Table of Contents

Abstract

This project is a competition project that we need to develop a Customer Relationship Management (CRM) System. Hong Kong IT Accountants Association (ITAA) holds this competition. We make co-operation with one of the catering enterprise to develop our system basic on their requirements and their needs. Our system will help boost sales and business, maintaining customers and performing business analysis.

This project is to develop both server-sided and client-sided system, which is web service that second party can adopt. Nowadays, smart devices are becoming much popular like Apple iPhone, iPad, Google Android devices…etc. The client side aims at using the mobile application (mobile app) by receive the data from the web database to achieve more information to their mobile.

For the server-sided system, we will adopt cloud-technology to develop. The web service will provide editing function about the company information. We will provide the app project generation tools on our website. It is a whole CRM package of our design. The company can use our system to generate xcode and android project to let the staff to download base on their selections of functions and information on our website. This means people can edit or create their own mobile app and they do not need to have any programming skills on smart devices.

In the world up to now there is no any standard Customer Management System. Different enterprise uses their own system to keep relationship with the customers. This environment happens in all industry. Therefore, we will need to build up a Customer Relationship Management System for catering management.

Up to now, there are no any tools from the Internet let the people to create a mobile app or app project in the Internet. Many people would like to learn creating

mobile application but it is so difficult to learn for people who do not have any programming skills before. So, we would like to use this method to many the things easier.

Environmentally, we want to reduce the usage of paper for leaflet or food menu. People can use a mobile device together with our system to receive the company latest information.

This system finally brings simplicity, ease-of-use to users in small-and-specific area and is environmental friendly together with a mobile device and a web browser.

Introduction

Smart Phones are becoming more popular in Hong Kong. Smart Phone contains an operating system inside and the two most popular systems are iOS and Android. They let people to use their mobiles to do more than before and make communication easily with peoples. Nowadays, smart phone is essentially to us in many peoples' eyes. So, we based on this trade to develop a mobile application to our people.

This project aims at building a server-sided system working with client's application. This system is a web service which provides map data to the client application automatically when the client startup our application on the mobile with valid network connection. User can use the mobile application to view the news, e-menu, request e-coupon, GPS location directing, and also send us their feedback in the mobile.

All of the user's habit that we receive will be used for data mining and analysis to improve the sales of the company. All of the data is under the user permitted circumstances and will protect safely in our database.

Customer Relationship Management System means that it is a system can used for not only one company or group of company. It should be able to let another company to adapt using this system. Therefore, we will concentrate not only the system, but also the further implementation to meet different company's requirements.

In this report, we will present the system architecture (design of the system) which shows how the client and the server communication and synchronize the data together. We also concerned about the limitation od developing the system based on reliability requirements, performance requirements, hardware environment,

future extension and implementation language. It also includes the problems and proposed solutions in terms of software and hardware. Lastly, the cloud technology we adapted in our project will also be discussed.

The Problems

Background

Our CRM system is designed for a catering company called "Enoteca Group" (http://www.enoteca.hk/). As we know that the catering company did not adopt much Information Technology method in it. They only have a website to show their news, menu, location…etc. They have a small ordering system in their shop only.

Problem Description

Mainly, the customer cannot get the company information easily from the Internet. Usually, the customer needs to walk it the shop to purchase their service. And this is quite difficult to encourage more customers to their shop. They are difficult to increase their popularity in the market.

Second, the company does not have any mobile application that customer can used to notice the latest news from it. In nowadays Hong Kong, the group should provide a mobile application to let increase the popularity to increase the sales simultaneously.

Third, the company does not have any feedback tools or forms on Internet or mobile. There are no any statistic or data mining tools to analysis the customers' comment or their order habit. If we take action on this, the system will help them to increase the profit.

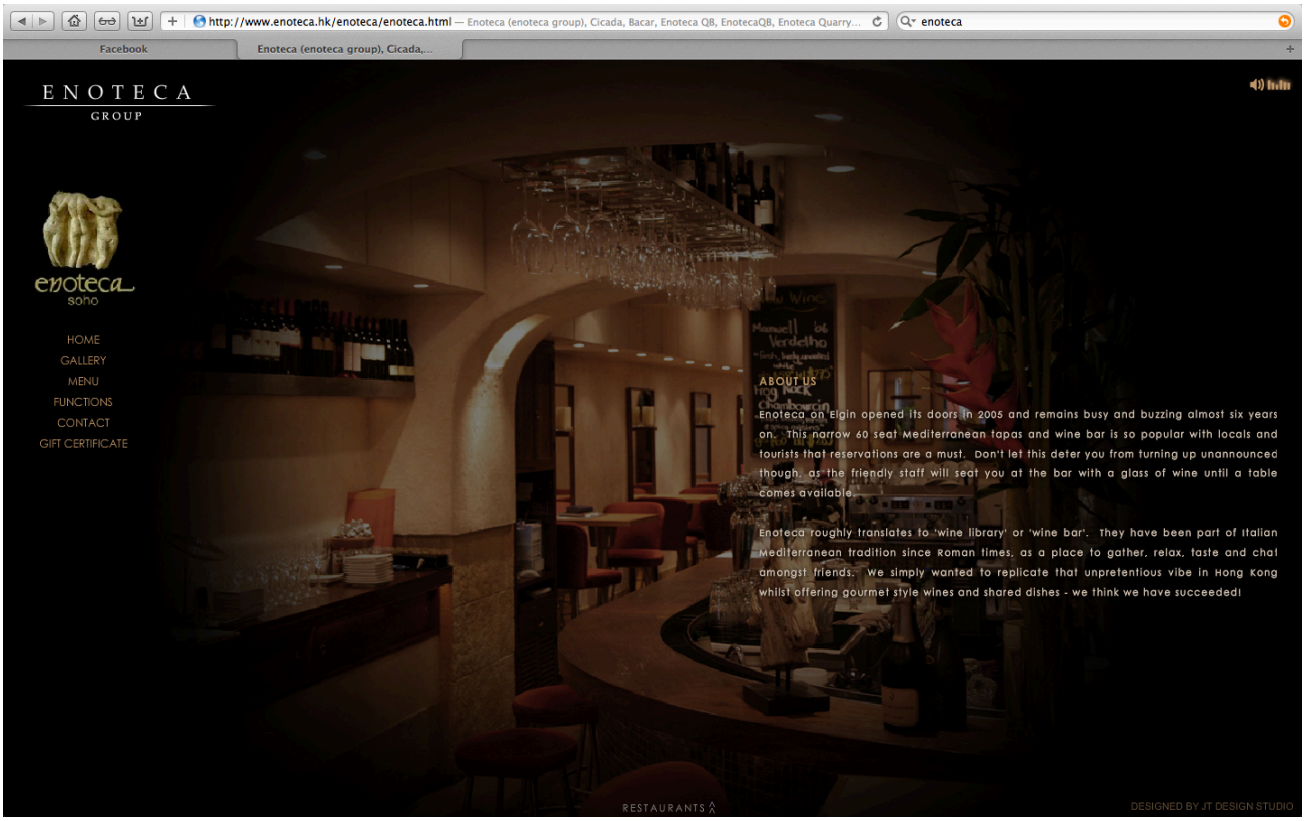
Problem Environment

The company is lack of using information technology to improve their services. For catering market in Hong Kong, many of them do not have any Customer

Relationship Management System. Catering in Hong Kong has a very important and large market. By adapting CRM System, the company can improve their relationship with customer. In a long run, they can improve their profit and attract more people to come and pay for your service.

Lack of communication between customers and company

(http://www.enoteca.hk/) The website is the company that we co-operate with and help them to build up a customer relationship management system.



We found that the company has their original official website for customer to find out the information. But a big problem in the situation is that the website was developed with ActionScript of Adobe Flash. Flash can make anything become beautiful and attractive, but they cannot communicate to other. They can only provide the views to the client. Therefore, we would provide a customer relationship system for the client and the staff to mark good relationship between the company and the customer.

Proposed Solution

This section is the discussion of solutions proposed to handle the problems encountered based on the design of the system, functionality and feasibility and the system requirements. Different aspects of problems have been concerned in terms of software and hardware layer. The discussed issues are as following:

- The way of client to archive server's address
- Automatically connect to the server and receiving latest information from the server
- The way of client and server to communication
- Consistency of data structure between server slide and client slide
- The structure of the mobile application
- The functions of the mobile application
- The functions of the website system
- The mobile application project generation on the website
- How to handle if the client can't connect the server in the specific area
- How to support multi-platform client
- How to make client easily to use our mobile application
- How the way to implement the system in the future
- Analysis reports for customers' habits

The way of client to archive server's address

Assume that the Client use smart devices to use the mobile application (mobile app) that we provided. When the client installs the mobile app and starts it in the first time, the app will auto connect to our server and received all the company information to the mobile app by using AJAX and JSON technology. After the data received, the data will store in the mobile local database. And then the app will put back the data to each tab and page on the mobile. The client can view the information, photos, map…etc. Once the client installs our mobile app, they should agree to provide some user information securely and also to turn on the push notification service.

The icon of the mobile app store on the server, when the user start the application, the photo will automatically download from our server to plug back to the related place which shows in the diagram.

The icon image will be put to the related area in the app

Automatically connect to the server and receiving latest information from the server

When the client exit and reopen the mobile app, the mobile app will first try to connect to the server if the connection is available to get the latest data. If the connect is not available, the mobile app will result timeout in the connection, therefore, it will use the mobile local database information.

```
try{
    String jsonArray=new URL("link").getContent();
}catch(UnkownHostException e){
    System.out.println("Timeout! Connection fail");
}
```

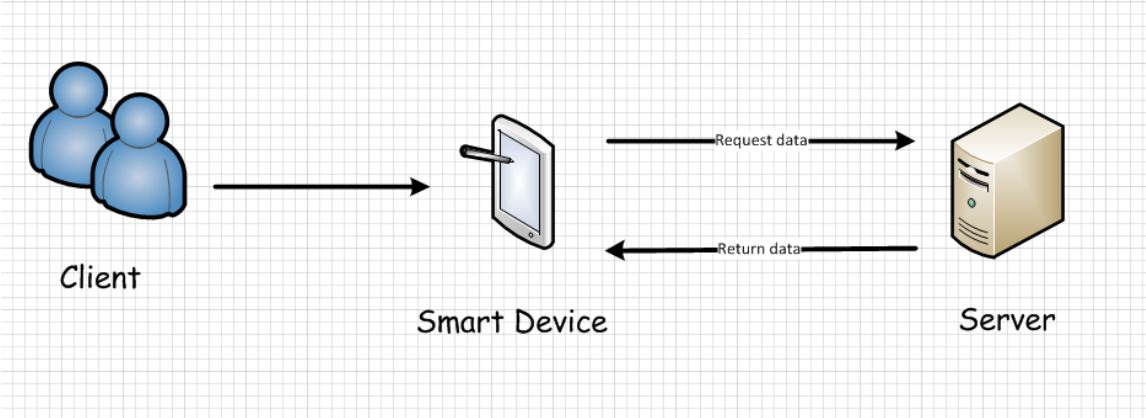
This diagram shows the code in Android when connecting to the database server. “link” is the server domain address or ip address.

```
+(BOOL) canConnectToServer;
+(void) errorWithNetworkConnection;
```

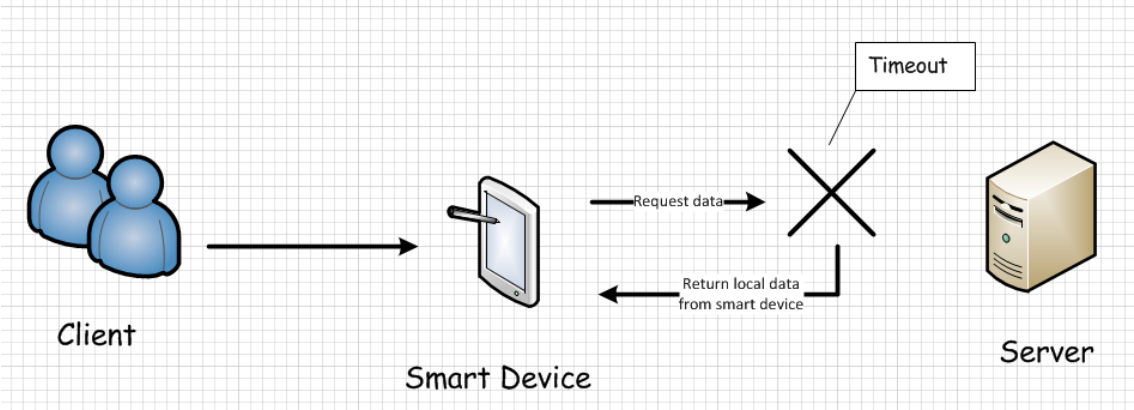
This is the header(.h) of iOS Objective C code of the method of checking the network connection.

```
+(BOOL) canConnectToServer {
    NSString *urlString = [config getServerURL];

    NSLog(@"canConnectToServer Testing %@", urlString);
    NSString *escapedUrlString = [urlString stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
    NSString *responseString;
    NSURLResponse *response;
    NSError *error;

    NSURLRequest *request = [[NSMutableURLRequest alloc]
                             initWithURL:[NSURL URLWithString:escapedUrlString]
                             cachePolicy:NSURLRequestUseProtocolCachePolicy
                             timeoutInterval:3]; // 3 second timeout

    NSData* data = [NSURLConnection sendSynchronousRequest:request returningResponse:&response error:&error];
    responseString = [[NSString alloc] initWithData:data encoding:NSUTF8StringEncoding];
    if(![responseString isEqualToString:@""]){
        NSLog(@"Y");
        return YES;
    } else {
        NSLog(@"N");
        return NO;
    }
}
```

The above is the main(.m) code in iOS that check the network connect to the server.



