Table of Contents

Abstract **6**

Acknowledgement **9**

List of Tables and Figures **10**

Introduction **14**

Document Structure **14**

The Description Of This Project **16**

Background **18**

Problems Description **19**

The requirements **21**

Scope Of The System **21**

The way of client to archive server's address 22

Mobile app automatically connect to the server and receiving latest information from the server 25

The way of client and server to communication 26

Consistency of data structure between server slide and client slide 26

The structure of the mobile application 27

The functions of the mobile application 27

The functions of the website and web service 39

Website for customer 58

How to handle if the client can't connect the server in the specific area 62

How to support multi-platform client 62

How to support multi restaurant 62

How to make client easily to use our mobile application 62

How the way to implement the system in the future 63

Analysis report for customers' habits 64

Description of hardware requirements 64

Non-Functional Requirement **65**

A server administrator must be added to the system 65

User-friendly GUI for the customer 65

Log keeping in database and server 65
Security Design is need in the server side 65
Automatically logout when the user stand-by for more than 15 minutes (website) ... 67
Database and system should be backup frequently 67
Use high effectively server 67
Use the stable software tools in the server 67

Constraints 68
Network stability problem 68
Limitation of iPhone API 68
Limitation of Android API 69

Documentation for problem analysis 71

UML Modeling 71
Entity-Relationship Diagram 72
Data Dictionary 74
Class Diagram (iPhone App) 88
Class Diagram (Android App) 92
Class Diagram (Website) 105
Use Case Diagram (Mobile) 106
Actor Description (Mobile) 107
Initial Use Case Description (Mobile) 108
Refined Use Case Description (Mobile) 111
Use Case Diagram (Website) 122
Actor Description (Website) 123
Initial Use Case Description (Website) 124
Refined Use Case Description (Website) 128
State Transition Diagram (iPhone) 139
State Transition Diagram (Android) 154
State Transition Diagram (Website) 165
Activity Diagram (iPhone) 177
Activity Diagram (Android) 189
Activity Diagram (Website) 199
Sequence Diagram (iPhone) 207
Sequence Diagram (Android) 223

Sequence Diagram (Website) 234
Object Collaboration Diagram (iPhone) 242
Object Collaboration Diagram (Android) 258
Object Collaboration Diagram (Website) 269

Documentation For Detailed Design 279
UI Design 279
Mobile App user interface design 280
Website User Interface Design 290

Implementation 304
iPhone Test Plan and Test Result 304
Login 304
Register 305
View Coupon Log 307
View My Order 308
..... 308
View Shop 309
..... 309
View News 310
..... 310
View Menu 311
..... 312
..... 312
View Map 314
..... 314
View Booking 315
..... 315
New Booking 316
Feedback 317
Coupon 318
Lucky Draw 320
Test Plan and Test Result (Android) 323
Login 323

Register 324
View My Order 325
View Coupon used 327
View news 328
View Menu 329
View Map 331
View Book 332
Booking 333
Feedback 334
Coupon 335
Lucky Draw 337
Test Plan and Test Result (Website) 339
View Restaurant E-news 339
New restaurant news 340
View E-Menu 341
New restaurant food 342
Search menu food 343
Send push notification 344
Insert new feedback 345
Modify mobile app icon 347
Modify mobile app function 349
Data analysis 350
Critical Evaluation 353
Problems and Difficulties encountered 353
Server side 353
iPhone development 354
Android mobile development 355
Delays / Changes in Project Schedule 355
Limitations of the proposed system 356
Unable to locate the shop 356
Network stability problem 356
References 357
Appendices 358

A1. HKITAA Completion 358
A2. PHP Framework 359
A3. Facebook Developer 359
A4. JQuery 360
A5. Project Plan 361
A6. Job Division 364
A7. Logsheet 365

Program Listing 435

ABSTRACT

In our final year, we need to develop a Final Year Project, we grouped up 4 students together last year and chose to work together to complete the project. In the beginning of the processes of this beginning, we discussed the topic, the aims, and the suggested system. Finally, we decided to develop a Customer Relationship Management (CRM) System. At that time, the Hong Kong IT Accountants Association (ITAA) and Software Factory Ltd. was organizing a competition that is use our final year project to join the Hong Kong Business Software Development Competition (HKBSDC).

We first select a type of the enterprise in Hong Kong and then co-operation with one of them to develop our system based on their requirements and their needs to help improving their business. The competition organized the company mentor for our group to follow, and the result is to co-operate with one of the catering company in Hong Kong that is “Enoteca Group”.

Our project is developing both server-sided and client-side systems. For the server-side, which is a web service for the restaurant staff. The client-side is for the customer to use the mobile devices like iPhone and Android. 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.

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 staff can edit or create their own mobile app and they do not need to have any programming skills on smart devices basically.

Now, there is no any standard Customer Relationship Management System. Different enterprises use their own system to keep relationship with the customers. This environment happens in all industry including catering, banking, accounting, insurance…etc. Therefore, we would like to build up a Customer Relationship Management System for catering management in this final year project.

Besides, there are no any tools from the Internet let the people to create a mobile app or modify mobile app project information or coding in the Internet. Many people would like to learn creating mobile application but it is difficult to learn for people who do not have any programming skills before, especially the iPhone/ iPad technology with the development tools XCode.

Wholeheartedly, we would like to create a platform for the user to modify the client-side information with having any programming skills. This action is related to user-friendly in the issue of HCICM.

Different companies always have different needs; we need to provide a Customer Relational Management System to fit the requirements for different companies. The CRM system will include some common features and functions in catering management in Hong Kong. Therefore, this project is a “multi company” and “multi platform” CRM project that supports the catering management in Hong Kong nowadays.

Environmentally, we want to reduce the use of paper for leaflet or food menu. By this project, people can use their mobile device (i.e. BYOD-Bring Your Own Device) to receive the company latest information including promotion details, booking, menu…etc. This

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

ACKNOWLEDGEMENT

We would like to thank for Mr. Alan PO wholeheartedly, he is our supervisor of this final year project whose guidance, help us made this project as a reality and become great succeed. With his valuable opinions, assistance and supports made the whole set of systems in this project more successfully, innovative and differences. He also led us to explore the new technology and IT knowledge.

We would also like to thank Ms. Esther YUEN, who is our co-supervisor of this final year project; with her suggestions and opinion is usefulness in making the project more practical.

We also like to thank the Hong Kong IT Accountants Association (ITAA) and Software Factory Limited, their competition let our group gain a lot experience in developing CRM system in real life.

Finally, we would like to thank for all the lecturers in this course. They taught us these four years and making us efficiently and practically apply what we have learnt on this project.

LIST OF TABLES AND FIGURES

FIGURE 1. THE WEBSITE HTTP://WWW.ENOTECA.HK 18
FIGURE 2. THE FLASH ENVIRONMENT IN THE COMPANY 19
FIGURE 3. THE ENVIRONMENT DO NOT PROVIDE SMART DEVICE APPLICATION 19
FIGURE 4. MOBILE REQUEST JSON FILE AND DATA 22
FIGURE 5. THE APP FIRST PAGE WITH THE ICONS RELATED X-AXIS AND Y-AXIS 24
FIGURE 6. THE APP FIRST PAGE WITH THE ICONS RELATED X-AXIS AND Y-AXIS 24
FIGURE 7. THE CONNECTION WHEN THE CUSTOMER ENTERS OUR MOBILE APP 25
FIGURE 8. THE CONNECTION IS FAILED DUE TO NETWORK PROBLEMS 25
FIGURE 9. WEB SERVICE ADOPTED BETWEEN THE SERVER AND CLIENT CONNECTION 26
FIGURE 10. THE LAYOUT OF THE MOBILE APP. THE LAYOUT CAN LATER BE MODIFIED IN THE WEBSITE BY THE RESTAURANT STAFF 27
FIGURE 11. THE E-MENU ON THE MOBILE 28
FIGURE 12. REFERENCED APPLICATION IN APP STORE 29
FIGURE 13. CUSTOMER CAN CHOOSE TO GET THE COUPON 29
FIGURE 14. SOME COUPON MAY NEED THE POINTS FROM THE MEMBERS 30
FIGURE 15. CUSTOMER CAN VIEW HOW MANY COUPONS DOES SHE HAS AND THE AVAILABILITY OF THE COUPONS 30
FIGURE 16. THE COUPON WITH THE COUPON CONDITIONS AT THE BACK, SLIDE THE COUPON TO VIEW THE CONDITIONS. THIS SIMILAR TO THE ENVIRONMENT THAT YOU ARE HOLDING A CARD AND TURN IT TO THE BACK 31
FIGURE 17. THE BOOKING WEB SERVICE 32
FIGURE 18. THE BOOKING WEB SERVICE WITH AUTOMATICAL APPROVE 32
FIGURE 19. CUSTOMER CAN CREATE BOOKING, FILL IN THE BOOKING INFORMATION, SHE CAN CHOOSE THE SPECIAL REQUIREMENTS FROM THE LIST OR INSERT THE SPECIAL REQUIREMENT HERSELF 33
FIGURE 20. THE BOOKING CREATED SUCCESSFULLY 33
FIGURE 21. THE CUSTOMER LUCKY DRAW 34
FIGURE 22. THE CUSTOMER SLIDE THE CARD TO VIEW THE RESULT 34
FIGURE 23. THE POINTS OF THE VIP DEDUCTED AND GIFT SEND TO THE CUSTOMER 35
FIGURE 24. GPS LOCATION ON MOBILE 35
FIGURE 25. RESTAURANT LOCATION ON GOOGLE MAP AND ROUTING FROM MY POSITION TO THE RESTAURANT 36

FIGURE 26.CUSTOMER'S FEEDBACK 37
FIGURE 27. CUSTOMER CAN CHOOSE DIFFERENT FEEDBACK FORMS TO FILL 37
FIGURE 28. CUSTOMER SUBMIT THE FEEDBACK SUCCESSFULLY 38
FIGURE 29. WEB SERVICE 39
FIGURE 30. PUSH NOTIFICATION MESSAGE 39
FIGURE 31. MOBILE RECEIVE PUSH NOTIFICATION 40
FIGURE 32. RESTAURANT NEWS 40
FIGURE 34. APPROVE THE NEWS 41
FIGURE 35. NEWS ADDED 42
FIGURE 36. E-MENU VIEW (CATEGORY: MAIN) 42
FIGURE 37. E-MENU VIEW (CATEGORY: STARTER) 43
FIGURE 38. INSERT TABLE FOR E-MENU TO INSERT A NEW ITEM 43
FIGURE 39. SEARCH FUNCTION FOR THE E-MENU 44
FIGURE 40. SEARCH FUNCTION FOR THE E-MENU 44
FIGURE 41. BOOKING LIST 45
FIGURE 42. PRINTING BOOKING LIST 45
FIGURE 43. FEEDBACK VIEW 46
FIGURE 44. FEEDBACK FORM - INSERT NEW QUESTIONS 47
FIGURE 45. MOBILE APP CONFIGURATION 48
FIGURE 46 MOBILE APP ICON POSITION 49
FIGURE 47 MOBILE APP FUNCTION CONFIRMATION 50
FIGURE 48. COUPON LIST 51
FIGURE 49. NEW COUPON 51
FIGURE 50. MODIFY RESTAURANT USER PERMISSION VIEW 52
FIGURE 51. EVENT CONTROL LIST 53
FIGURE 52. SHOP DETAILS 54
FIGURE 53. SHOP LOCATION WITH GOOGLE MAP API 54
FIGURE 54. LUCKY DRAW LIST 55
FIGURE 55. CREATE NEW LUCKY DRAW 55
FIGURE 56. DATA ANALYSIS CENTER 56
FIGURE 57. DATA ANALYSIS WITH BAR CHART 56
FIGURE 58. DATA ANALYSIS OF CUSTOMER FEEDBACKS 57
FIGURE 59. MODEL VIEW CONTROLLER 63
FIGURE 60. SERVER ADMINISTRATOR 65

FIGURE 61. WEBSITE CHALLENGE AND RESPONSE LOGIN 66
FIGURE 62. TABLE STRUCTURE CRM_APPROVE_EMENU 74
FIGURE 63. TABLE STRUCTURE CRM_APPROVE_NEWS 74
FIGURE 64. TABLE STRUCTURE CRM_BOOKING 75
FIGURE 65. TABLE STRUCTURE CRM_BOOKING_PROPERTY 75
FIGURE 66. TABLE STRUCTURE CRM_COMPANY 75
FIGURE 67. TABLE STRUCTURE CRM_COUPON 76
FIGURE 68. TABLE STRUCTURE CRM_COUPON_CODE 76
FIGURE 69. TABLE STRUCTURE CRM_DEVICE 76
FIGURE 70. TABLE STRUCTURE CRM_EMENU 77
FIGURE 71. TABLE STRUCTURE CRM_EMENU_TYPE 77
FIGURE 72. TABLE STRUCTURE CRM_ERROR_LOG 77
FIGURE 73. TABLE STRUCTURE CRM_FEEDBACK 77
FIGURE 74. TABLE STRUCTURE CRM_FEEDBACK_CUST 78
FIGURE 75. TABLE STRUCTURE CRM_FEEDBACK_CUST_ANS 78
FIGURE 76. TABLE STRUCTRUE CRM_FEEDBACK_QUESTION_MC_ANS 78
FIGURE 77. TABLE STRUCTURE CRM_FEEDBACK_SHOP 78
FIGURE 78. TABLE STRUCTURE CRM_LUCKYDRAW 79
FIGURE 79. TABLE STRUCTURE CRM_LUCKYDRAW_PRIZE 79
FIGURE 80. TABLE STRUCTURE CRM_LUCKYDRAW_PRIZE_REG 79
FIGURE 81. TABLE STRUCTURE CRM_LUCKYDRAW_SHOP 80
FIGURE 82. TABLE STRUCTURE CRM_MEMBER 80
FIGURE 83. TABLE STRUCTURE CRM_NEWS 80
FIGURE 84. TABLE STRUCTURE CRM_ORDER 81
FIGURE 85. TABLE STRUCTURE CRM_ORDER_DETAIL 81
FIGURE 86. TABLE STRUCTURE CRM_QUEUE 81
FIGURE 87. TABLE STRUCTURE CRM_SHOP 82
FIGURE 88. TABLE STRUCTURE CRM_SHOP_SPECIAL_EVENT 83
FIGURE 89. TABLE STRUCTURE CRM_TABLE 83
FIGURE 90. TABLE STRUCTURE CRM_TABLE_PROPERTY 83
FIGURE 91. TABLE STRUCTURE CRM_THEME 84
FIGURE 92. TABLE STRUCTURE CRM_THEME_ANDROID 85
FIGURE 93. TABLE STRUCTURE CRM_THEME_IPHONE 86
FIGURE 94. TABLE STRUCTURE CRM_USER 86

FIGURE 95. TABLE STRUCTURE CRM_WEB_USER 87
FIGURE 96. TABLE STRUCTURE CRM_WEB_USERGROUP 87
FIGURE 97. MOBILE APP ICON 280
FIGURE 98. APPLICATION MAIN MENU 281
FIGURE 99. MAIN INTERFACE 282
FIGURE 100. USER INTERFACE NEWS 283
FIGURE 101. RESTAURANT INFORMATION INTERFACE 284
FIGURE 102. RESTAURANT LOCATION INTERFACE 285
FIGURE 103. RESTAURANT MENU INTERFACE 286
FIGURE 104. SHARING TO SOCIAL NETWORK 287
FIGURE 105. BOOKING INTERFACE 288
FIGURE 106. VIEW BOOKING STATUS INTERFACE 289
FIGURE 107. WEBSITE INDEX PAGE 290
FIGURE 108. WEBSITE FOOD MENU 291
FIGURE 109. WEBSITE BOOKING LIST 292
FIGURE 110. ABUSE BOOKING USER 292
FIGURE 111. PRINT BOOKING LIST 293
FIGURE 112. WEBSITE PUSH NOTIFICATION 294
FIGURE 113. WEBSITE MOBILE CONFIGURATION 295
FIGURE 114.WEBSITE CONTROL FUNCTION ON MOBILE 296
FIGURE 115. WEBSITE CHANGE MOBILE APP ICON 297
FIGURE 116. WEBSITE E-COUPON 298
FIGURE 117. WEBSITE PERMISSION SETTING 299
FIGURE 118. WEBSITE SET SHOP LOCATION 300
FIGURE 119. WEBSITE DATA ANALYSIS 301
FIGURE 120WEBSITE DATA ANALYSIS 301
FIGURE 121. WEBSITE BAR CHART 302
FIGURE 122. CSV REPORT OF DATA ANALYSIS 303

INTRODUCTION

Document Structure

This report contains nine major sections:

- Introduction

  A general description of this document structure, the project and its background.

- The requirements

  This part is the descriptions of the scope of the proposed system, which means the system boundary, descriptions of the functions provided and the data processed by the system, other non-functional requirements, such as the interface with existing system and sub-systems.

- Documentation for problem analysis

  This part is the data modeling with ERD and Class diagram, functional modeling, use case diagrams and use case descriptions, state transition diagram, sequence diagram and object collaboration diagram, data dictionary.

- Documentation for detailed design

  Data design, software and hardware architectural design, procedural design (module) and user interface design.

- Implementation

  Records of the implementation work including the test plans, test results, changes to design and justification of changes.

- Results and Conclusion

  A summary and a critical discussion of the results, conclusions, any problems or difficulties encountered in our project, any delays or changes in project schedule, limitations of the proposed system, and in some cases a subsection suggesting further developments to be undertaken.

- References

  The reference website hyperlinks, books, leaflet, company's opinion and material we reference to across this project.

- Appendices

  The original project plan with segmentation of activities and estimated times for those completion, which we submitted in our Initial Report, together with a revised plan, which records how you actually spent the time. Also, the copy of our project log book.

The Description Of This Project

Smart Phones are becoming much popular in Hong Kong. Smart Phone contains an operating system inside the mobile. Nowadays, there are two most popular systems, which are iOS and Android. They let people to use their mobiles to do more than before and build up communication easily between peoples. 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 mobile application. The web service will provides map data to the client application automatically when the client startup our application on the mobile with valid network connection via WiFi or 3G. User can use the mobile application to view the news, e-menu, request e-coupon, lucky draw, GPS location directing, and also send us their feedback on the mobile.

All of the user's habit that we received will be used for data analysis to further make improvement on the sales of the company. All of the data is under the user permitted circumstances and will be protected safely in our database. Customer Relationship Management System (CRM) means that it is a system can be used for not only one company or group of companies. It should be able to let another company to adapt using this system without a lot of modification on the program template. Therefore, we will concentrate on not only the system functions, but also the further implementation to meet different companys' requirements.
To provide further development, we should first develop a CRM system based on the reference mentor "Enoteca Group" for our 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.

Besides the competition mentor, we have also got opinions from another company Café DeCo Group (http://www.cafedecogroup.com/). One of their restaurants “Dim Sum Bar” general manager Mr. Eagle PO gave us a lot of experience in catering manager. They reviewed our project and suggested us to do more useful things to meet different requirements of different company. Also, the PR & Marketing Executive of Café DeCo Group Ms. Quinnie Suen also gave us the experience on marketing of catering company.

Base on the completion mentor Enoteca Group and the Café DeCo Group, we further make our project become more successful and userful.

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.

Figure 1. The website http://www.enoteca.hk

The company does not have any mobile app on any platforms of the smart devices.

Problems Description

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

Second, the company does not have any mobile application that let customer to be noticed 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.

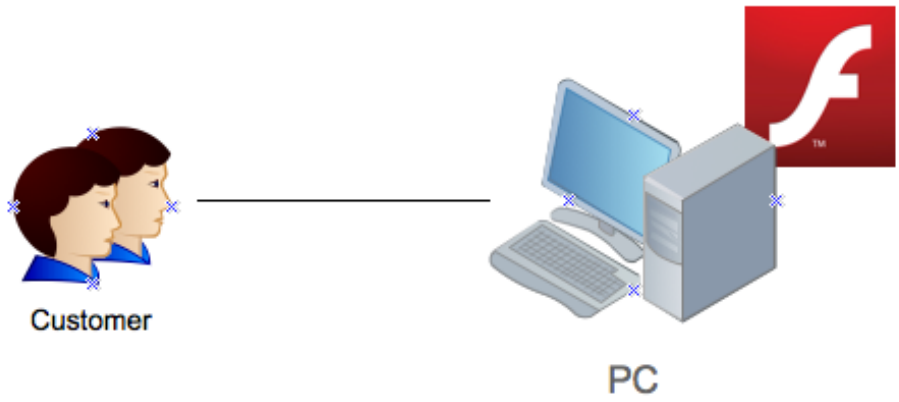


Figure 2. The flash environment in the company

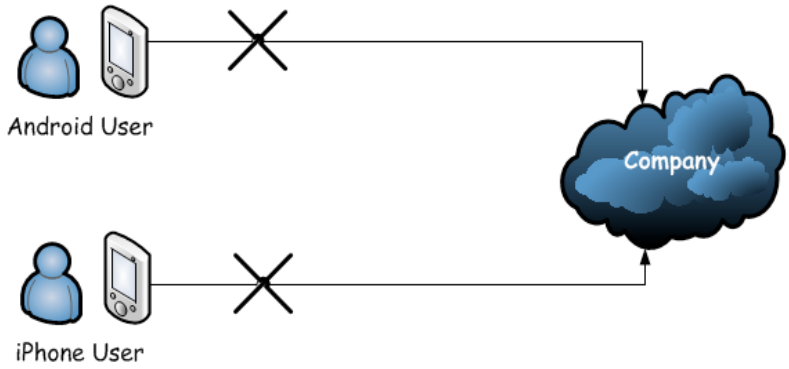


Figure 3. The environment do not provide smart device application

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 analysis tools to analysis the customers' comment or their order habits. 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.

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.

THE REQUIREMENTS

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
- Mobile app 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
- The mobile configuration on the website
- How to handle if the client can't connect the server in some specific area
- How to support multi-platform client
- How to support multi-restaurant
- How to make client easier to use our mobile application
- How the way to implement the system in the future
- Analysis reports for customers' habits
- Analysis reports for customers' feedback
- Description for the hardware requirement

The way of client to archive server's address

We must assume that the customer uses smart devices with our provided mobile app. The smart device is Apple iPhone or Android mobile. When the customer installs the mobile app and starts it in the first time, the app will automatically connect to our server at "http://alanpo.com/myfyp/crm/json" and receive all the company information to the mobile app by using AJAX and JSON technology.

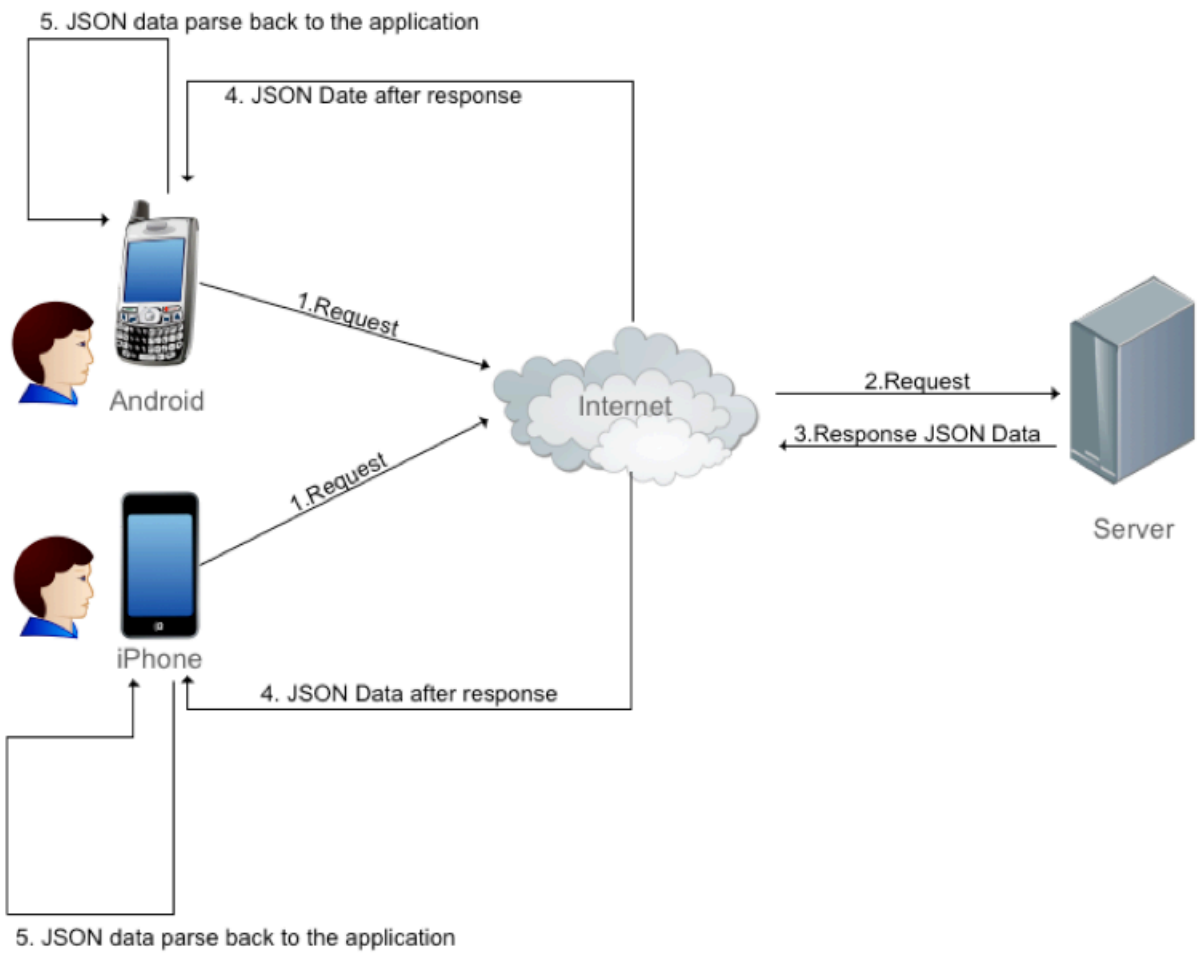


Figure 4. Mobile request JSON file and data

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 SQLite technology. And then the app will parse back the data to each tab and page on the mobile with the related index stored in the database. 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 and the location of 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 X-axis and Y-axis on the screen which shows in the following diagram.

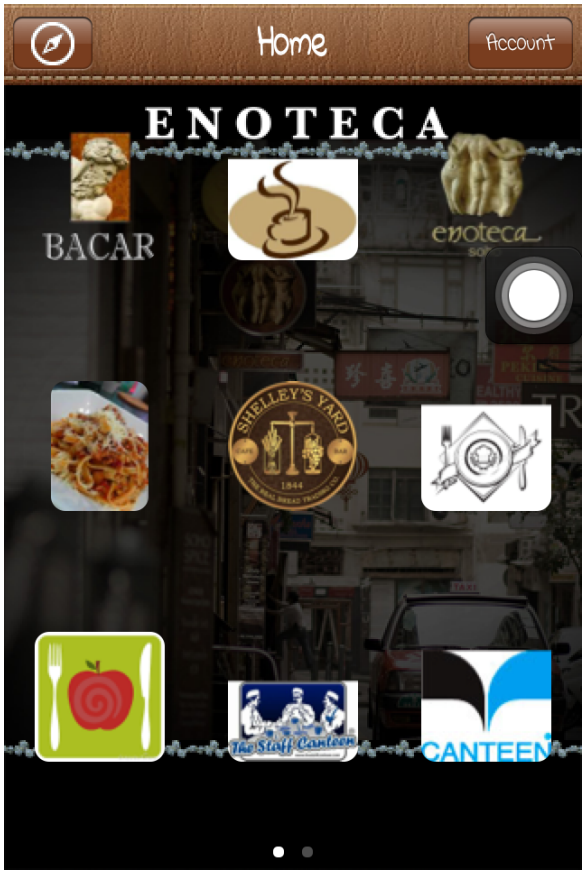


Figure 5. The app first page with the icons related x-axis and y-axis

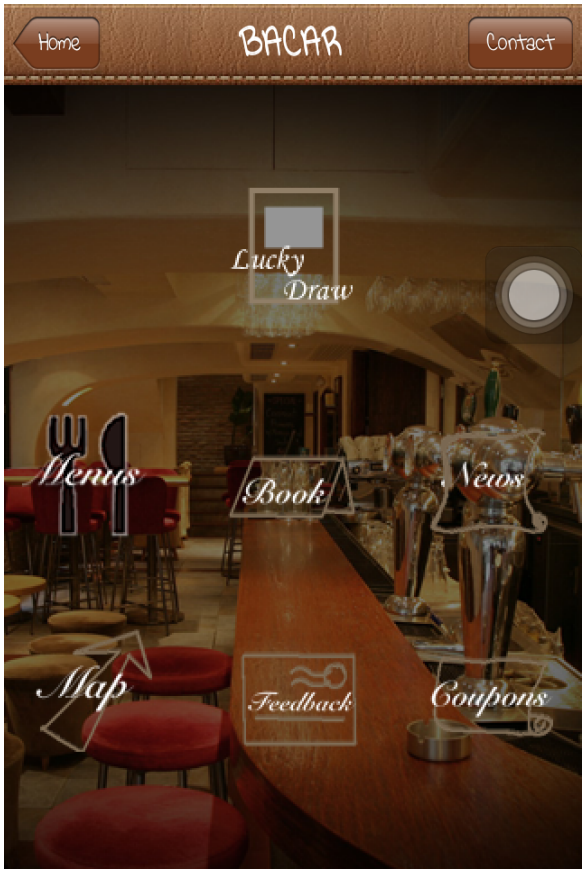


Figure 6. The app first page with the icons related x-axis and y-axis

Mobile app automatically connect to the server and receiving latest information from the server

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

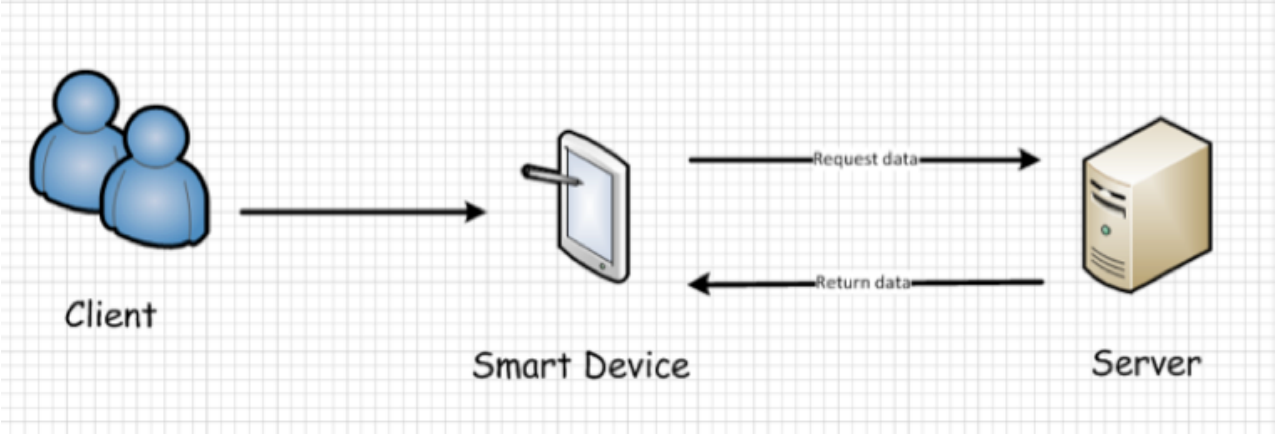


Figure 7. The connection when the customer enters our mobile app

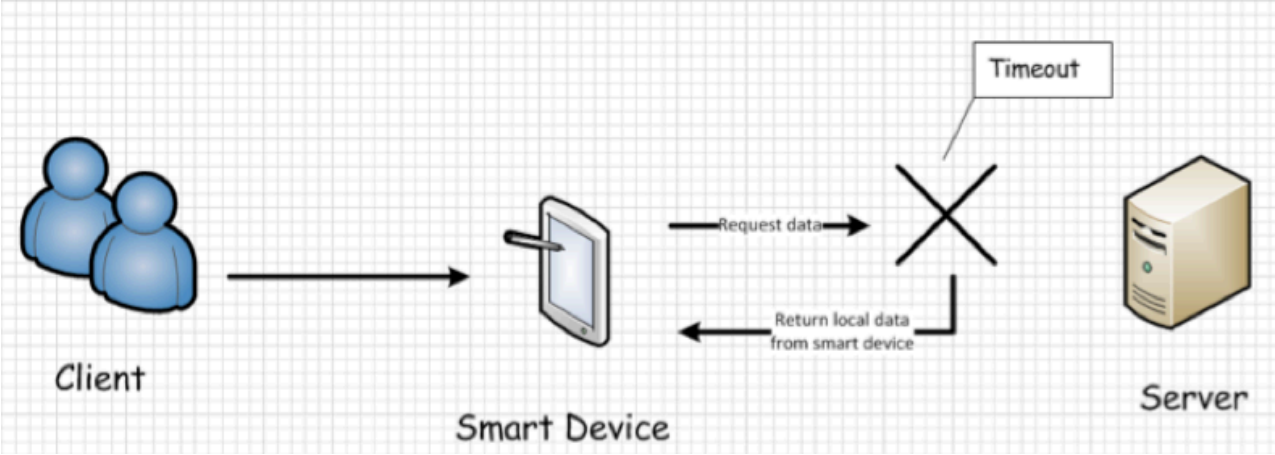


Figure 8. The connection is failed due to network problems

The Client opens the application and not able to connect to the server due to network connection problem. The app will use the information from the mobile local database since last updated on her own device.

The way of client and server to communication

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

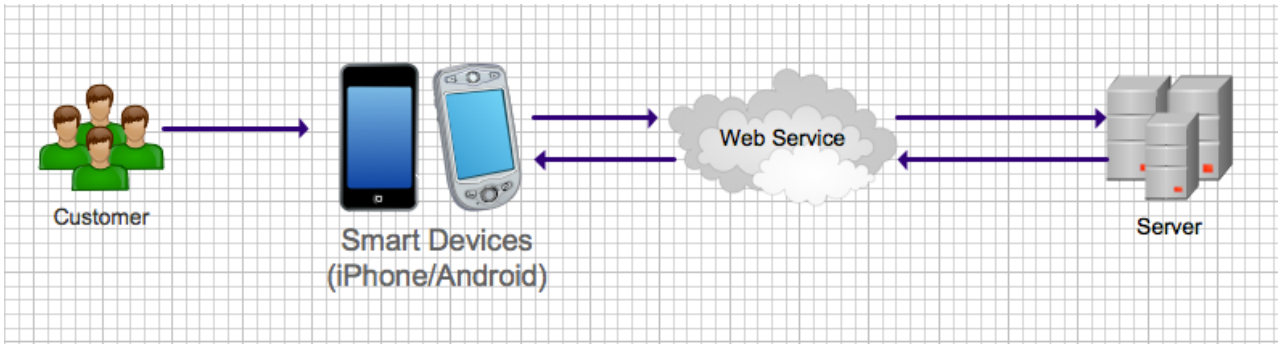


Figure 9. Web service adopted between the server and client connection

Consistency of data structure between server slide and client slide

We suggest the customer use the mobile in a network available location that means they use our app with Wi-Fi or 3G networks. The company staff can use the provided website to modify their latest news, information, data into the database. The customer must have 3G or Wi-Fi network provider to keep their app data updated. When the user uses our app, they can receive the latest information automatically.

Besides, we will use push notification to notice our customer 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.

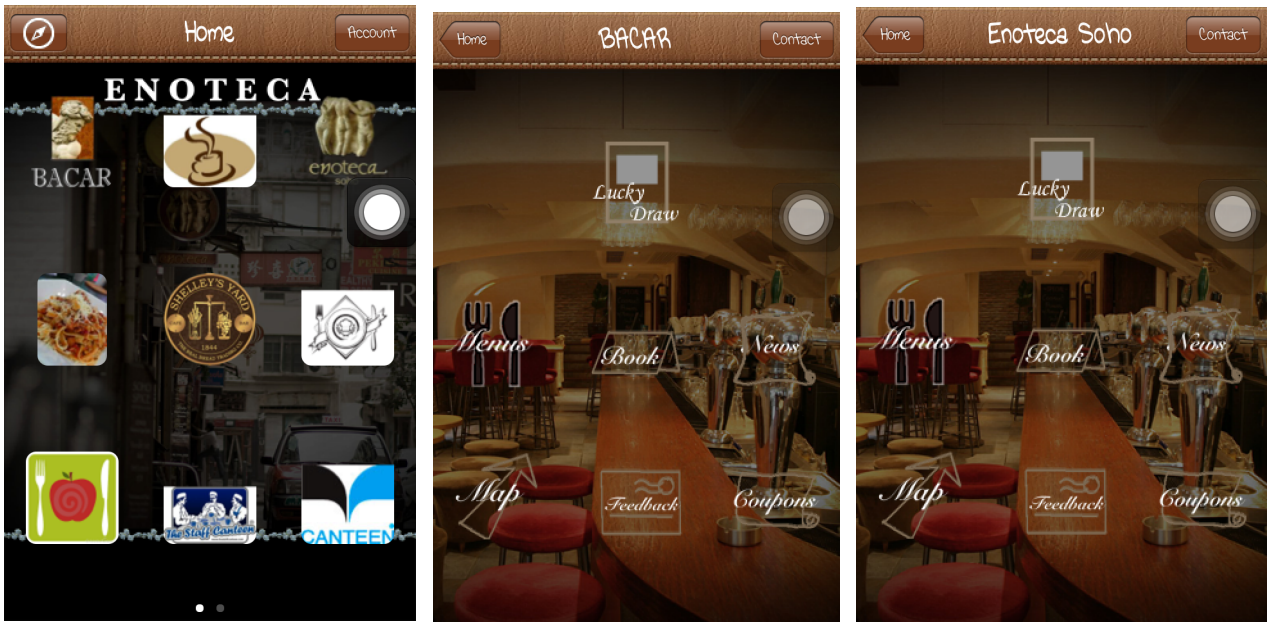


Figure 10. The layout of the mobile app. The layout can later be modified in the website by the restaurant staff

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.

Figure 11. The E-Menu on the mobile

- Electrical Coupon

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

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

Figure 12. Referenced Application in App store

The E-coupon we provided contains a 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.

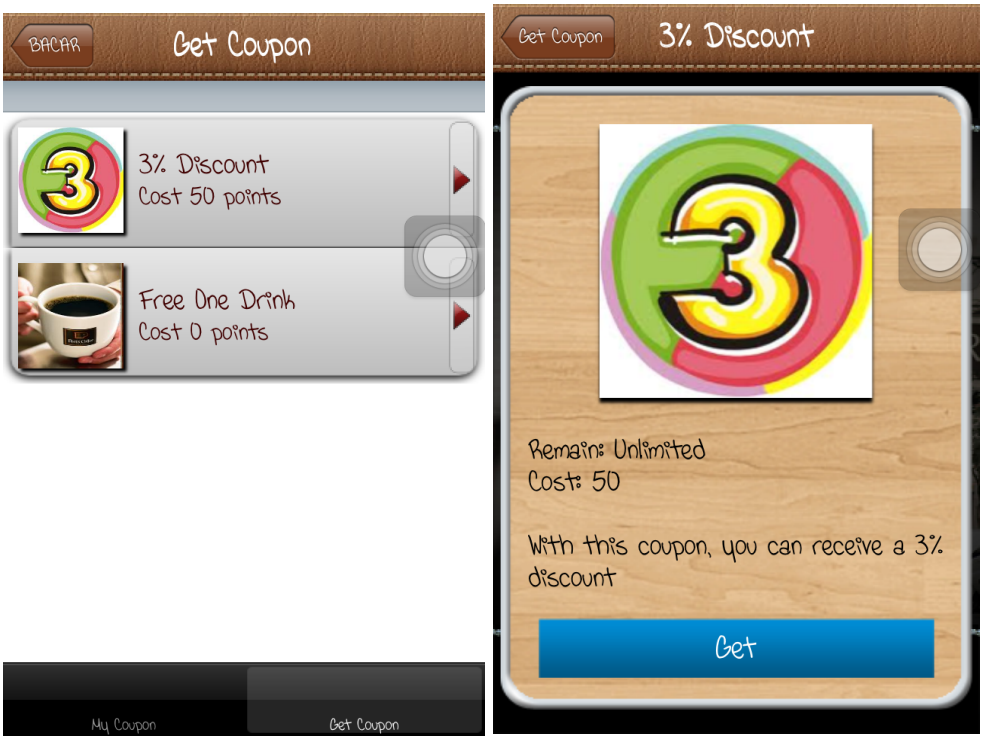


Figure 13. Customer can choose to get the coupon

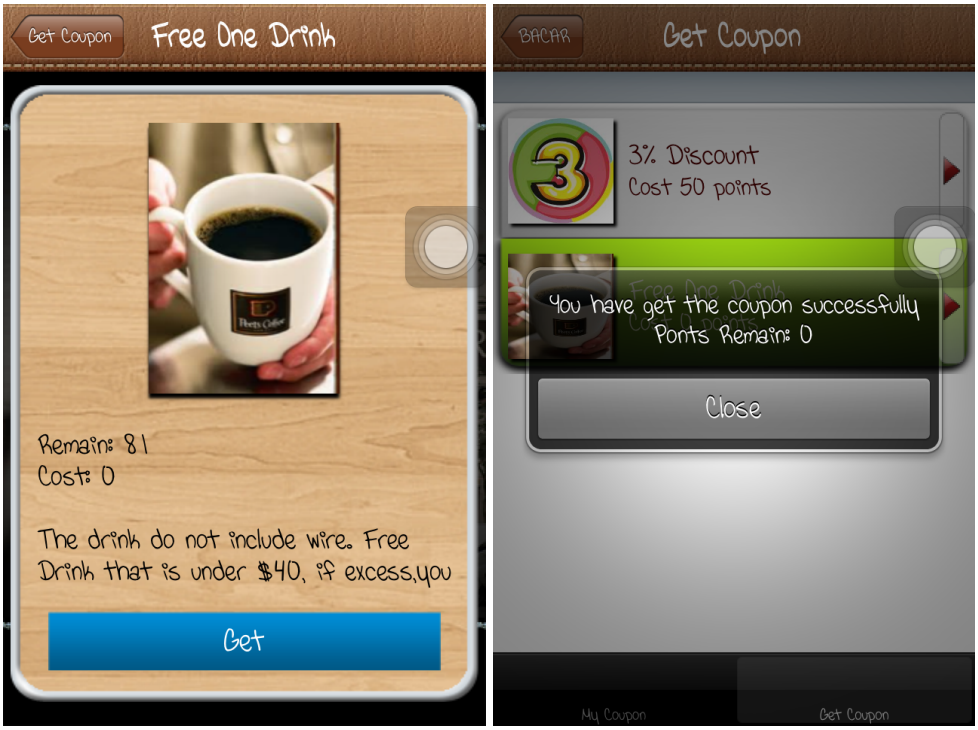


Figure 14. Some coupon may need the points from the members

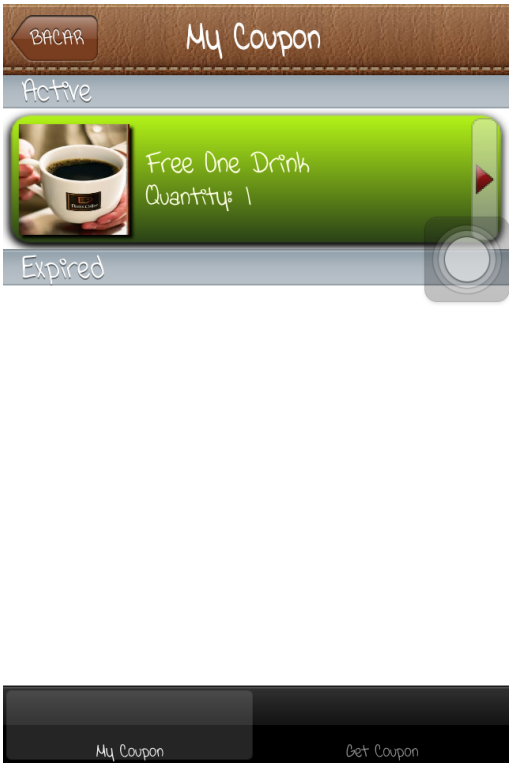


Figure 15. Customer can view how many coupons does she has and the availability of the coupons

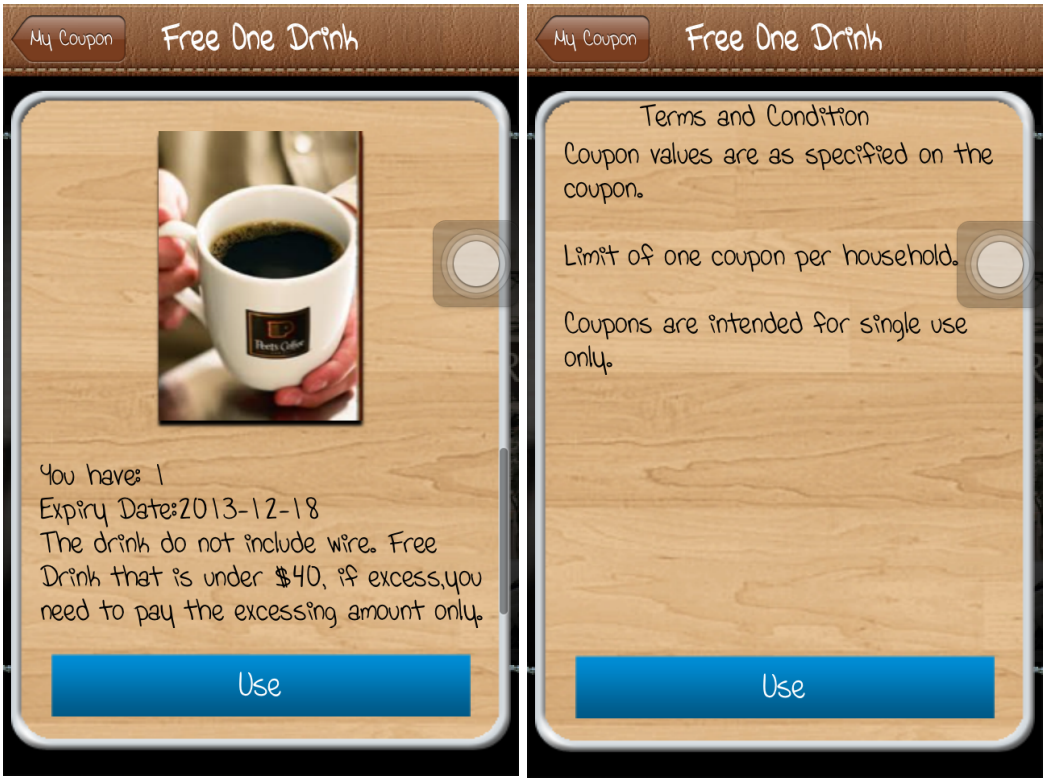


Figure 16. The coupon with the coupon conditions at the back, slide the coupon to view the conditions. This similar to the environment that you are holding a card and turn it to the back

- Booking Service

  The booking function let the customer to reserve tables in the restaurant for a specific time slot, customer uses their mobile to make reservation for a months. The data will sent to the server first and store in the database. The system will automatically approve the booking if the requirements are fit the timeslot or special requirement in the restaurant. If the requirements cannot fulfill in the restaurant, the system will suggest a coupon of timeslots or special requirement to the customer or just put it in the waiting list and wait for the staff to approve or reject in the website.

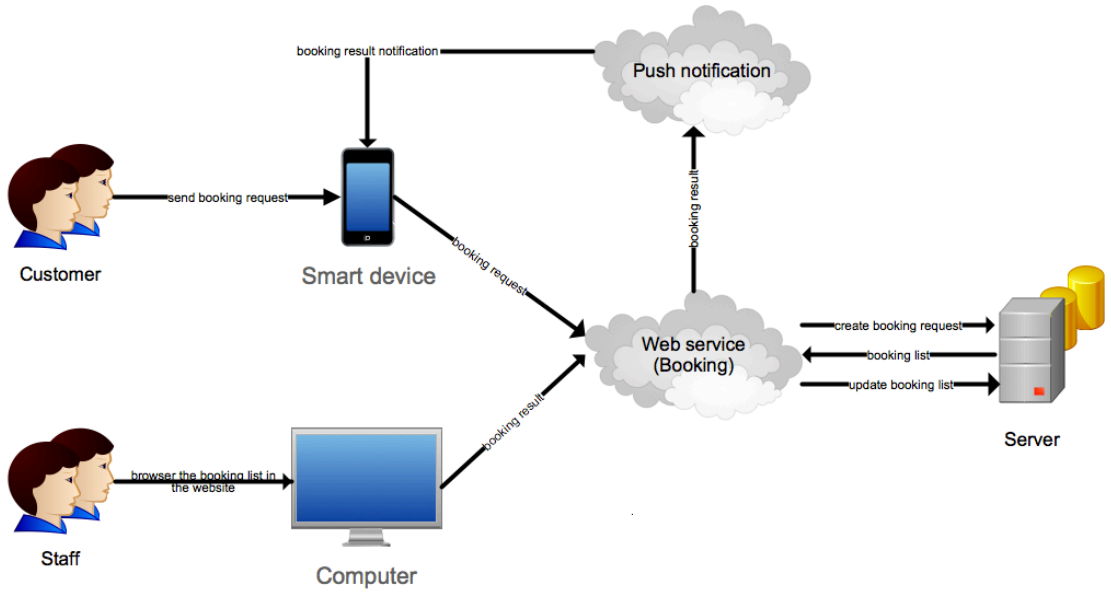


Figure 17. The booking web service

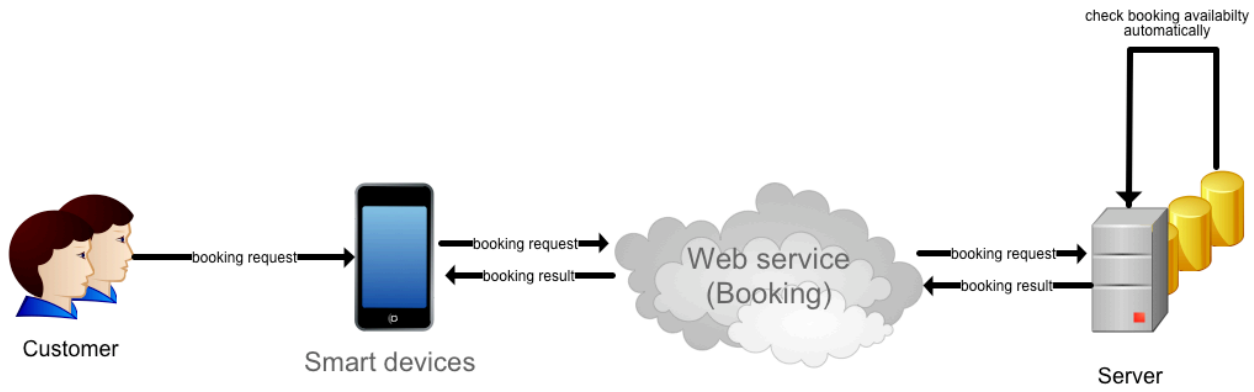


Figure 18. The booking web service with automatical approve

The user can also cancel the booking if she does not need the reservation. If the user book for a lot of timeslots in the same date, the system will notice the restaurant staff to do some following action.

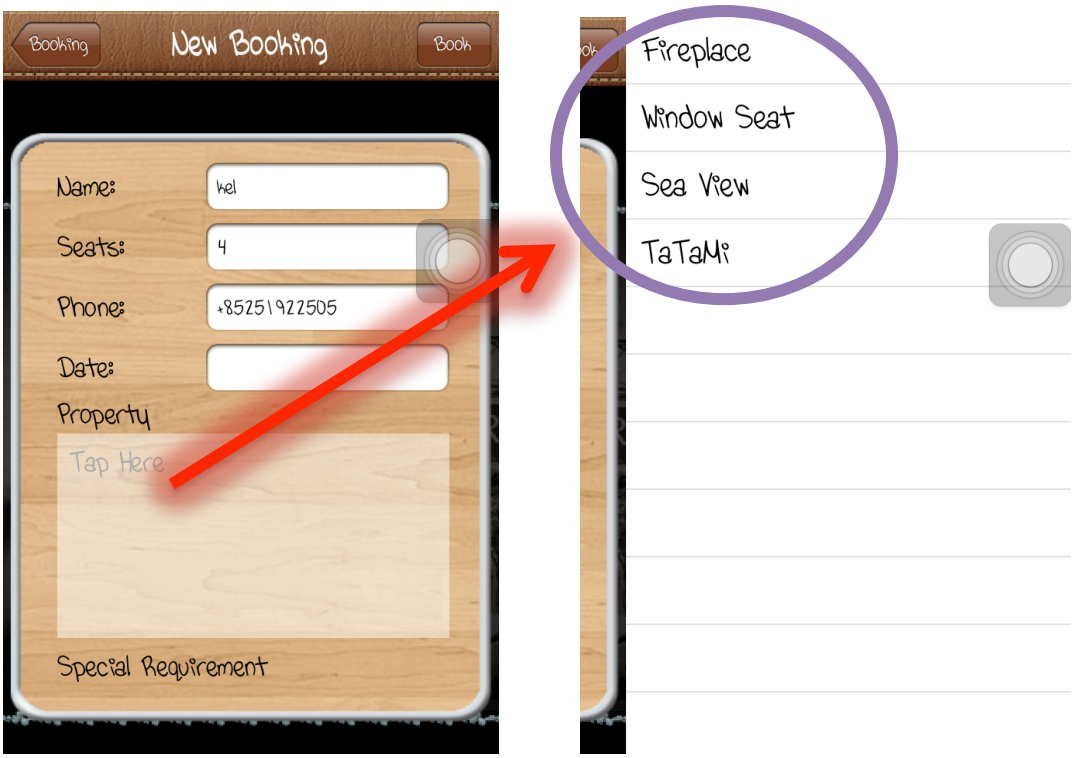


Figure 19. Customer can create booking, fill in the booking information, she can choose the special requirements from the list or insert the special requirement herself

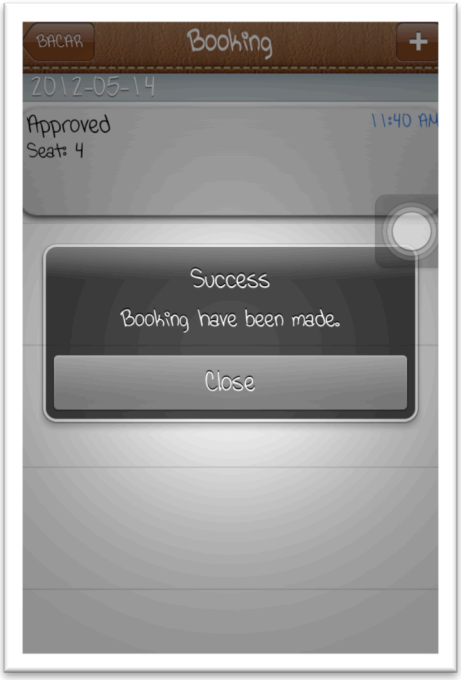


Figure 20. The booking created successfully

- Lucky Draw

  For the VIP of our restaurant, they can collect points when they purchase our service, the points can used to trade coupon of play the lucky draw on the mobile app. After they play the lucky draw, gift will be sent to the customer.

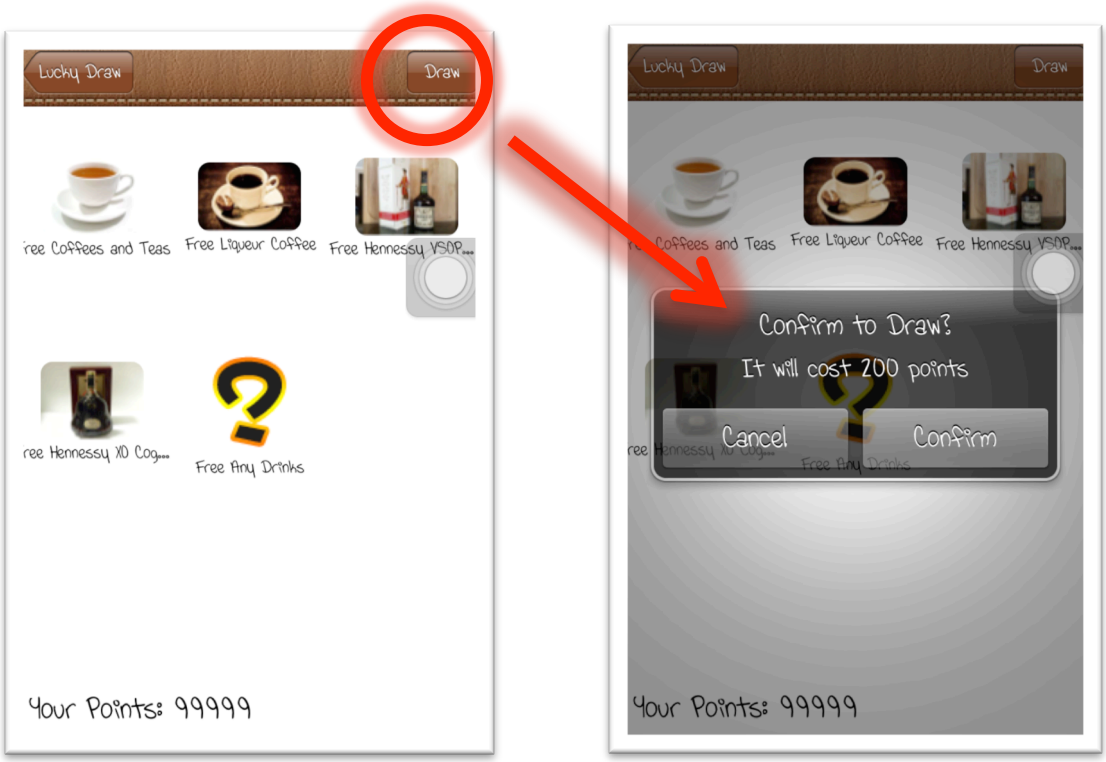


Figure 21. The customer lucky draw

Figure 22. The customer slide the card to view the result

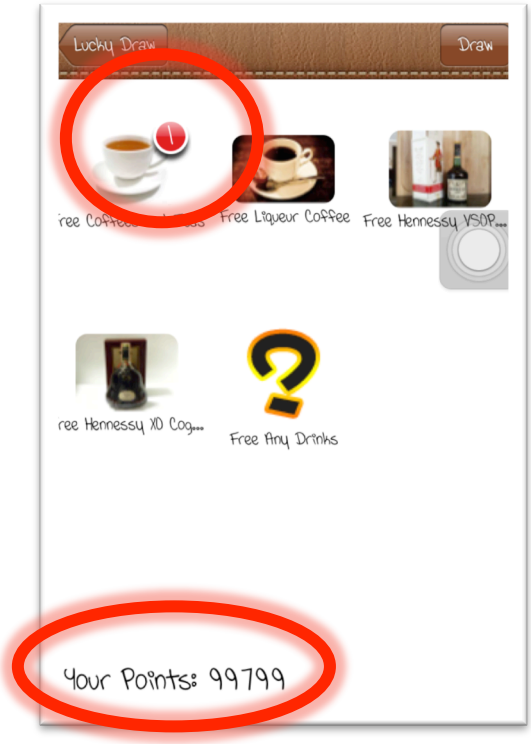


Figure 23. The points of the VIP deducted and gift send to the customer

- Location and path direction on our 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.

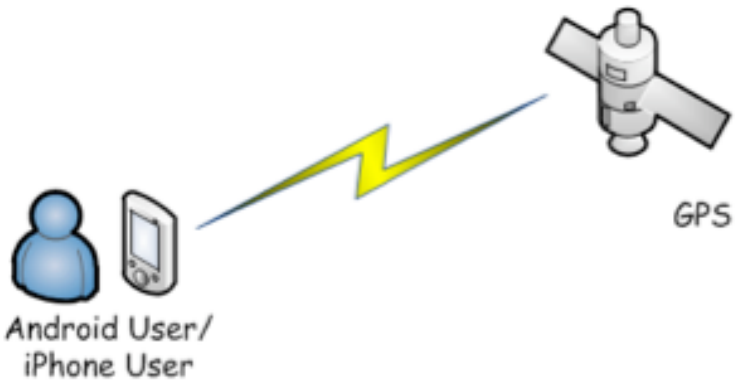


Figure 24. GPS Location on mobile

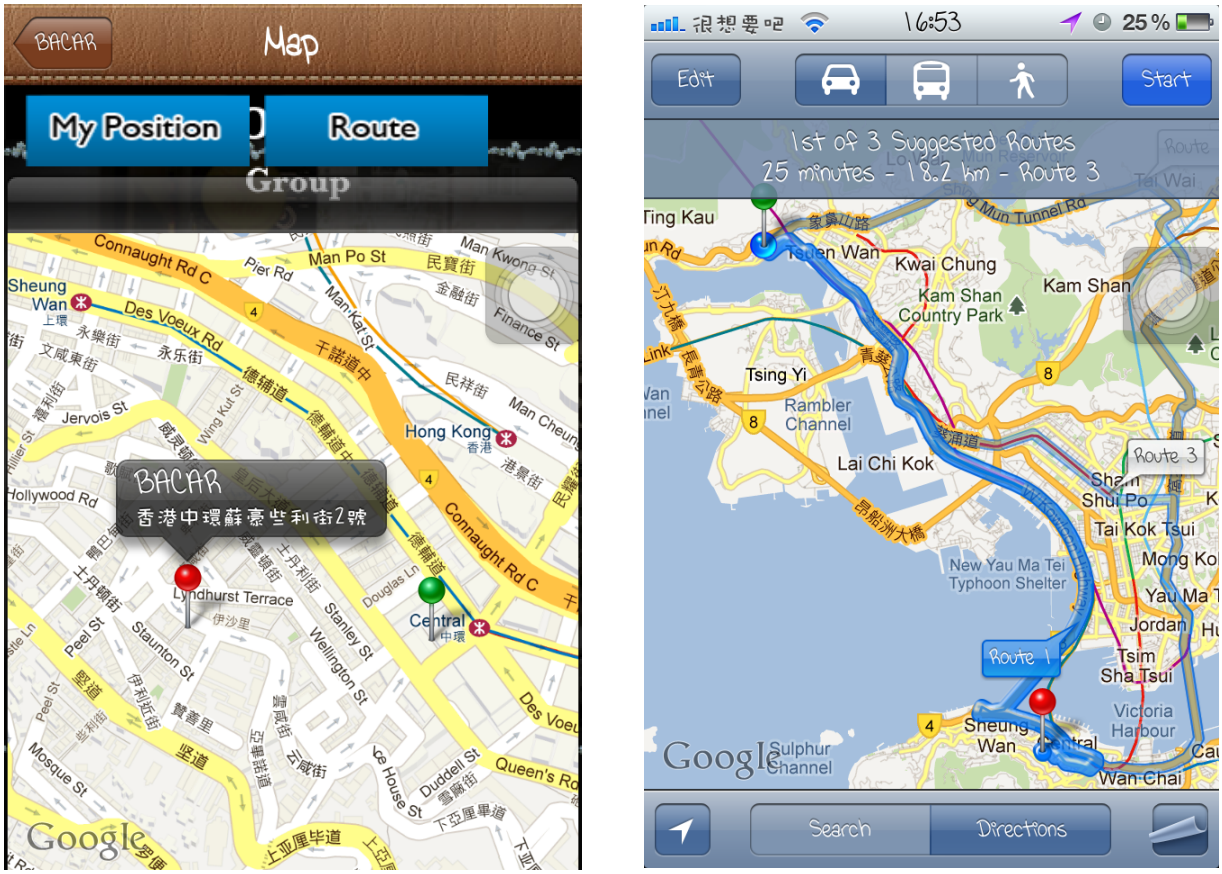


Figure 25. Restaurant Location on Google Map and Routing from my position to the restaurant

- 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.

Figure 26.Customer's Feedback

Figure 27. Customer can choose different feedback forms to fill

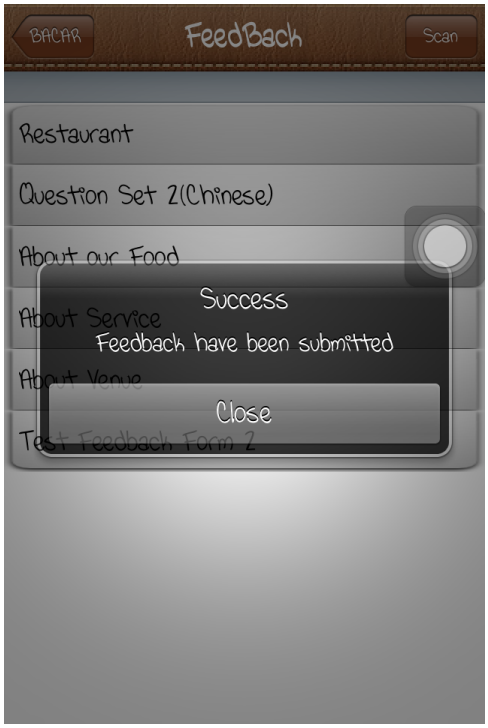


Figure 28. Customer submit the feedback successfully

- Receiving push notification message

  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.

  The mobile devices can receive the notification message sent from the server, the method of sending different notification message will be discussed later.

The functions of the website and web service

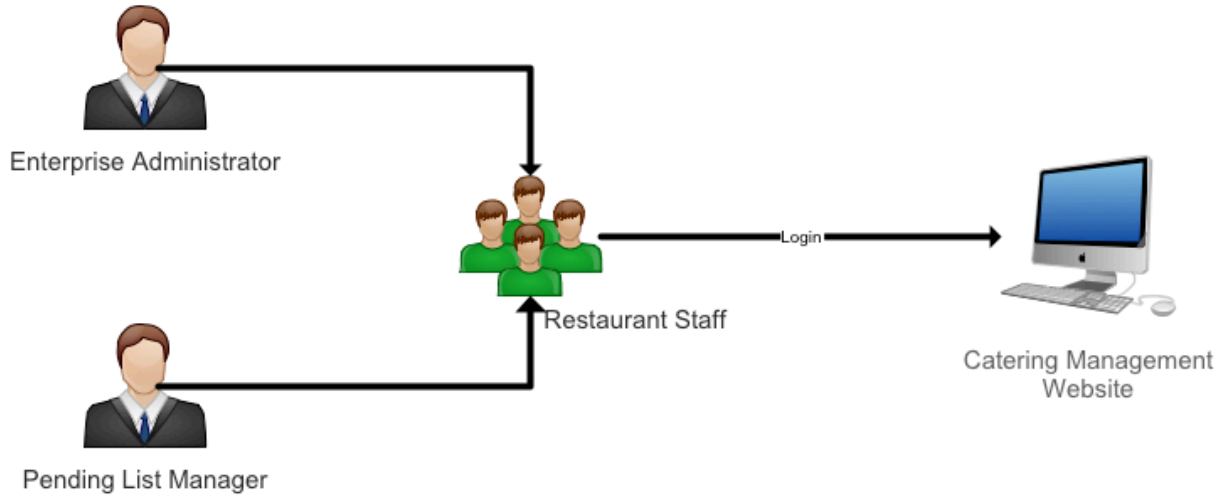


Figure 29. Web service

The website contains a lot of useful functions such as the following:

- Send push notification message to mobile

  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.

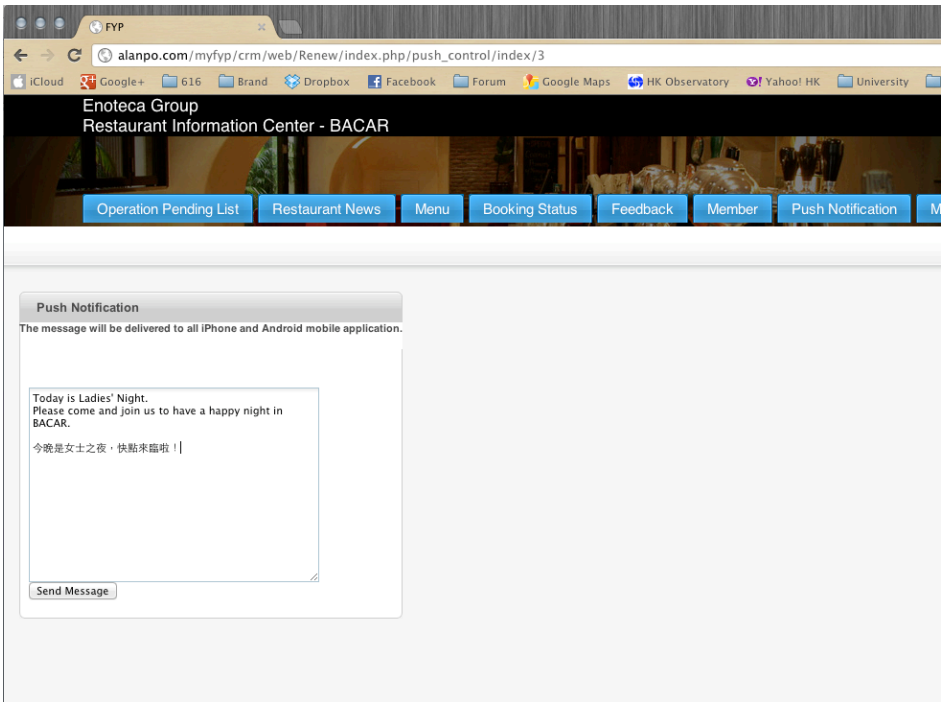


Figure 30. Push notification message

When the customer receive the message, a notification will be prompted out and return to the related app.

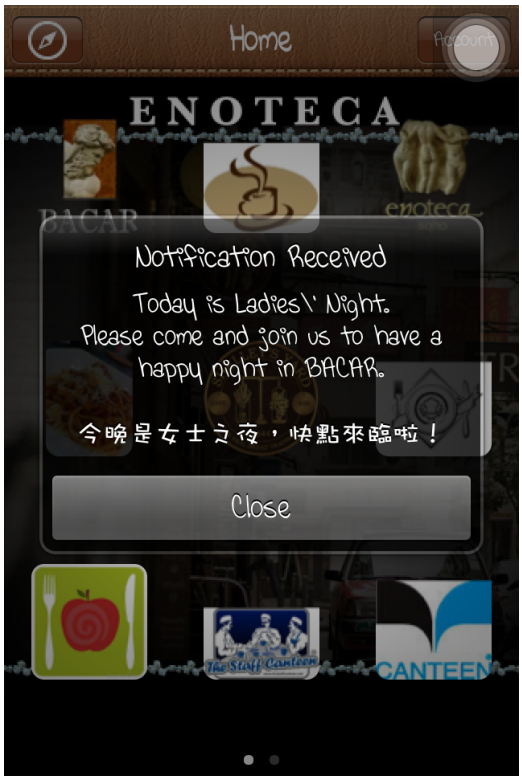
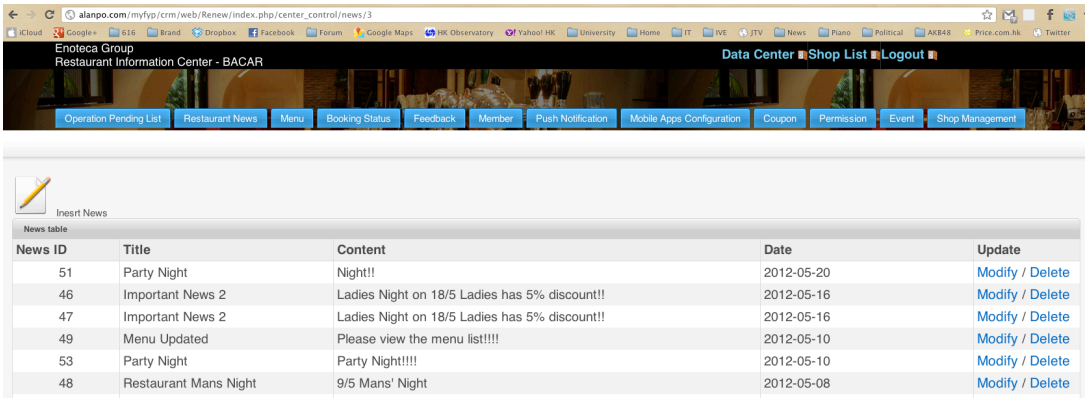


Figure 31. Mobile receive push notification

- Update restaurant news

  The restaurant staff can insert, modify or delete restaurant news from her company web page. After the news is updated, it will send to the approve list in the website, the enterprise administrator and the pending list manager can approve the update. Once the update is approved, the customer can view on their mobile app.



News ID	Title	Content	Date	Update
51	Party Night	Night!!	2012-05-20	Modify / Delete
46	Important News 2	Ladies Night on 18/5 Ladies has 5% discount!!	2012-05-16	Modify / Delete
47	Important News 2	Ladies Night on 18/5 Ladies has 5% discount!!	2012-05-16	Modify / Delete
49	Menu Updated	Please view the menu list!!!!	2012-05-10	Modify / Delete
53	Party Night	Party Night!!!!	2012-05-10	Modify / Delete
48	Restaurant Mans Night	9/5 Mans' Night	2012-05-08	Modify / Delete

Figure 32. Restaurant news

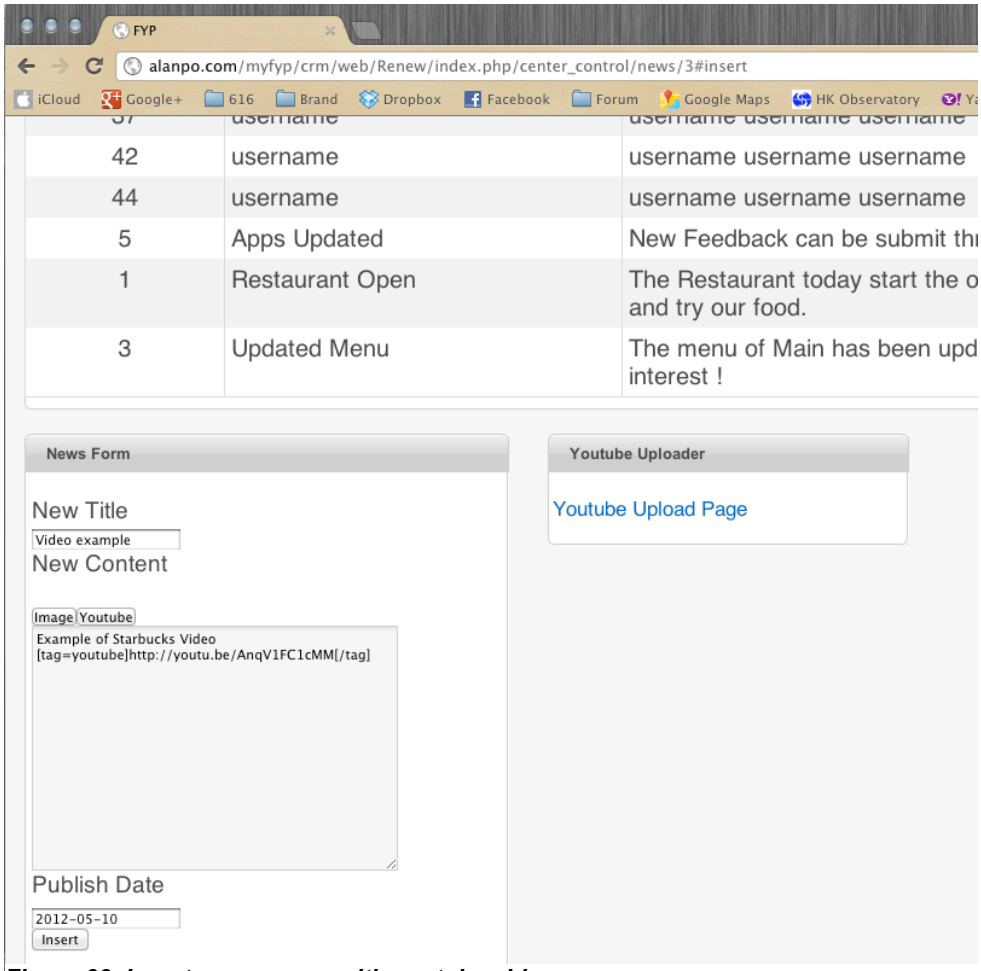


Figure 33. Insert a new news with youtube video

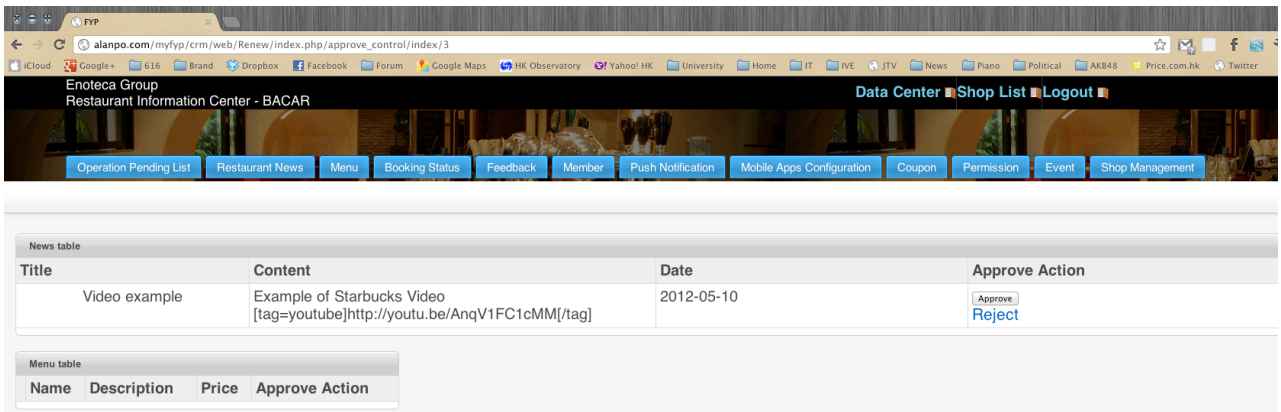
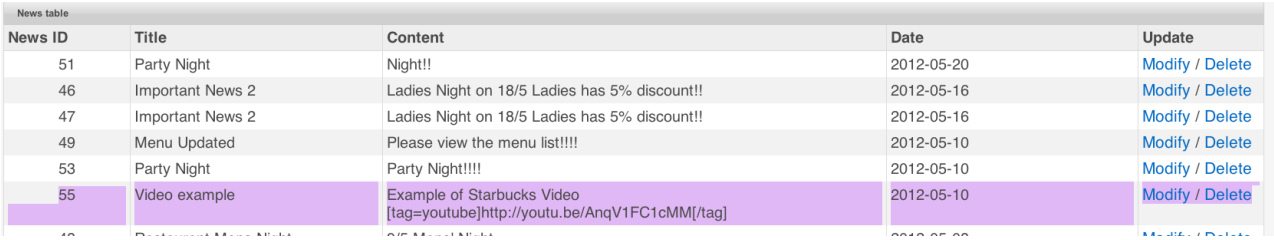


Figure 34. Approve the news



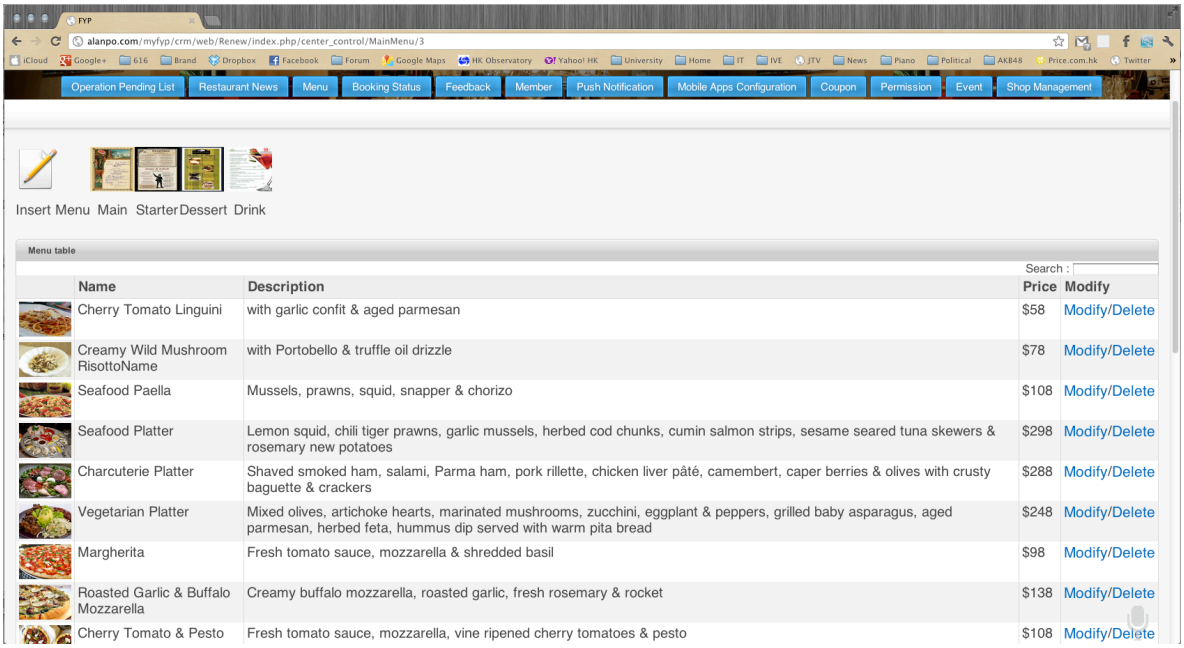
News table

News ID	Title	Content	Date	Update
51	Party Night	Night!!	2012-05-20	Modify / Delete
46	Important News 2	Ladies Night on 18/5 Ladies has 5% discount!!	2012-05-16	Modify / Delete
47	Important News 2	Ladies Night on 18/5 Ladies has 5% discount!!	2012-05-16	Modify / Delete
49	Menu Updated	Please view the menu list!!!!	2012-05-10	Modify / Delete
53	Party Night	Party Night!!!!	2012-05-10	Modify / Delete
55	Video example	Example of Starbucks Video [tag=youtube]http://youtu.be/AnqV1FC1cMM[/tag]	2012-05-10	Modify / Delete

Figure 35. News added

- Update E-Menu

  The restaurant staff 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. They can delete the item in the e-menu if it is outdated. 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. Same as the above function, the things will need to be approve by the enterprise administrator or the pending list manager.



alanpo.com/myfyp/crm/web/Renew/index.php/center_control/MainMenu/3

Operation Pending List | Restaurant News | Menu | Booking Status | Feedback | Member | Push Notification | Mobile Apps Configuration | Coupon | Permission | Event | Shop Management

Insert Menu Main StarterDessert Drink

Menu table

Search :

	Name	Description	Price	Modify
	Cherry Tomato Linguini	with garlic confit & aged parmesan	$58	Modify/Delete
	Creamy Wild Mushroom RisottoName	with Portobello & truffle oil drizzle	$78	Modify/Delete
	Seafood Paella	Mussels, prawns, squid, snapper & chorizo	$108	Modify/Delete
	Seafood Platter	Lemon squid, chili tiger prawns, garlic mussels, herbed cod chunks, cumin salmon strips, sesame seared tuna skewers & rosemary new potatoes	$298	Modify/Delete
	Charcuterie Platter	Shaved smoked ham, salami, Parma ham, pork rillette, chicken liver pâté, camembert, caper berries & olives with crusty baguette & crackers	$288	Modify/Delete
	Vegetarian Platter	Mixed olives, artichoke hearts, marinated mushrooms, zucchini, eggplant & peppers, grilled baby asparagus, aged parmesan, herbed feta, hummus dip served with warm pita bread	$248	Modify/Delete
	Margherita	Fresh tomato sauce, mozzarella & shredded basil	$98	Modify/Delete
	Roasted Garlic & Buffalo Mozzarella	Creamy buffalo mozzarella, roasted garlic, fresh rosemary & rocket	$138	Modify/Delete
	Cherry Tomato & Pesto	Fresh tomato sauce, mozzarella, vine ripened cherry tomatoes & pesto	$108	Modify/Delete

Figure 36. E-Menu View (Category: Main)

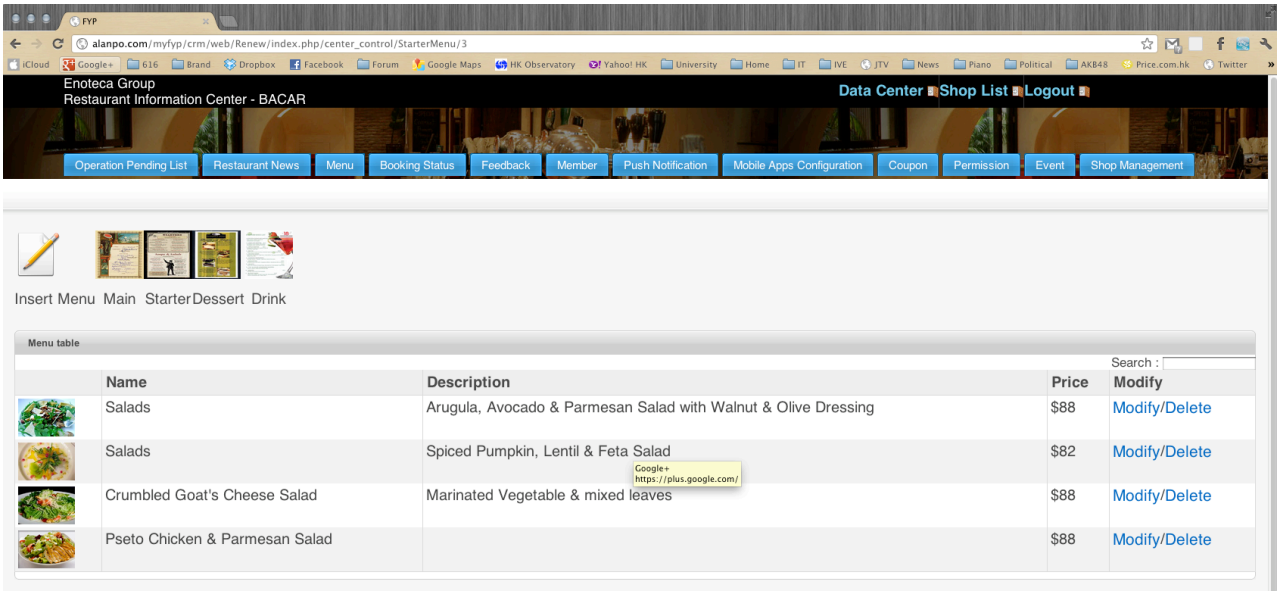


Figure 37. E-Menu view (Category: starter)

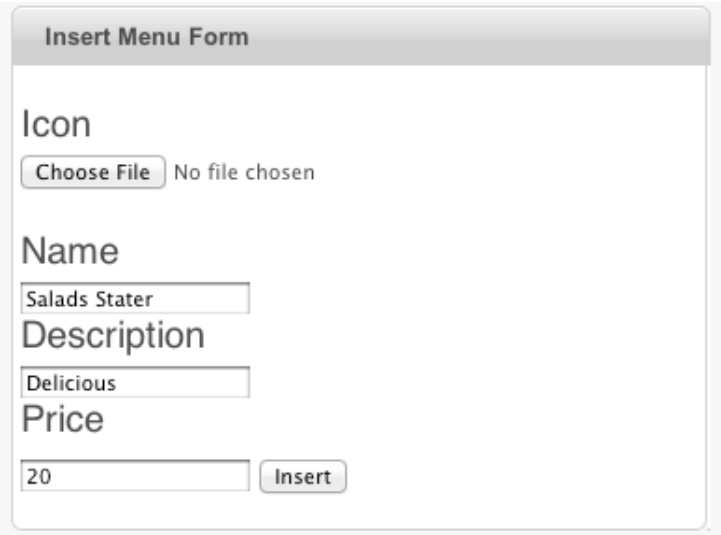


Figure 38. Insert table for E-Menu to insert a new item

To make the process much user-friendly, we provide a search function in the E-Menu for the restaurant manager to find and modify the data more quickly. She can type some key words of the food name, and then the data will be displayed if the food name is match with the letters.

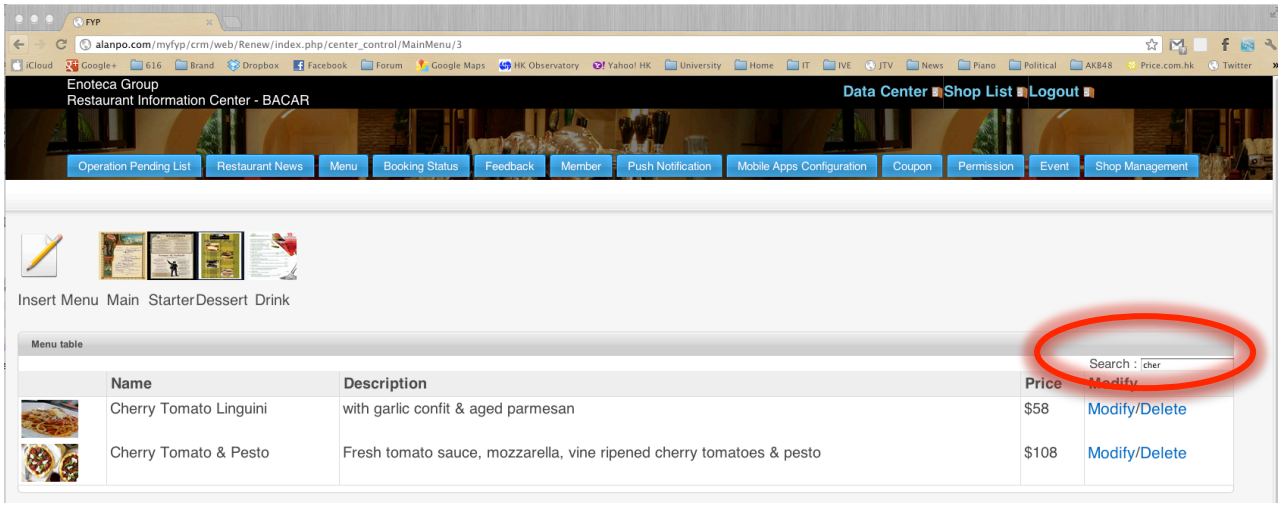


Figure 39. Search function for the E-Menu

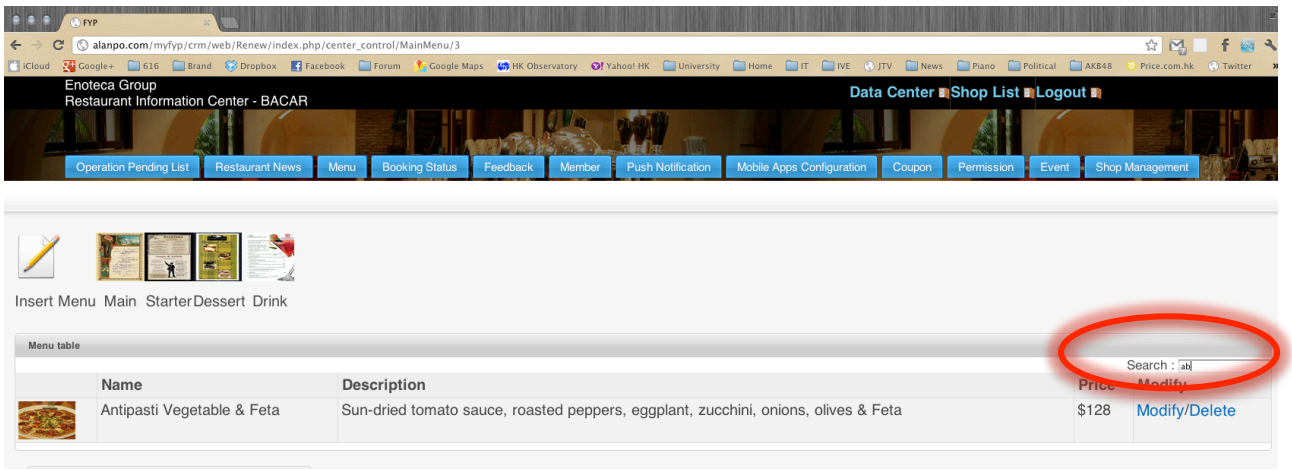
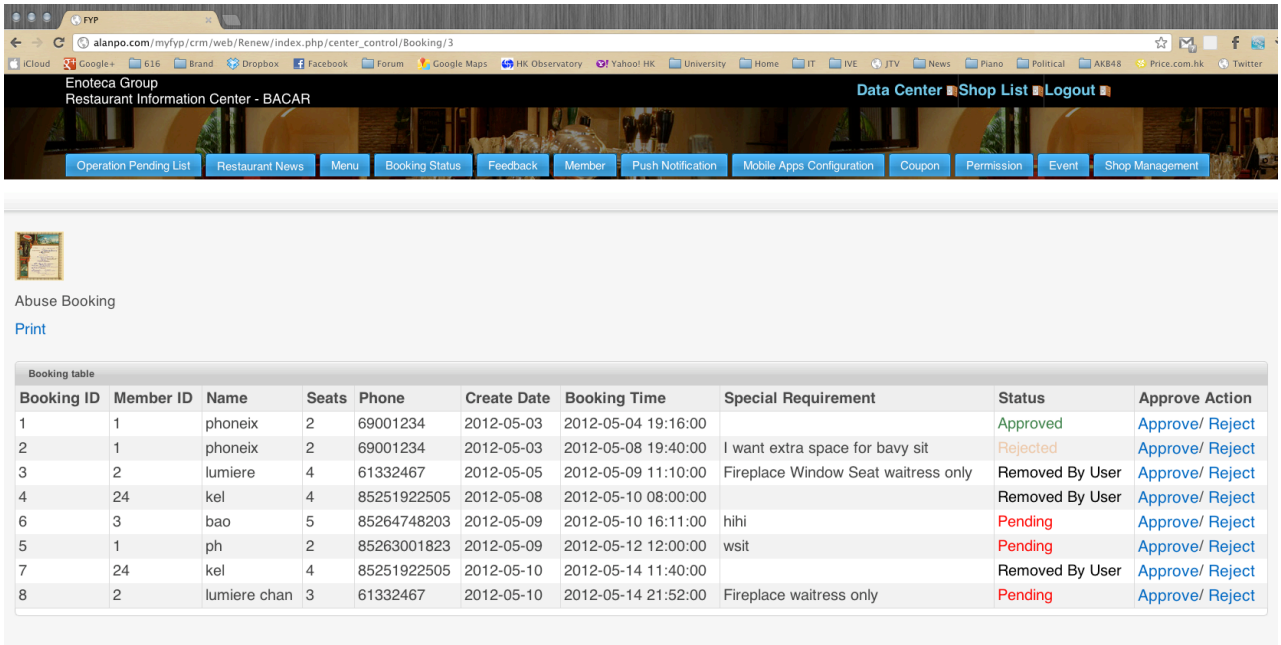


Figure 40. Search function for the E-Menu

- Check/ Approve/ Reject Booking Status

  The Booking Staff can view the booking status of customers in the website. She can reject a booking if the timeslot is not available or other reasons. Or can approve the booking if the system do not approved it automatically. Also, to avoid the customer make booking for a lot of timeslot, the system will put the customer in a abuse booking list if the system think that the customer is abuse. For the restaurant staff easily to control the flow, she can print on the booking list in an A4 paper.



Booking ID	Member ID	Name	Seats	Phone	Create Date	Booking Time	Special Requirement	Status	Approve Action
1	1	phoneix	2	69001234	2012-05-03	2012-05-04 19:16:00		Approved	Approve/ Reject
2	1	phoneix	2	69001234	2012-05-03	2012-05-08 19:40:00	I want extra space for bavy sit	Rejected	Approve/ Reject
3	2	lumiere	4	61332467	2012-05-05	2012-05-09 11:10:00	Fireplace Window Seat waitress only	Removed By User	Approve/ Reject
4	24	kel	4	85251922505	2012-05-08	2012-05-10 08:00:00		Removed By User	Approve/ Reject
6	3	bao	5	85264748203	2012-05-09	2012-05-10 16:11:00	hihi	Pending	Approve/ Reject
5	1	ph	2	85263001823	2012-05-09	2012-05-12 12:00:00	wsit	Pending	Approve/ Reject
7	24	kel	4	85251922505	2012-05-10	2012-05-14 11:40:00		Removed By User	Approve/ Reject
8	2	lumiere chan	3	61332467	2012-05-10	2012-05-14 21:52:00	Fireplace waitress only	Pending	Approve/ Reject

Figure 41. Booking List

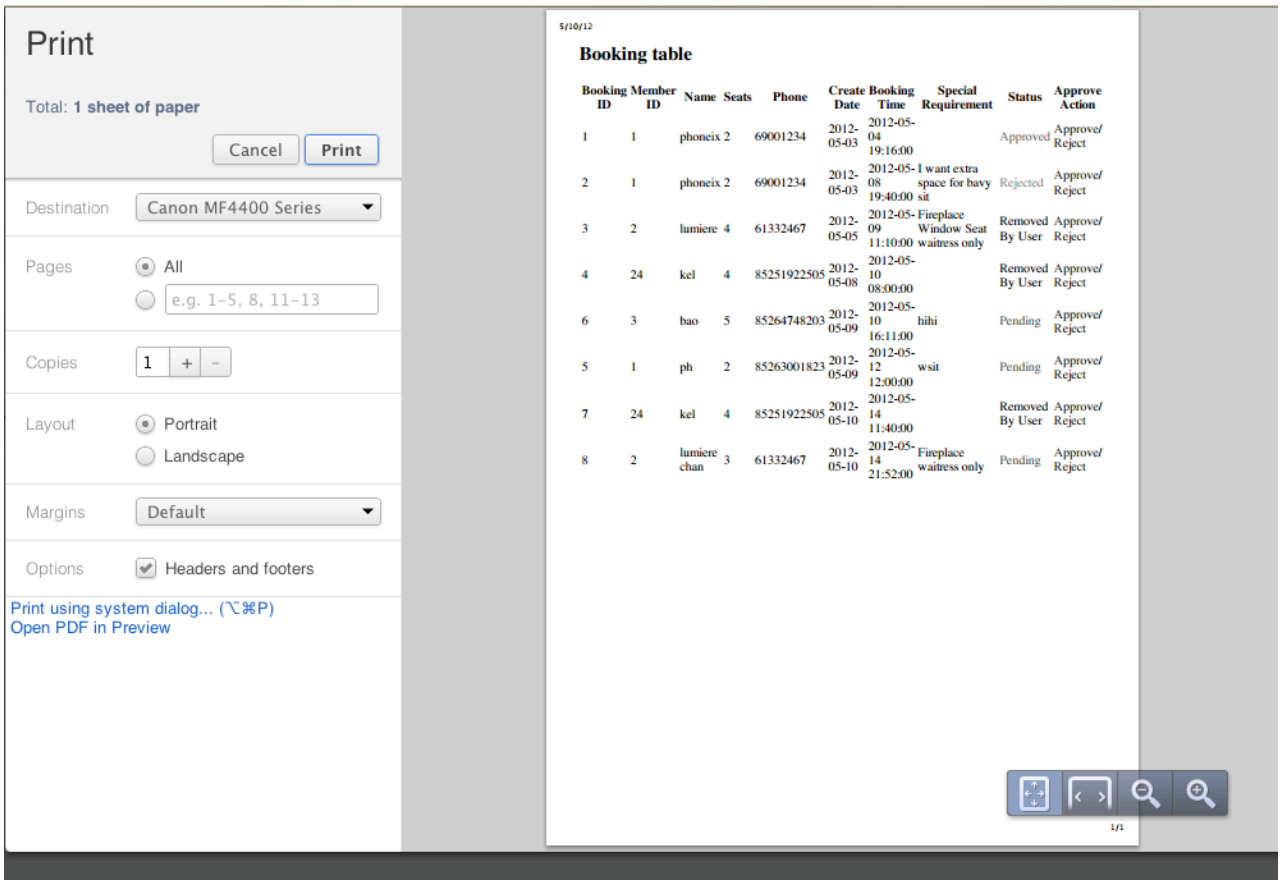


Figure 42. Printing booking list

- Modify feedback form

  The restaurant staff can modify the feedback form for the customer. She can create a new feedback form or delete the old form.

