<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Car park reservation system</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Yip, Kai Wing (葉啓榮)</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>Yip, K. W. (2012). Car park reservation system (Outstanding Academic Papers by Students (OAPS)). Retrieved from City University of Hong Kong, CityU Institutional Repository.</td>
</tr>
<tr>
<td><strong>Issue Date</strong></td>
<td>2012</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/2031/6751">http://hdl.handle.net/2031/6751</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>This work is protected by copyright. Reproduction or distribution of the work in any format is prohibited without written permission of the copyright owner. Access is unrestricted.</td>
</tr>
</tbody>
</table>
City University of Hong Kong
Department of Computer Science

BSCCS Final Year Project Report 2011-2012

(11CS126)

Car park reservation system

(Volume 1 of 1)

Student Name : Yip Kai Wing
Student No. : 
Programme Code : BSCCS
Supervisor : Dr. Lee, Chung Sing Victor
1st Reader : Dr. Li, Shuai cheng
2nd Reader : Prof Li, Qing
Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

Car park reservation system

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name: Yip Kai Wing  Signature: ____________________________

Student ID: ________________  Date: 10th April 2012
Abstract

Transportation is one of the well-known problems in metropolitan areas around the world. As revealed by the Transport Department, the total number of vehicle’s registration keep raising year on year since 2006, with a latest number of 682,777 total registrations which includes 465,453 registrations for private cars were recorded on September 2011. On average, it is a very high people per vehicle ratio considering we have approximately seven million people living in Hong Kong. With such a high ratio, it puts a high pressures and requirements on facilities supporting driver’s daily needs, it brings in a challenging situation for company to provides a good quality of service to their customers.

Although there are many different kinds of offer provided by car park company to smoothen and relieve this situation, for instance, different kinds of payment method, park and ride scheme, various award and reward schemes as well as free parking scheme, it is still not a very satisfactory experience and hard to fulfill drivers’ immediate parking need.

Our prototype introducing membership system, overbooking strategy and instance booking for car park reservation, car park operators can adjust the parameters such as available slot, overbook setting and reservation setting, which is in another words, transforming the physical parking spaces into logical spaces, and transform physical time into logical time. By overlapping virtual spaces and virtual travel time, it has potential to maximize the parking space utilization and generate more revenue. From the user’s point of view, they can reserve a guaranteed car park space in advance with minimal cost of instance reservation, and receive membership discount and benefit along with a convenience system for searching and making reservation which reduce their unnecessary queueing and cycling time looking for an available car park space everywhere.

Ultimately, saving in time, fuel and a better parking experience as well as enhancing the quality of service could be achieved by using this system and lead to a win-win situation for both parties.
Acknowledgement

I would like to take this opportunity to express my sincere thanks to Dr. Lee, Chung Sing Victor who is my supervisor of this project for his helpful guidance on this project.

Throughout the year, he had spent a lot of time having meetings and discussions with me with his professional knowledges and rich experiences. He was not only assisting me to facing and tackle the problems but also guiding me to analyse and approach for a better solution, he had also providing very valuable comments base on my output and put me back onto the right track even when it comes to the very last minute. Without his inspiriting comments, this project would become very inrealistic and impractical for general use.

I would like to express my appreciation to the Department of Computer Science for offering us various resources including hardware, software.

Last but not least, I would like to thanks to all my friends who share their driving experience with me through the year.
# Table of Contents

Abstract .................................................................................................................................. 3

Acknowledgement ................................................................................................................. 4

1. Introduction ....................................................................................................................... 7
   1.1. Background information .............................................................................................. 7
   1.2. Existing problems ......................................................................................................... 7
   1.3. Aims and objectives ...................................................................................................... 9
   1.4. Scope ............................................................................................................................. 9

2. Literature Review ............................................................................................................ 10
   2.1. Related technologies .................................................................................................. 10
      2.1.1. Mobile platforms ................................................................................................. 10
      2.1.2. Mobile application development .......................................................................... 12
         2.1.2.1. iOS software development kit (iPhone) .............................................................. 12
         2.1.2.2. Android software development kit .................................................................. 13
      2.1.3. Mobile browsers .................................................................................................. 14
      2.1.4. Mobile frameworks .............................................................................................. 16
         2.1.4.1. jQTouch ............................................................................................................ 16
         2.1.4.2. jQuery Mobile .................................................................................................. 17
         2.1.4.3. Sencha Touch .................................................................................................. 18
      2.1.5. Identification and recognition technologies ......................................................... 19
      2.1.6. Barcode ................................................................................................................ 20
      2.1.7. Quick Response (QR) code .................................................................................. 21
   2.2. Related web based applications ................................................................................. 22
      2.2.1. Car.com.hk ............................................................................................................ 22
      2.2.2. Parkinghk.com ..................................................................................................... 23
      2.2.3. Wilsonparking.com.hk .......................................................................................... 24
      2.2.4. The Link Real Estate investment Trust ................................................................. 25

3. Requirement analysis and specification ....................................................................... 27
   3.1. Terminologies ............................................................................................................. 27
   3.2. Use case diagram ....................................................................................................... 28
      3.2.1. Actor description ................................................................................................. 29
   3.3. Functional requirement ............................................................................................... 29
   3.4. Non-functional requirement ....................................................................................... 33

4. System design .................................................................................................................. 34
   4.1. System Overview ....................................................................................................... 34
   4.2. System architecture .................................................................................................... 36
   4.3. Three-tier architecture ............................................................................................... 39
   4.4. Restful Web Service .................................................................................................... 41
   4.5. Data communication in JSON .................................................................................. 42
      4.5.1. Skeleton of JSON object ....................................................................................... 43
4.6. Package Diagram ............................................................................................................. 44
4.7. Database design ............................................................................................................. 45
4.8. System implementation ................................................................................................. 46
  4.8.1. Multiple hierarchies ................................................................................................. 46
  4.8.2. Car park setting ....................................................................................................... 48
  4.8.3. Member management ............................................................................................... 50
  4.8.4. Application parameters ........................................................................................... 51
  4.8.5. Reservation ............................................................................................................. 52
    4.8.5.1. Overview of reservation process ......................................................................... 52
    4.8.5.2. Overbooking ....................................................................................................... 53
    4.8.5.3. Reservation requests ......................................................................................... 56
    4.8.5.4. Reservation modification ................................................................................... 59
    4.8.5.5. Check-in ............................................................................................................ 60
    4.8.5.6. Check-out .......................................................................................................... 61
    4.8.5.7. Reservation log ................................................................................................. 63
    4.8.5.8. Mobile client ...................................................................................................... 64
    4.8.5.9. Registration ...................................................................................................... 65
    4.8.5.10. Profile and vehicle management ..................................................................... 66
    4.8.5.11. Car park searching ......................................................................................... 69
    4.8.5.12. Car park searching ......................................................................................... 73
5. System testings .................................................................................................................. 74
  5.1. Test cases ..................................................................................................................... 74
  5.2. Performance evaluation ............................................................................................... 85
6. Limitations ....................................................................................................................... 89
  6.1. Pre-defined membership scheme and car park settings ............................................ 89
  6.2. Car park services and payment type ......................................................................... 89
  6.3. Overbooking setting and small car park ................................................................. 90
  6.4. Accuracy of map view in mobile client ...................................................................... 90
7. Difficulties ....................................................................................................................... 91
8. Future extension ............................................................................................................... 92
  8.1. Advance booking ....................................................................................................... 92
  8.2. Carpark profile matching ......................................................................................... 92
  8.3. Integration with reward system ............................................................................... 92
  8.4. Another approache ................................................................................................... 93
9. Summary ......................................................................................................................... 94

References .......................................................................................................................... 95

Appendix ............................................................................................................................ 97
  1. Monthly log ................................................................................................................... 97
1. Introduction

1.1. Background information
Transportation is one of the well-known problems in metropolitan areas around the world. There is no doubt that it is one of the most basic necessities of life, millions of people travel back and forth to work or school on a daily basis. According to the statistic result released by Census and Statistics Department, as of 2010, Hong Kong has a density of 6,349 people per sq. k.m. [1]. With such a high density of population, the transportation system could be extremely busy and could be observed easily, especially during rush hour and at major business areas, traffic congestion happened very often and is very common.

In Hong Kong, we can have a wide range of choice for public transport such as railways, buses, minibuses as well as taxis, and the government always encourages people make use of public transport instead of driving their own cars. As revealed by the Transport Department, the total number of vehicle’s registration keep raising year on year since 2006, with a latest number of 682,777 total registrations which includes 465,453 registrations for private cars were recorded on September 2011 [2]. On average, it is a very high people per vehicle ratio considering we have approximately seven million people living in Hong Kong. With such a high ratio, it puts a high pressures and requirements on facilities supporting driver’s daily needs, for example, road construction, gas station and parking., it also bring in a challenging situation for company to provides a good quality of service to their customers.

1.2. Existing problems
Along with the sustained economic recovery in Hong Kong these years [3,4,5], and the rise of information technology, the quality of life has been improving and people begin to demand more. To improve the quality of life, one might easily think of and start with the four basic necessities of life. For Chinese community, we have a old saying – “clothing, food, housing and transportation” to represent the most basic neccesities, which is basically truth worldwide. While clothing and food are easier to achieve, and housing is way too expensive, many people, especially for family, would consider purchasing of a private car which is relatively affordable. Even though driving might not always be an enjoyable activity in Hong Kong, it could bring us lots of convenient and saving lot of time for travelling.
The very first common problem in which most drivers are facing would be parking issues. In Hong Kong, there are typically two type of public parking facilities provided – parking meters and car parks. There are about 18,000 parking spaces provided by parking meters, which are located on-street where they do not obstruct traffic [6], and an estimated 631,000 spaces provided by various car park, of which 203,000 are for public use, and 428,000 are designated for private use [7].

According to the study conducted by Transport Department, the forecast demand-supply ratio for 2011 of ownership-related car space is 0.91 [8]. Although it sounds to be a good ratio, it might not reflect the actual situation and needs for daily drivers as it was an overall ratio including over-night parking, coaches and goods vehicles. Currently, there are some major parking operators in Hong Kong such as Wilson Parking [9], The Link [10], and Sino Parking Services Limited [11]. There are also many car parks located everywhere and operated by various companies, e.g. Hang Lung Properties, shopping malls’ owner as well as some private company. However, most car parks did not employ an efficient and systematic system. With such a high volume of parking spaces usage at urban area, immediate needs of parking service is often not available, time is being wasted for drivers in queuing, circling, and searching for an available parking space at surrounding areas. It is a very ineffective and frustrated situation for driver.

