

Table of Contents

ABSTRACT	6
THE OBJECTIVE OF THE INTERIM REPORT.....	9
INTRODUCTION OF PROJECT	10
THE DESCRIPTION OF DOCUMENT STRUCTURE.....	11
<i>The problems.....</i>	11
<i>The requirements.....</i>	12
<i>Documentation for problems analysis.....</i>	12
<i>Critical evaluation</i>	12
<i>Detailed project plan</i>	12
<i>References.....</i>	12
<i>Appendices.....</i>	13
THE PROBLEMS	14
BACKGROUND.....	14
PROBLEM DESCRIPTION.....	15
PROBLEM ENVIRONMENT.....	17
PROJECT BASIC OUTLINE & JOB DIVISION	18

BASIC OUTLINE.....	18
JOB OUTLINE DIVISION.....	19
REQUIREMENTS WITH PROPOSED SYSTEM	20
SCOPE OF THE SYSTEM	20
<i>The way of client to archive server's address.....</i>	21
<i>Automatically connect to the server and receiving latest information from the server.....</i>	24
<i>The way of client and server to communicate.....</i>	26
<i>Consistency of data structure between server slide and client slide.....</i>	26
<i>The structure of the mobile application.....</i>	27
<i>The functions of the mobile application.....</i>	28
<i>The functions of the website and web service.....</i>	38
<i>How to handle if the client can't connect the server in the specific area.....</i>	51
<i>How to support multi-platform client.....</i>	51
<i>How to make client easily to use our mobile application.....</i>	52
<i>How the way to implement the system in the future</i>	52
<i>Analysis report for customers' habits.....</i>	54
<i>Brief description of server and database set-up.....</i>	54
NON-FUNCTIONAL REQUIREMENTS	56

<i>A server administrator must be added to the system.....</i>	56
<i>User-friendly GUI for the customer.....</i>	56
<i>Log keeping in database and server</i>	57
<i>Security Design is need in the server side</i>	58
<i>Automatically logout when the user stand-by for more than 15 minutes (website).....</i>	58
<i>Database should be backup frequently.....</i>	59
<i>Use high effectively server</i>	59
<i>Use the stable software tools in the server.....</i>	59
<i>Keep Network Stable</i>	60
CONSTRAINTS.....	61
NETWORK STABILITY PROBLEM	61
LIMITATION OF iPHONE API.....	61
LIMITATION OF ANDROID API.....	62
USE CASES.....	66
USE CASE DIAGRAM OF WEB SERVICE.....	66
USE CASE DESCRIPTION OF WEB SERVICE.....	67
USE CASE DIAGRAM OF MOBILE DEVICES.....	78
USE CASE DESCRIPTION OF MOBILE DEVICES.....	79

CLASS DIAGRAMS.....	91
CLASS DIAGRAM OF iPHONE APPLICATION.....	91
CLASS DIAGRAM OF ANDROID APPLICATION.....	92
CLASS DIAGRAM OF WEB SERVICE	93
ERD DESIGN	94
ERD DIAGRAM.....	94
ERD DESCRIPTIONS.....	95
STATE TRANSITION DIAGRAM.....	110
STATE TRANSITION DIAGRAM OF iPHONE APPLICATION.....	110
STATE TRANSITION DIAGRAM OF ANDROID APPLICATION.....	123
STATE TRANSITION DIAGRAM OF WEB SERVICE	134
SEQUENCE DIAGRAM.....	146
SEQUENCE DIAGRAM OF iPHONE APPLICATION.....	146
SEQUENCE DIAGRAM OF ANDROID APPLICATION.....	159
SEQUENCE DIAGRAM OF WEB SERVICE	172
USER INTERFACE DESIGN	179
iPHONE/ ANDROID APPLICATION UI DESIGN.....	180
WEB SERVICE USER INTERFACE DESIGN.....	188
CRITICAL EVALUATION	191

PROBLEMS AND DIFFICULTIES ENCOUNTERED.....	191
<i>Server side</i>	191
<i>iPhone development</i>	192
<i>Android mobile development</i>	193
DELAYS / CHANGES IN PROJECT SCHEDULE	194
LIMITATIONS OF THE PROPOSED SYSTEM	195
<i>Unable to locate the shop</i>	195
<i>Network stability problem</i>	195
DETAILED PROJECT PLAN.....	197
HARDWARE FACILITY.....	198
REFERENCES.....	200
APPENDICES	202

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 first select a type of the enterprise and then make co-operation with one of them to develop our system basic on their requirements and their needs to help improving their business. The competition decided which company mentor for us to reference, and they decided our group to co-operate with one of the catering company in Hong Kong that is "Enoteca Group".

This project is to develop a multi-platform Customer Relationship Management (CRM) System for catering management. Our system will help boost sales and business, maintaining customers data and performing business analysis in order to help the company to increase their profit.

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 mobile devices, Android tablet...etc. The client side aims at using the mobile application (mobile app) by receive the data from the web database to achieve more the catering company information to their mobile.

In nowadays, we always heard of a word "cloud technology". Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a metered

service over a network, typically Internet. So, our project will also follow the rules of cloud technology to develop the whole set of system.

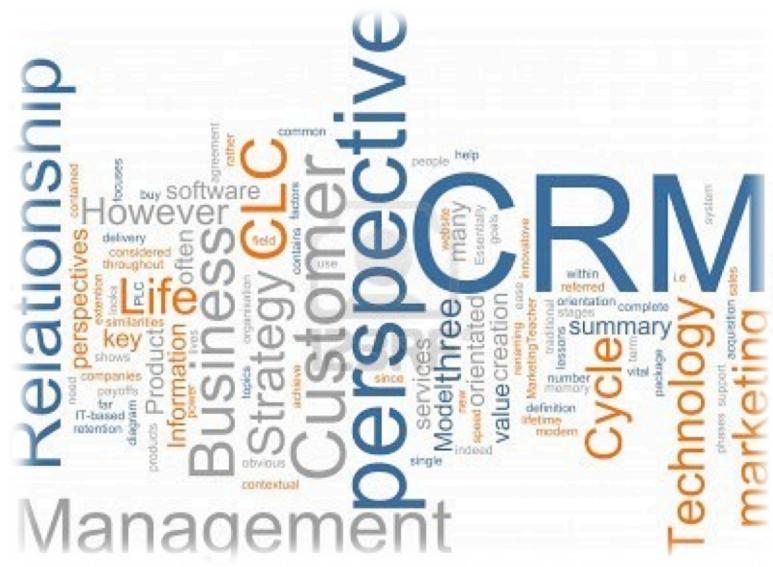
For the server-sided system, the web service will provide editing function about the company information. We will provide the app project configuration tools on our website for multi companies to design their own application on the cloud. It is a whole CRM package of our design. The company can use our system to configure their iOS 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 basically.

In the world up to now there is no any standard Customer Relationship 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. Besides, there are no any tools from the Internet let the people to create a mobile app or mobile 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.

Different company have different needs, we need to provide a Customer Relational Management System to fit the requirement for different company. The CRM system will include some common features and functions in catering management in Hong Kong. So,

this project is a multi company multi platform CRM project that supports the catering management in Hong Kong nowadays.

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.



The Objective of the Interim Report

This report is mainly describes what function the server provided to the staff and what functions the mobile devices will include and how the server work with client. This report consists of two critical. The first half of the report is documentation about the functions in server and client side, the requirements. The second half mostly is the UML modeling which include ERD design, Class Diagram in server side and client side, sequence diagram of the program flow and state machine diagram of the program states.



Introduction of Project

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 target customers.

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.

To provide further development, we should first develop a CRM system for based on the reference mentor Enoteca Group in blue print and then modify it to a CRM system for catering company.

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. UML Diagram will be shown to discuss the program design, how to adopt MVC in our system. We also concerned about the limitation of 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 description of document structure

This report has 7 major sections to help understanding process of development of the new CRM System.

The problems

This part describes the problem that occurring in the company. The problem we meet and the environment. The description of function provided is show a list of function can solve the problem.

The requirements

This part describes the scope of the system, description of the functions provided in both server and client side, the functional requirement and non-functional requirement and data processed by the system.

Documentation for problems analysis

This part is one of the UML modeling which are use case diagrams, use case description, class diagram, state transition diagram and sequence diagrams.

Critical evaluation

Any problems or difficult encountered, any delays or changes in project schedule, limitations of the proposed system, potential difficulties associated with the suggested progression route, etc.

Detailed project plan

This section will firstly show a chart of our project schedule and what deliverables will handout. Finally, the software and hardware we need during the project.

References

The reference links, books, leaflet, and material we reference to for this project.

Appendices

This part is the description that is related to our project but not included before in the report.

The Problems

Background

Our CRM system is designed for the catering management, so it is a Catering CRM System. As reference to the mentor "Enoteca Group" in Hong Kong, their website is "<http://www.enoteca.hk/>", we know that the tis 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 materials ordering system in their shop only.

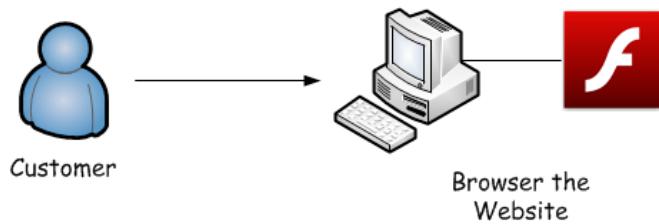
You can see that their website is quite beautiful in graphic design, but the website is used Adobe Flash ActionScript to built up. Therefore, there is no any function to be interacting with the customer.





Figure: The website of <http://www.enoteca.hk>

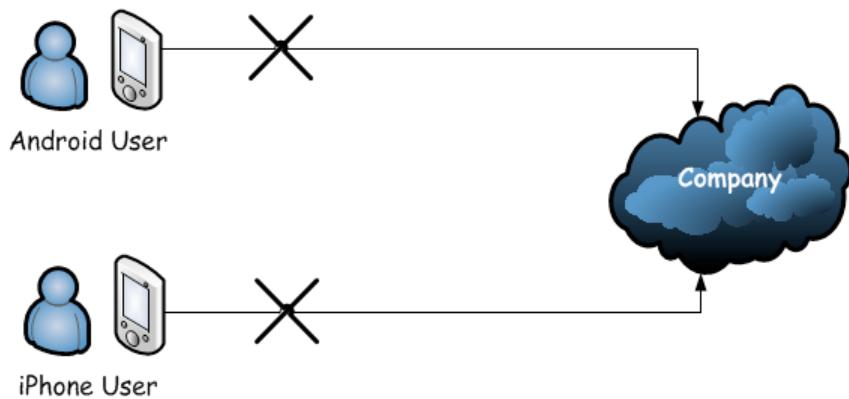
Problem Description



Mainly, the customer cannot get much 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. For the people who want to use iPhone to browse their website, it is not possible because iOS do not support Adobe Flash Player. In nowadays Hong Kong, the group should provide a mobile application to increase the popularity in order to increase the sales simultaneously.



Third, the company does not have any feedback tools or forms on Internet or mobile. If the customer would like to give them feedback of their service, it is difficult. Besides, there are no any statistic or data mining tools to analysis the customers' comment or their order habit. They should provide different ways for the customer to feedback comments about the service, food quality, the range of the meals they provided. After that, group the data together and make some analysis in specific area to increase the profit.

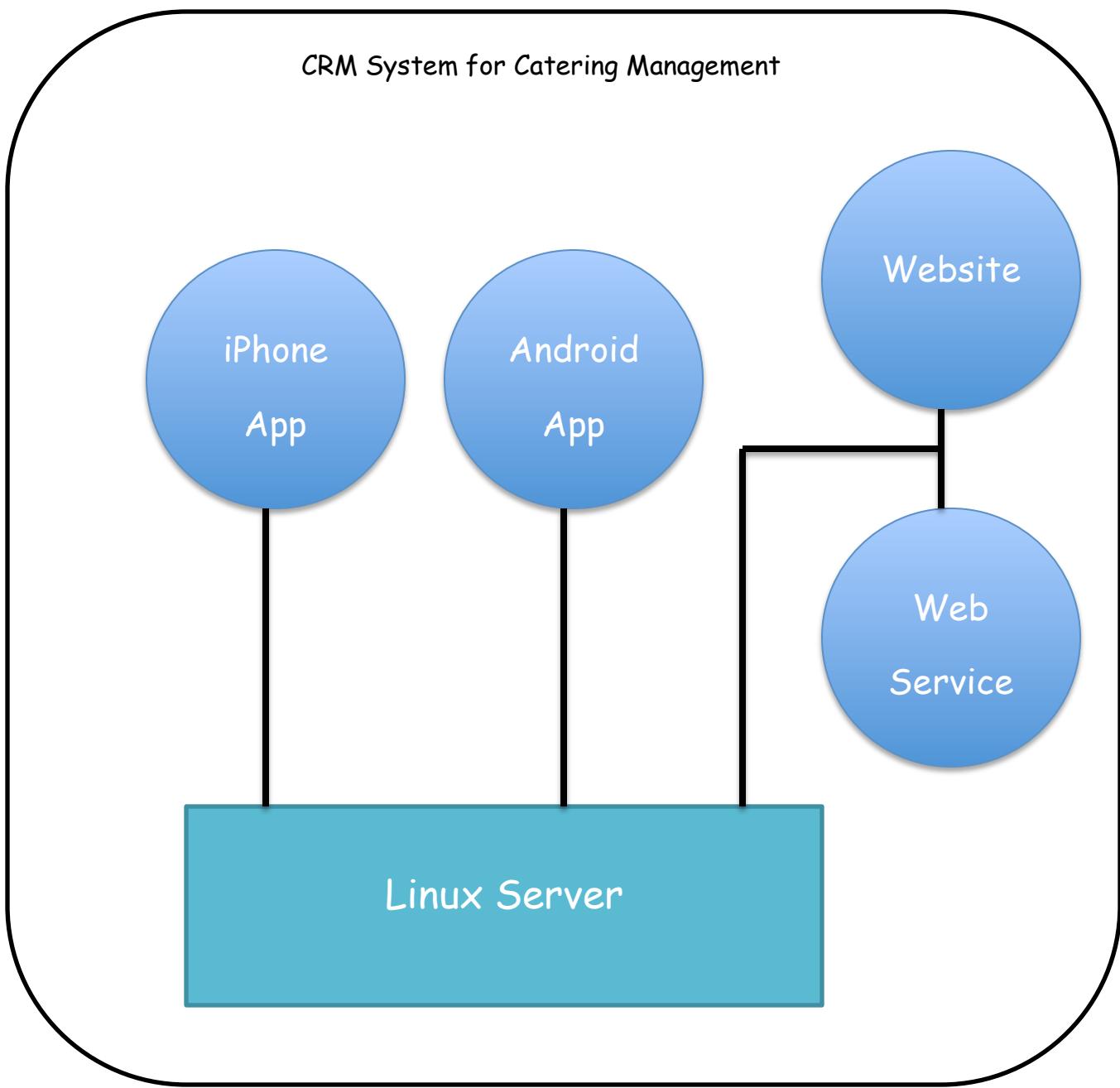
Problem Environment

The company is lack of using information technology (IT) to help improving their services, sales, relationship between customers and the company. For catering market in Hong Kong, many of them do not have any CRM System. Catering in nowadays Hong Kong has a very important status and large market. By adapting CRM System, the company can improve their relationship with customer more. In a long run, they can improve their profit and attract more people to come and pay for your service.

Their original official website for customer to find out the information got a 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 catering company and the customer.

Project Basic Outline & Job Division

Basic Outline



Job Outline Division

Members	Job in this group project
CHAN Chun Wa	<ul style="list-style-type: none">• Android Application development• Android Application maintenance• Data Synchronization between Server and Android• Co-operation with other members' job
CHU Kai Hang	<ul style="list-style-type: none">• Website development• Web service development• Web maintenance• Database Management• Co-operation with other members' job
KWOK Chun Hin	<ul style="list-style-type: none">• iPhone Application development• iPhone Application maintenance• Data Synchronization between Server and iPhone• Co-operation with other members' job
TSANG Tsz Lok	<ul style="list-style-type: none">• Website development• Web service development• Linux Server build up and maintenance• Database Management• Co-operation with other members' job

Requirements with Proposed System

Scope of the System

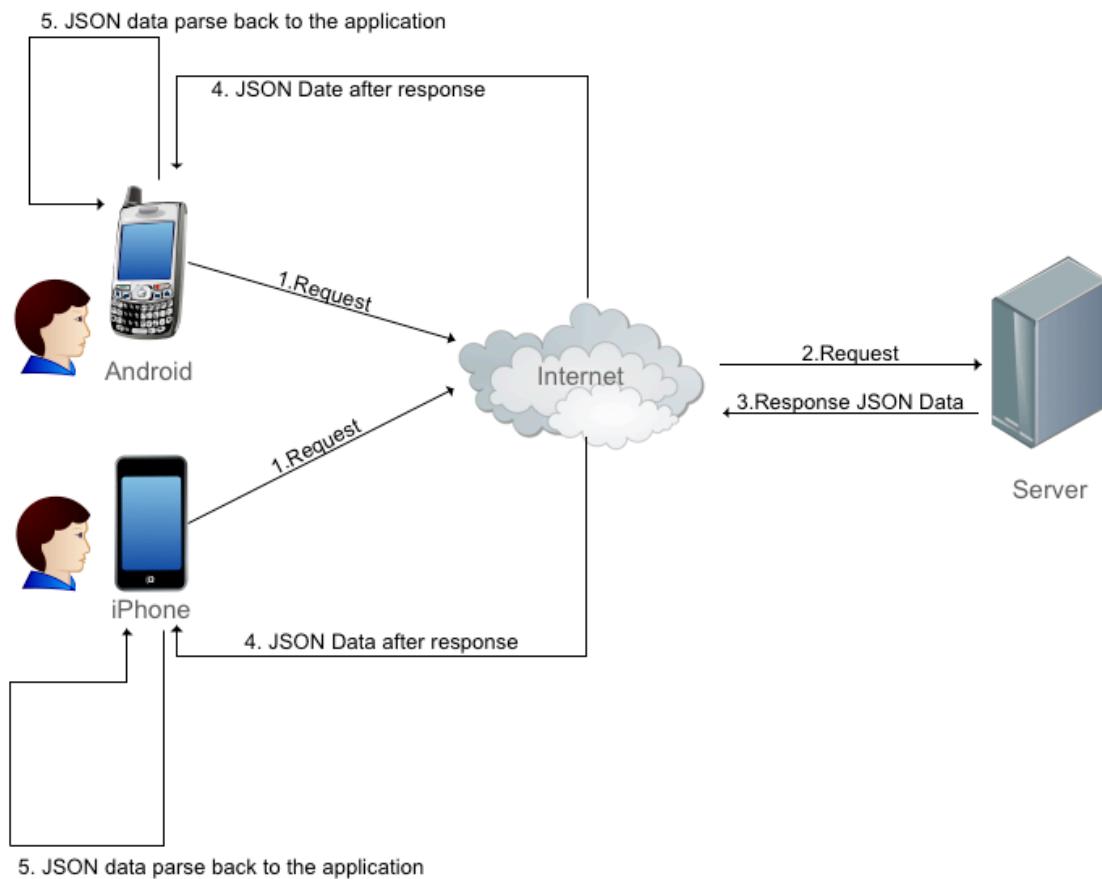
This section is the discussion of system proposed to handle the problems encountered based on the design of the system, functionality and feasibility and the system requirements. Different aspects of problems have been concerned in terms of software and hardware layer. The discussed issues are as following:

- 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 and web service
- 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

- Description for the Server, Database set up

The way of client to archive server's address

We must assume that the Client use smart devices to use the mobile app that we provided. Smart devices are including iPhone and Android mobile. When the client installs the mobile app and starts it in the first time, the app will automatically connect to our server at "http://kelvinlok.homeftp.net/fyp" and receive all the company information to the mobile app by using **AJAX** and **JSON** technology.



AJAX is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously in the background without interfering with the display and behavior of the existing page. Data is usually retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not needed JSON is often used instead, and the requests do not need to be asynchronous. AJAX is not a single technology, but a group of technologies. HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display, and to allow the user to interact with the information presented. JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

JSON (JavaScript Object Notation), is a lightweight text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for most languages. The JSON format was originally specified by Douglas Crockford, and is described in RFC 4627. The official Internet media type for JSON is application/json. The JSON filename extension is .json.

After the data received, the data will store in the mobile local database using SQL Lite technology. 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, location, coupon...etc. Once the client

installs our mobile app, it means the user agree to provide some fair user information securely and also to turn on the push notification service. The data that we collect will not excess the uses of our project development in data analysis.

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



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(UnknownHostException 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.

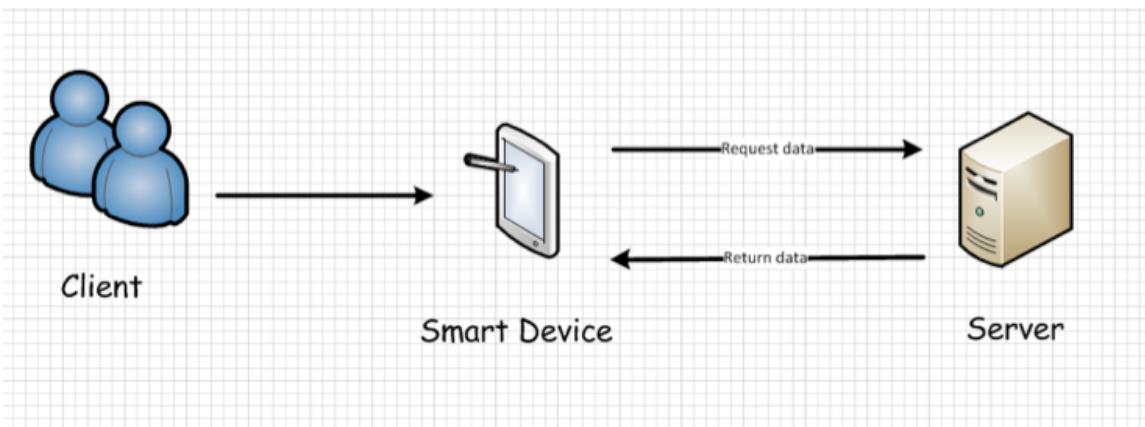
```
+ (BOOL) canConnectToServer;
+ (void) errorWithNetworkConnection;
```

This two are the header of iOS Objective C code for checking the network connection in iPhone.

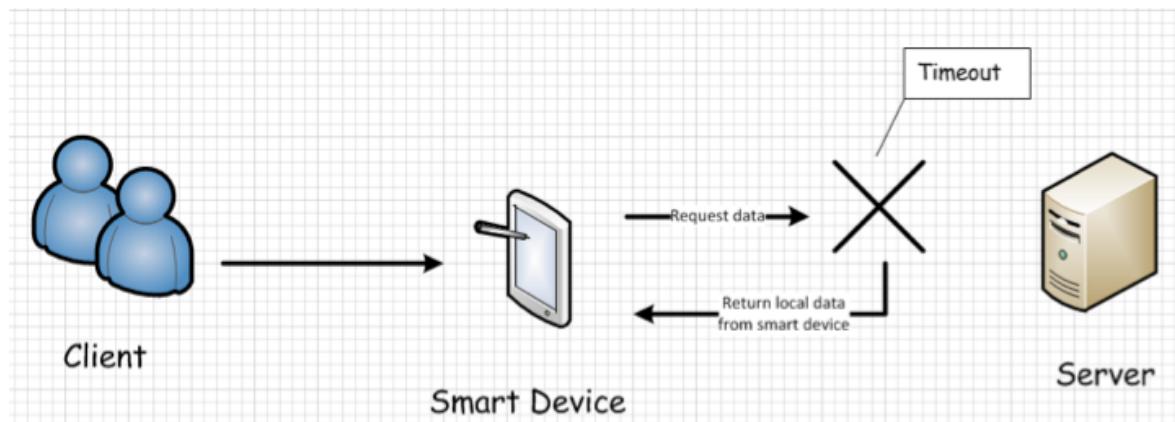
```
+ (BOOL) canConnectToServer {
    NSString * urlString = [config.getServerURL];
    NSLog(@"canConnectToServer Testing %@", urlString);
    NSString * escaped urlString = [urlString stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
    NSString * responseString;
    NSURLResponse * response;
    NSError * error;

    NSURLRequest * request = [[NSMutableURLRequest alloc]
        initWithURL:[NSURL URLWithString:escaped urlString]
        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;
    }
}
```



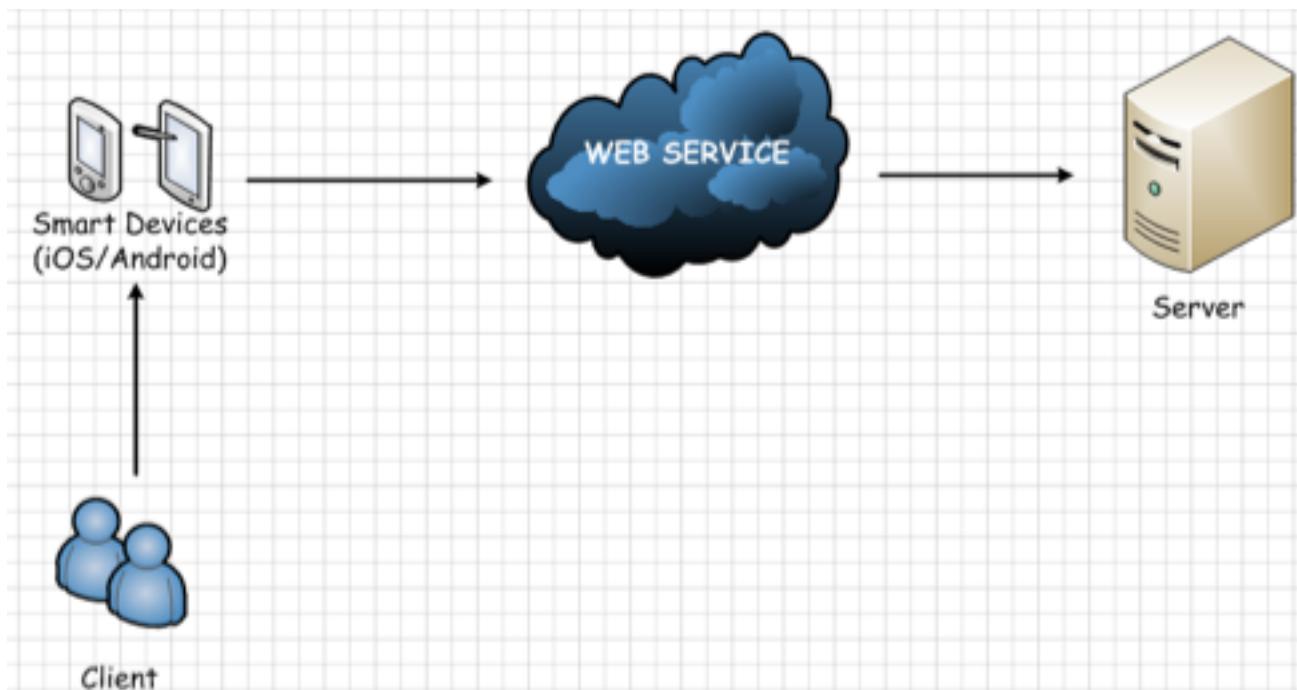
The above diagram shows the Client open the application and then successfully connects to the server to receive the latest information.



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, that means they use our app in Wi-Fi or 3G network. The company staff can use the provided website to update and insert their latest news, information, data into the database. The Client must

have 3G or Wi-Fi network provider to keep their data updated. When the user uses 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 should be in different layers. The first view show the group of the company, it display a list of the branches and shops with different type of restaurant that belongs to the company with their logo and name. After the client choose the branch or shop, it display the functions that we provided for the client that related to the branch or shop, the functions will be discussed later.



The functions of the mobile application

- Restaurant News

The company can update their restaurant information, 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. They can update the price, food details, food type, and images of the food.

- 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 date 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.

We make reference to another application in App store, and we would like to follow them to design the new one.



The E-coupon we provided 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.

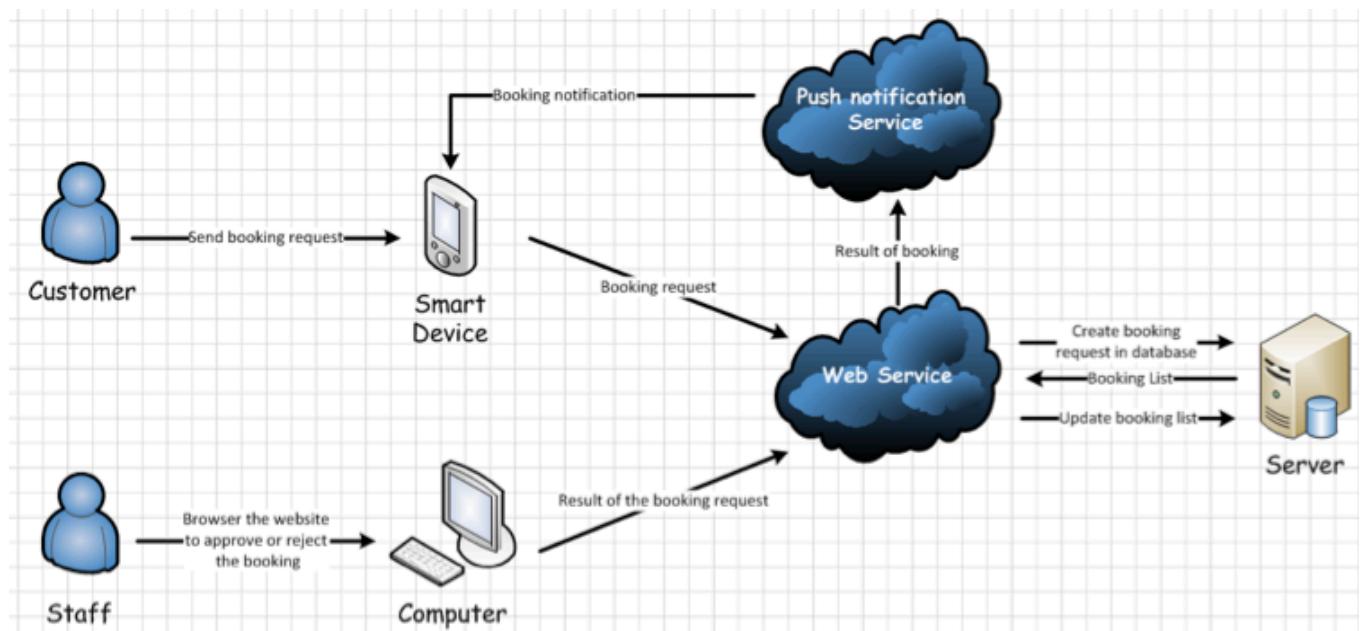
- View the Queuing States

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.



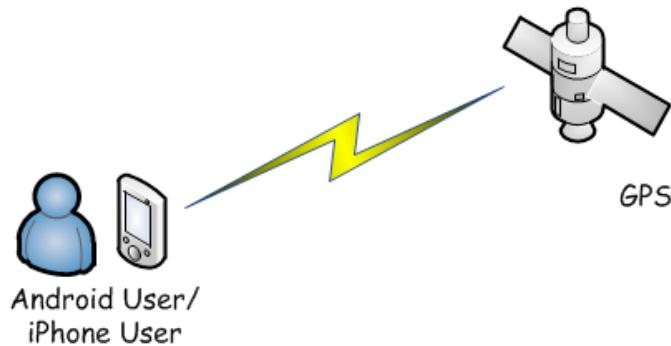
The staff needs to confirm the booking manually because the shop does not have any booking system, and the staff are receiving the booking phone call and note it

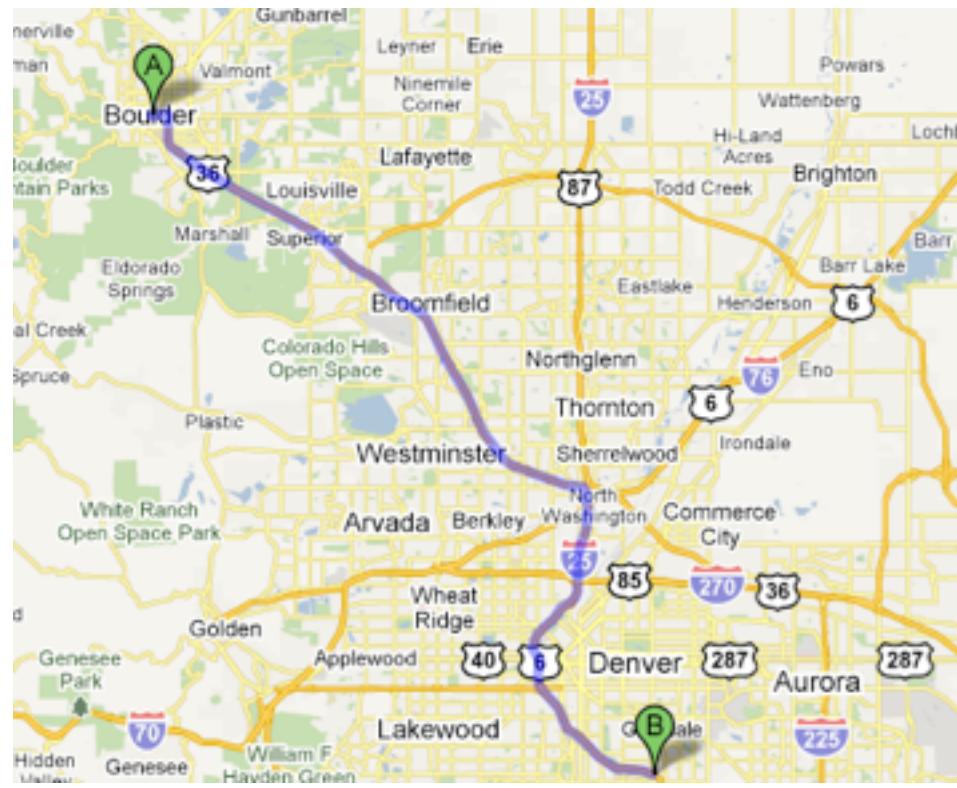
down only. So, when the staff receive the booking on the website, they will check call the customer to confirm and note it down, after that update the booking record in the database.

A cancel booking function also be provided to the customer if they would like to cancel the booking on before or after the confirmation. This function is use to provide user-friendly system.

