CSIS0801 Final Year Project 2014 – 2015

FYP14007
A Smart Phone Application to Assist Driving

Intermediate Project Report

Supervisor: Dr. Chim T W
Second Examiner: Dr. Yiu S M

Students:
Leung Kin Ning  3035040891
Lee Jin Man  3035040011
Wong Kwan Wai  2013718589
# Table of Contents

EXECUTIVE SUMMARY .................................................................................................................. 3  

SECTION 1:         PROJECT OVERVIEW ......................................................................................... 4  
  1.1       BACKGROUND .................................................................................................................. 4  
  1.2       RELATED WORK .............................................................................................................. 5  

SECTION 2: OBJECTIVE ................................................................................................................... 7  

SECTION 3: STRATEGY .................................................................................................................... 8  

SECTION 4: TECHNOLOGY AND TOOL .............................................................................................. 9  
  4.1       DEVELOPMENT TOOL ..................................................................................................... 9  
  4.2       TECHNOLOGY USED ....................................................................................................... 11  

SECTION 5: SYSTEM DESIGN .......................................................................................................... 12  
  5.1       MAIN APPLICATION FLOW ............................................................................................. 12  
    5.1.1       Overview ..................................................................................................................... 12  
    5.1.2       Client-Server Communication .................................................................................... 13  
    5.1.3       Data manipulation in web server and DB ................................................................. 13  
    5.1.4       Algorithm Implementation ....................................................................................... 16  
    5.1.5       Information Retrieval ............................................................................................... 18  
    5.1.6       Data Fetching and Storing ...................................................................................... 19  

SECTION 6: DATABASE DESIGN ...................................................................................................... 20  

SECTION 7: FUTURE DEVELOPMENT .............................................................................................. 22  

SECTION 8: DIVISION OF WORK ..................................................................................................... 24  

SECTION 9: REFERENCES ................................................................................................................ 25
Executive Summary

Mobile applications are becoming more and more useful in our life. It is not difficult to observe that people use a variety of mobile applications anywhere and anytime. For example, a majority of people use WhatsApp, a famous mobile application nowadays, for communication. Also, there are many utility applications and game applications in the market. Applications play a very important role in the society. They do not only provide us a platform of entertainment and communication, but also assistance to users. In this project, we will integrate a voice recognition function to a mobile application.

Our project aims to provide drivers a more convenient way to obtain the instant road conditions and traffic information. We know that tying is painful, especially when you are driving. With the aid of the voice recognition function, we hope that drivers can retrieve useful traffic information in a user-friendly way. At the same time, they can pay more attention on the road conditions while driving. We expect our mobile application can assist drivers efficiently.
Section 1: Project Overview

1.1 Background

In this century, private cars seem not luxury goods when compared to the past. According to the statistical data provided by the Transport Department of Hong Kong, the number of licensing of vehicles has an increasing trend from about 37000 in 2009 to about 42000 in 2013. It infers that more and more people will have driving licenses and they can go everywhere by their own car.

![Graph showing the number of licensing of vehicles]

Fig. 1.1 – Statistical Data from the Transport Department of Hong Kong

Because of the increasing number of drivers in the society, a number of mobile apps are available in the market to assist driving. Some mobile apps can be used as a black box to record information for car accident investigation while some provide functions that the smart phone camera can be used to record road conditions.

![Black box used to record information of the car]

Fig. 1.2 – Black box used to record information of the car
conditions and the GPS receiver can be used to log vehicle speed.

We think that the market of driving mobile app still have a potential for development. Although there are a variety of user-friendly driving apps in app store for drivers, nearly most of them are needed to input commands by touching on the screen on the mobile phone.

Thus, in order to provide a more convenient way for drivers to input commands to the driving app, we are going to embed a voice recognition function so that drivers can retrieve traffic information by saying some commands to the app.

