

**Mobile Tour Assistant for Malaysia**

by

NGUYEN THI CAM LUONG

Dissertation submitted in partial fulfillment of

the requirements for the

Bachelor of Technology (Hons)

(Business Information System)

NOVEMBER 2007

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

**CERTIFICATION OF APPROVAL**

**Mobile Tour Assistant for Malaysia**

by

Nguyen Thi Cam Luong

A project dissertation submitted to the

Computer Information Systems

Universiti Teknologi PETRONAS

in partial fulfilment of the requirement for the

BACHELOR OF TECHNOLOGY (Hons)

(BUSINESS INFORMATION SYSTEM)

Approved by,

A handwritten signature in black ink, appearing to read 'Low Tan Jung', written over a horizontal line.

Low Tan Jung)

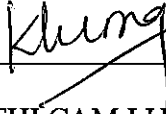
UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

November 2007

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



---

NGUYEN THI CAM LUONG

## **ABSTRACT**

Malaysia's tourism industry has seen a rapid growth rate for the past years, and played a significant role in the development of the country's economy. 2007 is a special year for Malaysian tourism since it is called the "Visit Malaysia" year in celebration of 50 years of Independence. Information Technology has made remarkable contributions to the growth of tourism industry in Malaysia. Its efforts to promote tourism range from websites providing information to systems to service tourists. However, while countries with well-developed tourism industry like France and Hong Kong have developed comprehensive tour assistant packages for tourists, Malaysia is yet to do so. Thus, a comprehensive mobile tour package is needed for Malaysia.

The objective of the project is to develop a well-designed tour assistant running on PDAs which makes use of current trends in mobile applications such as the adoption of AI technique in scheduling. Due to time and resource constraints, the project focuses on areas within and around Kuala Lumpur and targets on short-stay and transit tourists.

Going through a thorough literature review and following the waterfall methodology, the project has successfully developed a full package of mobile tour assistant including the back end and front end. The project also makes contribution to creatively applying Genetic Algorithm, an AI technique in plan auto-scheduling. The author hopes that the package will be of high commercial value and contribute significantly to boosting Malaysia's tourism.

## **ACKNOWLEDGEMENT**

First and foremost, I would like to express my deepest gratitude to my supervisor, Mr. Low Tan Jung for his helpful guidance and constructive feedbacks throughout my project endeavor. His thoughtful comments have been the guideline to help me have a better vision of the system and motivate me to continue enhancing it.

I would also like to thank my family, friends and beloved ones for their great support and encouragement to help me complete the project to the best of my ability. Without them, I might not have been able to achieve as much as I have for the project like today. I have learnt to understand that distance means nothing when sincere care from the beloved is felt all around.

My gratitude also goes to local and international friends who have given me valuable inputs for the system and helped me gain necessary background knowledge to develop the project.

## TABLE OF CONTENTS

|  |     |
|--|-----|
| ABSTRACT.....                                    | i   |
| ACKNOWLEDGEMENT.....                             | i   |
| TABLE OF CONTENTS.....                           | iii |
| LIST OF FIGURES.....                             | v   |
| LIST OF TABLES.....                              | vi  |
| ABBREVIATIONS AND NOMENCLATURES.....             | vii |
| INTRODUCTION.....                                | 1   |
| 1 BACKGROUND OF THE SYSTEM.....                  | 1   |
| 2 PROBLEM STATEMENT.....                         | 2   |
| 3 OBJECTIVES AND SIGNIFICANCE OF THE SYSTEM..... | 2   |
| 4 SCOPE OF THE SYSTEM.....                       | 3   |
| LITERATURE REVIEW.....                           | 5   |
| 1 TOURISM AND INFORMATION TECHNOLOGY.....        | 5   |
| 2 AI TECHNIQUE IN MOBILE APPLICATION.....        | 7   |
| 3 DATABASE AND MOBILE APPLICATIONS.....          | 9   |
| METHODOLOGY.....                                 | 11  |
| 1 SOFTWARE MODEL.....                            | 11  |
| 2 SYSTEM ANALYSIS.....                           | 12  |
| 3 SYSTEM DESIGN.....                             | 15  |
| 4 SYSTEM DEVELOPMENT.....                        | 19  |

|                                    |    |
|------------------------------------|----|
| RESULTS AND DISCUSSION .....       | 47 |
| 1 TESTING .....                    | 47 |
| 2 DISCUSSION .....                 | 50 |
| CONCLUSION AND RECOMMENDATION..... | 51 |
| 1 CONCLUSION .....                 | 51 |
| 2 RECOMMENDATIONS .....            | 51 |
| REFERENCES .....                   | 53 |
| APPENDICES .....                   | 56 |

## LIST OF FIGURES

|   |    |
|---|----|
| Figure 2.1: MVC model.....                              | 6  |
| Figure 2.2 Generic Algorithm's pseudo-code.....         | 8  |
| Figure 3.1: Waterfall model.....                        | 11 |
| Figure 3.2 Use case Diagram.....                        | 12 |
| Figure 3.4: System design.....                          | 16 |
| Figure 3.5: Sybase's interface.....                     | 18 |
| Figure 4.1: Login Form for Backend Module.....          | 19 |
| Figure 4.2: Tourism synchronization model.....          | 21 |
| Figure 4.3: Synchronization interface.....              | 22 |
| Figure 4.4: Attraction Manager.....                     | 23 |
| Figure 4.5: Event Manager.....                          | 24 |
| Figure 4.6: Map Manager.....                            | 25 |
| Figure 4.7: Hotel Manager.....                          | 26 |
| Figure 4.8: Contact Manager.....                        | 27 |
| Figure 4.9: Activity diagram for itinerary planner..... | 29 |
| Figure 4.10: Itinerary planner's interface.....         | 31 |
| Figure 4.12: Preference selection.....                  | 34 |
| Figure 4.13: Tourist Kits.....                          | 38 |
| Figure 4.14: Tourist Attractions' list.....             | 39 |
| Figure 4.15: Tourist attraction's display.....          | 39 |
| Figure 4.16: View Map.....                              | 40 |



|  |    |
|--|----|
| Figure 4.17: Events & Festivals.....               | 41 |
| Figure 4.18: Useful Information .....              | 42 |
| Figure 4.19: Activity diagram of hotel finder..... | 43 |
| Figure 4.20: Find hotels .....                     | 44 |

### **LIST OF TABLES**

|  |    |
|--|----|
| Table 4.11: Major tables for Itinerary Planner module..... | 32 |
|--|----|

## **ABBREVIATIONS AND NOMENCLATURES**

AI: Artificial Intelligence

GA: Genetic Algorithm

PDA: Personal Digital Assistant

IT: Information Technology

MVC: Model-View-Controller

DBMS: Database Management System

# CHAPTER 1

## INTRODUCTION

### 1 BACKGROUND OF THE SYSTEM

Tourism has played an important role in the economy. Together with globalization and increased social and economic cooperation, tourism will enjoy much faster growth. According to Malaysia's Ministry of Tourism [7], between 2002 and 2006, the number of arrivals in Malaysia increases by 4.5 million, from 13.29 million to 17.55 million and bring a 40% increase in revenue, from 25,781.1 million to 36,271.7 million.

2007 is the "Visit Malaysia" year- the year to promote Malaysia to the world. A lot of efforts have been put forward to attract foreigners to visit Malaysia. These efforts include using Information Technology as a significant role in creating a successful "Visit Malaysia" year. Websites promoting Malaysian tourism such as <http://allmalaysia.info> and <http://www.virtualmalaysia.com/> are examples of the roles of Information Technology in disseminating tourist information and attracting visitors to Malaysia.

Tourism has posed as an interesting field for the exploration and application of information technology. In its turn, information technology brings tremendous benefits to the growth of tourism.

## **2 PROBLEM STATEMENT**

Nowadays, it is very common that tourists have their own PDAs. Tourists always want to access information instantly, and plan an itinerary and schedule that can maximize their satisfaction and minimize the required travel time.

In countries with well-developed tourism industry such as France, Korea and Hong Kong [1], mobile assistance packages for tourists are readily available to make it easy and convenient for them to access information while traveling and assist them in making plans and schedules for their trips.

Information technology's roles in promoting tourism in Malaysia mostly focus on websites providing information and guidance for tourists. Websites such as *www.virtualmalaysia.com* and *www.tourism.gov.my* have been reliable source of information for tourists coming to Malaysia. However, efforts on developing mobile applications are limited to providing information, maps and guidance and have not yet provided a comprehensive system covering information guidance, tour scheduler and hotel recommender that suits the characteristics of the tourism industry in Malaysia. There is a need for Malaysia to develop a comprehensive mobile tourist assistance package as one of the ways to boost our tourism.

## **3 OBJECTIVES AND SIGNIFICANCE OF THE SYSTEM**

The objective of this project is to develop a tour assistance application on PDAs for tourists visiting Malaysia. The application will be readily available for use by tourists visiting Malaysia through relevant downloads (or push/pull technology) and it is aimed to make some contribution to promote tourism in Malaysia.

The application will deploy advanced mobile programming techniques such as multimedia display, synchronization with the desktop applications, and AI techniques.

These are the benefits the system is aimed to bring to users:

1. Save time and efforts looking for information about places of interest in Malaysia;
2. Plan and schedule their trips in a way that can optimize individual's preferences based on budget and time constraints;
3. Provide updated information about events and festivals in Malaysia;
4. Access to directions to tourist attractions by maps;
5. Find suitable hotels based on preferred area, ranking and budget;
6. Quick access to important contacts on urgent or emergent occasions.

From the potential benefits the system can bring, the system aims to contribute its part to promoting Malaysia's tourism.

#### **4 SCOPE OF THE SYSTEM**

Due to limited time frame of the project and resource constraints (in terms of memory of the handheld device), the system focuses on areas within and around Kuala Lumpur. The reasons for

taking Kuala Lumpur as the focused area are its ability to attract tourists and its reputation as the most well-known destination in Malaysia. However, the system will be developed in a way that it can accommodate for customization and expansion to other areas in Malaysia.

Due to the cost of and limited access to wireless network in Malaysia, the package is built as a stand-alone application without requiring any subscription to a wireless network.

## CHAPTER 2

### LITERATURE REVIEW

#### 1 TOURISM AND INFORMATION TECHNOLOGY

One undeniable feature of tourism is mobility [6], and information system serving the industry must take that into account. Tourists would like to have assistance at hand wherever they go.

In an article published on the ACM portal, Hannes Werthner and Francesco Ricci [6] say that travel and tourism represent approximately 11% of the worldwide GDP, according to the World Travel Tourism Council, and is expected to grow faster than any other economic sectors. In such a scenario in which technology becomes transparent and invisible to consumers, IT system in tourism should:

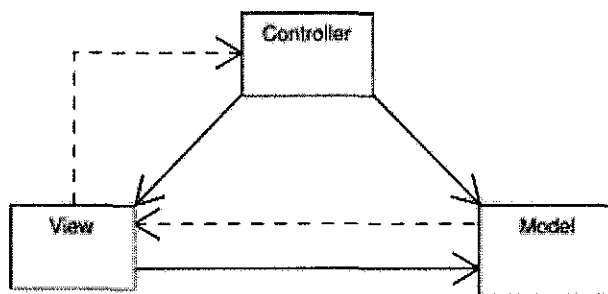
- Be scalable and open with regard to geographical and functional extensions. They will support the entire consumer life cycle and all business phases;
- Integrate mobile and fixed services, enabling multi-channel access to services provided by the various players;
- Support attentive user interfaces and personalization through extensive exploitation of user modeling, taking into consideration user behavior and cognition as well as emotional aspects.

