

**ACCOMMODATION SEARCH ENGINE BASED ON  
FREQUENT-VISIT-LOCATION USING MULTI-ATTRIBUTE UTILITY THEORY**

By

CHONG KAR WENG (10ACB07304)

A PROPOSAL

SUBMITTED TO

University Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

**BACHELOR OF INFORMATION SYSTEM (HONS)**

**BUSINESS INFORMATION SYSTEMS**

Faculty of Information and Communication Technology

(Perak Campus)

JANUARY 2013

## DECLARATION OF ORIGINALITY

I declare that this report entitled “Accommodation Search Engine based on Frequent-Visit-Location using Multi-Attribute Utility Theory” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_

Name : \_\_\_\_\_

Date : \_\_\_\_\_

## **ACKNOWLEDGEMENT**

The completion of this project would not be made possible without the abundance of guidance, help and encouragement from my supervisor, Dr. Alex Ooi Boon Yaik. I would like to take this opportunity to express my greatest gratitude towards Dr. Alex for the patient for giving suggestions and improvements on this project.

I would like to thanks to Faculty of Information Technology, for providing the facilities such as computer laboratory and Internet connection in UTAR to facilitate the work.

Besides, I would like to express my deepest appreciation to my family members, for the financial and mentally support and encouragement that support me all the time. Last but not least, I offer the gratitude to my fellow friends for willingness and to share knowledge and give suggestion and support during the project.

## **ABSTRACT**

The proposed project is an accommodation search engine based on frequent-visit-location using multi-attribute utility theory. The purpose of developing the search system is to provide more relevant results to the user, reduces the amount of time spend on searching accommodation and also support user in their accommodation selection decision making by displaying the results in the tabular format.

This progress report details the introduction to the project which includes identifying the disadvantages and problems of using area-based search currently used by most of the accommodation search websites and explains how using frequent visit location search with rating features are used to address these issues.

The scope of the project and the technologies involved are also discussed as well as the literature review on selected accommodation search websites to identify their strengths as well as weaknesses and how improvements can be made by introducing frequent visit location search with rating feature. Proposed methodologies used are also presented, which includes descriptions of database implementation and the Google Map Services.

## TABLE OF CONTENT

TITLE.....	i
DECLARATION OF ORIGINALITY .....	ii
ACKNOWLEDGEMENT .....	iii
ABSTRACT.....	iv
TABLE OF CONTENT .....	v
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
LIST OF ABBREVIATIONS.....	xiv
CHAPTER 1: INTRODUCTION .....	1
1.1    A scenario that motivates this work.....	3
1.2    Problem Statement.....	4
1.3    Objectives.....	6
1.4    Scope .....	8
CHAPTER 2: LITERATURE REVIEW .....	10
2.1.    Scenario.....	11
2.2.    Websites reviewed according to their strength and weaknesses. ....	13
2.2.1.    iProperty (Local website).....	13
2.2.2.    Trulia (Oversea website).....	14
2.2.3.    Hotpads (Overseas website).....	14
2.2.4.    Thinkproperty (Local website), Realtor (Oversea website), House Locator (Oversea website).....	15
2.2.5.    Room4rent and iBilik (Local website).....	16
2.3.    Review Criteria for the 8 websites in listed in Table 1 .....	17
2.3.1.    Search by specific location.....	17

2.3.2.	Multiple search location at once.....	18
2.3.3.	Rate the search location (Multi-Attribute Utility Theory).....	18
2.3.4.	Search result.....	20
2.3.5.	Inclusion of a Map in the website .....	21
2.4.	Discussion .....	22
CHAPTER 3: METHODOLOGY .....		24
3.1.	The Stages of the Iterative Process (Figure 3-1-F1).....	25
3.2.	Technology Involved .....	26
3.3.	Analysis Diagrams.....	28
3.3.1.	Analysis of the Use Case Diagram.....	28
3.3.2.	Analysis of the Activity Diagram.....	30
3.4.	Behavioural Diagram.....	31
3.4.1.	Sequence Diagram .....	31
3.4.2.	Collaboration Diagram.....	33
3.4.3.	State Diagram.....	35
3.5.	Structural Diagram .....	36
3.5.1.	Entity Relationship Diagram.....	36
3.5.2.	Class diagram .....	38
3.6.	System Progress Conceptual Website Step.....	39
3.6.1.	Search Bar .....	40
3.6.2.	Verification.....	41
3.6.3.	Result display page .....	42
3.7.	Gantt Chart .....	44
CHAPTER 4: IMPLEMENTATION.....		46
4.1.	Overview of system structure .....	46
4.2.	Web Server .....	47
4.3.	Client .....	50

4.4.	Web Browser .....	53
4.5.	Database .....	54
4.	4.5.1 Accommodation Table .....	55
	4.5.2 Member Table.....	56
4.6.	Google Server .....	56
4.7.	Multi-attribute Utility Theory (MAUT) .....	60
Chapter 5: Testing .....		63
5.1.	Scenario 1 - Rated versus Non-rated Result on two frequent-visit-location.....	63
5.2.	Scenario 2 - Rated versus Non-rated Result on three frequent-visit-location.....	68
5.3.	Scenario 3 - Rated versus Non-rated Result on three frequent-visit-location.....	73
5.4	Scenario 4 - Multiple Point Traveling method .....	77
CHAPTER 6: CONCLUSION .....		83
REFERENCES.....		85
APPENDIX A: SCREENSHOT OF RELATED WEBSITES.....		A-1

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 2-1-T1	Criteria for comparison among 4 local and 4 overseas accommodation websites	12



## LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 2-3-F1	Search result different between area and specific search	17
Figure 2-3-F2	MAUT formula applied in this system	19
Figure 2-3-F3	List format and table format listing information	20
Figure 3-1-F1	Spiral Model	23
Figure 3-2-F1	Client and server relationship	25
Figure 3-3-F1	Analysis Use Case Diagram	27
Figure 3-4-F1	Analysis Activity Diagram	29
Figure 3-5-F1	Sequence Diagram	30
Figure 3-6-F1	Collaboration Diagram	32
Figure 3-7-F1	State Diagram	34

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 3-8-F1	Entity Relationship Diagram	35
Figure 3-9-F1	Class Diagram	37
Figure 3-10-F1	Conceptual Search Bar	39
Figure 3-11-F1	Conceptual Verification Page	40
Figure 3-12-F1	Conceptual Result Display Page	41
Figure 3-13-F1	Gantt Chart	43
Figure 4-1-F1	Overview of the system structure.	45
Figure 4-2-F1	The Main page for the project	47
Figure 4-2-F2	Sample JavaScript and AJAX code of requesting validation from Web Server on validate.php page inside the Web Server (Back-End)	48

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 4-2-F3	Sample Validate.PHP which search MySQL database to validate user.	49
Figure 4-3-F1	The search area for user input (Front-End)	50
Figure 4-3-F2	The code for find function (Back-End)	50
Figure 4-3-F3	Display address returned by Google Servers	51
Figure 4-3-F4	Action performed when confirm button pressed.	52
Figure 4-4-F1	Sample Code for Event Handler JQuery	53
Figure 4-6-F1	Search address send by JavaScript to Google Services.	57
Figure 4-6-F2	Google Map with location plotted on map.	58
Figure 4-6-F3	Sample JavaScript Code to calling the Google Service and obtain the return result.	59
Figure 4-7-F1	Formula used in calculation of MAUT in this system	60

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 4-7-F2	The back-end calculation for MAUT.	61
Figure 5-1-F1	Google map with 2 frequent-visit-locations.	63
Figure 5-1-F2	Result of search plotted on Google Map.	63
Figure 5-1-F3	Partial Result of the search for rated and non-rated cases.	64
Figure 5-2-F1	Google map with 3 frequent-visit-locations	68
Figure 5-2-F2	Result of search plotted on Google Map.	68
Figure 5-2-F3	Partial Result of the search for rated and non-rated cases.	69
Figure 5-3-F1	Google map with 3 frequent-visit-locations	73
Figure 5-3-F2	Result of search plotted on Google Map.	73
Figure 5-3-F3	Partial Result of the search for rated and non-rated cases.	74

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 5-4-F1	Google map with 2 frequent-visit-locations.	76
Figure 5-4-F2	Result of search plotted on Google Map.	77
Figure 5-4-F3	Partial Result of the search.	77

## LIST OF ABBREVIATIONS

### A

API                    Application Interface Programming

AJAX                  Asynchronous JavaScript

### H

HTML                 Hyper Text Markup Language

### M

MYSQL               Structure Query Language

### P

PHP                    Hypertext: Preprocessor

## CHAPTER 1: INTRODUCTION

There are many ways available to gather accommodation information, such as from newspapers, by asking information from friends, and search online websites. But housing information in newspapers changes daily and it is troublesome for user to keep track of the newspapers everyday to search for houses. Yet, the information available in newspapers also may be mostly from large companies advertising their newly released houses, apartments or even villas. Information obtained from friends may be limited or similar as they may obtain them from the same source, unless one has friend who are housing brokers or agents. Therefore, there are many websites or online portals created to help people to find houses.

However, looking for housing information online may not be a simple and easy task as one might think it is, as there are many online websites that contain housing information in the world wide web. But the majority of these websites only provide information and do not help users to make decision. In a way, as no other options is available to assist the user in decision-making, the user will have more data to analyze.

In order to improve on the relevance and usefulness of the search for housing information for a particular location and therefore to reduce the time the user would need to spend on analyzing the search result, **an accommodation search engine based on frequent-visit-location using multi-attribute utility theory (MAUT) is**

**proposed.** The system is designed to allow the user to input search by frequent-visit-location (for example, work place).

The purpose of the frequent-visit-location search is to pinpoint the point of search area. It could be a work place location, a shopping center location or a school address. The location will be confirmed by the user before a search is carried out for the possible available accommodation. This function or feature increases the number of relevant results returned by the system compared with area-based search by other websites because area-based searches tend to have a larger search scope compared with specific location search. Besides that, not all users are familiar with the area. Existing solution assume users are familiar with the search area. Please refer to scenario 1.1.

The search system should also provide an additional feature that allows the user to enter more than one frequency of visits on the location. This feature is not included on most of the available accommodation websites on the internet. Rating the frequency of visits on location is used by the proposed solution to find the most suitable location. Therefore the implementation of the rating features could improve the search results by providing a ranking information to the user on which are the suitable accommodation, as suggested by the system, using the multi-attribute utility theory.



### **1.1 A scenario that motivates this work.**

Including a rating function could allow the user to have a general idea about which are the houses that are more suitable to the user. The purpose of the rating system is to provide ranking on the result(s) returned to user as it could be an important consideration for the user. For example, if a fresh graduate who grew up in Johor Bharu, finds a job in Kuala Lumpur, he or she will be moving over to Kuala Lumpur to work but potential problems may arise if he or she seldom travels to Kuala Lumpur and thus may not be familiar with the area, roads and buildings of K.L.. Hence, it would be difficult for him or her to search accommodation based on area. However, it is much convenient for him or her to specify his or her work place. Apart from searching for houses or rooms, he or she may want find out additional details like how far the potential house or room from his or her work place or having the work place as the initial location, he or she might need to go a shopping centre for groceries.

Besides that, The search results is designed in the tabular format to allow the user to compare each and every result such as distance between the accommodation result with ease so that he or she could choose the best choice of houses available.

Thus, the motivation for doing this project is to provide a new search paradigm which is more effective and efficient for accommodation search engine.

## 1.2. Problem Statement

Based on the research conducted, some problems that appear on most of the accommodation websites are identified. Below are some of the problems found during the research.

- Most of the accommodation websites available on the world wide web are only offering area-based search engines for users, which lead to the search results returning random locations encompassing a very wide area, such as multiple locations spreading over an entire state of Malaysia, which resulted in a large scope in terms of reference or area of interest (Thinkproperty.com.my, 2012; Room4rent.my, 2012; Ibilik.my, n.d.; Trulia.com, 2012; Realtor.com, 2012; Houselocator.com, 2012). This area-based search is beneficial for the users, provided the users know the location area fairly well, but if the users are not familiar with the search locations, they might select an accommodation that are beyond their scope or area of interest, in terms of the proximity and distance, for example, from their workplace. If a user input or typed "Selangor" in the area-based search engine, the system will display all the accommodation results located within Selangor. Since the area of the state of Selangor is fairly large, the search engine may return numerous results, spanning the whole state, but the results may or may not fall within user's area of interest. For example, the result displayed might be two houses, one located in PJ and the other in Ampang, but both of these two places are located far apart from each other, as well as they may be outside the user's area of interest, but the user would not know it, if he or

she does not know the various areas within Selangor well.

- The decision made by users while browsing through the websites for accommodation, most likely they find it hard to make a decision on which accommodation is best for them. As most people may know, rating is a feature for ranking a series of results by showing the best as first ranked and the worst as the last based on some specified criteria or condition(s). However, the researched website does not implement the rating features to help users by providing a supporting ranking for them to identify which is better for the user in reference (Iproperty.com.my, 2012; Thinkproperty.com.my, 2012; Room4rent.my, 2012; Ibilik.my, n.d.; Trulia.com, 2012; Hotpads.com, 2012; Realtor.com, 2012; Houselocator.com, 2012).
- Search results provided by most of the websites available are in the list format (Iproperty.com.my, 2012; Thinkproperty.com.my, 2012; Ibilik.my, n.d.; Trulia.com, 2012; Hotpads.com, 2012; Realtor.com, 2012; Houselocator.com, 2012). It is hard to compare one accommodation with another by using this format as not all the useful information is displayed. For example, if there are 10 potential results arranged in a list, the user needs to view the first result's details first, before he or she could do a comparison with other results. However, the user would have to memorise or note down the details of each of the houses, for the next 9 houses. If the number of results returned by the search engines increases, so does the amount of information the user would need to memorise or

write down.

### **1.3. Objectives**

Below are the few objectives of the proposed project websites, an accommodation search engine based on frequent-visit-location using Multi-Attribute Utility Theory. The objectives are specified to solve the problems mentioned in the previous problem statement section, with the aim to provide more relevant results, to reduce the amount of time the user spent on searches, as well as to enhance the user's search experience.

- The proposed search system with a frequent-visit-location (for example, a non-residential building, like the user's place of work) search engine differs with most of the available accommodation websites, by allowing the user to specify their preferred location. That identified location will be the center point of interest of the search and it could be more than one location, thus the system will expand and span the search starting from the pinpointed frequent-visit-location. For example, if Lily tries to look for accommodation near her work place, she will input the address of her work place and the website will locate her work place and generate the search based on the work place location as the center point of interest of the search. The results displayed will be of relevance to Lily because the area used by the search system uses Lily's work place as central location for the search. This search methodology will ensure that the search results fall within Lily's area of interest.

- The search system will allow the user to rate the frequent-visit-location entered based on the frequency of visit to that particular location. The reason for implementing rating features into the accommodation search website is because rating allows the system to rank the accommodation results based on the frequency of travel of the user to the particular location and determine which accommodation is ranked best among all the returned result and this method is based on multi-attribute utility theory. However, the researched websites do not implement this feature on their accommodation search websites. For example, if Peter wanted to look for accommodation near his new work place and he also prefer to go to shopping centre on weekends. The travel frequency to the work place for Peter was once a day (to and back from his work place) for five days and he travels to the shopping centre once a week. Therefore the accommodation that should be first recommended to Peter should be an accommodation that is nearer to his work place instead of an accommodation nearer to the shopping centre.
- The system will generate a table of search results for the user. The table will contain distances of the house(s) and the frequent-visit-location(s) that the user had filled in. The table would also contain agent contact details and price information of each of the houses. The information displayed in table will be more informative as it simplifies the comparison process, which user may do on most of the accommodation websites that display search results in the list format.

#### **1.4. Scope**

There are many methods and platforms to design and develop an accommodation search system, such as creating websites, mobile apps, displaying interesting advertisements in various locations and so on. But this project is mainly focused on developing the search system using websites.

A fully functional accommodation website has a large structure and it is a combination of various sections such as the administrator area to monitor and maintain the website, the agent or landlord area, which they could post housing information on the website, the information area, which caters for user searches and provides adequate information, the advertisement area, which handles the advertisement, as well as the design section, which is responsible for the look and feel of the website. The proposed system scope will only on the information area.

The area that will be covered by the prototype website is the Petaling Jaya (PJ) area. Petaling Jaya is a place filled with a lot of job opportunities, amenities, vacation locations, and industrial areas; this means the PJ area is an area of popular demand for accommodation.

