

HKUST

# **Economic Path Calculator Based on HK Real-time Traffic Condition**

CSIT 6910A Independent Project

Supervisor: Prof. David Rossiter

Student: HUANG Xinyi

Student ID: 20222719

Email: [xhuangap@connect.ust.hk](mailto:xhuangap@connect.ust.hk)

## Content Table

1.	Introduction .....	4
1.1.	Background .....	4
1.2.	Overview .....	4
2.	Design .....	4
2.1.	Work flow and Activity Diagrams .....	4
2.1.1.	Find the path .....	4
2.1.2.	Show existing roads in database .....	5
3.	Implementation .....	6
3.1.	Technology .....	6
3.1.1.	Development Environment .....	6
3.1.2.	Programming Language .....	7
3.1.3.	Third party libraries and API .....	7
3.2.	Main Functionalities .....	7
3.2.1.	Web crawling .....	7
3.2.2.	XML Parsing .....	7
3.2.3.	Data Querying .....	7
3.2.4.	Path Computing .....	7
3.2.5.	Result Visualization .....	7
3.3.	System Components .....	8
3.3.1.	Database .....	8
3.3.2.	Core Program .....	8
3.3.2.1.	Web Data Crawler .....	8
3.3.2.2.	XML Parser .....	8
3.3.2.3.	Path Computation Program .....	8
3.3.2.4.	Database Connector .....	9
3.3.2.5.	Common Gate Interface(CGI) .....	9
3.3.3.	Server .....	9
3.3.4.	Front-end Webpage .....	9
4.	Demonstration .....	10
4.1.	Initial Webpage .....	10
4.2.	Functions Selection .....	10

Economic Path Calculator Based on HK Real-time Traffic Condition

5. Conclusion ..... 17

6. Minutes ..... 17

6.1. Minutes of the 1<sup>st</sup> Project Meeting .....17

6.2. Minutes of the 2<sup>nd</sup> Project Meeting .....17

6.3. Minutes of the 3<sup>rd</sup> Project Meeting .....18

6.4. Minutes of the 4<sup>th</sup> Project Meeting .....18

6.5. Minutes of the 5<sup>th</sup> Project Meeting .....19

6.6. Minutes of the 6<sup>th</sup> Project Meeting .....19

7. Appendix ..... 20

# 1.Introduction

## 1.1.Background

In megalopolises, like Hong Kong, the shortest driving path might not be the most economic and time-saving path because traffic condition affects the fuel consumption of cars as well as the length of path. When there is traffic congestions, cars should repeat stopping and going several times, which might wastes more times and consumes more fuel.

After seeking information from Hong Kong government[[1]], It is possible to get an more economic path for drivers to travel from place to place in one certain district. Here, in this project, "Economic path" defines as the path which causes driver encounter less traffic congestions in order to prevent waste of time or fuel.

## 1.2.Overview

In users perspective, the user inputs name of starting point and destination into the system to the current economic path from starting point to destination with direction instructions.

In terms of technology, this system is consist of four main parts: Database, Core program, Apache Server and Front-end webpage. Moreover, the main functions of the system are: web crawling, XML parsing, path computing, database designing and result visualization.

# 2.Design

## 2.1.Work flow and Activity Diagrams

### 2.1.1.Find the path

The work flow described as follow and activity shows in Figure 2.1:

- 1.When the system gets the names of S and E from a webpage through Apache server, the system geocodes two pairs of longitude and latitude from these two names;
- 2.Crawls the latest traffic condition of main roads in Hong Kong from web, which is in XML format;

3. Parse the XML files to get current condition and initialize data structure for the core program;
4. Run the core program and keep querying in database to get the result;
5. Visualize the result in the webpage;