These characteristics provide guidelines on the design and implementation of an information technology system, especially in tourism.

In software engineering, one of the most popular architectural patterns is the Model-View-Controller (MVC). Adopting the MVC model can help build a system that satisfies the abovementioned characteristics required of a tourism application.

The MVC separates data (model) and user interface (view) concerns so that changes to the user interface will not affect data handling, and that the data can be reorganized without changing the user interface. The model-view-controller solves this problem by decoupling data access and business logic from data presentation and user interaction, by introducing an intermediate component: the controller. [9]

In more general terms, Model is database, View is presentation and Controller is application. Figure 2.1 depicts the relationship between Model, View and Controller.



**Figure 2.1: MVC model**



## 2 AI TECHNIQUE IN MOBILE APPLICATION

Many applications in tourism adopt AI techniques, for example in travel planning and scheduling and recommender systems [6]. In tourism, some notable applications focus on destination selection and travel products bundling. These systems are called “recommender” which suggest products and provide consumer information to facilitate their decision process.

One of the most popular AI techniques applied in searching problems is Genetic Algorithm (GA). A genetic algorithm is one of a class of algorithms that searches a solution space for the optimal solution to a problem [8]. This search is done in a fashion that mimics the operation of evolution – a “population” of possible solutions is formed, and new solutions are formed by “breeding” the best solutions from the population’s members to form a new generation. The population evolves for many generations; when the algorithm finishes the best solution is returned.

There are other techniques for searching such as Dijkstra’s algorithm or A\* algorithm. Dijkstra’s algorithm is a greedy algorithm that solves the single-source shortest path problem for a directed graph with non negative edge weights [15]. For example, if the vertices (nodes) of the graph represent cities and edge weights represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between two cities. The A\* algorithm is a generalization of Dijkstra's algorithm that cuts down on the size of the subgraph that must be explored [16]. These two algorithms are useful for simple searching problems which take relatively little time to process. They also do not accommodate for expansion of routes or nodes.

Considering the complexity of the itinerary system for this system and possible expansion of routes in the future, Genetic Algorithm is the right candidate. This is because Genetic Algorithm is particularly useful for problems where it is extremely difficult or impossible to get an exact solution or for difficult problems where an exact solution may not be required. They offer an interesting alternative to the typical algorithmic solution methods, and are highly customizable, which make them an interesting challenge for exploration.

This is the pseudo-code of genetic algorithm:

1. Create a population of random candidate solutions named pop.
2. Until the algorithm termination conditions are met, do the following (each iteration is called a generation):
  - (a) Create an empty population named new-pop.
  - (b) While new-pop is not full, do the following:
    - i. Select two individuals at random from pop so that individuals which are more fit are more likely to be selected.
    - ii. Cross-over the two individuals to produce two new individuals.
  - (c) Let each individual in new-pop have a random chance to mutate.
  - (d) Replace pop with new-pop.
3. Select the individual from pop with the highest fitness as the solution to the problem.

**Figure 2.2 Generic Algorithm's pseudo-code**

The population is the collection of candidate solutions that we are considering during the course of the algorithm. Over the generations of the algorithm, new members are “born” into the population, while others “die” out of the population. A single solution in the population is referred to as an individual. The fitness of an individual is a measure of how “good” the solution

represented by the individual is. The better the solution, the higher the fitness – obviously, this is dependent on the problem to be solved.

One typical example of using GA in computing is the problem of Travel Salesman Problem (TSP). The problem states that there is a traveling salesman who services some number of cities, including his home city. He needs to travel on a trip such that he starts in his home city, visits every other city exactly once, and returns home. He wants to set up the trip so that it costs him the least amount of money possible.

The TSP shares some similarities with a typical tour schedule. However, differences may exist among different circumstances, designs and approaches to tackle the scheduling. Thus, it requires an adopt-and-adapt approach to using Genetic Algorithm to solve the problem.

### 3 DATABASE AND MOBILE APPLICATIONS

Data management and data access on mobile devices is central to mobile applications. As mobile applications achieve widespread adoption in the enterprise, mobile and embedded DBMSs which are needed to support such applications become an important part of the IT infrastructure. [12]

These are some characteristics an embedded DBMS should have:

- *Embeddable in applications:* Such DBMSs form an integral part of the application or the application infrastructure, often requiring no administration. Database technology is delivered as part of the application. While the database must be embeddable as a DLL in applications, it must also be possible to deploy it as a stand-alone DBMS with support for multiple transactions and applications.

- *Small footprint:* For many applications, especially those that are downloadable, it is important to minimize DBMS footprint. Since the database system is part of the application, the size of the mobile and embedded database system software affects the overall application footprint.
- *Synchronization with back-end data sources:* In the case of mobile data and cached data, it must be possible to synchronize the data with the back-end data. In typical mid-tier (application server) caches, the data is fetched from the back-end databases into the cache, operated on, and synchronized with the back-end database.

These characteristics are important in the selection of a suitable database for the system.

## CHAPTER 3

### METHODOLOGY

#### 1 SOFTWARE MODEL

This section will present the methodology to be adopted for the project. The software model that will be used is Waterfall Model. As illustrated in Figure 2, the software process model has five phases: requirement analysis, system design, system development, testing and implementation and maintenance.

The model theoretically requires each phase to be completed and “signed-off” before moving to the next phase. However, the system development and unit testing will be executed side by side to step-by-step stabilize the system. Evaluating the system performance by users will be done during the implementation and maintenance phase.

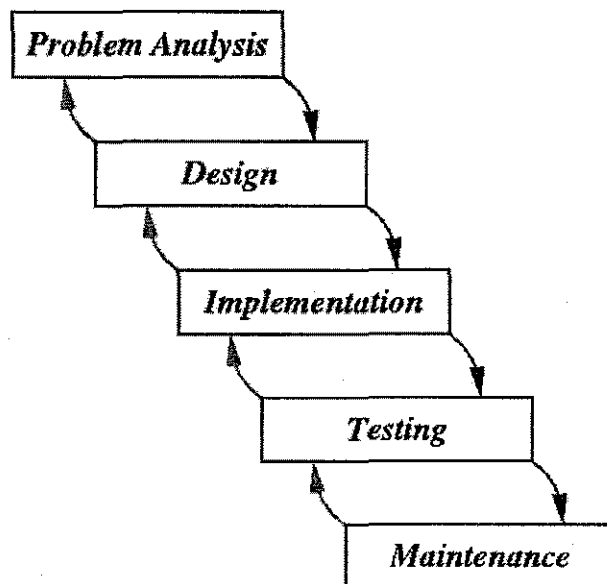


Figure 3.1: Waterfall model

## 2 SYSTEM ANALYSIS

### 2.1 Use Case Diagram

Below is the use case diagram for the system:

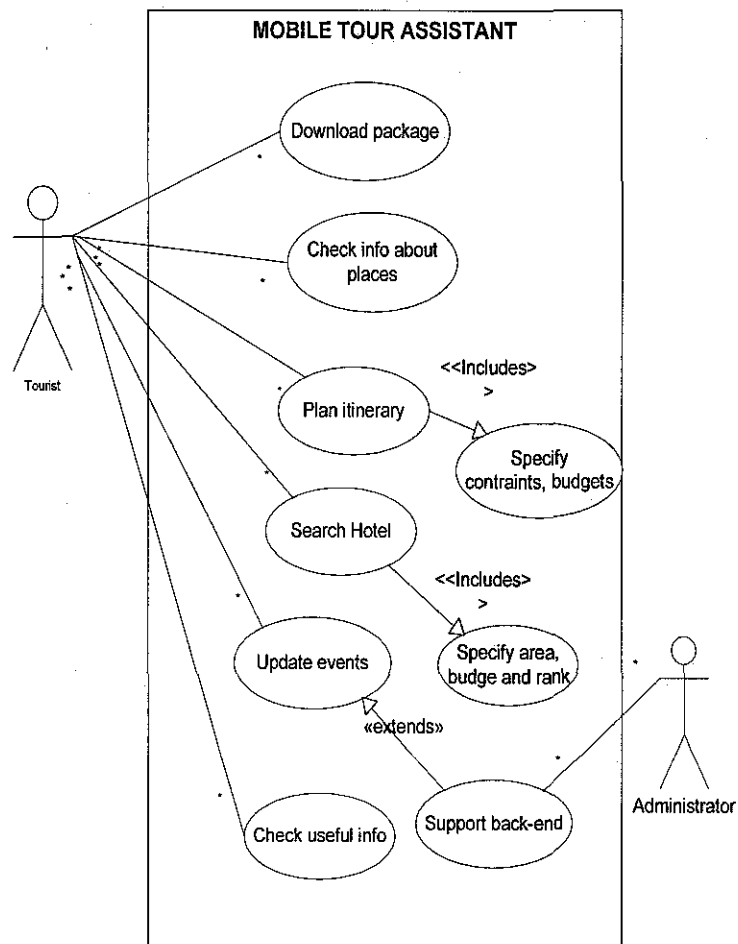


Figure 3.2 Use case Diagram

As can be seen in the diagram, the system involves two users: database administrator and tourist. The database administrator is in charge of inserting, updating and maintaining the central database for the system. The tourist downloads the package into their mobile device and performs necessary steps for the functions they need such as planning the itinerary and searching for hotels.

## **2.2 System's functionalities**

The analysis phase was carried out to determine what functions to include in the system as well as their scopes. Below is the description of the functions and scopes for the system:

- **Tourist attraction**

This function introduces Malaysian tourist attractions by making use of intensive multimedia such as movies, maps, and pictures. Information about places of interest such as location, opening hours, photos, and safety and traveling tips will be presented to the user. Movies and maps are meant to make the presentation more attractive.

The scope of the tourist attractions is focusing on places in Kuala Lumpur.

- **Itinerary planner**

This feature shall auto-generate a practical tour based on the user's time constraints and preferences. Based on users' preferences, for example on the type of sightseeing, dining taste or shopping styles, the system shall recommend relevant places for tourists to visit. This can be accomplished by applying some AI algorithm.

The itinerary shall be synchronized with the calendar organizer of the PDA to provide information for users' scheduling and the possible alarming functions.

- **Map viewer**

Maps are important to tourists. These may include transportation system maps, such as LRT maps and tourist attraction maps. The maps focus on areas within Kuala Lumpur.

- **Application download**

The system shall also provide a desktop-based website to introduce about the system, guide users how to use the system and how to download the package into the PDA. Download can be done via a wireless network, Infrared, Bluetooth or USB cradle.

- **Information Update**

Updating information about occasional events and festivals can be done by loading from the updated executable from the website. The information will be collected from relevant websites on Malaysian tourism such as <http://www.tourism.gov.my/> to look for updates.

