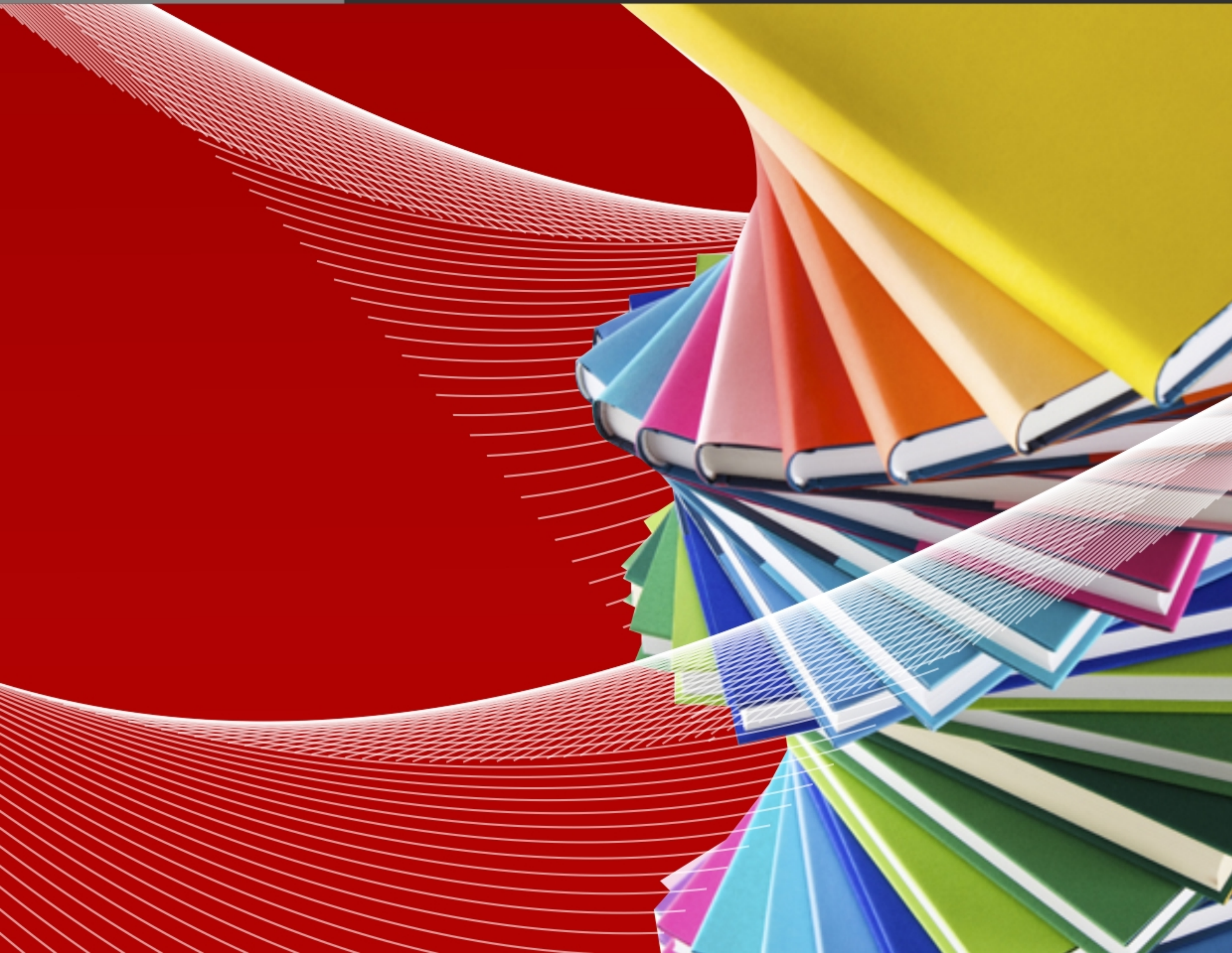
澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Outstanding Academic Papers by Students

學生優秀作品



University of Macau

Faculty of Science and Technology

澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Travel Route Recommendation System

by

IONG WENG UN, Student No: D-B1-2706-7

Final Project Report submitted in partial fulfilment
of the requirements of the Degree of
Bachelor of Science in Computer Science

Project Supervisor

Dr. Leong Hou U

21 May 2015

DECLARATION

I sincerely declare that:

1. I and my teammates are the sole authors of this report,
2. All the information contained in this report is certain and correct to the best of my knowledge,
3. I declare that the thesis here submitted is original except for the source materials explicitly acknowledged and that this thesis or parts of this thesis have not been previously submitted for the same degree or for a different degree, and
4. I also acknowledge that I am aware of the Rules on Handling Student Academic Dishonesty and the Regulations of the Student Discipline of the University of Macau.

Signature : ____________________________

Name : IONG WENG UN

Student No. : Dxxxxxx

Date : 21 May 2015

ACKNOWLEDGEMENTS

As a partial fulfilment of the requirement for the degree of Bachelor of Science, we are utmost gratitude to University of Macau for providing the opportunity to carry out a project.

In performing our final year project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much Pleasure. We would like to show our gratitude **Dr. Leong Hou U**, University of Macau for giving us a good guideline for out project throughout numerous consultations. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this project.

Many people, especially our classmates and team members itself, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our project. We thank all the people for their help directly and indirectly to complete our project.

ABSTRACT

The geotagged photos enable people to share their personal experiences as tourists at specific locations and time. Assuming that the collection of each user's geotagged photos is a sequence of visited locations, photo-sharing sites are important sources to collect the footprint of tourists. By carefully analysing their footprint movements, our objective is to extract representative and diverse travel routes.

In this paper, we propose a travel route recommendation method that makes use of (1) the travel experiences extracted from Flickr geotagged photos and (2) the landmark from Open Street Map. 639,680 geotagged photos are included in Flickr and 9,227 locations are included in Open Street Map.

We first propose a mapping algorithm to map a geotagged photo is to a landmark based on their longitude and latitude. This mapping enables us to extract important landmarks from our photo collection. Then we use these data to estimate the relationship of the landmarks based on association rule. Lastly, we apply a genetic algorithm to generate tourist routes based on different criterion to satisfy user needs.

TABLE OF CONTENTS

LIST OF FIGURES

Figure 1: Generous concept of Travel Route Recommendation.................................10

Figure 2: Generous concept of Travel Route Recommendation.................................11

Figure 3: Example of journey planner. 14

Figure 4: Another type of route recommendation system. 15

Table 5: Example of the rule...27

Figure 6: Example of visiting time evaluation...38

LIST OF TABLES

CHAPTER 1. Introduction

1.1 Motivation

Travelling is one of the most critical things of our life. We can broaden our views and explore our world through travel. Furthermore, many people find it as an efficient way to relief stress and purify soul. Therefore, a well-planned travel is very important.

Nowadays, many people prefer independent travel rather than group tour because the schedule of independent travel is more flexible and personalized that you can choose your dates of travel as well as decide your own travel itinerary. Although there are lots of benefits on independent travel, the preparation is really difficult and annoying since there are some factors need to be concerned such as budget, travel route, transportation and accommodation. In real cases, most travellers enjoy the trip but not on the planning time because they find that planning itinerary is really hard and time-wasting and many of them get trouble on it. Most travellers want to visit all famous places and attractions in a limited time but they do not know the way to begin planning. The traditional method to solve the planning problem is buying travelling books, searching travelling blogs on the internet or asking friends' recommendation. However, those methods are not efficient and objective.

We are in the age of advanced information technology today, computer is commonly used and browsing internet is the thing which we almost do every day. Because of the popularity of using computer, we want to establish a system which can make our daily life more convenience. As the above mentioned, planning travel route is inevitable in independent travel and it is not an easy work. Therefore, we want to build a tool in order to help traveller to plan their travel route more easily. Since taking photos to record travel is the thing that most people would do, we use it as a source to establish the Travel Route Recommendation System. We want to summarize the experiences of travellers, analysed the locations where most people would go and when would go to provide users a best travel route.

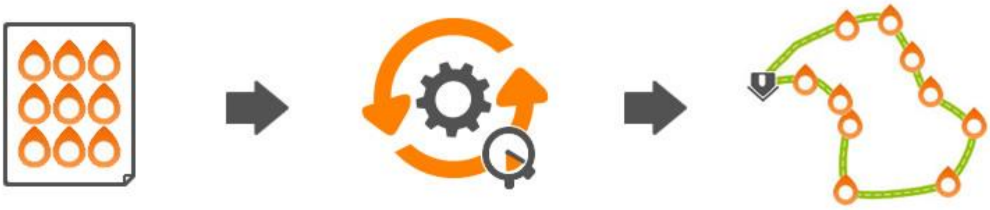


Figure 1: Generous concept of Travel Route Recommendation.

1.2 Challenge

Existing route recommendation systems have three main weaknesses. (1) They usually recommend the route using some official website information so that they cannot cover some special point that the official website does not provide or the point that only the resident in that place known. (2) They do not consider the time parameter in the point of interest. For instance, suppose there a supermarket that the opening time is nine o'clock and the closing time twenty-one o'clock. There should be recommending the user to that supermarket within the opening time. However, current existing recommendation system always ignores this point. In addition, the period of the traveller staying at the point of interest should also be a distribution not just the fixed value. (3) Existing route recommended systems usually consider the distance between two points of interest. Actually it is not enough.

To address these problems, we develop a new route recommendation system. Aim to providing a route that can depend on tourists experience in the past instead of some official data on the internet. However, it is not an easy job to collect tourist route information. It is because tourists usually planning their trip with their own computer or just writing a blog. First, it is hard to find the source since we are hard to define which one should be included. Second, it is impossible to parse the data that we fetched on the internet due to the format is not the same. Third, each article provided different information so that it is difficult to do the analysis. As a result, we cannot provide useful information.

With the growth of social network site and global positioning system widely use, we find out one of possible solution in here. Nowadays, many people like posting photos on social network site. Thus, we are able to collect the geotagged photos on the social network site, then according to the user who posted the photos and the date, who posted, we can easily to figure out that user's route in the past. Furthermore, this route has detail information about the time that the user visited and also the coordinate.

Despite of we come up with the solution of the source data. The task to fetch source data is much more difficult than imagine. There are some challenges for us to collect the users' data. It is because the users always setting the limitation that who can see the photos and only friend can view is the common setting nowadays. The social network site will have limitation also. In addition of this, geotagged photo is much less than the photo without geotagged relatively.

In analyse part, we choose association analysis as our main analyse method. It is the most common and popular method that widely use to discovering the interesting relations between variable in large databases. As a result, we occur a problem in our analyse module. The data is really discrete due to the data is fetched on the social network site so that the point of interest's support is really low. Therefore, we need to find out a method that can increase the support, in the meanwhile make sure the data is used completely so that we can create a trusty weight table.

Figure 2: Generous concept of Travel Route Recommendation.

1.3 System overview

With the popularity of smart phones and digital cameras that embed GPS technology, numerous geotagged photos were produced in these recent years. As a result, huge amounts of travellers recorded their journeys with photos and all photos are reflecting their travel behaviours and experiences. Therefore, there is a lot of useful travellers' knowledge accumulated on the internet. Those data are important and valuable for find the travellers' interest place and their travellers' route which can help other travellers as reference experiences or government to planning the public transportation.

The task of Travel route recommendation system is going to provide a custom route in city base on other travellers' record. And our work is to mine out the travel route in a city from huge amount of photos form Flickr. We have to find out the most useful data from the photo. When the huge amount photos goes together and we have to discover the relation between photos or users. It may be the user interest, sequence of photos or spending time. First of all, we choose Taipei as our testing city because which language is using Chinese and we have some travel experiences in there. All photos data is capture from Flickr legally.

Every single photo's data basically includes some information about the photo taken time, taken location's latitude and longitude. But all the locations of photos are dispersed and difficult to identify the taken place. We use Open Street Map's point of interest (POI) database and filter out other POI except viewpoint. After POI is ready, we used MPR (Minimum Bounding Rectangle) method to group the photos and it can easier to identify this area's photos is for which place. Also using the POI can classify the photos location to a place. Each place has a group of photos which are identified. Therefore, the traveller can be distinguished in three types (shopping, visit, food) according to the POI types of place. And we can measure the most popular visit time in a place by those photos taken time in one day. Similarly, the staying time in that place also can be analysed by traveller first photo and last photo taken in that place. The advanced way is to combine a traveller's photos in a group, and then we process many groups' photos to figure out the location sequences of all users.

Furthermore, all the traveller routes, place type, location, visit time and distances will process by the algorithm which included three method Genetic Algorithm, distance between the place and visit time. It calculate the weight for each place between other which is to evaluate the most reasonable route by those method, system will assemble all places rate, visiting time, traveling type and users to consider things like starting traveling times, visiting place types and starting places etc.

Finally, the Travel route recommendation system which will provide a recommend travel routes for user base-on many previous experiences from other travellers.