The rating function will be implemented in this prototype, and there is no limit as to how many frequent-visit-location(s) that user could choose. The rating function was based on the Multi-Attribute Utility Theory and adjusted according to the various situations in this prototype.

The proposed system is a prototype, and the intention of designing and developing the system is to prove that the concept of an accommodation search system with frequent-visit-location and rating features is feasible. Therefore the area covered will be smaller than those fully functional websites.

## **CHAPTER 2: LITERATURE REVIEW**

Search features and listing information are two important parts of an accommodation search website that can determine the success or failure of a particular website. This is due to the fact that the search engine is the major part of an accommodation website for users to rapidly search for their accommodation without browsing through the whole website. The listing information is the main reason for users to visit accommodation websites as one of the listings may be their potential accommodation. Websites that have the best search features and the best listing information provided to users will have a higher traffic flow on their websites. This will, in turn, leads to higher chances that the accommodation will be rented out or sold to users of the websites.

Therefore this project will concentrate on providing more relevant search results to the user. In order for this objective to be achieved, it requires equally important support from the search engine and the listing information features of a website. They are the major determinant that could affect the search results because they are the back-end system that processes the search and provides the results to the user.

The project will review on 8 accommodation websites among all the available accommodation websites on world wide web. The websites reviewed consists of 4 local websites (based in Malaysia) and 4 overseas websites. The review criteria are based on their search features and listing information. The local websites are



iProperty (Iproperty.com.my, 2012), thinkproperty (Thinkproperty.com.my, 2012), room4rent (Room4rent.my, 2012), and iBilik (Ibilik.my, n.d.) whereas the overseas websites are Trulia (Trulia.com, 2012), Hotpads (Hotpads.com , 2012), Realtor (Realtor.com, 2012), and House Locator (Houselocator.com, 2012).

### **2.1. Scenario**

The local websites were tested with the input "Swisslog, Kelana Jaya" (IT logistic Company) as the specific search location, and the input "Kelana Jaya" as the area-based search. The overseas websites were tested with input "Forbes Inc, New York" (Magazine company) as the specific search location, and input "New York" as the area-based search. The table below summarises the criteria that may be implemented in the proposed system and are of importance to users searching for accommodation. A point to note is that not all these criteria are commonly featured in all the reviewed websites.

**Table 2-1-T1: Criteria for comparison among 4 local and 4 overseas accommodation websites**

Website	Can consumer search by		Can consumer refine search criteria by rating?	Does search result generated by system can easily compare between each accommodation?	Map included to facilitate the search process?
	Specific Location? (Work place)	Multiple specific location at once?			
Malaysia					
iProperty	Yes.(Only Amenities-school, transport, food, shopping , hospital and place of worship)	No.	No.	No.	Yes
thinkproperty	No. (by area only)	No.	No.	No.	Yes
Room4rent	No. (by area only)	No.	No.	No. (Display result in 3x4 box format)	No.
iBilik	No. (by area only)	No.	No	No	No.
Overseas					
Trulia	No. (by area only)	No.	No.	Yes. Compare between the selected accommodation and other accommodation (maximum 4). Criteria - (Distance,price,bedroom,bathroom,property type)	Yes
Hotpads	Yes. (List all available accommodation with the specific location on map. )	No.	No.	No.	Yes. (Interactive map)
Realtor	No. (by area only)	No.	No.	No.	Yes
House Locator	No. (by area only)	No.	No.	No.	Yes.

## **2.2. Websites reviewed according to their strength and weaknesses.**

### **2.2.1. iProperty (Local website)**

The iProperty's website not only provides area-based search but also allows users to select amenities from the available list as their specific location search. These features are useful for people who are familiar with the amenities' names and their categories.

iProperty also have an additional feature, which is Geolocation. This feature can be beneficial because it detects current location of the users and perform searches on based on their location. However Geolocation's location feature is based on the location of the internet service provider of the internet user, therefore it might not accurately identify the actual location of the user.

But there are limitations as only the amenities category is available for user selection; they only provide 6 categories of amenities for selection. The categories are school, transport, food, shopping ,hospital/clinic and place of worship. Users are not allowed to input specific location themselves, such as their work place. There is only one search location for this search engine therefore ranking features is not included on the websites which allows user to rank based on the frequency of travel to the location.

### **2.2.2. Trulia (Oversea website)**

Trulia has the general features that most of the other review websites have such as area-based search and filtering option. It has an additional advantage over the others as it allows users to filter the accommodation based on school area (District-Apply in U.S). Another advantage of Trulia is that the search results could be selected and grouped to compare the criteria contained in the listing information.

However, there is a limitation to the comparable results, as it is only limited to 4 listings to be compared at a time. Trulia have a simple area-based search system but it does not associated with a rating features to allows user rate based on frequency travel to the location. This reason might because Trulia search location is limited to only one per search therefore the rating is not available.

### **2.2.3. Hotpads (Overseas website)**

The Hotpads website has the advantage of having an Interactive Map feature shown with search results plotted on the map. It allows the user to interact with the website by clicking the house icon on the map. Hotpads also provides search results in a list format. While this interactive map showed the nearest accommodation to the furthest distance as icons on the map from the pinpointed specific location, but when the user switches to the list format, the sequence is not shown by the nearest to the furthest as search results are listed in a random order.

This website has the advantage of allowing the user to perform specific location search. The location found will be displayed on the interactive map. But it does not use the pinpointed, user-defined location to perform future action such as listing the distances among available accommodation with respect to the pinpointed location. Instead it shows all the available accommodation with the pinpointed location on the interactive map. However, throughout the Hotpads search engine there is no ranking features as others search engine as well. The search location was also limited to only one places at a time which rating is not necessary in this situation.

#### **2.2.4. Thinkproperty (Local website), Realtor (Oversea website), House Locator (Oversea website)**

These three websites have the similar features as each other such as area-based search function. Filtering search results are by price, bedroom, bathrooms, property type. They also provide map features such as the Hotpads website mentioned above.

But there are limitations that are common for these three websites. These websites could not perform specific location searches, and they could not perform multiple search address at once as well. Thus it will not have rating features since it does not support multiple search address because rating is used to perform evaluation on the attribute which in this case the location.

### **2.2.5. Room4rent and iBilik (Local website)**

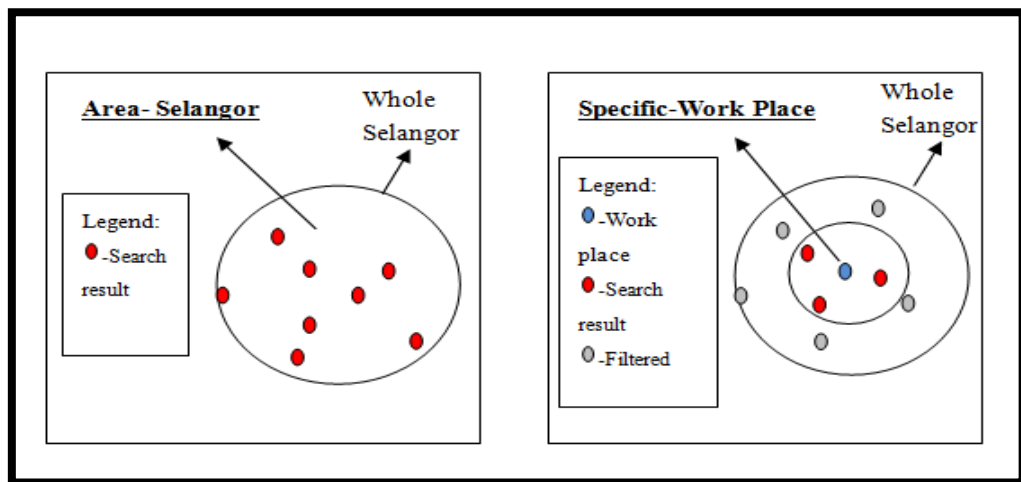
These two websites have the least features compared to the rest of the review websites; they are accommodation websites that provide searches for long term or short term room renting services.

These websites lack of filtering features even for the basic criterion such as price. Maps are not included to show the locations of the accommodation. The websites here are single location search engine which allow user to enter only one location on every search. Therefore, rating features will not usable in this situation and rating features was not found in these two website as well.

### 2.3. Review Criteria for the 8 websites in listed in Table 1

The explanation the review criteria of the 8 websites of the Table 1 is given below:

#### 2.3.1. Search by specific location



**Figure 2-3-F1: Search result different between area and specific search**

- The user is allowed to specifically inform the system of which area they would prefer to begin the search. This feature offers improvements to display the relevant information as compared to area-based search. Websites that have this feature would have an advantage over those without this feature because it could reduce and narrow down the search area to focus on providing more relevant results to the user.

### **2.3.2. Multiple search location at once**

- Multiple search location in one search was introduced to allow user to enter more than one specific location such as their frequent visit location. The purpose for this multiple location is allow user to list out their location and based on importance of the location provided the system could locate which accommodation is better compare to others accommodations. Furthermore by having multiple location on a single search, system could generate a radius which the search boundaries was auto calculated and accommodation result will be within the multiple specific location. Multiple location allows the rating feature to incorporate with the search engine.

### **2.3.3. Rate the search location (Multi-Attribute Utility Theory)**

- Most of the websites do not provide rating features for their user to perform rating on which of the search locations are more important. Different locations that user insert might have different weightage of importance, such as when comparing the work place with a shopping centre, the user may rate the work place as of a higher importance, as the user may travel to and from his accommodation, 5 times a week; than to a shopping centre that the user may only travel once or twice a week. Therefore, a better fit-to-user result will be returned by the system if a rating feature is implemented in the search system for user.



- The proposed system with rating features using the Multi-Attribute Utility Theory (MAUT) could return a more relevant result based on which frequent-visit-location is more important to the user. MAUT is a decision-support algorithm to support the each of the attributes that are of concerned to the user to evaluate which attribute is more important than the others. The decision result will be calculated and chosen based on the importance of each attribute assigned by the user. Each of the attribute should be independent of each other in order to have the best output using this theory, which in the case of this project, would be the frequent-visit-location provided by the user to be input into the prototype accommodation search system. A weightage would be assigned to each attribute based on its importance, using the rating entered by the user, to allow the evaluation of which attribute is better than the others. The general formula is  $u(v_1, v_2, \dots, v_k) = f(v_1, v_2, \dots, v_k)$ , where  $v_k$  is the  $k$ -th attribute of concern in reference (Fico.com n.d.).

$$u(v_1, v_2, \dots, v_k) = \sum_k w_k u_k(v_k)$$

**Figure 2-3-F2 : MAUT formula applied in this system**

For example, if the user input 3 frequent-visit-locations into the prototype accommodation search system, with their respective ratings assigned, a decision result can be calculated by the prototype system by using above the MAUT formula. In this case,  $w_k$  was representing rating chosen by user,  $u_k(v_k)$  was the distance obtained from the system and it will be summed up all the results to be the weighted result for the particular attribute (Fico.com n.d.).

2.3.4. Search result

<u>List Format</u>	<u>Table Format</u>				
<u>Accommodation 1</u> Price : RM 200 Bedroom : 2 Bathroom : 1 Agent Contact : Agent A	Price	Bedroom	Bathroom	Agent Contact	
<u>Accommodation 2</u> Price : RM 340 Bedroom : 1 Bathroom : 1 Agent Contact : Agent B	Accommodation 1	RM 200	2	1	Agent A
<u>Accommodation 3</u> Price : RM 500 Bedroom : 3 Bathroom : 2 Agent Contact : Agent C	Accommodation 2	RM 340	1	1	Agent B
<u>Accommodation 4</u> Price : RM 620 Bedroom : 1 Bathroom : 1 Agent Contact : Agent D	Accommodation 3	RM 500	3	2	Agent C
	Accommodation 4	RM 620	1	1	Agent D

Figure 2-3-F3: List format and table format listing information.

- The result generated by the system would be what the user is looking for when the user visits the accommodation website. Therefore, the format of results displayed to the user is one of the major areas that needs careful consideration in the design and development phase of the website.
- Displaying the accommodation available in a list would not be informative enough for user, because the user would have to view all of the available accommodation one by one in order to find out what are the available amenities or some other details (such as price, property size, property type, number of

bedrooms, as mentioned earlier). It will take a considerable amount of time and effort for the user to note down the accommodation that they like and compare each of them, one by one.

- The proposed system would create a table for the search result with details such as bedroom, bathroom, price, travel distance, agent contact details, as it would be more informative because it displays descriptions for both internal conditions as well as the external environment for a particular accommodation. The search result shown in the table format would facilitate the user to compare easily the available accommodation.

### **2.3.5. Inclusion of a Map in the website**

- A map could be used to briefly describe the location of the accommodation gathered by the search result. The map would give the user a general sense of where their potential accommodation are located in the map, and if they are interested to find out more they could travel along the route or path to the accommodation based on the map provided.
- The proposed system will be implemented with a map to allow the user to have a broad idea of the potential accommodation's location on a map.

## **2.4. Discussion**

The literature review aimed to gather and research on information that could be useful to improve on the initial concepts of the proposed design by analysing and comparing information found on other currently available websites. The search function, rating features and listing information are the major considerations in order to produce a good accommodation search website as most of these elements are available on almost every accommodation websites. However, each website has its own features and styles on the search function and also on the listing information display. One of aims of this proposed accommodation website is to provide more relevant search results for users.

The list of available amenities is another important consideration in this project other than the search features and listing information. This is because most of the websites provide the amenities information but did not include a rating function to the analyse the information. Therefore the proposed accommodation system will enable the user to search by frequent-visit-location of which could produce more relevant results to the user.

The rating feature analyses the result and allows user to know which accommodation is better among the others. The rating considered the travel distance of the frequent-visit-location and the accommodation itself to reduce the travel distance required by the user based on the Multi-Attribute Utility Theory (MAUT)

and proposed the best accommodation location for the user. However other accommodation will be displayed in result table as well to allow the user make better informed choices by themselves. The weightage on rating is just a piece of information for the user to consider and the user would still have to decide on which accommodation that he or she prefers.

**CHAPTER 3: METHODOLOGY**

In this project, **spiral methodology** is used. This method is chosen because it is a methodology which combines prototyping with waterfall methodology. One of the advantages of using this approach is that the design flexibility allows changes to be implemented at several stages of the project and quality assurance with the iterative process. This approach starts with minimal requirement and develops with more information throughout the life cycle. There are 4 stages in this approach:

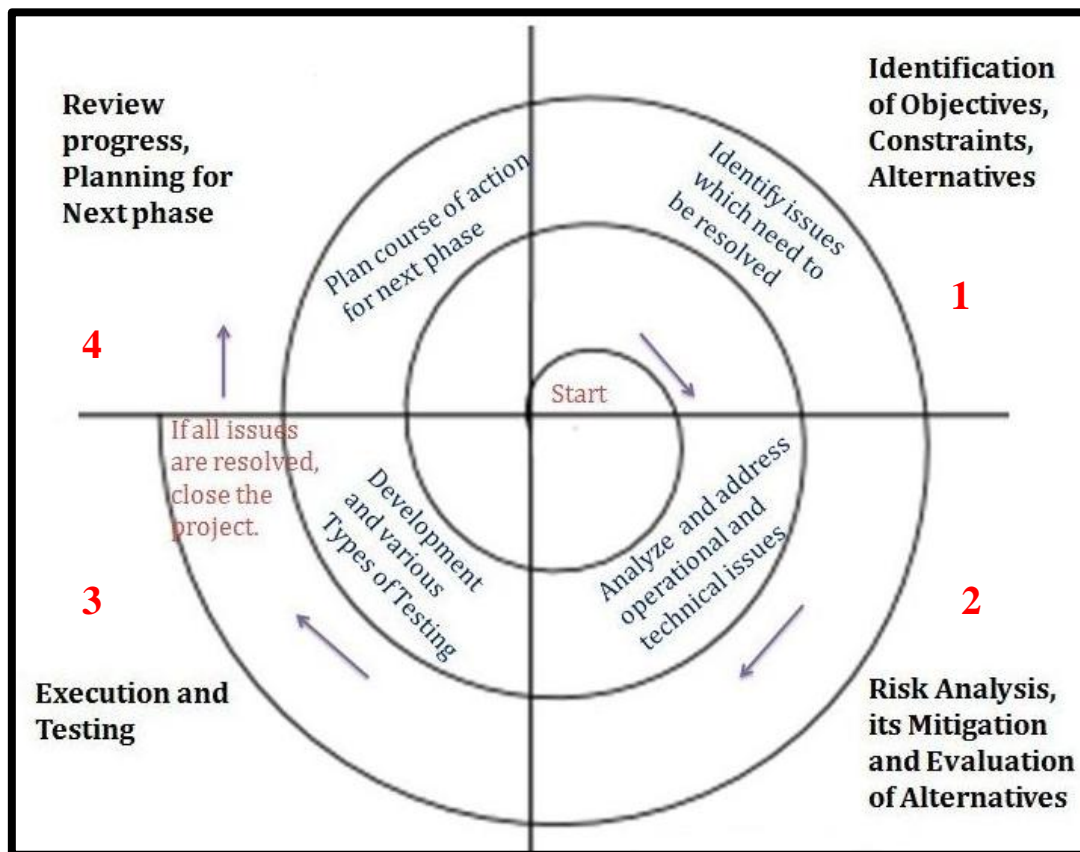


Figure 3-1-F1 : Spiral Model (Ianswer4u.com, n.d.)