Although there are many different kinds of offer provided by car park company to smoothen and relieve this situation, for instance, different kinds of payment method, park and ride scheme, various award and reward schemes as well as free parking scheme, it is still not a very satisfactory experience and hard to fulfill drivers’ immediate parking need. Based on the above considerations, there would be value in study these issues and developing an computerize and systematic solution to assist driver from numerous way, such as allowing drivers to search for an available car park nearby, and reserve a guaranteed parking space at a convenient location at their concern in advance. From the operator and business point of views, it could also help to manage their car park’s user through membership scheme, optimize and predict the parking spaces utilization through statistic result, eventually, building up a sense of belonging and loyalty for their customers, it has a potential to generate more revenue and lead to a win-win situation for both parties.
1.3. Aims and objectives

This project aims to develop a car park reservation system to address and tackle the above daily problems. This system will consist of a membership system which allow car park operators to manage their carparks and their customers’ activities, it also includes a mobile-based client for car park searching and making reservation, which would be used to search for and locate the nearby available car park with geometry information as well as reserving for a guaranteed parking space in advance. This project will be focusing on how reservation could be done, and how drivers’ parking experience could be improved for the above situation through employing and using software system with latest technology. Ultimately, saving in time, fuel and a better parking experience could be achieved by using this system.

1.4. Scope

This project will be developing a car park reservation system which will be running on a smart phone, and supported by a back end web service. The scope of this project is defined as follow:

- Investigate the value and the benefits of developing a system for assisting drivers’ parking needs.
- Study the feasibility and different options to enhance the drivers’ experience and efficiency of parking.
- Explore geometry application development and location-based service.
- Study the application development process on smart phone and using latest technologies and various software development techniques.
- Design and develop prototype system which includes a backend system and a mobile client running on smart phone.
- Evaluate and review the strengths and weaknesses of the project outcome, seek and identify for further improvements.
2. Literature Review

In Hong Kong, we have thousands of car parks located everywhere, including large, medium and small sized car park. Most car parks provides a wide range of services and charging models to fulfill customers’ needs, the most common services are monthly, daily and hourly parking reservation service, where monthly and long term parking service are provided via prior registration, daily and hourly services are provide in a drop-in/walk-in and first come first serve (FCFS) manner. Currently, there are some car web-based park searching systems existed, however, there is no online car park reservation system existed in Hong Kong. The following are some related topics that have been studied, including related technologies and related applications.

2.1. Related technologies

2.1.1. Mobile platforms

Nowadays, almost everyone owns a mobile phone in Hong Kong. In recent years, smart phone devices become more popular and more affordable. They are feature-rich and possessing
different capabilities such as web browsing, Wi-Fi connections, and media playback. According to the survey conducted by The Nielsen Company, as of Q1’ 11, 37% of mobile consumers owns a smart phone [12].

The current smart phone market are mainly share by four competitor while each of them having their own operating system, namely Research In Motion (RIM) BlackBerry, Microsoft Windows Mobile, Apple iPhone and Google Android, they occupied almost 90% of the market share. While considering the OS, hardware features and popularity, Apple iPhone and Google Android stand out from the competition even though neither of them is sharing the highest ratio in the market.

According to the data analyzed by The Nielsen Company in Q1’ 2011, Android dominates the smartphone market share of 36 percent in the United States [13]. In terms of software development, both iPhone and Android provided software development kit (SDK) for developer to develop application running on their OS and have the facilities to implement all the functions in this project. Besides, Android is being used in a wide-range of mobile devices, and iPhone is using innovative design but single device per generation.

Based on the above reasons, it seems that an all-rounded application which supports both IOS and Android could benefit the most for developer, business operator, and application user., i.e. drivers, and covering majority of the consumers.

Figure 2. Smartphone Market Share, The Nielsen Company
2.1.2. Mobile application development

In general, mobile application development is similar to desktop application development, but using different software tools, languages and technique along with different architecture and resource constraints. It can be spitted into two approaches – native application and web-based application.

2.1.2.1. iOS software development kit (iPhone)

The iOS software development kit [14] is developed by Apple Inc. and available to third-party development for developing iOS application for iPhone and iPod touch devices since March 2008. It consists of full set of tools such as Xcode, Interface builder, Instruments and iOS simulator to assist programmer making iOS application. The official development environment is Xcode, and the programming language is Objective-C.

In short, the development process through the entire software development cycle is smooth, however, MacOS is the only operation to support Xcode, which mean the developer must own an MacBook and iPhone beforehand. Also, debugging application on actual device or publishing of the software would require user to enroll for iOS developer program [15], which is not free and costing USD $99 per year.

Figure 3. Screenshot of Xcode
2.1.2.2. Android software development kit

Android software development kit [16] is the software tools first release its beta in November 2007 and developed by Google, which allow developer to make software to run and execute on mobile devices running Google Android OS.

Unlike iOS software development kit, the software does not bundle with a single development environment; it can be run and executed in console mode. It’s official supported development is Eclipse IDE, and provided as a plugin to ease developer needs to start writing Android application. It also provides full set of tools to assist development needs such as debugger and profiler, and the programming language is Java. Besides, one major differences comparing to iOS SDK is that, it can be used on Windows, MacOS, as well as Linux, and its free of charge to publish software to the market.

![Android SDK running on MacOS](image)

*Figure 4. Android SDK running on MacOS*
2.1.3. Mobile browsers
Among all the popular mobile devices and mobile OS, although they all shipped with stock browser, user can have the option to install other third parties browser to fit their needs, such as Opera, NetFront, and Skyfire. According to the top 9 mobile browser statistic results from StatCounter Global Stats as shown in figure 5 [17], iPhone and Android together played a major role and occupied over 60% of mobile traffic in Hong Kong from January 2011 to January 2012.

Figure 5. Top 9 mobile browsers in Hong Kong
However, if we take a look at the top 9 mobile browser recorded worldwide within the same period as shown in figure 6 [18], it shows that other players including BlackBerry’s, Nokia’s and Opera’s browser on the market are also playing a competitive role.

Even though iPhone and Android is dominating the mobile world, it seems that an all-round application which could be run and execute on these browsers and/or its platform would be a great step to achieve in order to facilitate as much user as possible, and could be more useful for this project.

Figure 6. Top 9 mobile browsers Worldwide
2.1.4. Mobile frameworks

There are over hundreds of different framework available on the market for developing mobile applications. By implement the mobile frontend using web-based approach, we can get rid of the boundary from one single device as discussed in previous section s. In this section, we will reviews three of the most popular choices which had been considered to use for this project.

2.1.4.1. jQTouch

jQuery is currently the most popular JavaScript library for web development. And jQtouch is a jQuery plugin built by David Kaneda, which is used for mobile web development and can be run on iPhone, iPod touch, Android and other WebKit enabled devices [19]. It simulates looks and feels of native iPhone application by using latest technology and technique in HTML, CSS and JavaScript. It featured with native WebKit animations, swipe detection, flexible theming support and also comes with extensions support.

However, during our prove of concept (POC) process, the overall performance is the worst on medium size device which is expected to be the mainstream of devices on the market, especially on animations and user input detection, when comparing to other.

![Figure 7. Screenshot of jQTouch](image)
2.1.4.2. jQuery Mobile

jQuery mobile is a unified, touch-optimized HTML5-based web development framework developed by The jQuery Project [20], which is used for mobile devices. Unlike jQTouch, it is built on top of the popular jQuery and jQuery codebase. It also provides theme which looks like native iPhone application, and coding in HTML, CSS and JavaScript. As it does not require WebKit for execution, it supports all the major smartphone and tablet platforms including iOS, Android, BlackBerry, Bada, Windows Phone, Palm webOS, Symbian, as well as MeeGo.

During our POC process, the overall performance is good, it is easy to learn and providing full set of general UI elements support, however, it suffering from a slightly unresponsive scrolling when viewing a list of information, it also suffering from randomly jumping address bar and screen flickering issues when tested on iOS and Android devices.

![Screenshot of jQuery Mobile](image)

*Figure 8. Screenshot of jQuery Mobile*
2.1.4.3. Sencha Touch

Similar to the previous framework, Sencha Touch developed by Sencha Inc. which make uses of HTML5, CSS3, and JavaScript and provides maximum degree of flexibility and optimization [21]. It allow developers to build mobile web application which look and feel native on different mobile devices. It is built on top of another popular JavaScript Library ExtJS.

Unlike previous framework, as it is a mobile JavaScript framework, developing application using Sencha Touch will be involving heavy coding in JavaScript while the others are coding in HTML as major programming language, and its learning curve is a lot deeper.

However, it gives the best result through the POC process, and provides acceptable performance and better animations without the jumping screen issues, it is also very responsive in general which will not affect the user experience a lot. It also provides better documentation to follow and thus, it is likely to be selected to use in this project.

![Sencha Touch running on Android and iOS](image)

*Figure 9. Screenshot of Sencha Touch running on Android and iOS*
2.1.5. Identification and recognition technologies

According to HTS Hi-Tech Solutions, one of the leading license plate recognition companies, “license plate and vehicle recognition (LPR/ANPR) detects and reads vehicle license plates for parking, access control, traffic, tolling, surveillance, law enforcement, security and other logistics applications” [22]. Currently, license plate recognition is commonly used. It can use existing closed-circuit television or camera to capture and recognize the vehicle number printed on the license plate.

As infrared light camera is being used, it is able to capture information under different light source and illumination situation with very high successful rate. Also, some designated camera can capture vehicle speeds and distances as well. With such technology exists, car park management or reservation system will be able to extract necessary information for further processing when automation is needed.

Figure 10. Camera located at car park entrance

Figure 11. Screenshot of license plate recognition system
2.1.6. Barcode

According to Association for Automatic Identification and Mobility, barcode was invented more than 50 years ago, its common usage are covered by international standards. Barcode is a type of symbology and a machine-readable representation of data. It could encode, decode, and represent numbers, letters as well as a few punctuation characters in a series of parallel lines with different widths in one dimension [23]. It is widely used in Point-of-Sale and manufacturing environment, it could be found on almost every packaging of products in the market. In Hong Kong, most grocery stores use barcode reader to identify the barcode on the goods for the check-out process to shorten check-out time and avoid error. It can also be found on almost every receipt and invoice letter.

It is a very useful technology for various kinds of identification such as receipt and coupon, and is widely used around the world. For car park operation, it might be possible to use barcode for ticketing or receipt identification purpose.
2.1.7. Quick Response (QR) code

Denso Wave Incorporated defined QR code as a kind of two-dimensional symbology. It is able to represent information in the vertical direction as well as the horizontal direction. QR code can carry up to several hundred times the amount of data while comparing with one dimensional representation like barcode [24]. Currently, QR code is very popular in Japan and mainly being used with a combination of mobile device in everywhere. This trend begins to spread out in Hong Kong nowadays. It could be used to identify product and represent more information.