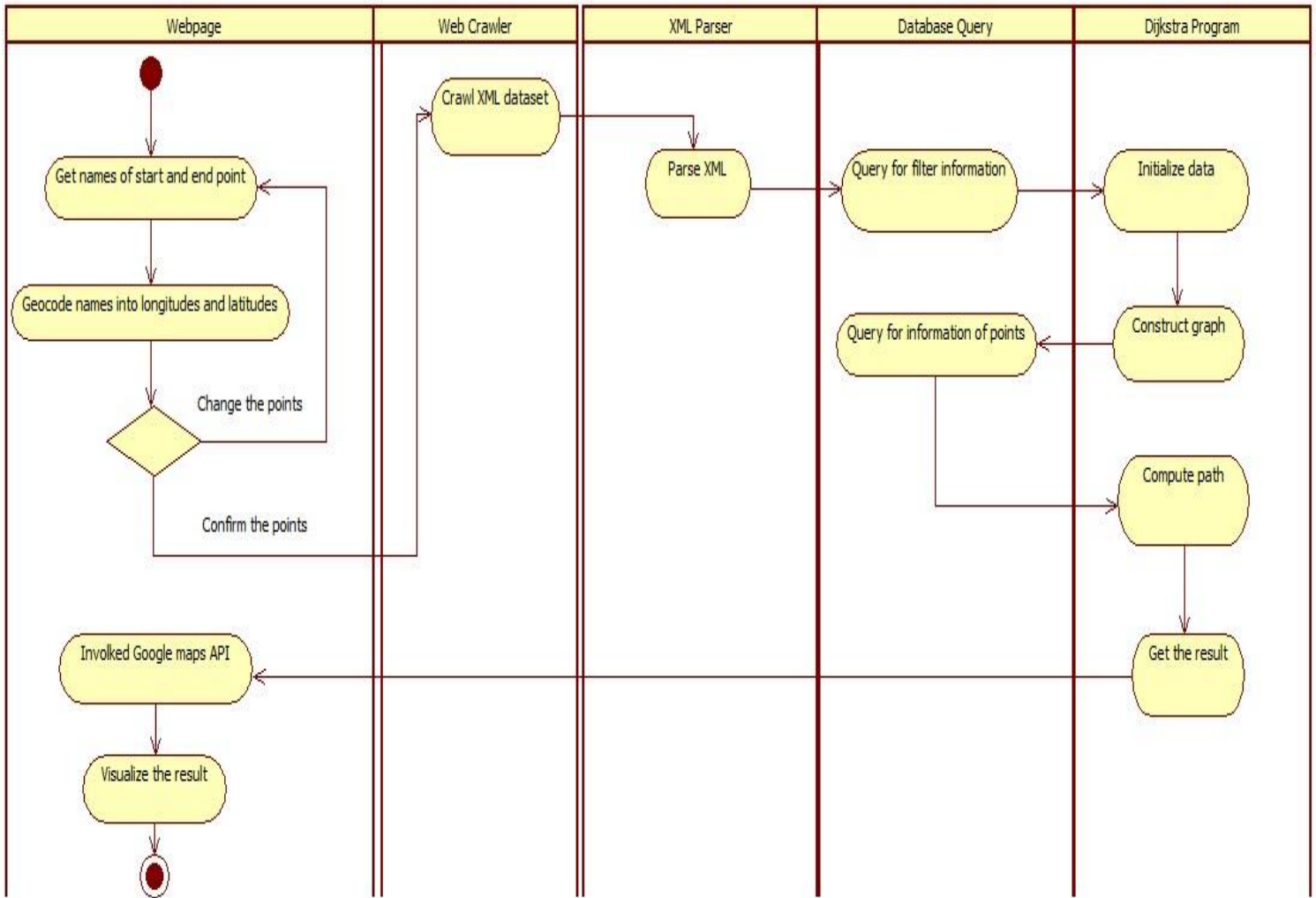


Figure 2.1

### 2.1.2. Show existing roads in database

The work flow described as follow and activity shows in Figure 2.2:

- 1) When the system gets the order to show all roads in database, the webpage keeps reading each link's information (start point and end point location) from database through background program.
- 2) When one pair of locations is read, the webpage invoked a callback function to visualize the route between start point and end point;

3)Keep loop between Step 1 and 2, if all roads are showed, stop the loop.

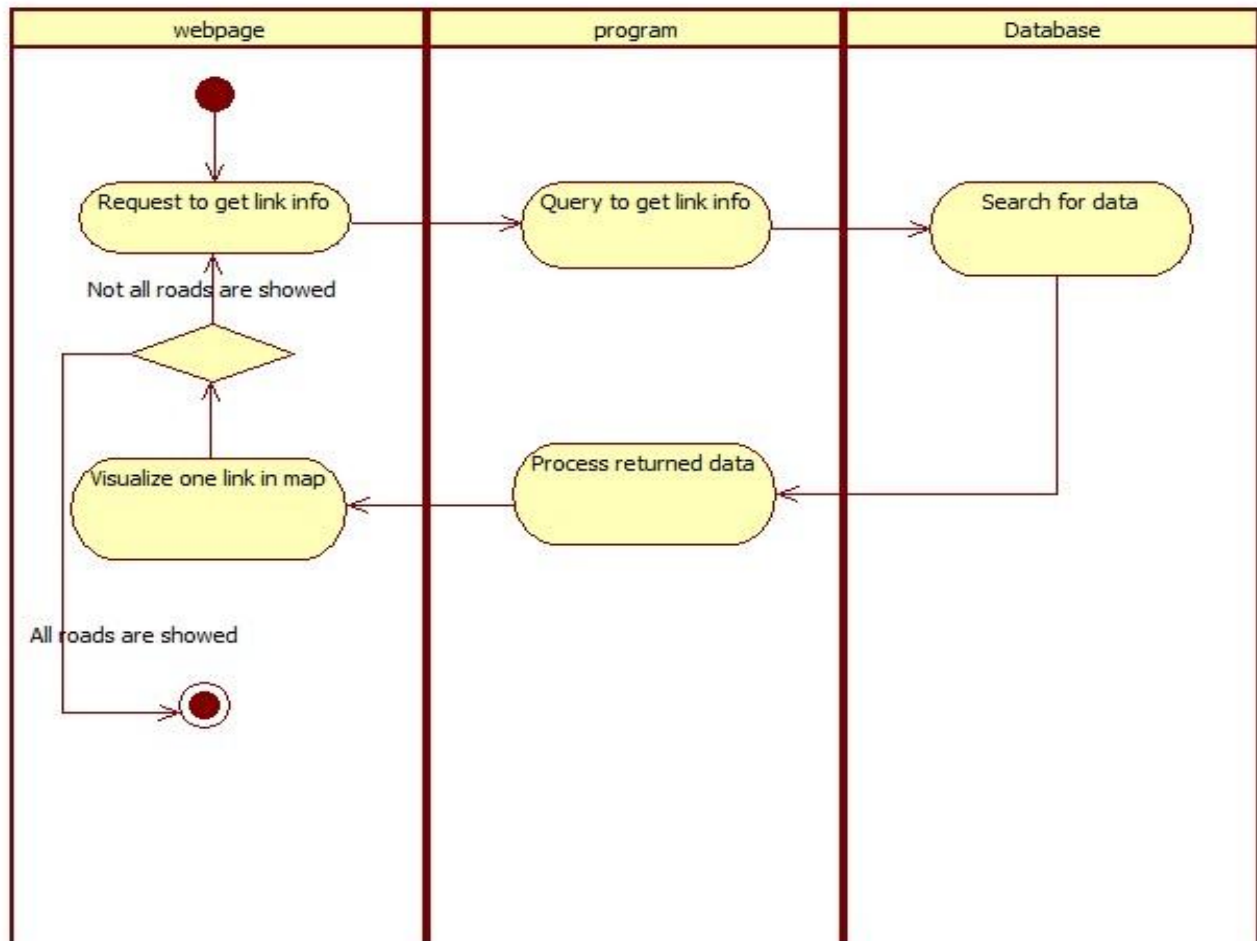


Figure 2.2

## 3.Implementation

### 3.1.Technology

#### 3.1.1.Development Environment

- Core Program IDE: Visual Studio 2010[2]
- Database: MySQL[3]
- Server: Apache[4]
- Browser: IE
- Code management: Github[5]

### **3.1.2. Programming Language**

- C++
- HTML, JavaScript, CSS
- CGI Script
- SQL

### **3.1.3. Third party libraries and API**

- RapidXml[6]
- MySQL Connector/C++[7]
- Google Maps API[8]

## **3.2. Main Functionalities**

### **3.2.1. Web crawling**

This function is triggered by the "Submit" from user, its responsibility is to download of current traffic condition of Hong Kong, which is in XML format and updates every 5 minute.