- **Useful information**

In case of emergency, tourists should have access to useful information such as immigration laws, emergency numbers, embassy contact numbers and travel tips.

- **Back-end support**

The system shall have a back end module to allow for information update, edition and any other possible changes. For example, the back-end module should allow making changes to information about places of attraction and events.



## 3 SYSTEM DESIGN

### 3.1 Overall system design

The system adheres to a popular architectural pattern in software engineering, the *Model-View-Controller* (MVC). In the system, Model handles the database maintenance, View presents the interfaces to the user and Controller processes the requests and processes between the interface and database.

Adopting the MVC is one of the reasons why the system is written on .NET framework, in which the patterns are the patterns for the view and controller are well defined and the model is left to the developer to design.

#### **Model:**

WinForms does not strictly require a model. The developer has the option to create a model class, but may choose to forget it and have the event handlers in the controller perform any calculations and data persistence. Again, using a model to encapsulate business rules and database access is both possible and preferable. It is left to developers to design the Model.

#### **View:**

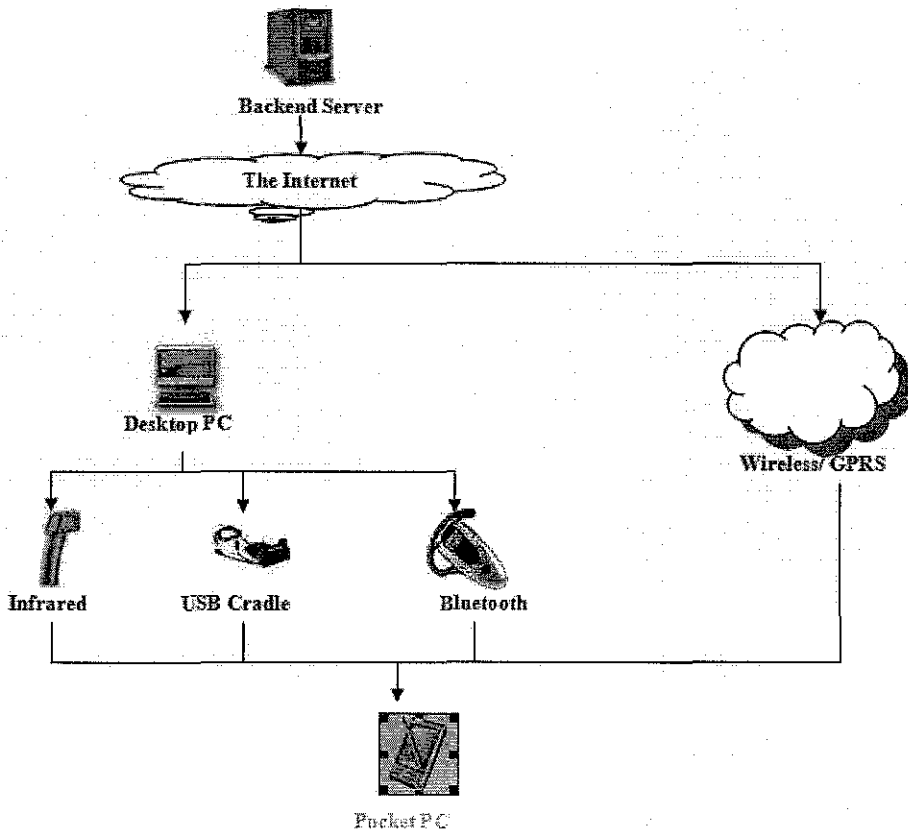
A class inheriting from either Form or Control handles the responsibilities of the view. In the case of WinForms, the View and Controller are compiled into the same class. This differs from ASP.Net, which uses inheritance, and Smalltalk, which have separate classes with pointers to one another.

#### **Controller:**

The duties of the controller are split between three places. The generation and passing of events start at the OS level. Inside the .Net framework, the Form and Control classes route the event to the proper event handler. The event handlers typically reside in the Form or Control class and are

implemented as delegates for the user interface events (e.g. button click, form load, listbox selection changed).

Below is the high-level view of the system:



**Figure 3.4: System design**

Figure 3.4 depicts the flow of the system. The backend is in charge of maintaining the central database and uploading the installation package to the Internet. The user then can download the installation package to the desktop PC and export to the mobile device via Infrared, Bluetooth or USB Cable. In the event of an available wireless or GPRS network, the package can be downloaded directly to the handheld device.

### 3.2 Database issues

Making the right choice of the mobile embedded database is a crucial issue in mobile development. After a thorough research on this, *Sybase* [2] has been chosen for the project because of the following benefits it brings:

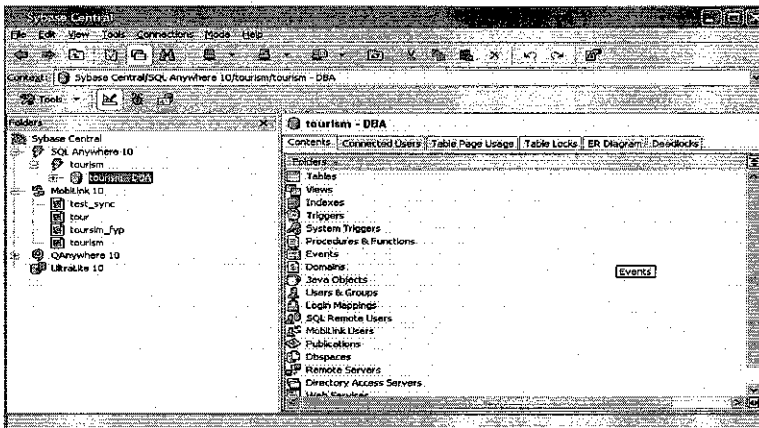
- provides a relational database management and synchronization system that can be deployed on small, mobile, and embedded devices;
- robust data management, powerful synchronization capabilities, multiple programming interfaces, and platform availability;
- can be integrated with MobiLink synchronization technology so the UltraLite database can synchronize data to a central database (SQL Anywhere in the project). [3]

Sybase has all in one for the back end (desktop) and front end (mobile) development and can be downloaded free from [10]:

- **SQL Anywhere 10:** an enterprise caliber database that scales to hundreds of concurrent users, hundreds of gigabytes of data, containing hundreds of millions of rows. Especially, it also offers a rich set of features that make it easy to operate in environments where I.T. support may not always be available, such as Self-tuning query optimizer.
- **UltraLite 10:** a database-management system designed for small-footprint mobile devices such as PDAs and smartphones. It only needs 150 KB of space [13]. It works on a wide range of mobile platforms such as Windows Mobile, Windows Mobile Smartphone Edition, Windows CE (Pocket PC, Handheld PC), Palm OS and Symbian OS

- **MobiLink 10:** a highly-scalable, session-based synchronization technology for exchanging data among relational databases and other non-relational data sources.

Figure 3.5 shows the interface of Sybase DBMS:



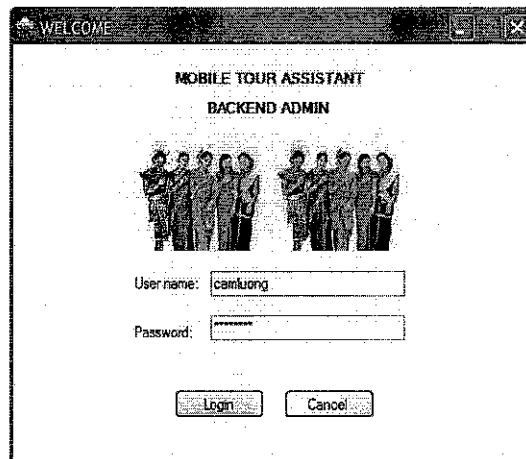
**Figure 3.5: Sybase's interface**

## 4 SYSTEM DEVELOPMENT

The system development consists of three components: Back-end on the desktop, Front-end on the mobile device and a website for package download.

### 4.1 Back-end Module

The backend module plays a very important role in storing, updating and managing master data for the system:



**Figure 4.1: Login Form for Backend Module**

The backend module includes:

- Tourist Attraction Manager
- Event Manager
- Map Manager
- Hotel Manager
- Contact Manager
- Data Synchronization

Each manager serves as the interface to insert, update and delete necessary data. For example, the map manager allows the system administrator to add map, associate each map with attractions and hotels within that map, update the information and delete unnecessary maps.

One of the critical features of the backend is managing images for all types of data. In this project, all images are stored in a folder called “*Image*” and their directories (paths) are stored in the database. Once the directory is called, the image will be displayed accordingly. The reason for this is to reduce the memory consumption on the PDA. The following section will discuss in detail the data synchronization and available managers of the back-end module.

#### **4.1.1 Data synchronization between central database and remote database via MobiLink**

Central data are stored in the *tourism* database which resides on SQL Anywhere 10 server. This database acts as the backend for the system. Mobile devices will retrieve information from the backend via the technology called “*data replication*” [4]. In this project, MobiLink [5] provides the synchronization mechanism between the backend and mobile devices.

The synchronization process replicates database from the central database in the remote device. With this mechanism, data can be easily inserted, edited and deleted at the backend and exported to the mobile device.

The MobiLink synchronization works through a synchronization model. Figure 3.7 shows the synchronization model “*tourism*” used for this project.

| Status | Remote Table              | Dir. | Consolidated Table        | DrId | Type      | Del.                                | DrId-Sub. | Cflc-Det. | Cflc-Res. |
|--------|---------------------------|------|---------------------------|------|-----------|-------------------------------------|-----------|-----------|-----------|
| 1      | embassy_info_del (DBA)    | ↔    | embassy_info_del (DBA)    |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 2      | embassy_info (DBA)        | ↔    | embassy_info (DBA)        |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 3      | event_inv_del (DBA)       | ↔    | event_inv_del (DBA)       |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 4      | event_inv (DBA)           | ↔    | event_inv (DBA)           |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 5      | hotel_inv_del (DBA)       | ↔    | hotel_inv_del (DBA)       |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 6      | hotel_inv (DBA)           | ↔    | hotel_inv (DBA)           |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 7      | interchange_inv_del (DBA) | ↔    | interchange_inv_del (DBA) |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 8      | interchange_inv (DBA)     | ↔    | interchange_inv (DBA)     |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 9      | map_inv_del (DBA)         | ↔    | map_inv_del (DBA)         |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 10     | map_inv (DBA)             | ↔    | map_inv (DBA)             |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |
| 11     | operation time del (DBA)  | ↔    | operation time del (DBA)  |      | Timestamp | <input checked="" type="checkbox"/> | None      | None      |           |

**Figure 4.2: Tourism synchronization model**

Steps of database synchronization between SQL Anywhere 10 and UltraLite using MobiLink server:

### 1. Creating the synchronization model:

This is a one time job. The model will match corresponding tables between the central (consolidated) database and the remote database. It will later be used for the *deployment* process to generate the new remote database (if no remote database has existed) or update any changes from the consolidated database to the remote one.

### 2. Deploying the synchronization model:

Before deploying the model, we must make sure the schema of the model is updated. We can do this by selecting the “*update schema*” function of the model. The deployment process is quite simple with all the clear guides from the wizard.