For example, PCCW Hong Kong Ltd. used QR code technology for confirmation and identification of movie ticket purchased; The Kowloon Motor Bus Co. used QR code to represent the advertisement’s website on their outdoor advertising campaign; some local magazine also printed with QR code on their cover page. It is also being used in other countries as well as some websites on the internet. Google Marketplace is one of the example website which used QR code to represent the web location of an Android application. It becomes more popular to use as a commercial and advertisement media in recent year. It can also provide a visual linkage between mobile devices and web addresses using camera as QR code scanner on mobile devices. Similar to barcode, it might be possible to use barcode for ticketing or receipt identification purpose.

Figure 14. Latest QR code implement in Hong Kong – using QR code for movie ticket

Figure 15. QR Code located at City University of Hong Kong
2.2. Related web based applications

2.2.1. Car.com.hk

“Car.com.hk” is an online website provided and managed by two Local website - discuss.com.hk and uwant.com. It is an online platform mainly used for second-hand vehicle’s buying and selling for the community. It provided wide range of information including promotions and some related news information. It also provides car park information and searching functionality in one of its sub-section. User can search for a car park using name or address; it can also filter the result by different car park type and charging mode.

Although the lists of available car parks are limited, the results are visualized clearly with appropriate icons indicating and comparing the search result. Google map is also integrated within the system; user can locate the car park via a map view. It is a very useful feature, especially for drivers looking for nearby car park, for experienced driver, they might able to recognize most nearby area and street names, and seek for a suitable car park at a convenience manner.

![Figure 16. Screenshot of car.com.hk](image-url)
2.2.2. Parkinghk.com

“Parkinghk.com” is an associate company of Hong Kong City Parking. It is a web application providing a platform for users to search for car park locations, parking space to lease and sales across various locations in Hong Kong. It also provided analyses results and reviews on price movement, monthly rent and parking space demand and supply in Hong Kong for references.

It provided a more detailed sub-section comparing to other related web applications, such as filtering of car parks spaces by beaches, swimming pools, shopping malls, meter parking, parks and motorbikes. This is a very convenience and user oriented design, which allow user to filter information and locate information very quickly by their intention. It also provides essentials information on most car park including their address, phone number, number of parking spaces and standard charges. It also provides reward scheme information for various car park, for example, a reward scheme of “Enjoy 2 hours free parking for spending $600 or above” provided by Festival Walk could be found and located easily on the system within the shopping malls section.

![Figure 17. Screenshot of parkinghk.com](image)
2.2.3. Wilsonparking.com.hk

“Wilsonparking.com.hk” is the official web site of Wilson Parking. It provides all the information of car parks under its management. In the web site, there are “easy locator” and “advanced locator” for user to search for car park with different criteria. The easy locator is a simplified one, which allows user to locate a car park by district easily, and provides an overview list of existing car park, while the advanced locator provides a full featured table view of its car park, with filtering option of different services provided, user can easily locate a car park by services such as hourly or day park.

Both searching function includes a detail view of each car park with services, rate, phone, payment method and photo of each car park; it also intergraded with Google map and provide a map view for navigation. These searching functions are well-designed and provide convenience services enabling user to search for its car park within a short period.

![Figure 18. Easy locator of wilsonparking.com.hk](image)

![Figure 19. Advance locator of wilsonparking.com.hk](image)
2.2.4. The Link Real Estate investment Trust

The Link Real Estate Investment Trust ("The Link REIT") is a real estate investment trust in Hong Kong, it is also currently the largest in Asia [25]. In recent years, it owns and manages 182 properties in Hong Kong, most of them are shopping malls and car parks located on the doorstep of over 40% of Hong Kong’s households.

In their official website, user can look up their properties by simply click on the list of location on the left hand side of the website, it list all type of property including car park, shopping center and estate retail. Besides, user can perform specific search for their managed properties by using ‘Property Locator’ as shown in the figure 13 and 14, search can be performed with options of district, category, property and keywords. Search results will be presented in a brief list view. User can also check for detail information by clicking “read more” button.

![Property locator of The Link](image)

*Figure 20. Property locator of The Link*
Comparing with other system, its search function is more simplified and harder to find a specific result, which might due to large list of properties the company currently managing. In the detail view, it also provide extra information such as management office address and phone number, it also provide a link to map view as well as download of leasing form. It also provides some picture of the surround area which will be very useful for user and driver to locate the car park.

![Hung Hom Estate Retail](image)

*Figure 21. Detail view of The Link property locator*
3. Requirement analysis and specification

3.1. Terminologies

Before we start discussing our prototype in details, we would like to define some terminologies first. During our study and observation, we found that different car park are using different terms to representing for the same things, same actions or same services, such as “move-in” comparing to “check-in”, “parking space”, “parking slot” as well as “parking bay”. The following table summarize some terminologies we used during reservation process in this prototype, i.e. The word being used in this document and in the user interface of the prototype:

<table>
<thead>
<tr>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parking space / slot</td>
<td>The physical space which a vehicle will be parking and staying in a carpark.</td>
</tr>
<tr>
<td>Hourly charge</td>
<td>The charges for parking per hour.</td>
</tr>
<tr>
<td>Check in</td>
<td>The vehicle arrive the carpark and make use of the parking service.</td>
</tr>
<tr>
<td>Check out</td>
<td>The vehicle leave the carpark and finish the usage of the parking service.</td>
</tr>
</tbody>
</table>

*Table 1. Terminologies used in this report*
3.2. Use case diagram

The prototype of our car park reservation system consists of a set of function to assist in carpark reservation and management. The following use cases as shown below provides a better understanding and overview of features and user requirements included, as well as the actors involved. It also represent the function requirements of our prototype.

![Use case diagram of car park reservation system](image)

*Figure 22. Use case diagram of car park reservation system*
3.2.1. Actor description

As shown in the user case diagram, our prototype consists of three main actors. The following table summarize their roles involved in our system.

<table>
<thead>
<tr>
<th>ID</th>
<th>Actor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC-1</td>
<td>Member</td>
<td>Member is the car owner who is of the primary user of the system, their major goal is to find and reserve car park slot through using the car park reservation system. They can also cancel reservation and view car park reservation information by using the system.</td>
</tr>
<tr>
<td>AC-2</td>
<td>Car park management user</td>
<td>Car park management user is the property owner or their employee who play administrative role in the system. They can manage car park parameter such as number of slot and extra slot. Also, they can manage reservation and member, as well as check-in, check-out vehicle.</td>
</tr>
<tr>
<td>AC-3</td>
<td>SMS service provider</td>
<td>External Actor who assist in handling and sending SMS acknowledgement/confirmation to system member.</td>
</tr>
</tbody>
</table>

Table 2. Actor description

3.3. Functional requirement

Our car park reservation system prototype can be divided into two parts – a mobile client for car park member and a web system for car park management user (i.e. car park company). Each of them involving different functionalities in our system as illustrated in the use case diagram.

For car park members:

- Manage user profile

Once the user is registered, they can login our system and manage their profile. They can manage their personal details such as username, password, name, mobile number, and car park preferences. They can also view their current membership group and manage their vehicle details in our system.
• **Search for car park**

User can search for a car park and view its information such as current utilization, available parking spaces, charging rate, phone and address. These information are essential for drivers to decide which car park to use. Considering we have user using various kind of mobile devices, our system provides two searching mode for different usage. The first one is text mode which can search by keyword or area, and list out car parks matched in our system. The second one is map view mode which integrated with Google map, user can have a better overview of car park location and availability nearby their current location. As drivers usually concerned about a certain area rather than a specific car park, apart from inputting searching for matching keywords in text mode, we also allow inputting of a custom keyword search for a destination which is in general, supporting all kind of addresses or locations, and view the car parks nearby their destination on the map. Besides, we also includes advanced features to provide driving distance and driving time estimation, as well as showing up the suggested driving path from their current location or destination on the map, it will be very useful especially when the driver is unfamiliar with the surround area or concerning about the distance or fuel.

• **Manage reservation**

Once the user located a car park, they can make an reservation and receive a guaranteed parking space in advance. Once a reservation is confirmed, our system will issue a SMS confirmation message to the user. This helps user to plan ahead for their trip, and avoid unnecessary time wasting for queueing and searching for parking space. Besides, we allow user to manage their reservation in our system such as modifying or cancelling a reservation within pre-defined allowed time, for each update or cancellation, our system will also issue a SMS acknowledgement to the user. Besides, by specifying their leaving time for the reservation, user can also enjoy their member benefit such as reservation discount.

**For car park management user:**

• **Manage reservation**

As some member might request for a reservation by other means such as phone call or in person, management user can also make an reservation on behalf of the member. Similar to making an reservation by member, the reservation can specify an leaving time to enjoy member benefit. As they know the current utilization of the car park beforehand, when they know a reservation is confirmed will be likely to be accepted in the system, they can optionally select issuing a SMS
confirmation or not. They can also update or cancel a reservation on behalf of the member in a similar manner, their SMS acknowledgement is also optional. In addition, they can select to charge for the cancellation. If the car park is full and the reservation cannot be made, the request will be recorded, once there is an available space within the pre-defined expired time, the system will issue a SMS acknowledgement to the selected user and let them decide to make a new reservation or not, this reservation recall also applied to making reservation by member using mobile client.

- **Manage reservation**

  Once the user located a car park, they can make a reservation and receive a guaranteed parking space in advance. Once a reservation is confirmed, our system will issue a SMS confirmation message to the user. This helps user to plan ahead for their trip, and avoid unnecessary time wasting for queueing and searching for parking space. Besides, we allow user to manage their reservation in our system such as modifying or cancelling a reservation within pre-defined allowed time, for each update or cancellation, our system will also issue a SMS acknowledgement to the user. Besides, by specifying their leaving time for the reservation, user can also enjoy their member benefit such as reservation discount.

- **Manage member**

  User is able to manage the member’s profile including their personal contact, car park preferences, running balance, credit balance, current rating as well as their membership.

- **Manage carpark**

  In our prototype, we introduced a wide range of parameters for each car park which model the carpark such as car park spaces, number of monthly user, day time charges, night time charges, reservation setting, membership discount, overbook ratio, etc. These parameters are used to calculate the current utilization and determine whether a reservation can be made. User is able to manage and adjust these setting in our system.

- **Check-in vehicle**

  Once a vehicle is arriving the carpark, user is able to view the current reservation list in the system and allow the member to check-in to the car park. As overbooking strategy is implemented in our prototype, the system will determine the current utilization and decide whether the reservation can be check-in or not. Upon a successful check-in, the system will also
record its arrival time for statistic usage. If a check-in is unsuccessful, the member can choose to wait or cancel the reservation, when the user decided to cancel the reservation, we assume that some sort of compensations such as credit balance or coupon, or free cancellation scheme will be provided by the car park company in such case, and it can be applied by the car park management user by cancelling the reservation with charges calculation opt-out. We also accept check-in of anonymous vehicles which would affect the car park’s current utilizations, but we only log and keep track of its one-off activities without detailed information such as driver personal details or past activities.

