



The University of Hong Kong

Faculty of Engineering

Department of Computer Science

COMP7704

Dissertation Title

A Smart Phone Application for Valet Parking

Submitted in partial fulfillment of the requirements for the admission to the
degree of Master of Science in Computer Science

By

LUO Xianyang

3035237420

Dr. T.W. Chim

Date of submission: 30 / 11 / 2016

Abstract



Declaration of Candidate

I, the undersigned, hereby declare that the work contained in this thesis is my own original work, and has not previously in its entirety or in part been submitted at any university for a degree.

Only the source cited in the document has been used in this draft. Parts that are direct quotes or paraphrases are identified as such.

I agree that my work is published, in particular that the work is presented to third parties for inspection or copies of the work are made to pass on to third parties.

The University of Hong Kong,

LUO Xianyang

Acknowledgments

I would like to thank~

Table of Contents

lalala

Chapter 1. INTRODUCTION

1.1. Background Information

Nowadays a lot of people will go to clubs for dinners, after-work drinks or having fun with friends. Since Hong Kong is an International and fast-tempo city, it is rather common for citizens to go to places like Lan Kwai Fong after a day's work or at weekend. It is convenient for customers to drive their own cars to the hotels, clubs or bars. But as a matter of fact that Hong Kong is one of the most crowded cities in the world, it is not easy for drivers to find a parking lot quickly.

Valet parking service can help customers park their cars. It is offered by some restaurants, shopping malls, clubs and so on. A person called valet will drive a customer's car to parking lot when the customer arrives at the gate of the hotel and return the car when the customer leaves. The main advantage of valet parking is convenience. On one point, customers do not need to find a parking lot by themselves. On the other point, they do not have to walk a long way from the parking

lot to the hotel, which saves lots of time. All they need to do is just dropping their cars at drop-off point.

However, in such a fast-tempo city and such a high-tech era, some problems of traditional valet parking for a customer are:

- 1) must use a valet parking ticket to redeem their cars. If the ticket is lost, customer need to prove that the car belongs to him or her by showing driver license or identity card
- 2) may do not know where is the drop-off point for a certain hotel, restaurant or clubs and may take time to find it
- 3) do not know current status of the car

To solve these problems, I would like to develop a mobile application for people in Hong Kong to use valet parking service easier and more efficient.

1.2.Project Description

This project mainly focuses on help drivers enjoy better valet parking service. A customer can register our service via its phone number.

After adding a new car by inputting plate, brand and color, he or she can choose a parking lot and generate a QR code. When a valet scans

the QR code successfully, the order will be generated and the customer can use the phone as a valet ticket. Whenever the customer wants to get the car back, he or she can just click the “recall” button. Then our valets will return the car back to the customer.

This project has three parts and functions are as follows

1) iOS version application for customer

- a. allows a user to register, login and reset password via phone number
- b. allows a user to add a new car, edit an existing car and delete a car. Information like plate, brand and color are requested
- c. allows a user to generate order by choosing parking lot and car
- d. allows a user to get the car back by just clicking a button
- e. allows a user to view all the historical orders

2) iOS version application for valet

- a. allows a valet to login and reset password via phone number
- b. allows a valet to add an order by scanning the QR code generating by a user

- c. allows a valet to view all the opening orders
 - d. allows a valet to end an order by just clicking a button
- 3) server and database
- a. processes all the http request and sends a proper response back to phone
 - b. stores data safely of all users, valets, cars and orders
sends notification to a user then an order has been generated and ended successfully

1.3. Project Objectives

Since the traditional valet parking service has matured, so the app need to be more attractive to gain users. We have to follow the current trend of design, follow the guideline of user interface design and take some of them into consideration to fulfill the goal, which is Shneiderman's Golden Rules of Interface Design[1], Jun Gong's Guideline for Mobile Application[2] and Nurul's Threes Layers Design Guideline for Mobile Application[3].