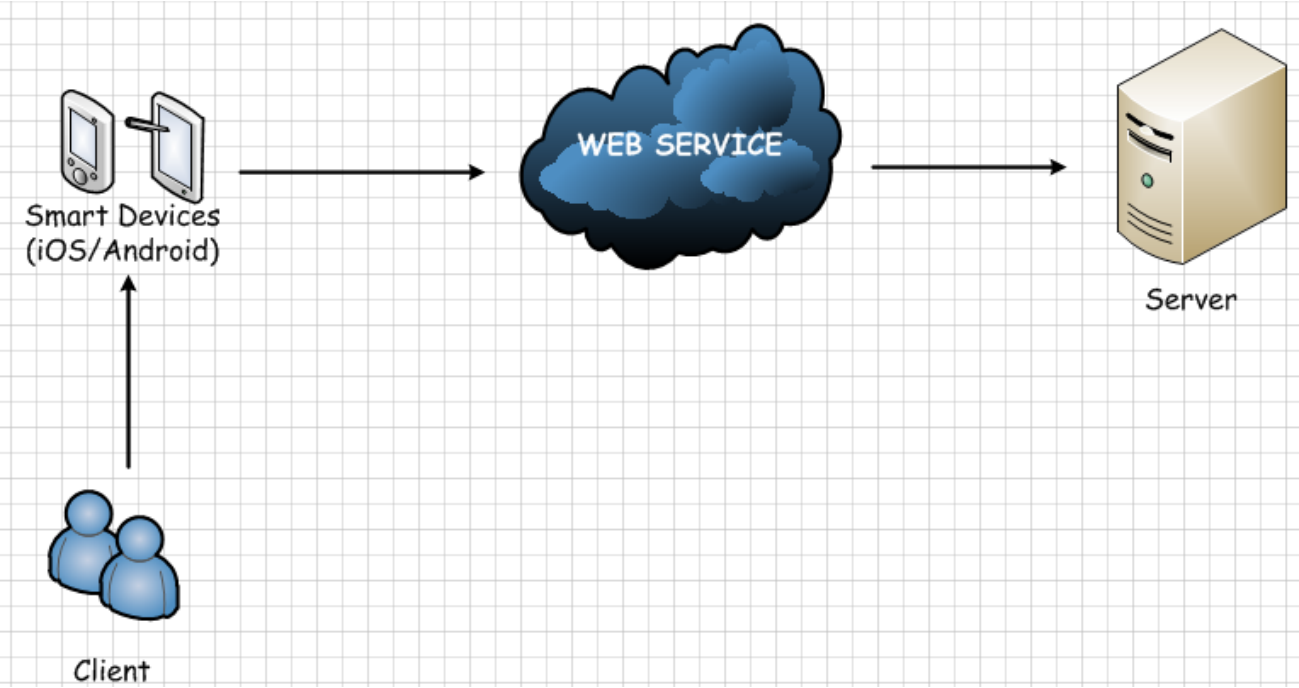
The above diagram shows the Client open the application and then successfully connects to the server to receive the latest information.



The above diagram shows the Client open the application and not be able to connect to the server due to some connection problem. The app will use the information from the mobile local database since last update.

The way of client and server to communicate

The Client and Server communicate through the web service, the Client application get all the data from the database, the database pass the data to the Client through php and AJAX. The Client send feedback to the Server will also use AJAX. This is the web service of our project.



Consistency of data structure between server slide and client slide

We suggest the Client use the mobile in a network acceptable way. The company staff can use our website to update and insert their latest news, information, data into the database. The Client must have good 3G or Wi-Fi network provider to keep their data updated. Each time the user use our app, they can receive the latest information automatically. Besides, we will use push notification to notice our Client that there are some updated information and request them to update our app. This method used to keep the data consistency between server side and client side.

The structure of the mobile application

We decided the mobile application into different layer. The first view show the group of the company, it display a list of the branch that belongs to the company with their icon and name. After the client choose the company, it display the functions that we provided for the client, the functions will be discussed after.

The image on the left hand side shows the home page of the mobile application in iOS. And the image on the right hand side shows the home page of the Android. Since the mobile application in iPhone and Android is the same and they will provide the same functions, therefore we provide the iPhone simulator screen capture for elaborate the processes.

The functions of the mobile application

Restaurant News

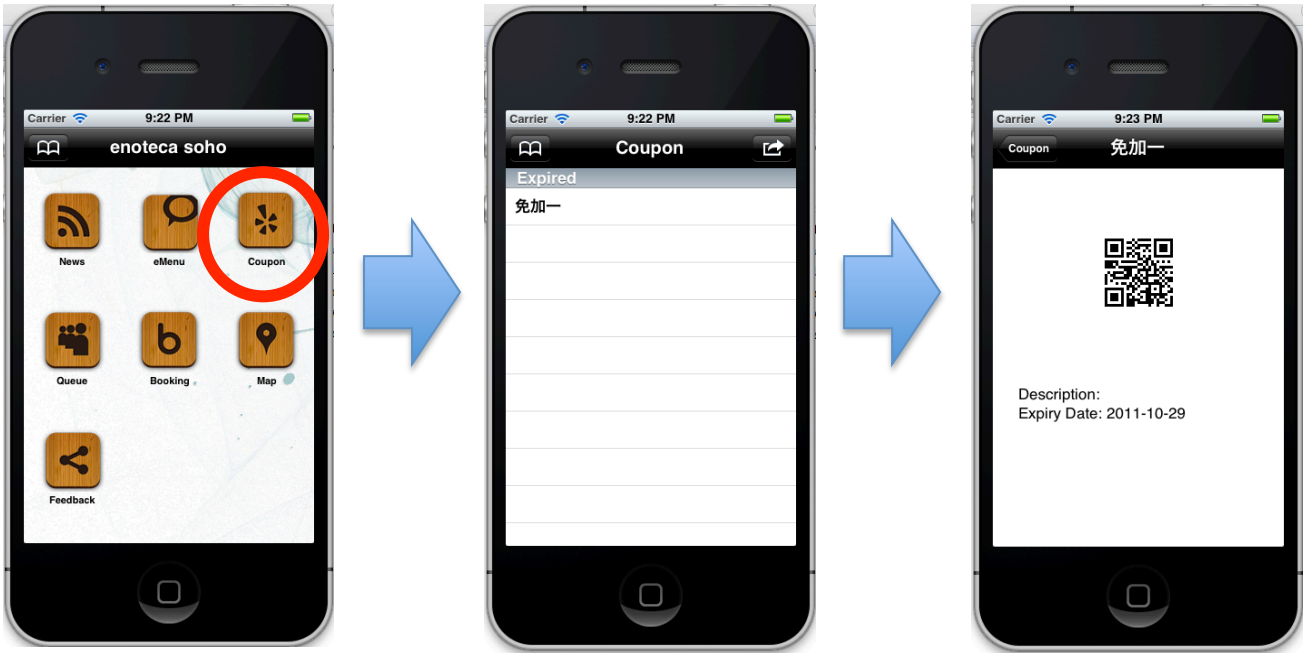
The company can update their restaurant information, latest promotion details, discount details, that they would like to share with client or notice to the client. The client can use their mobile app to view the latest news. The news will be store in our database.

Electrical Menu

The company can update their menu in the website that we provided for the staff. After that, the client can view the food menu on their mobile app.

Electrical Coupon

The company can provide their electrical coupons (E-coupon) to the customer. There are two type of E-coupon, one is unlimited and the other is limited. The unlimited E-coupon means that the client can use it at any time and each time can receive a discount provided. While the limited E-coupon is only for members or VIP of the company, and there is expiry data and time. After the client use the coupon, the coupon will be marked as used and will never be available until she receive a new one.



The E-coupon contains a QR code that allow us to store data in the code, when the client show the code to the company, the company will use a QR code reader to read the data and update the states of the E-coupon on the server.

The company can provide the image or poster to decorate the E-coupon background or even to change the design.

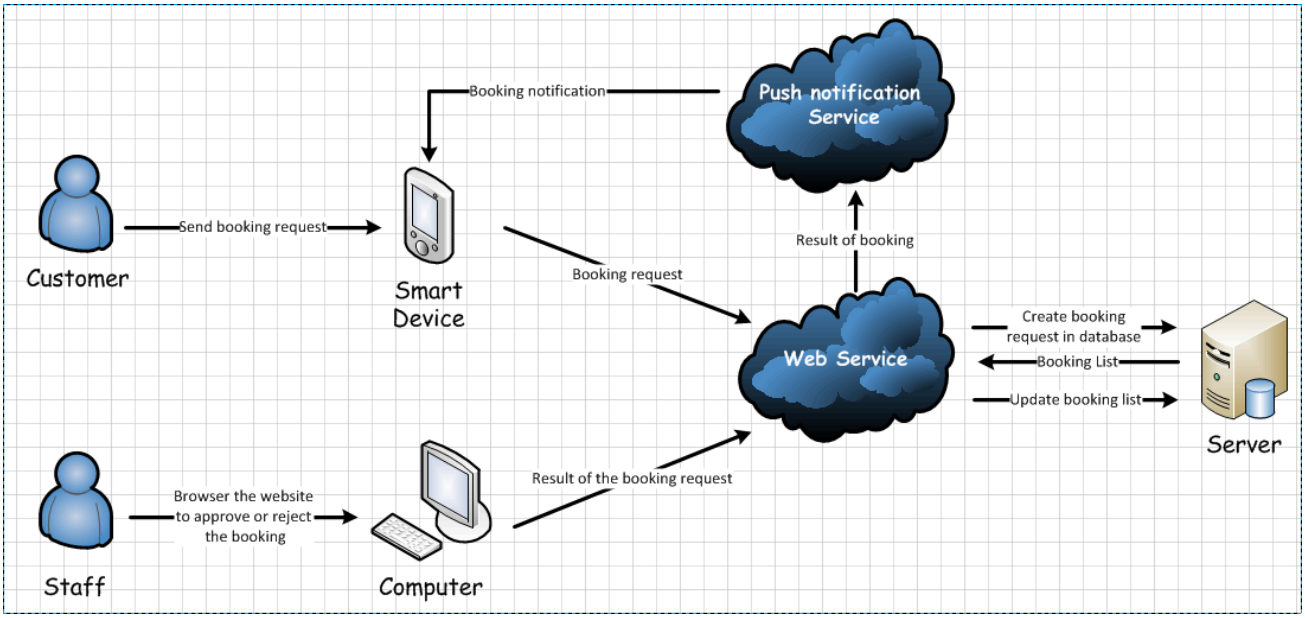
View the Queuing States

This function is similar to the people walk in to the restaurant and gets a queuing ticket and wait. The client can use their mobile to request for the queue and receive a queuing number on the mobile.

Booking Service

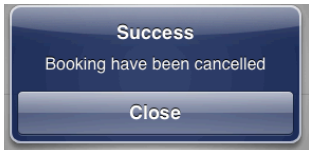
The booking service let the client to reserve tables in the restaurant, the client can use their mobile to make reservation for one months. The data will sent to the server first and wait for the staff to confirm in the restaurant. After the staff confirms the booking, we will use push notification to notice the client.

The following diagram in the next page show the process of the booking service. It includes the push notification. The details of the operation of the push notification will be discussed later.



The following diagrams show the process on the mobile application.

If the client wants to cancel the booking, she can press the “Cancel Booking” button, the booking will be cancel and removed from the booking list in the database.



Location and path direction on the app

The mobile app let the client to use location directing with the GPS on their mobile. First, the mobile will search your GPS or AGPS location on the map and show your position. Second, the client can choose to view the restaurant locations or to direct a path from your position to the location.