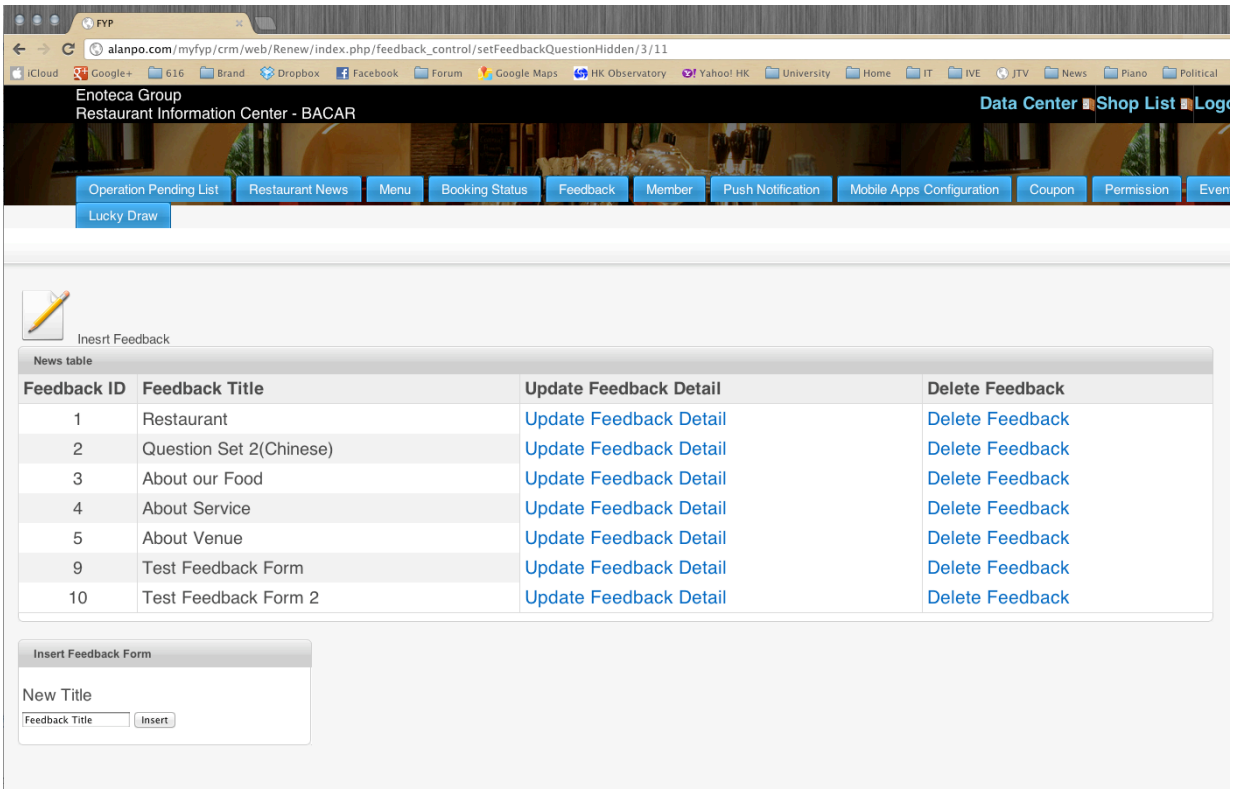


Figure 43. Feedback view

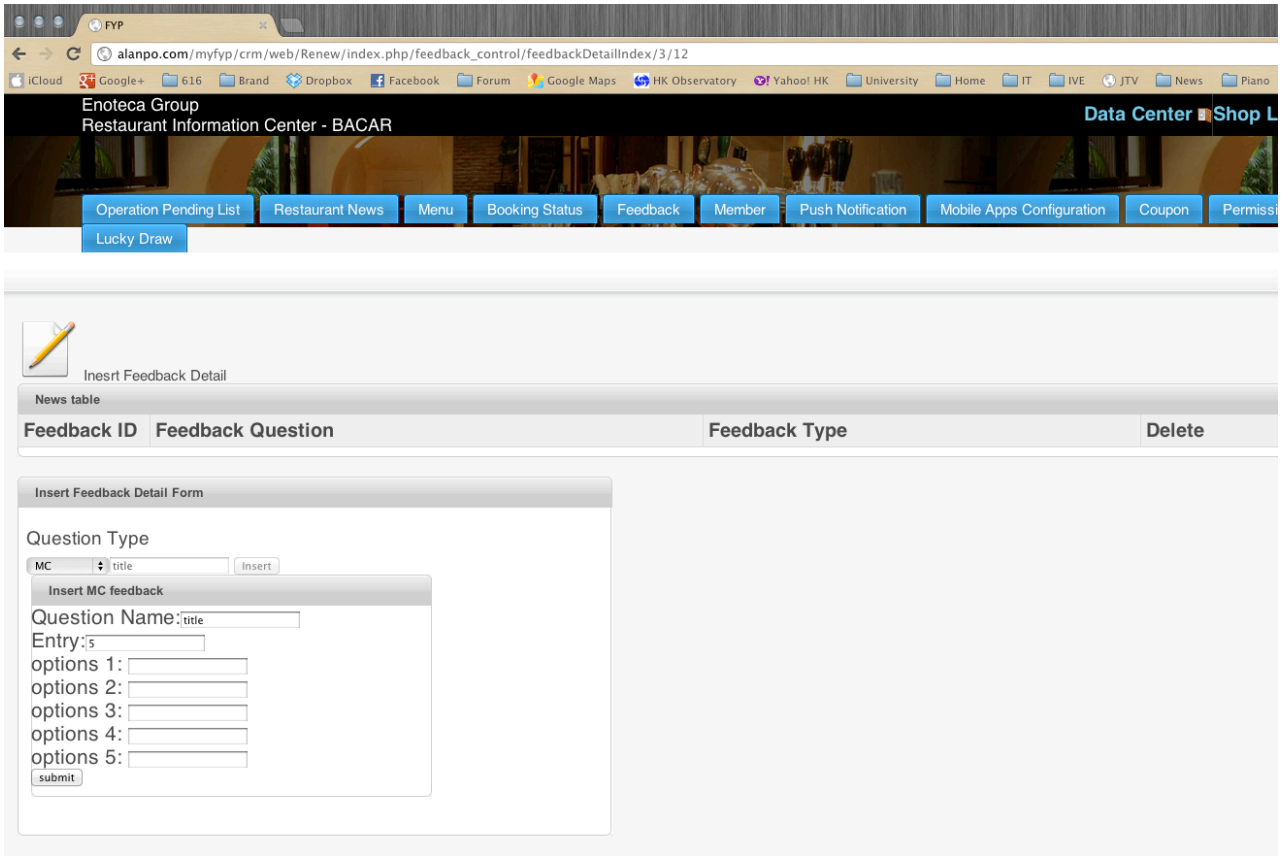


Figure 44. Feedback Form - insert new questions

- App configuration

  The restaurant manager can modify their mobile app in her restaurant, choose the functions in each type of restaurant for the app for the customers, change the theme in different type of restaurant, change the layout in different order of function, change the icon of the functions.

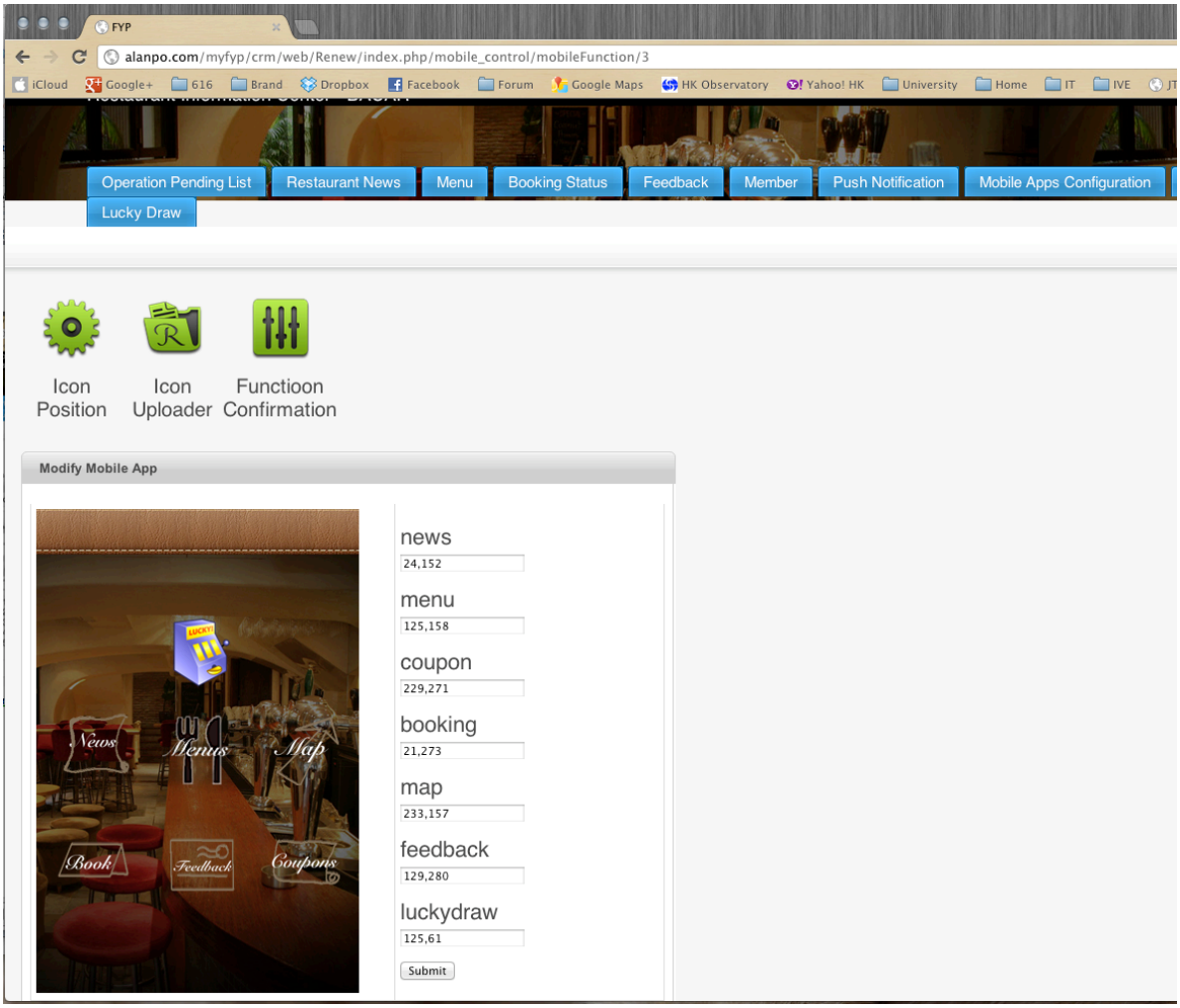


Figure 45. Mobile App Configuration

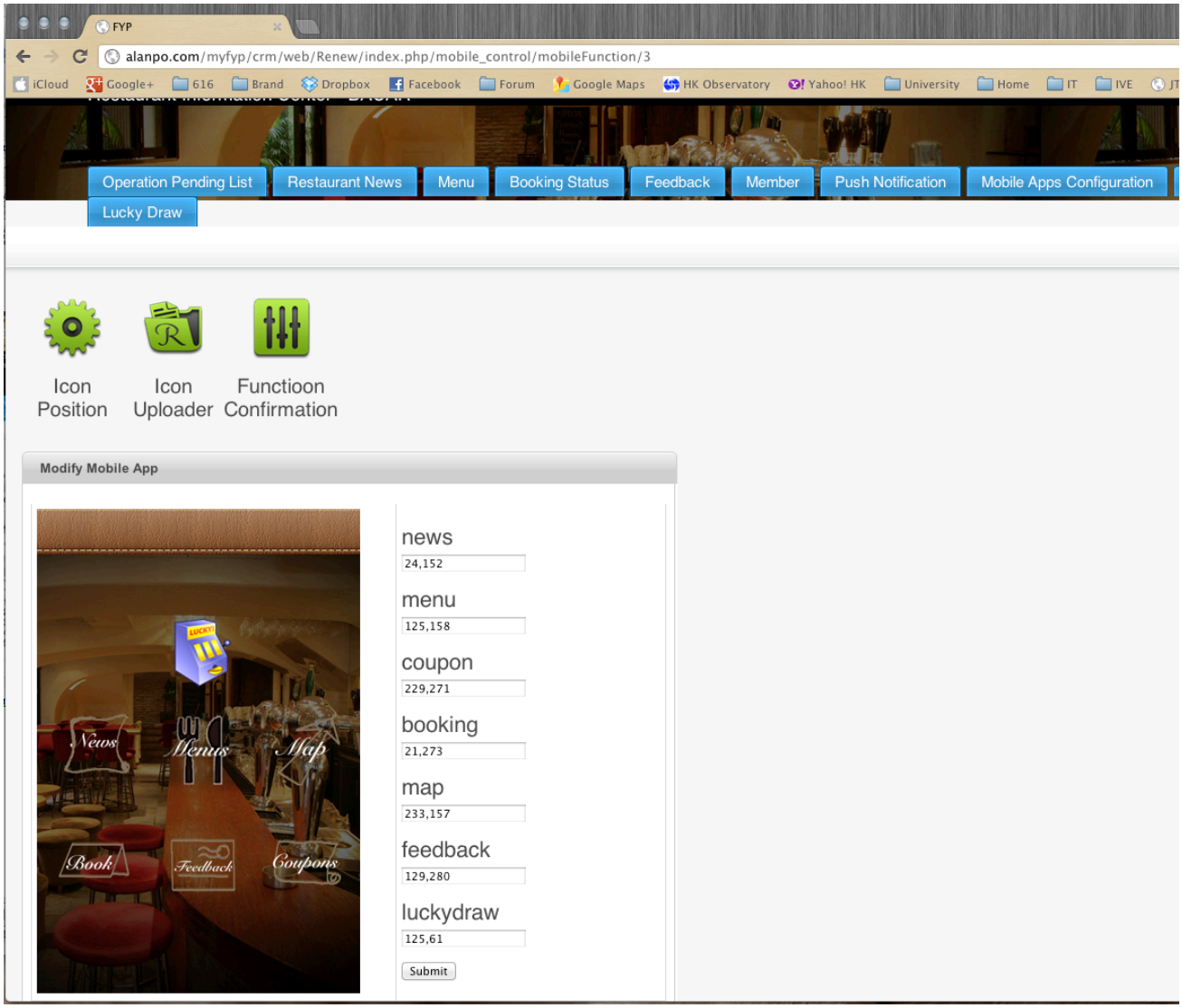


Figure 46 Mobile App icon position

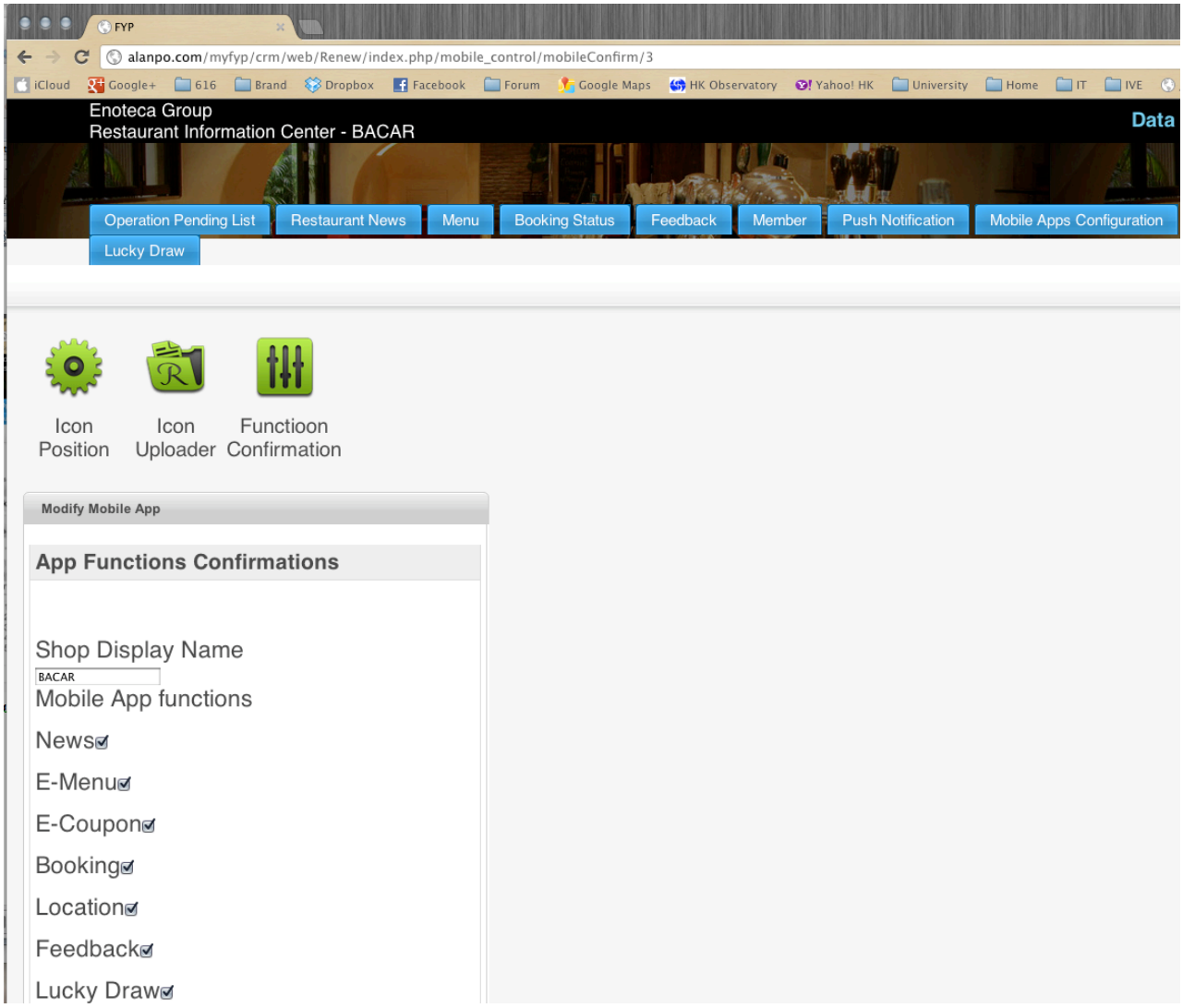


Figure 47 Mobile App function confirmation

- Modify E-Coupon

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

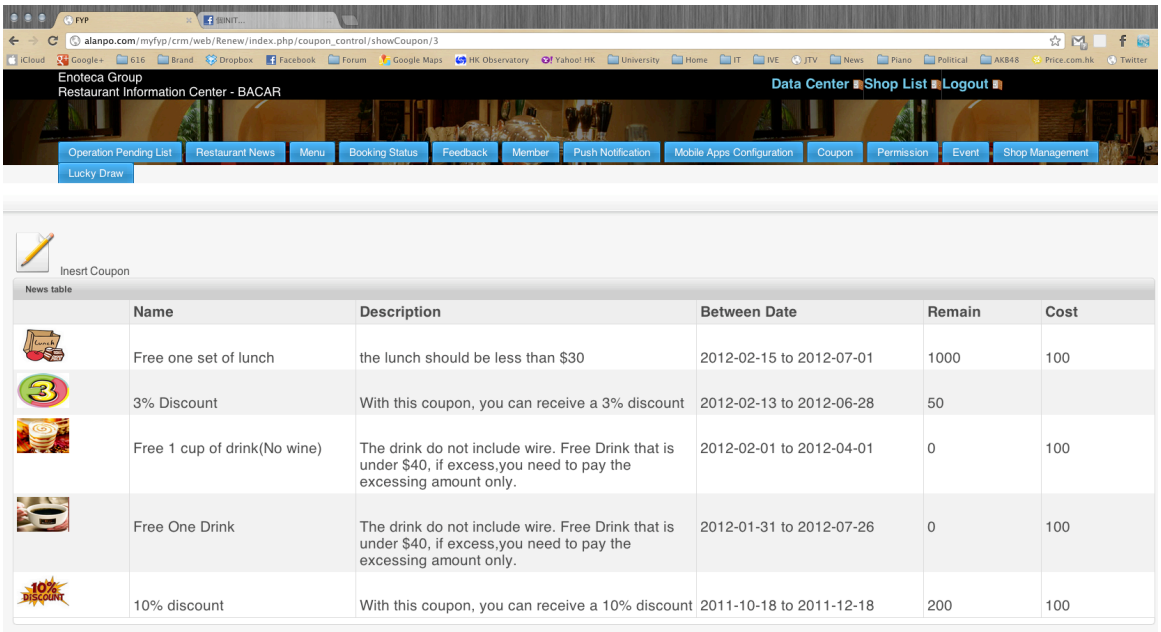


Figure 48. Coupon List

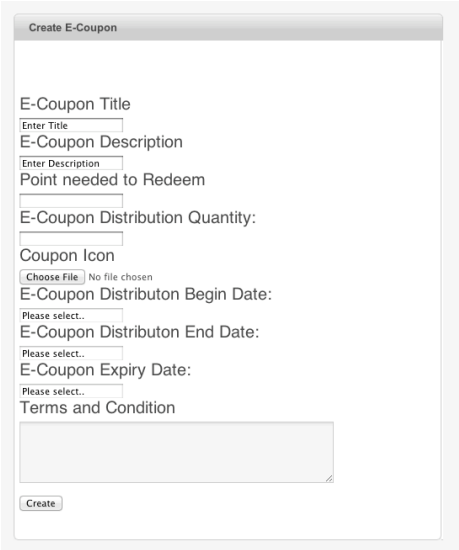


Figure 49. New Coupon

- Modify restaurant user permission

  The enterprise administrator can modify the permissions for different group of people.

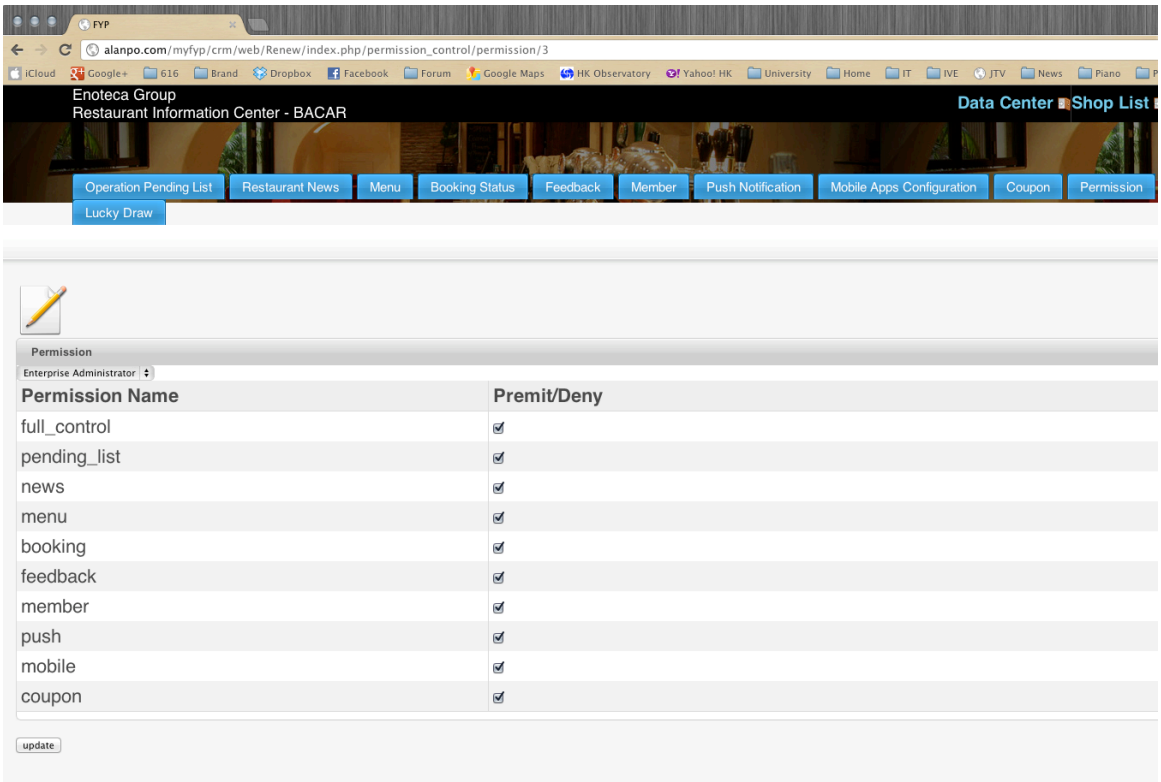


Figure 50. Modify restaurant user permission view

- Event control

  The restaurant manager can set for a special date or period then deny all booking on that day.

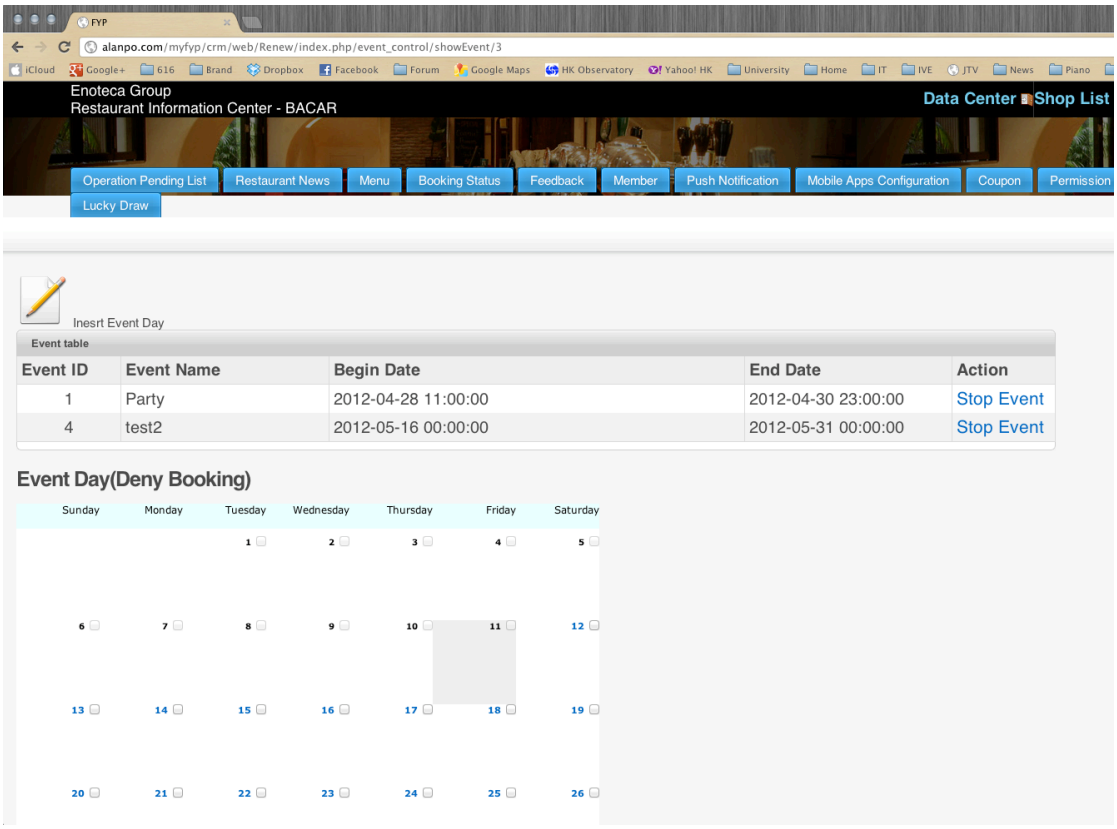


Figure 51. Event control list

- Shop Management

  The restaurant manager can modify the shop information, including the location of the branch shop, the details including the shop name, description, opening hours. Also, we use the Google API for the location service. The user can point to the location on the map easily.

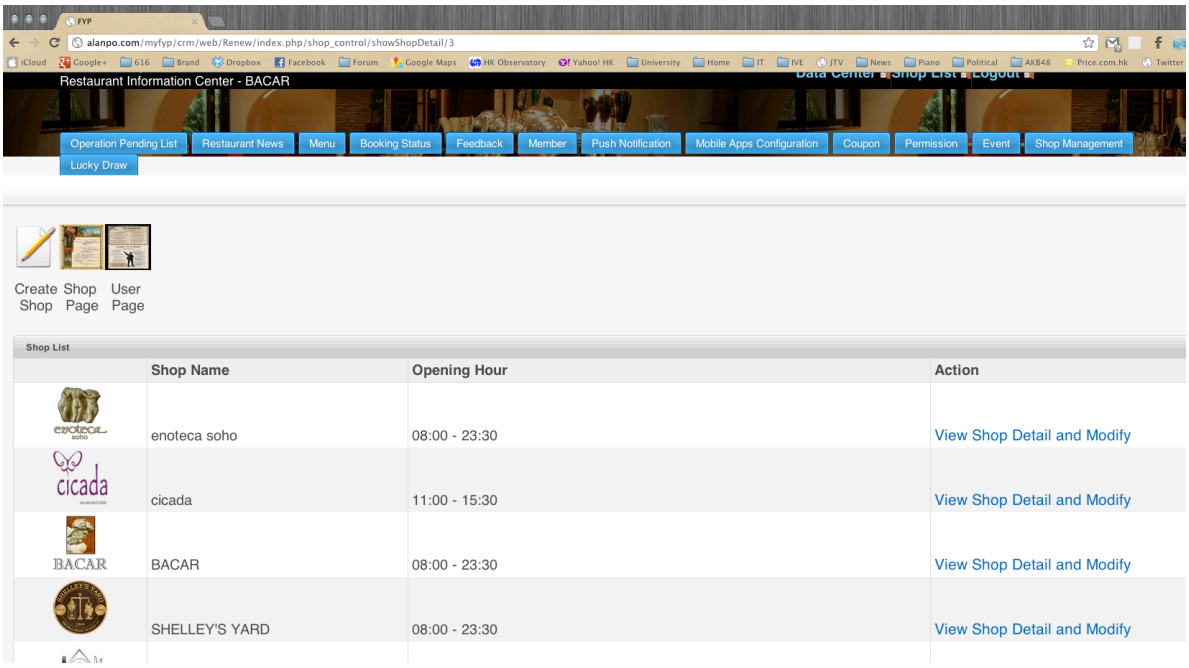


Figure 52. Shop details

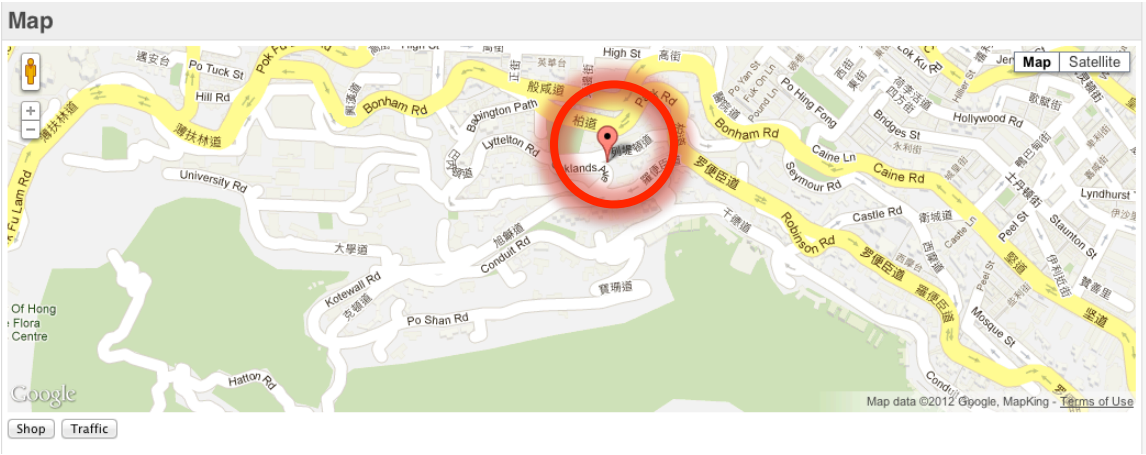


Figure 53. Shop location with Google Map API

- Modify lucky draw data

  The restaurant manager can insert new lucky draw information or modify the information.

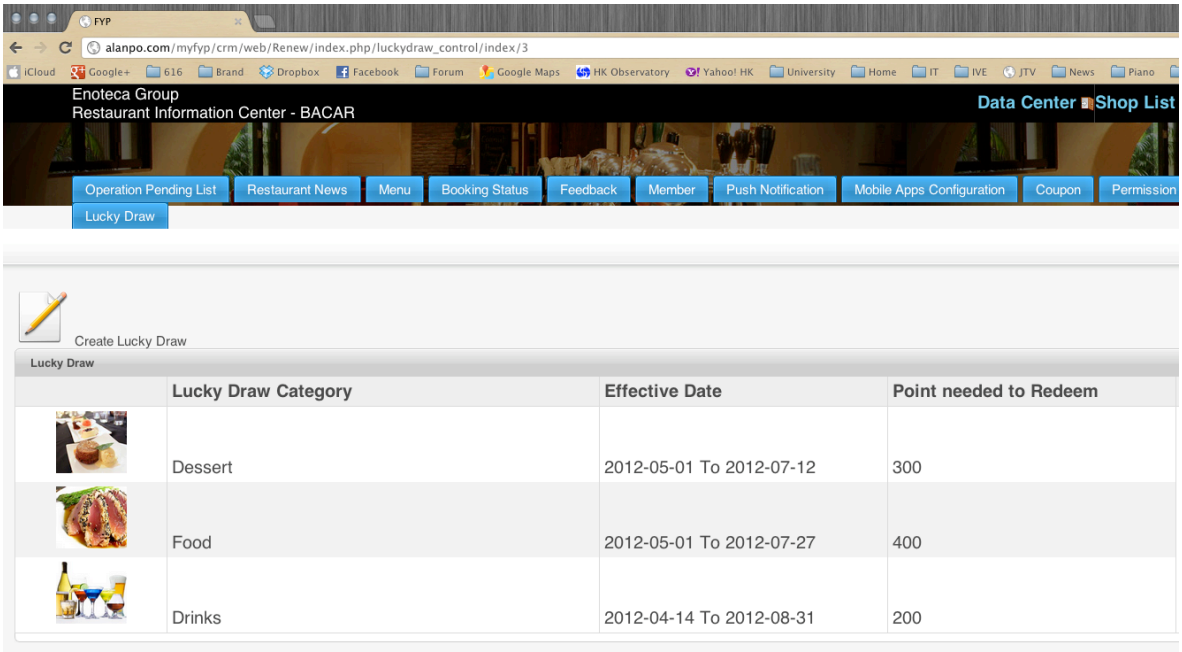


Figure 54. Lucky draw list

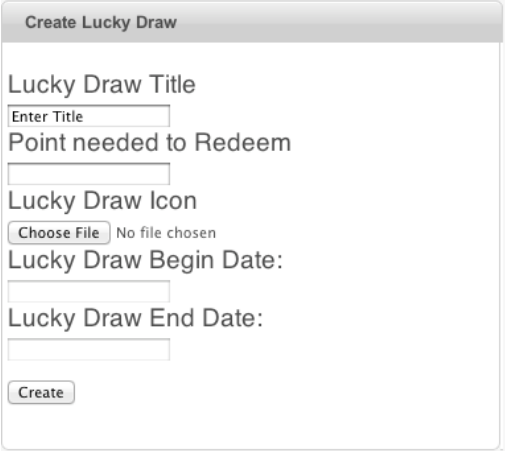


Figure 55. Create new lucky draw

- Data Analysis Center

  The restaurant manager can view the data analysis sub-system. The result can be exported to a .csv file.

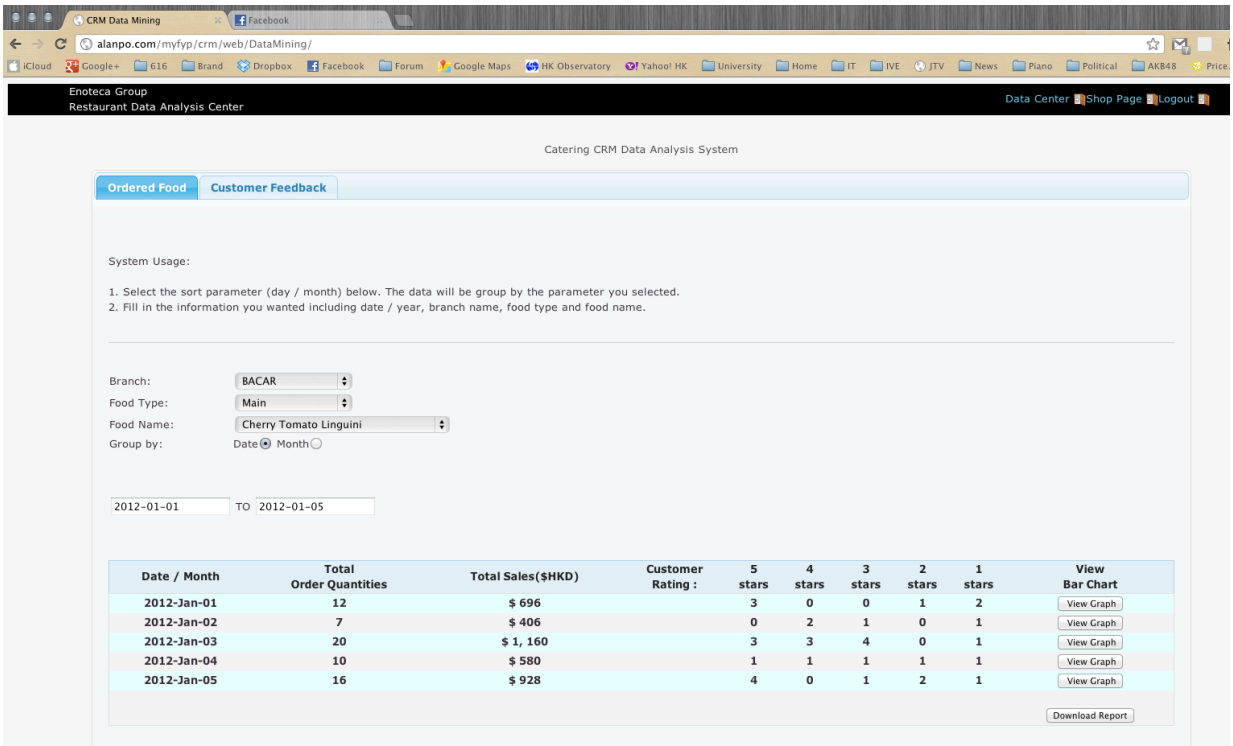


Figure 56. Data analysis center

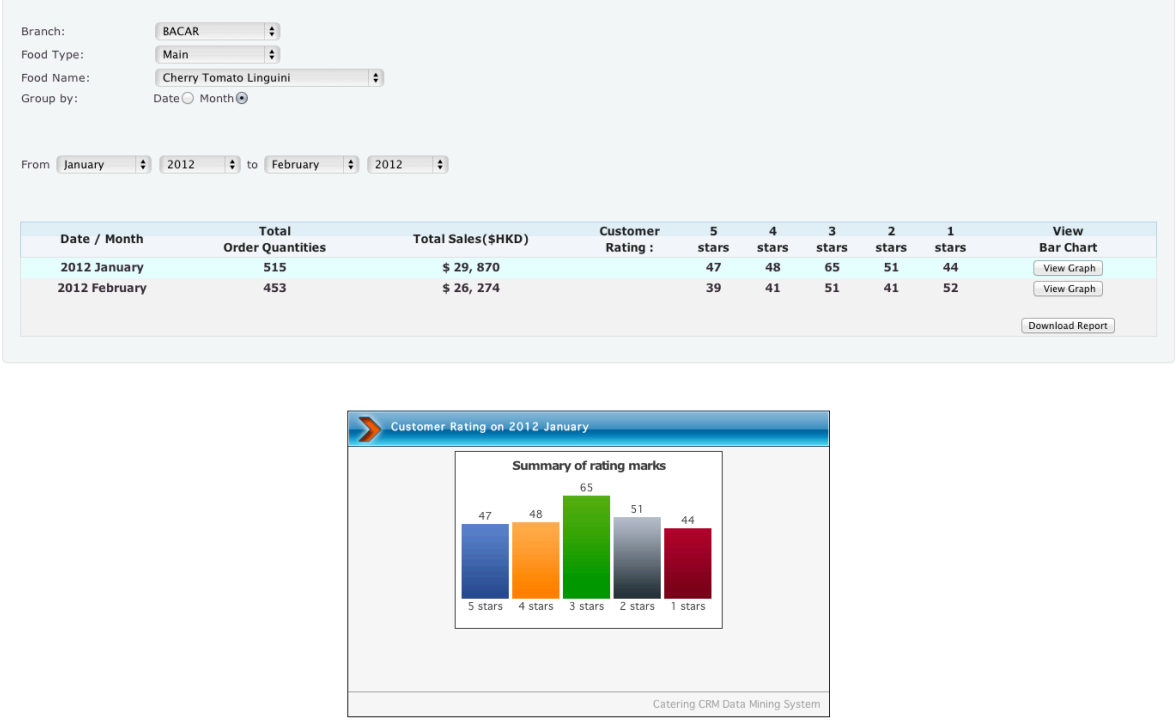


Figure 57. Data analysis with bar chart

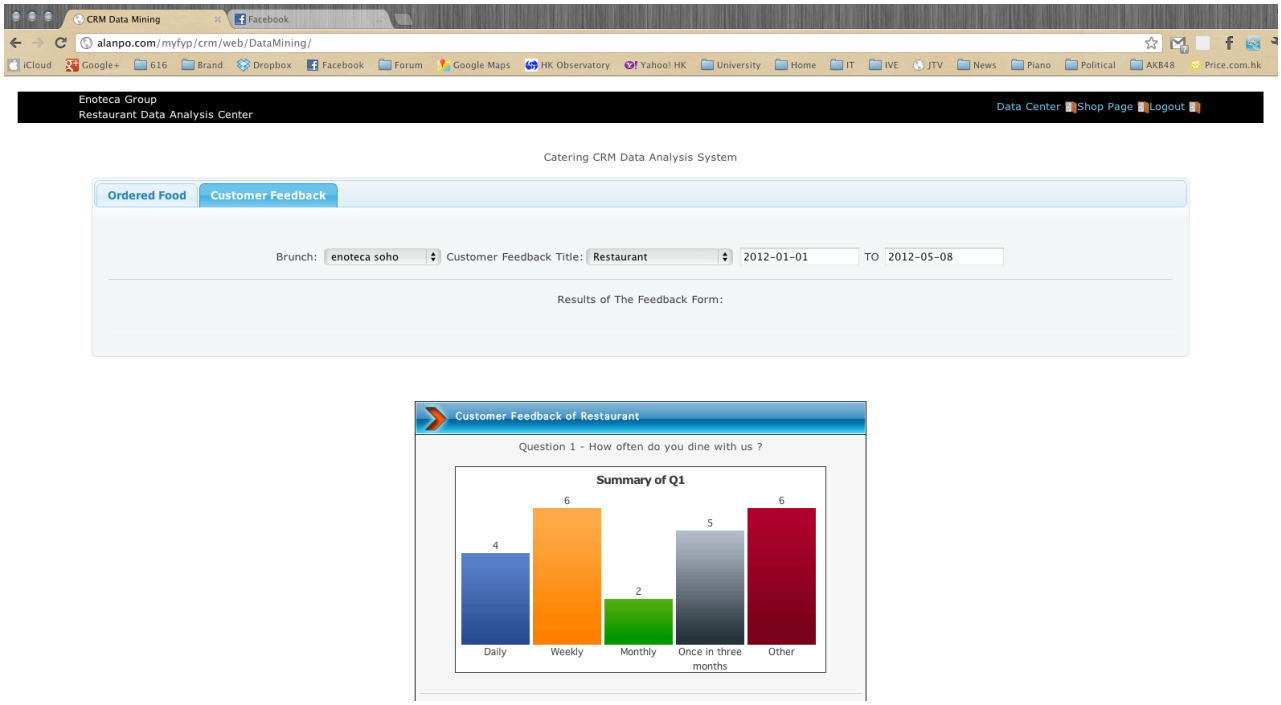


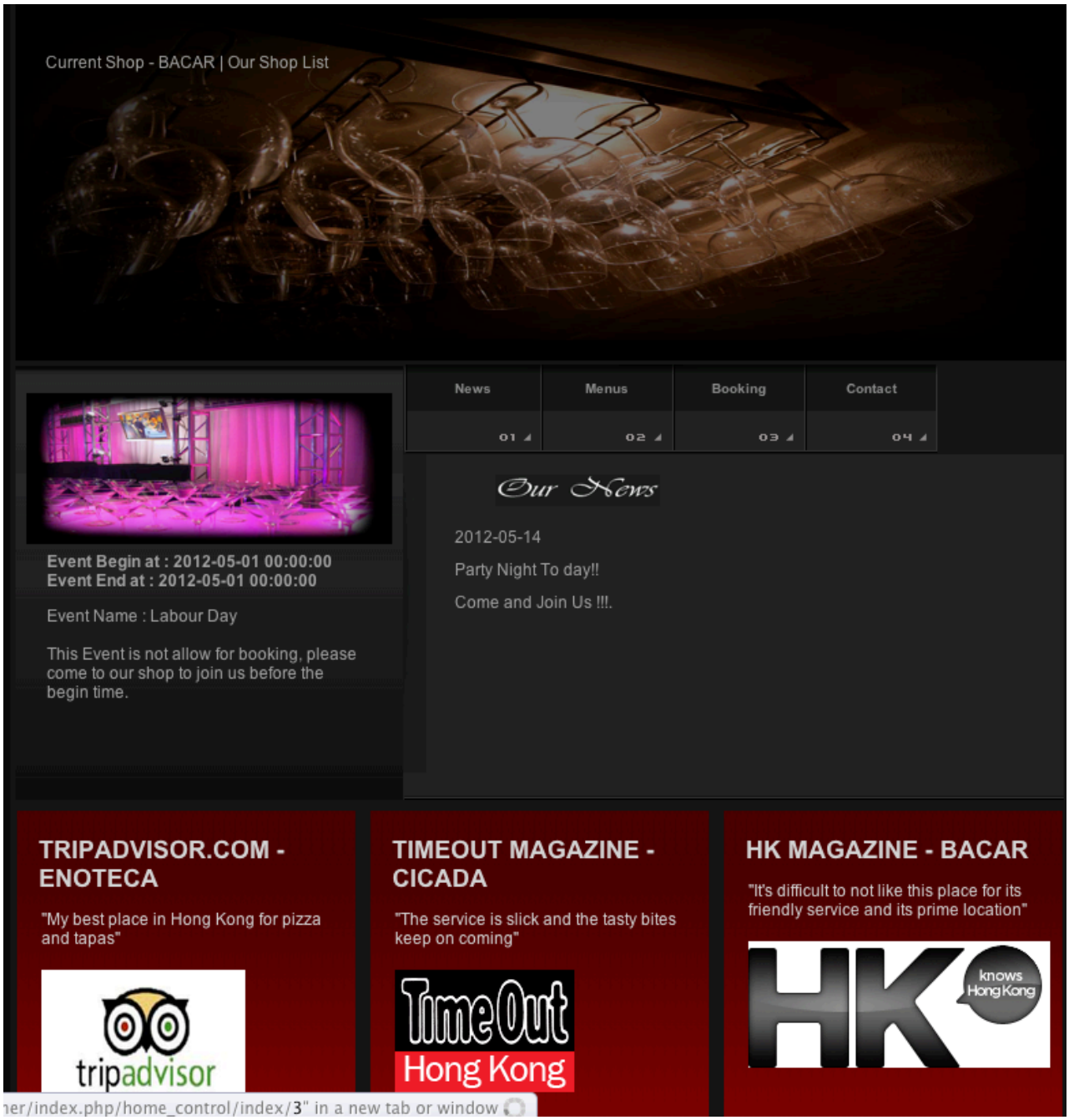
Figure 58. Data analysis of customer feedbacks

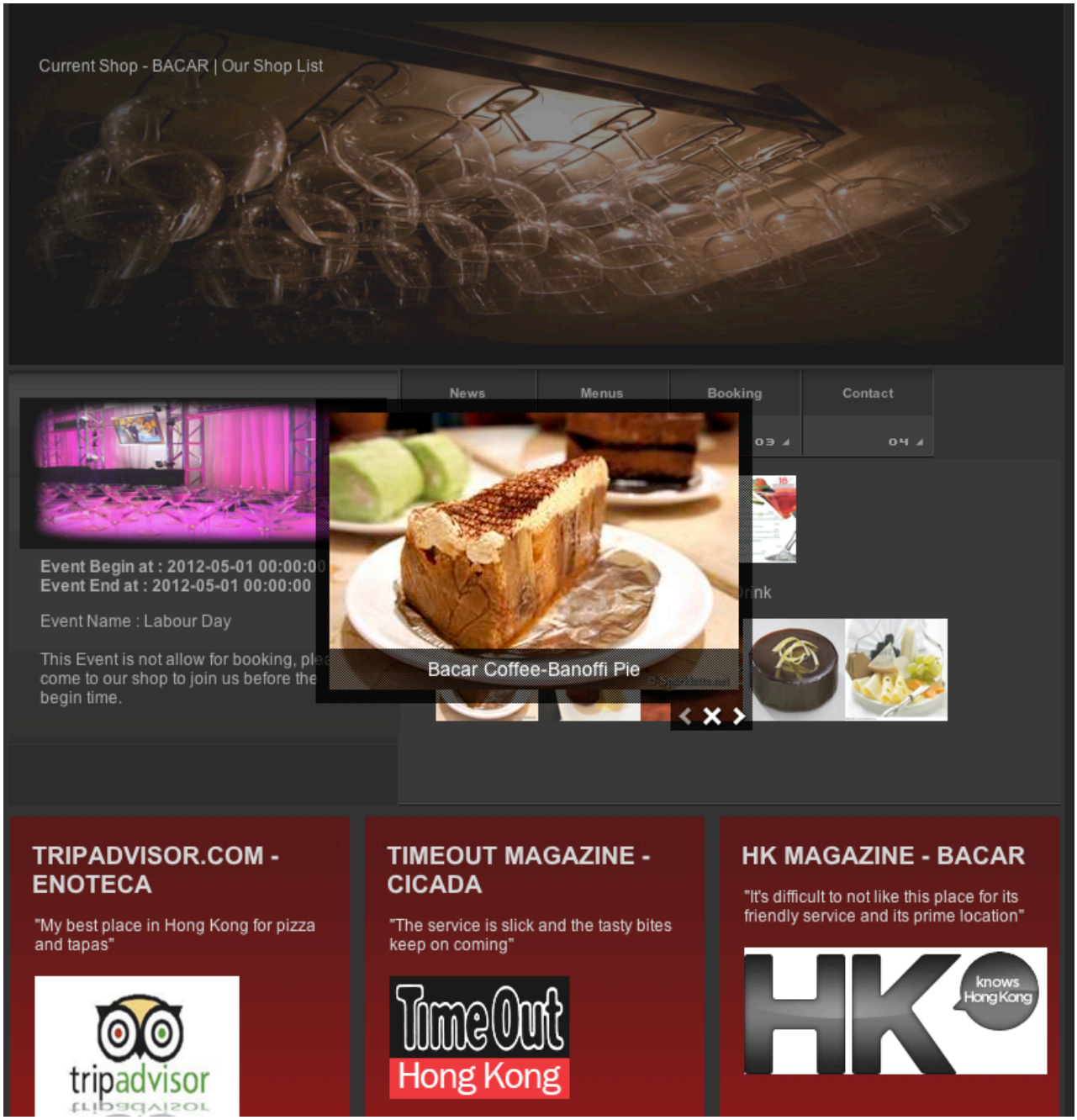
- Evaluation of Data

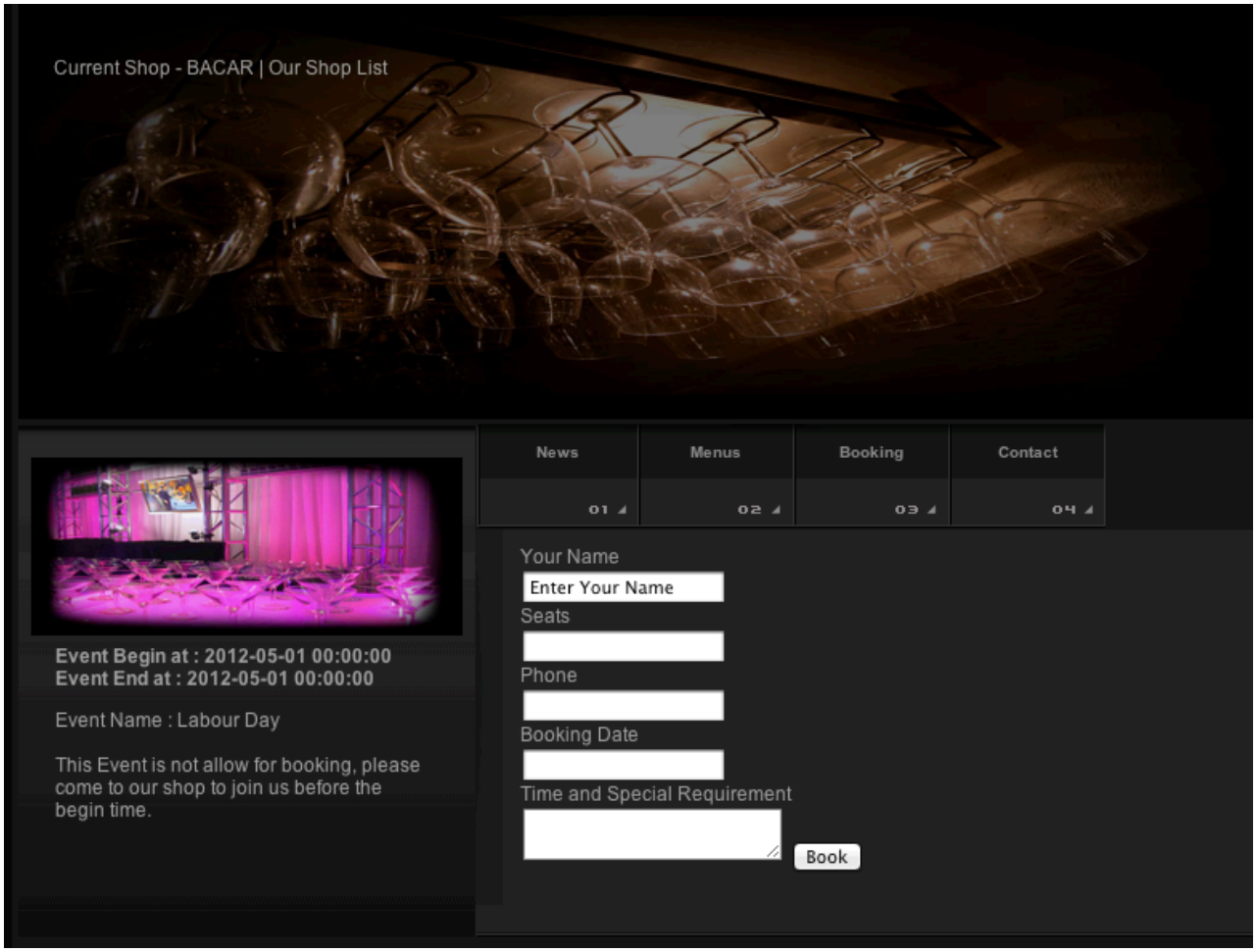
  After data analysis is done, we will periodically evaluate the analyzed data. We will change different method of data analysis to meet your company's need. Data evaluation usually insists of privacy problem, we will protect our analyzed data well.

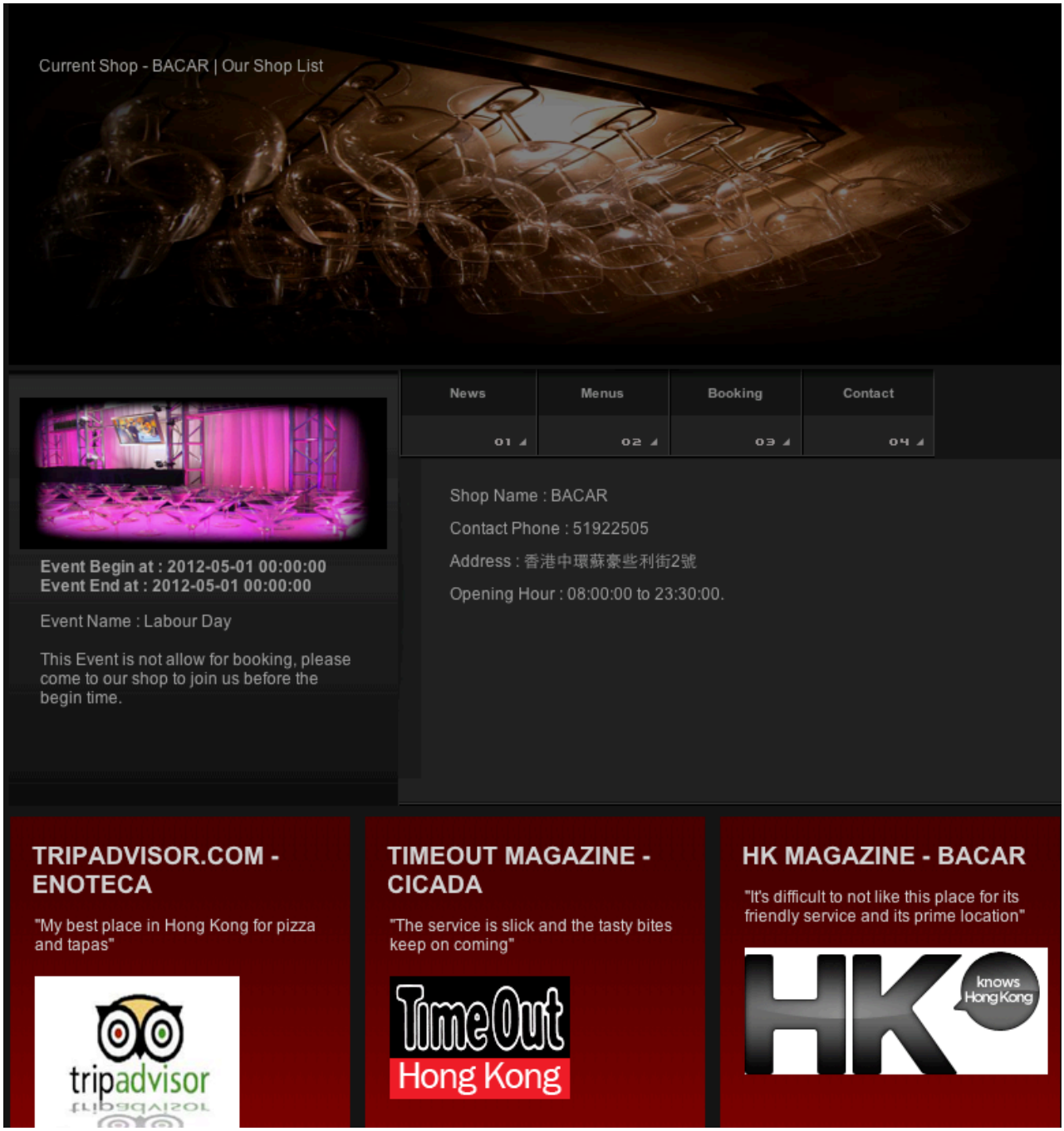
Website for customer

The customer can view some basic information in the Customer Website page http://alanpo.com/myfyp/crm/web/Customer/index.php/index_control




Current Shop - BACAR | Our Shop List
News
Menus
Booking
Contact
Event Begin at : 2012-05-01 00:00:00
Event End at : 2012-05-01 00:00:00
Event Name : Labour Day
This Event is not allow for booking, please come to our shop to join us before the begin time.
Bacar Coffee-Banoffi Pie
TRIPADVISOR.COM - ENOTECA
"My best place in Hong Kong for pizza and tapas"
tripadvisor
TIMEOUT MAGAZINE - CICADA
"The service is slick and the tasty bites keep on coming"
Time Out Hong Kong
HK MAGAZINE - BACAR
"It's difficult to not like this place for its friendly service and its prime location"
HK knows Hong Kong


Current Shop - BACAR | Our Shop List
News
Menus
Booking
Contact
01
02
03
04
Event Begin at : 2012-05-01 00:00:00
Event End at : 2012-05-01 00:00:00
Event Name : Labour Day
This Event is not allow for booking, please come to our shop to join us before the begin time.
Your Name
Enter Your Name
Seats
Phone
Booking Date
Time and Special Requirement
Book


Current Shop - BACAR | Our Shop List
News
Menus
Booking
Contact
01
02
03
04
Event Begin at : 2012-05-01 00:00:00
Event End at : 2012-05-01 00:00:00
Event Name : Labour Day
This Event is not allow for booking, please come to our shop to join us before the begin time.
Shop Name : BACAR
Contact Phone : 51922505
Address : 香港中環蘇豪些利街2號
Opening Hour : 08:00:00 to 23:30:00.
TRIPADVISOR.COM - ENOTECA
"My best place in Hong Kong for pizza and tapas"
tripadvisor
TIMEOUT MAGAZINE - CICADA
"The service is slick and the tasty bites keep on coming"
Time Out
Hong Kong
HK MAGAZINE - BACAR
"It's difficult to not like this place for its friendly service and its prime location"
HK
knows Hong Kong

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 support multi restaurant

When the customer uses our mobile app, she needs to select restaurant. For a group of restaurant in the same group, they will be displayed with the icon and the name of the restaurant. Each restaurant can have different functions and information inside. And all the data can be modify in the website.

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 using simple English or Tradition Chinese.

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.

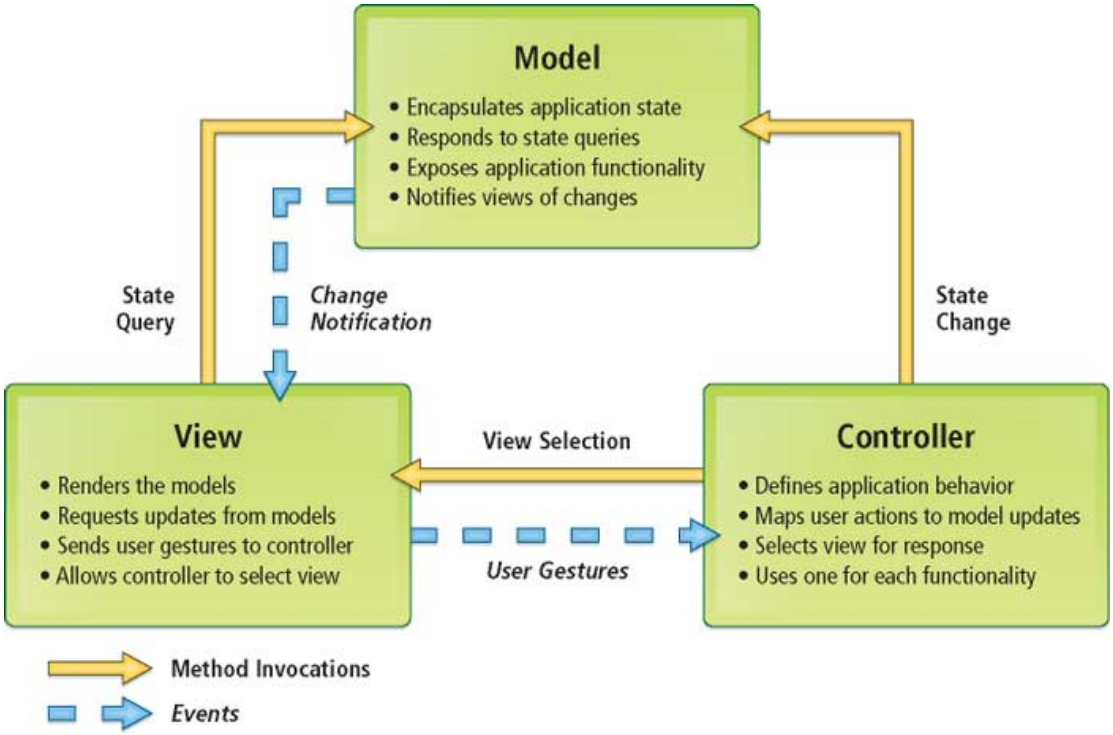


Figure 59. Model View Controller

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: 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. Bar Chart will also provide to the staff for the improvement in the restaurant to gain more profit.

Description of hardware requirements

- Server side

Motherboard:	Intel Socket LGA 775/ LGA1156 Motherboard
CPU:	Intel Core 2 Duo 2.6Ghz or above
RAM:	4GB or above
DVD-ROM:	at least DVD-Read
Hard Disk:	500GB or above
Network:	1Gbps Ethernet
Power Supply:	500W 24-hrs Power Supply

- Client side

Apple iPhone / iPad / iPod Touch with stable network connection
or
Android Mobile with Google Android 2.3.3 or above

Non-Functional Requirement

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.

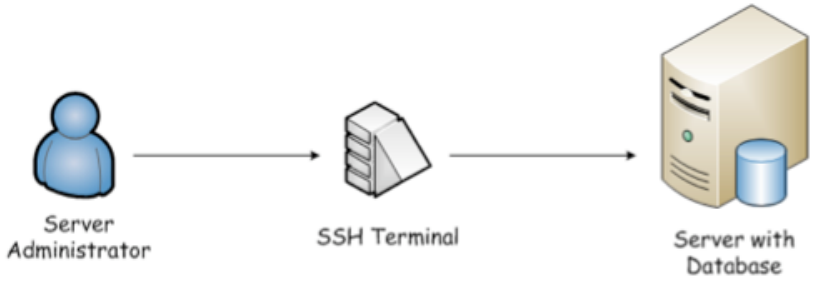


Figure 60. Server Administrator

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.

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.

Besides, we uses challenge and response to control the user login on the website. Each time the user login with her password the system will generate a random number back to the user and encrypt the password with that random number and then send it to the server, the server will change the string is valid or not with the random number and the password.

Challenge-response authentication is a family of protocols in which one party presents a question ("challenge") and another party must provide a valid answer ("response") to be authenticated. The simplest example of a challenge-response protocol is password authentication, where the challenge is asking for the password and the valid response is the correct password. Clearly an adversary that can eavesdrop on a password authentication can then authenticate itself in the same way. One solution is to issue multiple passwords, each of them marked with an identifier. The verifier can pick any of the identifiers, and the prover must have the correct password for that identifier. Assuming that the passwords are chosen independently, an adversary who intercepts one challenge-response message pair has no more chance of responding correctly to a different challenge than an adversary who has intercepted nothing.

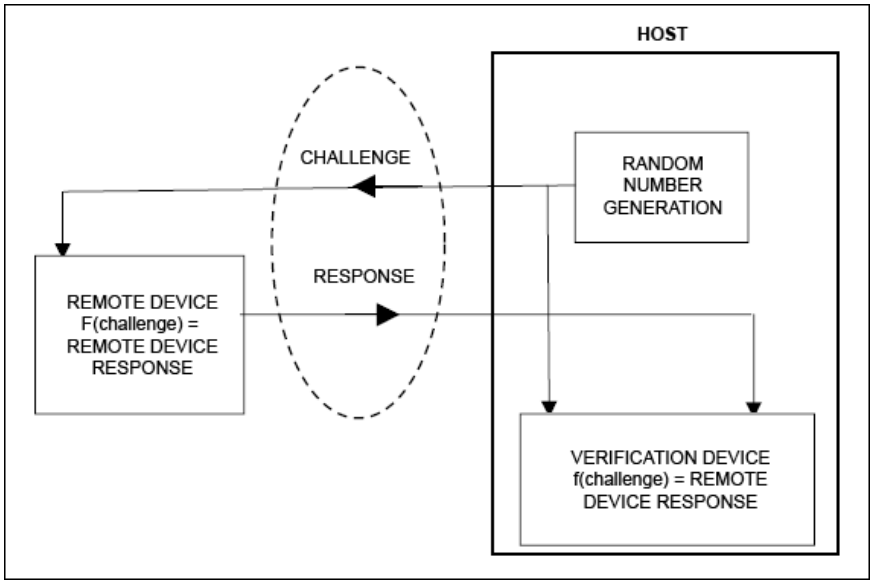


Figure 61. Website challenge and response login

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 and system 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.

CONSTRAINTS

Network stability problem

For our projects, we use a public domain http://www.alanpo.com but a big problem is the connection problems. Sometimes, the network connect was too slow, the mobile app need a few time to update the information to keep the data consistency.

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

Limitation of iPhone API

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

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

Objective-C implementations use a thin runtime system written in C, which adds little to the size of the application. In contrast, most object-oriented systems at the time that it was created used large virtual machine runtimes. Programs written in Objective-C tend to be not much larger than the size of their code and that of the libraries (which generally do not need to be included in the software distribution), in contrast to Smalltalk systems where a large amount of memory was used just to open a window. Objective-C applications tend to be larger than similar C or C++ applications because Objective-C dynamic typing does not allow methods to be stripped or inlined. Since the programmer has such freedom to delegate, forward calls, build selectors on the fly and pass them to

the runtime system, the Objective-C compiler cannot assume it's safe to remove unused methods or to inline calls.

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

Limitation of Android API

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

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

Android consists of a kernel based on the Linux kernel, with middleware, libraries and APIs written in C and application software running on an application framework, which includes Java-compatible libraries, based on Apache Harmony. Android uses the Dalvik virtual machine with just-in-time compilation to run Dalvik dex-code (Dalvik Executable), which is usually translated from Java bytecode. Android has a large community of developers writing applications ("apps") that extend the functionality of the devices. Developers write primarily in a customized version of Java.

While most Android applications are written in Java, there is no Java Virtual Machine in the platform and Java byte code is not executed. Java classes are compiled into Dalvik executable and run on Dalvik, a specialized virtual machine designed specifically for

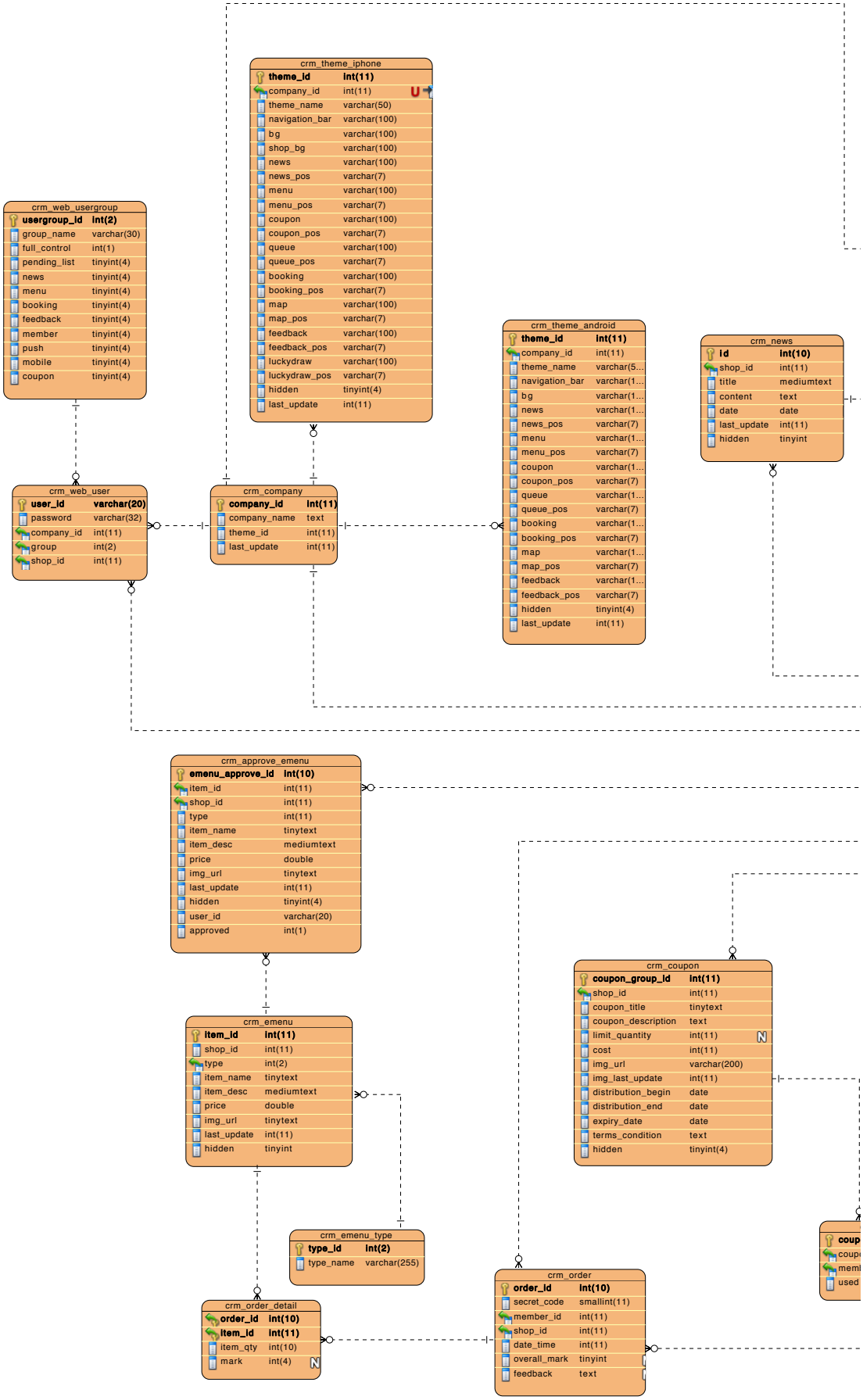
Android and optimized for battery-powered mobile devices with limited memory and CPU. J2ME support can be provided via third-party applications.

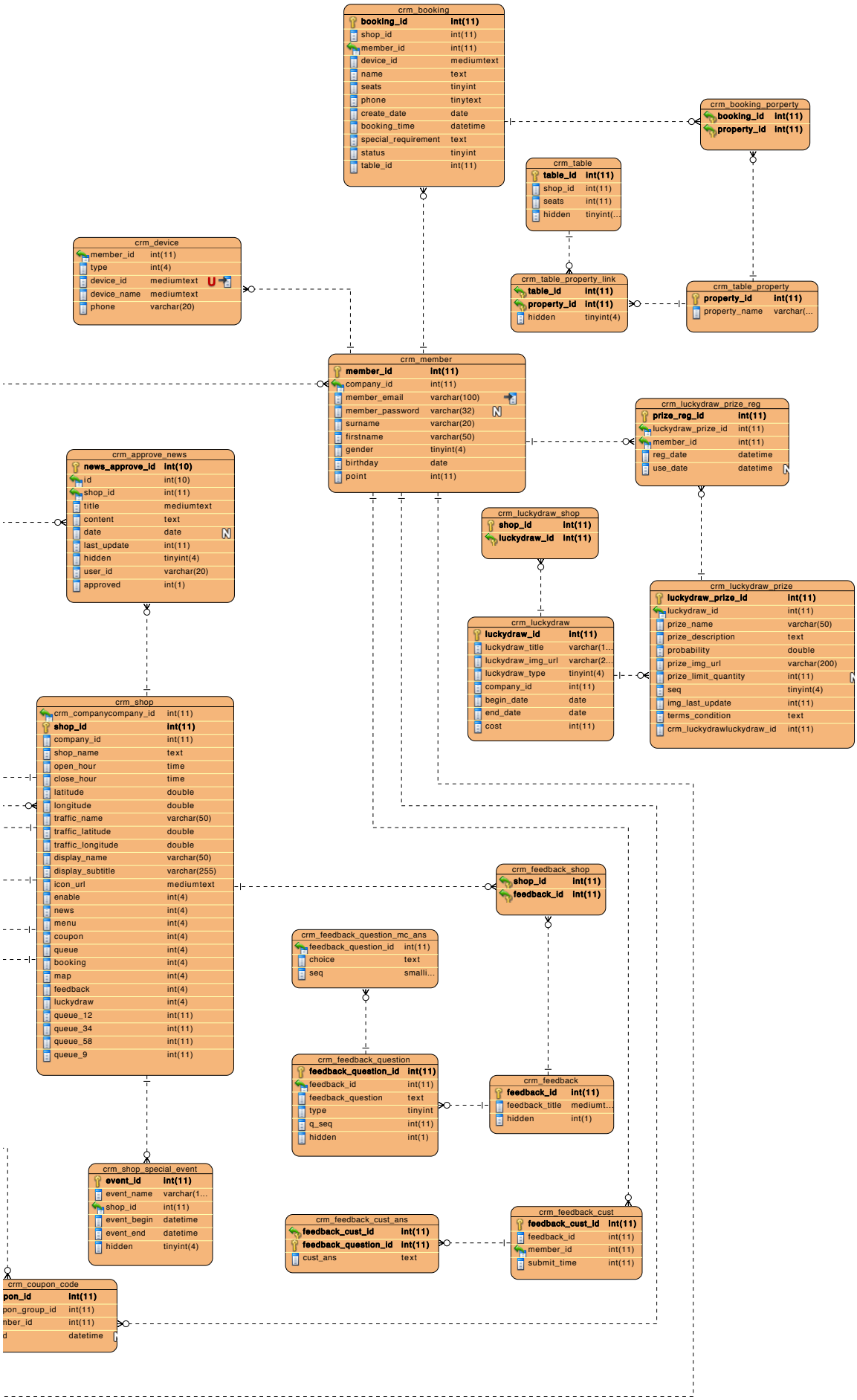
Android do not provide any server to do the push notification service. It is totally different to Apple. If we want to do push notification, we need to build up the server and service ourselves. Also, the API level is not same in different devices; we need to use a common API level to support all the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

DOCUMENTATION FOR PROBLEM ANALYSIS

UML MODELING

Entity-Relationship Diagram




crm_booking
booking_id int(11)
shop_id int(11)
member_id int(11)
device_id mediumtext
name text
seats tinyint
phone tinytext
create_date date
booking_time datetime
special_requirement text
status tinyint
table_id int(11)
crm_booking_porperty
booking_id int(11)
property_id int(11)
crm_table
table_id int(11)
shop_id int(11)
seats int(11)
hidden tinyint(...
crm_device
member_id int(11)
type int(4)
device_id mediumtext
device_name mediumtext
phone varchar(20)
crm_table_property_link
table_id int(11)
property_id int(11)
hidden tinyint(4)
crm_table_property
property_id int(11)
property_name varchar(...
crm_member
member_id int(11)
company_id int(11)
member_email varchar(100)
member_password varchar(32)
surname varchar(20)
firstname varchar(50)
gender tinyint(4)
birthday date
point int(11)
crm_luckydraw_prize_reg
prize_reg_id int(11)
luckydraw_prize_id int(11)
member_id int(11)
reg_date datetime
use_date datetime
crm_approve_news
news_approve_id int(10)
id int(10)
shop_id int(11)
title mediumtext
content text
date date
last_update int(11)
hidden tinyint(4)
user_id varchar(20)
approved int(1)
crm_luckydraw_shop
shop_id int(11)
luckydraw_id int(11)
crm_luckydraw_prize
luckydraw_prize_id int(11)
luckydraw_id int(11)
prize_name varchar(50)
prize_description text
probability double
prize_img_url varchar(200)
prize_limit_quantity int(11)
seq tinyint(4)
img_last_update int(11)
terms_condition text
crm_luckydrawluckydraw_id int(11)
crm_luckydraw
luckydraw_id int(11)
luckydraw_title varchar(1...
luckydraw_img_url varchar(2...
luckydraw_type tinyint(4)
company_id int(11)
begin_date date
end_date date
cost int(11)
crm_shop
crm_companycompany_id int(11)
shop_id int(11)
company_id int(11)
shop_name text
open_hour time
close_hour time
latitude double
longitude double
traffic_name varchar(50)
traffic_latitude double
traffic_longitude double
display_name varchar(50)
display_subtitle varchar(255)
icon_url mediumtext
enable int(4)
news int(4)
menu int(4)
coupon int(4)
queue int(4)
booking int(4)
map int(4)
feedback int(4)
luckydraw int(4)
queue_12 int(11)
queue_34 int(11)
queue_58 int(11)
queue_9 int(11)
crm_feedback_shop
shop_id int(11)
feedback_id int(11)
crm_feedback_question_mc_ans
feedback_question_id int(11)
choice text
seq smalli...
crm_feedback_question
feedback_question_id int(11)
feedback_id int(11)
feedback_question text
type tinyint
q_seq int(11)
hidden int(1)
crm_feedback
feedback_id int(11)
feedback_title mediumt...
hidden int(1)
crm_shop_special_event
event_id int(11)
event_name varchar(1...
shop_id int(11)
event_begin datetime
event_end datetime
hidden tinyint(4)
crm_feedback_cust_ans
feedback_cust_id int(11)
feedback_question_id int(11)
cust_ans text
crm_feedback_cust
feedback_cust_id int(11)
feedback_id int(11)
member_id int(11)
submit_time int(11)
crm_coupon_code
pon_id int(11)
pon_group_id int(11)
nber_id int(11)
d datetime

Data Dictionary

Table: crm_approve_emenu

Field Name	Type	Key**	Extra Info
emenu_approve_id	INT(10)	PK	AUTO_INCREMENT
item_id	INT(11)		
shop_id	INT(11)		
type	INT(11)		
item_name	TINYTEXT		
item_desc	VARCHAR(255)		
price	DOUBLE		
img_url	TINYTEXT		
last_upDATE	INT(11)		
hidden	INT(4)		
user_id	VARCHAR(20)		
approved	INT(1)		

** PK = PRIMARY KEY

Figure 62. Table structure crm_approve_emenu

Table: crm_approve_news

Field Name	Type	Key**	Extra Info
news_approve_id	INT(10)	PK	AUTO_INCREMENT
id	INT(10)		
shop_id	INT(11)		
title	VARCHAR(255)		
content	TEXT		
DATE	DATE		
last_upDATE	INT(11)		
hidden	INT(4)		
user_id	VARCHAR(20)		
approved	INT(1)		

** PK = PRIMARY KEY

Figure 63. Table structure crm_approve_news

Table: crm_booking

Field Name	Type	Key**	Extra Info
booking_id	INT(11)	PK	AUTO_INCREMENT
shop_id	INT(11)		
member_id	INT(11)		
device_id	VARCHAR(255)		
name	TEXT		
seats	INT(4)		
phone	TINYTEXT		
create_DATE	DATE		
booking_time	DATETIME		
special_requirement	TEXT		
status	INT(4)		
table_id	INT(11)		

** PK = PRIMARY KEY

Figure 64. Table structure crm_booking

Table: crm_booking_property

Field Name	Type	Key**	Extra Info
booking_id	INT(11)	PK	
property_id	INT(11)	PK	

** PK = PRIMARY KEY

Figure 65. Table structure crm_booking_property

Table: crm_company

Field Name	Type	Key**	Extra Info
company_id	INT(11)	PK	AUTO_INCREMENT
company_name	TEXT		
theme_id	INT(11)		
last_upDATE	INT(11)		

** PK = PRIMARY KEY

Figure 66. Table structure crm_company

Table: crm_coupon

Field Name	Type	Key**	Extra Info
coupon_group_id	INT(11)	PK	AUTO_INCREMENT
shop_id	INT(11)		
coupon_title	TINYTEXT		
coupon_description	TEXT		
limit_quantity	INT(11)		
cost	INT(11)		
img_url	VARCHAR(200)		
img_last_upDATE	INT(11)		
distribution_begin	DATE		
distribution_end	DATE		
expiry_DATE	DATE		
terms_condition	TEXT		
hidden	INT(4)		

** PK = PRIMARY KEY

Figure 67. Table structure crm_coupon

Table: crm_coupon_code

Field Name	Type	Key**	Extra Info
coupon_id	INT(11)	PK	AUTO_INCREMENT
coupon_group_id	INT(11)		
member_id	INT(11)		
used	DATETIME		

** PK = PRIMARY KEY

Figure 68. Table structure crm_coupon_code

Table: crm_device

Field Name	Type	Key**	Extra Info
member_id	INT(11)		
type	INT(4)		
device_id	VARCHAR(255)	UNI	
device_name	VARCHAR(255)		
phone	VARCHAR(20)		

** PK = PRIMARY KEY

Figure 69. Table structure crm_device

Table: crm_emenu

Field Name	Type	Key**	Extra Info
item_id	INT(11)	PK	AUTO_INCREMENT
shop_id	INT(11)		
type	INT(11)		
item_name	TINYTEXT		
item_desc	VARCHAR(255)		
price	DOUBLE		
img_url	TINYTEXT		
last_upDATE	INT(11)		
hidden	INT(4)		

** PK = PRIMARY KEY

Figure 70. Table structure crm_emenu

Table: crm_emenu_type

Field Name	Type	Key**	Extra Info
type_id	INT(2)	PK	AUTO_INCREMENT
type_name	VARCHAR(255)		

** PK = PRIMARY KEY

Figure 71. Table structure crm_emenu_type

Table: crm_error_log

Field Name	Type	Key**	Extra Info
id	INT(11)	PK	AUTO_INCREMENT
content	TEXT		

** PK = PRIMARY KEY

Figure 72. Table structure crm_error_log

Table: crm_feedback

Field Name	Type	Key**	Extra Info
feedback_id	INT(11)	PK	AUTO_INCREMENT
feedback_title	VARCHAR(255)		
hidden	INT(1)		

** PK = PRIMARY KEY

Figure 73. Table structure crm_feedback

Table: crm_feedback_cust

Field Name	Type	Key**	Extra Info
feedback_cust_id	INT(11)	PK	AUTO_INCREMENT
feedback_id	INT(11)		
member_id	INT(11)		
submit_time	INT(11)		

** PK = PRIMARY KEY

Figure 74. Table structure crm_feedback_cust

Table: crm_feedback_cust_ans

Field Name	Type	Key**	Extra Info
feedback_cust_id	INT(11)	PK	
feedback_question_id	INT(11)	PK	
cust_ans	TEXT		

** PK = PRIMARY KEY

Figure 75. Table structure crm_feedback_cust_ans

Table: crm_feedback_question_mc_ans

Field Name	Type	Key**	Extra Info
feedback_question_id	INT(11)	MUL	
choice	TEXT		
seq	smallINT(6)		

** PK = PRIMARY KEY

Figure 76. Table structrue crm_feedback_question_mc_ans

Table: crm_feedback_shop

Field Name	Type	Key**	Extra Info
shop_id	INT(11)	PK	
feedback_id	INT(11)	PK	

** PK = PRIMARY KEY

Figure 77. Table structure crm_feedback_shop

Table: crm_luckydraw

Field Name	Type	Key**	Extra Info
luckydraw_id	INT(11)	PK	AUTO_INCREMENT
luckydraw_title	VARCHAR(100)		
luckydraw_img_url	VARCHAR(200)		
luckydraw_type	INT(4)		
company_id	INT(11)		
begin_DATE	DATE		
end_DATE	DATE		
cost	INT(11)		

** PK = PRIMARY KEY

Figure 78. Table structure crm_luckydraw

Table: crm_luckydraw_prize

Field Name	Type	Key**	Extra Info
luckydraw_prize_id	INT(11)	PK	AUTO_INCREMENT
luckydraw_id	INT(11)		
prize_name	VARCHAR(50)		
prize_description	TEXT		
probability	DOUBLE		
prize_img_url	VARCHAR(200)		
prize_limit_quantity	INT(11)		
seq	INT(4)		
img_last_upDATE	INT(11)		
terms_condition	TEXT		

** PK = PRIMARY KEY

Figure 79. Table structure crm_luckydraw_prize

Table:
crm_luckydraw_prize_reg

Field Name	Type	Key**	Extra Info
prize_reg_id	INT(11)	PK	AUTO_INCREMENT
luckydraw_prize_id	INT(11)		
member_id	INT(11)		
reg_DATE	DATETIME		
use_DATE	DATETIME		

** PK = PRIMARY KEY

Figure 80. Table structure crm_luckydraw_prize_reg

Table: crm_luckydraw_shop

Field Name	Type	Key**	Extra Info
shop_id	INT(11)	PK	
luckydraw_id	INT(11)	PK	

** PK = PRIMARY KEY

Figure 81. Table structure crm_luckydraw_shop

Table: crm_member

Field Name	Type	Key**	Extra Info
member_id	INT(11)	PK	AUTO_INCREMENT
company_id	INT(11)	PK	
member_email	VARCHAR(100)	PK	
member_password	VARCHAR(32)		
surname	VARCHAR(20)		
firstname	VARCHAR(50)		
gender	INT(4)		
birthday	DATE		
poINT	INT(11)		

** PK = PRIMARY KEY

Figure 82. Table structure crm_member

Table: crm_news

Field Name	Type	Key**	Extra Info
id	INT(10)	PK	AUTO_INCREMENT
shop_id	INT(11)		
title	VARCHAR(255)		
content	TEXT		
DATE	DATE		
last_upDATE	INT(11)		
hidden	INT(4)		

** PK = PRIMARY KEY

Figure 83. Table structure crm_news

Table: crm_order

Field Name	Type	Key**	Extra Info
order_id	INT(10)	PK	AUTO_INCREMENT
secret_code	INT(11)		
member_id	INT(11)		
shop_id	INT(10)		
DATE_time	INT(11)		
overall_mark	INT(11)		
feedback	TEXT		

** PK = PRIMARY KEY

Figure 84. Table structure crm_order

Table: crm_order_detail

Field Name	Type	Key**	Extra Info
order_id	INT(10)	PK	
item_id	INT(10)	PK	
item_qty	INT(10)		
mark	INT(4)		

** PK = PRIMARY KEY

Figure 85. Table structure crm_order_detail

Table: crm_queue

Field Name	Type	Key**	Extra Info
create_DATE	DATE	PK	
shop_id	INT(11)	PK	
queue_id	INT(11)	PK	
name	TEXT		
seats	INT(4)		
seat_group	INT(11)	PK	
type	INT(4)		
phone	TINYTEXT		
device_id	TEXT		

** PK = PRIMARY KEY

Figure 86. Table structure crm_queue

Table: crm_shop

Field Name	Type	Key**	Extra Info
shop_id	INT(11)	PK	AUTO_INCREMENT
company_id	INT(11)		
shop_name	TEXT		
open_hour	TIME		
close_hour	TIME		
latitude	DOUBLE		
longitude	DOUBLE		
traffic_name	VARCHAR(50)		
traffic_latitude	DOUBLE		
traffic_longitude	DOUBLE		
display_name	VARCHAR(50)		
display_subtitle	VARCHAR(255)		
icon_url	VARCHAR(255)		
enable	INT(4)		
news	INT(4)		
menu	INT(4)		
coupon	INT(4)		
queue	INT(4)		
booking	INT(4)		
map	INT(4)		
feedback	INT(4)		
luckydraw	INT(4)		
queue_12	INT(11)		
queue_34	INT(11)		
queue_58	INT(11)		
queue_9	INT(11)		

** PK = PRIMARY KEY

Figure 87. Table structure crm_shop

Table: crm_shop_special_event

Field Name	Type	Key**	Extra Info
event_id	INT(11)	PK	AUTO_INCREMENT
event_name	VARCHAR(100)		
shop_id	INT(11)		
event_begin	DATETIME		
event_end	DATETIME		
hidden	INT(4)		

** PK = PRIMARY KEY

Figure 88. Table structure crm_shop_special_event

Table: crm_table

Field Name	Type	Key**	Extra Info
table_id	INT(11)	PK	AUTO_INCREMENT
shop_id	INT(11)		
seats	INT(11)		
hidden	INT(4)		

** PK = PRIMARY KEY

Figure 89. Table structure crm_table

Table: crm_table_property

Field Name	Type	Key**	Extra Info
property_id	INT(11)	PK	AUTO_INCREMENT
property_name	VARCHAR(50)		

** PK = PRIMARY KEY

Figure 90. Table structure crm_table_property

Table: crm_theme

Field Name	Type	Key**	Extra Info
company_id	INT(11)	PK	
navigation_bar	VARCHAR(100)		
bg	VARCHAR(100)		
news	VARCHAR(100)		
menu	VARCHAR(100)		
coupon	VARCHAR(100)		
queue	VARCHAR(100)		
booking	VARCHAR(100)		
map	VARCHAR(100)		
feedback	VARCHAR(100)		
last_upDATE	INT(11)		

** PK = PRIMARY KEY

Figure 91. Table structure crm_theme

Table: crm_theme_android

Field Name	Type	Key**	Extra Info
theme_id	INT(11)	PK	AUTO_INCREMENT
company_id	INT(11)		
theme_name	VARCHAR(50)		
navigation_bar	VARCHAR(100)		
bg	VARCHAR(100)		
news	VARCHAR(100)		
news_pos	VARCHAR(7)		
menu	VARCHAR(100)		
menu_pos	VARCHAR(7)		
coupon	VARCHAR(100)		
coupon_pos	VARCHAR(7)		
queue	VARCHAR(100)		
queue_pos	VARCHAR(7)		
booking	VARCHAR(100)		
booking_pos	VARCHAR(7)		
map	VARCHAR(100)		
map_pos	VARCHAR(7)		
feedback	VARCHAR(100)		
feedback_pos	VARCHAR(7)		
hidden	INT(4)		
last_upDATE	INT(11)		

** PK = PRIMARY KEY

Figure 92. Table structure crm_theme_android

Table: crm_theme_iphone

Field Name	Type	Key**	Extra Info
theme_id	INT(11)	PK	AUTO_INCREMENT
company_id	INT(11)	UNI	
theme_name	VARCHAR(50)		
navigation_bar	VARCHAR(100)		
bg	VARCHAR(100)		
shop_bg	VARCHAR(100)		
news	VARCHAR(100)		
news_pos	VARCHAR(7)		
menu	VARCHAR(100)		
menu_pos	VARCHAR(7)		
coupon	VARCHAR(100)		
coupon_pos	VARCHAR(7)		
queue	VARCHAR(100)		
queue_pos	VARCHAR(7)		
booking	VARCHAR(100)		
booking_pos	VARCHAR(7)		
map	VARCHAR(100)		
map_pos	VARCHAR(7)		
feedback	VARCHAR(100)		
feedback_pos	VARCHAR(7)		
luckydraw	VARCHAR(100)		
luckydraw_pos	VARCHAR(7)		
hidden	INT(4)		
last_upDATE	INT(11)		

** PK = PRIMARY KEY
UNI = UNIQUE

Figure 93. Table structure crm_theme_iphone

Table: crm_user

Field Name	Type	Key**	Extra Info
user_id	INT(11)	PK	AUTO_INCREMENT
type	INT(4)		
device_id	VARCHAR(255)		
device_name	VARCHAR(255)		
phone	VARCHAR(20)		

** PK = PRIMARY KEY

Figure 94. Table structure crm_user

Table: crm_web_user

Field Name	Type	Key**	Extra Info
user_id	VARCHAR(20)	PK	
password	VARCHAR(32)		
company_id	INT(11)		
group	INT(2)		
shop_id	INT(11)		

** PK = PRIMARY KEY

Figure 95. Table structure crm_web_user

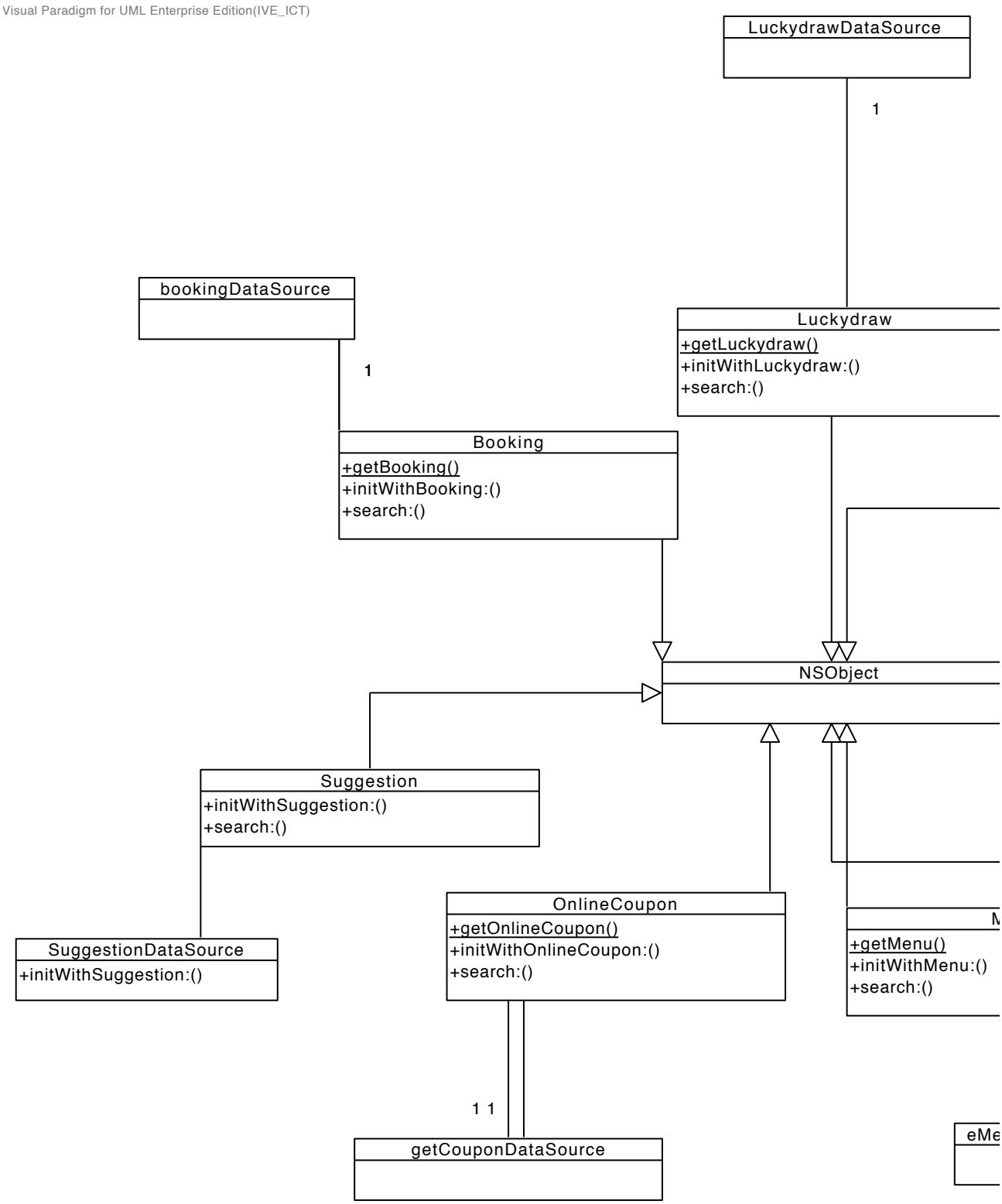
Table: crm_web_usergroup

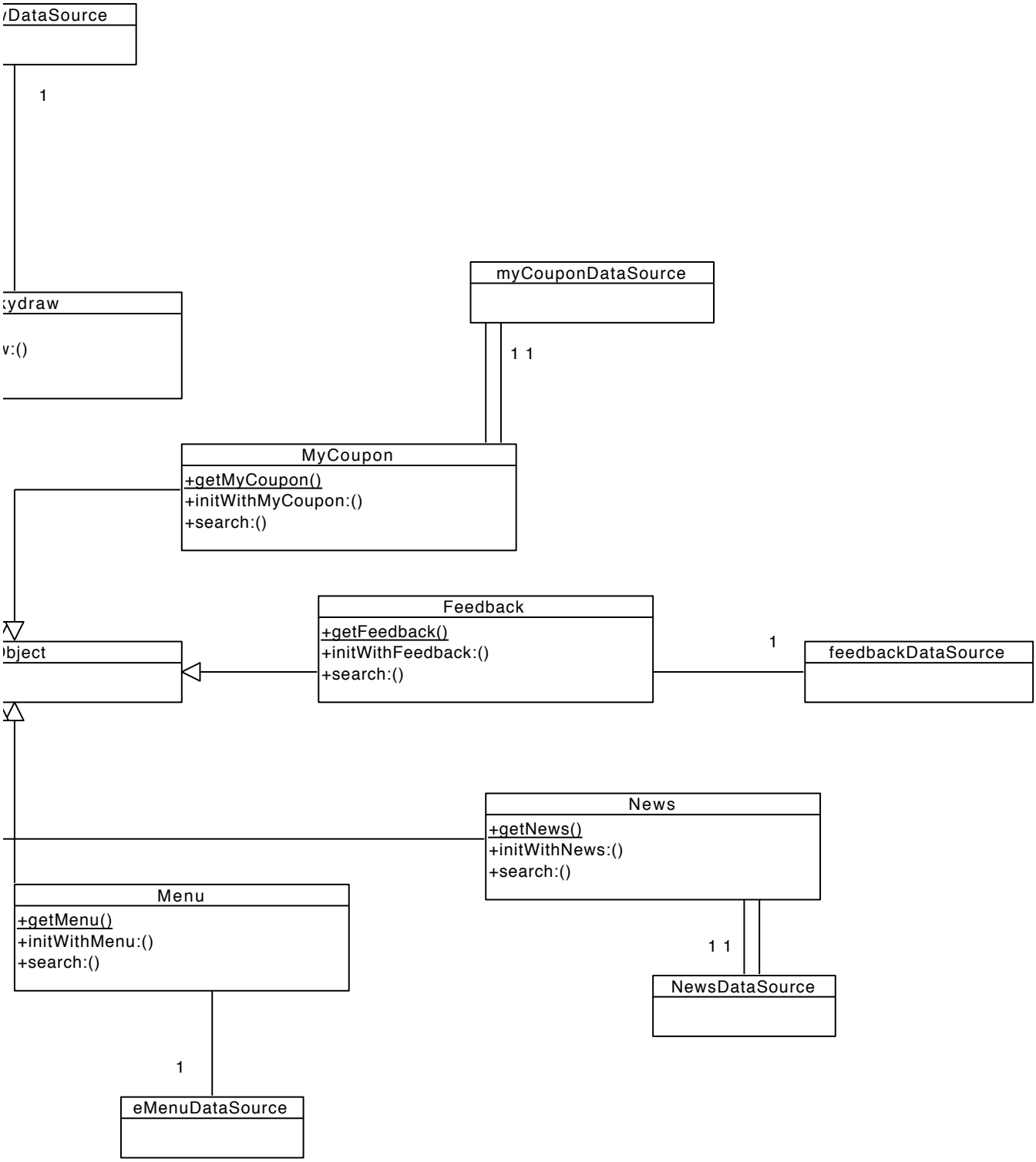
Field Name	Type	Key**	Extra Info
usergroup_id	INT(2)	PK	AUTO_INCREMENT
group_name	VARCHAR(30)		
full_control	INT(1)		
pending_list	INT(4)		
news	INT(4)		
menu	INT(4)		
booking	INT(4)		
feedback	INT(4)		
member	INT(4)		
push	INT(4)		
mobile	INT(4)		
coupon	INT(4)		