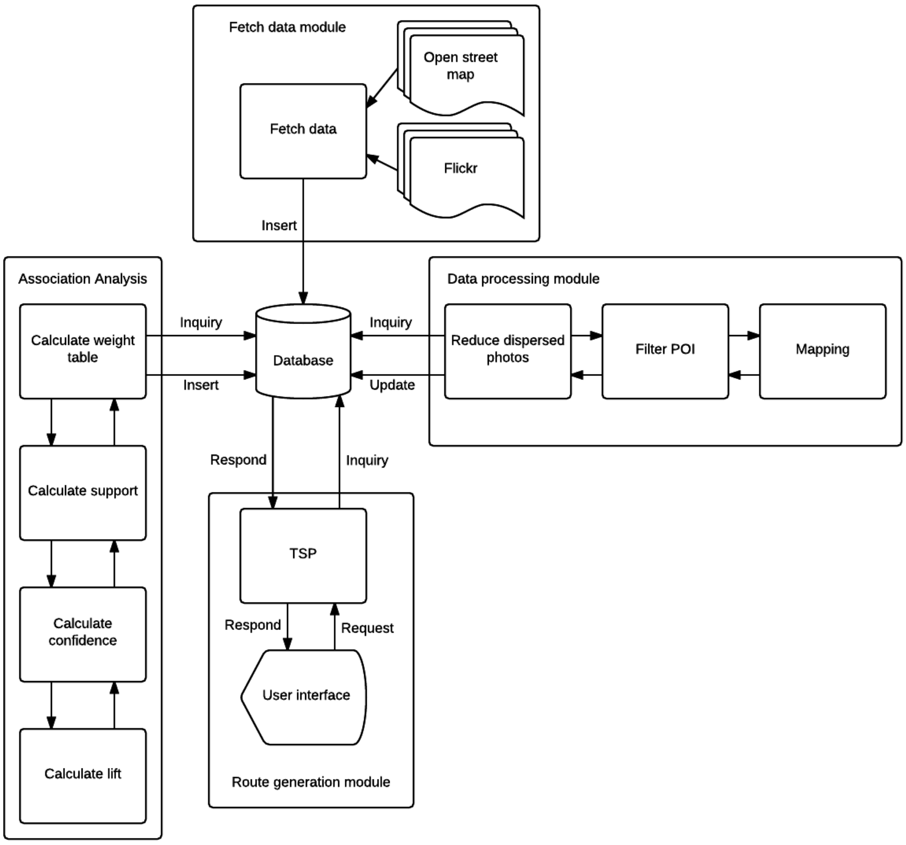
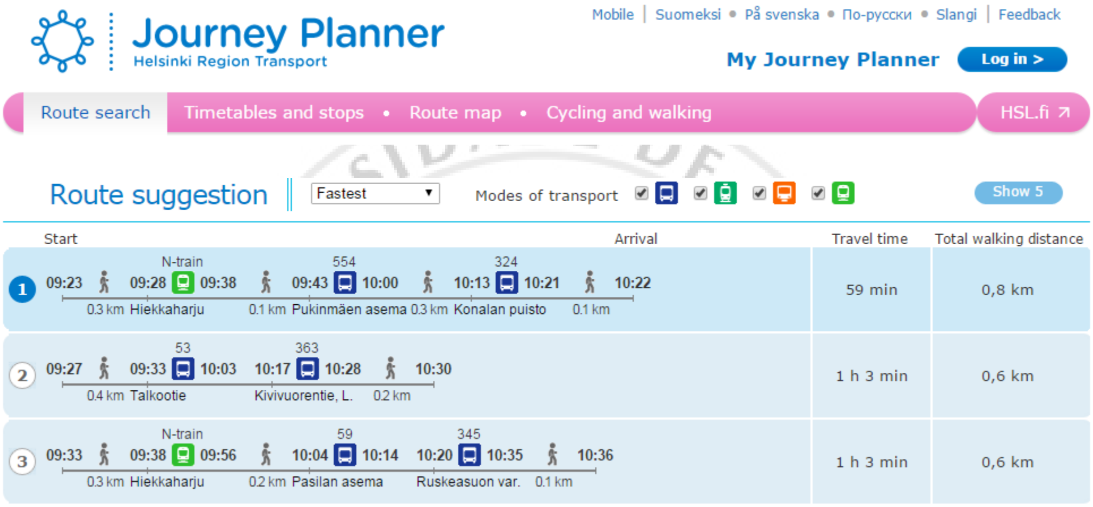


Figure 3: System overview.

CHAPTER 2. Literature survey and related work

2.1 Current exist system

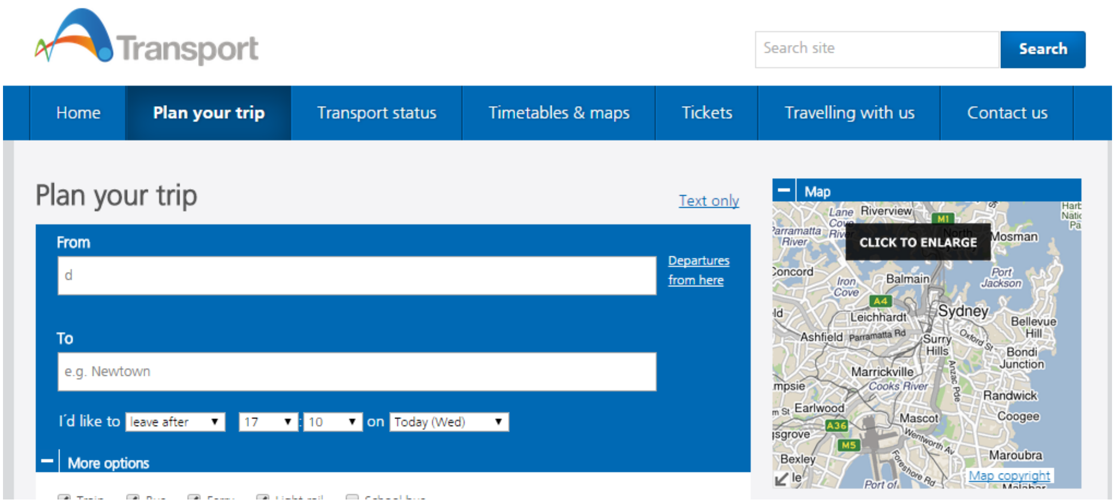
Our concept of the system is user select their interest place and system will generate a route ordered by other place food print. We have some research for this topic that most travel/Journey planning website or apps are separate into two categories. One is provide two points transportation in a city. This journey planning system only provide you to select two point in a city and they will suggest three route to go there but the traveller most force on the viewpoint but most travel have no idea where is the popular place so that the information is not comprehensive.


Journey Planner
Helsinki Region Transport
Mobile | Suomeksi • På svenska • По-русски • Slangi | Feedback
My Journey Planner
Log in >
Route search
Timetables and stops • Route map • Cycling and walking
HSL.fi
Route suggestion
Modes of transport
Show 5

Fastest

	Start / Arrival	Travel time	Total walking distance
1	09:23 – 0.3 km Hiekkaharju – N-train 09:28 – 09:38 – 0.1 km Pukinmäen asema – 554 09:43 – 10:00 – 0.3 km Konalan puisto – 324 10:13 – 10:21 – 0.1 km – 10:22	59 min	0,8 km
2	09:27 – 0.4 km Talkootie – 53 09:33 – 10:03 – Kivivuorentie, L. – 363 10:17 – 10:28 – 0.2 km – 10:30	1 h 3 min	0,6 km
3	09:33 – 0.3 km Hiekkaharju – N-train 09:38 – 09:56 – 0.2 km Pasilan asema – 59 10:04 – 10:14 – Ruskeasuon var. – 345 10:20 – 10:35 – 0.1 km – 10:36	1 h 3 min	0,6 km

Figure 3: Example of journey planner.



Plan your trip

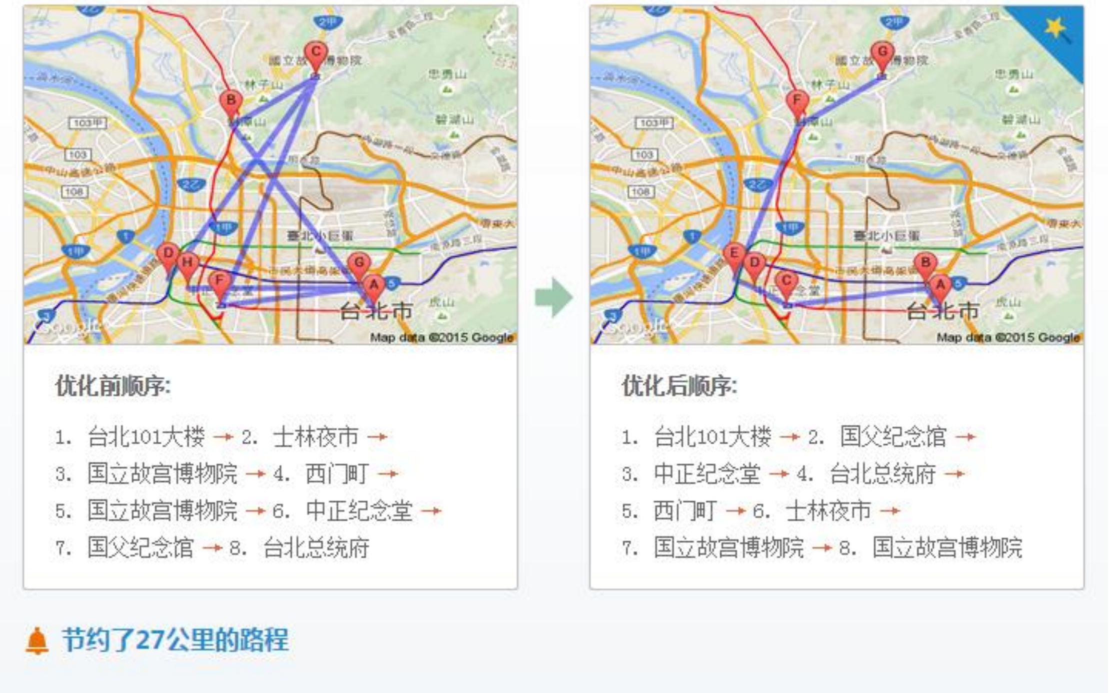
From: d

To: e.g. Newtown

I'd like to leave after 17 : 10 on Today (Wed)

Figure 5: Example of transport.

Another type website is providing other users travel planning. Both of them are inadequate, in first category that. They just allow user to select some popular place in the city base on other user's pervious travel planning. We also find that some website have provided travel route optimization but only focus on the distance and not independent location opening and closing time or the transportation. For example in qyer.com which is a screen capture in route optimization but the result is not acceptable because they arranged our route such as museum to the last of our route. Actually, if I follow their route and when I arrive the museum will be in evening. In my initial guess their system is only using one method to calculate the distance of each location and find out a shortest way for the suggestion. But this is very not enough for a route recommendation because the transportation and place visit time are also importance for user. If you suggest a route is not reasonable that this result will let user confused.



优化前顺序:

1. 台北101大楼 → 2. 士林夜市 →
3. 国立故宫博物院 → 4. 西门町 →
5. 国立故宫博物院 → 6. 中正纪念堂 →
7. 国父纪念馆 → 8. 台北总统府

优化后顺序:

1. 台北101大楼 → 2. 国父纪念馆 →
3. 中正纪念堂 → 4. 台北总统府 →
5. 西门町 → 6. 士林夜市 →
7. 国立故宫博物院 → 8. 国立故宫博物院

节约了27公里的路程

Figure 4: Another type of route recommendation system.

2.2 Our system

For our route generation module, we use the generic algorithm to solve problem. About generic algorithm, we reference article “Genetic Algorithms: Concepts and Applications” [1]. This paper introduces genetic algorithms as a complete entity, in which knowledge of this emerging technology can be integrated together to form the framework of a design tool for industrial engineers. An attempt has also been made to explain “why’’ and “when” genetic algorithm should be used as an optimization tool. In addition of this, this paper also explains the limitation of genetic algorithm and the advantage that using parallelism computing on genetic algorithm. Therefore, we get the basic idea of genetic algorithm so that we come up with the idea of how to design our route generation module.

For the purpose of the efficiency in genetic algorithms, Korea’s article “An efficient genetic algorithm for the traveling salesman problem with precedence constraints” [2] provide us an efficient genetic algorithm to solve the travelling salesman problem with precedence constraints is presented. The key concept of the proposed genetic algorithm is a topological sort, which is defined as an ordering of vertices in a directed graph. Also, a new crossover operation is developed for the proposed genetic algorithm. The results of numerical experiments show that the proposed genetic algorithm produces an optimal solution and shows superior performance compared to the traditional algorithms. Therefore, we have an optimal solution to reduce the running time on the calculation of genetic algorithms.

For the selection method in genetic algorithms, we also do the study. We read the paper “Comparison of Performance between Different Selection Strategies on simple genetic algorithms” [3]. This paper presents the comparison of performance on a simple genetic algorithm using roulette wheel selection and tournament selection. A simple genetic algorithm is mainly composed of three genetic operations, which are selection, crossover and mutation. With the same crossover and mutation operation, the simulation results are studied by comparing different selection strategies which are discussed in this paper. Qualitative analysis of the selection strategies is depicted, and the numerical experiments show that simple genetic algorithm with tournament selection strategy converges much faster than roulette wheel selection.