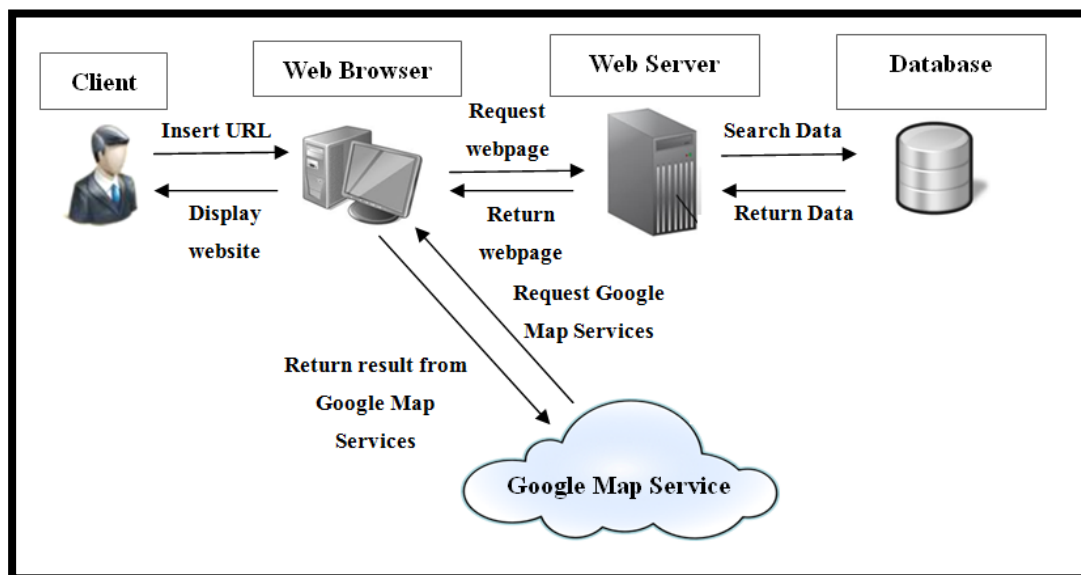
### **3.1. The Stages of the Iterative Process (Figure 3-1-F1)**

1. The project is defined by the 3 objectives which are to create a website with a search engine system that allows frequent-visit-location search features, implements a rating feature, and displays search information in a table format. The first stage iterative process focused on the design and creation of the prototype website with search engine capable of specific location searches. Other objectives will be developed and implemented once search engine prototype is created.
2. Risk Analysis is carried before the prototype is designed and created. The various risks associated with the project are identified. One of the risks that can be identified in the 1st iteration is that the search engine system might not work well with the Google Map technology, which would be implemented together to display the various locations in the website. The alternative would be to implement other map technologies if the implementation of the Google Map technology failed.
3. After the risk analysis in step 2, the prototype is created. The prototype website will be a preliminary design of the final product that would have basic functions to meet one of the 3 objectives of the project. The prototype will tested and refined as needed. The testing of the feasibility of implementing of the Google Map technology will be done at this stage.
4. At this review stage, the prototype will be evaluated with its strengths and weaknesses identified together with the risk(s) involved. Any improvement needed from step 3 will be implemented in next iteration planning. The next objective

requirement of the project will also be determined. The planning and design for second prototype to combine with the rating features will be carried out.

The iteration process will continue until all the objectives are achieved or the allocated timeframe of the project is reached.

### 3.2. Technology Involved



**Figure 3-2-F1: Client and server relationship.**

A prototype website is one of the outcomes of this project. Therefore, to create a dynamic website with dynamic data such as accommodation details, and map information, various website technologies are used. There are five technologies used to create a dynamic website. They are :

- HTML(Hyper Text Markup Language) (Graham, I., 1998).



- PHP (Hypertext: Preprocessor) (Webopedia.com , n.d.).
- JavaScript in reference (Quirksmode.org , 2008).
- AJAX (Asynchronous JavaScript) (Kyrnin, J. n.d.).
- JQuery (Techterms.com 2013).

Figure 3-2-F1 show that the system will be using a database to store the important data, such as accommodation information (Roseindia.net , 2008). The data storage is required because all the accommodation information could not be stored in the webpage itself. A database is used in this project to store the accommodation details, user details and agent and/or landlord details. These are important data sets to which an accommodation search website should contain and be properly recorded and stored.

The project presented a search for accommodation website which allows the user to perform frequent-visit-location searches. The system is an enhancement project which covers improvements to the search engine functions and listing information display area. These two areas present two major advantages of an accommodation search website over other currently available accommodation websites because they are the core functions of an accommodation website. The system allows the user to do searches using the non-residential buildings as the points of search and rates the locations using the Multi-Attribute Utility Theory (MAUT) (Fico.com n.d.). The rating allow the system to evaluate which accommodation is more suitable for the user based on the rate provided by the user

itself. This could increase the number of search results that are relevant to the user. The search results will be displayed in table format that would make it easier for the user to compare the results as well as to enhance the user’s search experience. The purpose of developing such search system is to provide more relevant results to the user, reduces the amount of time spend on searching accommodation and also support the user in his or her accommodation selection decision making.

### 3.3. Analysis Diagrams

#### 3.3.1. Analysis of the Use Case Diagram

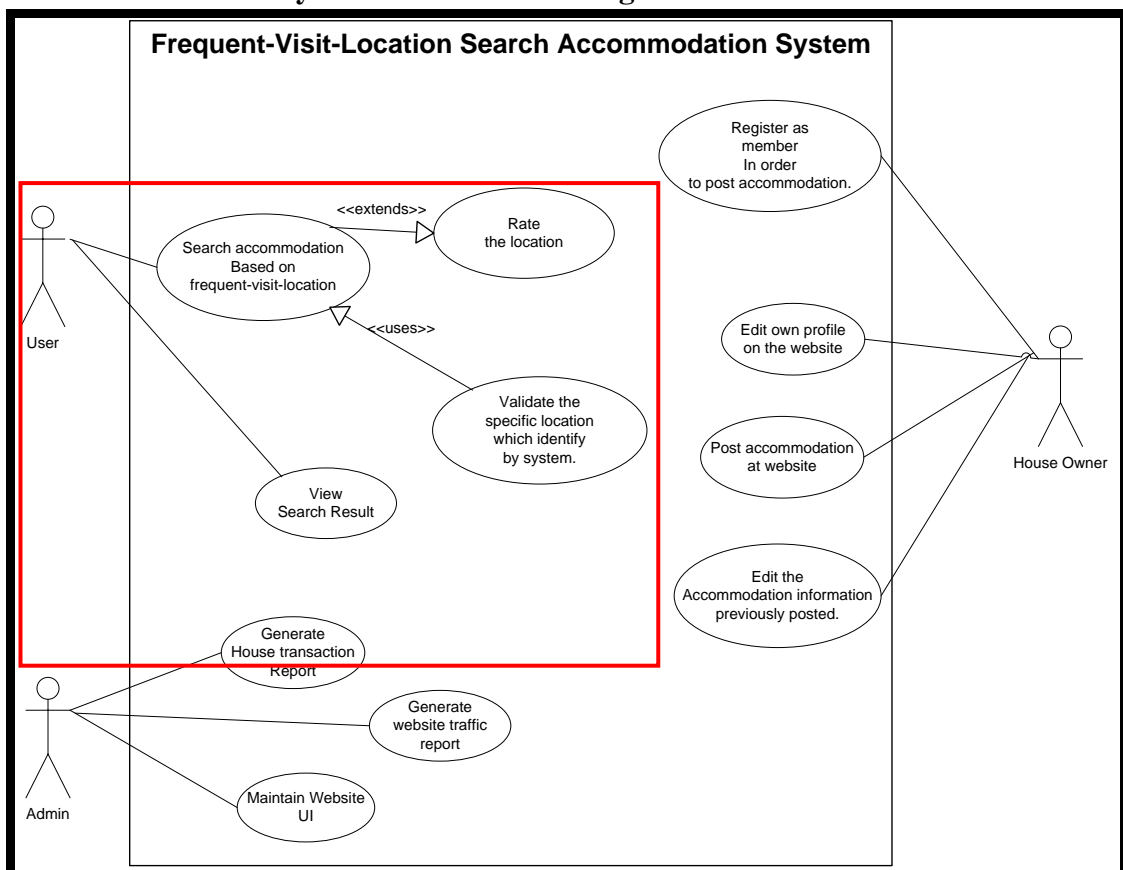


Figure 3-3-F1 Analysis Use Case Diagram

The Figure 3-3-F1 use case above shows how the structure of the full functional accommodation search system should be. It includes all the relevant areas such as admin, house owner and users action. However in this project, the area that will be implemented in the prototype website is the user area which is specified within the red box in the diagram.

The user could perform two actions, one of which is to search for relevant houses or rooms and the other is to view the result generated by system. According to the diagram above, the user have the choices to rate the location input by the user to get more relevant results. In order to provide more relevant results to the user, the user have to perform a series verification steps to enable the system to determine the specific location which identified by system.

In addition, the user will be viewing the result generated by the system search. The result will be displayed in table format to enable the user to compare ~~between~~ each accommodation result easily.

3.3.2. Analysis of the Activity Diagram

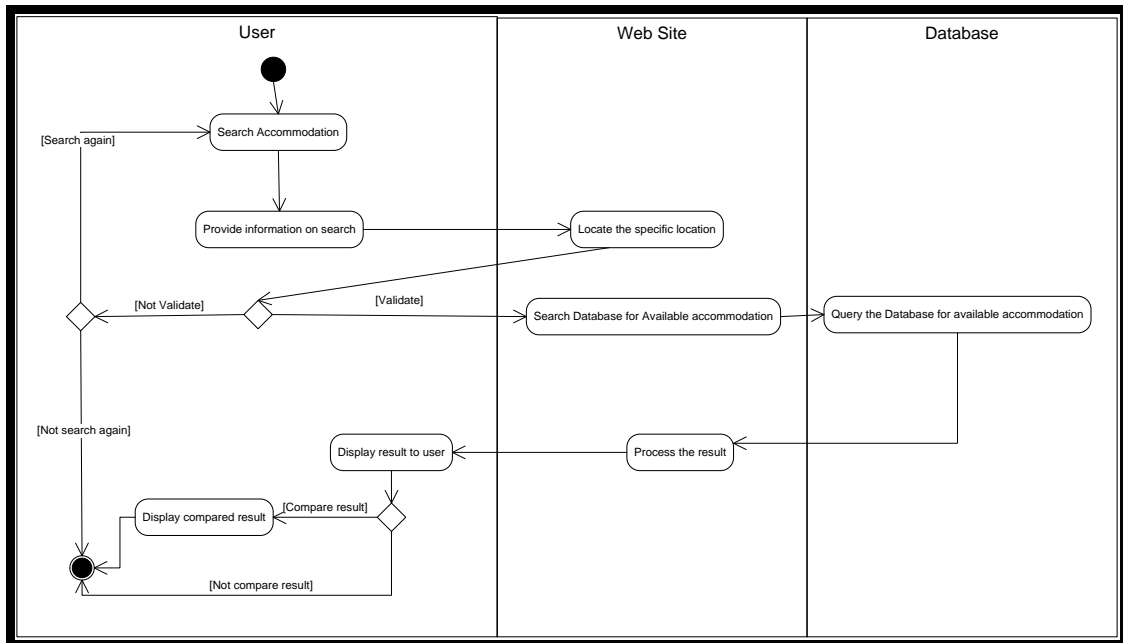


Figure 3-4-F1 Analysis Activity Diagram

Figure 3-4-F1 shows that the system activity diagram above shows the role performed by the user, website and database based on the whole process of the accommodation search. It begins when the user enters to the accommodation website and performs an accommodation search by inserting the specific location for identification. Once the user identified the location and performed rate or non-rate action, the relevant accommodation information will be taken from the database and displayed to the user through some process.

### 3.4. Behavioural Diagram

#### 3.4.1. Sequence Diagram

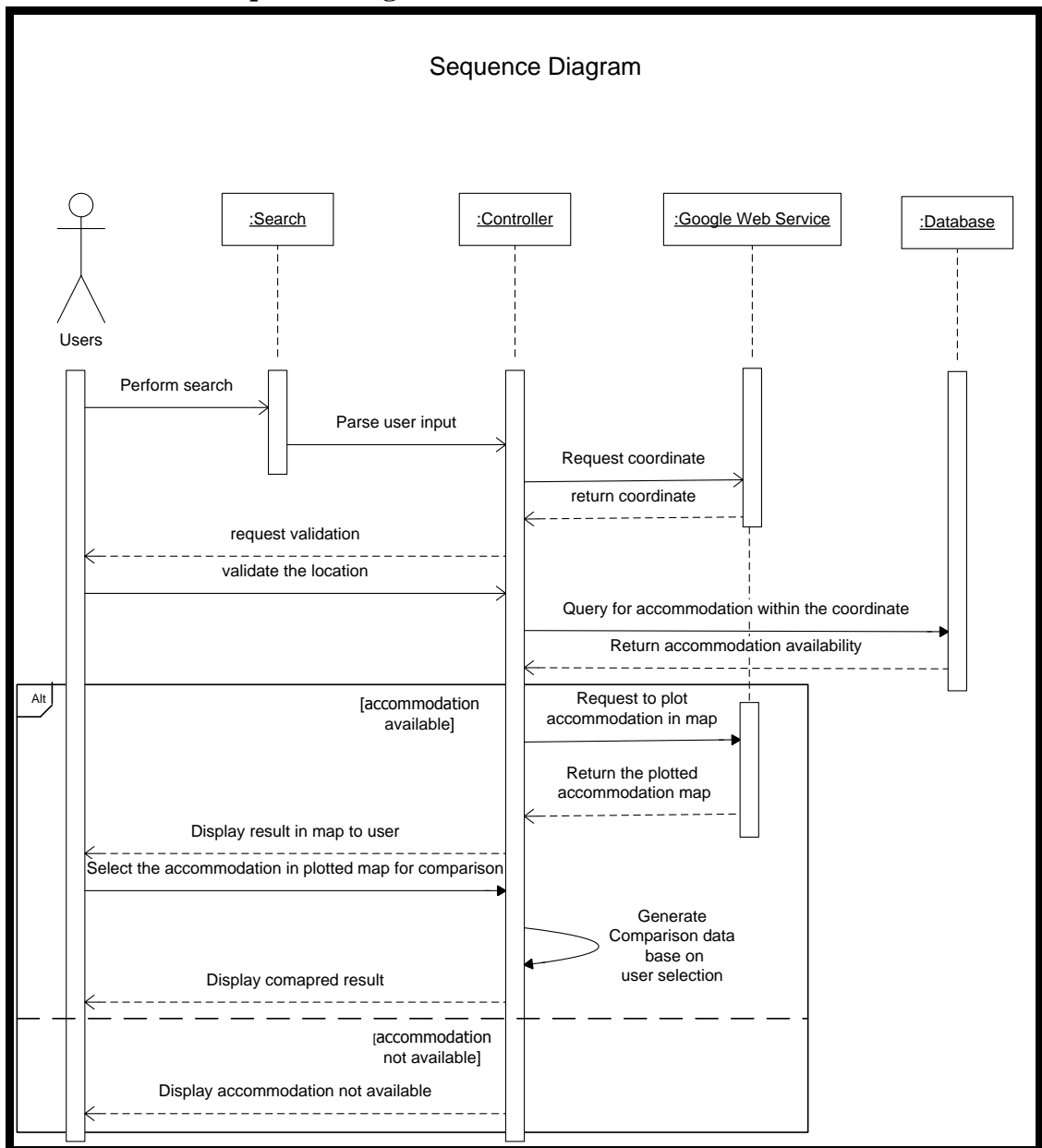


Figure 3-5-F1 Sequence Diagram

The Figure 3-5-F1 sequence diagram above shows the action user, and the classes perform in the search accommodation process throughout time. The sequence of the process is important in this diagram as it shows the actions performed at a specific time.