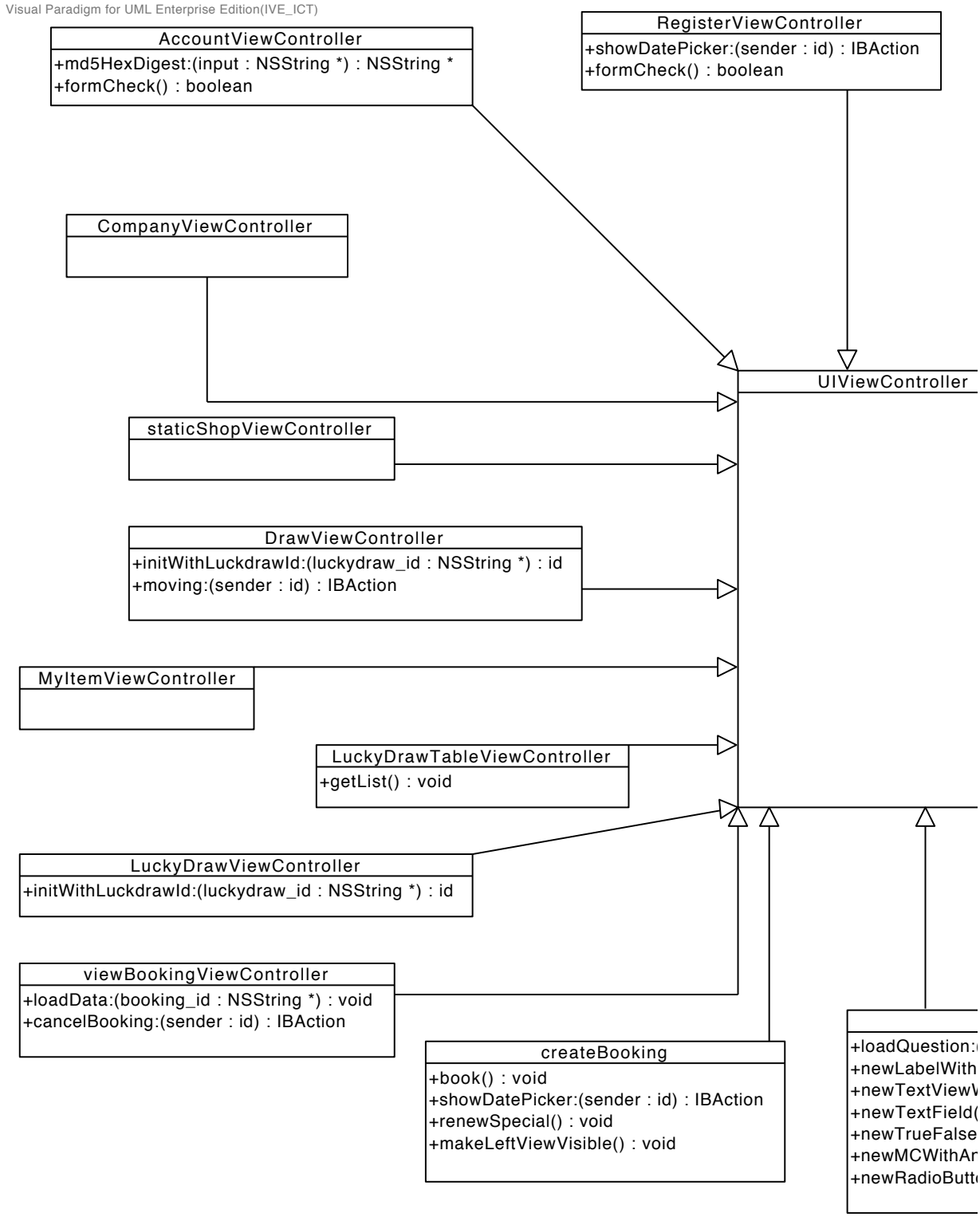
** PK = PRIMARY KEY

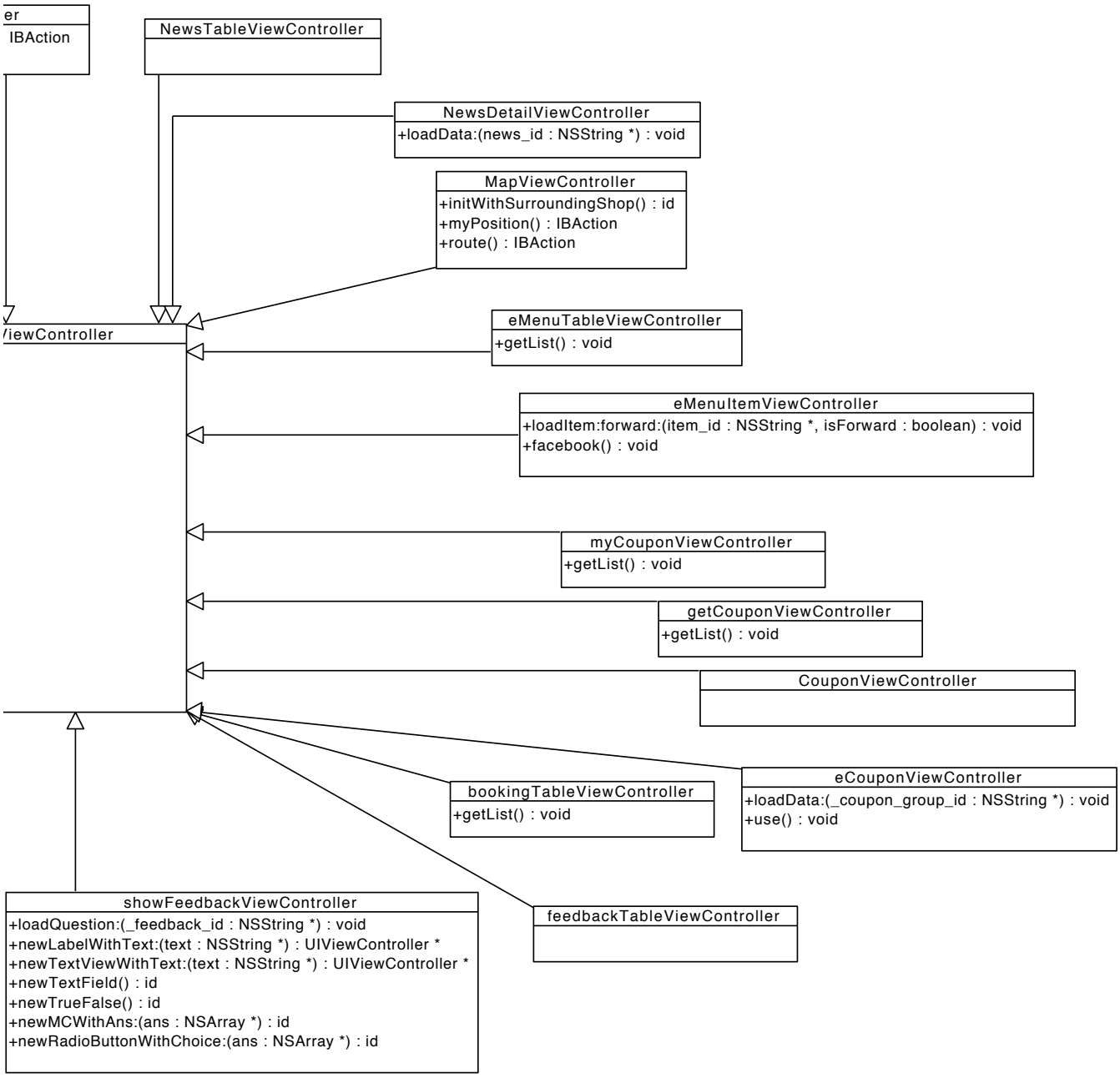
Figure 96. Table structure crm_web_usergroup

Class Diagram (iPhone App)

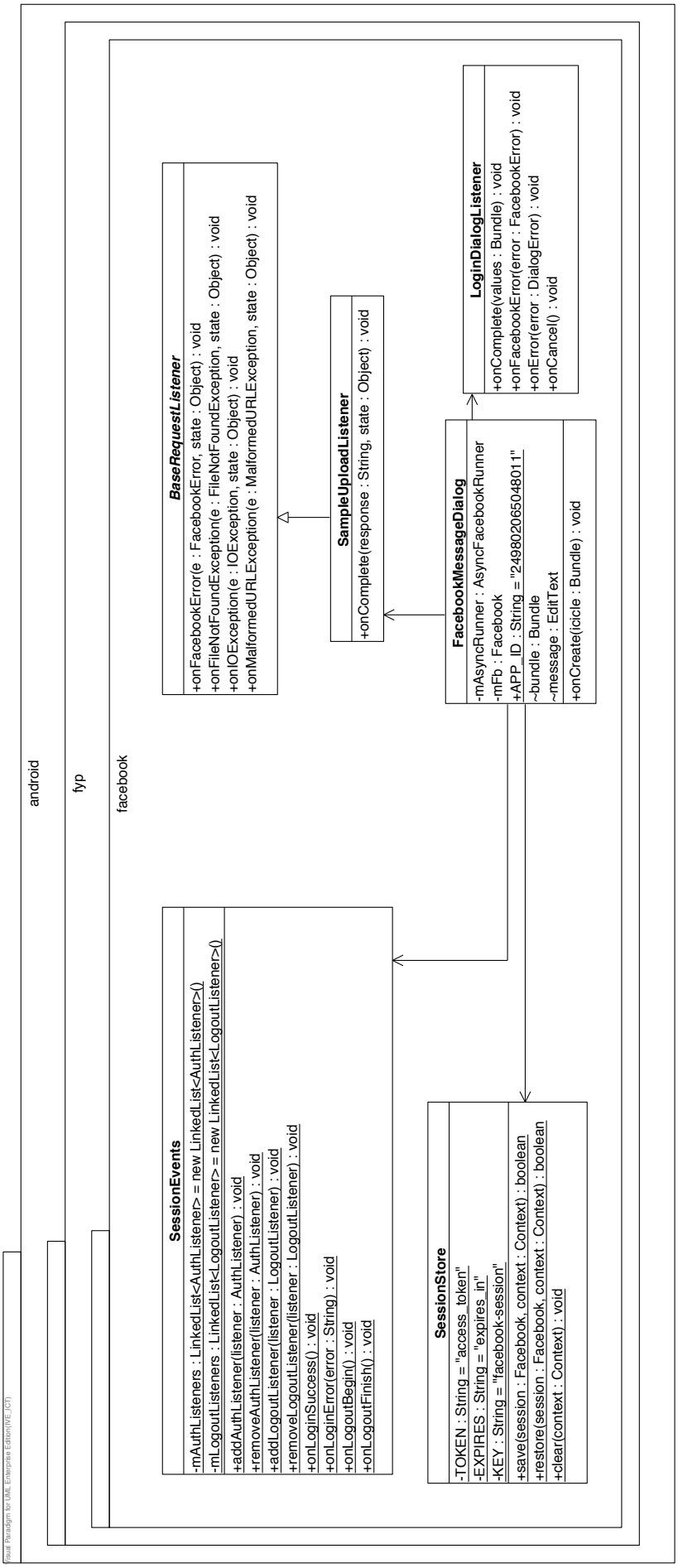


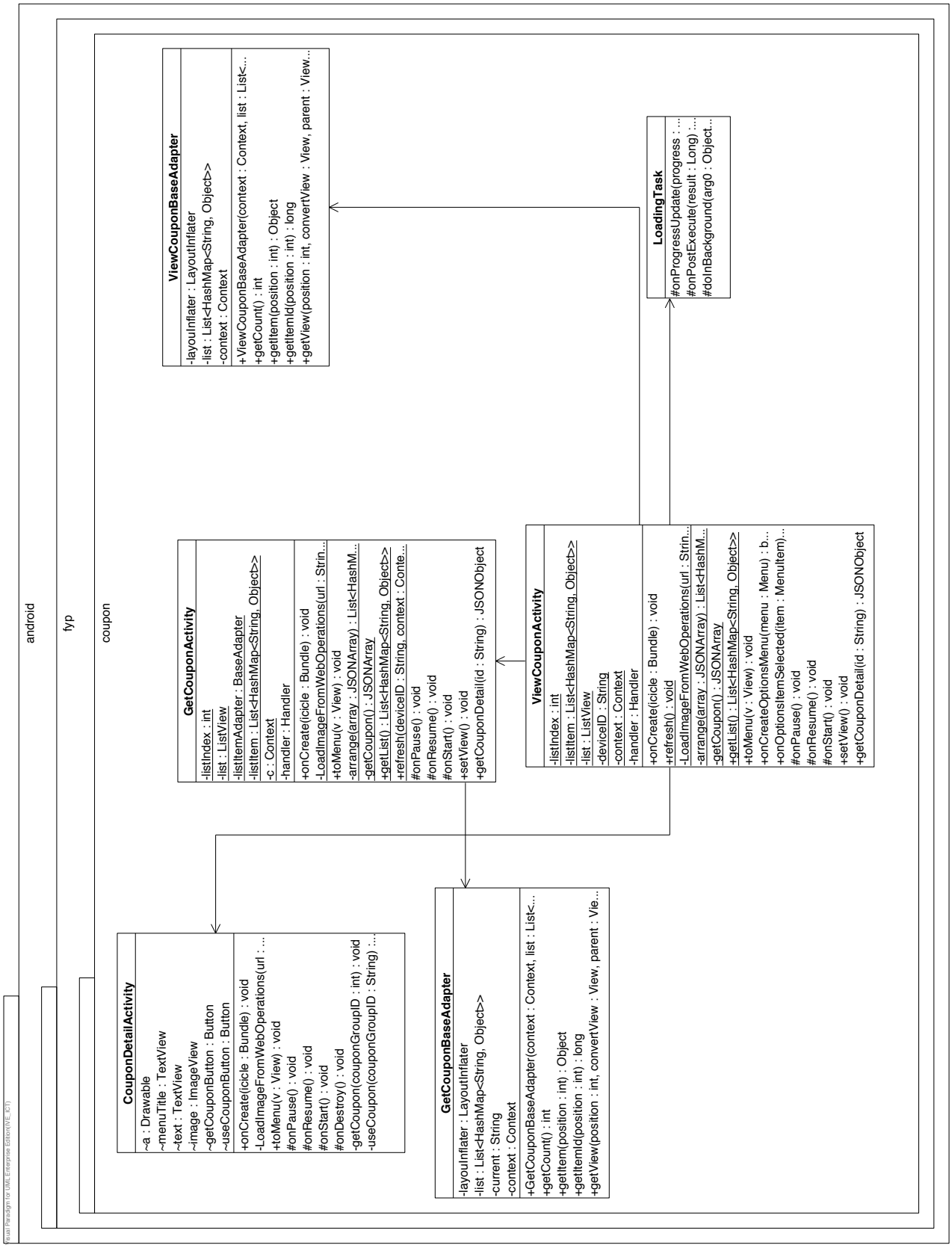

/DataSource
1
ydraw
v:()
myCouponDataSource
1 1
MyCoupon
+getMyCoupon()
+initWithMyCoupon:()
+search:()
Feedback
+getFeedback()
+initWithFeedback:()
+search:()
1
feedbackDataSource
bject
News
+getNews()
+initWithNews:()
+search:()
Menu
+getMenu()
+initWithMenu:()
+search:()
1 1
NewsDataSource
1
eMenuDataSource

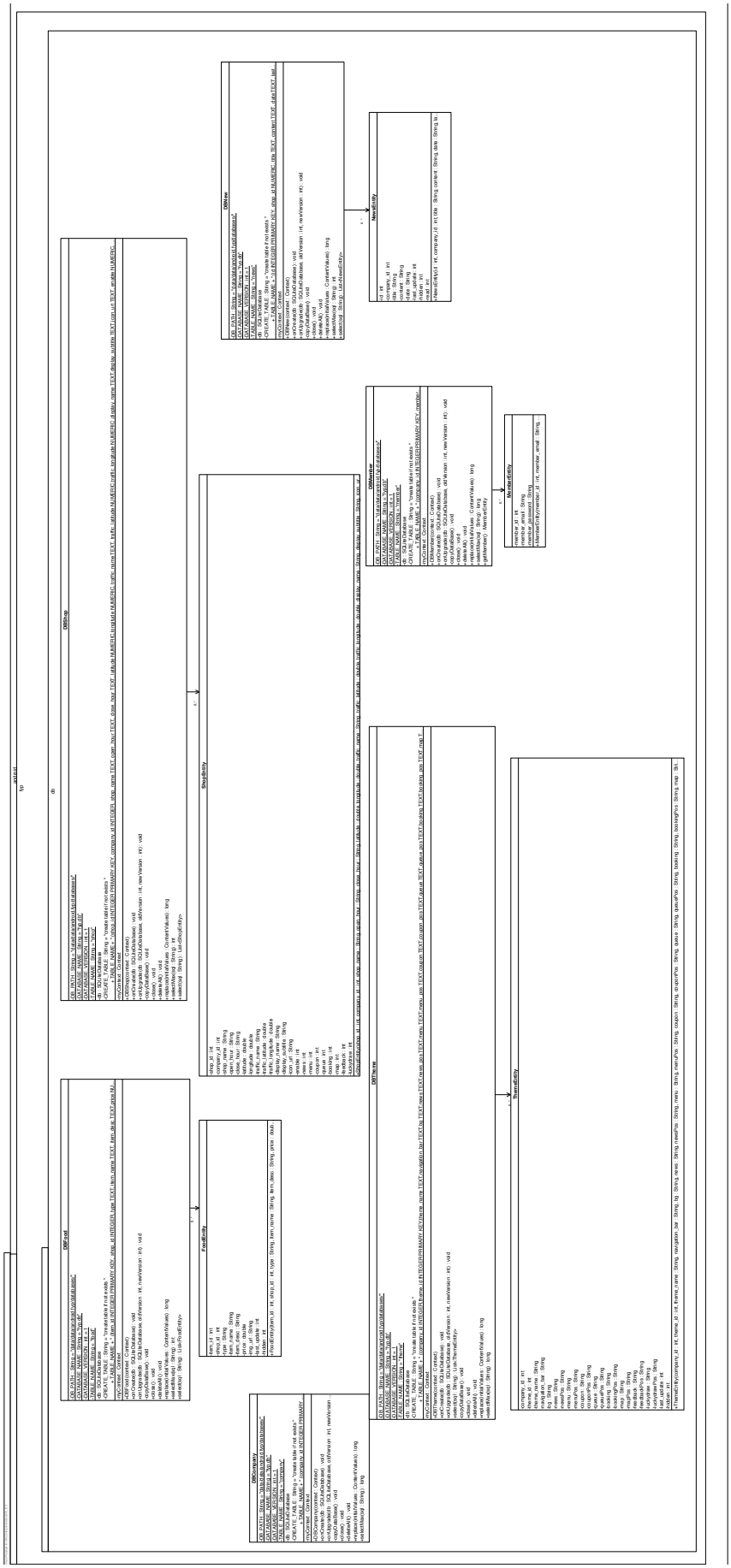

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
AccountViewController
+md5HexDigest:(input : NSString *) : NSString *
+formCheck() : boolean
RegisterViewController
+showDatePicker:(sender : id) : IBAction
+formCheck() : boolean
CompanyViewController
UIViewController
staticShopViewController
DrawViewController
+initWithLuckdrawId:(luckydraw_id : NSString *) : id
+moving:(sender : id) : IBAction
MyItemViewController
LuckyDrawTableViewController
+getList() : void
LuckyDrawViewController
+initWithLuckdrawId:(luckydraw_id : NSString *) : id
viewBookingViewController
+loadData:(booking_id : NSString *) : void
+cancelBooking:(sender : id) : IBAction
createBooking
+book() : void
+showDatePicker:(sender : id) : IBAction
+renewSpecial() : void
+makeLeftViewVisible() : void
+loadQuestion:
+newLabelWith
+newTextView
+newTextField(
+newTrueFalse
+newMCWithAr
+newRadioButt

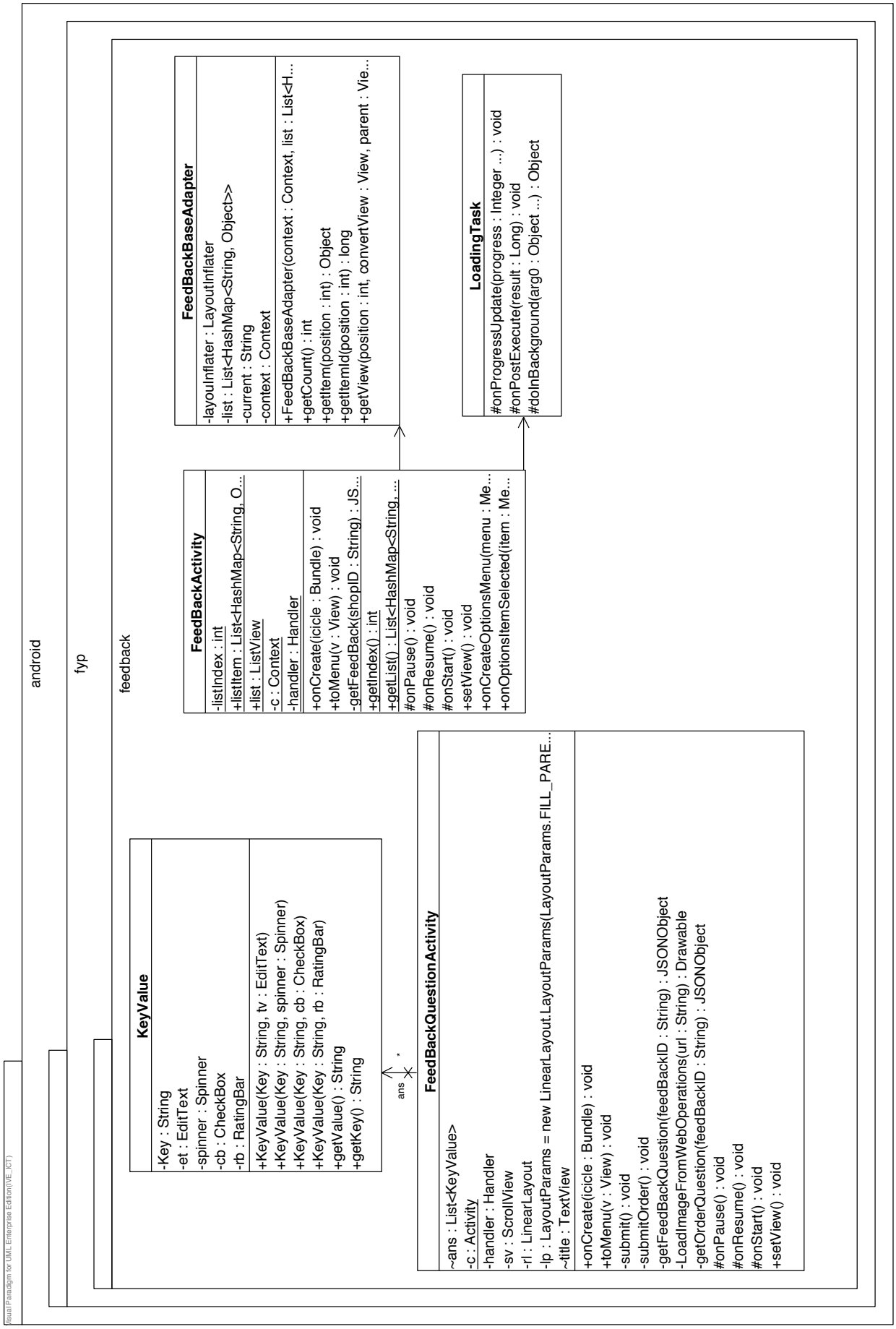

er
IBAction
NewsTableViewController
NewsDetailViewController
+loadData:(news_id : NSString *) : void
MapViewController
+initWithSurroundingShop() : id
+myPosition() : IBAction
+route() : IBAction
iewController
eMenuTableViewController
+getList() : void
eMenuItemViewController
+loadItem:forward:(item_id : NSString *, isForward : boolean) : void
+facebook() : void
myCouponViewController
+getList() : void
getCouponViewController
+getList() : void
CouponViewController
eCouponViewController
+loadData:(_coupon_group_id : NSString *) : void
+use() : void
bookingTableViewController
+getList() : void
showFeedbackViewController
+loadQuestion:(_feedback_id : NSString *) : void
+newLabelWithText:(text : NSString *) : UIViewController *
+newTextViewWithText:(text : NSString *) : UIViewController *
+newTextField() : id
+newTrueFalse() : id
+newMCWithAns:(ans : NSArray *) : id
+newRadioButtonWithChoice:(ans : NSArray *) : id
feedbackTableViewController

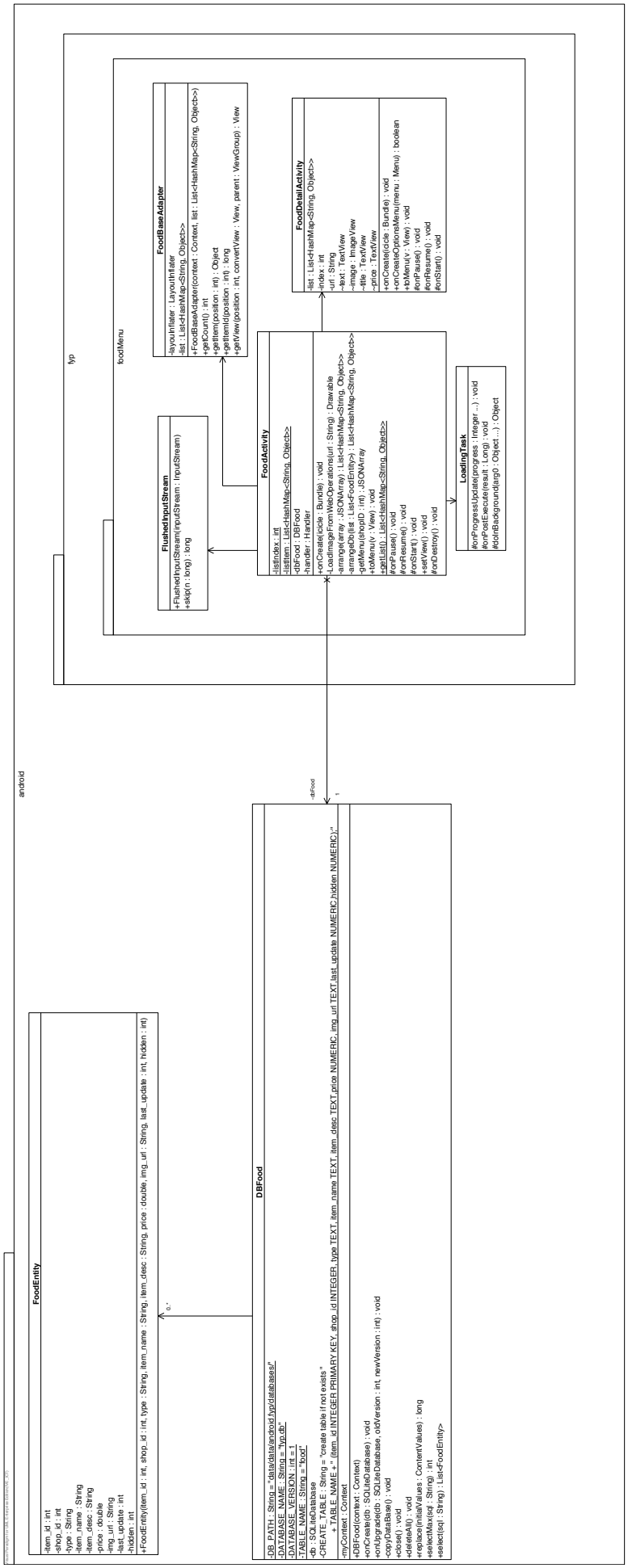
Class Diagram (Android App)

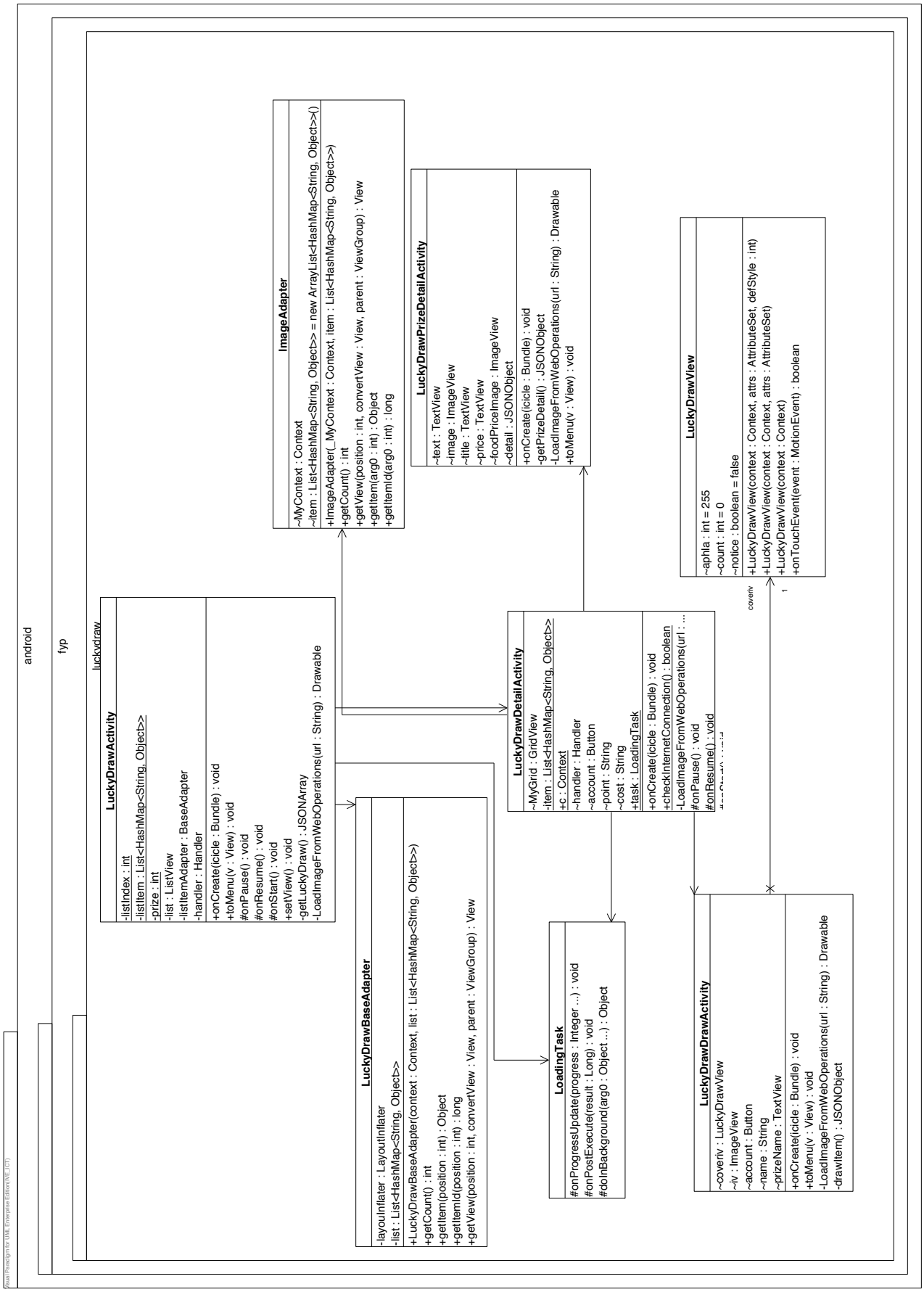

Visual Paradigm for UML Enterprise Edition(VE, ICT)
android
fyp
facebook
SessionEvents
-mAuthListeners : LinkedList<AuthListener> = new LinkedList<AuthListener>()
-mLogoutListeners : LinkedList<LogoutListener> = new LinkedList<LogoutListener>()
+addAuthListener(listener : AuthListener) : void
+removeAuthListener(listener : AuthListener) : void
+addLogoutListener(listener : LogoutListener) : void
+removeLogoutListener(listener : LogoutListener) : void
+onLoginSuccess() : void
+onLoginError(error : String) : void
+onLogoutBegin() : void
+onLogoutFinish() : void
SessionStore
-TOKEN : String = "access_token"
-EXPIRES : String = "expires_in"
-KEY : String = "facebook-session"
+save(session : Facebook, context : Context) : boolean
+restore(session : Facebook, context : Context) : boolean
+clear(context : Context) : void
BaseRequestListener
+onFacebookError(e : FacebookError, state : Object) : void
+onFileNotFoundException(e : FileNotFoundException, state : Object) : void
+onIOException(e : IOException, state : Object) : void
+onMalformedURLException(e : MalformedURLException, state : Object) : void
SampleUploadListener
+onComplete(response : String, state : Object) : void
FacebookMessageDialog
-mAsyncRunner : AsyncFacebookRunner
-mFb : Facebook
+APP_ID : String = "249802065048011"
~bundle : Bundle
~message : EditText
+onCreate(icicle : Bundle) : void
LoginDialogListener
+onComplete(values : Bundle) : void
+onFacebookError(error : FacebookError) : void
+onError(error : DialogError) : void
+onCancel() : void

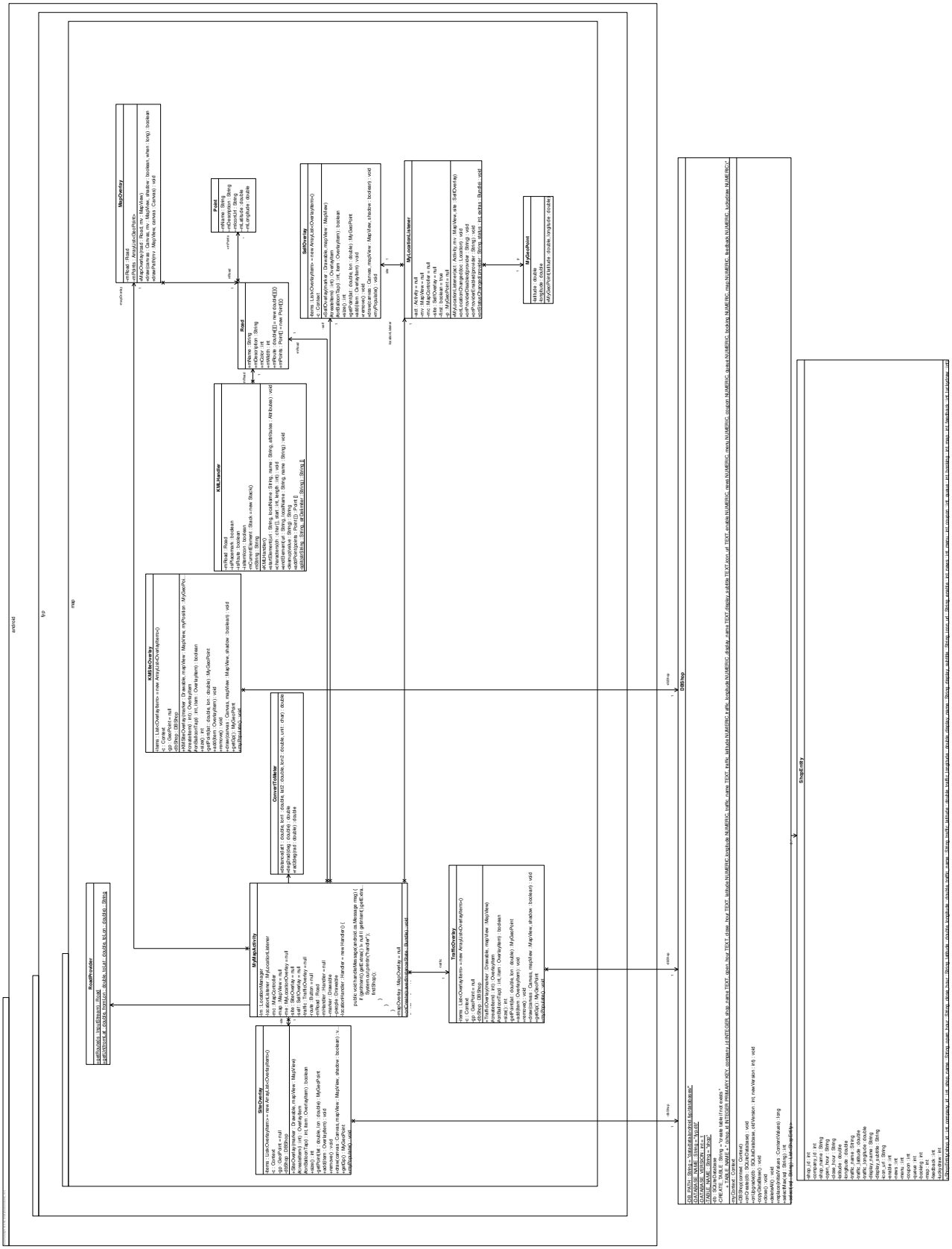

Visual Paradigm for UML Enterprise Edition(VE, JC1)
android
fyp
coupon
CouponDetailActivity
~a : Drawable
~menuTitle : TextView
~text : TextView
~image : ImageView
~getCouponButton : Button
~useCouponButton : Button
+onCreate(icicle : Bundle) : void
-LoadImageFromWebOperations(url : ...
+toMenu(v : View) : void
#onPause() : void
#onResume() : void
#onStart() : void
#onDestroy() : void
-getCoupon(couponGroupID : int) : void
-useCoupon(couponGroupID : String) : ...
GetCouponBaseAdapter
-layoutInflater : LayoutInflater
-list : List<HashMap<String, Object>>
-current : String
-context : Context
+GetCouponBaseAdapter(context : Context, list : List<...
+getCount() : int
+getItem(position : int) : Object
+getItemId(position : int) : long
+getView(position : int, convertView : View, parent : Vie...
GetCouponActivity
-listIndex : int
-list : ListView
-listItemAdapter : BaseAdapter
-listItem : List<HashMap<String, Object>>
-c : Context
-handler : Handler
+onCreate(icicle : Bundle) : void
-LoadImageFromWebOperations(url : Strin...
+toMenu(v : View) : void
-arrange(array : JSONArray) : List<HashM...
-getCoupon() : JSONArray
+getList() : List<HashMap<String, Object>>
+refresh(deviceID : String, context : Conte...
#onPause() : void
#onResume() : void
#onStart() : void
+setView() : void
+getCouponDetail(id : String) : JSONObject
ViewCouponBaseAdapter
-layoutInflater : LayoutInflater
-list : List<HashMap<String, Object>>
-context : Context
+ ViewCouponBaseAdapter(context : Context, list : List<...
+getCount() : int
+getItem(position : int) : Object
+getItemId(position : int) : long
+getView(position : int, convertView : View, parent : View...
LoadingTask
#onProgressUpdate(progress : ...
#onPostExecute(result : Long) : ...
#doInBackground(arg0 : Object...
ViewCouponActivity
-listIndex : int
-listItem : List<HashMap<String, Object>>
-list : ListView
-deviceID : String
-context : Context
-handler : Handler
+onCreate(icicle : Bundle) : void
+refresh() : void
-LoadImageFromWebOperations(url : Strin...
-arrange(array : JSONArray) : List<HashM...
-getCoupon() : JSONArray
+getList() : List<HashMap<String, Object>>
+toMenu(v : View) : void
+onCreateOptionsMenu(menu : Menu) : b...
+onOptionsItemSelected(item : MenuItem)...
#onPause() : void
#onResume() : void
#onStart() : void
+setView() : void
+getCouponDetail(id : String) : JSONObject

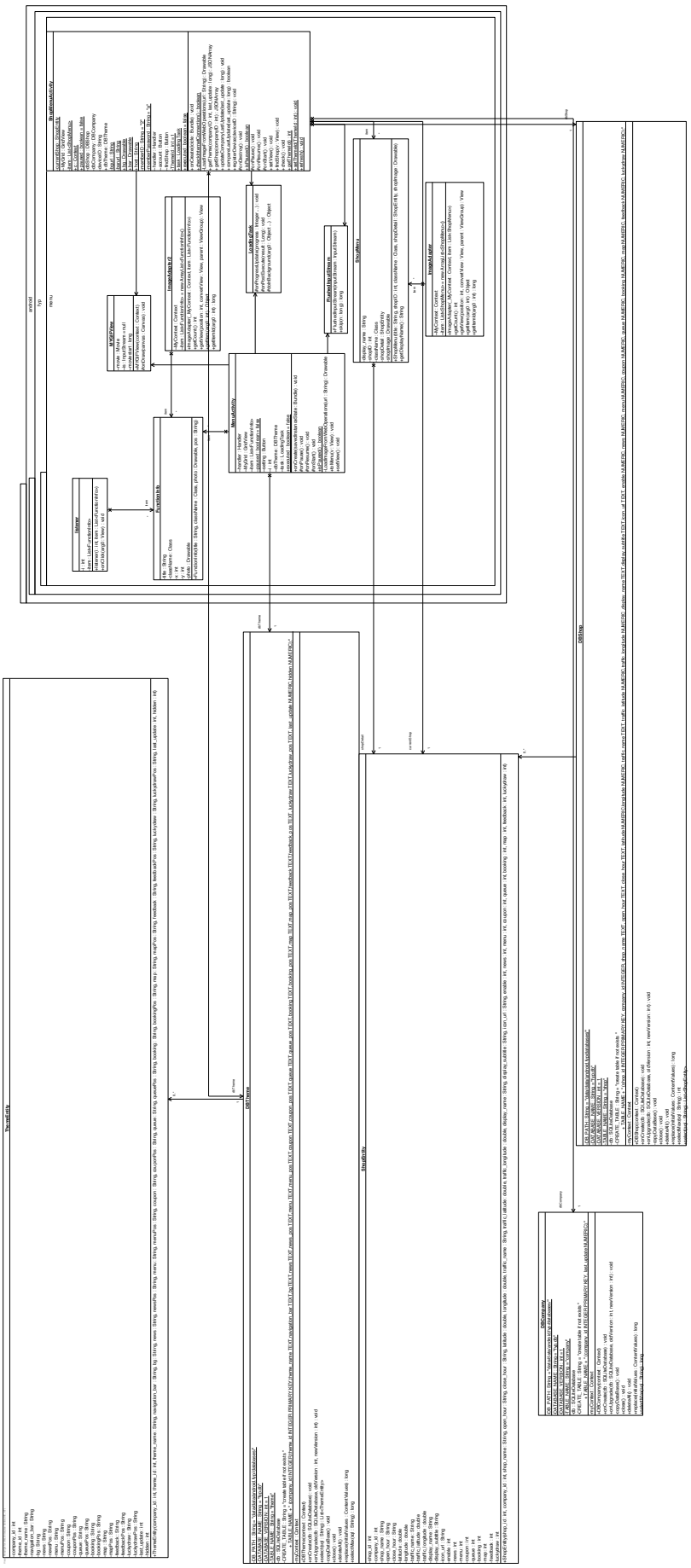



Visual Paradigm for UML Enterprise Edition(VE_ICT)
android
fyp
feedback
KeyValue
-Key : String
-et : EditText
-spinner : Spinner
-cb : CheckBox
-rb : RatingBar
+KeyValue(Key : String, tv : EditText)
+KeyValue(Key : String, spinner : Spinner)
+KeyValue(Key : String, cb : CheckBox)
+KeyValue(Key : String, rb : RatingBar)
+getValue() : String
+getKey() : String
ans
*
FeedBackQuestionActivity
~ans : List<KeyValue>
-c : Activity
-handler : Handler
-sv : ScrollView
-rl : LinearLayout
-lp : LayoutParams = new LinearLayout.LayoutParams(LayoutParams.FILL_PARE...
~title : TextView
+onCreate(icicle : Bundle) : void
+toMenu(v : View) : void
-submit() : void
-submitOrder() : void
-getFeedBackQuestion(feedBackID : String) : JSONObject
-LoadImageFromWebOperations(url : String) : Drawable
-getOrderQuestion(feedBackID : String) : JSONObject
#onPause() : void
#onResume() : void
#onStart() : void
+setView() : void
FeedBackActivity
-listIndex : int
+listItem : List<HashMap<String, O...
+list : ListView
-c : Context
-handler : Handler
+onCreate(icicle : Bundle) : void
+toMenu(v : View) : void
-getFeedBack(shopID : String) : JS...
+getIndex() : int
+getList() : List<HashMap<String, ...
#onPause() : void
#onResume() : void
#onStart() : void
+setView() : void
+onCreateOptionsMenu(menu : Me...
+onOptionsItemSelected(item : Me...
FeedBackBaseAdapter
-layoutInflater : LayoutInflater
-list : List<HashMap<String, Object>>
-current : String
-context : Context
+FeedBackBaseAdapter(context : Context, list : List<H...
+getCount() : int
+getItem(position : int) : Object
+getItemId(position : int) : long
+getView(position : int, convertView : View, parent : Vie...
LoadingTask
#onProgressUpdate(progress : Integer ...) : void
#onPostExecute(result : Long) : void
#doInBackground(arg0 : Object ...) : Object

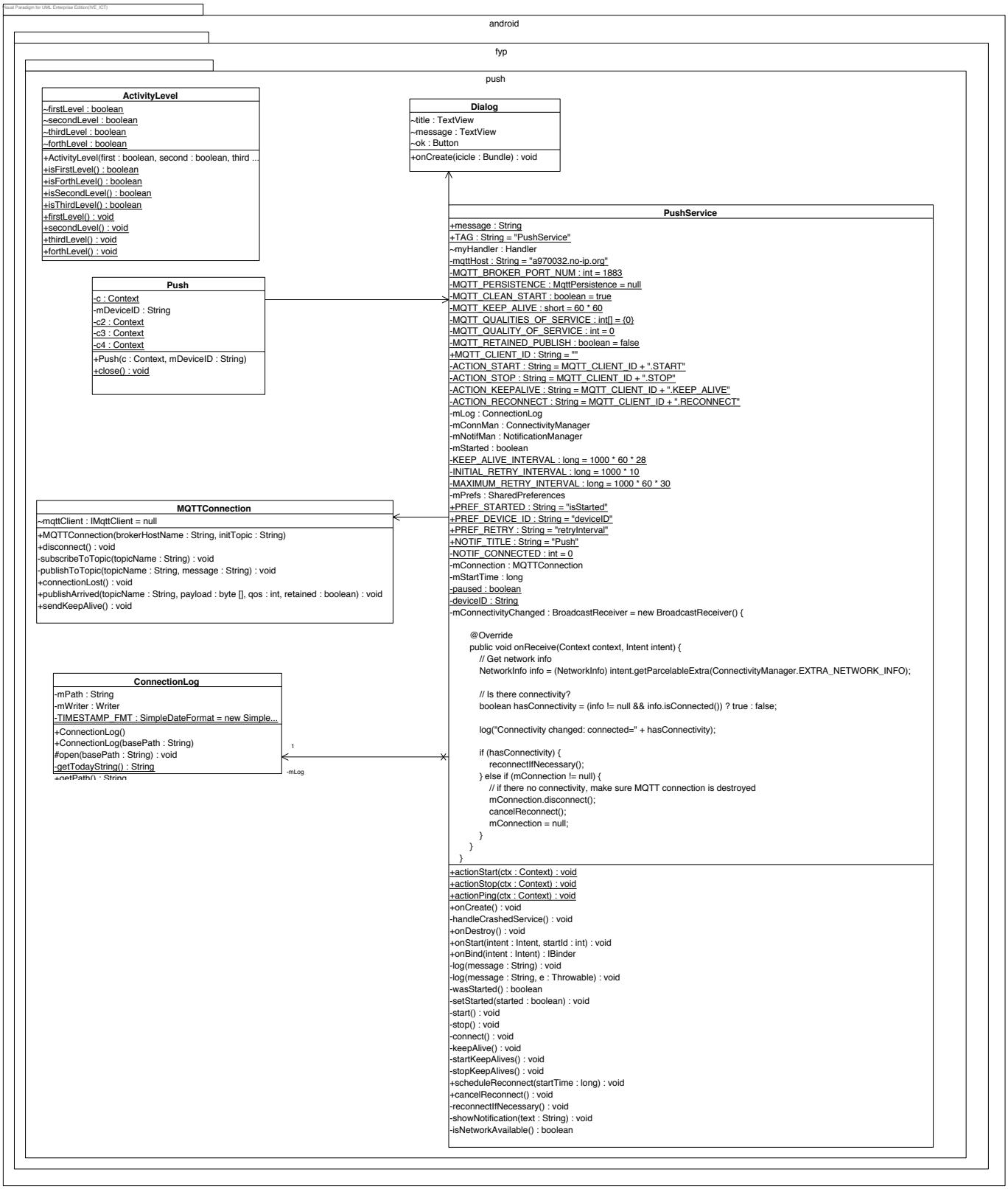

FoodEntity
-item_id : int
-shop_id : int
-type : String
-item_name : String
-item_desc : String
-price : double
-img_url : String
-last_update : int
-hidden : int
+FoodEntity(item_id : int, shop_id : int, type : String, item_name : String, item_desc : String, price : double, img_url : String, last_update : int, hidden : int)
0..*
DBFood
-DB_PATH : String = "/data/data/android.fyp/databases/"
-DATABASE_NAME : String = "fyp.db"
-DATABASE_VERSION : int = 1
-TABLE_NAME : String = "food"
-db : SQLiteDatabase
-CREATE_TABLE : String = "create table if not exists "
+ TABLE_NAME + " (item_id INTEGER PRIMARY KEY, shop_id INTEGER, type TEXT, item_name TEXT, item_desc TEXT, price NUMERIC, img_url TEXT, last_update NUMERIC, hidden NUMERIC);"
-myContext : Context
+DBFood(context : Context)
+onCreate(db : SQLiteDatabase) : void
+onUpgrade(db : SQLiteDatabase, oldVersion : int, newVersion : int) : void
-copyDataBase() : void
+close() : void
+deleteAll() : void
+replace(initialValues : ContentValues) : long
+selectMax(sql : String) : int
+selectSql(sql : String) : List<FoodEntity>
-dbFood
1
android
fyp
foodMenu
FlushedInputStream
+FlushedInputStream(inputStream : InputStream)
+skip(n : long) : long
FoodBaseAdapter
-layoutInflater : LayoutInflater
-list : List<HashMap<String, Object>>
+FoodBaseAdapter(context : Context, list : List<HashMap<String, Object>>)
+getCount() : int
+getItem(position : int) : Object
+getItemId(position : int) : long
+getView(position : int, convertView : View, parent : ViewGroup) : View
FoodActivity
-listIndex : int
-listItem : List<HashMap<String, Object>>
-dbFood : DBFood
-handler : Handler
+onCreate(icicle : Bundle) : void
+LoadImageFromWebOperations(url : String) : Drawable
-arrange(array : JSONArray) : List<HashMap<String, Object>>
-arrangeb(list : List<FoodEntity>) : List<HashMap<String, Object>>
-getMenu(shop) : int) : JSONArray
+onMenu(v : View) : void
+getList() : List<HashMap<String, Object>>
#onPause() : void
#onResume() : void
#onStart() : void
+setView() : void
#onDestroy() : void
FoodDetailActivity
-list : List<HashMap<String, Object>>
-index : int
-url : String
~text : TextView
~image : ImageView
~title : TextView
~price : TextView
+onCreate(icicle : Bundle) : void
+onCreateOptionsMenu(menu : Menu) : boolean
+onMenu(v : View) : void
#onPause() : void
#onResume() : void
#onStart() : void
LoadingTask
#onProgressUpdate(progress : Integer ...) : void
#onPostExecute(result : Long) : void
#doInBackground(arg0 : Object ...) : Object

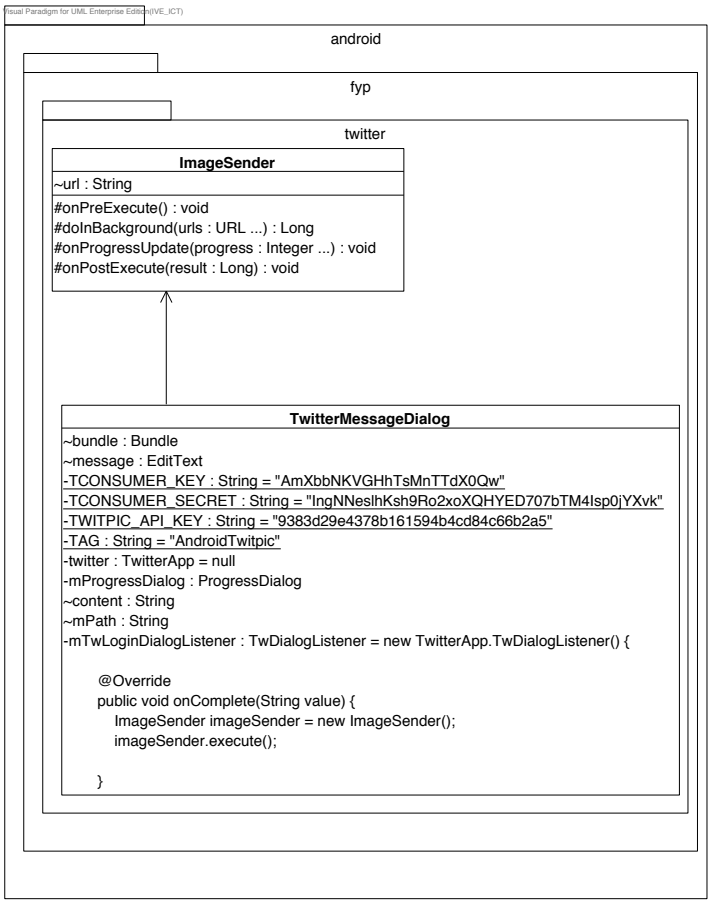

Alan Paradigm for UML Enterprise Edition(VE, JCT)
android
fyp
luckydraw
LuckyDrawActivity
-listIndex : int
-listItem : List<HashMap<String, Object>>
-prize : int
-list : ListView
-listItemAdapter : BaseAdapter
-handler : Handler
+onCreate(icicle : Bundle) : void
+toMenu(v : View) : void
#onPause() : void
#onResume() : void
#onStart() : void
+setView() : void
-getLuckyDraw() : JSONArray
-LoadImageFromWebOperations(url : String) : Drawable
ImageAdapter
~MyContext : Context
~item : List<HashMap<String, Object>> = new ArrayList<HashMap<String, Object>>()
+ImageAdapter(_MyContext : Context, item : List<HashMap<String, Object>>)
+getCount() : int
+getView(position : int, convertView : View, parent : ViewGroup) : View
+getItem(arg0 : int) : Object
+getItemId(arg0 : int) : long
LuckyDrawPrizeDetailActivity
~text : TextView
~image : ImageView
~title : TextView
~price : TextView
~foodPriceImage : ImageView
~detail : JSONObject
+onCreate(icicle : Bundle) : void
-getPrizeDetail() : JSONObject
-LoadImageFromWebOperations(url : String) : Drawable
+toMenu(v : View) : void
LuckyDrawView
~aphla : int = 255
~count : int = 0
~notice : boolean = false
+LuckyDrawView(context : Context, attrs : AttributeSet, defStyle : int)
+LuckyDrawView(context : Context, attrs : AttributeSet)
+LuckyDrawView(context : Context)
+onTouchEvent(event : MotionEvent) : boolean
coveriv
1
LuckyDrawDetailActivity
~MyGrid : GridView
-item : List<HashMap<String, Object>>
+c : Context
~handler : Handler
~account : Button
~point : String
~cost : String
+task : LoadingTask
+onCreate(icicle : Bundle) : void
+checkInternetConnection() : boolean
-LoadImageFromWebOperations(url : ...
#onPause() : void
#onResume() : void
LuckyDrawBaseAdapter
-layoutInflater : LayoutInflater
-list : List<HashMap<String, Object>>
+LuckyDrawBaseAdapter(context : Context, list : List<HashMap<String, Object>>)
+getCount() : int
+getItem(position : int) : Object
+getItemId(position : int) : long
+getView(position : int, convertView : View, parent : ViewGroup) : View
LoadingTask
#onProgressUpdate(progress : Integer ...) : void
#onPostExecute(result : Long) : void
#doInBackground(arg0 : Object ...) : Object
LuckyDrawDrawActivity
~coveriv : LuckyDrawView
~iv : ImageView
~account : Button
~name : String
~prizeName : TextView
+onCreate(icicle : Bundle) : void
+toMenu(v : View) : void
-LoadImageFromWebOperations(url : String) : Drawable
-drawItem() : JSONObject

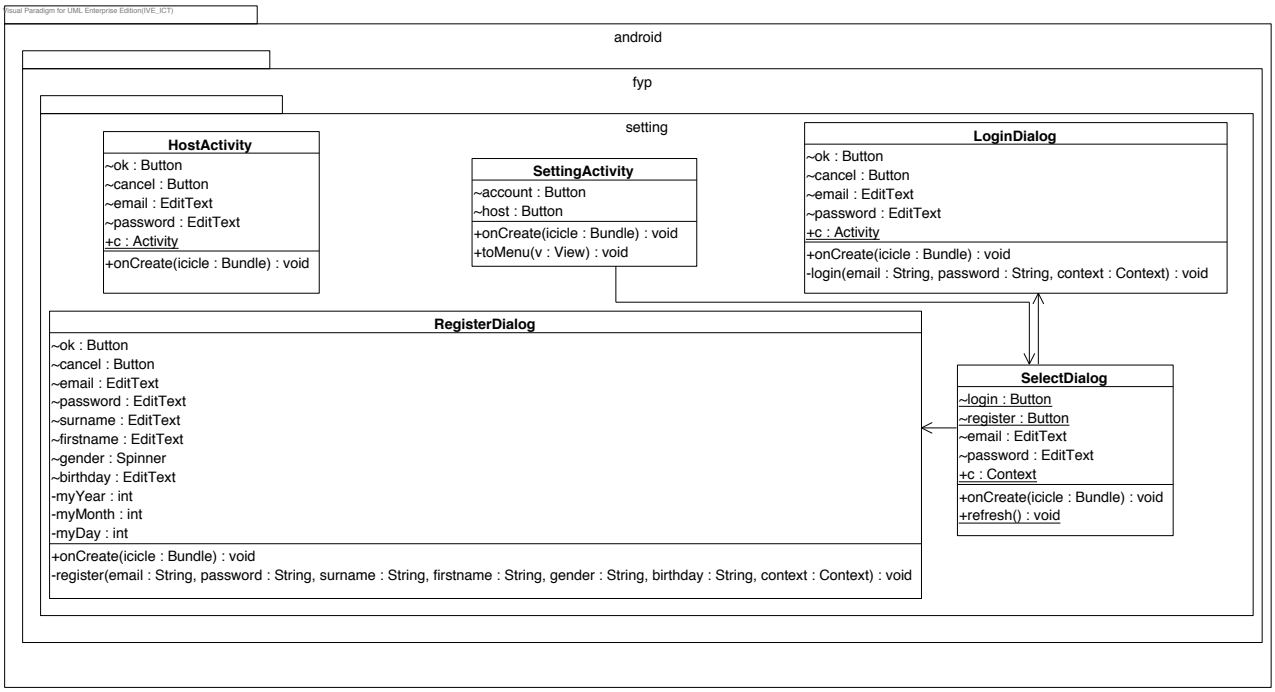

android
tp
map
RouteFinder
setRoute(s: InputStream): Route
getRoute(s: double, slat: double, elat: double, slon: double): String
MapOverlay
-mRoad: Road
-mPoints: ArrayList<GeoPoint>
+MapOverlay(road : Road, mv : MapView)
+draw(shadow : boolean, when : long) : boolean
+drawPath(mv : MapView, canvas : Canvas) : void
Point
-mName: String
-mDescription: String
-mLat: double
-mLongitude: double
Road
-mName: String
-mDescription: String
-mPoints: ArrayList
-mWidth: int
-mRouteColor: int
+mRoute: Route[0] = new Route[0]
+mPoints: Point[] = new Point[0]
KMLHandler
-mRoad: Road
-mPlace: boolean
-mBank: boolean
-mDemon: boolean
-mCurrentElement: Stack <new Stack()>
-mString: String
+KMLHandler()
+startElement(uri : String, localName: String, name: String, attributes: Attributes) : void
+characters(ch : char[], start : int, length : int) : void
+endElement(uri : String, localName : String, name: String) : void
+parse(pInstr : String) : String
+addPoints(pPts : Point[]) : Point[]
+getRoad(pInstr : String) : Road
+getStringFromInputStream(pIs : String) : String []
KMLHandler
items: LinkOverlayItem> =new ArrayList<OverlayItem>()
-c: Context
-gp: GeoPoint = null
-sDBHelp: DBHelp
+KMLHandler(mv : Drawable, mapView: MapView, myPoint: MyGeoPo...
#populate() : void
#createItem(pIdx) : LinkOverlayItem
#size() : int
+getPoint(lat : double, lon : double) : MyGeoPoint
+addItem(ovi : OverlayItem) : void
+removeAll() : void
+draw(canvas : Canvas, mapView : MapView, shadow : boolean) : void
+getPoint() : MyGeoPoint
+getMyPoint() : MyGeoPoint
DistanceToMeter
+distance(lat : double, lon : double, lat2 : double, lon2 : double, unit : char) : double
+degrees(rad : double) : double
+radians(deg : double) : double
SetOverlay
items: LinkOverlayItem> =new ArrayList<OverlayItem>()
-c: Context
-sDBHelp: DBHelp
+SetOverlay(mv : Drawable, mapView : MapView)
#resetItem() : int, OverlayItem
#createItem(pIdx) : int, OverlayItem
#size() : int
+getPoint(lat : double, lon : double) : MyGeoPoint
+addItem(ovi : OverlayItem) : void
+removeAll() : void
+draw(canvas : Canvas, mapView : MapView, shadow : boolean) : void
+getPoint() : MyGeoPoint
+addOverlayItem() : void
MyLocationOverlay
items: LinkOverlayItem> =new ArrayList<OverlayItem>()
-c: Context
-gp: GeoPoint = null
-sDBHelp: DBHelp
+TrafficOverlay(mv : Drawable, mapView : MapView)
#createItem(pIdx) : int, OverlayItem
#size() : int
+getPoint(lat : double, lon : double) : MyGeoPoint
+addItem(ovi : OverlayItem) : void
+removeAll() : void
+draw(canvas : Canvas, mapView : MapView, shadow : boolean) : void
+getPoint() : MyGeoPoint
+addOverlayItem() : void
MyLocationListener
-ctx: Activity = null
-mv: MapView = null
-myOverlay: MyLocationOverlay = null
-site: SetOverlay = null
-dbg: boolean = true
-sDBHelp: DBHelp = null
+MyLocationListener(pCtx: Activity, mv : MapView, site : SetOverlay)
+onLocationChanged(pLoc : Location) : void
+onProviderDisabled(provider : String) : void
+onProviderEnabled(provider : String) : void
+onStatusChanged(provider : String, status : int, extras : Bundle) : void
MyGeoPoint
-latitude: double
-longitude: double
+MyGeoPoint(latitude: double, longitude: double)
DBHelp
DB PATH : String = "/data/data/com.alp.db/databases/"
DATABASE_NAME : String = "db.db"
DATABASE_VERSION : int = 1
+sDb : SQLiteDatabase
+CREATE_TABLE_CMD : String = "create table if not exists "
+ " (_id INTEGER PRIMARY KEY, company id INTEGER, shop_name TEXT, open_hour TEXT, close_hour TEXT, latitude NUMERIC, longitude NUMERIC, traffic_name TEXT, traffic_latitude NUMERIC, traffic_longitude NUMERIC, display_name NUMERIC, news NUMERIC, menu NUMERIC, coupon NUMERIC, queue NUMERIC, booking NUMERIC, map NUMERIC, feedback NUMERIC, loyalty NUMERIC
+DBHelp(pCtx: Context)
+onCreate(db : SQLiteDatabase) : void
+onUpgrade(db : SQLiteDatabase, oldVersion: int, newVersion: int) : void
+copyDatabase() : void
+close() : void
+deleteAll() : void
+replaceValueValues : ContentValues) : long
+getDb() : SQLiteDatabase
ShopEntry
-shop_id : int
-company_id : int
-shop_name : String
-open_hour : String
-close_hour : String
-latitude : double
-longitude : double
-traffic_name : String
-traffic_latitude : double
-traffic_longitude : double
-display_name : String
-display_latitude : double
-ch_url : String
-news : int
-menu : int
-coupon : int
-queue : int
-booking : int
-feedback : int
-loyalty : int
+ShopEntry(pShop_id : int, pCompany_id : int, pShop_name : String, pOpen_hour : String, pClose_hour : String, pLatitude : double, pLongitude : double, pTraffic_name : String, pTraffic_latitude : double, pTraffic_longitude : double, pDisplay_name : String, pDisplay_latitude : double, pCh_url : String, pNews : int, pMenu : int, pCoupon : int, pQueue : int, pBooking : int, pMap : int, pFeedback : int, pLoyalty : int)



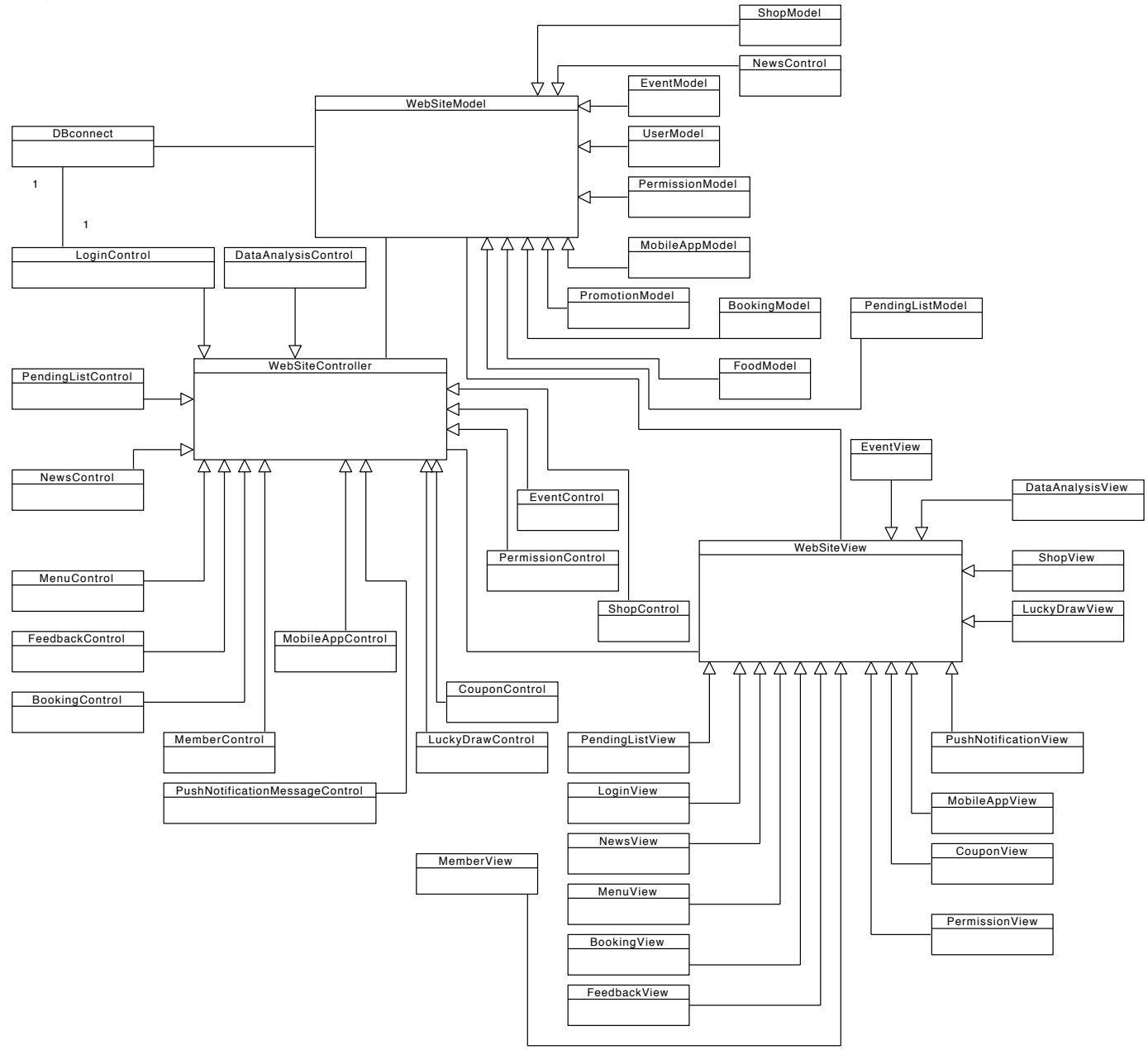
android
fyp
booking
BookDetailActivity
-date : TextView
-time : TextView
-seats : TextView
-status : TextView
-bookingId : String
-cancelButton : Button
-deviceId : String
-confirmButton : Button
+onCreate(icicle : Bundle) : void
+toMenu(v : View) : void
+confirmBooking(bookingID : String, deviceID : Stri...
+cancelBooking(bookingID : String, deviceID : Strin...
YesNoDialog
~title : TextView
~message : TextView
~yes : Button
~no : Button
+onCreate(icicle : Bundle) : void
LoadingTask
#onProgressUpdate(progress : Integer ...) : ...
#onPostExecute(result : Long) : void
#doInBackground(arg0 : Object ...) : Object
YesNoWaitingDialog
~title : TextView
~message : TextView
~yes : Button
~no : Button
~waiting : Button
+onCreate(icicle : Bundle) : void
BookActivity
-bYear : int
-bMonth : int
-bDay : int
-bHour : int
~name : TextView
~seat : TextView
~phone : TextView
~normalRequirement : TextView
~requirement : TextView
~mDeviceID : String
+dateTime : EditText
~selection : List<HashMap<String, Object>>
~suggestion : List<HashMap<String, Object>>
+onCreate(icicle : Bundle) : void
+getProperty() : List<HashMap<String, Object>>
-book(deviceID : String) : void
-getSuggestion(suggestion : JSONArray) : List<HashMap<Strin...
+getChosePropertyI() : String
+checkCreate(s : String) : boolean
+setDateTime(bYear : int, bMonth : int, bDay : int, bHour : int, b...
+toMenu(v : View) : void
#onPause() : void
#onResume() : void
#onStart() : void
+updateSelection() : void
SuggestionActivity
-listItem : List<HashMap<String, Object>>
-list : ListView
-listItemAdapter : BaseAdapter
-c : Context
+handler : Handler
+onCreate(icicle : Bundle) : void
+toMenu(v : View) : void
#onPause() : void
#onResume() : void
#onStart() : void
+setView() : void
BookSelectionDialog
-list : ListView
~selection : List<HashMap<String, Object...
+onCreate(icicle : Bundle) : void
#onDestroy() : void
SuggestionBaseAdapter
-layoutInflater : LayoutInflater
-list : List<HashMap<String, Object>>
-current : String
-context : Context
+SuggestionBaseAdapter(context : Context, list : List<HashMap<S...
+getCount() : int
+getItem(position : int) : Object
+getItemId(position : int) : long
+getView(position : int, convertView : View, parent : ViewGroup) : ...
BookSelectionDialogAdapter
-layoutInflater : LayoutInflater
-list : List<HashMap<String, Object>>
-current : String
-context : Context
+BookSelectionDialogAdapter(context : Context, list : List<HashMap<...
+getCount() : int
+getItem(position : int) : Object
+getItemId(position : int) : long
+getView(position : int, convertView : View, parent : ViewGroup) : View
DateTimeDialog
~datePicker : DatePicker
~timePicker : TimePicker
-bYear : int
-bMonth : int
-bDay : int
-bHour : int
-bMin : int
+onCreate(icicle : Bundle) : void
ViewBookActivity
-listItem : List<HashMap<String, Object>>
-list : ListView
-listItemAdapter : BaseAdapter
-c : Context
+handler : Handler
+onCreate(icicle : Bundle) : void
-getBookingid(deviceId : String) : JSONArray
+refresh(mDeviceID : String, context : Context)...
+onCreateOptionsMenu(menu : Menu) : boolean
+onOptionsItemSelected(item : MenuItem) : bo...
+toMenu(v : View) : void
#onPause() : void
#onResume() : void
#onStart() : void
+setView() : void
BookBaseAdapter
-layoutInflater : LayoutInflater
-list : List<HashMap<String, Object>>
-current : String
-context : Context
+BookBaseAdapter(context : Context, list : List<HashMap<Stri...
+getCount() : int
+getItem(position : int) : Object
+getItemId(position : int) : long
+getView(position : int, convertView : View, parent : ViewGrou...


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
android
fyp
push
ActivityLevel
~firstLevel : boolean
~secondLevel : boolean
~thirdLevel : boolean
~forthLevel : boolean
+ActivityLevel(first : boolean, second : boolean, third ...
+isFirstLevel() : boolean
+isForthLevel() : boolean
+isSecondLevel() : boolean
+isThirdLevel() : boolean
+firstLevel() : void
+secondLevel() : void
+thirdLevel() : void
+forthLevel() : void
Dialog
~title : TextView
~message : TextView
~ok : Button
+onCreate(icicle : Bundle) : void
Push
-c : Context
-mDeviceID : String
-c2 : Context
-c3 : Context
-c4 : Context
+Push(c : Context, mDeviceID : String)
+close() : void
PushService
+message : String
+TAG : String = "PushService"
~myHandler : Handler
-mqttHost : String = "a970032.no-ip.org"
-MQTT_BROKER_PORT_NUM : int = 1883
-MQTT_PERSISTENCE : MqttPersistence = null
-MQTT_CLEAN_START : boolean = true
-MQTT_KEEP_ALIVE : short = 60 * 60
-MQTT_QUALITIES_OF_SERVICE : int[] = {0}
-MQTT_QUALITY_OF_SERVICE : int = 0
-MQTT_RETAINED_PUBLISH : boolean = false
+MQTT_CLIENT_ID : String = ""
-ACTION_START : String = MQTT_CLIENT_ID + ".START"
-ACTION_STOP : String = MQTT_CLIENT_ID + ".STOP"
-ACTION_KEEPALIVE : String = MQTT_CLIENT_ID + ".KEEP_ALIVE"
-ACTION_RECONNECT : String = MQTT_CLIENT_ID + ".RECONNECT"
-mLog : ConnectionLog
-mConnMan : ConnectivityManager
-mNotifMan : NotificationManager
-mStarted : boolean
-KEEP_ALIVE_INTERVAL : long = 1000 * 60 * 28
-INITIAL_RETRY_INTERVAL : long = 1000 * 10
-MAXIMUM_RETRY_INTERVAL : long = 1000 * 60 * 30
-mPrefs : SharedPreferences
+PREF_STARTED : String = "isStarted"
+PREF_DEVICE_ID : String = "deviceID"
+PREF_RETRY : String = "retryInterval"
+NOTIF_TITLE : String = "Push"
-NOTIF_CONNECTED : int = 0
-mConnection : MQTTConnection
-mStartTime : long
-paused : boolean
-deviceID : String
-mConnectivityChanged : BroadcastReceiver = new BroadcastReceiver() {
@Override
public void onReceive(Context context, Intent intent) {
// Get network info
NetworkInfo info = (NetworkInfo) intent.getParcelableExtra(ConnectivityManager.EXTRA_NETWORK_INFO);
// Is there connectivity?
boolean hasConnectivity = (info != null && info.isConnected()) ? true : false;
log("Connectivity changed: connected=" + hasConnectivity);
if (hasConnectivity) {
reconnectIfNecessary();
} else if (mConnection != null) {
// if there no connectivity, make sure MQTT connection is destroyed
mConnection.disconnect();
cancelReconnect();
mConnection = null;
}
}
}
+actionStart(ctx : Context) : void
+actionStop(ctx : Context) : void
+actionPing(ctx : Context) : void
+onCreate() : void
-handleCrashedService() : void
+onDestroy() : void
+onStart(intent : Intent, startId : int) : void
+onBind(intent : Intent) : IBinder
-log(message : String) : void
-log(message : String, e : Throwable) : void
-wasStarted() : boolean
-setStarted(started : boolean) : void
-start() : void
-stop() : void
-connect() : void
-keepAlive() : void
-startKeepAlives() : void
-stopKeepAlives() : void
+scheduleReconnect(startTime : long) : void
+cancelReconnect() : void
-reconnectIfNecessary() : void
-showNotification(text : String) : void
-isNetworkAvailable() : boolean
MQTTConnection
~mqttClient : IMqttClient = null
+MQTTConnection(brokerHostName : String, initTopic : String)
+disconnect() : void
-subscribeToTopic(topicName : String) : void
-publishToTopic(topicName : String, message : String) : void
+connectionLost() : void
+publishArrived(topicName : String, payload : byte [], qos : int, retained : boolean) : void
+sendKeepAlive() : void
ConnectionLog
-mPath : String
-mWriter : Writer
-TIMESTAMP_FMT : SimpleDateFormat = new Simple...
+ConnectionLog()
+ConnectionLog(basePath : String)
#open(basePath : String) : void
-getTodayString() : String
+getPath() : String
1
-mLog

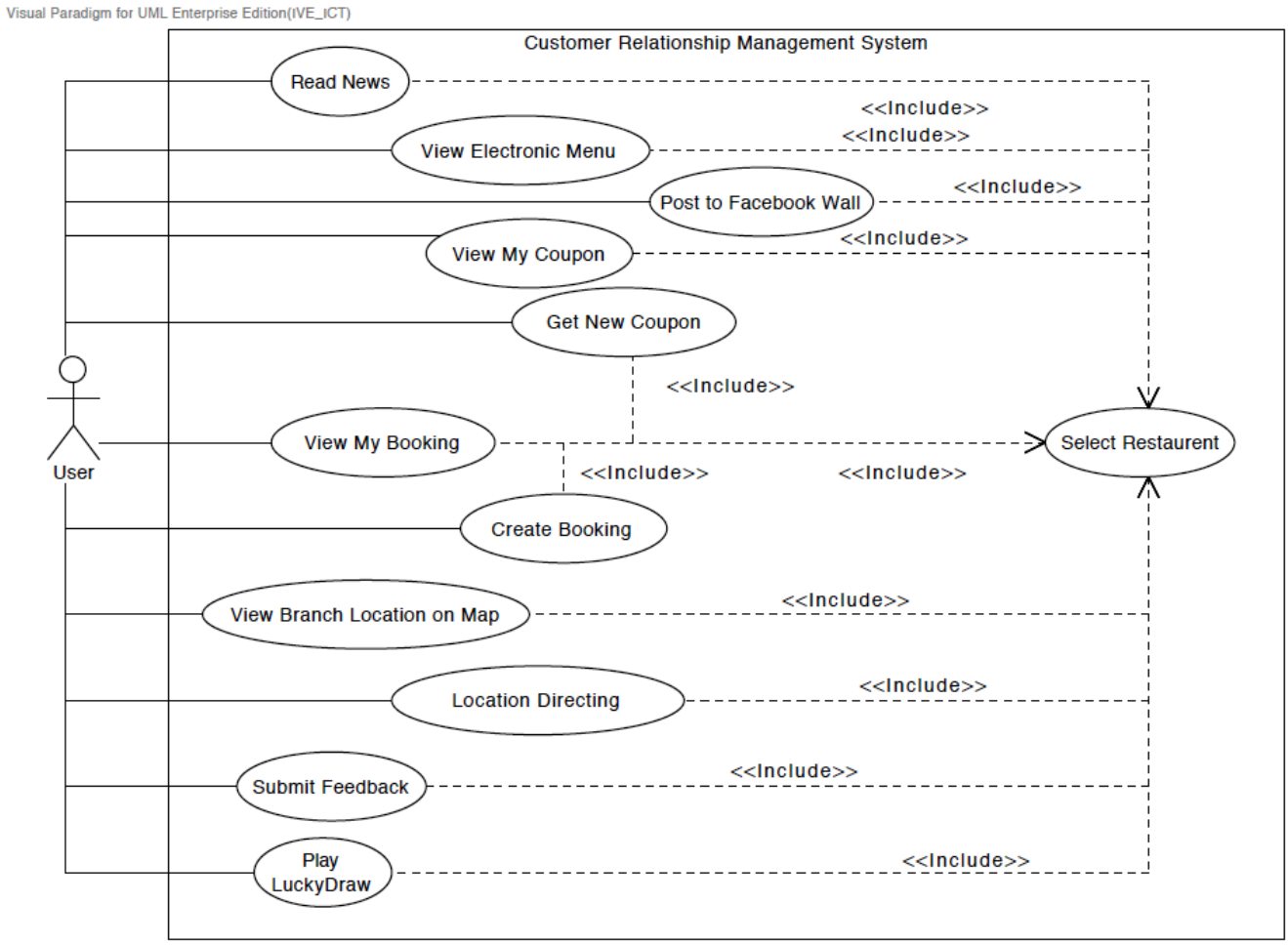

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
android
fyp
twitter
ImageSender
~url : String
#onPreExecute() : void
#doInBackground(urls : URL ...) : Long
#onProgressUpdate(progress : Integer ...) : void
#onPostExecute(result : Long) : void
TwitterMessageDialog
~bundle : Bundle
~message : EditText
-TCONSUMER_KEY : String = "AmXbbNKVGHhTsMnTTdX0Qw"
-TCONSUMER_SECRET : String = "IngNNeslhKsh9Ro2xoXQHYED707bTM4Isp0jYXvk"
-TWITPIC_API_KEY : String = "9383d29e4378b161594b4cd84c66b2a5"
-TAG : String = "AndroidTwitpic"
-twitter : TwitterApp = null
-mProgressDialog : ProgressDialog
~content : String
~mPath : String
-mTwLoginDialogListener : TwDialogListener = new TwitterApp.TwDialogListener() {
@Override
public void onComplete(String value) {
ImageSender imageSender = new ImageSender();
imageSender.execute();
}


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
android
fyp
setting
HostActivity
~ok : Button
~cancel : Button
~email : EditText
~password : EditText
+c : Activity
+onCreate(icicle : Bundle) : void
SettingActivity
~account : Button
~host : Button
+onCreate(icicle : Bundle) : void
+toMenu(v : View) : void
LoginDialog
~ok : Button
~cancel : Button
~email : EditText
~password : EditText
+c : Activity
+onCreate(icicle : Bundle) : void
-login(email : String, password : String, context : Context) : void
RegisterDialog
~ok : Button
~cancel : Button
~email : EditText
~password : EditText
~surname : EditText
~firstname : EditText
~gender : Spinner
~birthday : EditText
-myYear : int
-myMonth : int
-myDay : int
+onCreate(icicle : Bundle) : void
-register(email : String, password : String, surname : String, firstname : String, gender : String, birthday : String, context : Context) : void
SelectDialog
~login : Button
~register : Button
~email : EditText
~password : EditText
+c : Context
+onCreate(icicle : Bundle) : void
+refresh() : void

Class Diagram (Website)



Use Case Diagram (Mobile)



Actor Description (Mobile)

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

Initial Use Case Description (Mobile)

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

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

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

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

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

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

Use case name:	LuckyDraw
Use case ID:	UC-0007
Actor(s):	Client
Brief description:	Many users may view the prize of the lucky draw.

Use case name:	Play Lucky draw
Use case ID:	UC-0008
Actor(s):	Client
Brief description:	Many users may use the point to play the lucky draw.

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

Use case name:	Create Booking
Use case ID:	UC-0010
Actor(s):	Client
Brief description:	The user can send a book request for the table for a specific time. If table is available, the booking will be approved automatically. If no table is available, she needs to wait for the reply from the restaurant staff.

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

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

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

Refined Use Case Description (Mobile)

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

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

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

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

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

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

Use case name:	LuckyDraw
Use case ID:	UC-0007
Actor(s):	Client
Brief description:	Users can use the bonus point to play lucky draw
Preconditions:	The user is logged in
Flow of events:	1. Include (select Restaurant). 2. The user click the button “LuckyDraw” 3. The system a list of lucky draw available 4. The user select a lucky draw 5. The system show a list of prize
Postconditions:	/
Alternative flows and exceptions:	
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

Use case name:	Play LuckyDraw
Use case ID:	UC-0008
Actor(s):	Client
Brief description:	Users can use the bonus point to play lucky draw
Preconditions:	The user is logged in
Flow of events:	1. Include (select Restaurant). 2. The user click the button “LuckyDraw” 3. The system a list of lucky draw available 4. The user select a lucky draw 5. The system show a list of prize 6. The user click “Draw” 7. The system show a prize covered by a silver cover
Postconditions:	/
Alternative flows and exceptions:	
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

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

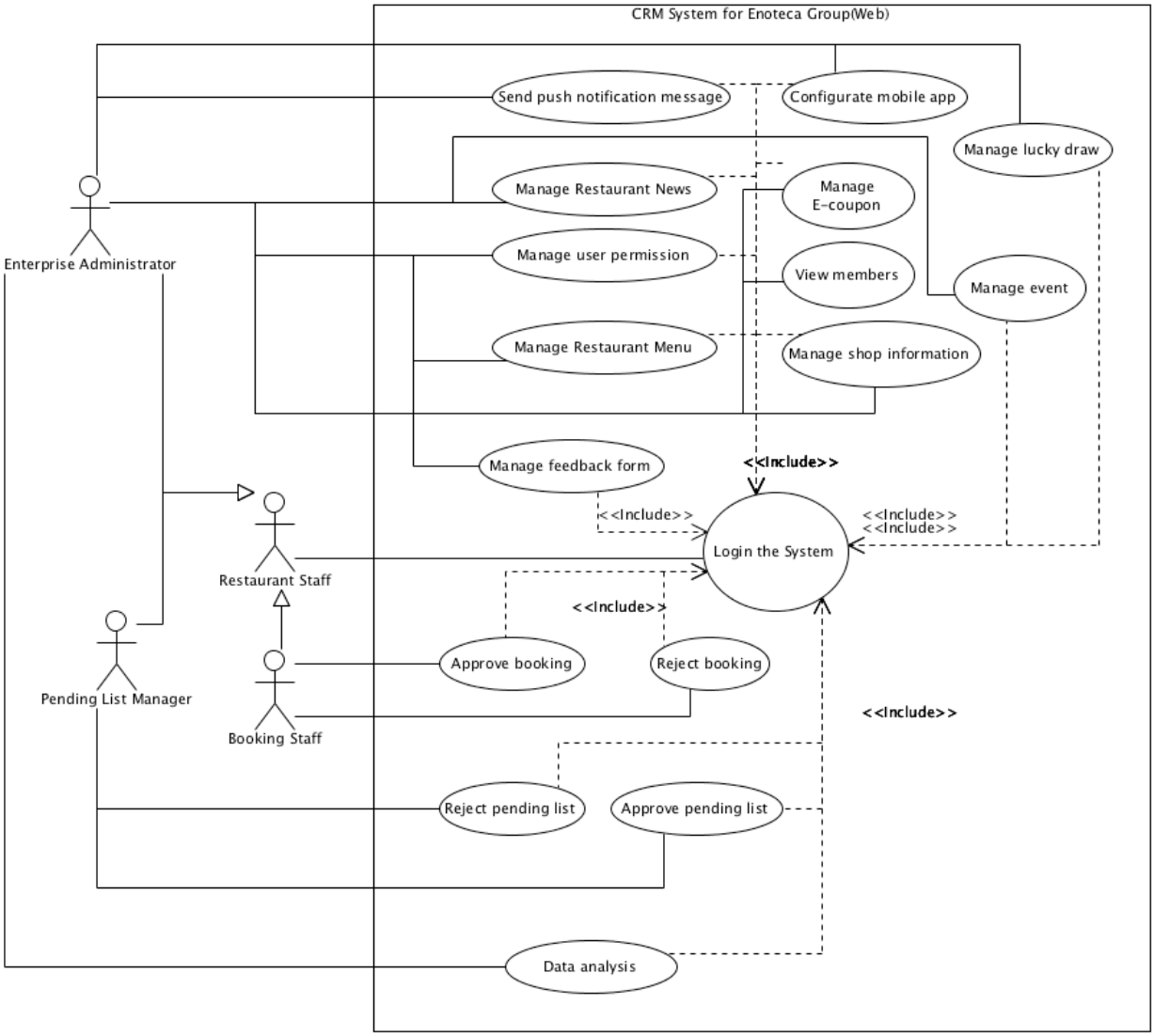
Use case name:	Create Booking
Use case ID:	UC-0010
Actor(s):	Client
Brief description:	The user can book a table for a specific time.
Preconditions:	/
Flow of events:	1. Include (select Restaurant). 2. The user clicks the button “Booking” and clicks the button of “Create Booking”. 3. The system shows form for creates booking. 4. The user enters their name, seat required, their contact phone number, date, special requirement and click “Book”. 5. The system check if there is table available match the requirement 6. The system shows the success message and return to page “View Booking”. 7. The user want for the booking states after the result sent back to the client.
Postconditions:	/
Alternative flows and exceptions:	If the network is not available, the client cannot view the booking.
Assumptions:	The user use iPhone or Android devices.
Source:	Functional Requirement from the Proposal.

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

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

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

Use Case Diagram (Website)



Actor Description (Website)

Actor Name:	Enterprise Administrator
Description:	Enterprise Administrator is the end-user interacts with the system by using the web site. Usually, the enterprise administrator contains the full control permission in the website. Each restaurant should have not much people having this permission.

Actor Name:	Pending List Manager
Description:	Pending List Manager is the end-user interacts with the system by using the web site. All the updated data in the website will store in the database pending list, all the data needed to be approve before updating to the customer mobile.

Actor Name:	Booking Staff
Description:	Booking Staff is the end-user interacts with the system by using the web site. The booking staffing is used to control the booking list only in the website, she can approve or reject booking from the mobile or to avoid some abuse booking. Making communication with to customer by using the list of data.

Actor Name:	Restaurant Staff
Description:	Restaurant Staff is the end-user interacts with the system by using the web site. A general staff in the restaurant.

Initial Use Case Description (Website)

Use case name:	Login the system
Use case ID:	UC-0001
Actor(s):	Restaurant Staff
Brief description:	Restaurant Staff needs to login the system before doing any function.

Use case name:	Send push notification
Use case ID:	UC-0002
Actor(s):	Enterprise Administrator
Brief description:	The enterprise administrator can send push notification to the mobile customer.

Use case name:	Configure mobile app
Use case ID:	UC-0003
Actor(s):	Enterprise Administrator
Brief description:	The user make modification on the mobile app

Use case name:	Manage lucky draw
Use case ID:	UC-0004
Actor(s):	Enterprise Administrator
Brief description:	User can make modification on the lucky draw information to the customer.

Use case name:	Manage Restaurant News
Use case ID:	UC-0005
Actor(s):	Enterprise Administrator
Brief description:	The user can modify the restaurant news.

Use case name:	Manage E-coupon
Use case ID:	UC-0006
Actor(s):	Enterprise Administrator
Brief description:	A Restaurant staff can insert, update or delete the information of E-coupon.

Use case name:	Manage user permission
Use case ID:	UC-0007
Actor(s):	Enterprise Administrator
Brief description:	The enterprise administrator can modify the user permission for different group of people.

Use case name:	View members
Use case ID:	UC-0008
Actor(s):	Enterprise Administrator
Brief description:	The enterprise administrator can view the VIP who has registered in the mobile.

Use case name:	Manage event
Use case ID:	UC-0009
Actor(s):	Enterprise Administrator
Brief description:	The user can manage the information of the events on the website.

Use case name:	Manage Restaurant Menu
Use case ID:	UC-0010
Actor(s):	Enterprise Administrator
Brief description:	The user can manage the restaurant menu information on the mobile for the customer.

Use case name:	Manage shop information
Use case ID:	UC-0011
Actor(s):	Enterprise Administrator
Brief description:	The user can manage shop information including the location, opening hours, shop details.

Use case name:	Manage feedback form
Use case ID:	UC-0012
Actor(s):	Enterprise Administrator
Brief description:	The user can manage the feedback on the mobile for the customer

Use case name:	Approve booking
Use case ID:	UC-0013
Actor(s):	Booking Staff
Brief description:	The user can approve the customer booking sent from the mobile.

Use case name:	Reject booking
Use case ID:	UC-0014
Actor(s):	Booking Staff
Brief description:	The user can reject the customer booking sent from the mobile.

Use case name:	Reject pending list
Use case ID:	UC-0015
Actor(s):	Pending List Manager
Brief description:	The user can reject the pending list. The pending list stores the modified data. Every modified data needed to be approve before upload to the mobile app to avoid the mistake of typing.

Use case name:	Approve pending list
Use case ID:	UC-0016
Actor(s):	Pending List Manager
Brief description:	The user can approve the pending list. The pending list stores the modified data. Every modified data needed to be approve before upload to the mobile app to avoid the mistake of typing.