For the number of generation in generic algorithm, we read the article “Population Size in Gas for TSP” [4]. This paper describes a simple estimate of appropriate population size in a genetic algorithm for travelling salesman problem whose crossover operator builds offspring tours using edges from parent tours. The estimate sets the population size large enough that the initial population is likely to include every edge in a shortest tour of the problem instance's cities. The estimate is tested in a straightforward steady-state genetic algorithm for travelling salesman problem that applies an edge-based crossover operator but not a mutation operator. In runs on several well-known instances of travelling salesman problem, the algorithm's performance deteriorates with populations significantly smaller or larger than the estimate recommends. Finally this article figure out that the max number of generation should be *2log2(N).*

For our data analyse module, we use the association analysis to reach the goal. Having read the article “Association Analysis: Basic Concepts and Algorithm” [5]. We understand a methodology known as association analyses, which is useful for discovering interesting relationships hidden in large data sets. Also we understand the basic concept of association analysis.

In order to further understand association analysis, we read the article “Support vs Confidence in Association Rule Algorithms” [6]. We discover that there are currently a variety of algorithms to discover association rules. Some of these algorithms depend on the use of minimum support to weed out the uninteresting rules. Other algorithms look for highly correlated items, that is, rules with high confidence. In this paper it present a description of these types of association rule algorithms and a comparison of two algorithms representative of these approaches, with the aim of understanding the pros and cons of the support- and confidence-based approaches. Therefore, we understand the pros and cons of the support and confidence. Also we read the article “Standardising the Lift of an Association Rule” [7], this article introduction the idea of lift. The lift of an association analysis is frequently used, both in itself and as a component in formulae, to gauge the interestingness of a rule. Also this article provides a method of visualising standardised lift, through the relationship between lift and its upper and lower bounds.

CHAPTER 3. System requirement

3.1 Environment

Programming language: Php, JQuery, Java

Server: Window server

Database: MySQL

3.2 Data source

We have two main data source Flickr and Open Street Map and designed two data collect module to both source. Flickr is more easy because which provide an API directly to access the data. But also have some limitation. And Open Street Map provided a very large data package and we have to build a program to read all useful data and copy out. In this project, we focus on Taipei city which is because of the language is Chinese and we have experience to visit this city then easier to evaluate the result of this system.

Flickr is a popular image hosting website which hosted over 5 billion photos and provides an Application Programming Interface for non-commercial use. Developers allow accessing all public photo and photo related data such as user id, taken date, image id, coordinate and so on. But there are some limitation for free access that is each Application Programming Interface access only responses 500 latest record and not allow to access over 5000 times per hour. The source we collected is from 1st January, 2010 to 1st September, 2014. Finally we got 639,680 photos record from Flickr.

Open Street Map is a free editable map of the world and open data for public. Offical will pubish the data package every month but which is a very large data over 20GB. Finally we got 9,227 point of interest from Open Street Map in Taipei. But this collection of point of interest includes some type we do not have to use, such as road, hospital, gas station etc.

CHAPTER 4. Application system software design

4.1 System database design

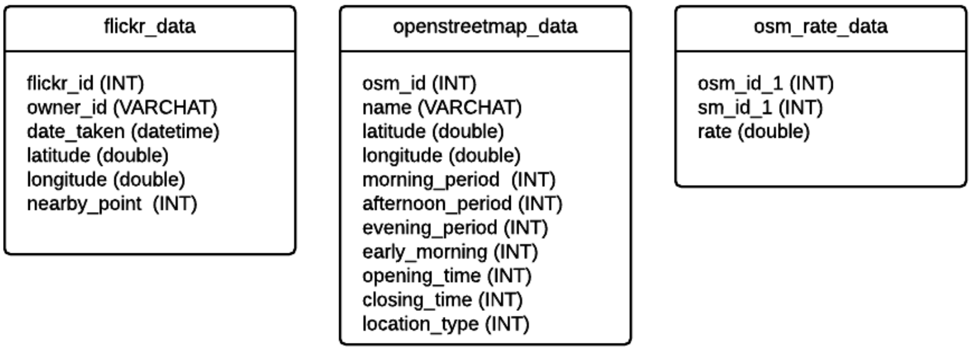


Figure 7: Database diagram

4.2 Fetch data module design

Travel route recommendation system used two data source for photos data processing, route analyses and viewpoint located. In this project, we focus on Taipei city which is because of the language is Chinese and we have experience to visit this city then easier to evaluate the result of this system. Therefore, our target data is all Taipei photos in Flickr in past 4 years and OpenStreetMap (Point Of Interest) in Taipei city.

4.3 Data processing module design

4.3.1 Reduce noise

The distribution of photo is dispersed. In order to deal with this problem, we have following steps. First, reduce dispersed photo. Second, filter point of interest. At last, map photo with point of interest

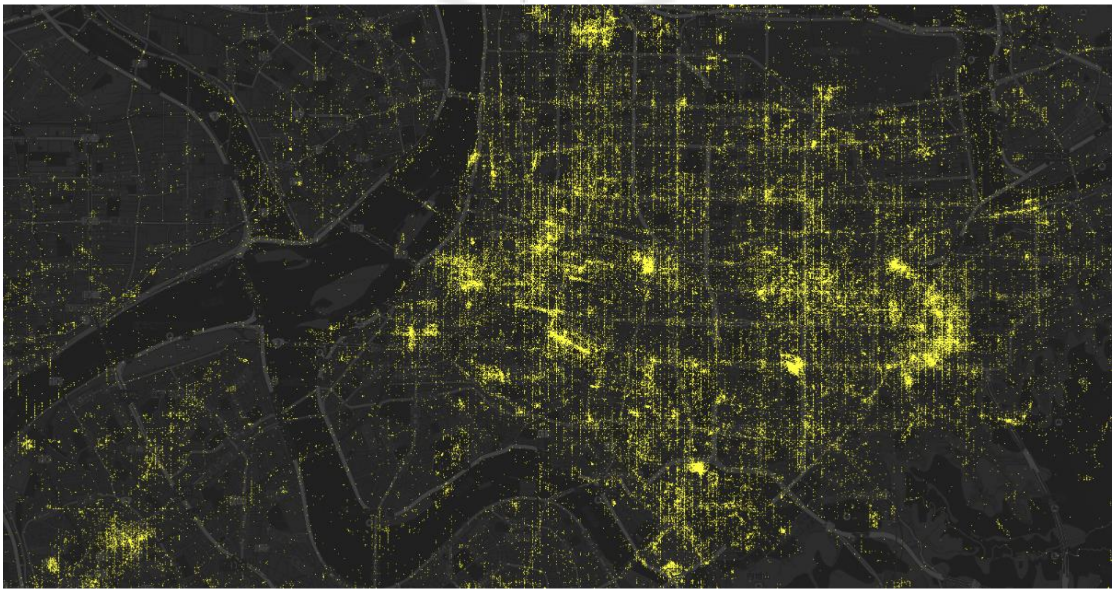


Figure 8: Original distribution of photo.

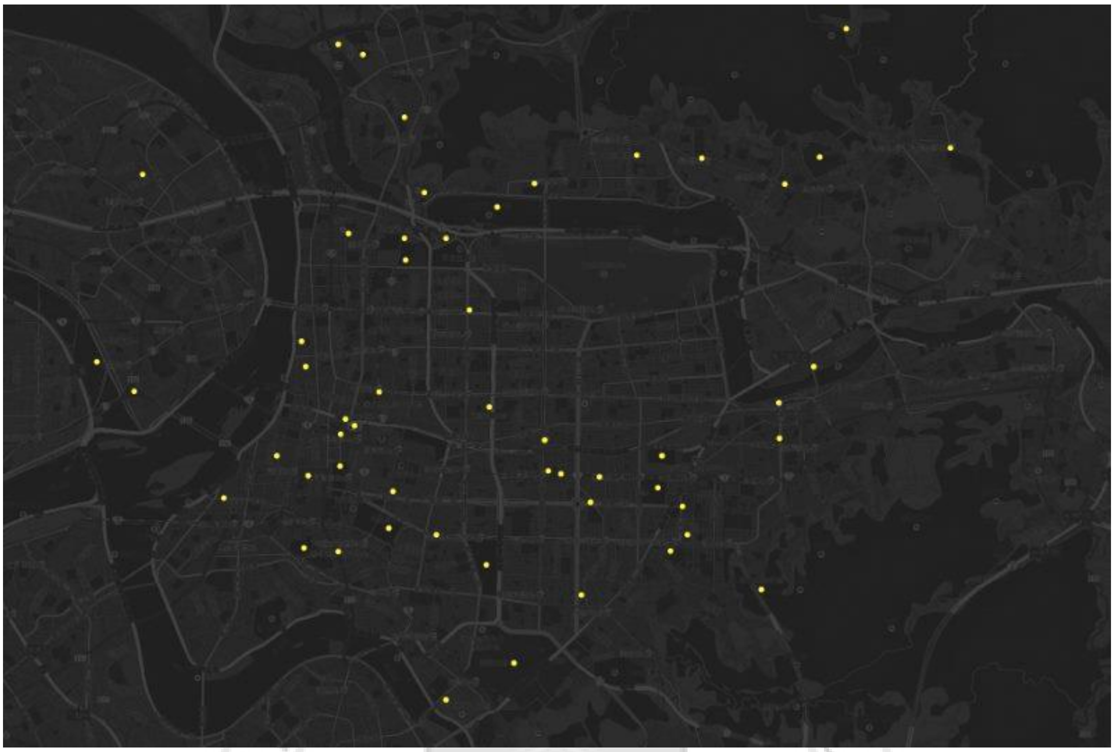


Figure 9: Distribution of photo after processing.

4.3.2 Time distribution

The photo taken time is useful information also. Therefore we calculate the distribution of it.

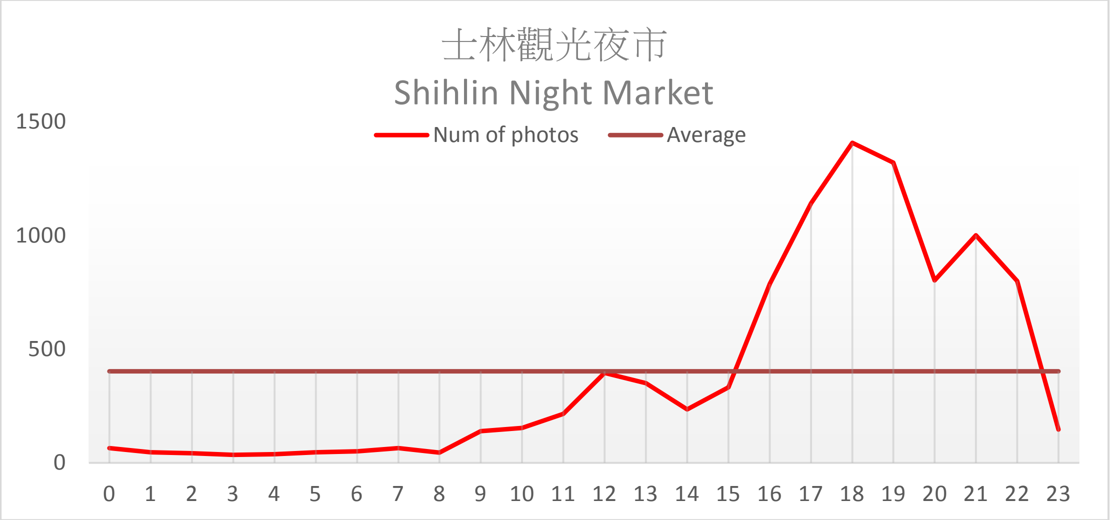


Figure 10: Calculate 4 years photo data in SNM and order by the taken time.

CHAPTER 5. Application System Methodology

In this part, we will concrete explain how our module implementation and the difference between current technology. Our application system totally has four modules: fetch data module, data processing module, data analyse module and route generation module. The modules that I take responsible for data analyse module and route generation module. Therefore, I will explain the detail of them.

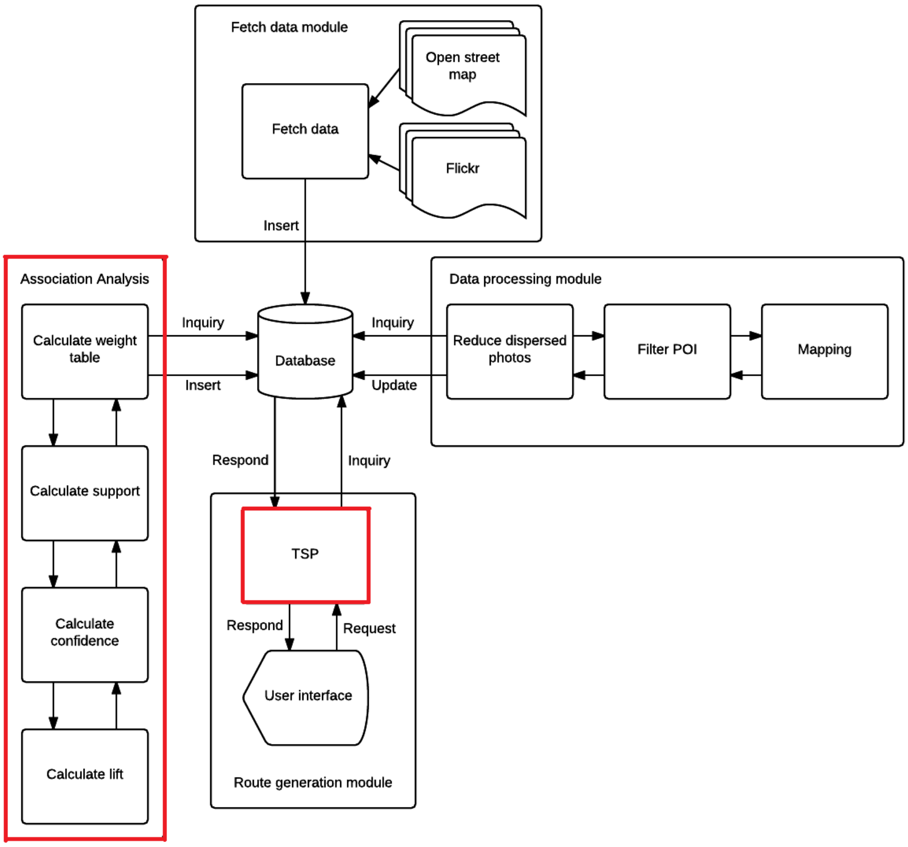


Figure 11: System overview that the part I take responsible for.