• Check-out vehicle

Once a vehicle is leaving the carpark, user is able to view the current check-in vehicle list in the system and allow the member to check-out of the car park. Leaving time will be recorded, charges and running balance will be calculated. If the reservation had specified an end time and the member is leaving on time, the system will automatically calculate the charges with discount set in the car park setting. For anonymous user, it is similar to check-in vehicle which we only log and keep track of its one-off activities.
3.4. Non-functional requirement

Apart from functional requirements, there are also many non-functional requirements in our concerns. Our major concerns for this prototype is as followings:

- **Availability**
  As wireless network connection might not be available for all users using the system, and the connectivity can be affected or even lost by the environment. The system should be able to make use of the browser’s cache to cache the retrieved data such as car park information and images whenever possible. When the data is cached, it should be able to open the application without required to reload from the internet again. It also improved the latency and reduced the connection cost for displaying data on the system comparing to retrieve and display the data in real-time.

- **Communication cost**
  Data transmission through telecommunication service (e.g. 3G network) could be expensive. The system should able to reduce the communication cost by reducing and minimizing the size of data being transferred. It could be done by using appropriate data structure to reduce communication overhead (e.g. JSON vs. XML), data compression such as zip or gzip, or allowing user to retrieve data selectively.

- **Usability**
  The system and the user interface should be easy to use and easy to understand by most users. In order to allow driver to use the system within short period of timeframe, such as stopping the car aside on the road, the complexity of the mobile frontend should be minimized. Moreover, the user interface should be designed to be simple and suitable to display concrete information on the limited screen of a mobile device.

- **Platform and browser compatibility**
  As discussed in section 2.1.3. Mobile browsers, there are a variety of popular browsers being used actively worldwide, including iPhone, Android, BlackBerry, Opera, etc. The system should be able to run and execute on as many platform as possible with small amount of user interface distortion among browsers and platforms. System functionalities should be remaining the same across supported browsers to minimize compatibility issues.
4. System design

4.1. System Overview

The prototype of car park reservation system consists of two parts – a web-based management system for car park management user, and a mobile-based system which consists of a range of features as revealed in section 3.

The server system is a web management front-end to support car park reservation from registering member, making reservation to vehicle check-in and check-out. It consists of a membership system which allow customer to join as different membership group which each group are having different benefit and restriction setting, it also allow car park operator to manage and monitor their customers’ behavior by logging and viewing activities history, it also consist of a reservation system which member to make reservation in advance, it allow car park operator to manage reservation, for instance, adding, updating and modifying reservation on behalf of the member, and supporting check-in of vehicle when user arriving as well as check-out of vehicles when user is leaving the car park.

Besides, we have conduct analyse and study on various strategies on how reservation can be done such as advance booking, instance booking, reservation overbooking, membership scheme, company scheme, expanding car park space limit, etc., and instance booking along with membership scheme is selected to be our final reservation approach. It also supports multiple hierarchies of company, carpark, membership scheme, vehicles for applying difference setting and reservation. The server side have been re-constructed, it is able to configure by parameters and fully customizable for each car park by the car park owners, such as car park setting, parking spaces, car park type, overbook ratio for each membership, membership benefit, membership discount, reservation time restriction, cancellation restriction, as well as membership upgrade requirements. Walk-in user/vehicles is also supported in which they would affect the car park’s current utilizations, the system will log and keep track of its one-off activities without detailed information such as driver personal details or past activities. The server back-end system also play an important role for supporting the mobile client’s functionalities.

The mobile client will be mainly used by driver, it is a mobile-based system which enable driver for searching car park and reserving a guaranteed parking space in advance. In our prototype, user can search and view the essential information of a carpark for drivers parking needs, such as
current utilization, available parking spaces, charging rate, phone and address. The searching module consists of two parts – text-based and map based, such that it could support for both high-end and low-end devices or when geo location information from GPS or Wifi positioning is not available. The mobile client have integrated the map view mode with Google map, which provides a better overview of carpark location and availability nearby their current location. It also supports inputting keywords for matched searching as well as inputting custom keyword search for a destination which is in general, supporting all kind of addresses or locations, such that user can view the car parks nearby a certain area of their destination on the map. For users who are unfamiliar with the surround area or concerning about the distance or fuel, we also provides advanced features which provides driving distance and driving time estimation from their current location or inputted destination to the car park, as well as highlighted the suggested driving path from their current location or destination on the map.

The primary goal of our car park reservation prototype is to enable users to reserve for a guaranteed car park space beforehand and enhance driver’s parking experience, while still ensure the car park operator can gain benefits from this systematic approach, and enhancing their car park’s utilization as well as profit margin. Together with membership scheme, the overbook strategy and systematic support from the car park reservation system prototype, the stated goal become possible.

From the car park operator’s point of view, it is expected to receive a set of member, which allows customer relationship study could be achieved from customers’ activities history, and brings potential extensions of using such information, as well as receiving good sense of belonging from member through membership system. Our system introducing overbooking and instance booking, car park operators can adjust the parameters such as available slot, overbook setting and reservation setting, which is in another words, transforming the physical parking spaces into logical spaces, and transform physical time into logical time. By overlapping virtual spaces and virtual travel time, it has potential to maximize the parking space utilization and generate more revenue. From the user’s point of view, they can reserve a guaranteed car park space in advance with minimal cost of instance reservation, and receive membership discount and benefit along with a convience system for searching and making reservation which reduce their unnecessary queueing and cycling time looking for an available car park space everywhere. Ultimately, saving in time, fuel and a better parking experience as well as enhancing the quality of service could be achieved by using this system and lead to a win-win situation for both parties.
4.2. System architecture

The car park reservation system prototype is implemented and deployed using Microsoft Windows, Apache Tomcat, MySQL Server, Apache HTTP server and Java environment. General activities of reserving a carpark as discussed in previous sections are supported. All user can access and perform different actions in our system through their preferred web browser, such as Google Chrome, Safari and Firefox. The following diagram illustrated the overall architecture of our system prototype.

Figure 23. System architecture of car park reservation system prototype
Our system can be divided into four sections – a database server, a main application server, administrative front-end, as well as a mobile-based frontend. The centralized application server is implemented by using Java servlet with assist of Grails and Groovy. It handle all the business logic including company, carpark, member, membership as well as reservation information. It also handles and process user’s requests and communicate with our web management front end, mobile client and the SMS service provider via HTTP interface. It will extract and validate user’s submitted request according to the validation rules against the actual data stored in the database, such as parking spaces availability, user’s login information, and reservation time constraint. Received requests which failure to validate will be rejected by the system while valid input will be stored and further processed by the server.

For data communications between client and server, standard HTTP request, XML and JSON and will be used according to the type of requests. With future development and extension in mind, Restful Web Service interface is also implemented to enable easy communication and extension with different type of client.

The database will store all the the data processed by the system, such as company, carpark, member, membership, reservation and activities history information. As we have implement the system with parameters in mind, the system, carpark and reservation configuration parameters are also stored in the database. All data will be accessed and manipulated by the back-end system deployed at the application server directly.

The management front-end will be used by the car park management user (i.e. car park operators). It is implemented in Java Servlet of JSP/GSP along with HTML5 and JavaScript. It major purpose is to interpret user’s input and submit requests to the application server for further processing. It includes functions allowing user to change and adjust car park information and parameters as discussed previously. It also allow user to register user as member, making and adjusting reservation on behalf of the customer, check-in as well as check-out vehicles. Besides, it also issues AJAX requests to the main application server for autocomplete features in reservation and searching modules which reduce the potential error rate and enhance the user experience for user input. Unlike the back-end system at the application server, it only includes basic validation of user input, e.g. date format, numeric value, etc., so that all operational logic can remain at the server and prevent potential security issues, this also make the system more maintainable.
The mobile-based client will be used by the member – drivers. It will be implemented using Sencha Touch, and programmed mainly and heavily in JavaScript. As a front-end client, it also interprets user’s input and submits requests to the application server for further processing. It allows user to login to the system, search and view car park information, view map information, and reserving parking space at desired car park as well as view/modify/cancel of reservation, by knowing guaranteed spaces available beforehand, queuing and waiting at the car park can be avoid. The mobile client’s features are restricted comparing to management front-end which is intended to be used by system administrator. As we have integrated with Google map, the mobile client will also communicate with Google map server to retrieve geo location information as well as address querying information. In our prototype, all business logic are located at the application server to minimized the risk from program bugs as well as security concerns, thus, the client operations will be depend on the communication with requests via the main server.

In our prototype, we also integrated with SMS service to provide SMS acknowledgement and confirmation service. It includes reservation confirmation, modification notification, cancellation notification as well as reservation request recalling. The SMS message will be sent to the member through the SMS service provider via standard SMS protocol.
4.3. Three-tier architecture

In order to make the car park reservation system more extensible and modular, an architectural pattern Model-View-Controller (MVC) is employed and implemented. Each layer consists of different groups of program code based on their purpose and aspect such as handling user input, business logic, and rendering UI of the application. Hence, it results in loose coupled, more flexible and maintainable program code and system.

![Figure 24. The MVC three tier architecture](image)

- **Model**
  
The model layer consists of a group of classes which represent the domain models and are responsible for data retrieval and storage in the system, which can be manipulate by the controller layer and the view layer. It includes all domain models involved in this project, including company, carpark, member, membership, reservation, reservation log as well as application config.

- **View (Presentation)**
  
The view layer represent the UI pages transported and rendered to the client. In our system, it includes pages coded and rendered in JSP/GSP in the adminitrative front-end system, while the mobile-based client are coded mainly in JavaScript, CSS and partially in HTML. This layer can manipulate the data from controller and model layer.
• **Controller**

The controllers play a major and important role in our system. It includes all Java/Groovy Servlet and business logics, such as validation rules and business rules. For each domain models, at least one controller is employed and implemented in our system. For instance, car park controller, member controller, reservation controller, etc. It interacts with the model layer and view layer to response and dispatch results to the user.

• **Service**

An extra layer of services are also employed in our system. It is a thin layer coupled with the controller layer. Its main responsibility is to interact with the controller and domain model when a transactional service is needed, such as performing long query, chained request as well as critical request such as carpark update, and reservation.

The following diagram illustrated the major components in each tier in our system.

![Diagram](image-url)
4.4. Restful Web Service

Representational State Transfer (REST) is a style of software architecture for distributed hypermedia system which is using the web-standards and HTTP protocol, and it is one of the current popular web service implementation. A RESTful web service is a basic web service which implemented to use HTTP and the principles and concepts of REST. It basic idea is to treat everything as resource identified by universal resource identifiers (URIs) and supported by HTTP common operations. The supported HTTP methods include GET, POST, PUT and DELETE.