Use case name:	Data analysis
Use case ID:	UC-0017
Actor(s):	Enterprise Administrator
Brief description:	The user can go to the data center and view all the shop data analysis in customer feedback, customer orders

Refined Use Case Description (Website)

Use case name:	Login the system
Use case ID:	UC-0001
Actor(s):	Restaurant Staf
Brief description:	Restaurant Staff needs to login the system before doing any function.
Preconditions:	/
Flow of events:	1. Go to the website index page. 2. Input username and password. 3. Login the system.
Postconditions:	
Alternative flows and exceptions:	If the username / password incorrect, the staff cannot login
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Send push notification
Use case ID:	UC-0002
Actor(s):	Enterprise Administrator
Brief description:	The enterprise administrator can send push notification to the mobile customer.
Preconditions:	
Flow of events:	1. Include (Loin the system) 2. The user click the button “Push Notification” 3. The system shows a message box 4. The user insert the message 5. The system send the message to the user
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Configure mobile app
Use case ID:	UC-0003
Actor(s):	Enterprise Administrator
Brief description:	The user make modification on the mobile app
Preconditions:	
Flow of events:	1. Include (Login) 2. The user click the button “mobile app” 3. The system shows the functions. 4. The user modify it 5. The system change the appearance of the mobile app.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Manage lucky draw
Use case ID:	UC-0004
Actor(s):	Enterprise Administrator
Brief description:	User can make modification on the lucky draw information to the customer.
Preconditions:	
Flow of events:	1. Include (Login the system) 2. The user click the button “lucky draw” 3. The system shows the information to the user. 4. The user modifies the data. 5. The system modify the data in the mobile app.
Postconditions:	A Fool detail is posted to the user’s Facebook wall
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Manage Restaurant News
Use case ID:	UC-0005
Actor(s):	Enterprise Administrator
Brief description:	The user can modify the restaurant news.
Preconditions:	
Flow of events:	1. Include (Login) 2. The user click the restaurant news 3. The user modifies the restaurant news. 4. The system update the data for the customer
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Manage E-coupon
Use case ID:	UC-0006
Actor(s):	Enterprise Administrator
Brief description:	A Restaurant staff can insert, update or delete the information of E-coupon.
Preconditions:	Database must have data
Flow of events:	1. Include (Login) 2. The user goes to the E-coupon page. 3. The user modifies the E-coupon data. 4. The system update the E-coupon data for the customer.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Manage user permission
Use case ID:	UC-0007
Actor(s):	Enterprise Administrator
Brief description:	The enterprise administrator can modify the user permission for different group of people.
Preconditions:	The user must already in a user group
Flow of events:	1. Include (Login) 2. The user click the button “Permission” 3. The system shows the permission on different group. 4. The user can modify the permit or deny permission on the function.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	View members
Use case ID:	UC-0008
Actor(s):	Enterprise Administrator
Brief description:	The enterprise administrator can view the VIP who has registered in the mobile.
Preconditions:	
Flow of events:	1. Include (Login) 2. The user click the button “member” 3. The system shows the member list. 4. The user can view the details of the VIP or view the feedback posted by the customer.
Postconditions:	The user have join the queue.
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Manage event
Use case ID:	UC-0009
Actor(s):	Enterprise Administrator
Brief description:	The user can manage the information of the events on the website.
Preconditions:	
Flow of events:	1. Include (Login) 2. The user choose the event page 3. The system shows the calendar to the user. 4. The user can choose the day and select to event that do not let the customer to make booking on the specified timeslot.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Manage Restaurant Menu
Use case ID:	UC-0010
Actor(s):	Enterprise Administrator
Brief description:	The user can manage the restaurant menu information on the mobile for the customer.
Preconditions:	
Flow of events:	1. Include (Login) 2. The user goes to the menu page and then click choose the food information or menu information. 3. The system shows the information to the user. 4. The user insert, update or delete the food item in the website. 5. The system shows the success message.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Manage shop information
Use case ID:	UC-0011
Actor(s):	Enterprise Administrator
Brief description:	The user can manage shop information including the location, opening hours, shop details.
Preconditions:	
Flow of events:	1. Include (Login) 2. The user go to the shop management page 3. The system shows the shop data to the user. 4. The user then inserts a new shop or modify a shop information. 5. The user can select the location in the google map if needed. 6. The system tell the user successful.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Manage feedback form
Use case ID:	UC-0012
Actor(s):	Enterprise Administrator
Brief description:	The user can manage the feedback on the mobile for the customer
Preconditions:	The restaurant can let the customer to submit feedback in mobile
Flow of events:	1. Include(Login) 2. The user choose the feedback page 3. The system shows the feedback data to the user 4. The user can insert, update or delete the feedback form on mobile 5. The system update the database. 6. The system update the information on mobile.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Approve booking
Use case ID:	UC-0013
Actor(s):	Booking Staff
Brief description:	The user can approve the customer booking sent from the mobile.
Preconditions:	/
Flow of events:	1. Include (Login) 2. The user go to the booking status page. 3. The system shows the booking list to the user. 4. The user can print out the booking list. 5. The user can approve the booking if the booking is not approved by the system automatically. 6. The system update the result and show to the user. 7. The system then send push notification to the user on their mobile.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

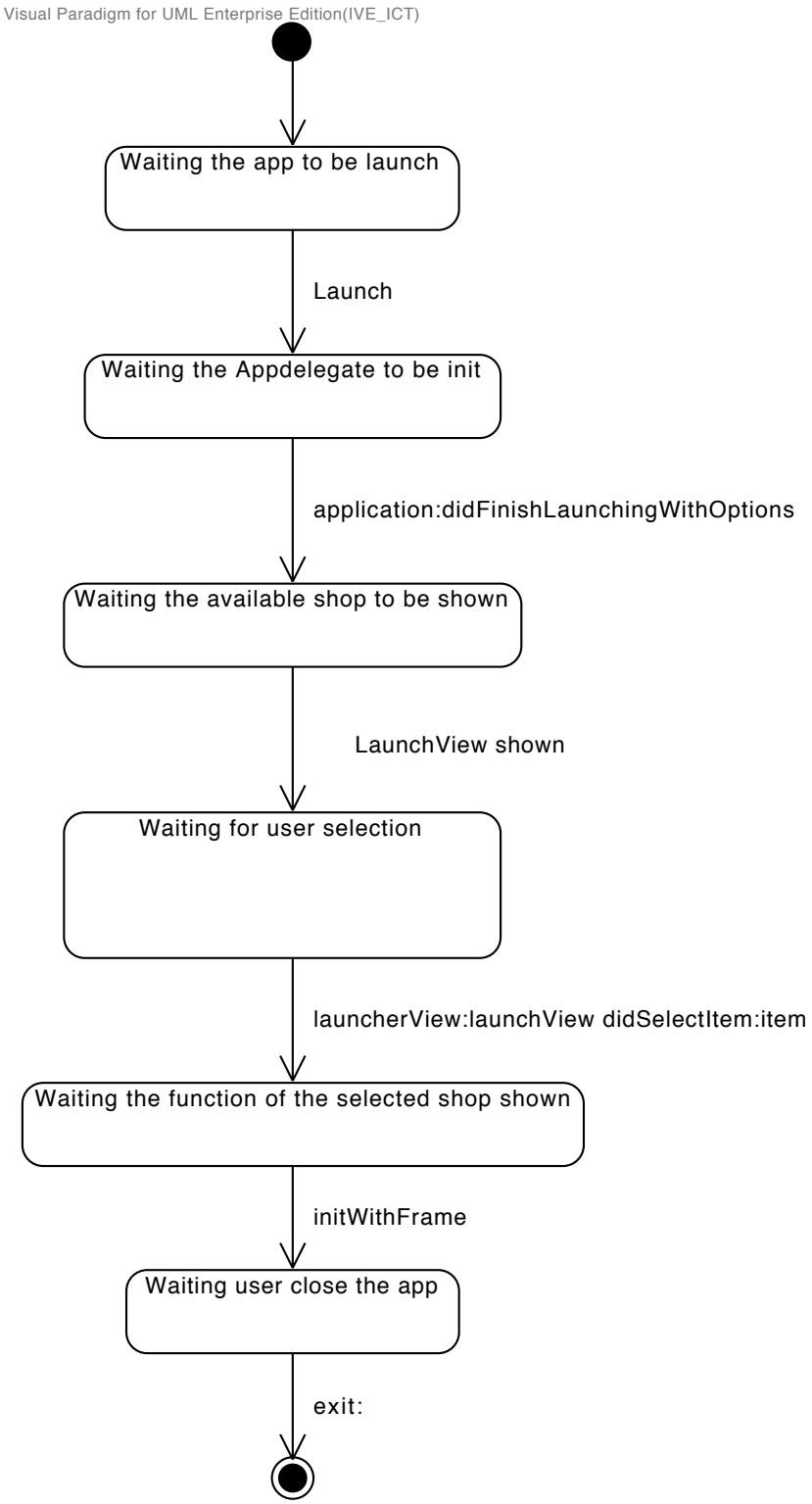
Use case name:	Reject booking
Use case ID:	UC-0014
Actor(s):	Booking Staff
Brief description:	The user can reject the customer booking sent from the mobile.
Preconditions:	/
Flow of events:	1. Include (Login) 2. The user goes to the booking status page. 3. The system shows the booking list to the user. 4. The user can print out the booking list. 5. The user can reject the booking if the booking is not valid. 6. The system updates the result and show to the user. 7. The system then send push notification to the user on their mobile.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Reject pending list
Use case ID:	UC-0015
Actor(s):	Pending List Manager
Brief description:	The user can reject the pending list. The pending list stores the modified data. Every modified data needed to be approve before upload to the mobile app to avoid the mistake of typing.
Preconditions:	/
Flow of events:	1. Include (Login) 2. The user goes to the pending list page. 3. The system shows the pending list to the user. 4. The user can view the pending list 5. The user can reject the pending list item if there is some problems of the data. 6. The system updates the result and show to the user. 7. The system keeps the record in the database.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

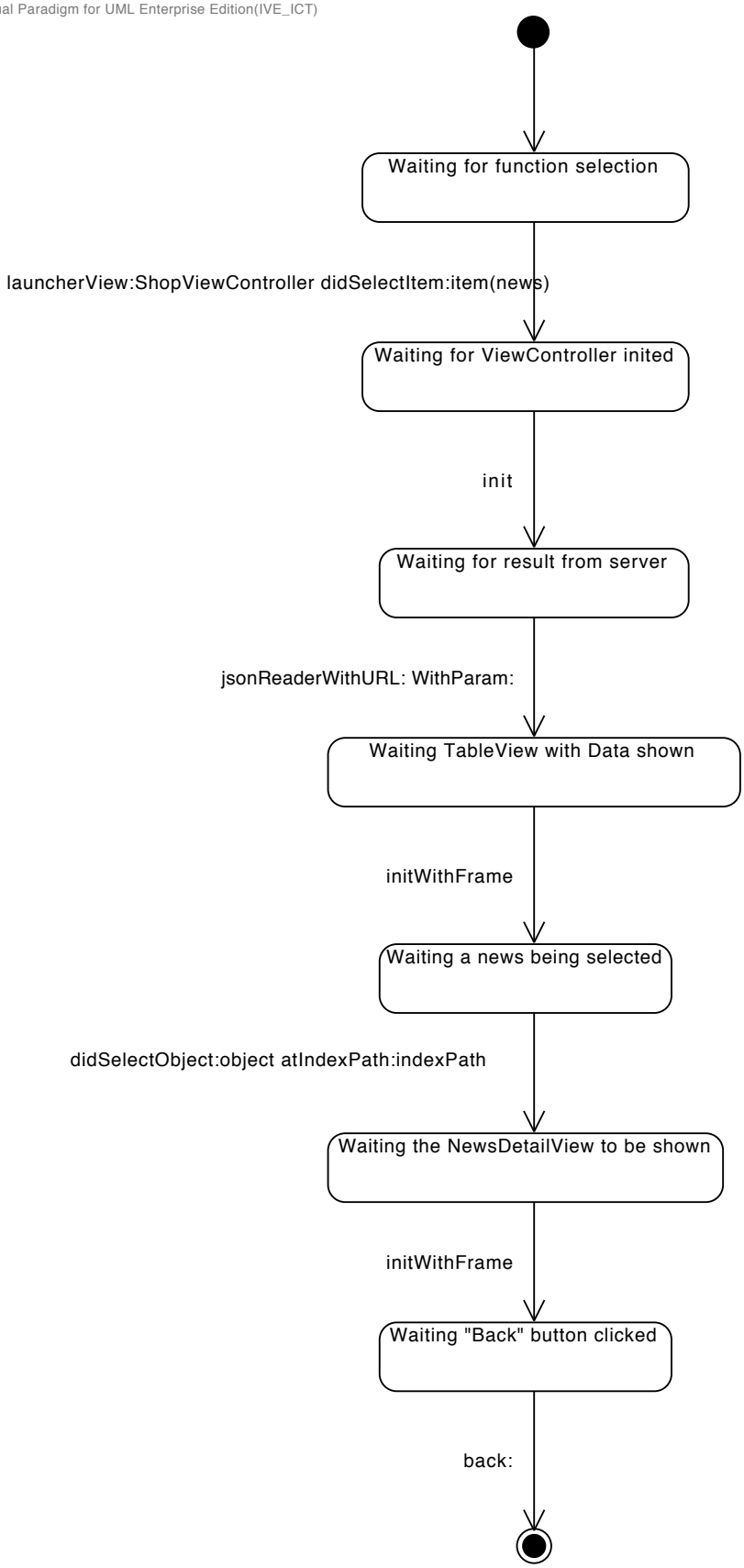
Use case name:	Approve pending list
Use case ID:	UC-0016
Actor(s):	Pending List Manager
Brief description:	The user can approve the pending list. The pending list stores the modified data. Every modified data needed to be approve before upload to the mobile app to avoid the mistake of typing.
Preconditions:	/
Flow of events:	1. Include (Login) 2. The user goes to the pending list page. 3. The system shows the pending list to the user. 4. The user can view the pending list 5. The user can approve the pending list item if the data is correct. 6. The system updates the result and show to the user. 7. The system keeps the record in the database. 8. The system update the mobile app information to the customers.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

Use case name:	Data analysis
Use case ID:	UC-0017
Actor(s):	Enterprise Administrator
Brief description:	The user can go to the data center and view all the shop data analysis in customer feedback, customer orders
Preconditions:	/
Flow of events:	1. Include (Login) 2. The user goes to the data center. 3. The system shows the data center to the users. 4. The user can follow the guide on the web and then input the information for data analysis in the database. 5. The system display the analyzed data to the user. 6. The user can download the csv report for the data.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	
Assumptions:	
Issue:	
Source:	

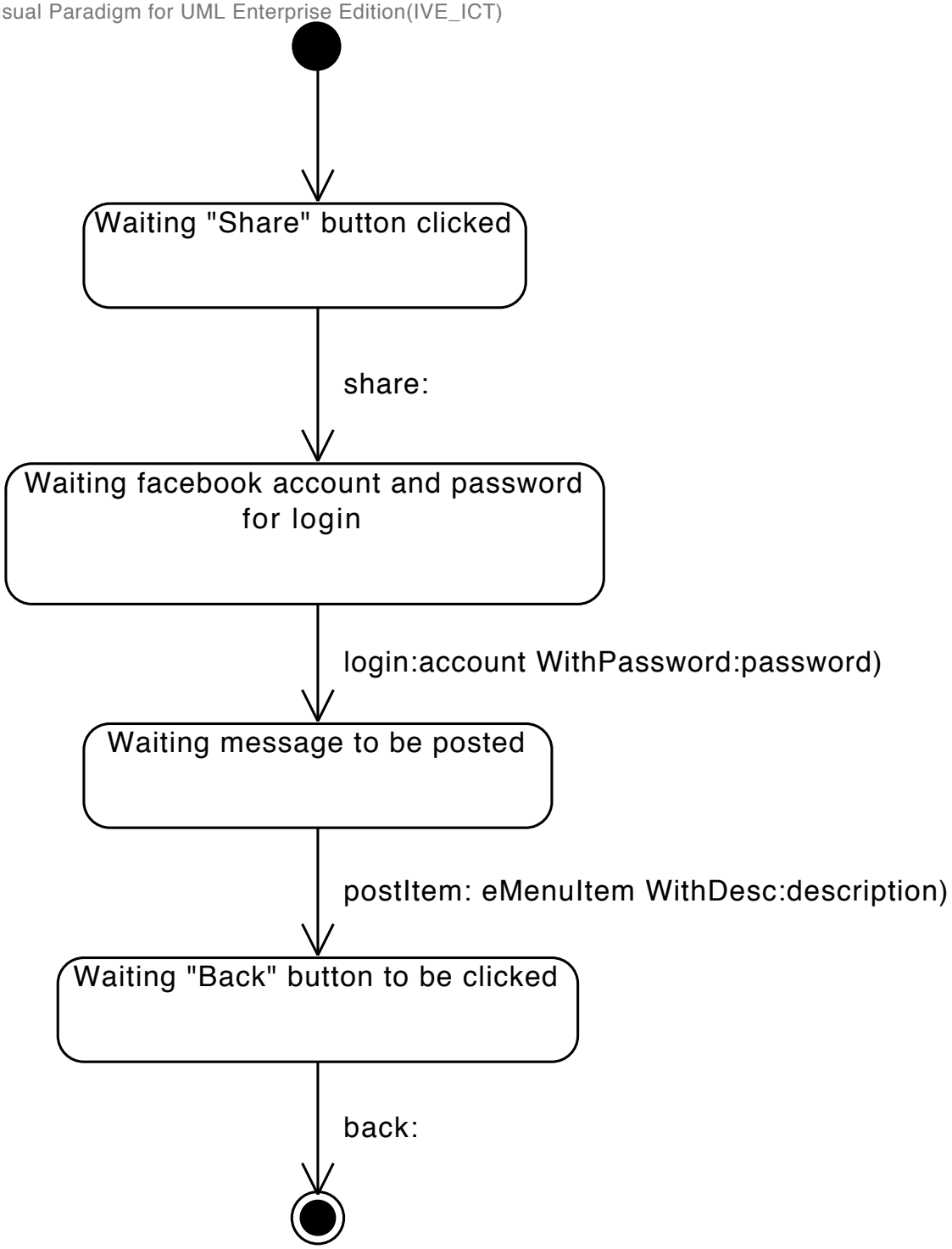
State Transition Diagram (iPhone)

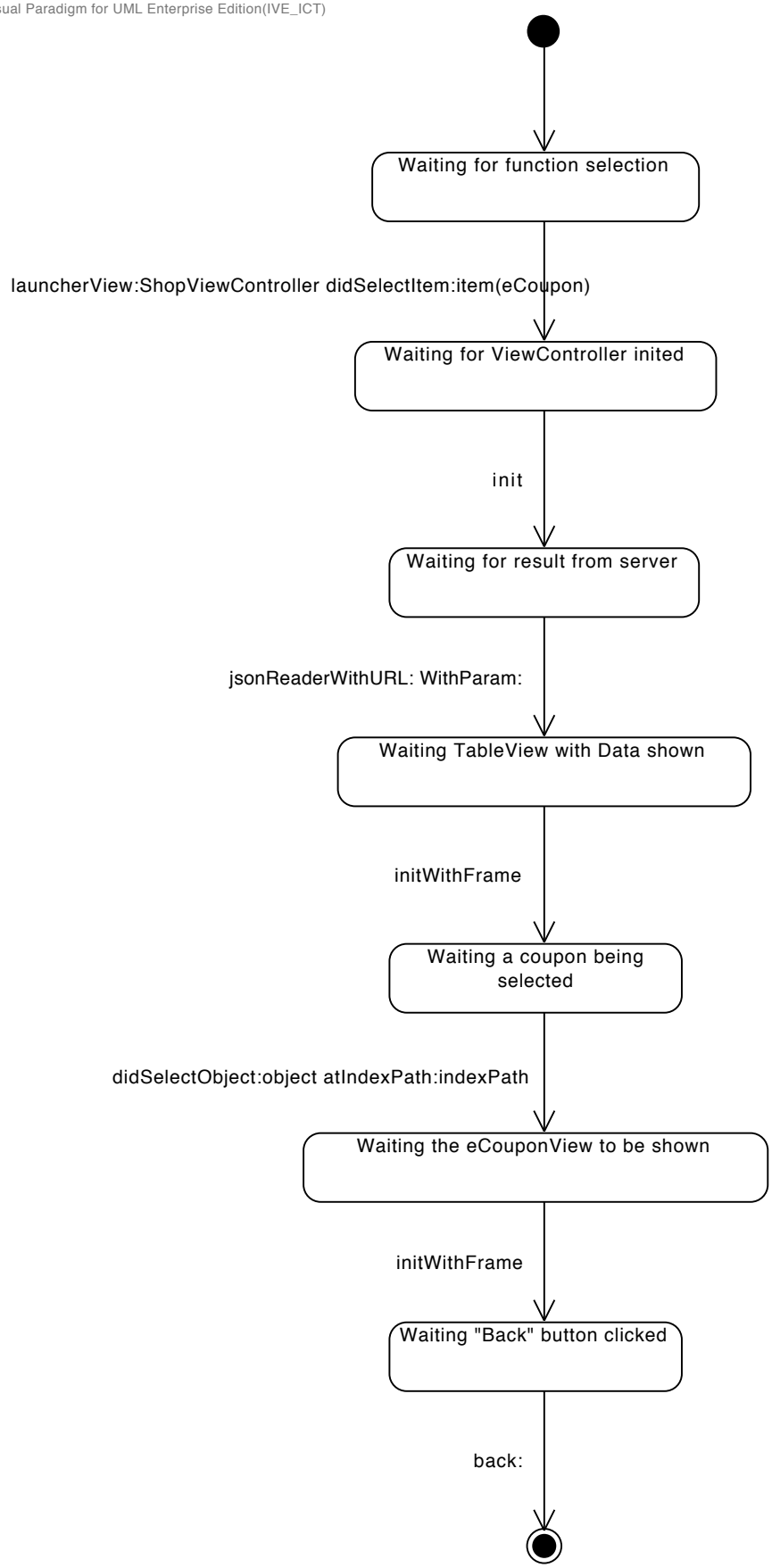


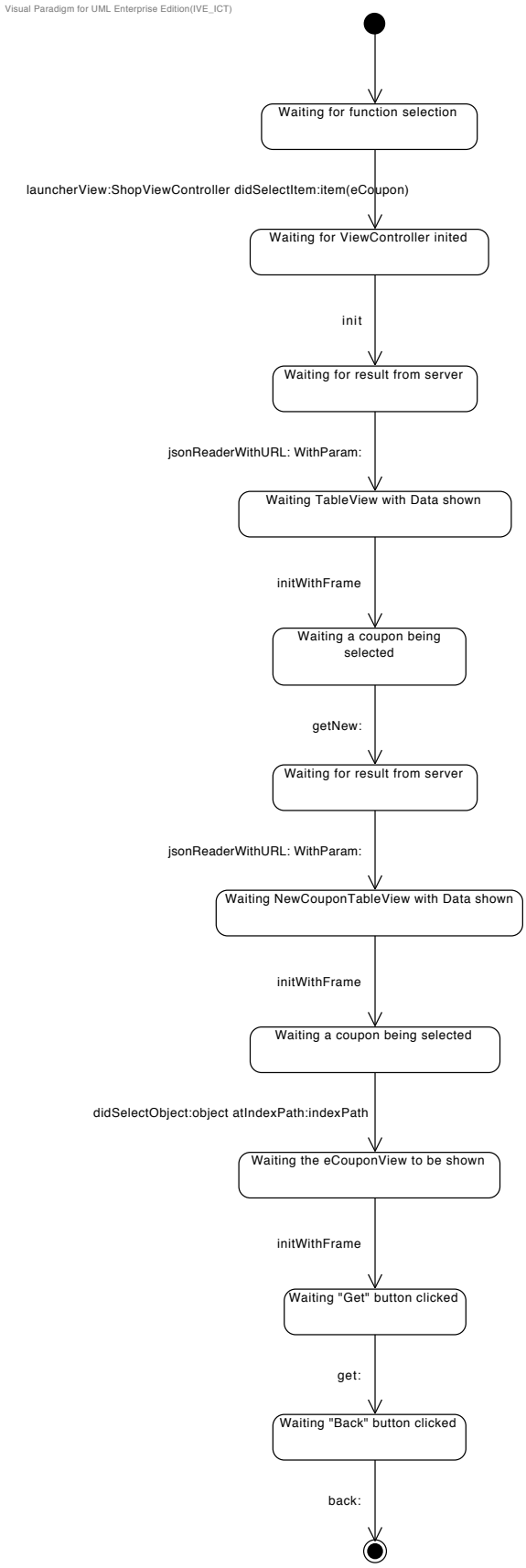
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

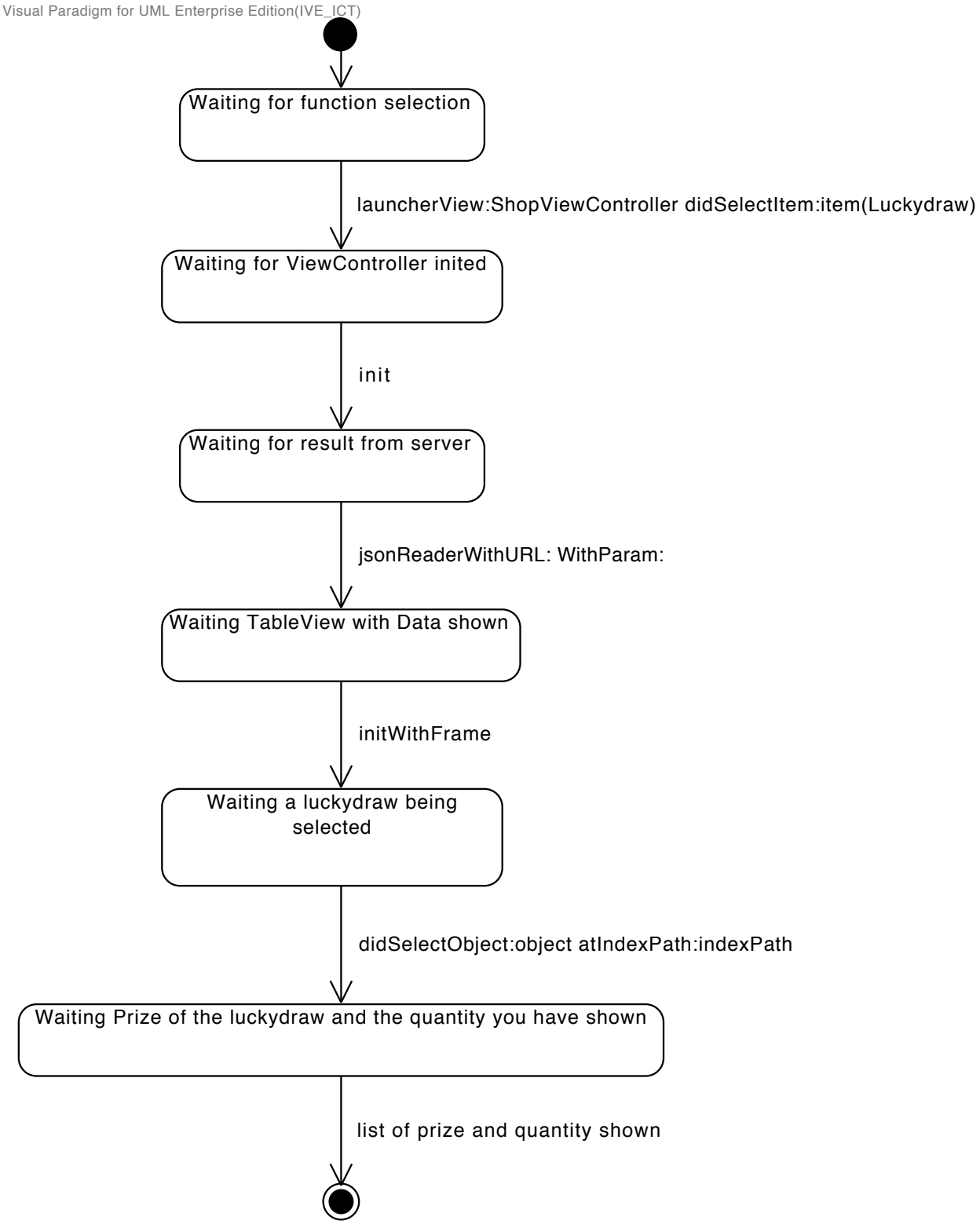


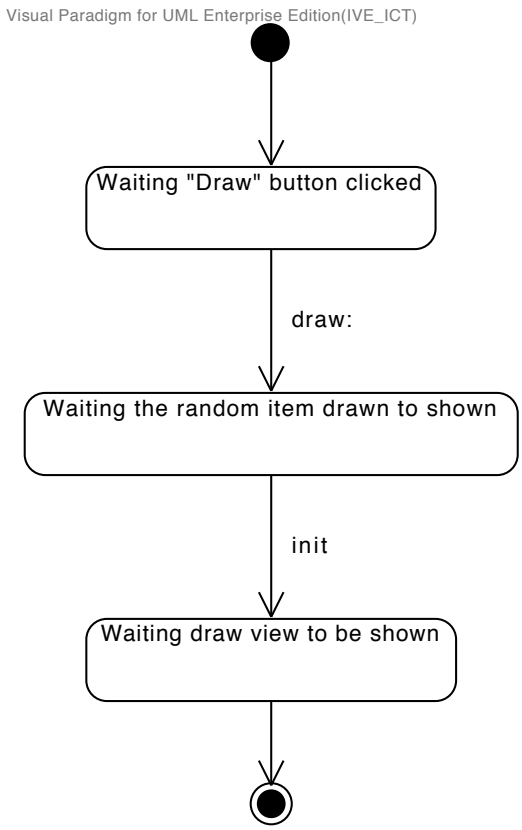
Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting for function selection
launcherView:ShopViewController didSelectItem:item(eMenu)
Waiting for ViewController inited
init
Waiting for result from server
jsonReaderWithURL: WithParam:
Waiting TableView with Data shown
initWithFrame
Waiting a item being selected
didSelectObject:object atIndexPath:indexPath
Waiting the eMenuItemView to be shown
initWithFrame
Waiting "Back" button clicked
back:


Waiting "Share" button clicked
share:
Waiting facebook account and password for login
login:account WithPassword:password)
Waiting message to be posted
postItem: eMenuItem WithDesc:description)
Waiting "Back" button to be clicked
back:

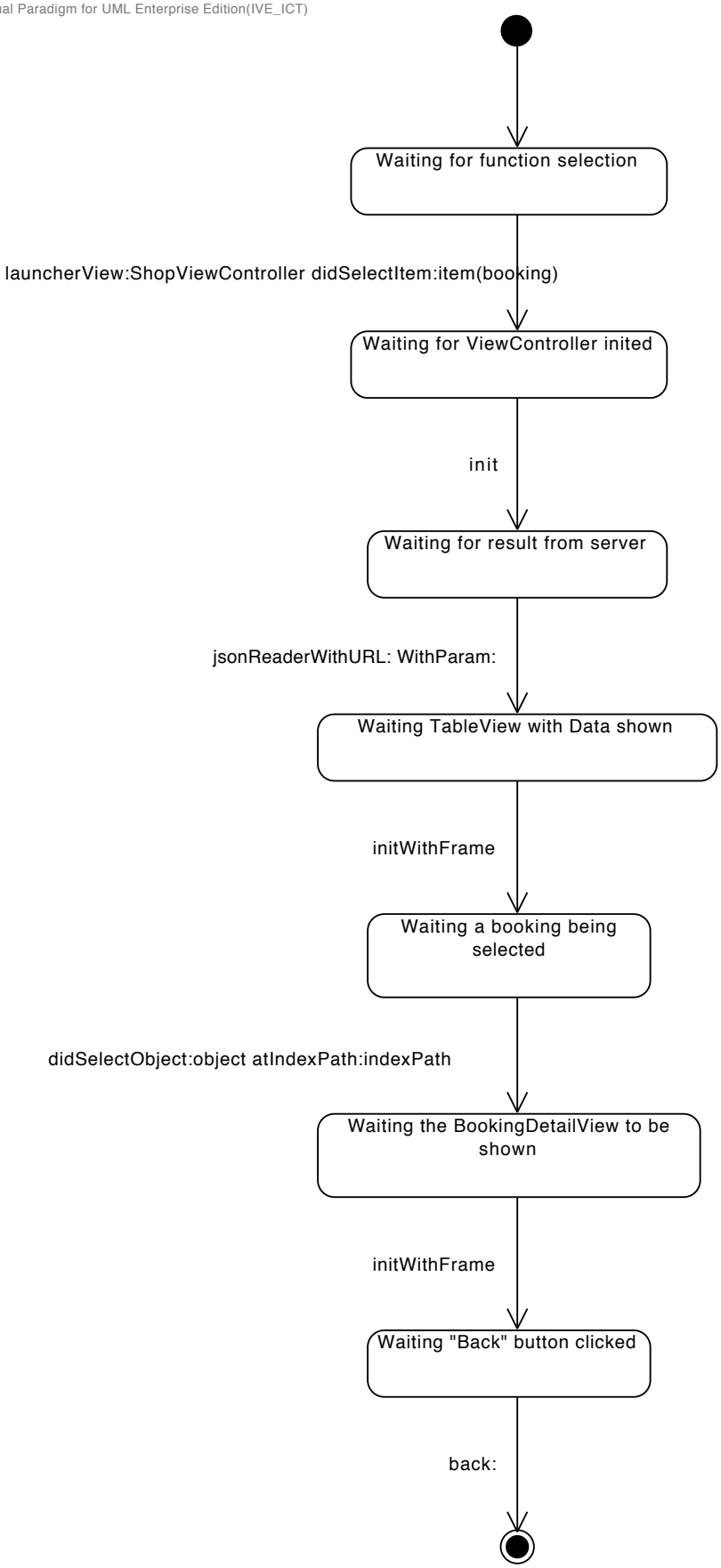

Waiting for function selection
launcherView:ShopViewController didSelectItem:item(eCoupon)
Waiting for ViewController inited
init
Waiting for result from server
jsonReaderWithURL: WithParam:
Waiting TableView with Data shown
initWithFrame
Waiting a coupon being selected
didSelectObject:object atIndexPath:indexPath
Waiting the eCouponView to be shown
initWithFrame
Waiting "Back" button clicked
back:

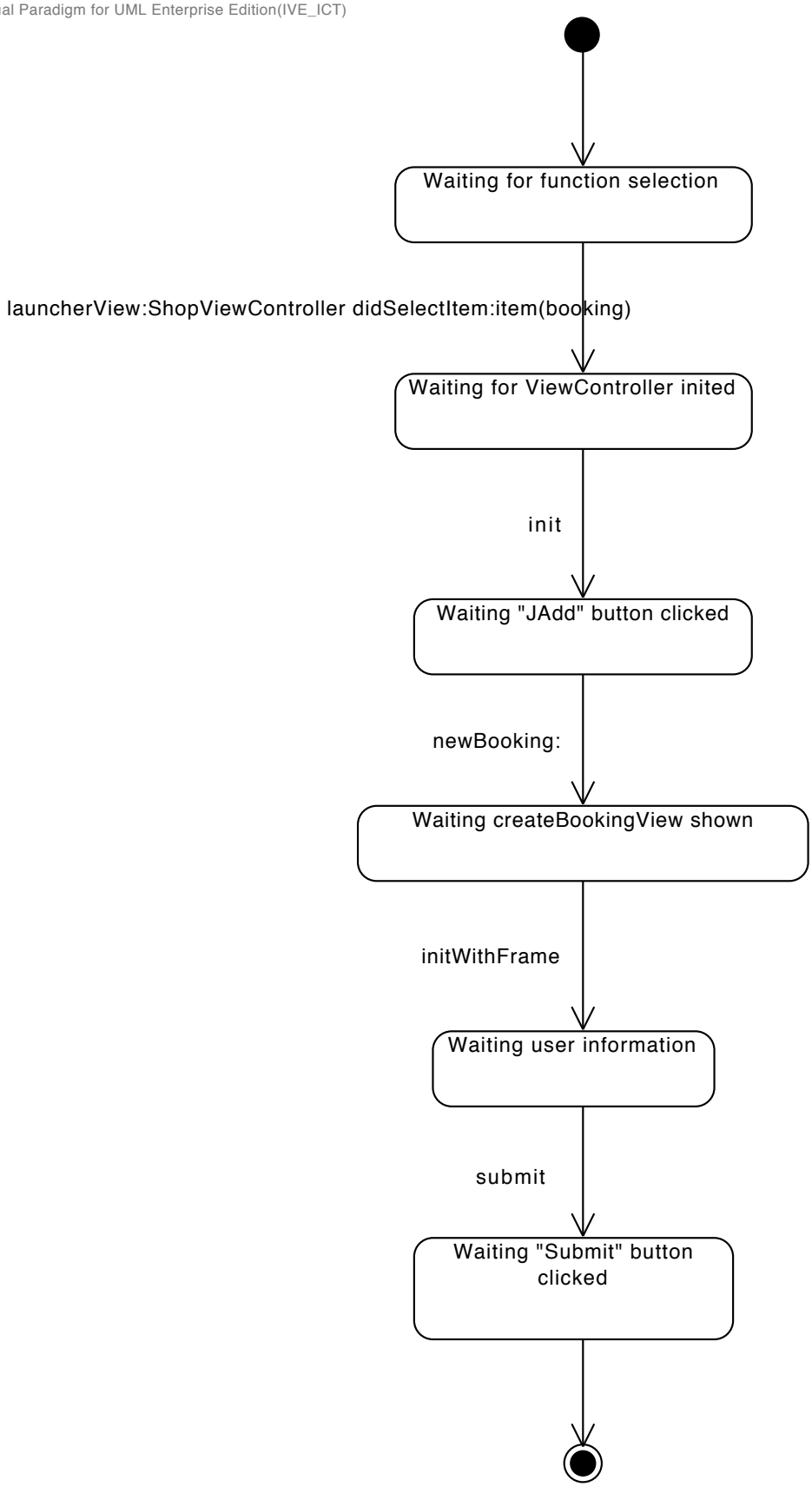

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting for function selection
launcherView:ShopViewController didSelectItem:item(eCoupon)
Waiting for ViewController inited
init
Waiting for result from server
jsonReaderWithURL: WithParam:
Waiting TableView with Data shown
initWithFrame
Waiting a coupon being selected
getNew:
Waiting for result from server
jsonReaderWithURL: WithParam:
Waiting NewCouponTableView with Data shown
initWithFrame
Waiting a coupon being selected
didSelectObject:object atIndexPath:indexPath
Waiting the eCouponView to be shown
initWithFrame
Waiting "Get" button clicked
get:
Waiting "Back" button clicked
back:

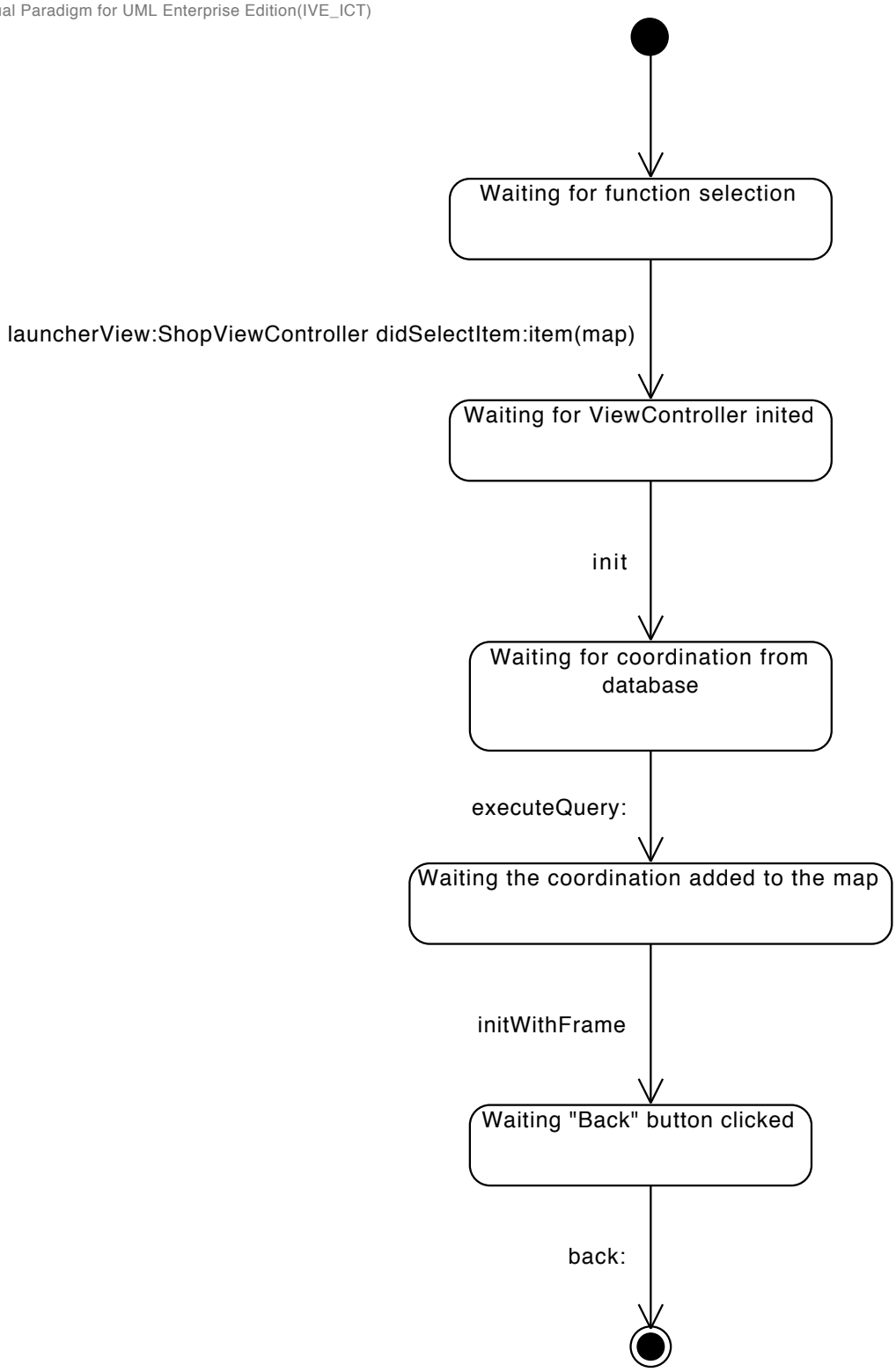

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting for function selection
launcherView:ShopViewController didSelectItem:item(Luckydraw)
Waiting for ViewController inited
init
Waiting for result from server
jsonReaderWithURL: WithParam:
Waiting TableView with Data shown
initWithFrame
Waiting a luckydraw being selected
didSelectObject:object atIndexPath:indexPath
Waiting Prize of the luckydraw and the quantity you have shown
list of prize and quantity shown

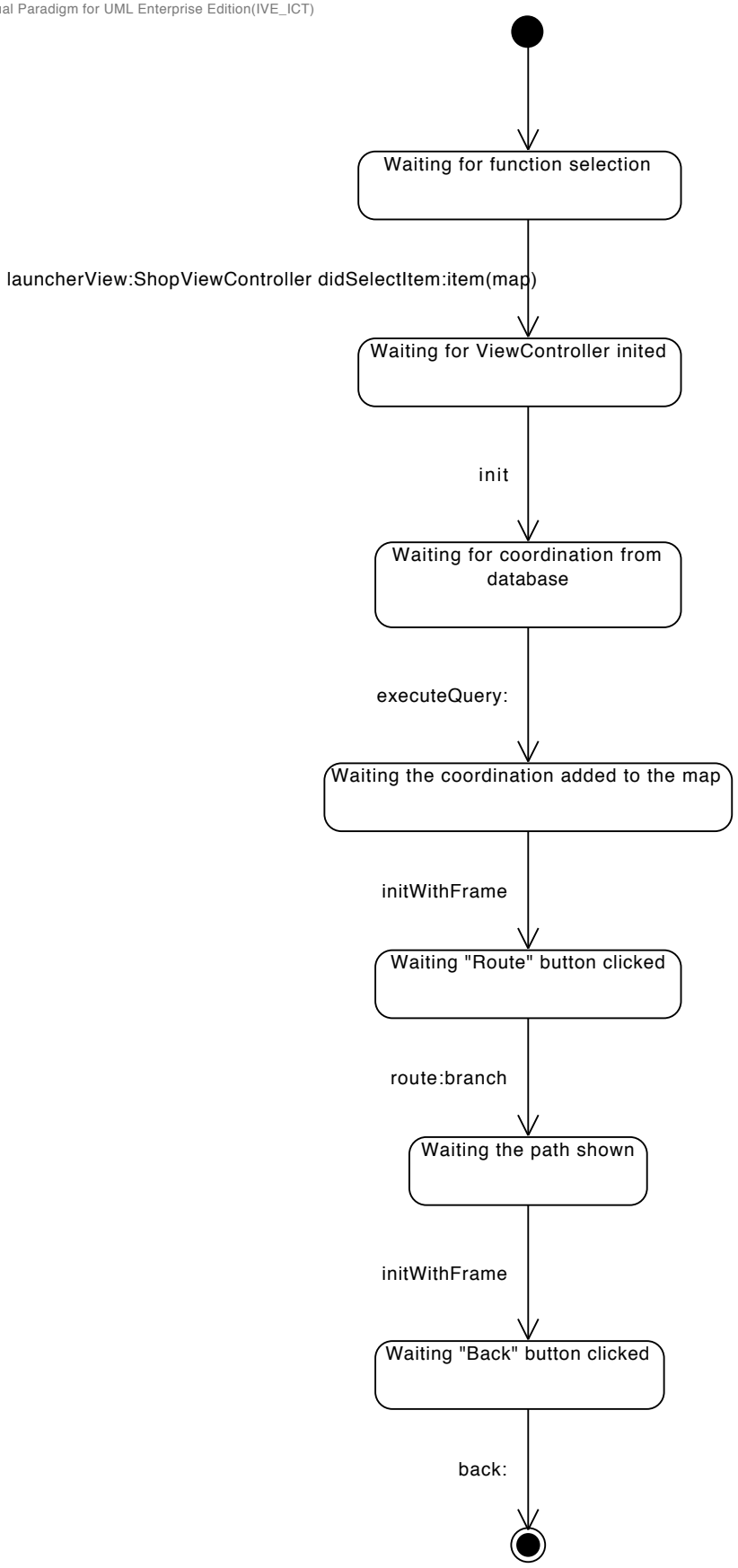

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting "Draw" button clicked
draw:
Waiting the random item drawn to shown
init
Waiting draw view to be shown

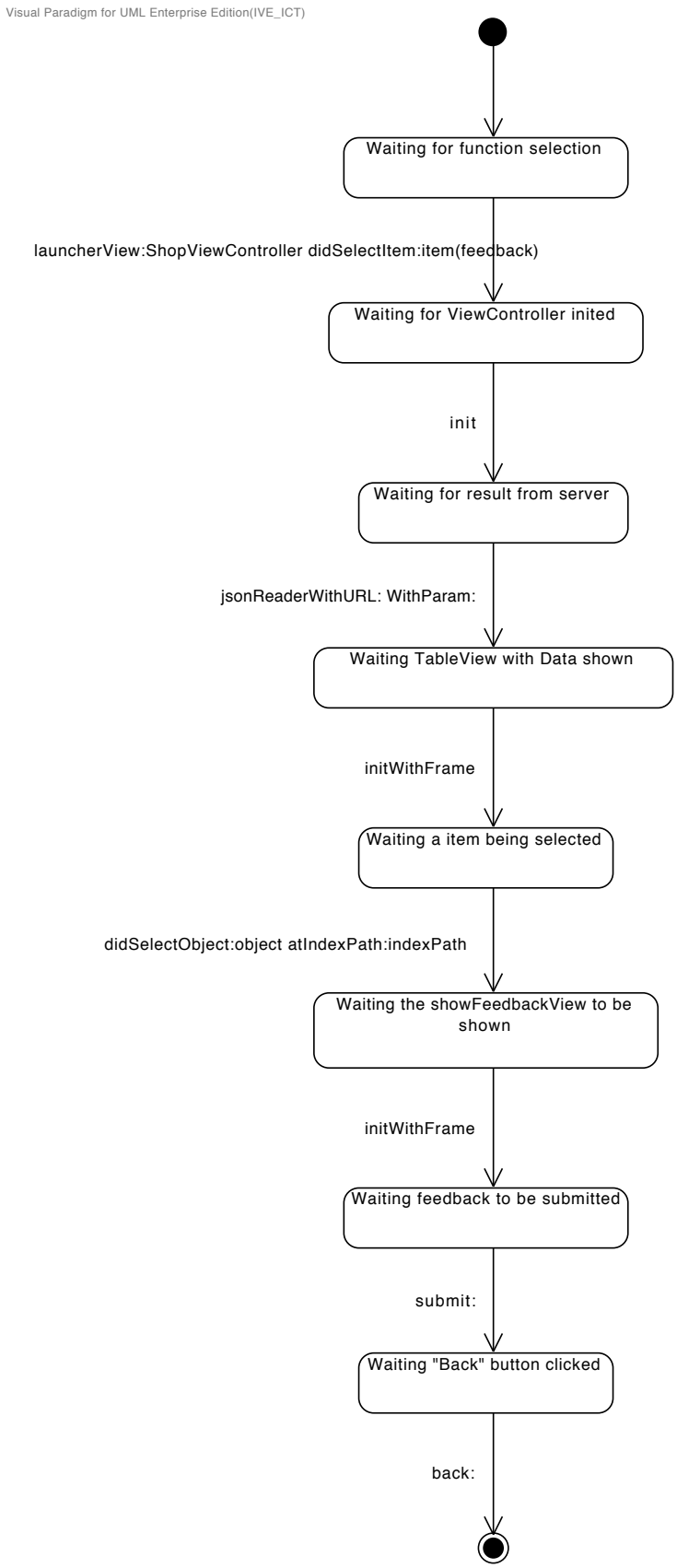
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

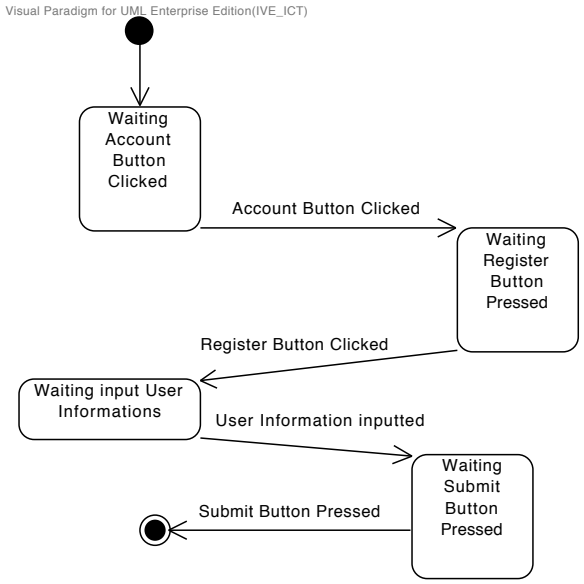


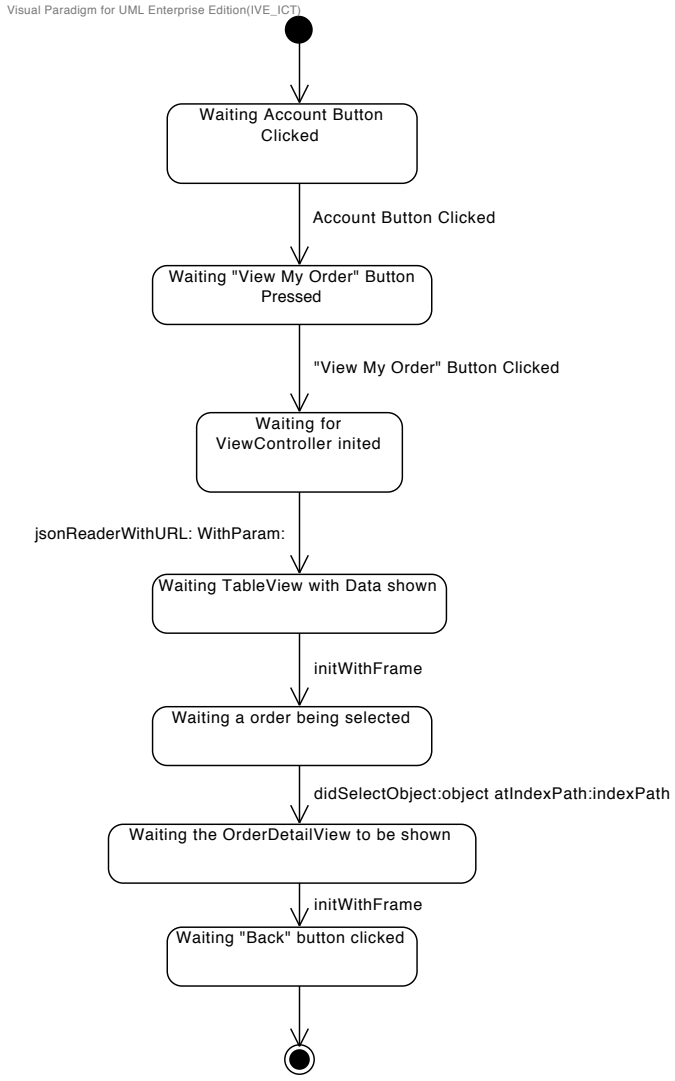

Waiting for function selection
launcherView:ShopViewController didSelectItem:item(booking)
Waiting for ViewController inited
init
Waiting "JAdd" button clicked
newBooking:
Waiting createBookingView shown
initWithFrame
Waiting user information
submit
Waiting "Submit" button clicked

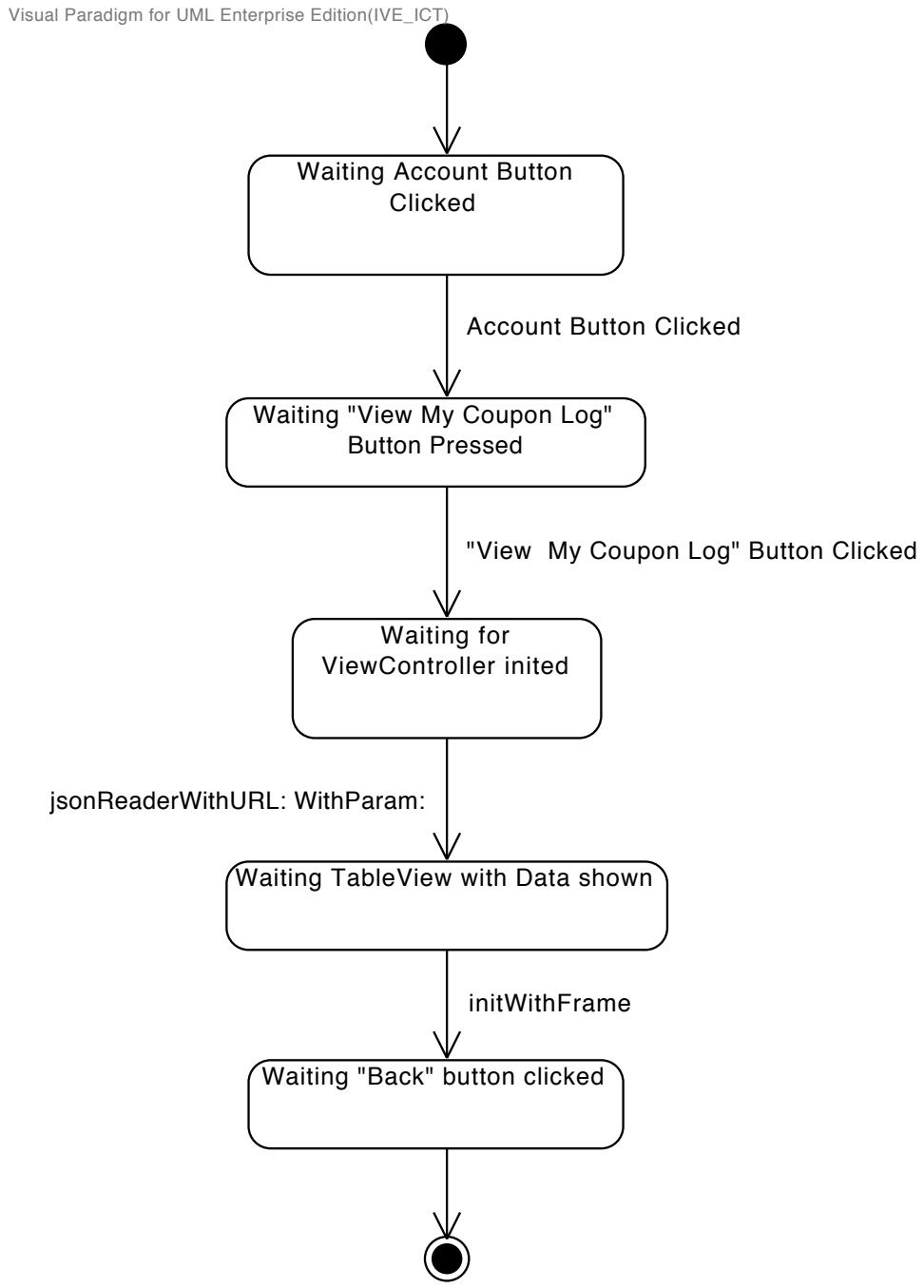

Waiting for function selection
launcherView:ShopViewController didSelectItem:item(map)
Waiting for ViewController inited
init
Waiting for coordination from database
executeQuery:
Waiting the coordination added to the map
initWithFrame
Waiting "Back" button clicked
back:


Waiting for function selection
launcherView:ShopViewController didSelectItem:item(map)
Waiting for ViewController inited
init
Waiting for coordination from database
executeQuery:
Waiting the coordination added to the map
initWithFrame
Waiting "Route" button clicked
route:branch
Waiting the path shown
initWithFrame
Waiting "Back" button clicked
back:

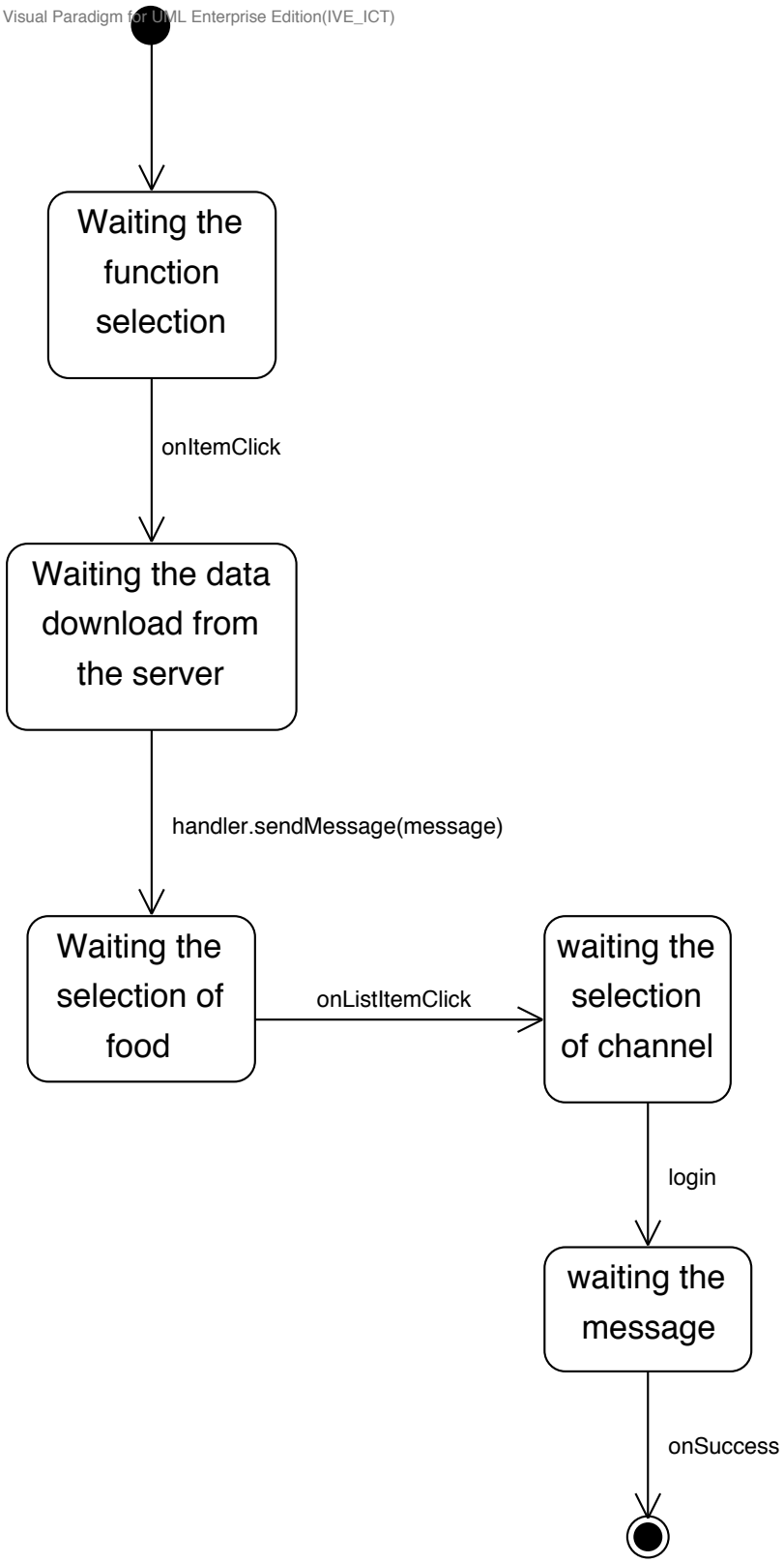

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting for function selection
launcherView:ShopViewController didSelectItem:item(feedback)
Waiting for ViewController inited
init
Waiting for result from server
jsonReaderWithURL: WithParam:
Waiting TableView with Data shown
initWithFrame
Waiting a item being selected
didSelectObject:object atIndexPath:indexPath
Waiting the showFeedbackView to be shown
initWithFrame
Waiting feedback to be submitted
submit:
Waiting "Back" button clicked
back:

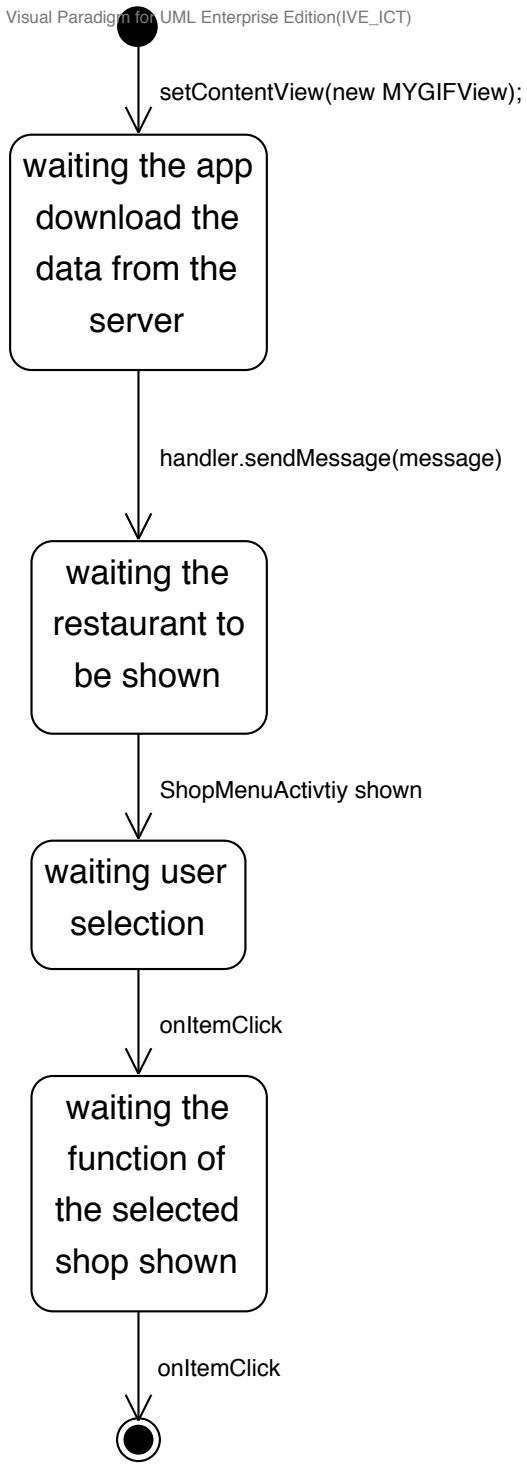

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting Account Button Clicked
Account Button Clicked
Waiting Register Button Pressed
Register Button Clicked
Waiting input User Informations
User Information inputted
Waiting Submit Button Pressed
Submit Button Pressed

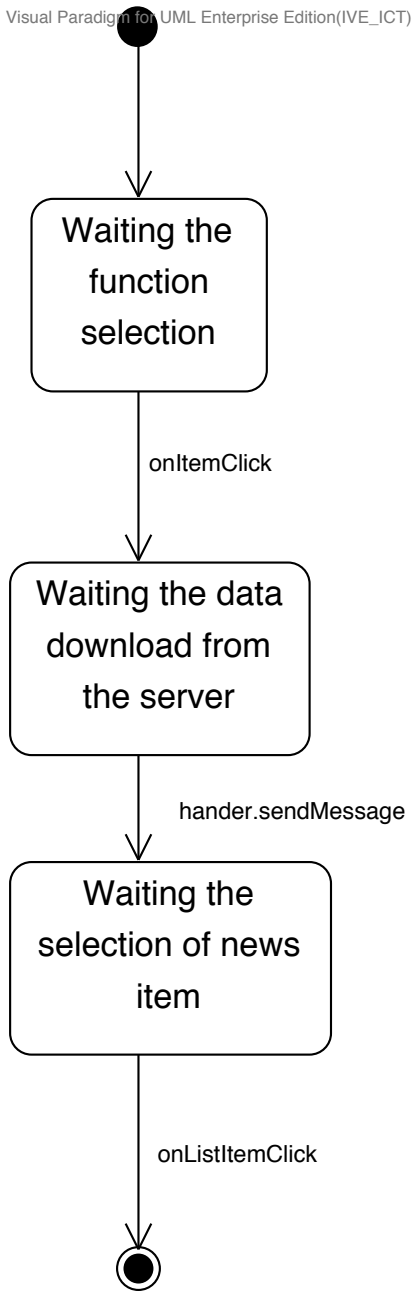

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting Account Button Clicked
Account Button Clicked
Waiting "View My Order" Button Pressed
"View My Order" Button Clicked
Waiting for ViewController inited
jsonReaderWithURL: WithParam:
Waiting TableView with Data shown
initWithFrame
Waiting a order being selected
didSelectObject:object atIndexPath:indexPath
Waiting the OrderDetailView to be shown
initWithFrame
Waiting "Back" button clicked

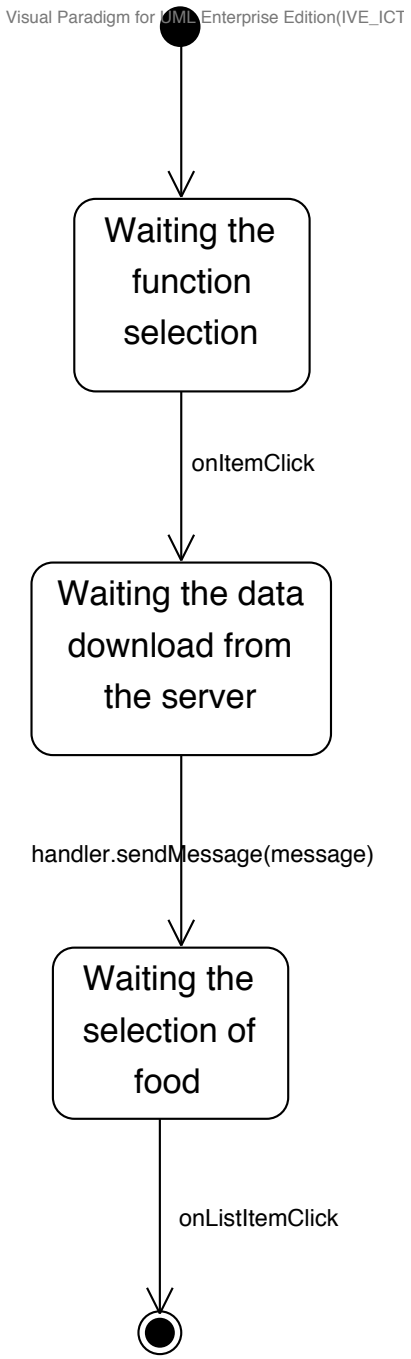

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting Account Button Clicked
Account Button Clicked
Waiting "View My Coupon Log" Button Pressed
"View My Coupon Log" Button Clicked
Waiting for ViewController inited
jsonReaderWithURL: WithParam:
Waiting TableView with Data shown
initWithFrame
Waiting "Back" button clicked

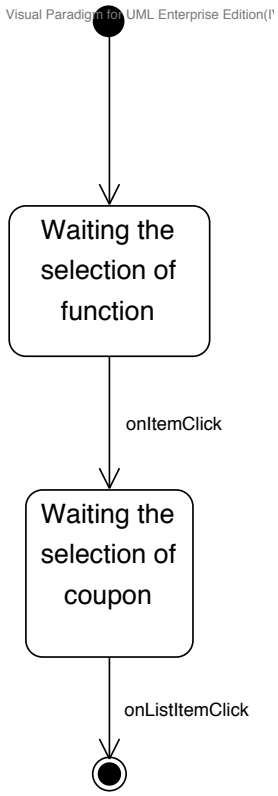
State Transition Diagram (Android)

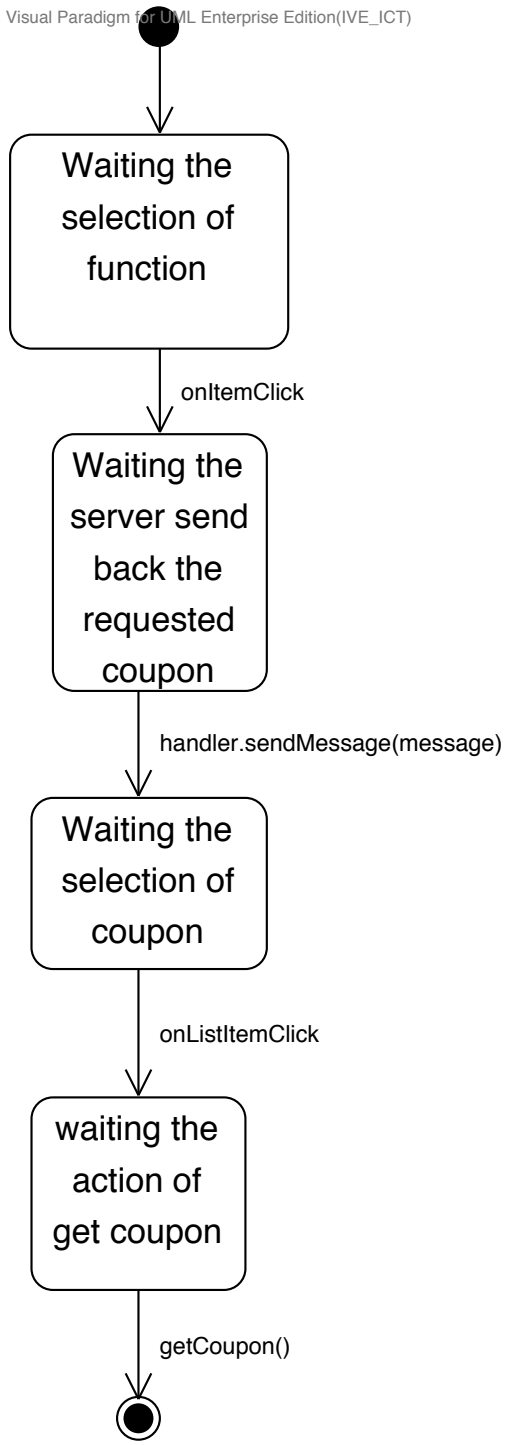


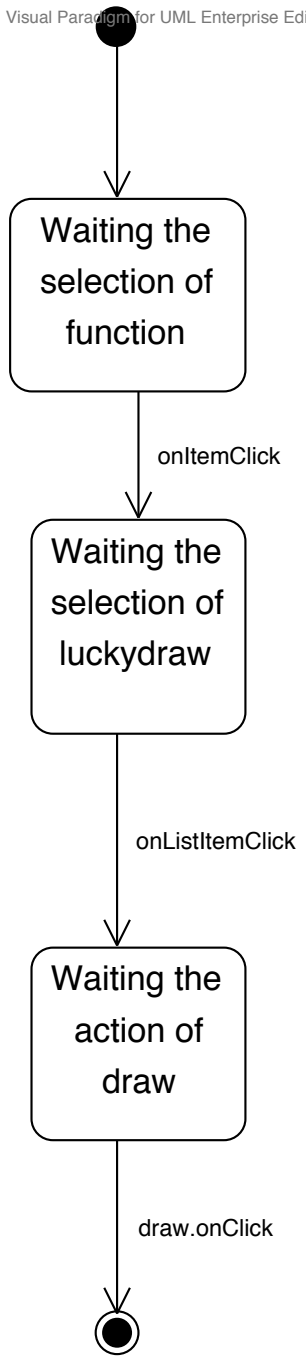

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
setContentView(new MYGIFView);
waiting the app download the data from the server
handler.sendMessage(message)
waiting the restaurant to be shown
ShopMenuActivtiy shown
waiting user selection
onItemClick
waiting the function of the selected shop shown
onItemClick

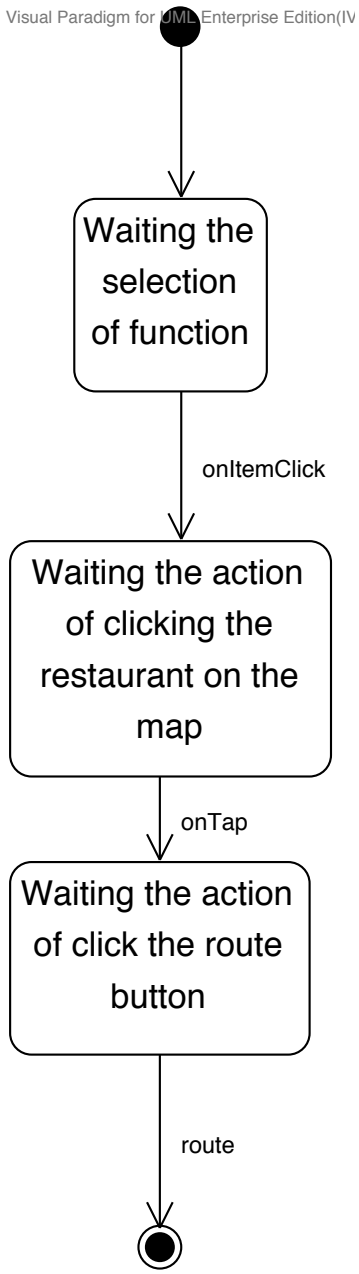

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting the function selection
onItemClick
Waiting the data download from the server
hander.sendMessage
Waiting the selection of news item
onListItemClick

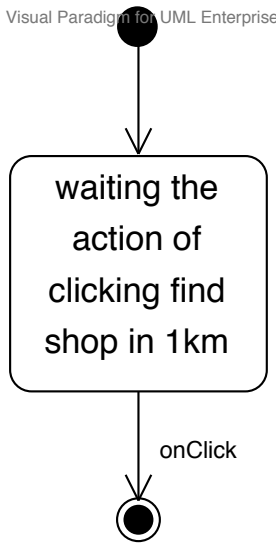

Visual Paradigm for UML Enterprise Edition(IVE_ICT
Waiting the function selection
onItemClick
Waiting the data download from the server
handler.sendMessage(message)
Waiting the selection of food
onListItemClick

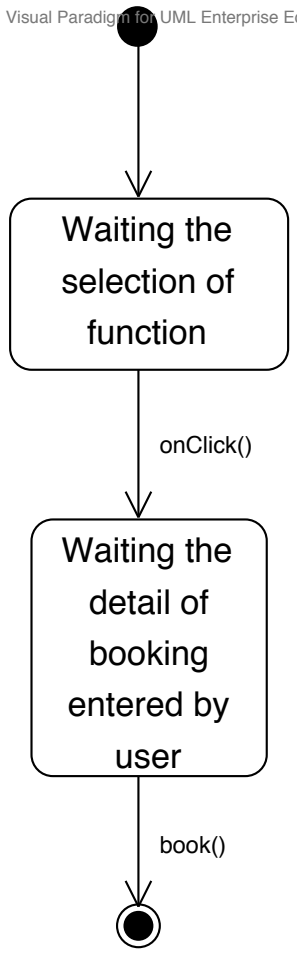

Visual Paradigm for UML Enterprise Edition(I
Waiting the selection of function
onItemClick
Waiting the selection of coupon
onListItemClick

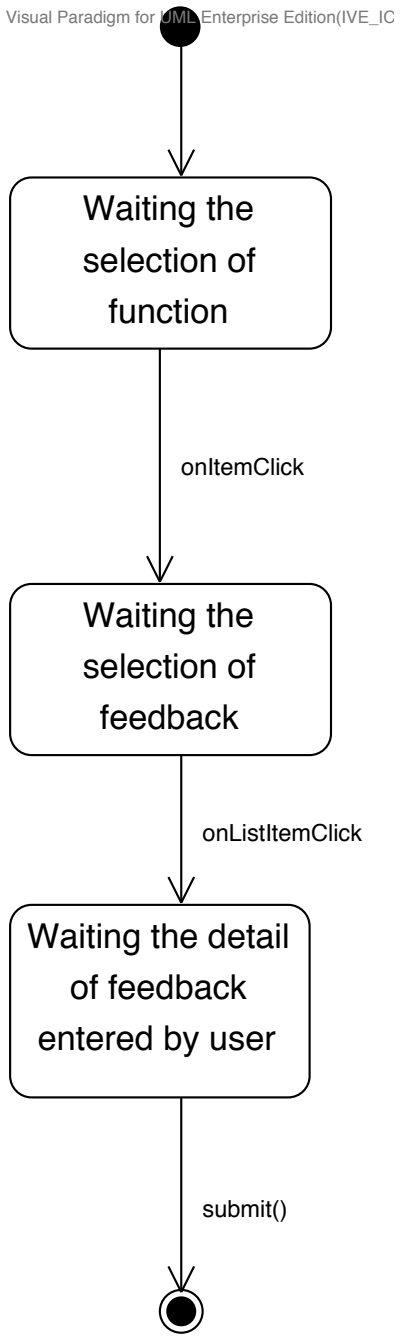

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Waiting the selection of function
onItemClick
Waiting the server send back the requested coupon
handler.sendMessage(message)
Waiting the selection of coupon
onListItemClick
waiting the action of get coupon
getCoupon()

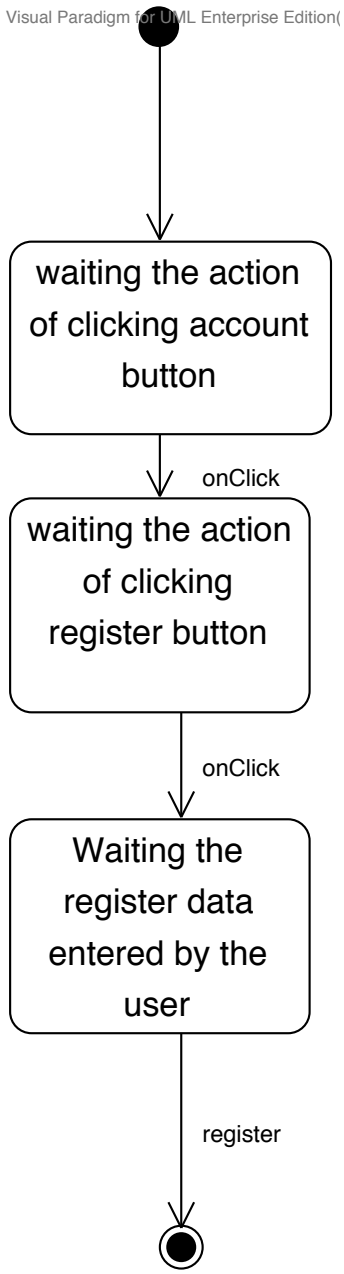

Visual Paradigm for UML Enterprise Editio
Waiting the selection of function
onItemClick
Waiting the selection of luckydraw
onListItemClick
Waiting the action of draw
draw.onClick


Visual Paradigm for UML Enterprise Edition(IV
Waiting the selection of function
onItemClick
Waiting the action of clicking the restaurant on the map
onTap
Waiting the action of click the route button
route

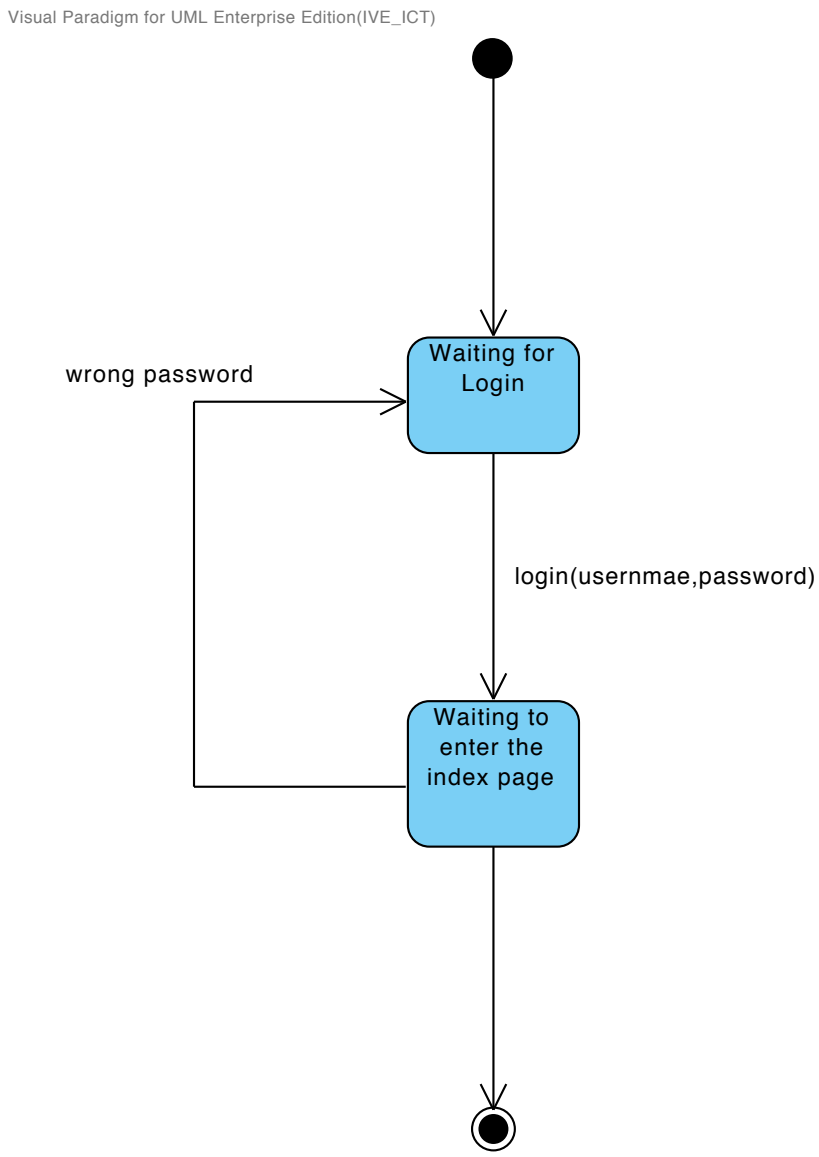

Visual Paradigm for UML Enterprise
waiting the action of clicking find shop in 1km
onClick

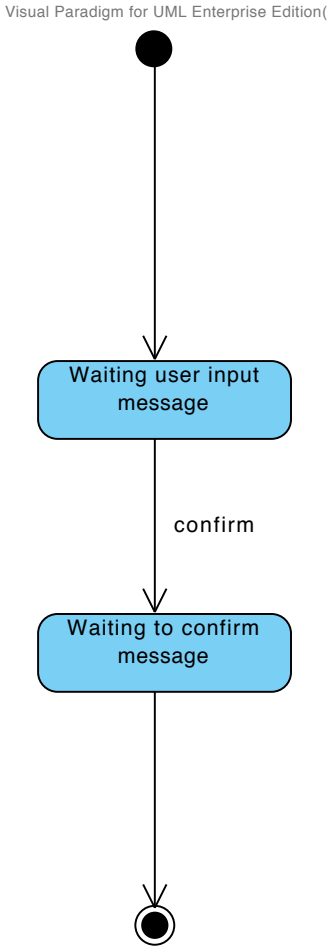

Visual Paradigm for UML Enterprise E
Waiting the selection of function
onClick()
Waiting the detail of booking entered by user
book()

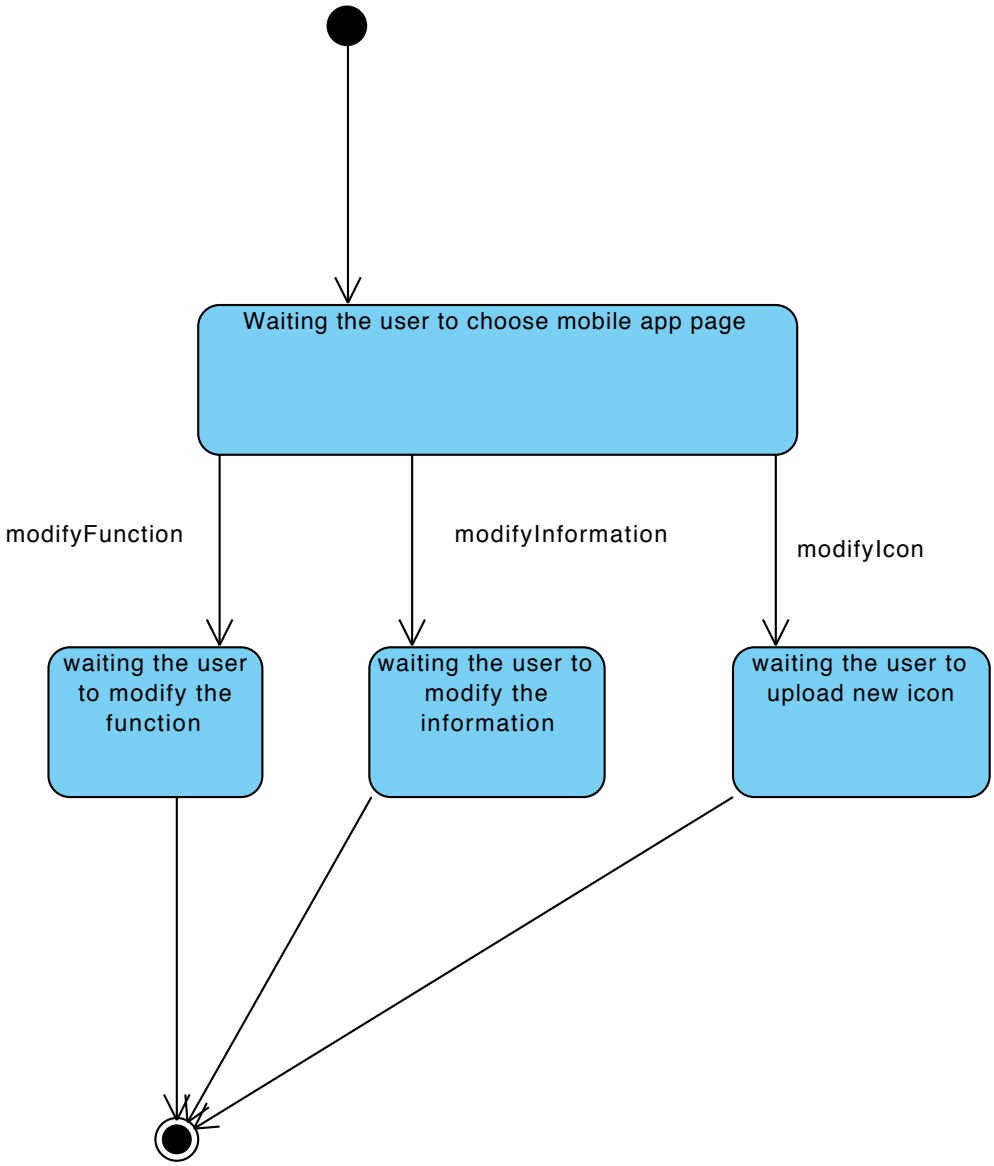

Visual Paradigm for UML Enterprise Edition(IVE_IC
Waiting the selection of function
onItemClick
Waiting the selection of feedback
onListItemClick
Waiting the detail of feedback entered by user
submit()

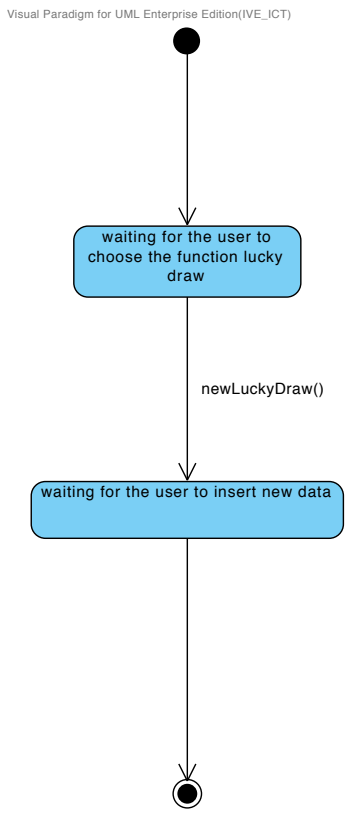

Visual Paradigm for UML Enterprise Edition(
waiting the action of clicking account button
onClick
waiting the action of clicking register button
onClick
Waiting the register data entered by the user
register

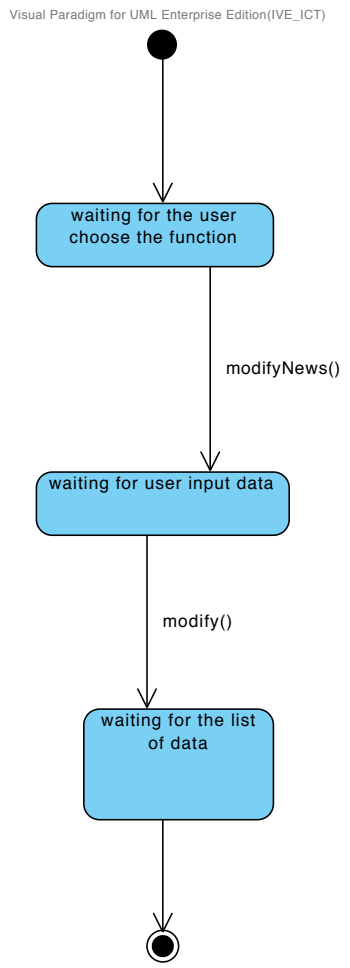
State Transition Diagram (Website)

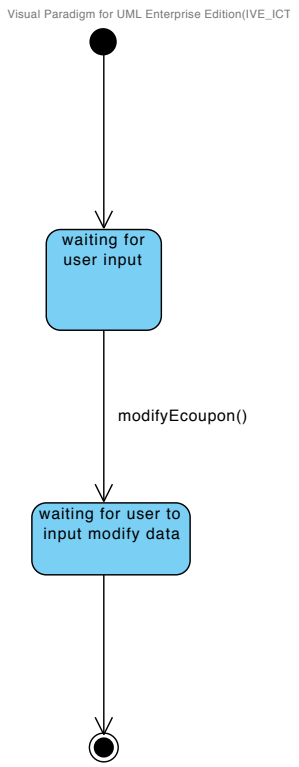


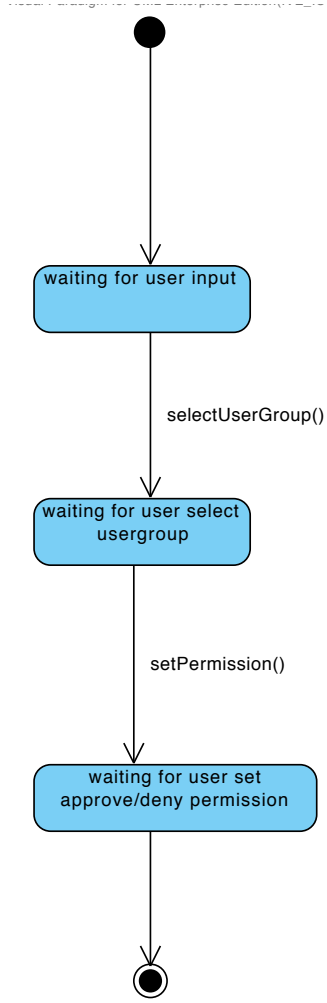

Visual Paradigm for UML Enterprise Edition(
Waiting user input message
confirm
Waiting to confirm message

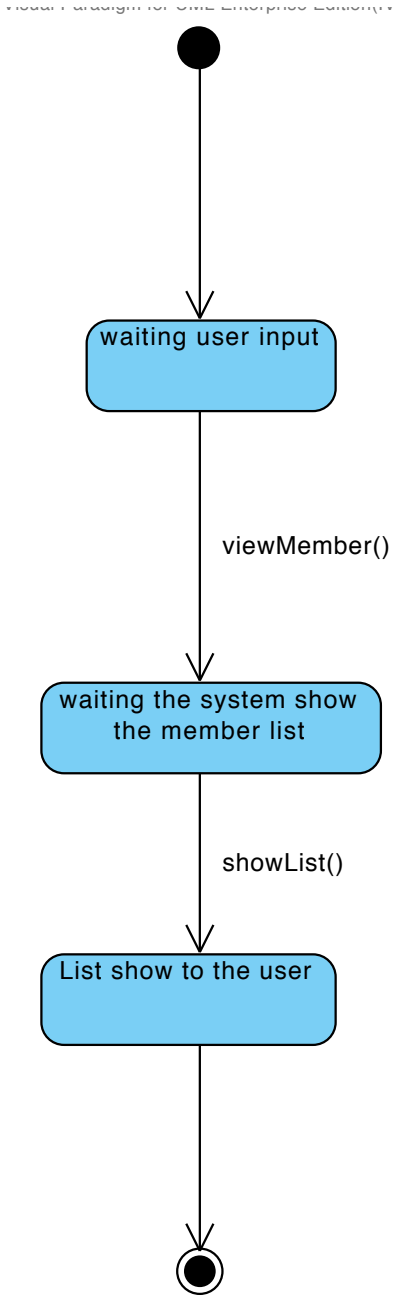

Waiting the user to choose mobile app page
modifyFunction
modifyInformation
modifyIcon
waiting the user to modify the function
waiting the user to modify the information
waiting the user to upload new icon

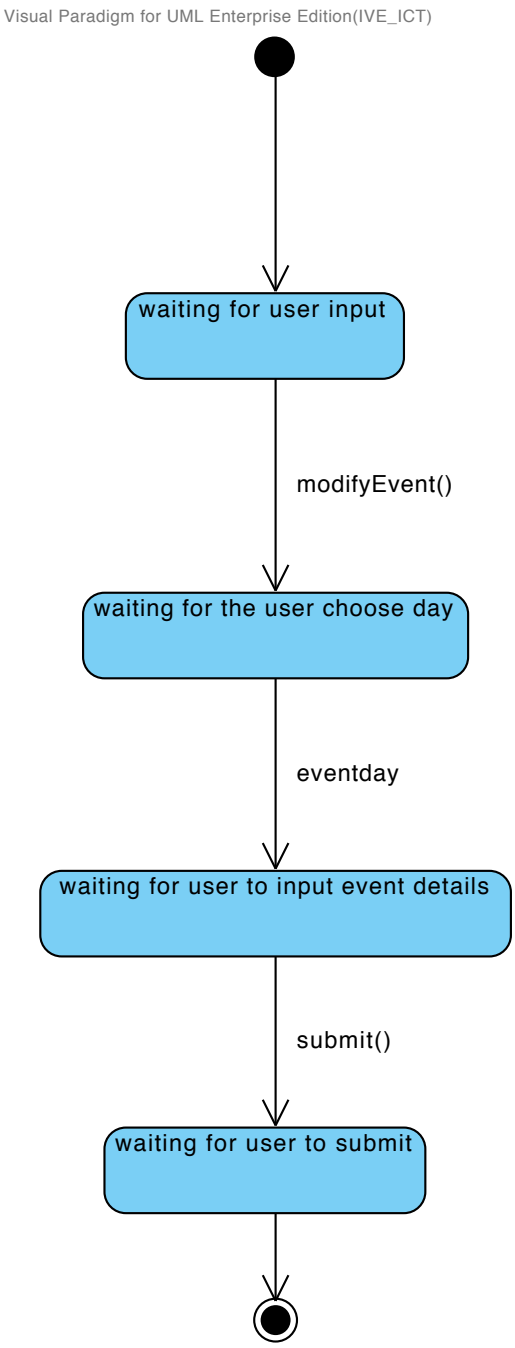

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
waiting for the user to choose the function lucky draw
newLuckyDraw()
waiting for the user to insert new data

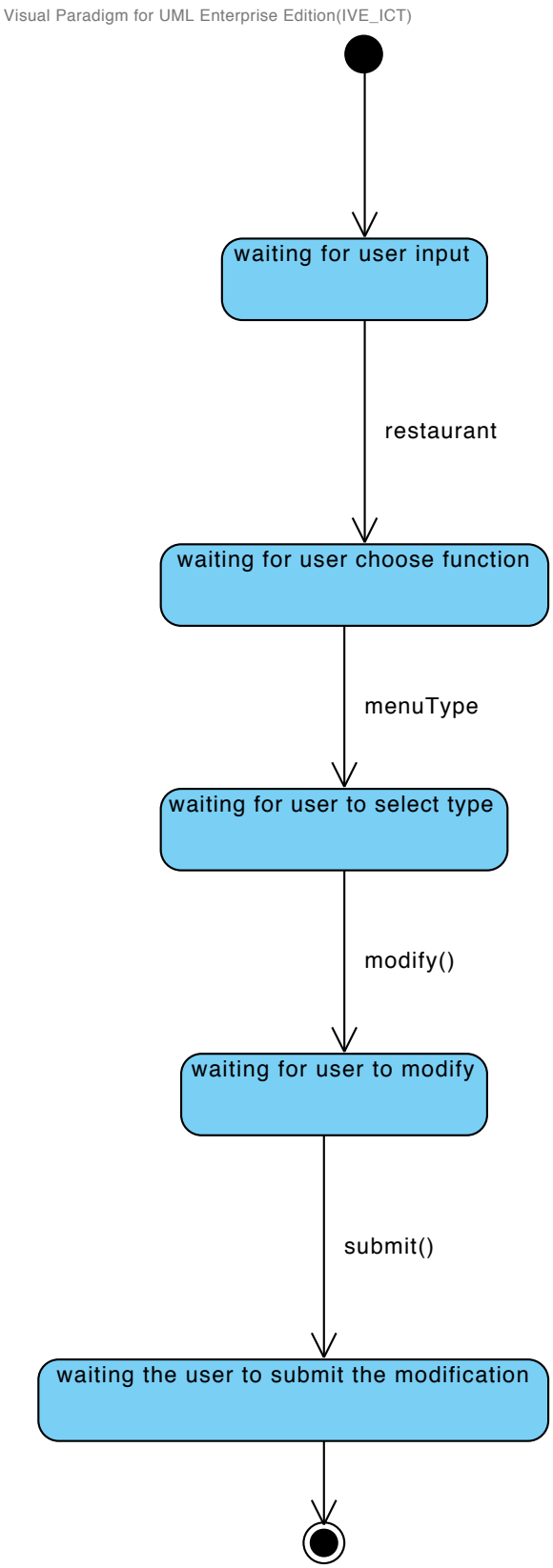

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
waiting for the user choose the function
modifyNews()
waiting for user input data
modify()
waiting for the list of data

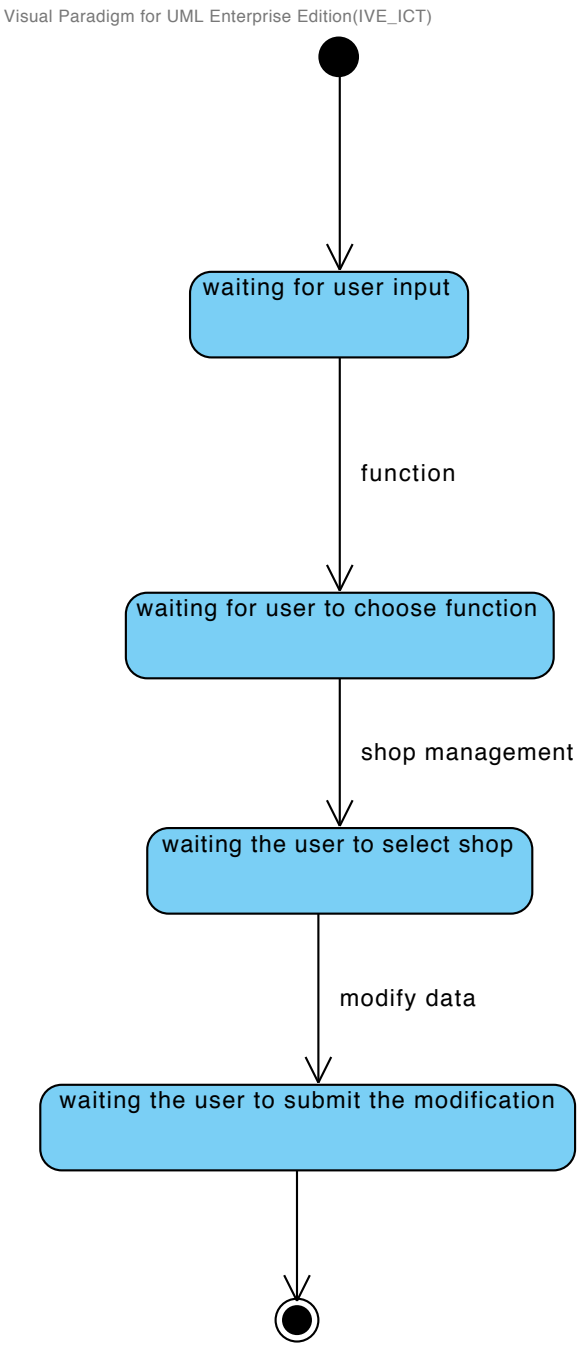

Visual Paradigm for UML Enterprise Edition(IVE_ICT
waiting for user input
modifyEcoupon()
waiting for user to input modify data

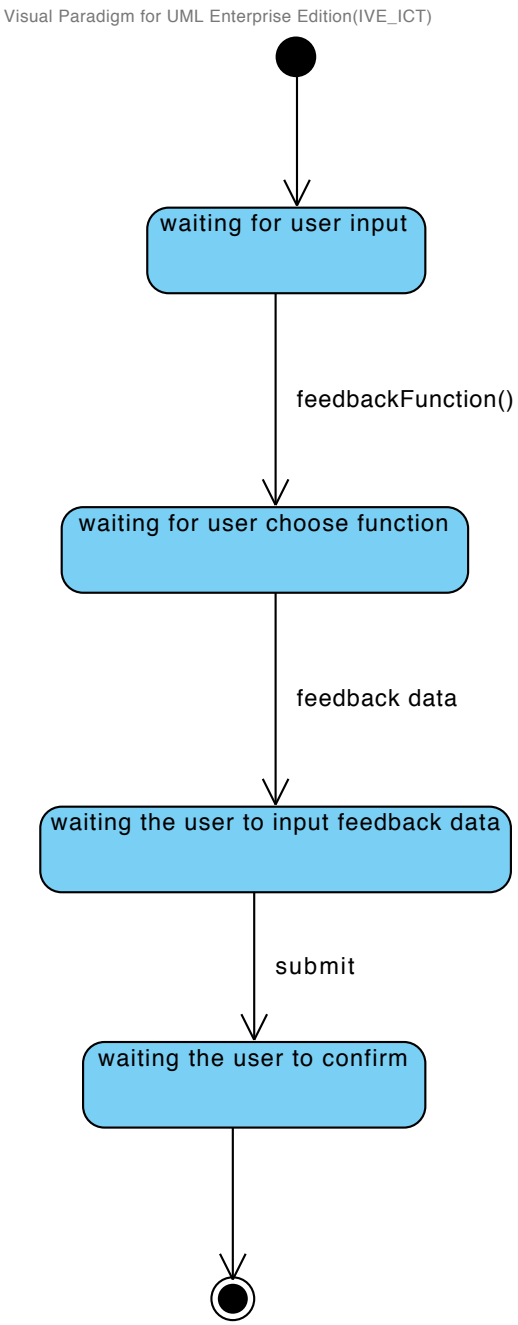

waiting for user input
selectUserGroup()
waiting for user select usergroup
setPermission()
waiting for user set approve/deny permission


waiting user input
viewMember()
waiting the system show the member list
showList()
List show to the user

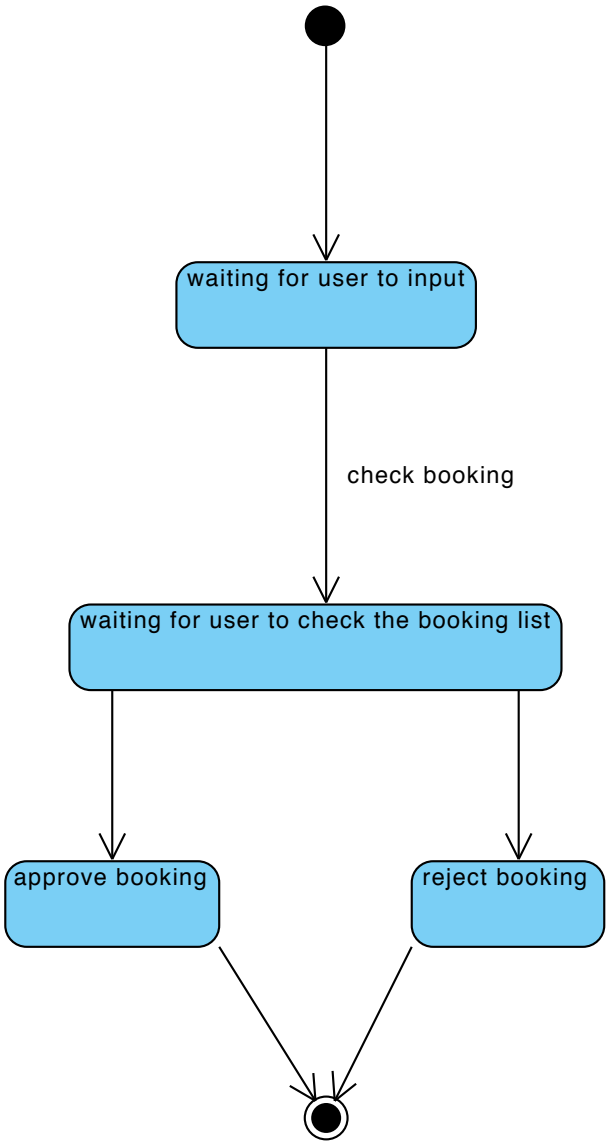

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
waiting for user input
modifyEvent()
waiting for the user choose day
eventday
waiting for user to input event details
submit()
waiting for user to submit

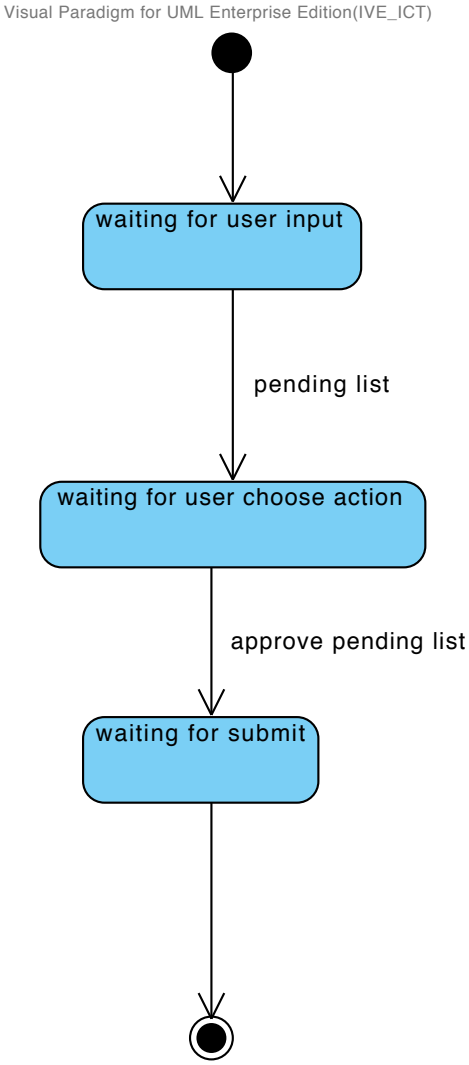

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
waiting for user input
restaurant
waiting for user choose function
menuType
waiting for user to select type
modify()
waiting for user to modify
submit()
waiting the user to submit the modification


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
waiting for user input
function
waiting for user to choose function
shop management
waiting the user to select shop
modify data
waiting the user to submit the modification

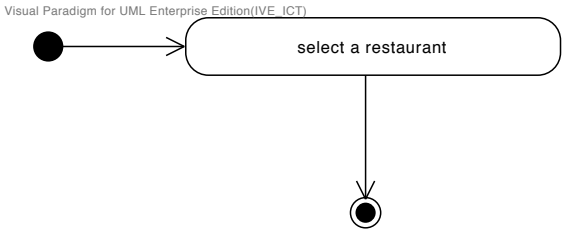

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
waiting for user input
feedbackFunction()
waiting for user choose function
feedback data
waiting the user to input feedback data
submit
waiting the user to confirm

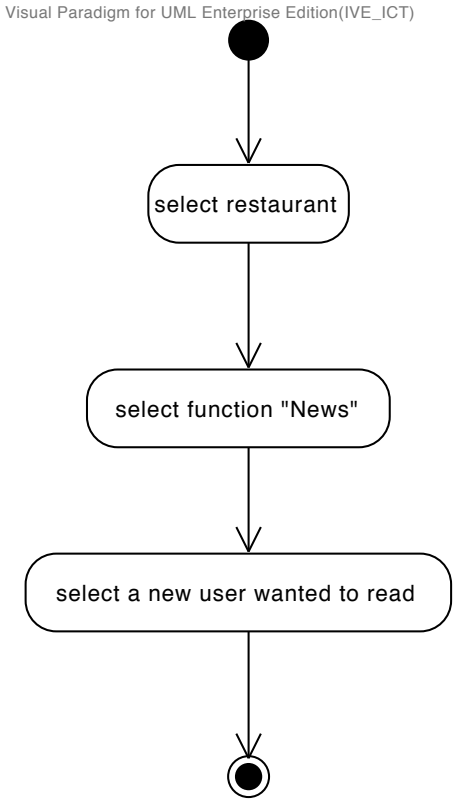
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

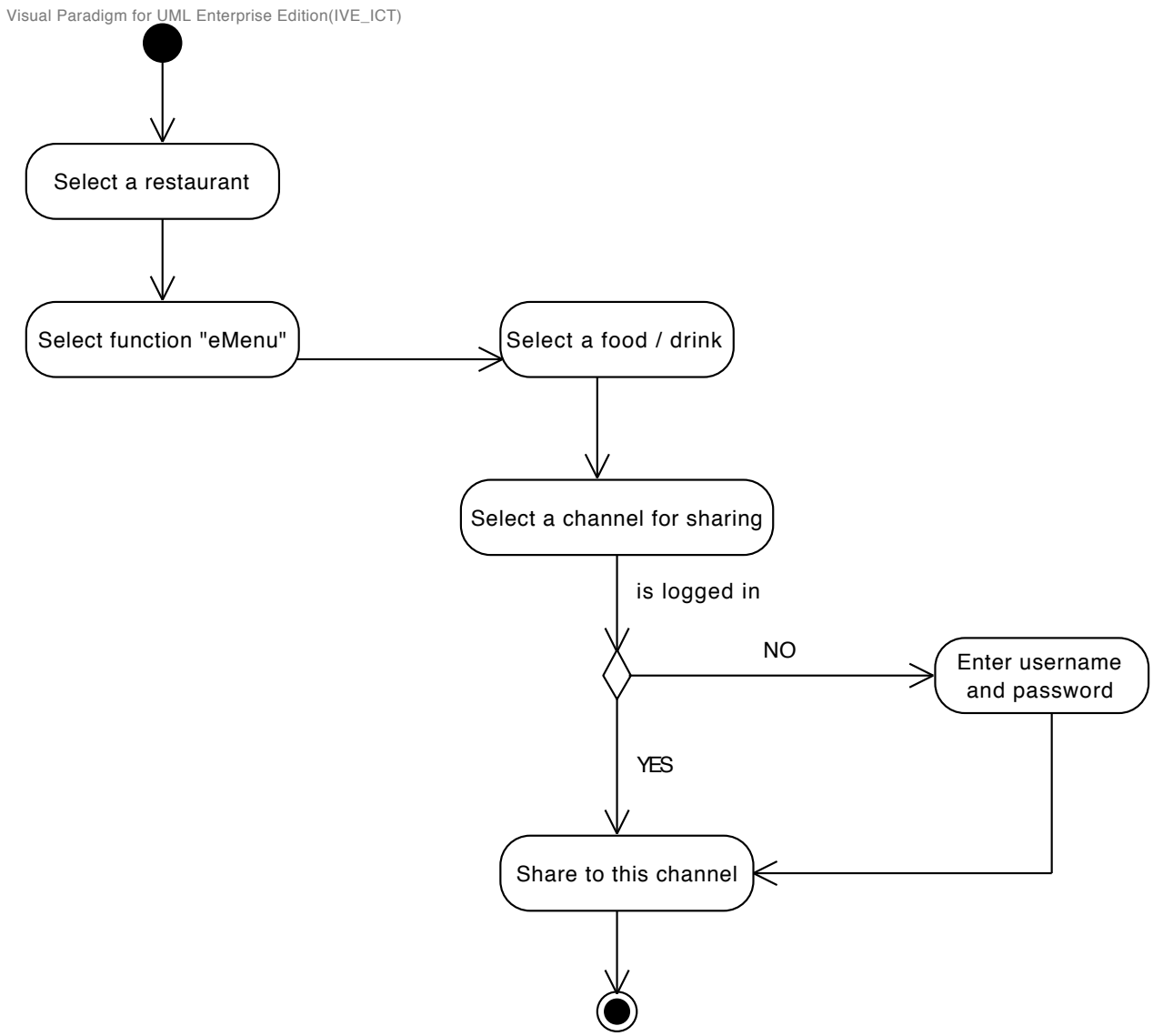


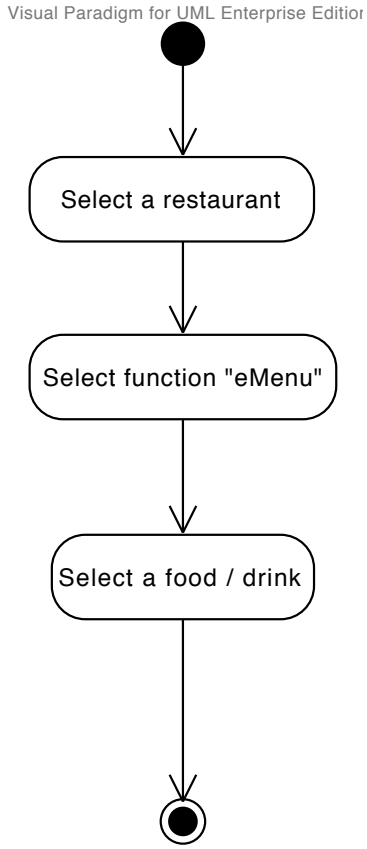

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
waiting for user input
pending list
waiting for user choose action
approve pending list
waiting for submit