### **3.2.2. XML Parsing**

The raw data downloaded by crawler contains lots of useless information for the path computation. The system parses the raw data line by line to get the data captured time, ID of roads, Roads saturation levels, Traffic Speed.

### **3.2.3. Data Querying**

After parsing the XML and while running the path computation program, the system needs to do some queries within established database. The system implements these query functions:

- 1)Search Roads index by ID;
- 2)Search Road's index by longitude and latitude;
- 3)Search Road's longitude by index;
- 4)Search Road's latitude by index;

### **3.2.4. Path Computing**

When the requirements from user and the current traffic condition are ensured, the system computes the economic path by Dijkstra algorithm[9]. which gives out the shorted weighted path.

### **3.2.5. Result Visualization**

In finding the shortest path part, when user enters the name of start/end point, the longitude and latitude will be showed on the webpage. After confirm the entered

information, webpage gets the results from background core program. Then the webpage draw a directed line from Start Point to End point with Passed points(s). Moreover, the direction also shows as several text messages in a panel which explains how to drive from passed point to next pass point.

In showing roads in the database part, all the roads in the database will be drawn in a new webpage.

### **3.3.System Components**

#### **3.3.1.Database**

Apart from the traffic data from the government website, the system has its own database which contains the detailed information of nodes(Start points and End points) and links(Roads between Start points and End points) in map. The detailed information includes: Road ID, Start Points and End Points ID, points longitudes and latitudes.

The system uses MySQL as database and the table in database constructed based on the dataset from Hong Kong government website.[10]

#### **3.3.2.Core Program**

##### **3.3.2.1.Web Data Crawler**

Crawler uses functions which included in URLMon.dll[11] to download real-time Hong Kong traffic condition file. Only starting the crawler after users confirm the locations of start point and end point;

##### **3.3.2.2.XML Parser**

XML parser uses functions which included in URLMon.dll[12] to download real-time Hong Kong traffic condition file. Only starting the crawler after users confirm the locations of start point and end point;

##### **3.3.2.3.Path Computation Program**

The path computation part first constructs a non-negative edge ambigram by real-time traffic condition because there are lots of one-way roads in Hong Kong. Secondly, a graph search algorithm is implemented to calculate the paths cost and produce a shortest path tree. For a pair of given start point and end point, the algorithm finds the path with lowest cost.



Moreover, in this system, the path computation part uses priority queue[13] by Fibonacci heap[14] as storage data structure to improve the time complexity of the computation. The improvement method is given in [15].

#### **3.3.2.4.Database Connector**

Both XML Parse and path computation needs to query in database. The system uses MySQL database connector in C++. It creates the query statement by given information, searches the database, sorts the results and outputs the query results.

#### **3.3.2.5.Common Gate Interface(CGI)**

The system builds the interaction between webpage and program by CGI to responds to requests from web browser and generate dynamic content on webpage.

#### **3.3.3.Server**

The system uses Apache as server which allows the CGI program communicates with Front-end webpage.

#### **3.3.4.Front-end Webpage**

In the Front-end part, HTML, CSS are used to describing the look and formatting the frame. By using JavaScript, Google Maps APIs are invoked to geocode the location name, visualize result route in the map.

Besides, when invoking callback function to show all the roads in the database. The limited number of request time set by Google and the nature of callback extends the whole running of drawing all lines. The visualization of all roads might take a long time;