5.1 Data analyse module

In this part, we will detail explain our analyse module. This module mainly has five sub-modules: rebuild route module, first level support calculation, second level support calculation, confidence calculation and lift calculation. The main function of this module is analyse tourist’s behaviour, and then found out the weight between point of interest. In order to reach our goal, we use association analysis as the base of our analyse module.

5.1.1 Introduction of association analysis

Before we explain what association analysis is. We want to tell you a store first. One day, Wal-Mart who is an analyst discovers that quite a few diapers and beer always are purchased together every Friday. After further market research, Wal-Mart discover that women often remind her husband to help children to buy diapers after work and the meanwhile husband will buy a few cans of beer enjoying the weekend. Diapers and beer are putting together with this market awareness so that the sales of diapers and beer grow as much as three per cent. This is what association analysis can do with us.

Association analysis is usually used in market-basket analyse. The purpose is to figure out the trends from the data. The data we analyse always is large, discrete and also heterogeneity. There are two important parts in association analysis, one is support and another one is confidence.

Support is representing the probability of a specific item set that occur in the total transactions. Frequency item set is the item set that support value greater or equal to the mini support.

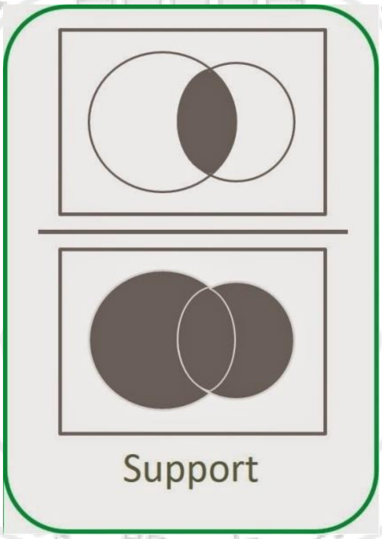


Figure 12: Relationship diagram of support.

Confidence is represent the probability of occur event Y under occur event X. This is also known as rule. Association rule is a rule that satisfy with mini support and also mini confidence.

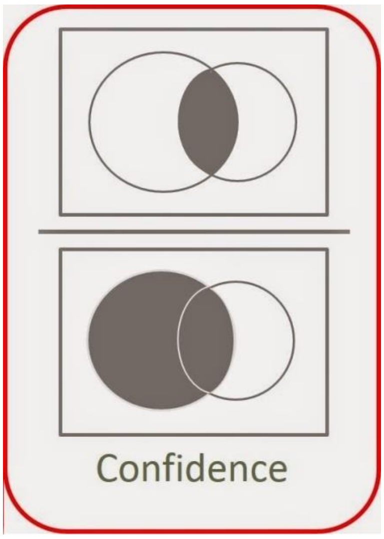


Figure 13: Relationship diagram of confidence.

Sometimes confidence is high however it may not bring us benefit. Since event Y and event X both support is high so that their confidence will high only. Therefore we need to use life to estimate our rule. Lift is the per cent of event X=>Y confidence over its support. When lift greater than one, the rule is useful, otherwise the rule is useless.

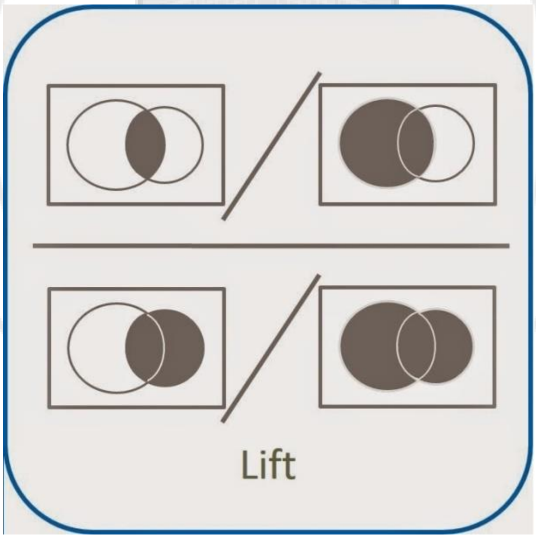


Figure 14: Relationship diagram of lift.

5.1.2 Why association analysis

Association analysis is useful for discovering interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of association rules or sets of frequent items. The data we trying to analyse is large, discrete and also heterogeneity. Therefore, it is really suitable for apply association analysis in our module.

5.1.3 Data analyse module structure

Route is the collection store the result that generate from rebuild route function. SupportTable is a hash map that records the result of calculateSupport. Rule is record two ids scores. Ruleset is the collection of rule. Analyse is the main function in data analyse module. This function accepts routes then generate the ruleset, calculateSupport, calculateConfidence, calculateLift is the function that will called by analyse.

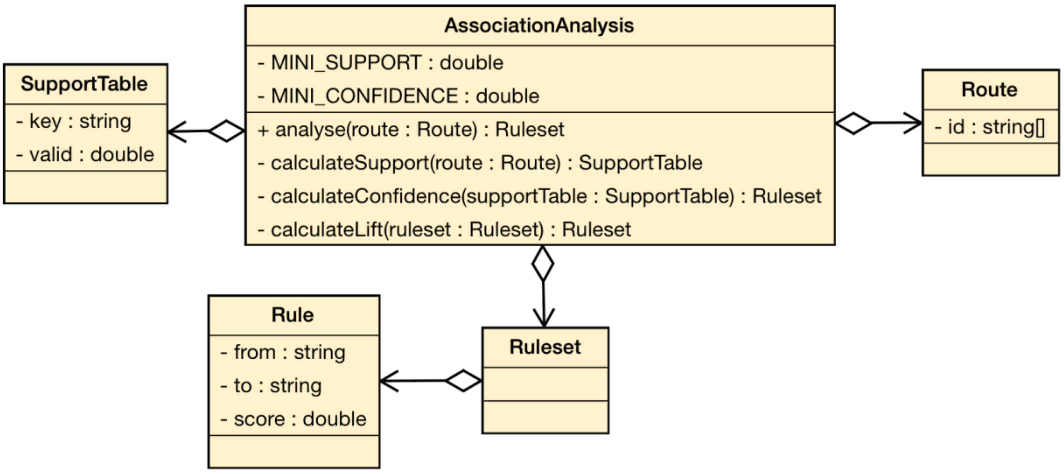

AssociationAnalysis
- MINI_SUPPORT : double
- MINI_CONFIDENCE : double
+ analyse(route : Route) : Ruleset
- calculateSupport(route : Route) : SupportTable
- calculateConfidence(supportTable : SupportTable) : Ruleset
- calculateLift(ruleset : Ruleset) : Ruleset
SupportTable
- key : string
- valid : double
Route
- id : string[]
Rule
- from : string
- to : string
- score : double
Ruleset

Figure 15: Class diagram of association analysis.

5.1.4 Rebuild route

In data analyse module, the first task to make used the record in database, and we need to use these record to rebuild the user's travel route in the past. Therefore our input is the collection of record in database and the output should be the collection of route, we define this collection as ***R***.

Algorithm

```
rebuildRoute(refNumbers)
{
    routes = null
    for(refNo in refNumbers)
    {
            records = the record with same refNo in database
            sortedRecords = sorting the records with date
            collectionOfSplitedRecords = split the sortedRecords with date
            for(oneDayRecords in collectionOfSplitedRecords)
            {
                    if(oneDayRecords.size > MIN_NO_OF_POI)
                    {
                            routes.add(oneDayRecords)
                    }
            }
    }
    return routes
}
```

For example, we get the five photos with a user that taken in the same day. Also know their order according to the time. Then we get that user route on that day.

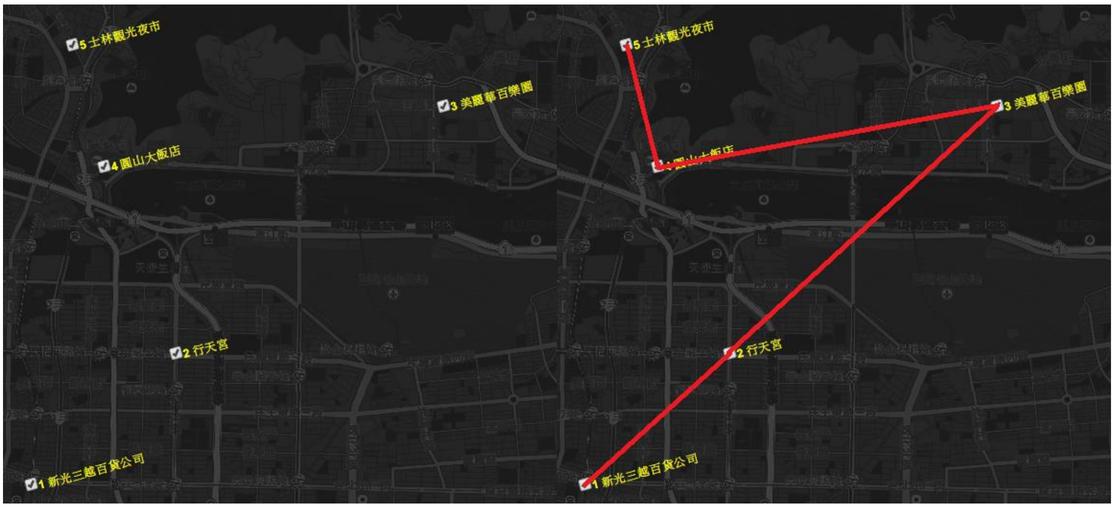


Figure 16: Example of rebuild route.

5.1.5 Support calculation

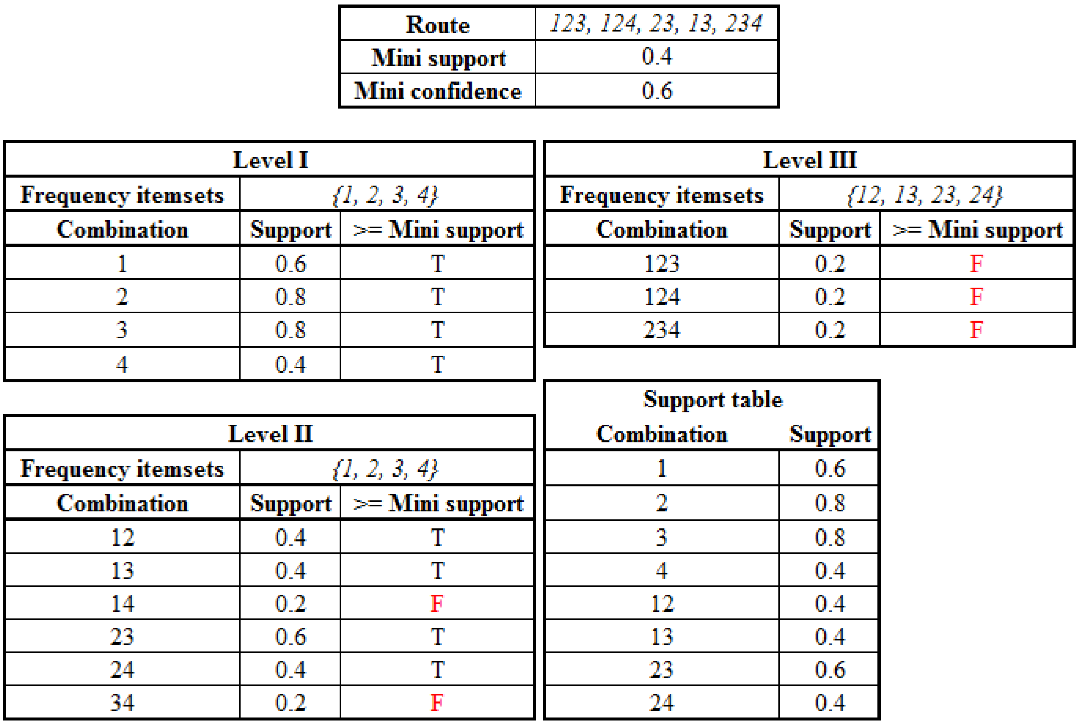
Next, our task is to calculate support. Actually we have two parts in calculate support. The second part will be introduced in next session. Here we will talk about first part. The first part is to calculate the number of every point of interest (Define this as ***POI***) that occur in ***R***, and also doing same calculation on their combination recursively unit do not satisfy our equation. In addition, we apply pruning method here to reduce the calculation time. Therefore our input is ***R***, the output is the first level support table, and we define this table as ***F***.