1.2 Related Work

In our society, there have been some driving assistant devices such as GPS navigation system that make use of synchronized voice to assist drivers when they are driving. For example, the driver can input a location where he or she are going to arrive, then the system tell the driver with the aid of
synchronized voice the details of the destination when the driver is driving to the location such as the time left to arrive at the destination, the distance to the destination and which road the driver should choose in order to arrive at the location in a shorter time. This type of system indeed brings an advantage to drivers and become very popular nowadays.

Besides, many new brands of mobile phones today have been installed a voice recognition function such as Siri used in Iphone and Vlingo used in most of the Android’s mobile phones. The function provides a more convenient way for users to input commands to the phone. For example, the mobile phone can play a song when a user just simply says “Play music” to the phone.

All of these examples show the potential of using voice as a medium in inputting commands and retrieving information in nowadays society is tremendous and feasible, and the value is considerable.
Section 2: Objective

In this project, we want to develop an app with a user-friendly interface and a convenient way to provide assistant to drivers. We observe that many car accidents happened in the past resulted from drivers which used their cell phones while driving. Thus, the main goal of our project is “Hands on the wheel, eyes on the road”.

Our app will offer drivers personalized access to content and services through deep voice integration, while ensuring seamless connectivity and end-to-end delivery of traffic information.

By using our driving app, we hope that drivers can pay more attention on the road conditions and reduce the number of car accidents resulting from using mobile apps when driving. Also, through using the app, drivers can easily get the on-time information about the traffic and road conditions in Hong Kong whenever they want.

Fig. 2.1 – Advertisement urges drivers not to text while driving

Fig. 2.2 – According to new U.S. government research data, more than 16,000 death in car accidents are caused by texting while driving from 2001 to 2007
Section 3: Strategy

Actually, at the present, we have 3 main objectives in this project. To achieve our objectives, we have identified some strategies to achieve each of the following:

In order to achieve Objective 1 – Develop a driving assistance application with user-friendly interface with simple operation, we will provide some handy ways such as buttons or a list of choices for drivers inputting data into our application and then get the desired information they want. We hope that the design of the application can be as simple as possible so that drivers can get the desired information quickly and conveniently.

To fulfill Objective 2 – Incorporate a convenient functionality into the application, we are going to embed the voice recognition function into the mobile application. We know that typing is very inconvenient and troublesome, especially when you are driving. We hope this assistive feature will be helpful and useful for the drivers.

Lastly, to accomplish objective 3 – Provide the comprehensive and updated traffic information and road conditions to drivers, data will be retrieved from the government websites time to time. Then, analysis on the data will be carried out and then the useful information will be available to the drivers. Besides, some useful tools will be specifically available such as map for users to view the shortest route when they are going to travel from one place to another place.
Section 4: Technology and Tool

4.1 Development Tools

4.1.1 Git (version control)

Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. We use this as a platform to share the source codes among each other. The source codes will be stored in the BitBucket and each of us can fetch and commit the source codes and know about the progress of every member.

4.1.2 Spring tool suite (customized eclipse)

We adopted the Spring Tool Suite in the Eclipse platform to develop our application. The Spring Tool Suite is an Eclipse-based development environment that is customized for developing Spring applications. It provides a ready-to-use environment to implement, debug, run, and deploy your Spring applications, including integrations for Git and Gradle.

4.1.3 Gradle

Gradle is a build automation system. Gradle can automate the building, testing, publishing, deployment and more of software packages or other types of projects such as generated static websites, generated documentation or indeed anything else.

We adopted Gradle to be our build automation system because Gradle combines the power and flexibility of Ant with the dependency management and conventions of Maven into a more effective way to build.
4.1.4 Android Studio

Android Studio is the official IDE for Android application development, based on IntelliJ IDEA. With its flexible Gradle-based build system, we can build variants and generate multiple .apk file. Also, the studio offers code templates to help us build common app features and rich layout editor with support for drag and drop theme editing.

We adopted Android Studio over Eclipse for our development because Android Studio offers better syntax highlighting, refactoring tools and better build system.

4.1.5 Apache Tomcat

Apache Tomcat is an open-source web server and servlet container developed by the Apache Software Foundation (ASF). We adopted Apache Tomcat as our web server for serving web services requests and running scheduled operations.