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Typical usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Read and access resource</td>
</tr>
<tr>
<td>PUT</td>
<td>Create new resource</td>
</tr>
<tr>
<td>DELETE</td>
<td>Remove resource</td>
</tr>
<tr>
<td>POST</td>
<td>Update an existing resource</td>
</tr>
</tbody>
</table>

Table 3– Restful HTTP typical usage

Our system has implemented to use RESTful style to represent a resource. The following table illustrated some example URI in our system

<table>
<thead>
<tr>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://localhost/carparkms2/carparks">http://localhost/carparkms2/carparks</a></td>
<td>Collection of carparks</td>
</tr>
<tr>
<td><a href="http://localhost/carparkms2/carpark/list">http://localhost/carparkms2/carpark/list</a></td>
<td>Request on car park controller’s list method</td>
</tr>
<tr>
<td><a href="http://localhost/carparkms2/carpark/list/1">http://localhost/carparkms2/carpark/list/1</a></td>
<td>Request on car park controller’s list method with carpark id 1</td>
</tr>
<tr>
<td><a href="http://localhost/carparkms2/member/show/1">http://localhost/carparkms2/member/show/1</a></td>
<td>Request on member controller’s show method with member id 1</td>
</tr>
</tbody>
</table>

Table 4– Example URI in Restful web service

In our current system design. We have exposed JSON interface for the mobile client to retrieve carpark and reservation information via Restful interface from the main application server. For XML format data communication, it can be done by injecting Accept header with application type in HTTP request.
4.5. Data communication in JSON

JSON is a lightweight data-interchange format. It is natively supported by Javascript engine as well as many mobile clients, it is commonly used in web services, thus, easier to process (consume).

The data communication in our prototype are mainly using JSON. The main reason is that, comparing data transfer using JSON with XML or SOAP. JSON utilizes less overhead as it does not require XML header or SOAP header for every transfer; it also reduced the amount of character in each message as it does not use standard tag like other format. JSON is used in our system to communicate between main application server and mobile client through RESTful interface. Data received by the mobile client will further processed and stored in the data store in memory for manipulation in our mobile-based client for driver.

Figure 26. Comparison between JSON and XML
4.5.1. Skeleton of JSON object

The following figure shows two skeleton of our current JSON object design which representing a member object and a reservation object. These JSON will be used for data communication between the main application server and the mobile client in most of the requests. Upon receiving these JSON objects, the client will consume and convert them into Javascript object and store temporarily in the data store located on the smart phone.

Figure 27. Skeleton of JSON object in car park reservation system
4.6. Package Diagram

There are total 3 packages 19 classes implemented in this prototype. All of them together to cover the functionalities of company modules, carpark modules, member modules, membership modules, reservation modules, reservation log modules as well as SMS module. The following diagram illustrated the relationship between each package and each classes.

![Package Diagram](image)

*Figure 27. Package diagram for car park reservation system*
4.7. Database design

The following ERD diagram shows the structure of the database of our prototype system. It is also representing the data structure of each of the domain model including in company module, carpark module, member module, membership module, reservation module, reservation log module as well as SMS module.

![ERD diagram of Car park reservation system prototype](image)

Figure 28. ERD diagram of Car park reservation system prototype
4.8. System implementation

Our prototype can be divided into two parts – the web management system and the mobile client. There are total seven modules including company module, carpark module, member module, membership module, reservation module, reservation log module, module. Each of them consist of Create, read, update and delete (CRUD) functions on the management side while the client side is restricted to access carpark searching, member profile management and reservation management. During our implement process, we aimed and decided to provide a system to support the critical functionalities needed to operate a car park reservation system with our suggested approach instead of designing our one single business rules or regulation with a concern of each car park might have difference physical, geometrical or operational limitations. Therefore, the system are re-constructed into supporting various parameters related to carpark setting and reservation setting, the system is full customizable and configurable by system administrator via web front end. As discussed in previous section, we suggested to incorporate with membership system and overbooking strategy to enhance car park performance and parking experience. In this section, we will cover the major features and work flow implemented in our prototype.

4.8.1. Multiple hierarchies

Our prototype support multiple hierarchies structure of entity relationships. The following diagram gives an overview of the relation of each of them.

Figure 29. Multiple hierarchies of Car park reservation system
In our prototype, modules includes multiple relations are company, carpark, membership and vehicles. These structures are used to apply different default settings to each of its underneath entity.

For company and carpark as illustrated in Figure 29, the company can setup default setting which can be applied to all of its carpark such as default membership overbook ratio for different group. By supporting multiple company, the system can also be viewed as another prospective where it is running in single company mode if there is only one company, and it can be viewed as running an carpark reservation agency mode when there is more than one company. Both prospective are also common in practice where a properties group can have multiple child companies operating carpark.

![Figure 30. Multiple hierarchies of Car park reservation system](image)

For each carpark, each car park can override the default setting base on their limitations and needs, and apply the new setting for each of the membership group, including overbook ratio, discount rate, maximum reservation length and reservation cancellation restriction as shown in Figure 31 below.

![Figure 31. Customizable car park setting](image)
Every membership scheme consists of a set of members, the membership group can be automatically upgraded based on application config setting, each member can have multiple vehicles and different vehicle type for making reservation. It is also common and realistic in real life especially in a family environment where all vehicles are belongs to one single family member.

![Image](image-url)

*Figure 32. Multiple vehicles belonging to one member*

### 4.8.2. Car park setting

Our prototype allows car park operator to adjust their car park setting through web management front-end. The following figures shows all the settings which are adjustable.
In the car park setting, we supported over 20 parameters to be configurable. As most of the setting parameters are self-explanatory, we will summarize and describe some of their usage and functionality. From Figure 33, the total are used to allow users to initialize and configure the total parking spaces, and unavailable parking spaces occupied by monthly users, the current statistic fields are not intended to be edited, however, it is enabled for administrators as a fallback tool to adjust their car park spaces under special needs.

In Figure 34, users can change the car park setting including charging rates in different time periods, they can also adjust the overbook ratio and discount rate, the Reservation Interval representing the maximum reservation length while the Cancellation setting representing the cancellation time constraint in our system. Moreover, users can configure their car park type and services provided.
such as open air car park, car washing service. This is intended for the use of car park profile matching and recommendation in advanced search module, however, it is not implemented as an enhanced map view is included, details will be covered in later chapter.

### 4.8.3. Member management

As the car park operator are assumed to be the administrator of the system, they can view and manage members’ profile. Initially, this profile is setup while user is registering to the system.

![Figure 35. Managing member profile](image)

In the member profile, apart from basic information and vehicles, user can adjust their user role and current membership. They can also change the company scheme which will be used for filtering carpark result to display the select company’s car park only in the client for reservation. The priorities setting are intended to use for matching the carpark setting discussed in previous section. The good records and late records are used to calculate the current user rating, these number will be updated automatically when a reservation has specified an end time, and the user is leaving on time or late accordingly, these rating will be used for calculating membership group’s promotion. The running balance is used to record the total balance of a member for statistic purpose while the credit balance is intended to be used as compensations and pay for another bills when an overbooked member decided to cancel their reservation, which is discussed in section 3.3.
4.8.4. Application parameters

With configuration and customization in mind, some application config are refactored and available for system administrator to adjust as shown below. For reservation related constraints and setting, these are reserved as application-wise default setting. We also include config value for membership upgrade as well as expired request checking, the membership upgrade is assumed to be execute once a month, however, for illustration and demonstration purposes, we implemented it to execute same as request checking, which will be triggered after a successful check out, it will be discussed in later section.

![City Carpark Reservation System](image)

**Figure 36. Managing member profile**
4.8.5. Reservation

In our prototype system, we have modelled the entire reservation process into computerized system. We study and tried different approach such as advance reservation and expanding carpark spaces from monthly user, we eventually come up with and suggesting the use of integrating membership system, overbooking with instance reservation which is more realistic and practical. In this section, we will cover most of the major system flow involved in the reservation process.

4.8.5.1. Overview of reservation process

The figures below illustrated the general process involved in a reservation process. The user will first login to the system, and request for a reservation, once the reservation is confirmed, the system will issues a SMS confirmation message to the user. As we are in instance reservation mode, the reservation will begin to be timed for charge, this transform the travel time into reservation time with minimal cost for the user. Once the user is arrive, they will check in to the car park, they will also check out the carpark when they finish using the parking service. Apart from the SMS acknowledgement, we have modelled each stage into different status as shown in Figure 38.

![Figure 37. Work flow of the reservation process](image)

![Figure 38. Status recorded in our system](image)
4.8.5.2. Overbooking

As revealed in section 4.8.2 Car park setting and 4.8.3 Member management, our system support membership scheme as well as overbooked reservation. Our primary goal is to maximize the utilization by transform physical car park spaces into logical car park spaces, as well as gaining a better profit margin by bringing the travel time from user into reservation time. The travel time is denoted as follow:

\[ \text{Travel time} = \{\text{arrival time} - \text{reservation start time}\} \]

In short, it is achieved by hoping or knowing a certain number of user might finish using the car park within the travel time of another incoming user, it is done by expanding the available parking space with different overbook ratio of the membership group. The value of the overbook ratio rely on the experience and statistic result from the car park operator and our system support adjusting of this value whenever needed. In our system, the utilization is denoted as follow:

\[ \text{max. available spaces} = \{\text{total parking space} - \text{monthly user}\} \text{ for each vehicle type} \]

\[ \text{utilization} = \{\text{max. available spaces} - \text{anonymous user} - \text{current reservations} - \text{current check-in car}\} \]

100% utilization = \{max. available spaces become zero\}

Currently, we have 3 kinds of membership supported – Gold, Silver and Bronze. By default, the overbooking ratio is as belows

![Current setting](image)

\[ \text{Figure 39. Default overbooking ratio} \]

We tried various setting for the overbooking ratio, such as negative ratio and reversing the default ratio order. As all extra slot are virtual spaces and we would like to provide the best quality of service and highest rate of guarantee parking, we found that having gold member with
smallest ratio is more practical and effective. For a car park with 100 available spaces, gold member can reserve for 5 extra car park spaces even when the car park have zero available spaces, while bronze member would receiving a 15 extra spaces, which have a higher risk of unavailable parking space upon arrival.