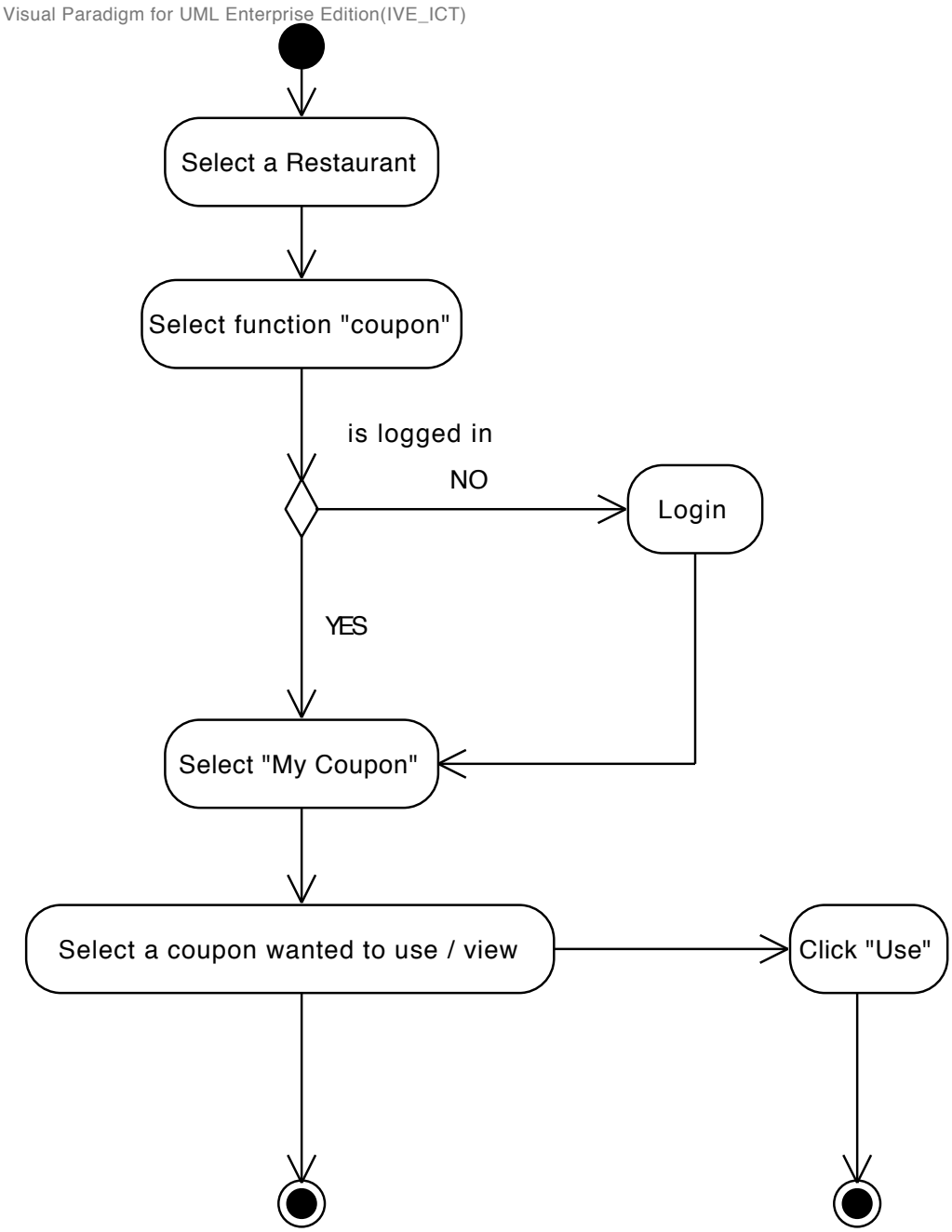
Activity Diagram (iPhone)

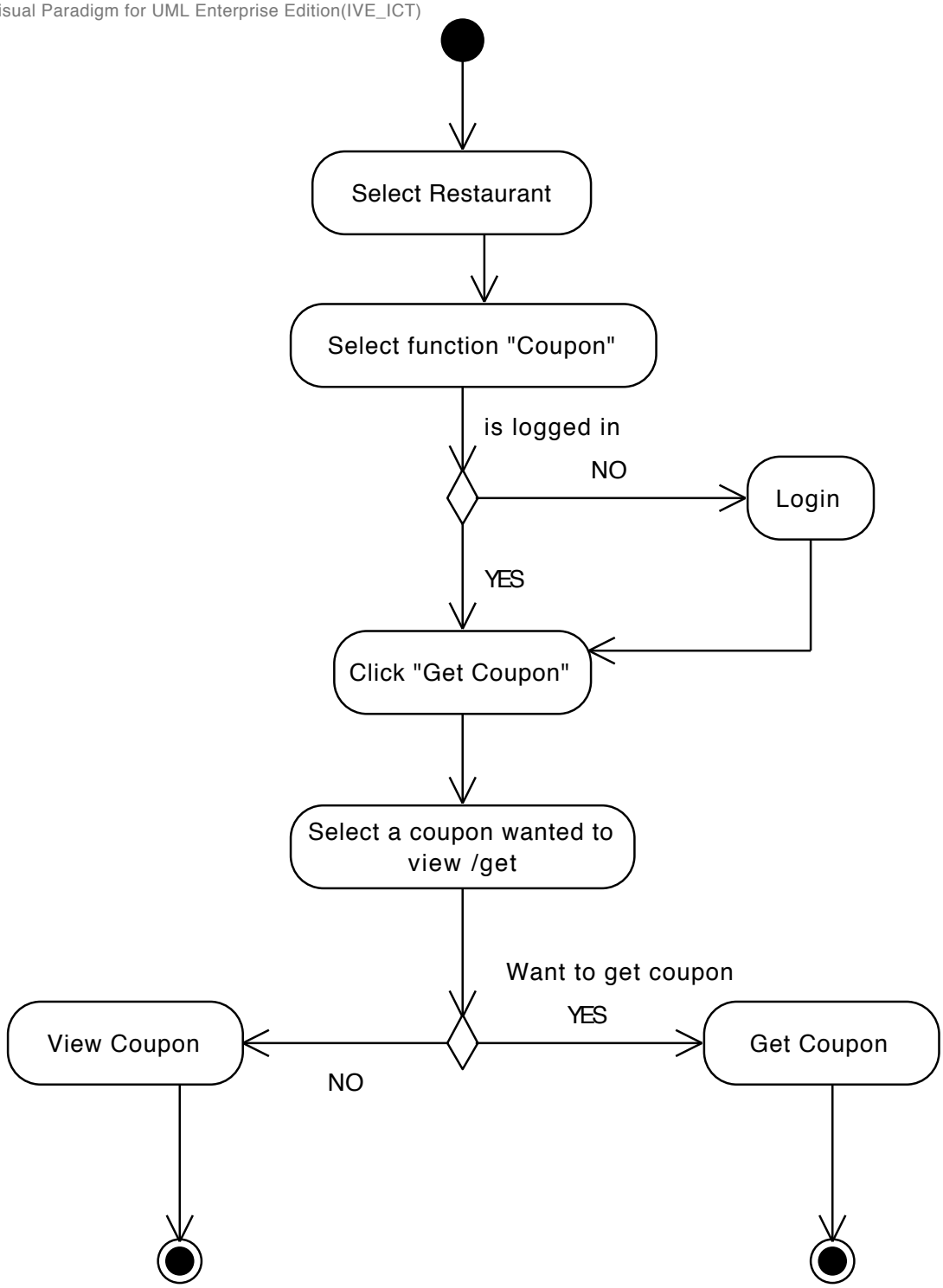


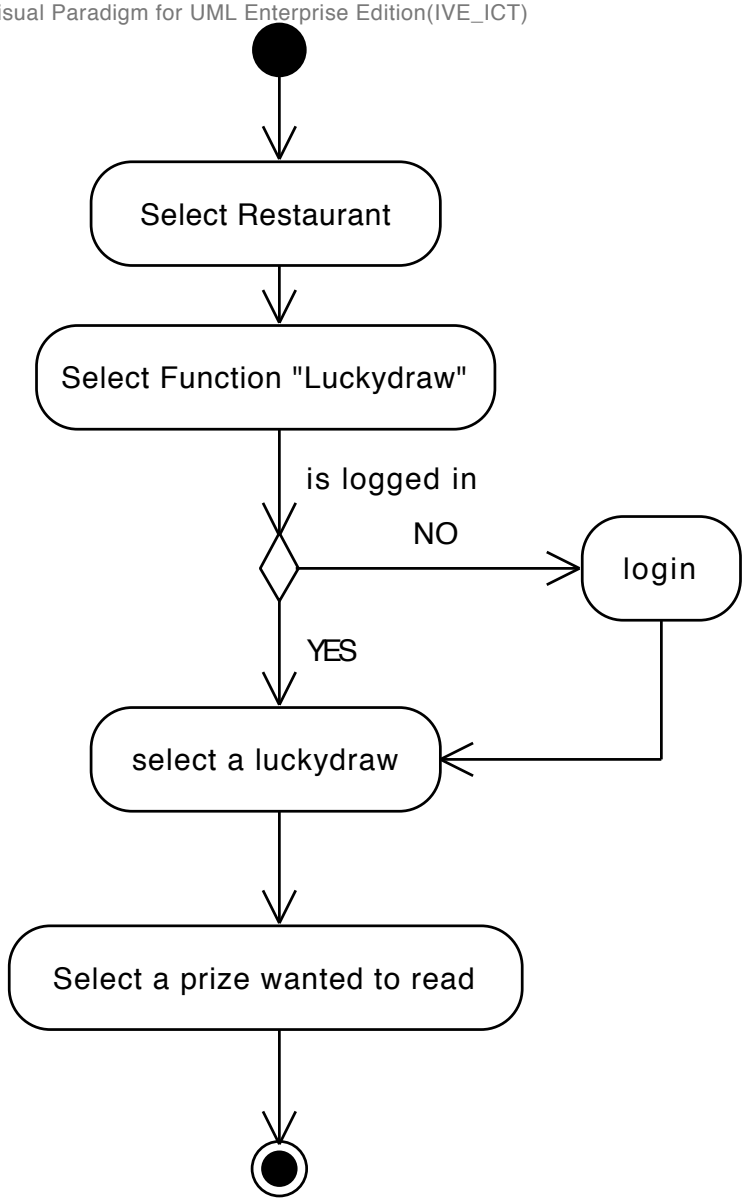


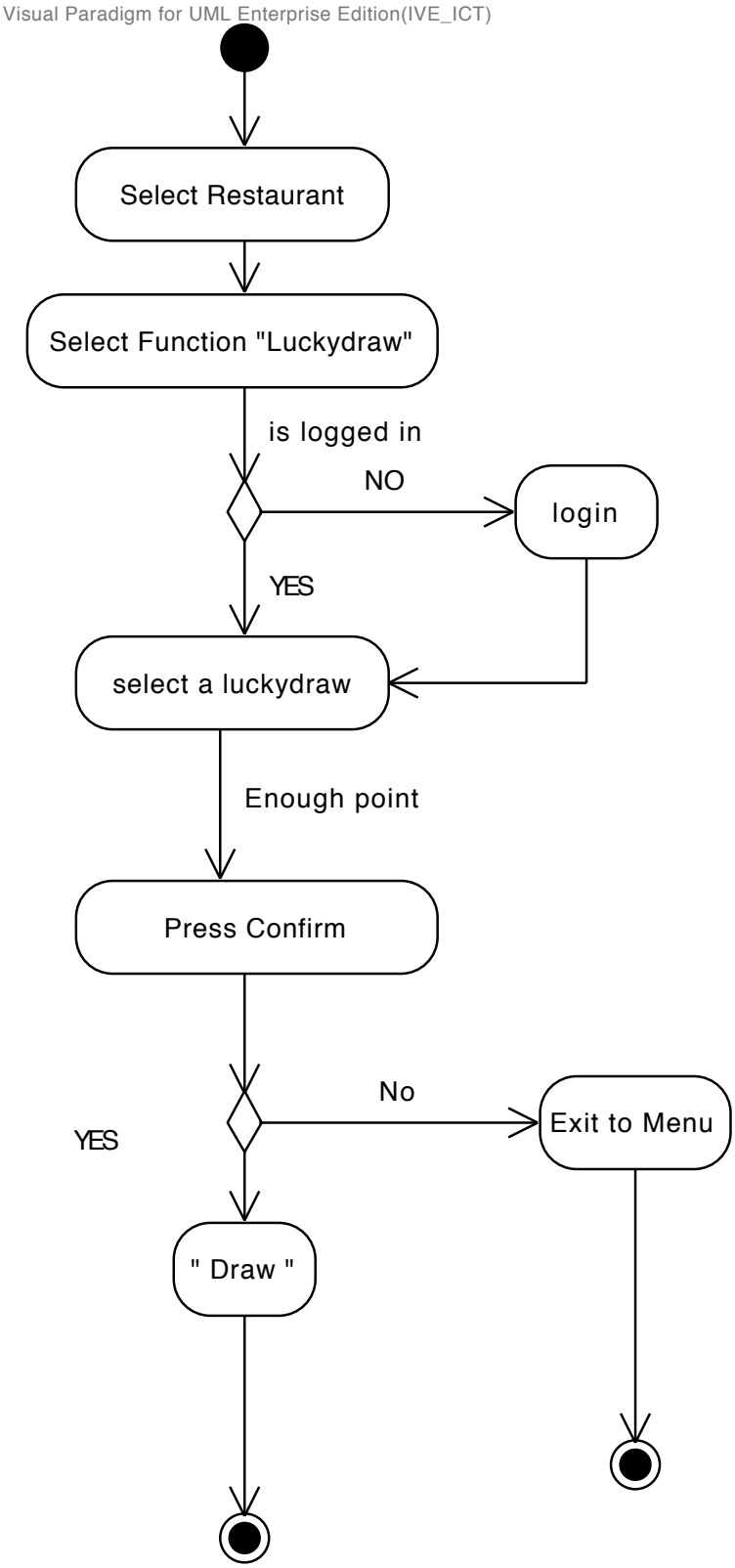

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Select a restaurant
Select function "eMenu"
Select a food / drink
Select a channel for sharing
is logged in
NO
Enter username and password
YES
Share to this channel

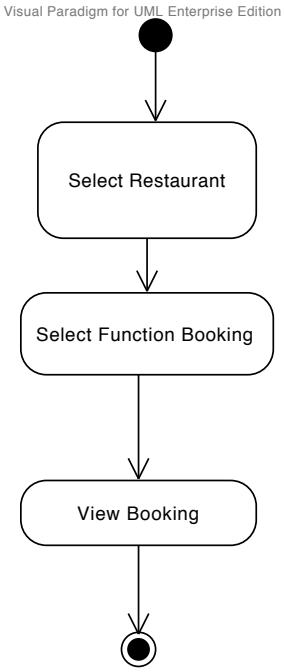

Visual Paradigm for UML Enterprise Editior
Select a restaurant
Select function "eMenu"
Select a food / drink

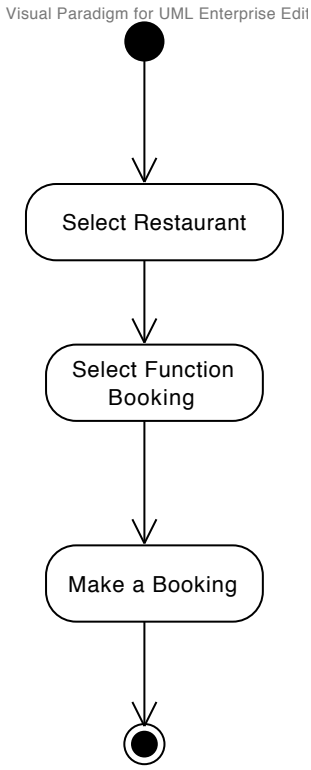

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Select a Restaurant
Select function "coupon"
is logged in
NO
Login
YES
Select "My Coupon"
Select a coupon wanted to use / view
Click "Use"

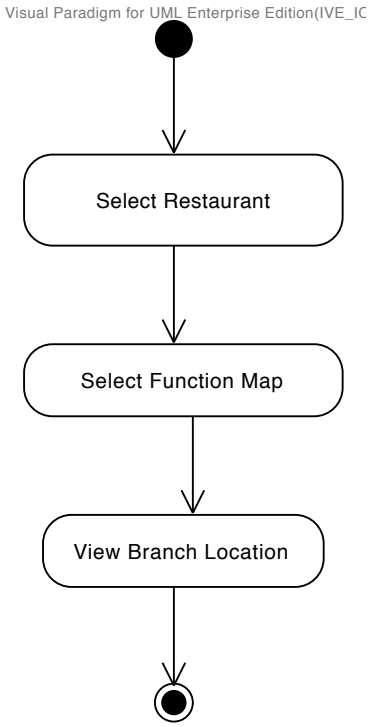

Select Restaurant
Select function "Coupon"
is logged in
NO
Login
YES
Click "Get Coupon"
Select a coupon wanted to view /get
Want to get coupon
YES
View Coupon
Get Coupon
NO

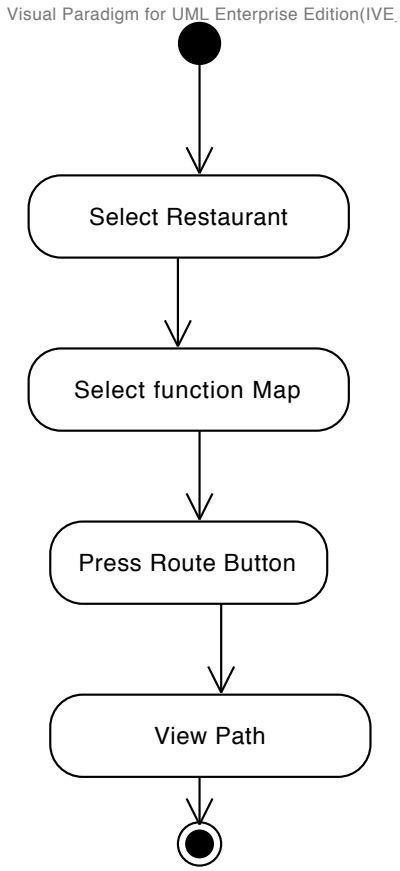

Select Restaurant
Select Function "Luckydraw"
is logged in
NO
login
YES
select a luckydraw
Select a prize wanted to read

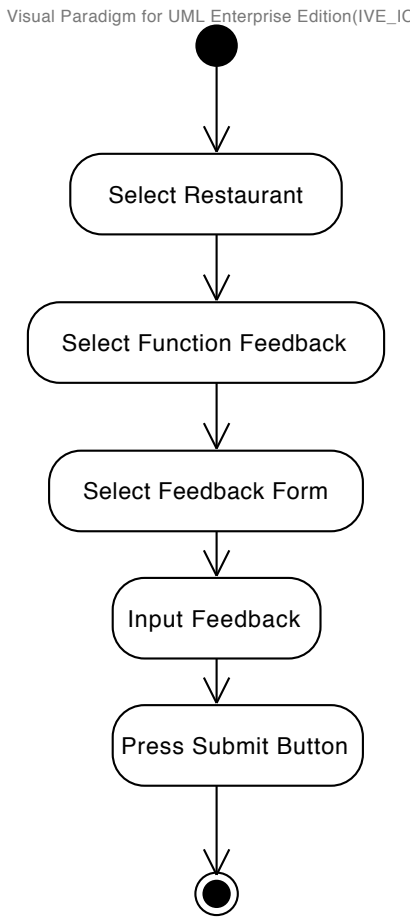

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Select Restaurant
Select Function "Luckydraw"
is logged in
NO
login
YES
select a luckydraw
Enough point
Press Confirm
No
Exit to Menu
YES
" Draw "

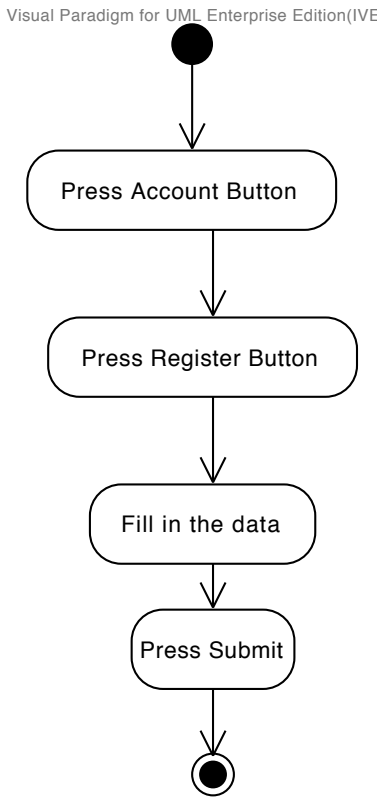

Visual Paradigm for UML Enterprise Edition
Select Restaurant
Select Function Booking
View Booking


Visual Paradigm for UML Enterprise Edit
Select Restaurant
Select Function Booking
Make a Booking

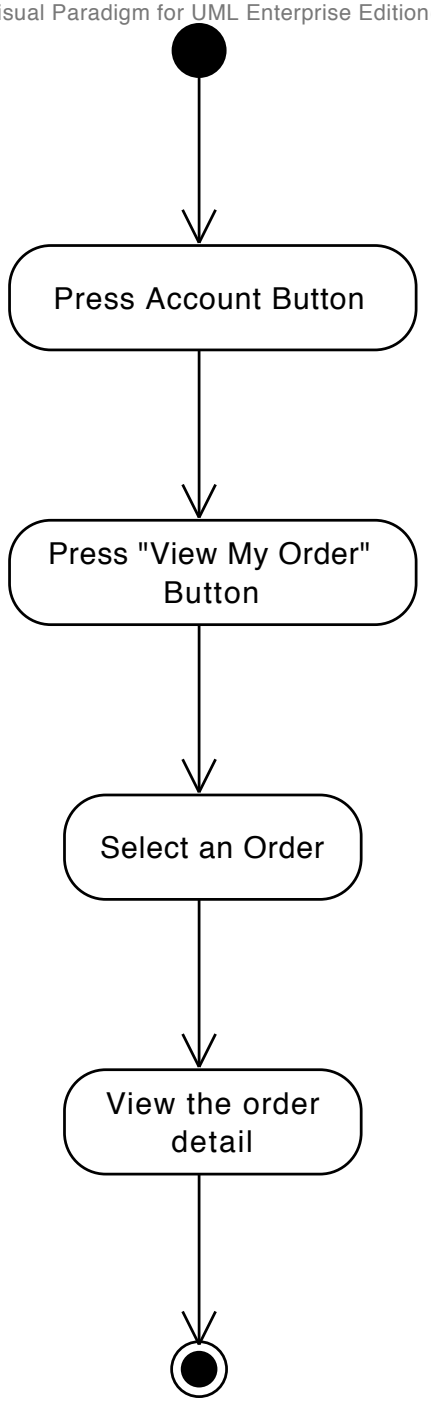

Visual Paradigm for UML Enterprise Edition(IVE_IC
Select Restaurant
Select Function Map
View Branch Location


Visual Paradigm for UML Enterprise Edition(IVE
Select Restaurant
Select function Map
Press Route Button
View Path

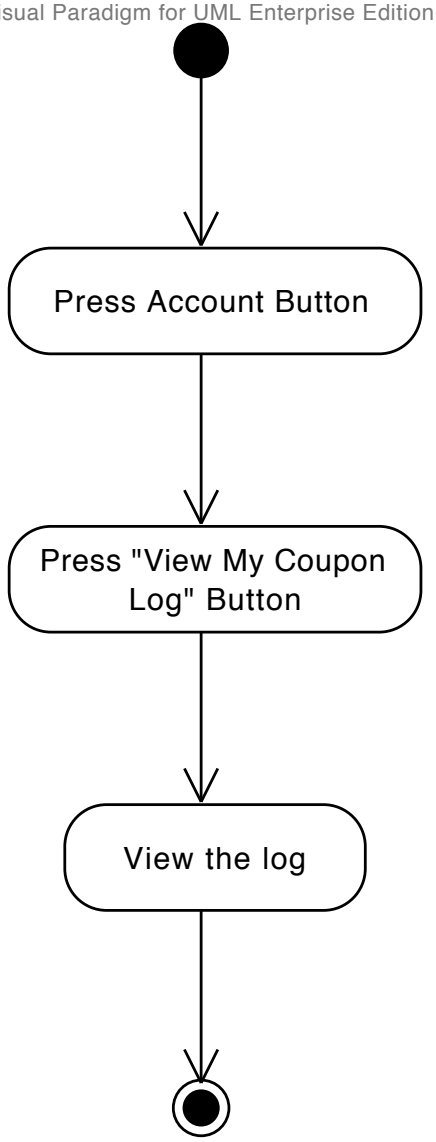

Visual Paradigm for UML Enterprise Edition(IVE_IC
Select Restaurant
Select Function Feedback
Select Feedback Form
Input Feedback
Press Submit Button


Visual Paradigm for UML Enterprise Edition(IVE
Press Account Button
Press Register Button
Fill in the data
Press Submit

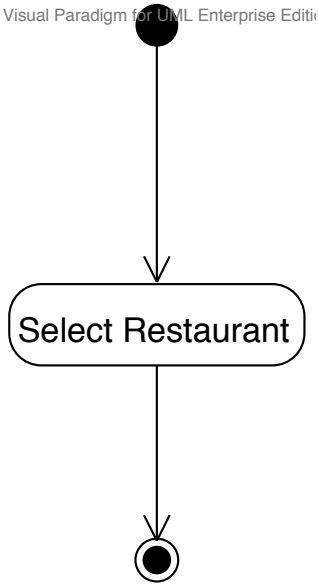
Visual Paradigm for UML Enterprise Edition(I\

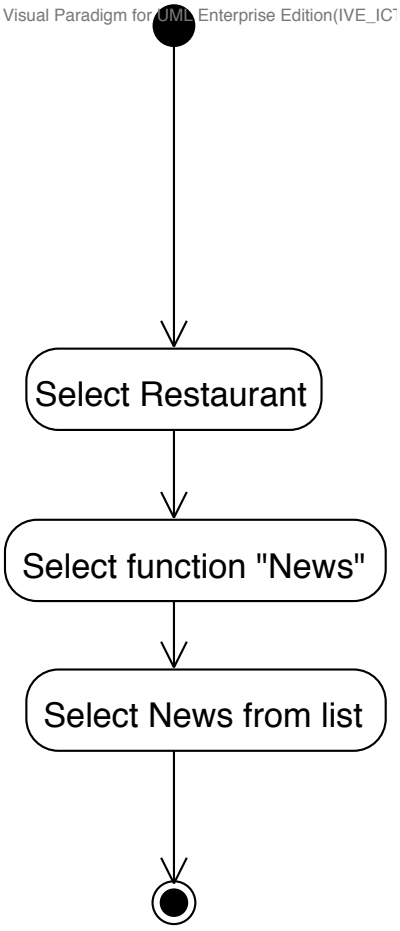


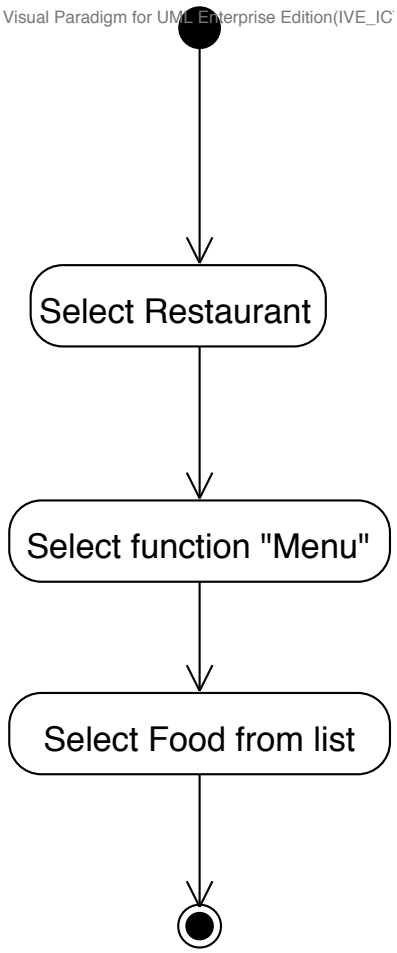
Visual Paradigm for UML Enterprise Edition(I



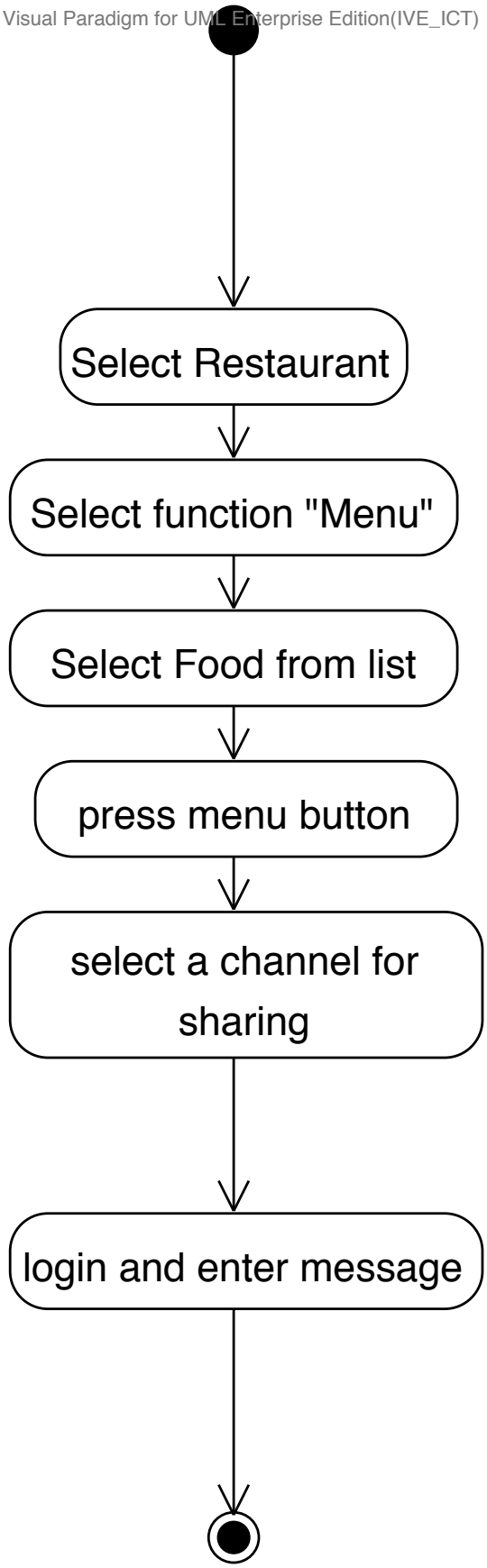
Activity Diagram (Android)

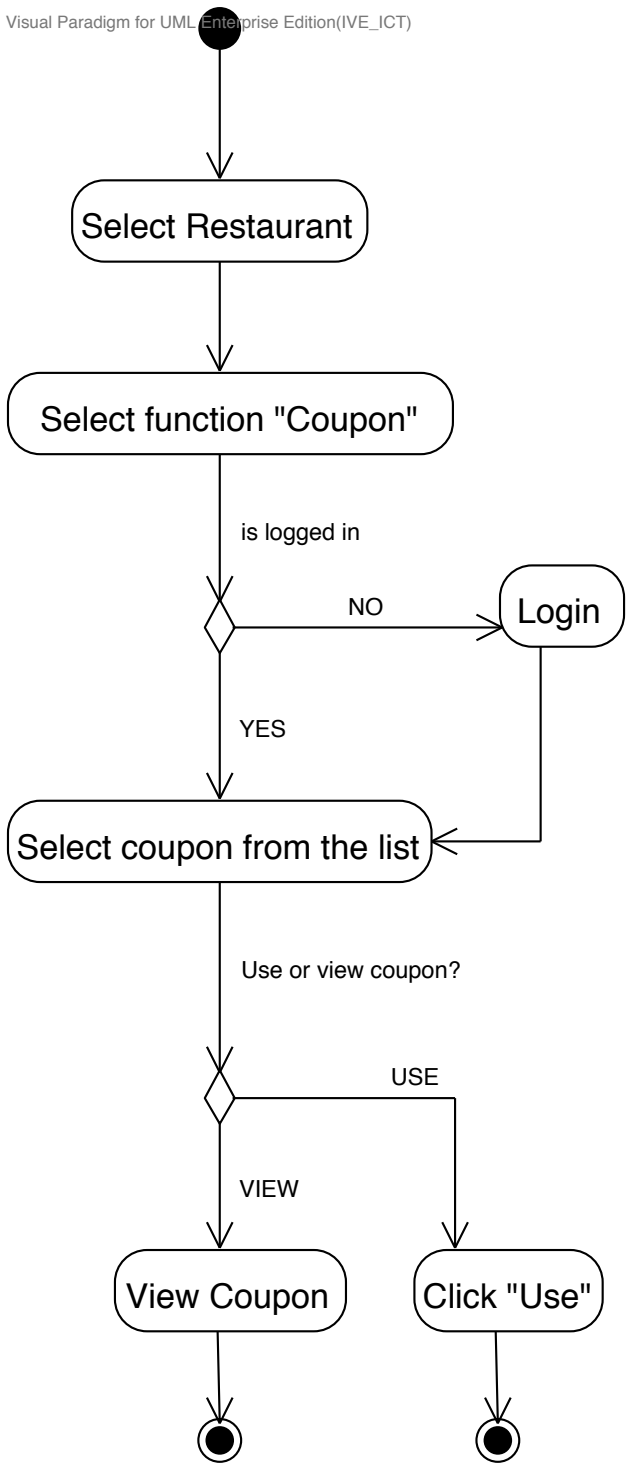


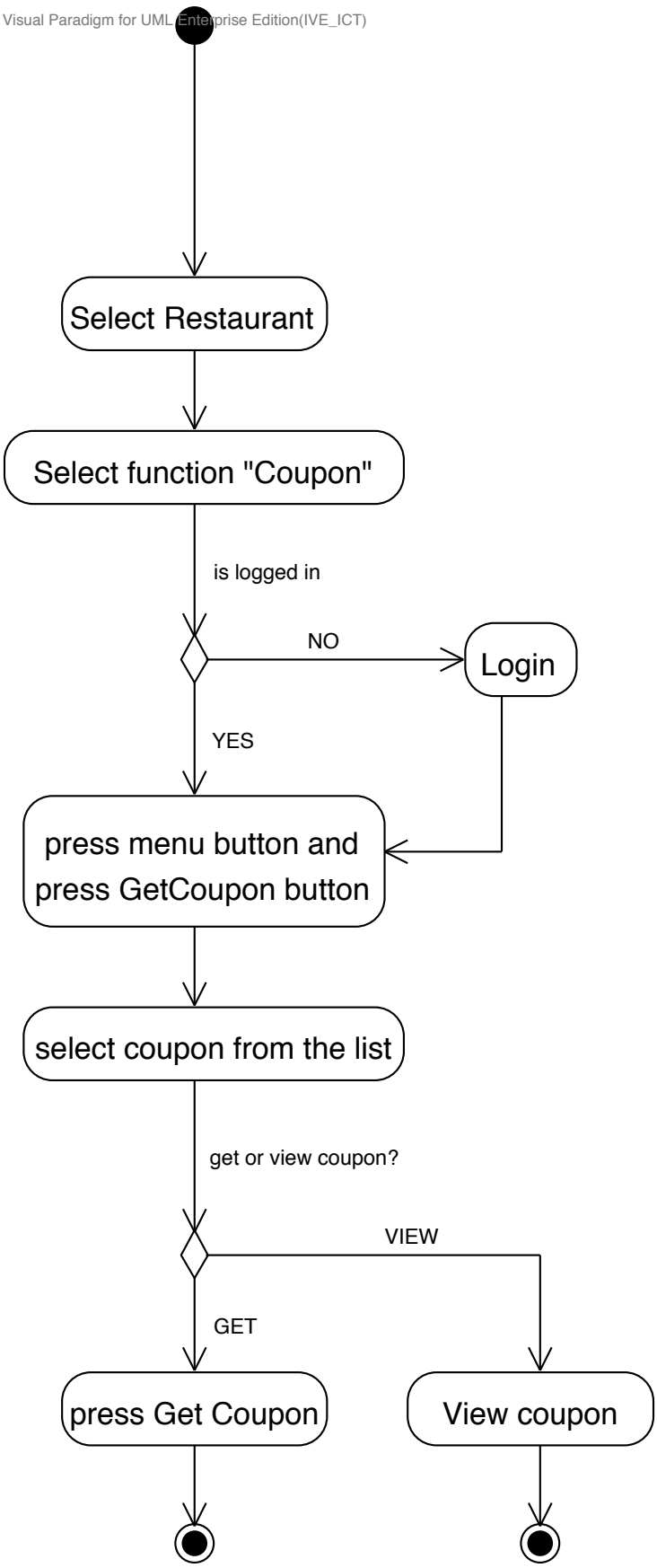


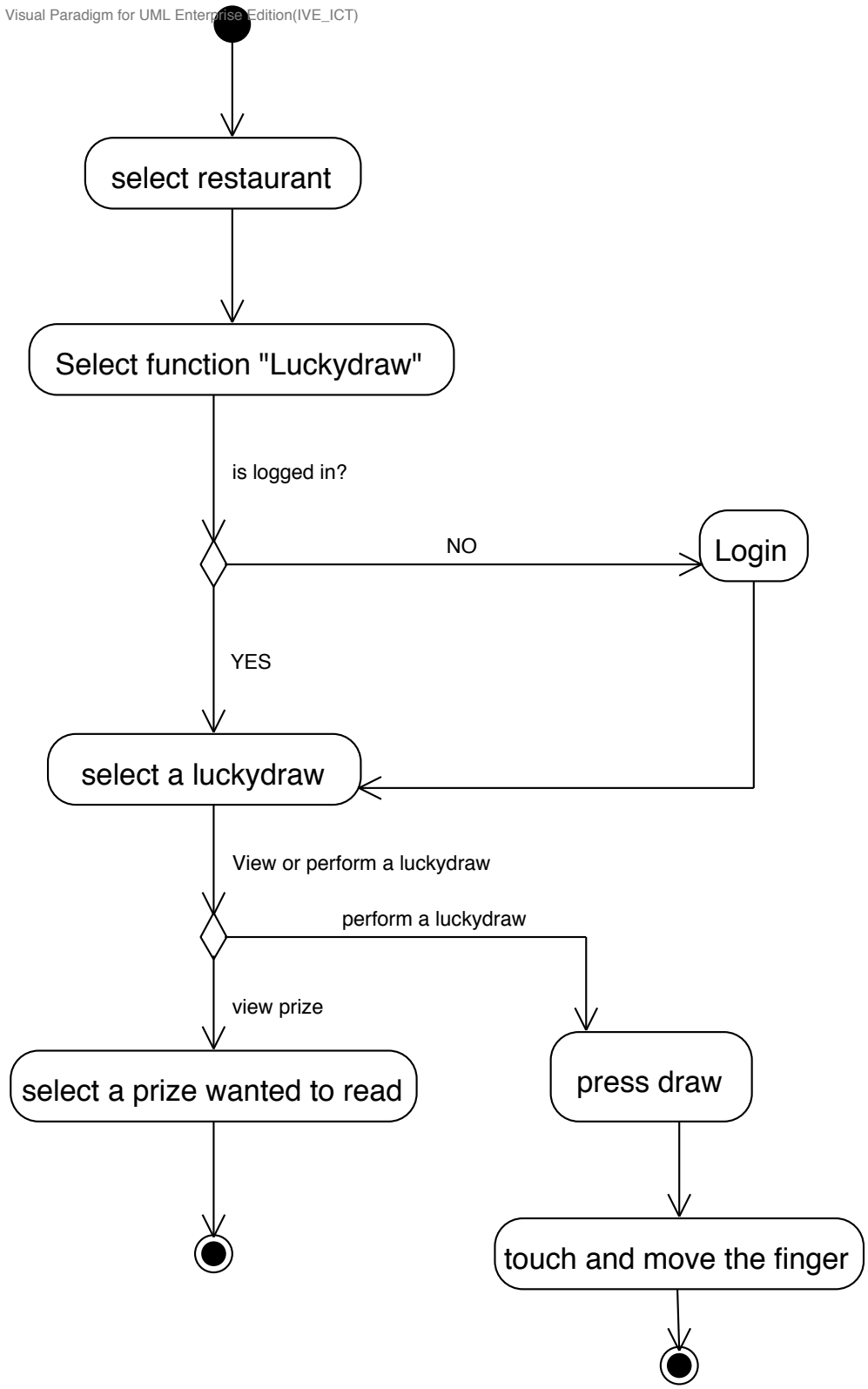

Visual Paradigm for UML Enterprise Edition(IVE_IC
Select Restaurant
Select function "Menu"
Select Food from list

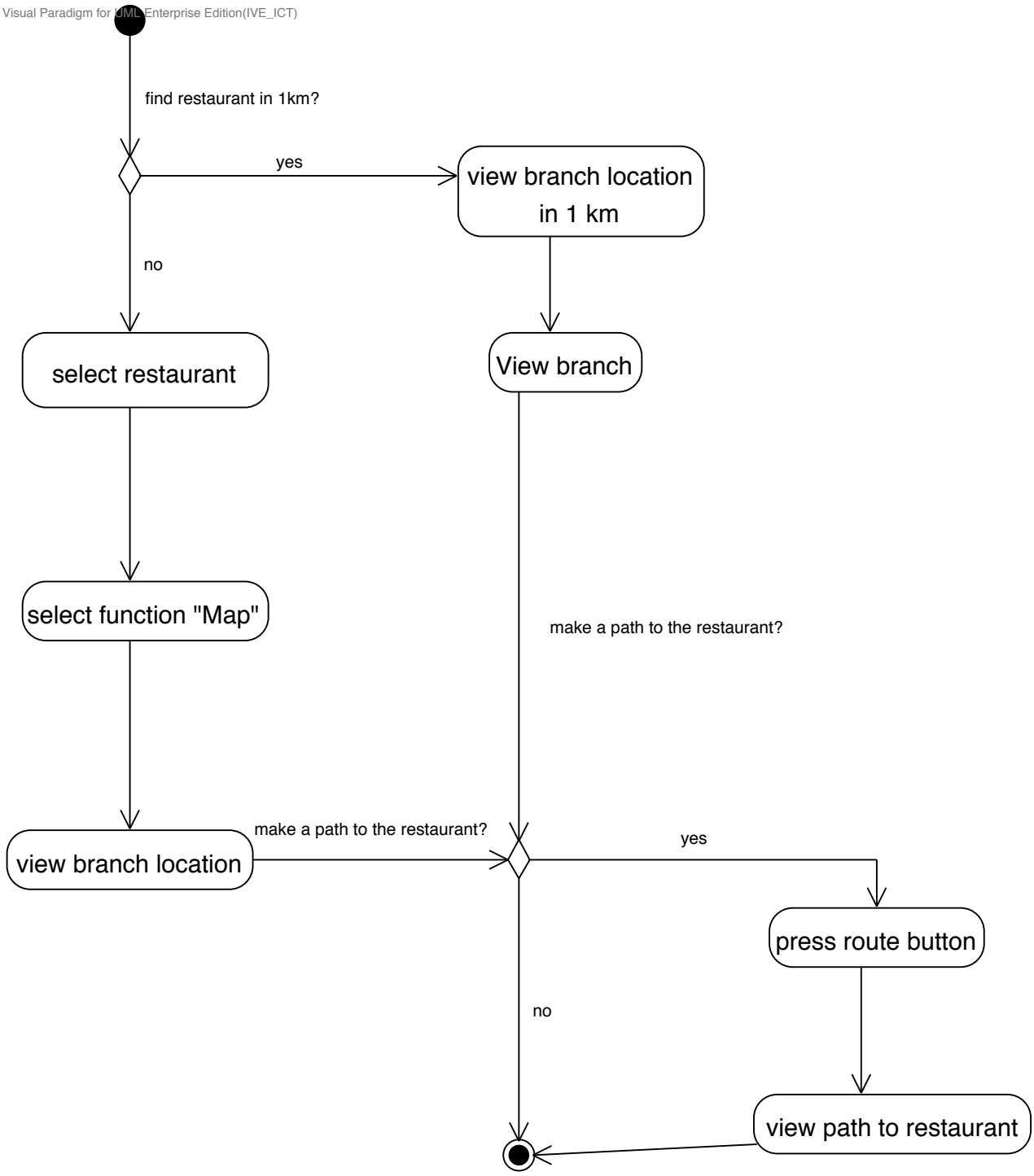
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

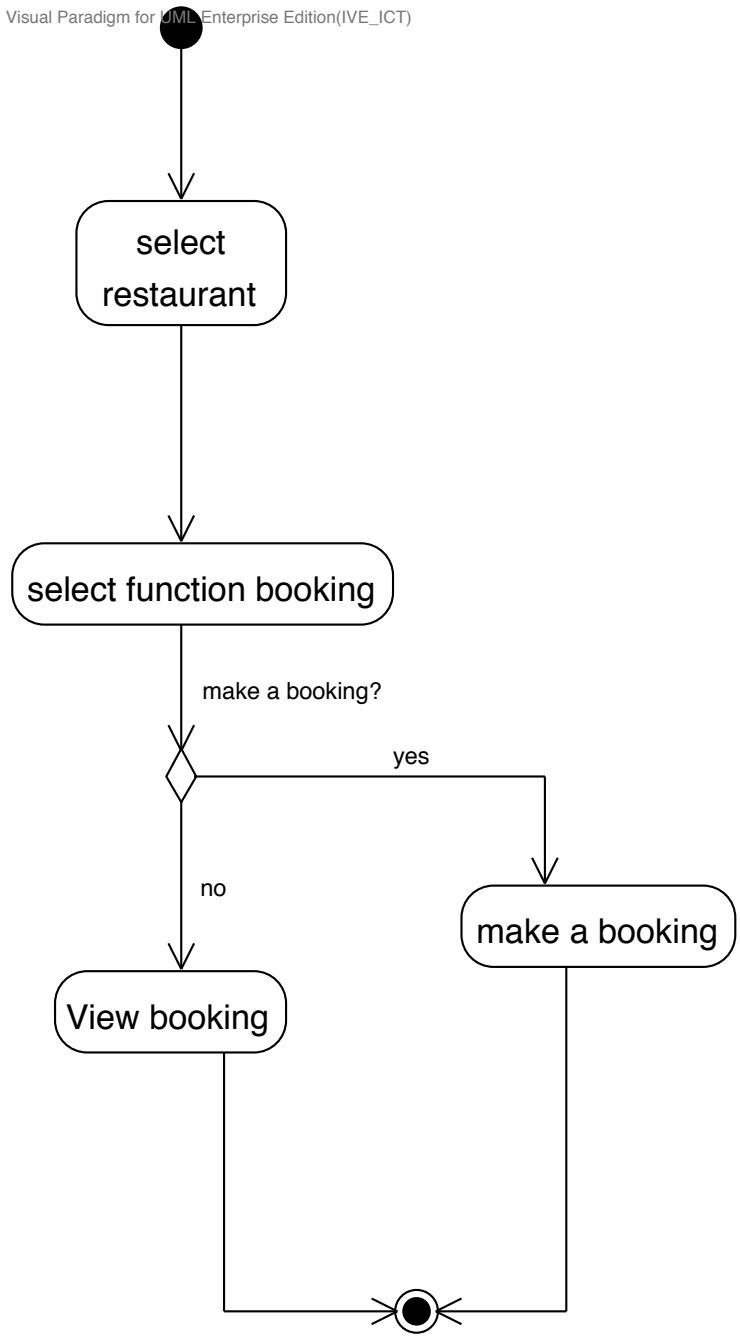


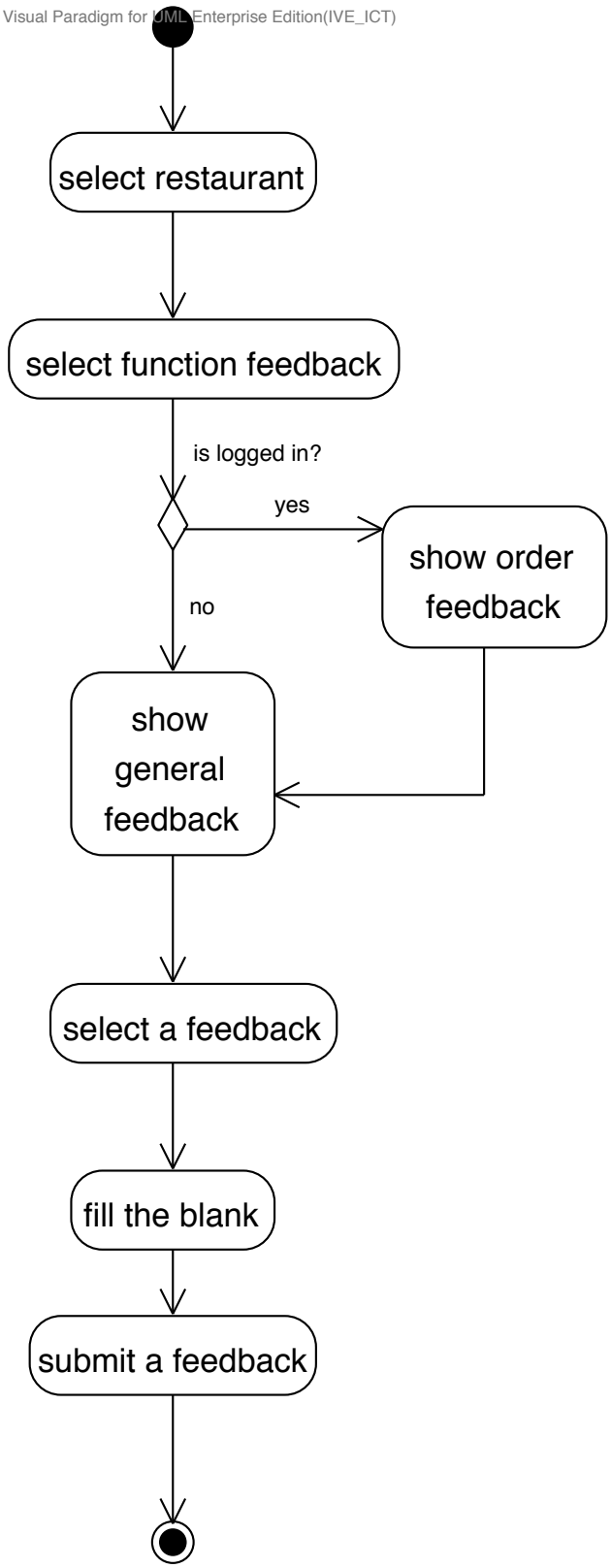

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Select Restaurant
Select function "Coupon"
is logged in
NO
Login
YES
Select coupon from the list
Use or view coupon?
USE
VIEW
View Coupon
Click "Use"

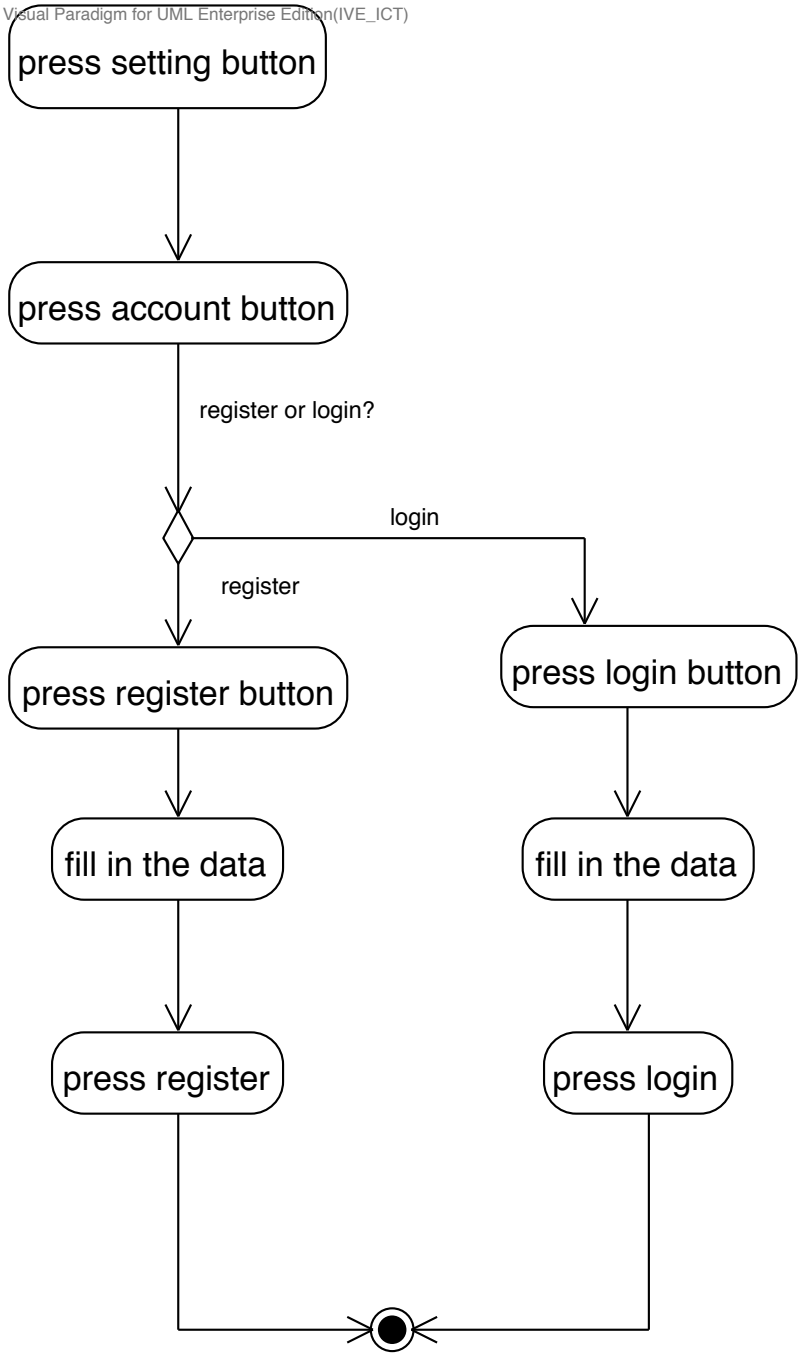

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Select Restaurant
Select function "Coupon"
is logged in
NO
Login
YES
press menu button and
press GetCoupon button
select coupon from the list
get or view coupon?
VIEW
GET
press Get Coupon
View coupon


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
select restaurant
Select function "Luckydraw"
is logged in?
NO
Login
YES
select a luckydraw
View or perform a luckydraw
perform a luckydraw
view prize
select a prize wanted to read
press draw
touch and move the finger

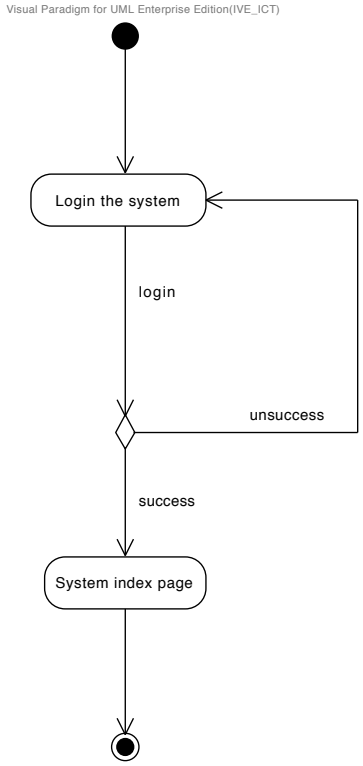

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
find restaurant in 1km?
yes
view branch location in 1 km
no
select restaurant
View branch
select function "Map"
make a path to the restaurant?
make a path to the restaurant?
yes
view branch location
press route button
no
view path to restaurant

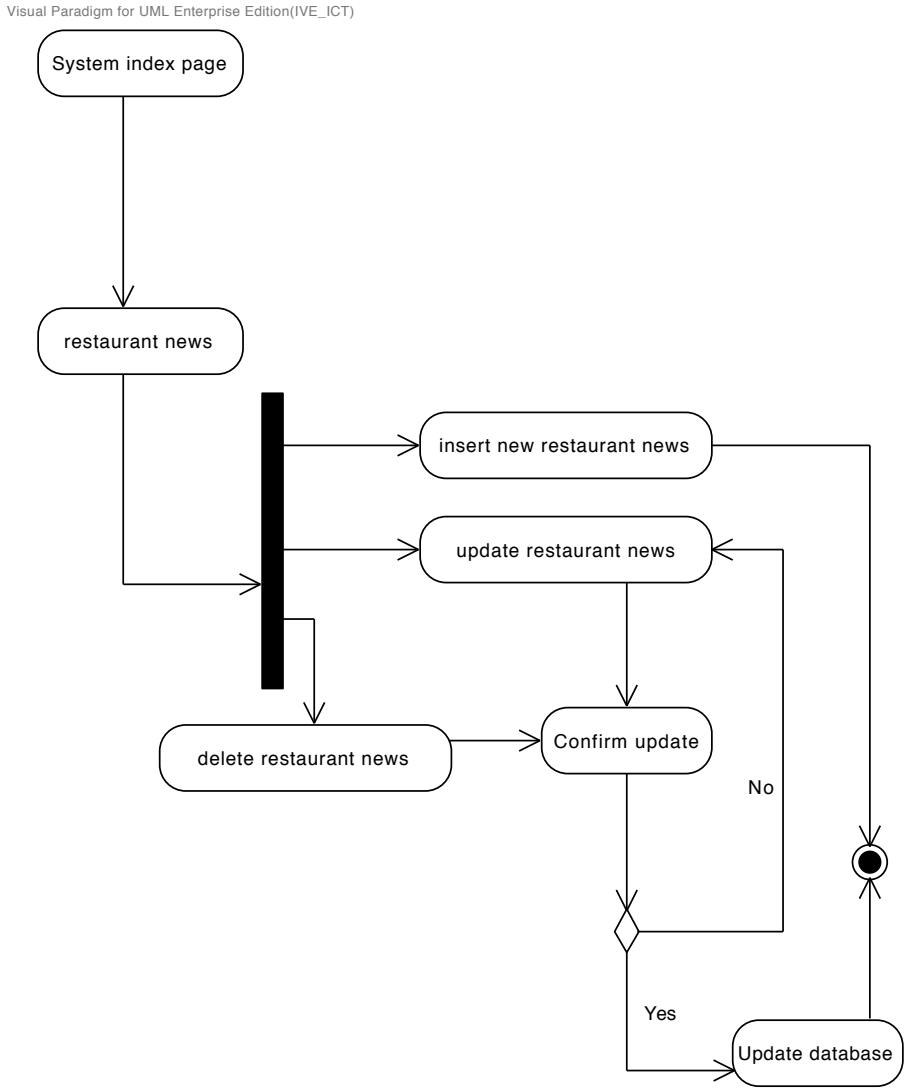

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
select restaurant
select function booking
make a booking?
yes
no
make a booking
View booking

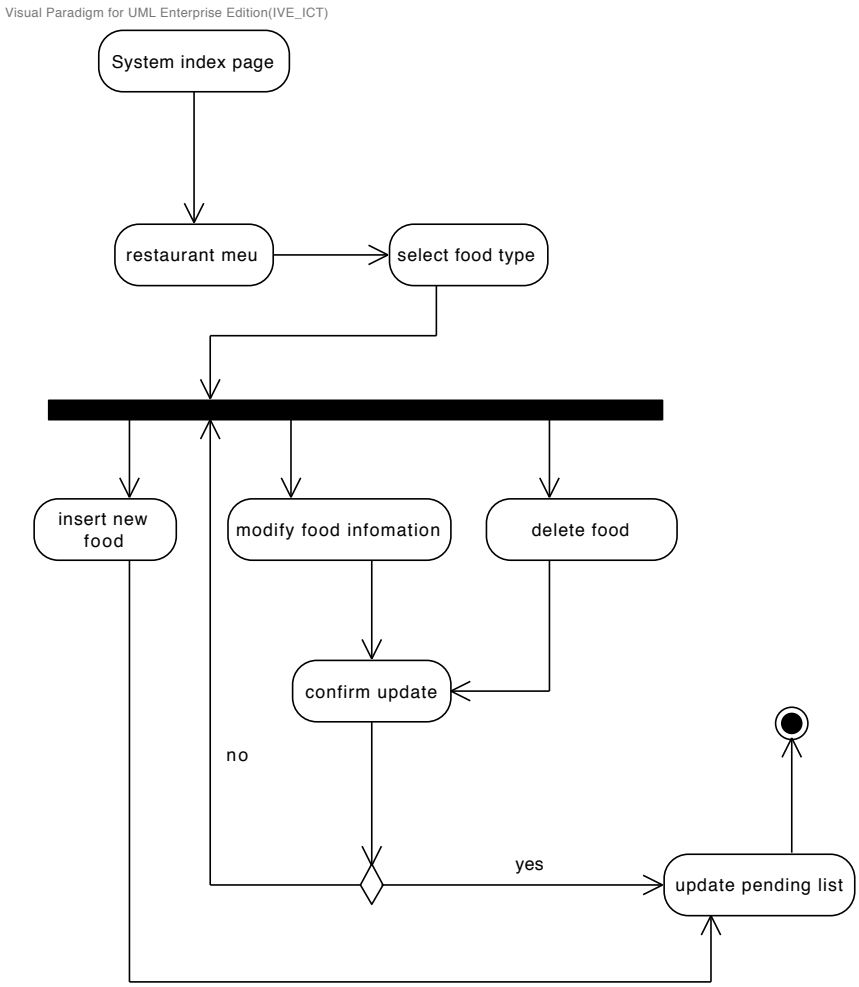

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
select restaurant
select function feedback
is logged in?
yes
show order feedback
no
show general feedback
select a feedback
fill the blank
submit a feedback

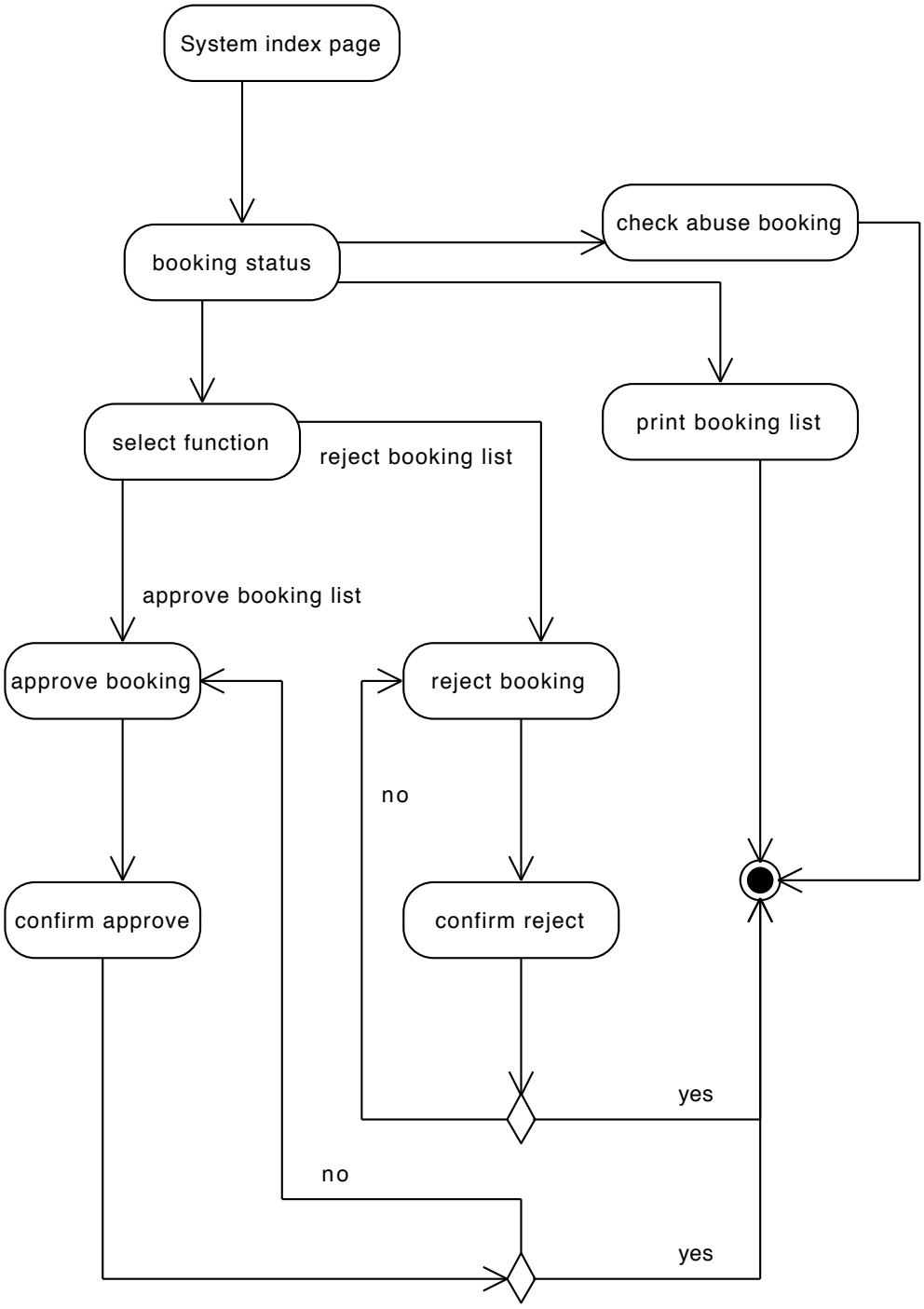

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
press setting button
press account button
register or login?
login
register
press register button
press login button
fill in the data
fill in the data
press register
press login

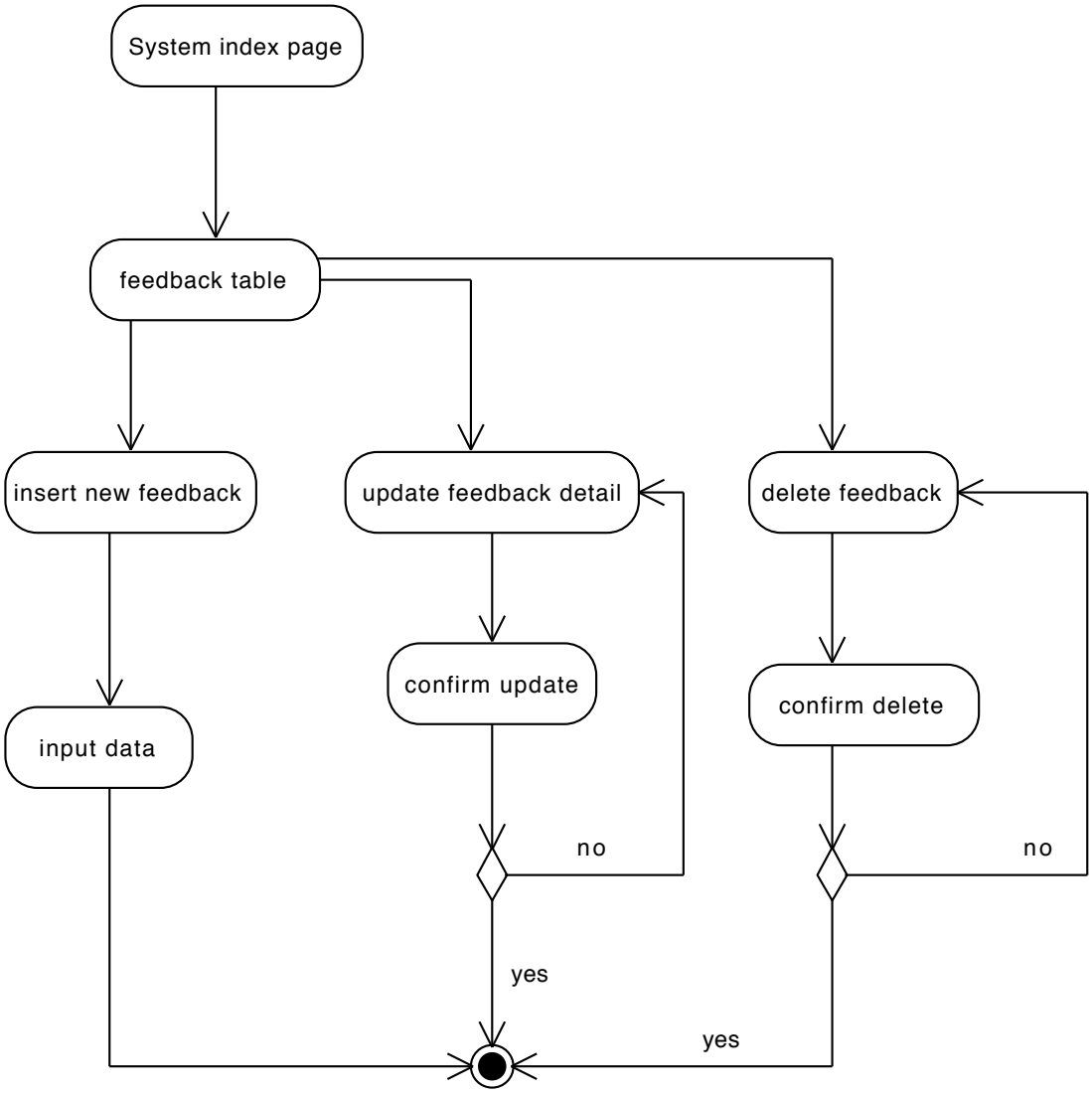
Activity Diagram (Website)

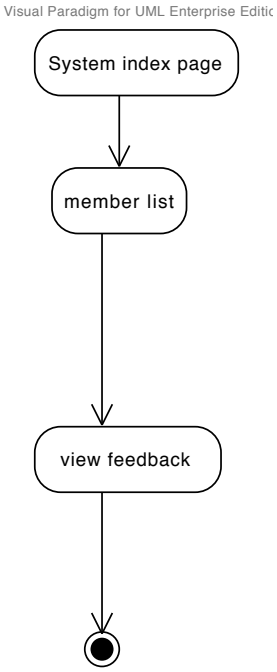


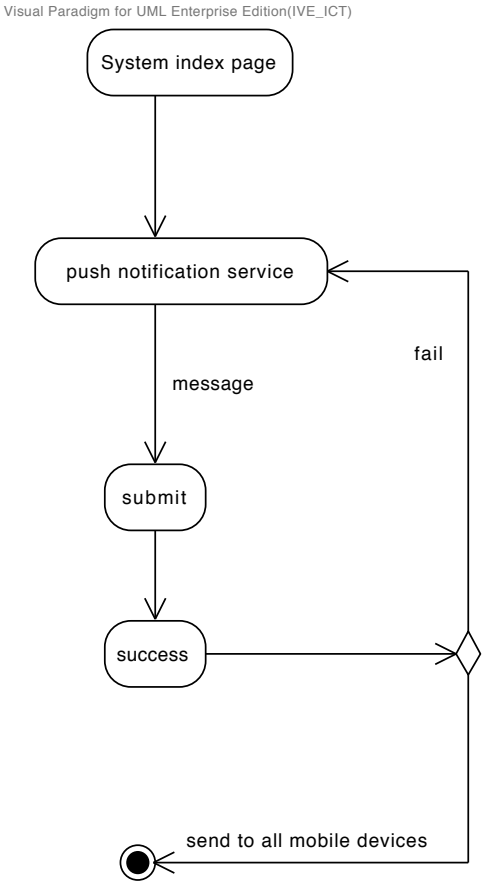

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
System index page
restaurant news
insert new restaurant news
update restaurant news
delete restaurant news
Confirm update
No
Yes
Update database


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
System index page
restaurant meu
select food type
insert new food
modify food infomation
delete food
confirm update
no
yes
update pending list

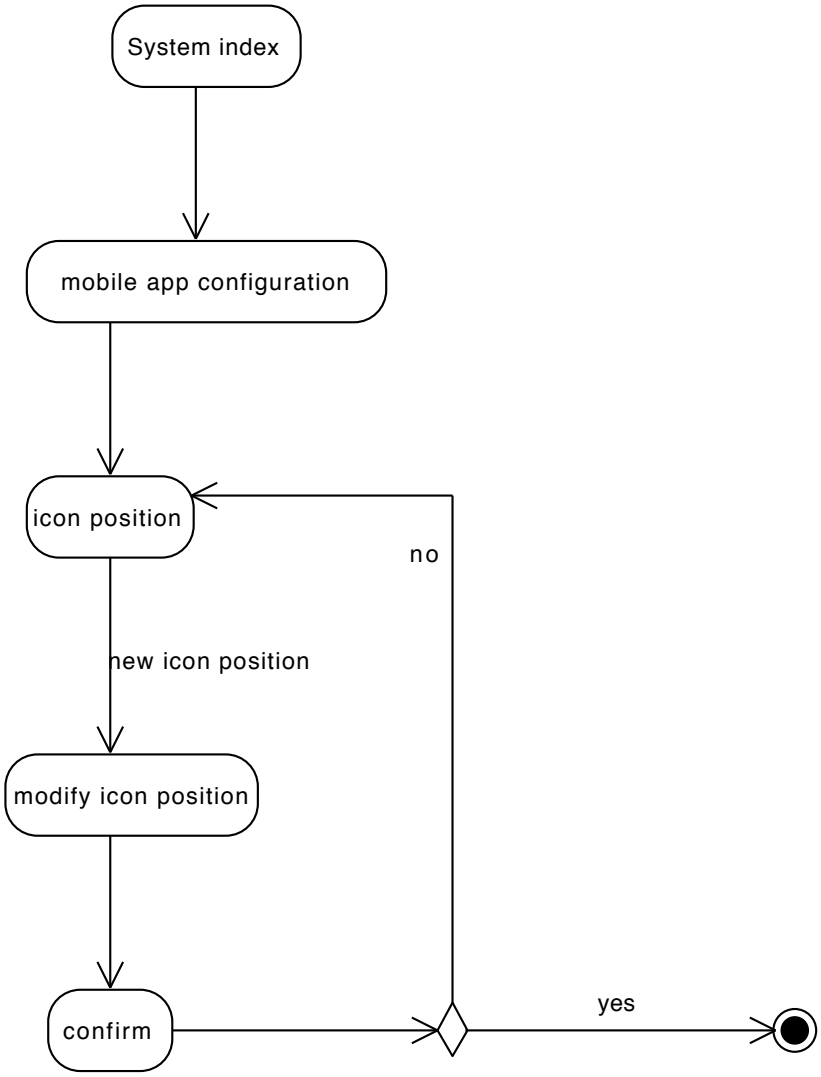

System index page
booking status
check abuse booking
print booking list
select function
reject booking list
approve booking list
approve booking
reject booking
no
confirm approve
confirm reject
yes
no
yes

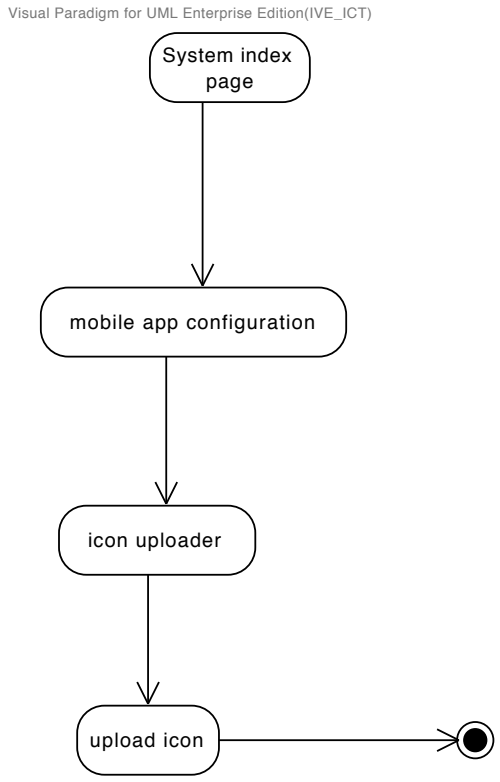

System index page
feedback table
insert new feedback
update feedback detail
delete feedback
input data
confirm update
confirm delete
no
no
yes
yes

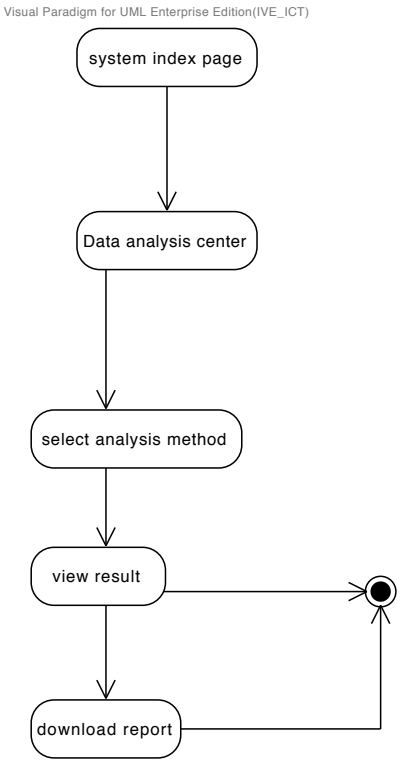

Visual Paradigm for UML Enterprise Editic
System index page
member list
view feedback


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
System index page
push notification service
fail
message
submit
success
send to all mobile devices

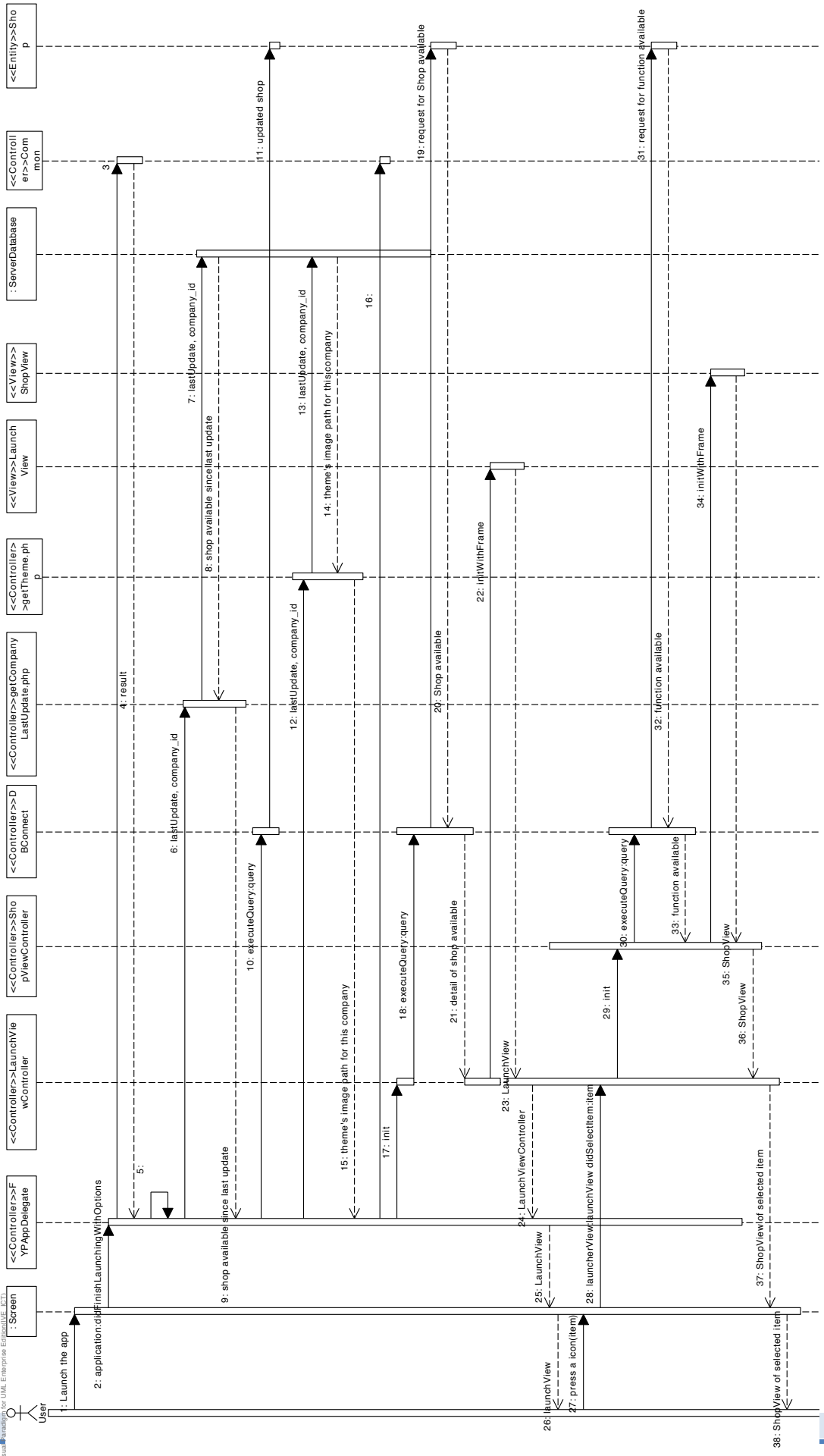
Visual Paradigm for UML Enterprise Edition(IVE_ICT)



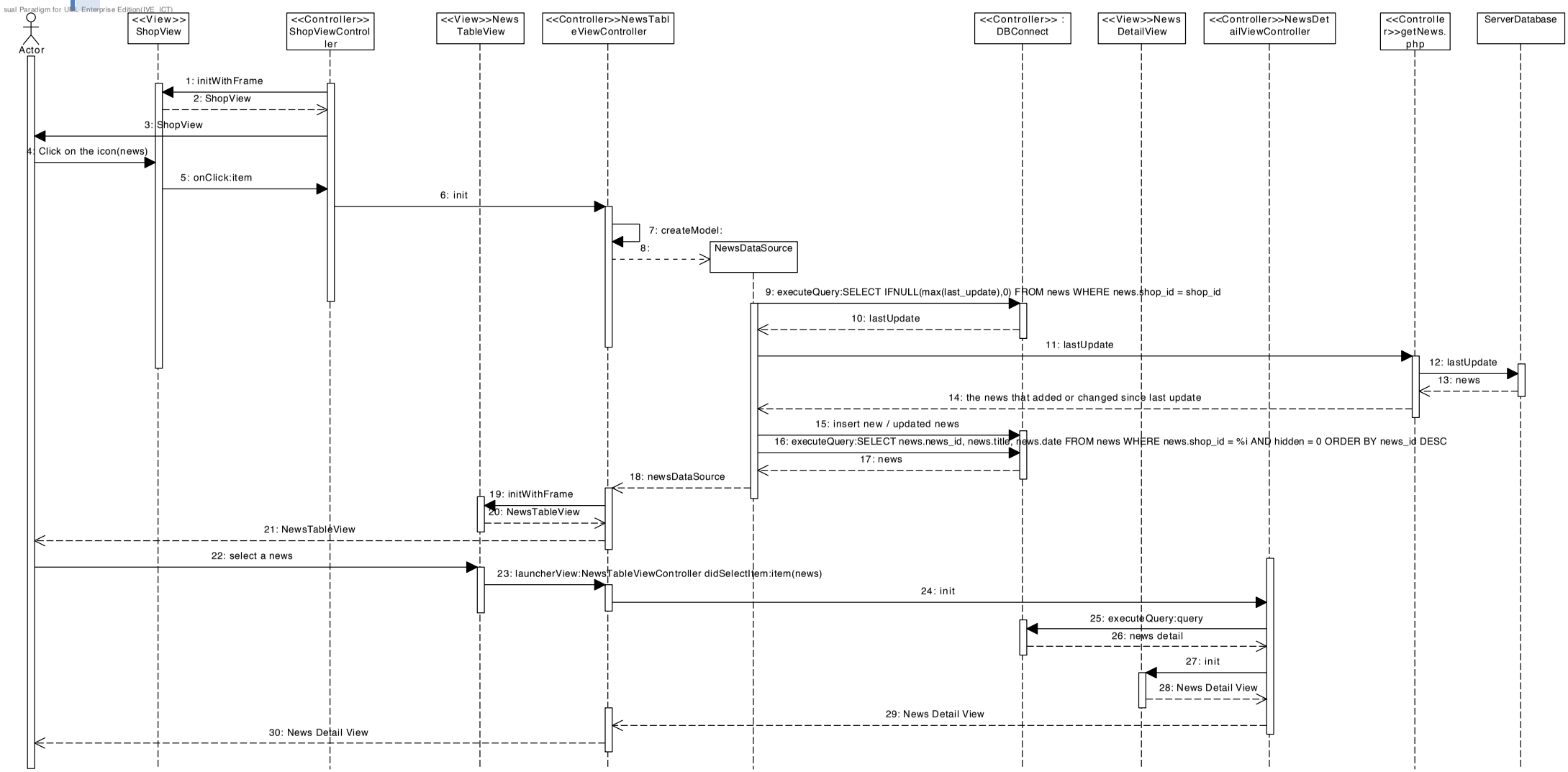

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
System index page
mobile app configuration
icon uploader
upload icon

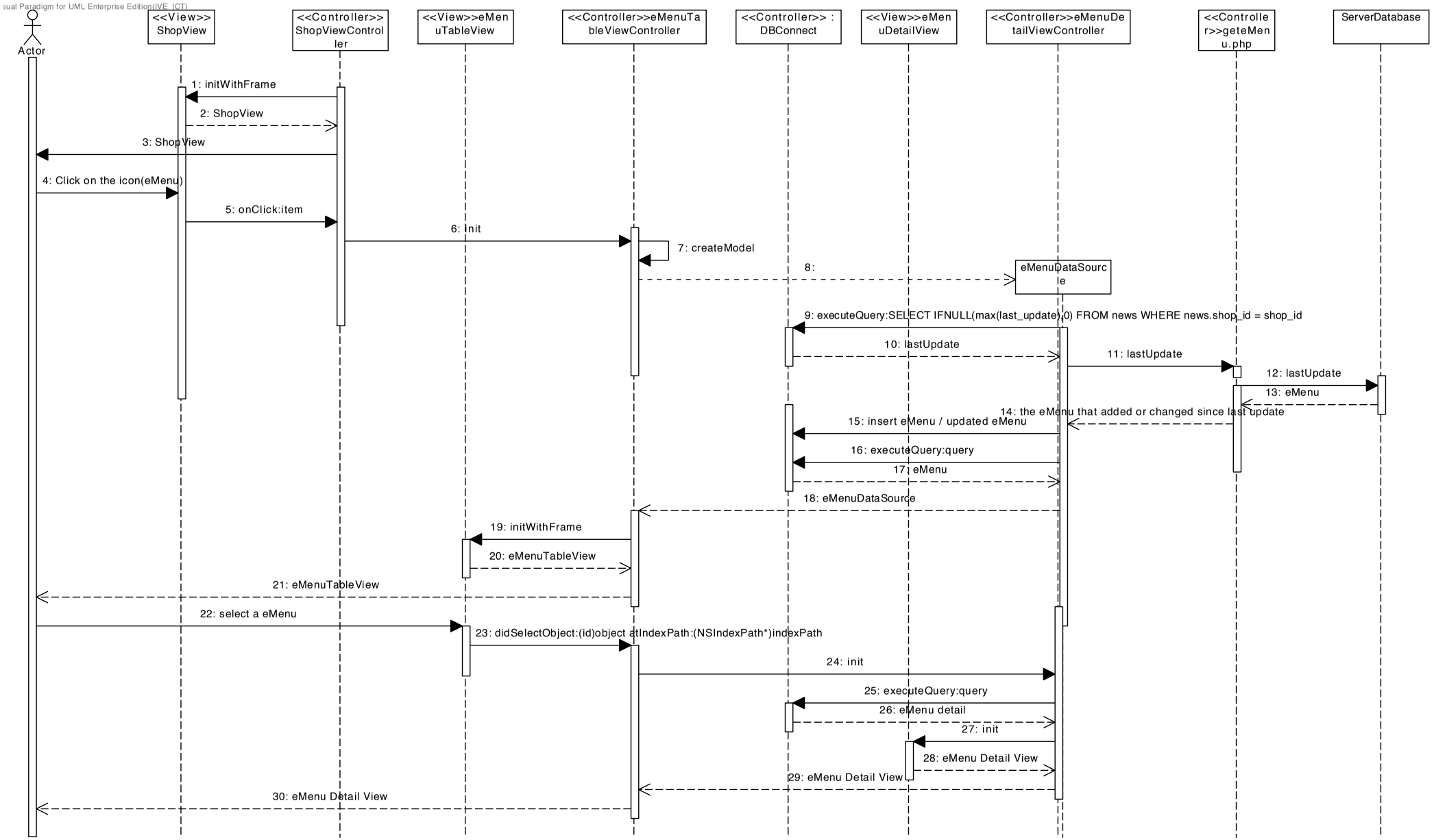

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
system index page
Data analysis center
select analysis method
view result
download report

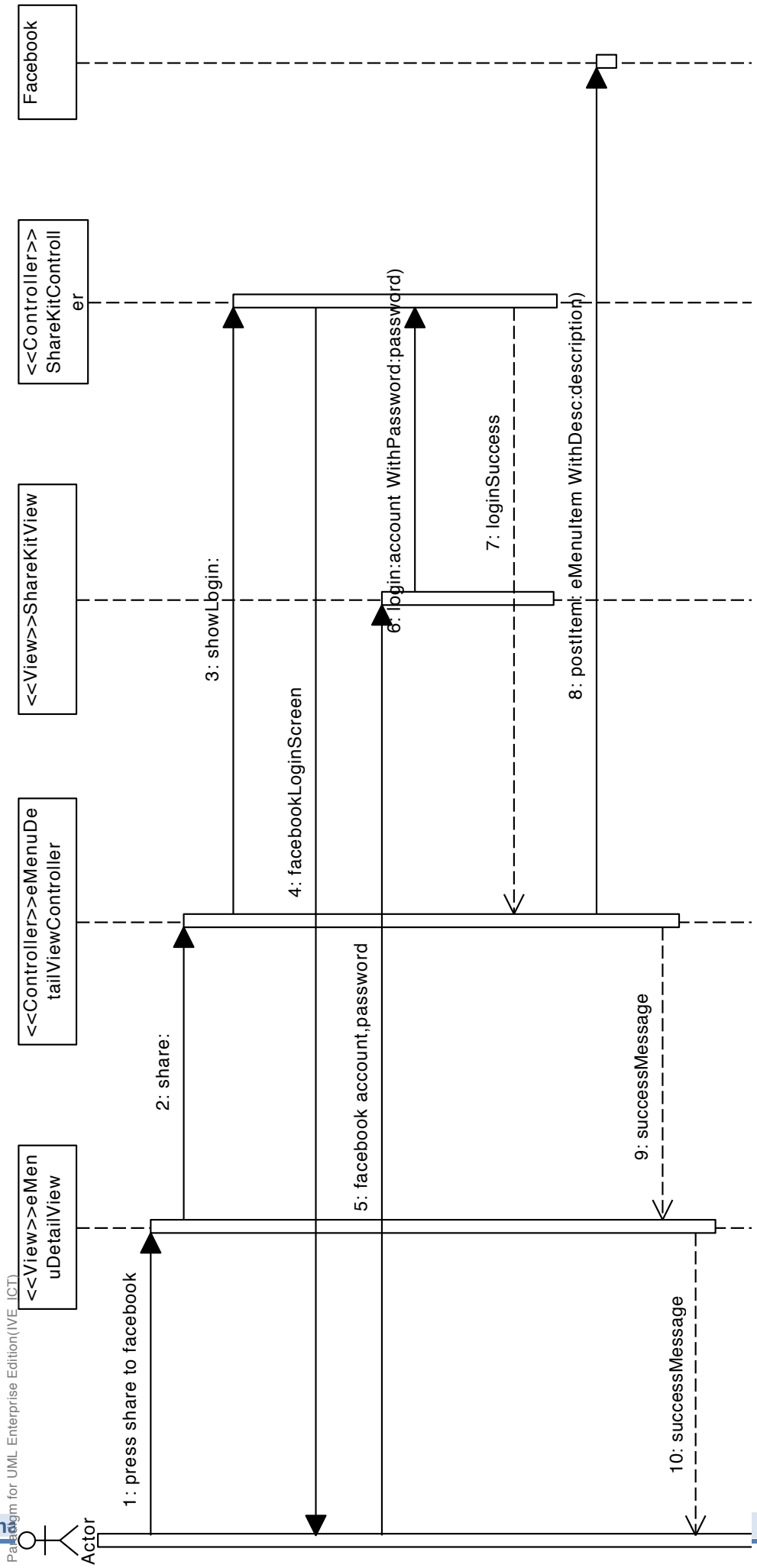
Sequence Diagram (iPhone)



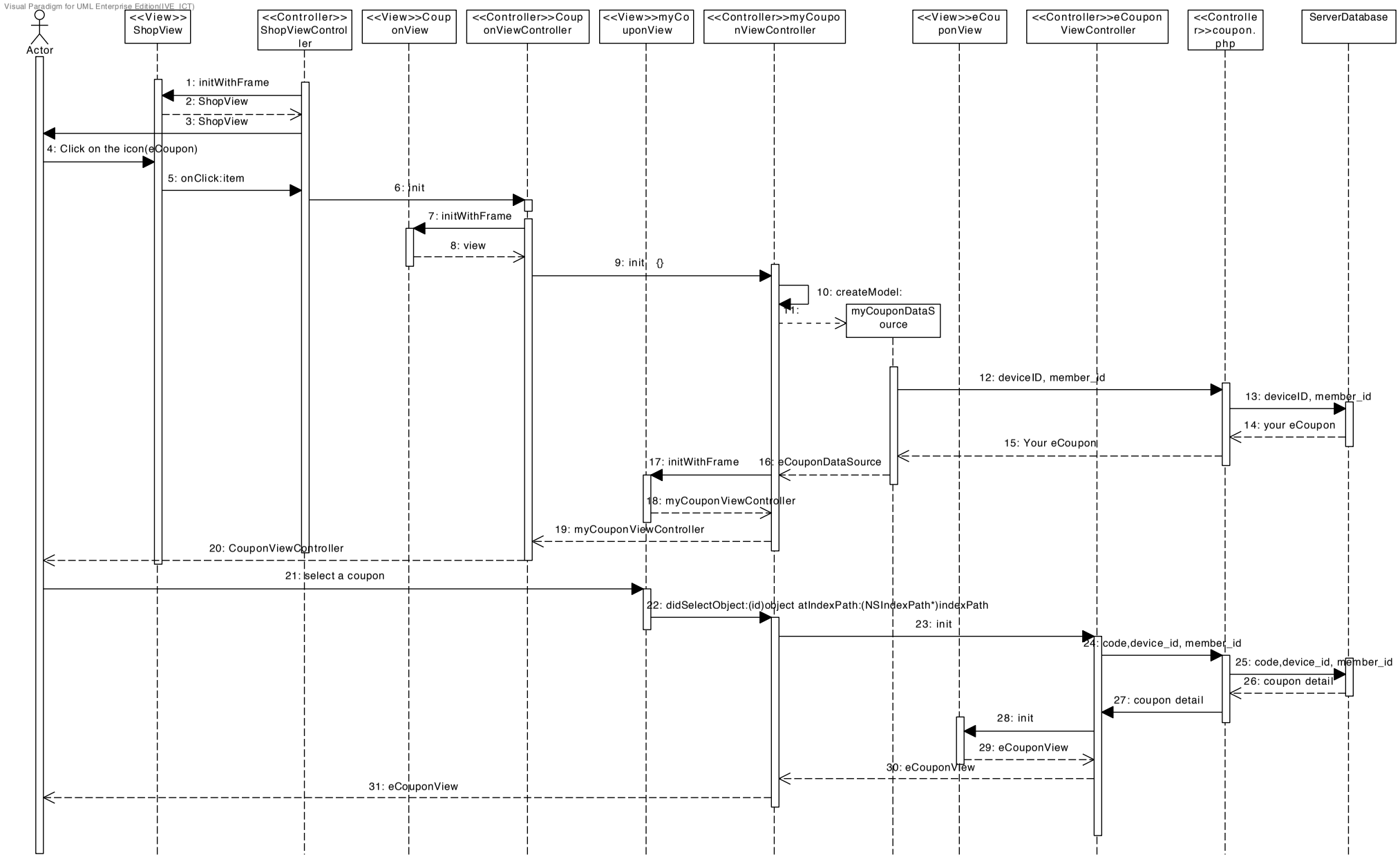
Visual Paradigm for UML Enterprise Edition(UE_621)

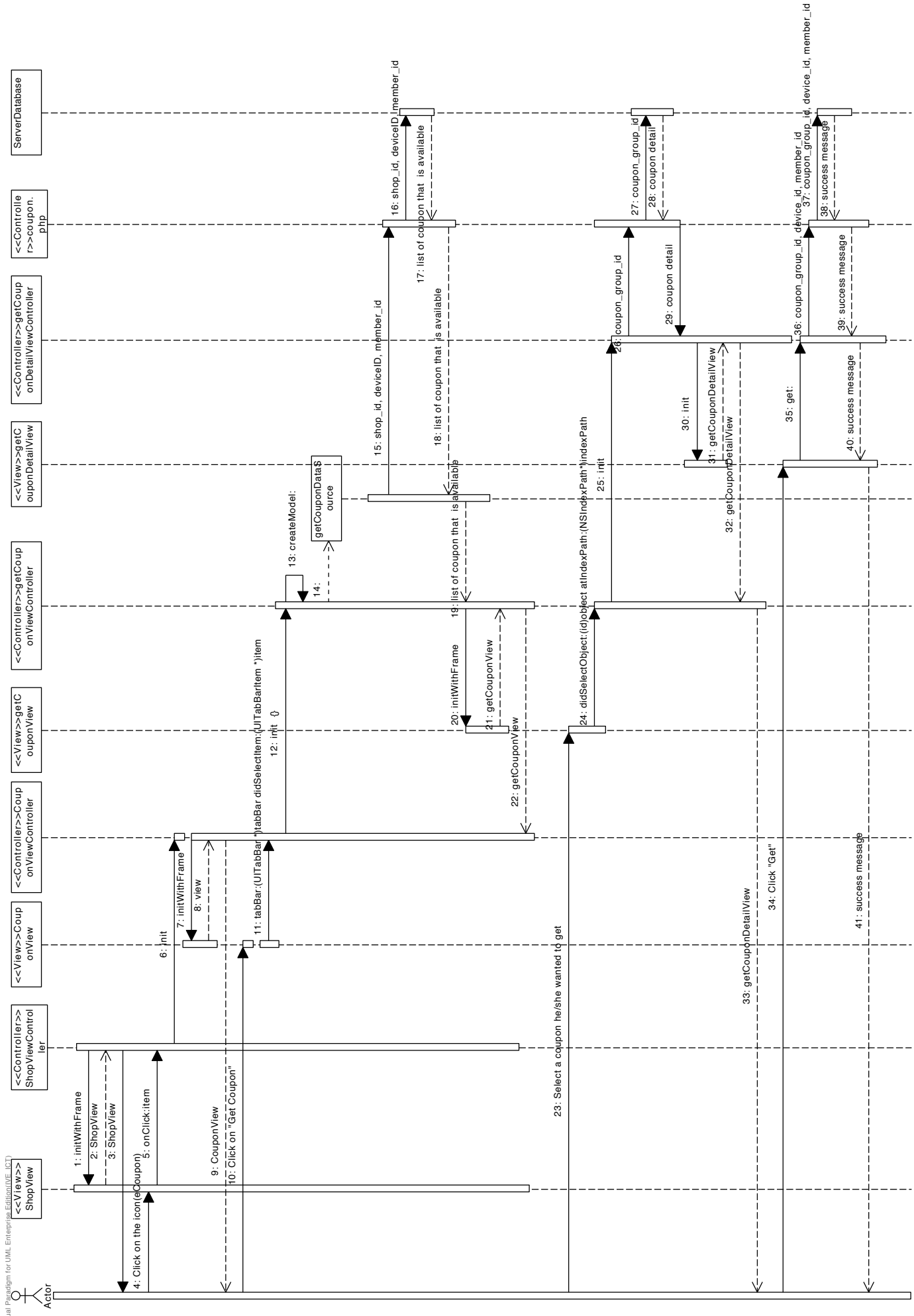

sual Paradigm for UML Enterprise Edition(IVE_ICT)
Actor
<<View>>
ShopView
<<Controller>>
ShopViewControl
ler
<<View>>News
TableView
<<Controller>>NewsTabl
eViewController
<<Controller>> :
DBConnect
<<View>>News
DetailView
<<Controller>>NewsDet
ailViewController
<<Controlle
r>>getNews.
php
ServerDatabase
1: initWithFrame
2: ShopView
3: ShopView
4: Click on the icon(news)
5: onClick:item
6: init
7: createModel:
8:
NewsDataSource
9: executeQuery:SELECT IFNULL(max(last_update),0) FROM news WHERE news.shop_id = shop_id
10: lastUpdate
11: lastUpdate
12: lastUpdate
13: news
14: the news that added or changed since last update
15: insert new / updated news
16: executeQuery:SELECT news.news_id, news.title, news.date FROM news WHERE news.shop_id = %i AND hidden = 0 ORDER BY news_id DESC
17: news
18: newsDataSource
19: initWithFrame
20: NewsTableView
21: NewsTableView
22: select a news
23: launcherView:NewsTableViewController didSelectItem:item(news)
24: init
25: executeQuery:query
26: news detail
27: init
28: News Detail View
29: News Detail View
30: News Detail View