- 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

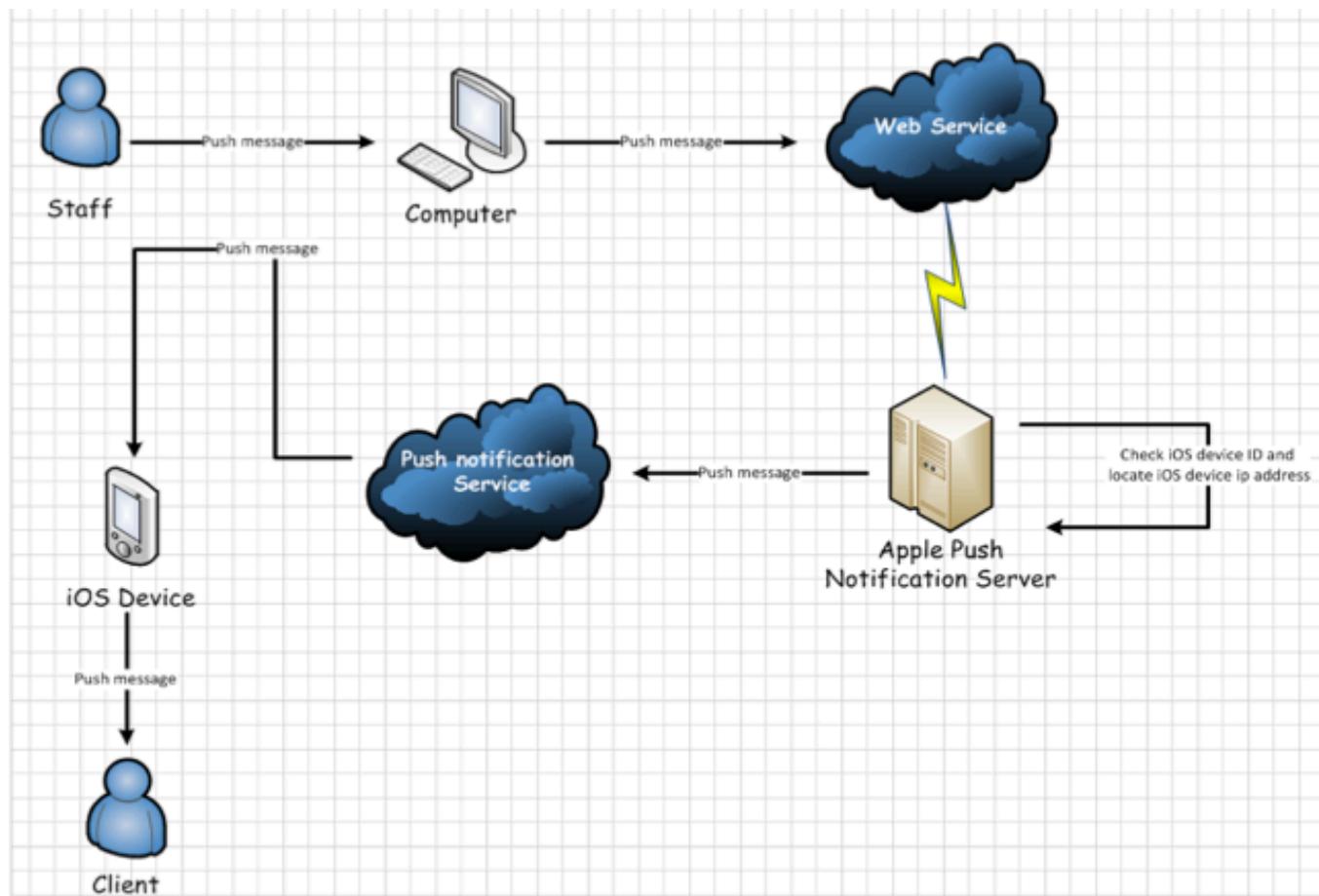
The client can use their mobile phone to send feedback to the server. The database will record the received data for further data mining process. The company can create different feedback forms on the website, the forms will automatically loaded to the mobile app when the client start our app.

- Push notification service

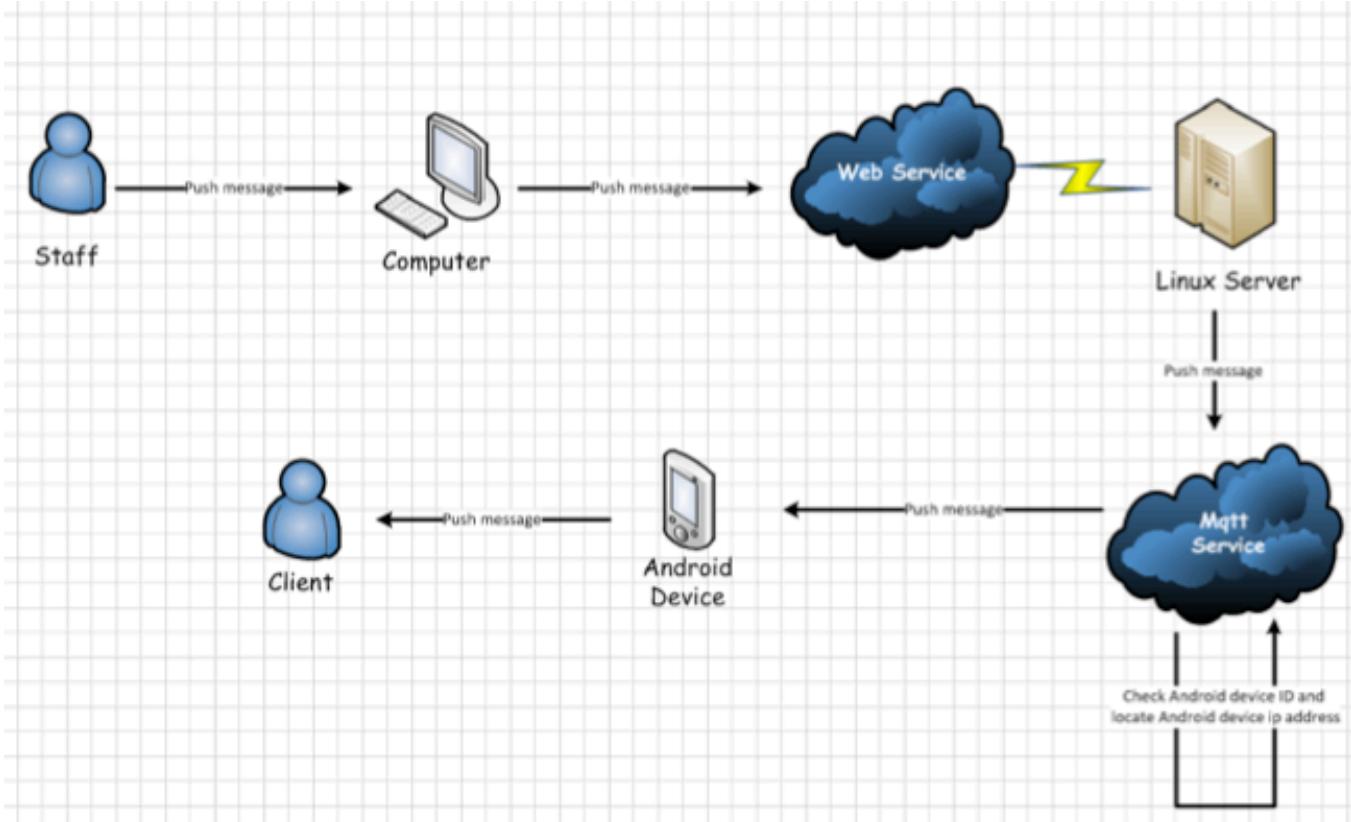
The method for doing the push notification is different in iOS and Android. Apple Inc. has provided a push server for developer to send the device id and message to

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.

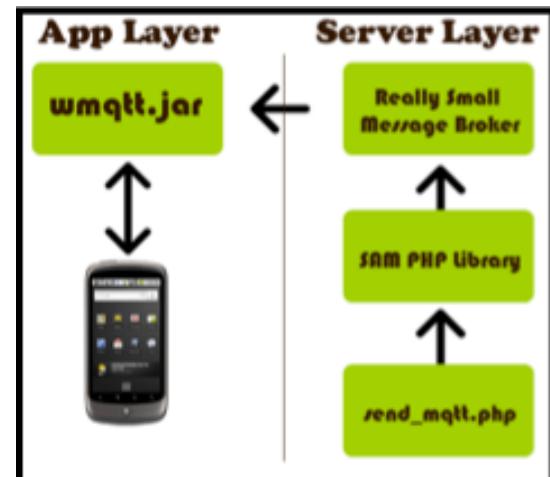
To send push notification message to Android devices are similar, but we need to build up a Linux server and install MQTT service to use the push notification. All the things need to be created and operate by the developers.



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

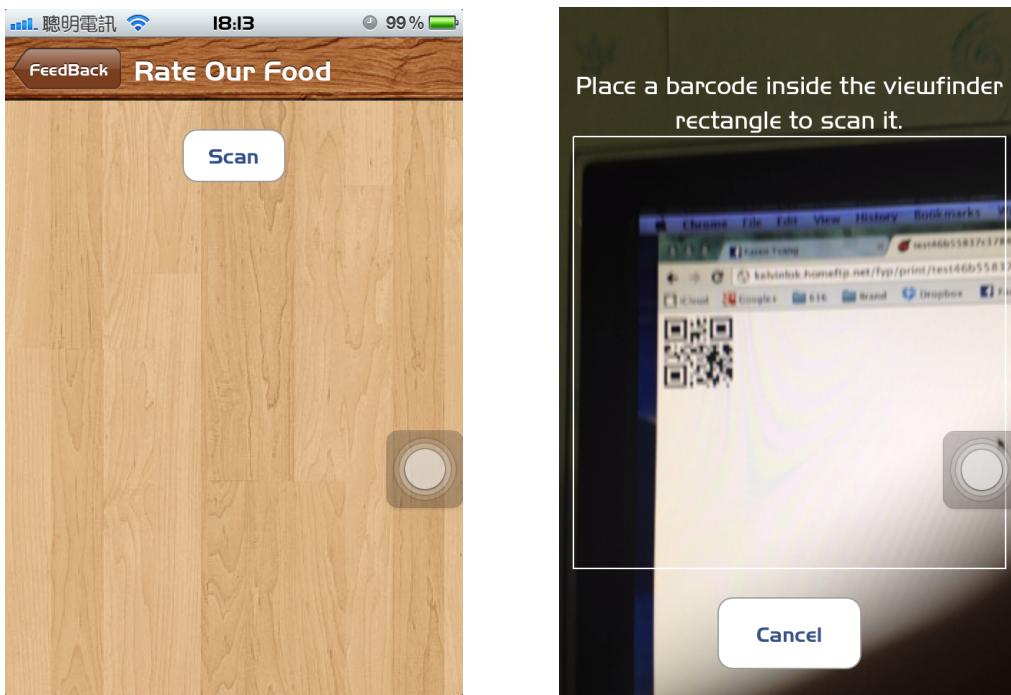


The above diagram shows the process of sending the push notification to Android devices.



- QR-Code Feedback function

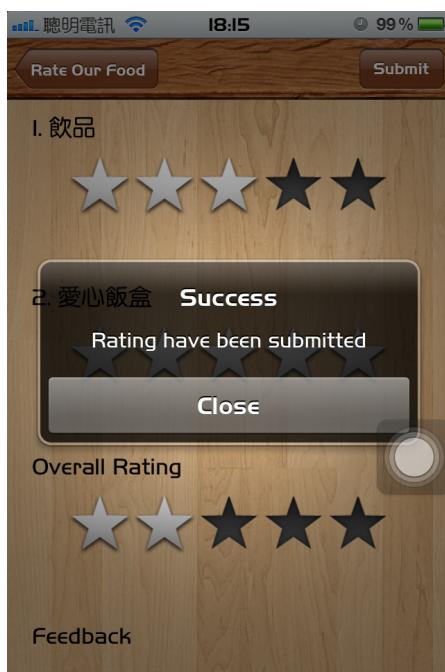
The mobile application will provide a QR-Code scanning function, once the customer purchases their food or service, a short receipt will be put on their table in the shop. The short receipt only records what food they have ordered and provide the QR-Code on the receipt, the QR-Code contains the information of the food that they ordered. The customer can use the scanning function in the app to scan the QR-Code and then rate the food instant. The following diagram shows you the processes.



First, go to the Feedback button and then press the button scan, place the photo camera under the QR-Code. After the app scans the QR code, it will direct the user to the page to rate for her ordered food.



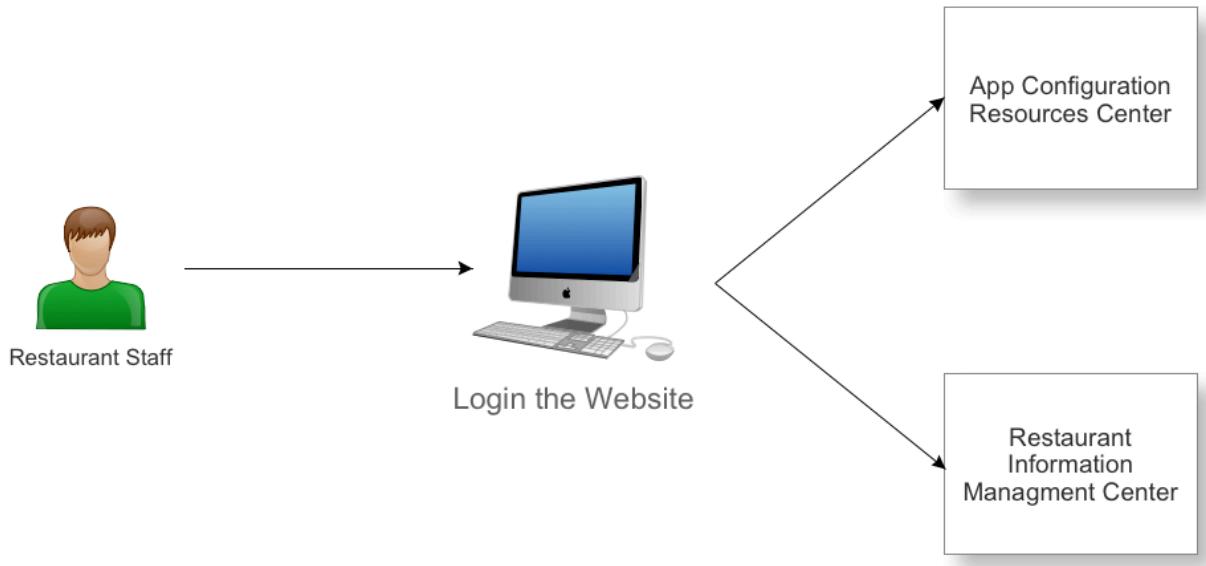
The user can touch the star to rate for the food.



Then, submit the rating.

The functions of the website and web service

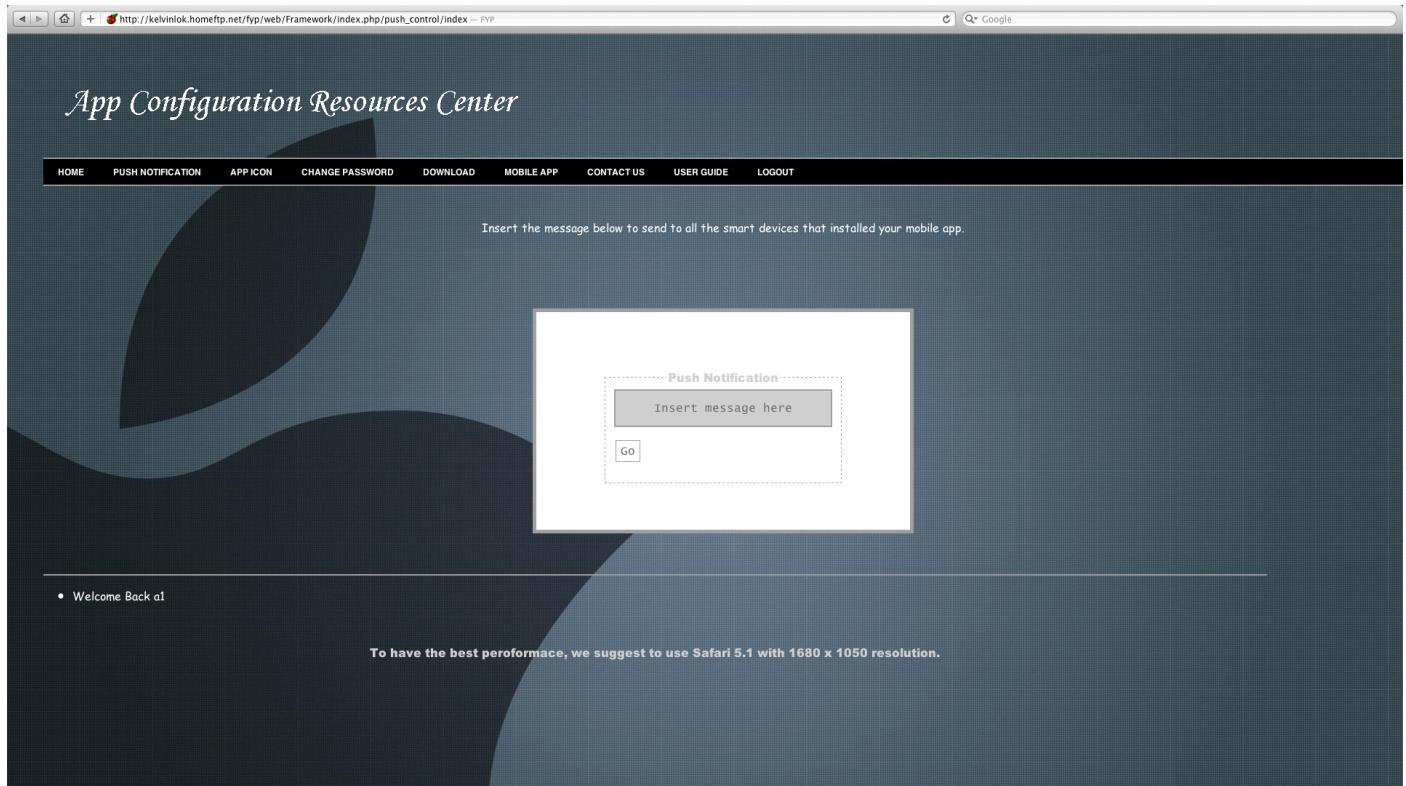
We divided the website into two parts: App Generation Resources Center and Restaurant Information Management Center.



The App Configuration Resources Centre and Restaurant Information Management Center require the staff to login, the password is stored in MD5 on the database for security issue. 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 Configuration Resources Center is used for the staff to modify their own application for their restaurant, change the image, background, themes or functions in the application and produce a new mobile application.

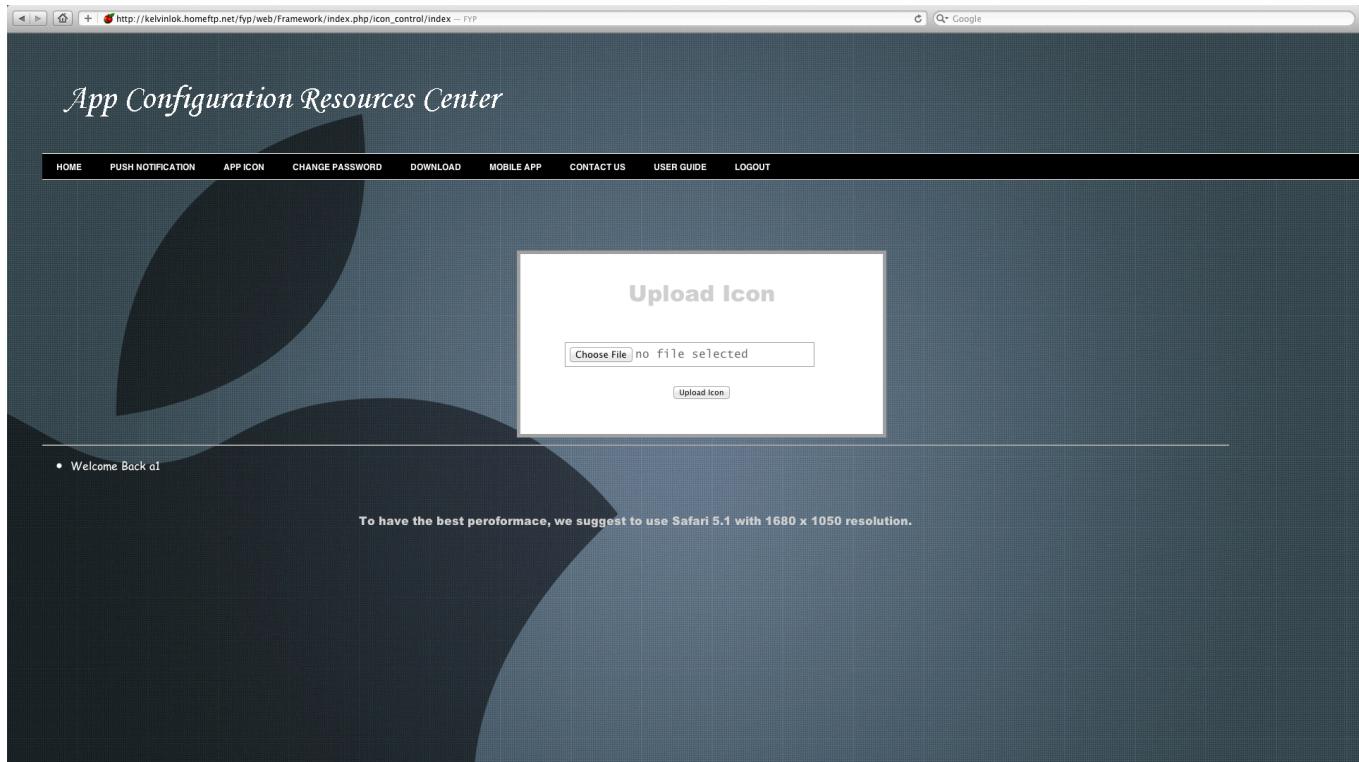
- Push notification message

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.



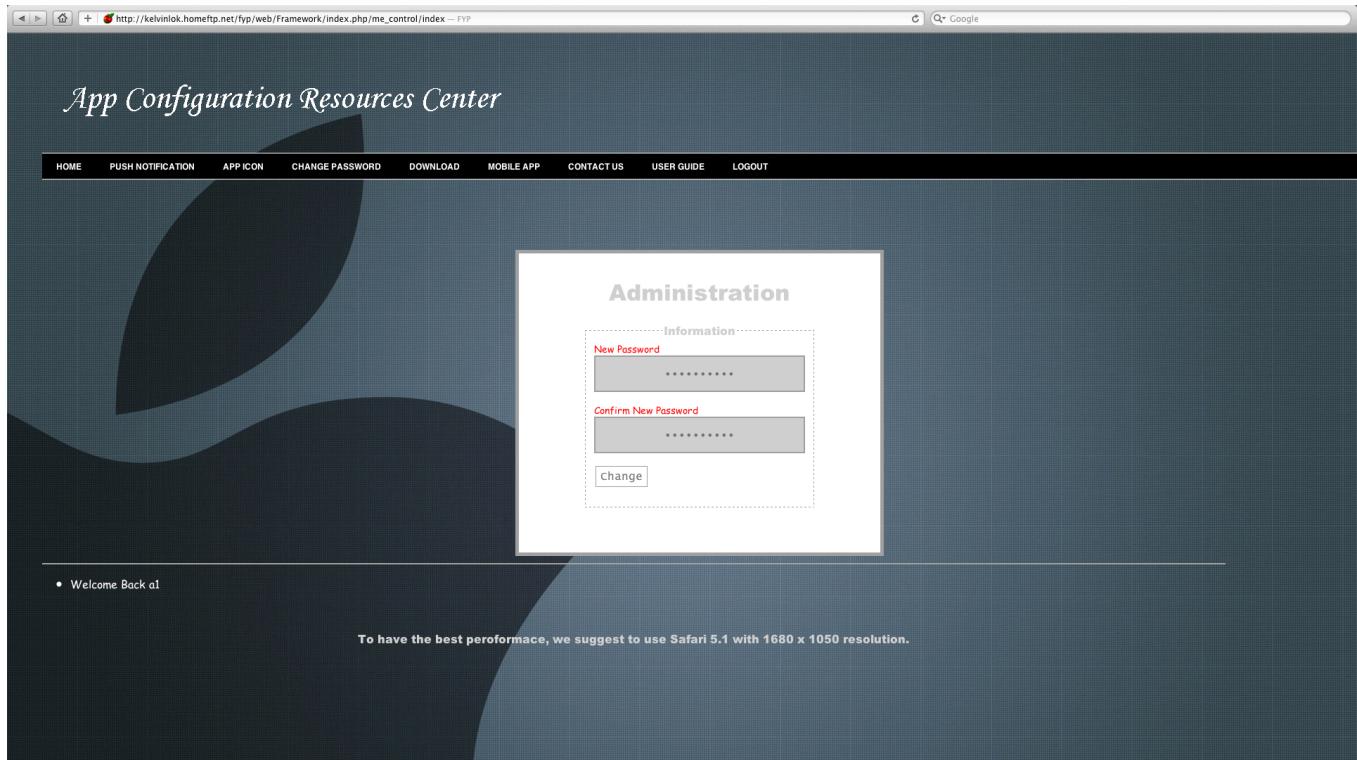
- Upload and manage app icon image

The staff can upload a 64×64 pixels image to the website. The image can later be used in the app generation process. The webpage will show 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 after all the development 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.

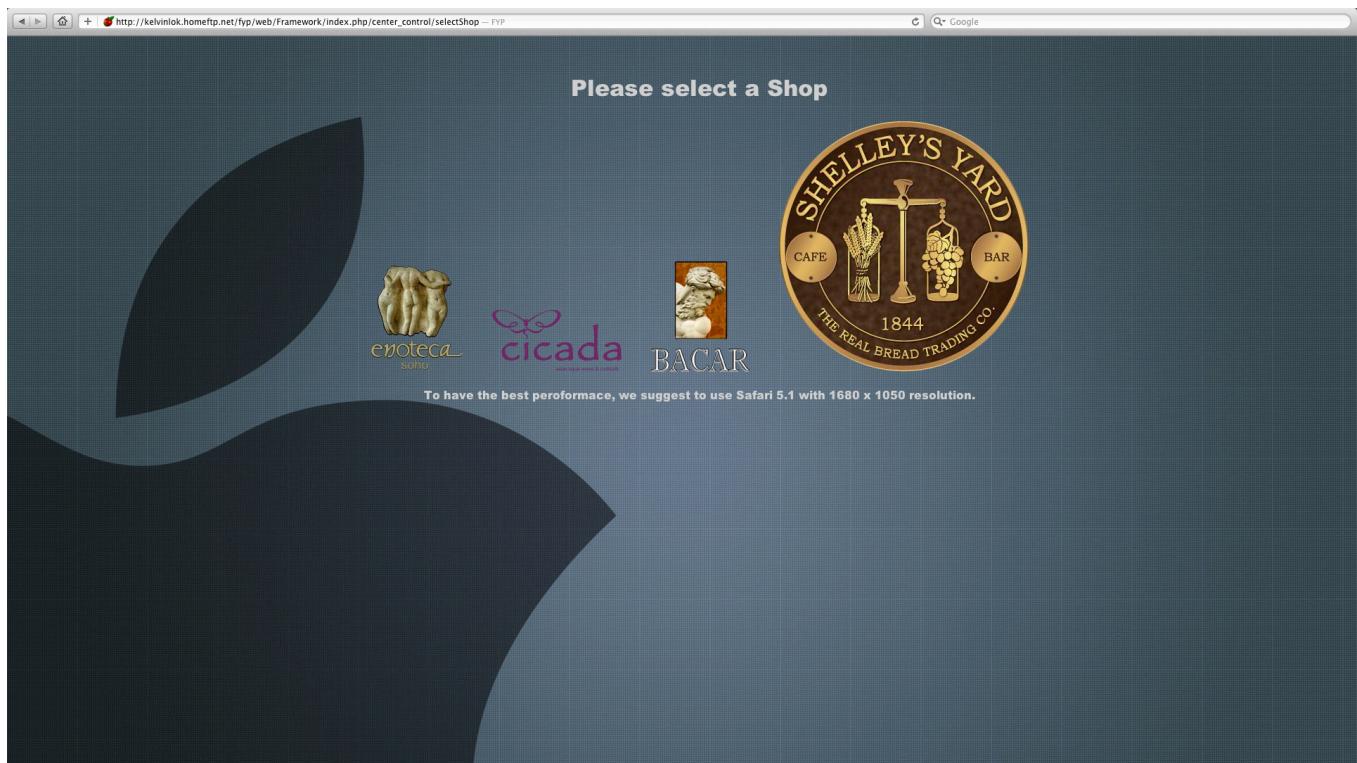
The user can then modify their mobile app, choose the functions in each type of restaurant for the app for the customers, check the theme in different type of restaurant, change the layout in different order of function, change the food menu table view layout.

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.

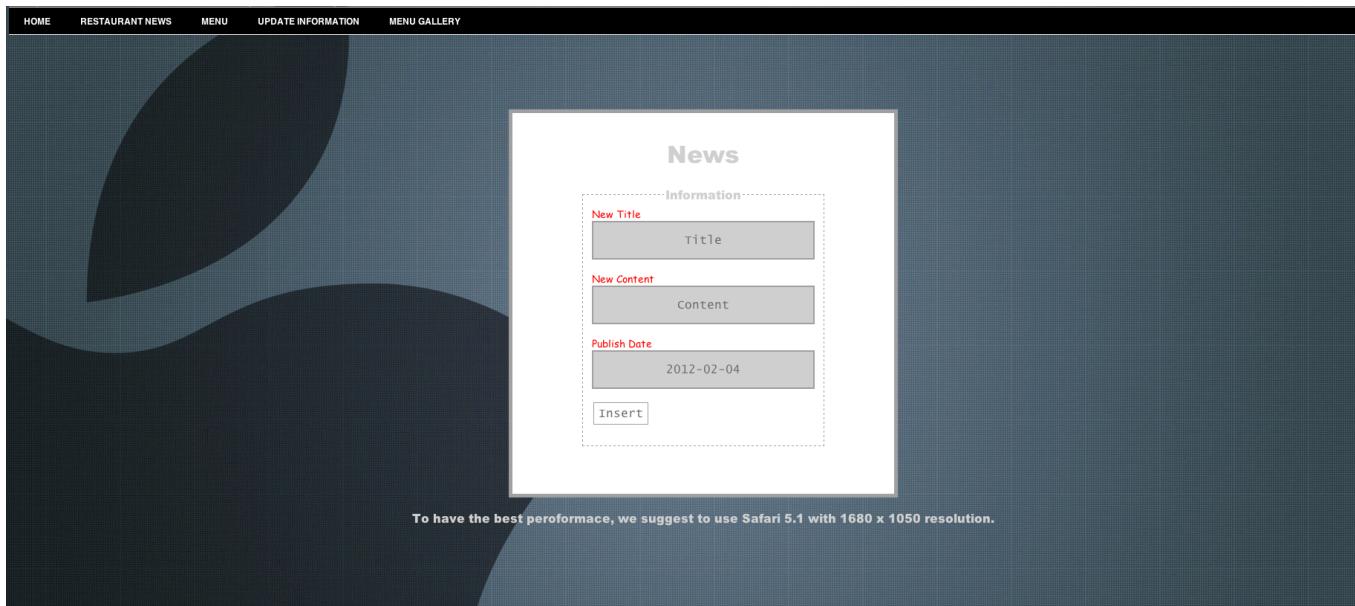
- Selection of the type of restaurant in the website

There are 4 types of restaurant in the website, Bur (Pub), Italian Restaurant, Café, Café and Bar. Just click on the type icon to enter the page to make modification.



- 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.

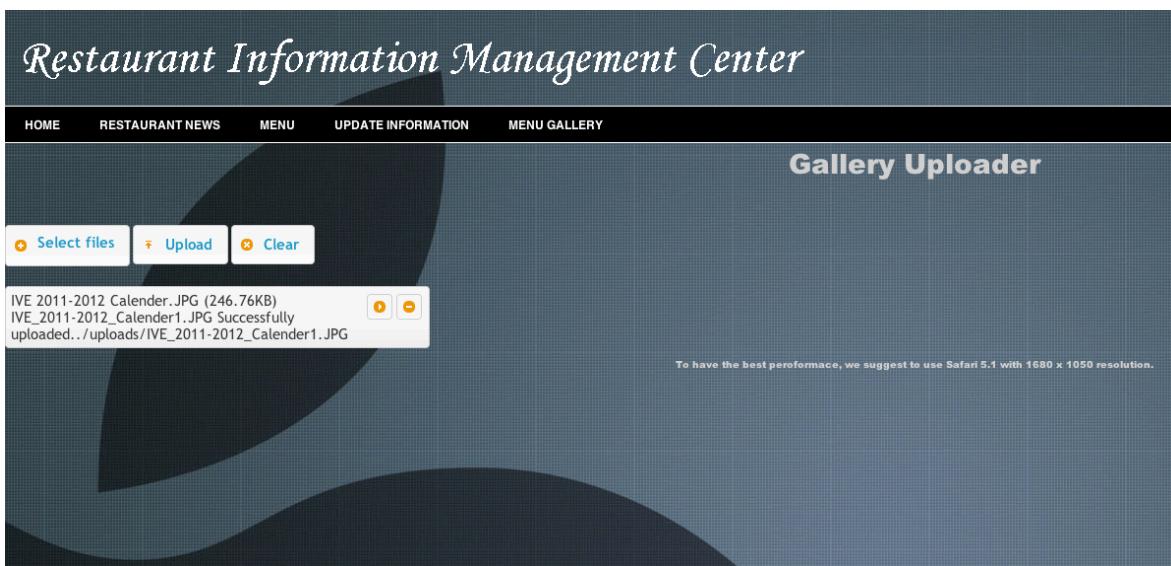
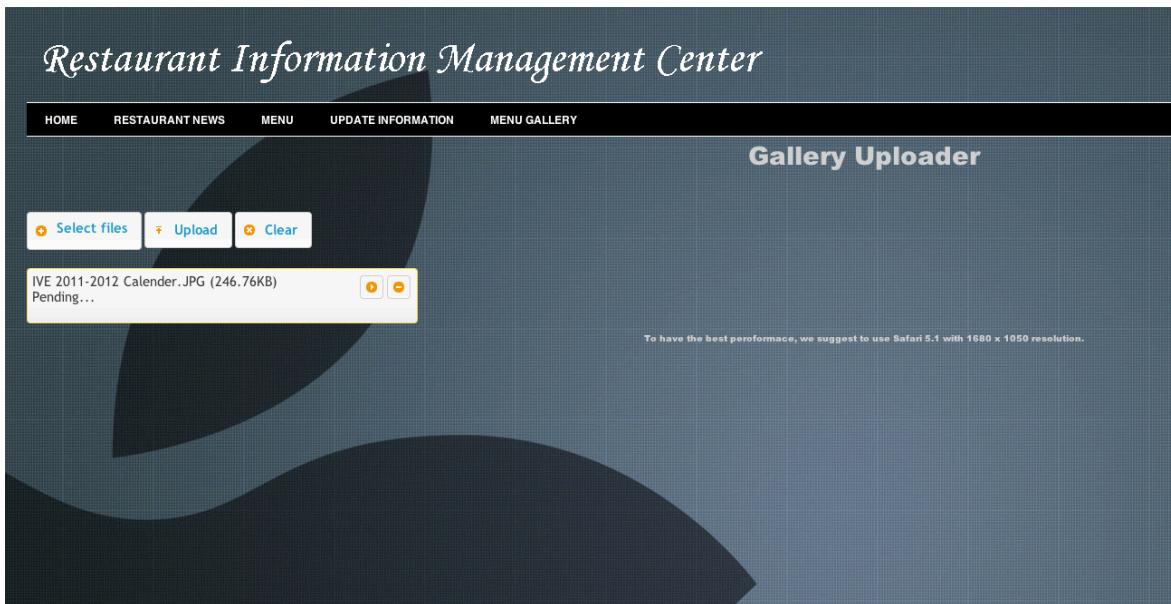


- Update electrical Menu

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.