The user timeline would start and end only when the user finished his or her search process. Once the user entered to the websites, the timeline for the user would be started, and when the user performed the search, the timeline for search class and perform action would also be activated. Once the search information was parsed to the Controller class, the search class would not be used anymore until the user requested to perform a new search of which the process would start all over again.

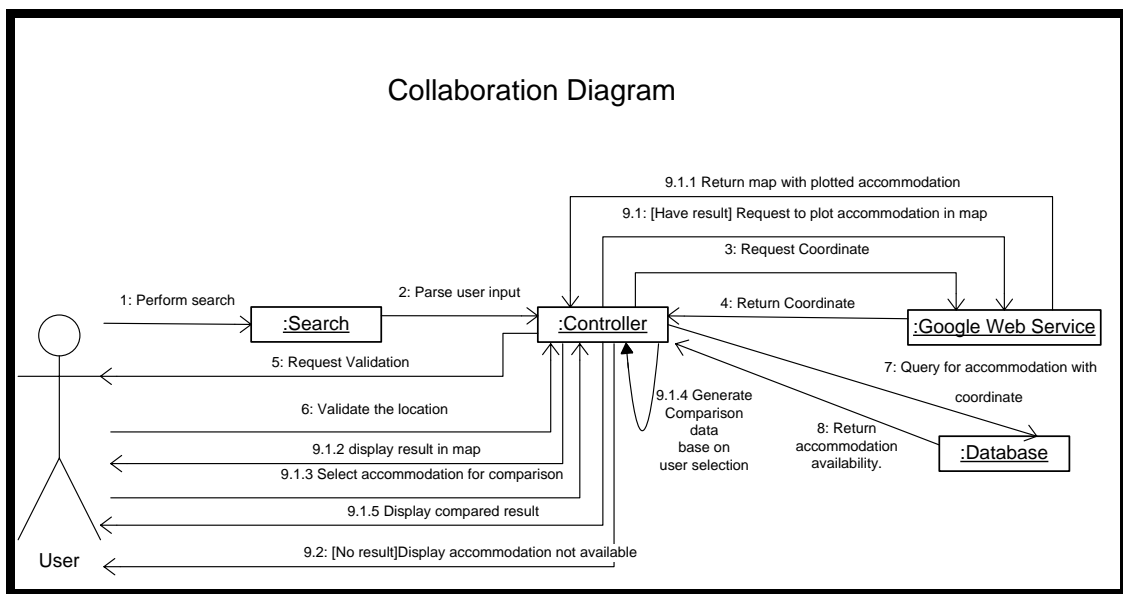
As mentioned earlier, Controller class is the major coordinator in this system; therefore the lifetime of this class is same as the user timeline until when user has stopped performing any actions, the controller class would end as well.

The Google Web Service class will have its lifetime activated when the Controller requests on the co-ordinates of a specific location and it would stop there until if there is available accommodation in the next process, when it will be called upon to plot the accommodation on the map and the lifetime of Google Web Service is ended after it returns the plotted map to the Controller class.

The Database class will be activated only if the specific location is available and

called upon to perform query for available accommodation; its lifetime will end after it returns the result to the controller class.

### 3.4.2. Collaboration Diagram



**Figure 3-6-F1 Collaboration Diagram**

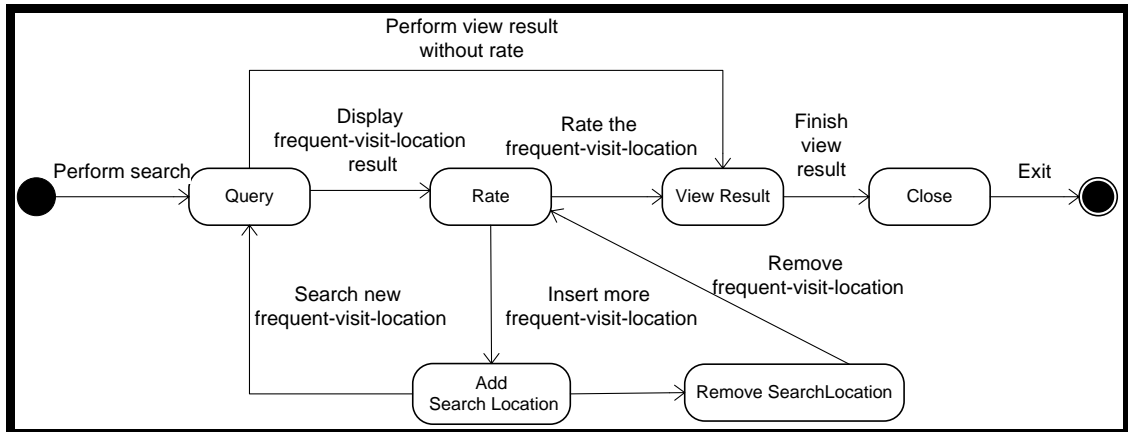
The Figure 3-6-F1 shows that the user will perform a search and parse the information such as frequent-visit-location and rating to the controller class which in turn will obtain the information and parse the information to Google Web Service class to validate the user input on a specific location. After that, the location will be transferred back to Coordinator class, which will then requests user to validate the location.

Once the location is validated, the Coordinator will assign a task to the database to query the available accommodation based on the location obtained by the Google Web Service class and the information obtained earlier from the Search class. Once the Database class has finished the query, it will return the result to the Coordinator. At this stage, there will two possible results given by database class; whether the accommodation is available or accommodation is not available.

For the accommodation is available scenario, the coordinator will request the Google Web Service to plot the available houses on the map and display the result to the user. The user is allowed to perform further action, that is to select which accommodation to perform a detailed comparison. The Coordinator will perform the comparison task and return back the compared result. However for the second scenario when accommodation is not available, the Coordinator will just return a message to the user stating that accommodation is not available.



### 3.4.3. State Diagram



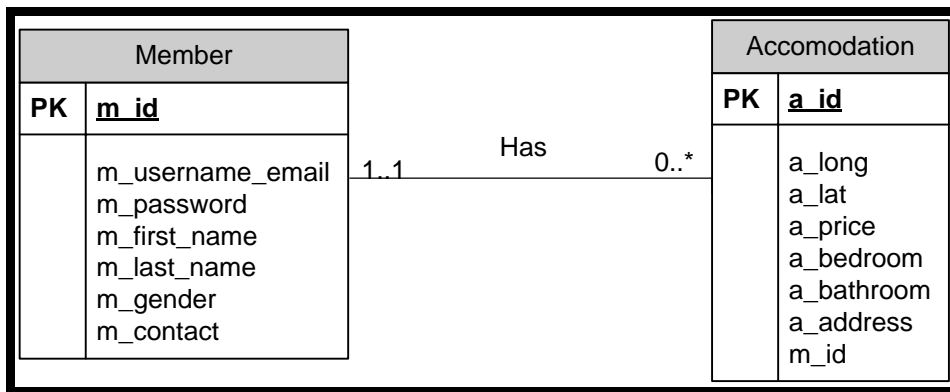
**Figure 3-7-F1 State Diagram**

Figure 3-7-F1 shows that the system will enter to query state when the user performs a search for accommodation. The system will enter query state and perform the query and search for available frequent-visit-location and display the result for user to verify. Once the location is verified, the user will proceed to the rate state, in which the user is given the option to view result with or without rating. After the rate state, the user has the optional action to insert more frequent-visit-location point(s), in which the system will enter to the add search location state. If the user wants to perform the search for frequent-visit-location once again for the newly added column, the system will return to the query state. However, the user is allowed to remove the newly added search column, if the user changed his or her mind afterwards, in which the remove search location state is entered, and the system will then proceed back to rate state and allow user to proceed to view result. Once the user is satisfied with the number of frequent-visit-location(s), the user could view the result, in which the

system will enter into the view result state as the result is generated by the system. Lastly, when the user has finished viewing the result, the user will enter into the close state; or in other words, the user has finished the search.

### 3.5. Structural Diagram

#### 3.5.1. Entity Relationship Diagram



**Figure 3-8-F1 Entity Relationship Diagram**

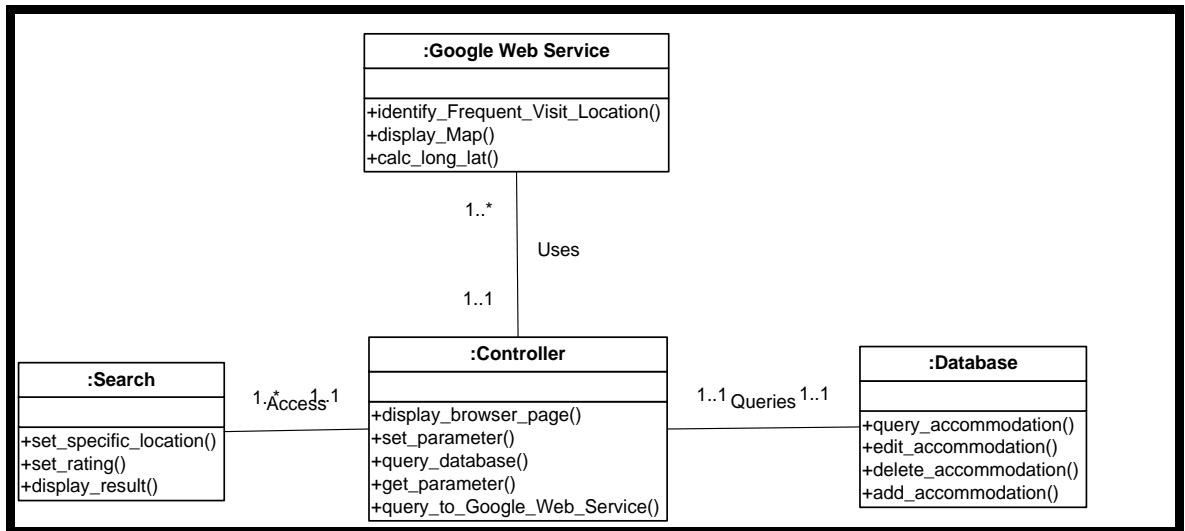
In the Figure 3-8-F1 ER Diagram above, there are 2 tables that are used in the proposed system. They are two table which is Member and Accommodation.

The Member table functions as a storage of the details of every member to uniquely identify the owner of the accommodation posted on our website. The attributes are all the member’s personal details; using m\_id as primary key to keep track of each member.

The m\_username\_email and m\_password is the id and password to login to the websites and to keep track of user activity, m\_contact is recorded to establish the communication channel between the landlord and the user. Other columns such as m\_first\_name, m\_last\_name and m\_gender are to store the other personal details of the member.

The Accommodation table is the most valuable asset on the website database because it consists of the accommodation details wanted by the user. The accommodation table will contain all the details of an accommodation such as latitude, longitude, price, the number of bedrooms, bathrooms, addresses and the landlord id for the accommodation and it is identified by the a\_id column within the table.

### 3.5.2. Class diagram



**Figure 3-9-F1 Class Diagram**

The Figure 3-9-F1 class diagram consists of four major classes; the Search class, the Controller class, the Google Web Service class and the Database class.

The Search class is used for handling actions and methods that are required to perform a search. It contains the method of setting a specific location, and to display the result. These functions are required for getting the result back to the user.

The Database class is a class which handles the database tables that could perform the query on searching for available houses, as well as adding, deleting and editing the accommodation.

The Google Web Service class is the API from Google, which allows the system to identify the specific location, display maps on the websites and calculate the longitude and latitude of a specific location. All these functions are pre-defined by Google when we are utilizing the API.

Last but not least, the Controller class is the most important component in this whole system, because it is the coordinator for each of the classes and it is the middle man for every process in the system. Therefore, it has the most functions such as displaying the browser page to the user, getting and setting the parameter obtained from the search class, and for the database and Google Web Service queries to obtain the search results.

### **3.6. System Progress Conceptual Website Step**

The following section describe the conceptual parts of the proposed accommodation search website. It mainly consist of the Search Bar section, the Verification section and the Results Display Page section.

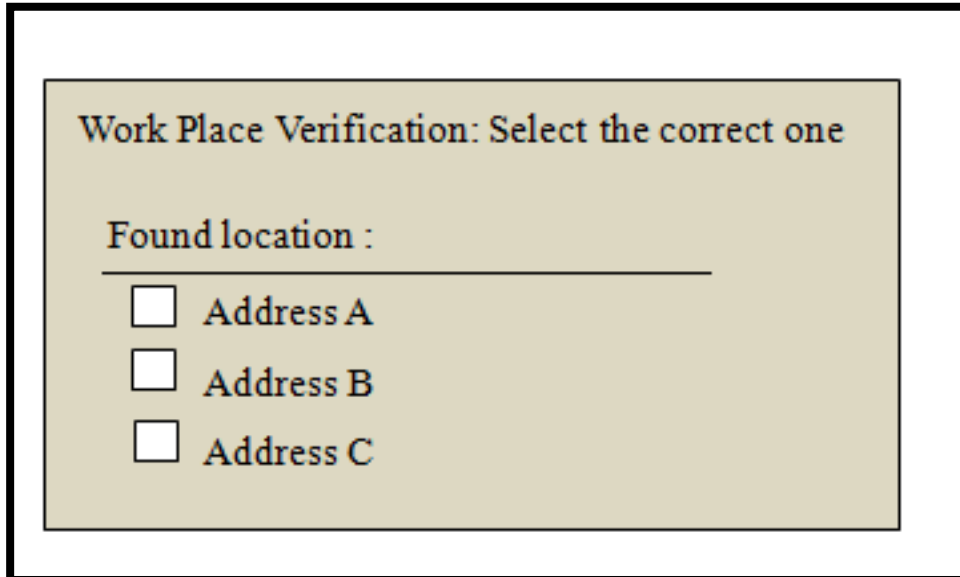
### 3.6.1. Search Bar

The image shows a conceptual search bar interface. It is titled "Search 1 :". Below the title, there are two rows of input fields. The first row contains a text field labeled "Frequent Visit Address 1" and a button labeled "Rate". The second row contains a text field labeled "Frequent Visit Address 2" and a button labeled "Rate". At the bottom of the interface, there are two buttons: "Add Location" and "Confirm".

**Figure 3-10-F1: Conceptual Search Bar**

Figure 3-10-F1 is the conceptual search bar allows the user enters the frequent-visit address as input in the text field, and the conceptual Search Bar allow user to choose to rate or not to rate, by simply select the rate displayed in the "Rate" area. User could add more addresses by pressing "Add Location" button. The "Confirm" button will lead to next figure to verify the location search by system.

**3.6.2. Verification**



The image shows a conceptual verification page with a light beige background. At the top, it says "Work Place Verification: Select the correct one". Below this, there is a label "Found location :" followed by a horizontal line. Underneath the line, there are three radio button options: "Address A", "Address B", and "Address C".

**Figure 3-11-F1: Conceptual Verification Page**

Figure 3-11-F1 is the conceptual verification page and purpose of this page is to confirm that the area searched is related to what the user has intended to search for. Therefore search result will be more relevant for the user.

