL&dl=ACM [Accessed 30 March 2013].

IDC, 2011. Smartphones surpass feature phone shipments for first time. *IDC*, [Online] 12 Sep. Available at: http://www.knowyourmobile.com/products/14432/smartphones-surpass-feature-phone-shipments-first-time [Accessed 22 March 2013].

KLIAExpress, n.d. *KLIA Express & KLIA Transit*. [Online] Available at: http://www.kliaekspres.com/erlsb/default.aspx. [Accessed 16 March 13].

KTM, n.d. *KTM Komuter*. [Online] Available at: http://www.ktmkomuter.com.my/ [Accessed 16 March 13].

Lee, W.M., 2012. *Android 4 Application Development*. 1st ed. Indiana: John Wiley & Sons.

MRT Corp, n.d. *Klang Valley My Rapid Transit*. [Online] Available at: http://www.mymrt.com.my/stations.html [Accessed 16 March 13].

Ngu, P.H. and Do, V.T., 2012. *Evaluation of mobile app paradigms*. Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia. Available at: http://dl.acm.org/citation.cfm?id=2428955.2428968&coll=DL&dl=GUIDE&CFID=204469111&CFTOKEN=49838060 [Accessed 30 March 2013].

Nielsen, 2012. Smartphones Account For Half Of All Mobile Phones, Dominate New Phone Purchase In The US. *Neilsen*, [Online] 29 March. Available at: http://www.nielsen.com/us/en/newswire/2012/smartphones-account-for-half-of-all-mobile-phones-dominate-new-phone-purchases-in-the-us.html [Accessed 22 March 2013].

Palmieri, M. and Singh, I., 2012. *Comparison of cross-platform mobile development tools*. Intelligence in Next Generation Network (ICIN) 16th International Conference. Available at: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6376023 [Accessed 12 March 2013].

RapidKL, n.d. *Your Public Transport Portal*. [Online] Available at: http://www.myrapid.com.my/ [Accessed 16 March 13].

Richard, F. and Behrouz, A., 2005. *Data Structures : A Pseudocode Approach with C*, USA: Course Technology.

SPAD, n.d. *Klang Valley Urban Rail Development Plan.* [Online] Available at: http://www.spad.gov.my/sites/default/files/URDP-MP-MOC_110913-v1.0.pdf
[Accessed 2013 March 14].

Telecompaper, 2012. Smartphone penetration at 27% in Malaysia – survey. Telecompaper, [Online] 9 May. Available at: http://www.telecompaper.com/news/smartphone-penetration-at-27-in-malaysia-survey--872016 [Accessed 18 Match 2013].

The Star, 2013. The Klang Valley has finally arrived to be in a top spot in world business. *The Star Online*, [Online] 2 Jan. Available at: http://thestar.com.my/news/story.asp?file=/2013/1/2/nation/12522345&sec=nation [Accessed 15 March 2013].

Zheng, Z.H. and Cai, W.X., 2008. *A dynamic shortest path algorithm based on real-time traffic information in the urban public transit network*. Service Operations and Logistics, and Informatics, IEEE International Conference, 2008. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4682762 [Accessed 13 March 2013].

APPENDICES

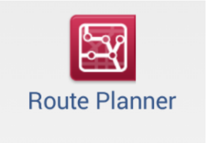
Appendix A: User Manual

The hardware requirements to use this application are as follow:

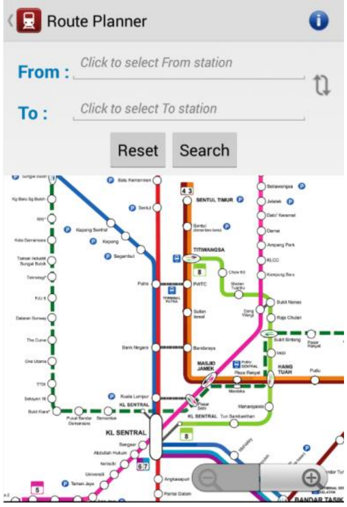
- Android Based Phone
  - Min 1Ghz Processor.
  - 512MB RAM and above.
  - Android 4.0 (Jelly Bean)