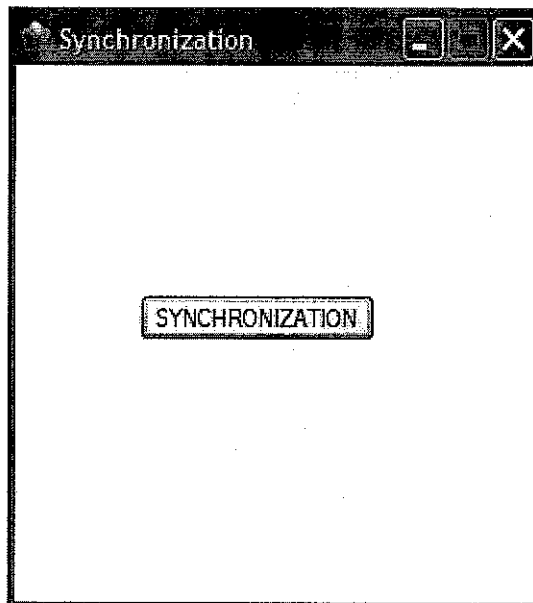
The results of the deployment process are three folders:

- “*consolidated*”: contains files to run the consolidated database
- “*mslv*”: contains files to run the MobiLink server

- “*remote*”: contains file to generate or update the remote database.

### 3. Synchronizing:

To start synchronization, MobiLink server must be started first. Then is the consolidated database, and lastly is the sync bat file in the “*remote*” folder. Each step is carried out by one particular .bat file in the three folders mentioned in step 2. However, we can combine three .bat files into one .bat file and call that single file from a program as in Figure 3.8. The synchronization can be done easily by a click. The result file is a .udb file that contains all data from the central database on the desktop.



**Figure 4.3: Synchronization interface**



## 4.1.2 Tourist Attraction Manager

The screenshot shows a window titled "Attraction Manager" with two tabs: "ADD ATTRACTION" (selected) and "UPDATE ATTRACTION". The form contains the following fields and controls:

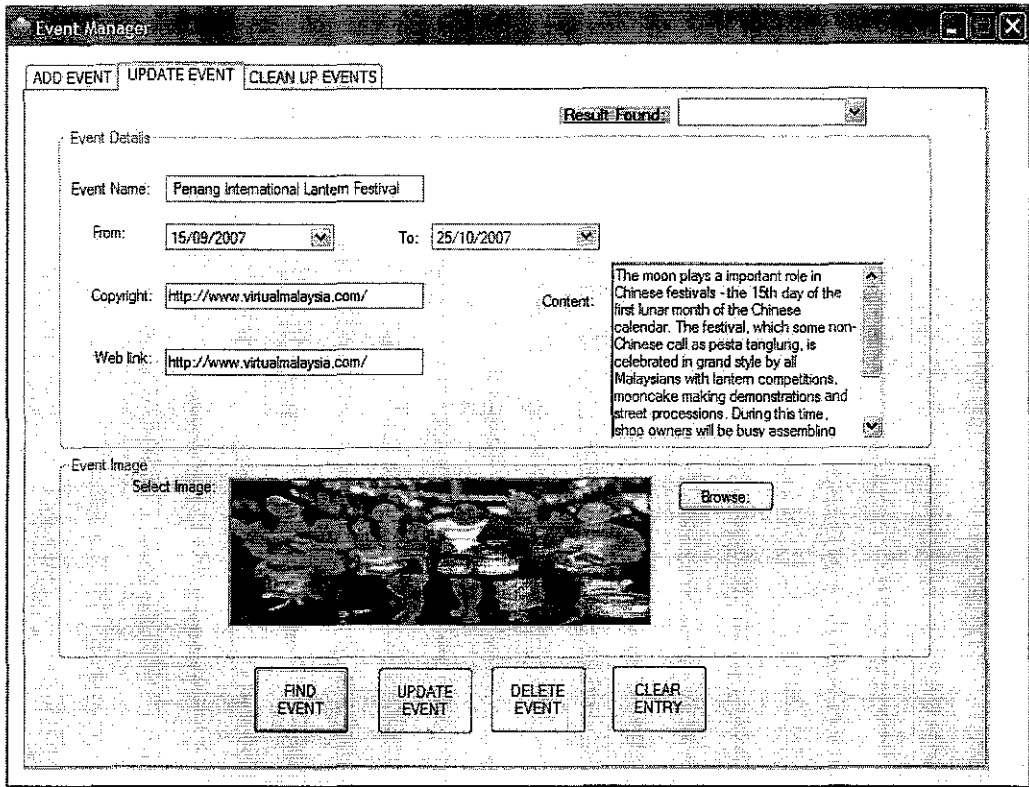
- Attraction's name: Lake Garden
- Putra Line station: [dropdown]
- Star Line Station: [dropdown]
- Monorail station: Hang Tuah
- Others: [dropdown]
- Time to attraction: [spinbox] (minutes)
- Stay Duration: [spinbox] (minutes)
- Notes on transportation: [text area]
- Opening hours: 0 to 0
- Break time: 13 to 14
- Shopping hours: 0 to 0
- Sightseeing section:
  - Natural: 3
  - Building: 4
  - Museum: 4
  - Malay culture: 3
  - Chinese culture: 2
  - Indian culture: 3
- Eating section:
  - Malay: 2
  - Chinese: 2
  - Indian: 2
  - Western: 2
- Shopping section:
  - Shopping mall: 0
  - Street shopping: 0
- Buttons: ADD ATTRACTION, CLEAR
- Image section: Open, Update Image (with a small image of a park)
- Attraction's Description: Situated just a short distance away from the Merdeka Square lies the Lake Gardens, Kuala Lumpur's most popular park. Located at Jalan Perdana, the sprawling 51.6 hectare

Figure 4.4: Attraction Manager

Besides storing necessary information about a tourist attraction, the attraction manager also allows inserting the score of different criteria for each attraction. These scores are used for recommending and devising the itinerary for users. As illustrated in Figure 3.21, the scoring criteria are divided into 3 groups:

- Sightseeing: Natural, Building, Museum, Malay, Chinese and Indian culture
- Eating: Malay, Chinese, Indian and Western cuisine-
- Shopping: Street shopping, Mall shopping

## 4.1.3 Event Manager

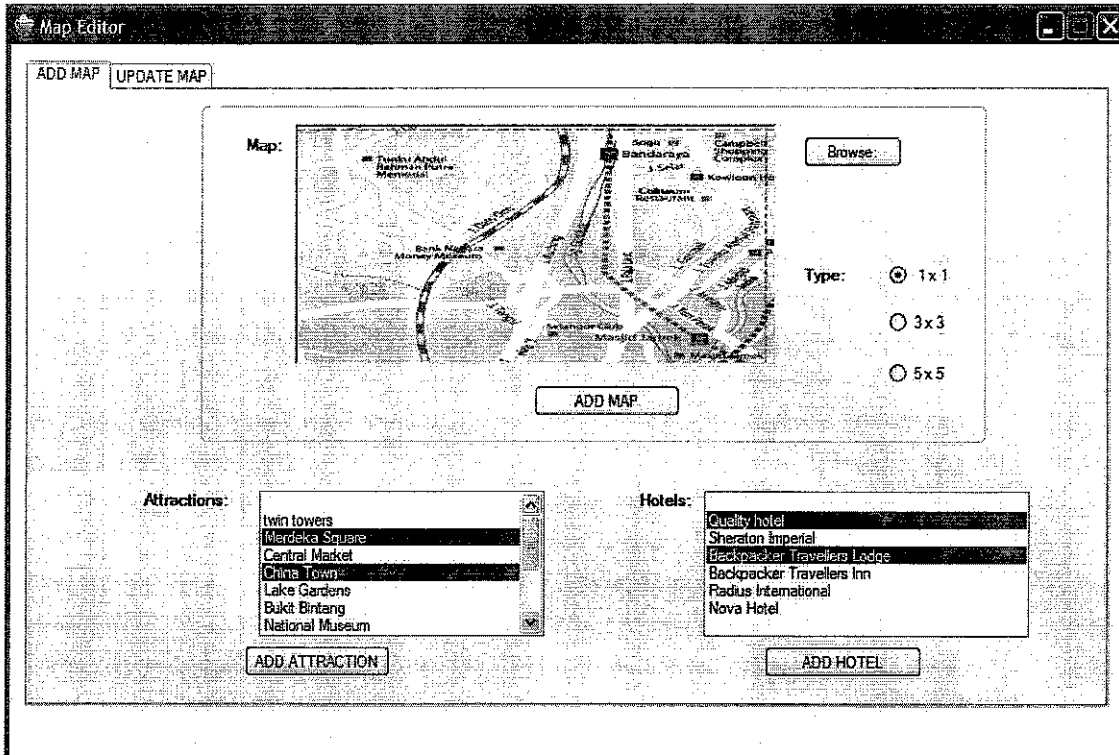


**Figure 4.5: Event Manager**

Information about upcoming events is managed through the event manger. Sources of information come from popular websites on Malaysian tourism such as *www.virtualmalaysia.com* and *www.tourism.gov.my*.

One feature of the event manager is cleaning up events happening before today.

#### 4.1.4 Map Manager



**Figure 4.6: Map Manager**

The map manager stores maps of the whole Kuala Lumpur and associates the attractions and hotels to the map. These data will be used to display map for each attraction and hotel.

#### 4.1.5 Hotel Manager

The screenshot shows a window titled "Hotel Editor" with two tabs: "ADD HOTEL" (selected) and "UPDATE HOTEL". The form contains the following fields:

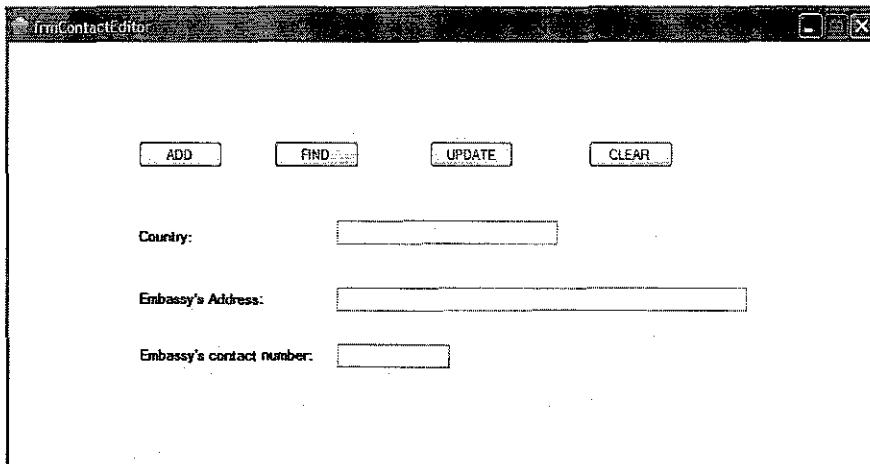
- Name:** PNB Darby Park
- Address:** 10 Jalan Beringin 50450 Kuala Lumpur
- Telephone:** +603 2141 9111
- Description:** PNB Darby Park an award-winning 5-star international hotel in Kuala Lumpur, is conveniently located adjacent to the KL International Airport's Main Terminal Building. Pan Pacific Kuala Lumpur International Airport is the preferred hotel choice for transit air travellers with a few long hours to 'kill' in-between flights.
- Area:** Petaling Jaya
- Rank:** 3 star
- Room Rate:** 200-300

At the bottom of the form are two buttons: "ADD HOTEL" and "CLEAR".