It should also noted that these 3 parameters are overlapping with each others, it implies that the maximum of extra spaces is 15 for all membership group. As car parks are most likely running in FCFS basis, racing condition could existed. In our system, we display a reservation queue on the system including reservation start time and membership group to the user as shown in Figure 40, it is up to the car park owner to operate as a FCFS basis or highest member group first. In a worst case, highest member group first could introduce an unnecessary waiting time for lower member group if a flexible approach is not used by the carpark operator. An example of such situation is illustrated below:

\[
\text{Gold member travel time} = 9 \text{ min}, \quad \text{Silver member travel time} = 1 \text{ min}
\]

\[
\text{Upon silver member arrival, he/she has to wait for } 9 \text{ min} - 1 \text{ min} = 8 \text{ min.}
\]
With such strategy implemented, it can boost up the maximum capacity of the car park, even when it is not fully utilized and not benefiting from extra overbooking slot introduced, the car park also receiving an extra profit margin from the travel time while user can enjoy a guaranteed parking slot as well as membership benefit. The statistic data from the figure below shows the utilization being boosted up to 112% after running a simulation run in our current system.

Figure 41. Utilization statistic after overbooking is implemented
4.8.5.3. Reservation requests

User can make reservation in our system, user friendly features such as auto completion and date picker are implemented to assists user input and reduce input error rate as showns below. Member is able to optionally specify an end time to enjoy member benefit or leave it empty. The car park operator is also well-informed with car park’s current utilization and availability of parking spaces beforehand so that they do not have to perform unlimited retry for making one single reservation on behalf of the member.

![Figure 42. Auto complete feature for searching](image)

![Figure 43. Date picker for inputting date time](image)
When the system receives a reservation request, it will first validate the user input, and validate if the reservation time input is correct, the minimum reservation time in our setting is 1 hour where the maximum reservation length for gold, silver and bronze is currently 4,5 and 6 hours respectively. The system will then check if vehicle is available, i.e. check-in in another car park due to incorrect user input, it will then check if car park availability as revealed in previous section – Overbooking, if the car park is not available, the request will be marked for reservation recall. Otherwise, the reservation will be confirmed, SMS acknowledgement will be sent to the user and the reservation will be marked as reserved state. The follow diagram illustrate the flow of making a reservation in our prototype.

![Figure 43. Work flow of making reservation](image)
Each rejected requested will be bounded to a 15 minutes expire time, these request will be checked and trigger during check out process, the system will select the oldest request within 15 minute from the gold membership group, as we assume that the driver might cycling arround and searching for an carpark space within 15 minute, the system issue a SMS acknowledgement which recalling the user to make another.

![Figure 44. SMS confirmation and reservation recall](image)

Due to technical limitation from the SMS provider, we are not able to read the incoming replied SMS message from the user, also to avoid undetermined turnaround time from user response, we use the recall approache and assuming the user will make an instance reservation via mobile client or phone call as soon as they needed the space.
4.8.5.4. Reservation modification

Sometimes, user are needed to modify their reservation such as changing or extending end time or cancel their reservation by various reasons. It can be done by both the management front end and the mobile client. The UI design is consistent with related pages. Modification from the management front-end will receive less restrict on modification time constraint, user can optionally choose to issue SMS acknowledgement for reservation update or cancellation. In some cases, cancellation might or might not introduces a charge, it can also be optionally selected by the car park operator. However, if it is requesting to modify from mobile client, the modification/cancellation time constraint will be applied, such that modification of reservation will be disabled and rejected.

![Reservation modification front end](image1.png)

*Figure 45. Reservation modification front end*

![SMS acknowledgement for reservation update / cancellation](image2.png)

*Figure 46. SMS acknowledgement for reservation update / cancellation*
### 4.8.5.5. Check-in

Once the user is arriving at the car park, they can check in to the car park to make use of the parking service. Car park operator can view and search from the web front-end to check-in a reserved vehicle.

However, as overbooking strategy is applied, and our system will determine the actual physical spaces instead of logical reservation spaces, it might not be always available instancelessly as discussed in previous section, user might have to wait or cancel the reservation. Also, depending on the car park operation, the sequence of check-in vehicle can be selected by the operator, member who arrive earlier or having highest membership grade could be check in first. In a worst case, when overbooked reservation is totally overloaded, we assume compensation would be made by the car park company as discussed in section 3.3. Upon a successful check in, arrival time and check in status will be recorded while the charge is still calculated from reservation start time, which bring a profit margin including travel time and waiting time to the car park.

![Check-in UI Screenshot](image_url)

*Figure 47. Screenshot of check in UI*
4.8.5.6. Check-out

When the user finish using the parking service, they will leave the carpark and check out at the gate. Similar to check in process, the car park operator can view and search for current vehicle/reservation in the check out UI.

![Screenshot of check out UI](image)

The system will first determine if end time is specified by the user. If it is specified, and the user is leaving earlier than end time, on time or late, and incrementing the good record and late record counter in our system respectively. These counter will be used to determine the user rating for a membership upgrade. No matter end time is specified or not, the charge will be calculated base on the usage time of the reservation period. The charge is calculated by the configurable parameter in car park setting including day time charges and night time charges. As two charging mode is included in our implementation, boundary case could exist, such as overlapped day time (09:00am-17:00pm) and night time (17:00pm-09:00am) charge. In our default configuration, we introduced a offset parameter for handling such issues. The setting is currently set to 30 minute, which means there would be a 30 minute allowance for calculating the boundary case. The following example illustrate our calculation of charges, the calculating applied to day to night overlapping as well as night to day overlapping cases.

*Usage starting from 4.30pm or before, the 1st hour would be day time charge*

*Usage starting from 4.31pm or afterward, the 1st hour would be night time charge*

Start time = 4.30pm, leaving time = 5.30pm

Usage hour = 5.30pm-4.30pm = 1 hour

Charges = 1 hour * day time charge
If an end time is specified by the user and their leaving time is earlier or on time, the member can enjoy their membership benefit of having reservation discount and will be applied to the overall usage charge for current reservation bill. This membership benefit together with the guaranteed parking spaces introduce an incentive for the user to specify their end-time, while this information would be critical and very useful for the car park company to manage their resource, collect statistic data and plan ahead for adjusting their available resources, carpark setting, overbook setting or membership benefit.

Once a check out is successful, the leaving time, charges, running balance, and a check out status will be recorded in our system. This finish a entire reservation process for the user. For the system, a successful check out is served as a trigger to execute some further processes, including expired request clean up, SMS reservation recall. As in our current implementation, membership upgrade will be also executed and check via the application config to upgrade member who fullfil the upgrade requirement, an addition SMS notification will also be issued to the user who upgraded.

![Membership upgrade notification via SMS](image)

*Figure 49. Membership upgrade notification via SMS*
4.8.5.7. Reservation log

During the reservation process in our prototype, there are lots of event triggered in each stages, such as status changes, reservation acceptance, rejection, SMS acknowledgement, etc. We have recorded all activities in our system during reservation for data, reservation and event tracking. From the company point of view, it will also be a very useful feature which allowing them to know what is happen in when and involving who, especially when they are not familiar with the overbook setting. It also provides some form of an overview of car park statistic at a certain period.

![Reservation log]

*Figure 50. Screenshot of reservation log*
4.8.5.8. Mobile client

In our prototype, we also provides a mobile client for car park user to make and manage their reservation virtually everywhere with internet access. It includes function of member registration, carpark searching which include text mode and map view mode, reservation viewing and modification, member profile management and vehicle management. As all business logic are implemented and stored at the application server, most of the functionalities remain the same as discussed in previous sections. In another words, most part of the mobile client is another frontend interface of the main application with features restrictions, it will process user input, issues requestes to main server, and render the results returned by the server with minimum business logic preprocessing. Compare to the main server application, the map view is newly introduced, we also included advanced feature to assist and enhance car park search and parking experience. In this section, we will briefly describe the features which is already introduced in previous section and further discuss the new map view feature.

Figure 51. Entry screen of the mobile client
4.8.5.9. Registration

In order to use the mobile client system, the user needed to be a member of our car park reservation system prototype. If they are not a member, they can register by clicking on the Register button at the entry screen and provides their personal information. The information will be sent to the server for processing, and the registration result will be returned to the mobile client.

![Registration screen of the mobile client](image)

*Figure 52. Registration screen of the mobile client*
4.8.5.10. Profile and vehicle management

Once the user login successfully to our system, they can see the entry screen as shown in Figure 51.

![Main functions screen and login screen of the mobile client](image)

Figure 53. Main functions screen and login screen of the mobile client
Starting from the options available from the bottom, user can manage their profile as well as their vehicle, the profile management has a similar UI as the registration screen, in which user can manage and update their personal details, however, membership is a read-only property and are not available for user to change. Upon submitting the data to the server, the server will determine an validate the user input, if there is no error, the member profile will be updated immediately in both application server and mobile client.

![Profile management]

*Figure 54. Profile management*
Similiar to profile management, user is able to view their vehicle listing in our current system. User can also create new vehicle which will be available for making new reservation. Upon submitting the request to the server, it will be further processed, the application server and the mobile client will then be updated immediately in a similar manner. However, as vehicle consists of limit number of info, editing of vehicle is disabled to avoid data inconsistency in the reservation log, server log and other statistic functions.

*Figure 55. Screenshot of vehicle management*
4.8.5.11. Car park searching

To make a reservation, the first step which a user will do is search for a car park. We provide two searching modes – text based and map view. Both modes accept keyword and area search, which will issue a request to our application server to perform string wildcard matching against car park name, address, and company name, which is within the selected area. Once the server returns the matching results, the mobile client will display a list of car parks including their available parking spaces and current hourly charges. We assume most users will make use of the map view instead of the text-based mode, however, we reserve this basic searching method which enables car park searching for low-end devices and for situations where geo location information is not accessible.

![Figure 56. Screenshot of text-base searching](image-url)
To provide an easy-to-use and user-friendly car park searching experience, we have integrated the car park searching and reservation functionalities with Google map. It accepts keyword search and area search as shown in Figure 56 as parameters to perform queries, where the returned results from the application server will be rendered within the map with custom map pins, enabling a great overview of the resultant car park, including location and distance and its surrounding areas.

Figure 57. Supported area and car parks in map view
As shown in Figure 57, all carparks are located easily on the map. We have also included customized map pins to illustrate the result. As drivers usually concern about parking spaces available around their destination area, we also support destination searching. In general, it accepts all kinds of location keywords such as address, building name or address in both Chinese or English, and covering virtually all areas as the result will querying and returned directly to and from Google map server. The meaning of the map pin is as described in the following table:

<table>
<thead>
<tr>
<th>Map pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="green-car" alt="Map Pin" /></td>
<td>The current geo location of the user’s mobile device. When a destination search is performed, it will locate and represent the destination as the current location.</td>
</tr>
<tr>
<td><img src="orange-car" alt="Map Pin" /></td>
<td>Car park with utilization lower than 50%</td>
</tr>
<tr>
<td><img src="yellow-car" alt="Map Pin" /></td>
<td>Car park with utilization lower than 80%</td>
</tr>
<tr>
<td><img src="red-car" alt="Map Pin" /></td>
<td>Car park with utilization of 100% or higher</td>
</tr>
</tbody>
</table>

*Table 5. Map pin description*

*Figure 58. Custom destination search in map view*
Besides, we also implemented advanced features and support displaying of car park details within the map info windows, including car park available parking slot. As driver also concern about the distance, fuel use, driving time, distance and driving path, we also implemented and includes estimated driving distance, estimating driving time in the info window, along with a suggested driving path colored in blue on the map. Furthermore, a straight line colored in green is between two destinations to indicate the straight direction is also included on the map, which will be useful for driver who having a driving habit of memorizing famous building or shopping mall instead of memorizing the driving path, especially when driving in a unfamiliar area.