3.6.3. Result display page

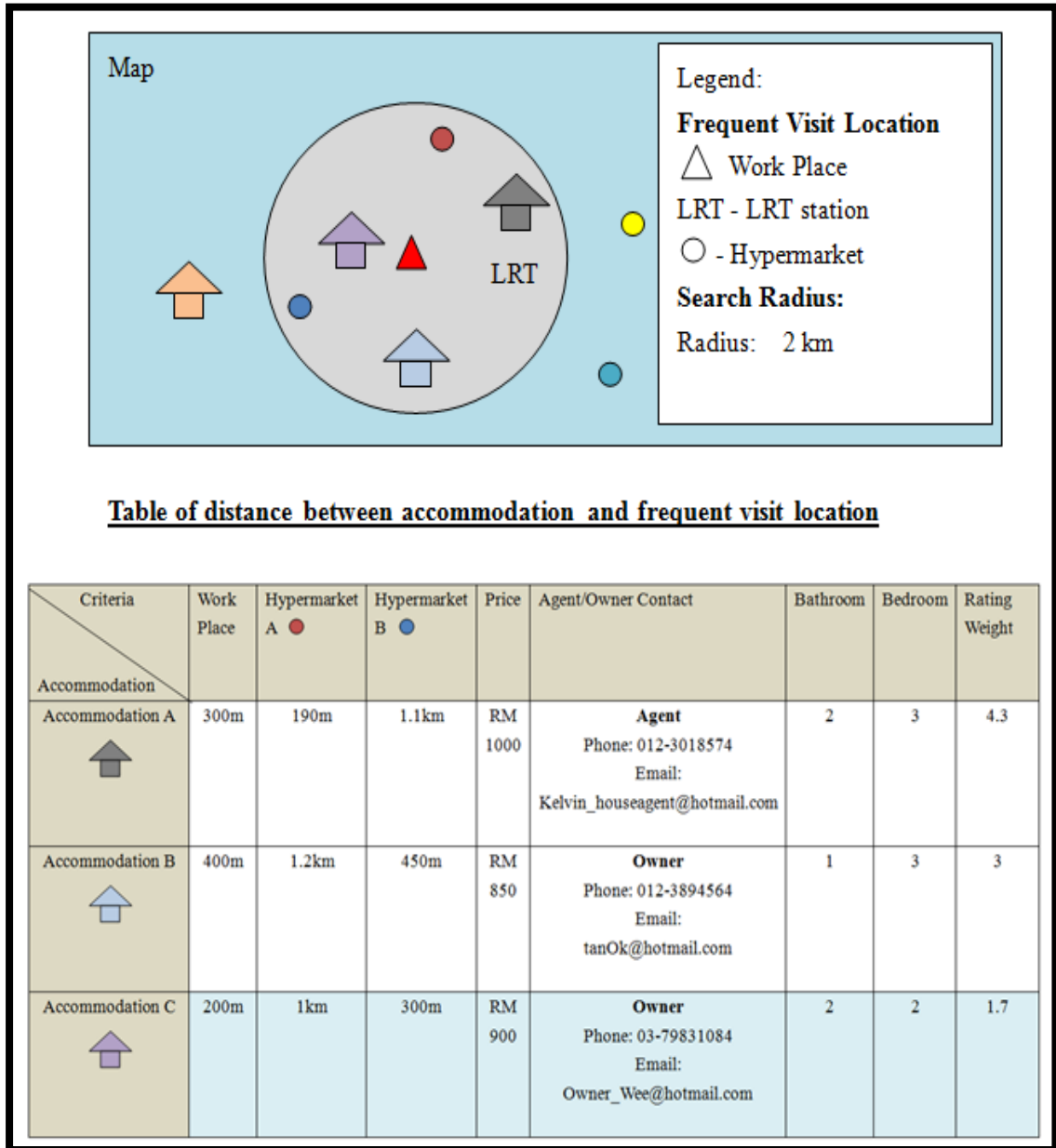


Figure 3-12-F1: Conceptual Result Display Page



The Figure 3-12-F1 displays the result found by the system in a tabular format. The criteria for comparison is based on user the selected options. The accommodation which will be listed on table is the distance between each accommodation and the frequent-visit-location input by the user. This could reduce the number of irrelevant accommodation results to the user. It is clearer and easier for the user to do comparison with the usage of the tabular format to display accommodation information. The rating was to determine which accommodation suits the user best. Based on the Figure 3-12-F1, the Accommodation C was recommended by the system with the lowest value in the rating weightage.

### 3.7. Gantt Chart

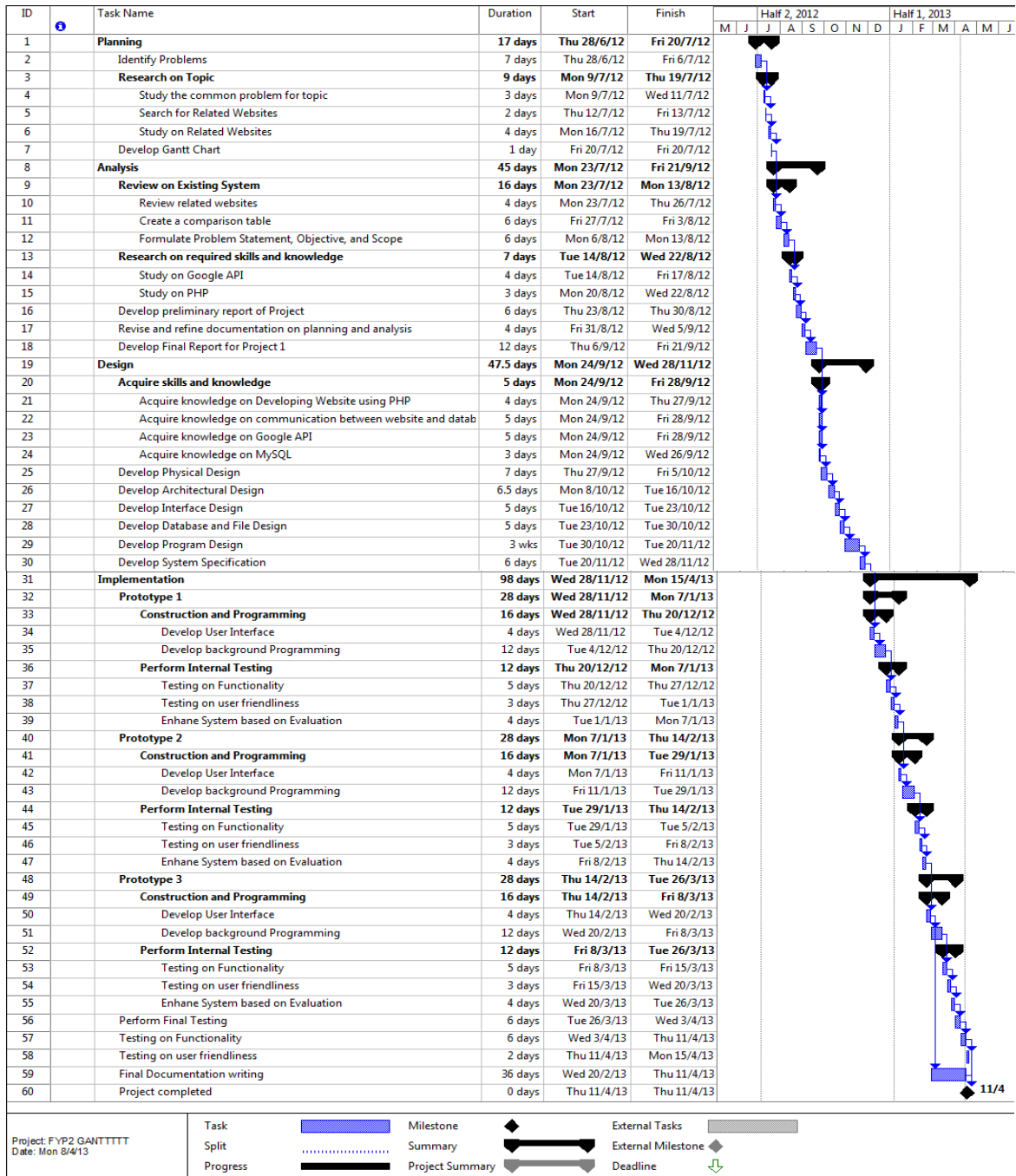
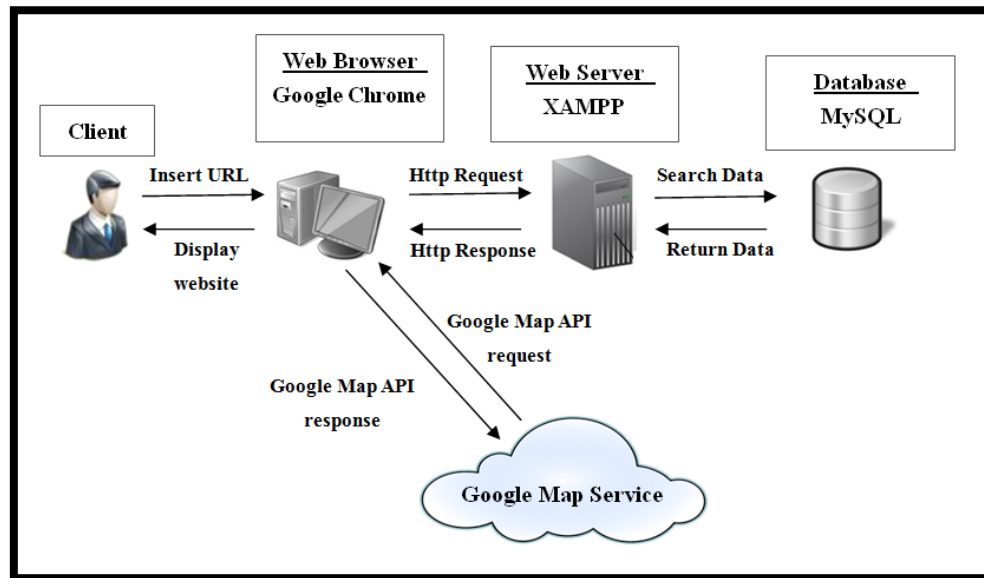


Figure 3-13-F1 Gantt Chart

The Gantt Chart in Figure 3-13-F1 was the progress planning for the overall system development phase which consist of Planning, Analysis, Design and Implementation phase. The time period for the whole system finish was from 28 May 2012 until April 11 2013. The figure shows the planned progress on every phases and the duration will perform the specific task listed in the chart

## CHAPTER 4: IMPLEMENTATION

## 4.1. Overview of system structure



**Figure 4-1-F1 Overview of the system structure.**

The Figure 4-1-F1 shows the overview flow of the system. There consist of a client to use the system though the web browser and the web browser will communicate with web server to obtain the websites based on the client input URL. The user will interact with the web browser on user's intention on performing which actions. The action then will trigger JavaScript/ Asynchronous JavaScript (AJAX) within the web pages to perform actions accordingly. Google Server will be communicated by web browser when there is Map, Place or travel distance is required to display on webpage or the user wanted to perform search on available houses will access to the Database system through web server. Lastly there will be HyperText Markup Language(HTML) contain in the web pages itself to display a more user friendly Graphical User Interface(GUI) to user such as

buttons and textbox.

#### **4.2. Web Server**

The server which implemented in this prototype website was XAMPP server. XAMPP is a acronym of **X**-cross platform, **A**pache Server , **M**ySQL , **P**HP, and **P**erl. XAMPP was a program which allows personal computer become a host web server. The XAMPP was chosen because it was flexible can be apply on personal computer and easier for development purpose. XAMPP supports MYSQL, PHP, Apache HTTP Server and Perl. The XAMPP version used in this project was Version 1.7.3 for windows.

The server perform as the middle man for the whole website structure. Web servers is like a container for web sites. Therefore Http Request send from Web browser will be searching the available websites which contain within the web server, if there is such websites it will return the main page, on the other hand error will displayed if website not found.

The web site inside web server will obtain parameters from web browser through Graphical User Interface (GUI) of web pages and client side language JavaScript or AJAX. The website will base on the programmed code to decide to communicate with which component.

The web site have multiple actions when receive different parameters and different action will be carried out. Below was the few action will perform by web server.

I) Reply a http request to the web browser when request for web site. E.g. User request to visit our project main page.



**Figure 4-2-F1 The Main page for the project**

II) When the JavaScript/AJAX from web browser request to change pages or perform some calculation on PHP pages. Web browser will request the server for another existing web pages. E.g. User perform login on the main screen and the web browser activates JavaScript/AJAX to request validation from another page inside Web Server to perform validation on the user login.

```

//User wish to login
if(buttonSelected == "loginBut")
{

    var userName = document.getElementById("username").value;
    var passWord = document.getElementById("password").value;

    //Check if user exist
    $.ajax({
        type: "post",
        url: "validate.php",
        dataType:"json",
        data: { hiddenUser: userName , hiddenPass: passWord , types: 'login' },
        success: function (response) {
            if(response.status == 'success') {
                document.getElementById("usernameE").value = userName ;
                document.getElementById("m_id").value = response.m_id;
                var myForm = document.getElementById("verified");
                myForm.submit();
            } else if(response.status == "error") {
                alert("Wrong user!" + response.message);
            }
        },error: function(xhr, textStatus, error){
            console.log(arguments);
            alert("ajax Error");

            alert("Error occured: "+error+" "+xhr+" "+testStatus);
        }
    })
}
}

```

**Figure 4-2-F2 Sample JavaScript and AJAX code of requesting validation from Web Server on validate.php page inside the Web Server (Back-End)**

III) Interact with Database on server sided language (PHP). The Server contains a few server-side language pages such as validate.php to validate user. The PHP code is directly interacted with the database from the web server itself because it works in back-end. Validation of user, Adding House, and Search available House is under this section to communicate with MySQL database using PHP server-sided language.

```

//Obtain the username and password
$usernameReceived = $_POST["hiddenUser"];
$passwordReceived = $_POST["hiddenPass"];
$type = $_POST["types"];
$con = mysql_connect('localhost','root','') ;
if (!$con)
{
echo 'could not connect<br /> ' ;
} else {
mysql_select_db('fyp', $con);

if($type == 'register') //Register only check user
{
$query = "SELECT * FROM member where m_user_email = '" . $usernameReceived . "' ";
} else if($type == 'login') //Login must check user and password
{
$query = "SELECT * FROM member where m_user_email = '" . $usernameReceived
. "' AND m_password = '" . $passwordReceived . "' ";
}
//echo $query;
$result = mysql_query($query);
if($result && mysql_num_rows($result) > 0 ) {

```

Communicate with MySQL database

**Figure 4-2-F3 Sample Validate.PHP which search MySQL database to validate user.**

### 4.3. Client

Client is the person who require assistant in search for accommodation which also called as user.



**Rating is based on how frequent the visitation to the location. The smaller rating represent more frequent visit**  
 E.g Work Place - Daily - rate 1  
 Shopping Center - Weekly - rate 2

Address 1:   Rate:

**Figure 4-3-F1 The search area for user input (Front-End)**

```

if(buttonS == "findBut"){
  var searchText = "searchText_"+numbers;
  $('#'+searchText).trigger(event);
}

```

Triggered the button press event

//Declare textbox ID with the //number obtained previously. //Trigger the event of textbox

Linked to here

```

//Key Up Event
$(".st").live("keyup",function(event){ // When text box event happen

  var stID = $(this).attr("id"); //Store the event happened text box ID
  var stringST = stID.split("_"); //Get the textbox ID name
  var numberST = stringST[1]; //Get the textbox ID number
  //Event is Enter(13)

  if(event.keyCode == 13){
    searchAdd(numberST);
  }
});

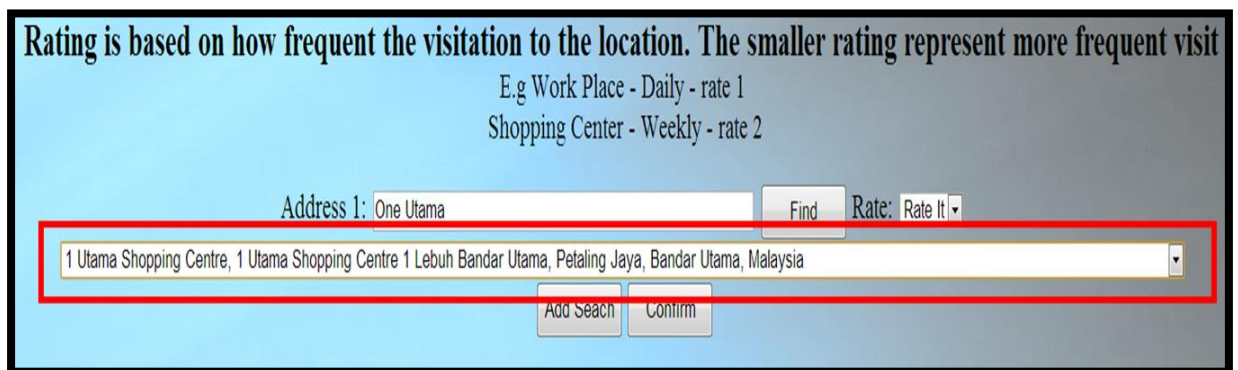
```

Call the method to search address.