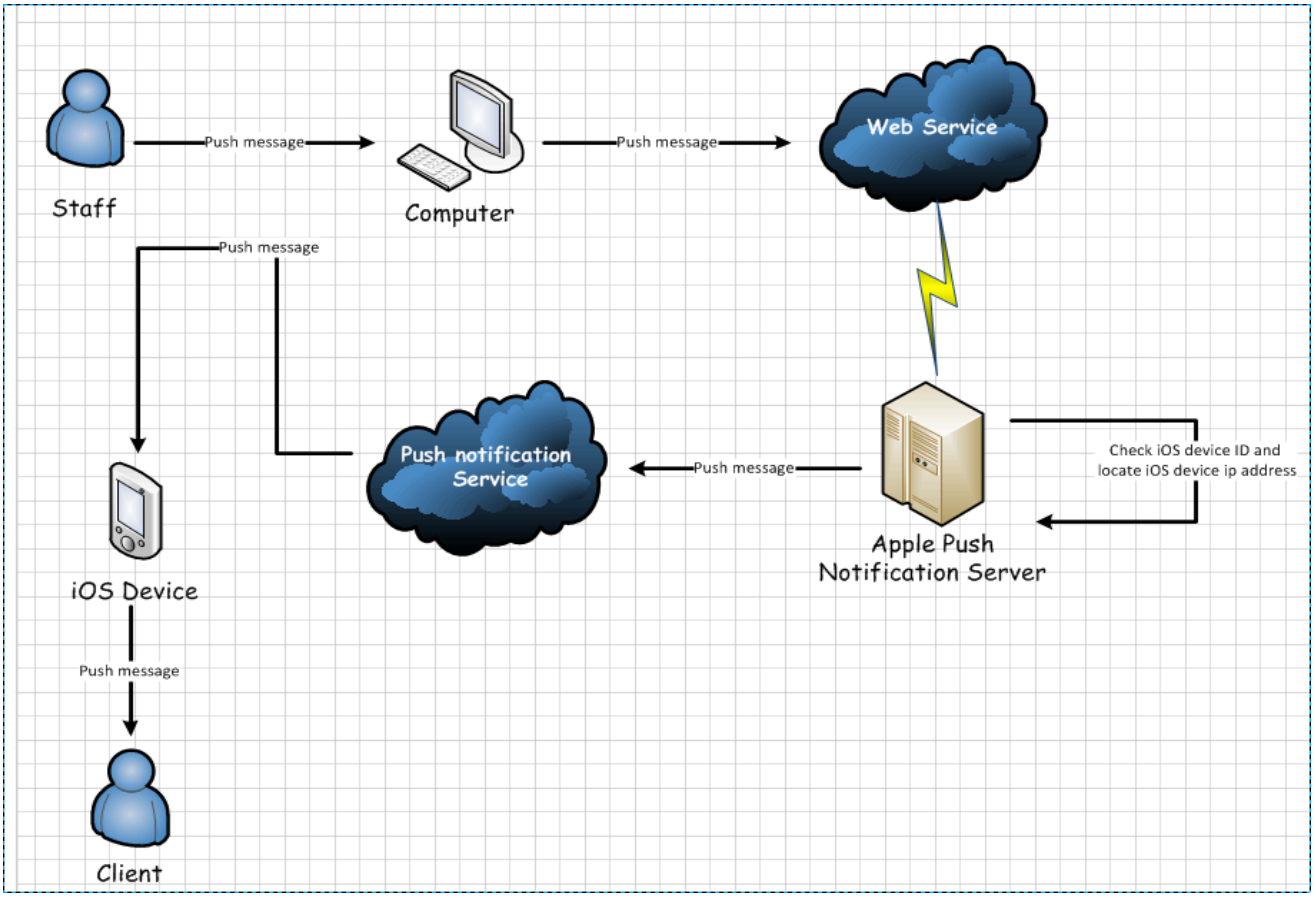
Customer send feedback to the company

The client can use their mobile phone to send feedback to the server. The database will record the received data for further data mining process. The company can create different feedback forms on the website, the forms will automatically loaded to the mobile app when the client start our app.

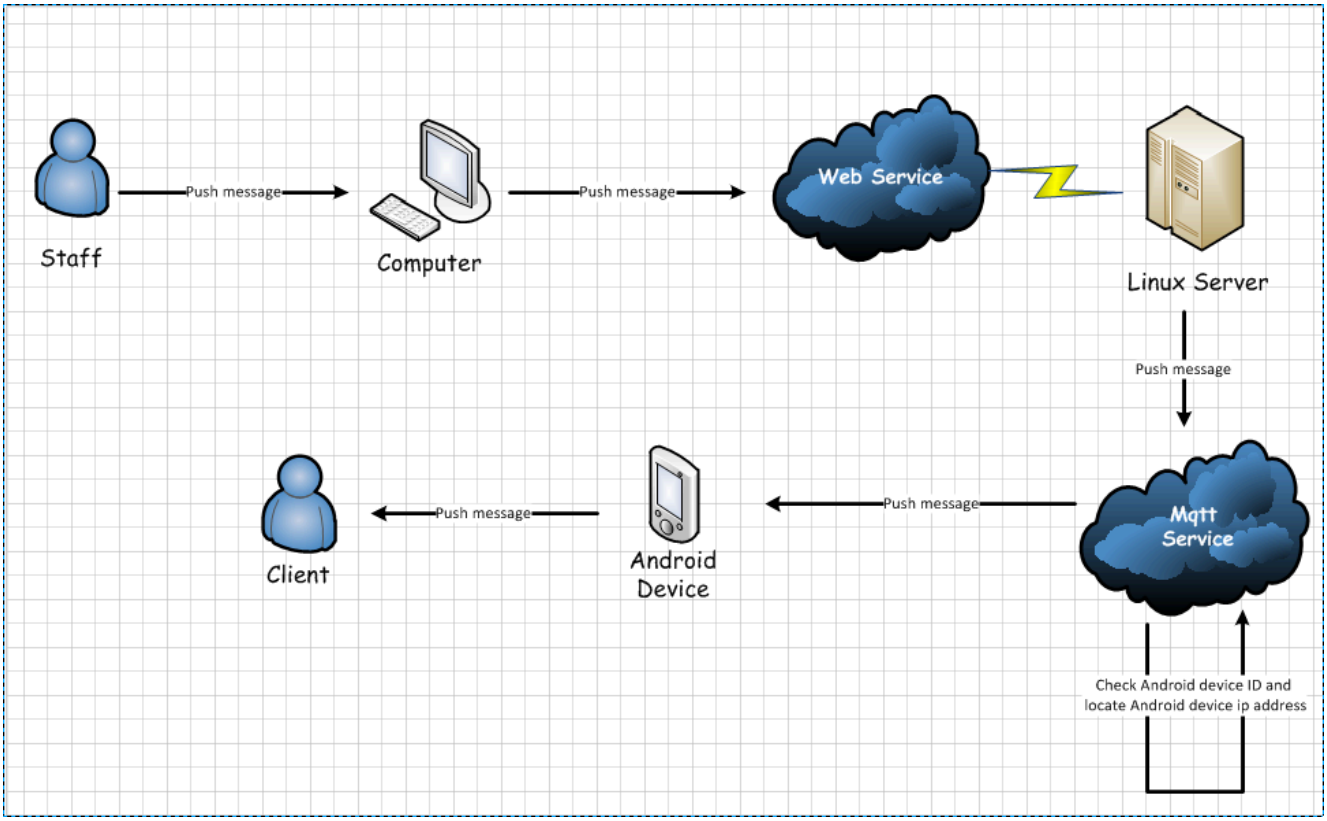
Push notification service

The method for doing the push notification is different in iOS and Android. Apple Inc. has provided a push server for developer to send the device id and message to this server; the server then will locate the iOS device ip address in the world and send the push message through the network. When the iOS client operating the mobile under a network available situation, the message will be pop up to the client.

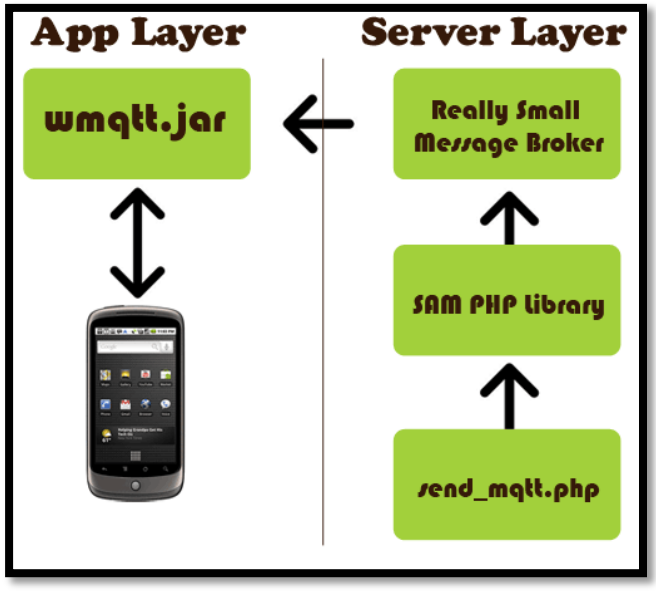
To send push notification message to Android devices are similar, but we need to build up a Linux server and install MQTT service to use the push notification. All the things need to be created and operate by the developers.



The diagram shows the process of sending the push notification to iOS devices.

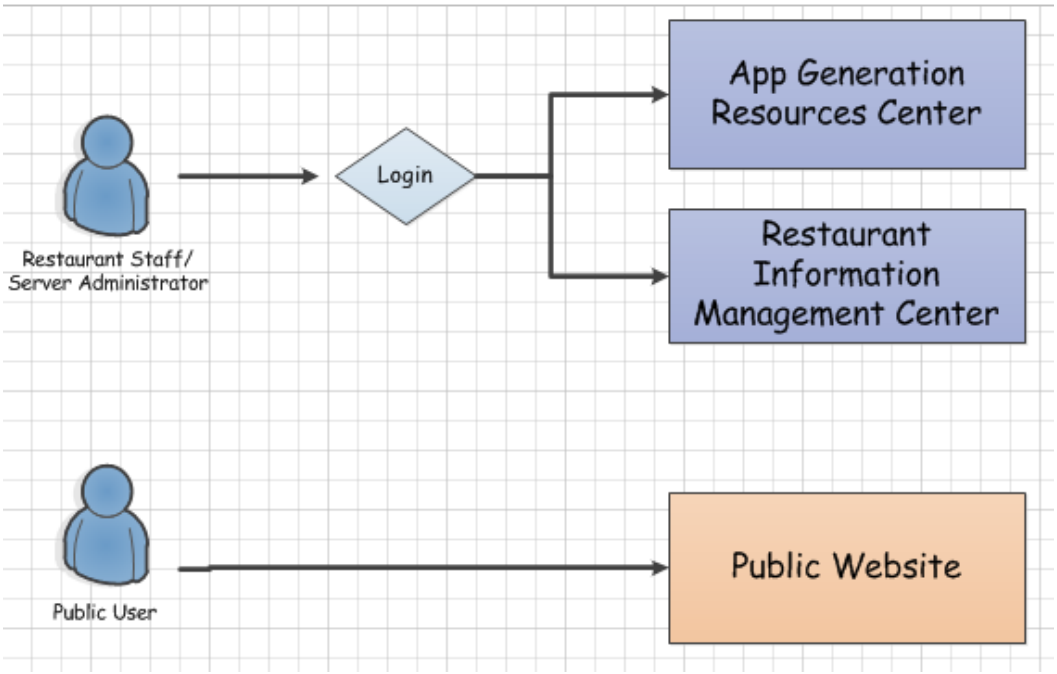


The above diagram shows the process of sending the push notification to Android devices.



The functions of the website system

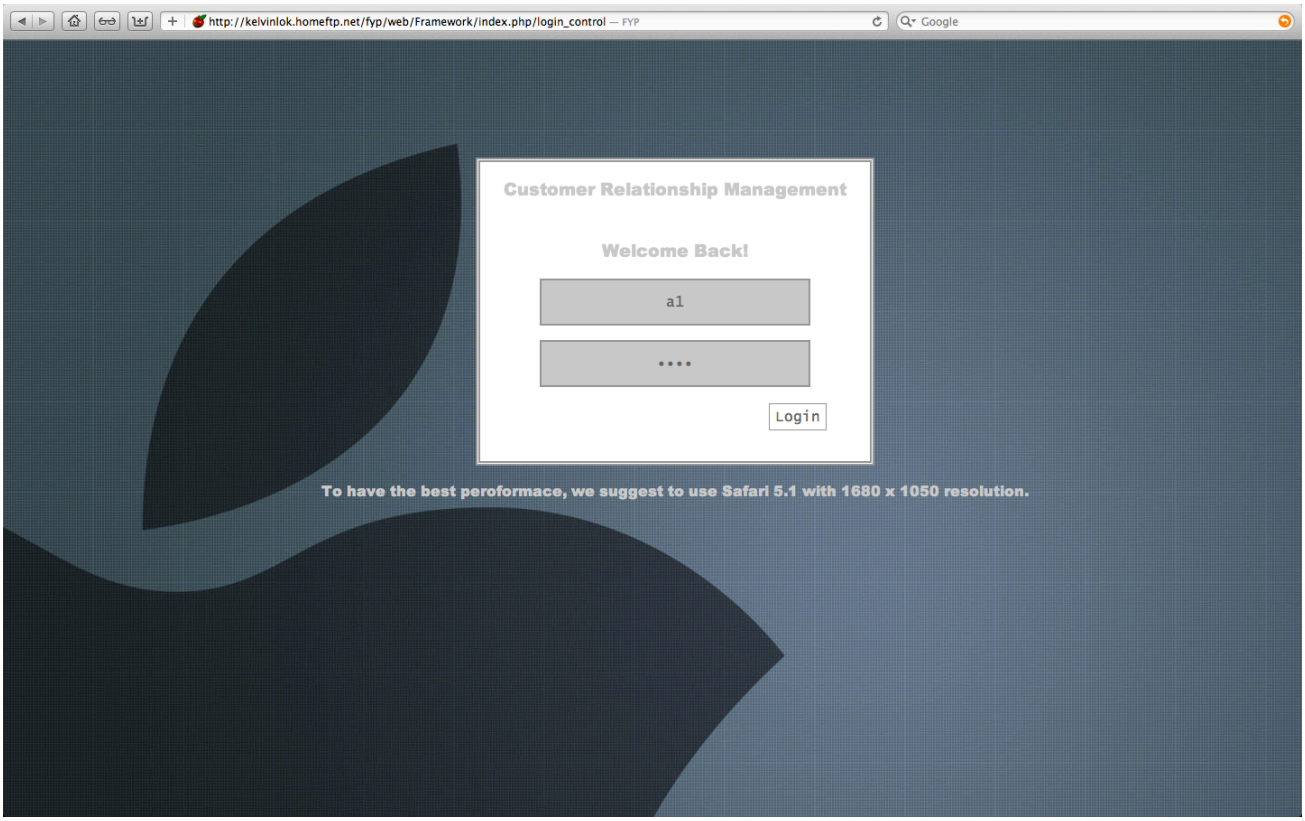
We divided the website into three parts: App Generation Resources Center, Restaurant Information Management Center and Public User Website.