Algorithm

```
calculateFirstLevelSupport(routes)
{
    passedCombination = null
    itemset = all poi in database
    r = 1
    do
    {
            n = itemset.size
            combination = apply c(n, r) to itemset
            itemset = null
            for(c in combination)
            {
                    support = the number that c occur in routes / routes.size
                    if(support >= MINI_SUPPORT)
                    {
                            passedCombination.add(c)
                            itemset.add(c)
                    }
            }
            r = r + 1
    }while(itemset != null)
    return passedCombination
}
```

For example, we have five routes 123, 124, 23, 13, 234. The minimum support is 0.4 and the minimum confidence is 0.6. Also we have the frequency item sets 1234. First we calculate the combination of frequency item sets with formula C (4, 1) so that we get 1, 2, 3 and 4. Then calculate their support. Support (1) = 3 / 5 = 0.6 >= 0.4. Second, the combination that its support >= 0.4 will be the next frequency item sets. So that we have new frequency item sets 1234. Then, we calculate the combination of frequency items sets with formula C (4, 2) so that we get 12, 13, 14, 23, 23 and 34.

Next calculate their support. Support (12) = 2 / 5 = 0.4 >= 0.4. And keep on doing unit there do not have new frequency item sets.



Route	*123, 124, 23, 13, 234*
Mini support	0.4
Mini confidence	0.6

Level I		
Frequency itemsets	*{1, 2, 3, 4}*	
Combination	**Support**	**>= Mini support**
1	0.6	T
2	0.8	T
3	0.8	T
4	0.4	T

Level II		
Frequency itemsets	*{1, 2, 3, 4}*	
Combination	**Support**	**>= Mini support**
12	0.4	T
13	0.4	T
14	0.2	F
23	0.6	T
24	0.4	T
34	0.2	F

Level III		
Frequency itemsets	*{12, 13, 23, 24}*	
Combination	**Support**	**>= Mini support**
123	0.2	F
124	0.2	F
234	0.2	F

Support table	
Combination	**Support**
1	0.6
2	0.8
3	0.8
4	0.4
12	0.4
13	0.4
23	0.6
24	0.4

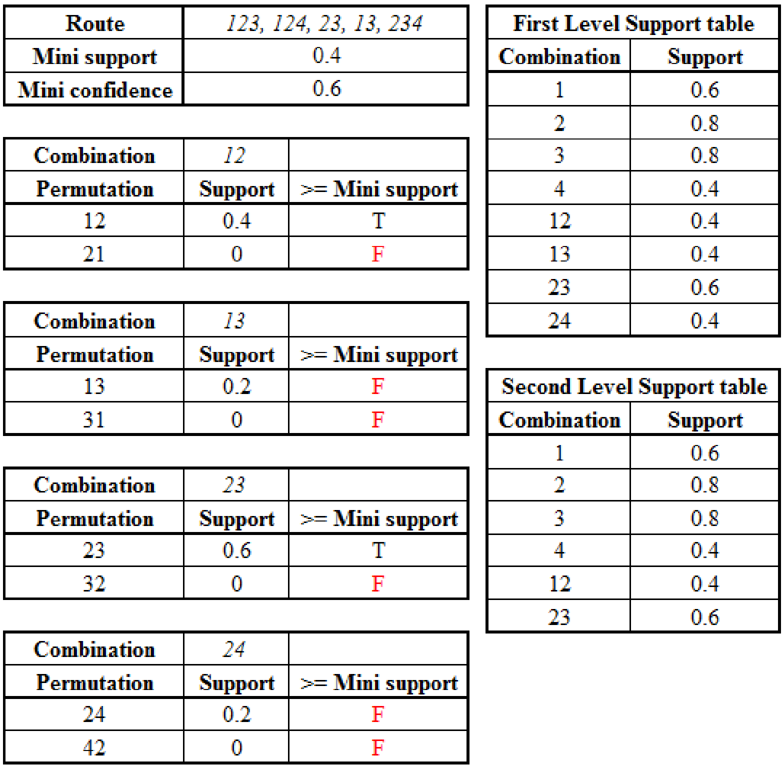
Table 1: Example of first level support calculation .

After first level support calculation, we will do the second part. Since the first part calculation is just count is that POI occurred in R, F does not guarantee the sequence order is correct. The second part is to deal with this problem. Therefore our input is ***F***, the output is the second level support table, we define this table as ***S***.

Algorithm

```
calculateSecondLevelSupport(passedCombination)
{
    supportTable = null
    for(c in passedCombination)
    {
            permutation = p(c)
            for(p in permutation)
            {
                    support = the number that p occur in routes / routes.size
                    if(support >= MINI_SUPPORT)
                    {
                            supportTable.add(p, support)
                    }
            }
    }
    return supportTable
}
```

For example, we have five routes 123, 124, 23, 13, 234. The minimum support is 0.4 and the minimum confidence is 0.6. Also, we have the first level support table. Therefore, we can find out all permutation of the first level support table. Then count the number that the permutation occurs in the route. Support (12) = 2 / 5 = 0.4 >= 0.4. Hence, this is an acceptable permutation.



Route	*123, 124, 23, 13, 234*
Mini support	0.4
Mini confidence	0.6

Combination	*12*	
Permutation	Support	>= Mini support
12	0.4	T
21	0	F

Combination	*13*	
Permutation	Support	>= Mini support
13	0.2	F
31	0	F

Combination	*23*	
Permutation	Support	>= Mini support
23	0.6	T
32	0	F

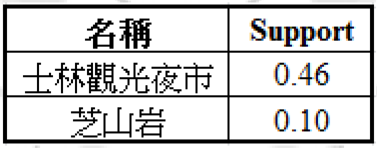
Combination	*24*	
Permutation	Support	>= Mini support
24	0.2	F
42	0	F

First Level Support table	
Combination	Support
1	0.6
2	0.8
3	0.8
4	0.4
12	0.4
13	0.4
23	0.6
24	0.4

Second Level Support table	
Combination	Support
1	0.6
2	0.8
3	0.8
4	0.4
12	0.4
23	0.6

Table 2: Example of second level support calculation.

The importance of poi in system is depends on its support. When support is high, that poi is more popular. As you can see the example, the support of first poi is higher than the second one. That is mean the first poi is more popular than the second one for most of people.



名稱	Support
士林觀光夜市	0.46
芝山岩	0.10

Table 3: Example of the support comparison.

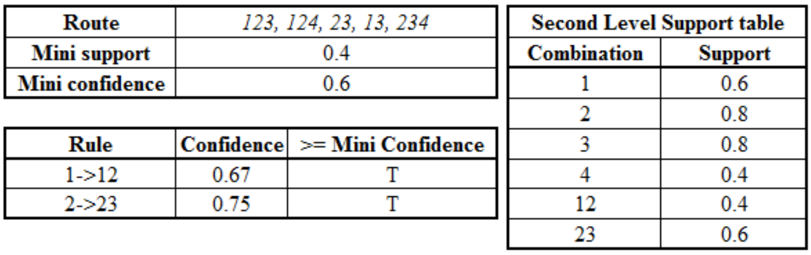
5.1.6 Confidence calculation

In this part, our task is to build up the rule. Here we will use confidence calculation result as rule. In order to suitable for our system, we have this limitations in calculate confidence. Whenever X=>Y, X always contains only one element, and Y should be always greater than one element, then we will only take the element after X in Y's collection. Therefore our input is ***S***, the output is confidence table, we define this table as ***C***, it is also known as rule.

Algorithm

```
calculateConfidence(supportTable)
{
    ruleset = null
    for(x in supportTable)
    {
            for(y in supportTable)
            {
                    if
                    (
                       x.key != y.key &&
                       x.key.size == 1 &&
                       x.key is found in y.key &&
                       x.key is not the last element in y.key
                    )
                    {
                           confidence = y.value / x.value
                           if(confidence >= MINI_CONFIDENCE)
                           {
                                   ruleset.add(x.key, y.key, confidence)
                           }
                    }
            }
    }
    return ruleset
}
```

For example, we have five routes 123, 124, 23, 13, 234. The minimum support is 0.4 and the minimum confidence is 0.6. Also, we have the support table. Therefore if we have a rule x=>y, then we can calculate its confidence using support of y divide support of x. Confidence (1->12) = Support (12) / Support (1) = 0.4 / 0.6 = 0.67 >= 0.6. Therefore, this is an acceptable rule.



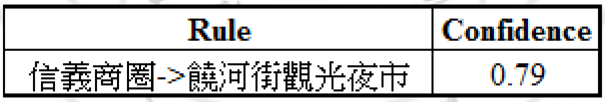
Route	*123, 124, 23, 13, 234*
Mini support	0.4
Mini confidence	0.6

Rule	Confidence	>= Mini Confidence
1->12	0.67	T
2->23	0.75	T

Second Level Support table	
Combination	Support
1	0.6
2	0.8
3	0.8
4	0.4
12	0.4
23	0.6

Table 4: Example of confidence calculation.

For example we get 0.79 from first poi to second poi. This means people will move like this. The confidence score has a positive relation with credibility.



Rule	Confidence
信義商圈->饒河街觀光夜市	0.79

Table 5: Example of the rule.

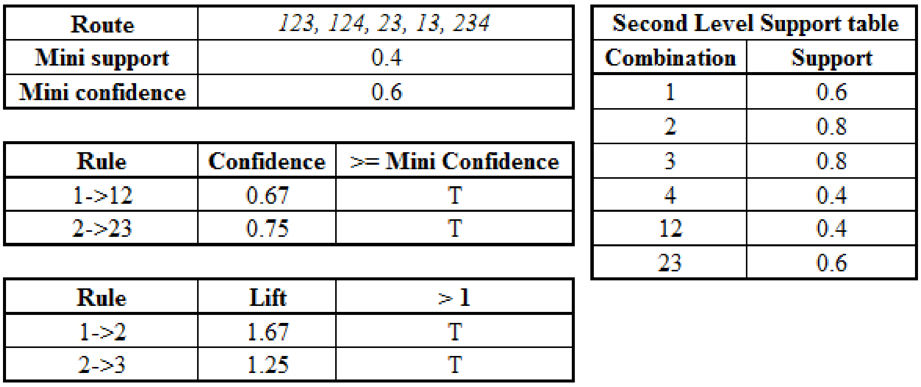
5.1.7 Lift calculation

Actually ***C*** is what we expect for. So what else should we do? The last but not least is to check whether the rules are useful or not. Hence, we do the lift calculation, after this calculation, we will get rid of the rules that are useless. Therefore our input is ***C***, the output is useful rule, we define this calculation result as ***L***, and this is what we use in weight function.

Algorithm

```
calculateLift(ruleset, supportTable)
{
    trustedRuleset = null
    for(r in ruleset)
    {
            support = supportTable(r.firstkey + r.secondkey)
            lift = r.value / support
            if(lift >= 1)
            {
            trustedRuleset.add(r)
            }
    }
    return trustedRuleset
}
```

For example, we have five routes 123, 124, 23, 13, 234. The minimum support is 0.4 and the minimum confidence is 0.6. Also, we have the support table and confidence table. Therefore, we can calculate lift using its confidence divide its support. Lift (1->2) = Confidence (1->12) / Support (12) = 0.67 / 0.4 = 1.67 > 1. Thus this rule should be having a positive effect.



Route	*123, 124, 23, 13, 234*
Mini support	0.4
Mini confidence	0.6

Rule	Confidence	>= Mini Confidence
1->12	0.67	T
2->23	0.75	T

Rule	Lift	> 1
1->2	1.67	T
2->3	1.25	T

Second Level Support table	
Combination	Support
1	0.6
2	0.8
3	0.8
4	0.4
12	0.4
23	0.6

Table 6: Example of lift calculation.

5.2 Route generation module

In this part, we would like to introduce our route generation module. Genetic algorithm is the core of our generation module. Also our route generation module is support three kind of method to generate the route: distance, weight and time.

5.2.1 Introduction of Genetic Algorithm

Genetic algorithm is suitable routinely used to generate useful solutions to optimization and complex search problems. So what does this algorithm doing? “Survival of the fittest” is the core idea of genetic algorithm. This algorithm is using some operation to figure out the fittest gene under specific condition in the population. There are a few important operations in genetic algorithm. They are selection, crossover and mutation.

Selection helps us to do this by discarding the bad designs and only keeping the best individuals in the population. There are a few different selection methods but the basic idea is the same, make it more likely that fitter individuals will be selected for our next generation. Crossover is an operation to generate a new gene using parent’s gene. Mutation is an operation that the gene change itself suddenly. Then, the method to measure a gene is good or not is fitness function.

5.2.2 Why genetic algorithm

When we've found a route that we believe is optimal, how can we test if it's really the optimal route? Well, in short, we can't - at least not practically. To understand why it's so difficult to prove the optimal route let's consider a similar map with just 3 locations instead of the original 20. To find a single route, we first have to choose a starting location from the three possible locations on the map. Next, we'd have a choice of 2 cities for the second location, and then finally there is just 1 city left to pick to complete our route. This would mean there are 3 x 2 x 1 different routes to pick in total. That means, for this example there are only 6 different routes to pick from. So for this case of just 3 locations it's reasonably trivial to calculate each of those 6 routes and find the shortest. If you're good at maths you may have already realized what the problem is here. The number of possible routes is a factorial of the number of locations to visit, and trouble with factorials is that they grow in size remarkably quick!

Although it may not be practical to find the best solution for a problem like ours, we do have algorithms that let us discover close to optimum solutions such as the nearest neighbour algorithm and swarm optimization. Genetic algorithms are capable of finding a 'good-enough' solution to the travelling salesman problem surprisingly quickly. That’s why we choose it.