Apart from user interface, the app should be rather easy to use. Users do not to do lots of setting or follow a complicate guideline to generate an order. There are some mobile applications in the market with

similar purpose like Meibo, Youbo and so on. After analyzing those applications, I found out that they were not that easy to use. Since there are a new technology in iOS called 3D touch **which brings a new powerful dimension to Multi-Touch interface**, users can enjoy the best convenience while parking their cars. Also, it is rather innovative if there is an Apple Watch application cooperating with application on the iPhone.

The final goal of this project is to change the traditional valet parking service by attractive and innovative features so our objectives can be conducted in **three** aspects:

- 1) **attractability**: to attract users, the application should have friendly and beautiful user interface, reasonable layout and overall clean look.
- 2) **innovation**: users are more willing to use the application by using new technology like 3D touch. This project allows user to enter the **"parking now"** and **"current orders"** views from the icon which makes it rather convenient and enjoyable.
- 3) **connectivity**: **a user may enjoy more convenience if he or she has an Apple Watch. The all the parking and recall request can be**

done through Apple Watch. A user do not even need to take his
or her phone out the pocket.

1.4. Summary of Chapters

Chapter 1 firstly introduces the background and motivation of this
project and then gives brief introduction and objectives of this project

Chapter 2 firstly introduces two similar mobile applications in the
market and analyze the advantages and disadvantages.

Chapter 3 design blaalala

Chapter 4 implementation balalallala

Chapter 5 conclusion balalaalal

Chapter 2. RELATED WORK

To develop a successful application, we need to refer to the existing applications and see if how we can do it better.

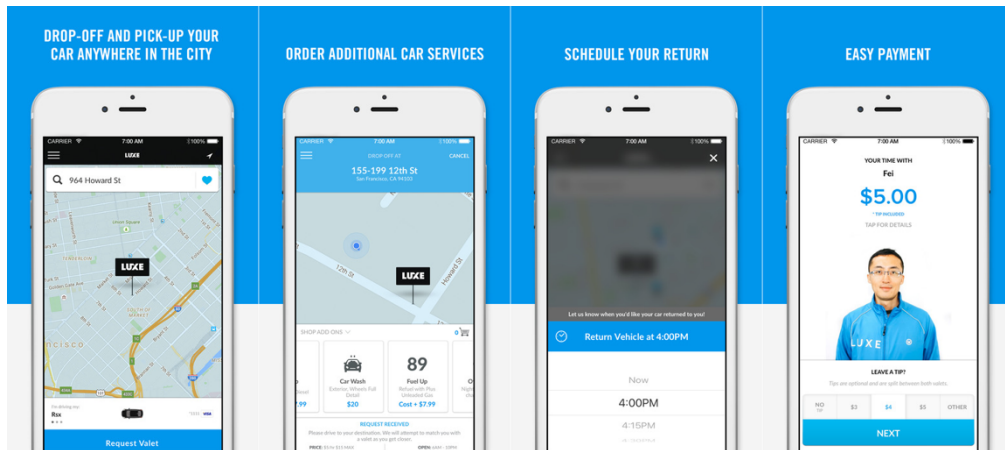
In this part, I will introduce two mobile applications offering valet parking service in the market. The first one is Luxe, which is available in American and then Youbo in China. I will firstly give a brief introduction and analyze the advantages and disadvantages of both applications respectively.

2.1. Luxe

<http://www.luxe.com/>

Luxe is a valet parking app available on both iOS and android. It is currently available in San Francisco, New York, Chicago, Seattle, Austin and Log Angeles. Figure 2-1 shows demo pictures for Luxe.