Figure 59. Driving path, distance and time colored and displayed in map view
4.8.5.12. Car park searching

To make an reservation on the mobile client, user can click on the car park list as shown in Figure 56 and click on the Reserve button in the car park detail view as shown below, or clicking the Reserve button as shown in Figure 59. Both action will redirect the user to the same reservation page as shown below, similar to reservation on management front-end, user can submit the request by specifying the end date base on the needs. The request will be submitted to the application server, the same validation and processing as discussed in Overbooking section and Reservation section will be applied, all business logic are process at the server, the result will then returned to the mobile client. If an reservation is placed successfully, a SMS confirmation message will be follow by and received on the mobile device with the member’s mobile number registered in our system.

![Car park detail and Make reservation](image)

*Figure 60. Making reservation in mobile client*
5. System testings

After the implementation or changes of each module, testings have been conducted to ensure the system is working as expected. In this section, we will include test cases of main functions and the major flow in our prototype.

5.1. Test cases

<table>
<thead>
<tr>
<th>Test case</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Making reservation when the car park is available.</td>
<td>As expected. Reservation is made and SMS confirmation is sent and received. Car park utilization is updated.</td>
</tr>
</tbody>
</table>

![Reservation](image)
Making reservation when the car park is full, but overbooking is not overloaded

As expected.
Reservation is made and SMS confirmation is sent and received.
Car park utilization is updated

Making reservation when the car park is totally overloaded.

As expected.
The server reject the reservation and prompted for max. reservation limit reached. The reservation request is also recorded.
Car park utilization is not updated
<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>One member reserved the last car park space, and another member try to make reservation immediately</td>
<td>As expected. The first member reserved successfully while the other member request is rejected. Car park utilization is not updated</td>
</tr>
<tr>
<td>Check in a reserved reservation vehicle into car park when car park is not full</td>
<td>As expected. Member check-in successfully. Car park available space is updated</td>
</tr>
<tr>
<td>Scenario</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Check in a reserved reservation vehicle with overbook, but the carpark is full and overloaded.</td>
<td>As expected. Member check-in unsuccessful. The system prompt for Max utilization reached. Car park available space is not updated</td>
</tr>
<tr>
<td>Member check out reservation without end time specified</td>
<td>As expected. Discount charge is correctly displayed.</td>
</tr>
<tr>
<td>Member check out reservation with end time specified and leaving on time</td>
<td></td>
</tr>
</tbody>
</table>
### Member check out reservation with end time specified but late

<table>
<thead>
<tr>
<th><img src="image1" alt="Check-out" /></th>
<th>As expected. Discount is not applied. Charge is correctly displayed</th>
</tr>
</thead>
</table>

### Anonymous use check into a carpark which is available

<table>
<thead>
<tr>
<th><img src="image2" alt="Check-in" /></th>
<th>As expected. Check in is successful. Car park utilization is update.</th>
</tr>
</thead>
</table>

### Anonymous use check into a carpark which is available

<table>
<thead>
<tr>
<th><img src="image3" alt="Check-in" /></th>
<th><img src="image4" alt="Error" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiving SMS reservation confirmation after placing a successful reservation</td>
<td>As expected. Confirmation SMS message is receive on the mobile number provided.</td>
</tr>
<tr>
<td>Receiving SMS update or cancellation acknowledgement</td>
<td>As expected. SMS message is received with correct message.</td>
</tr>
<tr>
<td>Receiving SMS acknowledgement for reservation request recall</td>
<td>As expected. SMS is received with correct message.</td>
</tr>
</tbody>
</table>
Membership upgrade and receiving upgrade notification

As expected. The message is sent and received correctly and membership is upgraded

User perform car park search without any keyword to display entire list

As expected. 30 car parks are listed with correct information

<table>
<thead>
<tr>
<th>Carpark results</th>
</tr>
</thead>
<tbody>
<tr>
<td>威爾斯地王停车场</td>
</tr>
<tr>
<td>Available: Car 12%, Motorcycle 0%</td>
</tr>
<tr>
<td>Daytime $5, Night $7</td>
</tr>
<tr>
<td>玄輝軒停車場</td>
</tr>
<tr>
<td>Available: Car 2%, Motorcycle 0%</td>
</tr>
<tr>
<td>Daytime $5, Night $14</td>
</tr>
<tr>
<td>博遠駐車場</td>
</tr>
<tr>
<td>Available: Car 22%, Motorcycle 0%</td>
</tr>
<tr>
<td>Daytime $5, Night $13</td>
</tr>
<tr>
<td>青銅停車場</td>
</tr>
<tr>
<td>Available: Car 31%, Motorcycle 0%</td>
</tr>
<tr>
<td>Daytime $5, Night $12</td>
</tr>
<tr>
<td>美域商場停車場</td>
</tr>
<tr>
<td>Available: Car 38%, Motorcycle 10%</td>
</tr>
<tr>
<td>Daytime $4, Night $8</td>
</tr>
<tr>
<td>美域商場停車場</td>
</tr>
<tr>
<td>Available: Car 19%, Motorcycle 2%</td>
</tr>
<tr>
<td>Daytime $5, Night $10</td>
</tr>
<tr>
<td>萬田停車場</td>
</tr>
<tr>
<td>Available: Car 25%, Motorcycle 15%</td>
</tr>
<tr>
<td>Daytime $10, Night $11</td>
</tr>
<tr>
<td>靜湖花園停車場</td>
</tr>
<tr>
<td>Available: Car 60%, Motorcycle 20%</td>
</tr>
<tr>
<td>Daytime $6, Night $8</td>
</tr>
<tr>
<td>花園城第二期</td>
</tr>
<tr>
<td>Available: Car 53%, Motorcycle 12%</td>
</tr>
</tbody>
</table>
User perform car park search without any keyword but selected area to filter the list

As expected. Car parks are filtered with correct information

User perform car park search with keyword and selected area to filter the list

As expected. Only 2 car park are found
<table>
<thead>
<tr>
<th>User search for car park in map view with keyword</th>
<th>As expected. Only car park matching keywords are shown</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User search for destination within map view</th>
<th>As expected. Map pin is moved to the destination</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Show car park info, driving distance, driving time, driving path on map view

As expected. Data is correct, estimated time, distance and path are close to accurate
Making reservation on mobile client for available car park

As expected. Reservation is made and SMS confirmation is received.

Making reservation on mobile client for unavailable car park

As expected. Reservation is rejected and request is recorded.
5.2. Performance evaluation

As of a late stage of this project, we would like to learn more about how the system can improve the car park current situation. Thus, we have adjusted part of our system source code and performing some experimental trial run with pre-defined configuration to simulate a continuous usage of the car park and collect some statistic results.

These trial run are performed on one single carpark, with generated members and anonymous users. In our latest trail, some settings are disabled such as discount rate, different car type and hourly charges. Each test will perform reservation for 24 hour as we are focusing in short term reservation, and assuming no vehicle will be leaving the queue because of out of patient, also FCFS will be used, we also round the result into integer value for calculation. The following table shows the default setting for our experiment.

<table>
<thead>
<tr>
<th>Setting name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car total space</td>
<td>100</td>
</tr>
<tr>
<td>Month private car</td>
<td>25</td>
</tr>
<tr>
<td>Private car available space</td>
<td>75</td>
</tr>
<tr>
<td>Anonymous car</td>
<td>0</td>
</tr>
<tr>
<td>Hourly charge</td>
<td>10</td>
</tr>
<tr>
<td>Overbook ratio (Gold)</td>
<td>5%</td>
</tr>
<tr>
<td>Overbook ratio (Silver)</td>
<td>10%</td>
</tr>
<tr>
<td>Overbook ratio (Bronze)</td>
<td>15%</td>
</tr>
<tr>
<td>Discount rate</td>
<td>0%</td>
</tr>
<tr>
<td>Member/Vehicle arrival rate</td>
<td>2~4 per 2 minutes, uniform distributed</td>
</tr>
<tr>
<td>Vehicle leaving time</td>
<td>1~3 hours, uniform distributed</td>
</tr>
<tr>
<td>Travel time for member</td>
<td>5-15 minutes, uniform distributed</td>
</tr>
</tbody>
</table>

*Table 8. Default setting for trial run*
The first test we are interested in is the profit gained from employing a membership system without overbooking. Our result shows that the total profit after execution without membership system is 15620, while the total profit with membership is 15970. It demonstrated a profit margin gained from using membership system, the extra profit is generated and coming from the travel time of the member.

![Profit](image)

*Figure 60. Comparison of profit*

The second test is to collect the utilization data with membership system but disabled overbooking. The result shows the the utilization with membership system is 94% while it is 93% without the membership system. It seems that it only generated a minor amount of differences.

![Utilization](image)

*Figure 61. Comparison of utilization*
We also conducted test for profit gained from employing a membership system with and without overbooking enabled. The data shows that with overbooking enabled, the profit increased to 16240, and the profit is 15570 without overbooking enabled. We should also noted that there is a slight differences when overbooking is not enabled comparing to our first test. We assume it is caused by the vehicle, membership generation. The extra profit gain from using overbooking approach should be coming from the extra virtual spaces we had added by calculating the overbooking ratio.

Figure 62. Comparison of profit

Apart from profit, we also observed the utilization when the same setting. We recorded a 96% of utilization for overbooking enabled comparing to 94% of utilization under the same environment. From reading the log, the increased utilization is similar to the profit gain in which it is boosted up by the extra virtual spaces.

Figure 62. Comparison of profit
The next thing we are interested in are the utilization of the system under heavy loading. We change the generation of vehicle to 4 vehicles per minutes, 5 vehicles per minutes and 6 vehicles per minutes, and observing the results. It shows that the utilization continue to increase from 99% to 106% when 6 vehicles are arriving per 2 minutes.

![Utilization Graph](image)

Figure 63. Comparison of utilization under heavy loading

Base on the results of the above experiments, although it is relatively basic, but it already prove that by employing the membership and reservation system reflect a position enhancement in terms of utilization and profit margin to the car park.
6. Limitations

“We are not perfect”, especially when the system is targeted as a prototype system. Although we have configurable parameter implemented in this system, which tackled and reduced a lot of limitations and logical issues. There are still some limitations we are concerning of and existed in the current system.