5.2.3 Route generation module structure

Poi is the base unit in route generation. A tour is a list of poi with random sequence. Population is a list of tour. Evolution is the main function in genetic algorithm module. This function accepts a population then generate a new population. We also call the new population as new generation. Selection, crossover and mutation is the function that will called by evolution. The last but not least, every population will use a fitness to design how fitness it is. Here we have three fitness functions: distance, time and weight.

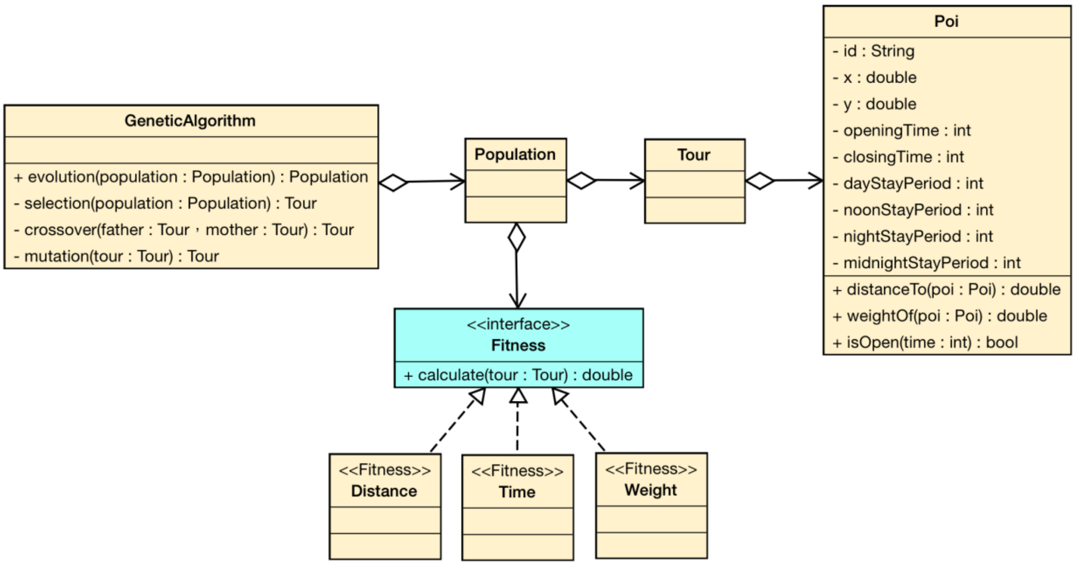

GeneticAlgorithm
+ evolution(population : Population) : Population
- selection(population : Population) : Tour
- crossover(father : Tour , mother : Tour) : Tour
- mutation(tour : Tour) : Tour
Population
Tour
Poi
- id : String
- x : double
- y : double
- openingTime : int
- closingTime : int
- dayStayPeriod : int
- noonStayPeriod : int
- nightStayPeriod : int
- midnightStayPeriod : int
+ distanceTo(poi : Poi) : double
+ weightOf(poi : Poi) : double
+ isOpen(time : int) : bool
<<interface>>
Fitness
+ calculate(tour : Tour) : double
<<Fitness>>
Distance
<<Fitness>>
Time
<<Fitness>>
Weight

Figure 17: Class diagram of route generation module.

5.2.4 Evolution function

In this function, we our input is population, then we will choose two tours from population as parent, next do the crossover of them, after that apply mutation function on each of tour. Finally output the new generation.

Algorithm

```
evolution(population)
{
    fittest = population.getFittest
    tours = population.getTours
    newGeneration = null
    for(tour in tours)
    {
            father = selection(population)
            mother = selection(population)
            child = crossover(father, mother)
            child = mutation(child)
            newGeneration.add(child)
    }
    return newGeneration
}
```

5.2.5 Selection function

In our selection function, we want to be constantly improving our populations overall fitness. Thus, we create a small population, then randomly put a few tour to our small population from the population that we input, and then output the fitness tour.

Algorithm

```
selection(population)
{
    selectedTours = randomly pick NO tours from population
    newPopulation.add(selectedTours)
    return newPopulation.getFittest
}
```

5.2.6 Crossover function

Our crossover method that's able to produce a valid route is ordered crossover. In this crossover method we select a subset from the first parent, and then add that subset to the offspring. Any missing values are then adding to the offspring from the second parent in order that they are found. Thus, our input of this function is two tours and output is a new tour.

Algorithm

```
crossover(father, mother)
{
    child = null
    r = random(0, 1)
    if(r <= CROSSOVER_RATE)
    {
            size = father.size
            startPos = random(0, size)
            endPos = random(0, size)
            if(startPos > endPos)
            {
                    swap(startPos, endPos)
            }

            for(i = 1; i < size; i = i + 1)
            {
                    if(i >= startPos && i < endPos)
                    {
                            child[i] = father[i]
                    }
            }

            excludedGene = the element that mother have but not in child
            for(i = 1, j = 1; i < size; i = i + 1)
            {
                    if(i < startPos or i >= endPos)
                    {
                            child[i] = mother[j]
                            j = j + 1
                    }
            }
    }
    return child
}
```

For example, we have a father gene with sequence one to nice and a mother gene with sequence nice to one. Then we have a random start position three and end position seven. Therefore, their child will copy the father's gene from start position to the position before the end position. The gene of father is 3456. After that we find the gene that mother have but not in child. Those values are 98721 At last, copy those values to child. Therefore, the child gene is 983456721.

Start position: 3, End position: 7									
Father	1	2	3	4	5	6	7	8	9
Mother	9	8	7	6	5	4	3	2	1
Child	9	8	3	4	5	6	7	2	1

Table 7: Example of crossover function.

5.2.7 Mutation function

The mutation method should only be capable of shuffling the route, it shouldn't ever add or remove a location from the route, and otherwise it would risk creating an invalid solution. One type of mutation method we could use is swap mutation. Therefore our input is one tour and the output is also one tour.

Algorithm

```
mutation(child)
{
    newGeneChild = child
    for(poi in newGeneChild)
    {
            m = random(0, 1)
            if(m <= MUTATION_RATE)
            {
                    swapPos = random(0, tour.size)
                    swap(newGeneChild[currentPos], newGeneChild[swapPos])
            }
    }
    return newGeneChild
}
```

For example, we have a gene with the sequence one to nice. Then we have a random start position three and end position seven. Therefore, we just simple swap two value of those positions so that the position three will become value seven, on the other hand, the position seven will become value three.

Start position: 3, End position: 7									
Before	1	2	3	4	5	6	7	8	9
After	1	2	7	4	5	6	3	8	9

Table 8: Example of mutation function.

5.2.8 Fitness functions

The least part of this module is the fitness functions. Our fitness functions are distance, weight and time. Distance fitness function is using distance between two points of interest as measurement, the shorter distance get the higher score. Weight fitness function is using the weight between two points of interest as measurement, and the weight between them are calculated by association rule, the confidence between two points of interest will have a positive relation with the score. Time fitness function is using the time information of point of interest as measurement. The time information of point of interest for measurement is opening time, closing time. The time that the user wants to visit is within opening time that will get the higher score. In addition to this, we also have their combinational fitness functions: distance with time, distance with weight and also distance with time and weight.

Algorithm of Distance fitness function

```
distance(tour)
{
    totalDistance = 0
    previousPoi = tour.lastElement
    for(i = 0; i < tour.size; i = i + 1)
    {
            currentPoi = tour[i]
            distance = previousPoi.distanceTo(currentPoi)
            totalDistance = totalDistance + distance
            previousPoi = currentPoi
    }
    return -totalDistance
}
```

Algorithm of Weight fitness function

```
weight(tour)
{
    totalWeight = 0
    previousPoi = tour.lastElement
    for(i = 0; i < tour.size; i = i + 1)
    {
            currentPoi = tour[i]
            weight = previousPoi.weight(currentPoi)
            totalWeight = totalWeight + weight
            previousPoi = currentPoi
    }
    return totalWeight
}
```

Algorithm of Time fitness function

```
time(tour)
{
    score = 0
    currentTime = START_TIME
    for(i = 0; i < tour.size; i = i + 1)
    {
            currentPoi = tour[i]
            if(currentPoi.isOpen(currentTime))
            {
                   currentTime = currentTime + currentPoi.getPeriod(currentTime)
                    score = score + currentPoi.getScore(currentTime)
            }
    }
    return score
}
```

Algorithm of Distance with time fitness function

```
distanceWithTime(tour)
{
    totalDistance = 0
    previousPoi = tour.lastElement
    currentTime = START_TIME
    for(i = 0; i < tour.size; i = i + 1)
    {
```

```
               currentPoi = tour[i]
               distance = previousPoi.distanceTo(currentPoi)
               if(currentPoi.isOpen(currentTime))
               {
                      currentTime = currentTime + currentPoi.getPeriod(currentTime)
                       distance = distance * 0.5
               }
               totalDistance = totalDistance + distance
               previousPoi = currentPoi
        }
        return -totalDistance
    }
```

Algorithm of Distance with weight fitness function

```
    distanceWithWeight(tour)
    {
        totalDistance = 0
        previousPoi = tour.lastElement
        for(i = 0; i < tour.size; i = i + 1)
        {
               currentPoi = tour[i]
               distance = previousPoi.distanceTo(currentPoi)
               weight = previousPoi.weight(currentPoi)
               totalDistance = totalDistance + distance * (1 - weight)
               previousPoi = currentPoi
        }
        return -totalDistance
    }
```

Algorithm of Distance with time and weight fitness function

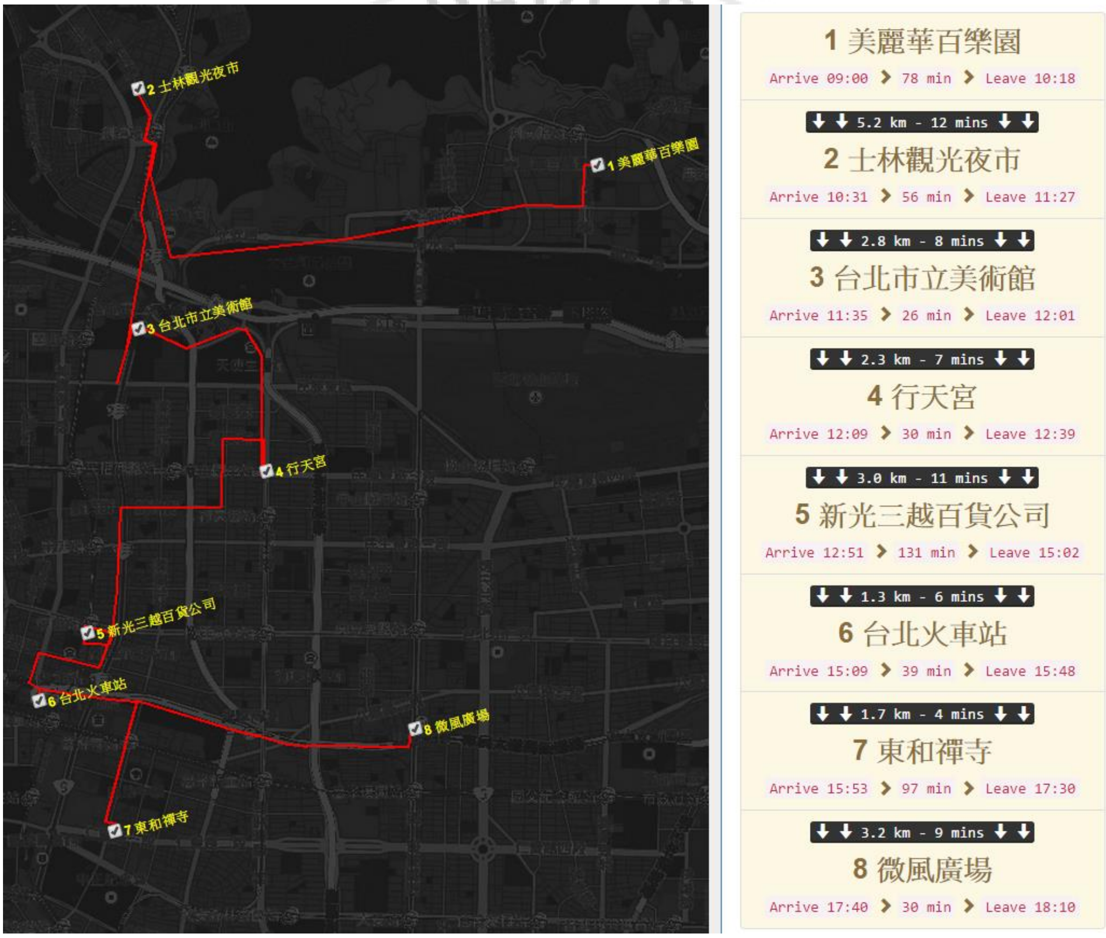
```
    distanceWithWeightAndTime(tour)
    {
        totalDistance = 0
        previousPoi = tour.lastElement
        currentTime = START_TIME
        for(i = 0; i < tour.size; i = i + 1)
        {
               currentPoi = tour[i]
               distance = previousPoi.distanceTo(currentPoi)
               weight = previousPoi.weight(currentPoi)
               if(currentPoi.isOpen(currentTime))
               {
                      currentTime = currentTime + currentPoi.getPeriod(currentTime)
                       distance = distance * 0.5
               }
               totalDistance = totalDistance + distance * (1 - weight)
               previousPoi = currentPoi
        }
        return -totalDistance
    }
```

CHAPTER 6. System evaluation

6.1 Different method evaluation

6.1.1 Distance method evaluation

For the distance method is compare the total distance of the route and find out the shortest route. But there are many unacceptable orders such as it arranged to visit night market at morning and arranged to visit temple at night. Actually, night market has to arrange at night or nightfall time and temple should be at morning or afternoon because different location have different opening time or recommend sightseeing time. If the system suggest in incorrect time for visitor then they will feel confused. So that only consider the distance between the locations is very not enough for the router suggestion. And it should be use some more method to optimize the travel route.