Figure 2-1 iPhone Screenshots for Luxe



After trying Luxe a few days, I found the advantages and disadvantages as below:

Advantages:

- 1) the user can drop off and gets returned anywhere and a valet will wait at the drop point. This is the best experience for a valet parking service. The user saves time since he or she does not need to find a certain drop-off point
- 2) the user can pay for the service using its phone, which makes it a Uber-like application. All the process can be done by the application.
- 3) the user interface is attractive and modern

Disadvantages:

- 1) the application does not use new technology like 3D touch and does not offer Apple Watch application

- 2) the register process is rather complicated, which requires both email account and phone number
- 3) all those service are only available in the USA

2.2.Youbo

<http://www.uboche.com/>

Youbo is a valet parking app available on both iOS and android. It is currently available in Beijing, Shanghai, Chengdu and other big cities in mainland China. Figure 2-2 shows demo pictures for Youbo

Figure 2-2 iPhone Screenshots for Youbo



Youbo offers similar services as for Luxe. After taking a real experience, I found the advantages and disadvantages below:

Advantages:

- 1) a valet can drive the customer to the destination and then park the car. And the customer can get the car returned at wherever he or she wants. This is rather similar to Luxe
- 2) a valet could help wash and refuel a customer's car
- 3) a user can register and login just using its phone number

Disadvantages

- 1) the application does not use new technology like 3D touch and does not offer Apple Watch application
- 2) the user interface does not look good.

Chapter 3. BACKGROUND KNOWLEDGE

There are a lot of third party frameworks to enhance the performance or to optimize the user interface of this project. This chapter mainly introduces some useful frameworks in the development. The first part talks about tools used in front-end which is the mobile application. And the second part talks about tools in back-end which is the server and database.

3.1. Tools used in Front-End

3.1.1. AFNetworking

AFNetworking is networking library used in iOS and Mac OS X development. It is built on top of the Foundation URL Loading System, extending the powerful high-level networking abstractions built into Cocoa. It is a high efficient networking module along with feature rich API which is rather easy to use. It powers some of most popular applications on iPhone and iPad.

The usage of AFNetworking is simple, which differs from the origin method offered in iOS. After initializing a session manager, the user just needs to configure some parameters, sends the request and then waits for the response. A simple post request can be implemented below:

```
AFURLSessionManager *manager = [AFURLSessionManager manager];
NSDictionary *parameters = @{@"foo": @"bar"};
[manager POST:@"http://example.com/resources.json"
 success:^(AFHTTPRequestOperation *operation, id responseObject) {
    parameters:parameters
    NSLog(@"JSON: %@", responseObject);
} failure:^(AFHTTPRequestOperation *operation, NSError *error) {
    NSLog(@"Error: %@", error);
}];
```

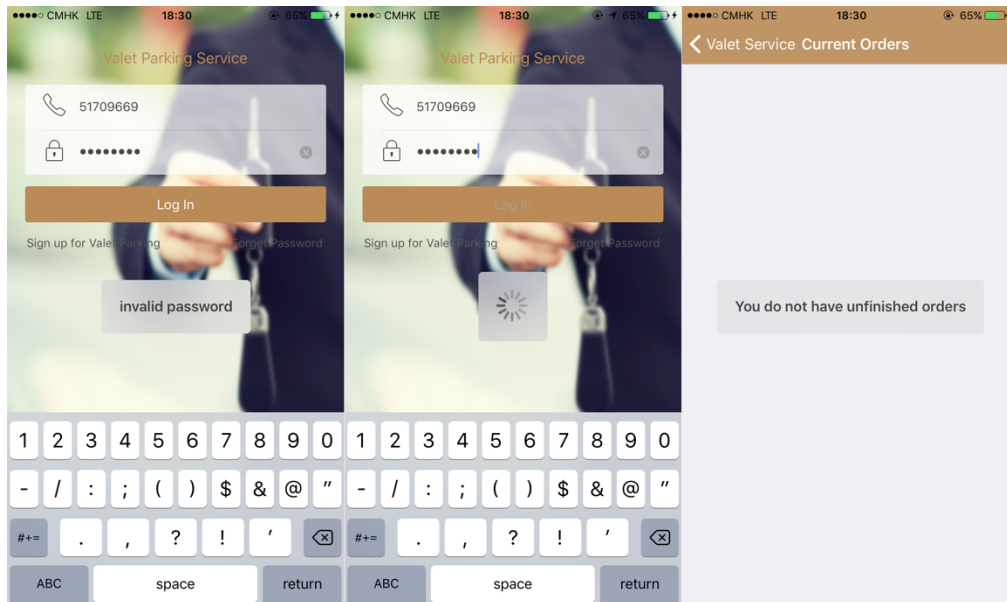
This code block sends an asynchronous request containing a parameter dictionary to the server. If the request is successful, the manager will get a *responseObject* containing request information or an *error* if the request is failed.