- Upload new photo for the food

The staff can upload new photos for the food menu in the mobile



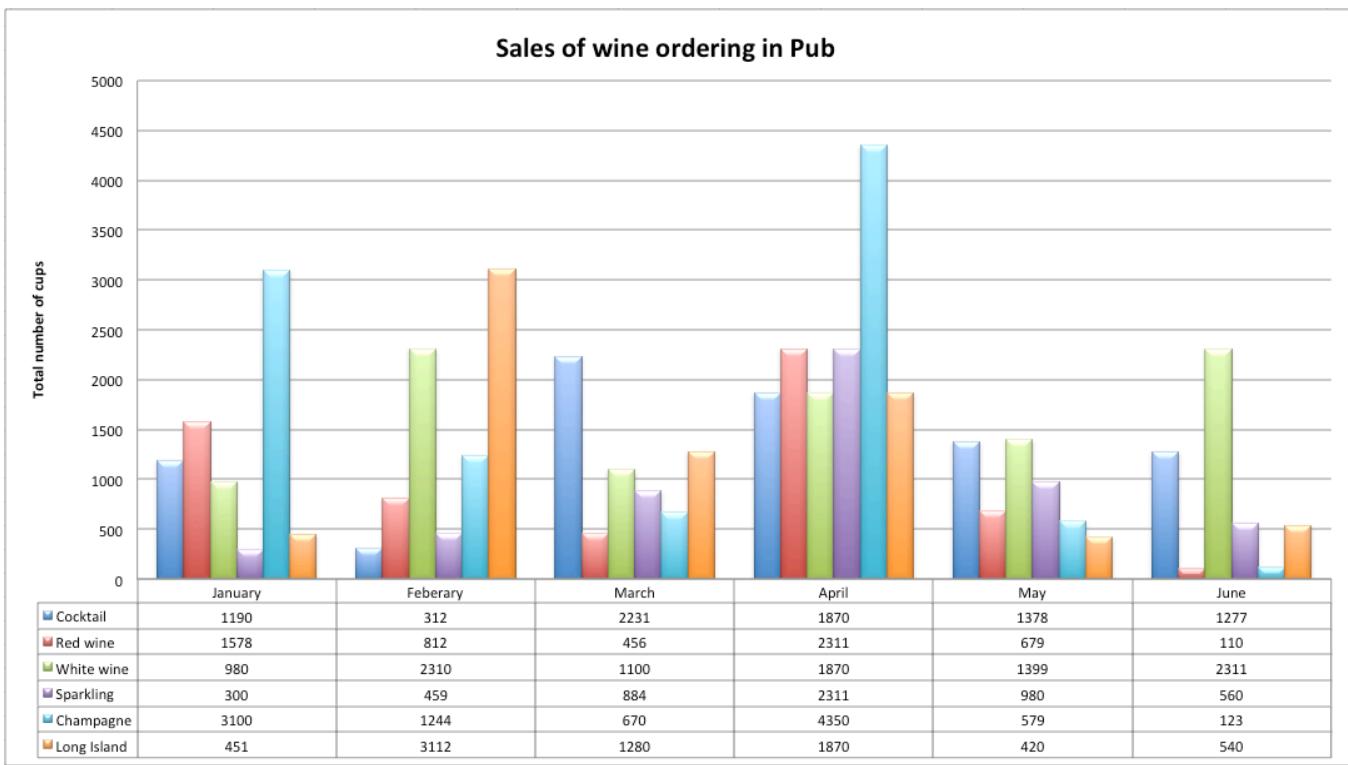
- Staff confirms booking

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 the company to find out the habits of their customers so that you can design the corresponding business strategies. We will find some of the data mining tools from the Internet and choose the best one to develop it. We will use the feedback data from the customers to analysis the company future business target. For example, we can find out which food the customers' most favorite, what the customer needs or wants but the company do not provide...etc. Here are some of the examples of data mining.

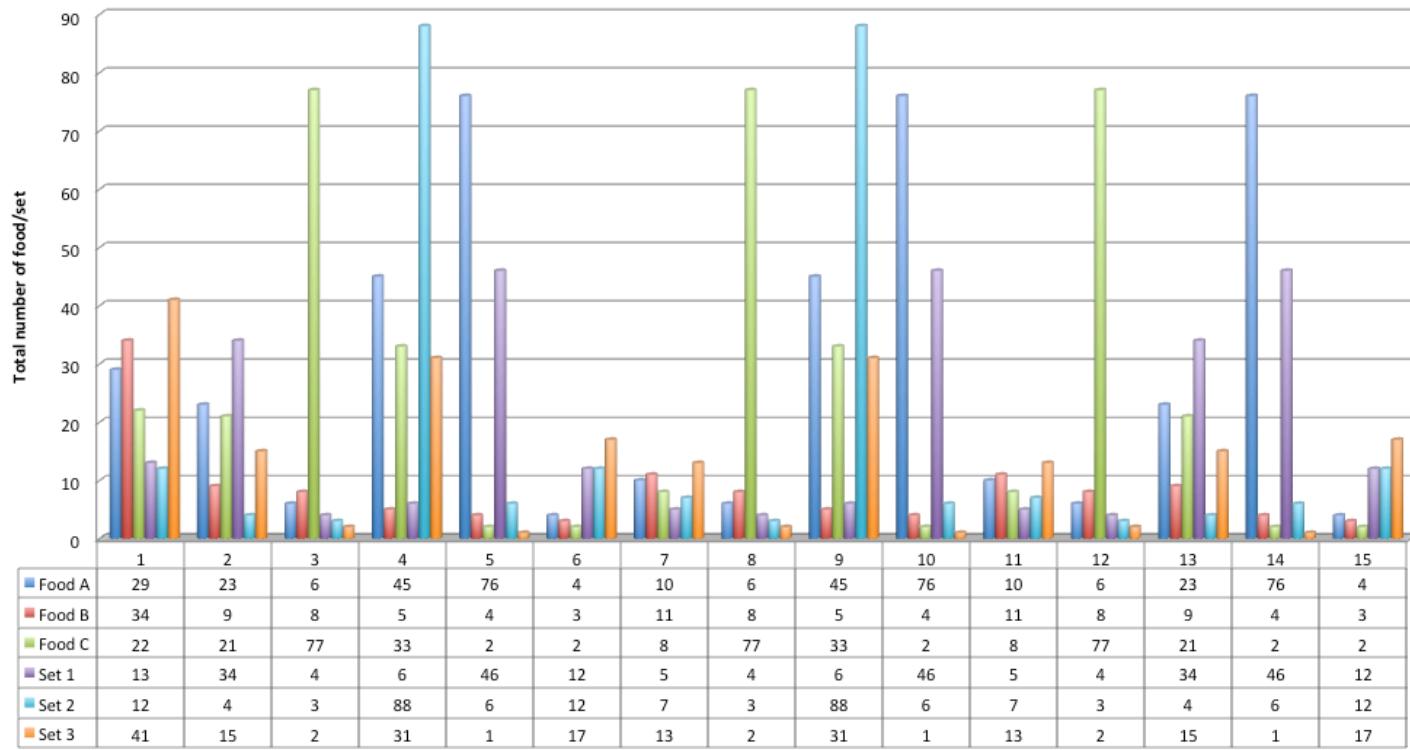
We will provide some selection type for the company to choose to analysis what data in our database. The critical can be selected by the restaurant staff.



The diagram shows the cup of the sales of customer ordering in Pub in different months in a year, the company can find out which kind of wine the customer like most and purchase more in the stock to keep their service and which kind of wine that is not much customer like, so they can purchase fewer in the stock. Also, you can find out which month there are most customer in the branch. From the diagram, you can see April is the month contains the most customers in the first half year.

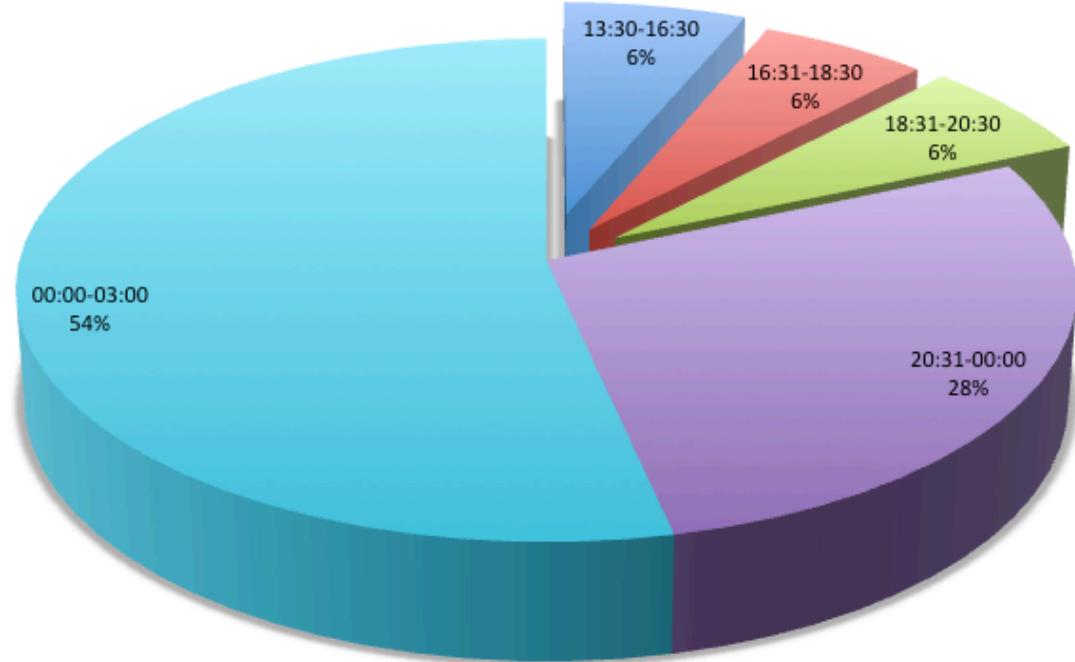
Chart Area

Food ordered in January 2011



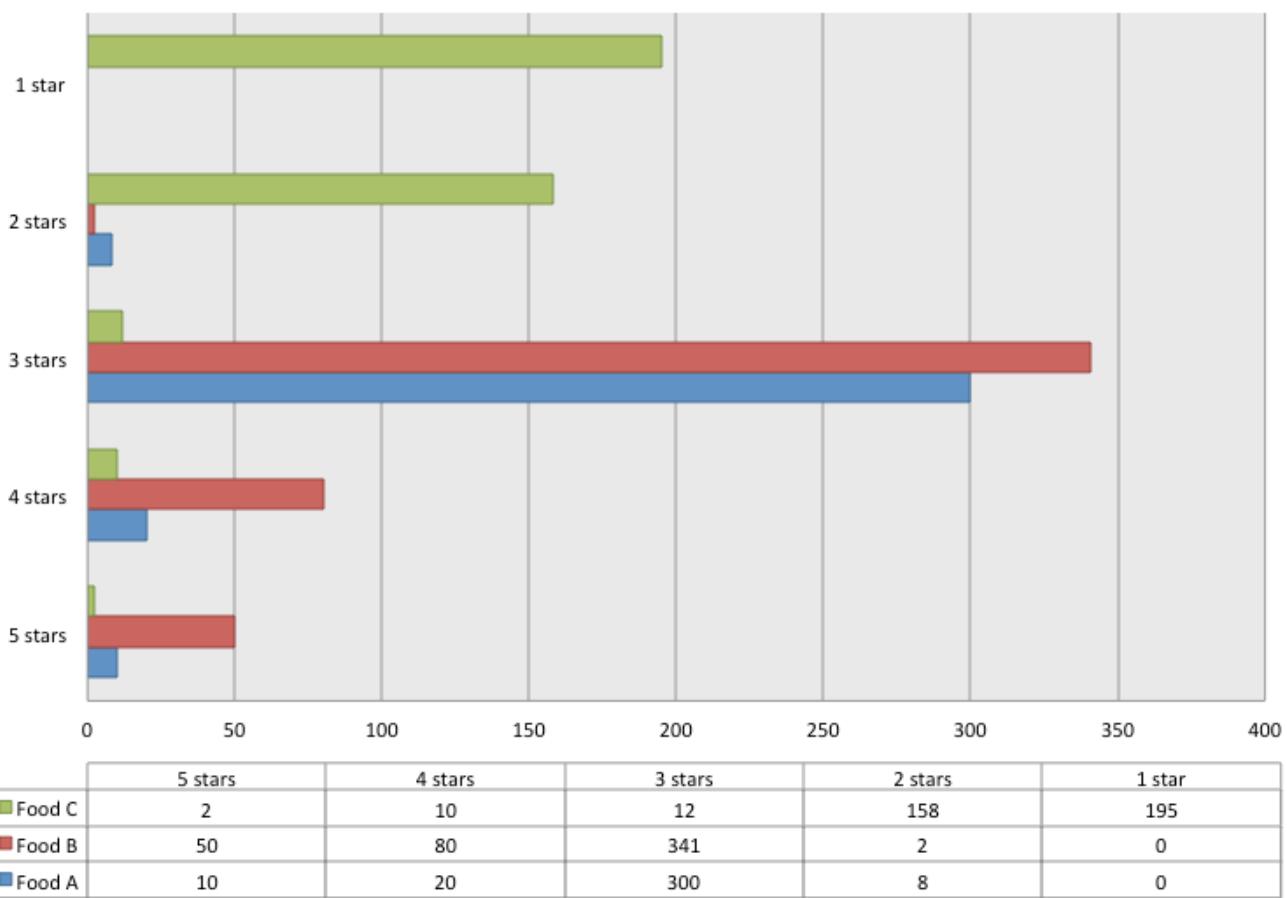
The diagram shows the food of the sales of customer ordering in one of their Italian Restaurant in different days in January, the company can find out which kind of food the customer like most and which type of food the customer do not like, so they can change the food or design a new food menu to seek for the customer needs.

Total precentage of customer order service in different time slot



The diagram shows the time slot of the sales of customer in one of their Cafe in a month. The company can make use of this statistic to manage not only to food but also the human resource, they can save energy and resource in the timeslot that is not much customer and spend more resource in the timeslot that is much customers.

Rating of the food in 1 February 2012



This diagram shows the rating from the customer feedback in the mobile application. The company can view the food-rating list in order to further develop their business path.

- 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.

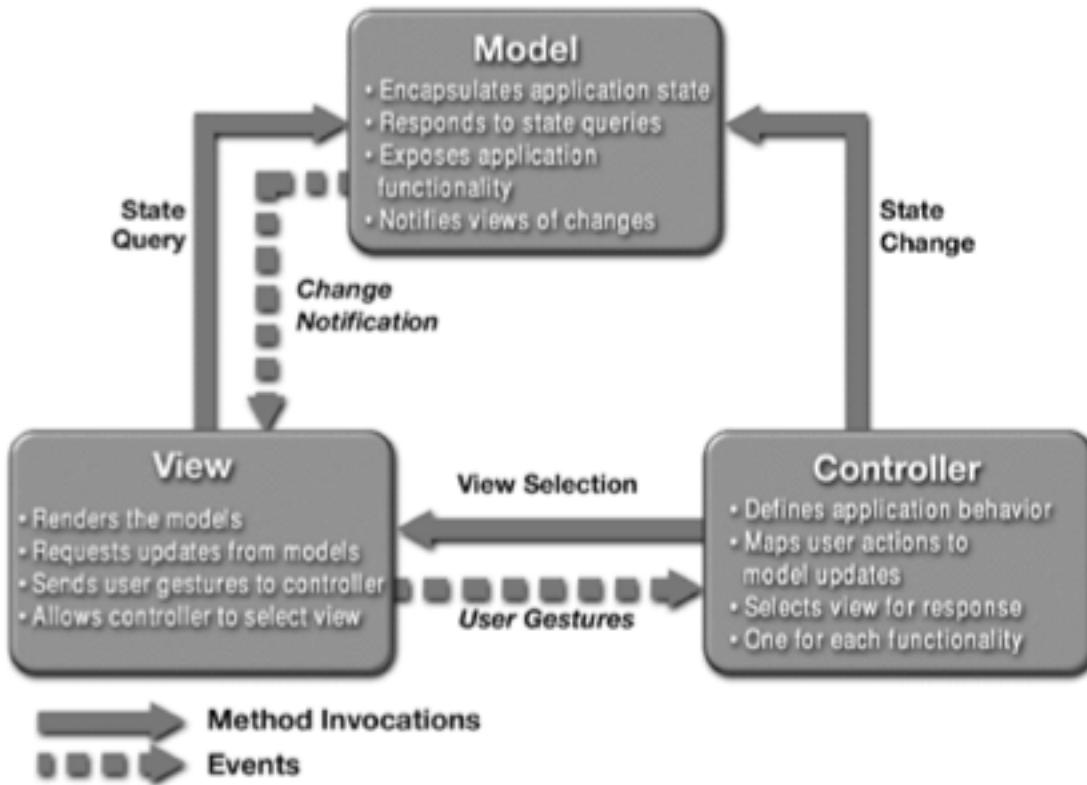
To encourage more people to install our mobile app, we develop and test it in many ways to make the things and jobs simple. And the language we provided for the user is use simple English.

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.

Brief description of server and database set-up

We need to build up a Linux Server in one of our members' home and connect with network. We choose to use a old version computer than can install drivers for Linux OS. Linux Cent OS 5.6 is the most suitable version for us.

First, install the Cent OS 5.6 into the computer and use the terminal to install the drivers for it including network and graphic drivers. After that, we install the service on the computer. Here are the services in Linux we need:

1. SSH Terminal
2. VSFTP Service

3. HTTP Service
4. PhpMyAdmin
5. MQTT Server

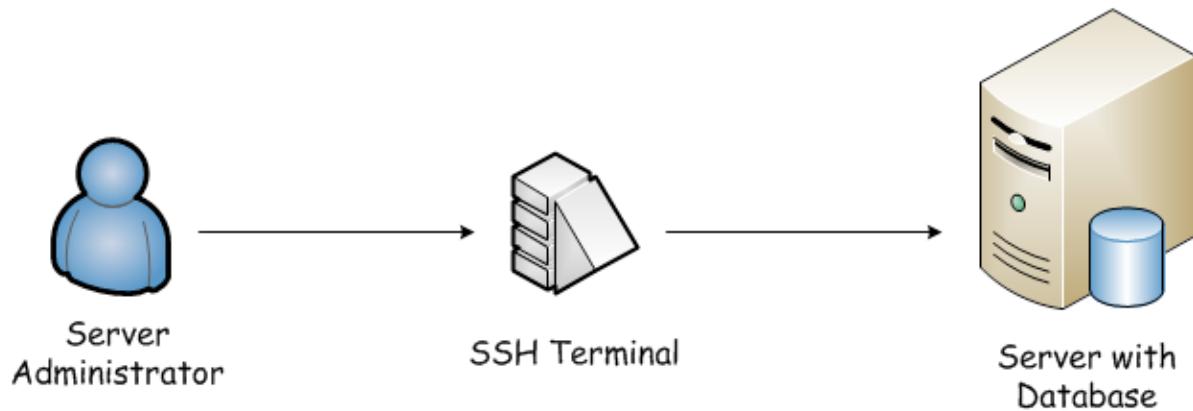
Then, we need to make configuration to different service. The VSFTP Service needed account and security for us to put all the resources in it. We need to change to permission to let us use SSH to login the server and maintain.

After the server set-up, we register a "dyndns.org" free domain name that is "kelvinlok.homeftp.net". We set up a router that is support Domain Name Transfer from "dyndns.org", therefore, the IP address will be transfer to kelvinlok.homeftp.net dynamically and we set the update time is one hours, once the IP address from the ISP is change, it will later be updated by the router automatically.

Non-Functional Requirements

A server administrator must be added to the system

In order to recover the server service when there server down time, we should find one member to maintain it and repair it as quick as possible. For example, she needs to restart the service by terminal or through SSH Terminal.



User-friendly GUI for the customer

We should provide a user friendly GUI for the customers by adopting the concept of human communication interaction (HCI), we found some of the mobile application in the Apple iPhone that we will mark reference on their design.



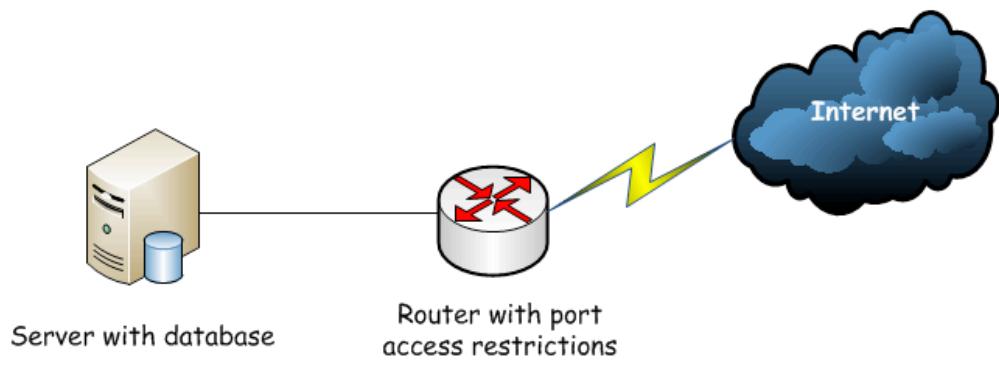
Log keeping in database and server

We should keep the log file in the server when there is any error occurred or any transition is crushed. We can trace back and rollback the transition or we can easily locate the bugs in the server. The following image shows the error log of the application in database.

+ Options	id content
Edit Inline Edit Copy Delete	1 {"booking_time": "2011-10-08 13:21:49", "type": 1, "sh...}

Security Design is need in the server side

The server keep our customer data in the database, in order to provide security, a router is set up between the Internet and the Server, the router will control the port access in order to the related server IP address, it only allow the port that is need to use the service only.



Automatically logout when the user stand-by for more than 15 minutes (website)

If the user stay on the website and do not perform any functions for more than 15 minutes, the system should request the user to re login her account and password to validate the user identification to keep the data in secure.

Database should be backup frequently

The database should be backup in the server frequently like once a week. When the database is crush, we can put back more than 60% data back to the database to keep the operation of the whole CRM system.

Use high effectively server

We choose Linux operating system to be our server, it is a free enterprise os and with secure configuration. It provides reliable and high effectively service to user.



Use the stable software tools in the server

Since the technology in nowadays change quickly, some of the software version also be updated like PHP, jQuery, operating system...etc. We need to choose not to updated version but the stable one. Therefore, the Linux Cent OS we choose version 5.6.0 and the website we choose to use PHP 5.5. For mobile, iPhone should be with ios 4.2.1 or above and android device should be in 2.3.1 or above.



Keep Network Stable

We should choose a better service ISP in Hong Kong in order to keep the network operating with the server under a stable environment. For mobile user, we suggest they use a stable 3G network in Hong Kong.

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 objects 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 executable 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 ourselves. 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.

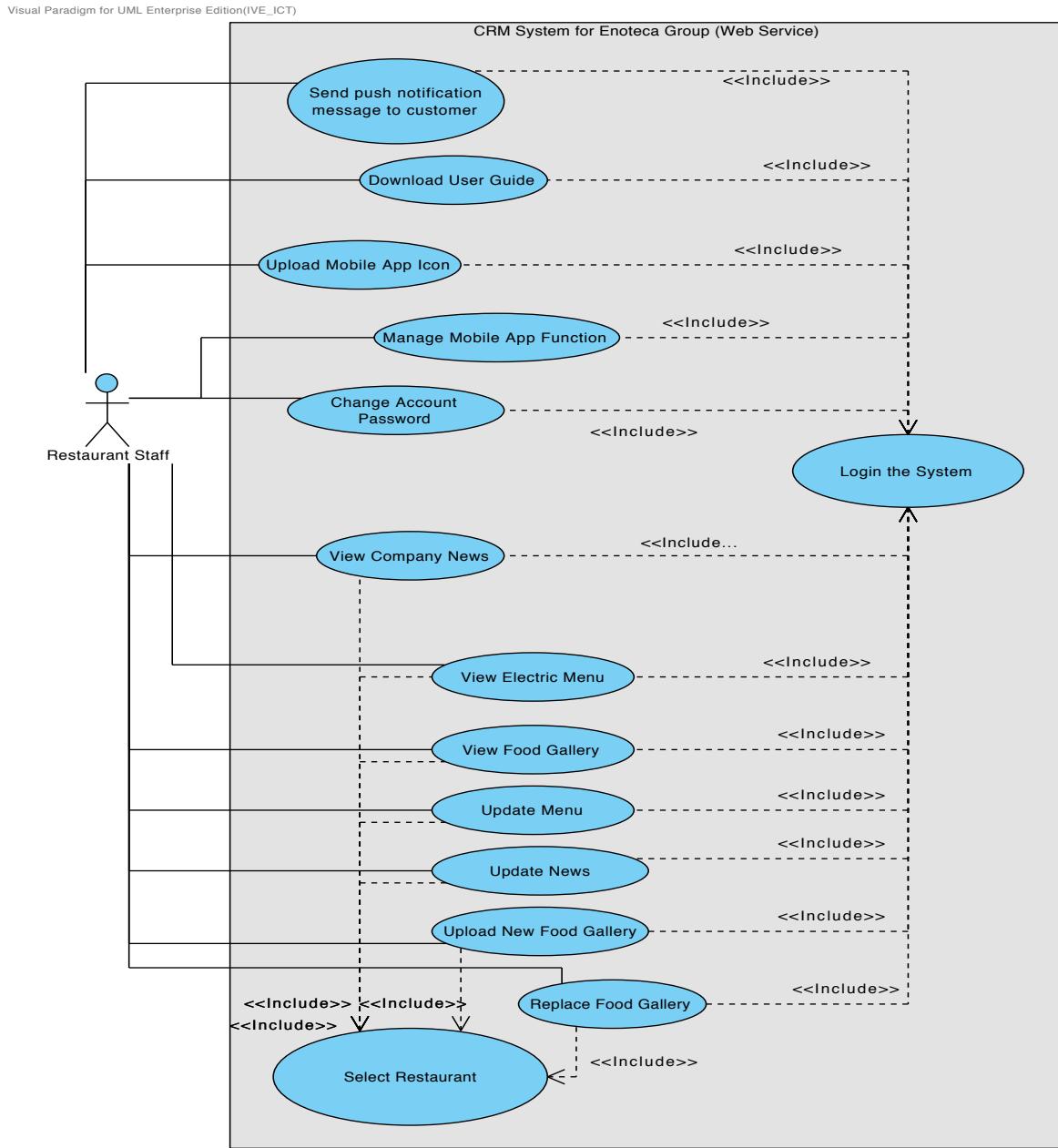
UML MODELING

65

Final Year Project Interim Report (2011/2012)

Use Cases

Use Case Diagram of Web Service



Use Case Description of Web Service

Use case name:	Login the System
Use case ID:	UC001
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff login to the system and choose which sub system she need to use.
Pre-conditions:	Restaurant Staff has a valid user account and password.
Flow of events:	<ol style="list-style-type: none">1. Restaurant Staff input username and password in the index page.2. The system shows the sub system page.3. The restaurant staff chooses the sub system.4. The system directs the related restaurant page to the Restaurant Staff.
Post-conditions:	Login Successfully.
Alternative flows and exceptions:	If the Restaurant Staff does not have a valid account, she cannot login the system.
Non-behavior requirements:	It suggested that the Restaurant Staff use a Internet browser with Safari 5.0 or above in resolution 1680 x 1050.
Assumptions:	Pair of username and password is created.

Use case name:	Send push notification message to customer
Use case ID:	UC002
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff can input a notification message, after press the confirm button, a message is send to the mobile device
Pre-conditions:	
Flow of events:	<ol style="list-style-type: none"> 1. (include UC001) 2. Restaurant Staff chooses the push notification function. 3. The system shows the input page. 4. Restaurant Staff input the content of notification 5. Restaurant Staff presses the send button. 6. The system sends the message to the mobile device. 7. The system returns to the notification page and display a successful message.
Assumptions:	<ol style="list-style-type: none"> 1. The mobile the device had install the mobile app of the restaurant. 2. The user have turn on the notification service in the mobile.

Use case name:	Download User Guide
Use case ID:	UC003
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff can download view the .pdf file full user guide through the web system.
Flow of events:	<ol style="list-style-type: none"> 1. (include UC001) 2. Restaurant Staff chooses the user guide function. 3. The system shows the user guide or start downloading it.
Post-conditions:	The user guide is uploaded before.
Non-behavior requirements:	The user computer should have Adobe Reader to open the pdf file or the user has installed any third-party pdf viewer in the computer.

Use case name:	Upload Mobile App Icon
Use case ID:	UC004
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff can upload the icons which use in the mobile app.
Flow of events:	<ol style="list-style-type: none"> 1. (include UC001) 2. Restaurant Staff chooses the upload app icon function. 3. The system shows the upload icon page. 4. Restaurant Staff presses the browse button to choose the icon. 5. The system shows the dialog. 6. Restaurant Staff chooses the image file. 7. Restaurant Staff presses the upload button. 8. The system then returns to the upload icon page and display successful message.
Post-conditions:	The image is on the server.
Non-behavior requirements:	The image should be in resolution 64 x 64, if the image is larger then the size, it will auto stretch by the server. Also, the image should be with file extension jpg, png, tiff, bmp only.

Use case name:	Manage Mobile App Function
Use case ID:	UC005
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff can choose which function will show in the app through the system, after submit, the system will update the function of the app.
Flow of events:	<ol style="list-style-type: none"> 1. (include UC001) 2. Restaurant Staff chooses the Manage Mobile App Function. 3. The system shows the mobile app function page. 4. Restaurant Staff chooses the functions. 5. The system updates the functions of the app.

Use case name:	Change Account Password
Use case ID:	UC006
Actor(s):	Restaurant Staff
Brief description:	Administrator can update the password of the login account
Pre-conditions:	
Flow of events:	<ol style="list-style-type: none"> 1. (include UC001) 2. Restaurant Staff chooses the update password page. 3. The system shows the update page. 4. Restaurant Staff input the required information. 5. Restaurant Staff press the submit button. 6. The system updates the information.

Use case name:	Select Restaurant
Use case ID:	UC007
Actor(s):	Restaurant Staff
Brief description:	Before update restaurant information, the Restaurant Staff needs to choose which restaurant needed to be updated.
Pre-conditions:	Administrator choose the restaurant information management system
Flow of events:	<ol style="list-style-type: none"> 1. (include UC001) 2. Administrator selects the Restaurant. 3. The System shows the restaurant page.

Use case name:	View Company News
Use case ID:	UC008
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff can view the news of the restaurant
Pre-conditions:	
Flow of events:	<ol style="list-style-type: none"> 1. (Include UC001) 2. (Include UC007) 3. Administrator chooses the news function. 4. The system shows the news of the restaurant.

Use case name:	View Electric Menu
Use case ID:	UC009
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff can view and choose different type of menu - starter, food, drink and dessert.
Pre-conditions:	
Flow of events:	<ol style="list-style-type: none"> 1. (Include UC001) 2. (Include UC007) 3. Restaurant Staff points to the menu bar. 4. The system shows the type of menu. 5. Restaurant Staff chooses the sub menu. 6. The system shows the sub menu.

Use case name:	View Food Gallery
Use case ID:	UC010
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff can view the Gallery of the Menu
Pre-conditions:	
Flow of events:	<ol style="list-style-type: none"> 1. (Include UC001) 2. (Include UC007) 3. Restaurant Staff chooses the gallery page. 4. System shows the menu gallery. 5. Restaurant Staff chooses and view gallery.
Assumptions:	There are galleries uploaded before in the system.