The App Generation Centre Require the staff to login, the password is stored in MD5 on the database for security reasons. Mainly, the Restaurant Information Management Center is used to update the information in the database. The data will be updated in the mobile app when the user uses the app with network connection. The App Generation Resources Center is used for the staff to create their own application for the restaurant, check the image/ background in the application and produce a new mobile application.

The following is the details of the functions of the website for the staff.

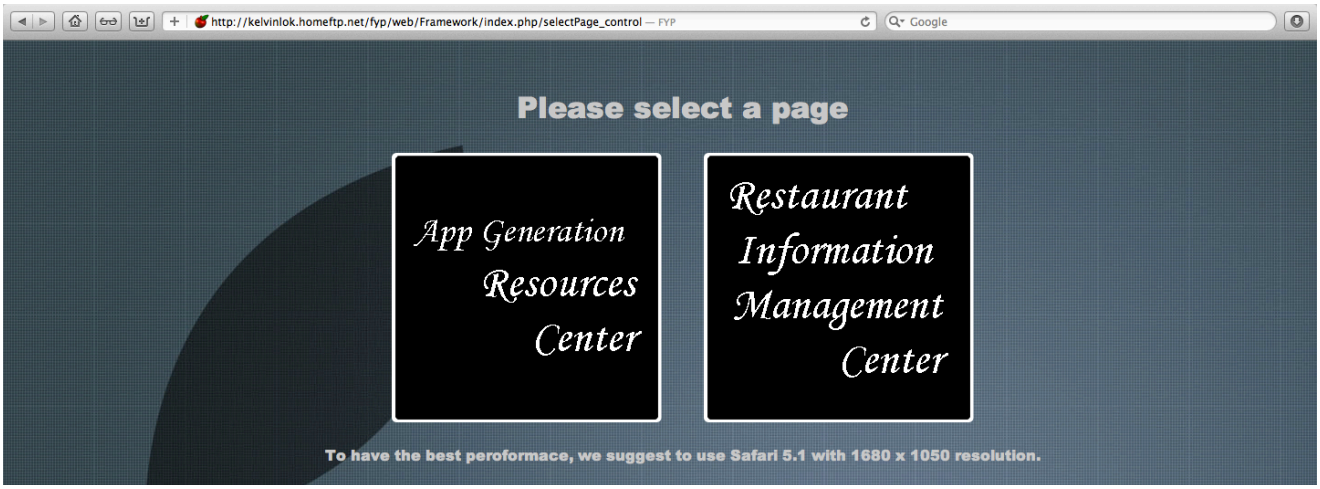
The following diagram is the first index page of our website provided for the staff to login.



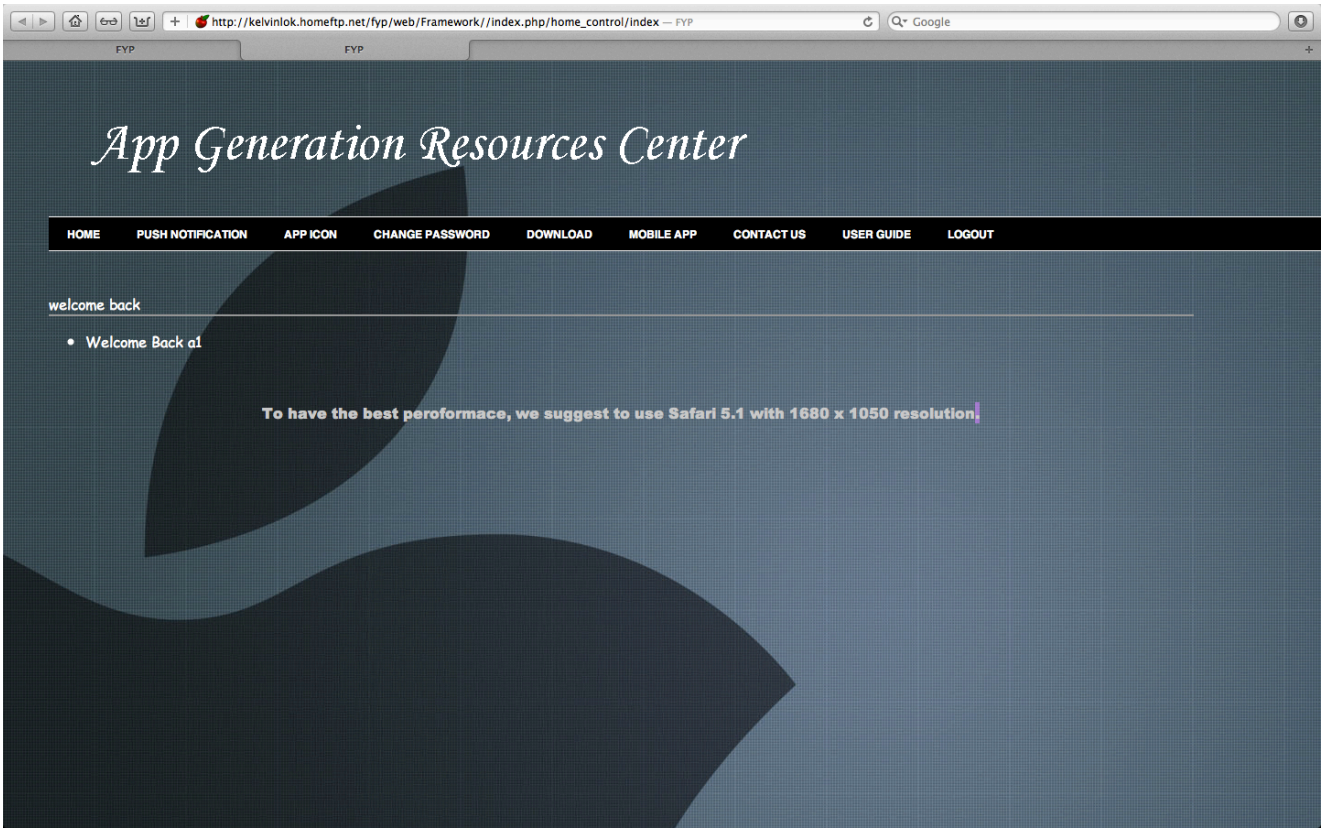
The following shows the hashed password in the database. MD5 is a password encryption technology that provided a one way hashing to a 32 bits Hex Characters. It is widely be used in the Internet.

	user_id	password	email	company_id	group
Edit Inline Edit Copy Delete	a1	81dc9bdb52d04dc20036dbd8313ed055	0	2	0
Edit Inline Edit Copy Delete	a2	81dc9bdb52d04dc20036dbd8313ed055	a2@alanpofyp.com	1	1

After the login process, there are two center provided for the staff showing in the following diagram.



Now, we are going to discuss the detailed function in the App Generation Resources Center.



- Push notification message

The staff can use this function under the push notification tab by insert the message in the textbox and press to "GO" button to send push notification message to the mobile client.

- Upload and manage app icon image

The staff can upload a 32 x 32 pixels image to the website. The image can later be used in the app generation process. The webpage will shows the icon that the staff uploaded. The staff can delete it or download it.

- Change account password

A simple change password administration function is also provided to the staff for keeping the data security.

- User guide

A full user guide of the app generation process will be provided for the staff to download when all the process is finished. It will discuss the beginning of the process to the finish process.

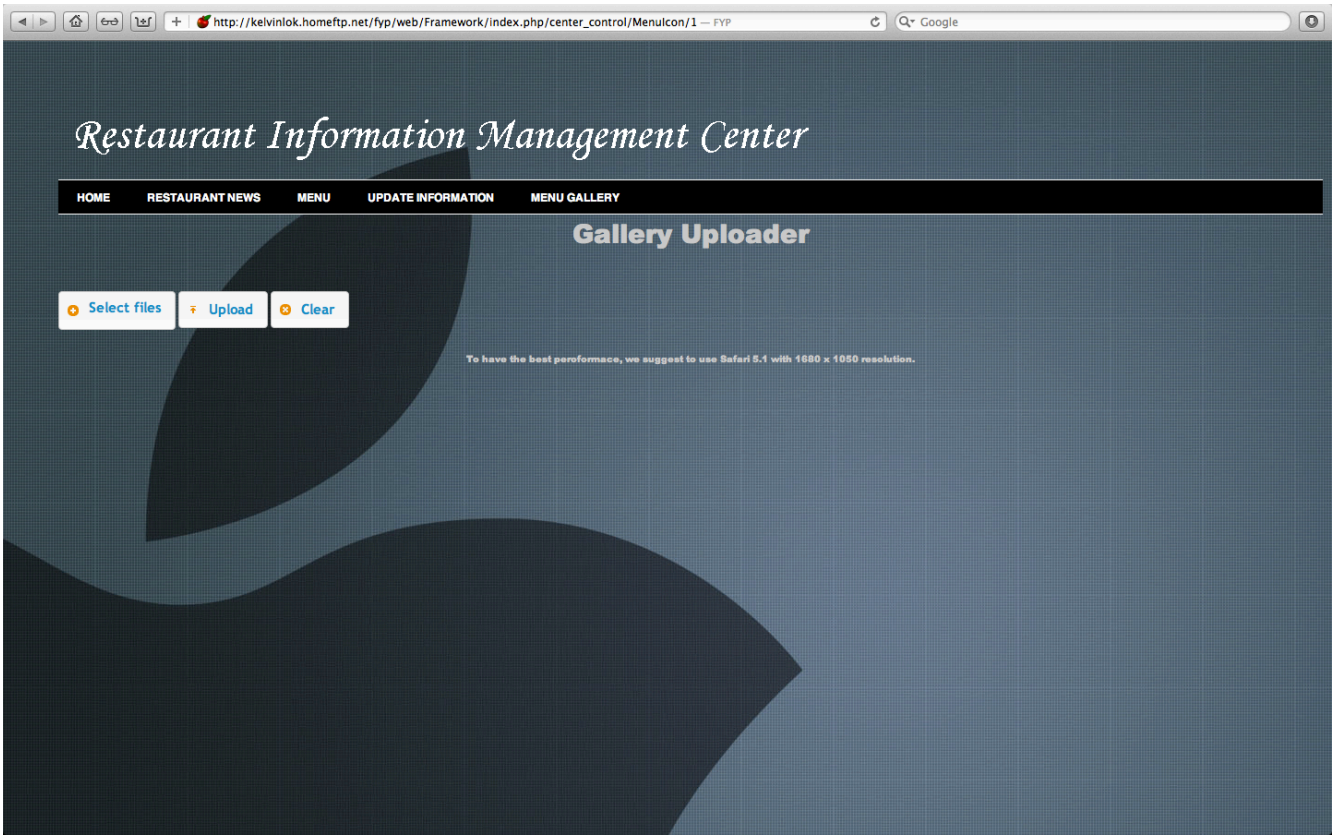
- Mobile App

When the staff click the tab "MOBILE APP", it will direct the user to a next website. That website contains the iPhone and Android simulators. The user can then modify the mobile app, choose the function for the client, and also the modification of the mobile application.

When the user finished their app in the website, they can save the app in the server. After that, select the “DOWNLOAD” tab in the App Generation Resources Center to download the Android project and iOS project. The project can be open in the XCode and Android development tools like Eclipse.

After the user download the project zip file in the website, she can choose to remove the file from the server.

Next, we will introduce the detailed function in the Restaurant Information Management Center.



- Update restaurant news

  The user can update the restaurant news in the database. After the successful update is completed, the latest news will be post in the top in the mobile app.

- Update electrical Menu

  The user can update the e-menu information. She can change the price, the contents or the images on the e-menu. They can also upload new information on the e-menu. It is simple and user-friendly. The Client can use their mobile to view the latest e-menu and do not need to walk in for having the information.

- Staff confirm booking

  The staff can confirm the client booking in this website and notice the client for the booking result.

- Data mining

  Data Mining means the knowledge discovery in database. It is the process of discovering new patterns from large data sets involving methods from statistics and artificial intelligence but also database management. We will use data mining technology to help your company to find out the habits of your customers so that you can design the corresponding business strategies.

- Evaluation of Data

  After data mining is done, we will periodically evaluate the analyzed data. We will change different method of data mining to meet your company's need. Data evaluation usually insists of privacy problem, we will protect our analyzed data well.

How to handle if the client can't connect the server in the specific area

Our mobile application suggested that the user use our app through the Internet connection like 3G, GPRS or Wi-Fi. Therefore, the latest information and data will automatically synchronize with the database server. Once the data is downloaded, it will store in the local mobile database. If the user use the mobile app and resulting in timeout, that means the connection between the mobile and the server is not available. The mobile app will use the local database data. Timeout is one of the most famous technologies that used to check the network connection in this world.

How to support multi-platform client

We use web service the support multi-platform client. A Linux server is needed in this process. To keep the data consistency, we only use one database. The mobile app in different platform is using the same data. Since the program language are different, but we can collect to one database to keep the data consistency. Web service is the best way to support the multi-platform.