Search For A Route

1. Click on the "Route Planner Icon" on the main screen.



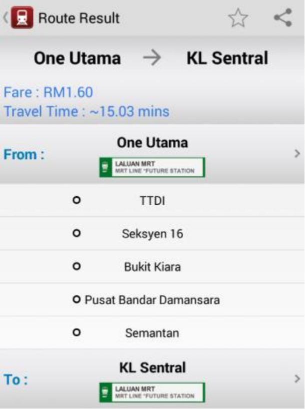
2. Select your To & From station by tapping on the text box or click on the map.



From : Click to select From station

To : Click to select To station

3. After selecting the stations, press on search button.
4. User will be redirected to route result page.



Route Result

One Utama → KL Sentral

Fare : RM1.60
Travel Time : ~15.03 mins

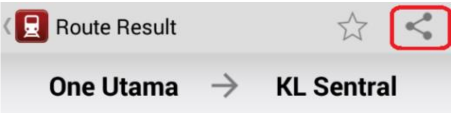
From : One Utama
LALUAN MRT MRT LINE *FUTURE STATION

TTDI
Seksyen 16
Bukit Kiara
Pusat Bandar Damansara
Semantan

To : KL Sentral
LALUAN MRT MRT LINE *FUTURE STATION

Share A Route Result

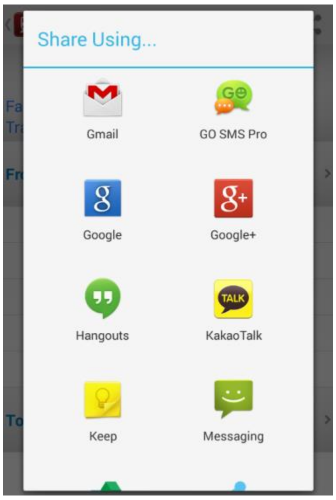
1. Perform steps on ***Search For a Route.***
2. Click on the Share Button on the top right corner of the route result page.



Route Result

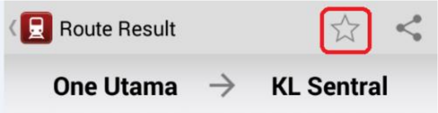
One Utama → KL Sentral

3. Select the application that you would like to share.



Save A Route Result

1. Perform steps on ***Search For a Route.***
2. Click on the Save Button on the top right corner of the route result page.

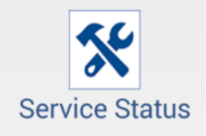


One Utama → KL Sentral

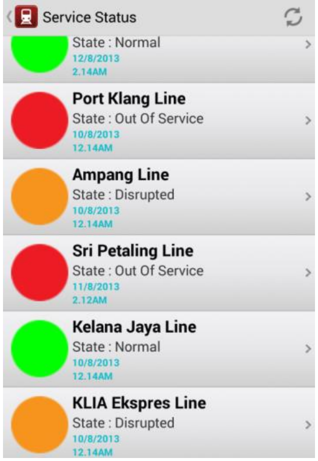
3. Wait for the “Route Saved!” message to appear.

Viewing Service Status

1. Turn on your data/WiFi connection.
2. Click on the “Service Status” icon on main screen.



3. Wait for the service status information to appear.



State : Normal
12/8/2013
2.14AM

Port Klang Line
State : Out Of Service
10/8/2013
12.14AM

Ampang Line
State : Disrupted
10/8/2013
12.14AM

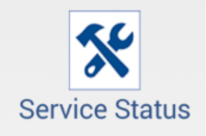
Sri Petaling Line
State : Out Of Service
11/8/2013
2.12AM