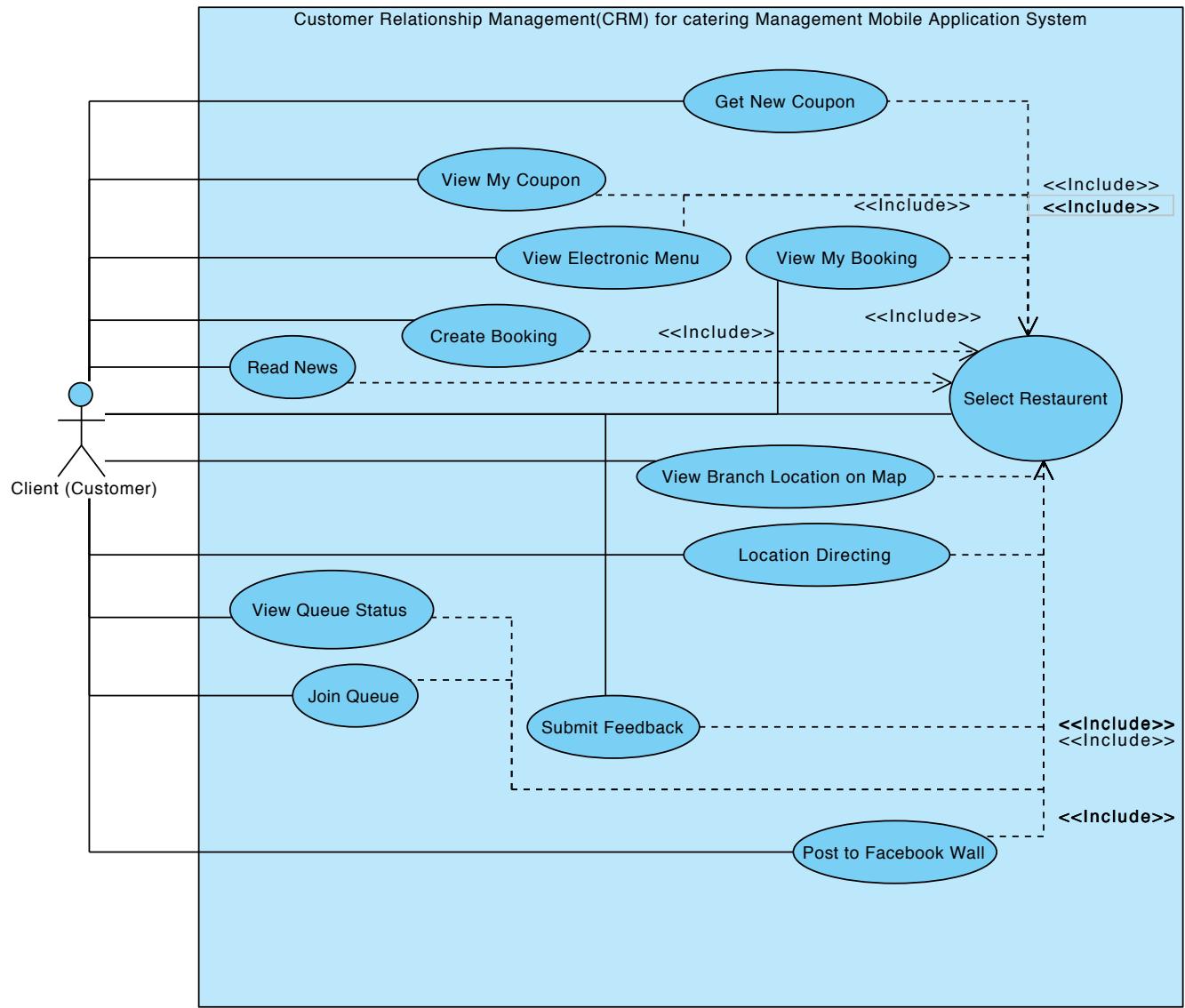
Use case name:	Update Menu
Use case ID:	UC011
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff can update the menu so that the user who use the app can view the restaurant latest menu and information
Pre-conditions:	
Flow of events:	<ol style="list-style-type: none"> 1. (Include UC001) 2. (Include UC007) 3. Restaurant Staff points to the update information bar. 4. The system shows the sub function. 5. Restaurant Staff chooses the update menu function. 6. The system shows the update menu page. 7. Restaurant Staff input the update menu and chooses the menu icon. 8. Restaurant Staff press the submit button. 9. The system updates the menu information.

Use case name:	Update News
Use case ID:	UC012
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff 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:	<ol style="list-style-type: none"> 1. (Include UC001) 2. (Include UC007) 3. Restaurant Staff points to the update information tab. 4. The system shows the sub function. 5. Restaurant Staff chooses the update news function. 6. The system shows the update news page. 7. Restaurant Staff input the news information. 8. Restaurant Staff presses the submit button. 9. The system updates the news information.

Use case name:	Upload New Food Gallery
Use case ID:	UC013
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff 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:	<ol style="list-style-type: none"> 1. (Include UC001) 2. (Include UC007) 3. Restaurant Staff points to the update information tab. 4. The system shows the sub function. 5. Restaurant Staff chooses the Upload Gallery function. 6. The system show the Upload Gallery Pages 7. Restaurant Staff press the browse button 8. The system shows the dialog. 9. Restaurant Staff chooses the photo. 10. Restaurant Staff presses the add button. 11. The system uploads the photo to the directory. 12. The system shows which photo can be update successfully.

Use Case Diagram of Mobile Devices

Visual Paradigm for UML Enterprise Edition(IVE_ICT)



Use Case Description of Mobile Devices

Use case name:	Select Restaurant
Use case ID:	UC-0001
Actor(s):	User
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:	<ol style="list-style-type: none">1. User opens our application.2. System synchronous the database with the server database.3. System shows the available Restaurant and wait for user input/4. User selects a restaurant5. System shows the available function of that restaurant to the user
Postconditions:	
Alternative flows and exceptions:	For the first launch, user require a network connection to receive the Restaurant data

Use case name:	Read News
Use case ID:	UC-0002
Actor(s):	User
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:	
Flow of events:	<ol style="list-style-type: none"> 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. 5. The system shows the detail (title, content) of that news.
Postconditions:	

Use case name:	View Electronic Menu
Use case ID:	UC-0003
Actor(s):	User
Brief description:	The user can read the menu of restaurant
Preconditions:	
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user click 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.
Postconditions:	

Use case name:	Post to Facebook Wall
Use case ID:	UC-0004
Actor(s):	User
Brief description:	After a user have browse a menu item, they can share it with their friends
Preconditions:	User have a Facebook Account
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user click 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 info and login 9. The system post the food detail to their Facebook Wall
Postconditions:	A Food detail is posted to the user's Facebook wall
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	View My Coupon
Use case ID:	UC-0005
Actor(s):	User
Brief description:	The user can view the coupon they owned
Preconditions:	The user have at least one coupon
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user click 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.

Use case name:	Get New Coupon
Use case ID:	UC-0006
Actor(s):	User
Brief description:	A Restaurant may release coupons periodically. The user can get the coupons
Preconditions:	
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user click the button "Coupon" 3. The system shows a list of coupons he / she owned 4. The user click 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 go back to "View My Coupon" and the coupon is added to my coupon.

Use case name:	View Queue Status
Use case ID:	UC-0007
Actor(s):	User
Brief description:	Many users may be queuing for a restaurant. The users can view their position in the queue.
Preconditions:	The user have joined the queue
Flow of events:	<ol style="list-style-type: none"> 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

Use case name:	Join Queue
Use case ID:	UC-0008
Actor(s):	User
Brief description:	When a restaurant is full, user have to join the queue to wait for a table.
Preconditions:	
Flow of events:	<ol style="list-style-type: none"> 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 has join the queue.

Use case name:	View My Booking
Use case ID:	UC-0009
Actor(s):	User
Brief description:	The user can book a table for a specific time.
Preconditions:	
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user clicks the button "Booking" 3. The system shows a list of booking the user have made. 4. The user selects a booking for details 5. The system show the details of the selected booking.

Use case name:	Create Booking
Use case ID:	UC-0010
Actor(s):	User
Brief description:	The user can book a table for a specific time.
Preconditions:	
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user click the button "Booking" and click the button of "Create Booking" 3. The system shows form for create 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"

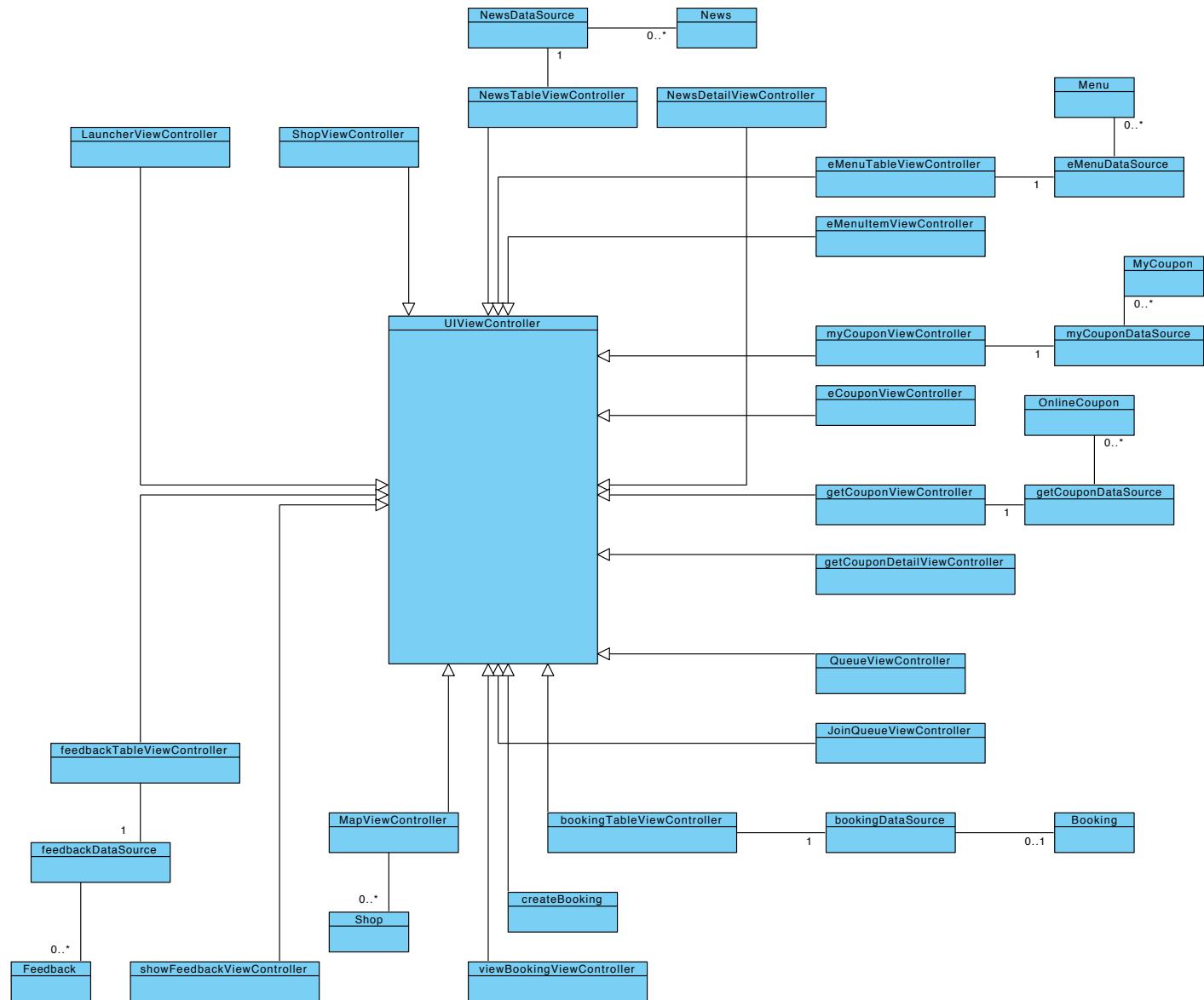
Use case name:	View Branch Location on Map
Use case ID:	UC-0011
Actor(s):	User
Brief description:	The user can find the location of restaurant on the map
Preconditions:	
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user click the button "Map" 3. The system shows a map and mark the location of restaurant and user

Use case name:	Location Directing
Use case ID:	UC-0012
Actor(s):	User
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:	
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user click the button "Map" 3. The system shows a map and mark the location of restaurant and user 4. The user selects the restaurant and click route. 5. The system shows a path from the user to the restaurant.

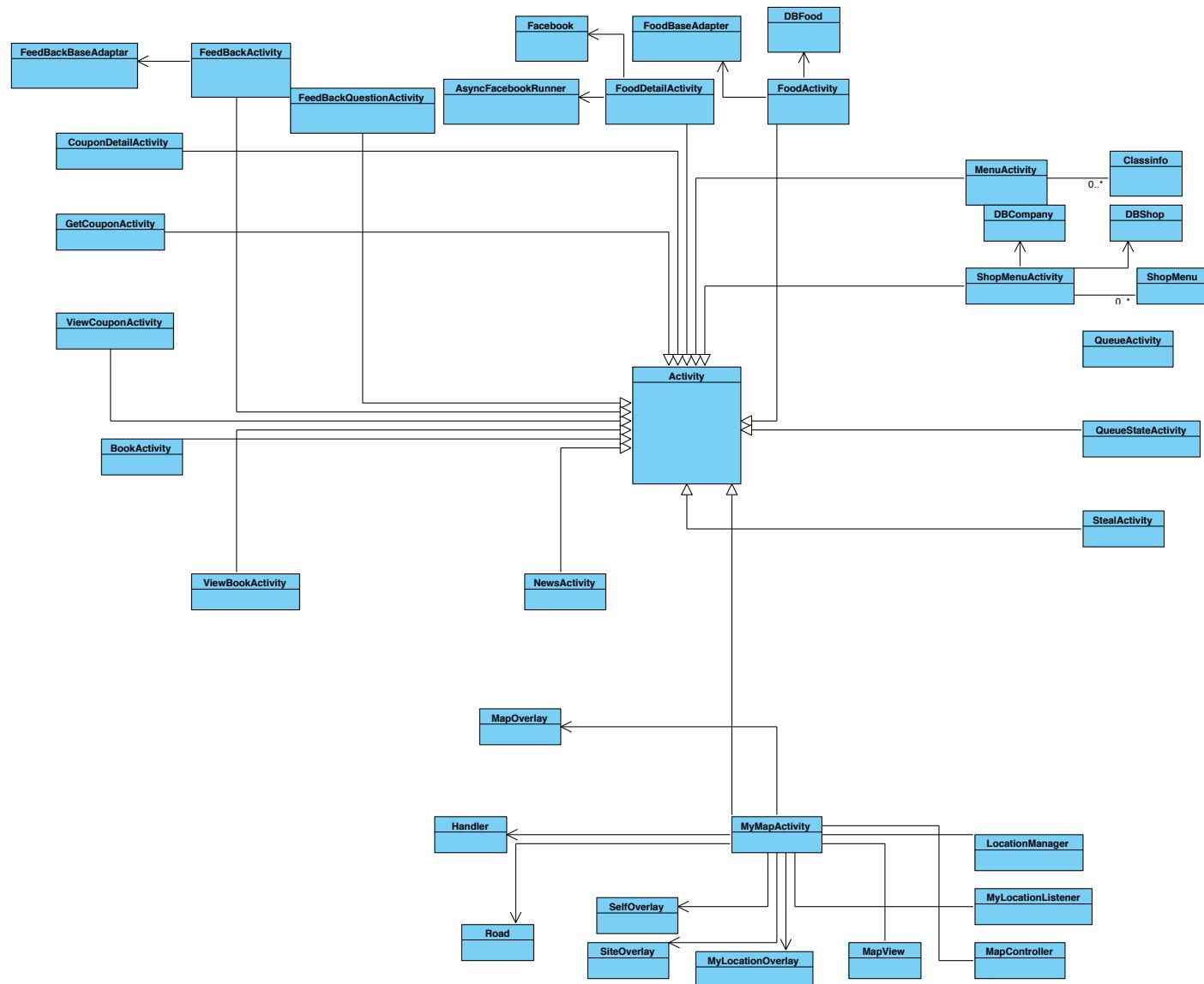
Use case name:	Submit Feedback
Use case ID:	UC-0013
Actor(s):	User
Brief description:	The user can submit feedback to restaurant.
Preconditions:	
Flow of events:	<ol style="list-style-type: none"> 1. Include (select Restaurant) 2. The user click the button "Feedback" 3. The system shows a list of available questionnaire 4. The user selects a questionnaire 5. The system shows the question of that questionnaire. 6. The user enters the answers and submits. 7. The system shows the success message.
Postconditions:	

Class Diagrams

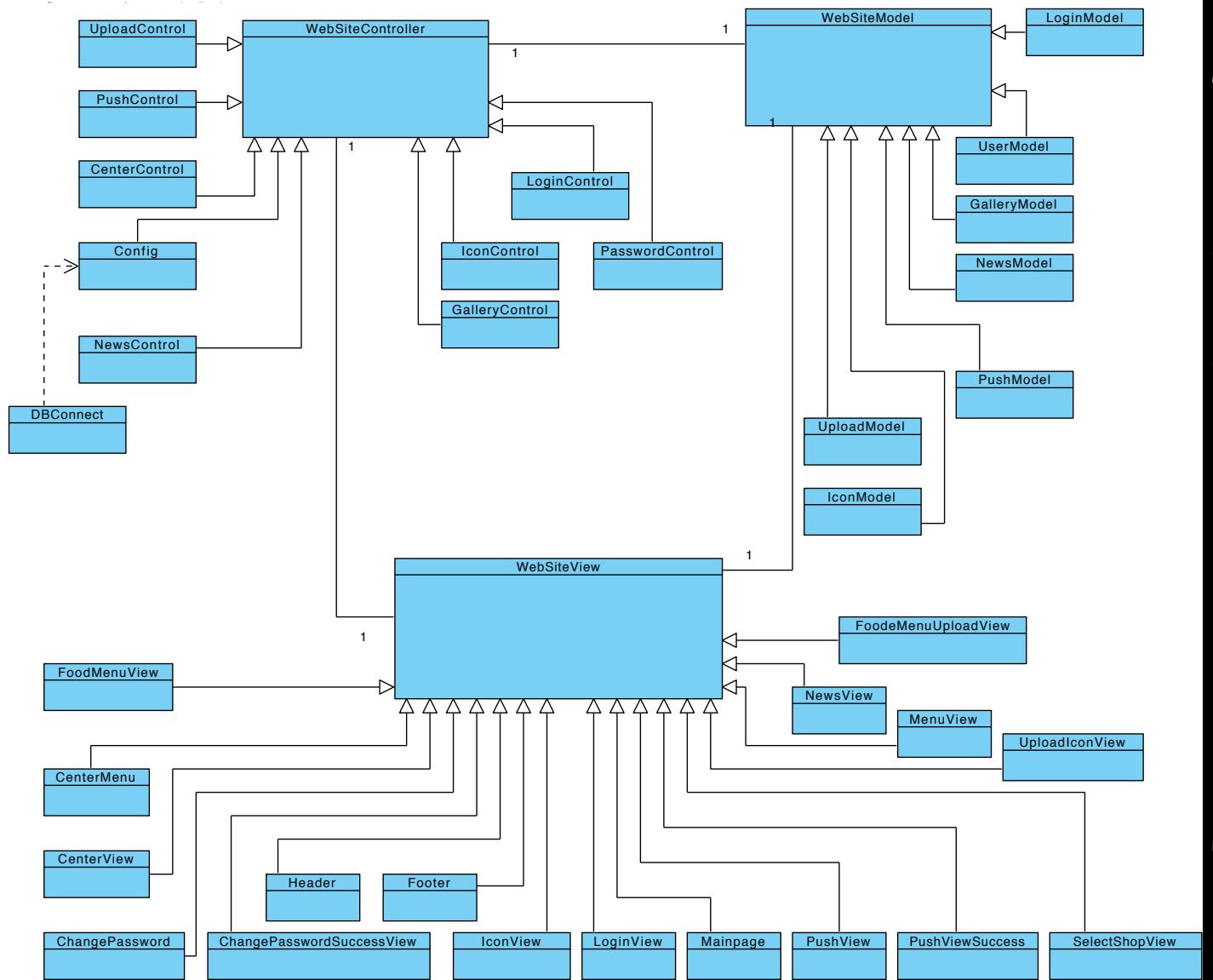
Class Diagram of iPhone Application



Class Diagram of Android Application



Class Diagram of Web Service



ERD Design

ERD Diagram

----- Stick ERD Diagram Here -----

ERD Descriptions

Crm_app_icon

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
User_id	VARCHAR(20)	No	The unique number of each user id
Icon_url	VARCHAR(255)	No	The actual path of the icon

Crm_booking

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Booking_id	Int(11)	No	Unique id of each booking
Shop_id	Int(11)	No	Unique id of each shop
Name	Text	No	The name of the customer who make a booking
Seats	Tinyint(4)	No	The seats number of the booking
Type	Tinyint(4)	No	The booking type
Phone	Tinytext	No	The phone number of the customer
Device_id	Text	No	The unique id of the booking device
Create_date	Date	No	The created date of the booking
Booking_time	Datetime	No	The booking time of the booking
Special_requirement	Text	No	The requirement of the booking
Status	Tinyint(4)	No	The booking state

Crm_company

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Company_id	Int(11)	No	The unique id of each company
Company_name	Text	No	The name of the company
Last_update	Int(11)	No	The status to check if there are any update

Crm_coupon

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Coupon_group_id	Int(11)	No	The unique id of the coupon_group
Shop_id	Int(11)	No	The unique id of the shop
Coupon_title	Tinytext	No	The title of the coupon
Coupon_description	Text	No	The content of the coupon
Distribution_begin	Date	No	The time of beginning distribute the coupon
Distribution_end	Date	No	The time of stop the coupon

Crm_coupon_code

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Coupon_group_id	Int(11)	No	The unique id of the coupon_group
Code	Varchar(64)	No	The unique code of each coupon
Type	Tinyint(4)	No	The type of the coupon
Device_id	Mediumtext	No	The unique id of the device
Expiry_date	Date	No	The expiry date of the coupon
Used	Datetime	No	The status of the coupon if it was used

Crm_device

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Id	Int(11)	No	The unique id of each device record
Type	Tinyint(4)	No	The type of the device
Device_id	Mediumtext	No	The unique id of the device
Device_name	Mediumtext	No	The name of each device
Phone	Varchar(20)	No	The phone number of the device

Crm_emenu

Entity	A abstract of all identities		
Description			
Name	Data Type	Nullable	Data Description
Item_id	Int(11)	No	The unique id of each item
Shop_id	Int(11)	No	The unique id of the shop
Type	Tinytext	No	The type of the item
Item_name	Tinytext	No	The name of the item
Item_desc	Mediumtext	No	The description of the item
Price	Double	No	The price of the item
Img_url	Tinytext	No	The actual path of the image of the item
Last_update	Int(11)	No	To check if the information of the item was updated
Hidden	Tinyint(4)	No	Used to show the item in the menu

Crm_error_log

Entity	A abstract of all identities		
Description			
Name	Data Type	Nullable	Data Description
Id	Int(11)	No	The unique log id
Content	Text	No	The content of the error log

Crm_feedback

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Feedback_id	Int(11)	No	The unique id of the feedback
Feedback_title	Mediumtext	No	The title of the feedback

Crm_feedback_cust

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Feedback_id	Int(11)	No	The unique id of the feedback
Feedback_question_id	Int(11)	No	The question id of the feedback
Cust_device_id	Varchar(64)	No	The device id of the customer who make the feedback
Cust_ans	Text	No	The content of the feedback from the customer

Crm_feedback_question

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Feedback_question_id	Int(11)	No	The question id of the feedback
Feedback_id	Int(11)	No	The unique id of the feedback
Feedback_question	Text	No	The question of the feedback
Type	Tinyint(4)	No	The type of the feedback

Crm_feedback_question_mc_ans

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Feedback_question_id	Int(11)	No	The question id of the feedback
Choice	Text	No	The choice of the mc question
seq	Smallint(6)	No	The sequence of the question

Crm_feedback_shop

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Feedback_id	Int(11)	No	The unique id of the feedback
Shop_id	Int(11)	No	The unique id of the shop

Crm_news

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Id	Int(10)	No	The unique id of each news
Shop_id	Int(11)	No	The unique id of the shop
Title	Mediumtext	No	The title of the news
Content	Text	No	The content of the news
Date	Date	yes	The date of the news
Last_update	Int(11)	No	the status to check if the news was updated
Hidden	Tinyint(4)	No	To check if the news need to show

Crm_order

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Order_id	Int(10)	No	The unique id of each order
Secret_code	Smallint(11)	No	The secret_code of the order
Shop_id	Int(10)	No	The unique id of the shop
Date_time	Datetime	No	The date and time of the order
Overall_mark	Tinyint(11)	yes	The mark of the order
Feedback	Text	yes	The content of feedback

Crm_order_detail

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Order_id	Int(10)	No	The unique id of each order
Item_id	Int(10)	No	The unique id of the item
Mark	Tinyint(4)	yes	The mark of the order

Crm_queue

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Create_date	Date	No	The create date of the queue
Shop_id	Int(11)	No	The unique id of the shop
Queue_id	Int(11)	No	The unique id of the queue
Name	Text	No	The name of the customer
Seats	Tinyint(4)	No	The seats number of the queue
Type	Tinyint(4)	No	The type of the phone
Phone	Tinytext	No	The phone number of the queue
Device_id	Text	No	The device id of the queue

Crm_shop

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Shop_id	Int(11)	No	The unique shop id
Company_id	Int(11)	No	The unique id of the company
Shop_name	Text	No	The name of the shop
Current_queue_id	Int(11)	No	The id of current queue
Latitude	Double	No	The latitude of the queue
longitude	Double	No	The long latitude of the queue
Display_name	Varchar(50)	No	The display name of the shop
Display_subtitle	Varchar(255)	No	The display subtitle of the shop
Icon_url	Mediumtext	No	The actual path of the icon of the shop
Enable	Tinyint(4)	No	To check if it is enable to show this shop on the device
News	Tinyint(4)	No	To check if it is enable to show the news function on the device
menu	Tinyint(4)	No	To check if it is enable to show the menu function on the device
coupon	Tinyint(4)	No	To check if it is enable to show the coupon function on the device
queue	Tinyint(4)	No	To check if it is enable to show the queue function on the device
booking	Tinyint(4)	No	To check if it is enable to show the booking function on the device
map	Tinyint(4)	No	To check if it is enable to show the map function on the device
feedback	Tinyint(4)	No	To check if it is enable to show the feedback function on the device

Crm_theme

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
Company_id	Int(11)	No	The unique id of the company
Navigation_bar	Varchar(100)	No	The actual path of the navigation bar
Bg	Varchar(100)	No	The actual path of the background
News	Varchar(100)	No	The actual path of the news icon
Menu	Varchar(100)	No	The actual path of the menu icon
Coupon	Varchar(100)	No	The actual path of the coupon icon
Queue	Varchar(100)	No	The actual path of the queue icon
Booking	Varchar(100)	No	The actual path of the booking icon
Map	Varchar(100)	No	The actual path of the map icon
Feedback	Varchar(100)	No	The actual path of the feedback icon
Last_update	Int(11)	No	The status to check if the theme was updated

Crm_user

Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
User_id	Int(11)	No	The unique id of the user
Type	Tinyint(4)	No	The type of the user's device
Device_id	Mediumtext	No	The unique id of the device
Device_name	Mediumtext	No	The name of the device
Phone	Varchar(20)	No	The phone number of the device

Crm_web_user

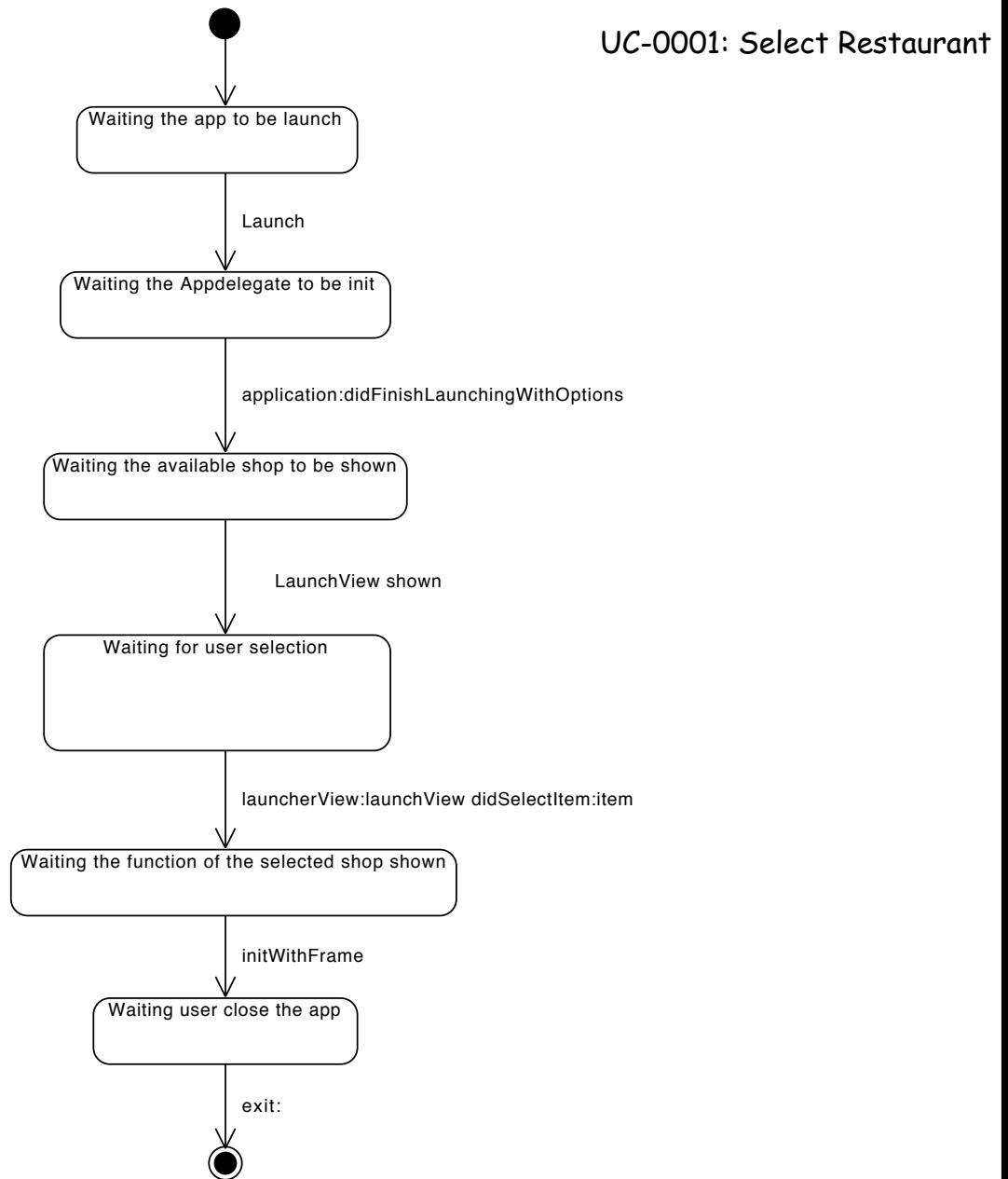
Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
User_id	Varchar(20)	No	The login id of the web
Password	Varchar(32)	No	The login password of the web
Email	Varchar(100)	No	The email address of the user
Company_id	Int(11)	No	The unique id of the company
Group	Int(2)	No	The group id of the user

Crm_web_usergroup

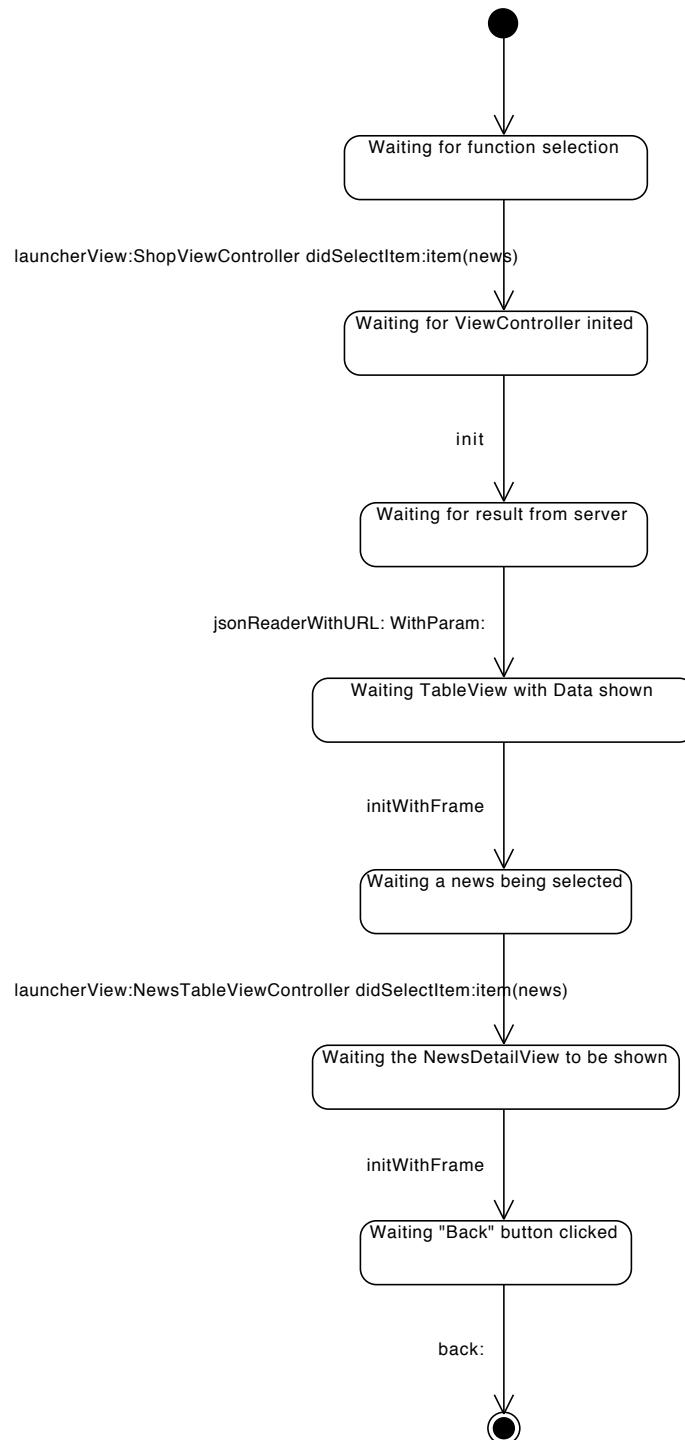
Entity Description	A abstract of all identities		
Name	Data Type	Nullable	Data Description
User_group_id	Int(2)	No	The id of the user group
Group_name	Varchar(30)	No	The name of the user group
Full_control	Int(1)	No	The full control permission of the user