6.1. Pre-defined membership scheme and car park settings

Our prototype consists of membership scheme of Gold, Silver and Bronze, it also consists of over 20 parameters for car park settings and 10 application-wise settings. As they are related closely with the implementation of early design stage of car park reservation system. These setting value are configurable but the name cannot be changed. Such as changing the membership name from Gold to Golden would introduce error in matching carpark setting, or vice-verse, it also affect some parts within our reservation calculation and discount calculation. A more flexible and better software design and database design such as reflection technique or mapping each of the keys into a relation could be one of the possible way to fix this issues.

6.2. Car park services and payment type

At the early stage of this project, we decided to include carpark searching with profile and service matching against member’s preferences for car park recommendations, it can be done in a very complex way or a simplier way such as weighting and ranking with preference level. However, we slightly adjusted the project scope in the middle of the project due to study interest. Thus, this two group of setting would have no affection to other functions included in this prototype. If there is more time to develop this prototype, I will be definitely interested in studying for various approaches for profile matching issues.
6.3. Overbooking setting and small car park

Currently, the overbooking ratio parameter is represented and only supported in percentage. It works well in medium sized car park (approx. 50-100 spaces). However, from our experiments, such percentage would make a big difference in large car park while there would be no effect on small car park if the percentage is relatively small, e.g. 5% or 10%., it is due to the calculation we implemented are actually round up the integer, which in real life it does make sense since 0.5 or 0.9 both does not represent a vehicle as a whole. This limitation can be resolved by changing it from percentages to plain old number, however, for a system supporting many car parks at the same time, a dual parameter might also be a possible solution.

6.4. Accuracy of map view in mobile client

With current implementation, the geo location information of the car park is pre-defined, i.e. assumed to be input by user, while the user’s current location is retrieved by GPS or Wifi positioning received on the mobile devices. Although Wifi positioning is one of the good alternative when the GPS signal is not accessible, it might not suitable for a concrete forest like Hong Kong and its accuracy is sometimes very poor. Besides, the destination searching function are rely on Google map query, there is also a chance of returning a location of a similar or same address from another places. Also, the driving path is something not always be the shortest driving path, an example of wrong driving path is as shown below where two destination is just one block away.

Figure 64. Driving path suggestion in map view
7. Difficulties

All in all, this project is a very challenging project to me and full of obstacles. First, carpark related system is not publicly existed or accessible in Hong Kong, there is basically similar system for no referencing. As I am not a driver, I know nothing about cars or car park, it brings a great amount of uncertainty to me throughout the year. Luckily, my supervisor and some of my friends are willing to share their experience to me, which helps me a lot in designing and gathering information needed for the prototype.

Also, one size does not fit all, instead of providing a single solution regulation to the system, we decided to implement a configurable system along with sample configuration for user to administrate, however, it involve quite a lot amount of changing to our existing design. We also uncertain that whether the user will satisfies with the overbooking aproach. Also, reservation without a end time is very hard to control which place a high risk on overbook usage which could introduced a high rejection rate which affecting the image of the company and the confidence level from the customers. Besides, although we provide incentives to attract user to specify their end time, but they might not always know when they would leave, with such a a flexible end time, it brings up lot of difficulties to design a better sample regulation for our prototype. Also, the performance evulation is very new to me, since we decided to add this in at a very late stage, even though my supervisor is wholeheartly to guide me on approach to conduct such simulation, I still found it very hard to implement and having a great sense of uncertainty lose confidence for the result I have come up with.

Last but not least, for the technique part, since the language used is very new to me, the learning curve is quite deep. Furthermore, the logical part are alsovery hard for me, as we are having win-win situation in mind for every design or implementation, we need to concern of both user benefit and car park benefit, which is a very tough decision to make for student with limited business sense, I was always fall into the wrong track and spent time on unrelated matter. Also, along with heavy workload from other course, time schedule is very tight and eventually, I need to adjust the project schedule frequently, it is also one of the minor difficulties I am facing during this project.
8. Future extension

8.1. Advance booking

We are currently using instance booking as our strategy, however, from user point of view, they are very likely would like to reserve in advance, but, from our experience, advance reservation is very hard to predict the available parking spaces without specifying the end time, also, schedule algorithms might not able to fit in due to uncertain end time with preferred start time. One possible solution could be introducing blocking time for anonymous user check-in, as well as introducing offset amount to calculate the nearby reservation at a certain time, but it would makes the user to receive a longer waiting time, a more thorough study is indeed needed in such cases.

8.2. Carpark profile matching

As discussed in Limitations section, carpark profile matching was intended to be included, such feature will be very useful for recommending the best carpark to user in both map view and text based view which best fit the user’s expectation such as distance, cost, car park type, and car park services provided.

8.3. Integration with reward system

In Hong Kong, car park parking is very common to be tighted with reward or award scheme, such as free parking allowance or reduced parking price. With entire membership system and car park general activities supported ready, it is possible to includes and integrate with such system to enlarge the user benefit and the profit margin for the carpark owner.
8.4. Another approach

As the last paragraph of this report, I would also like to include an idea of another approach in my mind during the development of this project as a future extension, which is also related to the car park utilization of current project. The idea is to monitor the car park performance base on the utilization of the car park, in which the popular car park is often overloaded, we increase the charge when utilization is high and reduce the charge when utilization is low. Such idea is to balance and distribute the car park usage across multiple car park own by the same company and trying to reduce the overloading of the popular carpark, by further recording of statistic data, car park company might able to analyse their car park usage, and better allocate their resource, such as closing, improving or relocating of low usage’s carpark. It might be possible to share a similar idea of managing land use planning, demand and supply of parking. Although it would involve a great amount of knowledges and experience beforehand, I believe it should be a quite interest project to deal with.
9. Summary

As far as my study and data collection goes, parking reservation of public car park is yet existed or not yet publicly accessible in Hong Kong. In this project, we have study the issues thoroughly, and developed a prototype system by integrating membership system with overbooking strategy, where instance booking will be used.

With a strong sense of win-win situation in mind, we developed a prototype consisting of web-based management front-end as well as a mobile client to assist the entire flow for carpark reservation from member registration, car park searching, making reservation to vehicle check-in and check-out as well as reservation logging. We delivered a prototype system which is fully customizable and is flexible for car park operators to adjust the parameters within our system through web interface. By implementing and using our system, the car park transformed the physical parking spaces into logical spaces and brings the physical travel time into logical reservation time with minimum cost from the user. By overlapping virtual spaces and virtual travel time, it has proven that the parking space utilization can be maximized and the profit margin is increased. Base on the trial run we experimented on, we recorded a 6% utilization boost when the car park is starting from empty, and a maximum of 9% of utilization boost with the setting of 5%, 10% and 15% overbooking rate for gold, silver and bronze user respectively, while the revenue gained is 6% if we fill up the car park in advance and feeding vehicle continously.

Besides, from the user’s point of view, they can reserve a guaranteed car park space in advance with minimal cost of instance reservation, and receive membership discount and benefit along with a convience system for searching and making reservation tackle the unnecessary queueing and cycling time issues for seeking an available car park space everywhere. Last but not least, together with the map view included in client which provides a rough estimate of driving distance, driving time and driving path, it tackled most of the concerns common to drivers heading toward one destination. To sum up, a win-win situation is already able to foresee from this system, the quality of service and parking experience are enhanced while car park company are receiving a positive profit margin, the ultimate goal is no more far away to be achieved.
References


http://www.sino-propertyservices.com/ps_about_en.html

http://blog.nielsen.com/nielsenwire/online_mobile/who-is-winning-the-u-s-smartphone-battle/


## Appendix

### I. Monthly log

| Progress log - Sept. 2011          | • Research and seek for various project topic  
|                                  | • Study on different latest technology including iphone, android, and web  
| Progress log - October 2011       | • Study background and requirements on suggested projects  
|                                  | • Research on related application  
|                                  | • Study on web and mobile related technologies  
|                                  | • Preliminary design of the system.  
|                                  | • Write project plan  
| Progress log - Nov. 2011          | • Revise and limit project scope  
|                                  | • Study feasibility of using web vs mobile as core for this project  
|                                  | • Study related web and mobile technology to be used in this project  
|                                  | • Environment and code base setup  
|                                  | • Development of basic functionality of the system (phrase 1)  
| Progress log – Dec. 2011          | • Investigate and research for better framework for development to support more mobile devices.  
|                                  | • Prove of concept with simple application on native iPhone development, jQTouch, jQuery Mobile and Sencha touch  
|                                  | • Research and study on ORM framework  
|                                  | • Research and study on Grails, Groovy, Spring and Hibernate  
|                                  | • Revise and adjust project scope, database structure, and system requirement as of discussed with Dr. Lee  
|                                  | • Study and Rewrite of phrase 1 Android application into Sencha touch based application  


- Revise database structure to fit latest changes.
- Continue study and rewrite of the application into Sencha Touch based application.
- Research and study on applying parameter to the system
- Further research and study on implementing booking or reservation system.
- Prepare and write interim report

### Progress log – Feb. 2012

- Prepare and write interim report
- Refine structure of DB to make it more feasible
- Fix and migrate code to adapt new structure for both server and client side
- Adjust and enhance the system to make it more practical and realistic as discussed with Dr. Lee
- Investigate feasibility for employing SMS as acknowledgment in the system.
- Change UI and DB to remove unnecessary and/or non-mandatory fields.
- Review result of interim report and seek for improvement

### Progress log – Mar. 2012

- Adjust project planning, adjust weighting between server side and client side.
- Meeting with Dr. Lee to discuss latest ideas and changes.
- Refine DB and module structure for latest changes.
- Make the system become configurable via parameter.
- Develop, analysis and implement different rules set for practical and realistic reservation
- Employ multiple-level hierarchy into company, car park, membership
- Add and implement SMS acknowledgment.
- Investigate the feasibility of reservation prediction without end time
specification.

- Refine and improve reservation, check in checkout module.
- Add and implement anonymous check in checkout.
- Fix and adjust broken mobile client module due to latest changes.
- Study and explore profile matching/ranking technique.
- Investigate feasibility for employing google map in client side.
- Add and implement event-based system log for reservation module.
- Explore, study and develop testing & data collection for performance evaluation.
- Prepare final report.

### Progress log – Apr. 2012

- Fine-tune system parameters
- Disable advance reservation, minor changes in coding/parameters to use instance reservation instead.
- Functional test on all server/client modules.
- Code changes for performance evaluation
- Fix logical bugs due to late changes to instance reservation mode.
- Add Google map integration in client, add free-form destination searching on map.
- Add map pin colors base on utilization, add driving path suggestion, driving time estimation.
- Minor changes in client to show more useful info for users such as membership and carpark spaces.
- Prepare demonstration and presentation slide
- Finalize code base and prepare for submission
- Prepare and compile final report

~ End of report ~