**Figure 4.7: Hotel Manager**

The hotel manager administers information about a hotel in the area of KL such as name, address, contact number, location, rank and nearby attractions. The location and rank are used for users to select the hotel they want.

#### 4.1.6 Contact Manager



**Figure 4.8: Contact Manager**

The contact manger stores information about the embassy of some countries. The information of each embassy includes its address and contact number. This serves some emergency needs of users.

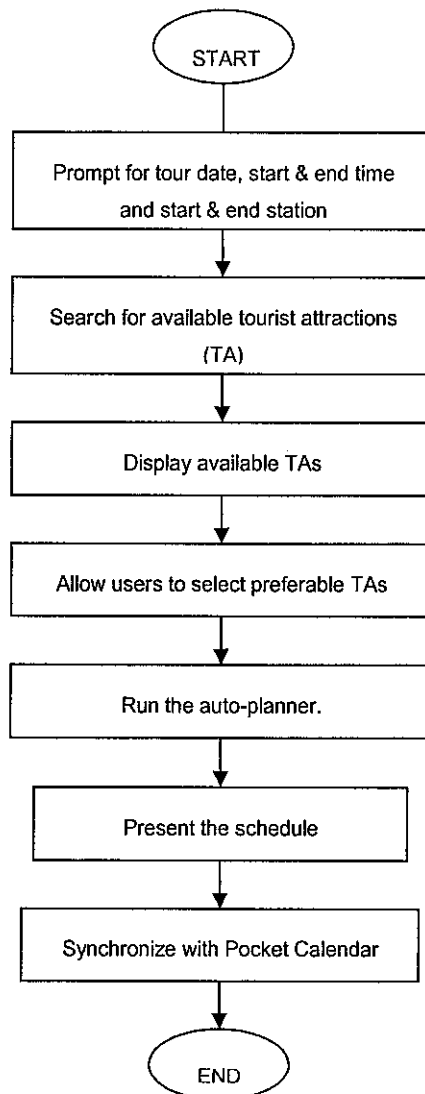
## 4.2 Front-end Module on Mobile Device

The front-end module on mobile device consists of three components:

- ***Itinerary Planner***: Using an AI technique, it schedules the trip for tourists based on their time constraint and tourist attraction preferences.
- ***Tourist Kits***: provides necessary information on tourist attractions, maps, events and festivals and useful information such as emergency contact numbers.
- ***Hotel Recommender***: recommends suitable hotels for tourists based on their chosen location, budget and hotel type.

## 4.2.1 Itinerary Planner

### 4.2.1.1 Activity diagram



**Figure 4.9: Activity diagram for itinerary planner**

### 4.2.1.2 Scope of the functionality

The itinerary planner functionality is focused on *transit tourists*, who visit Malaysia for a short time while on their journey to another place. They usually do not have a specific travel plan due to the uncertainty about their availability for a tour while in transit. When they decide to go for an ad hoc tour, their main concern is the time constraint. Their wish would then be to be able to visit as many places of their interest while keeping up with their tight schedule.

Meanwhile, tourists on holiday packages usually have their itinerary arranged by tour agents. Thus the itinerary planner may not be as useful to them. Nonetheless, the system can still accommodate almost any type of tourist.

Therefore, a system that is able to automatically recommend an itinerary based on tourists' schedule, preference and budget will be of good value to them.

#### **4.2.1.3 How it works**

The Itinerary Planner is composed of 3 main steps:

- *Provide Tour Information*: to get users' time constraints and start point.
- *Select Attractions*: to allow users to select preferred places among available ones.
- *Save To Calendar*: to allow users to save the suggested itinerary to their Pocket PC Calendar, together with alarm capability.

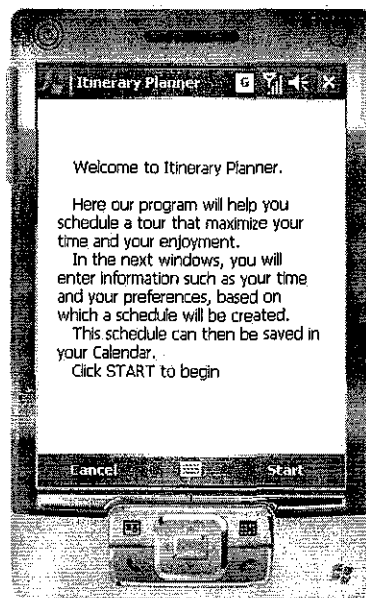
#### **4.2.1.4 User interfaces**

The interface of this module is simple as can be seen from Figures 3.8a-g.

The module works by guiding user step-by-step through each screen, until the final step when a schedule is added to their Calendar (Figure 4.10g).

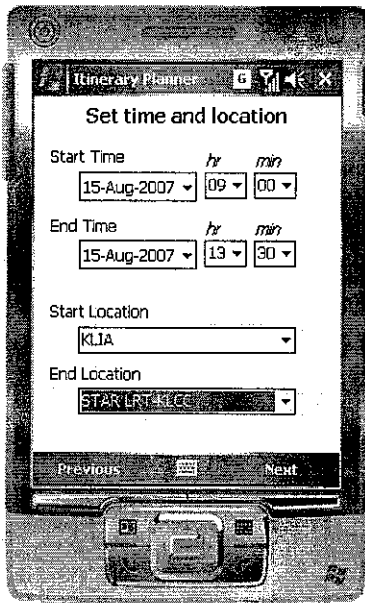
Below are the explanations of each of the screen:

- Figure 4.10a: A welcome screen to display a welcome message to user.
- Figure 4.10b: A form for user to enter their time and location constraint.
- Figure 4.10c: Another form for user to enter their preferences.
- Figure 4.10d: A summary display of the conditions the user has input.
- Figure 4.10e: A display of tourist attractions which best matches users' requirements. User can then customize further by choosing places of their best interest.
- Figure 4.10f: The result of the Itinerary Planner, detailing the auto-planned schedule for user.
- Figure 4.10g: User's Pocket Calendar after user has chosen to add schedule to Calendar.

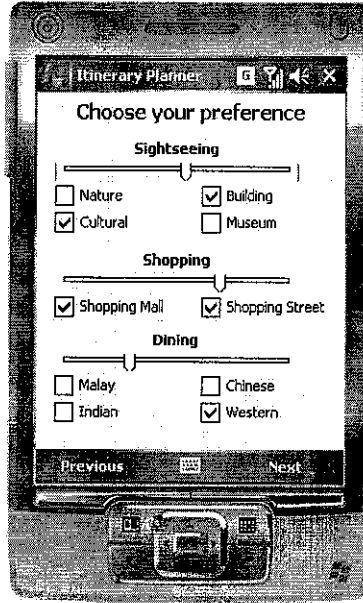


a)

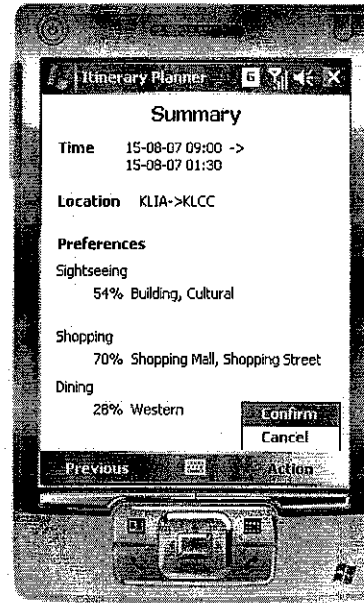




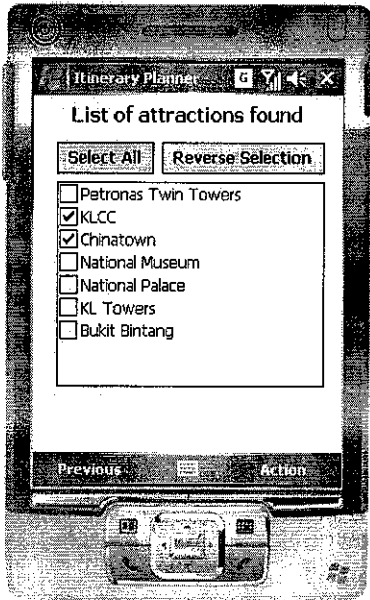
b)



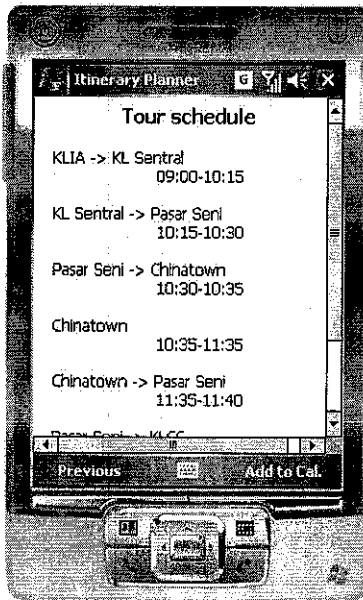
c)



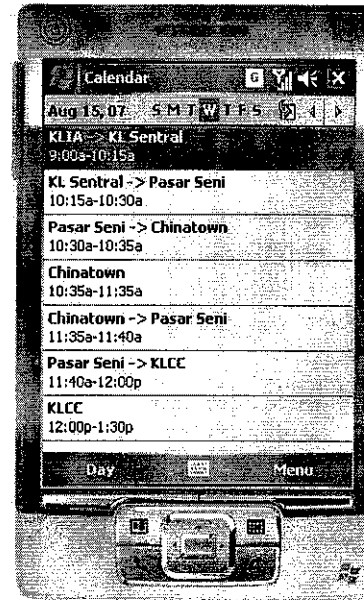
d)



e)



f)