## 4. Demonstration

### 4.1. Initial Webpage

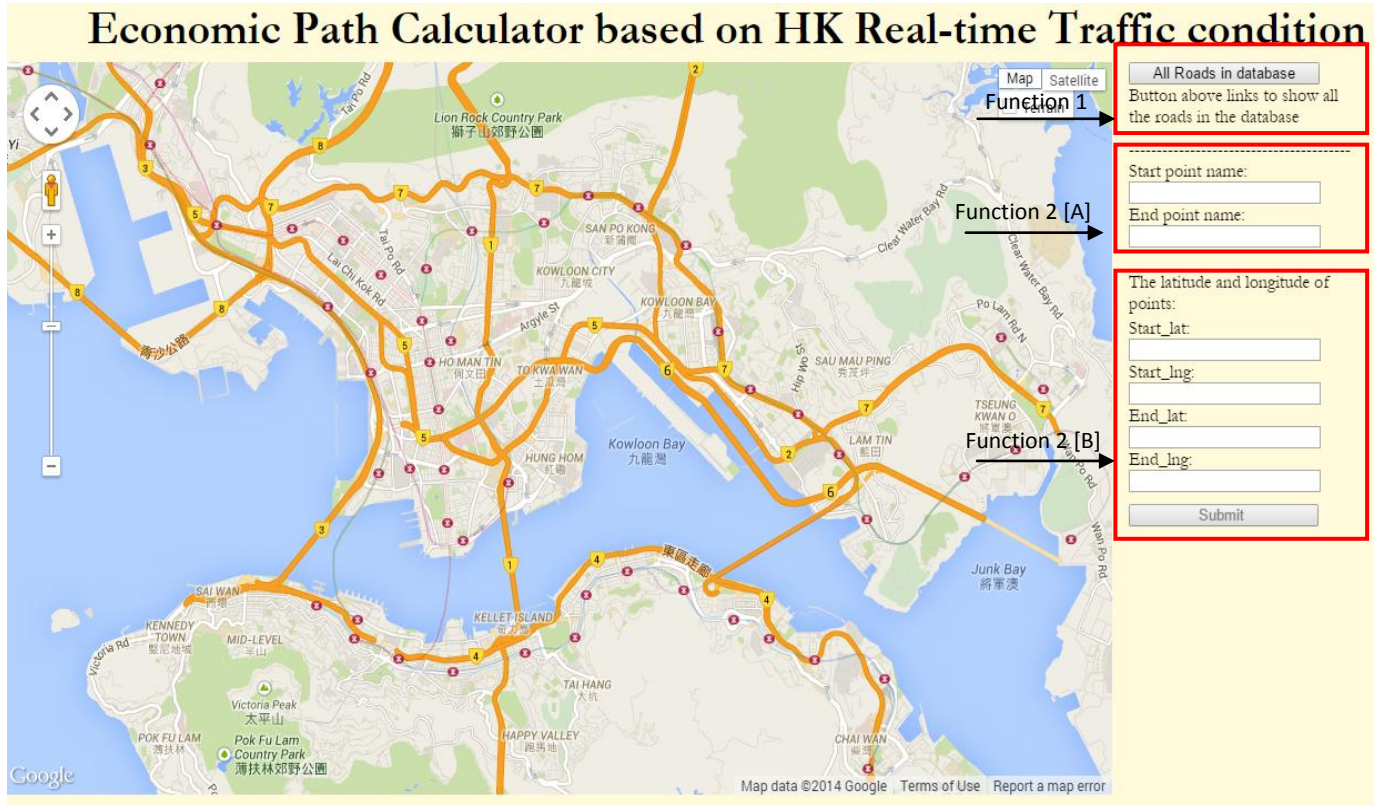


Figure 4.1.1

### 4.2. Functions Selection

In the initial webpage (Figure 4.1.1), users can select two functions:

- Function 1: After clicking the "All Roads in database" button, all roads in database could be visualized in a map (Figure 4.2.1 - Figure 4.2.2).



### End point: Sheung Wan Station

1. Enter the name of start point and end point separately, and the longitude and latitude of them will be filled in the "Start\_lat", "Start\_lng" and "End\_lat", "Eng\_lng" immediately. (Figure 4.2.3)