How to make client easily to use our mobile application

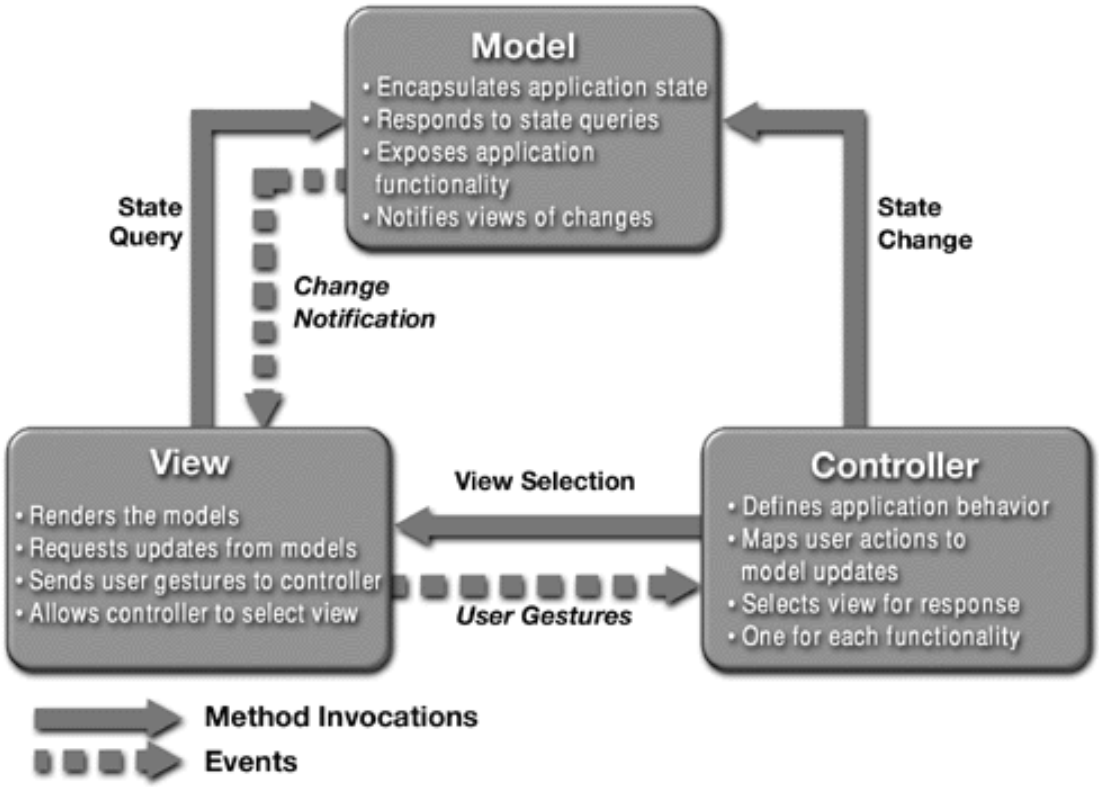
We develop simple function on the mobile app and use a 3x3 grid layout to design the main page in the mobile. All the functions include an image and a word of description. The client can easily use the function by touch on the screen and following some simple instruction. The client does not need to have any information technology knowledge to use our app.

To encourage more people to install our mobile app, we develop and test it in many ways to make the things and jobs simple. And the language we provided for the user is use simple English.

How the way to implement the system in the future

To make the implement in the future easier, we adapt high value of coding and pattern. We use Model-View-Controller to develop all the platforms. It is a reusable, flexible and extendable coding design. To achieve that, different kind of design pattern and methodology will be used. Each data need to be very reusable, otherwise, memory leaking during runtime will be a very common case.

Model–view–controller (MVC) is a software architecture, currently considered an architectural pattern used in software engineering. The pattern isolates "domain logic" (the application logic for the user) from the user interface (input and presentation), permitting independent development, testing and maintenance of each. Model View Controller (MVC) pattern creates applications that separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.



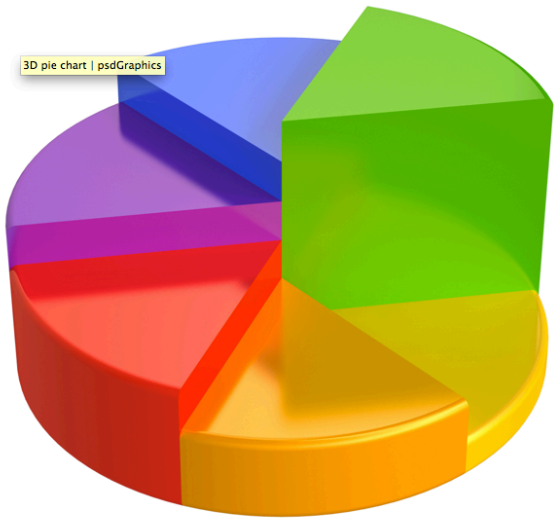
Use Model-view-controller can support multiple views using the same data, because the view is separated from the model and there is no direct dependency from the model to the view, the user interface can display multiple views of the same data at the same time.

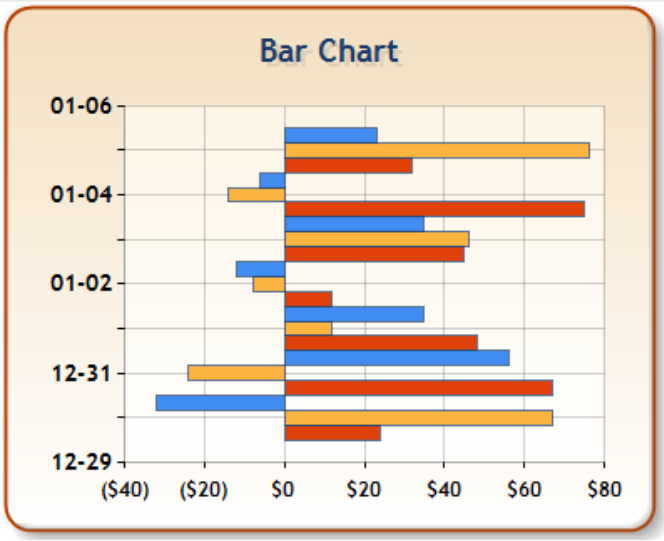
The MVC pattern introduces new levels of indirection and therefore increases the complexity of the solution slightly. It also increases the event-driven nature of the user-interface code, which can become more difficult to debug.

Analysis Report for customers' habits

We will provide a analysis function in the website, we will use the feedback data, user habits, user comments to the restaurant to make a report with diagram to shows to the restaurant staff. We can group the data in different segments: age, gender, menu, habits, date…etc. The website will let the staff to choose the data group in different segment, the system will then generate a written report similar to the SAP system to the staff by analyzing the data.

In this case, we will provide some tools that let the staff to build up a report in graph with the help of some PHP Framework in the Internet. Pie Chart or Bar Chart will also provide to the staff for the improvement in the restaurant to gain more profit.





Constraints

Network stability problem

For our projects, we use a Dyn-dns software to register a free domain name. This can resolve our server IP address into domain name, but a big problem is the connection problems. Sometimes, the network connect was too slow, the mobile app need a few time to update the information to keep the data consistency.

Besides, the push notification in iPhone needs the help of using the Apple Push Notification Server to resolve the device ID. When there are lots of users using the same service, the push notification will be delay. But this problem cannot be solved because the service provider is not controllable.

Limitation of iPhone API

iPhone use iOS which is written by Objective-C computer language. Objective-C is a reflective, object-oriented programming language that adds Smalltalk-style messaging to the C programming language.

Today, it is used primarily on Apple's Mac OS X and iOS: two environments derived from the OpenStep standard, though not compliant with it. Objective-C is the primary language used for Apple's Cocoa API, and it was originally the main language on NeXT's NeXTSTEP OS. Generic Objective-C programs that do not use these libraries can also be compiled for any system supported by gcc or Clang.

Objective-C implementations use a thin runtime system written in C, which adds little to the size of the application. In contrast, most object-oriented systems at the time that it was created used large virtual machine runtimes. Programs written in Objective-C tend to be not much larger than the size of their code and that of the libraries (which generally do not need to be included in the software distribution),

in contrast to Smalltalk systems where a large amount of memory was used just to open a window. Objective-C applications tend to be larger than similar C or C++ applications because Objective-C dynamic typing does not allow methods to be stripped or inlined. Since the programmer has such freedom to delegate, forward calls, build selectors on the fly and pass them to the runtime system, the Objective-C compiler cannot assume it's safe to remove unused methods or to inline calls.

Most of the object in the Objective-C use NSObject as the data type. We must delegate the object and destroy it when we do not need to use it. The memory control in iPhone is very important, because if we do not destroy the object, it will use a lot of memory in the mobile, after the memory address is full the app will be closed and shutdown. It is not user-friendly to the client.

Limitation of Android API

Android is an operating system for mobile devices such as smartphones and tablet computers. It is developed by the Open Handset Alliance led by Google.

Google purchased the initial developer of the software, Android Inc., in 2005. The unveiling of the Android distribution on November 5, 2007 was announced with the founding of the Open Handset Alliance, a consortium of 84 hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. Google released most of the Android code under the Apache License, a free software license. The Android Open Source Project (AOSP) is tasked with the maintenance and further development of Android.

Android consists of a kernel based on the Linux kernel, with middleware, libraries and APIs written in C and application software running on an application framework, which includes Java-compatible libraries, based on Apache Harmony. Android uses the Dalvik virtual machine with just-in-time compilation to run Dalvik dex-code (Dalvik Executable), which is usually translated from Java

bytecode. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices. Developers write primarily in a customized version of Java.

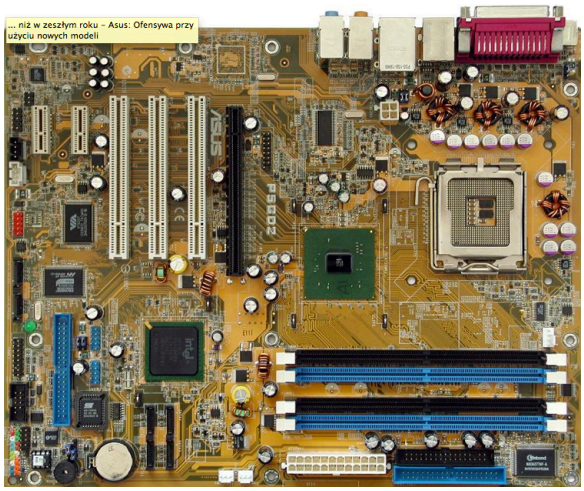
While most Android applications are written in Java, there is no Java Virtual Machine in the platform and Java byte code is not executed. Java classes are compiled into Dalvik executables and run on Dalvik, a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU. J2ME support can be provided via third-party applications.

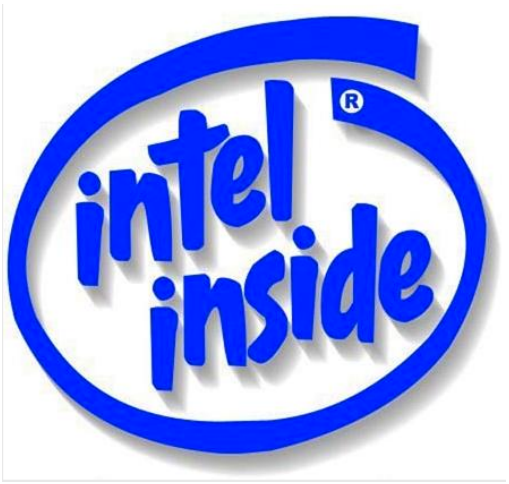
Android do not provide any server to do the push notification service. It is totally different to Apple. If we want to do push notification, we need to build up the server and service ourself. Also, the API level is not same in different devices, we need to use a common API level to support all the mobile devices with android system.