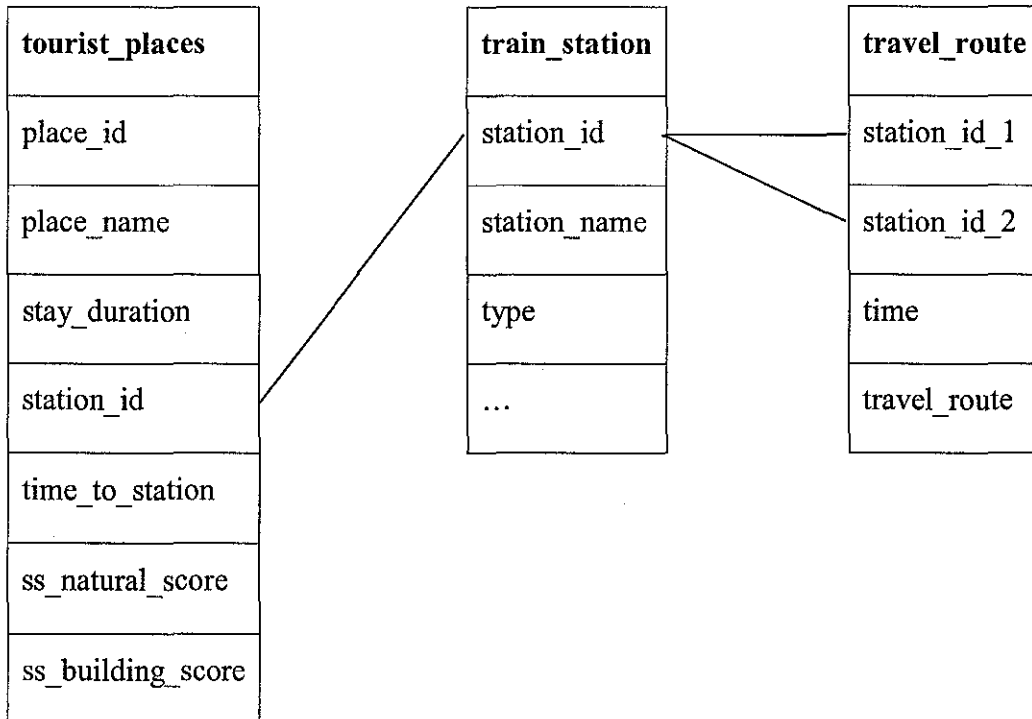


g)

Figure 4.10: Itinerary planner's interface

#### 4.2.1.5 Database Design

In this module, the following tables play the main roles:



**Table 4.11: Major tables for Itinerary Planner module**

Table **tourist\_places** contains its scoring information, to be used in ranking attractions based on user's preferences. It also contains the train station information which allows the application to know which train station is nearest to the attraction, as well as the amount of time to go from the station to the attraction.

Table **train\_station** contains just the important information of each station, while table **travel\_route** details the time required to go between any two station as well as the route to take. The route here is the list of *important* intermediate stations, which usually are places where user has to change from one type of train to another type. The route for 2 stations of the same type is thus empty. For example, to go from KLCC (Putra LRT) to Bangsar (Putra LRT), the route is

empty, as it is direct route. Meanwhile, to go from Bukit Bintang (Monorail) to Bangsar (Putra LRT), users need to know where to change line. Thus the route information would let users know that from Bukit Bintang, users have to go to KL Sentral (Monorail) first, then moving on to KL Sentral (Putra LRT) to change line, and finally to Bangsar.

#### 4.2.1.6 Algorithm

##### **Problem definition:**

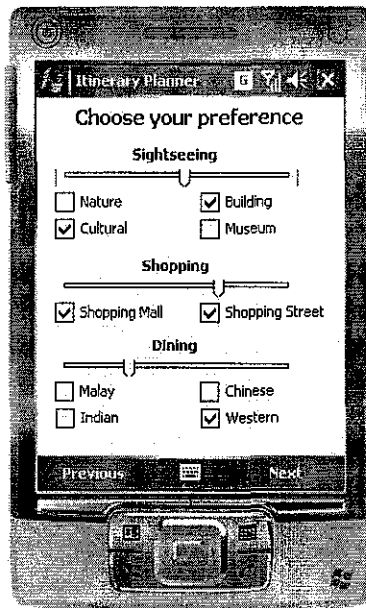
The main problem is to find a suitable schedule/route which has the maximum score but still satisfy the user's time and location constraint. Score means the level at which the attractions in the route matches user's preferences.

##### **Analysis:**

The problem requires the application to first choose the places according to their suitability (score) to the user's preferences. Then from the list of places, design a fitting schedule.

##### *i Finding suitable attractions*

The first step can be done in a simple way. User's preferences are broken down into a list of activities such as Sightseeing, Shopping and Dining. Each of these activities can be broken down further, for example, sightseeing may encompass nature sightseeing, building sightseeing and culture sightseeing. Each type of activities is given a weight, while each subtype can be either chosen or excluded.



**Figure 4.12: Preference selection**

For each attraction, all score of various subtypes has already been provided in the table `tourist_places`. To determine an attraction's suitability, these scores are combined with the weight that user has given, to form the final score for the attraction with respect to that user's preference. After the user has confirmed his/her preference, the places will be ranked according to their scores, and a number of top attractions will be chosen from there for further customization by user.

*ii Arranging the attractions to form a tour*

Once a list of desired attractions is formed, we move on to the more difficult task, arranging the attractions for a schedule with the best total score while staying under the time constraint. To best achieve this objective, we try to match this problem with an existing problem which already has an effective solution. This problem has some similarity to the traveling salesman problem

(TSP), which can be solved effectively using genetic algorithm, an algorithm popularly used in the field of Artificial Intelligence.

TSP is defined as “Given a number of cities and the costs of traveling from any city to any other city, what is the cheapest round-trip route that visits each city exactly once and then returns to the starting city?” [14]. In TSP, the salesman has to find the route with least distance (best time) while in this problem, the application needs to find the route with best score.

A major difference between the two problems is that the salesman always has to visit ALL places, while in this problem, a route might consist of a full set of attractions or a subset of attractions. Herein lies the complexity of the problem at hand. While TSP can be solved in a straightforward manner, the problem at hand requires additional computations.

Nonetheless, we would want to take advantage of the known effectiveness in solving the TSP problem, and apply that to the problem at hand. We thus implement an algorithm that first ignore the time constraint and consider all of the attractions that the user has chosen, i.e., we consider all tour routes that go through all attractions. Then, the algorithm should find the route with best TIME (here time is considered important instead of score since all routes that go through the same set of locations has the same total score). The problem of finding a least time-consuming route going through all attractions is essentially the same as TSP. We name the solution to this problem the minimum-duration complete-coverage route.

Once a minimum-duration complete-coverage route has been found, the time constraint has to be satisfied. From the minimum-duration complete-coverage route, the algorithm will try to find the subset satisfying the time constraint and with the best score. To decide which attractions to remove to form the subset, the algorithm assign an individual fitness to each attraction. This individual fitness is best taken as the ratio of an attraction’s score and the time wasted for it. The

time wasted for an attraction A in a route R is calculated by taking the time difference between route R and a new route R' with A removed. The ratio of score and time wasted is calculated for each attraction in a route. The attraction with the lowest score is then removed from the route. This way, attractions with lower score or which takes too much time to visit are more likely to be removed. The removal process is gradual, that is, attractions are removed one-by-one and after each removal the ratio of score and time wasted of each attraction are reevaluated. Attractions are removed until a subset is found to satisfy the time limit.

The subset found is the solution to the main problem.

### **Application of Genetic Algorithm:**

The main motivation for the use of genetic algorithm is its ability to achieve near-optimal solution within reasonable limitations (in time and space). A brute force approach would need to compute  $n!$  different combinations (where  $n$  is the number of attractions). This is not effective, and may be impossible to achieve even for a Windows Desktop environment let alone the Windows Mobile environment on which the Itinerary Planner works.

Below is a pseudo code of genetic algorithm [14]:

1. Choose initial population
2. Evaluate the fitness of each individual in the population
3. Repeat
  - a. Select best-ranking individuals to reproduce
  - b. Breed new generation through crossover and mutation (genetic operations) and give birth to offspring
  - c. Evaluate the individual fitness of the offspring
  - d. Replace worst ranked part of population with offspring
4. Until <terminating condition>

First we need to know what composes the *population*. These are essentially complete-coverage routes; meaning that routes which go through all attractions (in a particular order). In order to perform genetic operations like crossover and mutation, we need to represent route as a list of attractions, sorted in the order that they are visited, with the start and end locations being those taken from user input.

The next important aspect is how we compute the *fitness* of the route. As we are looking for the minimum-duration route, the fitness should correspond the duration of the route. The duration of the route is computed as the sum of time required to go from the starting point then to each of the attractions in the order we have found until we reach the ending point. The quickest routes from a sample are chosen for crossover and new routes created are to replace lengthy ones. Here a sample means a randomly selected subset of the population, since taking the whole population does not produce much variation in the population (every time only the best 2 of the population are crossovered), leading to slower path to the optimal solution.

The *crossover* will take every link that exists in both parents and place those links in both children. Then, for Child 1 it alternates between taking links that appear in Parent 2 and then Parent 1. For Child 2, it alternates between Parent 2 and Parent 1 taking a different set of links. For either child, there is a chance that a link could create an invalid tour where instead of a single path in the tour there are several disconnected paths. These links must be rejected. To fill in the remaining missing links, cities are chosen at random. Since the crossover is not completely random, this is considered a greedy crossover. [13]

Eventually, this GA would make every solution look identical. This is not ideal. Once every tour in the population is identical, the GA will not be able to find a better solution. There are two ways around this. The first is to use a very large initial population so that it takes the GA longer to make all of the solutions the same. The second method is *mutation*, where some child tours are randomly altered to produce a new unique tour.

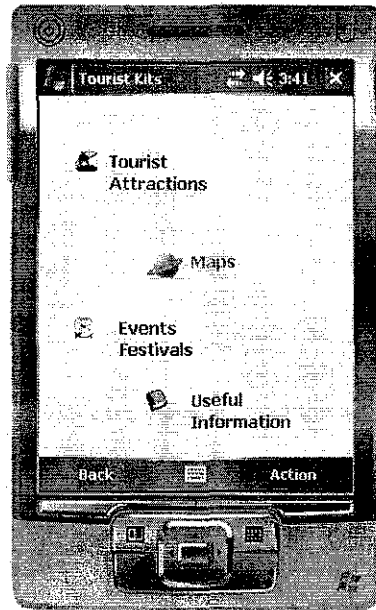
Once the child has been produced, its fitness will be calculated. If its fitness is higher than the worst member's, the child will replace the existing worst. This maintains the size of the population. If its fitness is higher than the existing best, it will hold the position of being the best of the population. The process of crossover and mutation is repeated until it cannot produce any better child than the existing best member.

**Main algorithm:**

1. Gather user's constraints and user's preference.
2. Find a number of attractions with the best scores after applying user's preference.
3. Let users choose which attractions to be included in the tour.
4. Use genetic algorithm to find the minimum-duration complete-coverage route
5. Repeat
  - a. Compute the time wasted for all remaining attractions
  - b. Compute the ratio of score against time wasted for all
  - c. Remove the attraction with the lowest ratio from the route
  - d. If the new total time of the route is less than the total time the user has, then move on to 6
6. Compute the exact time (& date) of visiting each attraction
7. Display the route to user
8. Add route to calendar



## 4.2.2 Tourist Kits

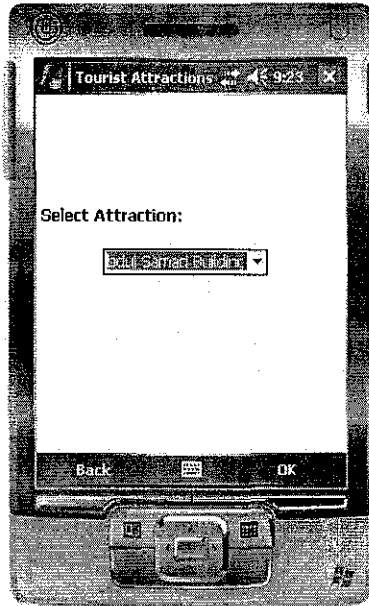


**Figure 4.13: Tourist Kits**

The Tourist Kits component consists of 4 parts: Tourist Attractions, Maps, Events and Festivals, and Useful Information.