**Figure 4-3-F2 The code for find function (Back-End)**

User will open a web browser on their personal computer to access to the web sites. The web browser will send a HTTP request to the web server to obtain a website through the Uniform Resource Locator(URL) typed at the user's web browser. The web site will be shown to user in GUI format. When user wanted to search for the information about available of accommodation on certain areas, they must have a

certain fixed location that they know such as work places or locality. Therefore the GUI designed with a search textbox allow user to key in their location and it will send the http request to web server with the string of the input data.



**Figure 4-3-F3 Display address returned by Google Servers**

The web browser will receive HTTP response on the address is found or not found indicated in the combo box GUI will be displayed after user press the GUI "Find" button. The address will be send to web server and it will parse the information to Google Server to obtain address.

Once user finish their locating the address, they could obtain the result by clicking the "Confirm" Button with or without rating selected. Once confirm button pressed the back-end will perform parsing parameter to another pages perform others action.



**Figure 4-3-F4 Action performed when confirm button pressed.**

User also allow to Login and Add House on this prototype system. In which the system will perform validation on searching database on the username and password for login. If user was not found will alert the client user is not exist. The houses will be added into MySQL database for searching purpose.

#### 4.4. Web Browser

Web browser is required in this project because the project developed is an websites. It could only be accessed through a web browser. The compatible web browser for this project was Google Chrome version 25.0.1364.. Web browser bridges the interaction between user(client) with the web server. Web browser is to performs the JavaScript, Asynchronous JavaScript(AJAX) and Multi-Browser JavaScript

Library (jQuery) which under the web pages returned by web server.

Web browser will take action according to the JavaScript/AJAX coded inside the web page itself. Once the condition meet it will activate the JavaScript and perform action such as searching for address from Google Services, storing house address, login, request display result and so on. These activities are controlled by the JavaScript/AJAX within the web page because the action performed by user are unpredictable, but we could predict what user intention next base on the buttons and event trigger.

```

//When button clicked event
$("#input").live('click', function(e){

var buttonSelected = e.target.id;
var event = jQuery.Event("keyup");
event.keyCode = 13;
var splitButton = buttonSelected.split("_");
var buttonS = splitButton[0];
var numberS = splitButton[1];

if(buttonS == "findBut"){
    var searchText = "searchText_"+numberS;
    $('#'+searchText).trigger(event);
}

```

JQuery code to detect button click event

//The button clicked ID  
//Declare key up event  
// Whatever keyCode you need

//Get the button name  
//Get the button number

//Declare textbox ID with the //number obtained previously.  
//Trigger the event of textbox

Check which condition user wish to perform

**Figure 4-4-F1 Sample Code for Event Handler JQuery**

#### 4.5. Database

Database is an component for storing permanent data. The data which important

to the project such as house address, details, user identification, users password and users personal detail. These data is essential for project to keep track of the record. Therefore what database to choose require to consider the languages we using as well. In our case, the database which will be integrated with this project was MySQL. MySQL was compatible with PHP coding language for web pages and it was used in this project. The other reason MySQL was chosen because of its reputation and the software simple version was free of charge for us to use. The version of MySQL used in this project was 1.2.11.

In this MySQL database it contains two table which is Accommodation and Member table. As the name itself mentioned, accommodation table is storing the information of available houses such as a\_lat and a\_long for Google Services, a\_price, a\_bathroom, a\_bedroom, a\_description, a\_address, a\_id and m\_id. Where the Member table consist few element which is m\_user\_email, m\_password for identification purpose and others personal details such as m\_contact, m\_gender, m\_first\_name, m\_last\_name, m\_middle\_name and m\_id.

### **4.5.1 Accommodation Table**

Accommodation id(a\_id) and member id(m\_id) is assign as INTEGER data type because it was the unique key for identify each and every row within the table. The value was auto increment when a new row of data was inserted into the table. Other than that, bedroom(a\_bedroom) and bathroom(a\_bathroom) data type also assigned as Integer because there will not be having half or three quarter room and it will only have a whole number room such as one, two or three. Latitude(a\_lat) and Longitude(a\_long) column in this table was assigned to FLOAT type data because the latitude and longitude contains decimals therefore FLOAT is the most suitable data

type. Description(a\_description) and Address name(a\_address) column was purely information column therefore it was assigned to VARCHAR data type to stores the string. When comes to monetary term column which is price(a\_price) Double data type is best match because monetary term used to be double format such as rm230.00, rm50.00.

### **4.5.2 Member Table**

Member id(m\_id) is to unique identify each row, and it was auto increment once a new row inserted into the database therefore will be assigning INTEGER. Member user name(m\_user\_email) and member password(m\_password) is stored to identify the existing of a member and the input can be combination of integer and string therefore assigned VARCHAR data type for this two column. Contact(m\_contact) usually comes in a series of number therefore INTEGER data type was assigned to this column. Gender(m\_gender) which approved in Malaysia is only two type which is Male and Female, to reduce database space only single character will stored represent each of them therefore CHAR data type will be used. Lastly first (m\_first\_name), middle(m\_middle\_name), and last(m\_last\_name) name was assign to VARCHAR data type due to name commonly is represented in string format.

### **4.6. Google Server**

In this project requires the supports of map services to locate, identify and search the existing location. The Google Map services is widely used on many websites and blogs. It shows that Google Map Service have high population in location based data. Google Map is well known Map service at least in Malaysia therefore it will be easy to search for solution online if there is problem in coding or displaying Google

Services. The map services provided by Google available in free and pay version which the limitation was set higher towards the free version but the free version Google Service is still applicable on this project prototype concept proving.

The Google Services have many Application Programming Interface(API) which pre-defined codes for developer to apply on their system. It reduces the difficulty and times to develop a good location search website.

The role of Google Service in this project was to search location address base on an address string input by user. The JavaScript/AJAX within the web page will perform capturing and posting the request to Google Server.

```

function searchAdd(text)
{
    var textField = "#searchText_" + text;
    var addField = "addressChoices" + text;
    var value = $(textField).val();
    infowindow = new google.maps.InfoWindow();

    var googleapi = 'AIzaSyBtC3hLBuiqQ_N8DT4U5v1cg3nWg8kNr_w';
    //Using ajax to access to url with parameters, parse the json result into array (if success)
    value = value.replace(/ /g, "+");
    //Obtain result from Google Server for search the Address
    $.getJSON('http://anyorigin.com/get?url=' +
    encodeURIComponent(
    'https://maps.googleapis.com/maps/api/place/textsearch/json?location=' +
    'pj+
    '&radius=1000&query=' + value +
    '&key=' + googleapi +
    '&sensor=false') +
    '&callback=?',
    function(data) {
        arrayLatLng = new Array();
        formatAdd = new Array();
        addName = new Array();
        var latLng;
        var addressChoices = document.getElementById(addField);
        addressChoices.options.length=0;

        if(data.contents.results.length > 0){
            addressChoices.style.display = '';
            //Loop the result to perform actions
            addChoice(0, "Please Select Exact Frequent Visit Location", addField)
            for (var i = 0; i < data.contents.results.length; i++) {
                var currLatLng = data.contents.results[i];
                latLng = new google.maps.LatLng(data.contents.results[i].geometry.location.lat,
                data.contents.results[i].geometry.location.lng);

                resultLatLng[i] = latLng;
                var format_add = data.contents.results[i].name + ", " + data.contents.results[i].formatted_address;
                var add_name = data.contents.results[i].name;
                arrayLatLng[i] = latLng;
                formatAdd[i] = format_add;
                addName[i] = add_name;
                addChoice((i+1), format_add, addField);
            }
            selectedAddList[(text-1)] = arrayLatLng;
            selectedFormatAddList[(text-1)] = formatAdd;
            selectedAddNameList[(text-1)] = addName;
        } else { //If no result returned inform user address not found
            addressChoices.style.display = '';
            var notFoundArray = new Array();
            notFoundArray.push("Address not found");
            addChoice(0, notFoundArray, addField);
        }
    });
}

```

Request to Google Map Service API

Google Map Service return result

Figure 4-6-F1 Search address send by JavaScript to Google Services.



The Google Service used in this project also have Google Map which allow the system to display the location of address and houses into map. This allows user to visualize the differences between location.



**Figure 4-6-F2 Google Map with location plotted on map.**

The Google Service of Distance matrix also used in this project. This service is to calculate the travel distance between two location. This services is used when calculating the travel distance between matching houses in database with the frequent visit location input by user.

```

//Call the Google Api service
service.getDistanceMatrix(
  {
    origins: latlngObjectArray,
    destinations: destArrayUsed,
    travelMode: google.maps.TravelMode.DRIVING,
    avoidHighways: false,
    avoidTolls: false
  }, callback2 );

function callback2(response, status) {
  if (status != google.maps.DistanceMatrixStatus.OK) {
  } else {
    var origins = response.originAddresses;
    var destinations = response.destinationAddresses;

    var rowResultDistance = "";

    //Loop the result base on the original address length.
    for (var i = 0; i < origins.length; i++) {
      var results = response.rows[i].elements;
    }
  }
}

```

**Figure 4-6-F3 Sample JavaScript Code to calling the Google Service and obtain the return result.**

All the above mentioned services are performed using JavaScript and AJAX to perform communication to the Google Service to obtain the result. The Result will be captured inside JavaScript and AJAX and perform further action and display result to user.

#### 4.7. Multi-attribute Utility Theory (MAUT)

The calculation analysis implemented in this project is to provide users a better decision-making tool by indicating which houses would have better advantages to be selected. This calculation is based on the Multi-attribute Utility Theory (MAUT). The prototype system allows the user to place a rating on the frequent-visit-location

based on their frequency of travel. The prototype system would then use the rating entered by the user to analyze which location is more frequently traveled by the user. The distance between an accommodation and the frequent-visit-location is obtained from the Google Distance Matrix API from Google Map Services. With the distance obtained, the proposed system could calculate and provide a number that indicates the weighted average of the distances between a particular accommodation and the various frequent-visit-locations entered by the user. The weighted average or weight for the particular accommodation is then shown as a rating to the user.

$$Weight(X) = \frac{Location\ A \times ((highest\ rating + 1) - ratingA)** + Location\ B \times ((highest\ rating + 1) - ratingB) + Location\ N \times (ratingN)}{(ratingA + ratingB \dots + ratingN)}$$

\*\*Rating representing most frequent visit is 1, therefore require to reverse the number when calculate the weight by highest rating plus 1 and then minus the rating.

**Figure 4-7-F1 Formula used in calculation of MAUT in this system**

The above calculation is to calculate the weight for each of the available houses so that the house that has the lowest weight can be obtained. The accommodation that has the lowest weight or rating indicates that it could be the best accommodation available for the user.

```

//Distance Data
var rateCounter = 0;
var rateTotalCounter = 0;

for( var k = i; k < eachDisElementArray.length ; k)
{
    var distances = eachDisElementArray[k];
    var distanceNumber = distances.split("km"); // 1st [0] is the number
    if( i == (highLightNumber))
    {
        bodyTag+=" " + distances + "</TD>"     }else     {         bodyTag+=" Calculate each of the frequent-visit-location distance  MAUT formula | |
```

**Figure 4-7-F2 The back-end calculation for MAUT.**

The back-end coding to evaluate on each accommodation to be displayed to the user in the result table was shown in Figure 4-7-F2. It will calculate the weight on every frequent-visit-location and display the result to the user, according to the respective weights as ratings; thus potentially allowing the user to select the best accommodation based on the lowest weight or rating.

## Chapter 5: Testing

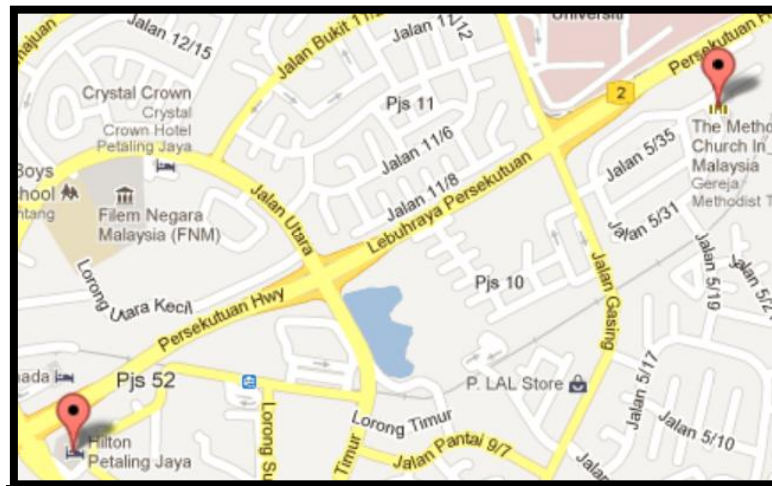
The system that is developed in this project was a website-based accommodation search system. Therefore the website will be used by users on the world wide web, in general. Testing is essential for a project because testing could provide vital information and useful insight on the usability of the system, error checking, so that improvement could be done based on the test cases developed to make the system more robust and solid. Hence, for this project several test cases were conducted based on several typical common scenarios to check whether the system was performing as it should or the other way around. The test cases are all based on the assumption related to real life scenarios.

### 5.1. Scenario 1 - Rated versus Non-rated Result on two frequent-visit-location

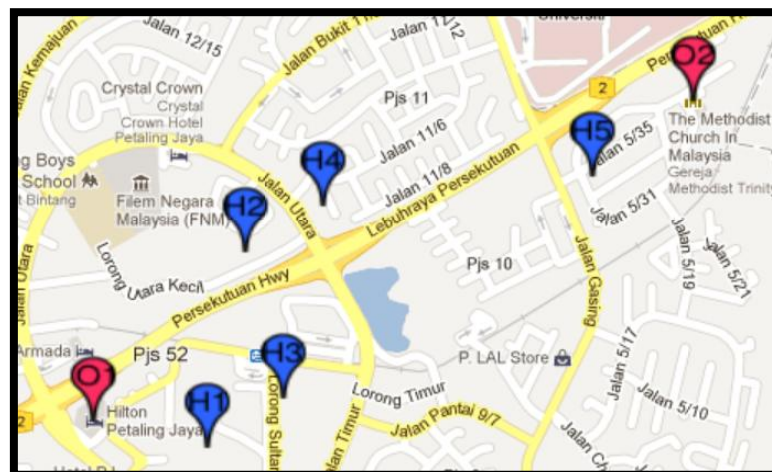
In this scenario, a person named **Steven** wish to search for accommodation after he got his position as a Hotel Manager inside Hilton Hotel located in Petaling Jaya. Steven's hometown is in **Johor** and Steven has rarely been to Selangor especially the Petaling Jaya area. Steven goes to church every Sunday for prayer and activities. The church which he know of was Methodist Church located in Petaling Jaya because he went there twice before when he visited his friend previously in Selangor.

Based on this assumption, the process of testing was conducted, and the parameters used in the testing process were the two addresses: **Hilton Hotel (Work Place)** and **Methodist Church**. It was assumed that Steven was required to travel to work place on a nearly **daily basis for the five working days at least**, and he

required to attend church prayer and activities on Sunday; thus the frequency of visits to these two locations would be such that the **work place would have a higher travel frequency compared to the church, in a week. Based on these criteria, a result was generated using the developed system.**



**Figure 5-1-F1 Google map with 2 frequent-visit-locations**



**Figure 5-1-F2 Result of search plotted on Google Map.**

Rated Result			
Result Table	Frequent Visit Location1 Hilton Petaling Jaya Hotel, Jalan Barat, Petaling Jaya, Selangor, Malaysia	Frequent Visit Location2 The Methodist Church In Malaysia, Trinity Methodist Church, Petaling Jaya, Malaysia	Rating Weight
Accommodation Available 4	1.7 km	1.8 km	1.73

Non-rated Result			
Result Table	Frequent Visit Location1 Hilton Petaling Jaya Hotel, Jalan Barat, Petaling Jaya, Selangor, Malaysia	Frequent Visit Location2 The Methodist Church In Malaysia, Trinity Methodist Church, Petaling Jaya, Malaysia	Average Distance
Accommodation Available 5	2.7 km	0.6 km	1.65 km

**Figure 5-1-F3 Partial Result of the search for rated and non-rated cases.**

Based on the result obtained from the system, with and without the rating function, on the two search addresses: Hilton Hotel and Methodist Church, a result table is generated. The partial result was captured and displayed as shown in Figure 5-1-F3. The result shown on top of Figure 5-1-F3 was the one of which the rating function was selected based on the scenario assumption where Steven would have a higher travel frequency to his work place for five days a week and thus the most frequently visited place goes to his work place and the 2nd most frequently visited location would be the Methodist church. The bottom part of Figure 5-1-F3 is the

result table without rating function being selected.