State Transition Diagram

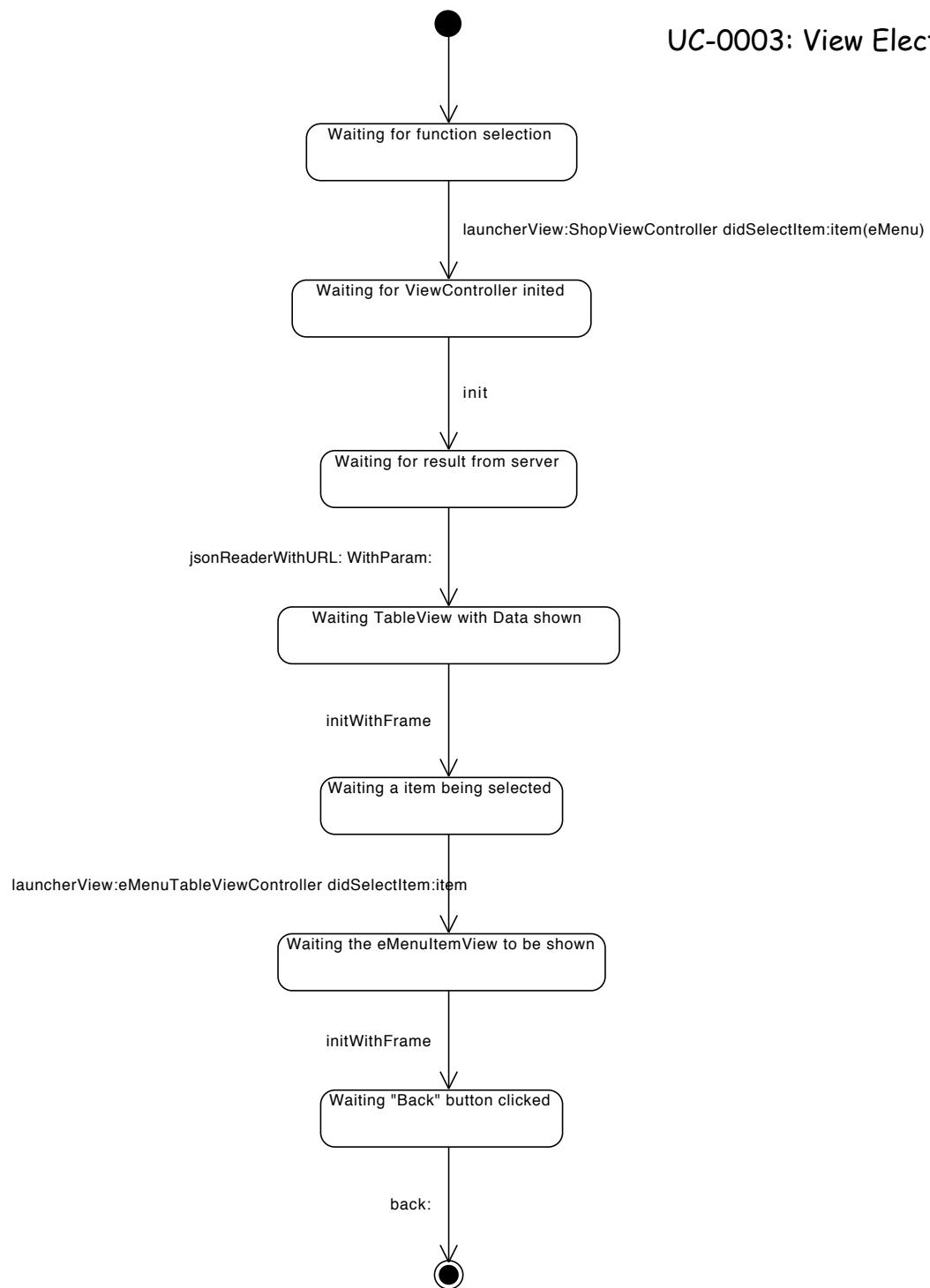
State Transition Diagram of iPhone Application