4.1.6 MySQL Database

We adopted MySQL Database in our project to store the data collected from the transport department.
4.2 Technology used

4.2.1 Restful Web Services

RESTful web services are built to work best on the Web. Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs). In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.

4.2.2 Spring Framework

The Spring Framework is an open source application framework and inversion of control container for the Java platform. We utilized some of the services provided the framework, such as Dependency Injection, Spring MVC web application and RESTful web service framework.

4.2.3 Google Maps Android API

With the Google Maps Android API, we can add maps based on Google Maps data to our application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. We make use of API calls to add markers and polylines a basic map. Each marker is associated with a road ID which is provided to our server in order to obtain traffic conditions of certain roads.
Section 5: System Design

5.1 Main Application Flow

5.1.1 Overview

Restful Web Service by HTTP GET request

Fetch Data every 5 minutes

Get data (in XML format)

Store Data

Web server

Get Data

DB server

Transport department, HKSAR
5.1.2 Client-Server Communication

Drivers can enquiry the information of a specific road by choosing the starting point and destination. Once drivers enter the corresponding data into the “From” and “To” boxes, the application will send the data into the web server and find the information of that specific route.

5.1.3 Data manipulation in web server and DB
Once the web server receives the starting and ending points, it will find out all the routes including the starting and ending point. Then, after finding all the routes, a table will be created which is used to store the information of each route and the information will be used for further analysis.

Take an example where 001 is the starting point and 006 is the destination. According to the data in the server, we find that there are 2 routes including the assigned starting point and destination. Then a table like the one below will be created to show the possible routes.

<table>
<thead>
<tr>
<th>Route</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001</td>
<td>002</td>
<td>003</td>
<td>004</td>
<td>006</td>
</tr>
<tr>
<td>2</td>
<td>001</td>
<td>005</td>
<td>006</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram showing possible routes between nodes 001 and 006]
After the table is created, the node number in each route in the table will be collected and then pass to the database. Once all the node numbers is passed to the database sequentially, database will return the speed and distance of each sub-route in each route. Take route 2 as an example, the database will return the speed and distance of the sub-routes from point 001 to point 005 and from point 005 to 006. Then, by using the formula indicated by (*) above, the total time of each route can be calculated. After comparing the total time of each possible route, Route 1 will be chosen.
route, the route with the shortest time will be chosen and the result will be returned to the driver.

5.1.4 Algorithm Implementation

To find out all the routes from the starting to the ending point, Breath-First-Search (BFS) have been used. In details, when the server side received the starting and ending location, BFS will take the starting point as a root node and inspects all the neighboring nodes. Since a neighboring node represents another location of the map that can be reached from the current node, the current and the neighbor nodes will all be visited and inspected one by one. By iterating the process, all the nodes that can be reached from the root node can be discovered, in other words, all the paths that can be reached from the starting point can be found. The time complexity would be $O(|V| + |E|)$ since every vertex and every edge will be explored in the worst case.

To represent the location on the map as a node and road as an edge, adjacency list has been used for storing the information. In details, array lists are employed as the data structure for implementing the adjacency list. The reason to use adjacency list is that the number of nodes are much more than the number of edges. By using adjacency list, it would be fast enough for BFS to search through the node for finding a path. Meanwhile, as array list is a kind of random access list, the access of data can be achieved by using direct indexing.