Figure 5-1-F3 shows that the system suggested different accommodation for the rating and without rating cases. Result details showed that the system recommended Accommodation No.4 with distance to the Hilton Hotel was 1.7 km and 1.8 km to the Methodist church for the rated search whereas the system recommended Accommodation No.5 to Steven for non-rated search with 2.7 km to the Hilton Hotel and 0.6 km to the Methodist church.

In order to distinguish the difference between rated and non-rated testing, a possible real life case was brought into this scenario to calculate and highlight the difference.

Calculation for travel distance for a month on suggested accommodation.

**Rated Accommodation Details - Accommodation No. 4**

Travel Distance : Hilton Hotel - 1.7 km  
 : Methodist Church - 1.8 km  
 Total Travel Distance : 3.5 km

**Travel Distance**

$1.7 \text{ km} \times 10$ (5 working day two way travel)	=	17 km / week
$+ 1.8 \text{ km} \times 2$ (Sunday only : two way travel)	=	3.6 km / week
<hr/>		
<b>Total Travel Distance per week</b>	<b>=</b>	<b>20.6 km</b>
<b>Total Travel Distance per month</b>	<b>=</b>	<b>82.4 km</b>



**Non-rated Accommodation Details - Accommodation No.5**

Travel Distance : Hilton Hotel - 2.7 km  
: Methodist Church - 0.6 km  
Total Travel Distance : 3.3 km

**Travel Distance**

$$\begin{aligned} 2.7 \text{ km} \times 10 \text{ (5 working day two way travel)} &= 27 \text{ km / week} \\ + 0.6 \text{ km} \times 2 \text{ (Sunday only : two way travel)} &= 1.2 \text{ km / week} \\ \hline \text{Total Travel Distance per week} &= \mathbf{28.2 \text{ km}} \\ \text{Total Travel Distance per month} &= \mathbf{112.8 \text{ km}} \end{aligned}$$

Based on the travel distance calculations for both the rated and non-rated accommodation results, it is shown that when the rating was selected, the system could suggest an accommodation that would save Steven 30.4 km of travel distance per month compared to the non-rated search result. If Steven selected the non-rated accommodation, it would possibly cause him to pay more on petrol fees or taxi fares to travel to these two places, since he had to travel 30.4 km more per month and in the long term, it would become 364.8 km a year! Steven would save 26.95% in his monthly travel distance if he chose the rated accommodation suggested by the proposed system. Therefore the rated accommodation result would best suit Steven in this scenario.

### **5.2. Scenario 2 - Rated versus Non-rated Result on three frequent-visit-location**

Assumptions were made in this scenario for a family who wants to move to Kelana Jaya to stay in a better environment and also because James, the head of the family, has found new job in the Kelana Jaya Furniture Mall. The family is concerned about the daily expenses if they are moving to a new place because they have a tight budget. The family members include James the father, Jane the housewife, and Jerry who still studying and the parents plan to enroll Jerry to a school nearby in Kelana Jaya called SMK Kelana Jaya.

The daily routine for James' family on based on their current house that they are staying is that James will fetch Jerry to go to school at 6 in the morning and return back to house and eat breakfast with Jane. On every two-day basis, James will drive Jane to the Giant Hypermarket to shop for ingredients for cooking. Jane insists to go on a two-day basis to make sure that the cooking ingredients will be fresh and thus she can cook a healthy meal for the family. James will go to work at 9 a.m and be back at 6 p.m on weekdays. This is the current daily routine for James' family. Since the family wants to move to Kelana Jaya, James wants to estimate the petrol expenses that the family may require to spend on a monthly basis. James assumes the petrol price per liter to be 1.90 Ringgit Malaysia per liter for the RON 95 petrol and 10 km will use up 1 liter of fuel.



Figure 5-2-F1 Google map with 3 frequent-visit-locations

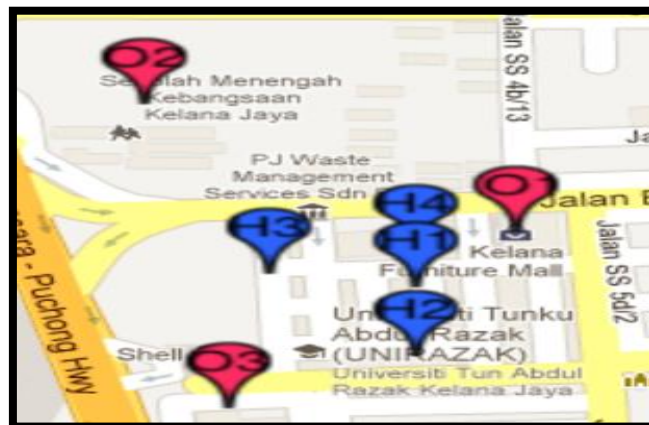


Figure 5-2-F2 Result of search plotted on Google Map.

Rated Result				
Result Table	Frequent Visit Location1 Kelana Furniture Mall, Selangor, Malaysia	Frequent Visit Location2 SMK KELANA JAYA, Smk Kelana Jaya Jalan Bahagia, Petaling Jaya, Kelana Jaya, Malaysia	Frequent Visit Location3 Giant Hypermarket Kelana Jaya, 33 Jalan SS 6/12, Ss 6, Petaling Jaya, Selangor, Malaysia	Rating Weight
Accommodation Available 4	0.3 km	0.5 km	1.0 km	0.48

Non-rated Result				
Result Table	Frequent Visit Location1 Kelana Furniture Mall, Selangor, Malaysia	Frequent Visit Location2 SMK KELANA JAYA, Smk Kelana Jaya Jalan Bahagia, Petaling Jaya, Kelana Jaya, Malaysia	Frequent Visit Location3 Giant Hypermarket Kelana Jaya, 33 Jalan SS 6/12, Ss 6, Petaling Jaya, Selangor, Malaysia	Average Distance
Accommodation Available 3	0.4 km	0.5 km	0.8 km	0.57 km

**Figure 5-2-F3 Partial Result of the search for rated and non-rated cases.**

Results were obtained based on searches with the rating function and without the rating function using the same search addresses: Kelana Jaya Furniture Mall, SMK Kelana Jaya and Kelana Giant Hypermarket. The partial result was captured and displayed as in the Figure 5-2-F3. The result shown in top of Figure 5-2-F3 was with rating involved as in the scenario assumption, in which James will travel to the work place most frequently. The second most frequent-visit-location will be the school and lastly the hypermarket.

The total number of travel trips per week for James was assumed to be 10 trips

for the work place (5 days; to go to work and back), 5 trips for sending Jerry to school (only for sending him to school in the morning), and 6 trips for the Hypermarket (to drive Jane to the hypermarket and back) .

Figure 5-2-F3 shows that the system suggested different accommodation for the rating and without rating cases. Result details showed that the system recommended Accommodation No.4 with distance to Work Place was 0.3 km, 0.5 km to school and 1.0 km to hypermarket for the rated search whereas the system recommended Accommodation No.3 to James for the non-rated search with 0.4 km to work place, 0.5 km to school and 0.8 km to hypermarket.

In order to differentiate the difference between the rated and non-rated testing, James' daily routine was used in the calculation to estimate the petrol expenses for James.

Calculation for travel distance for a month on suggested accommodation

Rated Accommodation Details - Accommodation No.4

Travel Distance : Work Place - 0.3 km  
 : School - 0.5 km  
 : Hypermarket - 1.0 km  
 Total Travel Distance : 1.8 km

Travel Distance

0.3 km × 10 ( 5 working day two way travel)	= 3 km / week
0.5 km × 10 (5 days - two way travel)	= 5 km / week
+ 1.0 km × 2 (once a week )	= 2 km / week
<b>Total Travel Distance per week</b>	<b>= 10 km</b>
<b>Total Travel Distance per month</b>	<b>= 40 km</b>

Petrol Price rated

40 km ÷ 10 km = 4 liter petrol  
 4liter × RM 1.90 = RM 7.60 / month

Rated Accommodation Details - Accommodation No.3

Travel Distance : Work Place - 0.4 km  
 : School - 0.5 km  
 : Hypermarket - 0.8 km  
 Total Travel Distance : 1.7 km

Travel Distance

0.4 km × 10 ( 5 working day two way travel)	= 4 km / week
0.5 km × 10 (5 days - two way travel)	= 5 km / week
+ 0.8 km × 2 (once a week )	= 1.6 km / week
<b>Total Travel Distance per week</b>	<b>= 10.6 km</b>
<b>Total Travel Distance per month</b>	<b>= 42.4 km</b>

Petrol Price rated

42.4 km ÷ 10 km = 4.24 liter petrol  
 4.24 liter × RM 1.90 = RM 8.056 / month

Based on the travel distance calculations calculated for both the rated and non-rated accommodation results, it is shown that with the rating function, the system could suggest an accommodation that will save James a total of 2.4 km travel distance per month as compared to the non-rated search result. The total petrol usage for a month with James' family staying in the respective new accommodation using the rating function as suggested by system was RM 7.60 and without rating was RM 8.056. Therefore James could save up to RM 0.456 per month, which is 5.66% of his petrol expenses per month if he chose the new accommodation using the rating function. Within a year James could save up to RM 5.472.

### **5.3. Scenario 3 - Rated versus Non-rated Result on three frequent-visit-location**

Assumptions are made in this situation for Steffy, who wants to move to a location near the her work place in the Brunsfield Company in Petaling Jaya. Steffy often travels to look for her friend Tony to have lunch or dinner together at the One Utama Shopping Centre in Petaling Jaya and she will go to the Tropicana City Mall every weekend. Steffy wants to find an accommodation that would best fit to her current situation as she likes to incorporate the activities that she wants to do, into her plan of finding a new house to stay in.



Figure 5-3-F1 Google map with 3 frequent-visit-locations

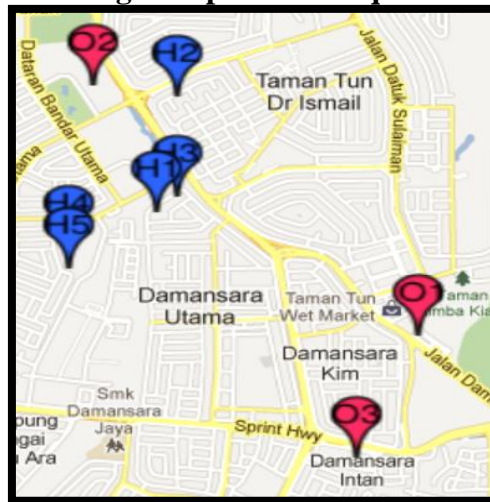


Figure 5.3.2 Result of search plotted on Google Map.



Rated Result				
Result Table	Frequent Visit Location1 Brunsfield International Group, 8, Jalan Wan Kadir 4, Off Jalan Damansara, Taman Tun Dr. Ismail, Wilayah Persekutuan, Federal Territory of Kuala Lumpur, Malaysia	Frequent Visit Location2 1 Utama Shopping Centre, 1 Utama Shopping Centre, 1 Lebuhr Bandar Utama, Petaling Jaya, Bandar Utama, Malaysia	Frequent Visit Location3 Tropicana City Mall, Tropicana City Mall, 3 Jalan SS20/27, 47400 P.J. Malaysia	Rating Weight
Accommodation Available 3	2.4 km	2.3 km	2.7 km	2.42

Non-rated Result				
Result Table	Frequent Visit Location1 Brunsfield International Group, 8, Jalan Wan Kadir 4, Off Jalan Damansara, Taman Tun Dr. Ismail, Wilayah Persekutuan, Federal Territory of Kuala Lumpur, Malaysia	Frequent Visit Location2 1 Utama Shopping Centre, 1 Utama Shopping Centre, 1 Lebuhr Bandar Utama, Petaling Jaya, Bandar Utama, Malaysia	Frequent Visit Location3 Tropicana City Mall, Tropicana City Mall, 3 Jalan SS20/27, 47400 P.J. Malaysia	Average Distance
Accommodation Available 3	2.4 km	2.3 km	2.7 km	2.47 km

**Figure 5.3.3 Partial Result of the search for rated and non-rated cases.**

Results were obtained from the search with rating and without rating cases based on the same search addresses: Brunsfield Company, One Utama Shopping Centre and Tropicana City Mall. The partial result was captured and displayed in Figure 5.3.3. The result shown on the top of Figure 5.3.3 was with the rating function involved as described in the scenario assumptions in which Steffy travels to her work place more often compare to the One Utama Shopping Centre because she does not go to the One Utama Shopping Centre to meet Tony every day and only go to the Tropicana City Mall on weekends.

Figure 5.2.3 showed that system suggested different accommodation for the

rating and without rating cases. Result details showed that the system recommended Accommodation No.3 with travel distance to the Work Place as 2.4 km, travel distance of 2.3 km to the One Utama Shopping Centre, and lastly 2.7 km of travel distance to Tropicana City Mall. The system displayed Accommodation No.3 to Steffy for the rated search as well for the non-rated search

Calculation for travel distance on suggested accommodation.

**Rated and Non-rated Accommodation Details - Accommodation No.3**

Travel Distance : Brunsfield Company - 2.4 km  
 : One Utama Shopping Centre - 2.3 km  
 : Tropicana City Mall - 2.7 km  
 Total Travel Distance : 7.4 km

Travel Distance

2.4 km × 10 ( 5 working day two way travel)	=	24 km / week
2.3 km × 6 ( assumption on 3 days every week for two way travel)	=	13.8 km / week
+ 2.7 × 4 ( weekend Saturday and Sunday for two way travel)	=	<u>10.8 km / week</u>
<b>Total Distance Per week</b>	=	<b>48.6 km</b>
<b>Total Distance per month</b>	=	<b>194.4 km</b>

Based on the accommodation result displayed by the system, both the rated and non-rated result showed Accommodation No.3, which means that Accommodation No.3 is the best choice for both cases; even for the non-rated case. Therefore the best choice will be displayed in rated or not-rated result in this scenario case. Therefore Accommodation No.3 is the best recommended suggestion to Steffy to fulfill her weekly travel routine.

#### 5.4 Scenario 4 - Multiple Point Traveling method

In this scenario, Kelvin who is working as a salesman wants to find an accommodation that will reduce his travel distance compare to the current place he is staying, which is in Shah Alam and thus reduce his petrol expenses. Kelvin needs to travel to the Pearl Point Hotel, Mid Valley, Taman Danau Desa and Kuchai Entrepreneurs Park daily to meet up with his customers and to build rapport with them. Therefore he hopes to find an accommodation with a shorter travel distance to travel to these multiple places.

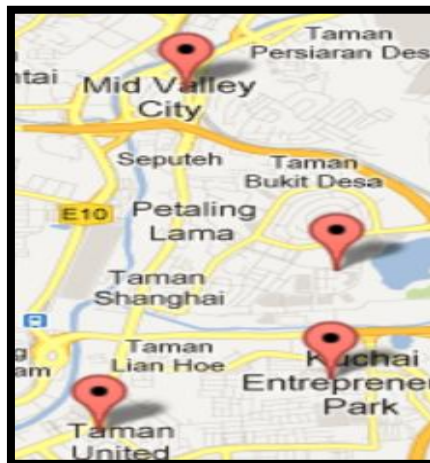


Figure 5-4-F1 Google map with 2 frequent-visit-locations



Figure 5-4-F2 Result of search plotted on Google Map.

Rated Result - System Select					
Result Table	Frequent Visit Location1 Pearl Point, Pearl Point Jalan Klang Lama, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia	Frequent Visit Location2 Mid Valley, Mid Valley, Mid Valley City, 58200 Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia	Frequent Visit Location3 Taman Danau Desa, Taman Danau Desa, 55100 Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia	Frequent Visit Location4 Kuchai Entrepreneurs Park, Jalan 1/116b, Kuala Lumpur, Malaysia	Rating Weight
Accommodation Available 5	1.9 km	7.4 km	2.7 km	2.1 km	3.73

Rated Result - Self Select					
Result Table	Frequent Visit Location1 Pearl Point, Pearl Point Jalan Klang Lama, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia	Frequent Visit Location2 Mid Valley, Mid Valley, Mid Valley City, 58200 Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia	Frequent Visit Location3 Taman Danau Desa, Taman Danau Desa, 55100 Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia	Frequent Visit Location4 Kuchai Entrepreneurs Park, Jalan 1/116b, Kuala Lumpur, Malaysia	Rating Weight
Accommodation Available 4	1.6 km	9.5 km	3.9 km	1.3 km	4.40

Figure 5-4-F3 Partial Result of the search.