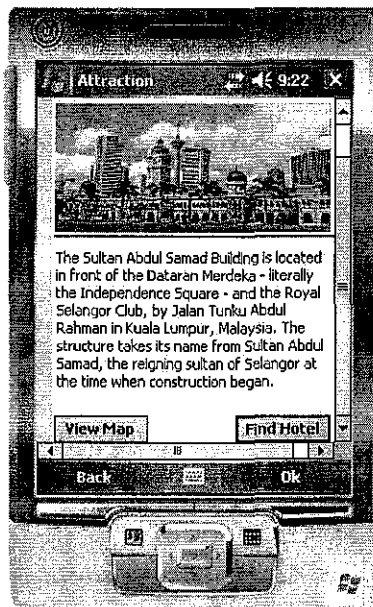
### 4.2.2.1 Tourist Attractions

The user can choose the attraction they want to view from the drop-down list:



**Figure 4.14: Tourist Attractions' list**

Information about the selected place will be displayed as follows:



**Figure 4.15: Tourist attraction's display**

#### 4.2.2.2 Map View

Users can select maps based on attraction or area. Maps can be zoomed in and out with certain sizes.

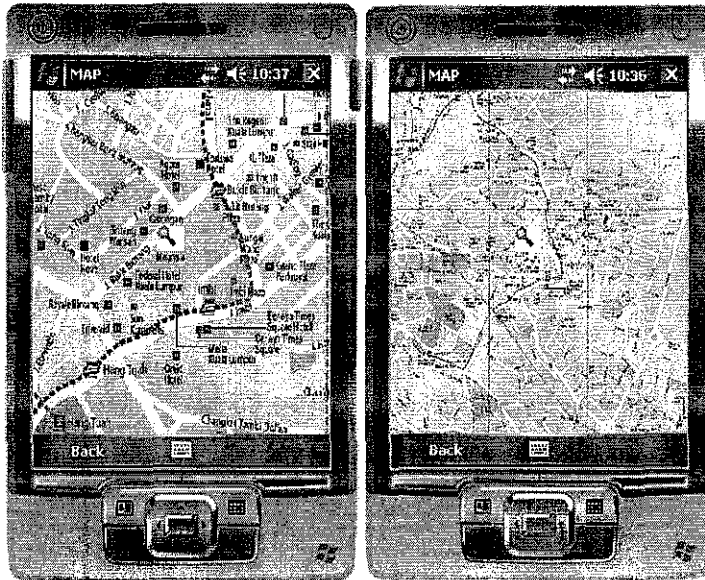


Figure 4.16: View Map

#### 4.2.2.3 Events and Festivals

The package also presents information about upcoming events in Malaysia. A list of events is displayed first and users can select to view the one they are interested in.

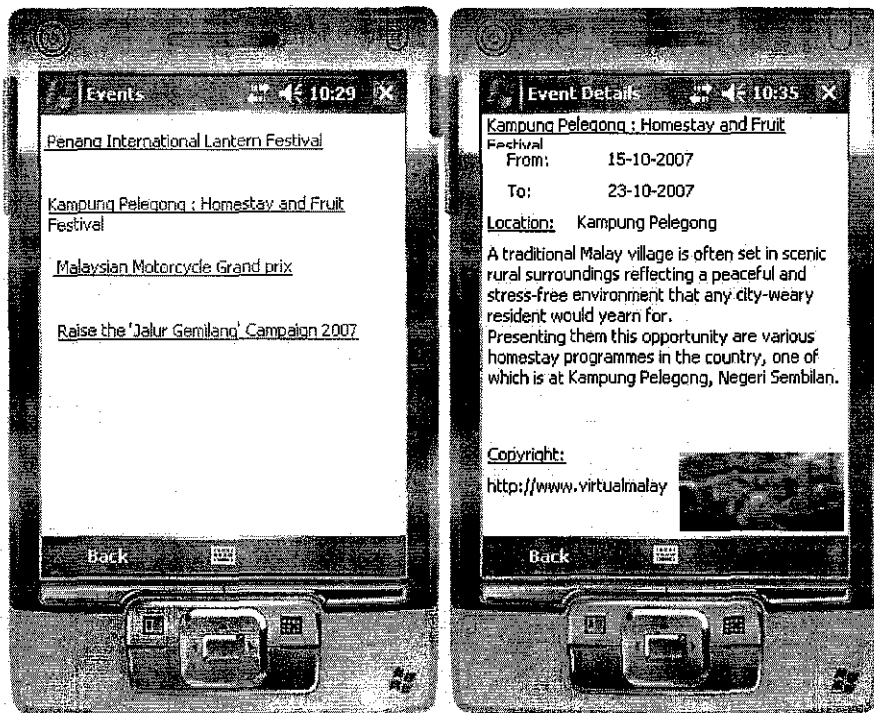
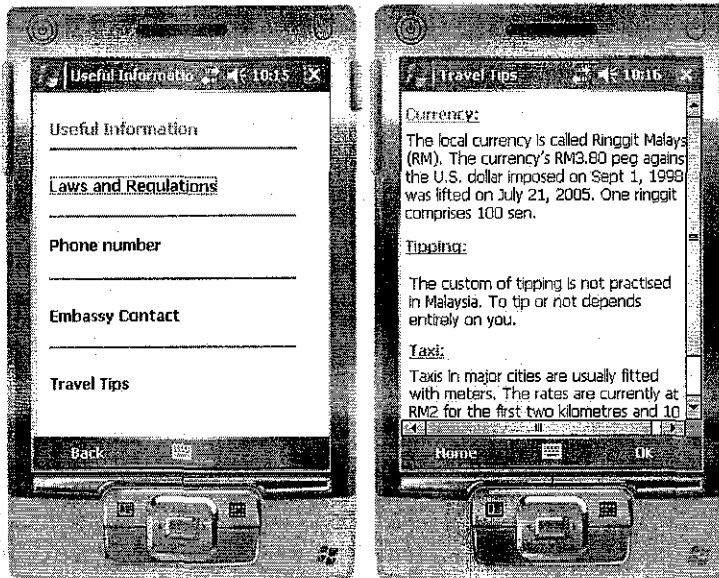


Figure 4.17: Events & Festivals

#### 4.2.2.4 Useful Information

This component presents the user with useful information about emergency contact numbers, embassy contact numbers, basic immigration laws and regulations and travel tips.

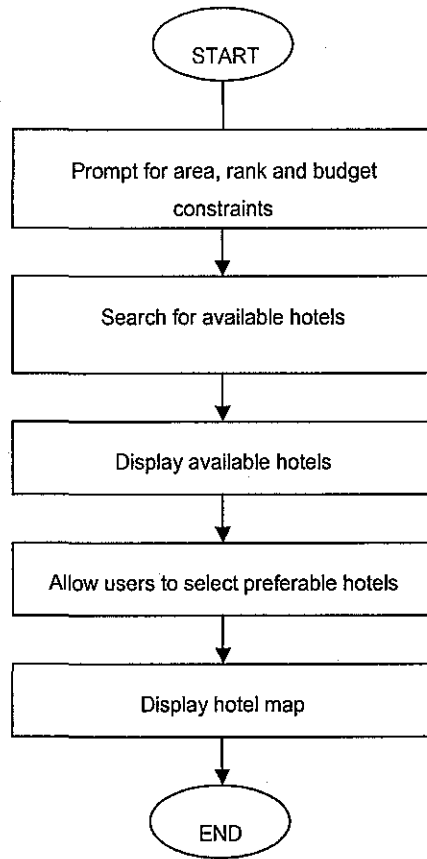


**Figure 4.18: Useful Information**

### **4.2.3 Hotel Recommender**

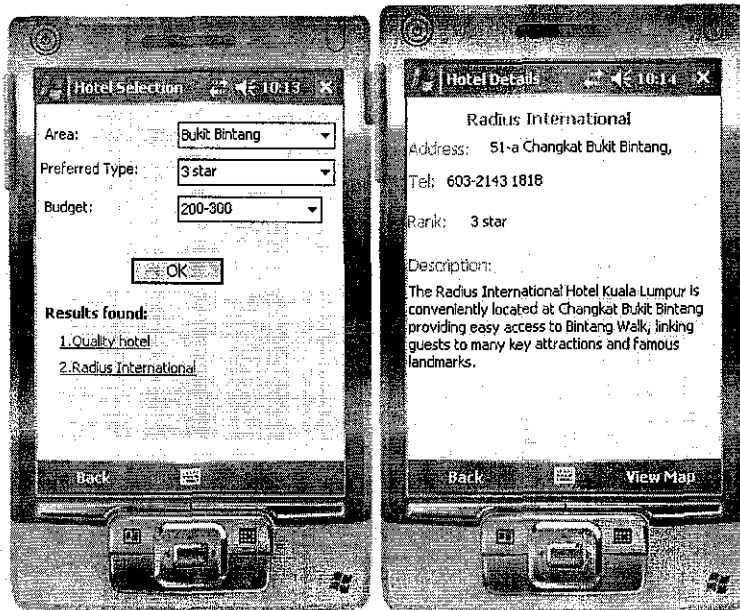
This functionality helps users to find the hotel based on their budget and preferred location.

Figure 3.16 illustrates the flow of the hotel finder.



**Figure 4.19: Activity diagram of hotel finder**

Some interfaces of the hotel recommender are below:



**Figure 4.20: Find hotels**

### 4.3 Website

The website is where interested users of the system can go to download the available packages. There are two types of download on the website. One is the download of the cab file for the first time installation. The other is the download of the updated data file.

The website is currently available at the address: <http://www.vilacen.com/mta>.

### 4.4 Tool requirements

#### 4.4.1 Hardware

PDA's running on Windows Mobile 5.0 operating system  
or a Windows Mobile emulator.

#### 4.4.2 Software

Languages:

- C# and .NET;

Database:

- Sybase including UltraLite, SQL Anywhere 10 and MobiLink.



## CHAPTER 4

### RESULTS AND DISCUSSION

#### 1 TESTING

##### 1.1 Backend Testing

The backend acts as the central database storage and synchronization centre for the system. Therefore, it requires a thorough testing on the backend to ensure the data entered are valid and exceptions are properly handled.

The purposes of the backend testing are:

1. To ensure data are correctly entered into the database;
2. To ensure exceptions are handled;
3. To prepare the data to be exported to the mobile device.

The test data are collected from all sources of information about Malaysia's tourism, mostly from the two popular websites: *www.tourism.gov.my* and *www.virtualmalaysia.com*. Testing has been carried on all modules of the backend including the place editor, map editor, event editor, contact editor and synchronization.

##### 1.2 Synchronization testing

As the system is stand-alone, synchronization is done to replicate data from the central database (SQL Anywhere 10) to the remote database (UltraLite). The synchronization process is

programmed and can be easily performed by clicking the Synchronization button in the Synchronization form of the Backend Module.