3.1.2. MBProgressHUD

MBProgressHUD is an iOS drop-in class that displays a translucent HUD with an indicator and/or labels while work is being done in a background thread. I use this framework in almost every view in this

project. It's easy to add a text indicator or progress indicator as shown in Figure 3-1

Figure 3-1 Screenshots for usage of MBProgressHUD



The usage MBProgressHUD is very simple by initializing an MBProgressHUD instance and show it in current view. By choosing the mode of a hud, it can display to show text or process indicator or both. The code block below shows how to add a hud to current view and hide until a certain process is finished.

```
[MBProgressHUD showHUDAddedTo:self.view animated:YES];
dispatch_time_t popTime = dispatch_time(DISPATCH_TIME_NOW, 0.01 *
NSEC_PER_SEC);
dispatch_after(popTime, dispatch_get_main_queue(), ^(void){
    // Do something...
    [MBProgressHUD hideHUDForView:self.view animated:YES];
});
```

3.1.3. JSONModel

MVC (Model, View, Controller) is widely used in object-oriented developing. A model is used to store and manage data. And the format of most data downloaded from server is JSON. In order to use this data, front-end needs to transfer it to model. JSONModel allows rapid creation of smart data models. By creating an model and inheriting it from JSONModel, the front-end can use the *initWithDictionary* method to generate instance of a model rapidly without reading each key-value from response data. The code block below shows an statement of class named *CountryModel* which is inherited from JSONModel. It has three attributes. As long as the JSON data has the same key, the application can generate a model using one line of code.

```
#import "JSONModel.h"
@interface CountryModel : JSONModel
@property (assign, nonatomic) int id;
@property (strong, nonatomic) NSString* country;
@property (assign, nonatomic) BOOL isInEurope;
@end
```

3.1.4. SMS_SDK

SMS_SDK is a framework used to send verification code to users' mobile phones. It's totally free and easy to be integrated in the application. Since our valets host customers' cars, it is important for valets to verify the identity of the user. The code block below shows how to use SMS_SDK to verify a user's phone number.

```
[SMSSDK getVerificationCodeByMethod:SMSGetCodeMethodSMS
```

```
phoneNumber:@"51709669"  
zone:@"852"  
customIdentifier:nil  
result:nil];
```

3.2. Tools used in Back-End

3.2.1. MongoDB

MongoDB is a free and open-source cross-platform document-oriented database program. Being different from database like MySQL, MongoDB is classified as NoSQL database program and uses JSON-like documents with schemas. It works well with Node.js and can implement common database method like create, read, update and delete rather easily and efficiently.

Compared to MySQL, MongoDB has some main advantages as follows:

- 1) rich data model, dynamic schema, typed data and data locality
- 2) rich query language which is easier to code and read
- 3) development is simplified as MongoDB documents map naturally to modern, object-oriented programming languages

3.2.2. Express

Express is a fast, unopinionated and minimalist web framework that provides a robust set of features for web and mobile applications. It is the most popular framework in Node.js as for now. Because it provides

lots of powerful and useful features including organizing application's routing and providing template solutions, the user can focus on developing the functions of applications. The main advantages of express are as follows:

- 1) event driven: this feature enables registering various profound functionalities by connecting them to other events that are being executed once the event is triggered.
- 2) javascript closures: this feature allow the user to use variables that are defined in the outer calling function inside the callback body, which is extremely useful to solve the conflict while working with various applications.

Chapter 4. REQUIREMENTS AND DESIGN

In this part ~~~~~

4.1. Requirements Analysis

4.1.1. Product Perspective

This project has front-end and back-end. The front-end is built on Xcode 8 using Objective-C and the back-end is built with Node.js and MongoDB. And there are two applications in the front-end. One is for customers and the other one is for valets. Both customers and valets use their phone number as login account.