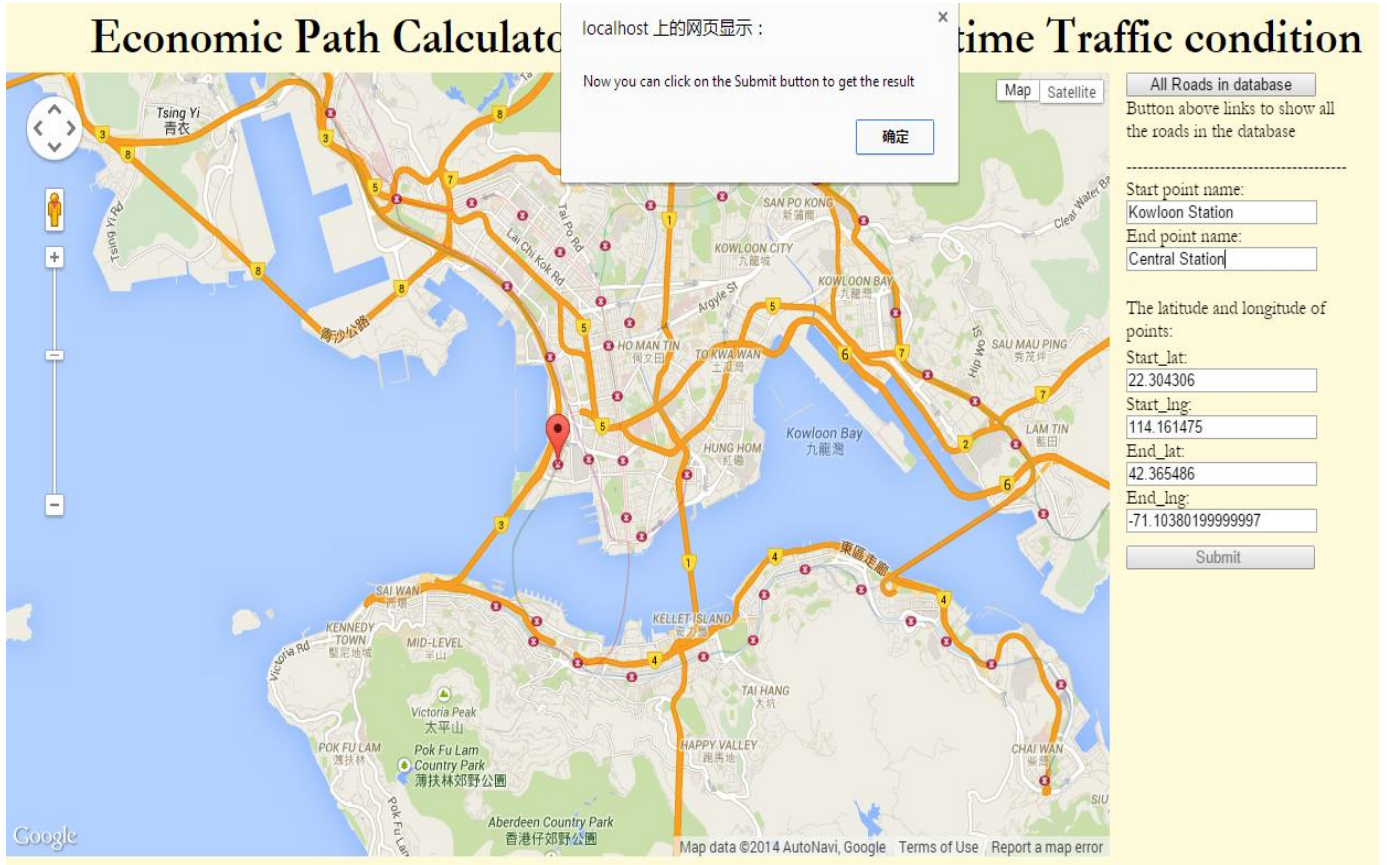


Figure 4.2.3

2. If two points are set, there will pop up a notification and "Submit" button will be activated(Figure 4.2.4).

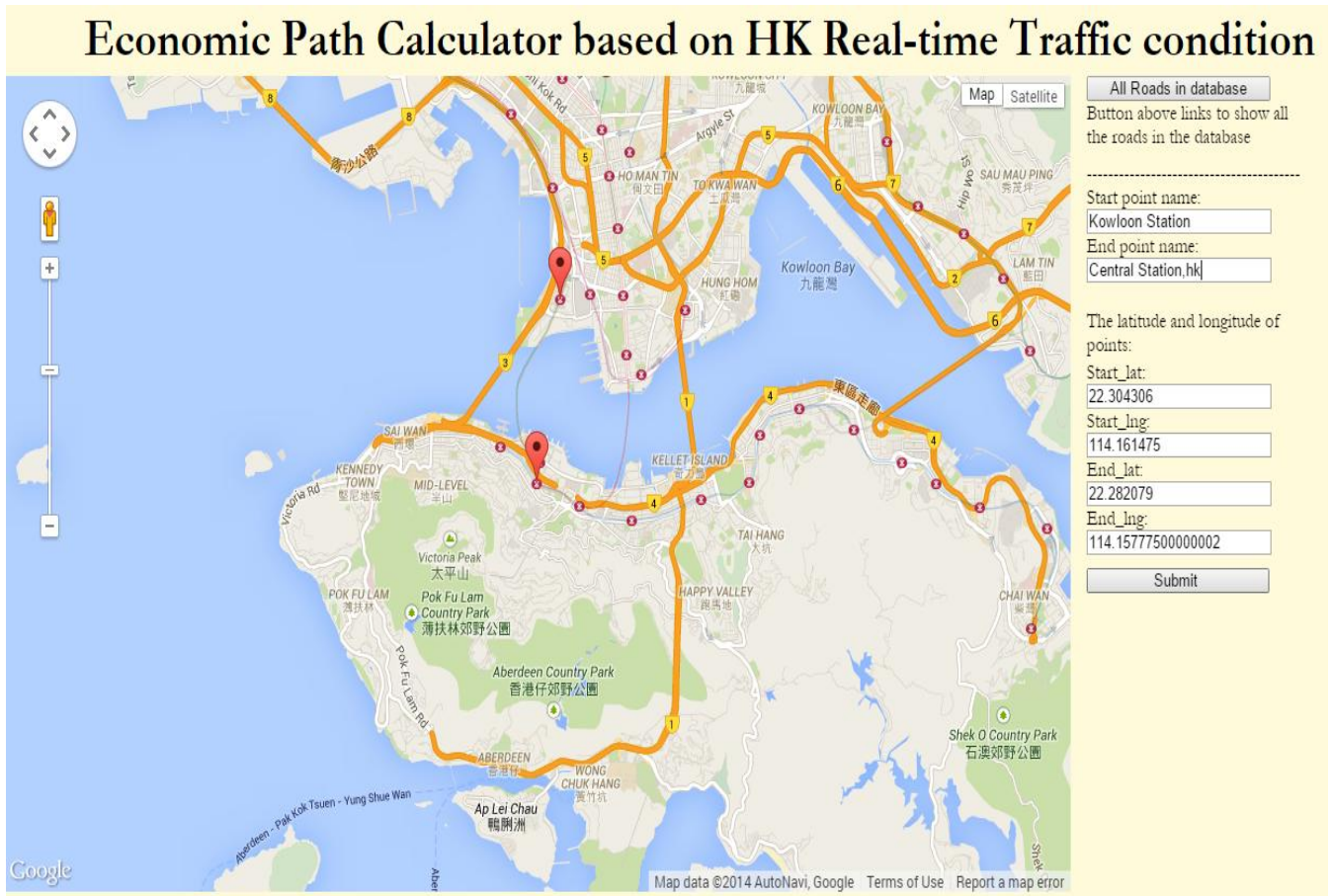


Figure 4.2.4

3. Then the user can click submit to see the result. The result will be drew as a line from start point to end point(Figure 4.2.5). The direction instruction will be showed as text message in a panel in the bottom right corner. When user clicks one of the instruction messages, there will be a dialogue showing on one points of route on the map. The result from the system(Figure 4.2.5). compares with the result from Google Map(Figure 4.2.6). Moreover, the traffic data captured time is showed in the bottom of the webpage(Areas outlined in yellow in Figure 4.2.5)