The search results were obtained from the search with rating for two accommodation – system selection and user selection, on the same search addresses: the Pearl Point Hotel, Mid Valley, Taman Danau Desa, and Kuchai Entrepreneurs Park. The partial result was captured and displayed as in the Figure 5-2-F3. The result shown on the top of Figure 5-4-F3 was with rating involved as according to the scenario description.

Figure 5-4-F3 shows that the system suggested Accommodation No.5. Result details shown that the system recommended Accommodation No.5 with distance to the Pearl Point Hotel was 1.9 km; a distance of 7.4 km to Mid Valley, 2.7 km to Taman Danau Desa, and lastly 2.1 km to Kuchai Entrepreneurs Park.

However, in this scenario, Accommodation No.4 (Self-picked) was used to compare with the Accommodation No.5 (System-selected) based on the travel sequence to multiple locations. The sequence of travel starts in the following order; from the Pearl Point Hotel to Mid Valley, from Mid Valley to Taman Danau Desa, and lastly from Taman Danau Desa to Kuchai Entrepreneurs Park . The calculation for the various travel distances is shown below.

Calculation for travel distance for a month on suggested accommodation.

Rated Accommodation Details - Accommodation No.5 (System)

Travel Distance	: A. Pearl Point	- 1.9 km
	: B. Mid Valley	- 7.4 km
	: C. Taman Danau Desa	- 2.7 km
	: D. Kuchai Entrepreneurs Park	- 2.1 km

Travel Distance of frequent-visit-location using Google Map Distance Calculate

Travel Distance	: A to B = 6.2 km
	: B to C = 7.8 km
	: C to D = 4.7 km

Total Travel Distance per trip from Accommodation No.5 to A,B,C,D and back to Accommodation No.5

Travel Distance	: No.5 to Pearl Point	1.9 km
	: Pearl Point to Mid Valley	6.2 km
	: Mid Valley to Taman Danau Desa	7.8 km
	: Taman Danau Desa to Kuchai Entrepreneurs Park	4.7 km
	: Kuchai Entrepreneurs Park to No.5	2.1 km

---

Total Travel Distance per day = 22.7 km

Rated Accommodation Details - Accommodation No.4 (Self-Pick)

Travel Distance	: A. Pearl Point	- 1.6 km
	: B. Mid Valley	- 9.5 km
	: C. Taman Danau Desa	- 3.9 km
	: D. Kuchai Entrepreneurs Park	- 1.3 km

Travel Distance of frequent-visit-location using Google Map Distance Calculate

Travel Distance	: A to B = 6.2 km
	: B to C = 7.8 km
	: C to D = 4.7 km

Total Travel Distance per trip from Accommodation No.5 to A,B,C,D and back to Accommodation No.5

Travel Distance	: No.5 to Pearl Point	1.6 km
	: Pearl Point to Mid Valley	6.2 km
	: Mid Valley to Taman Danau Desa	7.8 km
	: Taman Danau Desa to Kuchai Entrepreneurs Park	4.7 km
	: Kuchai Entrepreneurs Park to No.5	1.3 km

---

Total Travel Distance per day = 21.6 km

Based on the calculation above, the result showed that the self-picked accommodation would give a shorter travel distance when compared to the case where the system selected an accommodation based on the rating function. This was due to the fact that in this situation, the travel pattern flow was different from the other scenarios. In his travel sequence, the only parts of the journey that include his accommodation location are when Kelvin travels from his accommodation to the Pearl Point Hotel and back from Kuchai Entrepreneurs Park to his accommodation. Therefore the distances for the other frequent-visit-locations, that is from the Pearl Point Hotel to Mid Valley, from Mid Valley to Taman Danau Desa and from Taman Danau Desa to Kuchai Entrepreneurs Park would be the same for both the system-selected and self-picked cases. Hence, the travel distances that involve his accommodation (from his accommodation to the Pearl Point Hotel, and from Kuchai Entrepreneurs Park to his accommodation) would be the determining factor in finding which accommodation would have a shorter total travel distance for this unique conditional travel pattern.

The calculation shows that Accommodation No.4 (Self-picked) is having a shorter total travel distance of 21.6 km compared to 22.7 km for Accommodation No.5 (System-selected). The shorter total travel distance for Accommodation No. 4 is because the travel distances from Accommodation No. 4 to the Pearl Point Hotel and Accommodation No. 4 to Kuchai Entrepreneurs Park are shorter compare to that of Accommodation No.5 to these two places. Accommodation No.4 (Self-picked) reduces the travel distance by 4.84% as compared to Accommodation No.5 (System-selected). In this scenario, the system would not know that the user's travel pattern includes travelling from one frequent-visit-location to another frequent-visit-location in sequence rather than starting from the accommodation

location to one particular frequent-visit-location and then back to the accommodation.

As to summarize the test cases, the system have could have performed better if the travel sequence is not important to the user and only the frequency of travel, to and from the accommodation to the frequent-visit-location, is considered. Therefore, based on the few test cases conducted and described in the section, the system works best if the frequencies of travel of the different places are not similar to each other, for example, travelling daily to Location A while travelling once a week or once a month to Location B. However, if both locations A and B have the same user travel frequency and the sequence of travel is important, the system may be unable to determine the best accommodation for the user. Although the system is unable to provide the best accommodation information to the user, the user could also select other accommodation based on his or her own choice from the result table without selecting the proposed accommodation recommended by the system.



## CHAPTER 6: CONCLUSION

In this project, a prototype of an accommodation search engine based on frequent-visit-location using Multi-Attribute Utility Theory, with its main target users from other states in Malaysia or other countries, was developed to provide more relevant accommodation search results for the majority of users by allowing the user to input information by specifying the frequent-visit-location(s).

A rating feature that allows the system to summarize which accommodation is suitable for the user was incorporated into the prototype search system. The proof-of-concept system demonstrated that the specific location search could provide a smaller scope and more relevant searches compared to area-based search and also highlight the importance of having a rating features to provide a better decision making guidance for the user.

The search results provide useful information to the user such as travel distance from the accommodation to the frequent-visit-location. The best accommodation will be highlighted by the lowest value of rating weightage generated by the search system. Therefore, the user could reduce his or her expenses such as petrol costs for travelling. The proposed system was not just an accommodation search engine but also it could help users in their expenses planning.

Since the system was a proof-of-concept system; therefore the system was not a fully functional system yet as it requires more development on others areas such are administration and landlord areas.

One of the technologies implemented in this system, the Google Web Service, has a user limitation in which the Google Web Service only allows a query of 100 elements per day with the Google API key (per Google user). Therefore future work could replace the Google Web Service with other Map Services to produce a functional system, such as the Bing Map, Yahoo Map or any other map services that have lesser restrictions or more features. Apart than that, there are many other theories, other than the Multi-Attribute Utility Theory, on calculating the importance of frequent-visit-location(s) near the accommodation, which could be used to improve on the number of relevant search results shown to the user.

Lastly, the test cases, which had been presented previously, showed that the search results might not relevant if the user has a unique conditional travel pattern. For example, if the user travels from the accommodation to point A, and continues his or her journey to point B and lastly back to the accommodation, the travel sequence would result in a loop condition called as multiple point travel condition. The system would not be able to provide the best recommended accommodation for the user if that is what the user intends to do. Hence, in order for the search system to perform better, this issue should be further addressed in the future to enhance the capability of the system.

## REFERENCES

### REFERENCES

Fico.com (n.d.) *Utility Theory*. [online] Available at: <http://www.fico.com/en/Communities/Analytic-Technologies/Pages/UtilityTheory.aspx> [Accessed: 25 Mar 2013].

Graham, I., 1998. *Introduction to HTML*. [online] Available at: [http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/html\\_intro.html](http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/html_intro.html) [Accessed: 21 Jul 2012].

Hotpads.com, 2012. *HotPads - Map Search for Real Estate, Apartments and Houses for Rent, Foreclosures and Homes for Sale*. [online] Available at: <http://hotpads.com/> [Accessed: 20 Jul 2012].

Houselocator.com, 2012. *Homes For Sale, For Sale By Owner, FSBO*. [online] Available at: <http://www.houselocator.com/> [Accessed: 20 Jul 2012].

Ianswer4u.com, n.d.. *Spiral Model : Advantages and Disadvantages*. [online] Available at: <http://www.ianswer4u.com/2011/12/spiral-model-advantages-and.html#axzz21NEnXg77> [Accessed: 22 Jul 2012].

Ibilik.my, n.d.. *Find Room For Rent / Room To Let / Homestay / Short Term Rental / Sublet in Malaysia, Kuala Lumpur, Ampang, Bangsar, Cheras, Setapak, Damansara, Petaling Jaya, Subang Jaya, and Penang | iBilik (formerly known as CariBilik)*. [online] Available at: <http://www.ibilik.my/> [Accessed: 20 Jul 2012].

Iproperty.com.my, 2012. *Malaysia Property and Real Estate for Sale and Rent*. [online] Available at: <http://www.iproperty.com.my/> [Accessed: 20 Jul 2012].

Kyrnin, J. (n.d.) *What is Ajax?*. [online] Available at: <http://webdesign.about.com/od/ajax/a/aa>

## REFERENCES

101705.htm [Accessed: 25 Mar 2013].

Quirksmode.org , 2008. *Javascript - General Introduction*. [online] Available at: <http://www.quirksmode.org/js/intro.html> [Accessed: 21 Jul 2012].

Realtor.com , 2012. *Find Real Estate, Homes for Sale & Houses for Rent on REALTOR.com - Realtor.com®*. [online] Available at: <http://www.realtor.com/> [Accessed: 20 Jul 2012].

Room4rent.my , 2012. *Room for Rent, Room to Let, Room for Lease / room4rent.my*. [online] Available at: <https://www.room4rent.my/> [Accessed: 20 Jul 2012].

Roseindia.net , 2008. *What is MySQL*. [online] Available at: <http://www.roseindia.net/mysql/mysql5/what-is-mysql.shtml> [Accessed: 22 Jul 2012].

Techterms.com (2013) *jQuery Definition*. [online] Available at: <http://www.techterms.com/definition/jquery> [Accessed: 25 Mar 2013].

Thinkproperty.com.my , 2012. *Malaysia property, real estate, apartments, houses for sale/rental*. [online] Available at: <http://thinkproperty.com.my/realestate/> [Accessed: 20 Jul 2012].

Trulia.com , 2012. *Trulia - Real Estate, Homes for Sale, Apartments for Rent, Local data*. [online] Available at: <http://www.trulia.com/> [Accessed: 20 Jul 2012].

Webopedia.com , n.d.. *What is PHP? - A Word Definition From the Webopedia Computer Dictionary*. [online] Available at: <http://www.webopedia.com/TERM/P/PHP.html> [Accessed: 21 Jul 2012].

**APPENDIX A: SCREENSHOT OF RELATED WEBSITES**



Figure A-1: iProperty Main Page



Figure A-2: thinkproperty Main Page



Figure A-3: room4rent Main Page



Figure A-4: iBilik Main Page

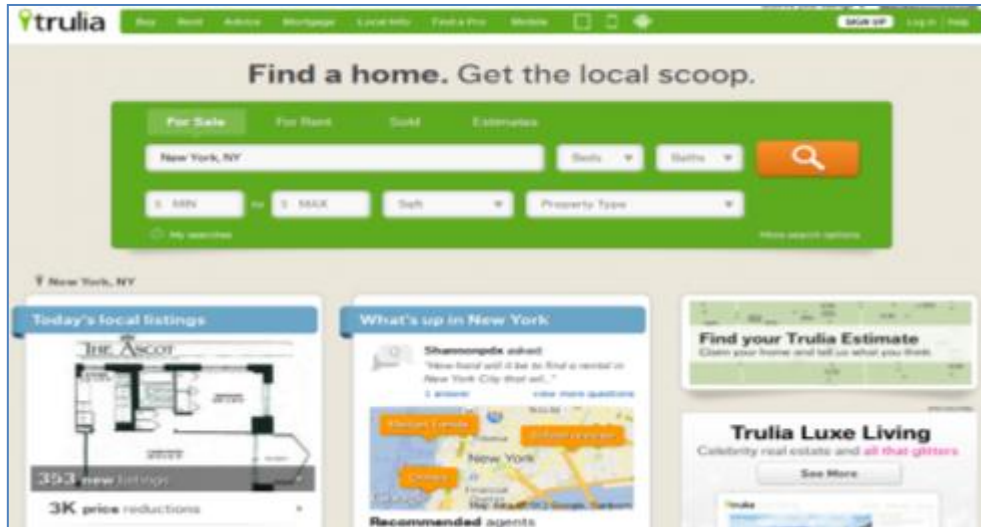


Figure A-5: Trulia Main Page



Figure A-6: Hotpads Main Page



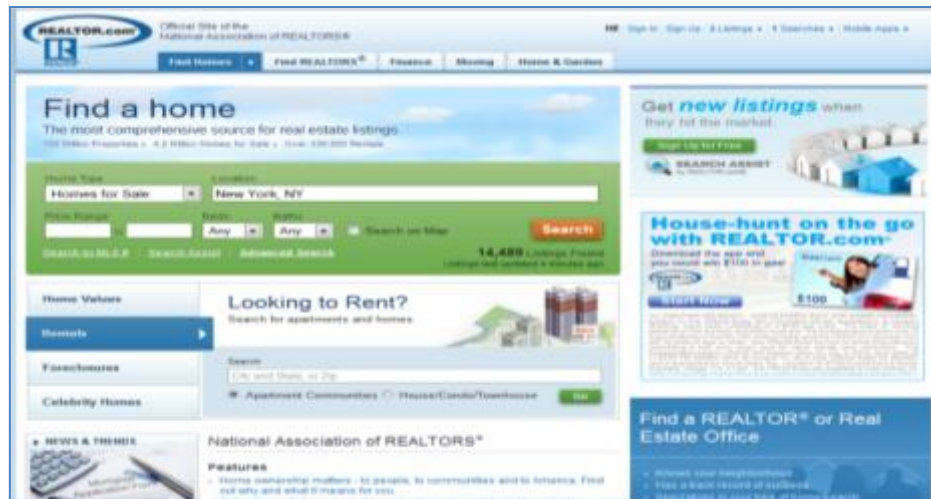


Figure A-7: Realtor Main Page

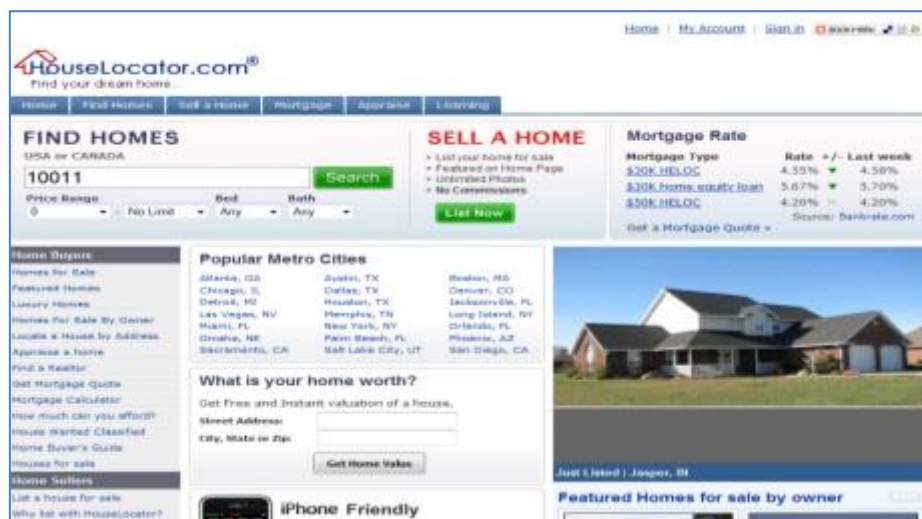


Figure A-8: House Locator