1 美麗華百樂園
Arrive 09:00 › 78 min › Leave 10:18

↓ ↓ 5.2 km - 12 mins ↓ ↓

2 士林觀光夜市
Arrive 10:31 › 56 min › Leave 11:27

↓ ↓ 2.8 km - 8 mins ↓ ↓

3 台北市立美術館
Arrive 11:35 › 26 min › Leave 12:01

↓ ↓ 2.3 km - 7 mins ↓ ↓

4 行天宮
Arrive 12:09 › 30 min › Leave 12:39

↓ ↓ 3.0 km - 11 mins ↓ ↓

5 新光三越百貨公司
Arrive 12:51 › 131 min › Leave 15:02

↓ ↓ 1.3 km - 6 mins ↓ ↓

6 台北火車站
Arrive 15:09 › 39 min › Leave 15:48

↓ ↓ 1.7 km - 4 mins ↓ ↓

7 東和禪寺
Arrive 15:53 › 97 min › Leave 17:30

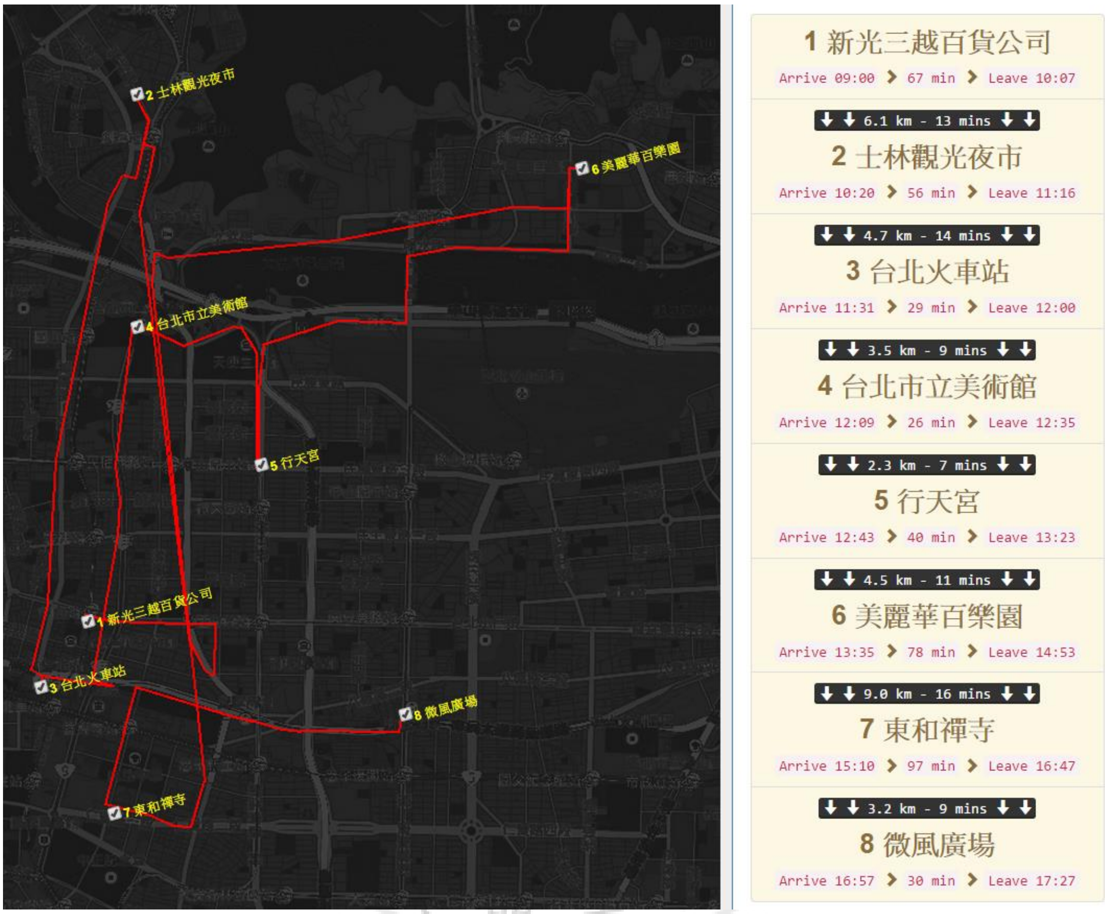
↓ ↓ 3.2 km - 9 mins ↓ ↓

8 微風廣場
Arrive 17:40 › 30 min › Leave 18:10

Figure 18: Distance method evaluation.

6.1.2 Weight method evaluation

The weight method is to calculate each location relational rank and compute a highest related route which mean all location are related and many other traveller using this route before. But we can see the result of this method which is easy to find out the defect in the route. The route is very disorder and difficult to follow this route because the distance between each location is very high. For example, after the first location *新光三越百貨公司* then the system suggested a farthest location to user which is the second location *士林觀光夜市*, then go to third location *台北火車站*. But we can find that the first location and third location are most near. Actually, after the first location we should go to third location *台北火車站* before second location *士林觀光夜市*. Also the second location should not suggest to visiting at morning because for the opening time. Beside of this problem, the distance of the route is not good because some we can see some location that user have to go retreat.

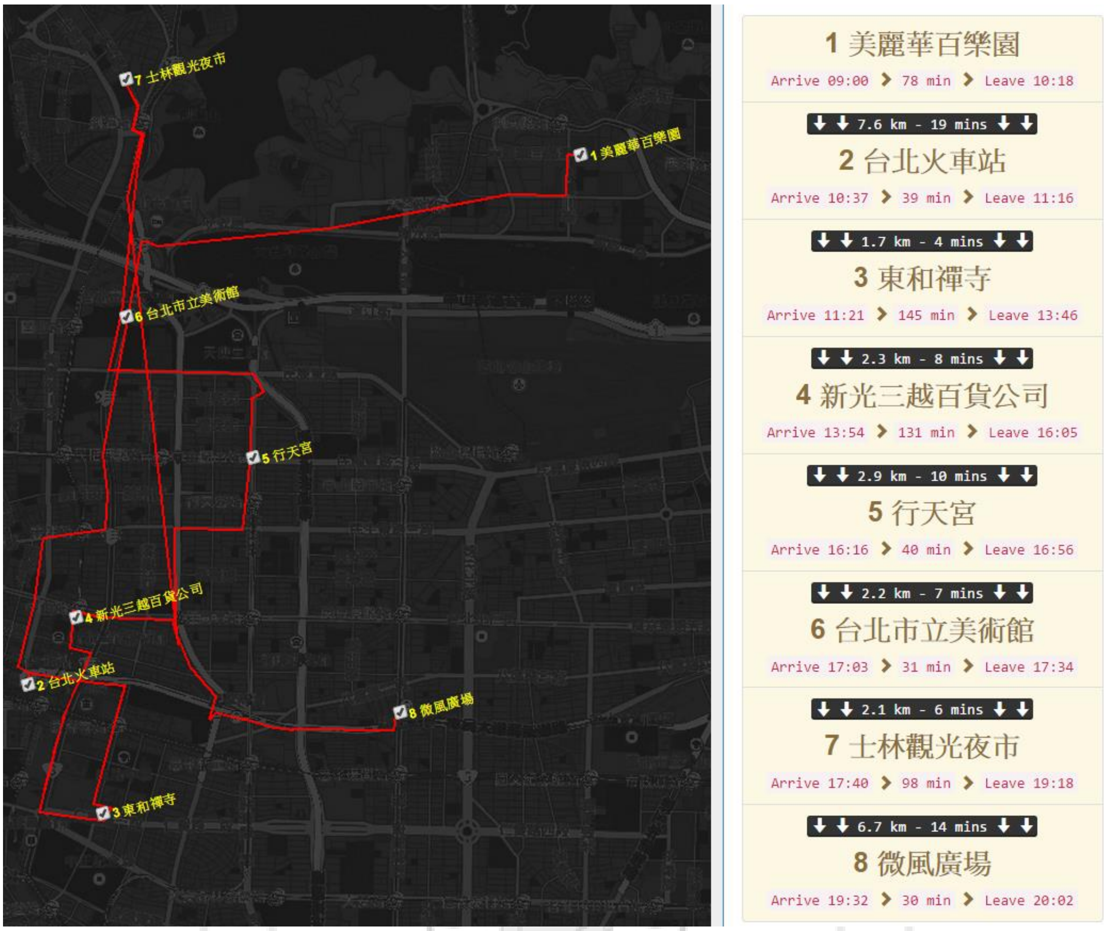


1 新光三越百貨公司
Arrive 09:00 › 67 min › Leave 10:07
↓ ↓ 6.1 km - 13 mins ↓ ↓
2 士林觀光夜市
Arrive 10:20 › 56 min › Leave 11:16
↓ ↓ 4.7 km - 14 mins ↓ ↓
3 台北火車站
Arrive 11:31 › 29 min › Leave 12:00
↓ ↓ 3.5 km - 9 mins ↓ ↓
4 台北市立美術館
Arrive 12:09 › 26 min › Leave 12:35
↓ ↓ 2.3 km - 7 mins ↓ ↓
5 行天宮
Arrive 12:43 › 40 min › Leave 13:23
↓ ↓ 4.5 km - 11 mins ↓ ↓
6 美麗華百樂園
Arrive 13:35 › 78 min › Leave 14:53
↓ ↓ 9.0 km - 16 mins ↓ ↓
7 東和禪寺
Arrive 15:10 › 97 min › Leave 16:47
↓ ↓ 3.2 km - 9 mins ↓ ↓
8 微風廣場
Arrive 16:57 › 30 min › Leave 17:27

Figure 19: Weight method evaluation.

6.1.3 Time method evaluation

The time method is to calculate the acceptable visiting time to user by other traveller's experience. User follow this order can go all location on the opening and popular visiting time. But there are some problem is about the total distance and relation between each location. Some locations also have to go retreat to the following location that is not a good route for user. User may have to spend more traffic time between locations. And it should be improved.



1 美麗華百樂園
Arrive 09:00 › 78 min › Leave 10:18

↓ ↓ 7.6 km - 19 mins ↓ ↓

2 台北火車站
Arrive 10:37 › 39 min › Leave 11:16

↓ ↓ 1.7 km - 4 mins ↓ ↓

3 東和禪寺
Arrive 11:21 › 145 min › Leave 13:46

↓ ↓ 2.3 km - 8 mins ↓ ↓

4 新光三越百貨公司
Arrive 13:54 › 131 min › Leave 16:05

↓ ↓ 2.9 km - 10 mins ↓ ↓

5 行天宮
Arrive 16:16 › 40 min › Leave 16:56

↓ ↓ 2.2 km - 7 mins ↓ ↓

6 台北市立美術館
Arrive 17:03 › 31 min › Leave 17:34

↓ ↓ 2.1 km - 6 mins ↓ ↓

7 士林觀光夜市
Arrive 17:40 › 98 min › Leave 19:18

↓ ↓ 6.7 km - 14 mins ↓ ↓

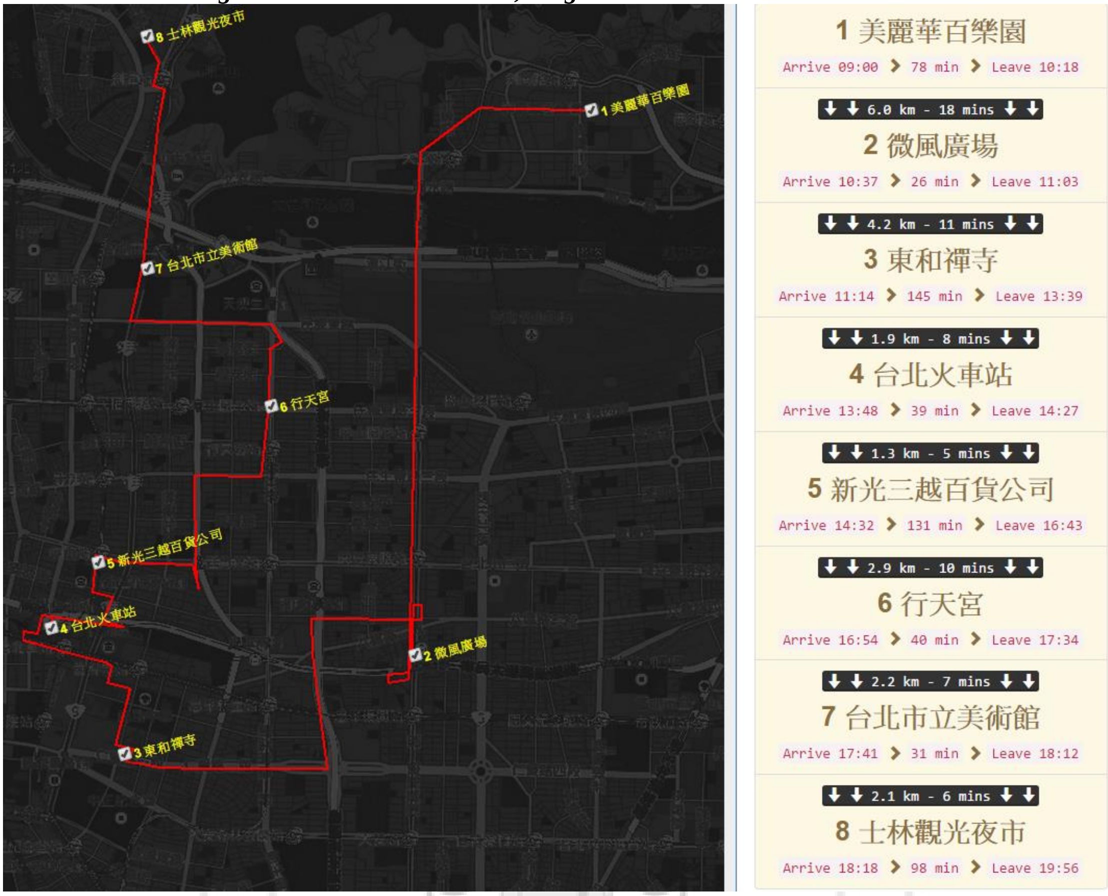
8 微風廣場
Arrive 19:32 › 30 min › Leave 20:02

Figure 20: Time method evaluation.