Hardware Facility - Server

- Server

Motherboard : Intel Socket LGA 775 Motherboard
CPU : Intel Pentium IV 3.0Ghz Dual-Core
RAM : 2GB
DVD-ROM : DVD-Read/Write, CR-Read/Write
Hard Disk : 320GB or above
Network : RJ-45 100Mbps/1Gbps Cat5e or above
Power Supply: 380W or above
Operating System : Linux Cent OS 5.6





Hardware Facility - Client

- Client

Supporting Web Browser:

Apple Safari 5 or above,

Google Chrome 11 or above

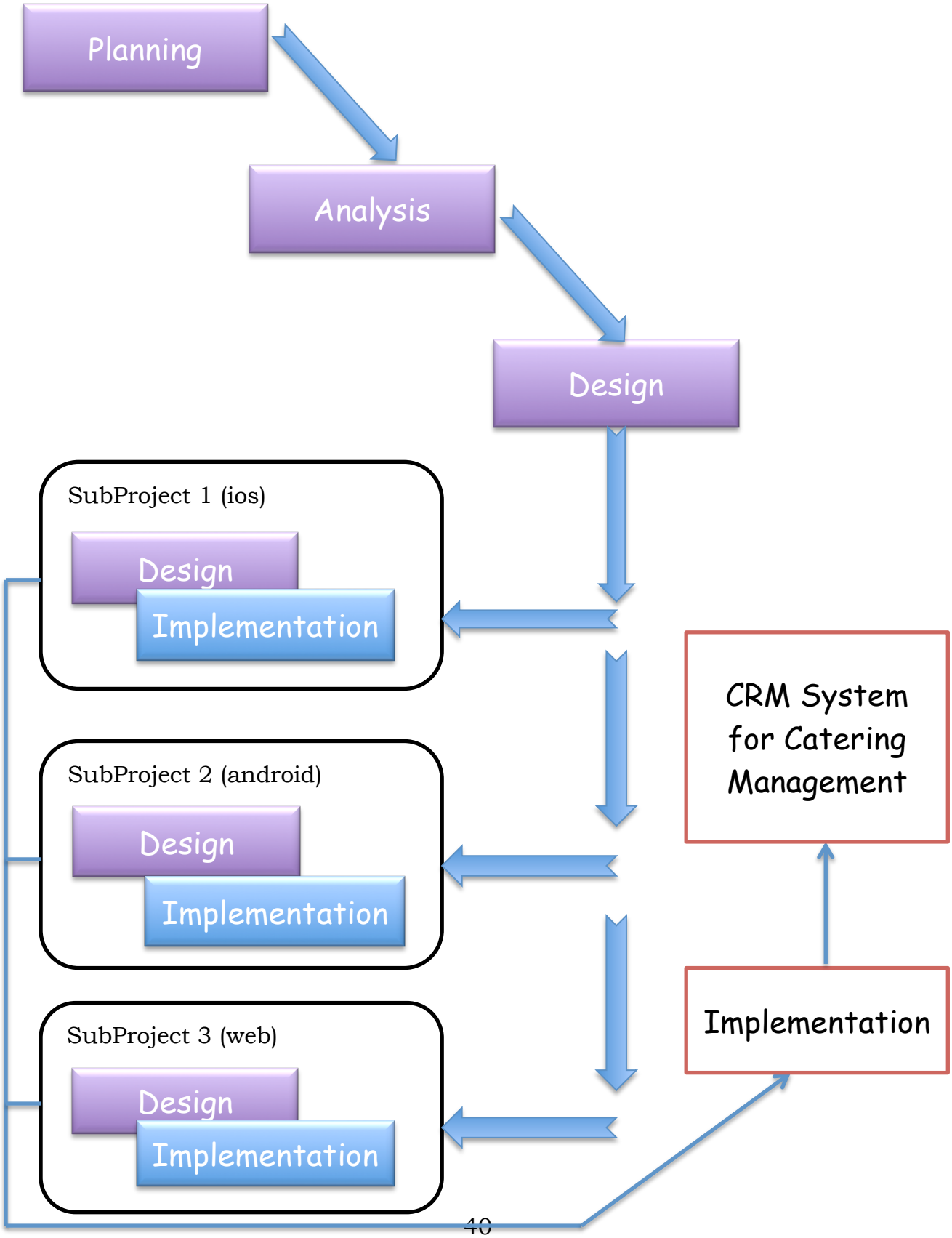
Support Smart Device:

Apple iPhone with iOS 4.2.x or above

Apple iPad with iOS 4.2.x or above

Mobile with Google Android 2.3.x or above

Software Process Model

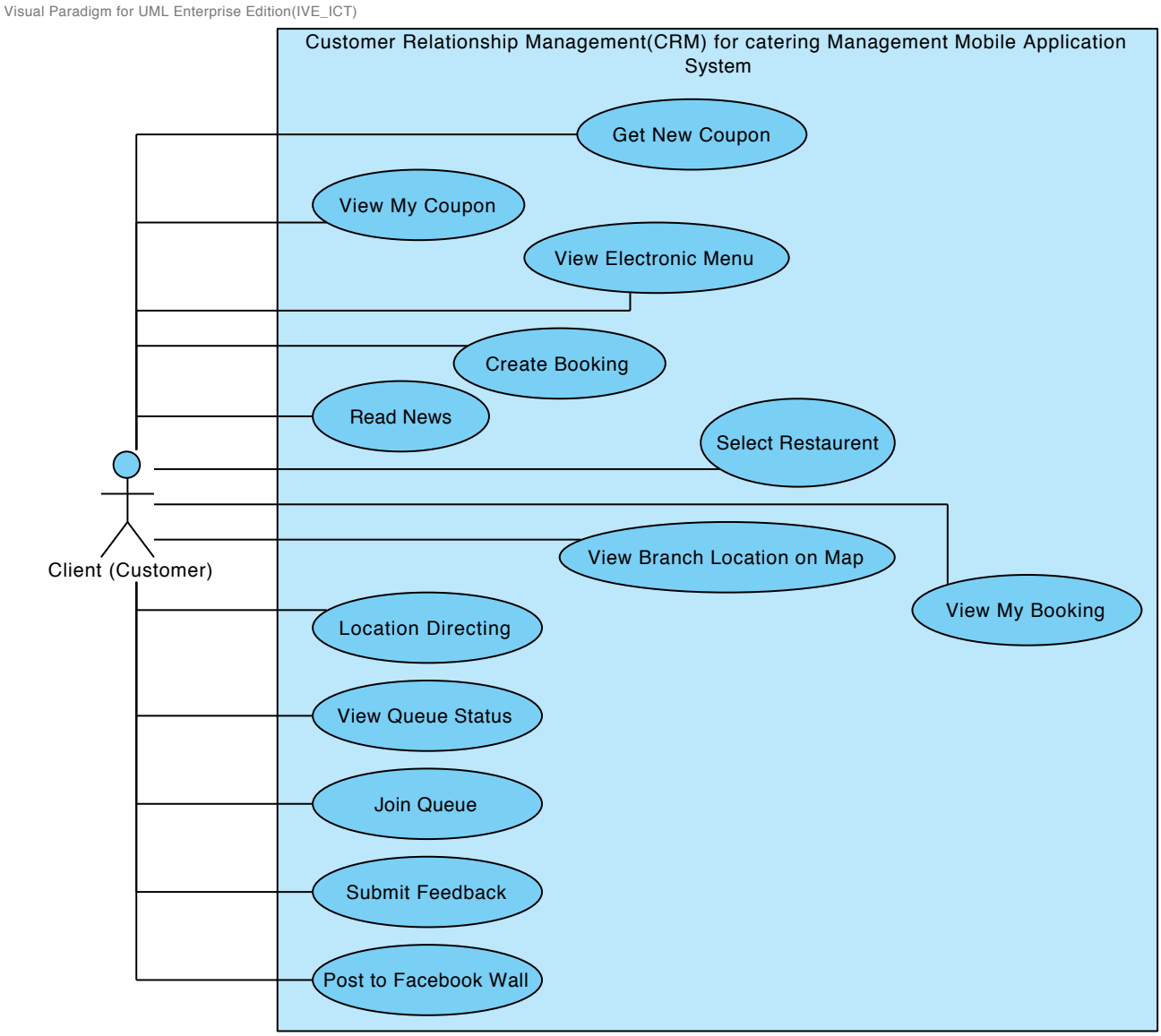


Project Plan

ID	Task Name	Start	Finish	Duration
1	Research Work for CRM in Catering Management	2011/7/1	2011/8/31	62d
2	Project Proposal	2011/8/1	2011/9/2	33d
3	Build up Linux Cent OS 5.6 server	2011/8/5	2011/8/25	21d
4	Website system for staff	2011/9/2	2012/1/19	140d
5	iOS Application development	2011/10/8	2011/12/10	64d
6	Android Application development	2011/10/8	2011/12/10	64d
7	Initial Report	2011/11/1	2011/11/21	21d
8	Competition Presentation I System Brief Description	2012/2/3	2012/2/7	5d
9	Competition Sharing View	2012/2/8	2012/2/8	1d
10	Debugging the system	2012/2/1	2012/3/31	60d
11	Checking Synchronization of the system and sub- system	2012/2/1	2012/3/31	60d
12	Competition Presentation II System Description and Demostratino	2012/5/7	2012/5/7	1d
13	Final Report	2012/4/7	2012/4/7	1d

UML Diagrams

Initial Use Case Diagram – Mobile App



Actor Descriptions – Mobile App

Actor Name:	Client (Customer)
Description:	Client is the end-user interacts with the system by using the mobile application. The team "Customer" is means the customer in the restaurant.

Initial Use Case Descriptions – Mobile App

Use case name:	Select Restaurant
Use case ID:	UC-0001
Actor(s):	Client
Brief description:	A Company can have more than one restaurant. User can chose which one to browse.

Use case name:	Read News
Use case ID:	UC-0002
Actor(s):	Client
Brief description:	A restaurant may have announced news through the web system. The user with this application would be able to read these news.

Use case name:	View Electronic Menu
Use case ID:	UC-0003
Actor(s):	Client
Brief description:	The user can read the menu of restaurant in the mobile, the menu including the images, price, contents, descriptions and type of the food.

Use case name:	Post to Facebook Wall
Use case ID:	UC-0004
Actor(s):	Client
Brief description:	After the user browsed a menu item on the mobile, they can share or recommend it with their friends by posting on its Facebook Wall. Or the user can post the whole menu on their Facebook Wall. Therefore, other friends can view your sharing.

Use case name:	View My Coupon
Use case ID:	UC-0005
Actor(s):	Client
Brief description:	The user can view the electric coupon they received. The user can view the type of coupon, discount offer and expiry date in this case.

Use case name:	Get New Coupon
Use case ID:	UC-0006
Actor(s):	Client
Brief description:	A Restaurant may publish coupons periodically. And the restaurant sets the type of the coupon. The user can receive the coupons.

Use case name:	View Queue Status
Use case ID:	UC-0007
Actor(s):	Client
Brief description:	Many users may be queuing for a table in the restaurant. The users can view their position in the queue under a network available situation on the mobile.

Use case name:	Join Queue
Use case ID:	UC-0008
Actor(s):	Client
Brief description:	When a restaurant is full and certainly unavailable, the user can use the mobile function to join the queue to wait for a table in the restaurant.

Use case name:	View My Booking
Use case ID:	UC-0009
Actor(s):	Client
Brief description:	After the user send the booking to the restaurant, she can view the result in the mobile.

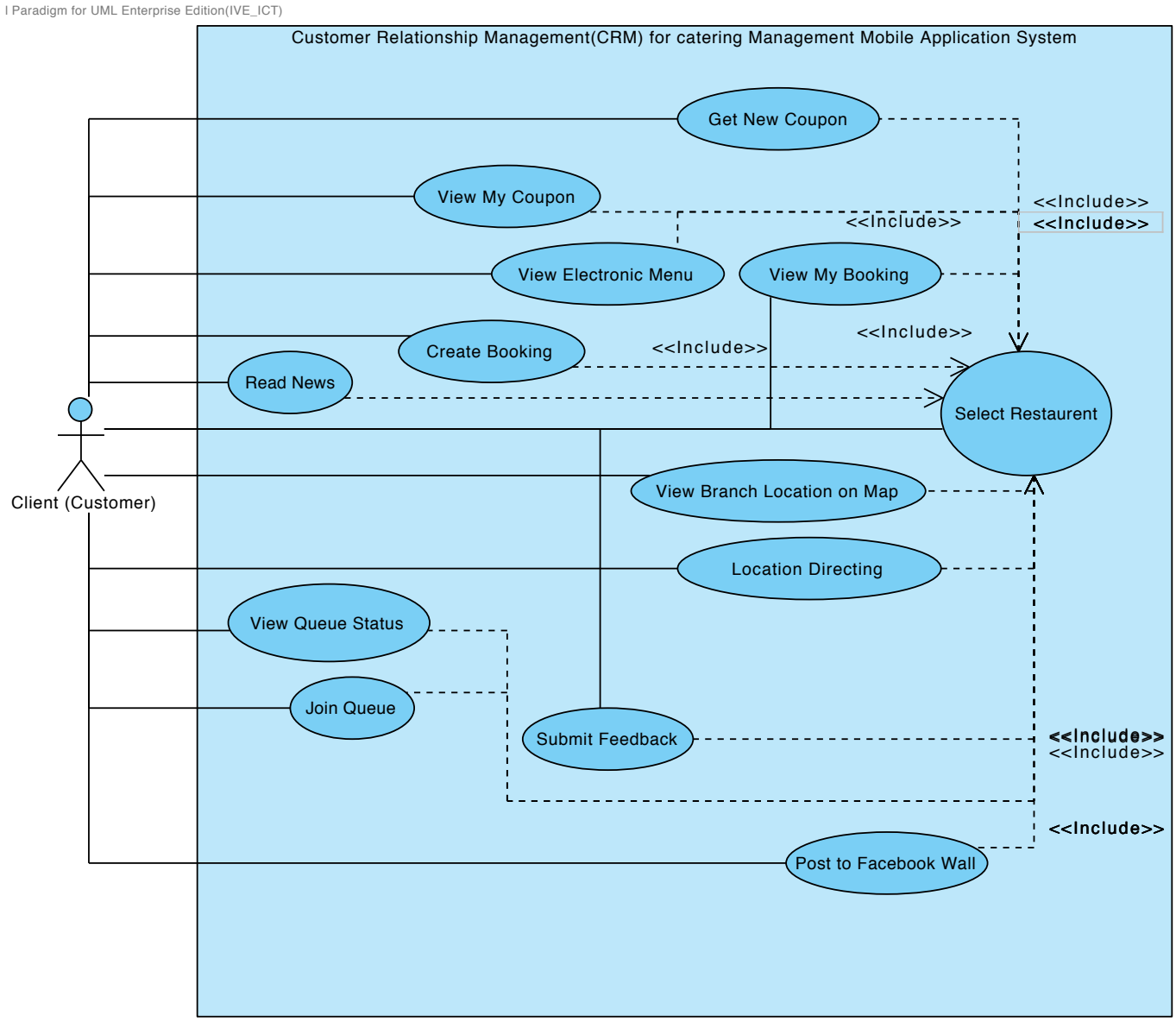
Use case name:	Create Booking
Use case ID:	UC-0010
Actor(s):	Client
Brief description:	The user can send a book request for the table for a specific time. After the user send the booking, she need to wait for the reply from the restaurant staff.