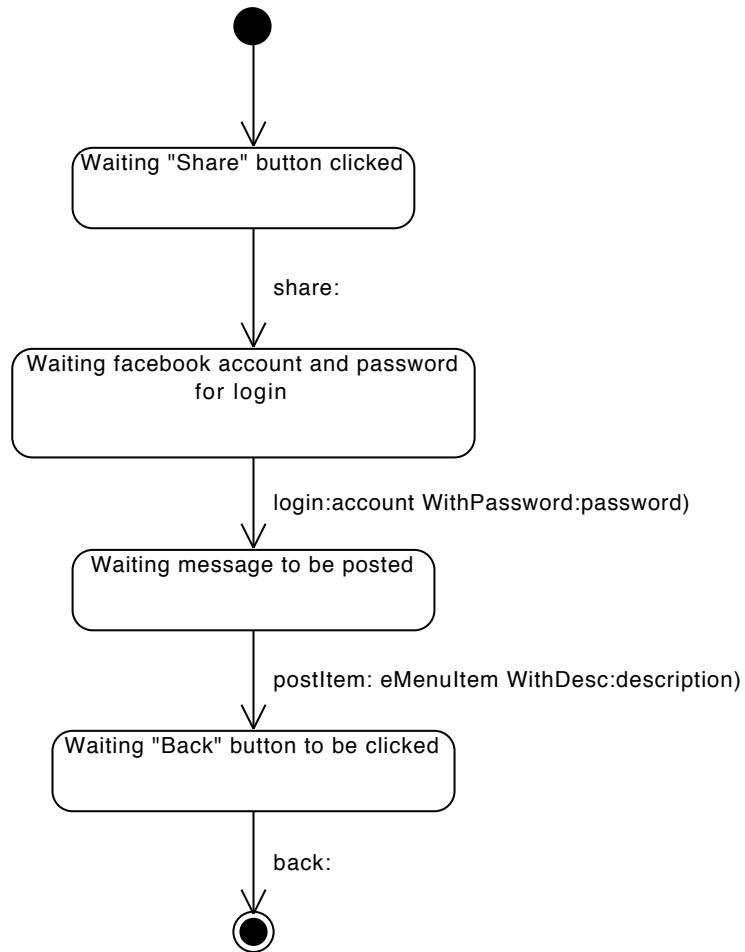


UC-0002: Read News

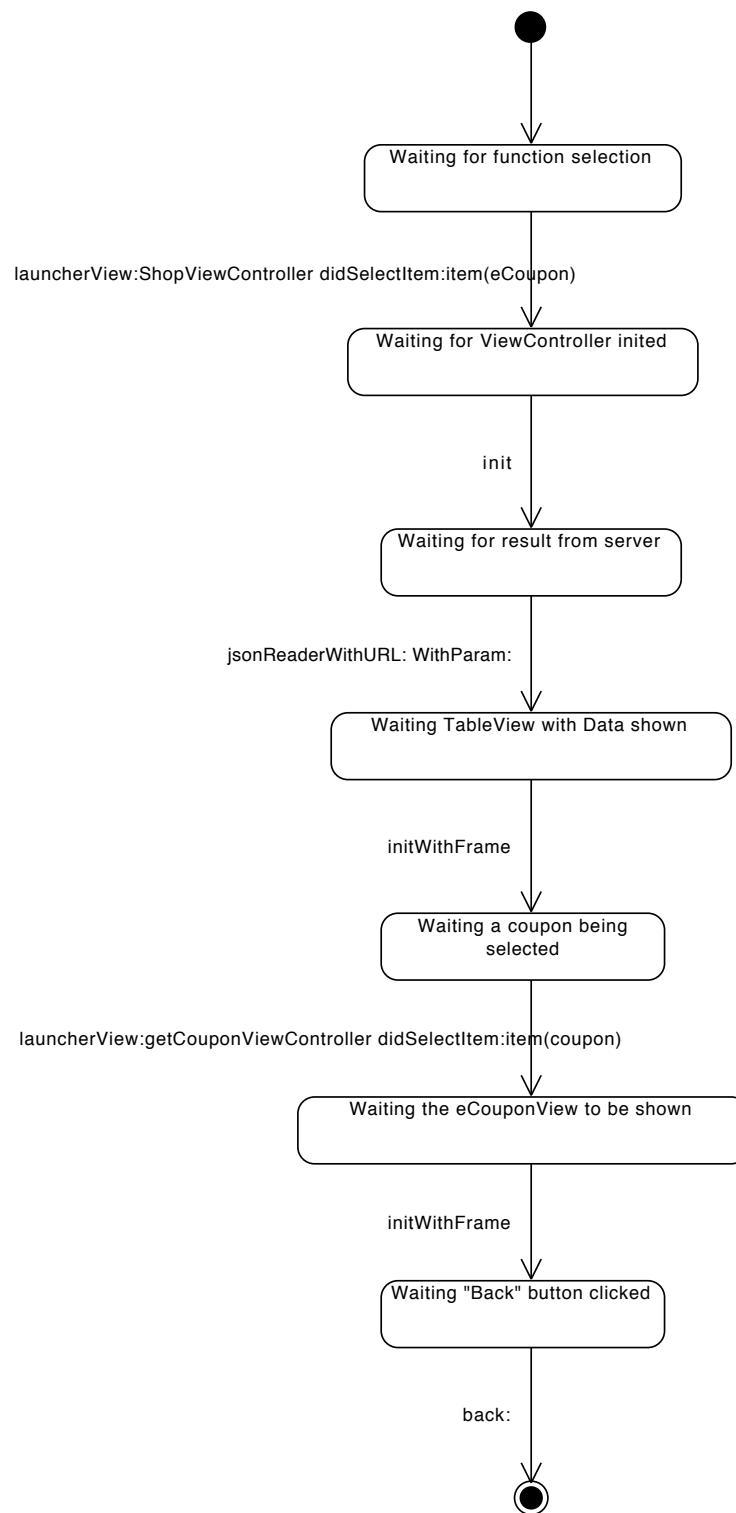


UC-0003: View Electronic Menu

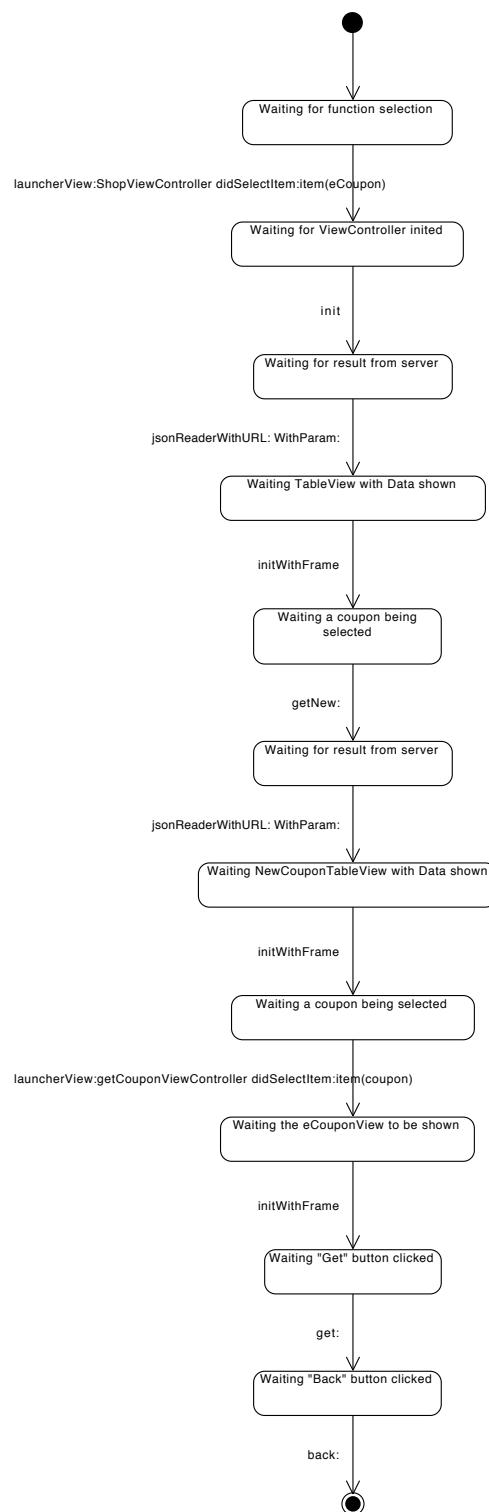




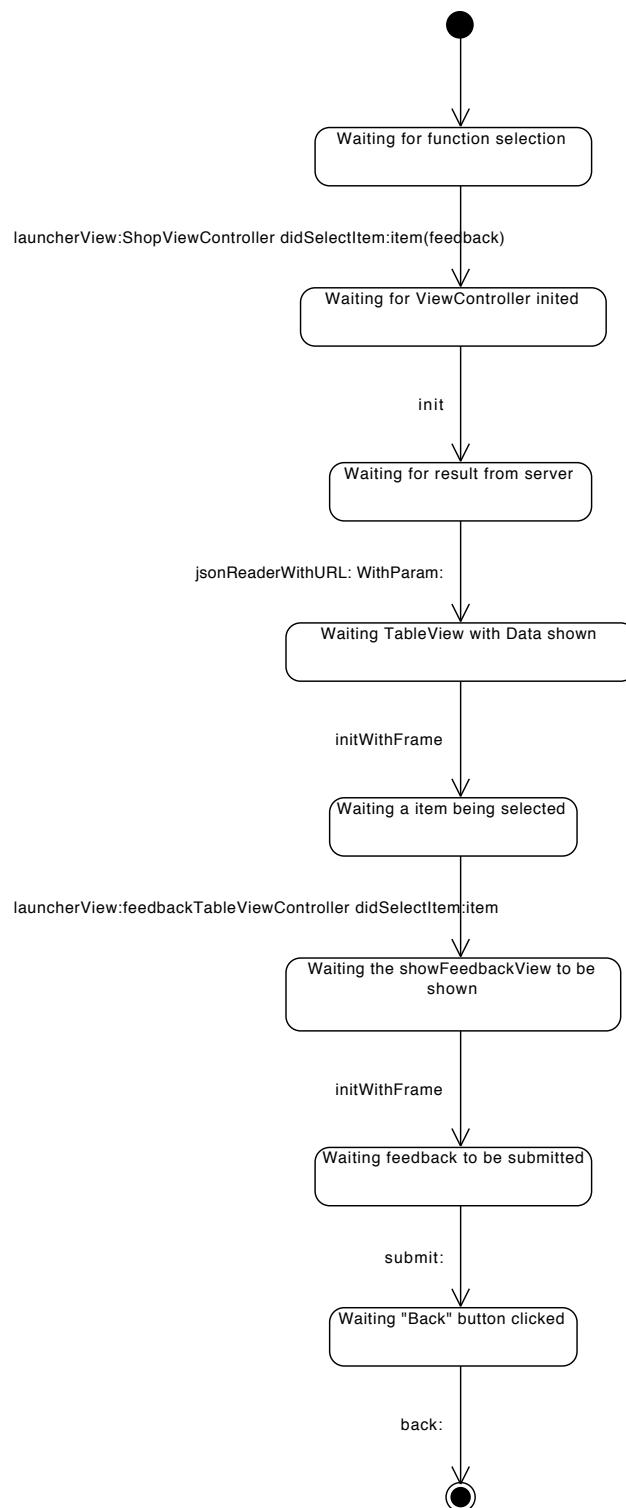
UC-0005: View My Coupon



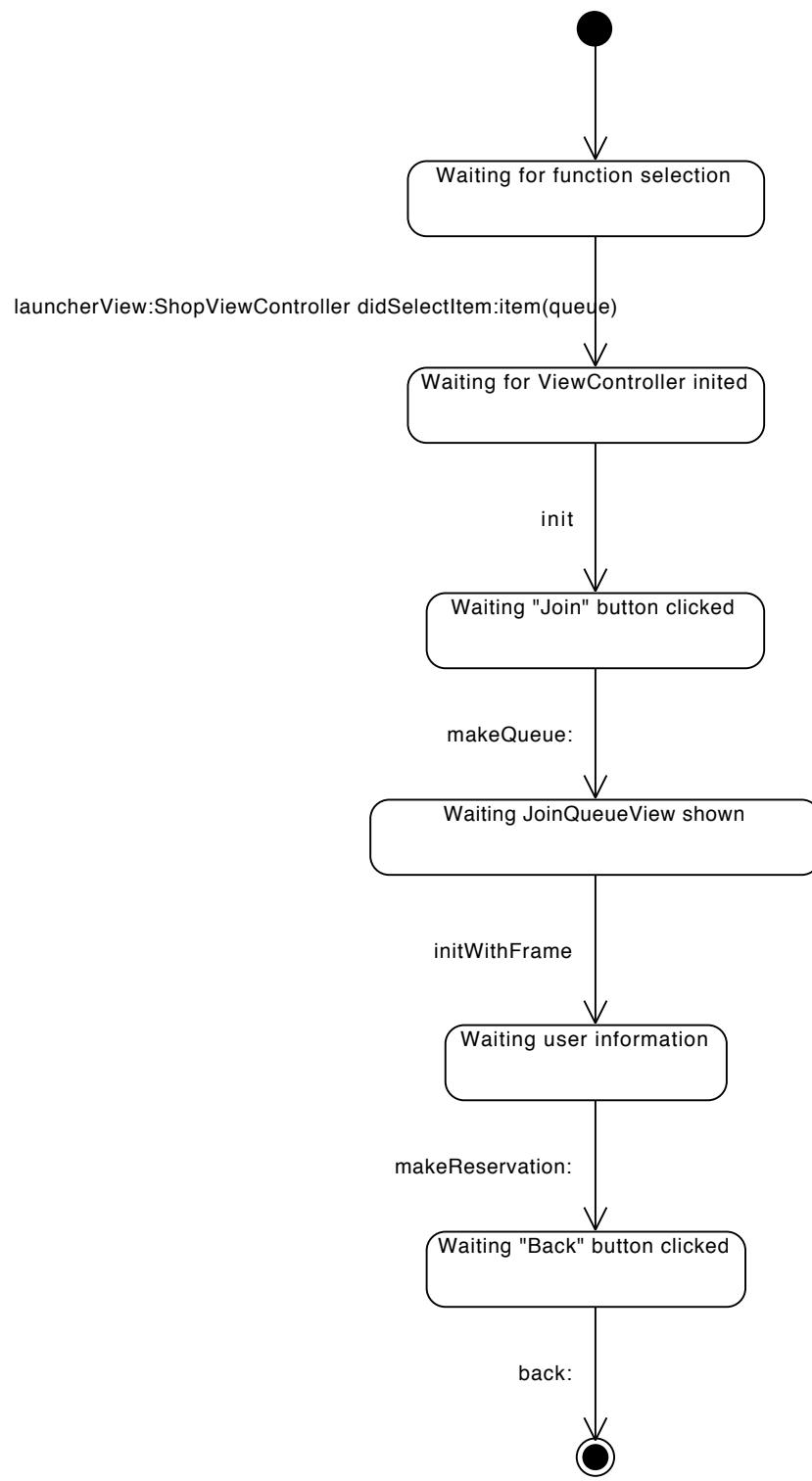
UC-0006: Get New Coupon



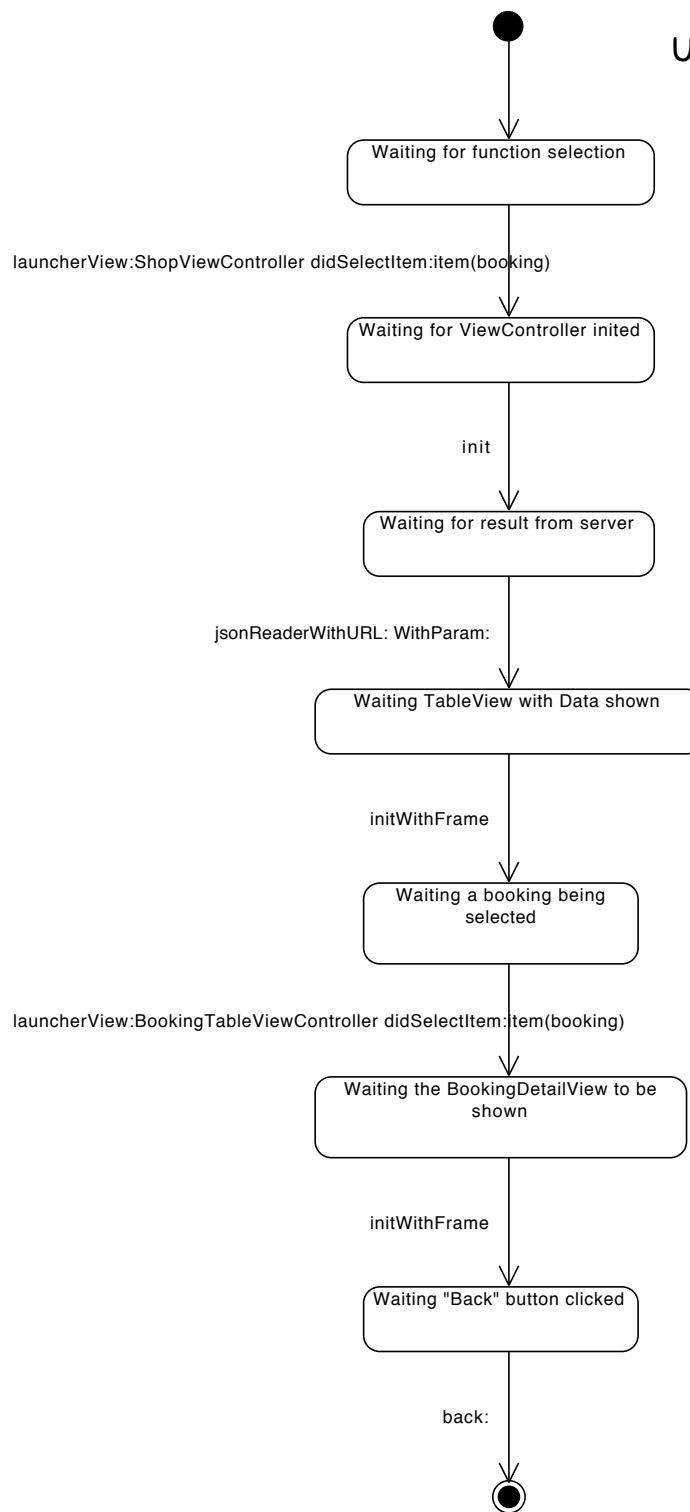
UC-0007: View Queue States



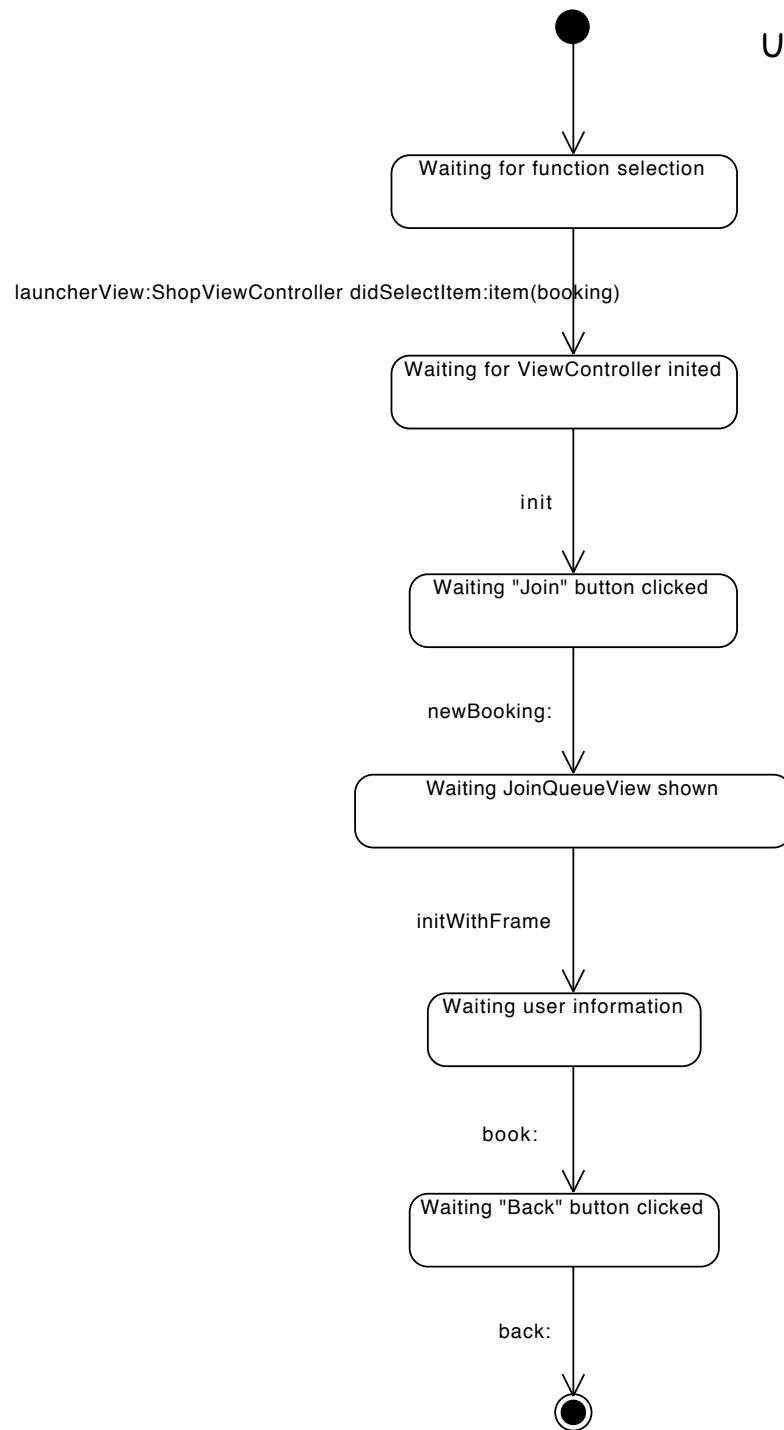
UC-0008: Join Queue



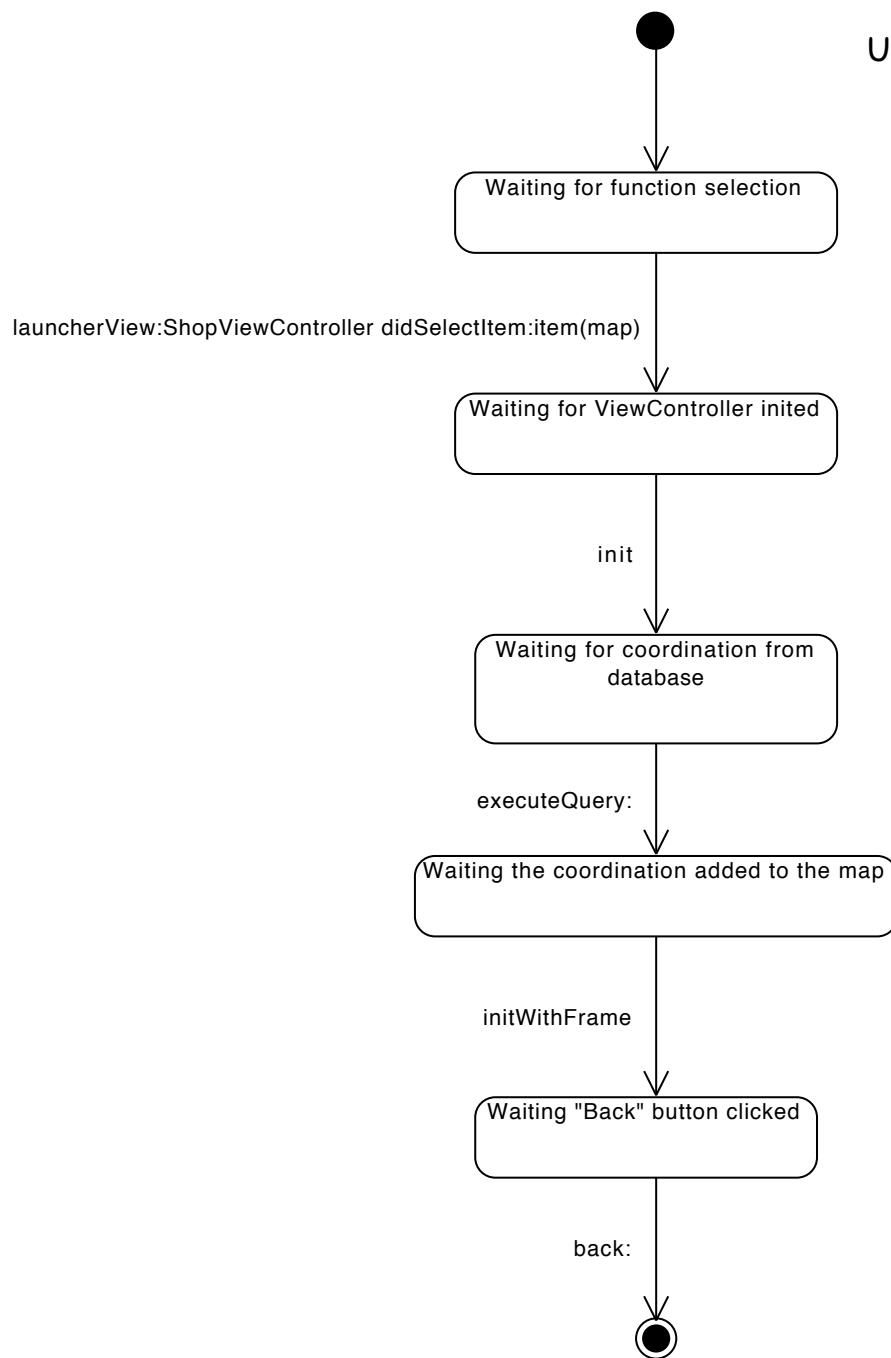
UC-0009: View My Booking

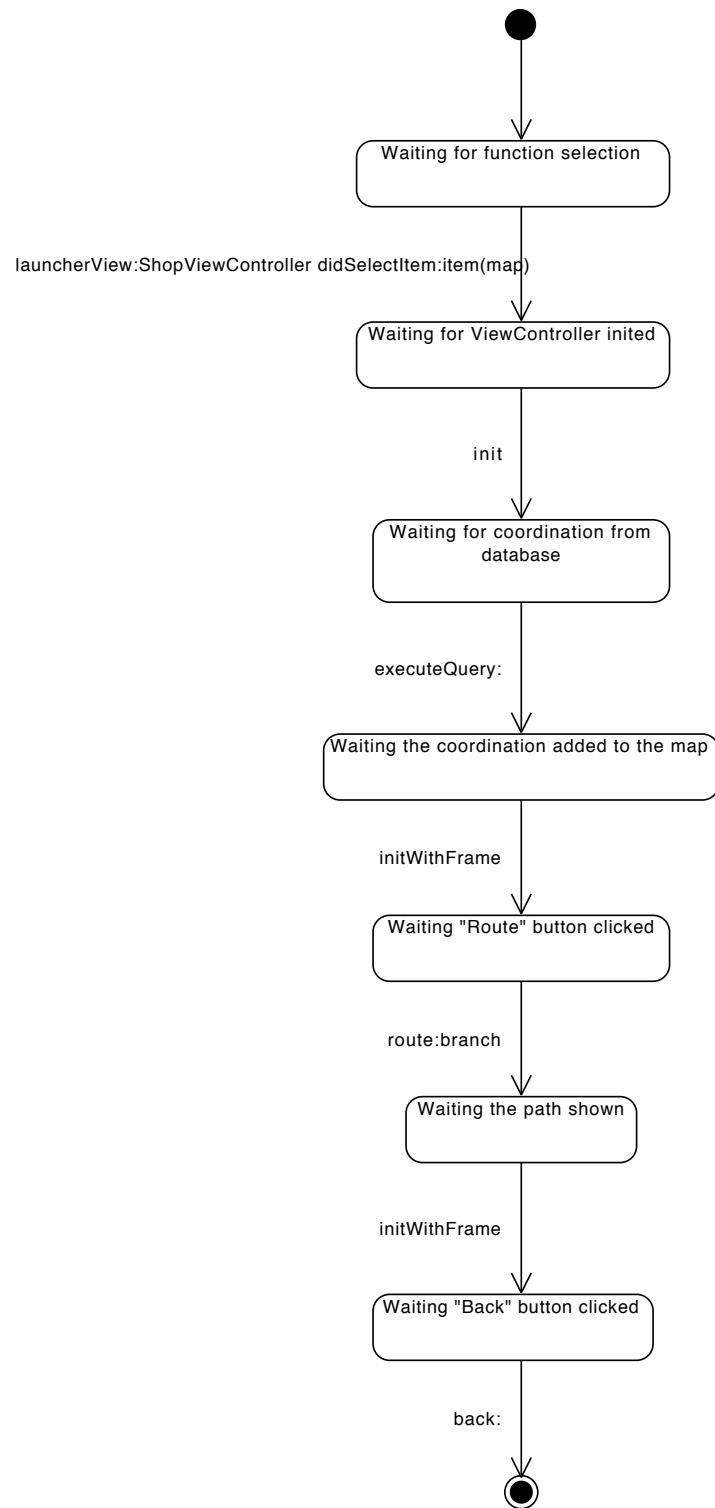


UC-0010: Create Booking

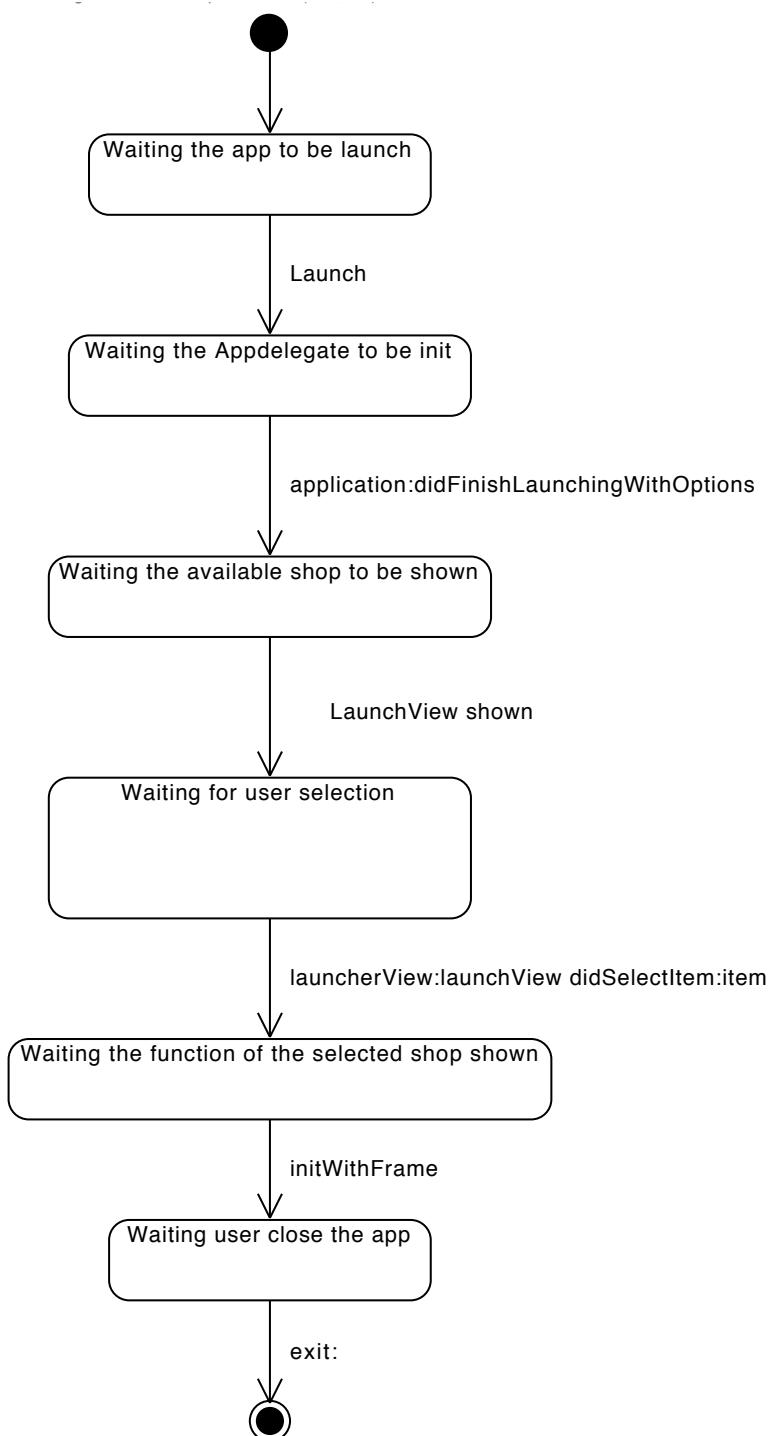


UC-0011: Create Booking

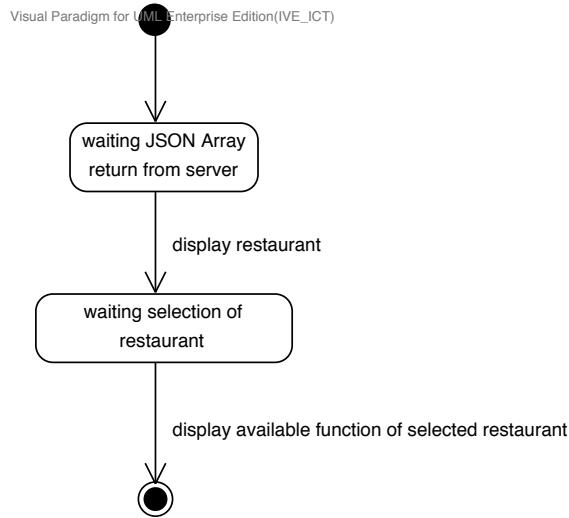




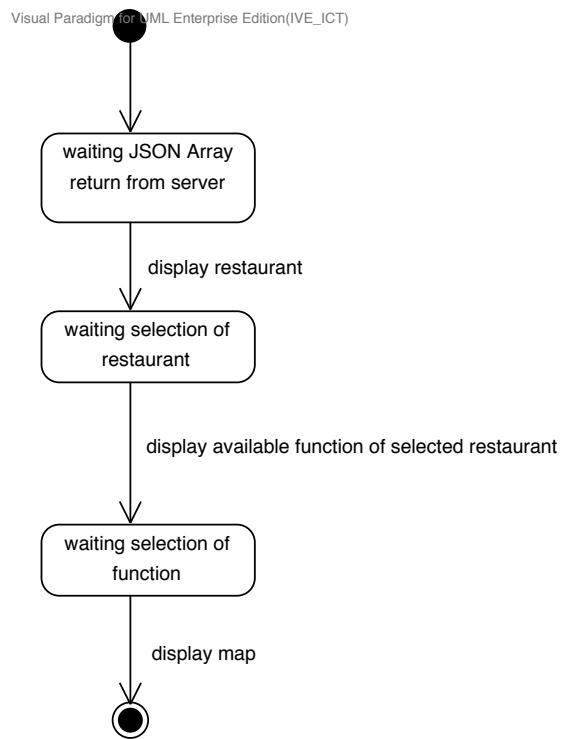
UC-0013: Submit Feedback



State Transition Diagram of Android Application

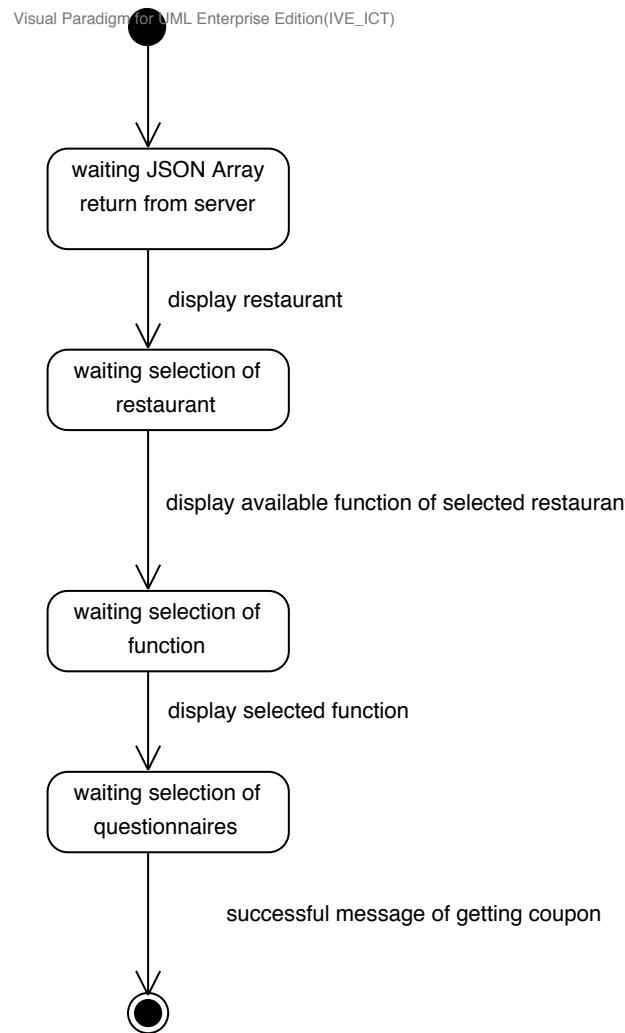


UC-0001: Select Restaurant

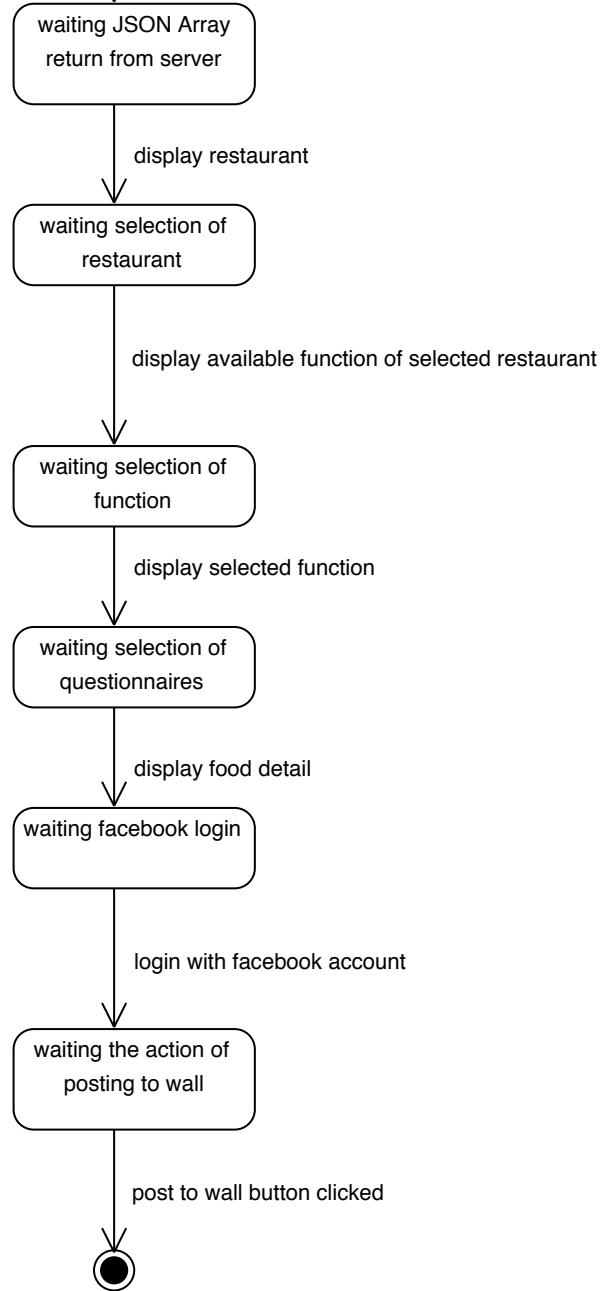


UC-0002: Read News

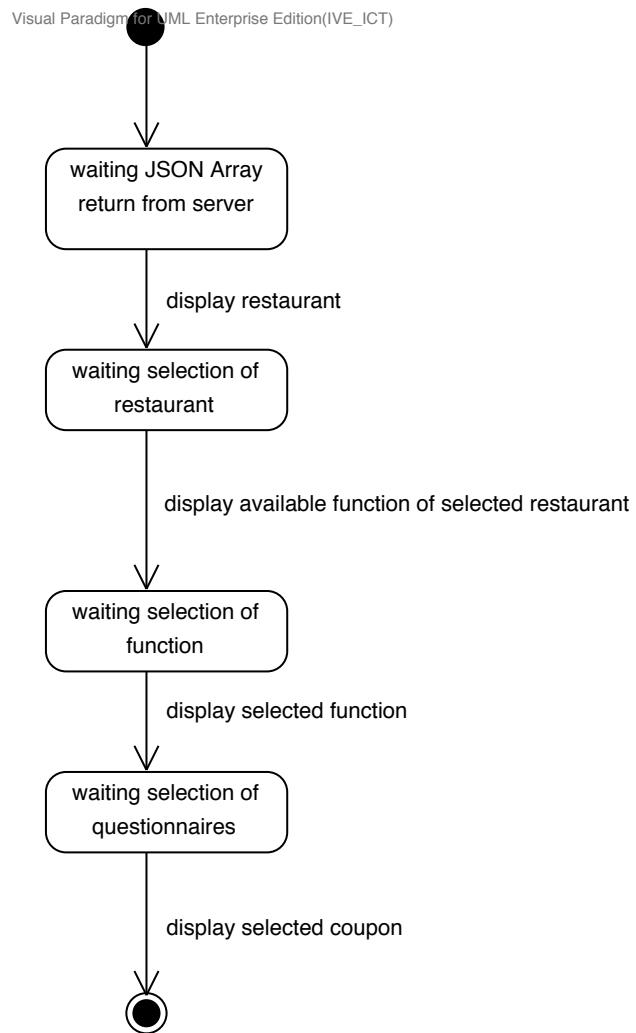
UC-0003: View Electronic Menu

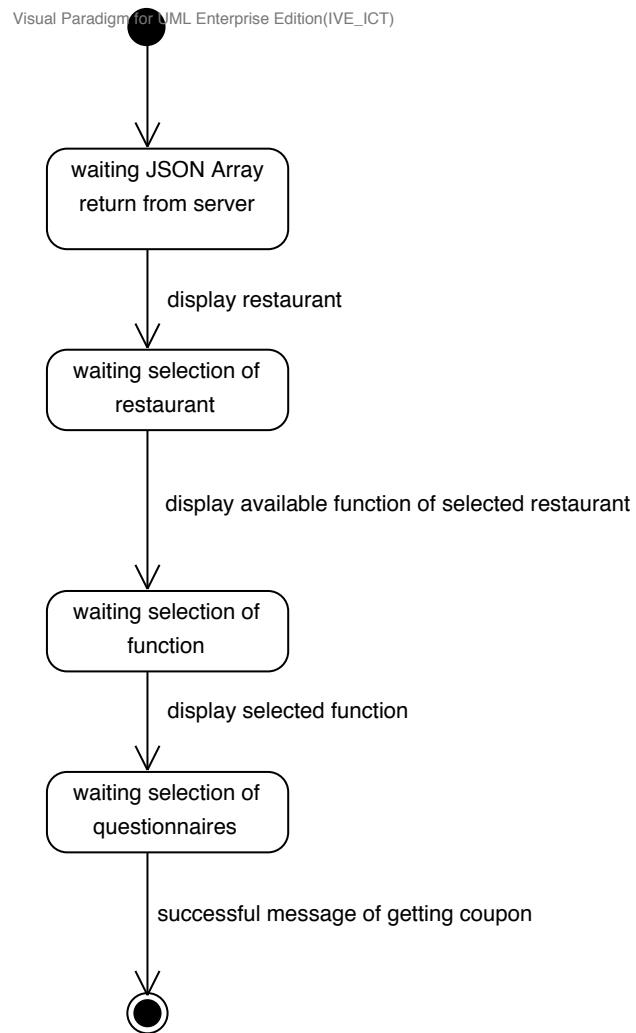


UC-0004: Post to Facebook Wall

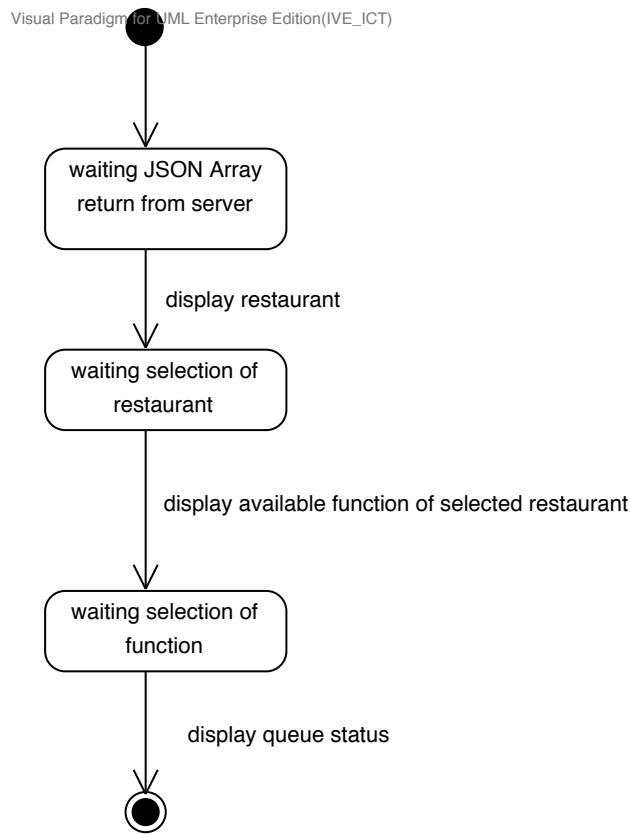


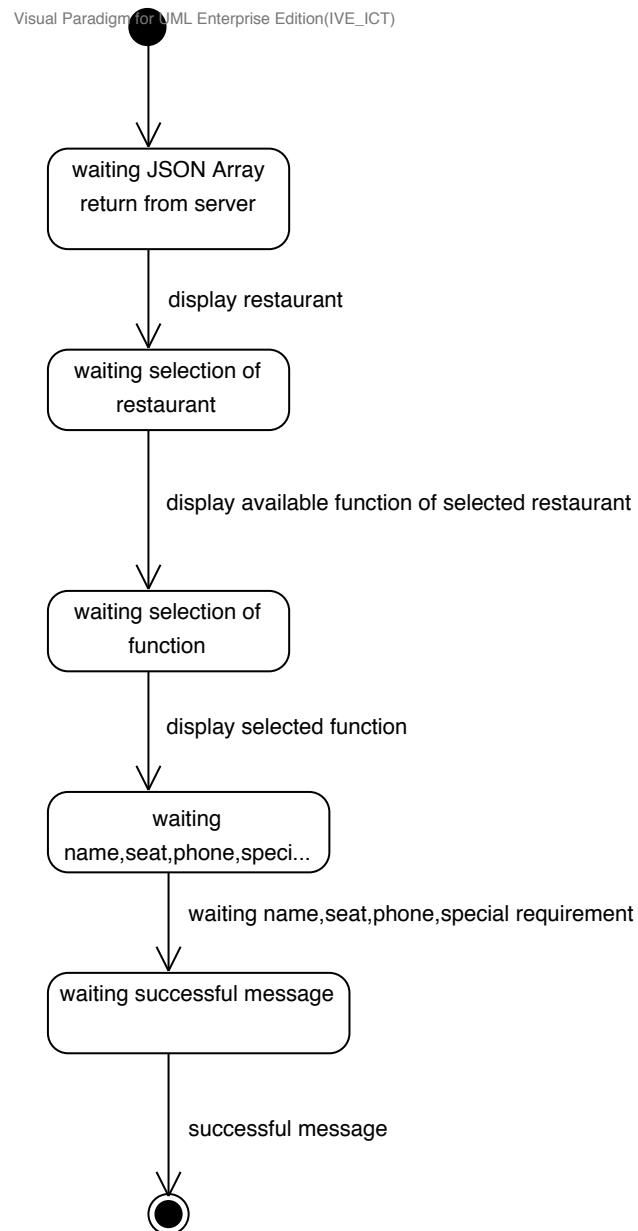
UC-0005: View My Coupon



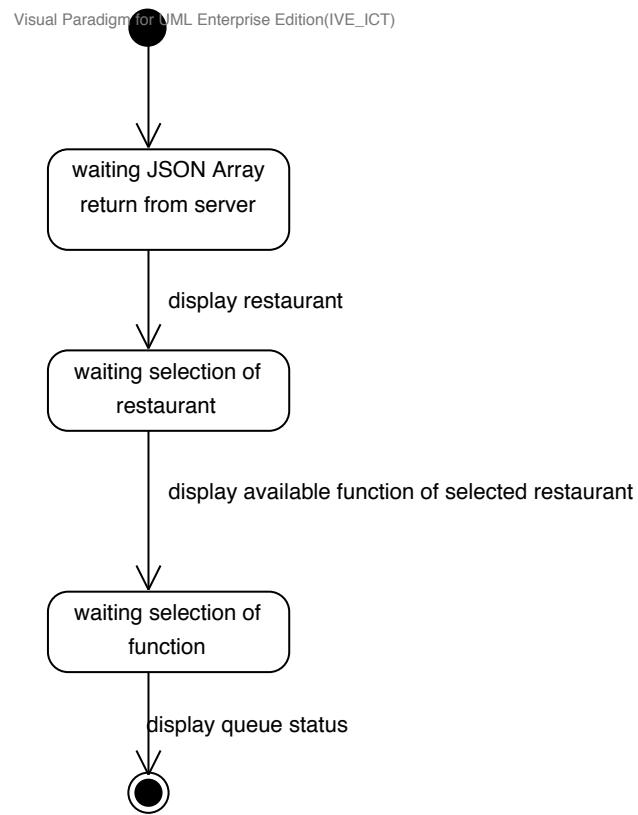


UC-0007: View Queue States

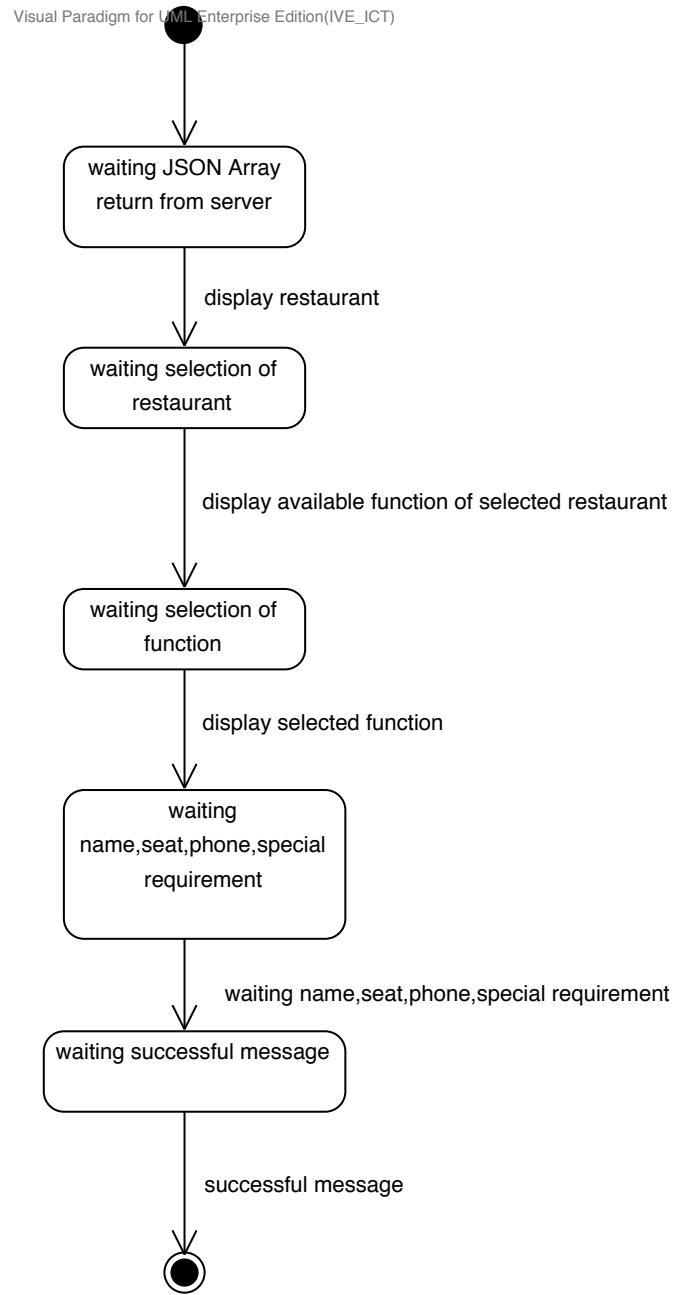




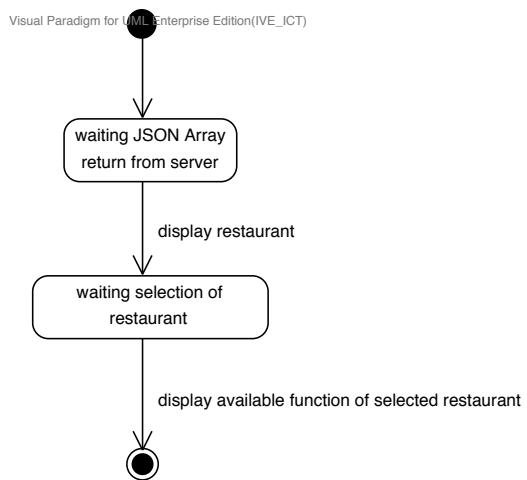
UC-0009: View My Booking



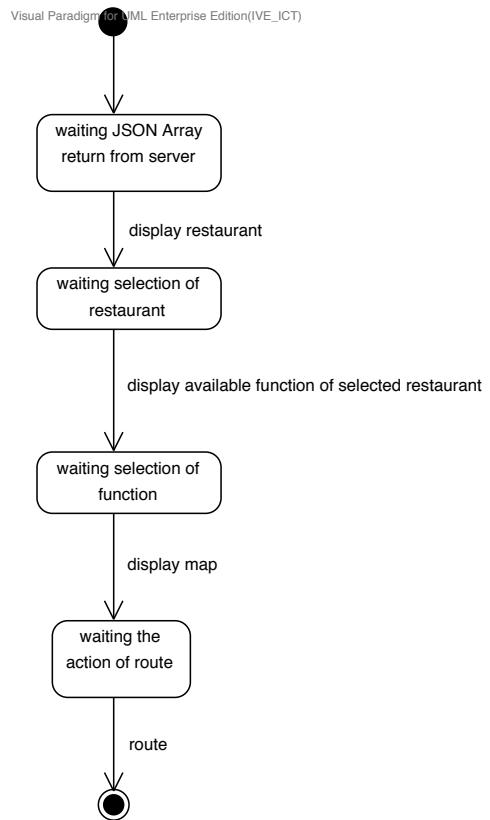
UC-0010: Create Booking



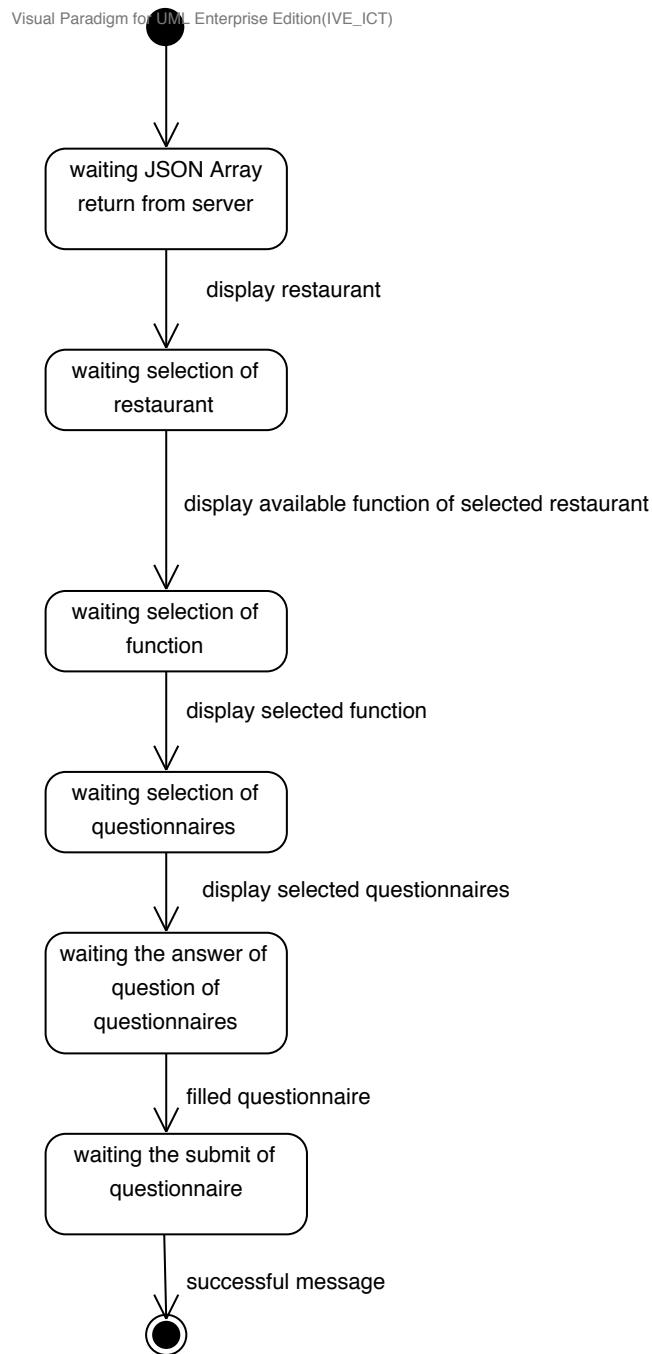
UC-0011: Create Booking



UC-0012: Location Directing



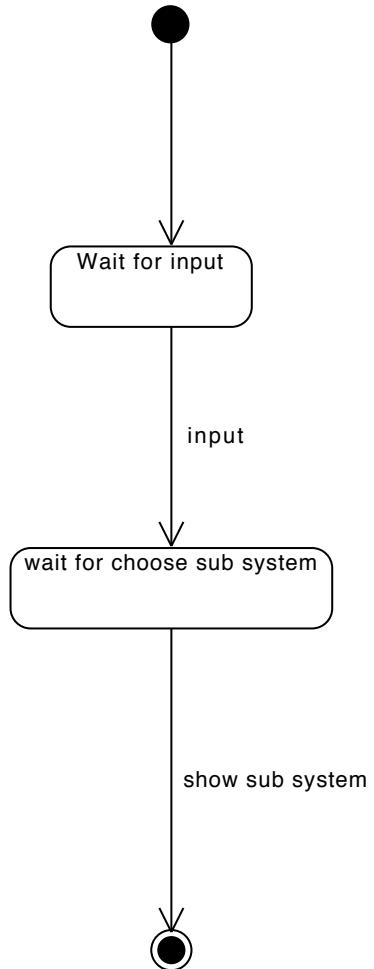
UC-0013: Submit Feedback



State Transition Diagram of Web Service

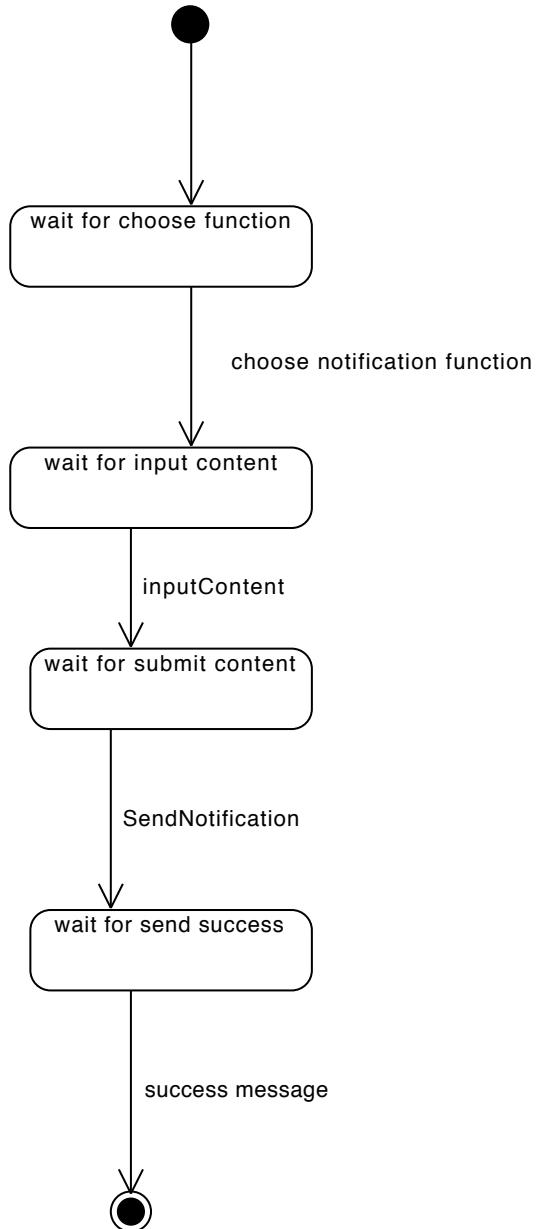
UC001: Login the System

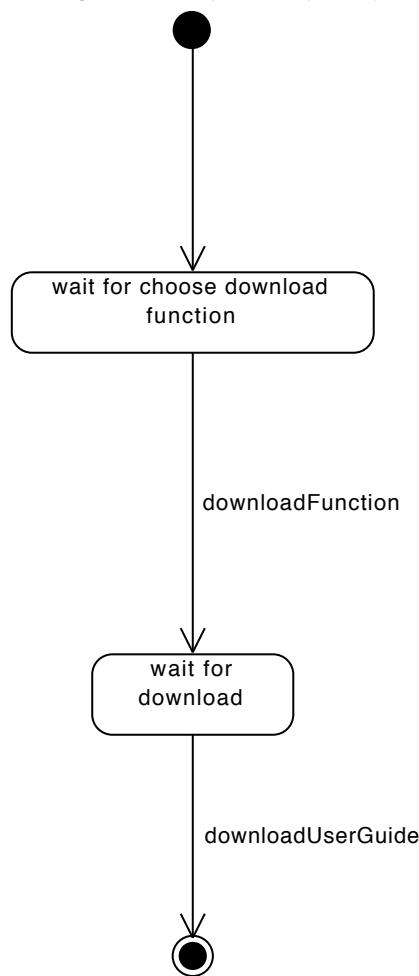
Visual Paradigm for UML Enterprise Edition(IVE_ICT)



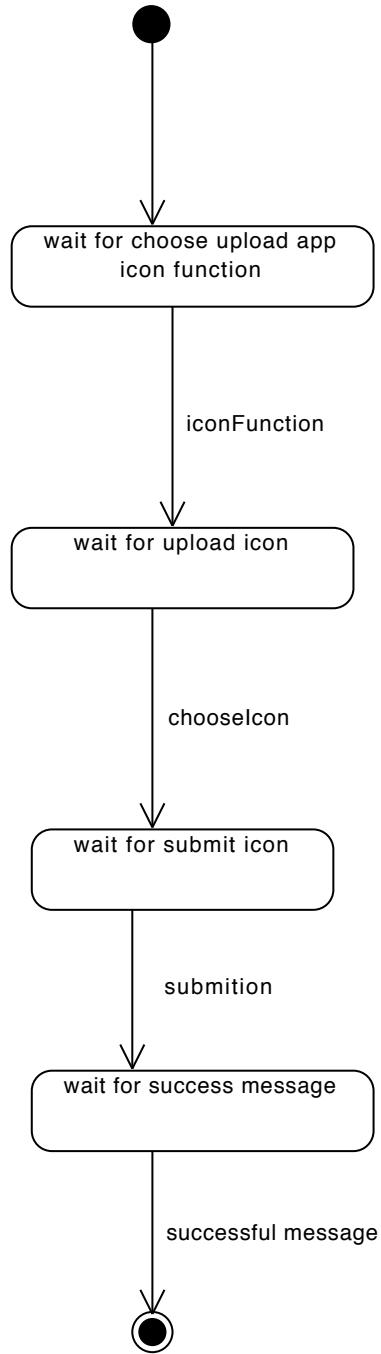
UC002: Send push notification message to customer

Visual Paradigm for UML Enterprise Edition(IVE_ICT)