4.1.2. User perspective

There are two types of user – customer and valet. After a customer logging in, he or she can add a car, choose a place to park, check orders and request his or her car. After a valet logging in, he or she can access to all the opening orders, generate an order by scan the QR code on a customer's phone and end an order after returning a customer's car.

4.1.3. Functional Requirements

Using this product, a customer will be able to choose a place and choose a car to park. Here is the list of high-level functional requirements that this project is focus on. The two apps and the back-end must:

- 1) Authenticate and authorize a customer or a valet
- 2) Store a number of data records describing cars. Each record will have attributes as below:
 - Car plate
 - Car brand
 - Car color
- 3) Store a number of data records describing orders. Each record will have attributes below:
 - Order date and time (create, update, end)
 - Order place
 - Customer name
 - Customer phone
 - Car plate
- 4) Allow customers to add, edit a car by typing in plate, brand and color. Also allow customers to delete a car.

- 5) Allow customers to choose a place and a car to park
- 6) Allow customers to check all current orders and historical orders
- 7) Allow customers to request his or her car back
- 8) Allow valets to generate an order by scanning QR code
- 9) Allow valets to access all the opening orders
- 10) Allow valets to end an order

Next I will introduce the details of design.

4.2. Design

4.2.1. System Architecture

The entire system can be divided into two parts: front-end and back-end. Back-end is a server based on Node.js with MongoDB as database. All the *get* and *post* requests including registering an account, logging in, adding a car, editing a car, deleting a car, creating an order, updating an order are handled by this server. It will check every request for correctness and validity and send corresponding response back. And front-end are two mobile applications built using Xcode and Objective-C. The database used in front-end is CoreData which is native in iOS and easy to use. Figure 3-1 shows the architecture of the system