Use case name:	View Branch Location on Map
Use case ID:	UC-0011
Actor(s):	Client
Brief description:	The user can find the position of the branches of the restaurant on the map.

Use case name:	Location Directing
Use case ID:	UC-0012
Actor(s):	Client
Brief description:	The user can make the path directing to the position of the selected branch of the restaurant. She can choose to path in the way by walk, car or transport. This function require the GPS access on the mobile to get the actually position.

Use case name:	Submit Feedback
Use case ID:	UC-0013
Actor(s):	Client
Brief description:	The user can submit feedback to restaurant after they finished the feedback form in the mobile.

Refined Use Case Diagram - Mobile App



Refined Use Case Descriptions – Mobile App

Use case name:	Select Restaurant
Use case ID:	UC-0001
Actor(s):	Client
Brief description:	A Company can have more than one restaurant. User can chose which one to browse.
Preconditions:	User has installed our mobile application.
Flow of events:	1. User opens our application in the mobile. 2. System automatically check the network connect and synchronous with the database server. 3. System shows the available Restaurant and wait for user input. 4. User selects a restaurant by clicking on it. 5. System shows the available functions of that restaurant to the user.
Postconditions:	/
Alternative flows and exceptions:	For the first launch, a network connection is required to receive the Restaurant data from the database server. Otherwise, the application will throw the exception and no data will be shown to the user.
Non-behavior requirements:	/
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	Read News
Use case ID:	UC-0002
Actor(s):	Client
Brief description:	A restaurant may have announced news through the web system. The user with this application would be able to read these news.
Preconditions:	The mobile is under a network available situation.
Flow of events:	1. Include (select Restaurant). 2. The user click the button “News” 3. The system shows a list of news title and grouped by date. 4. The user selects a news title. 5. The system shows the detail (title, content) of that news. 6. If the user wants to view another news, she can press the back button to return to the “News” page.
Postconditions:	/
Alternative flows and exceptions:	If the network is not available, the system will show the last updated news stored in the mobile local database.
Non-behavior requirements:	/
Assumptions:	The user use iPhone or Android devices.
Issue:	/
Source:	Functional Requirement from the Proposal.

Use case name:	View Electronic Menu
Use case ID:	UC-0003
Actor(s):	Client
Brief description:	The user can read the menu of restaurant in the mobile, the menu including the images, price, contents, descriptions and type of the food.
Preconditions:	The mobile is under a network available situation.
Flow of events:	1. Include (select Restaurant). 2. The user clicks the button “eMenu”. 3. The system shows a list of menu items including image, name and price of a food and grouped by menu type (Eg. Main Dish, Soup, Desert..etc). 4. The user selects an item that she wants to view. 5. The system shows the detailed information with a full size image, name, type and price of that item.
Postconditions:	/
Alternative flows and exceptions:	If the network is not available, the system will show the last updated news stored in the mobile local database.
Non-behavior requirements:	/
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	Post to Facebook Wall
Use case ID:	UC-0004
Actor(s):	Client
Brief description:	After the user browsed a menu item on the mobile, they can share or recommend it with their friends by posting on its Facebook Wall. Or the user can post the whole menu on their Facebook Wall. Therefore, other friends can view your sharing.
Preconditions:	User has a Facebook Account.
Flow of events:	1. Include (select Restaurant) 2. The user clicks the button “eMenu”. 3. The system shows a list of menu items (image, name and price of a food) and grouped by type. 4. The user selects an item. 5. The system shows the detail (a full size image, name, type and price) of that item. 6. The user clicks the button “share”. 7. A login screen of Facebook is prompt. 8. The user enters his/her Facebook account username and password to login. 9. The system will post the food detail to their Facebook Wall.
Postconditions:	A Fool detail is posted to the user’s Facebook wall.
Alternative flows and exceptions:	If the network is not available, the system will show the last updated news stored in the mobile local database.
Assumptions:	The user use iPhone or Android devices.

Use case name:	View My Coupon
Use case ID:	UC-0005
Actor(s):	Client
Brief description:	The user can view the electric coupon they received. The user can view the type of coupon, discount offer and expiry date in this case.
Preconditions:	The user have at least one coupon
Flow of events:	1. Include (select Restaurant). 2. The user clicks the button “Coupon”. 3. The system shows a list of coupons he / she owned. 4. The user selects a coupon. 5. The system shows the detail (QRcode, Description and Expiry date) of that coupon.
Postconditions:	/
Alternative flows and exceptions:	If the network is not available, the system will show the last updated news stored in the mobile local database.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	Get New Coupon
Use case ID:	UC-0006
Actor(s):	Client
Brief description:	A Restaurant may publish coupons periodically. And the restaurant sets the type of the coupon. The user can receive the coupons.
Preconditions:	Restaurants have the e-coupon.
Flow of events:	1. Include (select Restaurant). 2. The user clicks the button “Coupon”. 3. The system shows a list of coupons he / she owned 4. The user clicks the button of “Get New”. 5. The system shows a list of coupon that is available 6. The user selects a coupon. 7. The system shows the detail (Name, Description and remain) of that coupon. 8. The user click “Get” 9. The system goes back to “View My Coupon” and the coupon is added to my coupon.
Postconditions:	The client received the new coupon.
Alternative flows and exceptions:	If the network is not available, the client cannot receive the new coupon.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	View Queue Status
Use case ID:	UC-0007
Actor(s):	Client
Brief description:	Many users may be queuing for a table in the restaurant. The users can view their position in the queue under a network available situation on the mobile.
Preconditions:	The user has joined the queue.
Flow of events:	1. Include (select Restaurant). 2. The user click the button “Queue” 3. The system shows the current queue no. and user’s position in queue.
Postconditions:	/
Alternative flows and exceptions:	If the network is not available, the client cannot view the queue states.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	Join Queue
Use case ID:	UC-0008
Actor(s):	Client
Brief description:	When a restaurant is full and certainly unavailable, the user can use the mobile function to join the queue to wait for a table in the restaurant.
Preconditions:	/
Flow of events:	1. Include (select Restaurant). 2. The user click the button "Queue" and click the button of "Join Queue". 3. The system shows form for joining the queue. 4. The user enters their name, seat required, their contact phone number and click "Join". 5. The system shows the success message and return to page "Queue".
Postconditions:	The user need join the queue.
Alternative flows and exceptions:	If the network is not available, the client cannot join the queue.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	View My Booking
Use case ID:	UC-0009
Actor(s):	Client
Brief description:	After the user send the booking to the restaurant, she can view the result in the mobile.
Preconditions:	The client sent the booking request before.
Flow of events:	1. Include (select Restaurant) 2. The user clicks the button “Booking” 3. The system shows a list of booking the user has made. 4. The user selects a booking for details. 5. The system shows the details of the selected booking.
Postconditions:	/
Alternative flows and exceptions:	If the network is not available, the client cannot view the booking.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

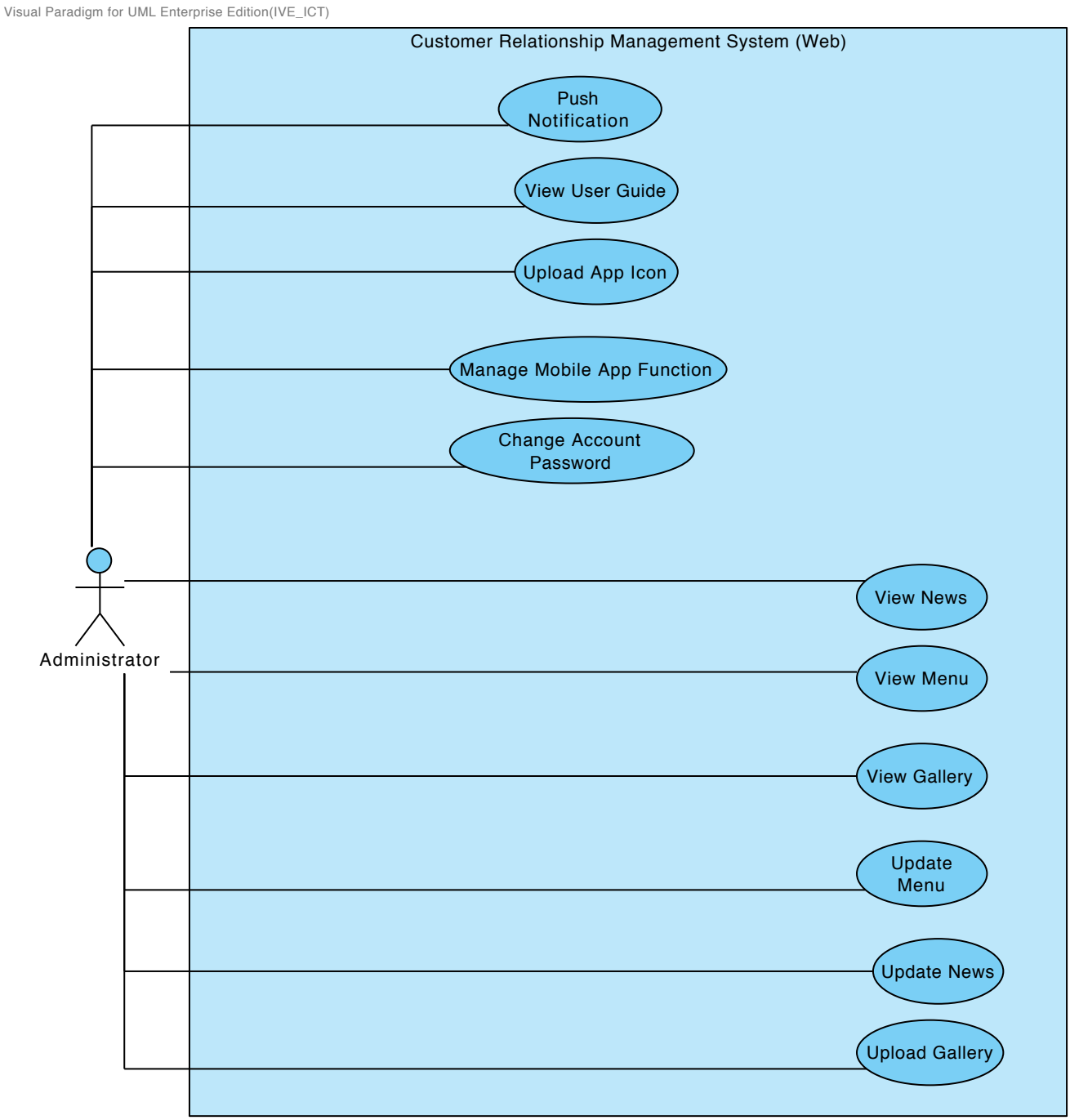
Use case name:	Create Booking
Use case ID:	UC-0010
Actor(s):	Client
Brief description:	The user can book a table for a specific time.
Preconditions:	/
Flow of events:	1. Include (select Restaurant). 2. The user clicks the button “Booking” and clicks the button of “Create Booking”. 3. The system shows form for creates booking. 4. The user enters their name, seat required, their contact phone number, date, special requirement and click “Book”. 5. The system shows the success message and return to page “View Booking”. 6. The user want for the booking states after the result sent back to the client.
Postconditions:	/
Alternative flows and exceptions:	If the network is not available, the client cannot view the booking.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	View Branch Location on Map
Use case ID:	UC-0011
Actor(s):	Client
Brief description:	The user can find the position of the branches of the restaurant on the map.
Preconditions:	1. The user turn on the GPS / AGPS in their mobile devices. 2. The mobile devices contain the mobile application Google Map in Android devices. 3. iPhone should turn on the location service and allow the mobile app to use this service.
Flow of events:	1. Include (select Restaurant). 2. The user clicks the button “Map”. 3. The system shows a map and marks the location of restaurant and user. 4. The user can touch to view more information in the Map.
Postconditions:	The location of the restaurants in marked by a flag in the map and the user position is marked by a people image.
Alternative flows and exceptions:	1. If the network is not available, the client cannot view the map. 2. If the GPS is stop in the mobile, the position may not be correct.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	Location Directing
Use case ID:	UC-0012
Actor(s):	Client
Brief description:	The user can make the path directing to the position of the selected branch of the restaurant. She can choose to path in the way by walk, car or transport. This function require the GPS access on the mobile to get the actually position.
Preconditions:	1. The user turn on the GPS / AGPS in their mobile devices. 2. The mobile devices contain the mobile application Google Map in Android devices. 3. iPhone should turn on the location service and allow the mobile app to use this service.
Flow of events:	1. Include (select Restaurant). 2. The user clicks the button “Map”. 3. The system shows a map and marks the location of restaurant and user. 4. The user selects the restaurant and click route. 5. The user selects the path method by walking, car or public transport. 6. The system shows a path from the user to the restaurant.
Postconditions:	The system returns a colored path in the map directing the user to the restaurant.
Alternative flows and exceptions:	1. If the network is not available, the client cannot view the map. 2. If the GPS is stop in the mobile, the position may not be correct.
Assumptions:	The user use iPhone or Android devices.