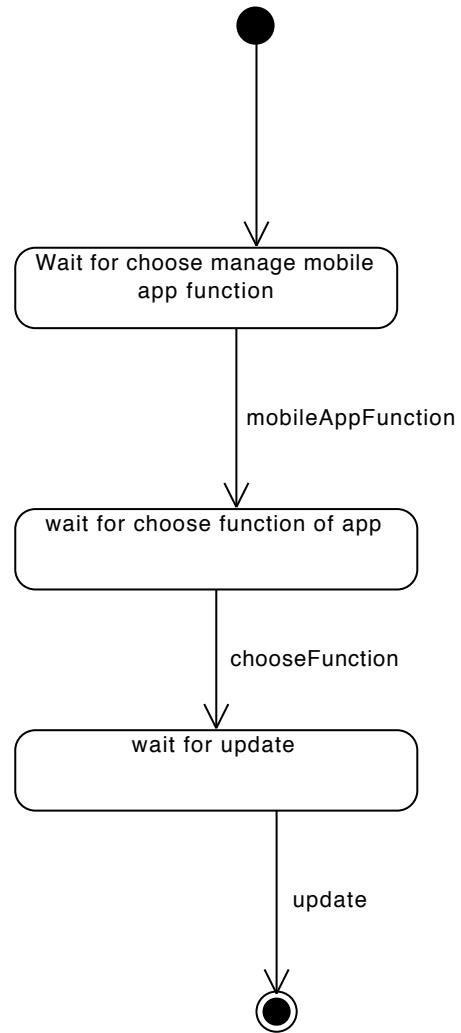




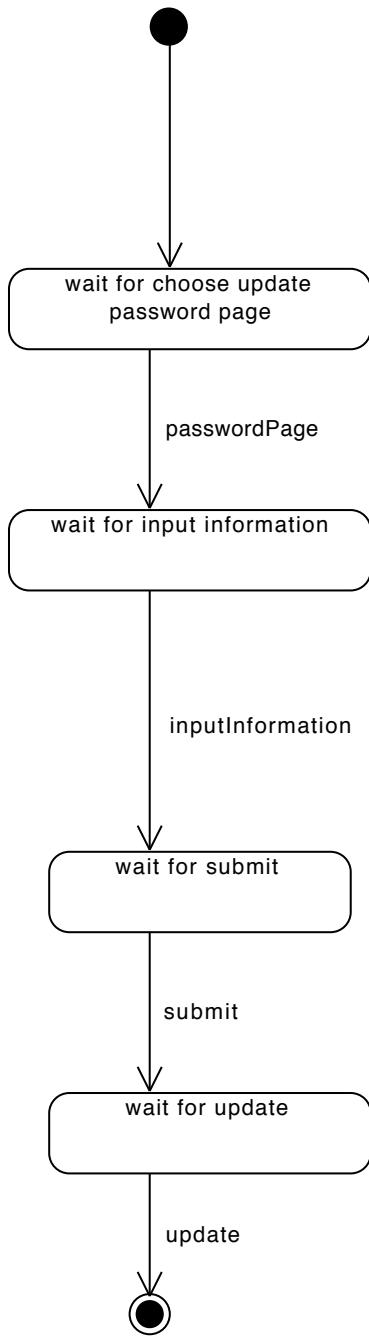
UC004: Upload Mobile App Icon



UC005: Manage Mobile App Function

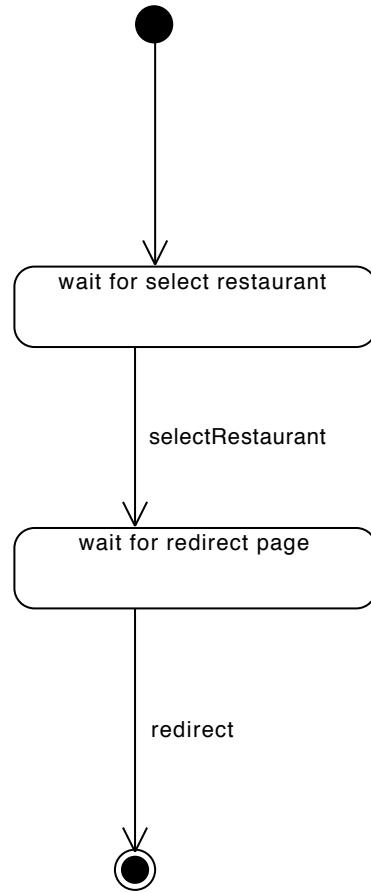


UC006: Change Account Password



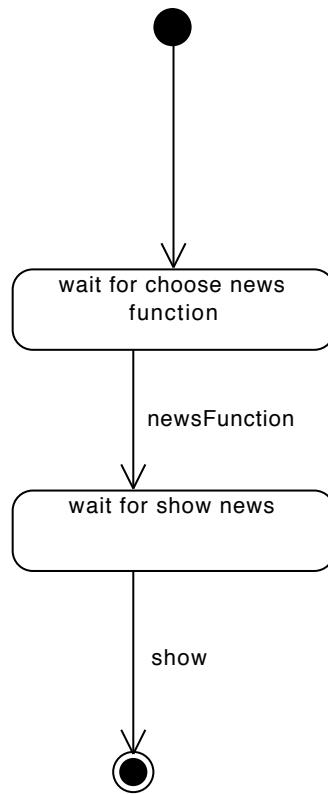
UC007: Restaurant Staff

Visual Paradigm for UML Enterprise Edition(IVE_ICT)



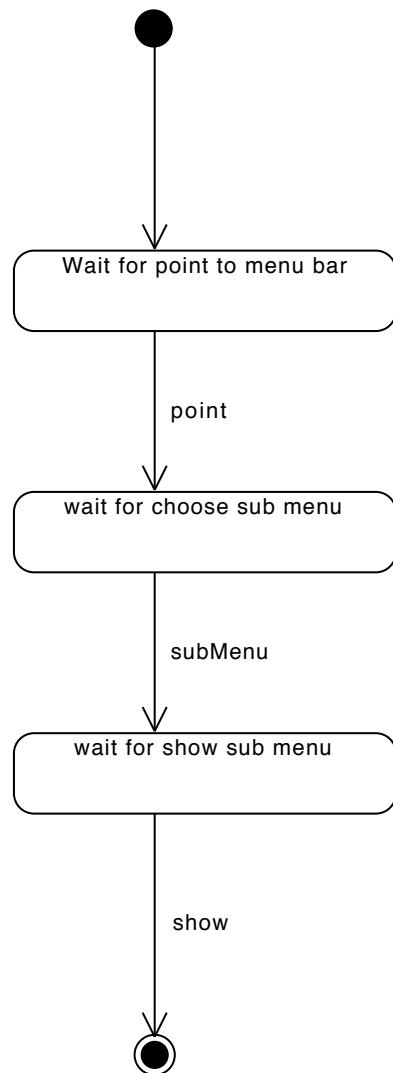
UC008: View Company News

Visual Paradigm for UML Enterprise Edition(IVE_ICT)



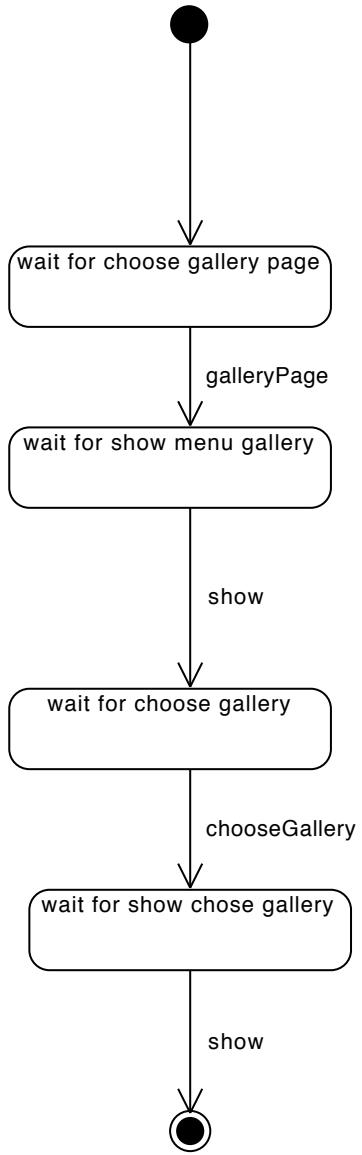
UC009: View Electric Menu

Visual Paradigm for UML Enterprise Edition(IVE_ICT)



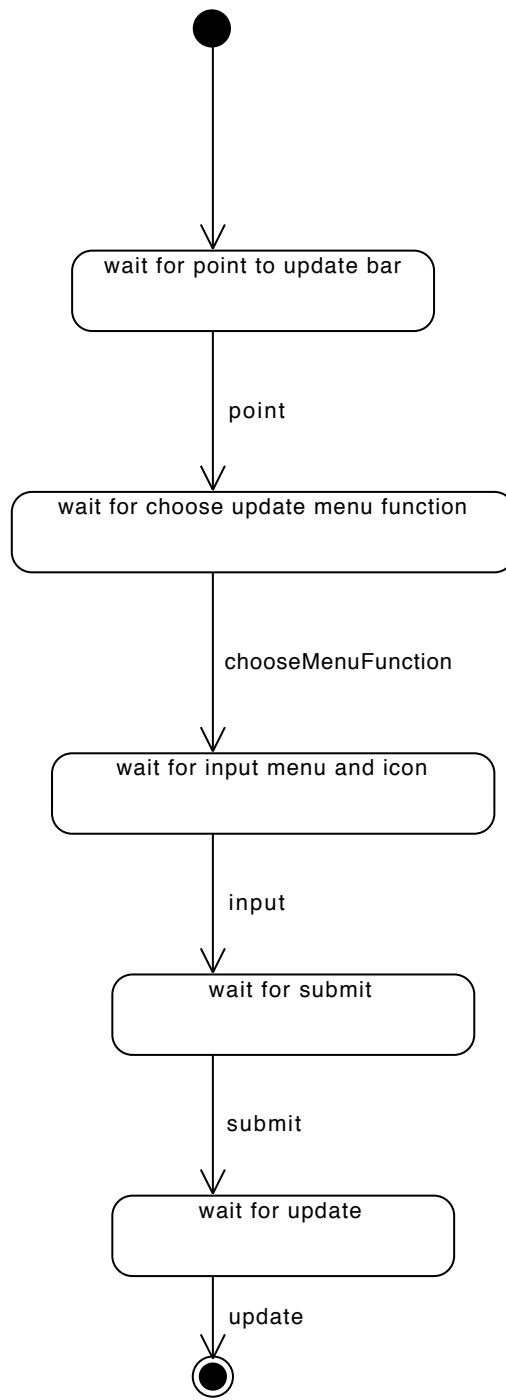
UC010: View Food Gallery

Visual Paradigm for UML Enterprise Edition(IVE_ICT)

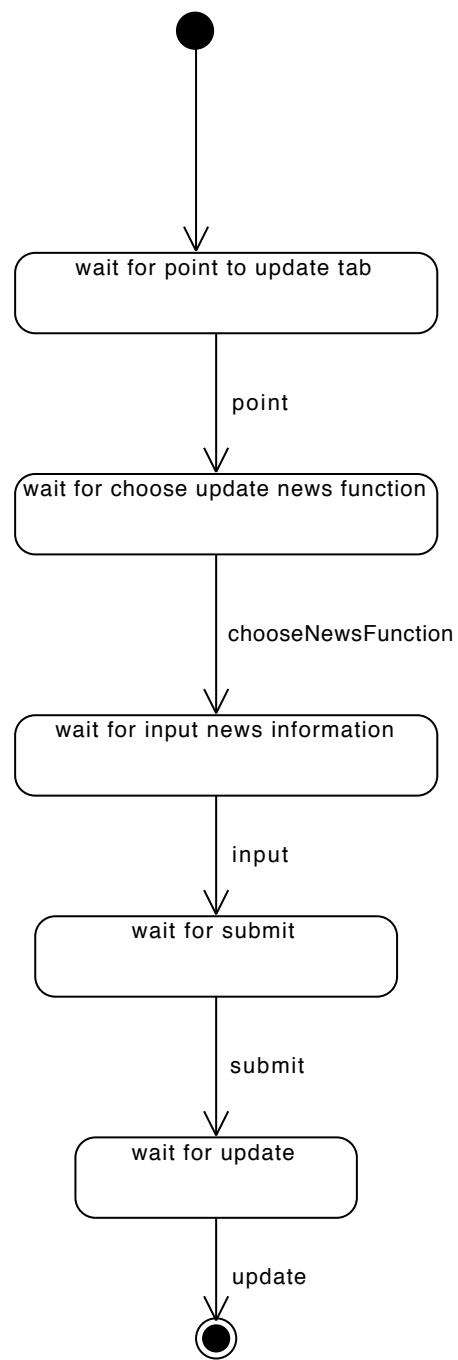


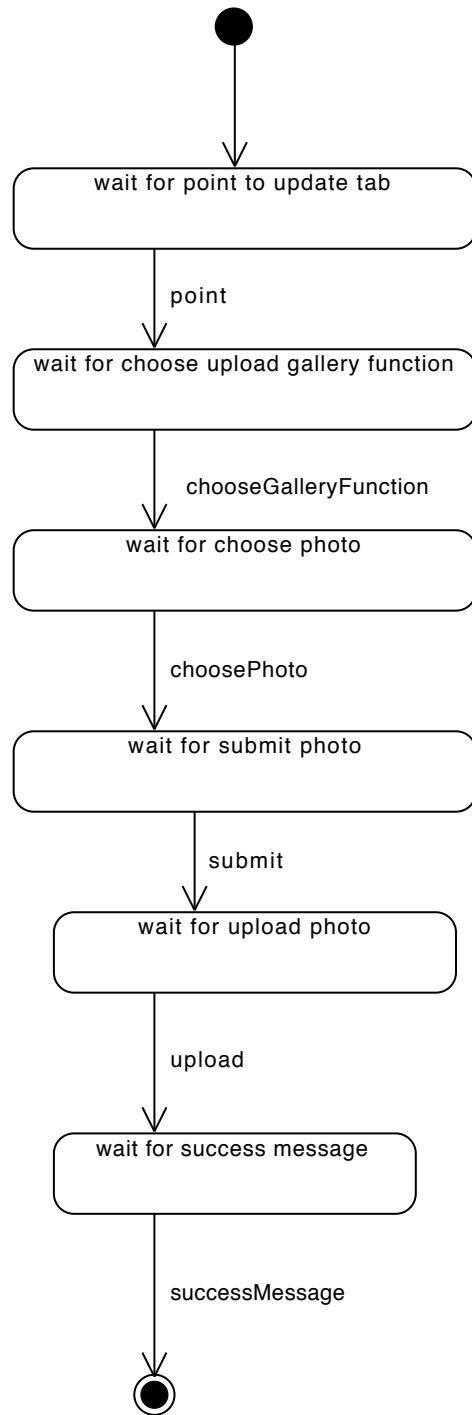
UC011: Update Menu

Visual Paradigm for UML Enterprise Edition(IVE_ICT)



UC012: Update News





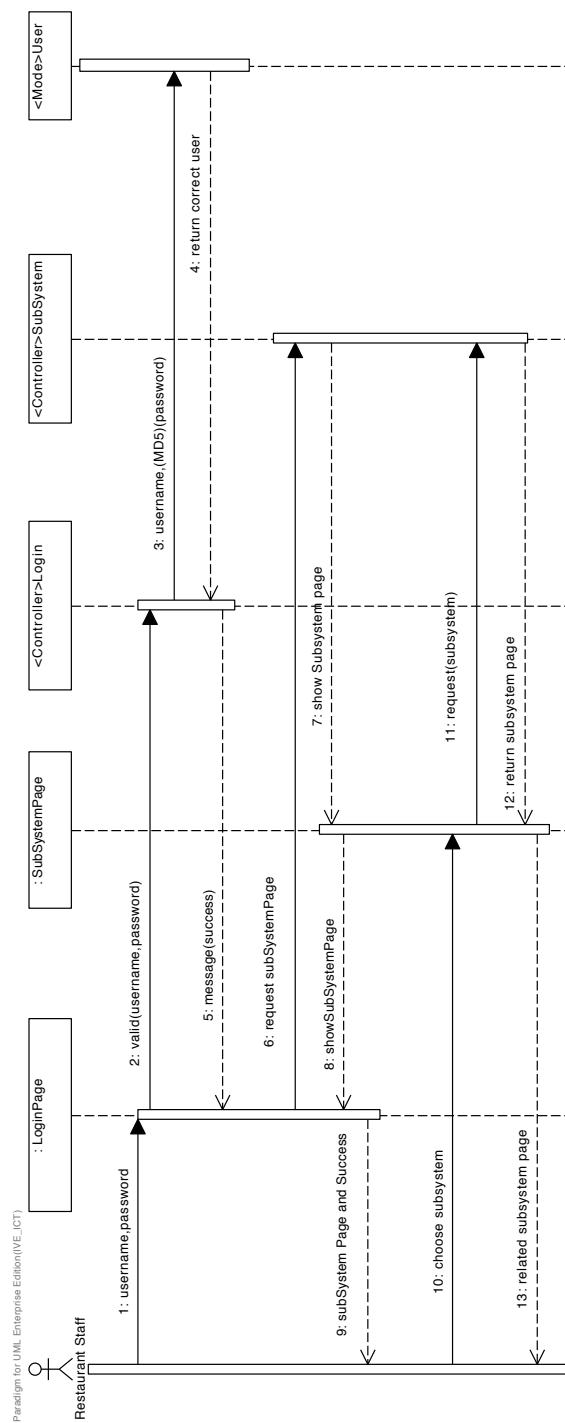
Sequence Diagram

Sequence Diagram of iPhone Application

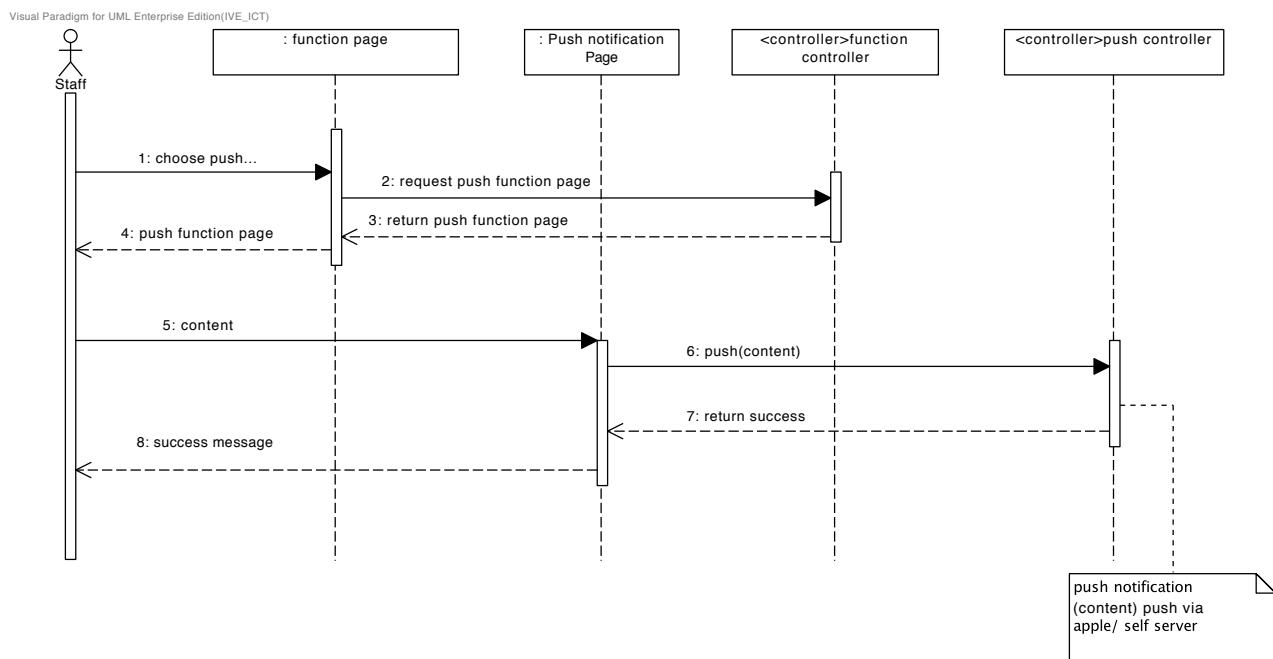
Sequence Diagram of Android Application

UC001: Login the System

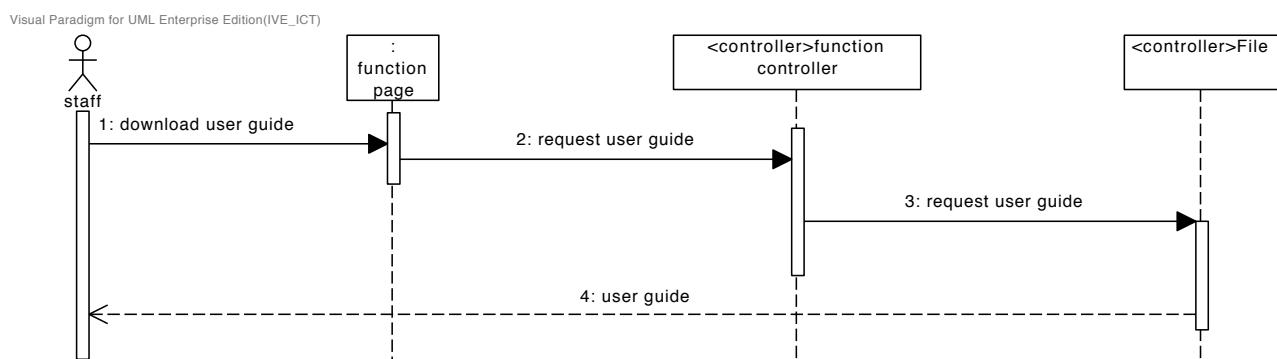
Sequence Diagram of Web Service



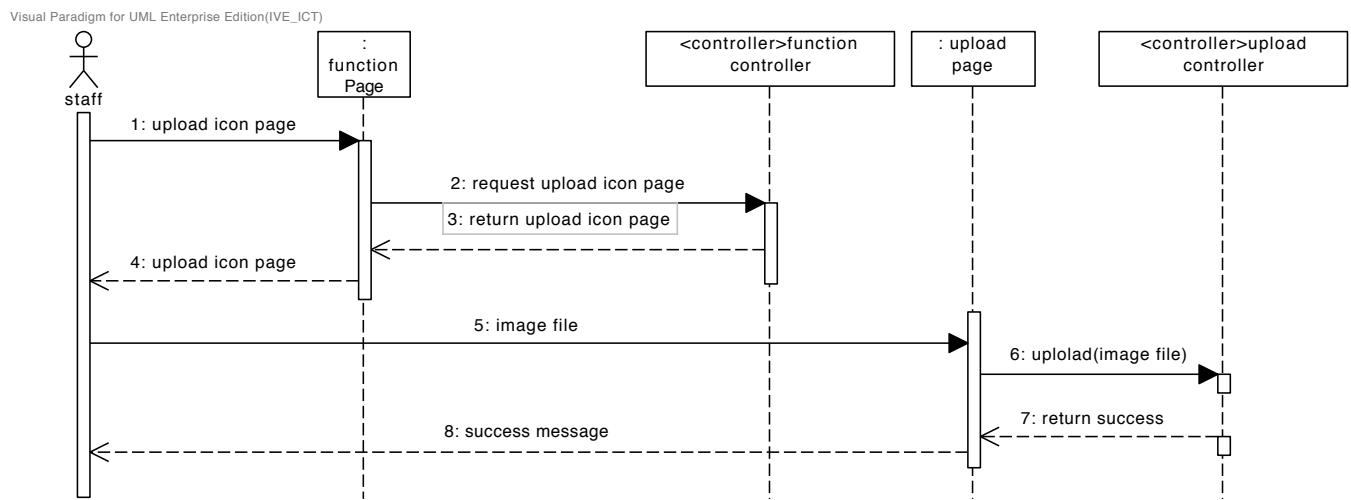
UC002: Send push notification message to customer



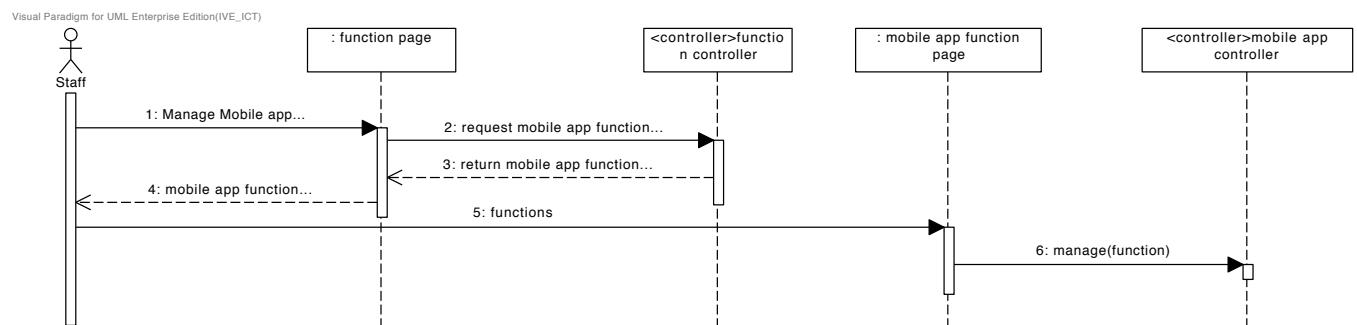
UC003: Download User Guide



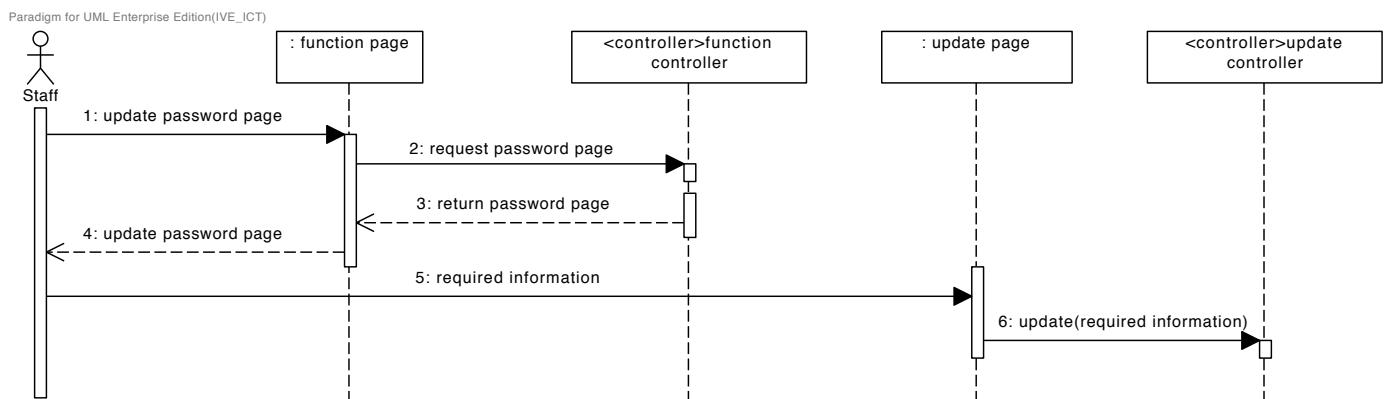
UC004: Upload Mobile App Icon



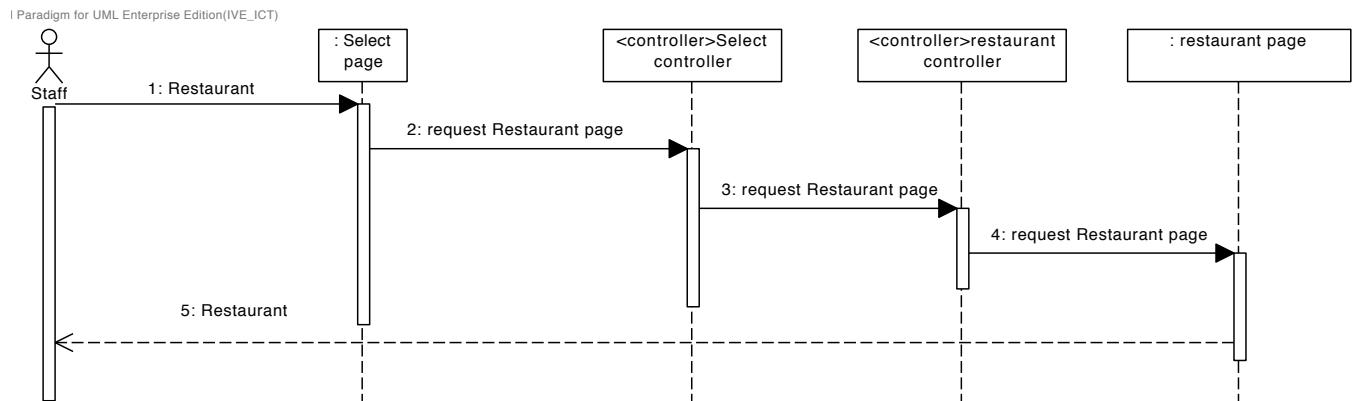
UC005: Manage Mobile App Function



UC006: Change Account Password

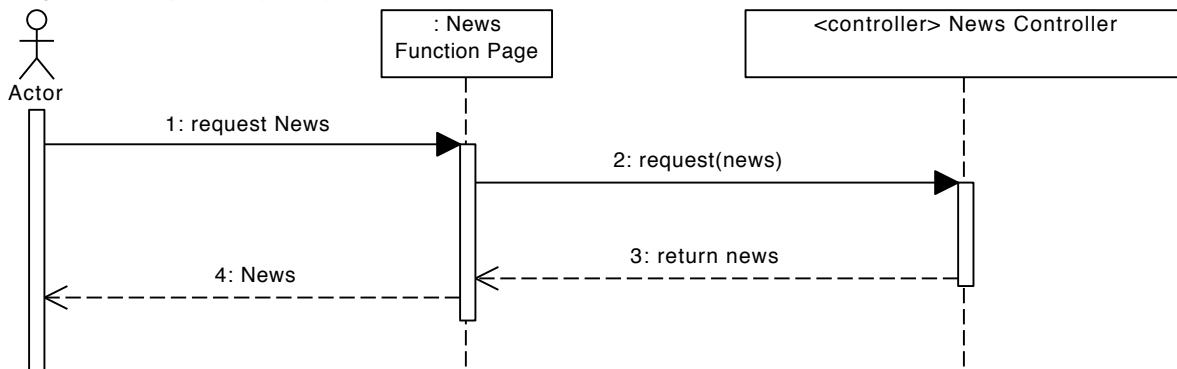


UC007: Select Restaurant



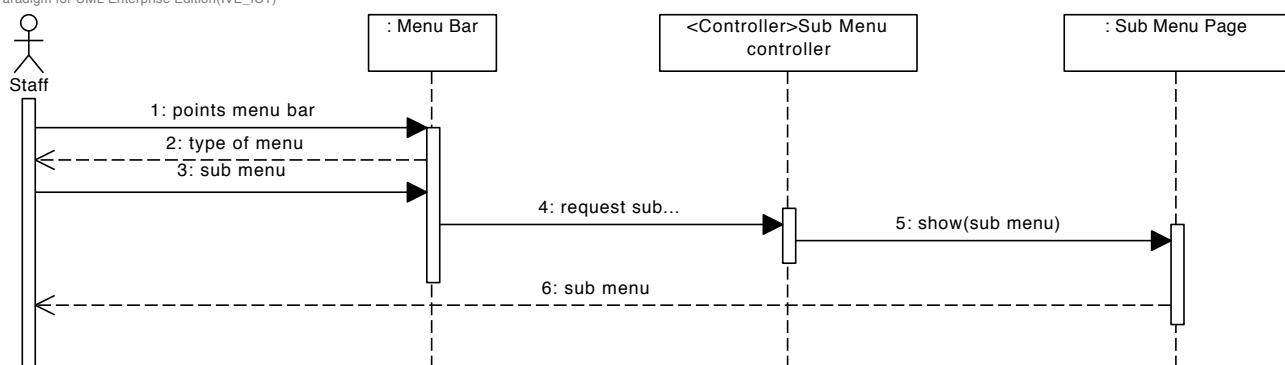
UC008: View Company News

Visual Paradigm for UML Enterprise Edition(IVE_ICT)

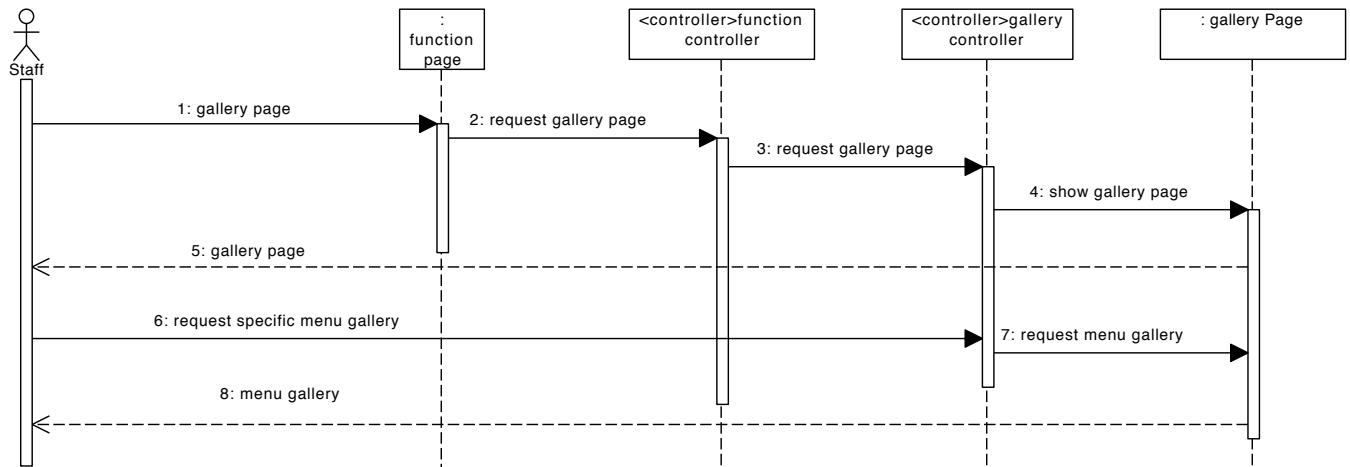


UC009: View Electric Menu

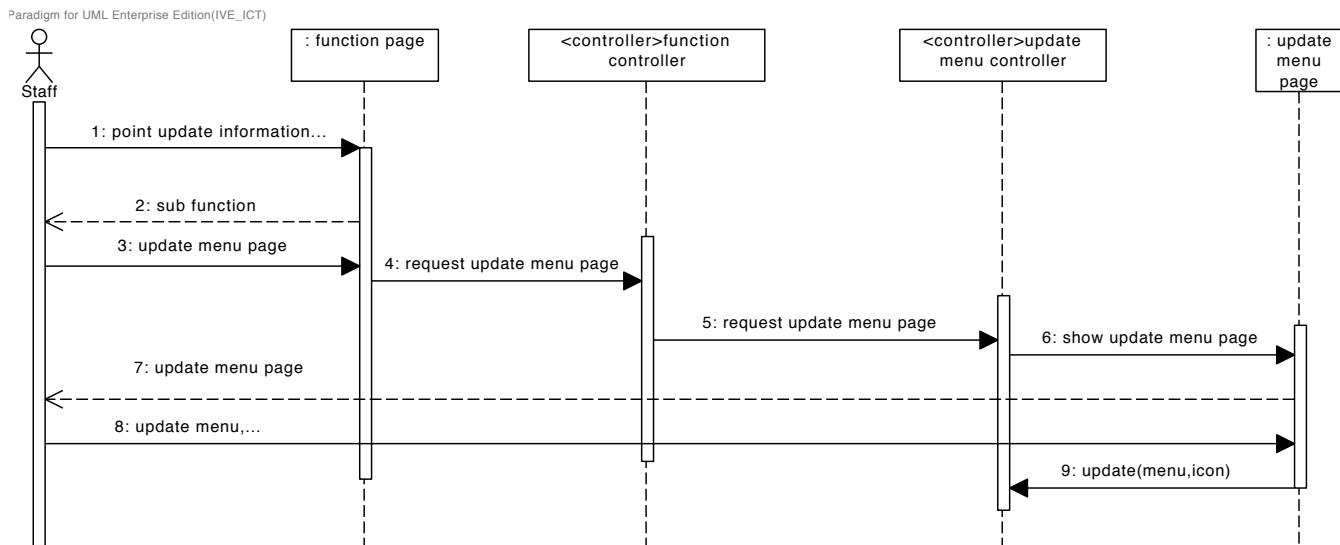
Paradigm for UML Enterprise Edition(IVE_ICT)