Besides, the reason of using BFS is that we concern about the run-time performance. Indeed, there are some algorithms such as Dijkstra’s Algorithm
that can perform much better than BFS since the algorithm can return a shortest path rather than all the paths that can be reached from the starting point. However, as Dijkstra's Algorithm takes the lowest cost/weight into account when calculating the shortest path, that means the algorithm may have to fetch data from the database to retrieve information e.g. the distance and speed limit of a road, for each visit of the nodes. Such implementation will certainly degrade the performance of the apps as database access will take place frequently.
5.1.5 Information Retrieval

<table>
<thead>
<tr>
<th>Route</th>
<th>Starting Position</th>
<th>Destination</th>
<th>Distance (m)</th>
<th>Average speed (km/hr)</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>006</td>
<td>8000</td>
<td>43.6</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Route Details</th>
<th>Starting Point</th>
<th>Destination</th>
<th>Distance (m)</th>
<th>Speed (km/hr)</th>
<th>Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>002</td>
<td>3000</td>
<td>70</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>003</td>
<td>2000</td>
<td>50</td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>004</td>
<td>1000</td>
<td>30</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>006</td>
<td>2000</td>
<td>40</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

The application will show the route and route details of the selected shortest time route to the driver on the phone's screen. The first table contains the summary of the route and it includes the starting position, ending position, distance travelled, average speed required and the time needed. Also, in the second table, it contains the details of the routes. It includes the road information of each sub-route of the selected route. Same as the first table, the starting point, ending point, distance, speed and time will be recorded in the second table.
5.1.6 Data Fetching and Storing

In the system design, our web server will fetch the data provided by the Transport Department, HKSAR every 5 minutes. The data return to the web server will be in XML format. After the web server receives the data, it will

```

<!DOCTYPE xml>  
<jtis_speedmap>  
  <LINK_ID>3006-30068</LINK_ID>  
  <REGION>K</REGION>  
  <ROAD_TYPE>URBAN ROAD</ROAD_TYPE>  
  <ROAD_SATURATION_LEVEL>TRAFFIC GOOD</ROAD_SATURATION_LEVEL>  
  <TRAFFIC_SPEED>30</TRAFFIC_SPEED>  
  <CAPTURE_DATE>2015-01-21T20:38:35</CAPTURE_DATE>  
</jtis_speedmap>  

<!DOCTYPE xml>  
<jtis_speedmap>  
  <LINK_ID>30069-300701</LINK_ID>  
  <REGION>K</REGION>  
  <ROAD_TYPE>URBAN ROAD</ROAD_TYPE>  
  <ROAD_SATURATION_LEVEL>TRAFFIC GOOD</ROAD_SATURATION_LEVEL>  
  <TRAFFIC_SPEED>31</TRAFFIC_SPEED>  
  <CAPTURE_DATE>2015-01-21T20:38:35</CAPTURE_DATE>  
</jtis_speedmap>
```
update the database and the database will store the new data. The database will only store the latest data and the old data will be discarded.

**Section 6: Database Design**

After collecting the data from the transport department, the data will be stored in the database. Also, when drivers need to know the traffic information of a specific road, data will be retrieved from the database.

In our database, in order to increase the efficiency in retrieving data and reduce redundancy, there is one table to store the road data. Each record in the table has 19 fields and the fields store the road which is specified in the road number, the starting point, the ending point, the Chinese name of the road, the English name of the road, the region the road is in, the category of the road, the Chinese starting and ending address of the road, the English starting and ending address of the road, the latitude of the starting and ending point, the longitude of the starting and ending point, the distance of the road, the traffic speed specified of the road, the road saturation level and the overview polyline points.

The screenshots of the records in the table is shown below.

<table>
<thead>
<tr>
<th>link_id</th>
<th>start_node</th>
<th>end_node</th>
<th>name_tc</th>
<th>name_en</th>
<th>region</th>
<th>road_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>34372-3450</td>
<td>34372</td>
<td>3450</td>
<td>5 阿柄線</td>
<td>Route 5</td>
<td>K</td>
<td>MAJOR ROUTE</td>
</tr>
<tr>
<td>34481-34371</td>
<td>34481</td>
<td>34371</td>
<td>5 阿柄線</td>
<td>Route 5</td>
<td>K</td>
<td>MAJOR ROUTE</td>
</tr>
</tbody>
</table>
### start_address_tc | start_address_en | start_location_lat | start_location_lng
--- | --- | --- | ---
香港土瓜灣東九龍走廊 | East Kowloon Corridor, To Kwa Wan, Hong Kong | 22.3140916 | 114.1873959