Figure 4.2.5

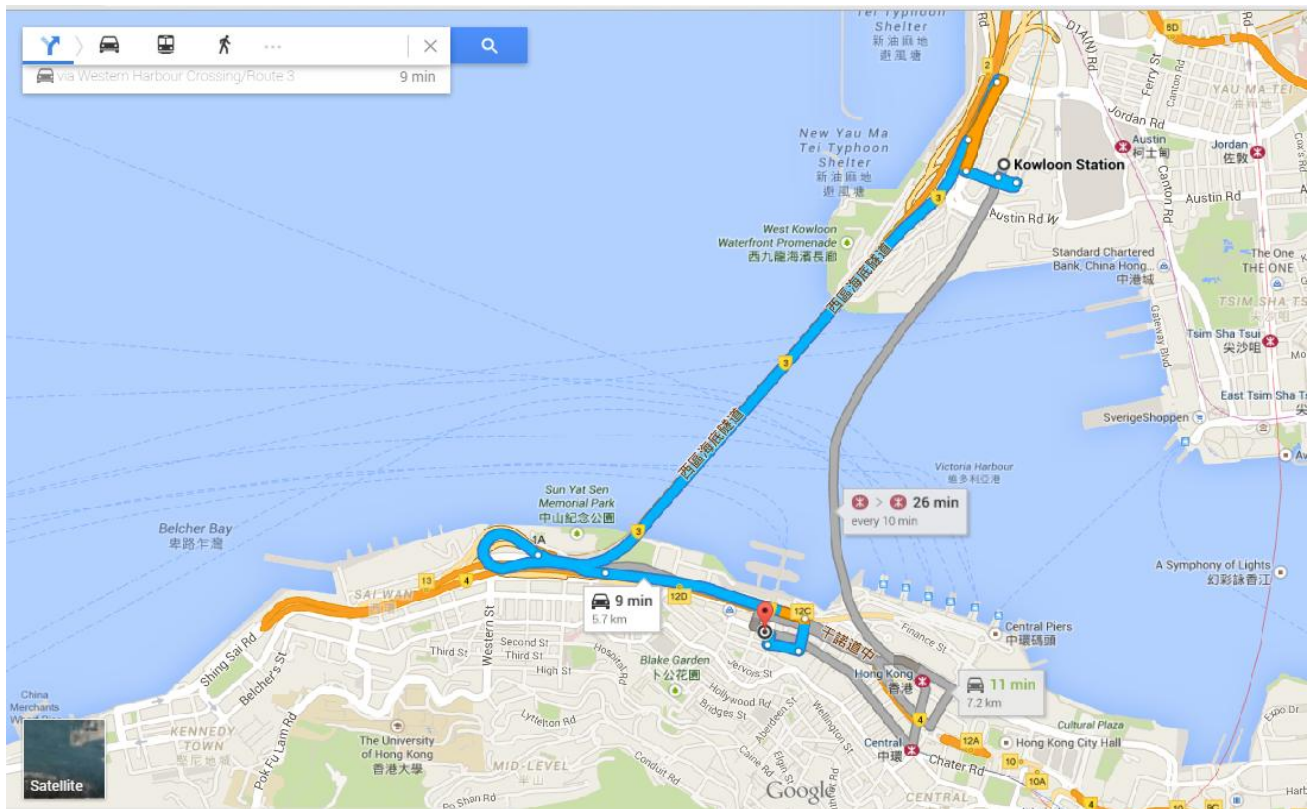


Figure 4.2.6

4. Compared Figure 4.2.5(Route A) with Figure 4.2.6(Route B), the length of Route A and Route B are almost the same, whereas they have a few difference in routes around end point. After observing the result in Function 1(Show all roads in the map), the end point in Route B is the nearest point the system have in the database because the government only provide information of main roads in Hong Kong(Figure 4.2.1 - Figure 4.2.2). However, if users want to get the path based on current traffic condition, they still can avoid the traffic congestions in main roads in Hong Kong.

**Example 2(New Terrier to Hong Kong Island)**

**Start point: Tuen Mun Station**

**End point: Island Eastern Corridor**

1. Steps are the same as Example 1;
2. The Route C from the system(Figure 4.2.7) and Route D from Google map (Figure 4.2.8) show as follows:

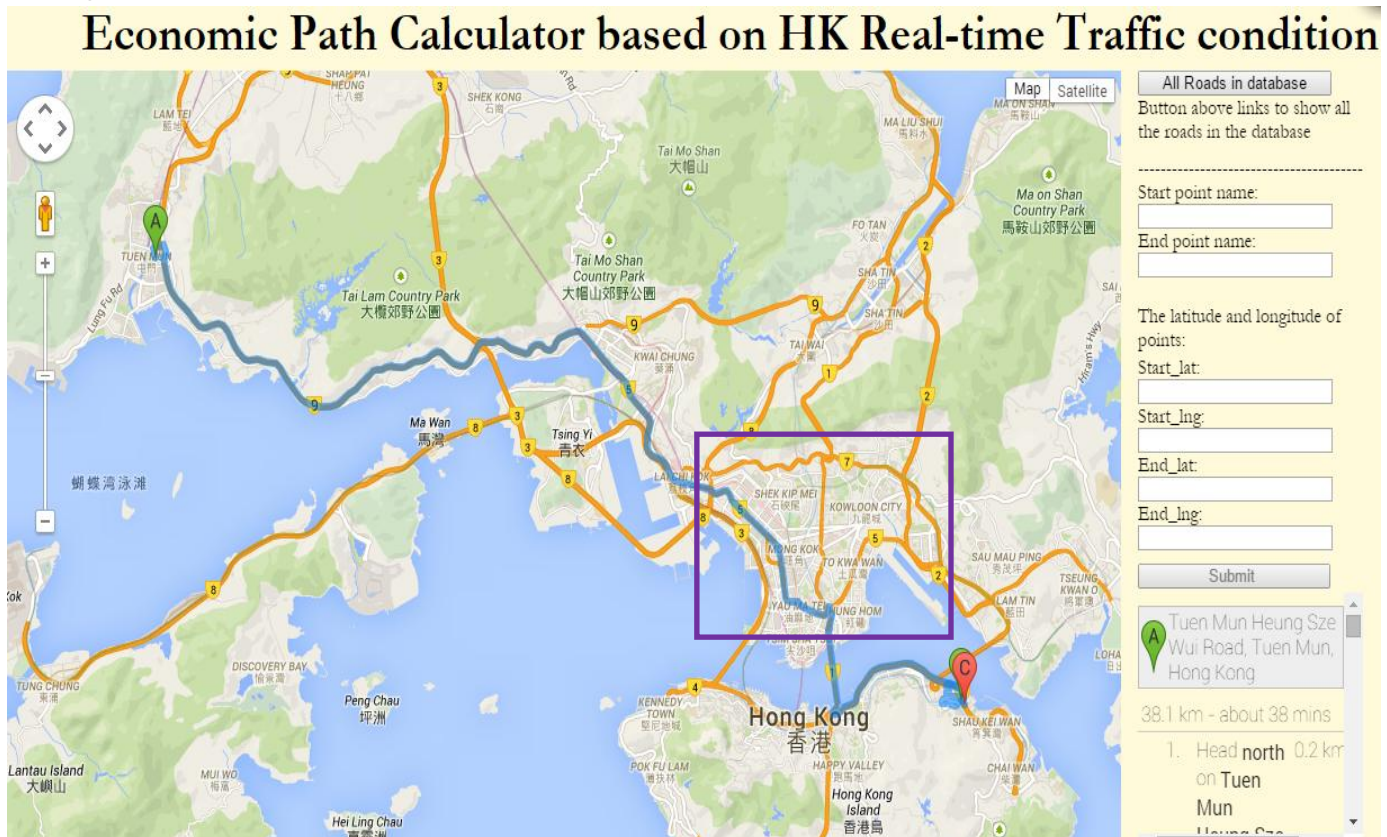


Figure 4.2.7

## Economic Path Calculator Based on HK Real-time Traffic Condition

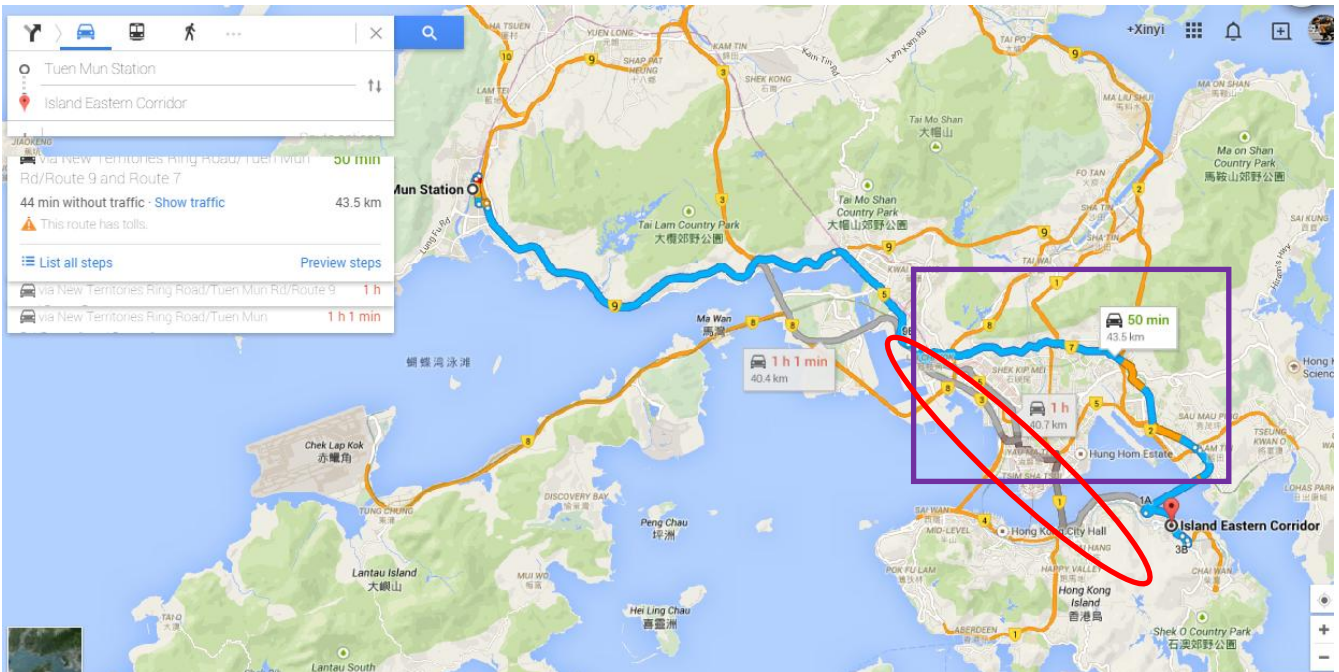


Figure 4.2.8

3. Figure 4.2.7 with Figure 4.2.8, the difference exists in Areas outlined in purple. After checking the roads on the map in this specific area(Figure 4.2.9), there is a lack of road information in that area(Area in red circle in Figure 4.2.9). Thus, the system found another route as the economic route, which is also the second choice route from Google's result(Area in red circle in Figure 4.2.8).

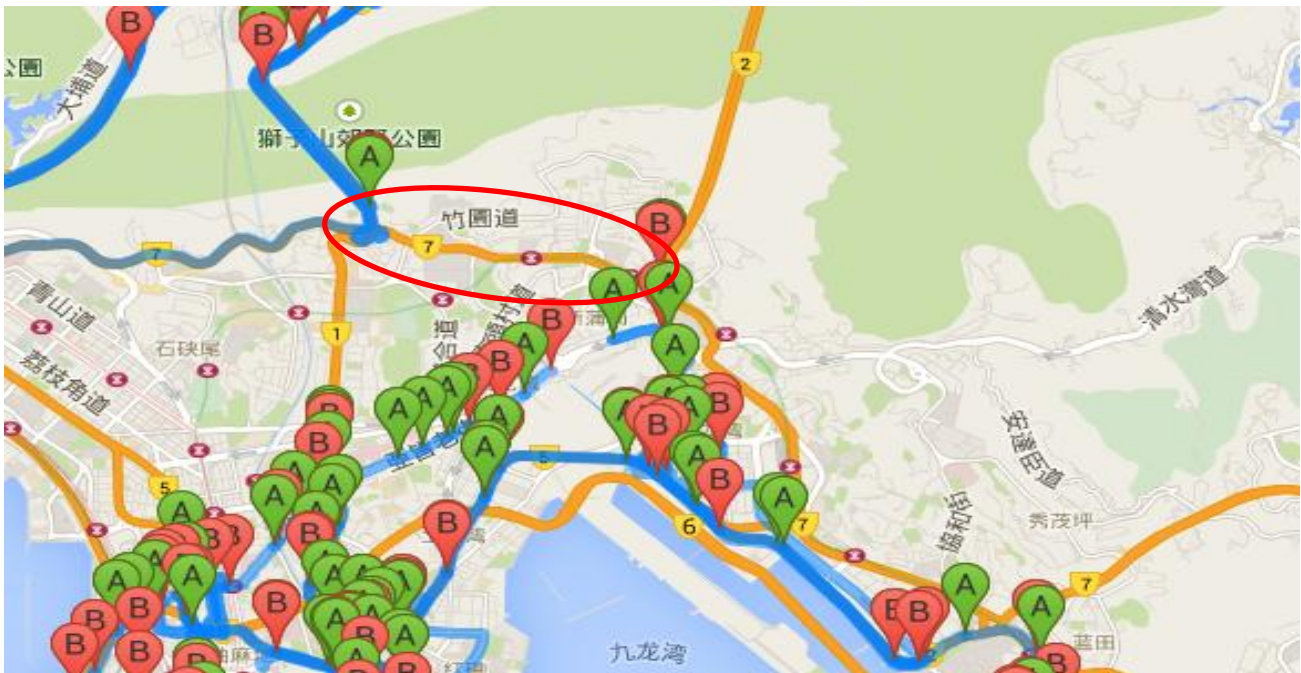


Figure 4.2.9



## 5. Conclusion

In this project, I implemented a completed system which consists of front-end, server and background program. The system can get the input of user from web browser, then crawls the real-time traffic condition data from other website. Next, the background program computes the path with the lowest cost and sends it to the server by CGI script. The final result is a path showing on the map and its text direction instruction in a panel.