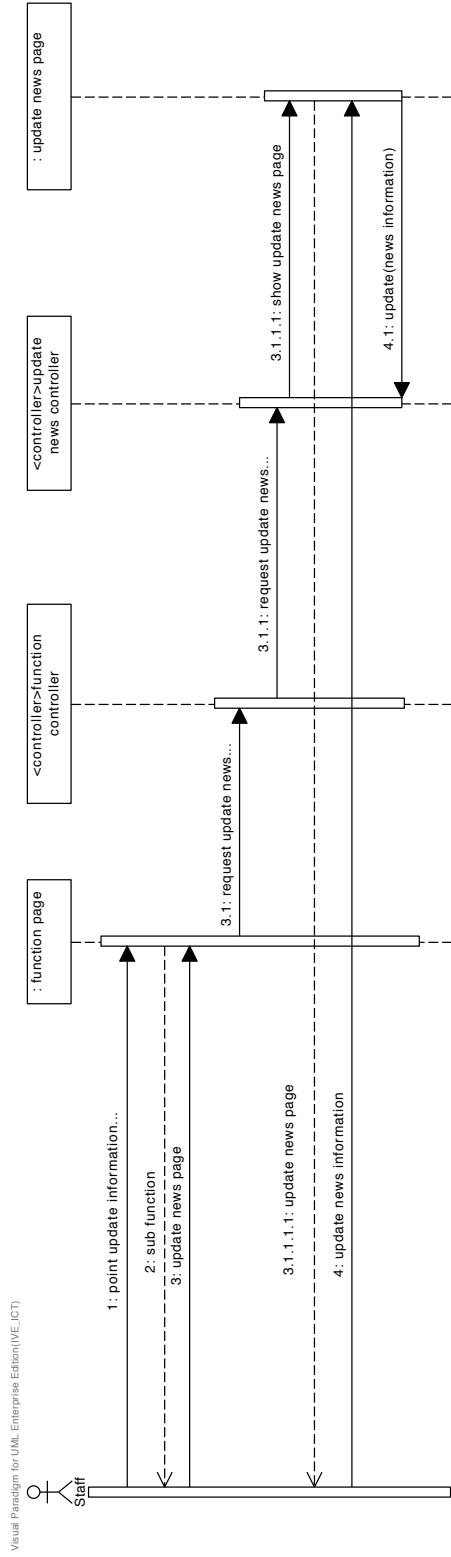
UC010: View Food Gallery



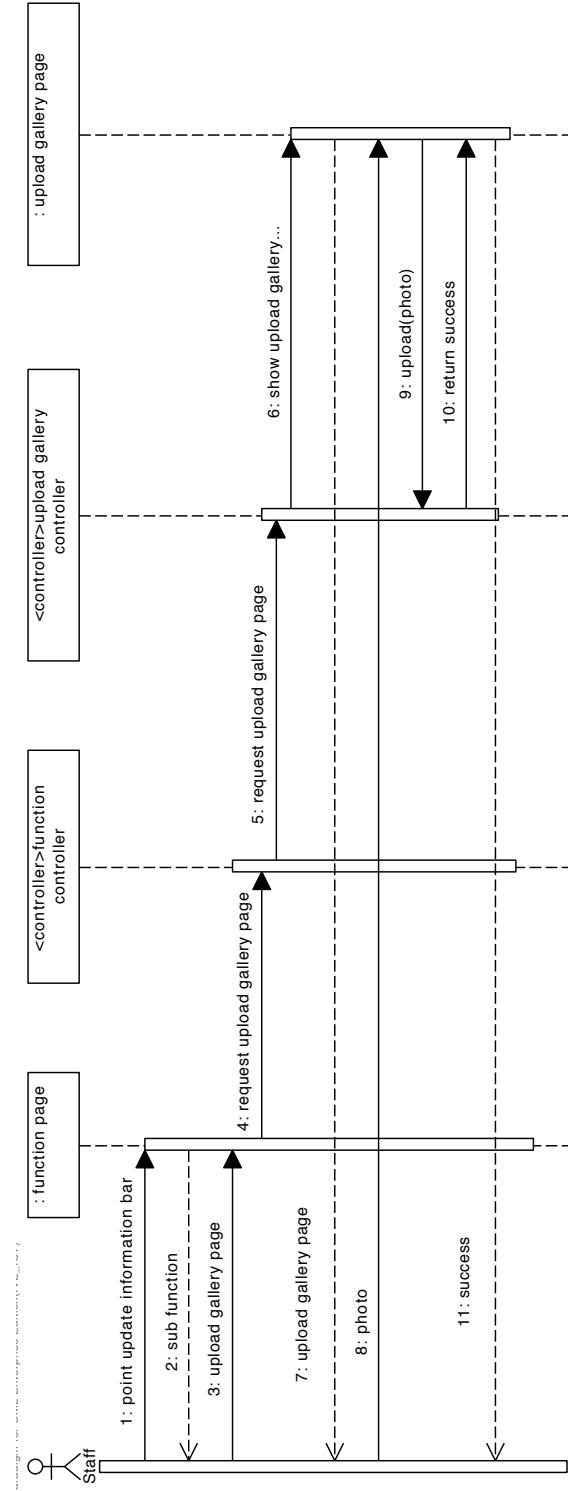
UC011: Update Menu



UC012: Update News



UC013: Upload New Food



User Interface Design

Usability is the ease of use and learnability of a human-made object. The object of use can be a software application, website, book, tool, machine, process, or anything a human interacts with. A usability study may be conducted as a primary job function by a usability analyst or as a secondary job function by designers, technical writers, marketing personnel, and others. It is widely used in consumer electronics, communication, and knowledge transfer objects and mechanical objects such as a door handle or a hammer.

Usability includes methods of measuring usability, such as Needs analysis and the study of the principles behind an object's perceived efficiency or elegance. In human-computer interaction and computer science, usability studies the elegance and clarity with which the interaction with a computer program or a web site (web usability) is designed. Usability differs from user satisfaction insofar as the former also embraces usefulness (see Computer user satisfaction).

The key principle for maximizing usability is to employ iterative design, which progressively refines the design through evaluation from the early stages of design. The evaluation steps enable the designers and developers to incorporate user and client feedback until the system reaches an acceptable level of usability.

iPhone/ Android Application UI Design

We assumed that the user interface design in iPhone and Android mobile are the same and providing the same functions, there should be only the coding in differences and this would not affecting any user design problem or design not the same.



The icon TestFYP is our application in mobile. We will change the name later. This icon design is attractive if the company would like to put the app in the App Store.



Application Main Menu

All function is represented by an image and short describe. It can make the user interface more interesting and clearer because user need not to find out any sub menu and they just need to touch one icon to use any function they want.



Main Interface

Each branch shop under the Enoteca group will have different functions and containing different data, as the diagram show, it use a 3×3 grid to fill in the icon, where there are more than 9 icons, it will swipe to another page. It let user to simply slide on the mobile to go to the next page. Each icon will contain the function name under it. This design makes the application livelier and more comfortable for user.



News Interface

The news menu is designed in a table view; each row displays the heading of the news. The user can touch on the heading to view the details of the news that related to the heading. Each shop contains different news. This is a simple and clear UI design.



Location Interface

User can View the Google map for the location, the blue point is the user GPS or AGPS location now. The red point is the shop. By the features provided by Google Map, the user can touch the tab bar on the top of the view to choose by private car, public transport or walk. The bold blue line in the second image is the routing path for the user.



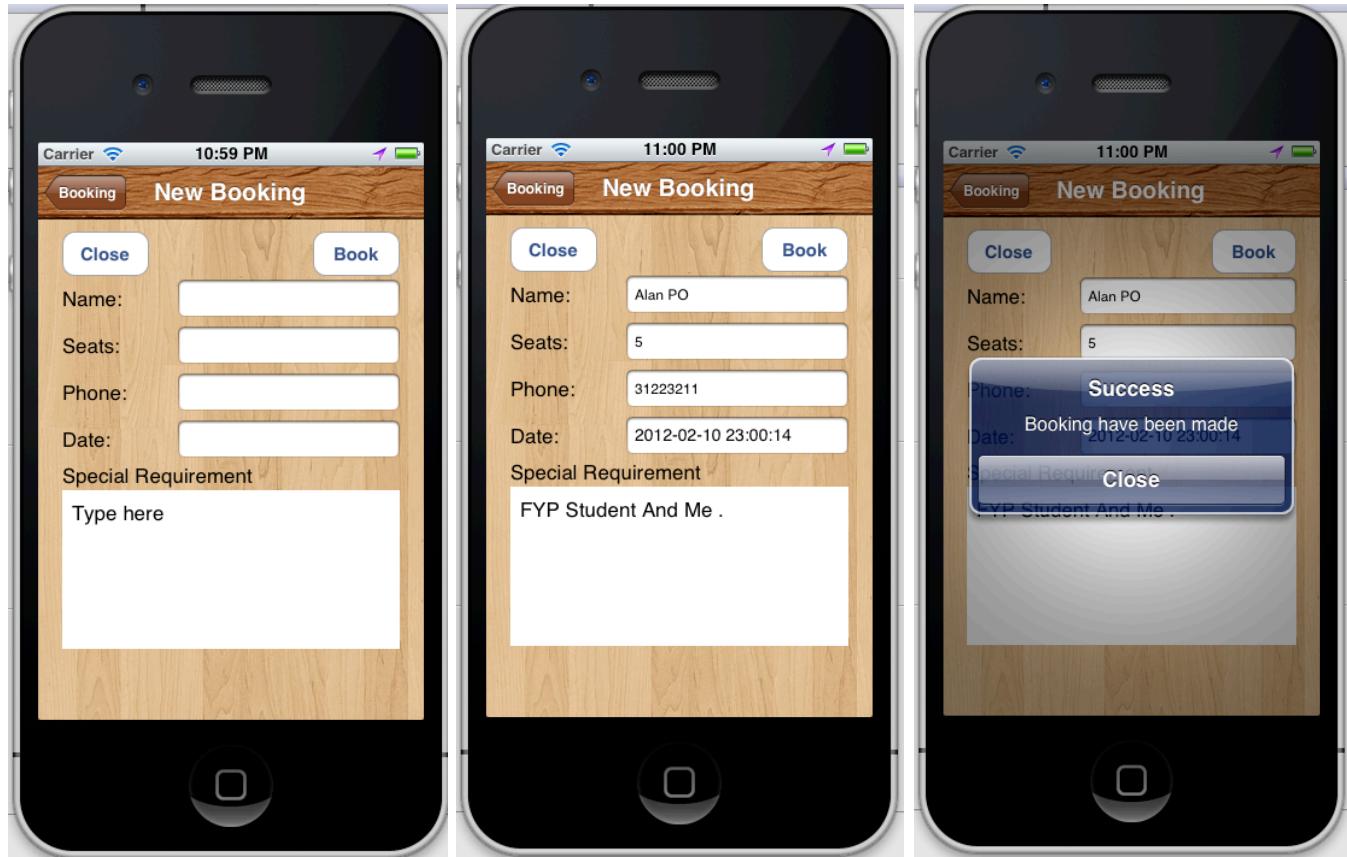
Menu Interface

The menu is designed in a table view; each row displays the preview image of the food and the name of the food. Brief information was following the food. The user can touch on the name of the food to view the details of the food that related to the name. Each shop contains different news. This is a simple and clear UI design.



Facebook Sharing Interface

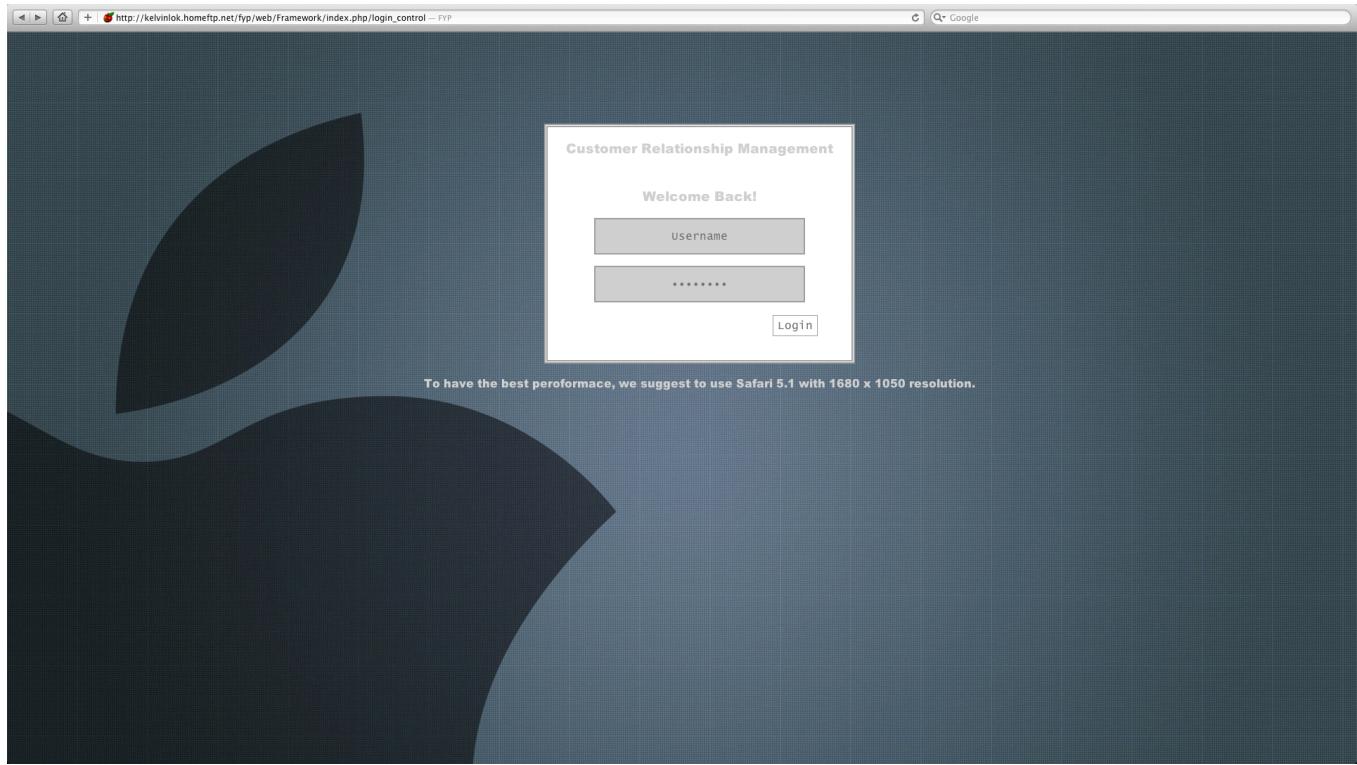
This is the user post the food to Facebook wall interface. The interface let the user to choose to share the food information by different social networks or copy to another application. The web-view can let the user to post the information to her own Facebook account.



Booking Interface

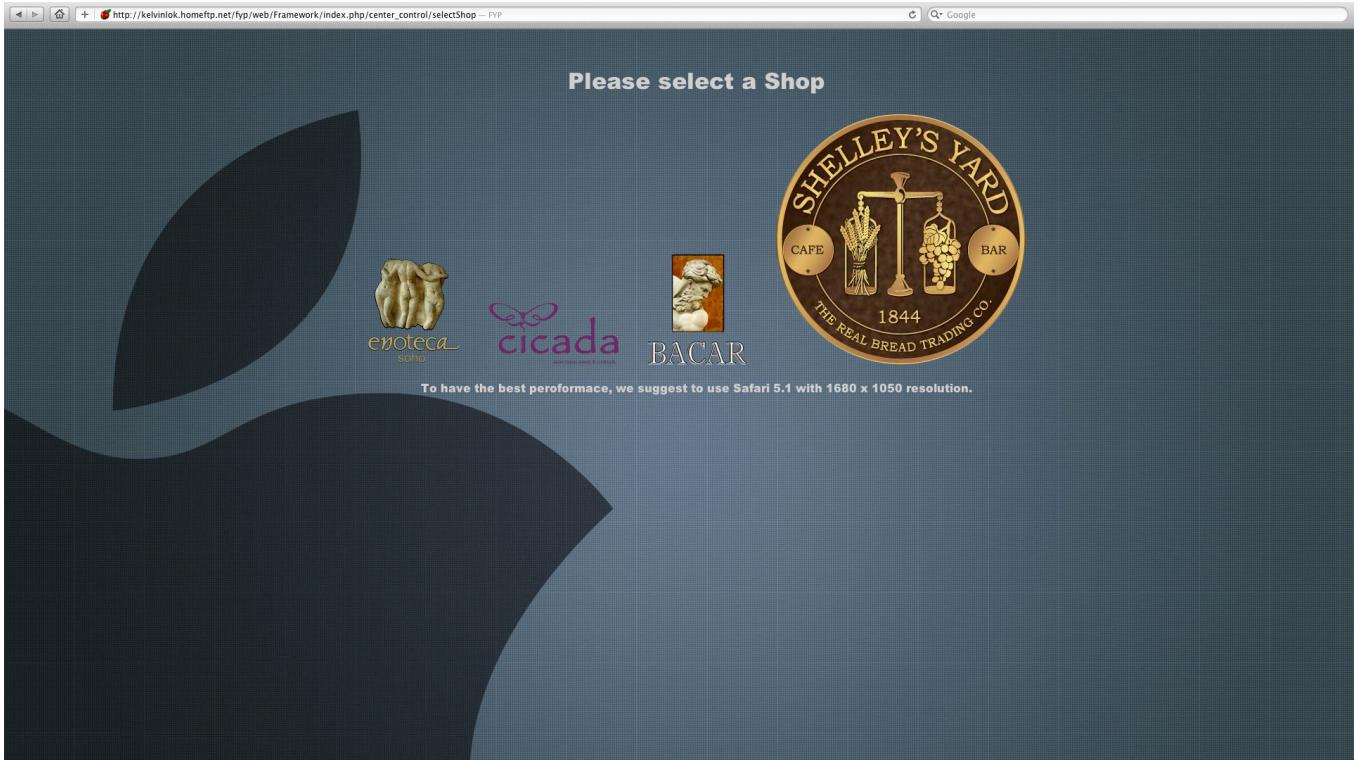
The booking interface contains a set of information that the user needs to provide to us. All UI elements displayed with a good alignment format and the related data is group together. The data can be input by the prompted data picker. The button "Book" is used for the user to submit their book to us.

Web Service User Interface Design



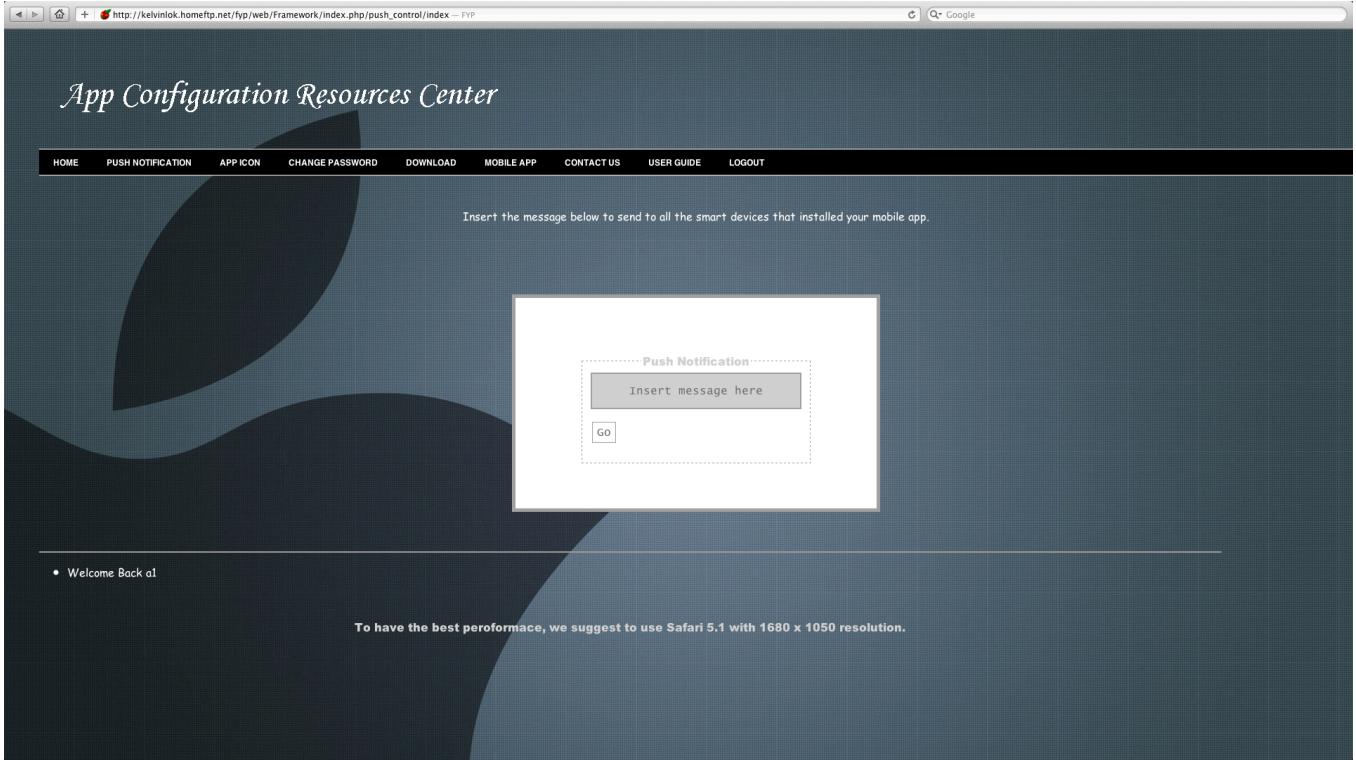
First Index Interface

This is the first interface of the website with a simple text box with the title of the system, The username field related to the user name of the user, and following by a password field.



Sub system Interface

This is the sub system interface of the website with the logo of the branch shop, the user just need to click on the logo to manage the information.



Main Interface Design

We design the main interface in a simple way, the tab contains the word of the function, it will direct the user to the related function and the middle is the function that provided to the user.

Critical Evaluation

Problems and Difficulties encountered

Server side

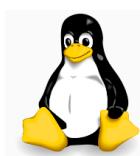
The server job is helping the communication between clients, so the server should provide a stable and reliable network connection; because the app needed to be update the information from the server database and the web service.

We build up the server in one of our home, but we do not 100 % sure that the network is stable because some a couple of problem from the ISP. Since we use PCCW in Hong Kong as our ISP, the upload speed is slow. Because we register a domain name in free from the <http://dyndns.org>. The domain name transfer is done in the router. The ISP IP address is dynamic, that means over a time the real IP will change. But the domain name transfer can only be updated hourly, that mean if the real IP address is change but the domain name cannot transfer at the same time, the client cannot connect to the server.

In this project, we used a push notification function to send some message to the client. The difficulties came here because we need to support multi-platform mobile devices. Since, Apple provided a push notification server itself in USA. We only need to send the application ID, device ID and message to Apple. The server will later send the push notification to the related devices. But for android, Google do not provided this service, we need to install the MQTT server in the Linux server to provide this function.

Therefore, delay of the push message may be happened because they do not use the same server.

Besides, to adopt the MVC concept in the website, we choose to use Code Igniter PHP Framework to complete our project. This is a chance for us to learn new technology but also a difficult to us because the PHP Framework code is a Object Oriented PHP code and we learnt the PHP and javascript before but not in object oriented base. Therefore, the coding standard cannot be used in the same ways. We need to follow some tutorial online to help use to learn and use it probably. Therefore, the website development speed need to be speed up as well.



[iPhone development](#)

iPhone is a quite new product and common nowadays. But the version of iOS in iPhone updated frequently, some of the features added. We can only use MAC computer to develop the iPhone application because the tools Xcode to develop iPhone only support the Mac system. Also, this new technology we need to start from zero because we have not learned the coding before. Also, we need to register the Apple developer in order to output our application and install it on iPhone.

The technic for communication between iPhone and the Linux server is the most difficult part, we found out a lot tutorials and user guide line from the Internet, but only with basic tools, we need to further think about the logic and method to keep the communication stable and in a security way.

Besides, the push notification in iPhone sometimes does not support Chinese character because the character memory is a bit different from the English letter even when we use Unicode. Apple develops the push server; we do not know that the problem is from our server or from the Apple push server.



Android mobile development

Android is an operating system in mobile that developed commonly in these few years. But up to now, more developer will choose to develop iPhone app more than Android, therefore, it is difficult to find out tutorials, example or user guide to learn from the Internet when company with iPhone.

Besides, API for Android is much fewer than iPhone. Some of the features are difficult to complete like the location routing, automatically updates from the database. The API level is one of the important thing that we need to control, some mobile use a lower Android system that we do not know, their API level maybe lower than our application. So, we need to use a more common API Level to complete our application.



Delays / Changes in Project Schedule

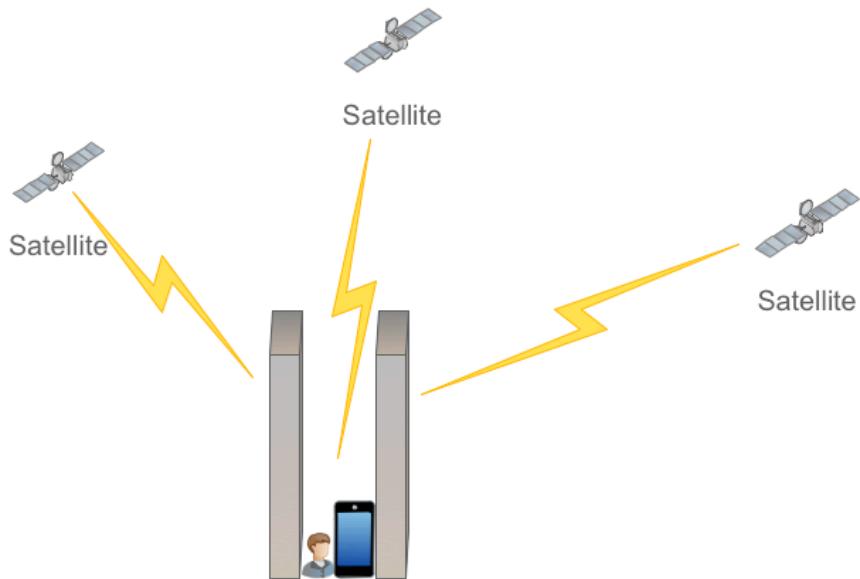
When we set up the server, sometimes we need to face some unknown problem or warning in Linux. We need to fix the problems in order to install the service to provide the stable server.

Some delays were happened and out of our expectation, the Web service is delay in developing process because the PHP Framework we need to take a couple of time to learn it and use.

Limitations of the proposed system

Unable to locate the shop

Some place cannot detect the GPS signal or inaccurate positioning, indoor and many high building around the user can affect the GPS signal and return nothing. GPS needs at least 3 GPS Satellite to locate the location. The following diagram shows the example.



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.

Detailed Project Plan

Hardware Facility

- Server

Motherboard : Intel Socket LGA 775 Motherboard

CPU : Intel Pentium IV 3.0Ghz Dual-Core

RAM : 2GB

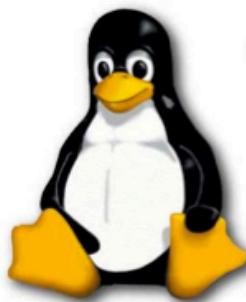
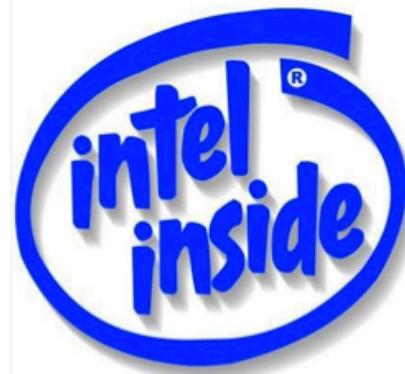
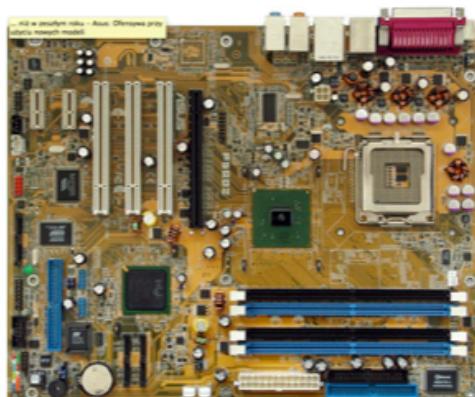
DVD-ROM : DVD-Read/Write, CR-Read/Write

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



- 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



References

<http://codeigniter.com/>

<http://developer.android.com/index.html>

<http://developer.apple.com/>

<http://github.com/>

<http://linux.vbird.org/>

<http://www.centos.org/>

<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

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

<http://www.digitalwall.com/scripts/display.asp?UID=270>

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

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

<http://adeem.me/blog/>

<http://maps.google.com/>

<http://code.google.com/intl/zh-TW/apis/maps/>

<http://api.jquery.com/jQuery/>

<http://findbook.tw/book/9789575279059/basic>

[http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))

<http://plugins.jquery.com/>

<http://www.web toolkit.info/ajax-file-upload.html>

<http://developers.facebook.com/>

Appendices

hksd.org/portal/en/

Google+ 616 Brand Dropbox Facebook Forum Google Maps HK Observatory Yahoo! HK University Home IT IVE JTV News

ITAA 商業軟件開發大賽

促進青少年對資訊科技在商業應用的認識和興趣，拉近商界與學界的關係

HOME ABOUT US Expand ABOUT US NEWS Expand NEWS RULES & GUIDELINES PROJECT Expand PROJECT ENQUIRY

LATEST NEWS

2012/05/25	Prize Award Ceremony
2012/05/25	Final Presentation
2012/02/08	Interim Sharing
2011/06/29	Orientation Day
2011/06/24	Project Day of Hong Kong Institute of Vocational Education (IVE)

>> More news

Topic
Customer Relationship Management (CRM)

- Participants can choose a CRM topic from different industries
- Programming tools is not limited
- The participating product should either (1) help boost sales & business (2) maintain customers (3) perform business analysis

Phase I
Participants should submit a 2,000 words report

Phase II
Participants should deliver a presentation with live demo

Key Important Dates

30 Sep 2011	Application Deadline
10 Feb 2012	Project Report Submission
17 Feb 2012	Interim Sharing
07-18 May 2012	First Screening: Presentation & Demo
25 May 2012	Final Presentation & Prize Award Ceremony

202

Final Year Project Interim Report (2011/2012)

The screenshot shows the official CodeIgniter website. At the top, there's a navigation bar with links to Google+, 616, Brand, Dropbox, Facebook, Forum, Google Maps, HK Observatory, Yahoo! HK, University, Home, IT, IVE, JTV, and News. Below the navigation is a banner featuring a flame icon and the text: "CodeIgniter is an Open Source Web Application Framework that helps you write incredible PHP programs". To the right of the banner are links for "Download CodeIgniter Version 2.1.0", "Read the User Guide", "View the Video Tutorials", and "Talk About It in the Forums". A sidebar on the right is titled "CODEIGNITER NEWS" and lists several news items with titles like "Pancake App", "ZoomShift", "CPA Site Solutions touts CI", "PHP Framework Usage Survey", and "View Past News". The main content area below the banner includes a section titled "Welcome to CodeIgniter!" and a list of reasons why CodeIgniter is right for you, such as "You want a framework with a small footprint." and "You need clear, thorough documentation."

The screenshot shows the GitHub homepage. At the top, there's a navigation bar with links to Google+, 616, Brand, Dropbox, Facebook, Forum, Google Maps, HK Observatory, Yahoo! HK, University, Home, IT, IVE, JTV, News, and Piano. Below the navigation is a banner with the text "1,311,669 people hosting over 3,921,382 git repositories". There are social sharing icons for Facebook, Twitter, Microsoft, VMware, Redhat, LinkedIn, and Yahoo!. Two boxes are shown: one for "git /'git/" describing it as a fast, efficient version control system, and another for "git·hub /'git hub/" describing it as a platform for collaborative development. A blue button at the bottom left says "Plans, Pricing and Signup Unlimited public repositories are free!". Below the button, text reads "Free public repositories, collaborator management, issue tracking, wikis, downloads, code review, graphs and much more...". At the bottom, there are four sections: "Team management", "Code review", "Reliable code hosting", and "Open source collaboration", each with a "More about" link.

The screenshot shows the Facebook Developers homepage. At the top, there's a banner with the title "Hack the Graph" and a subtext: "Build with the Open Graph. Integrate deeply into the Facebook experience. Grow lasting connections with your users." Below the banner are three sections: "Build for Websites" (with an icon of a laptop), "Build for Mobile" (with an icon of a smartphone), and "Build Apps on Facebook" (with an icon of a cube). Under "Build for Websites", there's a link to "Platform Live Status: Facebook Platform is Healthy". The "Latest Updates" section lists several items, including "How-To: Migrate your App Profile Page" and "How-To: Subscribing to data changes using the Real-time Updates API". The "HTML5 Resource Center" section features a large "HTML5" logo and a brief description about its capabilities. The "Showcase" section displays logos for various companies like Spotify, Pinterest, and Airbnb. At the bottom, there are social sharing buttons and a copyright notice: "Facebook © 2012 - English (US)".

The screenshot shows the jQuery API documentation website. The header includes the jQuery logo and navigation links for "jQuery", "UI", "Mobile", "Meetups", "Forum", "Blog", "About", and "Donate". The main content area has a dark background. On the left, there's a sidebar with "jQuery API" and "Browse the jQuery API" sections. The main content area shows the "jQuery()" page, which includes a table of contents, a "Description" section, and two code snippets for "jQuery(selector [, context])" and "jQuery(element)". The "jQuery(selector [, context])" snippet includes a "Returns: jQuery" note and a "Description" explaining it accepts a CSS selector. The "jQuery(element)" snippet includes a "version added: 1.0" note.

204

Final Year Project Interim Report (2011/2012)

END OF REPORT

205

Final Year Project Interim Report (2011/2012)