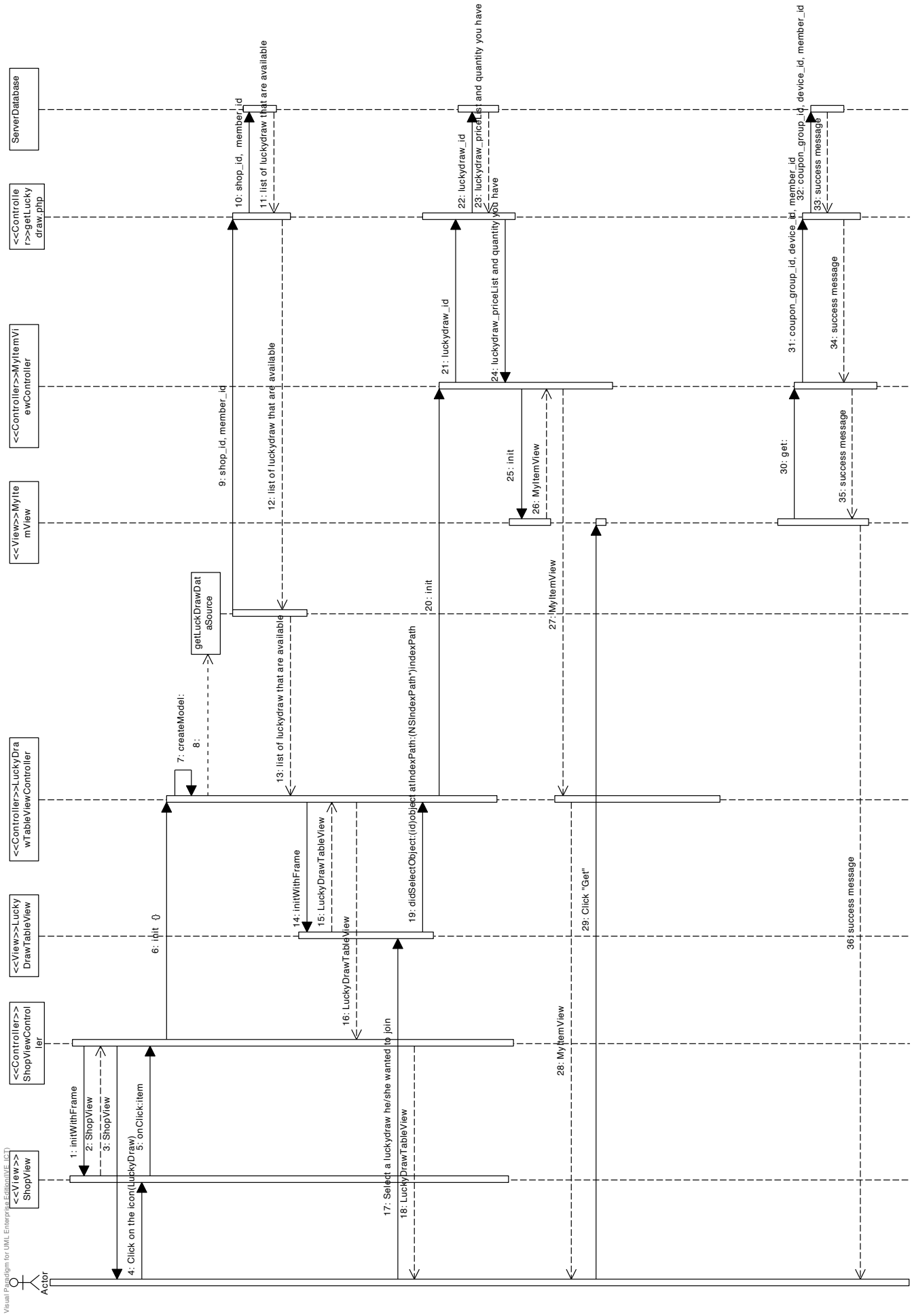

sual Paradigm for UML Enterprise Edition(IVE_ICT)
Actor
<<View>>
ShopView
<<Controller>>
ShopViewControl
ler
<<View>>eMen
uTableView
<<Controller>>eMenuTa
bleViewController
<<Controller>> :
DBConnect
<<View>>eMen
uDetailView
<<Controller>>eMenuDe
tailViewController
<<Controlle
r>>geteMen
u.php
ServerDatabase
1: initWithFrame
2: ShopView
3: ShopView
4: Click on the icon(eMenu)
5: onClick:item
6: Init
7: createModel
8:
eMenuDataSourc
e
9: executeQuery:SELECT IFNULL(max(last_update),0) FROM news WHERE news.shop_id = shop_id
10: lastUpdate
11: lastUpdate
12: lastUpdate
13: eMenu
14: the eMenu that added or changed since last update
15: insert eMenu / updated eMenu
16: executeQuery:query
17: eMenu
18: eMenuDataSource
19: initWithFrame
20: eMenuTableView
21: eMenuTableView
22: select a eMenu
23: didSelectObject:(id)object atIndexPath:(NSIndexPath*)indexPath
24: init
25: executeQuery:query
26: eMenu detail
27: init
28: eMenu Detail View
29: eMenu Detail View
30: eMenu Detail View

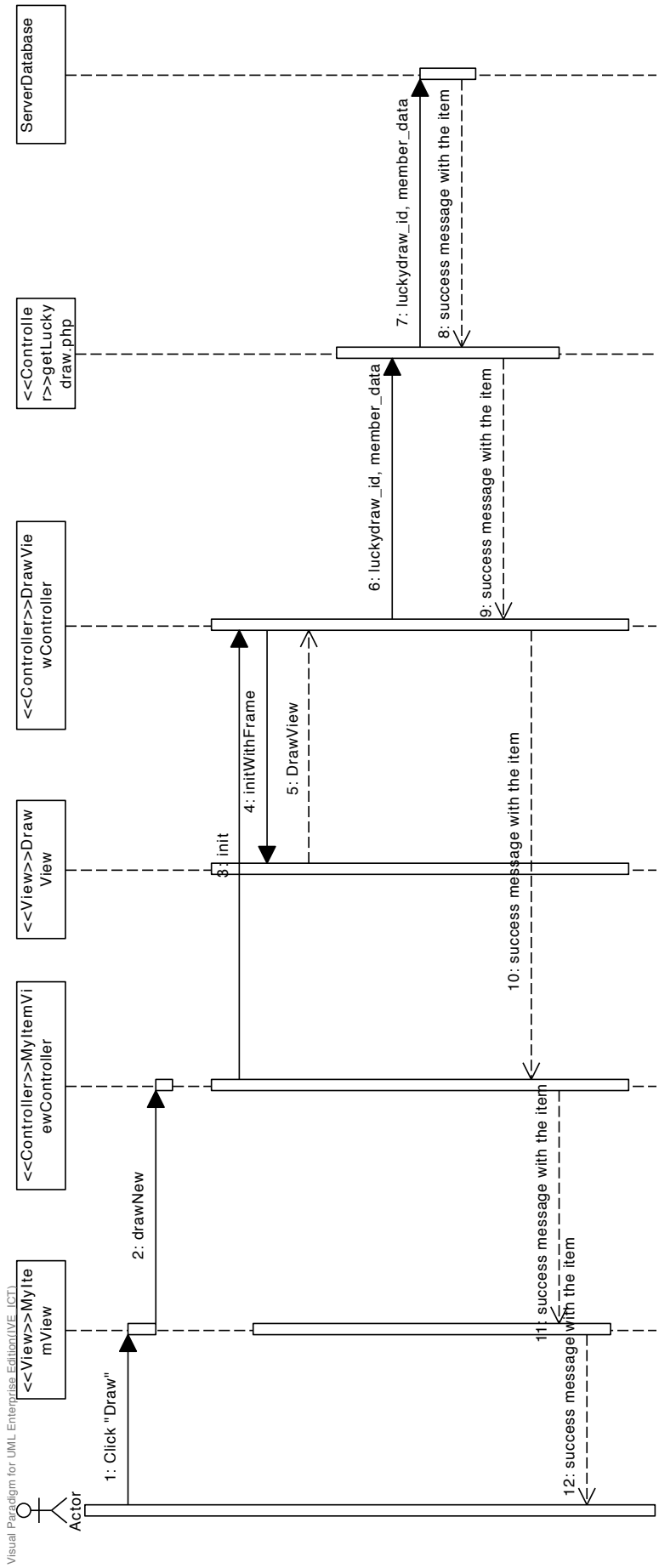


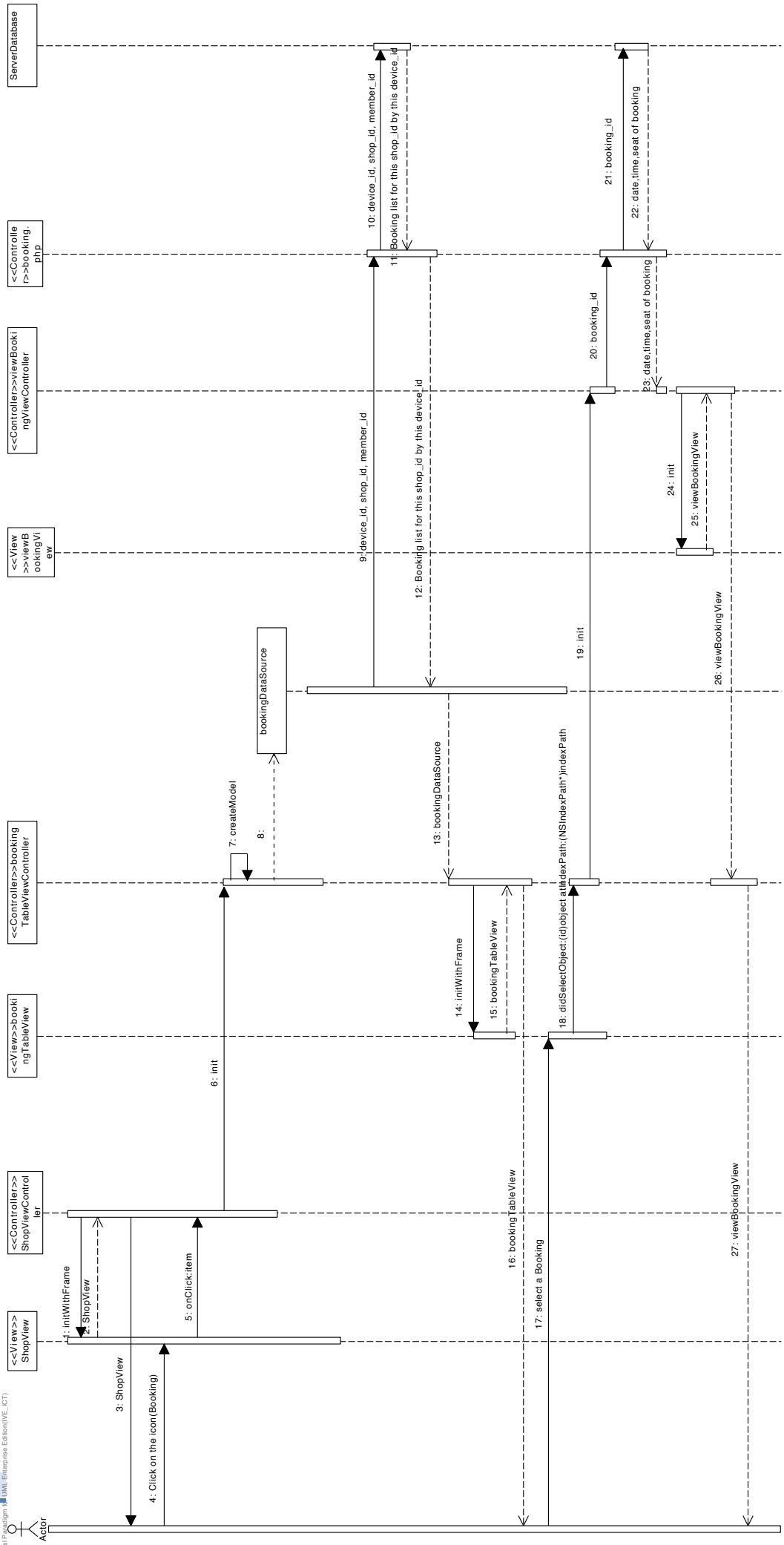
Visual Paradigm for UML Enterprise Edition(IVE_ICT)


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Actor
<<View>> ShopView
<<Controller>> ShopViewControl ler
<<View>>Coup onView
<<Controller>>Coup onViewController
<<View>>myCo uponView
<<Controller>>myCoupo nViewController
<<View>>eCou ponView
<<Controller>>eCoupon ViewController
<<Controlle r>>coupon. php
ServerDatabase
1: initWithFrame
2: ShopView
3: ShopView
4: Click on the icon(eCoupon)
5: onClick:item
6: init
7: initWithFrame
8: view
9: init {}
10: createModel:
11:
myCouponDataS ource
12: deviceID, member_id
13: deviceID, member_id
14: your eCoupon
15: Your eCoupon
16: eCouponDataSource
17: initWithFrame
18: myCouponViewController
19: myCouponViewController
20: CouponViewController
21: select a coupon
22: didSelectObject:(id)object atIndexPath:(NSIndexPath*)indexPath
23: init
24: code,device_id, member_id
25: code,device_id, member_id
26: coupon detail
27: coupon detail
28: init
29: eCouponView
30: eCouponView
31: eCouponView

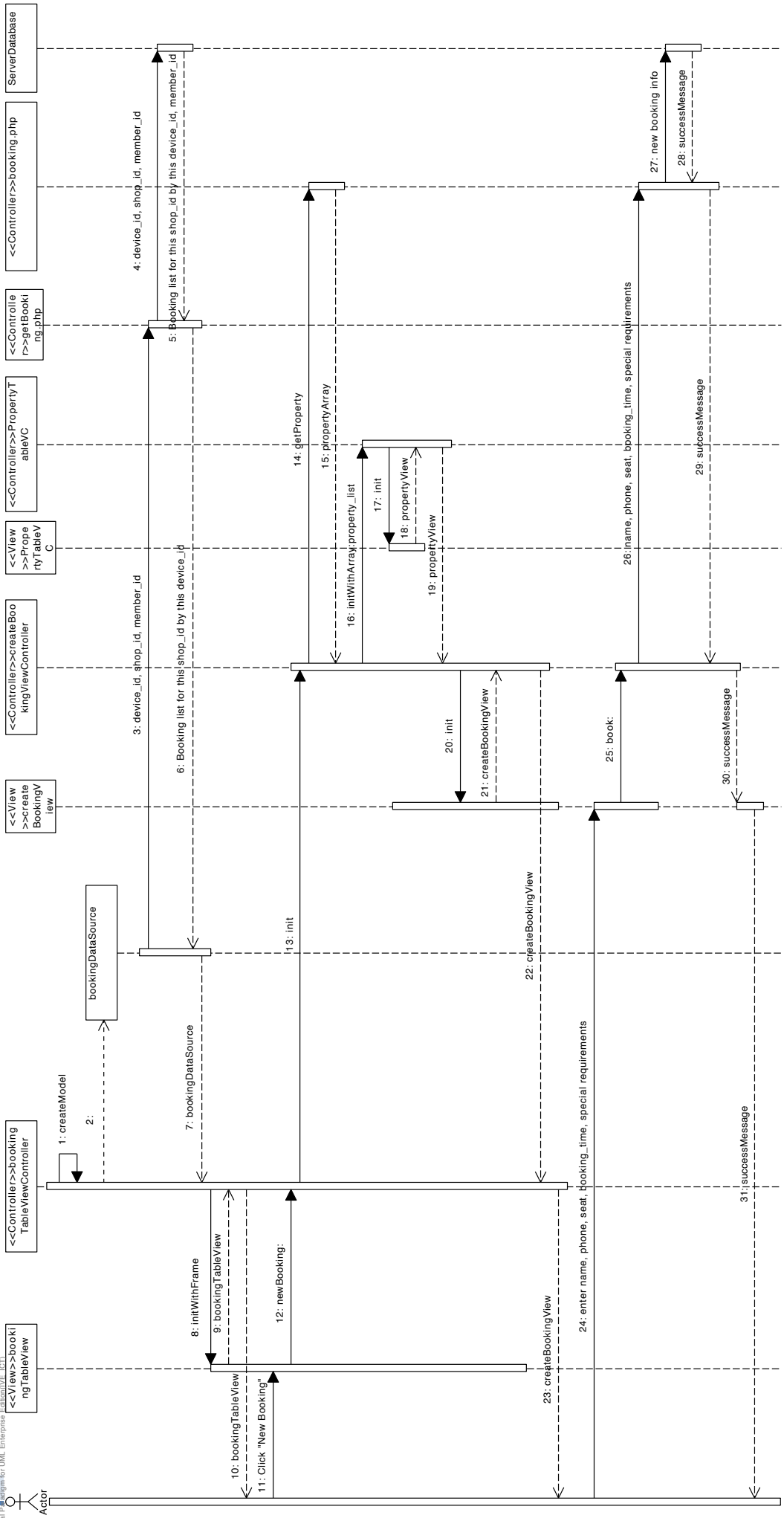

Actor
<<View>>
ShopView
<<Controller>>
ShopViewControl
ler
<<View>>Coup
onView
<<Controller>>Coup
onViewController
<<View>>getC
ouponView
<<Controller>>getCoup
onViewController
<<View>>getC
ouponDetailView
<<Controller>>getCoup
onDetailViewController
<<Controlle
r>>coupon.
php
ServerDatabase
1: initWithFrame
2: ShopView
3: ShopView
4: Click on the icon(eCoupon)
5: onClick:item
6: init
7: initWithFrame
8: view
9: CouponView
10: Click on "Get Coupon"
11: tabBar:(UITabBar *)tabBar didSelectItem:(UITabBarItem *)item
12: init ()
13: createModel:
14:
getCouponData$
ource
15: shop_id, deviceID, member_id
16: shop_id, deviceID, member_id
17: list of coupon that is available
18: list of coupon that is available
19: list of coupon that is available
20: initWithFrame
21: getCouponView
22: getCouponView
23: Select a coupon he/she wanted to get
24: didSelectObject:(id)object atIndexPath:(NSIndexPath *)indexPath
25: init
26: coupon_group_id
27: coupon_group_id
28: coupon detail
29: coupon detail
30: init
31: getCouponDetailView
32: getCouponDetailView
33: getCouponDetailView
34: Click "Get"
35: get:
36: coupon_group_id, device_id, member_id
37: coupon_group_id, device_id, member_id
38: success message
39: success message
40: success message
41: success message


Actor
<<View>>
ShopView
<<Controller>>
ShopViewControl
ler
<<View>>Lucky
DrawTableView
<<Controller>>LuckyDra
wTableViewController
<<View>>MyIte
mView
<<Controller>>MyItemVi
ewController
<<Controlle
r>>getLucky
draw.php
ServerDatabase
getLuckDrawDat
aSource
1: initWithFrame
2: ShopView
3: ShopView
4: Click on the icon(LuckyDraw)
5: onClick:item
6: init {}
7: createModel:
8:
9: shop_id, member_id
10: shop_id, member_id
11: list of luckydraw that are available
12: list of luckydraw that are available
13: list of luckydraw that are available
14: initWithFrame
15: LuckyDrawTableView
16: LuckyDrawTableView
17: Select a luckydraw he/she wanted to join
18: LuckyDrawTableView
19: didSelectObject:(id)object atIndexPath:(NSIndexPath*)indexPath
20: init
21: luckydraw_id
22: luckydraw_id
23: luckydraw_priceList and quantity you have
24: luckydraw_priceList and quantity you have
25: init
26: MyItemView
27: MyItemView
28: MyItemView
29: Click "Get"
30: get:
31: coupon_group_id, device_id, member_id
32: coupon_group_id, device_id, member_id
33: success message
34: success message
35: success message
36: success message
Visual Paradigm for UML Enterprise Edition(UVE_ICT)

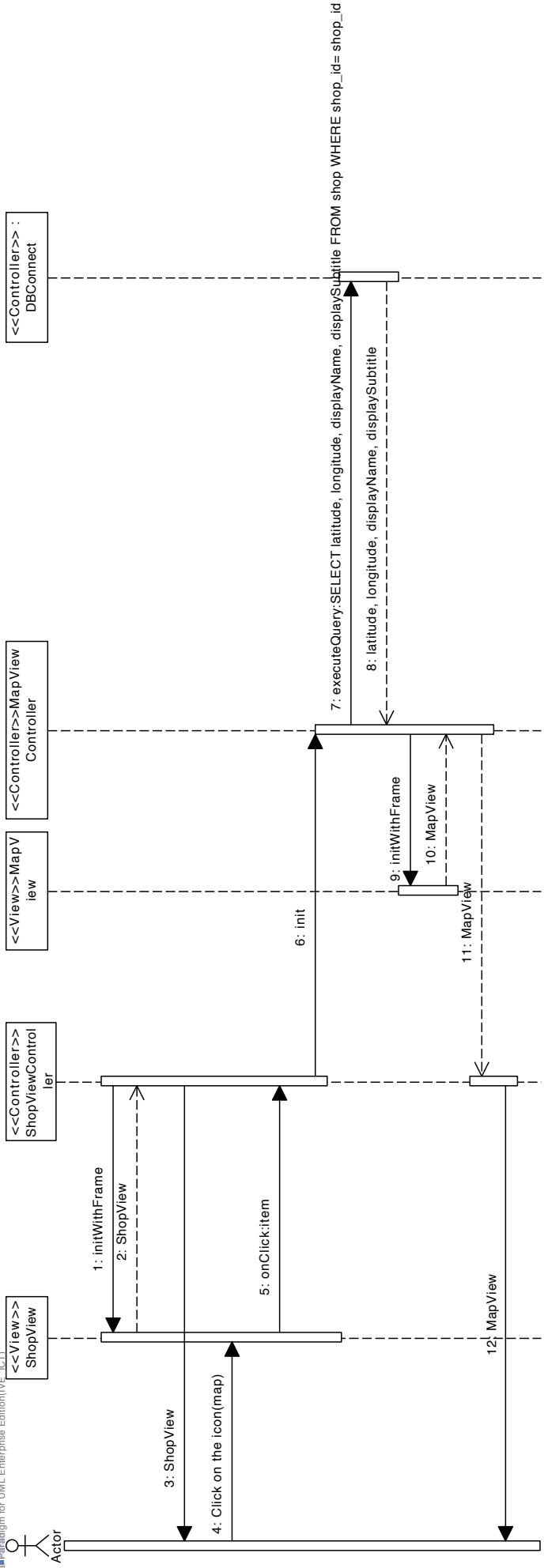

Actor
<<View>>MyItemView
<<Controller>>MyItemViewController
<<View>>DrawView
<<Controller>>DrawViewController
<<Controller>>getLuckydraw.php
ServerDatabase
1: Click "Draw"
2: drawNew
3: init
4: initWithFrame
5: DrawView
6: luckydraw_id, member_data
7: luckydraw_id, member_data
8: success message with the item
9: success message with the item
10: success message with the item
11: success message with the item
12: success message with the item
Visual Paradigm for UML Enterprise Edition(IVE_ICT)



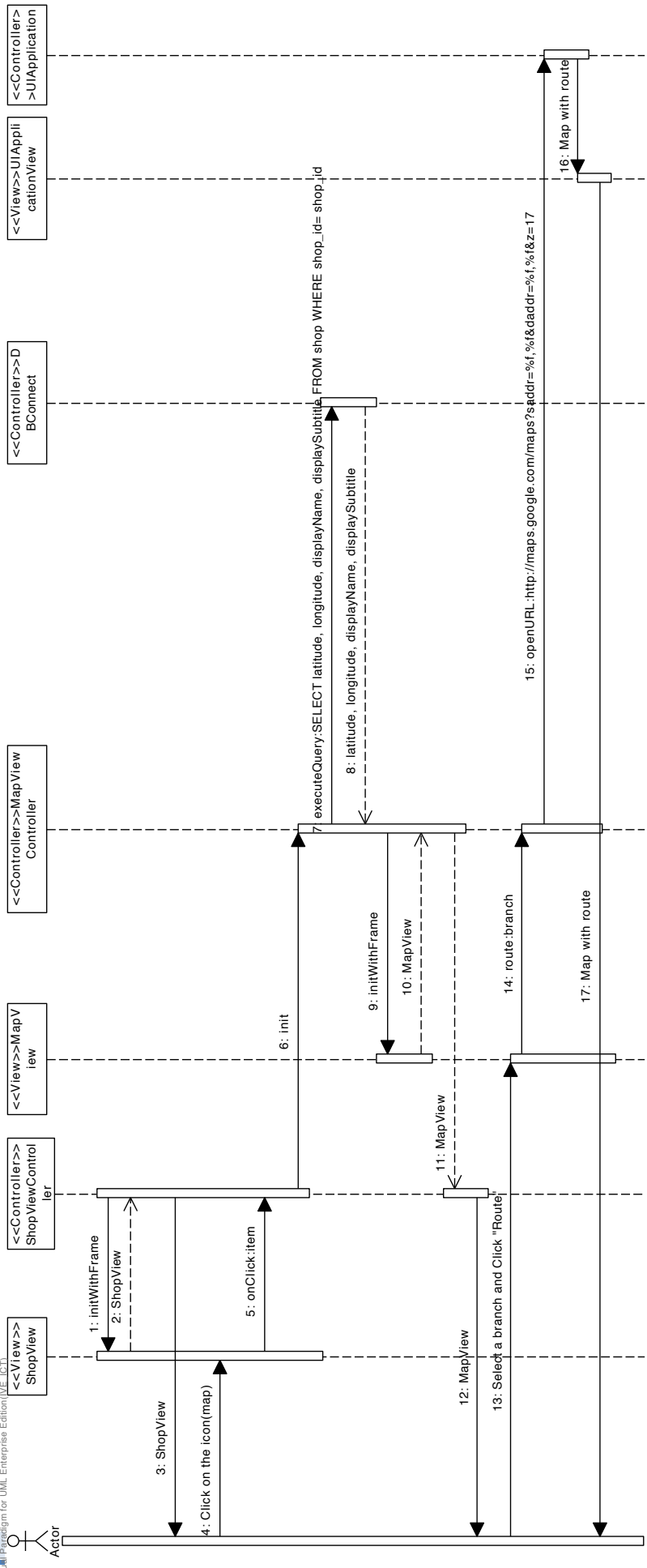
Visual Paradigm for UML Enterprise Edition(VE_ICT)



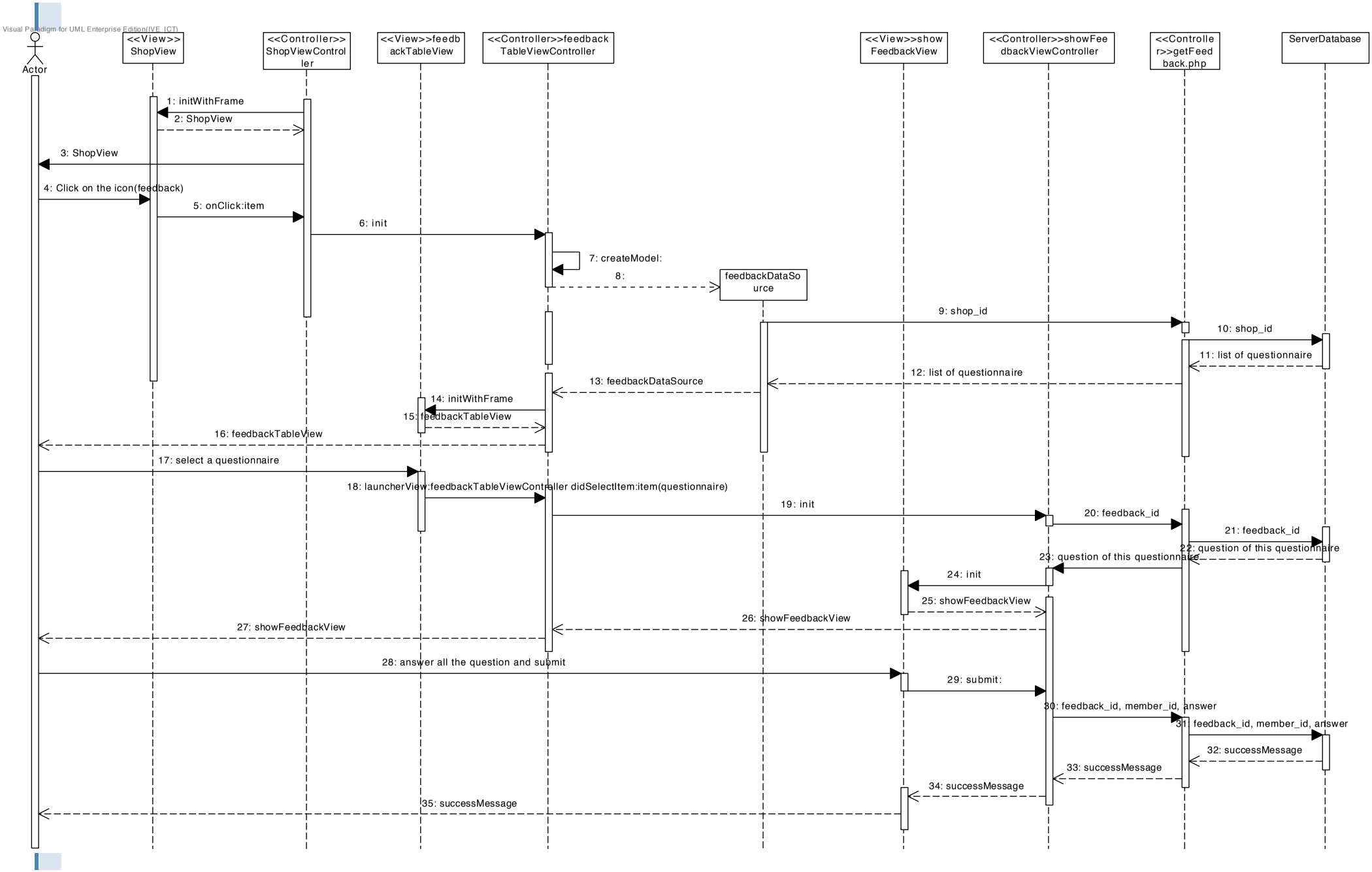
Visual Paradigm for UML Enterprise Edition(IVE, ICT)

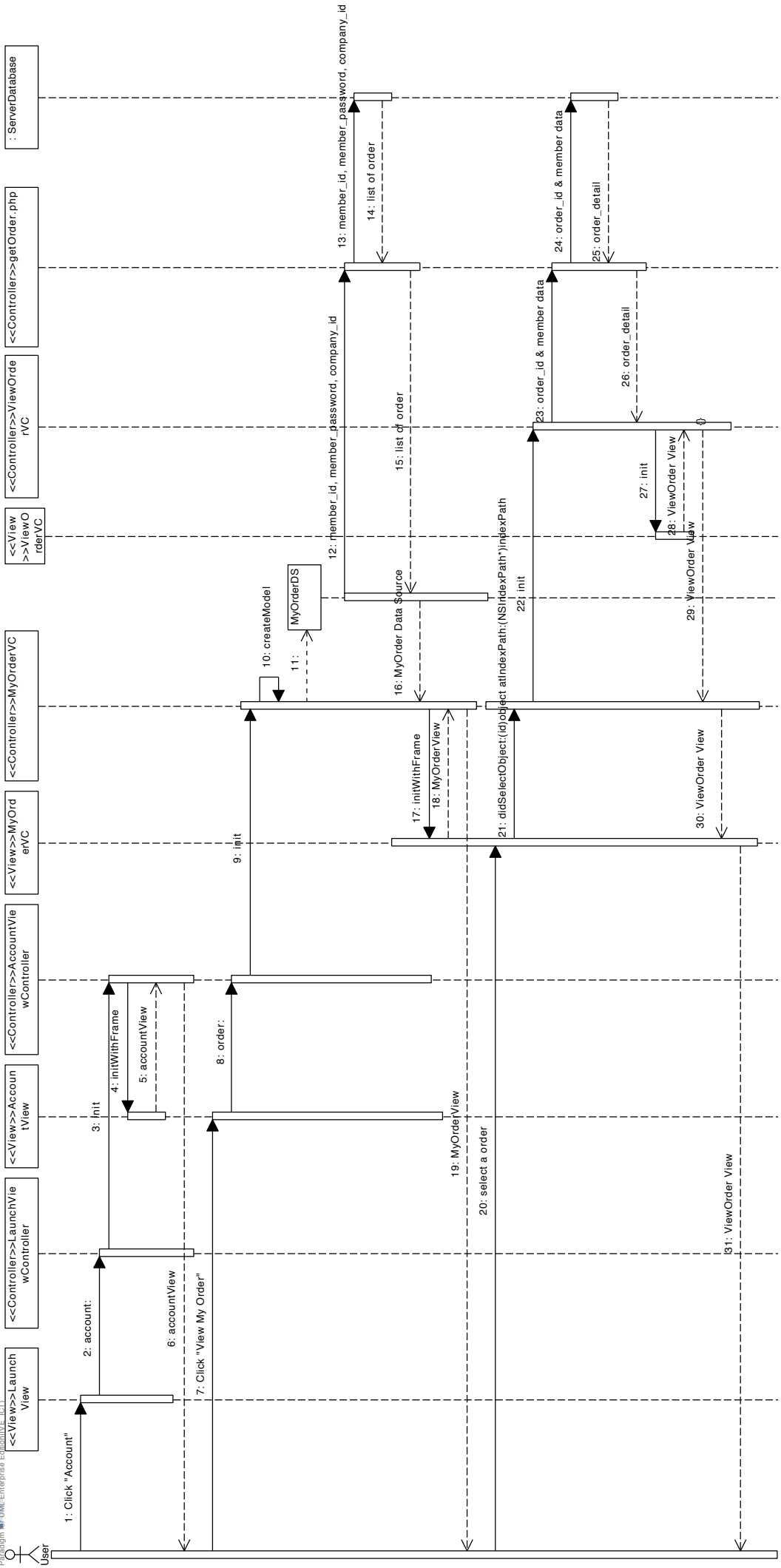


Visual Paradigm for UML Enterprise Edition(IVE_ICT)

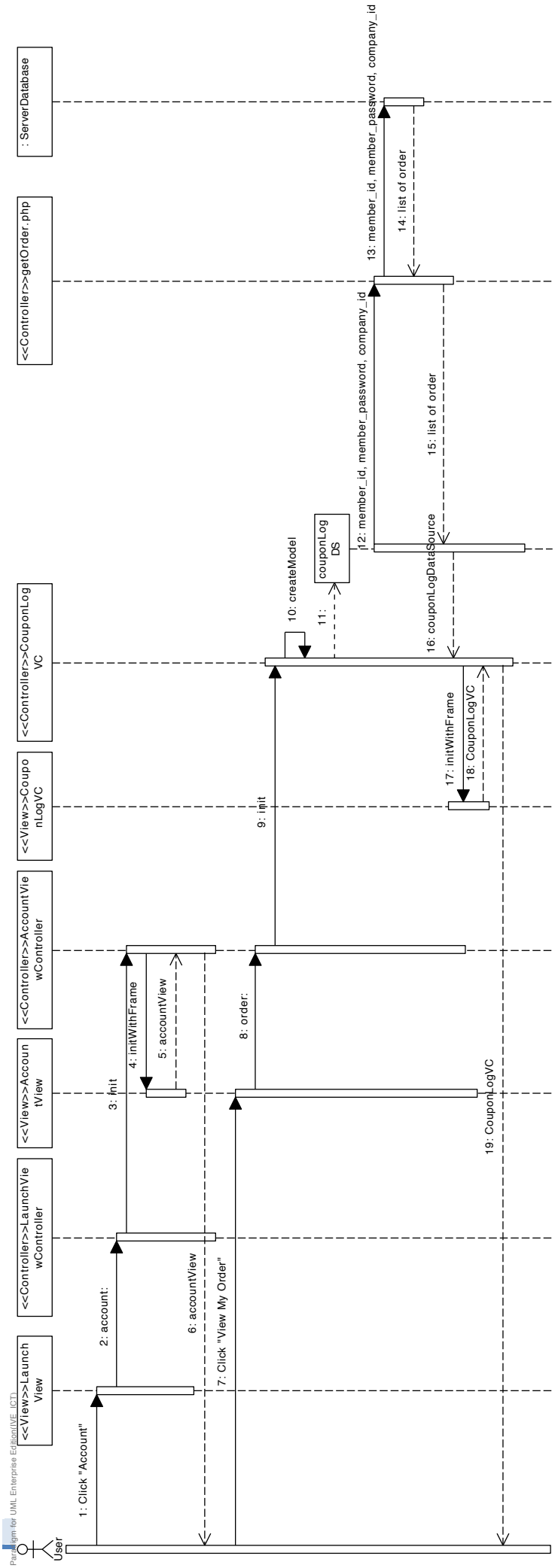


Visual Paradigm for UML Enterprise Edition(IVE_ICT)

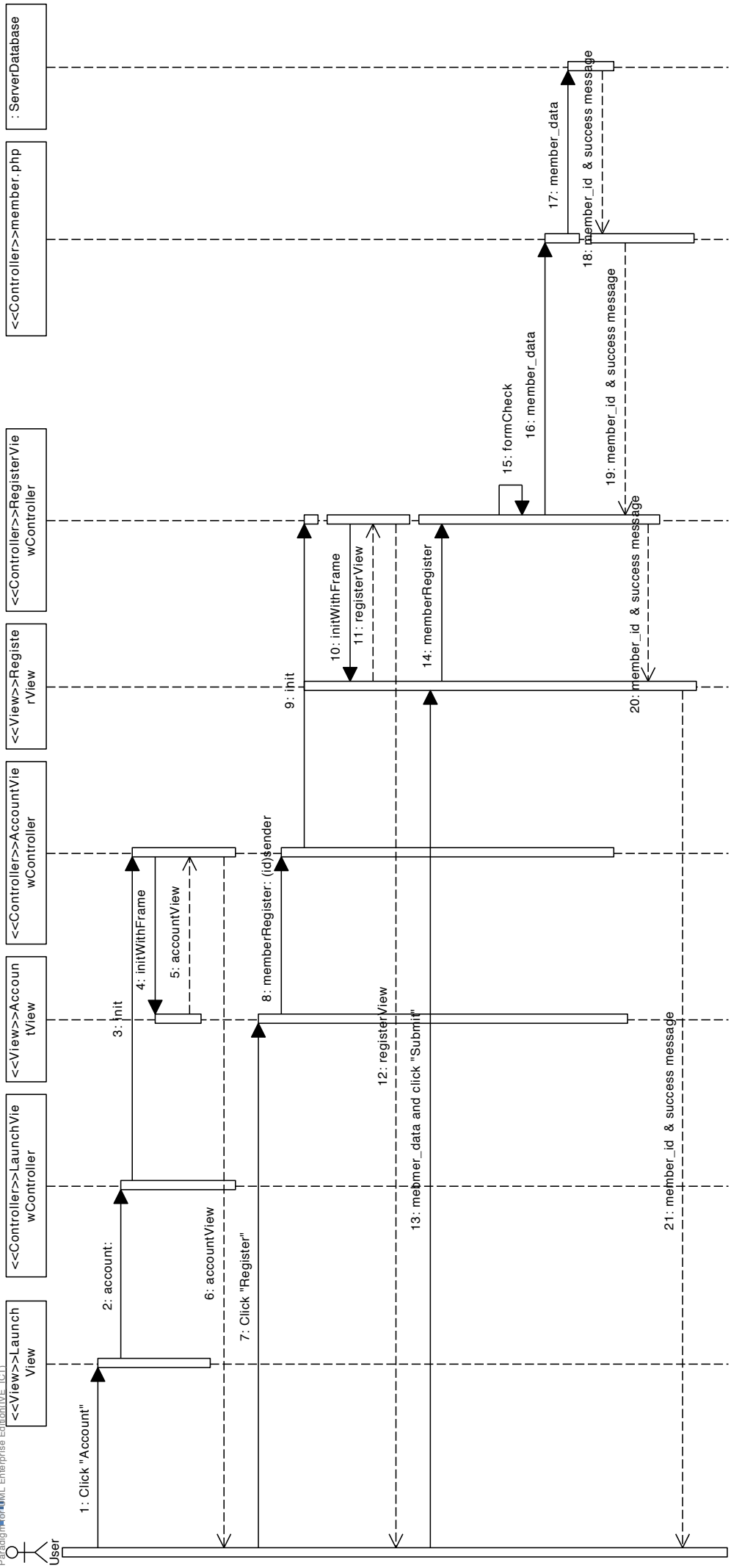

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Actor
<<View>>
ShopView
<<Controller>>
ShopViewControl
ler
<<View>>feedb
ackTableView
<<Controller>>feedback
TableViewController
<<View>>show
FeedbackView
<<Controller>>showFee
dbackViewController
<<Controlle
r>>getFeed
back.php
ServerDatabase
1: initWithFrame
2: ShopView
3: ShopView
4: Click on the icon(feedback)
5: onClick:item
6: init
7: createModel:
8:
feedbackDataSo
urce
9: shop_id
10: shop_id
11: list of questionnaire
12: list of questionnaire
13: feedbackDataSource
14: initWithFrame
15: feedbackTableView
16: feedbackTableView
17: select a questionnaire
18: launcherView:feedbackTableViewController didSelectItem:item(questionnaire)
19: init
20: feedback_id
21: feedback_id
22: question of this questionnaire
23: question of this questionnaire
24: init
25: showFeedbackView
26: showFeedbackView
27: showFeedbackView
28: answer all the question and submit
29: submit:
30: feedback_id, member_id, answer
31: feedback_id, member_id, answer
32: successMessage
33: successMessage
34: successMessage
35: successMessage



Visual Paradigm for UML Enterprise Edition(VE_ICT)

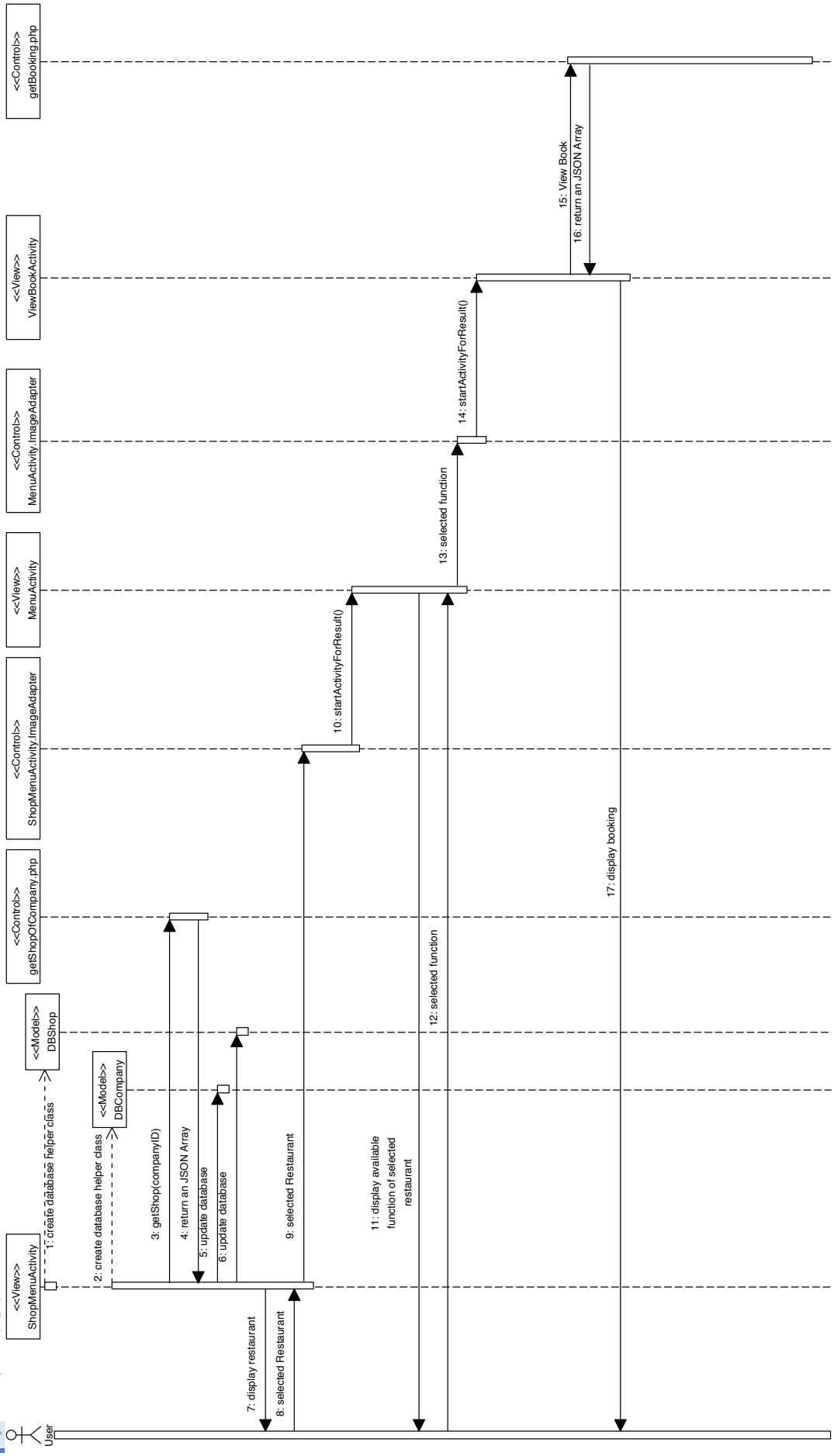


Visual Paradigm for UML Enterprise Edition(IVE_ICT)

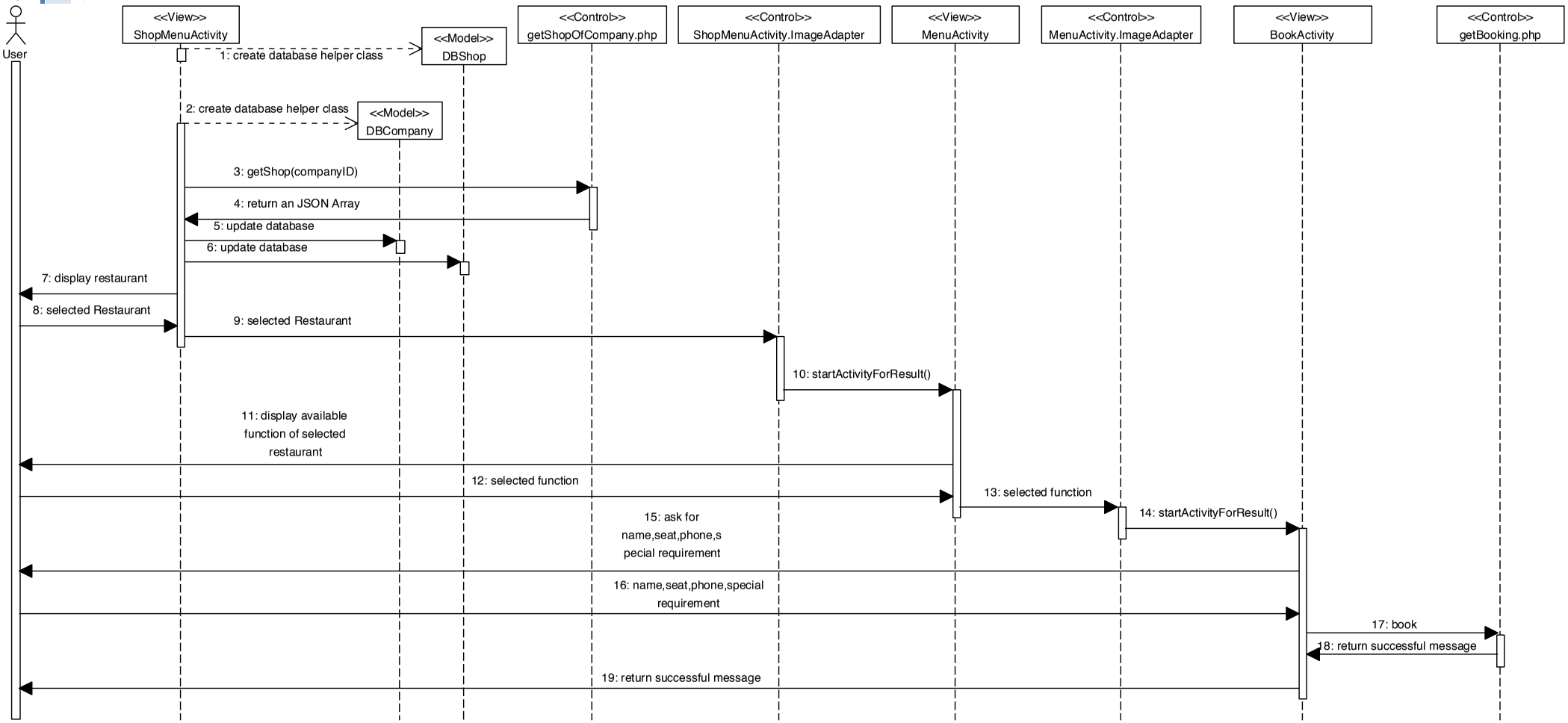


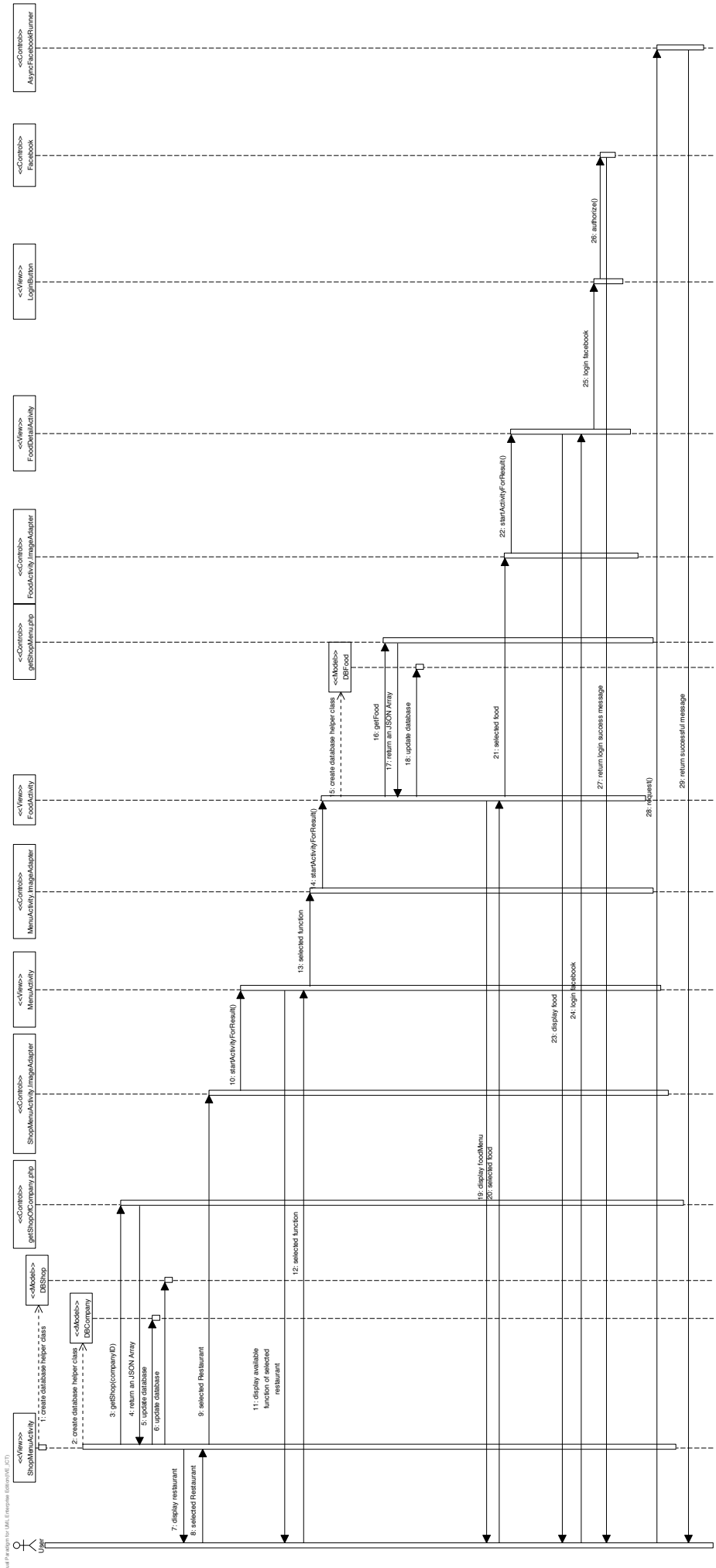
Visual Paradigm for UML Enterprise Edition(VE_ICD)

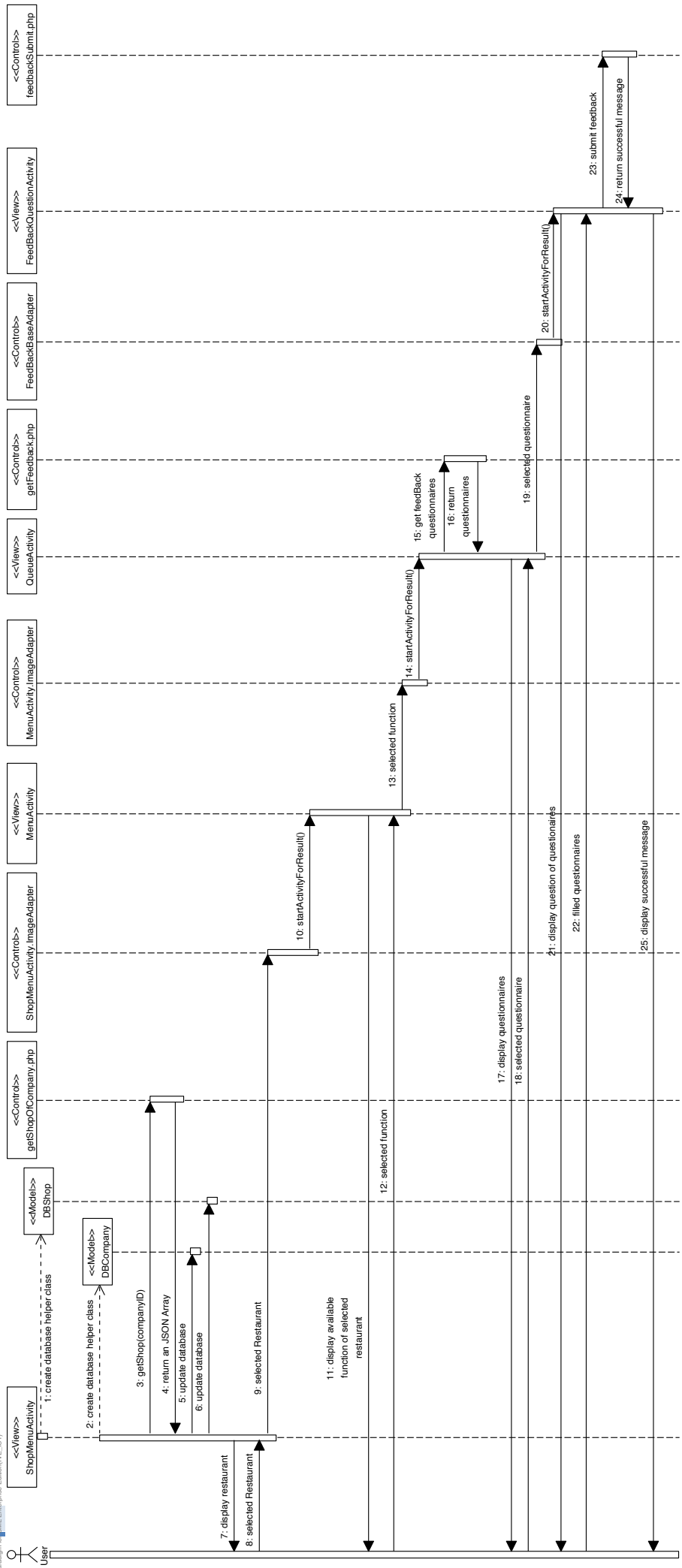
Sequence Diagram (Android)



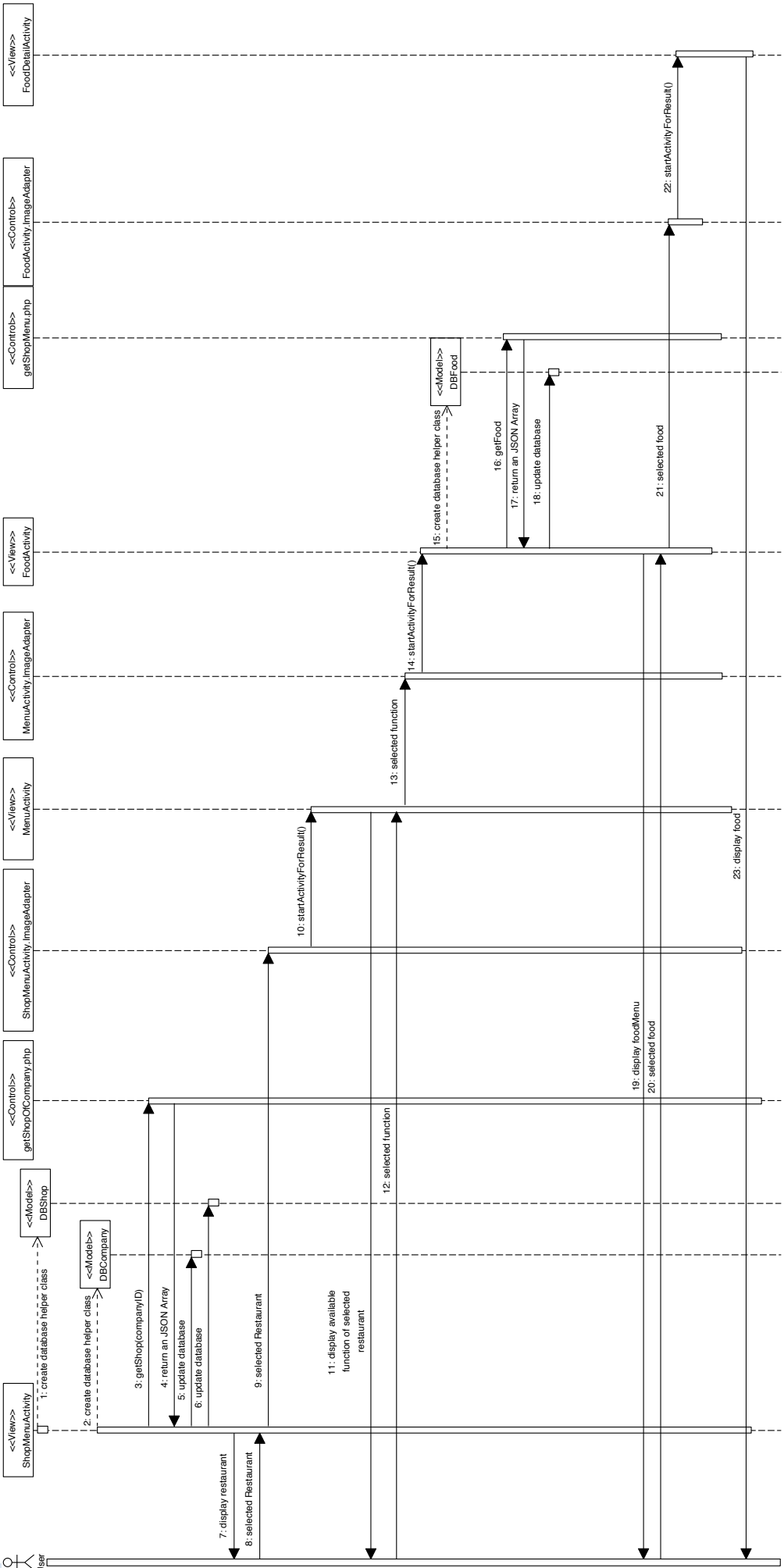
ual Paradigm for UML Enterprise Edition(IVE_ICT)



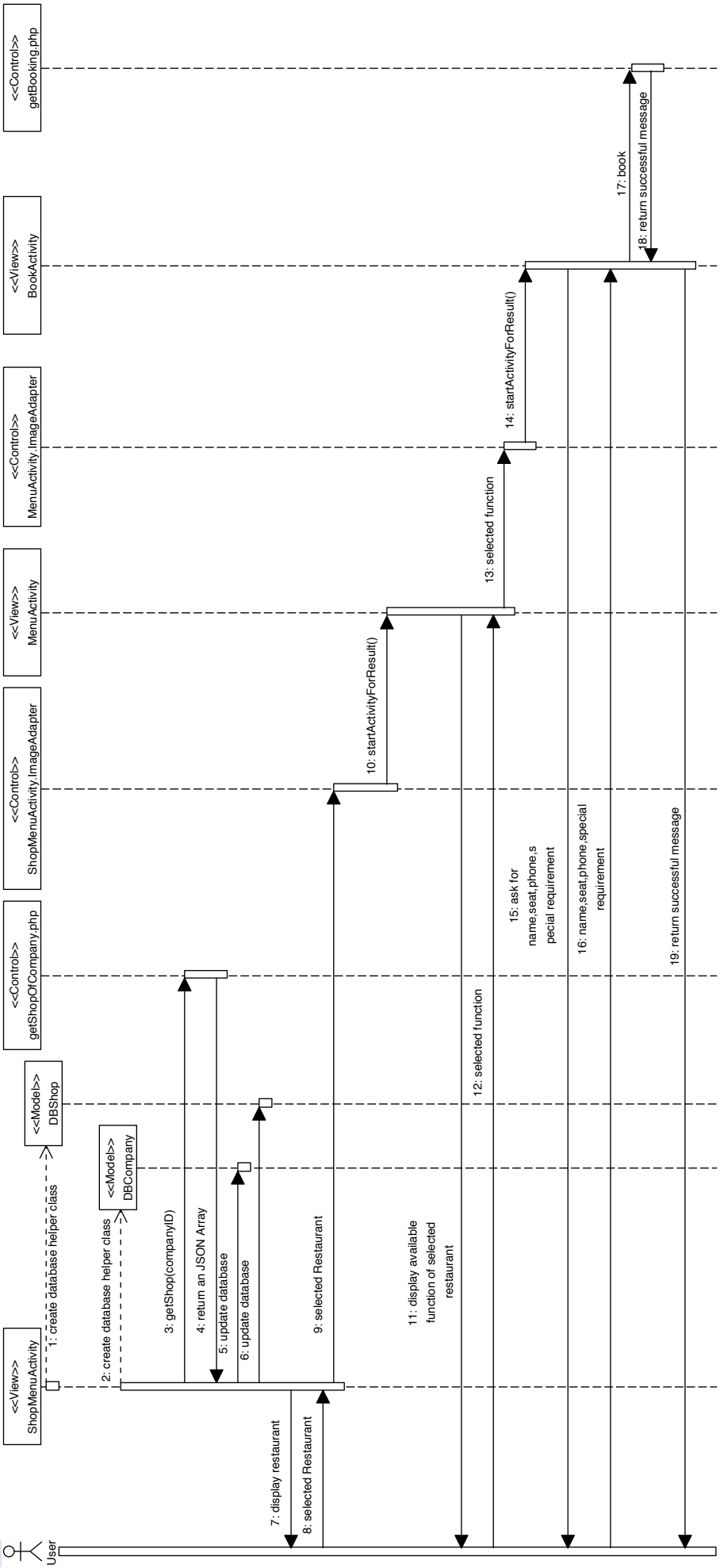

User
<<View>> ShopMenuActivity
<<Model>> DBShop
<<Model>> DBCompany
<<Controlo>> getShopOfCompany.php
<<Controlo>> ShopMenuActivityImageAdapter
<<View>> MenuActivity
<<Controlo>> MenuActivityImageAdapter
<<View>> FoodActivity
<<Model>> DBFood
<<Controlo>> getShopMenu.php
<<Controlo>> FoodActivityImageAdapter
<<View>> FoodDetailActivity
<<View>> LoginButton
<<Controlo>> Facebook
<<Controlo>> AsyncFacebookRunner
1: create database helper class
2: create database helper class
3: getShop(companyID)
4: return an JSON Array
5: update database
6: update database
7: display restaurant
8: selected Restaurant
9: selected Restaurant
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function
13: selected function
14: startActivityForResult()
15: create database helper class
16: getFood
17: return an JSON Array
18: update database
19: display foodMenu
20: selected food
21: selected food
22: startActivityForResult()
23: display food
24: login facebook
25: login facebook
26: authorize()
27: return login success message
28: request()
29: return successful message



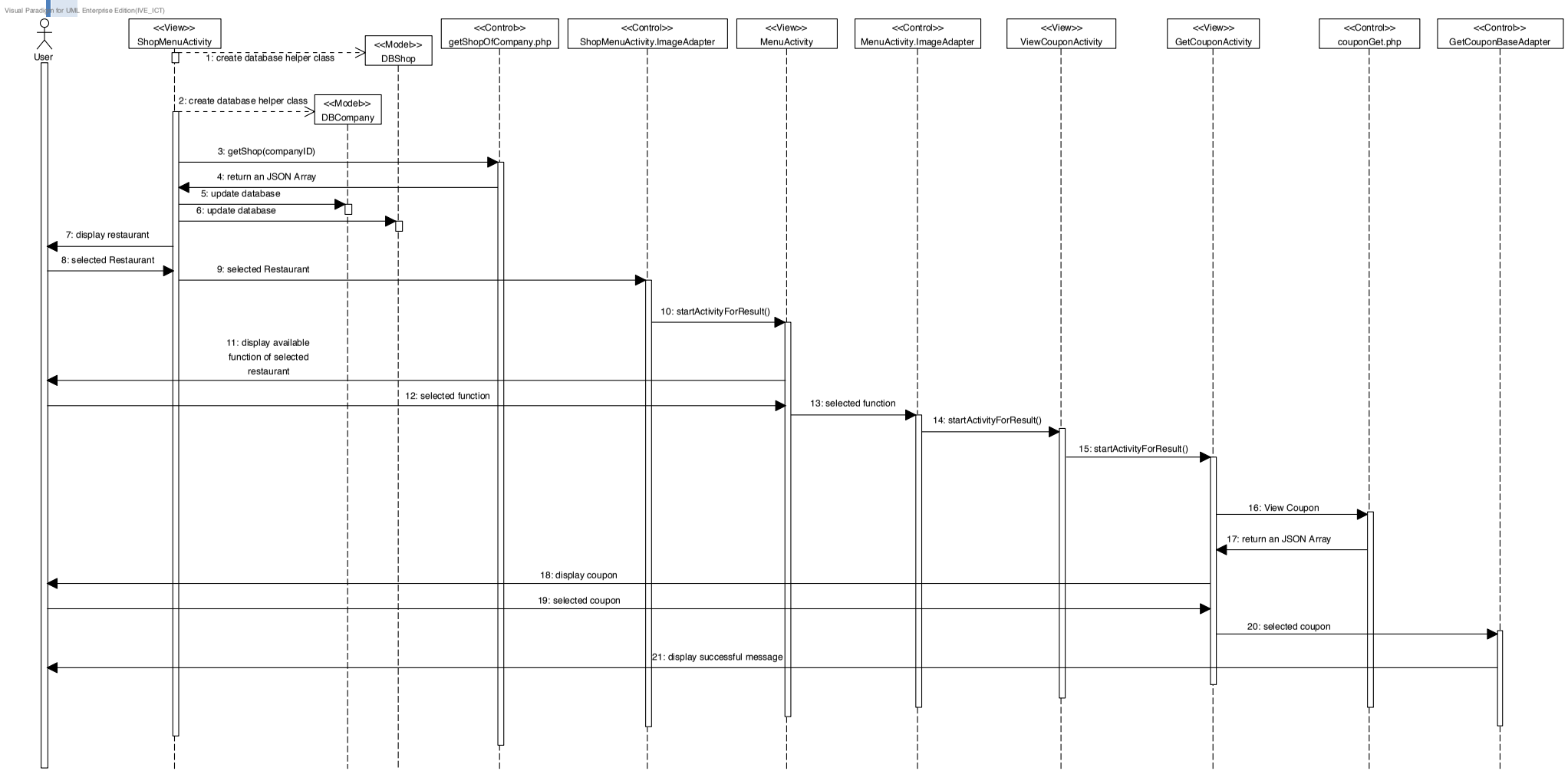
Visual Paradigm for UML Enterprise Edition(VE_ICT)

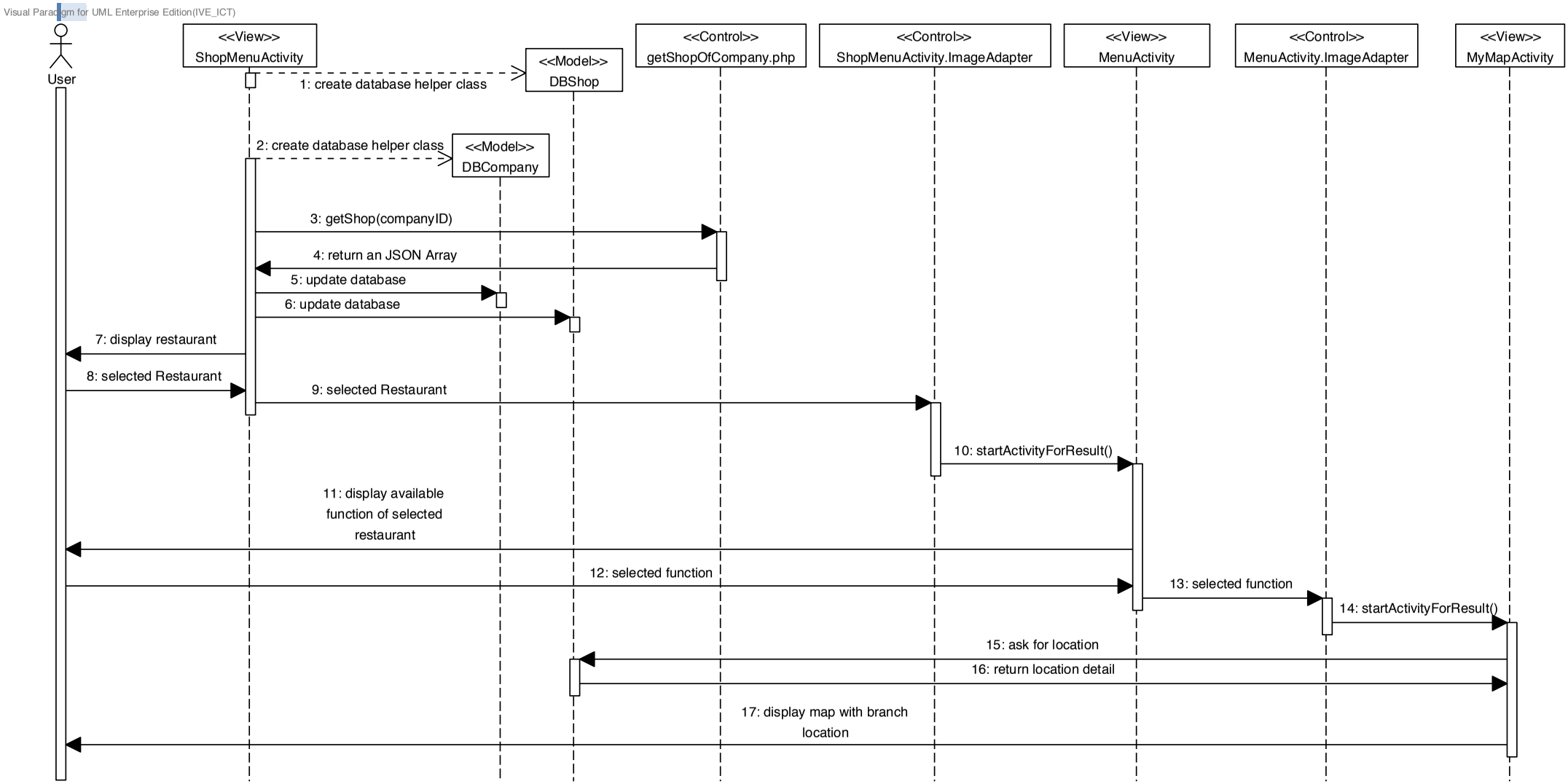


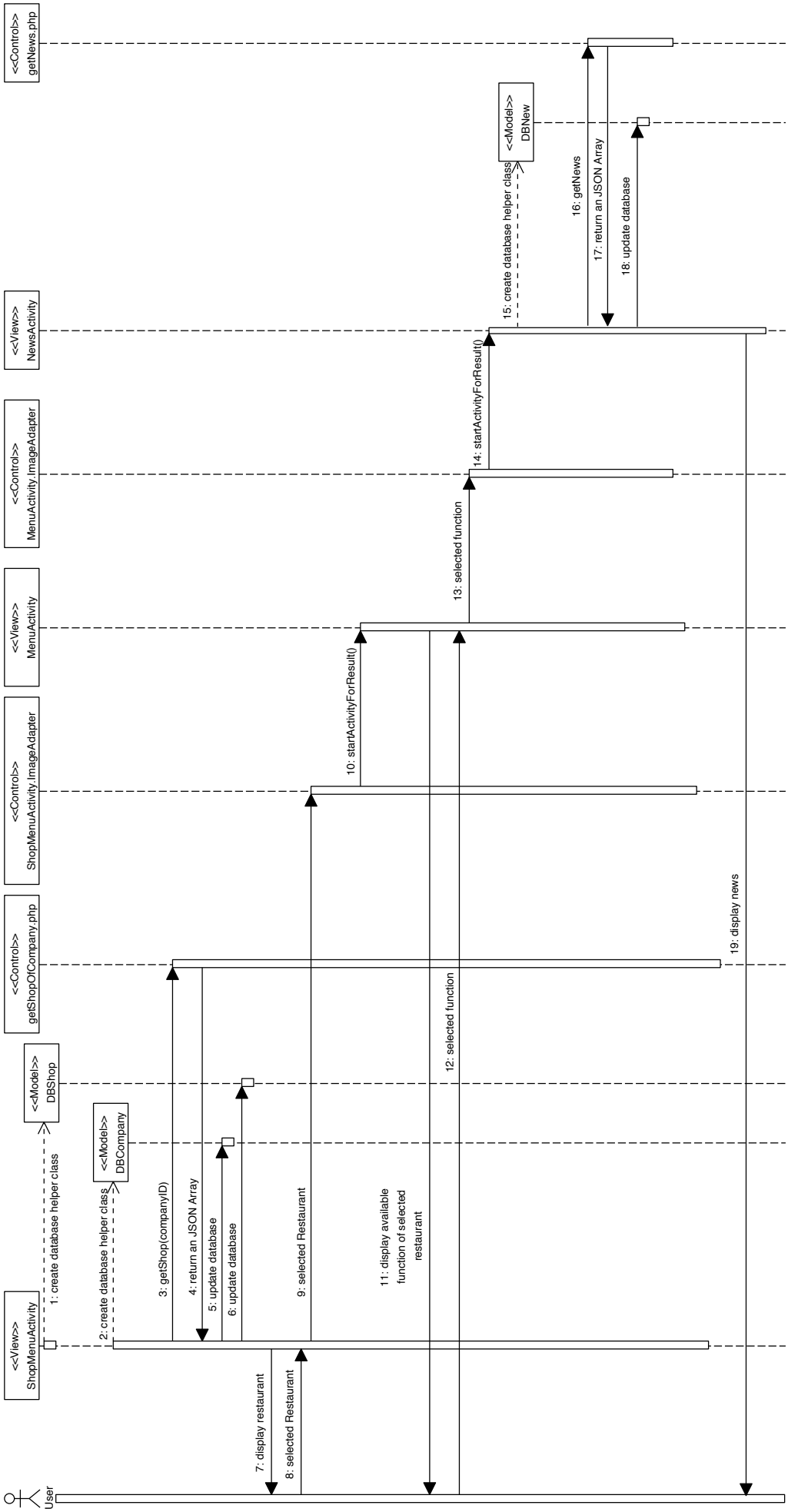
Visual Paradigm for UML Enterprise Edition(IVE, ICT)



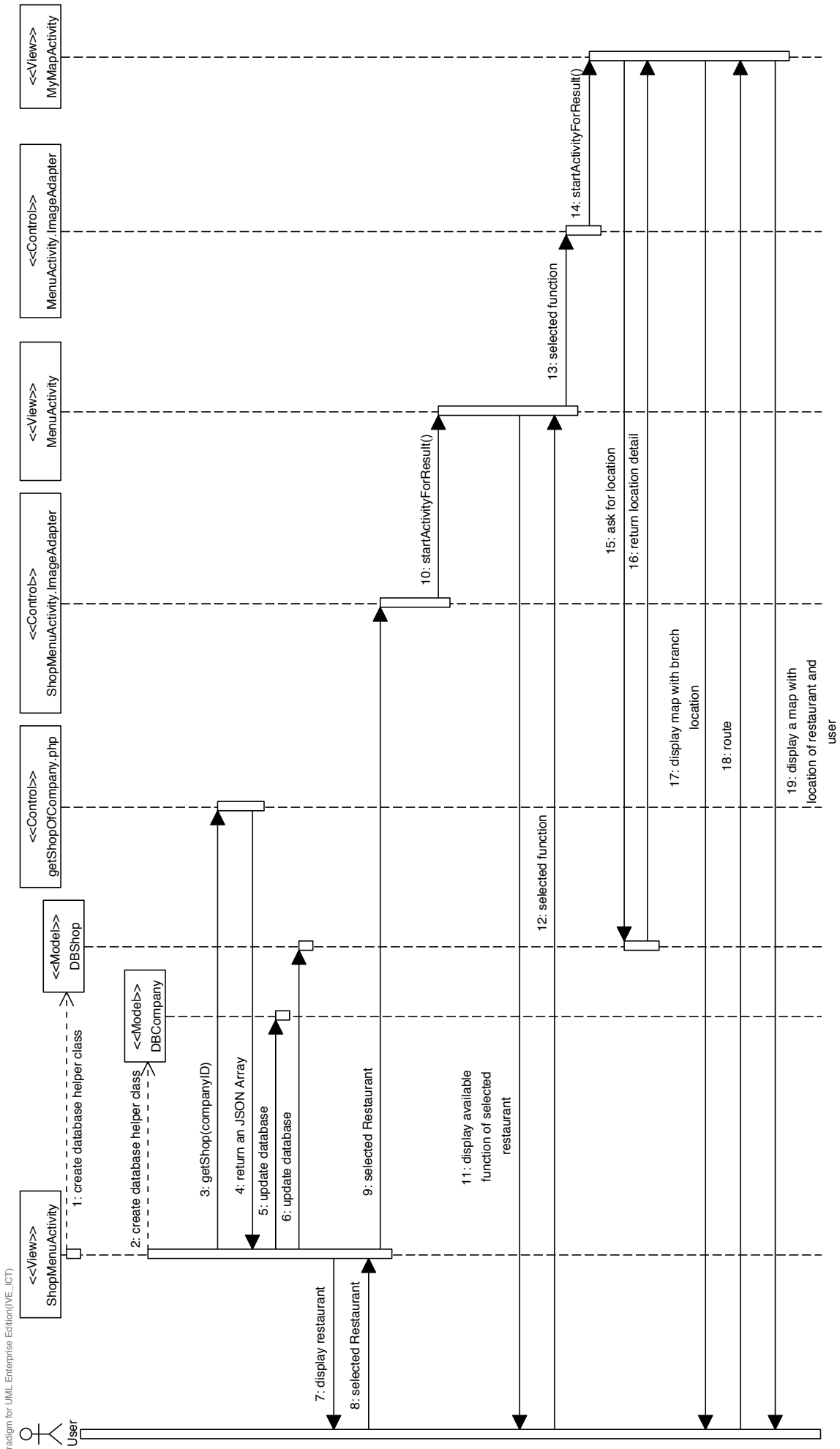
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

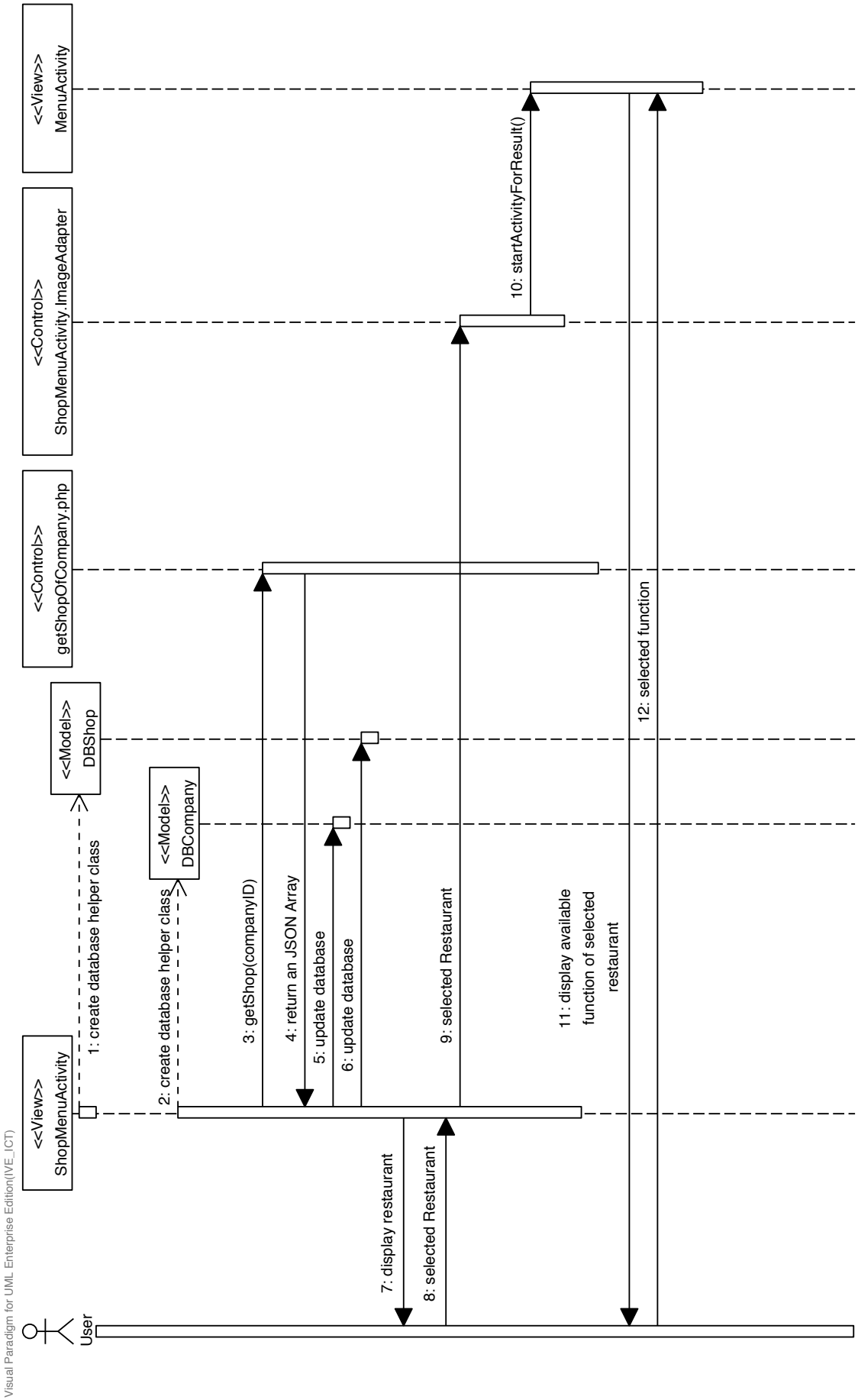

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
User
<<View>>
ShopMenuActivity
<<Model>>
DBShop
<<Control>>
getShopOfCompany.php
<<Control>>
ShopMenuActivity.ImageAdapter
<<View>>
MenuActivity
<<Control>>
MenuActivity.ImageAdapter
<<View>>
ViewCouponActivity
<<View>>
GetCouponActivity
<<Control>>
couponGet.php
<<Control>>
GetCouponBaseAdapter
1: create database helper class
2: create database helper class
<<Model>>
DBCompany
3: getShop(companyID)
4: return an JSON Array
5: update database
6: update database
7: display restaurant
8: selected Restaurant
9: selected Restaurant
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function
13: selected function
14: startActivityForResult()
15: startActivityForResult()
16: View Coupon
17: return an JSON Array
18: display coupon
19: selected coupon
20: selected coupon
21: display successful message


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
User
<<View>>
ShopMenuActivity
<<Model>>
DBShop
<<Control>>
getShopOfCompany.php
<<Control>>
ShopMenuActivity.ImageAdapter
<<View>>
MenuActivity
<<Control>>
MenuActivity.ImageAdapter
<<View>>
MyMapActivity
1: create database helper class
<<Model>>
DBCompany
2: create database helper class
3: getShop(companyID)
4: return an JSON Array
5: update database
6: update database
7: display restaurant
8: selected Restaurant
9: selected Restaurant
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function
13: selected function
14: startActivityForResult()
15: ask for location
16: return location detail
17: display map with branch location

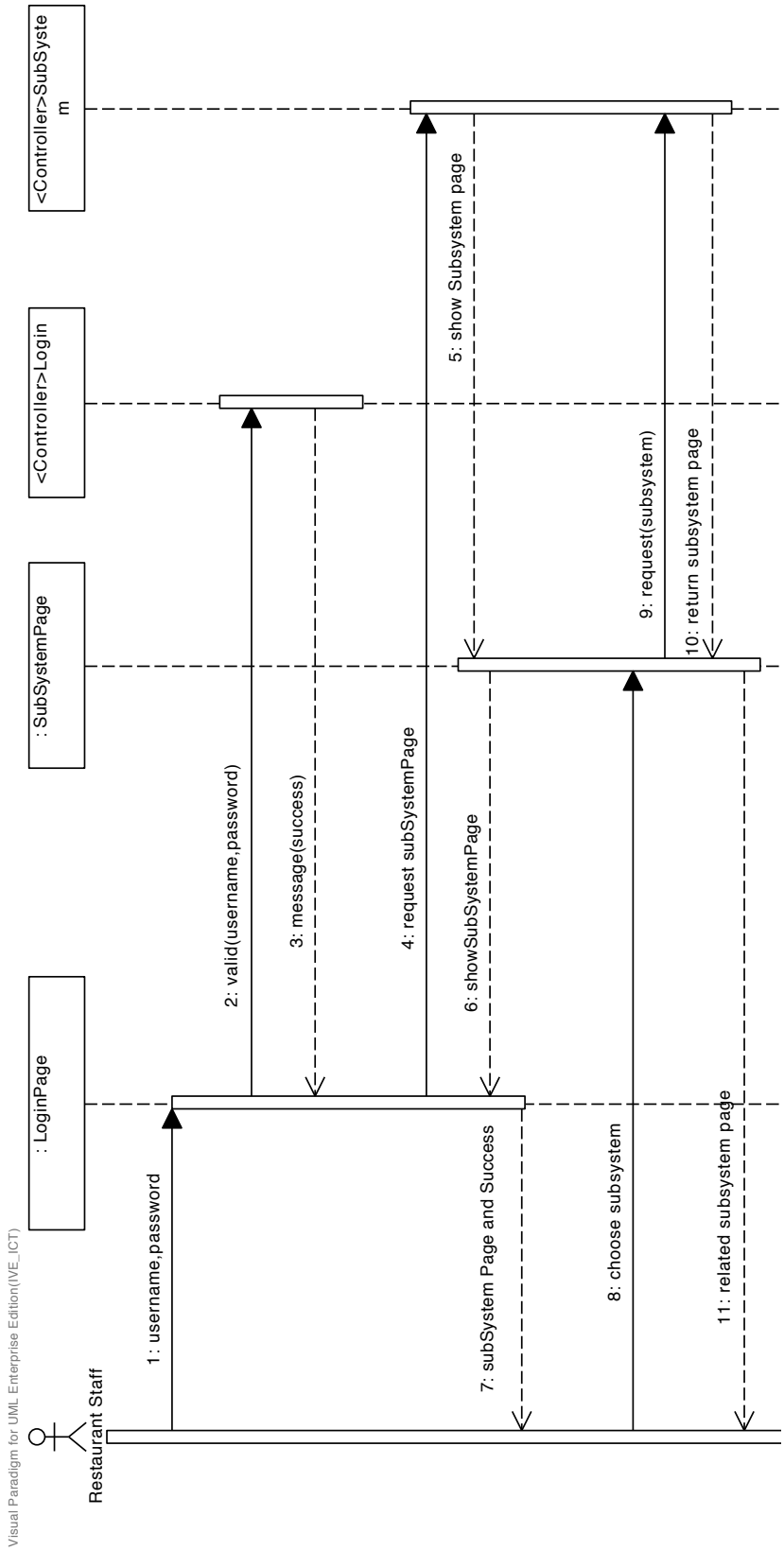


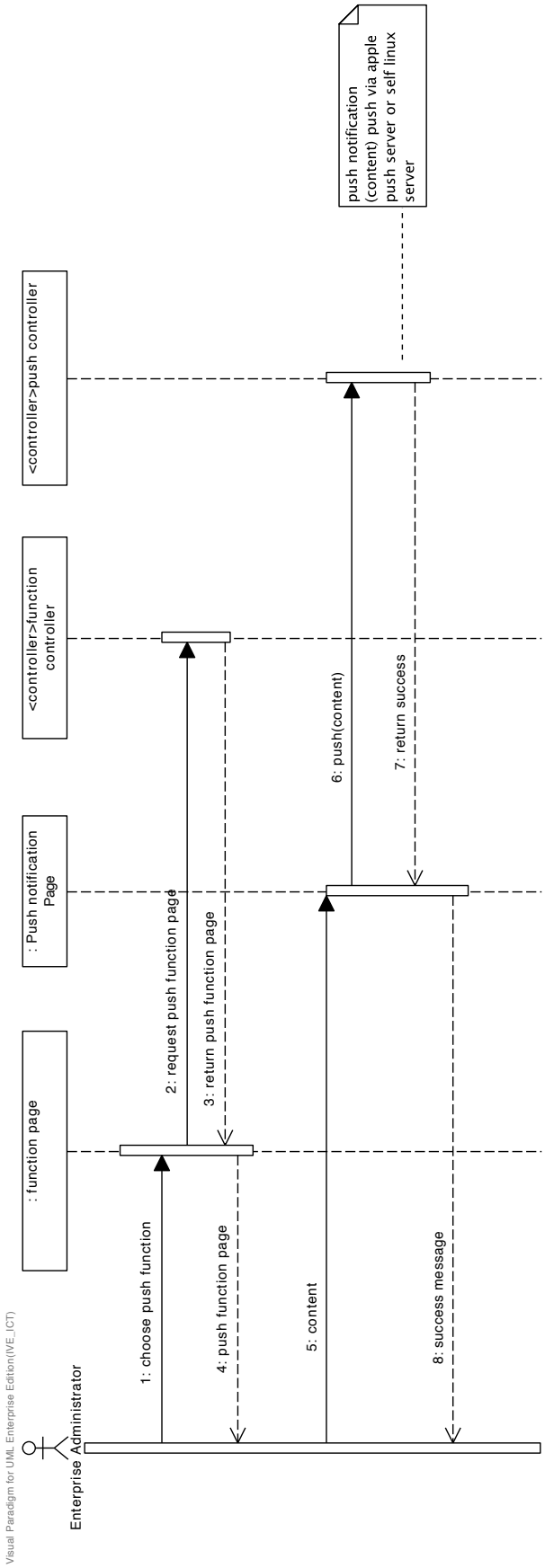
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

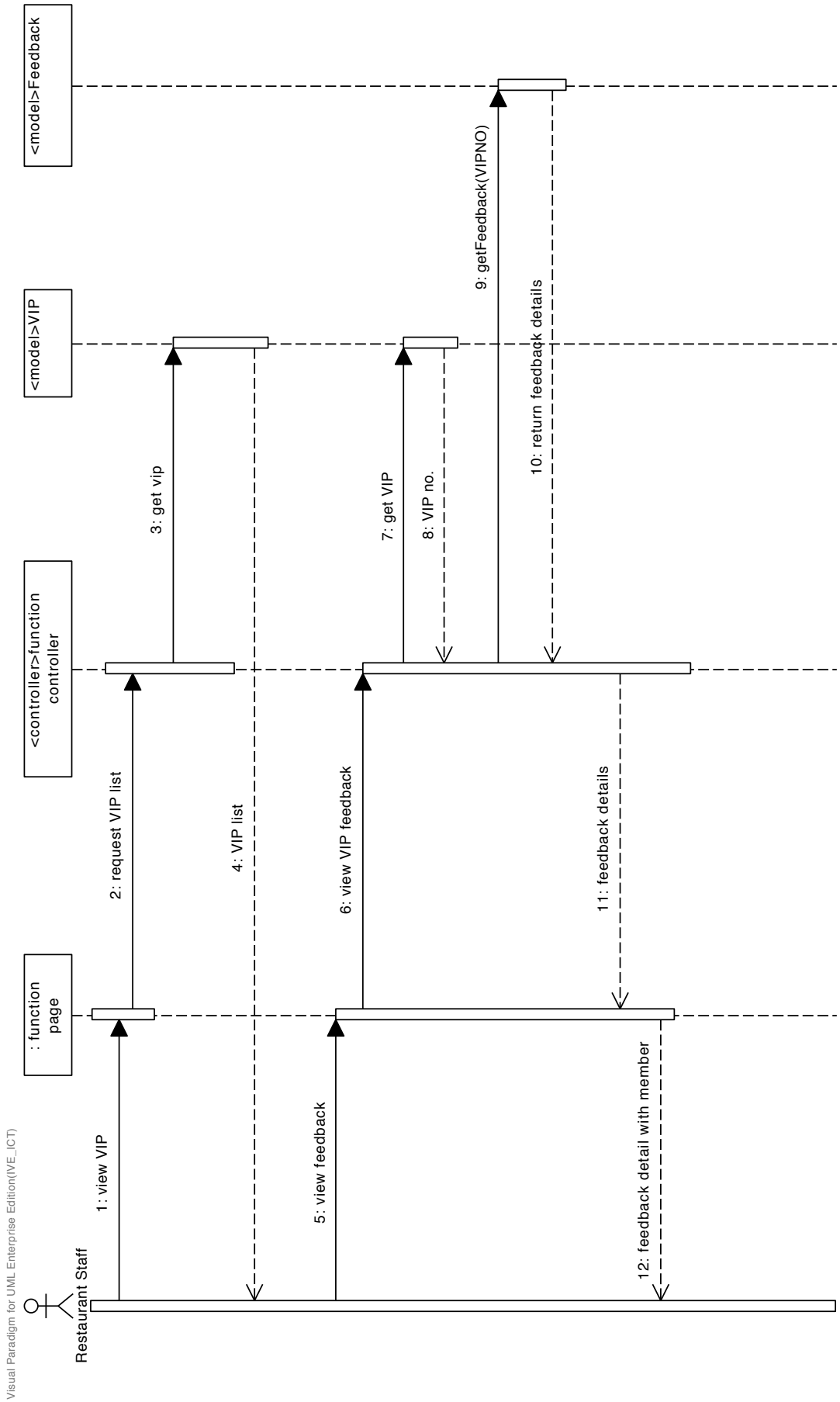

User
<<View>>
ShopMenuActivity
<<Model>>
DBShop
<<Model>>
DBCompany
<<Control>>
getShopOfCompany.php
<<Control>>
ShopMenuActivity.ImageAdapter
<<View>>
MenuActivity
<<Control>>
MenuActivity.ImageAdapter
<<View>>
MyMapActivity
1: create database helper class
2: create database helper class
3: getShop(companyID)
4: return an JSON Array
5: update database
6: update database
7: display restaurant
8: selected Restaurant
9: selected Restaurant
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function
13: selected function
14: startActivityForResult()
15: ask for location
16: return location detail
17: display map with branch location
18: route
19: display a map with location of restaurant and user
isual Paradigm for UML Enterprise Edition(IVE_ICT)

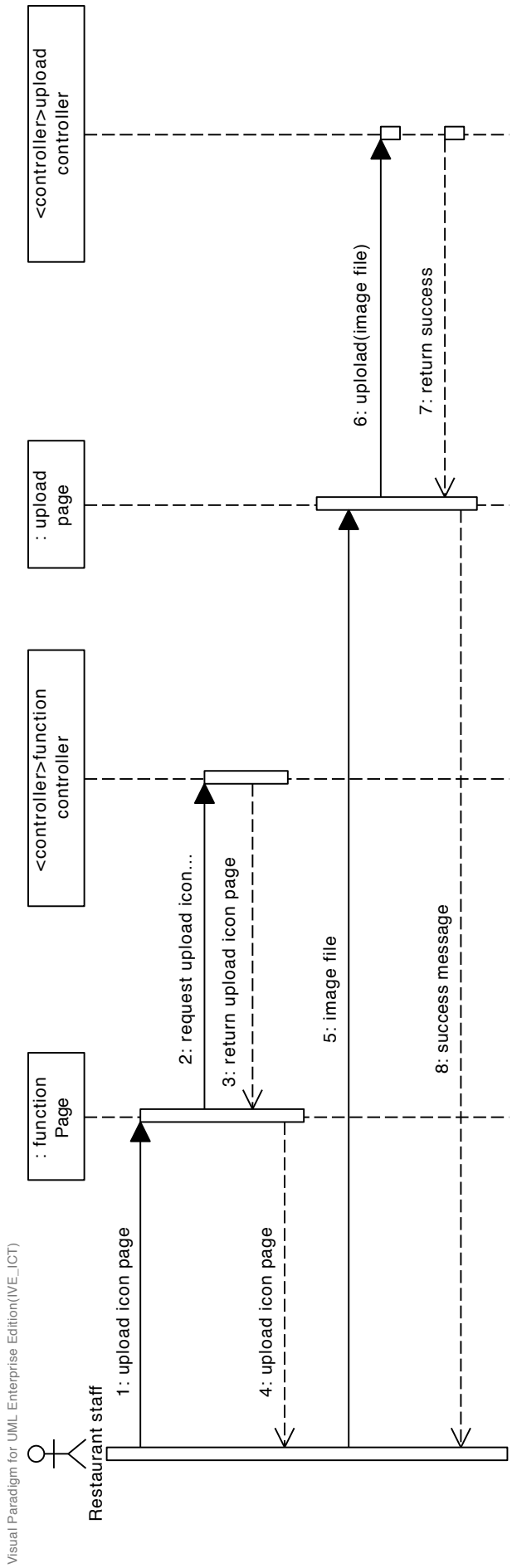

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
User
<<View>>
ShopMenuActivity
<<Model>>
DBShop
<<Control>>
getShopOfCompany.php
<<Control>>
ShopMenuActivity.ImageAdapter
<<View>>
MenuActivity
<<Model>>
DBCompany
1: create database helper class
2: create database helper class
3: getShop(companyID)
4: return an JSON Array
5: update database
6: update database
7: display restaurant
8: selected Restaurant
9: selected Restaurant
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function

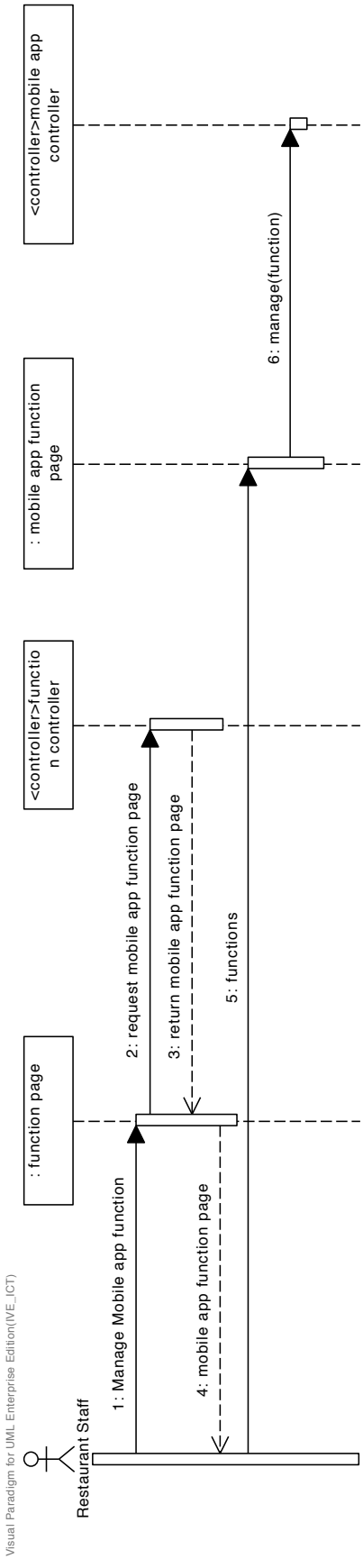
Sequence Diagram (Website)

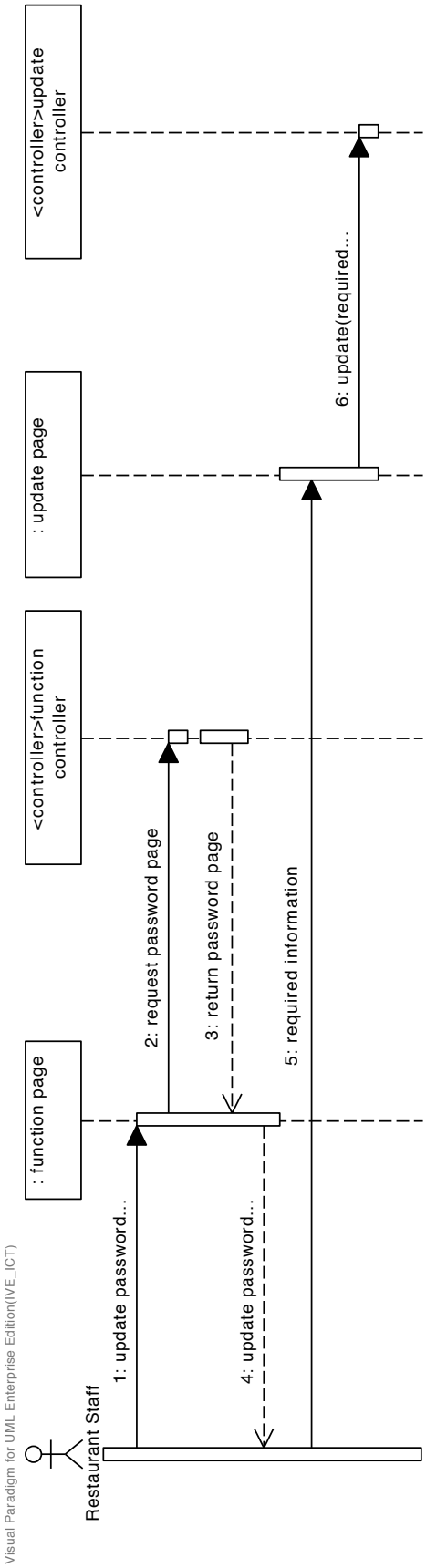


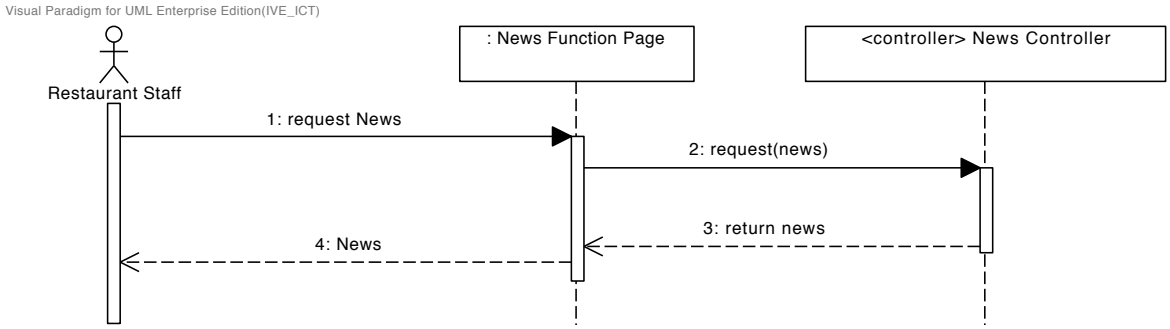

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Enterprise Administrator
: function page
: Push notification Page
<controller>function controller
<controller>push controller
push notification (content) push via apple push server or self linux server
1: choose push function
2: request push function page
3: return push function page
4: push function page
5: content
6: push(content)
7: return success
8: success message