6.1.4 Combine method evaluation

This method is combined three methods together which method are distance, weight and time method. We try to test this method using same POIs as the period evaluation. So the result can compare with other result. In this testing result, we find that the route is the best. All location can visit on popular visit time and also can stay more time in every location. For the total distance, it is a direct route and there are no situation that user have to go retreat to visit the following location. User can save more time in traffic and spend more time in the viewpoint. The route is shorter that using weight and time method. The time control is very good. Combine all method and which can balance the method advantage and reduce the weak point in the route so that the result is acceptable and useful.

Figure 21: Combine distance, weight and time method evaluation.



1 美麗華百樂園

Arrive 09:00 ❯ 78 min ❯ Leave 10:18

↓ ↓ 6.0 km - 18 mins ↓ ↓

2 微風廣場

Arrive 10:37 ❯ 26 min ❯ Leave 11:03

↓ ↓ 4.2 km - 11 mins ↓ ↓

3 東和禪寺

Arrive 11:14 ❯ 145 min ❯ Leave 13:39

↓ ↓ 1.9 km - 8 mins ↓ ↓

4 台北火車站

Arrive 13:48 ❯ 39 min ❯ Leave 14:27

↓ ↓ 1.3 km - 5 mins ↓ ↓

5 新光三越百貨公司

Arrive 14:32 ❯ 131 min ❯ Leave 16:43

↓ ↓ 2.9 km - 10 mins ↓ ↓

6 行天宮

Arrive 16:54 ❯ 40 min ❯ Leave 17:34

↓ ↓ 2.2 km - 7 mins ↓ ↓

7 台北市立美術館

Arrive 17:41 ❯ 31 min ❯ Leave 18:12

↓ ↓ 2.1 km - 6 mins ↓ ↓

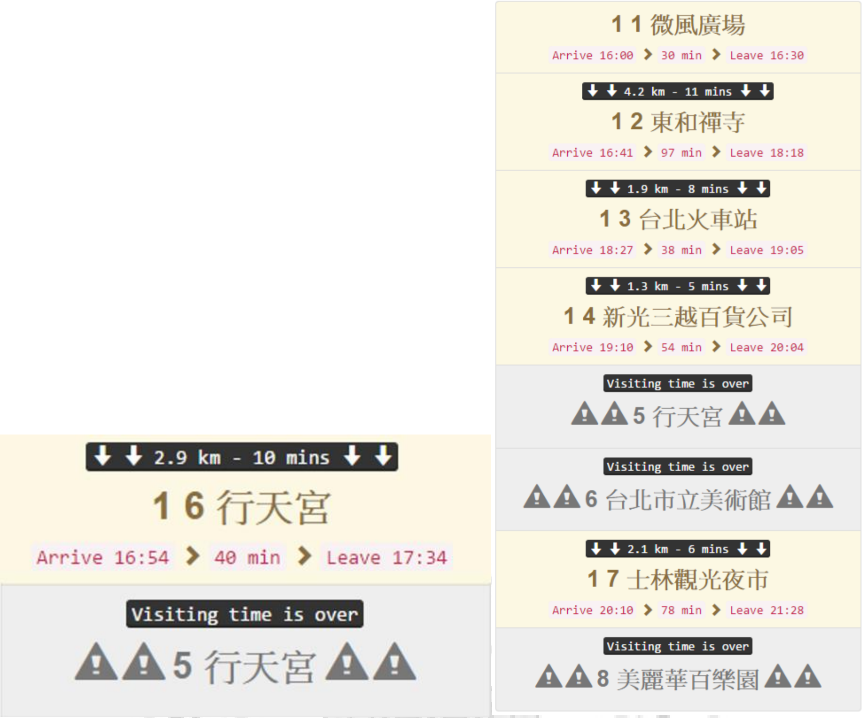
8 士林觀光夜市

Arrive 18:18 ❯ 98 min ❯ Leave 19:56

6.2 Supplement method evaluation

6.2.1 Visiting time evaluation

Visiting time is to calculate the location popular visiting time, if the route cannot arrange a suitable time to this location using the time method, the distance and weight method will arrange a shortest way and related to the suggest route. For example, the user select the starting visiting time is very late (starting time: 16:00) but want to visit 8 location. Then the system will generate a route with 8 points but some location may be over we will try to find out the most location you can visit. When some location the visiting time is over then we will show that for you.



1 1 微風廣場

Arrive 16:00 › 30 min › Leave 16:30

↓↓ 4.2 km - 11 mins ↓↓

1 2 東和禪寺

Arrive 16:41 › 97 min › Leave 18:18

↓↓ 1.9 km - 8 mins ↓↓

1 3 台北火車站

Arrive 18:27 › 38 min › Leave 19:05

↓↓ 1.3 km - 5 mins ↓↓

1 4 新光三越百貨公司

Arrive 19:10 › 54 min › Leave 20:04

Visiting time is over

⚠⚠ 5 行天宮 ⚠⚠

Visiting time is over

⚠⚠ 6 台北市立美術館 ⚠⚠

↓↓ 2.1 km - 6 mins ↓↓

1 7 士林觀光夜市

Arrive 20:10 › 78 min › Leave 21:28

Visiting time is over

⚠⚠ 8 美麗華百樂園 ⚠⚠

↓↓ 2.9 km - 10 mins ↓↓

1 6 行天宮

Arrive 16:54 › 40 min › Leave 17:34

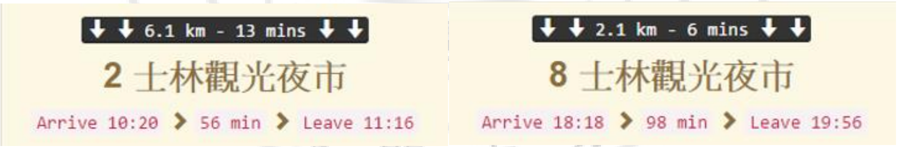
Visiting time is over

⚠⚠ 5 行天宮 ⚠⚠

Figure 6: Example of visiting time evaluation.

6.2.1 Staying time evaluation

The staying time calculation method is separate 4 parts per day. Each part of time will analyse the period traveller staying time. So that in different arrive time will show you different staying time. When the route suggest you go to *士林觀光夜市* at morning then the suggest staying time is ***56min***. If suggest you at night then the staying time of this location will be ***98min.***



↓↓ 6.1 km - 13 mins ↓↓

2 士林觀光夜市

Arrive 10:20 › 56 min › Leave 11:16

↓↓ 2.1 km - 6 mins ↓↓

8 士林觀光夜市

Arrive 18:18 › 98 min › Leave 19:56

Figure 23: Example of staying time evaluation.

CHAPTER 7. Ethics and professional conduct

7.1 Privacy

We use our fetch data module in our route recommendation system fetched a great number of photos from Flicker. The photos we fetched are public to outside. Anyone can access them. Therefore, we use these photos as our source data. We untie the information in photo. We use these information for analyse.

However, the action we did is invasion of privacy?

The user posted his photo in public on Flicker. Therefore, we have right to access the photos and we an able to download it for analyse use. However, we must follow the rule 1.2. Make sure that our system only uses their photo as analyse but not use their photo as some illegal purpose so that harm the photos owner.

1.2 Avoid harm to others

7.2 Unauthorized Access

In this project we are using Flickr and OpenStreetMap as the main data source. The both of them are legal for all users to access their public data for free. If one day, we using this project to found a profit company. One day, Flickr announce that all data are close the access for public and only able to using for studies and non-profit use. It means that if our company continues using Flickr data and we have to pay. But you discover that your id is using student level and the access permission is allow have you to access their data in free.

According rule 1.3, 2.3 and 2.6, we have to be honest if we collect the data using student identity but on my profit project which is broken the rule with Flickr. It is because their data is only for studies and non-profit use. So that I cannot use illegal way to collect the data, we have to pay for that.

1.3 Be honest and trustworthy

2.3 Know and respect existing laws pertaining to professional work

2.6 Honour contracts, agreements, and assigned responsibilities

CHAPTER 8. Conclusion

Throughout the software development phases of this project, we understand that in order to complete the task well and on time, commutation with each other is essential. A good communication could bring along a good atmosphere, and increase the efficiency. We divided the task into two parts based on the functional difference. Although most of the job is separated and can be done independently, but we always come together to discuss the solution and idea of the problems we meet, and the way of our developing method. We know that the opinions from the group mates are important, so we choose to hold meetings constantly to have the effect like brain-storming.

In the programming aspect, it is our first experience to fetch such great number of data from the internet and deal with it. We met several problems that the program does not come out with the result we expect. Since our source data social network site has limitation of the photo. We only can fetch 5,000 photos per hour and also the photos will have opportunity that gets the photo that we already have. We have to care about the execution flow of program statements very carefully. In the system structure, we are using Model-View-Controller framework to construct our web-based information system, now we have a general idea of MVC, but we still can improve the structure of our web pages. Frameworks components can be separated into smaller pieces then take the advantage on separating system functions and easier maintenance.

From cooperation we get lots of useful information and idea, more communications and discussions opened our eyes, enriching our knowledge and helped to make the system to be born in a better way. As the final and biggest project in the university life, we think this project is not only testing about out design and development power, but also tests us the cooperation ability, communication skill and team spirit. We believe that teamwork is very important in software development, as most of the IT people are working in the form of team and group. For the sake of future career, it is a good opportunity for us to be working in a team.

We are very thankful to our supervisor who has given his greatest support to our team. Without his experience in system design and user experience, we could be able to finish our system successfully. Almost for the whole year he is showing his patience, professional skills, experience on software, and his heart in education. We are very content to have him as our supervisors.

This project gives a chance for us to unify the things we have learnt throughout this major in the university life. At the first two years of university life, we have learnt many theories and techniques, but actually there are not much big projects for us to fuse all the knowledge learnt from each course. Now this final year project has given us a good chance to realise what is ‘software’, from zero to all, from ideas to real product, to accomplish this project, we realized what we have learnt, in order to put them all together to get the work done.

To conclude, this project is a good experience to our team. It gives our team a chance to communicate with the others, to try to understand the user requirements well so that less conflict will come out later. We had used some programming languages that we have not used in school projects before in the development stage. It had great improved our programming skill and knowledge on different type of programming language. Though there are many improvements can be done from the system in this project. We hope that later version of this system prototype could appear in the society and contributes as much as possible in the environmental protection.

CHAPTER 9. REFERENCES

[1] K. F. Man, K. S. Tang and S. Wong "Genetic Algorithms: Concepts and Applications".1996.[Online].

Available:http://www.researchgate.net/profile/Sam_Kwong/publication/3217198_Genetic_algorithms_concepts_and_applications_in_engineeringdesign/links/00b49517b05c06d776000000.pdf

[2] Korea, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints".2001.[Online].

Available:http://www.researchgate.net/profile/Yoonho_Seo/publication/222540460_An_efficient_genetic_algorithm_for_the_traveling_salesman_problem_with_precedence_constraints/links/0912f505affe7aad59000000.pdf

[3] Jinghui Zhong, Xiaomin Hu, Min Gu2 and Jun Zhang "Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms". 2009. [Online].

Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.3747&rep=rep1&type=pdf

[4] Tommi Rintala, "Population Size in Gas for TSP".1996.[Online].

Available: http://lipas.uwasa.fi/cs/publications/2NWGA/node8.html

[5] "Association Analysis: Basic Concepts and Algorithm".[Online].

Available: http://www-users.cs.umn.edu/~kumar/dmbook/ch6.pdf

[6] Kenneth Lai, "Support vs Confidence in Association Rule Algorithms".[Online].

Available: http://www.academia.edu/648890/Support_vs_Confidence_in_Association_Rule_Algorithms

[7] P. D. McNicholas, T. B. Murphy and M. O'Regan, "Standarding the Lift of an Association Rule". [Online]. Available: https://www.scss.tcd.ie/disciplines/statistics/tech-reports/07-01.pdf