Use case name:	Submit Feedback
Use case ID:	UC-0013
Actor(s):	Client
Brief description:	The user can submit feedback to restaurant after they finished the feedback form in the mobile.
Preconditions:	
Flow of events:	1. Include (select Restaurant). 2. The user clicks the button “Feedback”. 3. The system shows a list of available questionnaire. 4. The user selects a questionnaire. 5. The system shows the question of that questionnaire. 6. The user enters the answers. 7. After the user finish the questionnaire, he / she press the “Submit” button. 8. The system sends the data to the database via network. 9. The systems show the success message.
Postconditions:	The system returns a successful message.
Alternative flows and exceptions:	If the network is not available, the client cannot send the feedback.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Initial Use Case Diagram - Website



Actor Description – Website

Actor Name:	Administrator
Description:	Administrator is the end-user interacts with the system by using the website. The team "Administrator" is means the staff in the restaurant, which has a valid user account to do maintenance work.

Initial Use Case Description – Website

Use case name:	Push Notification
Use case ID:	UC002
Actor(s):	Administrator
Brief description:	Administrator can input a notification message, after press the confirm button, a message is send to the mobile device.

Use case name:	View User Guide
Use case ID:	UC003
Actor(s):	Administrator
Brief description:	The administrator can view the user guide of the website through the web system.

Use case name:	Upload app icon
Use case ID:	UC004
Actor(s):	Administrator
Brief description:	Administrator can upload the icons which use in the mobile app, after upload, the system will update the icon of the mobile app.

Use case name:	Manage Mobile App Function
Use case ID:	UC005
Actor(s):	Administrator
Brief description:	Administrator can choose which function will show in the app through the system, after submit, the system will update the function of the app

Use case name:	Change Account Password
Use case ID:	UC006
Actor(s):	Administrator
Brief description:	Administrator can update the password of the login account.

Use case name:	Select Restaurant
Use case ID:	UC007
Actor(s):	Administrator
Brief description:	Before update restaurant information, the administrator needs to choose which restaurant needs to be updated.

Use case name:	View News
Use case ID:	UC008
Actor(s):	Administrator
Brief description:	Administrator can view the news of the restaurant.

Use case name:	View Menu
Use case ID:	UC009
Actor(s):	Administrator
Brief description:	Administrator can view and choose different type of menu – starter, food, drink and dessert.

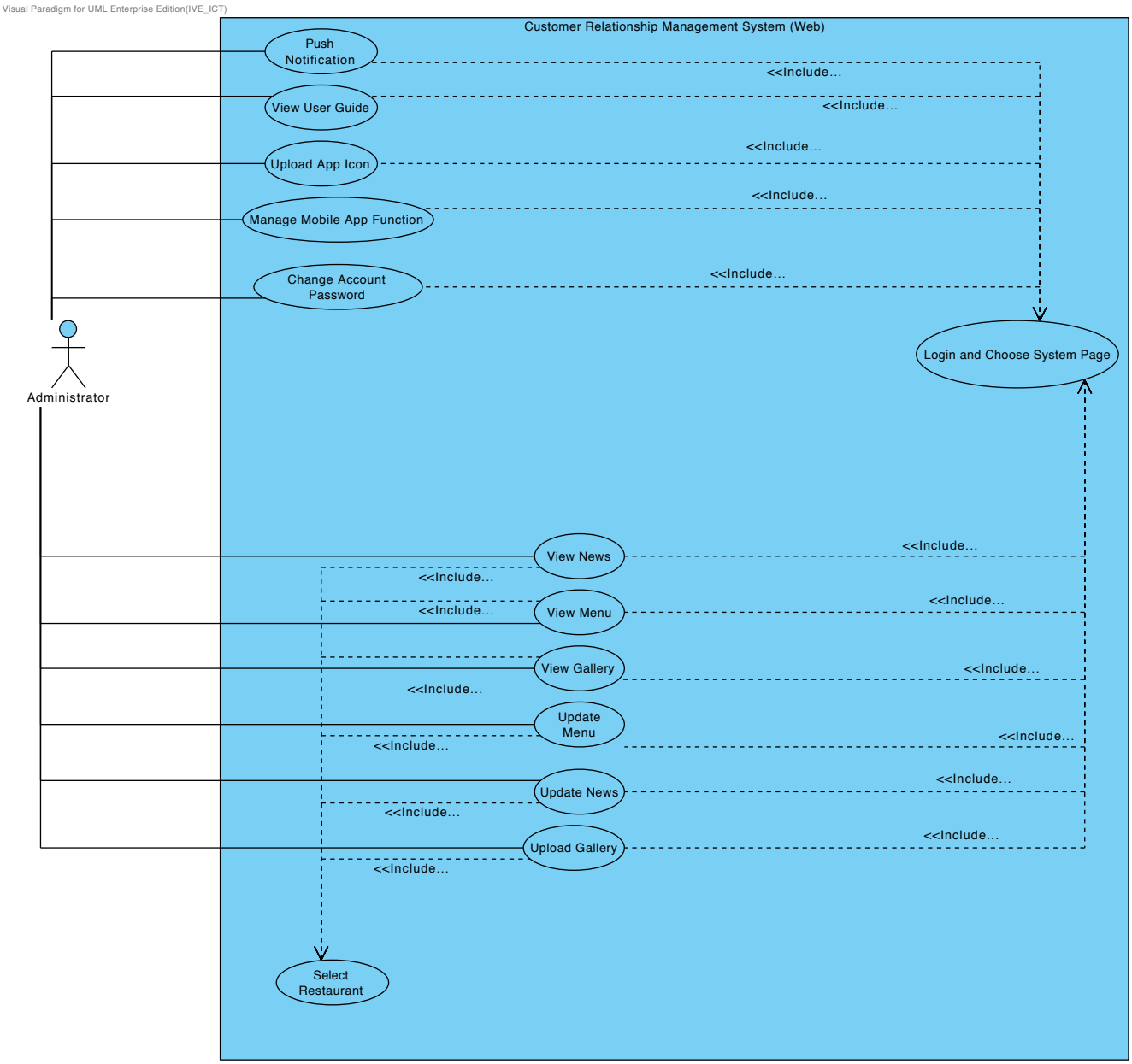
Use case name:	View Gallery
Use case ID:	UC010
Actor(s):	Administrator
Brief description:	Administrator can view the Gallery of the Menu.

Use case name:	Update Menu
Use case ID:	UC011
Actor(s):	Administrator
Brief description:	Administrator can update the menu so that the user who uses the app can view the restaurant latest menu and information.

Use case name:	Update News
Use case ID:	UC012
Actor(s):	Administrator
Brief description:	Administrator can update the restaurant latest information and news through the system, so that the users can view the latest news.

Use case name:	Upload Gallery
Use case ID:	UC013
Actor(s):	Administrator
Brief description:	Administrator can upload the icon or photo of the menu, after that users can view the menu gallery, also administrator can choose the photo which use in the menu when he need to update the menu.

Refined Use Case Diagram - Website



Refined Use Case Description – Website

Use case name:	Login and Choose System Page
Use case ID:	UC001
Actor(s):	Administrator
Brief description:	Administrator login to the system and choose which sub system he need to use.
Pre-conditions:	
Flow of events:	1. Administrator input username and password. 2. The system shows the sub system page. 3. Administrator chooses the sub system.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	A username and password are created.
Issue:	/
Source:	/

Use case name:	Push Notification
Use case ID:	UC002
Actor(s):	Administrator
Brief description:	Administrator can input a notification message, after press the confirm button, a message is send to the mobile device
Pre-conditions:	
Flow of events:	1. (include UC001). 2. Administrator chooses the push notification function. 3. The system shows the input page. 4. Administrator input the content of notification. 5. Administrator presses the send button. 6. The system sends the message to the mobile device.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	The mobile the device had install the mobile app of the restaurant.
Issue:	/
Source:	/

Use case name:	View User Guide
Use case ID:	UC003
Actor(s):	Administrator
Brief description:	The administrator can view the user guide through the web system.
Pre-conditions:	/
Flow of events:	1. (include UC001). 2. Administrator chooses the view user guide function. 3. The system shows the user guide.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/
Issue:	/
Source:	/

Use case name:	Upload app icon
Use case ID:	UC004
Actor(s):	Administrator
Brief description:	Administrator can upload the icons which use in the mobile app, after upload, the system will update the icon of the mobile app.
Pre-conditions:	/
Flow of events:	1. (include UC001). 2. Administrator chooses the upload app icon function. 3. The system shows the upload icon page. 4. Administrator presses the browses button. 5. The system shows the dialog. 6. Administrator chooses the photo. 7. Administrator press the add button.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/
Issue:	/

Use case name:	Manage Mobile App Function
Use case ID:	UC005
Actor(s):	Administrator
Brief description:	Administrator can choose which function will show in the app through the system, after submit, the system will update the function of the app
Pre-conditions:	
Flow of events:	1. (include UC001) 2. Administrator chooses the Manage Mobile App Function. 3. The system shows the mobile app function page. 4. Administrator chooses the functions. 5. The system update the functions of the app.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	The mobile device is connected to the network so that the users can view the updated information.
Issue:	/

Use case name:	Change Account Password
Use case ID:	UC006
Actor(s):	Administrator
Brief description:	Administrator can update the password of the login account.
Pre-conditions:	/
Flow of events:	1. (include UC001). 2. Administrator chooses the update password page. 3. The system shows the update page. 4. Administrator input the required information. 5. Administrator press the submit button. 6. The system updates the information.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/
Issue:	/
Source:	/

Use case name:	Select Restaurant
Use case ID:	UC007
Actor(s):	Administrator
Brief description:	Before update restaurant information, the administrator needs to choose which restaurant needs to be updated.
Pre-conditions:	Administrator choose the restaurant information management system
Flow of events:	1. (include UC001) 2. Administrator selects the Restaurant. 3. The System show the restaurant page
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/
Issue:	/
Source:	/

Use case name:	View News
Use case ID:	UC008
Actor(s):	Administrator
Brief description:	Administrator can view the news of the restaurant.
Pre-conditions:	/
Flow of events:	1. (Include UC001). 2. (Include UC007). 3. Administrator chooses the news function. 4. The system shows the news of the restaurant.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/
Issue:	/
Source:	/

Use case name:	View Menu
Use case ID:	UC009
Actor(s):	Administrator
Brief description:	Administrator can view and choose different type of menu – starter, food , drink and dessert
Pre-conditions:	/
Flow of events:	1. (Include UC001) 2. (Include UC007) 3. Administrator point to the menu bar. 4. The system shows the type of menu. 5. Administrator chooses the sub menu. 6. The system shows the sub menu.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/
Issue:	/
Source:	/

Use case name:	View Gallery
Use case ID:	UC010
Actor(s):	Administrator
Brief description:	Administrator can view the Gallery of the Menu
Pre-conditions:	/
Flow of events:	1. (Include UC001) 2. (Include UC007) 3. Administrator chooses the gallery page. 4. System shows the menu gallery. 5. Administrator chooses and view gallery.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	There are galleries uploaded to the system.
Issue:	/
Source:	/

Use case name:	Update Menu
Use case ID:	UC011
Actor(s):	Administrator
Brief description:	Administrator can update the menu so that the user who uses the app can view the restaurant latest menu and information.
Pre-conditions:	
Flow of events:	1. (Include UC001) 2. (Include UC007) 3. Administrator point to the update information bar. 4. The system shows the sub function. 5. Administrator chooses the update menu function. 6. The system shows the update menu page. 7. Administrator input the update menu and chooses the menu icon. 8. Administrator press the submit button. 9. The system updates the menu information.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/
Issue:	/

Use case name:	Update News
Use case ID:	UC012
Actor(s):	Administrator
Brief description:	Administrator can update the restaurant latest information and news through the system, so that the users can view the latest news.
Pre-conditions:	
Flow of events:	1. (Include UC001) 2. (Include UC007) 3. Administrator point to the update information bar. 4. The system shows the sub function. 5. Administrator chooses the update news function. 6. The system shows the update news page. 7. Administrator input the news information. 8. Administrator press the submit button. 9. The system updates the news information.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/
Issue:	/
Source:	/

Use case name:	Upload Gallery
Use case ID:	UC013
Actor(s):	Administrator
Brief description:	Administrator can upload the icon or photo of the menu, after that users can view the menu gallery, also administrator can choose the photo which use in the menu when he need to update the menu.
Pre-conditions:	/
Flow of events:	1. (Include UC001) 2. (Include UC007) 3. Administrator point to the update information bar. 4. The system shows the sub function. 5. Administrator chooses the Upload Gallery function. 6. The system shows the Upload Gallery Page. 7. Administrator presses the “browse” button. 8. The system shows the dialog. 9. Administrator chooses the photo. 10. Administrator press the add button 11. The system upload the photo to the directory 12. The system show which photo can be update successfully.
Post-conditions:	/
Alternative flows and exceptions:	/
Non-behavior requirements:	/
Assumptions:	/

References

Google

http://www.google.com.hk/

Apple Inc.

http://www.apple.com/

Google Android

http://www.android.com/

Facebook Developer

http://developers.facebook.com/

http://forum.developers.facebook.net/

Wikipedia

http://www.wikipedia.org/

http://en.wikipedia.org/wiki/Objective_c

http://en.wikipedia.org/wiki/Android_(operating_system)

http://en.wikipedia.org/wiki/Apple_Push_Notification_Service

http://en.wikipedia.org/wiki/Push_technology

TOKUDU android push

http://tokudu.com/2010/how-to-implement-push-notifications-for-android/

Codeigniter PHP Framework

http://codeigniter.com/

End of report