The expected results are:

1. First, the MobiLink server is started;
2. Second, the consolidate (central) database connection is opened;
3. Finally, the data is synchronized from the consolidated database to the remote one.

The output of the synchronization process is a file with .udb (UltraLite database) extension. This .udb file is later embedded in the mobile device.

### **1.3 Emulator testing**

The testing on the emulator has been done on Windows Mobile 5 and Pocket PC 2003. The system has run well on those platforms. Connecting the system to the emulator can simply be done by the debug utility of Visual Studio 2005.

The purposes of emulator testing are:

1. To test the functioning of all the components on the mobile end of the system and the interaction among them;
2. To test the display of the forms and images;
3. To test the memory resources, make sure that the device is not overloaded and the processing takes as little time as possible.

The functionalities include three components:

- Itinerary Planner
- Tourist Kits
- Hotel Recommender

#### 1.4 Device testing

The testing has been carried out on the HTC Touch running on Windows Mobile 6. (Refer to the Appendix 3 for more information on the HTC Touch). Testing on the device has two parts: installation testing and functionality testing.

##### *Installation testing*

In Microsoft.NET, the installation package for mobile devices is called a cab file. The cab file contains the executable file and embedded database of the system. The cab file can be downloaded from the following website: [www.vilacen.com/mta](http://www.vilacen.com/mta). Once the download to the desktop has been completed, the connection between the device and desktop can be done via Bluetooth, Infrared or USB Cradle.

Due to limited resources, the testing has been done only on the USB Cradle connection. The testing process is as follows:

- *Connecting via Active Sync*: Usually, the connection is automatic upon detecting the connection with the device.
- *Exploring the files*: The cab file can be copied to any location in File Explorer. Double click the file and the installation process will start.

### *Functionality testing*

The three components of the mobile part are tested to ensure all functionalities are working.

## **2 DISCUSSION**

The backend testing proves that the system is stable in terms of maintaining the central database as well as synchronizing for data replication between the central and remote database.

Since images are stored in a separate folder residing on the device, processing of images does not take much resource from the device while it is running. Thus, it prevents out of memory exceptions and improves the application process. However, the recommended practice is to keep the image size as small as possible in order to reduce memory processing.

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **1 CONCLUSION**

The mobile tour assistant will certainly benefit tourists in Malaysia by providing everything they need in a package at hand. They can easily access information about places, maps, events and hotels. Moreover, the system helps them schedule a satisfactory itinerary based on their preferences and constraints. The stand-alone feature also makes it suitable to short-stay and transit tourists who usually do not subscribe to any local wireless network.

The project has been putting efforts into making use of and improving available tools, techniques and methodology in current wireless application development such as embedded database and AI technique. However, the project is not simply written on flat programming. On the other hand, it deploys the intelligent technique in the itinerary planning. This is some contribution of the project in terms of computing technique.

The major achievements of the project are an intelligent mobile tour guide and backend supporting modules. The design of the system makes way for customization and with the installation package ready, the system can be sellable to the market.

#### **2 RECOMMENDATIONS**

These are the additional features that the can be added in enhancements of the system

### **1. Online connection**

Currently the system is offline without direct connection to wireless networks. Further enhancements can allow the system to have online connection for instant download of updates and information and access to dynamic map such as Google Map.

### **2. Interactive map searching**

The current system allows users to view map and zoom in and out in certain sizes. The map view function can be expanded in a way that allows users to search a particular place using maps and draws out the graphs of how to reach there.

### **3. Additional features in itinerary scheduling**

The current itinerary planner considers three means of transportation in scheduling the trip: Monorail, Putra line and Star line. Further enhancements of the system can include other means of transportation such as buses and taxi. The current planner recommends the time duration users should spend in a particular place without letting them make any changes. Later improvements should both recommend the duration and allow users to adjust the time spent in each location.

### **4. Multilingual support**

The current system supports in one language, which is English. Other popular languages among tourists in Malaysia such as Chinese, Arabic and French can be included in future enhancements.

## REFERENCES

- [1] Department of Computer Science and Engineering, Hong Kong University June 2007.  
<<http://www.cse.ust.hk/News/IITEC2004/>>
- [2] Sybase Inc. August 2007. “*SQL Anywhere- UltraLite*”  
<<http://www.iAnywhere.com/products/ultralite.html>>
- [3] Sybase Inc. August 2007. “*Overview of UltraLite Development and MobiLink Synchronization Technote: Mobile Enterprise, Database Management - Sybase Inc*”  
<<http://www.sybase.com/detail?id=1038632>>
- [4] IAnywhere.com. August 2007. “*Choosing a replication technology*”  
<[http://www.iAnywhere.com/developer/product\\_manuals/sqlanywhere/0902/en/html/dbfgen9/000040.htm](http://www.iAnywhere.com/developer/product_manuals/sqlanywhere/0902/en/html/dbfgen9/000040.htm)>
- [5] Sybase Inc. August 2007. “*Developer/SDN Mobile & Wireless Developer SQL Anywhere Studio MobiLink - Sybase Inc*”.  
<<http://www.sybase.com/developer/mobile/sqlanywhere/mobilink>>
- [6] Werthne, Hannes; Ricci, Francesco. 2004. “*E-commerce and Tourism*”  
Communications of the ACM. ACM 0002-0782/04/1200
- [7] Ministry of Tourism, Malaysia. September 2007.  
<[http://www.tourism.gov.my/tourismbiz/media\\_centre/statistic.asp](http://www.tourism.gov.my/tourismbiz/media_centre/statistic.asp)>

[8] Scott M. Thede. 2004, "*An introduction to generic algorithm*", Consortium for Computing Sciences in College. September 2007.

<<http://portal.acm.org/results.cfm?coll=ACM&dl=ACM&CFID=30280262&CFTOKEN=629040>>

[9] "*Model-View-Controller- Wikipedia, the free encyclopedia*". September 2007.

<<http://en.wikipedia.org/wiki/Model-view-controllers>. . >

[10] Sybase Inc. September 2007. <[http://www.iAnywhere.com/products/sql\\_anywhere.html](http://www.iAnywhere.com/products/sql_anywhere.html)>

[11] Anil Nori. June 2007. "*Mobile and Embedded Databases*". Proceedings of the 2007 ACM SIGMOD international conference on Management of data SIGMOD '07.

[12] Jim Geier. September 2007. "*Applications of Wireless Networks*".

<[http://www.wireless-nets.com/papers/wireless\\_choosing\\_database.htm](http://www.wireless-nets.com/papers/wireless_choosing_database.htm)>

[13] "*Traveling Salesman Problem Using Genetic Algorithms*". September 2007

<<http://www.lalena.com/AI/TSP/>. >

[14] Wikipedia. "*Traveling Salesman Problem*". September 2007.

< [http://en.wikipedia.org/wiki/Travelling\\_salesman\\_problem](http://en.wikipedia.org/wiki/Travelling_salesman_problem)>

[15] Wikipedia. "*Dijkstra's algorithm*". ". September 2007.

<[http://en.wikipedia.org/wiki/Dijkstra's\\_algorithm](http://en.wikipedia.org/wiki/Dijkstra's_algorithm)>



[16] Wikipedia. "*A\* Search Algorithm*". September 2007.

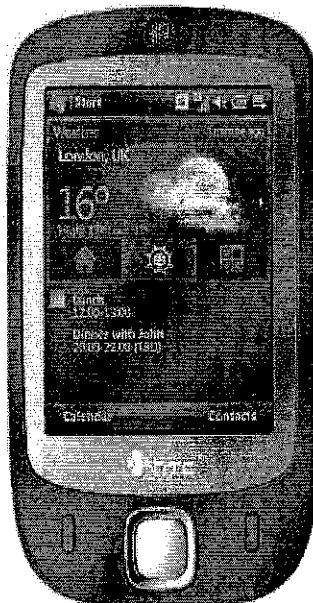
<[http://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](http://en.wikipedia.org/wiki/A*_search_algorithm)>

## APPENDICES

### Pocket PC 2003 Emulator



### HTC Touch



## Specification of the HTC Touch

|                  |   |
|------------------|---|
| Processor        | TI's OMAP™ 850, 201 MHz   |
| Operating System | Windows Mobile® 6 Professional  |
| Memory           | ROM: 128 MB<br>SDRAM: 64 MB DDR   |
| Dimension        | 99.9mm (L) X 58mm (W) X 13.9mm (T)  |
| Weight           | 112g with battery   |
| Display          | 2.8" LCD touch screen with backlight<br>240 X 320 dots resolution with 65,536 colors  |
| Network          | GSM/GPRS/EDGE Tri-band: 900, 1800, 1900 (The device will operate on frequencies available from the cellular network.)                           |
| Device Control   | HTC TouchFLO™<br>5-Way navigation control   |
| Connectivity     | Bluetooth® 2.0<br>Wi-Fi® IEEE 802.11 b/g<br>HTC ExtUSB™ (11-pin mini-USB and audio jack in one)   |
| Camera           | 2.0 megapixel CMOS color camera   |
| Audio            | Built-in microphone and 3-in-1 speaker<br>Windows Media Player supported formats: AAC, AAC+, eAAC+, MP3, WMA, WAV, QCELP, MPEG4, AMR-NB, AMR-WB |

|                |  |
|----------------|--|
| Battery        | Rechargeable Li-Ion battery<br>Capacity: 1100 mAh<br>Standby time: Up to 200 hours<br>Talk time: Up to 5 hours |
| Expansion Slot | microSD™ memory card (SD 2.0 compatible)   |
| Adapter        | Voltage range/frequency: 100 ~ 240V AC, 50/60Hz<br>DC output: 5V and 1A  |

## The Website

File Edit View History Bookmarks Tools Help

http://www.vilacem.com/mta/

Getting Started Latest Headlines FYR Career **B** New Bookmark Study Business Reading UTP WebMail Sing Jobs Tech

Definitio... Loading... HAB2043... Results (... W Dijkstra's a W A\* search W Greedy alg www.fibc

# mobile tourist assistant

Home | about | contact

**Mobile Tour Assistant is your helpful companion on your journey to explore and enjoy Kuala Lumpur. The package features the followings:**

- **Tourist Kits:** provide you necessary tourist information about Kuala Lumpur (KL) such as its Best Place, Events and Festivals, Travel Tips and Contact Phone Numbers. Tourist Kits also bring you multi-media with images of places and maps of KL.
- **Itinerary Planner:** We understand the arduousness of scheduling a trip, especially when you are under time constraints. Itinerary Planner is in place to help you lessen that burden. Just give your time, date, start and end location and select the preferences of tourist attractions. The rest of the work will be handled by Itinerary Planner. Using an intelligent travel scheduler technique, Itinerary Planner has been devised in a way that maximizes your satisfaction under your time constraints.
- **Hotel Recommender:** If you are wondering what hotel to stay over during your visit, just click on Hotel Finder in our package, specify your choices and the list of recommended hotels will be presented to you.

The package is available for **FREE** download.  
Visit our website frequently for the latest updates!

[Download App](#) [Download Database](#)

*\* Full App including database file*