Though the data from Hong Kong government is not completed, the system can work as my expectation in some parts of Hong Kong. Thus, there are still some improvements that can be made for this project except lack of data, like estimating driving time or find out alternative routes.

## 6. Minutes

There are 6 meetings with Prof. Rossiter for my project.

### 6.1. Minutes of the 1<sup>st</sup> Project Meeting

Date: Wednesday, 10 September 2014

Time: 4:30 PM

Place: Room 3512

Attending: Prof. Rossiter, HUANG Xinyi

Absent: None

Recorder: HUANG Xinyi

Approval of minutes: This is first formal meeting, so there were no minutes to approve.

Report on progress: Discussion about the expectation about the whole project

Discussion Items and Things To Do: The structure and workflow of the whole system, and consideration of implementation ways

Meeting adjournment: 5:00 PM

### 6.2. Minutes of the 2<sup>nd</sup> Project Meeting

Date: Tuesday, 14 October 2014

Time: 11:30 AM

Place: Room 3512

Attending: Prof. Rossiter, HUANG Xinyi

Absent: None

Recorder: HUANG Xinyi

Approval of minutes: I was asked to talk about the expected structure and implementation ways of the project. And the progress of core algorithm has been demonstrated.

Report on progress: Discussion about the usability and sources of data from websites.

Discussion Items and Things To Do: Confirmation of data authenticity

Meeting adjournment: 11:50 AM

### **6.3.Minutes of the 3<sup>rd</sup> Project Meeting**

Date: Tuesday, 4 November 2014

Time: 11:30 AM

Place: Room 3512

Attending: Prof. Rossiter, HUANG Xinyi

Absent: None

Recorder: HUANG Xinyi

Approval of minutes: I was asked to talk about the expected structure and implementation ways of the project. And the progress of core algorithm has been demonstrated.

Report on progress: Discussion about the usability and sources of data from websites.

Discussion Items and Things To Do: Confirmation of data authenticity and visualization of result in webpage

Meeting adjournment: 11:50 AM

### **6.4.Minutes of the 4<sup>th</sup> Project Meeting**

Date: Tuesday, 11 November 2014

Time: 11:10 AM

Place: Room 3512

Attending: Prof. Rossiter, HUANG Xinyi

Absent: None

Recorder: HUANG Xinyi

Approval of minutes: I was asked to showed the C++ program running in Visual Studio, HTML webpage, database search results with C++ connector;

Report on progress: Discussion about the connection between webpage and C++ program;

Discussion Items and Things To Do: Start to use Apache as server and build the connection between front-end and background program

Meeting adjournment: 12:10 AM

### **6.5.Minutes of the 5<sup>th</sup> Project Meeting**

Date: Tuesday, 20 November 2014

Time: 3:05AM

Place: Room 3512

Attending: Prof. Rossiter, HUANG Xinyi

Absent: None

Recorder: HUANG Xinyi

Approval of minutes: I was asked to show the Apache server worked successfully between the webpage and background program;

Report on progress: Discussion about how to capture the real-time XML files immediately after user submit the request

Discussion Items and Things To Do: Download the XML files by C++ by command send out indirectly from webpage

Meeting adjournment: 4:20 PM

### **6.6.Minutes of the 6<sup>th</sup> Project Meeting**

Date: Tuesday, 27 November 2014

Time: 3:30 PM

Place: Room 3512

Attending: Prof. Rossiter, HUANG Xinyi

Absent: None

Recorder: HUANG Xinyi

Approval of minutes: I was asked to show in integration of the whole project

Report on progress: Discussion about incomplete dataset and the method to revised the algorithm. Add another function to show all the roads in the database;

Discussion Items and Things To Do: Implement another function and optimized the webpage

Meeting adjournment: 5:00 PM

## 7. Appendix

---

- [1] <http://www.gov.hk/tc/theme/psi/datasets/>
- [2] <http://www.visualstudio.com/>
- [3] <http://www.mysql.com/>
- [4] <http://httpd.apache.org/>
- [5] <https://github.com/IsabellaGarcia/>
- [6] <http://rapidxml.sourceforge.net/>
- [7] <http://dev.mysql.com/doc/connector-cpp/en/connector-cpp-examples-complete-example-1.html>
- [8] <https://developers.google.com/maps>
- [9] [http://en.wikipedia.org/wiki/Dijkstra%27s\\_algorithm](http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm)
- [10] <http://www.gov.hk/en/theme/psi/datasets/speedmap.htm>
- [11] [http://en.wikipedia.org/wiki/Internet\\_Explorer](http://en.wikipedia.org/wiki/Internet_Explorer)
- [12] <http://www.dll-files.com/dllindex/dll-files.shtml?urlmon>
- [13] [http://en.wikipedia.org/wiki/Priority\\_queue](http://en.wikipedia.org/wiki/Priority_queue)
- [14] [http://en.wikipedia.org/wiki/Fibonacci\\_heap](http://en.wikipedia.org/wiki/Fibonacci_heap)
- [15] Leyzorek, M.; Gray, R. S.; Johnson, A. A.; Ladew, W. C.; Meaker, Jr., S. R.; Petry, R. M.; Seitz, R. N. (1957). *Investigation of Model Techniques — First Annual Report — 6 June 1956 — 1 July 1957 — A Study of Model Techniques for Communication Systems*. Cleveland, Ohio: Case Institute of Technology.