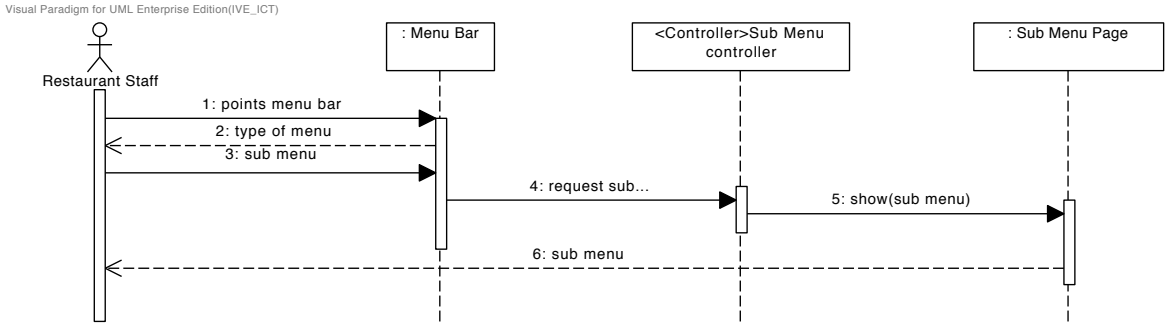

Restaurant Staff
: function page
<controller>function controller
<model>VIP
<model>Feedback
1: view VIP
2: request VIP list
3: get vip
4: VIP list
5: view feedback
6: view VIP feedback
7: get VIP
8: VIP no.
9: getFeedback(VIPNO)
10: return feedback details
11: feedback details
12: feedback detail with member
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

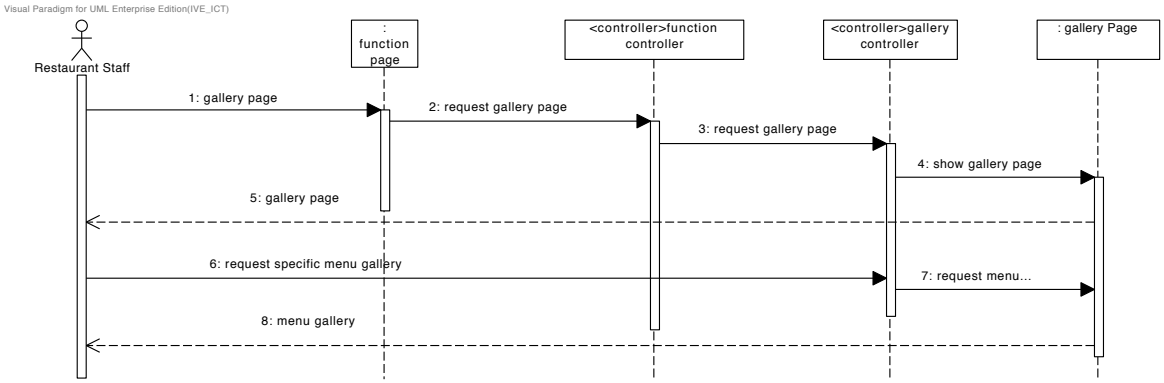

Restaurant staff
: function Page
<controller>function controller
: upload page
<controller>upload controller
1: upload icon page
2: request upload icon...
3: return upload icon page
4: upload icon page
5: image file
6: uplolad(image file)
7: return success
8: success message
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

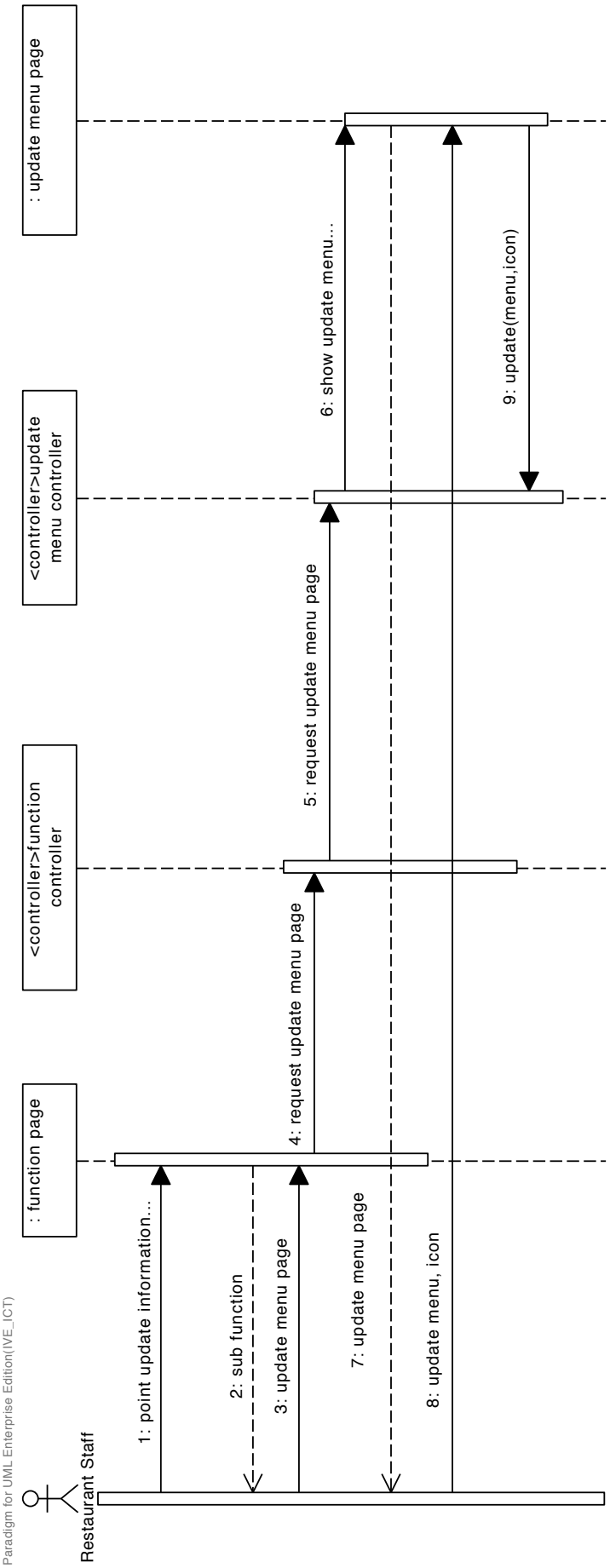

Restaurant Staff
: function page
<controller>functio
n controller
: mobile app function
page
<controller>mobile app
controller
1: Manage Mobile app function
2: request mobile app function page
3: return mobile app function page
4: mobile app function page
5: functions
6: manage(function)
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

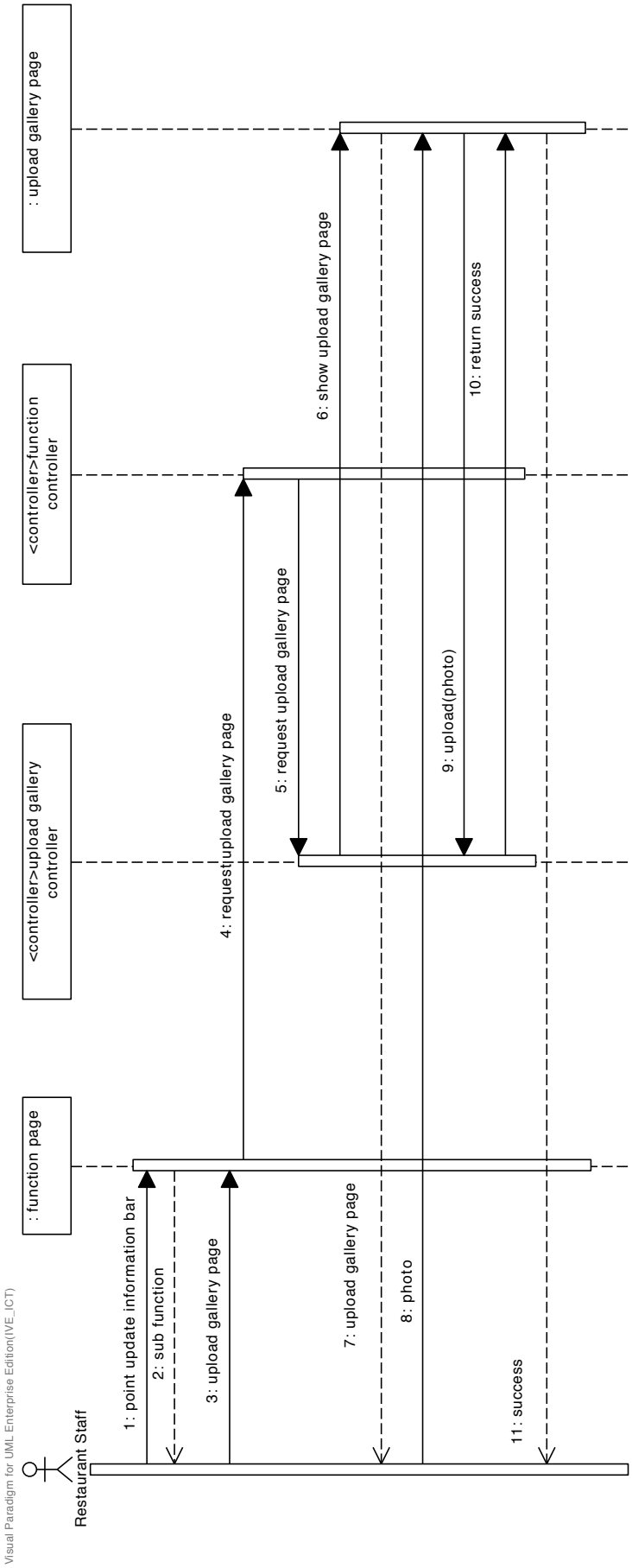

Restaurant Staff
: function page
<controller>function
controller
: update page
<controller>update
controller
1: update password...
2: request password page
3: return password page
4: update password...
5: required information
6: update(required...
Visual Paradigm for UML Enterprise Edition(IVE_ICT)


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Restaurant Staff
: News Function Page
<controller> News Controller
1: request News
2: request(news)
3: return news
4: News

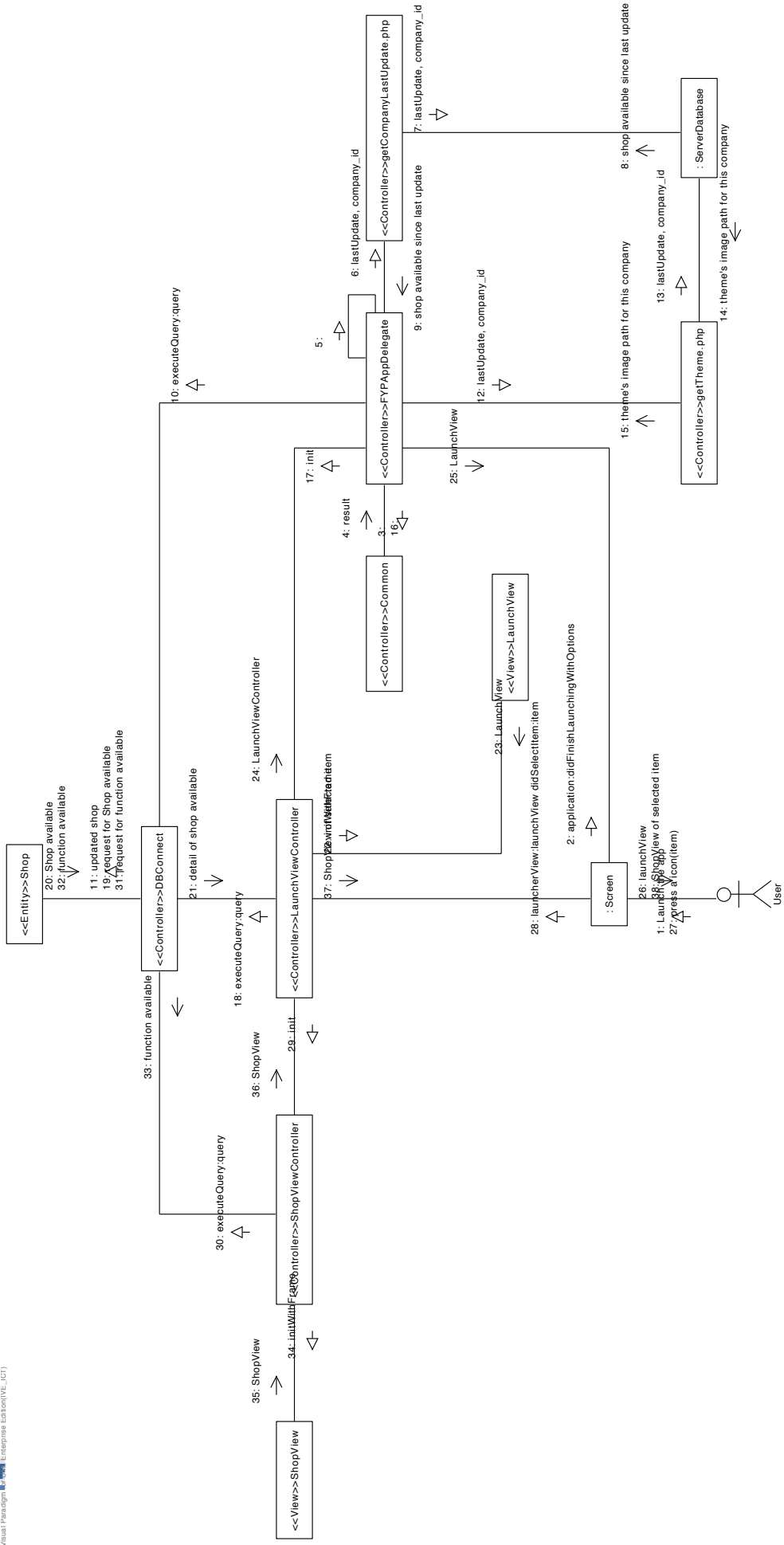

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Restaurant Staff
: Menu Bar
<Controller>Sub Menu controller
: Sub Menu Page
1: points menu bar
2: type of menu
3: sub menu
4: request sub...
5: show(sub menu)
6: sub menu


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Restaurant Staff
: function page
<controller>function controller
<controller>gallery controller
: gallery Page
1: gallery page
2: request gallery page
3: request gallery page
4: show gallery page
5: gallery page
6: request specific menu gallery
7: request menu...
8: menu gallery


Restaurant Staff
: function page
<controller>function controller
<controller>update menu controller
: update menu page
1: point update information...
2: sub function
3: update menu page
4: request update menu page
5: request update menu page
6: show update menu...
7: update menu page
8: update menu, icon
9: update(menu,icon)
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

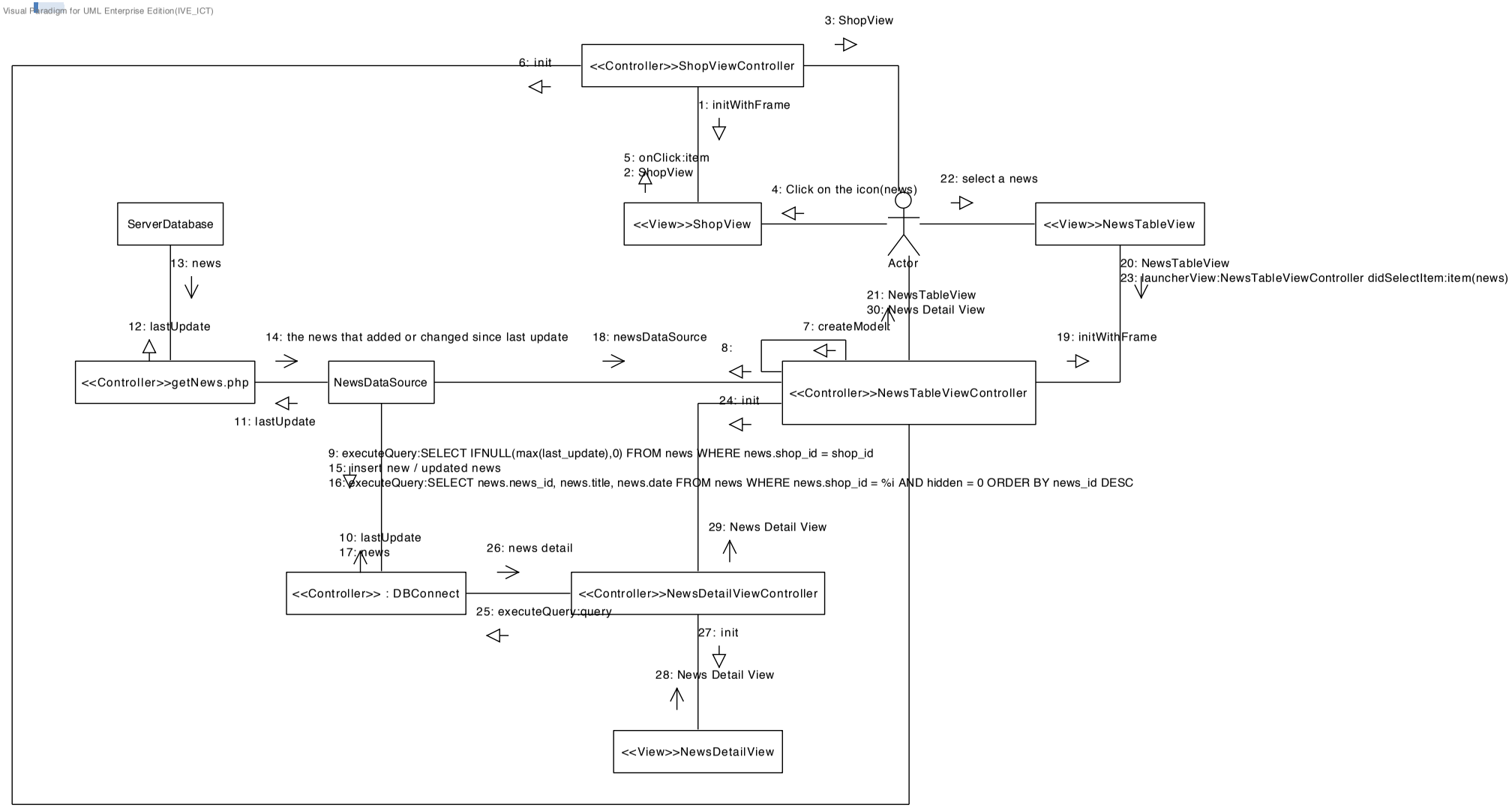

Restaurant Staff
: function page
<controller>upload gallery controller
<controller>function controller
: upload gallery page
1: point update information bar
2: sub function
3: upload gallery page
4: request upload gallery page
5: request upload gallery page
6: show upload gallery page
7: upload gallery page
8: photo
9: upload(photo)
10: return success
11: success
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

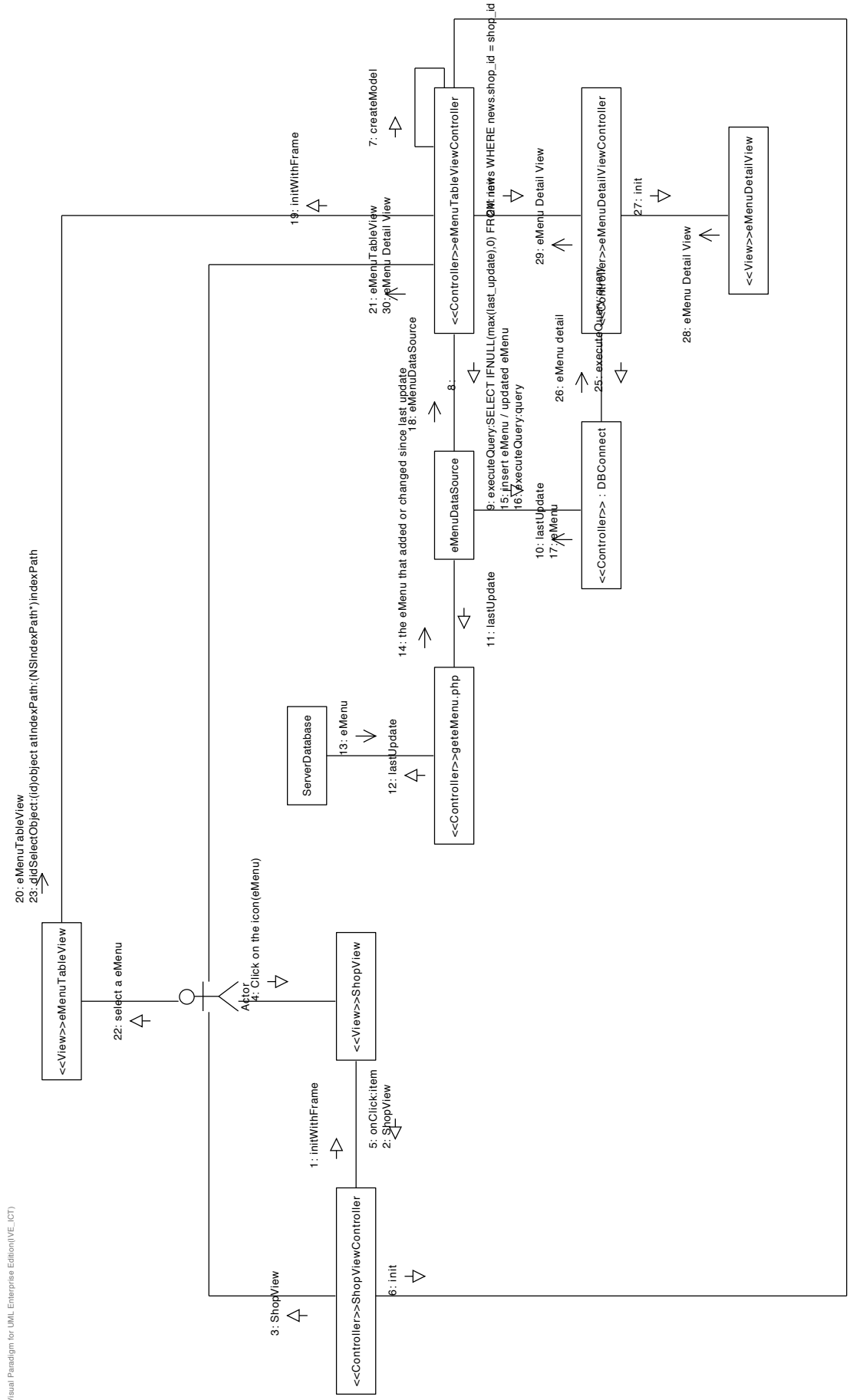
Object Collaboration Diagram (iPhone)

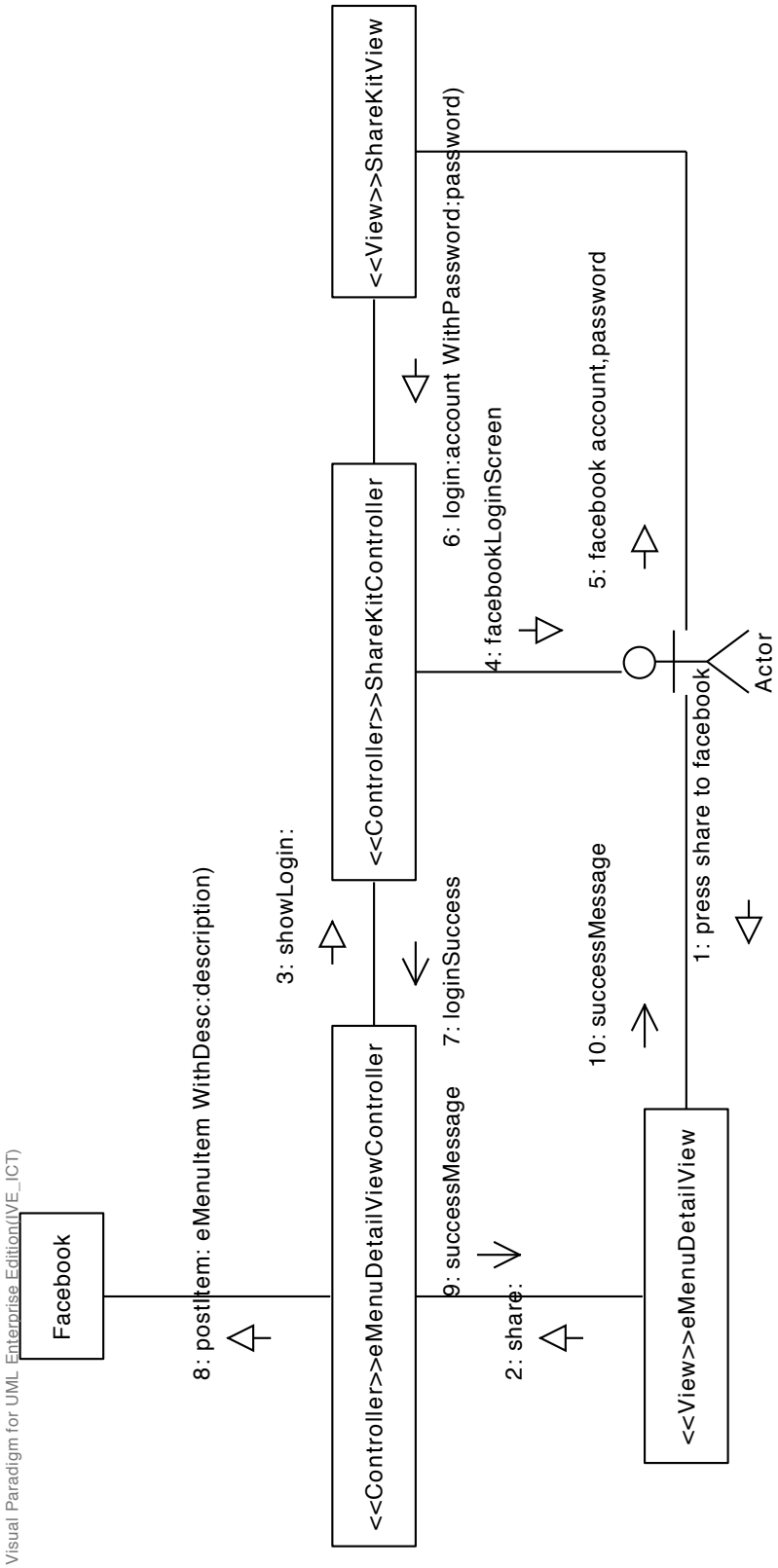


Visual Paradigm for UML Enterprise Edition(VE_ICT)

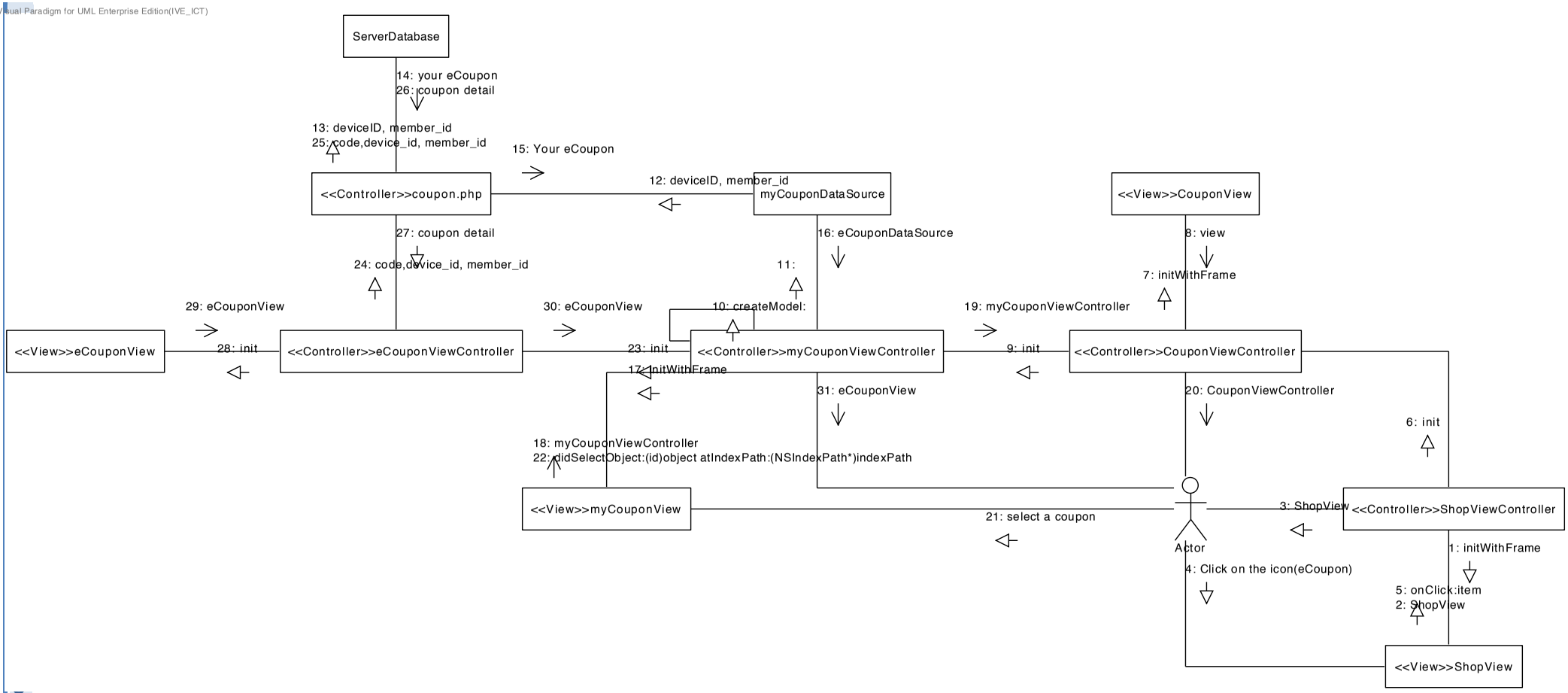
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

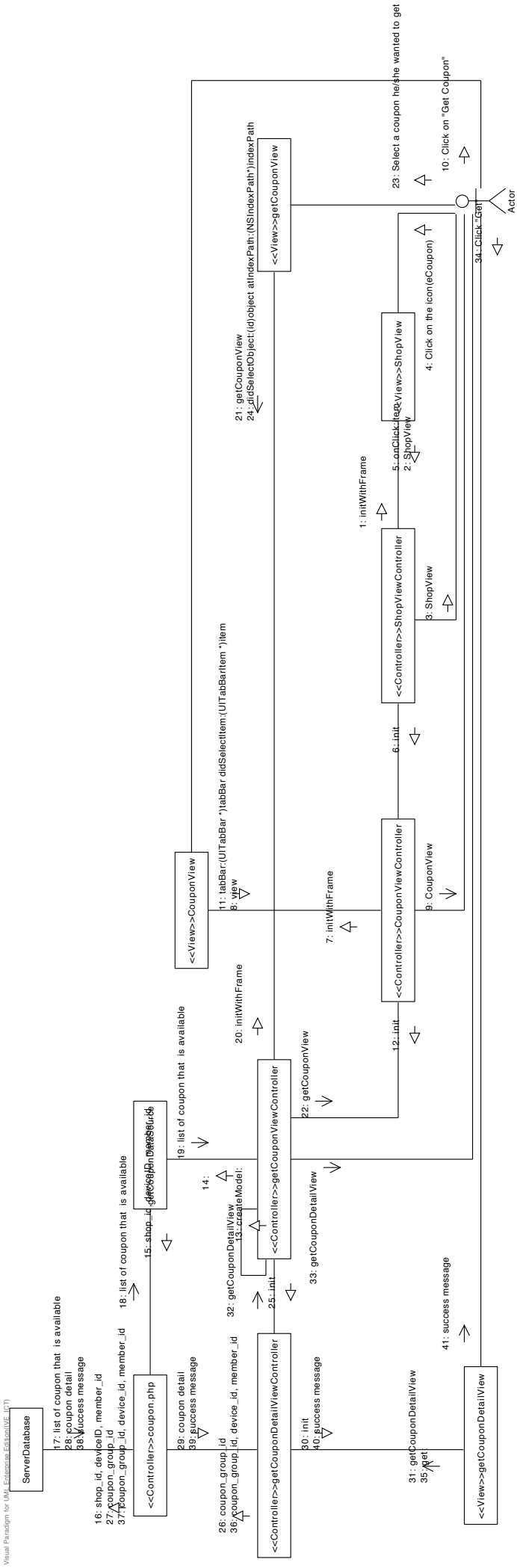


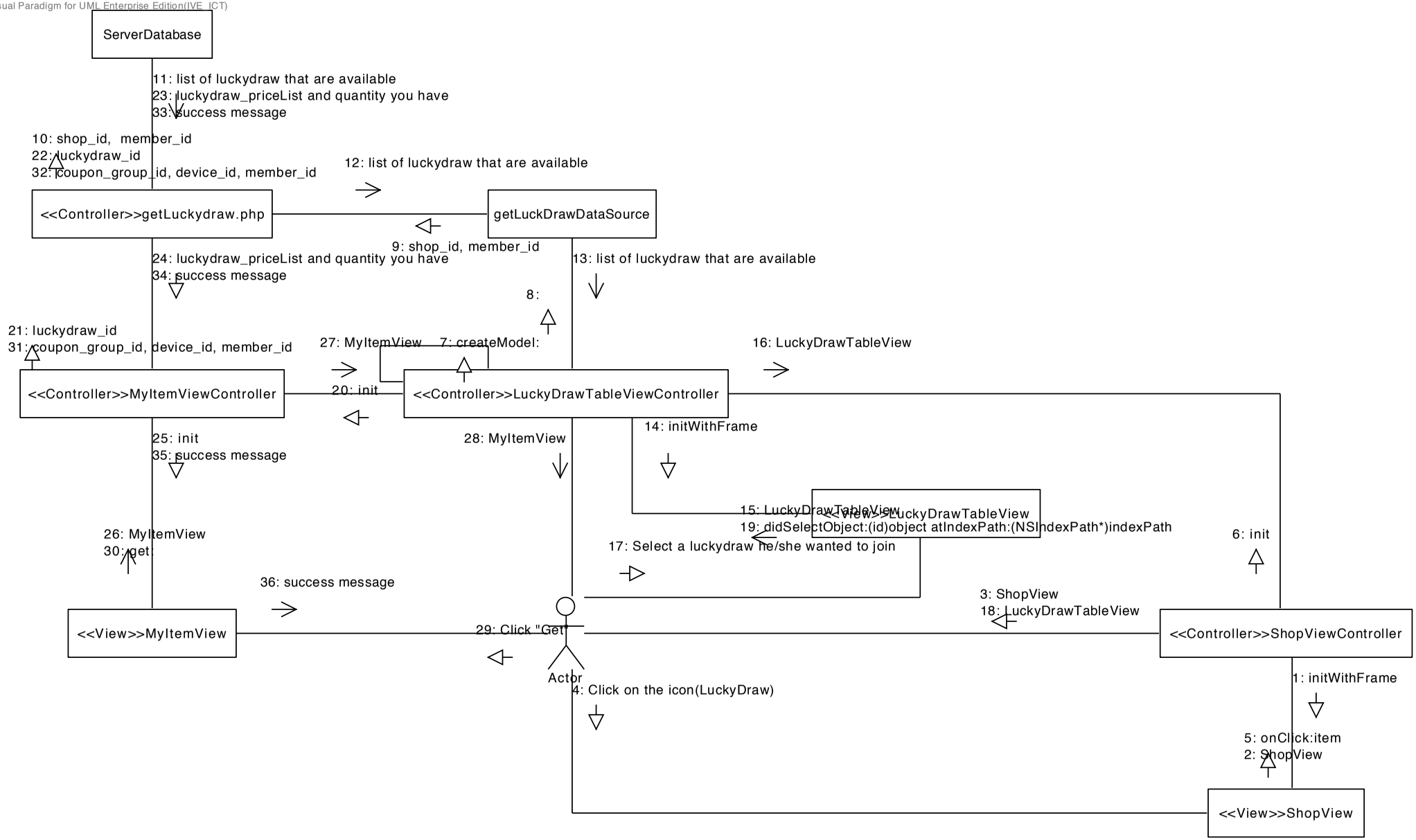

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
<<View>>eMenuTableView
<<Controller>>ShopViewController
<<View>>ShopView
Actor
<<Controller>>geteMenu.php
ServerDatabase
eMenuDataSource
<<Controller>> : DBConnect
<<Controller>>eMenuTableViewController
<<Controller>>eMenuDetailViewController
<<View>>eMenuDetailView
1: initWithFrame
2: ShopView
3: ShopView
4: Click on the icon(eMenu)
5: onClick:item
6: init
7: createModel
8:
9: executeQuery:SELECT IFNULL(max(last_update),0) FROM news WHERE news.shop_id = shop_id
10: lastUpdate
11: lastUpdate
12: lastUpdate
13: eMenu
14: the eMenu that added or changed since last update
15: insert eMenu / updated eMenu
16: executeQuery:query
17: eMenu
18: eMenuDataSource
19: initWithFrame
20: eMenuTableView
21: eMenuTableView
22: select a eMenu
23: didSelectObject:(id)object atIndexPath:(NSIndexPath*)indexPath
24: eMenu
25: executeQuery:query
26: eMenu detail
27: init
28: eMenu Detail View
29: eMenu Detail View
30: eMenu Detail View

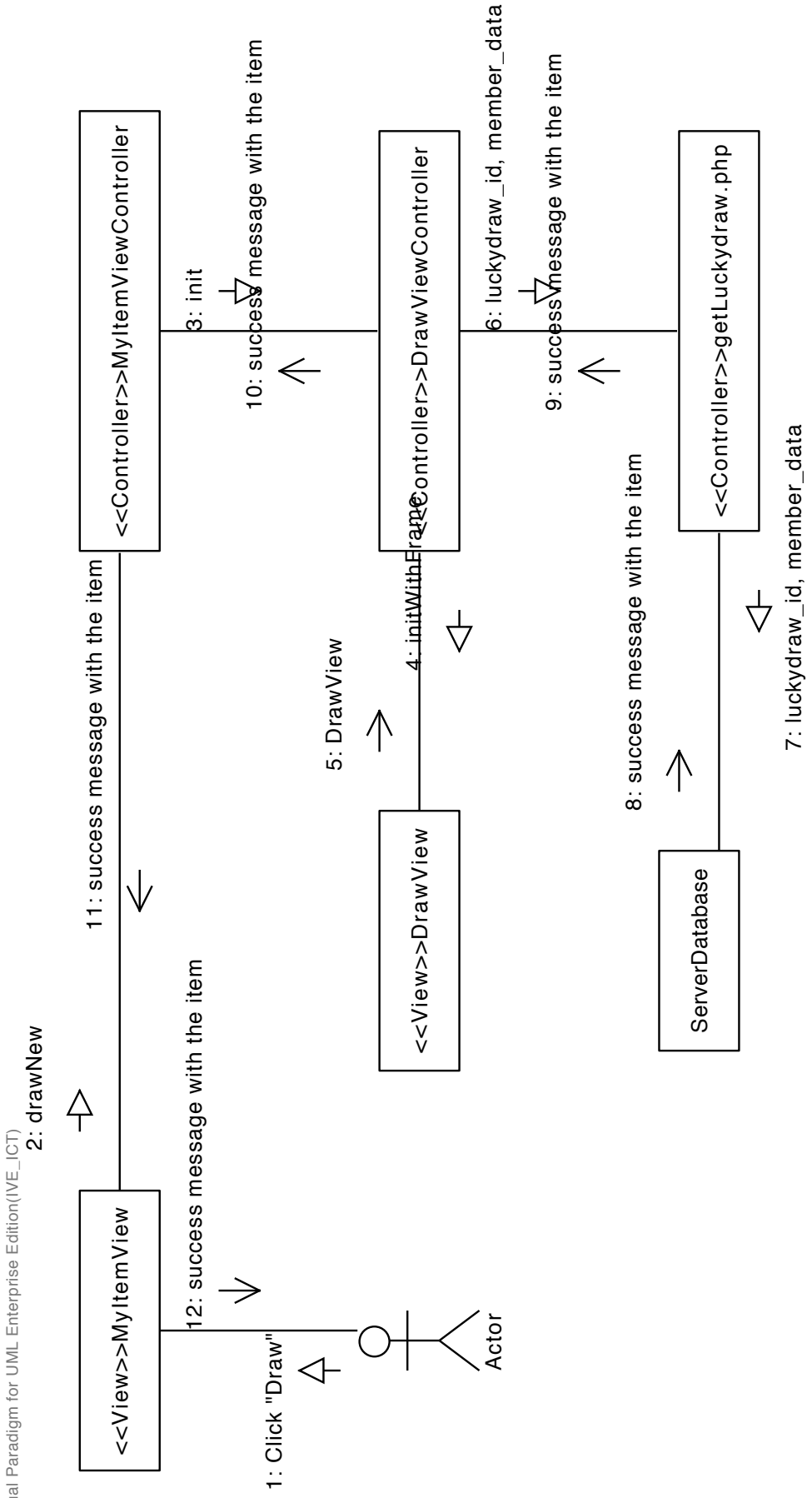

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Facebook
8: postItem: eMenuItem WithDesc:description)
<<Controller>>eMenuDetailViewController
3: showLogin:
<<Controller>>ShareKitController
<<View>>ShareKitView
6: login:account WithPassword:password)
7: loginSuccess
9: successMessage
2: share:
4: facebookLoginScreen
5: facebook account,password
10: successMessage
1: press share to facebook
<<View>>eMenuDetailView
Actor

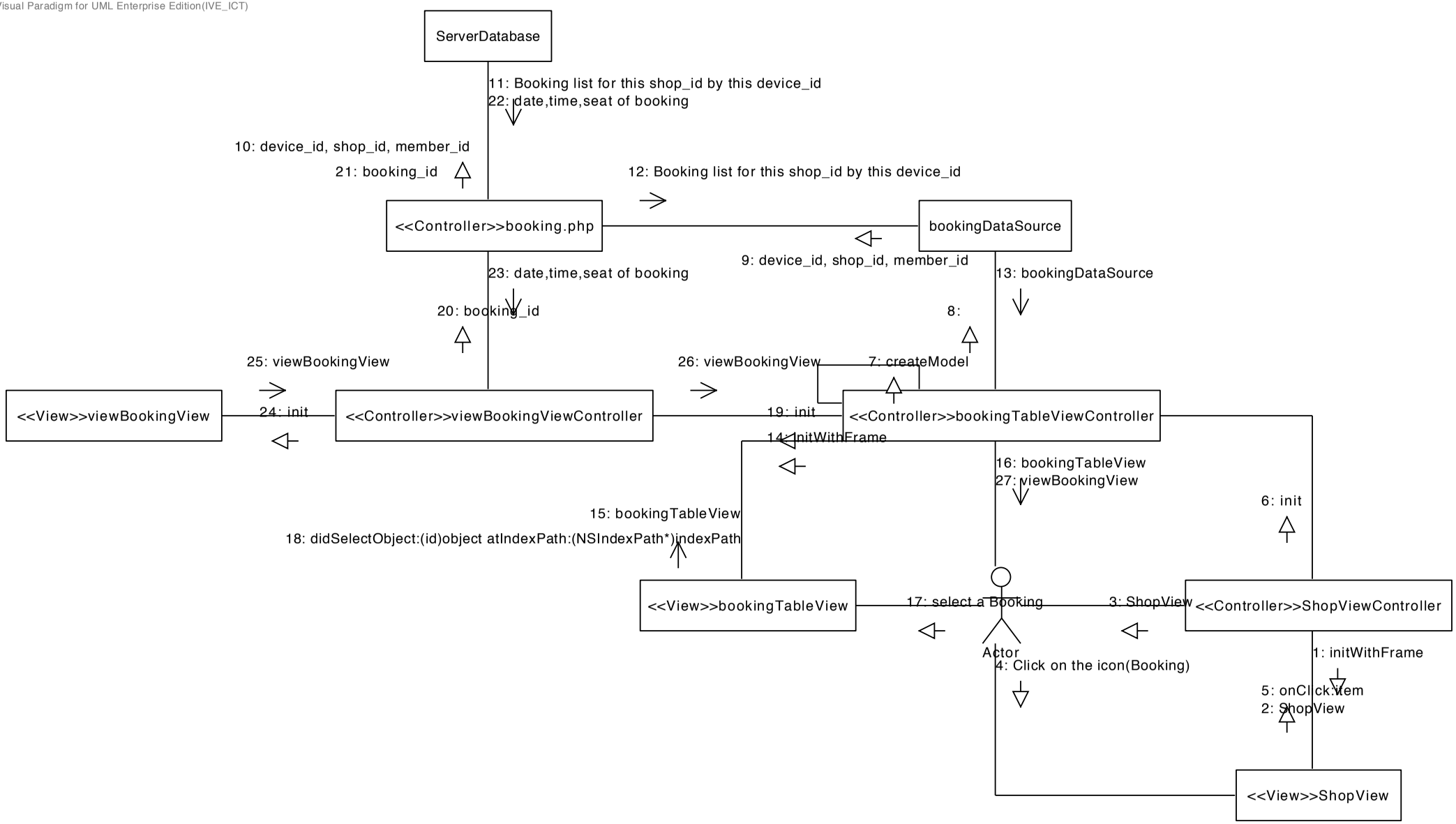
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

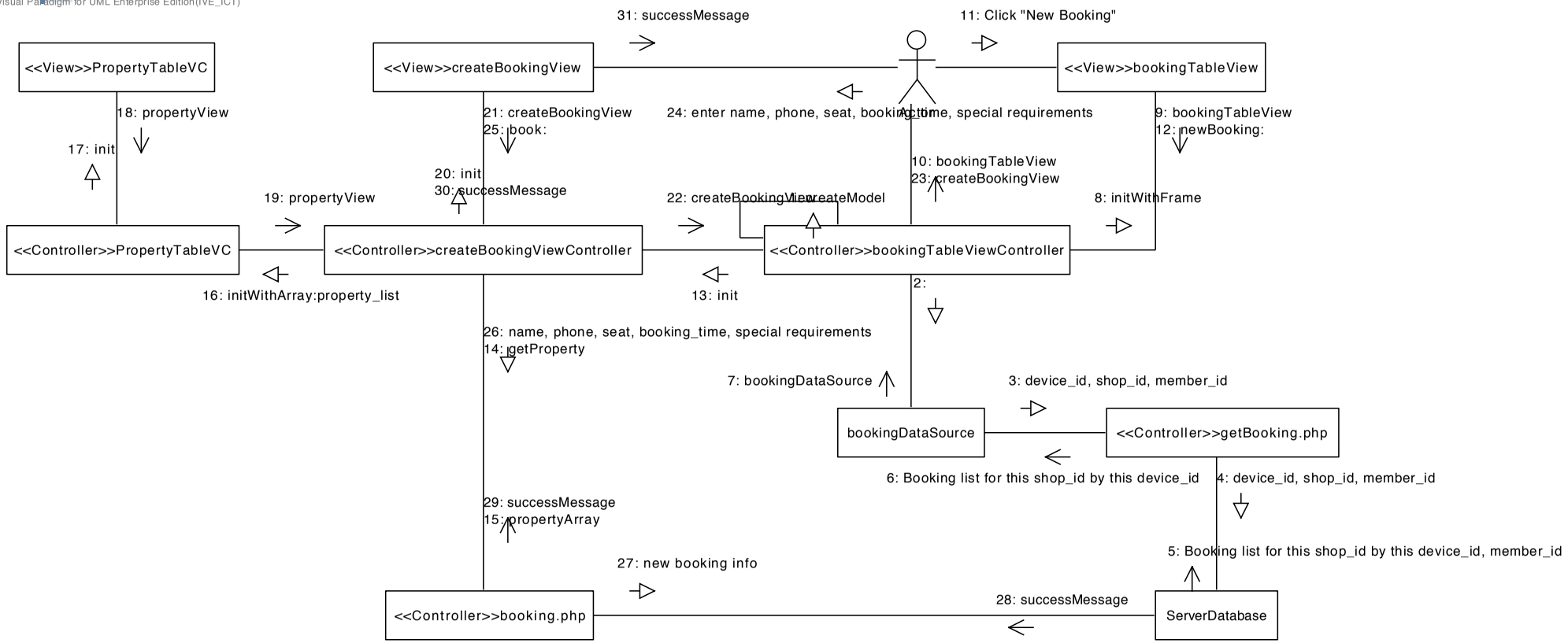


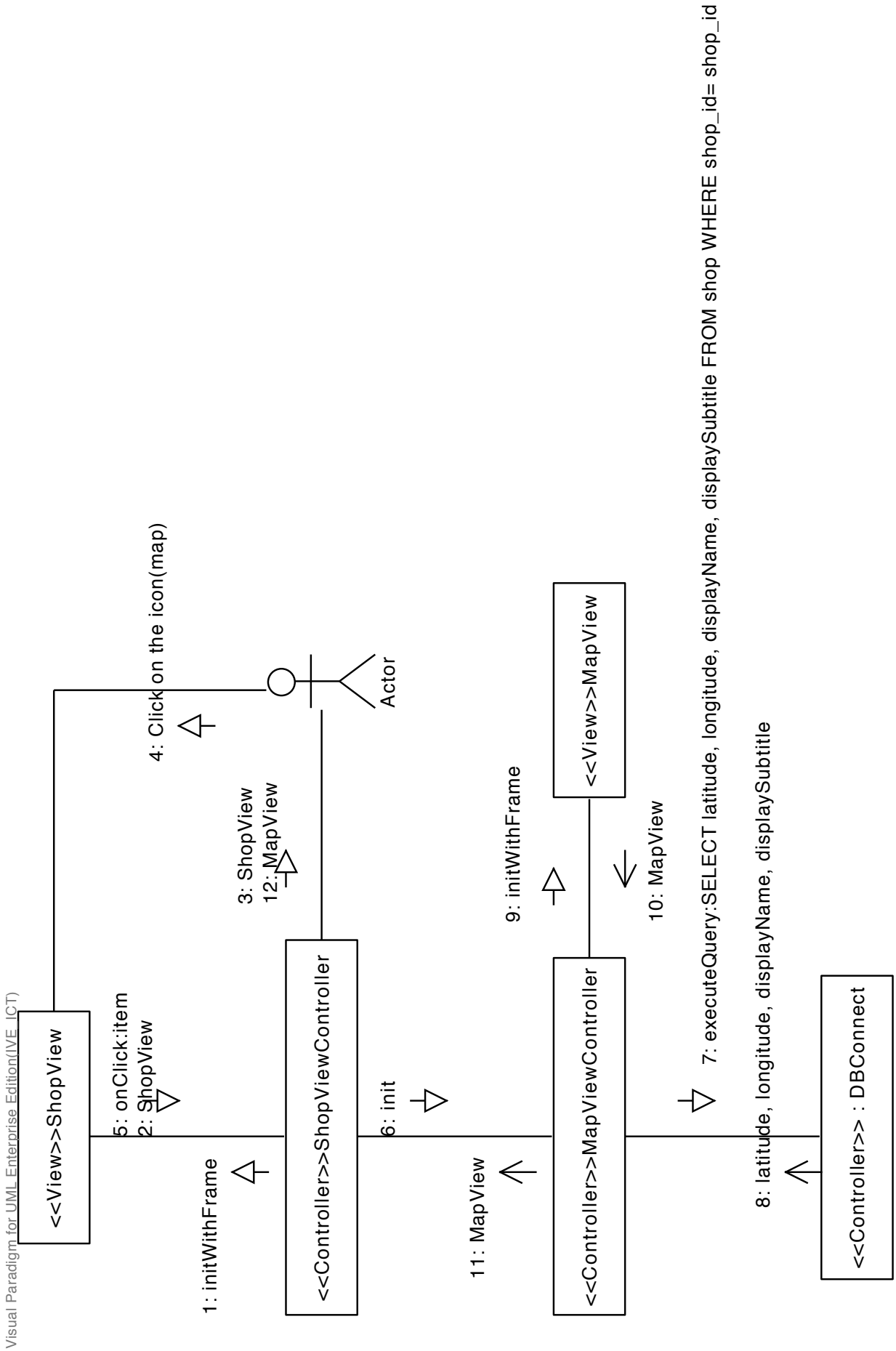

Visual Paradigm for UML Enterprise Edition(VE_UT)
ServerDatabase
<<Controller>>coupon.php
<<Controller>>getCouponDetailViewController
<<View>>getCouponDetailView
<<View>>CouponView
<<Controller>>getCouponViewController
<<Controller>>CouponViewController
<<Controller>>ShopViewController
<<View>>ShopView
<<View>>getCouponView
Actor
17: list of coupon that is available
28: coupon detail
38: success message
16: shop_id, deviceID, member_id
27: coupon_group_id
37: coupon_group_id, device_id, member_id
18: list of coupon that is available
15: shop_id, deviceID, member_id
19: list of coupon that is available
14:
29: coupon detail
39: success message
26: coupon_group_id
36: coupon_group_id, device_id, member_id
32: getCouponDetailView
13: createModel:
25: init
20: initWithFrame
11: tabBar:(UITabBar *)tabBar didSelectItem:(UITabBarItem *)item
8: view
21: getCouponView
24: didSelectObject:(id)object atIndexPath:(NSIndexPath*)indexPath
7: initWithFrame
12: init
22: getCouponView
33: getCouponDetailView
30: init
40: success message
31: getCouponDetailView
35: get
41: success message
9: CouponView
6: init
1: initWithFrame
3: ShopView
5: onClick:fromViews>>ShopView
2: ShopView
4: Click on the icon(eCoupon)
23: Select a coupon he/she wanted to get
10: Click on "Get Coupon"
34: Click "Get"

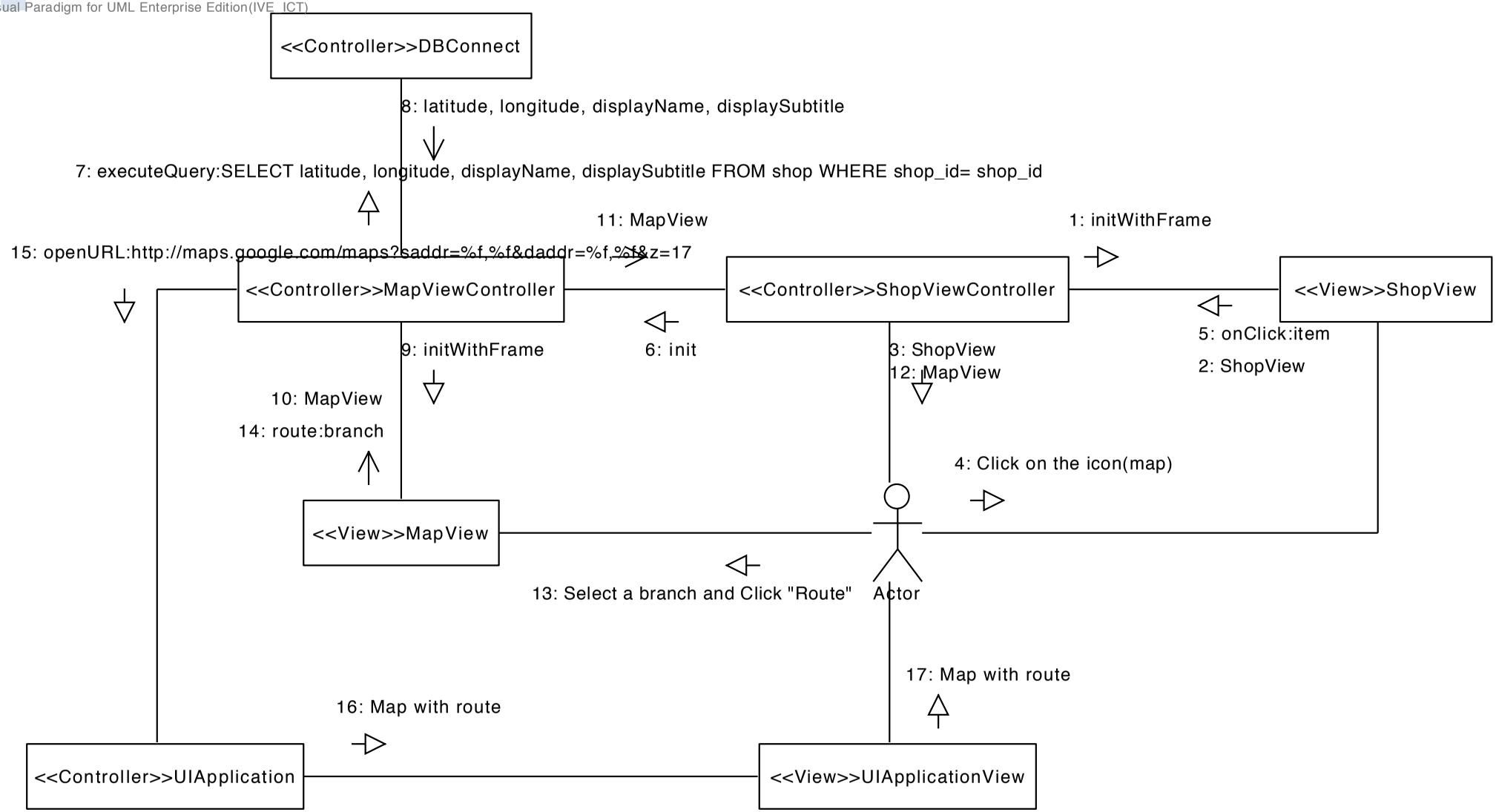

ServerDatabase
11: list of luckydraw that are available
23: luckydraw_priceList and quantity you have
33: success message
10: shop_id, member_id
22: luckydraw_id
32: coupon_group_id, device_id, member_id
12: list of luckydraw that are available
<<Controller>>getLuckydraw.php
getLuckDrawDataSource
9: shop_id, member_id
24: luckydraw_priceList and quantity you have
34: success message
13: list of luckydraw that are available
8:
21: luckydraw_id
31: coupon_group_id, device_id, member_id
27: MyItemView
7: createModel:
16: LuckyDrawTableView
<<Controller>>MyItemViewController
20: init
<<Controller>>LuckyDrawTableViewController
14: initWithFrame
25: init
35: success message
28: MyItemView
15: LuckyDrawTableView
<<View>>LuckyDrawTableView
19: didSelectObject:(id)object atIndexPath:(NSIndexPath*)indexPath
26: MyItemView
30: get
17: Select a luckydraw he/she wanted to join
6: init
36: success message
3: ShopView
18: LuckyDrawTableView
<<View>>MyItemView
29: Click "Get"
<<Controller>>ShopViewController
Actor
4: Click on the icon(LuckyDraw)
1: initWithFrame
5: onClick:item
2: ShopView
<<View>>ShopView

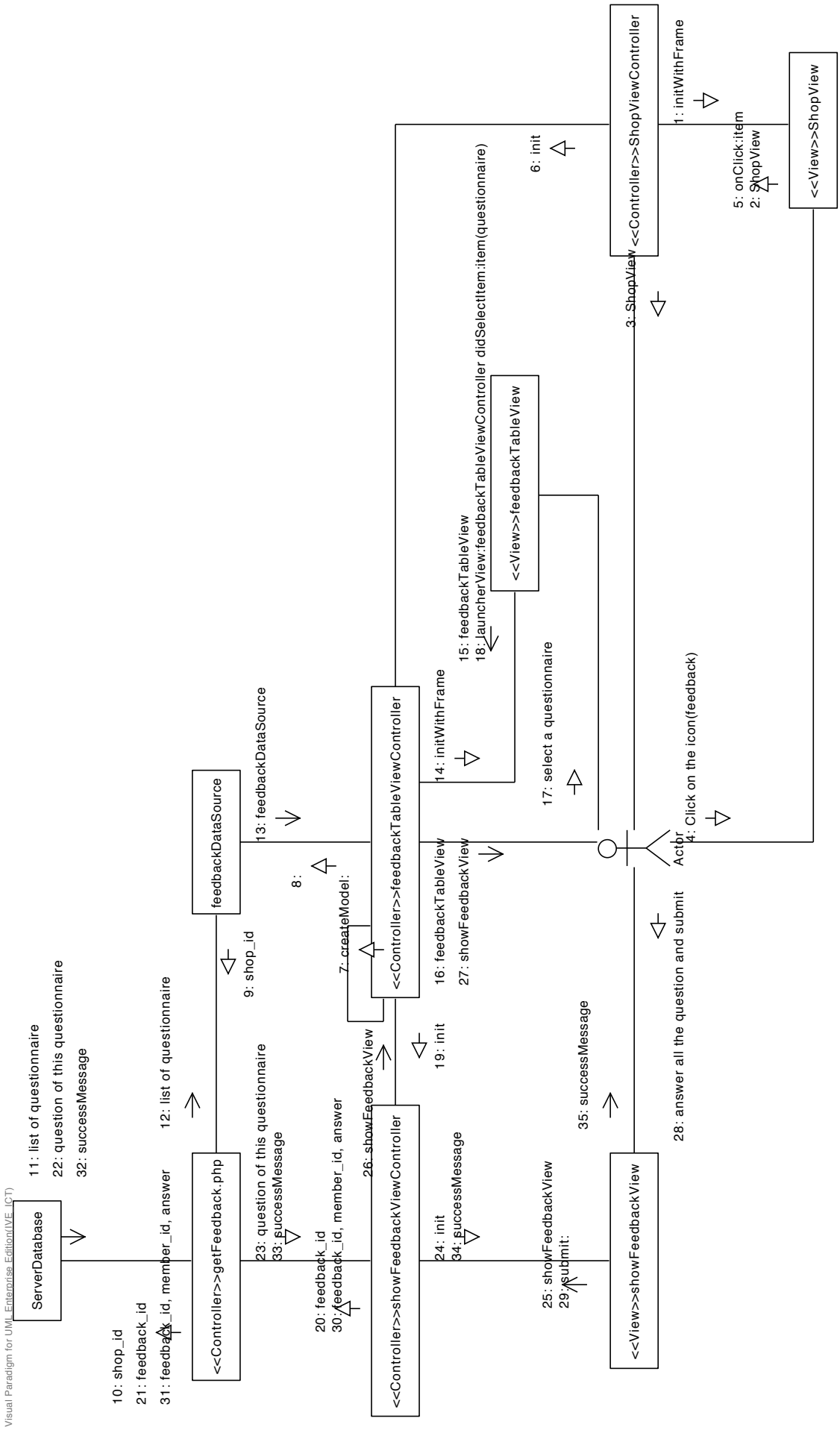

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
<<View>>MyItemView
<<Controller>>MyItemViewController
<<Controller>>DrawViewController
<<View>>DrawView
<<Controller>>getLuckydraw.php
ServerDatabase
Actor
1: Click "Draw"
2: drawNew
3: init
4: initWithFrame
5: DrawView
6: luckydraw_id, member_data
7: luckydraw_id, member_data
8: success message with the item
9: success message with the item
10: success message with the item
11: success message with the item
12: success message with the item

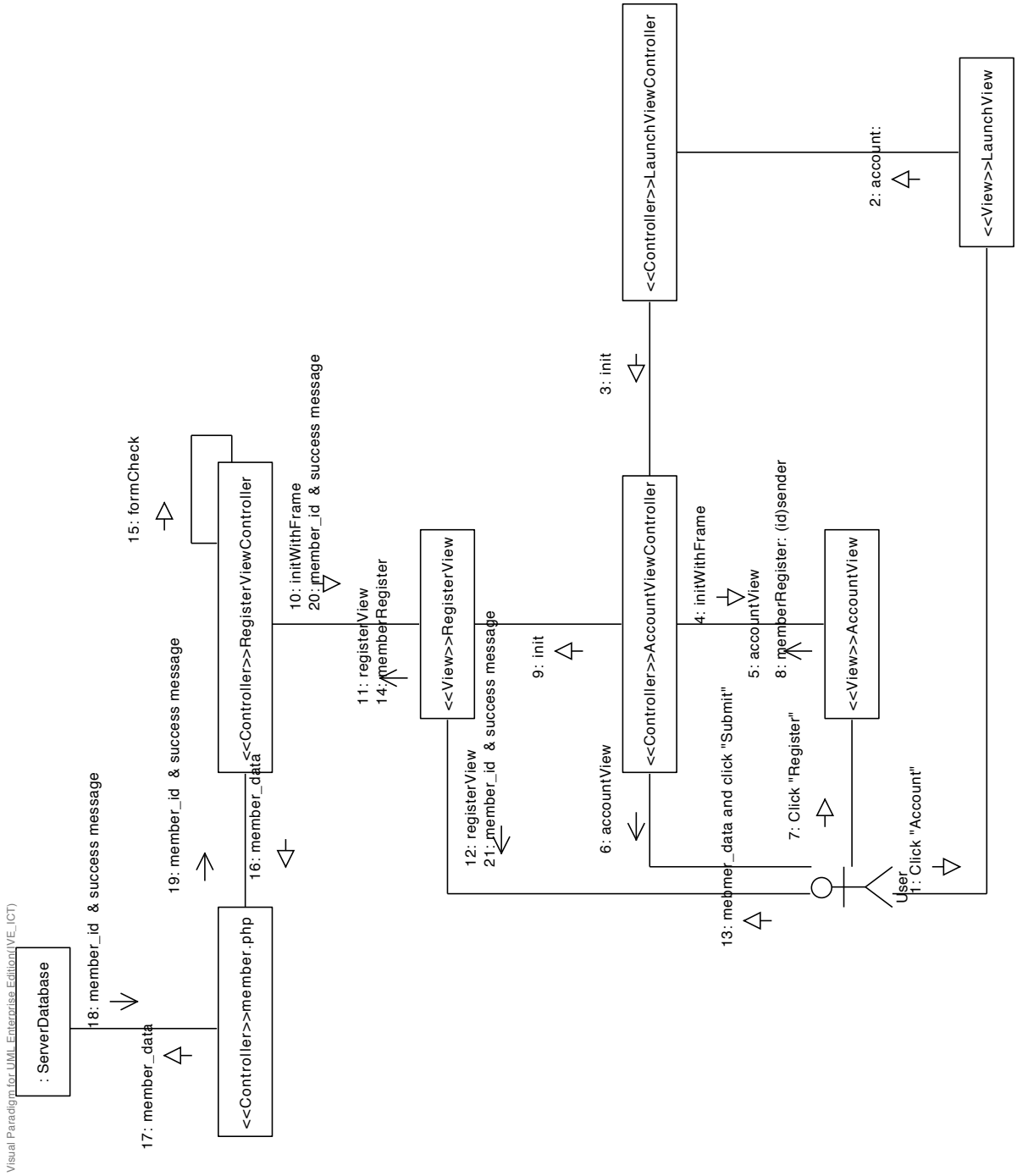

ServerDatabase
11: Booking list for this shop_id by this device_id
22: date,time,seat of booking
10: device_id, shop_id, member_id
21: booking_id
12: Booking list for this shop_id by this device_id
<<Controller>>booking.php
bookingDataSource
9: device_id, shop_id, member_id
23: date,time,seat of booking
13: bookingDataSource
20: booking_id
8:
25: viewBookingView
26: viewBookingView
7: createModel
<<View>>viewBookingView
24: init
<<Controller>>viewBookingViewController
19: init
<<Controller>>bookingTableViewController
14: initWithFrame
16: bookingTableView
27: viewBookingView
6: init
15: bookingTableView
18: didSelectObject:(id)object atIndexPath:(NSIndexPath*)indexPath
<<View>>bookingTableView
17: select a Booking
3: ShopView
<<Controller>>ShopViewController
Actor
4: Click on the icon(Booking)
1: initWithFrame
5: onClickItem
2: ShopView
<<View>>ShopView

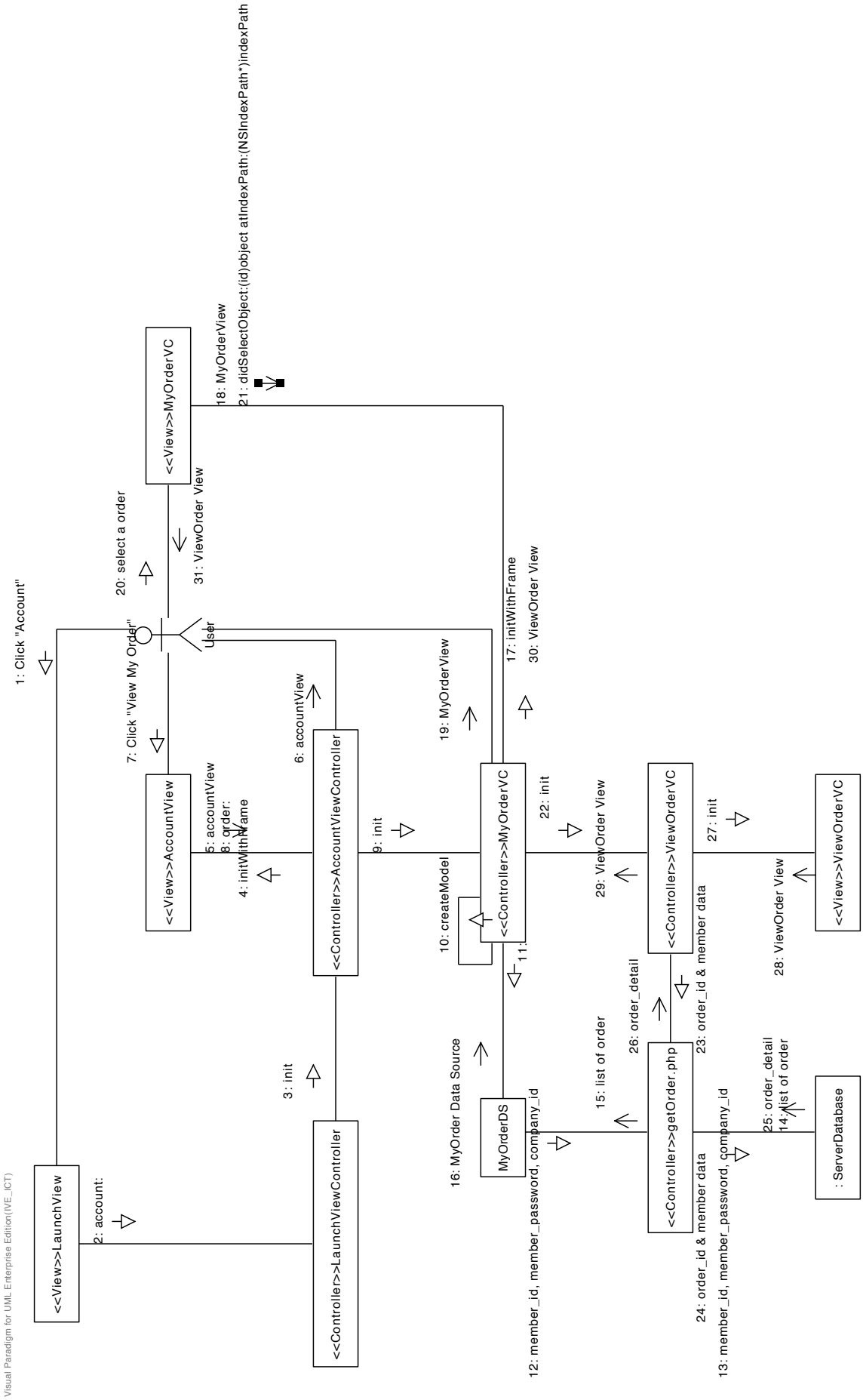

<<View>>PropertyTableVC
<<View>>createBookingView
<<View>>bookingTableView
31: successMessage
11: Click "New Booking"
18: propertyView
17: init
21: createBookingView
25: book:
24: enter name, phone, seat, booking_time, special requirements
9: bookingTableView
12: newBooking:
20: init
30: successMessage
10: bookingTableView
23: createBookingView
19: propertyView
22: createBookingViewcreateModel
8: initWithFrame
<<Controller>>PropertyTableVC
<<Controller>>createBookingViewController
<<Controller>>bookingTableViewController
16: initWithArray:property_list
13: init
2:
26: name, phone, seat, booking_time, special requirements
14: getProperty
7: bookingDataSource
3: device_id, shop_id, member_id
bookingDataSource
<<Controller>>getBooking.php
6: Booking list for this shop_id by this device_id
4: device_id, shop_id, member_id
29: successMessage
15: propertyArray
5: Booking list for this shop_id by this device_id, member_id
27: new booking info
<<Controller>>booking.php
28: successMessage
ServerDatabase

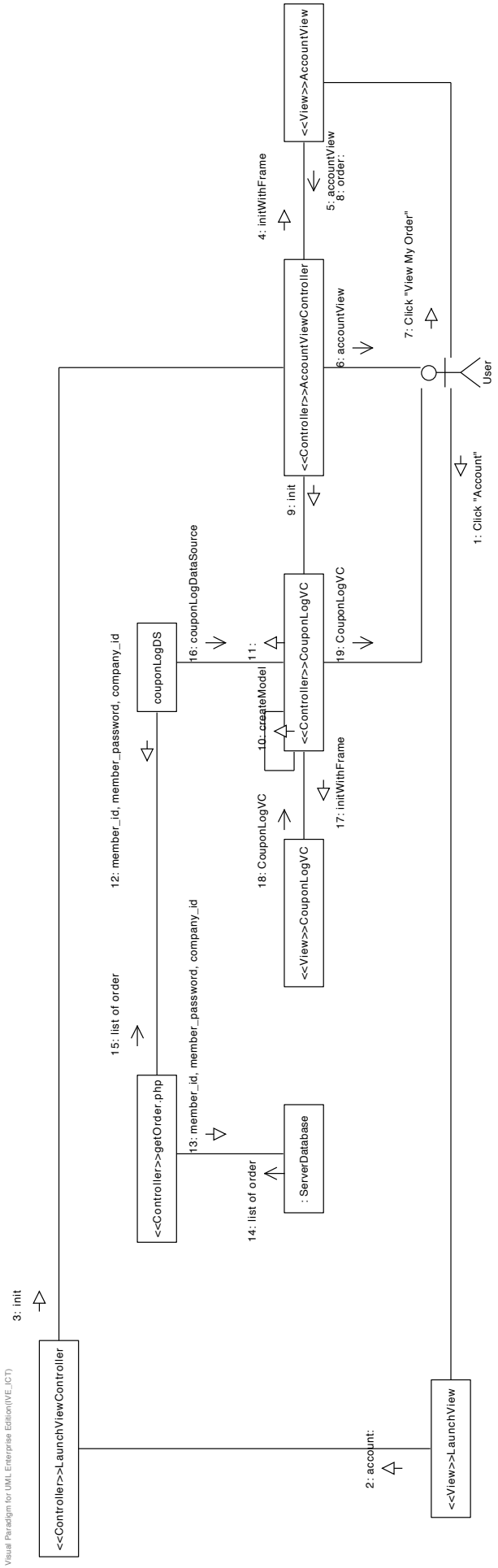

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
<<View>>ShopView
4: Click on the icon(map)
Actor
5: onClick:item
2: ShopView
1: initWithFrame
<<Controller>>ShopViewController
3: ShopView
12: MapView
6: init
11: MapView
<<Controller>>MapViewController
9: initWithFrame
<<View>>MapView
10: MapView
7: executeQuery:SELECT latitude, longitude, displayName, displaySubtitle FROM shop WHERE shop_id= shop_id
8: latitude, longitude, displayName, displaySubtitle
<<Controller>> : DBConnect


<<Controller>>DBConnect
8: latitude, longitude, displayName, displaySubtitle
7: executeQuery:SELECT latitude, longitude, displayName, displaySubtitle FROM shop WHERE shop_id= shop_id
11: MapView
1: initWithFrame
15: openURL:http://maps.google.com/maps?saddr=%f,%f&daddr=%f,%f&z=17
<<Controller>>MapViewController
<<Controller>>ShopViewController
<<View>>ShopView
5: onClick:item
2: ShopView
9: initWithFrame
6: init
3: ShopView
12: MapView
10: MapView
14: route:branch
4: Click on the icon(map)
<<View>>MapView
13: Select a branch and Click "Route"
Actor
17: Map with route
16: Map with route
<<Controller>>UIApplication
<<View>>UIApplicationView

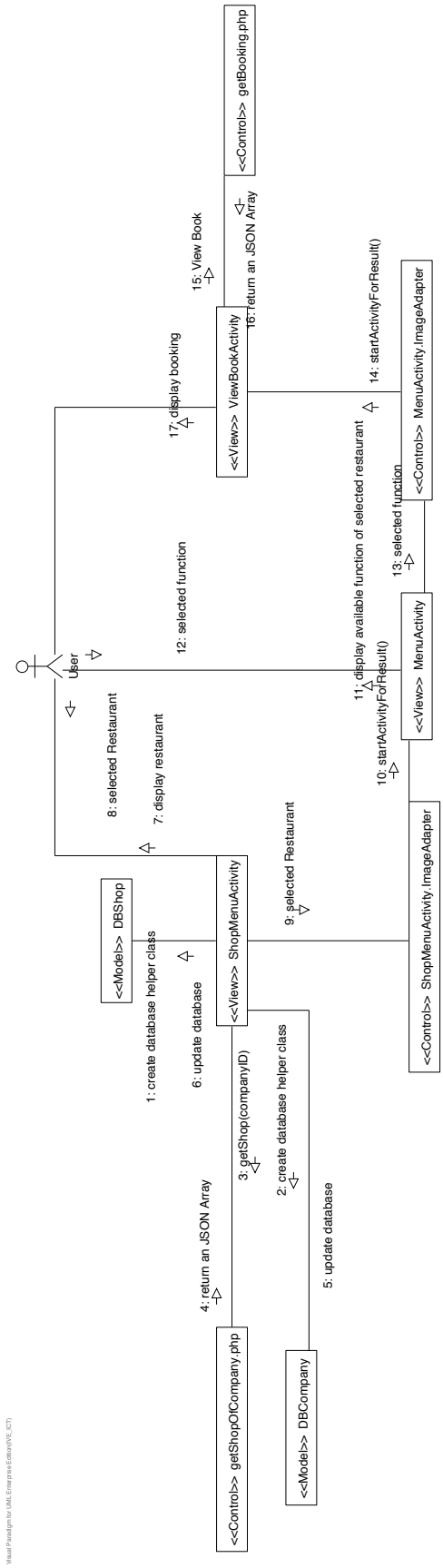

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
ServerDatabase
11: list of questionnaire
22: question of this questionnaire
32: successMessage
<<Controller>>getFeedback.php
12: list of questionnaire
10: shop_id
21: feedback_id
31: feedback_id, member_id, answer
feedbackDataSource
9: shop_id
13: feedbackDataSource
23: question of this questionnaire
33: successMessage
<<Controller>>feedbackTableViewController
7: createModel:
8:
14: initWithFrame
15: feedbackTableView
18: launcherView:feedbackTableViewController didSelectItem:item(questionnaire)
<<View>>feedbackTableView
16: feedbackTableView
27: showFeedbackView
<<Controller>>showFeedbackViewController
26: showFeedbackView
19: init
20: feedback_id
30: feedback_id, member_id, answer
24: init
34: successMessage
25: showFeedbackView
29: submit:
<<View>>showFeedbackView
35: successMessage
28: answer all the question and submit
17: select a questionnaire
Actor
4: Click on the icon(feedback)
3: ShopView
<<Controller>>ShopViewController
6: init
1: initWithFrame
5: onClick:item
2: ShopView
<<View>>ShopView

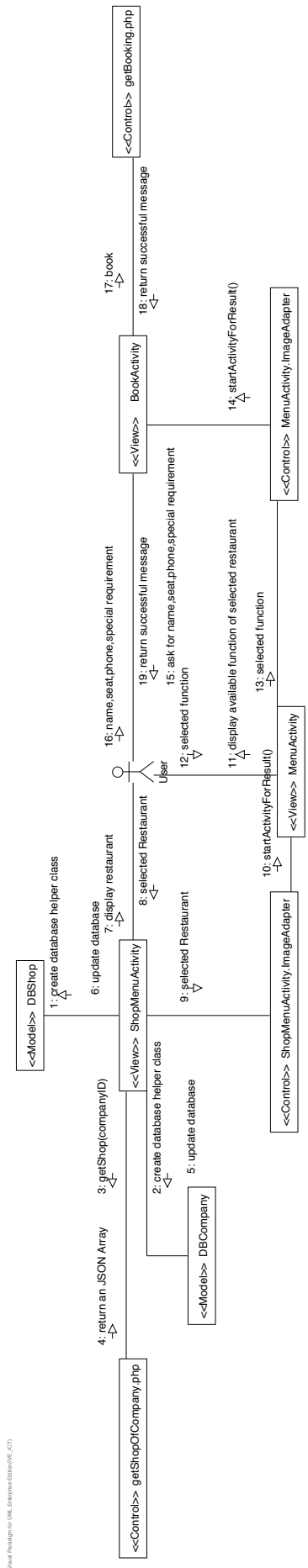

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
: ServerDatabase
<<Controller>>member.php
<<Controller>>RegisterViewController
<<View>>RegisterView
<<Controller>>AccountViewController
<<Controller>>LaunchViewController
<<View>>LaunchView
<<View>>AccountView
User
1: Click "Account"
2: account:
3: init
4: initWithFrame
5: accountView
6: accountView
7: Click "Register"
8: memberRegister: (id)sender
9: init
10: initWithFrame
11: registerView
12: registerView
13: member_data and click "Submit"
14: memberRegister
15: formCheck
16: member_data
17: member_data
18: member_id & success message
19: member_id & success message
20: member_id & success message
21: member_id & success message

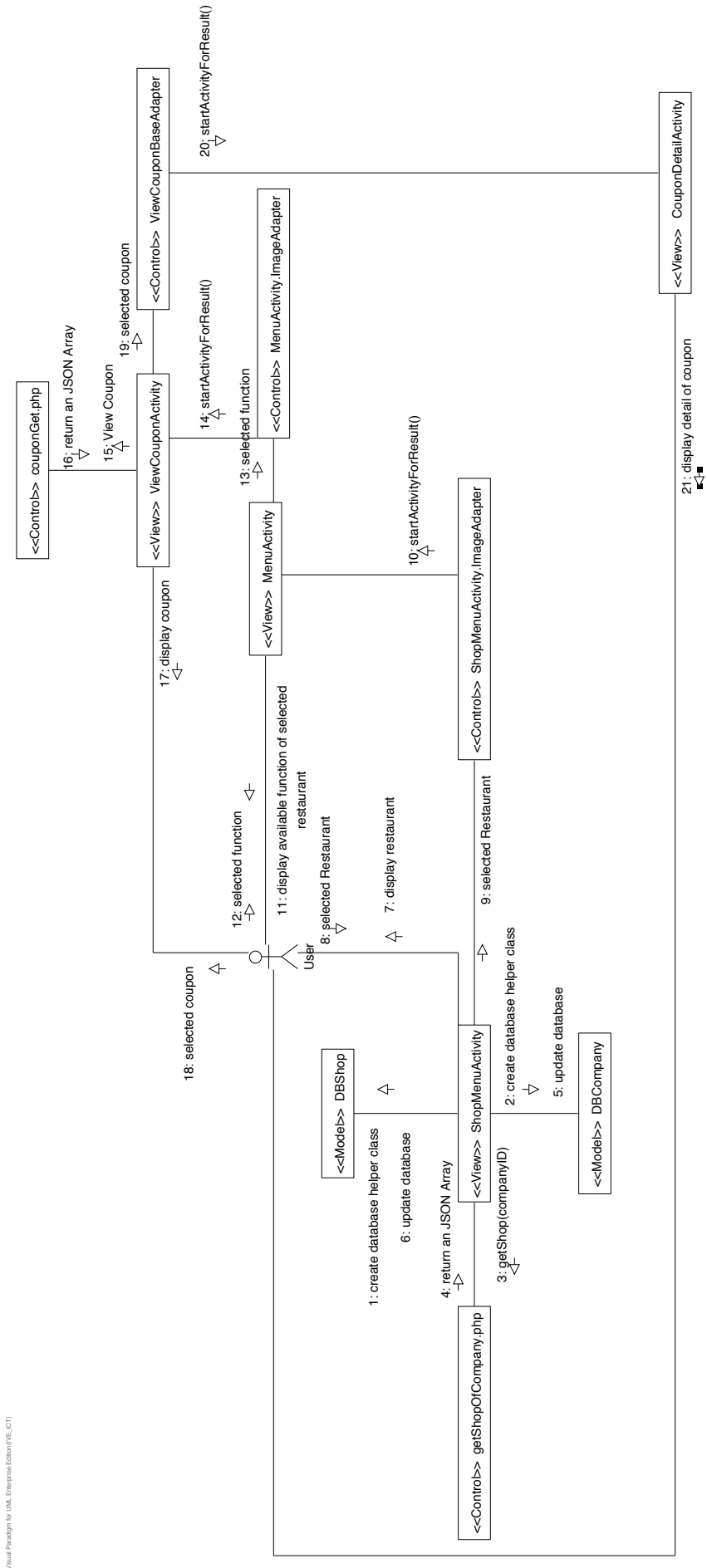

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
<<View>>LaunchView
<<View>>AccountView
<<View>>MyOrderVC
<<Controller>>LaunchViewController
<<Controller>>AccountViewController
<<Controller>>MyOrderVC
<<Controller>>ViewOrderVC
<<Controller>>getOrder.php
<<View>>ViewOrderVC
MyOrderDS
: ServerDatabase
User
1: Click "Account"
2: account:
3: init
4: initWithFrame
5: accountView
6: accountView
7: Click "View My Order"
8: order:
9: init
10: createModel
11:
12: member_id, member_password, company_id
13: member_id, member_password, company_id
14: list of order
15: list of order
16: MyOrder Data Source
17: initWithFrame
18: MyOrderView
19: MyOrderView
20: select a order
21: didSelectObject:(id)object atIndexPath:(NSIndexPath*)indexPath
22: init
23: order_id & member data
24: order_id & member data
25: order_detail
26: order_detail
27: init
28: ViewOrder View
29: ViewOrder View
30: ViewOrder View
31: ViewOrder View

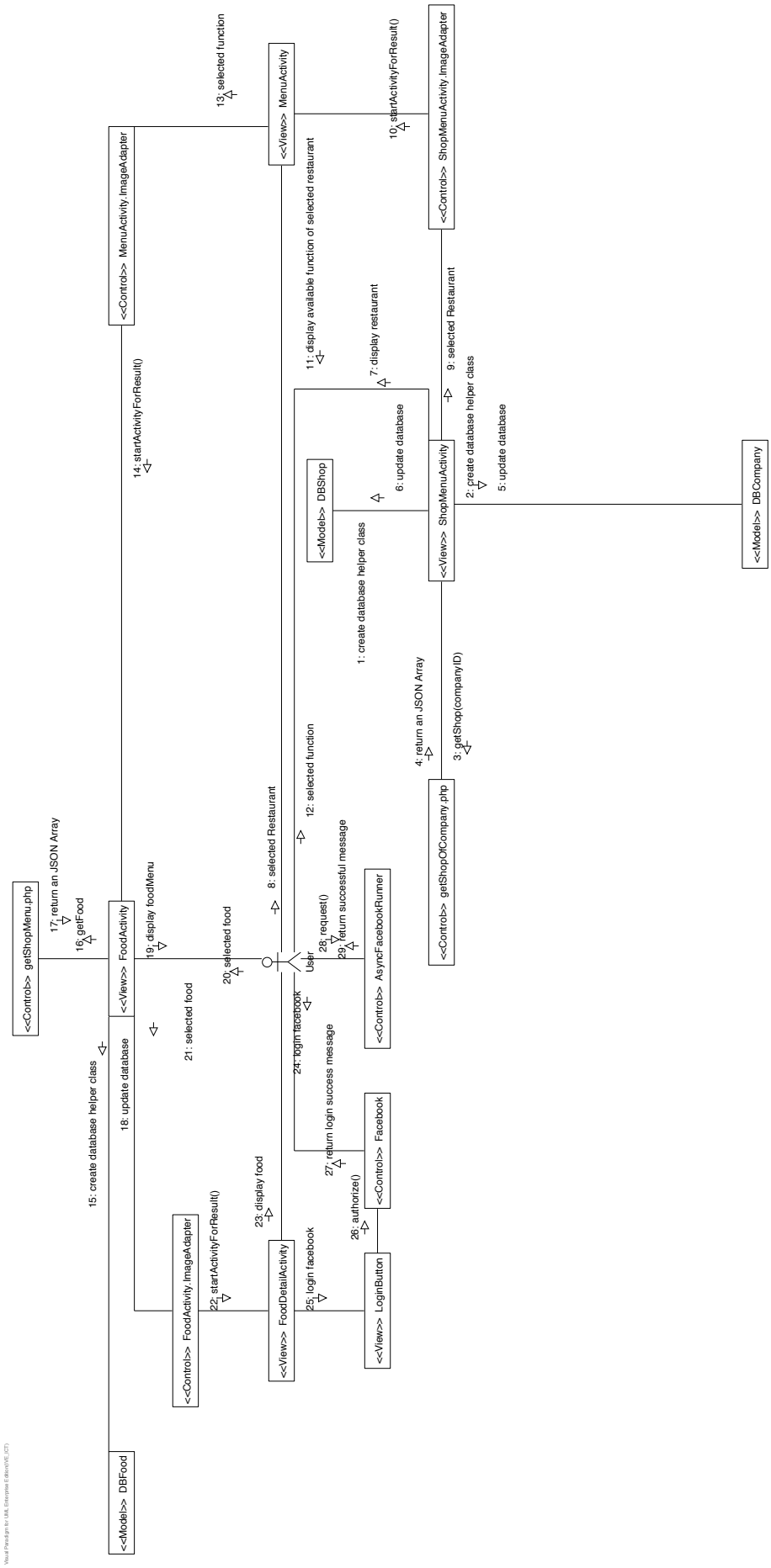

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
<<Controller>>LaunchViewController
3: init
<<Controller>>getOrder.php
15: list of order
12: member_id, member_password, company_id
13: member_id, member_password, company_id
: ServerDatabase
14: list of order
couponLogDS
16: couponLogDataSource
<<View>>LaunchView
2: account:
<<Controller>>AccountViewController
9: init
<<View>>AccountView
4: initWithFrame
5: accountView
8: order:
6: accountView
7: Click "View My Order"
User
1: Click "Account"
<<Controller>>CouponLogVC
11 :
10: createModel
19: CouponLogVC
17: initWithFrame
<<View>>CouponLogVC
18: CouponLogVC

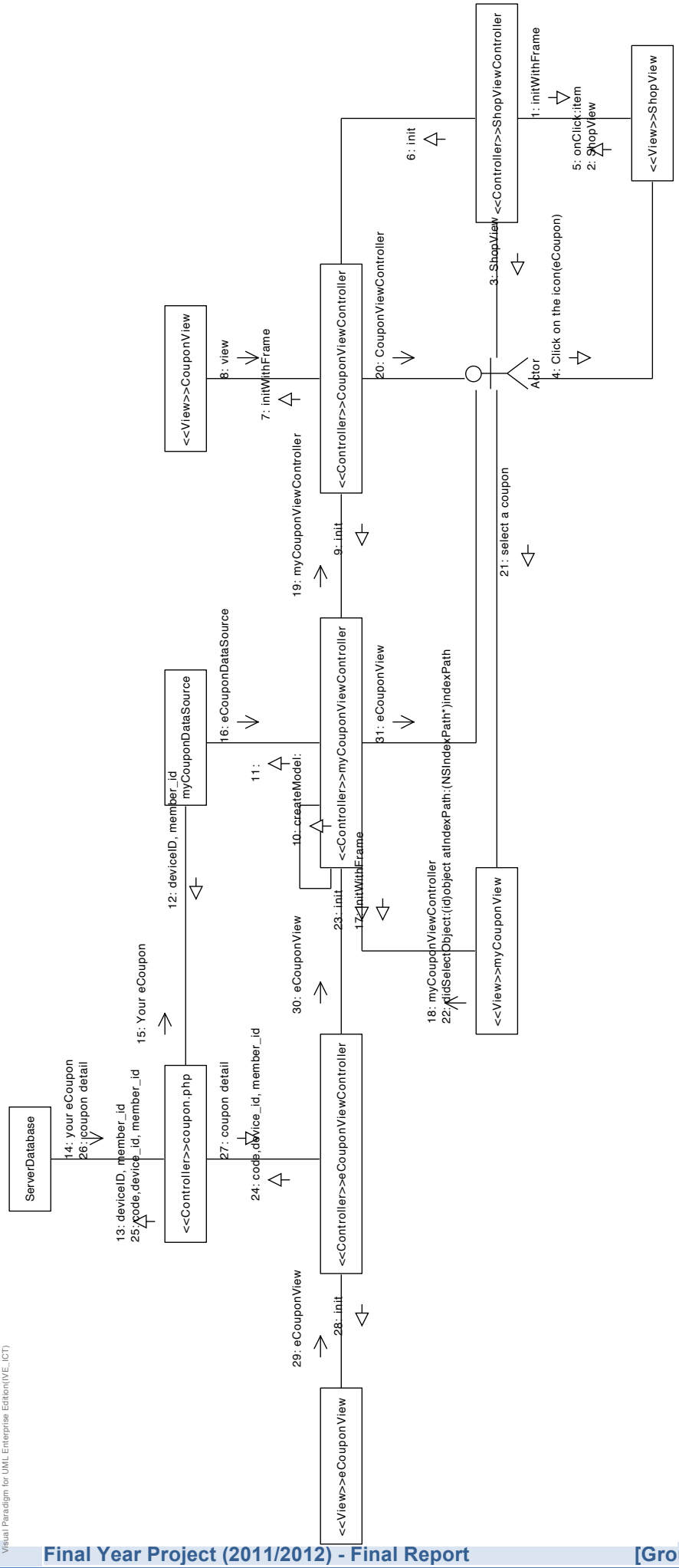
Object Collaboration Diagram (Android)



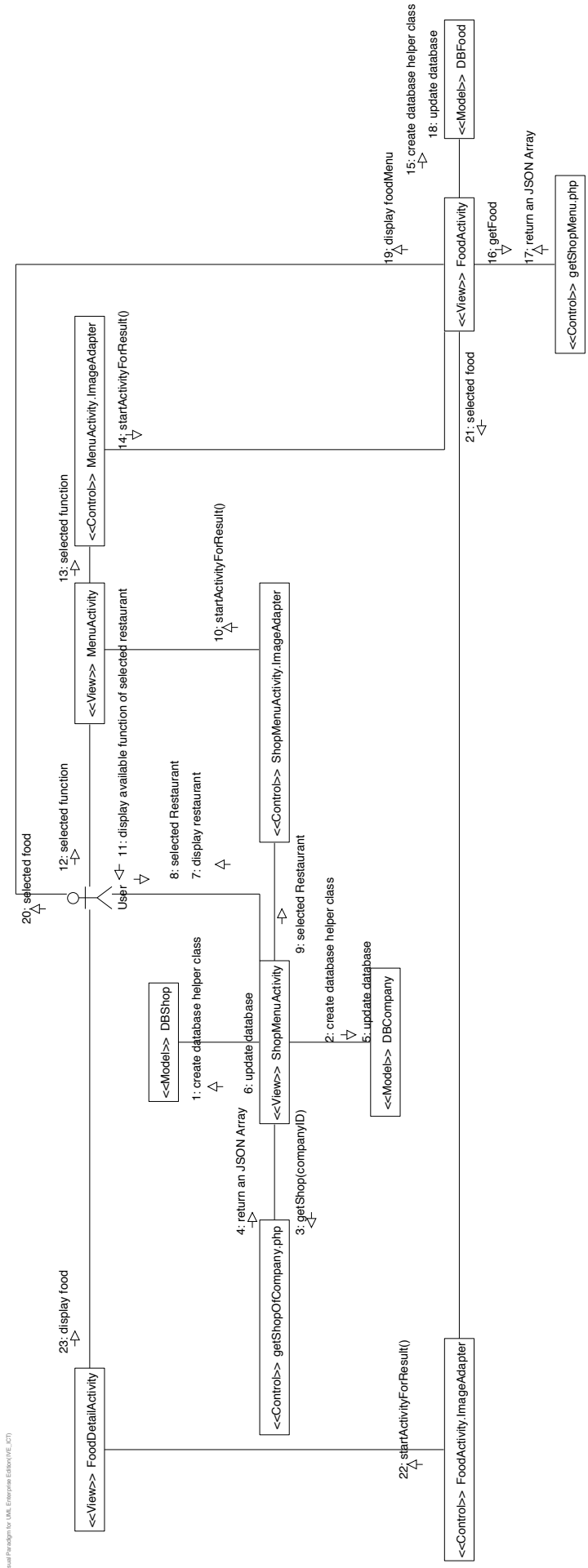

<<Model>> DBShop
1 : create database helper class
6: update database
<<View>> ShopMenuActivity
7: display restaurant
3: getShop(companyID)
2: create database helper class
5 : update database
<<Model>> DBCompany
4: return an JSON Array
<<Control>> getShopOfCompany.php
8: selected Restaurant
9: selected Restaurant
<<Control>> ShopMenuActivity.ImageAdapter
10: startActivityForResult()
<<View>> MenuActivity
User
11 : display available function of selected restaurant
12: selected function
13: selected function
<<Control>> MenuActivity.ImageAdapter
14: startActivityForResult()
<<View>> BookActivity
15 : ask for name,seat,phone,special requirement
16: name,seat,phone,special requirement
17: book
<<Control>> getBooking.php
18 : return successful message
19 : return successful message
Visual Paradigm for UML Enterprise Edition(NE_ICT)

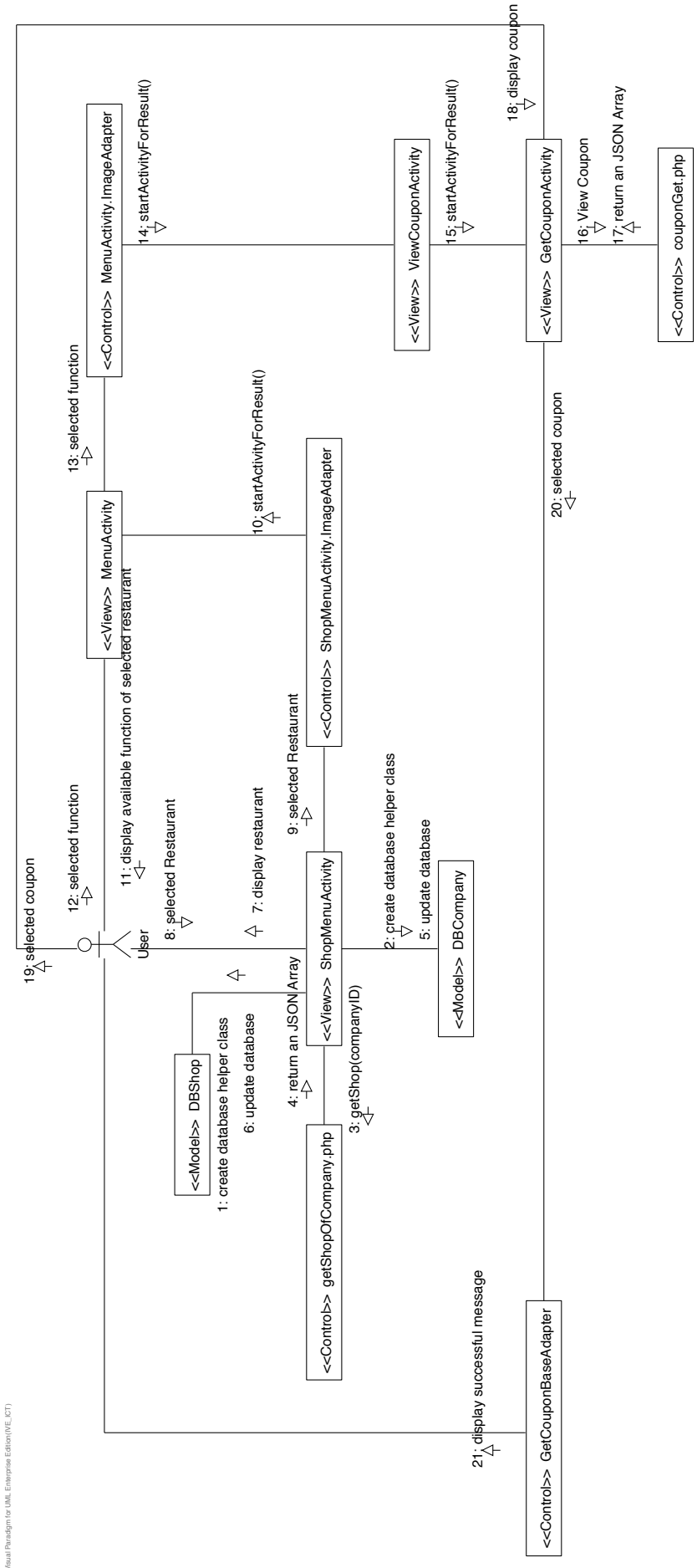

<<Control>> couponGet.php
16: return an JSON Array
15: View Coupon
<<View>> ViewCouponActivity
19: selected coupon
<<Control>> ViewCouponBaseAdapter
20: startActivityForResult()
<<View>> CouponDetailActivity
21: display detail of coupon
14: startActivityForResult()
13: selected function
<<Control>> MenuActivity.ImageAdapter
<<View>> MenuActivity
10: startActivityForResult()
<<Control>> ShopMenuActivity.ImageAdapter
17: display coupon
11: display available function of selected restaurant
12: selected function
8: selected Restaurant
7: display restaurant
9: selected Restaurant
User
18: selected coupon
2: create database helper class
5: update database
<<Model>> DBShop
<<View>> ShopMenuActivity
<<Model>> DBCompany
1: create database helper class
6: update database
4: return an JSON Array
3: getShop(companyID)
<<Control>> getShopOfCompany.php
Visual Paradigm for UML Enterprise Edition(VE_JCT)