Kelana Jaya Line
State : Normal
10/8/2013
12.14AM

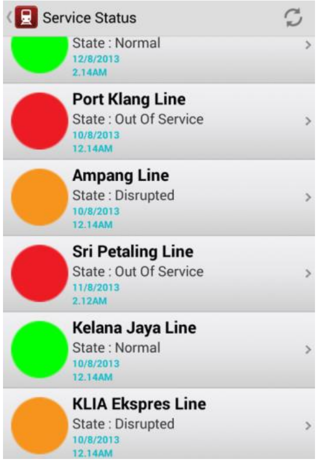
KLIA Ekspres Line
State : Disrupted
10/8/2013
12.14AM

Viewing Service Status Details

1. Turn on your data/WiFi connection.
2. Click on the “Service Status” icon on main screen.



3. Wait for the service status information to appear.
4. Tap on the desired line that you want to view.



Service Status

State : Normal
12/8/2013
2.14AM

Port Klang Line
State : Out Of Service
10/8/2013
12.14AM

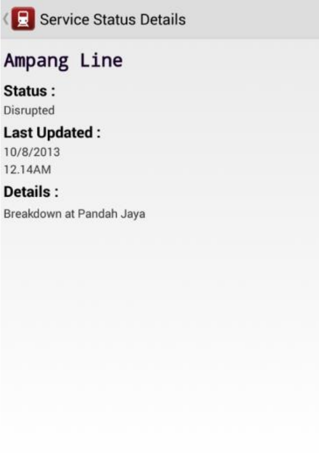
Ampang Line
State : Disrupted
10/8/2013
12.14AM

Sri Petaling Line
State : Out Of Service
11/8/2013
2.12AM

Kelana Jaya Line
State : Normal
10/8/2013
12.14AM

KLIA Ekspres Line
State : Disrupted
10/8/2013
12.14AM

5. The service status details for line will be displayed.



Service Status Details

Ampang Line

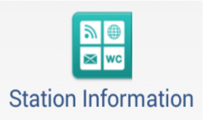
Status :
Disrupted

Last Updated :
10/8/2013
12.14AM

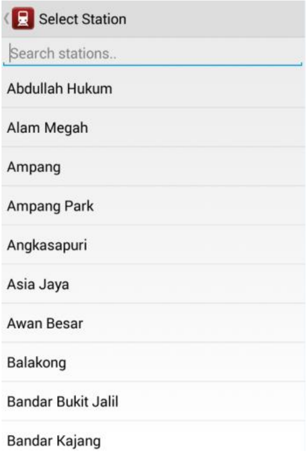
Details :
Breakdown at Pandah Jaya

Viewing Station Information

1. Click on the “Station Information” icon on main screen.



2. Select your desired station from the list.

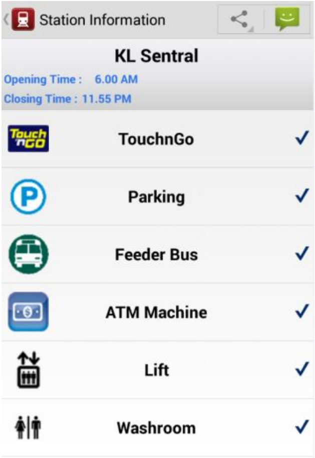


Select Station

Search stations..

Abdullah Hukum
Alam Megah
Ampang
Ampang Park
Angkasapuri
Asia Jaya
Awan Besar
Balakong
Bandar Bukit Jalil
Bandar Kajang

3. Wait for the station information to be displayed.



Station Information

KL Sentral

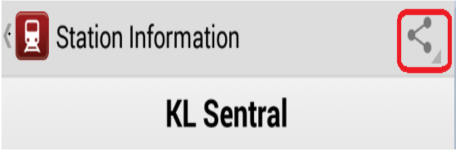
Opening Time : 6.00 AM

Closing Time : 11.55 PM

TouchnGo	✓
Parking	✓
Feeder Bus	✓
ATM Machine	✓
Lift	✓
Washroom	✓

Share Station Information

1. Perform steps on ***Viewing Station Information.***
2. Click on the Share Button on the top right corner of the station information page.