### end_address_tc | end_address_en | end_location_lat | end_location_lng
--- | --- | --- | ---
香港紅磡漆咸道北 | Chatham Road North, Hung Hom, Hong Kong | 22.3091797 | 114.1840819
香港土瓜灣東九龍走廊 | East Kowloon Corridor, To Kwa Wan, Hong Kong | 22.3141631 | 114.1873527

### distance | traffic_speed | road_saturation_level
--- | --- | ---
0.6 km | 52 | TRAFFIC GOOD

0.6 km | 64 | TRAFFIC GOOD
Section 7: Future Development

In the next stage of development, we are going to continue our implementation. We have 3 main areas for implementation.

The implementation of locating more points on the map
As a road may be very long, drivers may be only need to retrieve the traffic information of a small portion of that road. In order to provide more tailed-made traffic information to drivers, we are going to pay more efforts to locate more points on the map. Thus, drivers can choose the starting point and the destination in a more specific way and obtain the traffic information with a higher accuracy.

The implementation of adding more routes on the map
As we mainly depend on the data provided by the government, it only provides the road data of main roads in Hong Kong. Therefore, we only display the routes of which the government has the road data on the map at present. However, we think that the routes displayed are not enough. In order to provide comprehensive traffic information to drivers, we are going to do some research on the Internet and do some data analysis to get more details of other roads.

The implementation of adding landmarks on the lists of starting point and destination
We are now providing a list for drivers to choose their starting point and the destination. However, one road may have many points and drivers may not clearly know the exact locations in the real world the points stand for. In order to provide a more user-friendly way for drivers to precisely choose the starting and ending point, we are going to add the name of landmarks near the points beside the road name.
The implementation of voice recognition function

At the present stage, we have to enter the starting location and destination by touch input. However, the main focus of our FYP is to implement the voice recognition function so as to provide a convenient way for drivers to input commands into the application. At the same time, drivers can retrieve the information from the application by synthetic voice other than on the screen.

The application first will provide a full map to drivers. When drivers say a place, such as “Causeway Bay”, to the application, the map will zoom to the area of Causeway Bay. Also, drivers can say the name of landmarks to the application, such as “Sogo”. After the application receives the command, the map will automatically zoom to the area near Sogo.

On the map, markers are numbered. Drivers only need to say the number on the map to the application to indicate the starting location and destination.

Besides, once the drivers input the starting location and destination in “From” and “To” boxes and the shortest route is displayed on the map, traffic news of the roads nearby the chosen path will be retrieved. The traffic news will be read out by synthetic voice from the application to drivers.
## Section 8: Division of Work

<table>
<thead>
<tr>
<th>Team member</th>
<th>Tasks</th>
<th>Contact</th>
</tr>
</thead>
</table>
| Leung Kin Ning     | ➢ Android application  
                     | ➢ Voice recognition function | kknleung@connect.hku.hk  |
| Lee Jin Man        | ➢ Database system  
                     | ➢ Server configuration    | jlee1213@connect.hku.hk   |
| Wong Kwan Wai      | ➢ Web-based system  
                     | ➢ Android application    | kewea@connect.hku.hk      |

The above tasks specified to each member are the main tasks for which each member is responsible. Indeed, throughout the whole FYP, we work very closely together. All of us actively participated in both front-end and back-end design and development process. So that every member can have a better knowledge and overall understanding of how our system works besides one’s main tasks.
Section 9: References

GovHK: Traffic Speed Map, Retrieved October 11, 2014 from

Google Maps API: Android, Retrieved October 13, 2014 from
https://developers.google.com/maps/documentation/android

Geodetic Survey Section: Web-based Transformation Tool, Retrieved October 19, 2014 from

Google Developers: Google Maps API, Retrieved October 19, 2014 from
https://developers.google.com/maps/?hl=zh-tw

GitHub Gist: Graph Adjacent List, Retrieved October 21, 2014 from
https://gist.github.com/tnhansel/11441647