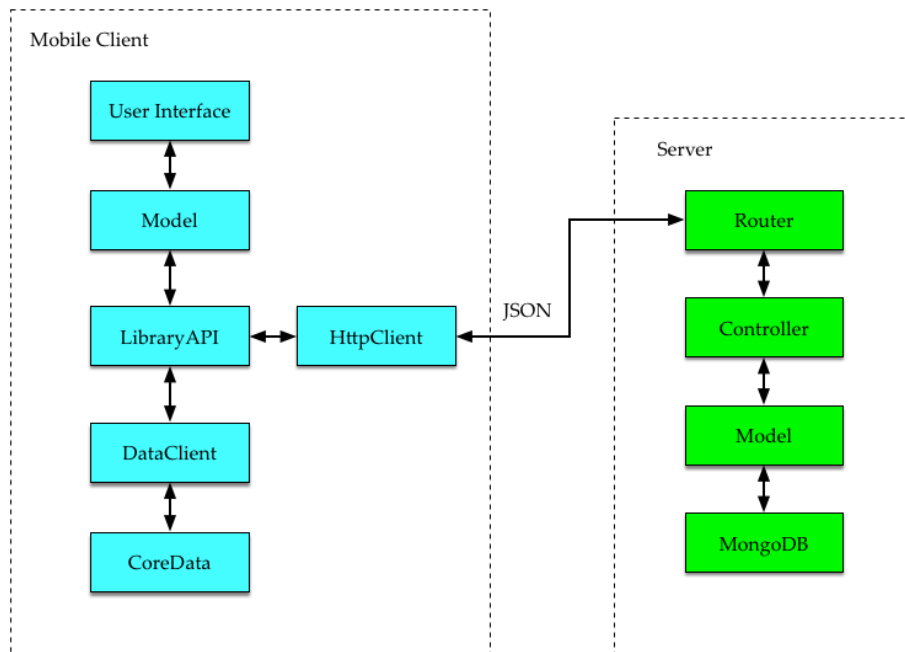


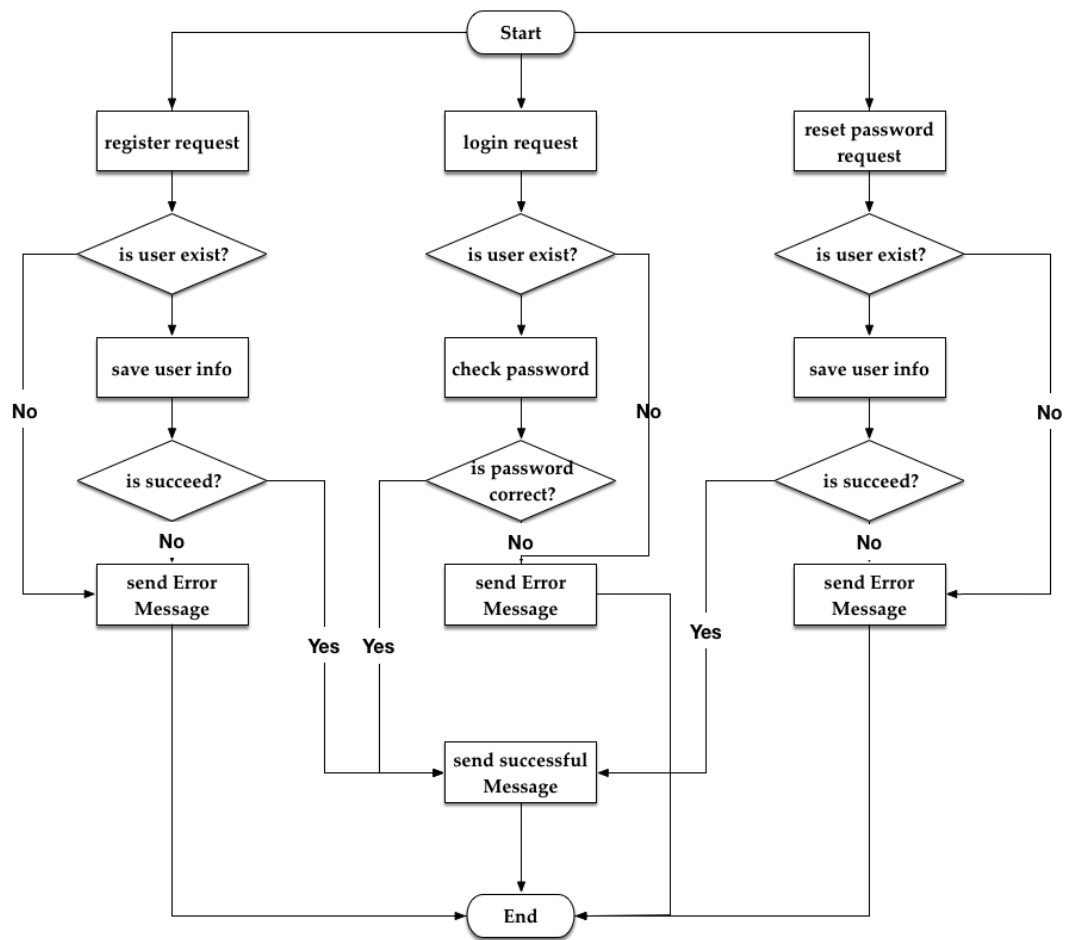
Figure 3-1 Architecture of System

4.2.2. System Workflow

The system workflow consists of two parts. The first part is the work flow in the server and the second part is the work flow in the mobile client. These two parts will be introduced separately.

4.2.2.1. Workflow for server

Server works as back-end in this project and will handle all the http request sent from front-end. There are three major functions in server: manage accounts, manage cars and manage orders. Figure 3-2 shows the flow chart for account management



4.2.3. Object-Oriented Design

4.2.3.1. Identify actors and use cases

4.2.3.2. Use case diagrams

4.2.3.3. Sequence diagrams

4.2.3.4. Design documents

4.2.3.5. Class diagrams

4.2.4. User Interface Design

4.2.5. Database Design

Chapter 5. IMPLEMENTATION AND TEST

Chapter 6. REVIEW AND FUTURE WORK