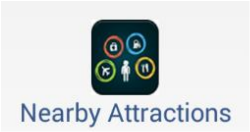
Station Information

KL Sentral

3. Choose the application that you want to share.

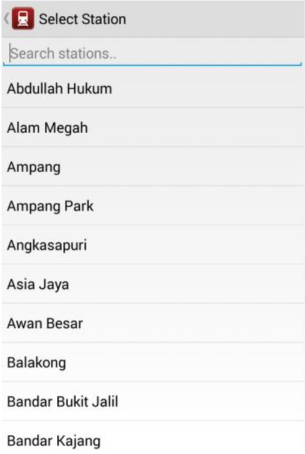
Viewing Nearby Attractions

1. Click on the “Nearby Attractions” icon on main screen.



Nearby Attractions

2. Select your desired station from the list.



Abdullah Hukum

Alam Megah

Ampang

Ampang Park

Angkasapuri

Asia Jaya

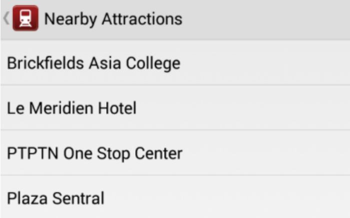
Awan Besar

Balakong

Bandar Bukit Jalil

Bandar Kajang

3. Click on the nearby attractions that you want to view.



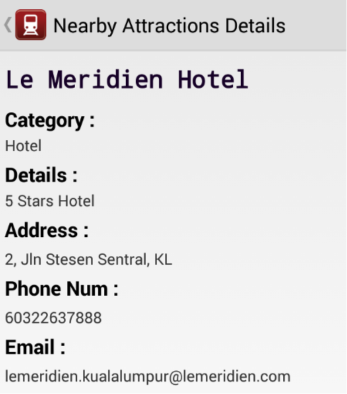
Brickfields Asia College

Le Meridien Hotel

PTPTN One Stop Center

Plaza Sentral

4. Wait for the nearby attraction details page to appear.



Le Meridien Hotel

Category :
Hotel

Details :
5 Stars Hotel

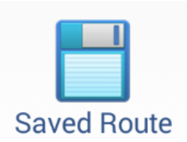
Address :
2, Jln Stesen Sentral, KL

Phone Num :
60322637888

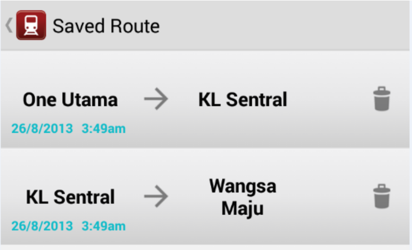
Email :
lemeridien.kualalumpur@lemeridien.com

Viewing Saved Route

1. Click on the “Saved Route” icon on main screen.



2. The list of saved route will be displayed.



Saved Route

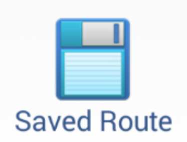
One Utama → KL Sentral
26/8/2013 3:49am

KL Sentral → Wangsa Maju
26/8/2013 3:49am

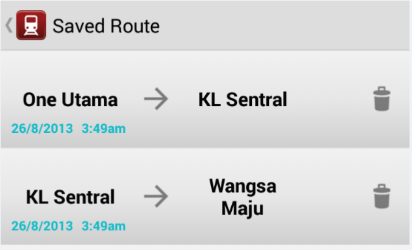
3. Click on the saved route to view route result.

Delete Saved Route

1. Click on the “Saved Route” icon on main screen.



2. The list of saved route will be displayed.



Saved Route

One Utama → KL Sentral
26/8/2013 3:49am

KL Sentral → Wangsa Maju
26/8/2013 3:49am

3. Click on Delete Button to erase the saved route.
4. Click confirm when prompted.
5. Wait for “Successfully Deleted!” message to appear.