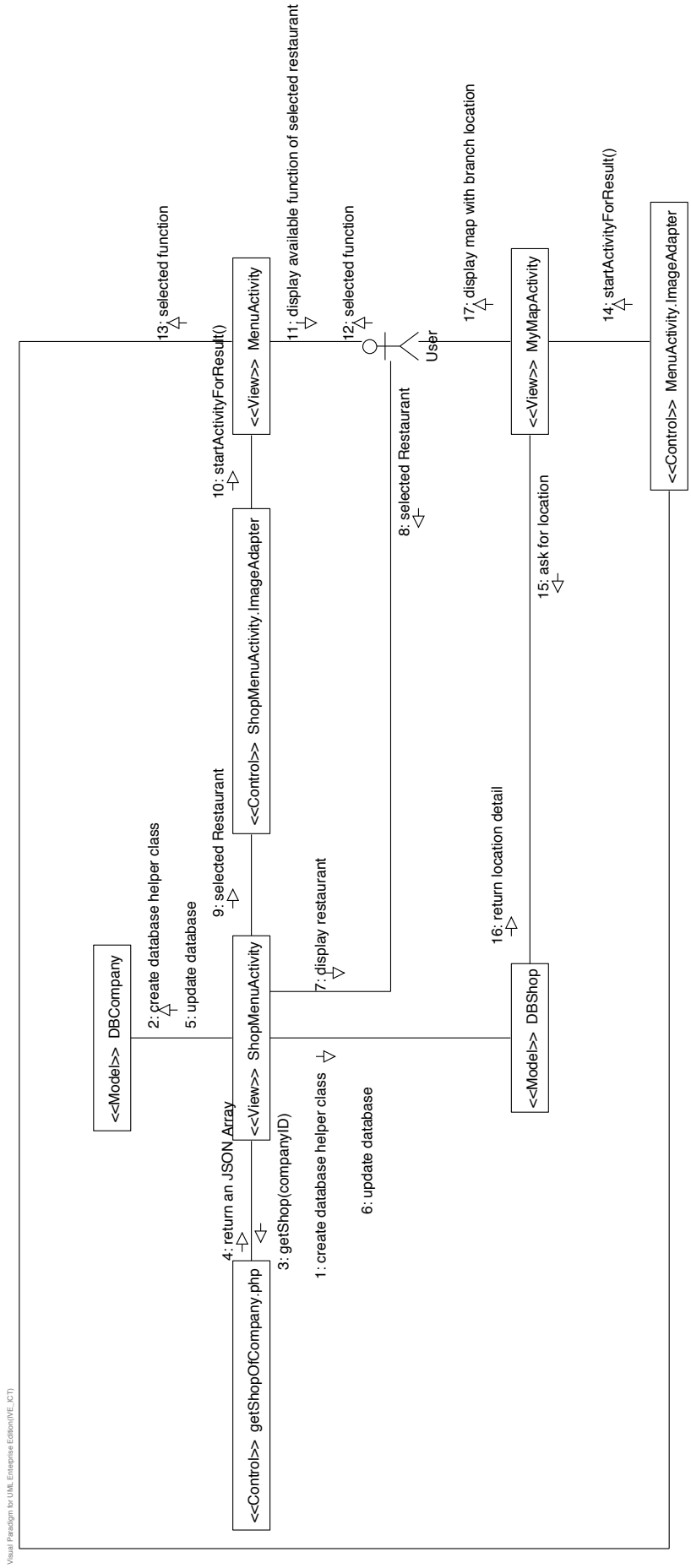

<<Control>> getShopMenu.php
17: return an JSON Array
16: getFood
<<View>> FoodActivity
19: display foodMenu
20: selected food
8: selected Restaurant
12: selected function
User
28: request()
29: return successful message
<<Control>> AsyncFacebookRunner
21: selected food
24: login facebook
27: return login success message
15: create database helper class
18: update database
<<Control>> FoodActivity.ImageAdapter
22: startActivityForResult()
23: display food
<<View>> FoodDetailActivity
25: login facebook
26: authorize()
<<Control>> Facebook
<<View>> LoginButton
<<Model>> DBFood
<<Control>> MenuActivity.ImageAdapter
14: startActivityForResult()
13: selected function
<<View>> MenuActivity
10: startActivityForResult()
<<Control>> ShopMenuActivity.ImageAdapter
11: display available function of selected restaurant
7: display restaurant
9: selected Restaurant
6: update database
<<Model>> DBShop
1: create database helper class
<<View>> ShopMenuActivity
2: create database helper class
5: update database
<<Model>> DBCompany
4: return an JSON Array
3: getShop(companyID)
<<Control>> getShopOfCompany.php

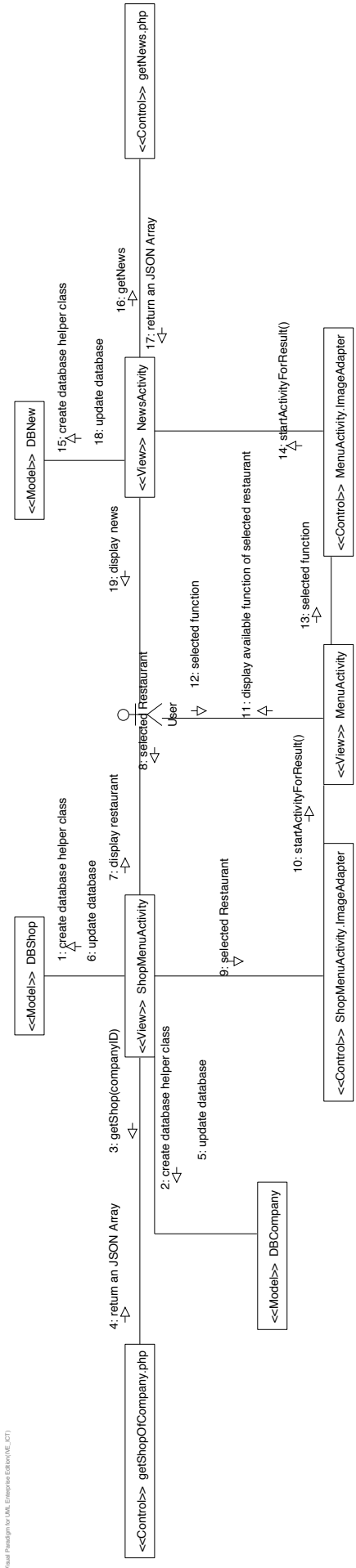


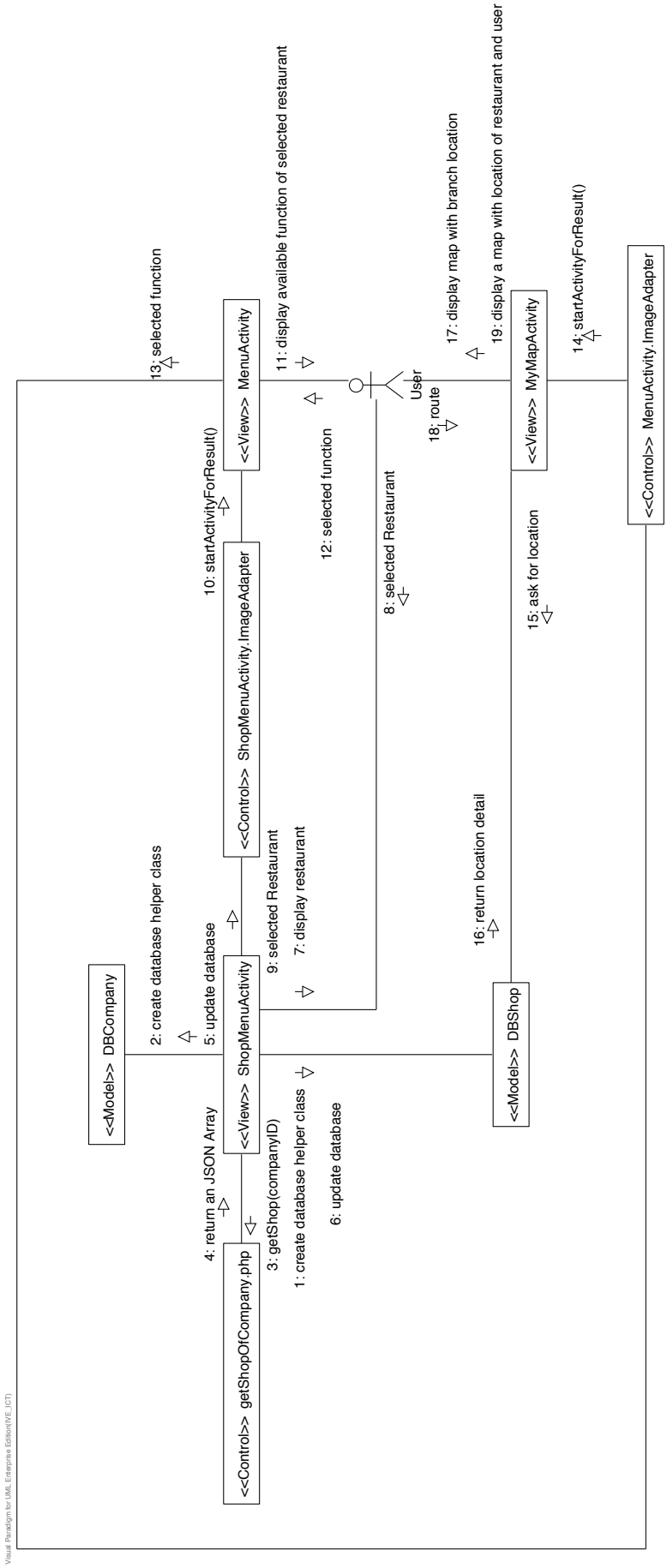
Visual Paradigm for UML Enterprise Edition(IVE_ICT)


User
<<View>> MenuActivity
<<Control>> MenuActivity.ImageAdapter
<<View>> FoodActivity
<<Model>> DBFood
<<Control>> getShopMenu.php
<<Model>> DBShop
<<View>> ShopMenuActivity
<<Control>> ShopMenuActivity.ImageAdapter
<<Model>> DBCompany
<<Control>> getShopOfCompany.php
<<View>> FoodDetailActivity
<<Control>> FoodActivity.ImageAdapter
1: create database helper class
2: create database helper class
3: getShop(companyID)
4: return an JSON Array
5: update database
6: update database
7: display restaurant
8: selected Restaurant
9: selected Restaurant
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function
13: selected function
14: startActivityForResult()
15: create database helper class
16: getFood
17: return an JSON Array
18: update database
19: display foodMenu
20: selected food
21: selected food
22: startActivityForResult()
23: display food
Visual Paradigm for UML Enterprise Edition(VE_ICT)

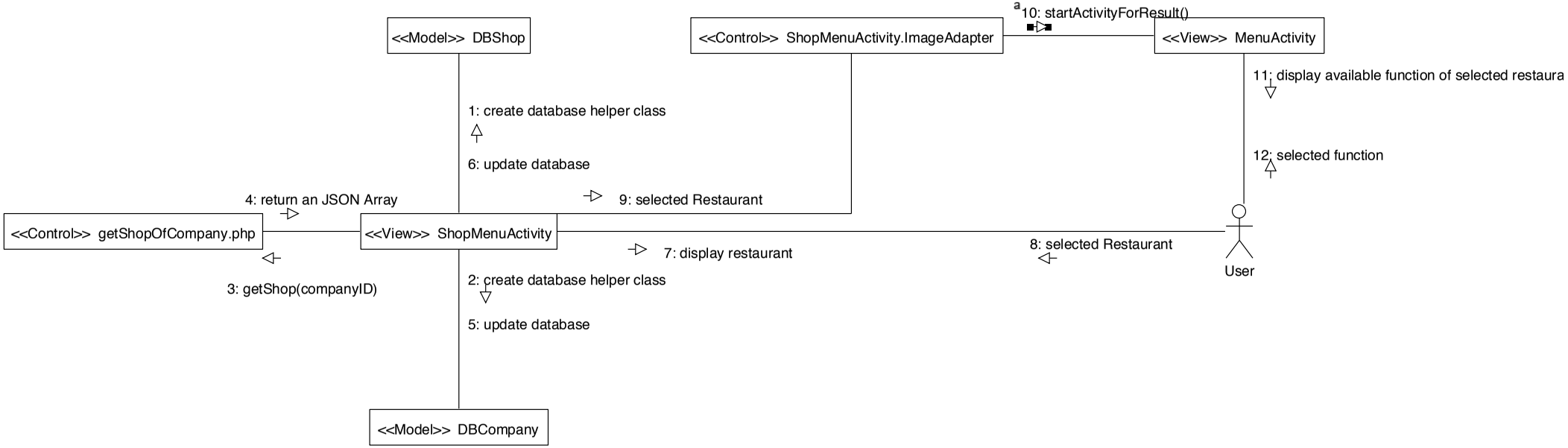

User
<<View>> MenuActivity
<<Control>> MenuActivity.ImageAdapter
<<View>> ViewCouponActivity
<<View>> GetCouponActivity
<<Control>> couponGet.php
<<Control>> ShopMenuActivity.ImageAdapter
<<View>> ShopMenuActivity
<<Model>> DBShop
<<Model>> DBCompany
<<Control>> getShopOfCompany.php
<<Control>> GetCouponBaseAdapter
1: create database helper class
2: create database helper class
3: getShop(companyID)
4: return an JSON Array
5: update database
6: update database
7: display restaurant
8: selected Restaurant
9: selected Restaurant
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function
13: selected function
14: startActivityForResult()
15: startActivityForResult()
16: View Coupon
17: return an JSON Array
18: display coupon
19: selected coupon
20: selected coupon
21: display successful message
Visual Paradigm for UML Enterprise Edition(IVE_ICT)


Visual Paradigm for UML Enterprise Edition(NVE_ICT)
<<Model>> DBCompany
<<Control>> getShopOfCompany.php
<<View>> ShopMenuActivity
<<Control>> ShopMenuActivity.ImageAdapter
<<View>> MenuActivity
User
<<View>> MyMapActivity
<<Control>> MenuActivity.ImageAdapter
<<Model>> DBShop
1: create database helper class
2: create database helper class
3: getShop(companyID)
4: return an JSON Array
5: update database
6: update database
7: display restaurant
8: selected Restaurant
9: selected Restaurant
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function
13: selected function
14: startActivityForResult()
15: ask for location
16: return location detail
17: display map with branch location

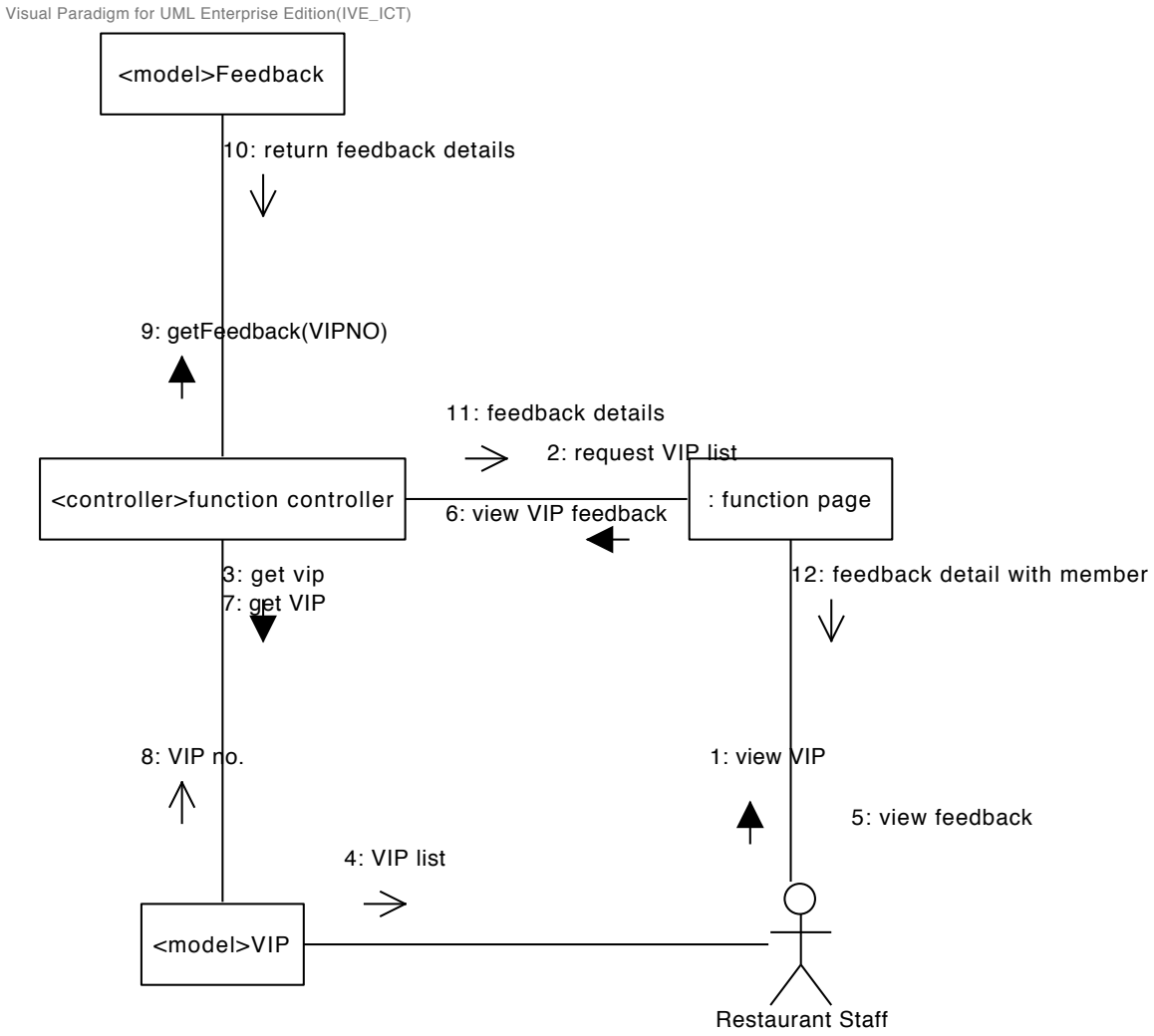

Visual Paradigm for UML Enterprise Edition(VE_ICT)
<<Control>> getShopOfCompany.php
4: return an JSON Array
3: getShop(companyID)
<<Model>> DBShop
1: create database helper class
6: update database
<<View>> ShopMenuActivity
2: create database helper class
5: update database
<<Model>> DBCompany
7: display restaurant
8: selected Restaurant
User
9: selected Restaurant
<<Control>> ShopMenuActivity.ImageAdapter
10: startActivityForResult()
11: display available function of selected restaurant
12: selected function
<<View>> MenuActivity
13: selected function
<<Control>> MenuActivity.ImageAdapter
14: startActivityForResult()
19: display news
<<View>> NewsActivity
15: create database helper class
<<Model>> DBNew
18: update database
16: getNews
17: return an JSON Array
<<Control>> getNews.php

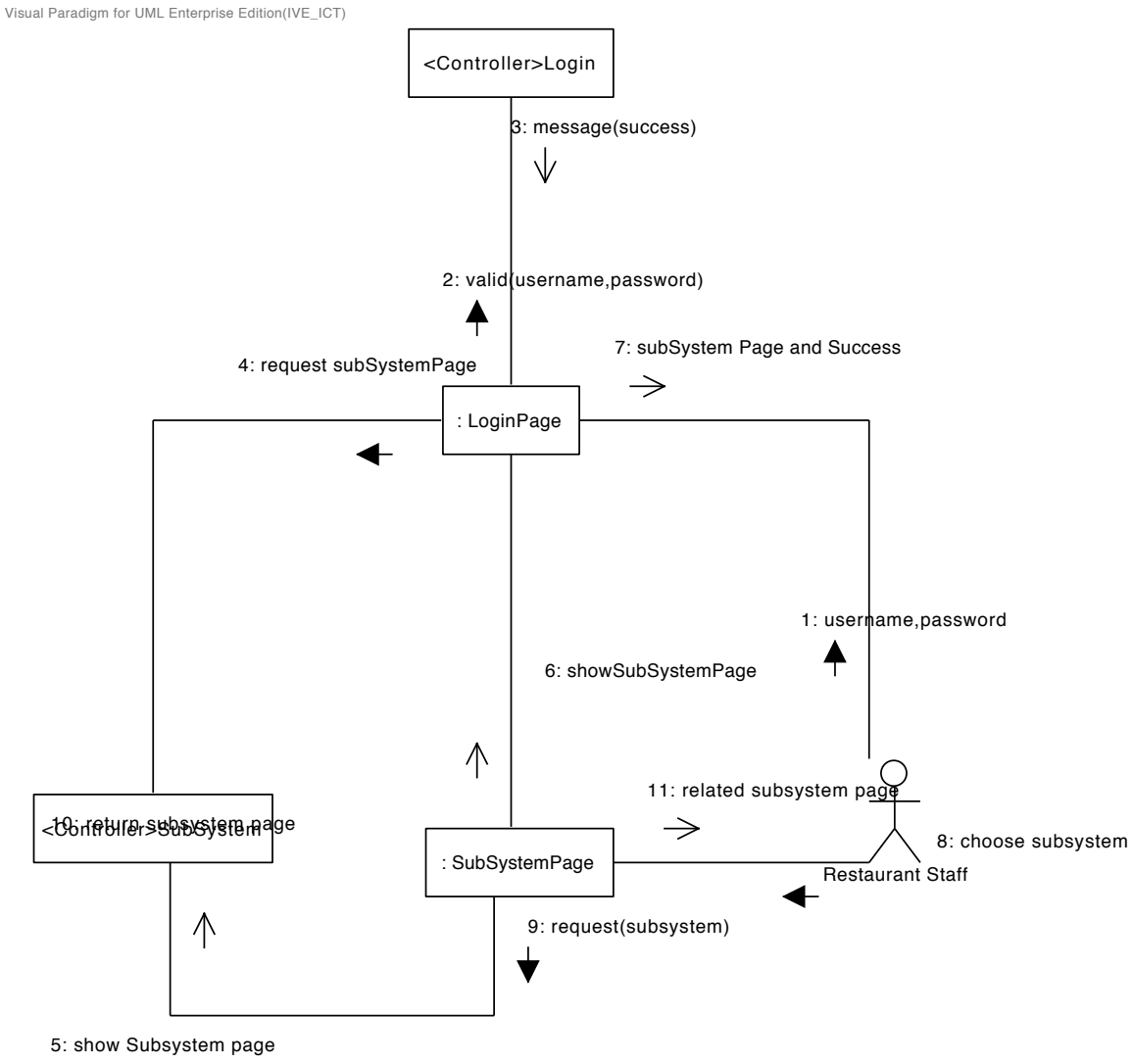

Visual Paradigm for UML Enterprise Edition(VE_ICT)
<<Control>> getShopOfCompany.php
4: return an JSON Array
3: getShop(companyID)
1: create database helper class
6: update database
<<Model>> DBCompany
2: create database helper class
<<View>> ShopMenuActivity
5: update database
9: selected Restaurant
7: display restaurant
<<Control>> ShopMenuActivity.ImageAdapter
10: startActivityForResult()
<<View>> MenuActivity
13: selected function
11: display available function of selected restaurant
12: selected function
8: selected Restaurant
User
18: route
17: display map with branch location
19: display a map with location of restaurant and user
<<View>> MyMapActivity
15: ask for location
16: return location detail
<<Model>> DBShop
14: startActivityForResult()
<<Control>> MenuActivity.ImageAdapter

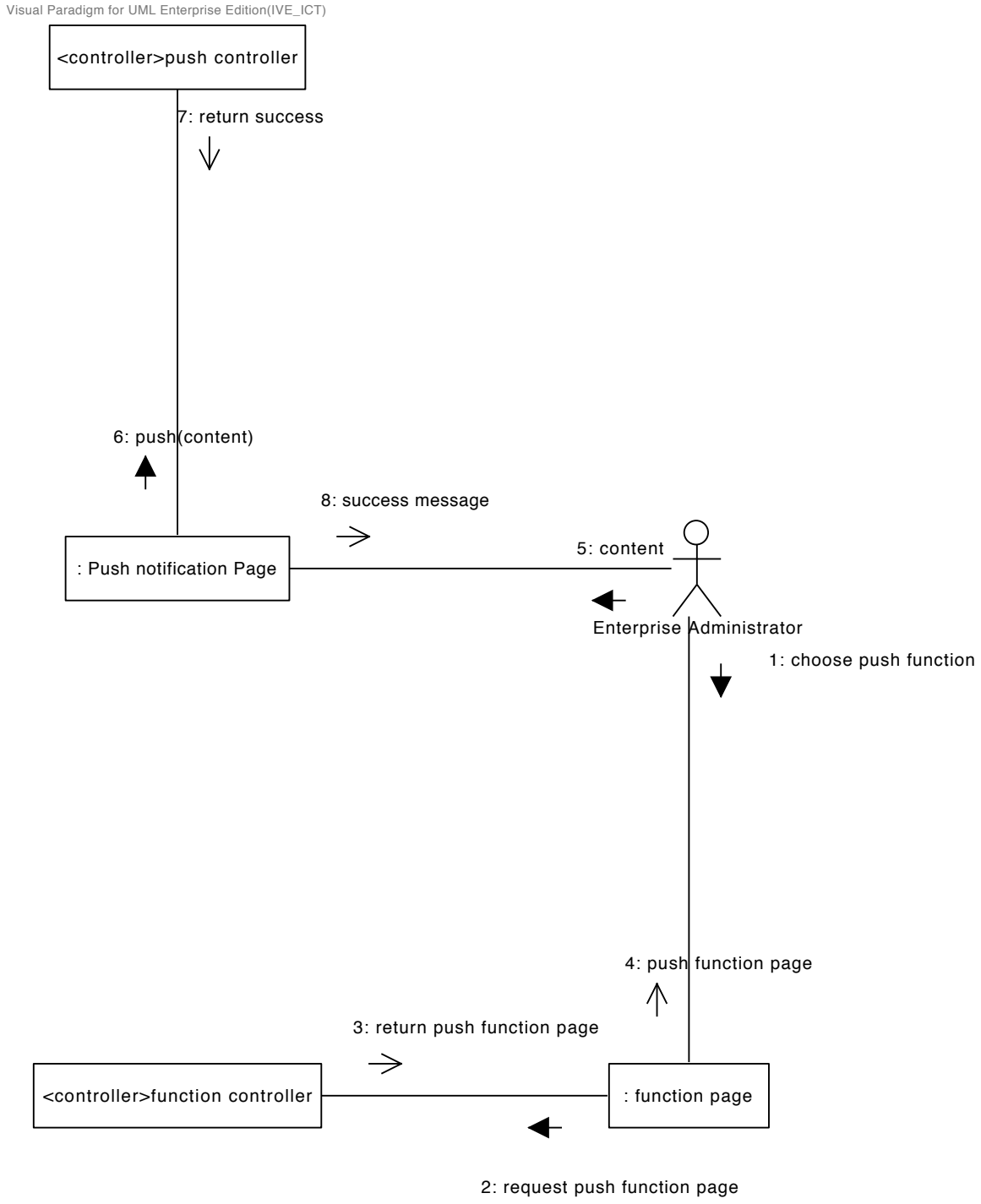
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

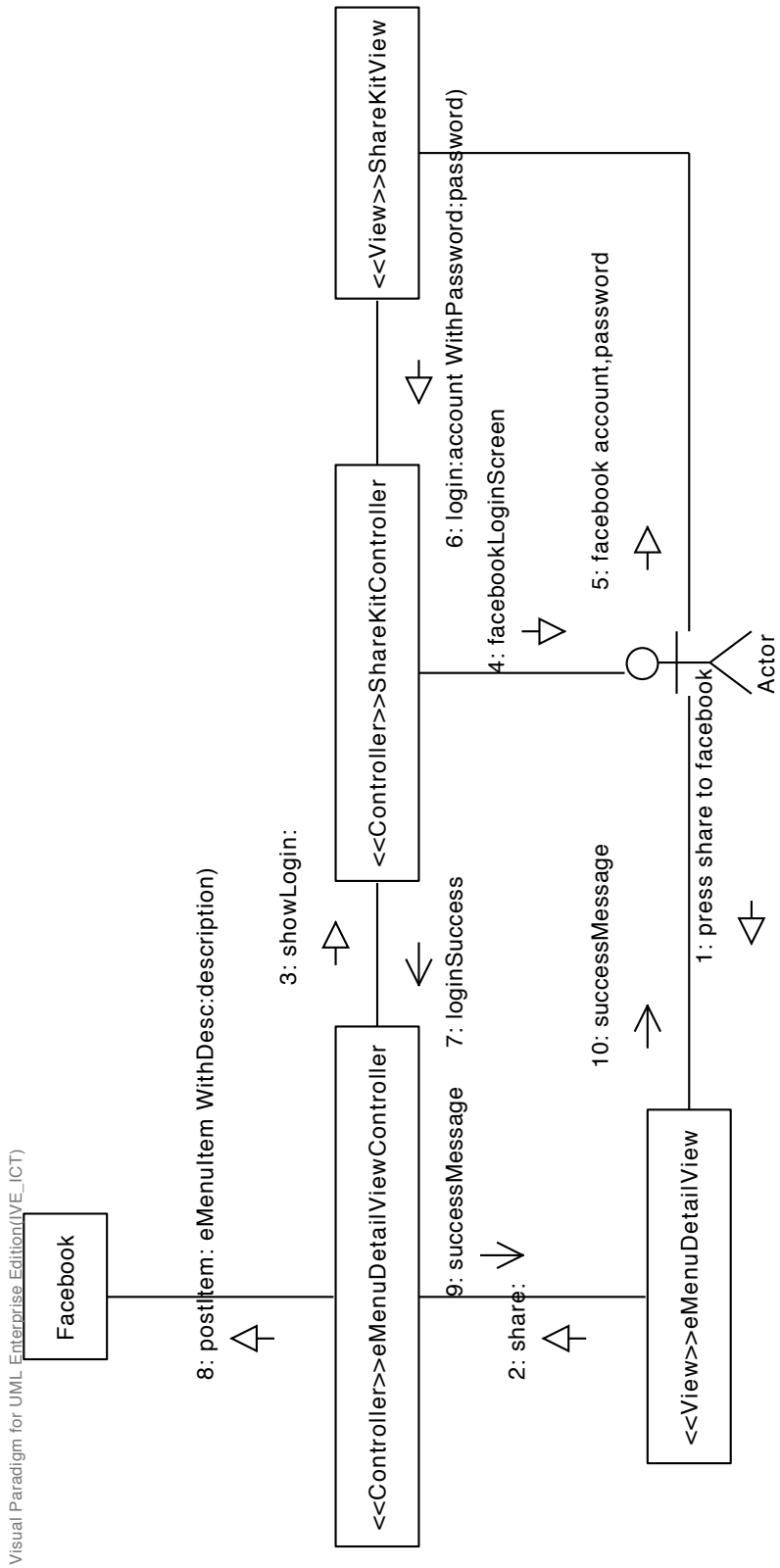


Object Collaboration Diagram (Website)

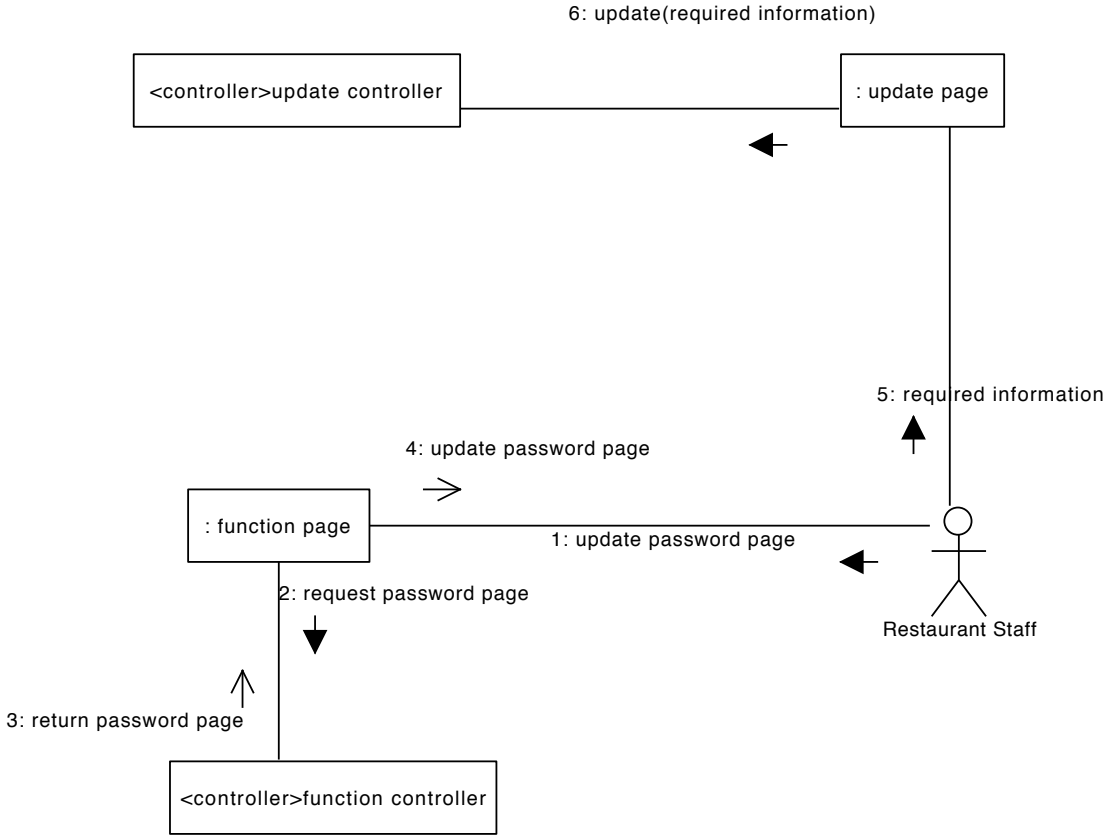



Visual Paradigm for UML Enterprise Edition(IVE_ICT)
<Controller>Login
3: message(success)
2: valid(username,password)
4: request subSystemPage
7: subSystem Page and Success
: LoginPage
1: username,password
6: showSubSystemPage
11: related subsystem page
10: return subsystem page
<Controller>SubSystem
: SubSystemPage
8: choose subsystem
Restaurant Staff
9: request(subsystem)
5: show Subsystem page

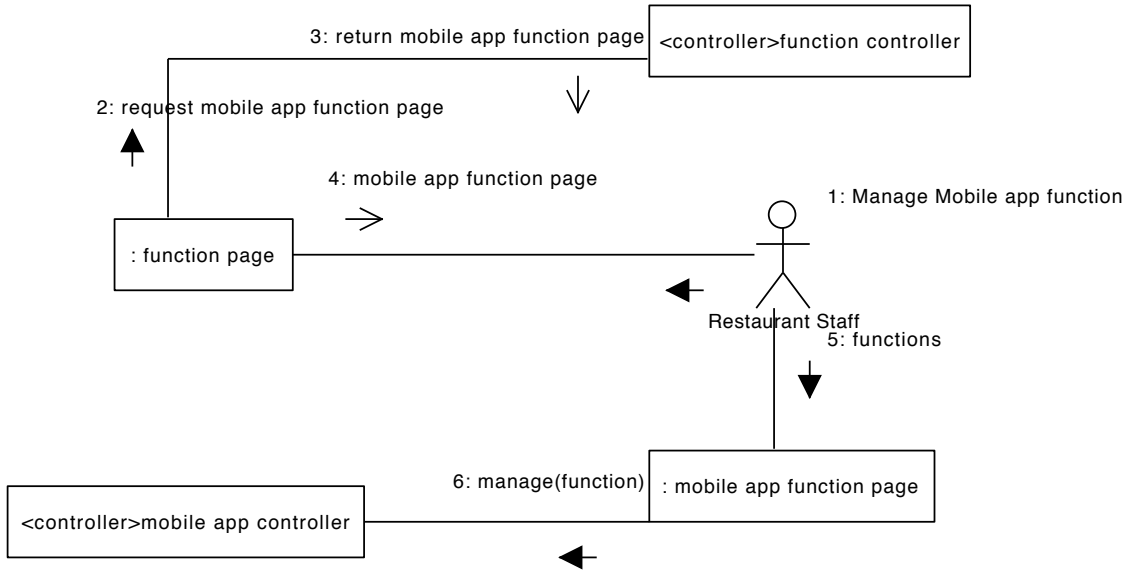

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
<controller>push controller
7: return success
6: push(content)
8: success message
: Push notification Page
5: content
Enterprise Administrator
1: choose push function
4: push function page
3: return push function page
<controller>function controller
: function page
2: request push function page

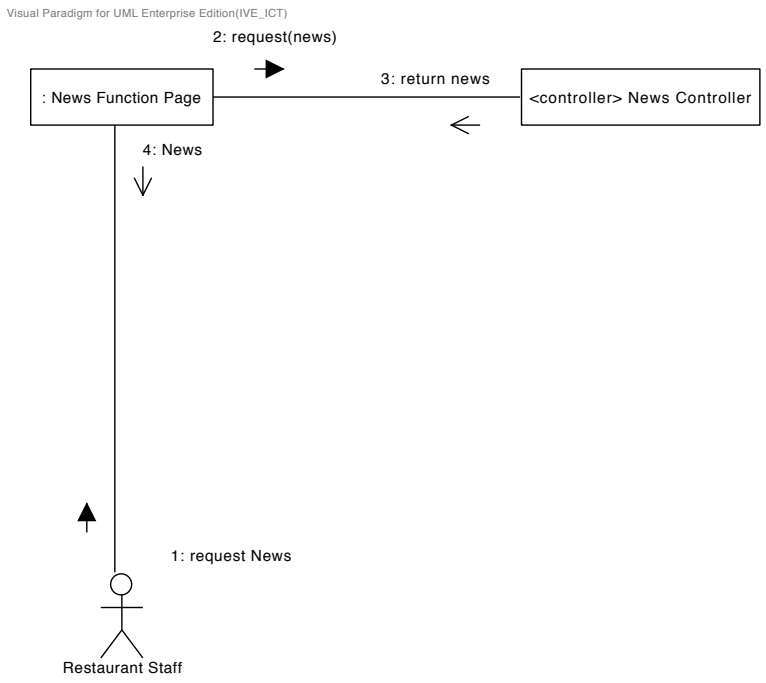

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
Facebook
8: postItem: eMenuItem WithDesc:description)
<<Controller>>eMenuDetailViewController
3: showLogin:
<<Controller>>ShareKitController
<<View>>ShareKitView
9: successMessage
7: loginSuccess
6: login:account WithPassword:password)
2: share:
4: facebookLoginScreen
<<View>>eMenuDetailView
10: successMessage
5: facebook account,password
1: press share to facebook
Actor

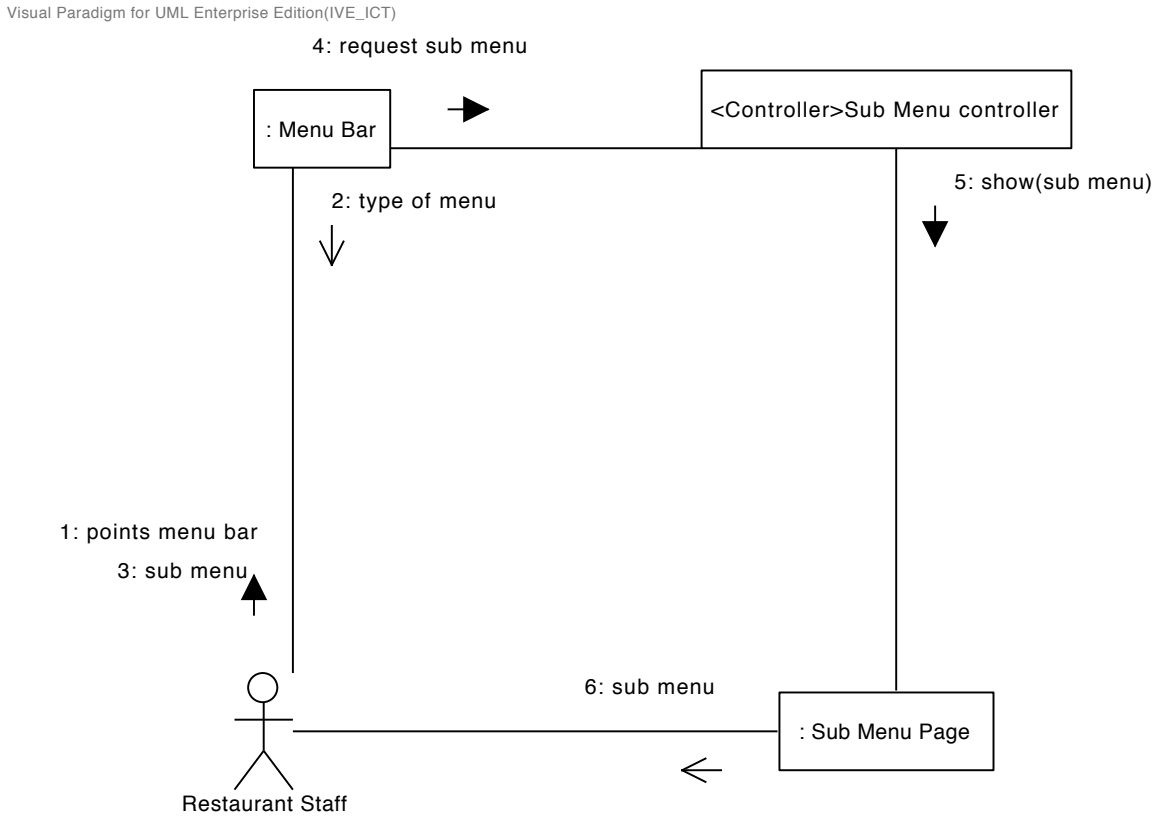
Visual Paradigm for UML Enterprise Edition(IVE_ICT)

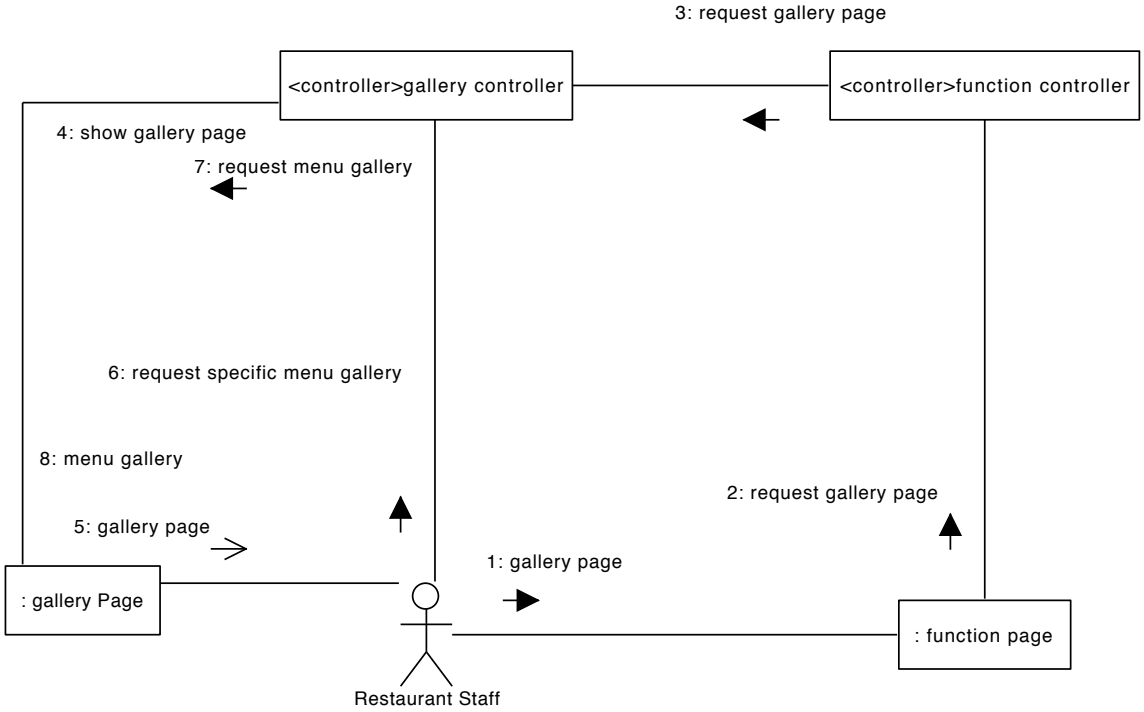


Visual Paradigm for UML Enterprise Edition(IVE_ICT)

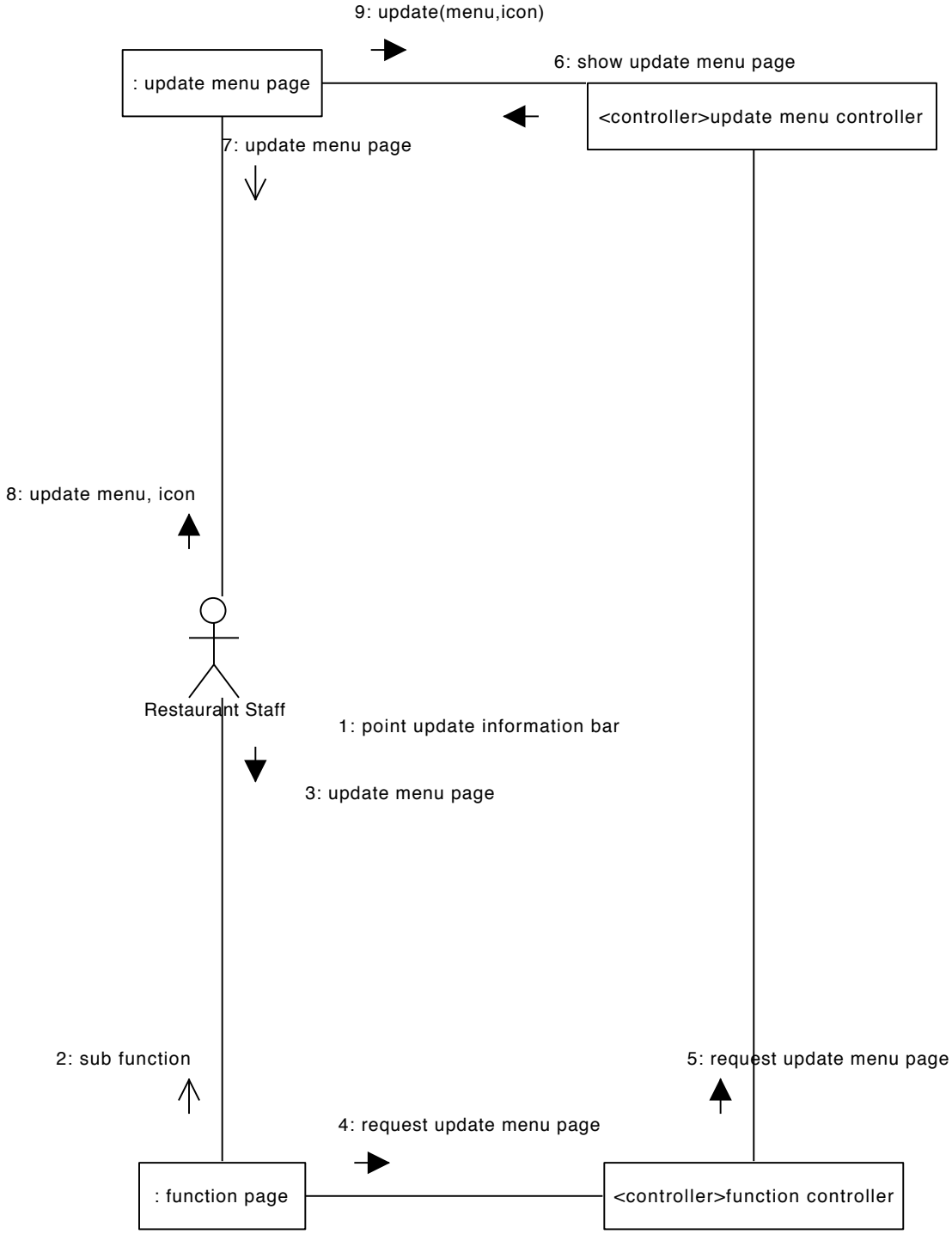


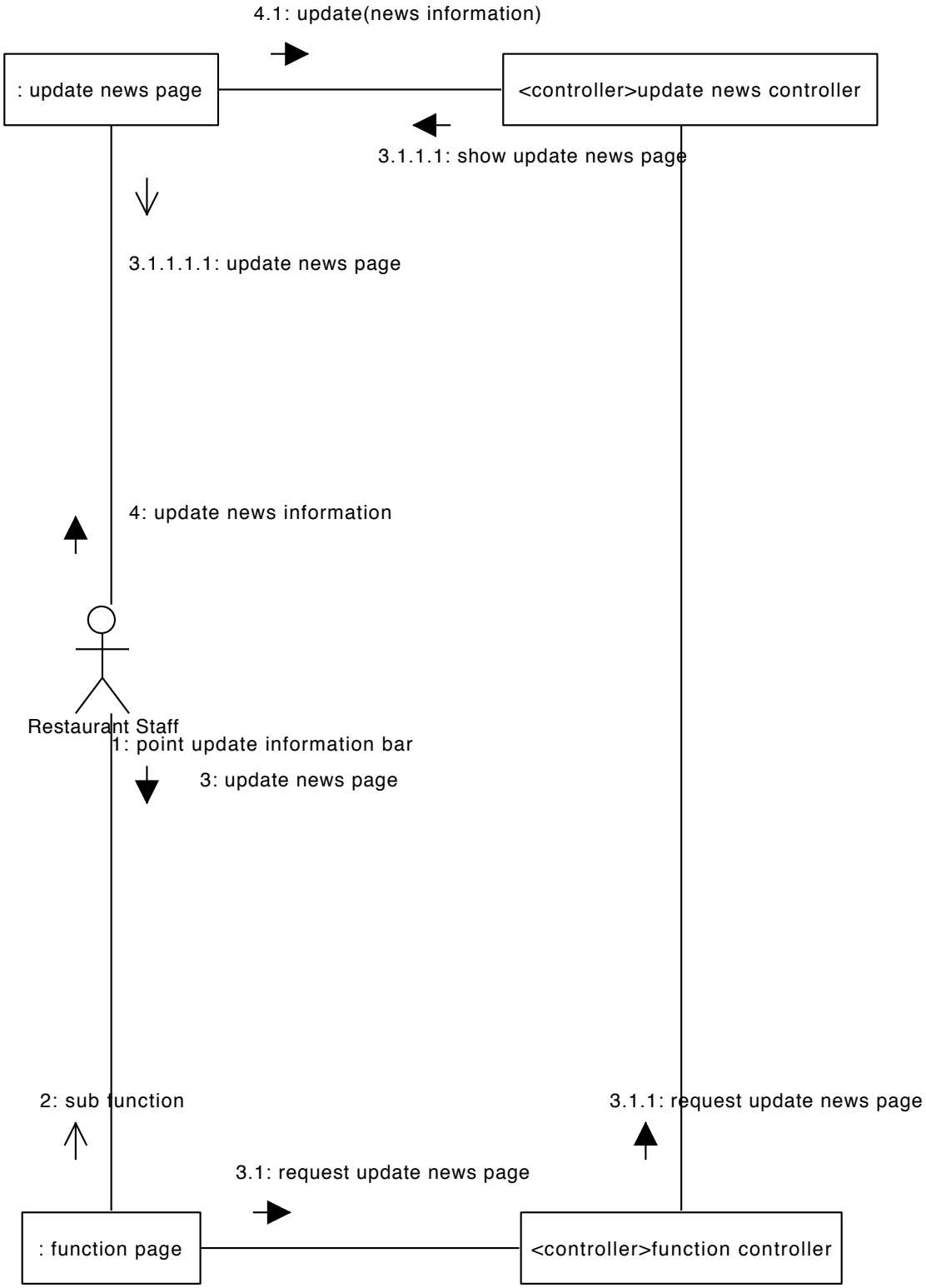

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
2: request(news)
: News Function Page
3: return news
<controller> News Controller
4: News
1: request News
Restaurant Staff

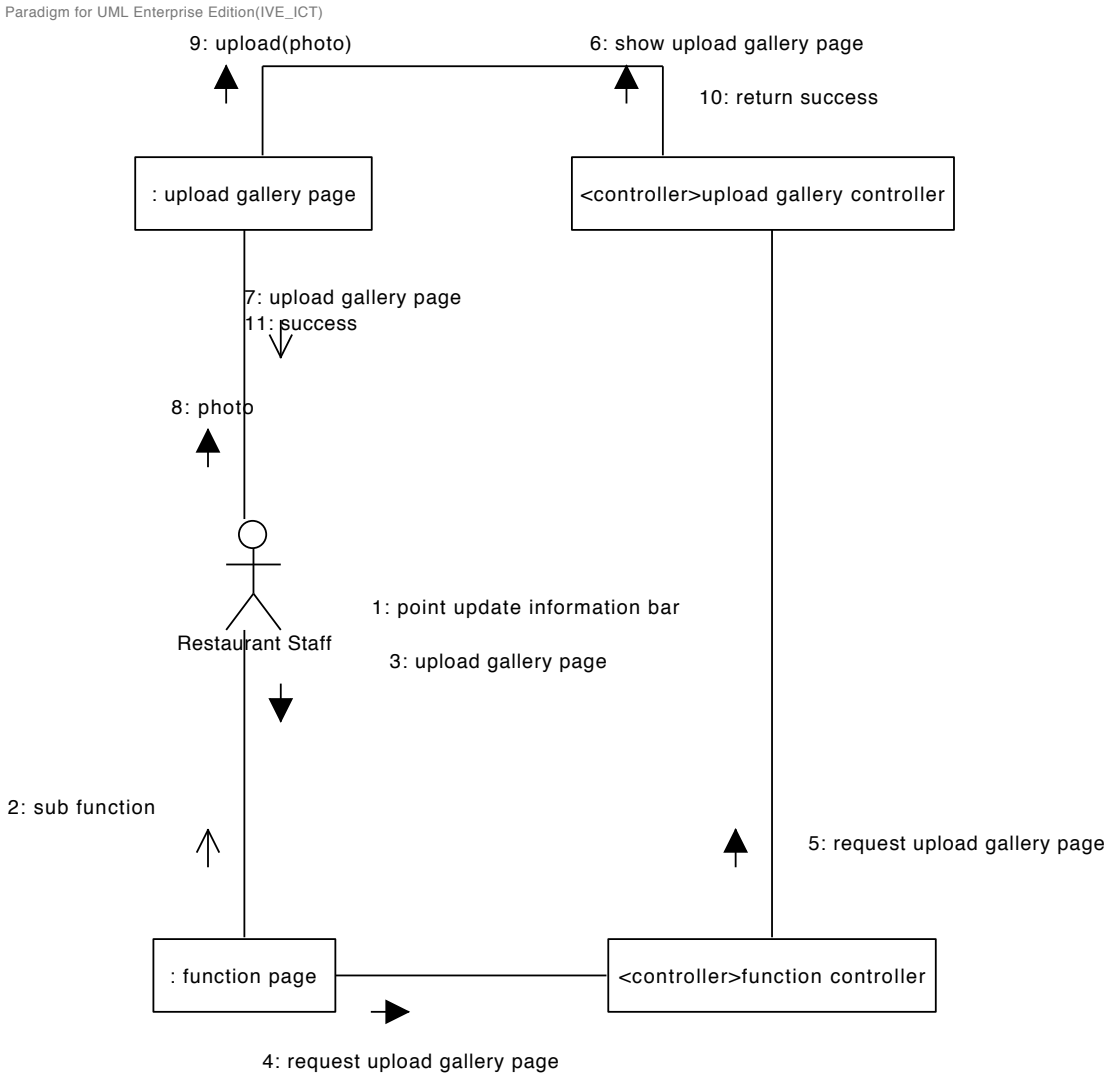

Visual Paradigm for UML Enterprise Edition(IVE_ICT)
4: request sub menu
: Menu Bar
<Controller>Sub Menu controller
5: show(sub menu)
2: type of menu
1: points menu bar
3: sub menu
6: sub menu
: Sub Menu Page
Restaurant Staff


3: request gallery page
<controller>gallery controller
<controller>function controller
4: show gallery page
7: request menu gallery
6: request specific menu gallery
8: menu gallery
2: request gallery page
5: gallery page
1: gallery page
: gallery Page
: function page
Restaurant Staff

Visual Paradigm for UML Enterprise Edition(IVE_ICT)




4.1: update(news information)
: update news page
<controller>update news controller
3.1.1.1: show update news page
3.1.1.1.1: update news page
4: update news information
Restaurant Staff
1: point update information bar
3: update news page
2: sub function
3.1.1: request update news page
3.1: request update news page
: function page
<controller>function controller


Visual Paradigm for UML Enterprise Edition(IVE_ICT)
9: upload(photo)
6: show upload gallery page
10: return success
: upload gallery page
<controller>upload gallery controller
7: upload gallery page
11: success
8: photo
Restaurant Staff
1: point update information bar
3: upload gallery page
2: sub function
5: request upload gallery page
: function page
<controller>function controller
4: request upload gallery page

DOCUMENTATION FOR DETAILED DESIGN

UI 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.

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.

Mobile App user interface design

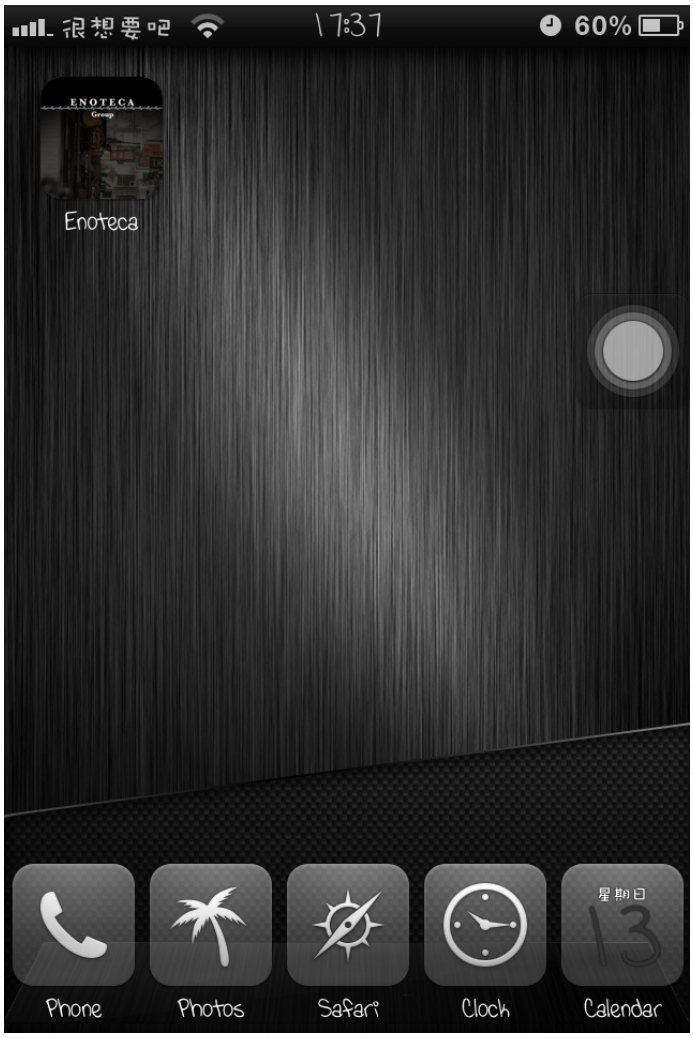


Figure 97. Mobile App icon

The icon “Enoteca” is our application in mobile. This icon design is attractive and related to the company mentor restaurant. If the company put the mobile app on the Apple App Store, the user can download it in the App Store.

Figure 98. Application Main Menu

All functions are 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. This page is the restaurant list; the user can see the icon of the restaurant easily.

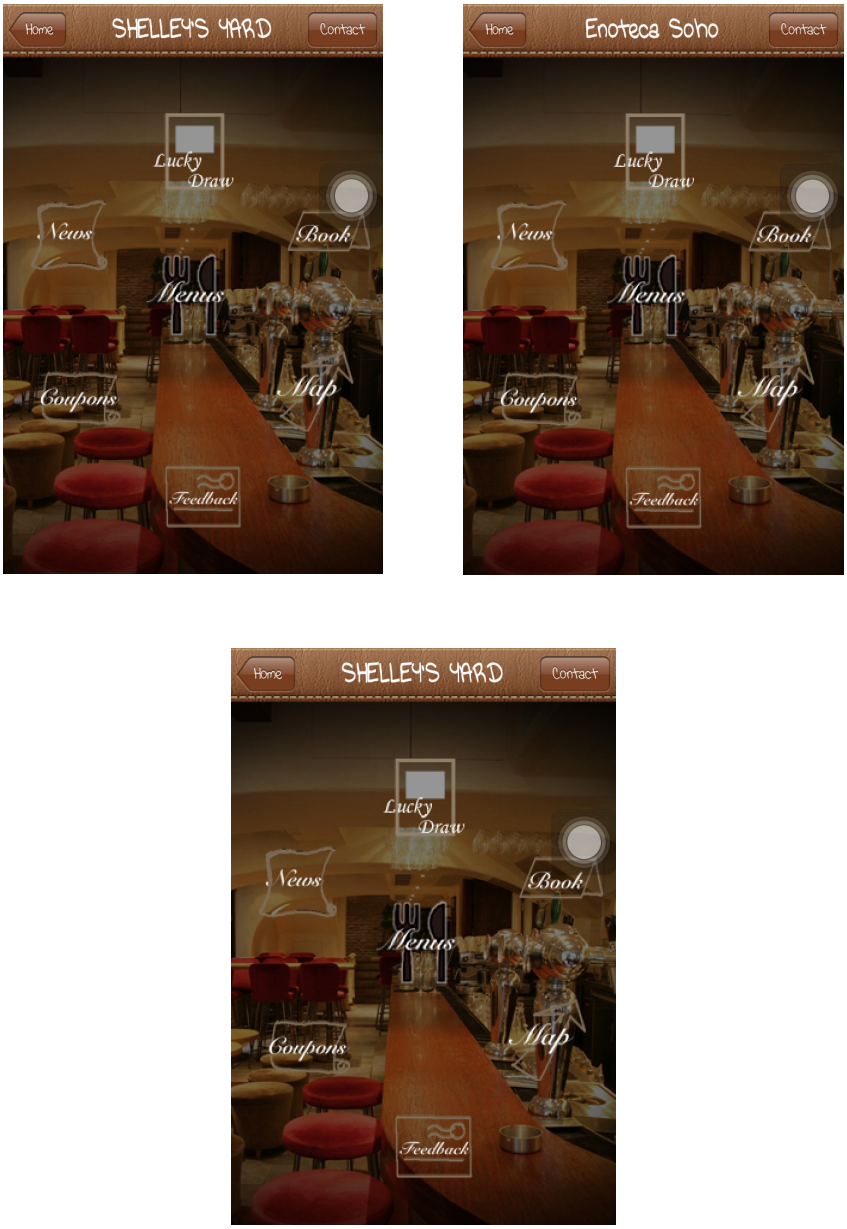


Figure 99. Main Interface

Each branch shop under the Enoteca group will have different functions and containing different data, as the diagram show, it use a 3 x 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.

Figure 100. User interface News

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.

Figure 101. Restaurant Information interface

This is the restaurant information interface. It is a table view and divided into different section with shop name, operating hours, phone number and address. The customer can quickly find out the information of that related company if they would like to go to there or require any services.

Figure 102. Restaurant 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.

Figure 103. Restaurant 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.

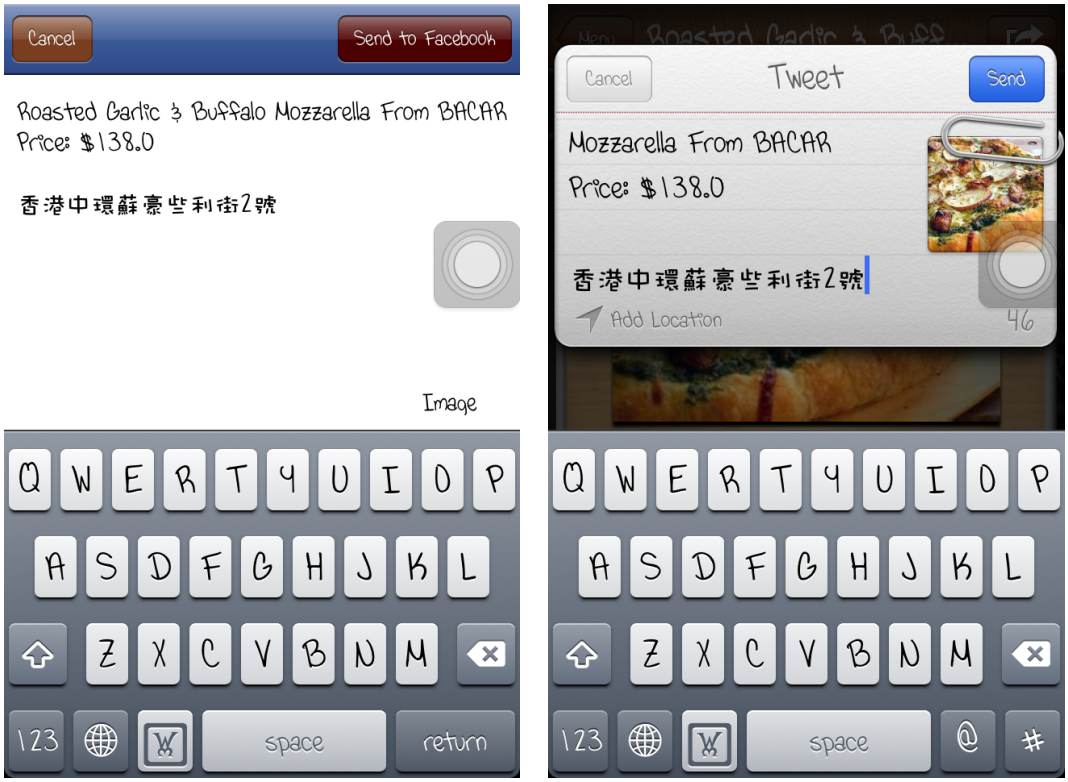


Figure 104. Sharing to Social Network

This is the user post the food to social network 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 Social Network account.

Figure 105. 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.

Figure 106. View booking status interface

This interface let the customer to review her booking details and status. The information includes the date, time, number of people and booking status.

Website User Interface Design

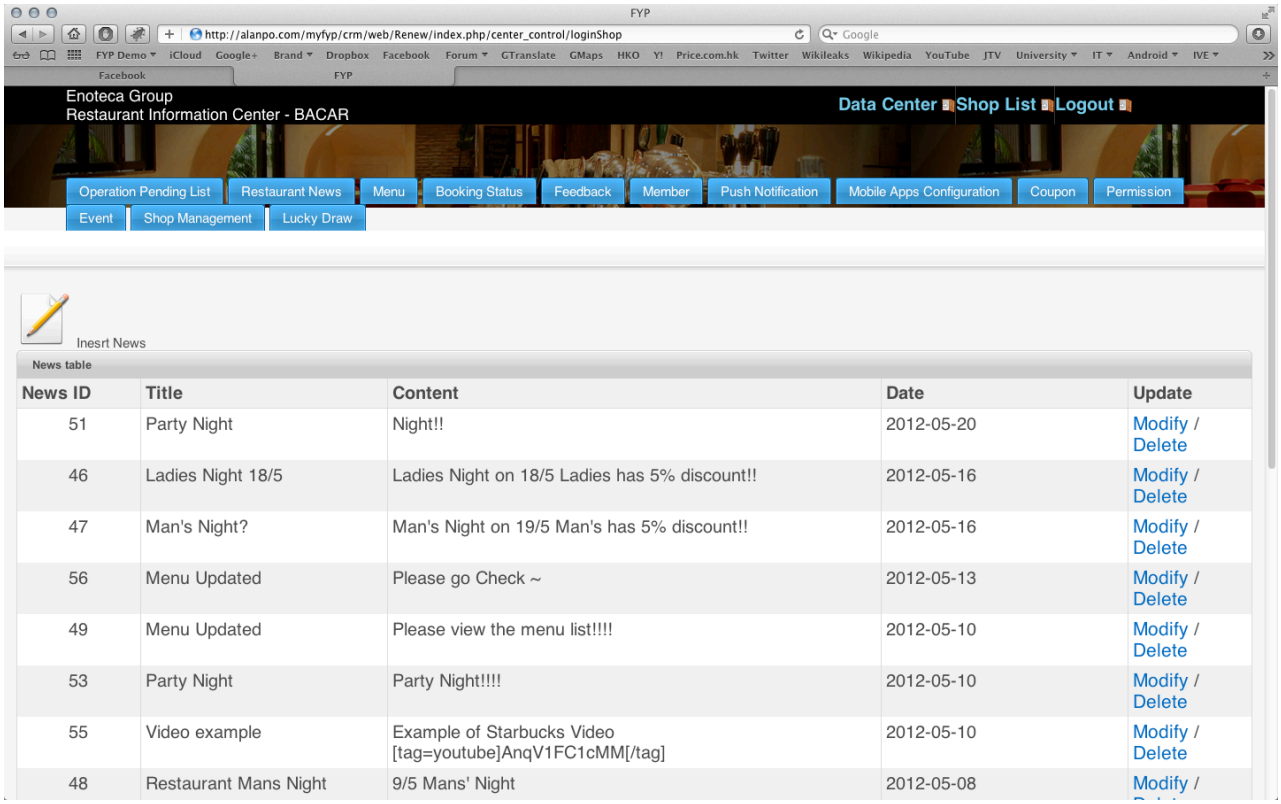


Figure 107. Website index page

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.

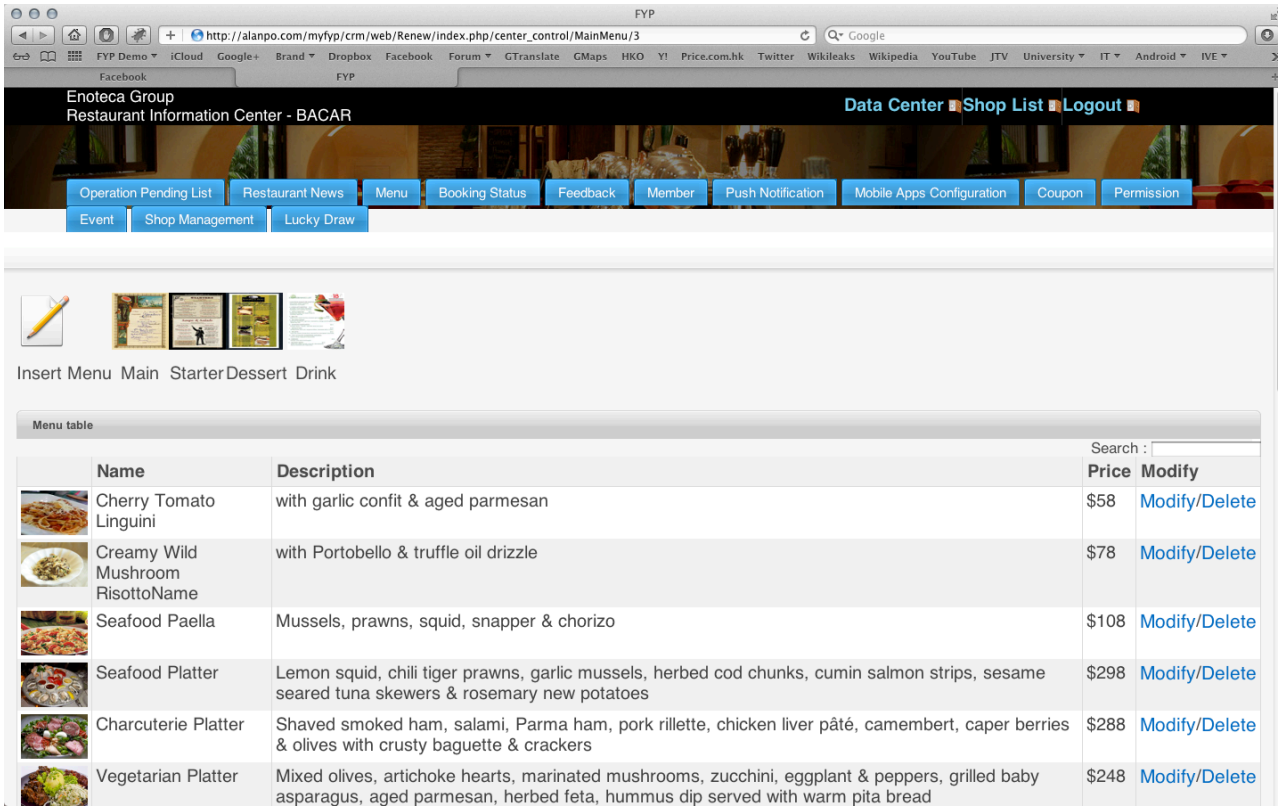


Figure 108. Website food menu

This is the food menu design. This first line is the food category. The menu table contains the information of each food including name, description, price and image.

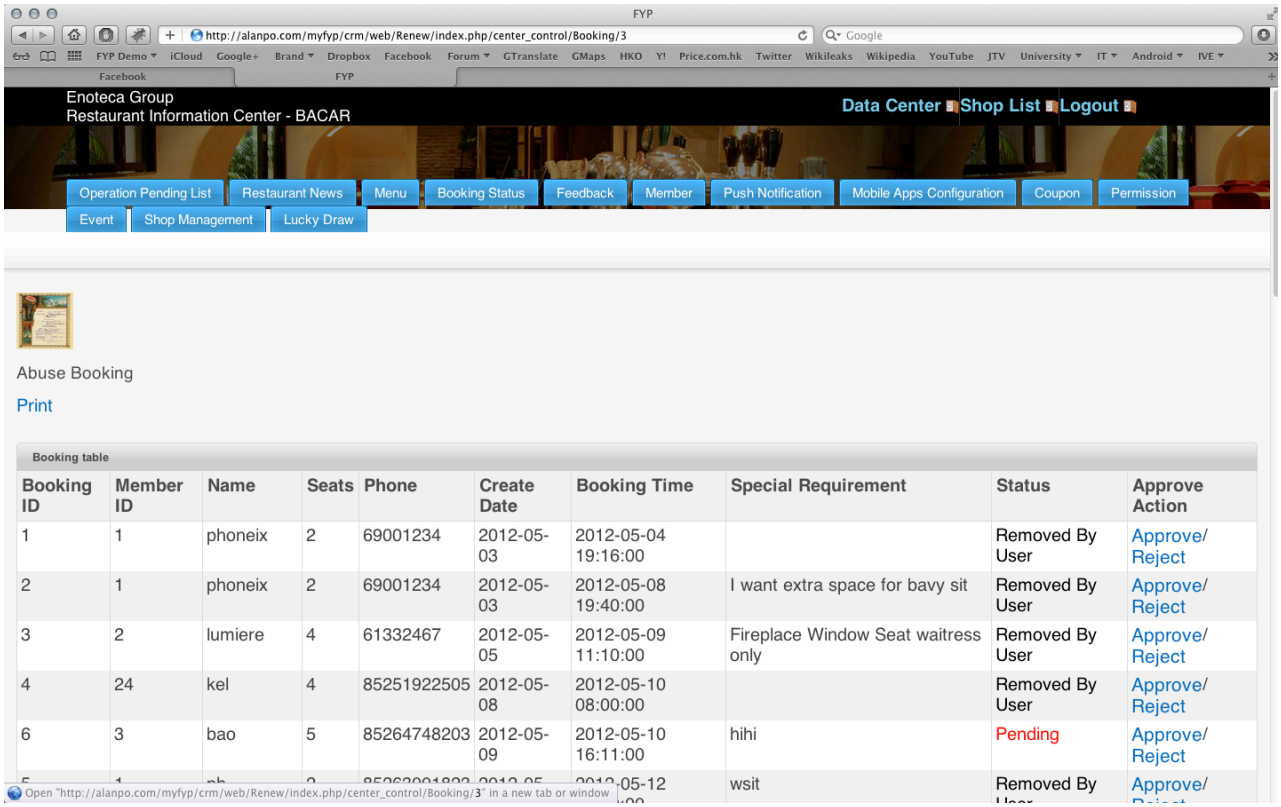


Figure 109. Website booking list

This is the booking list in the website. The booking list includes the booking ID from the restaurant, member ID if the customer already registered as a VIP, name of the customer, phone number, date and time, any special requirements, booking status and the approve function for the staff.

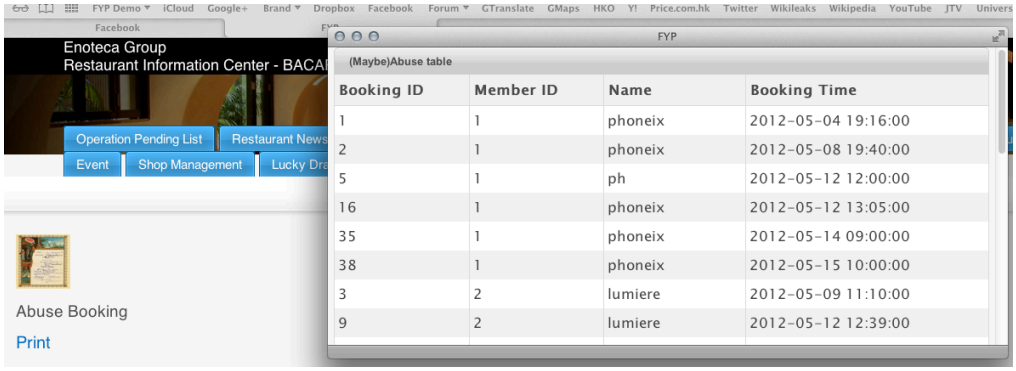


Figure 110. Abuse booking user

A abuse booking list is designed to trace the booking if the user makes a coupon of booking in the same day.

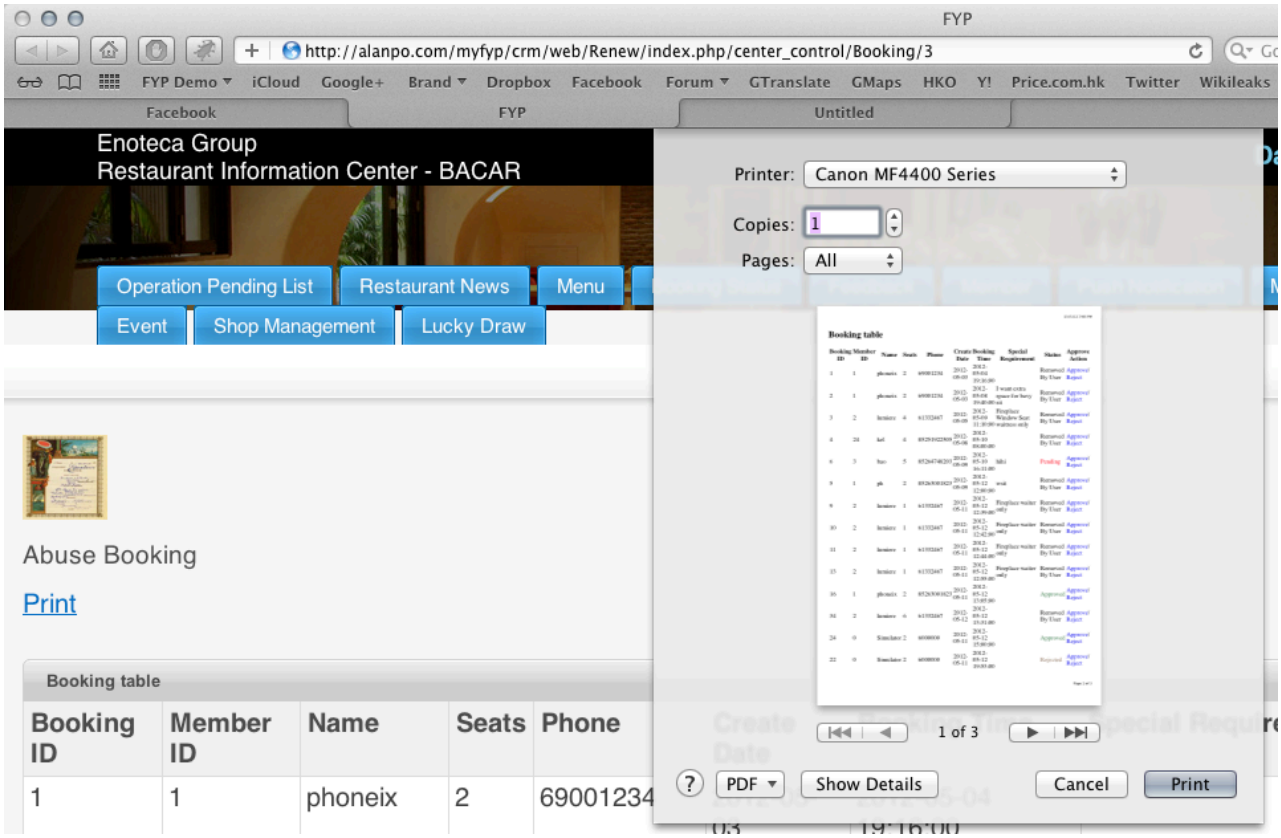


Figure 111. Print booking list

This is the printed version of the booking list. It is designed for the restaurant staff to print out in an A4 paper to follow up with the customers or manage the tables in the restaurant.

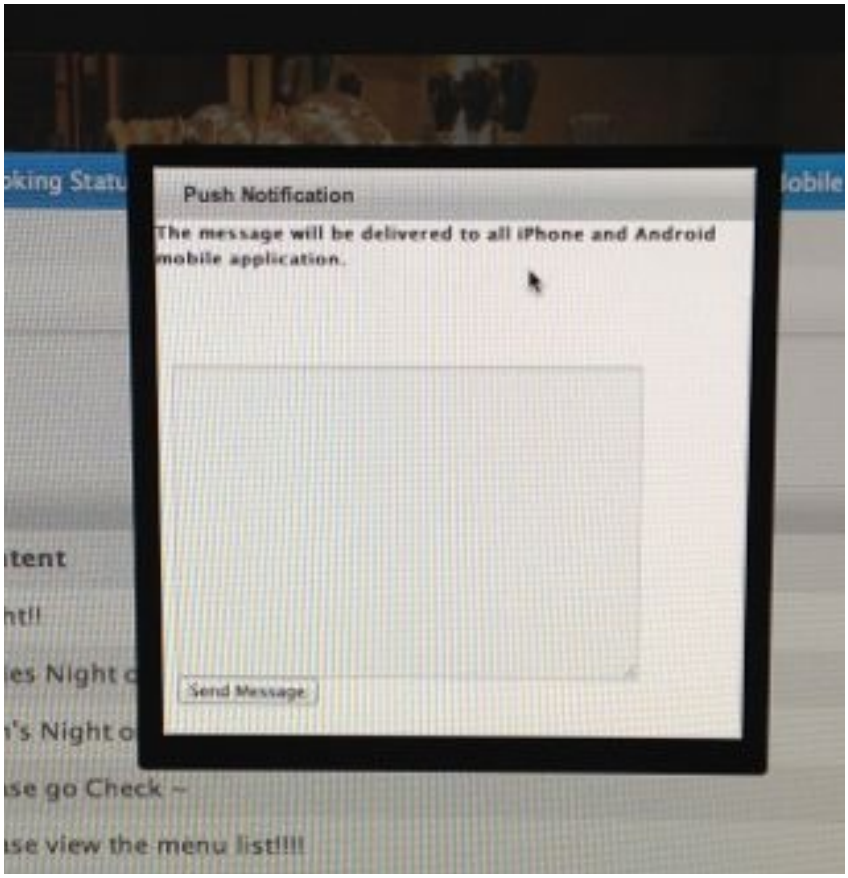


Figure 112. Website push notification

This is the push notification dialog for the user to input message and push to all mobile devices.

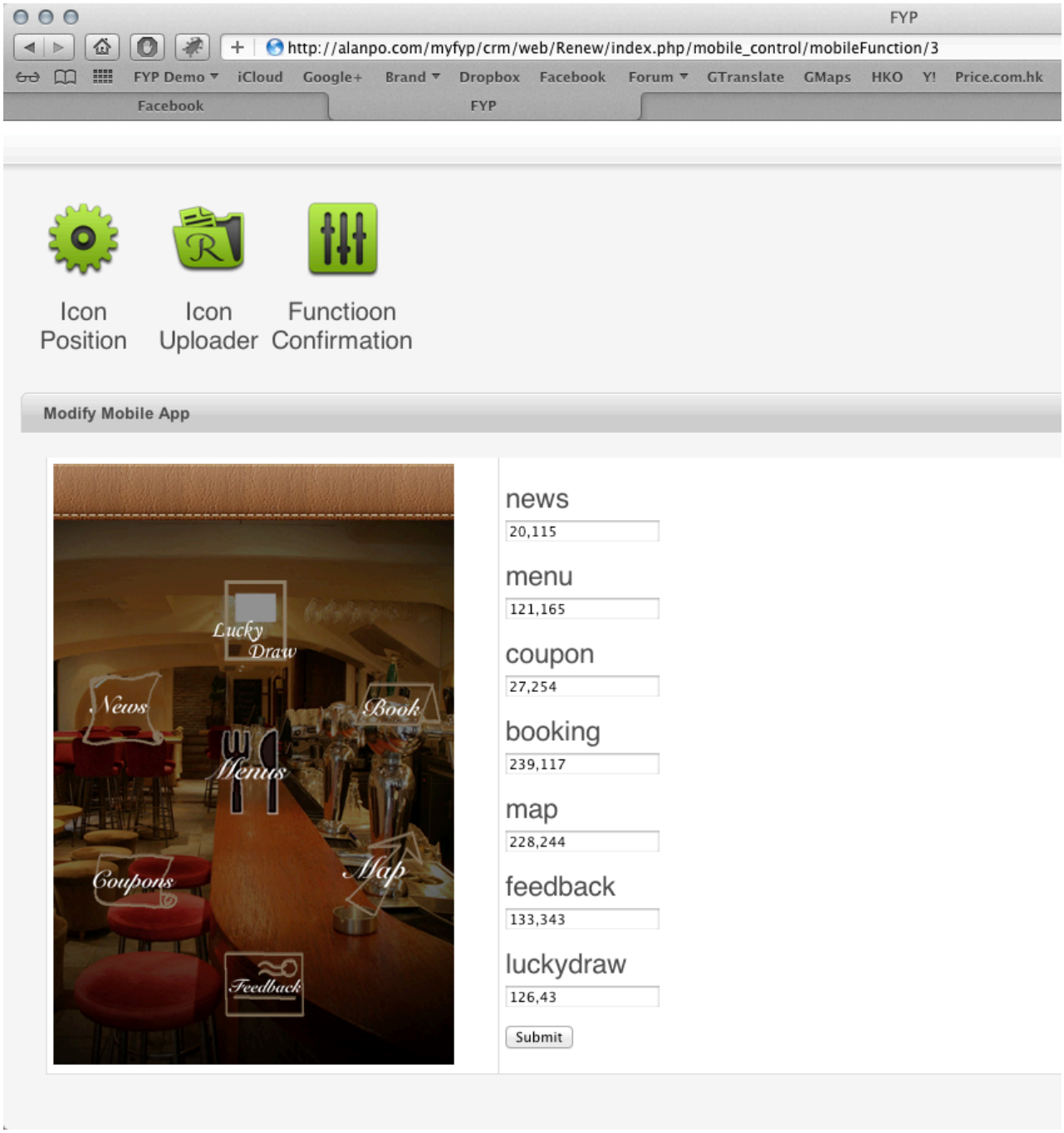


Figure 113. Website mobile configuration

This is the mobile app simulator that let the user to make modification on the position of the mobile app. This action makes anything become simple.

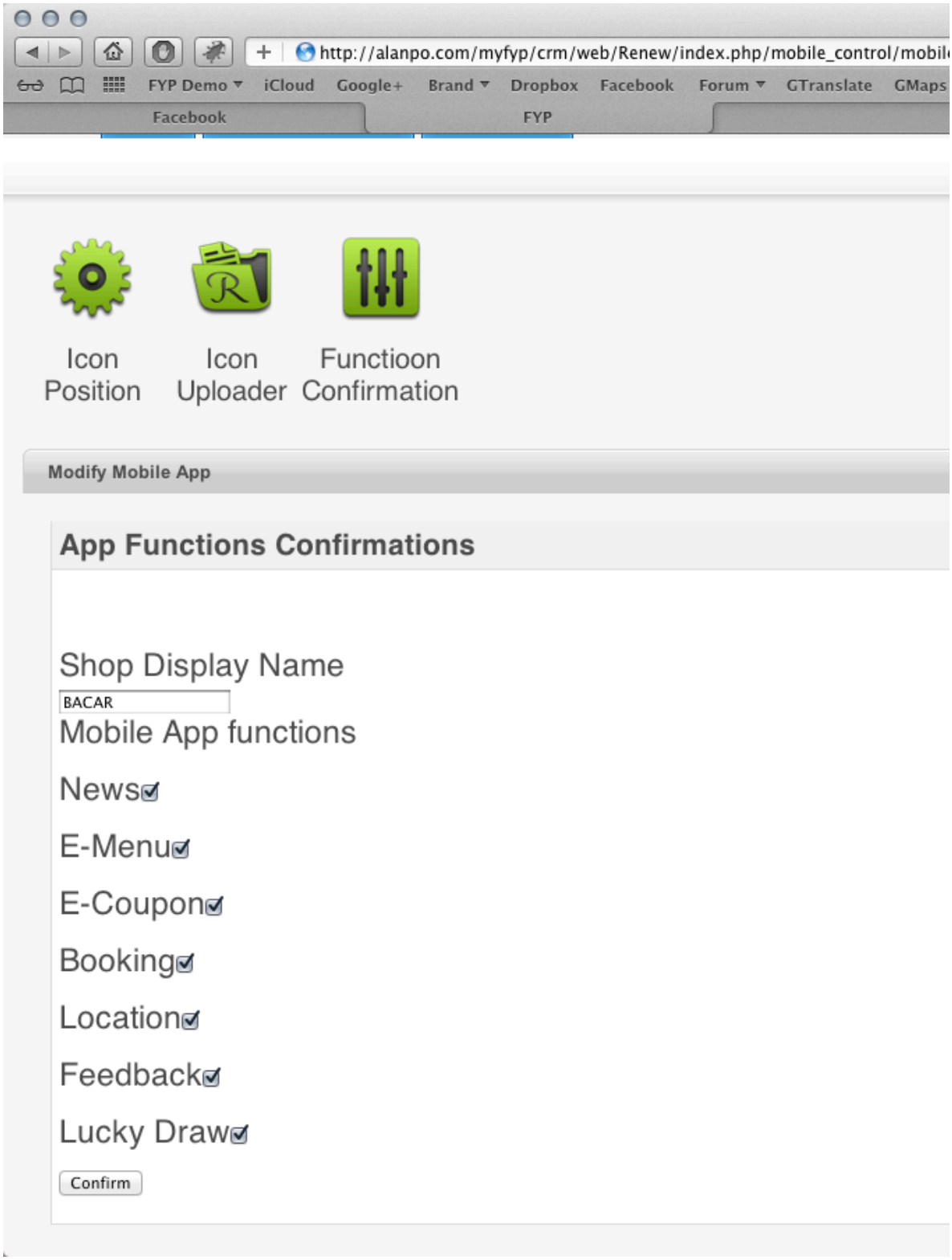


Figure 114.Website control function on mobile

This is the mobile function page that let the user to make modification on the functions allowed to the customer or not.

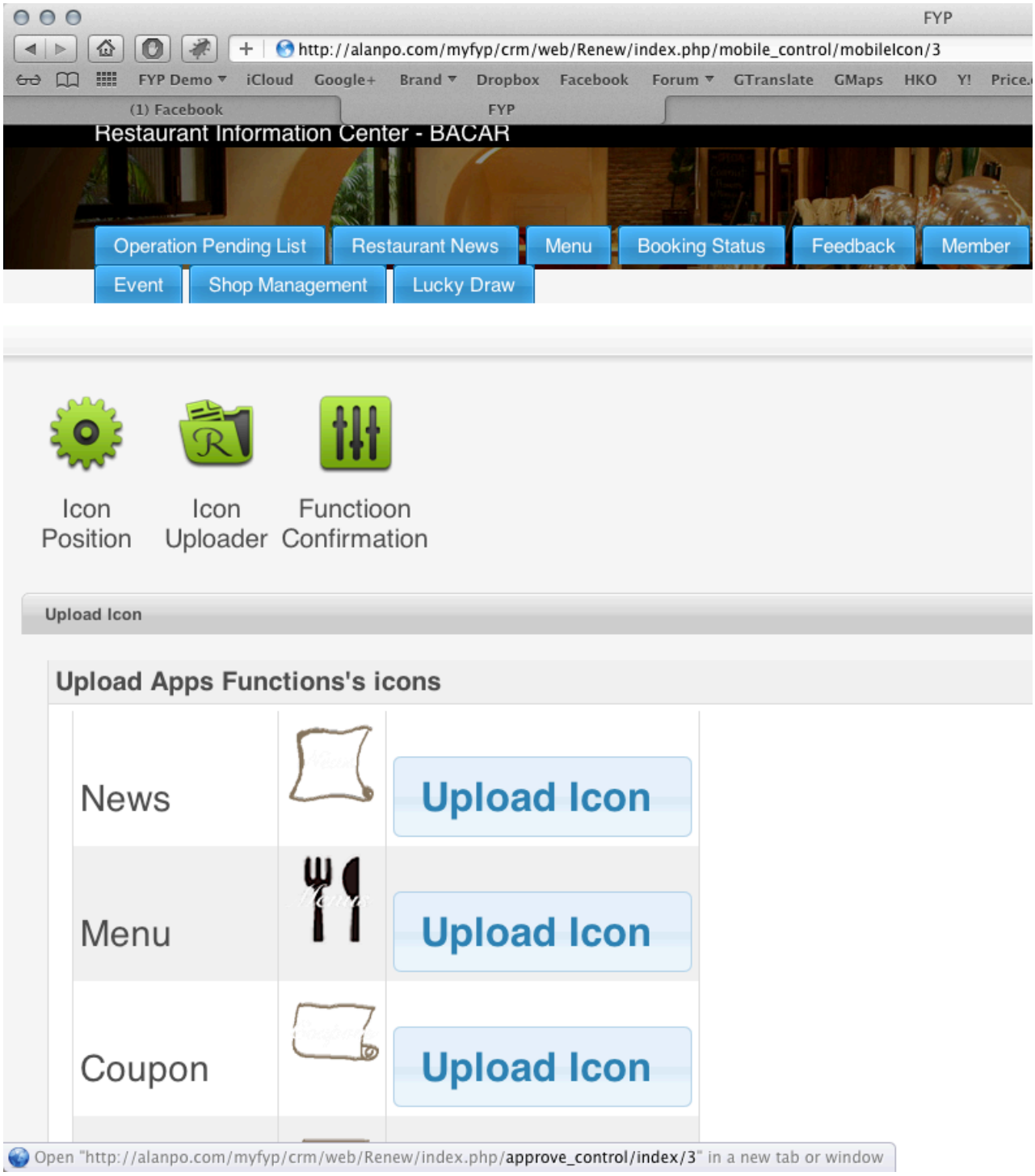


Figure 115. Website change mobile app icon

This is designed for the user to change the icon of the mobile app in the website.

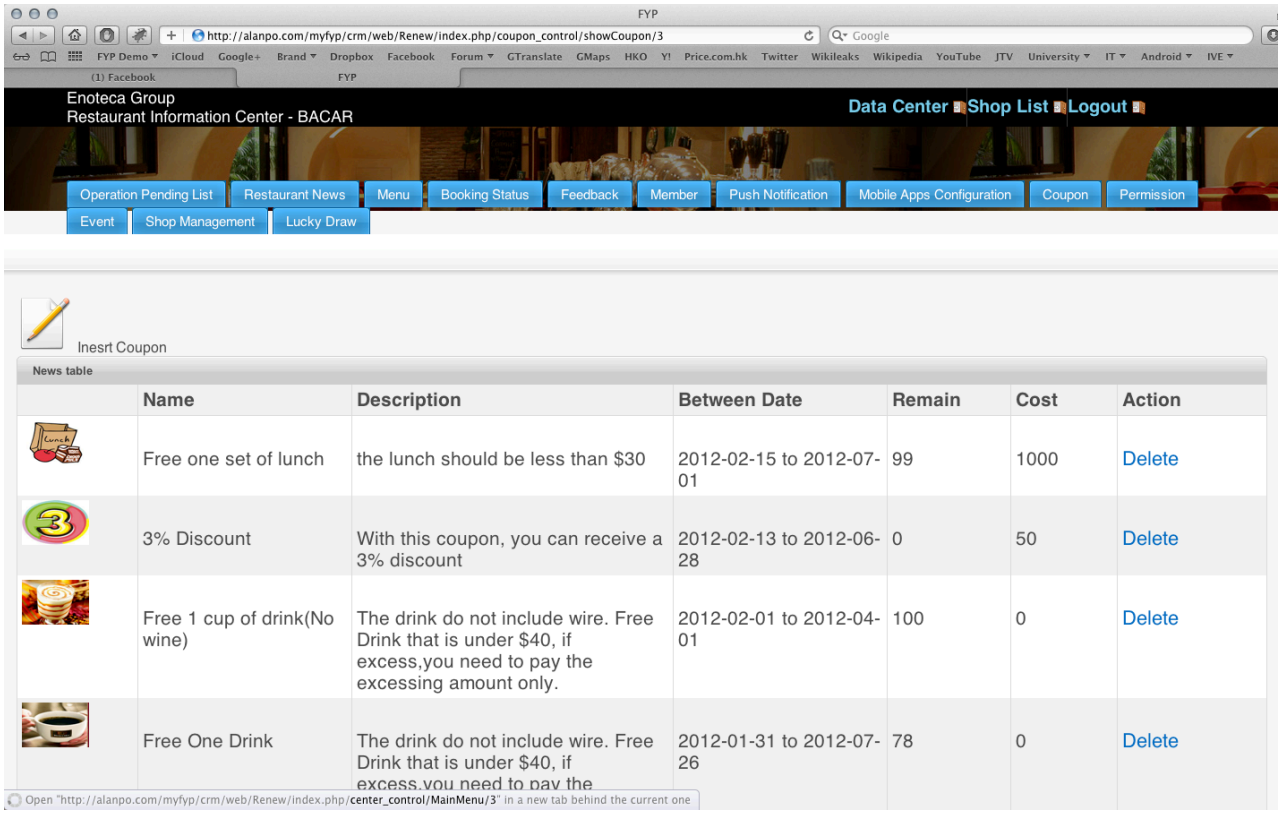


Figure 116. Website E-Coupon

This is designed for the user to make modification on the E-coupon, the table includes the information of each e-coupon.

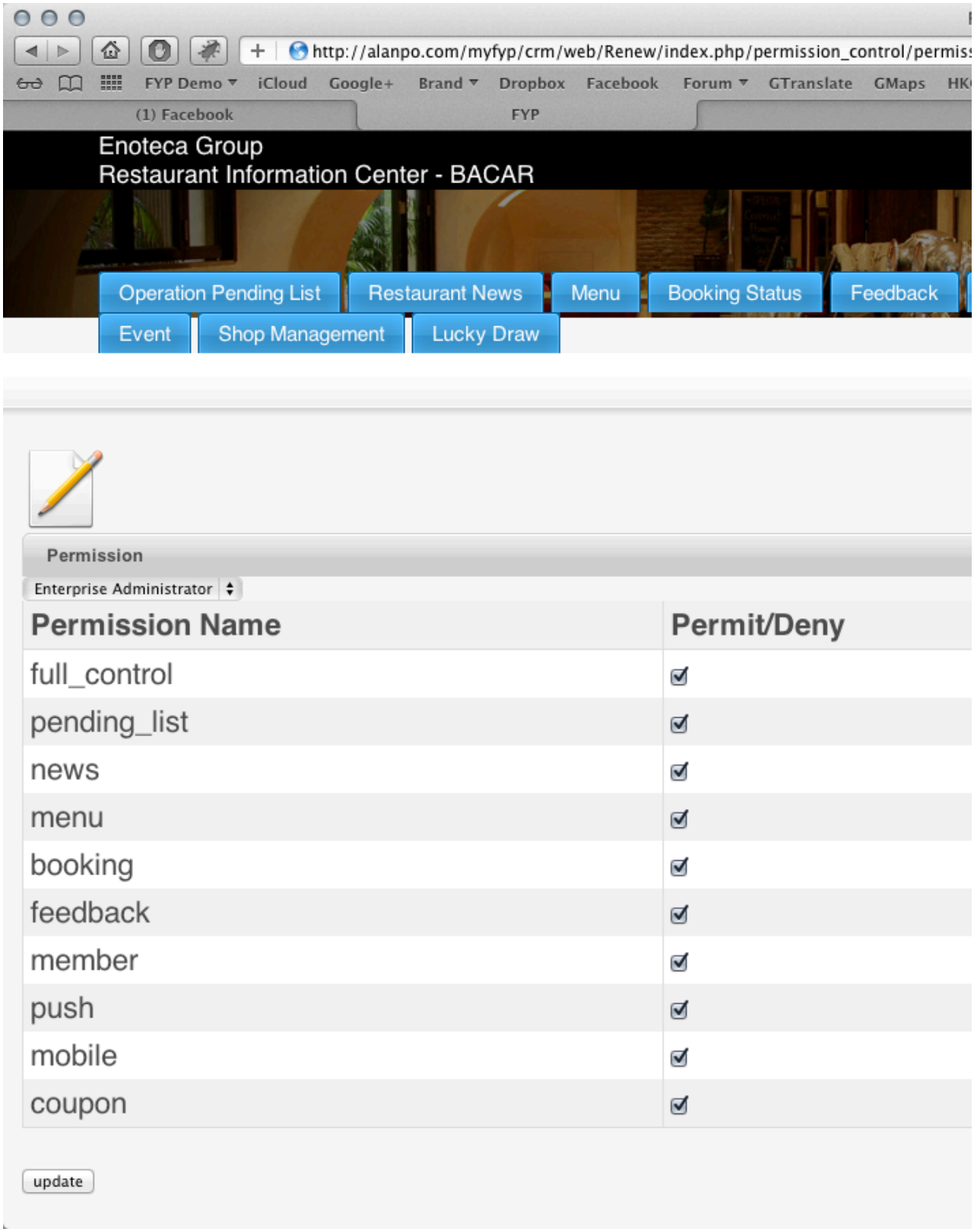


Figure 117. Website permission setting

This is the design of the permission control of different group of people.

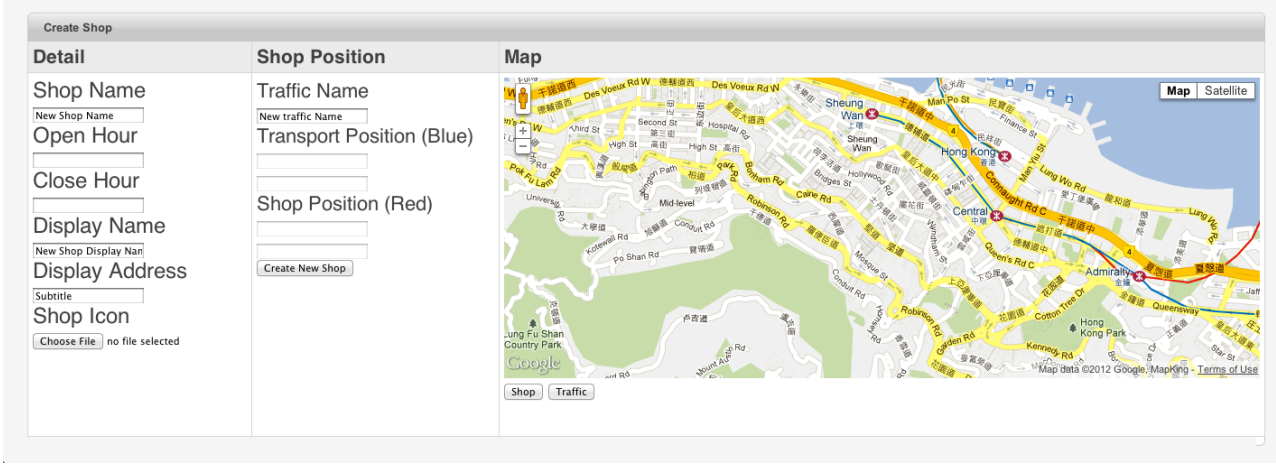


Figure 118. Website set shop location

This is the google map design for the user to set the shop location; the user can magnify the map to set the location of the shop.

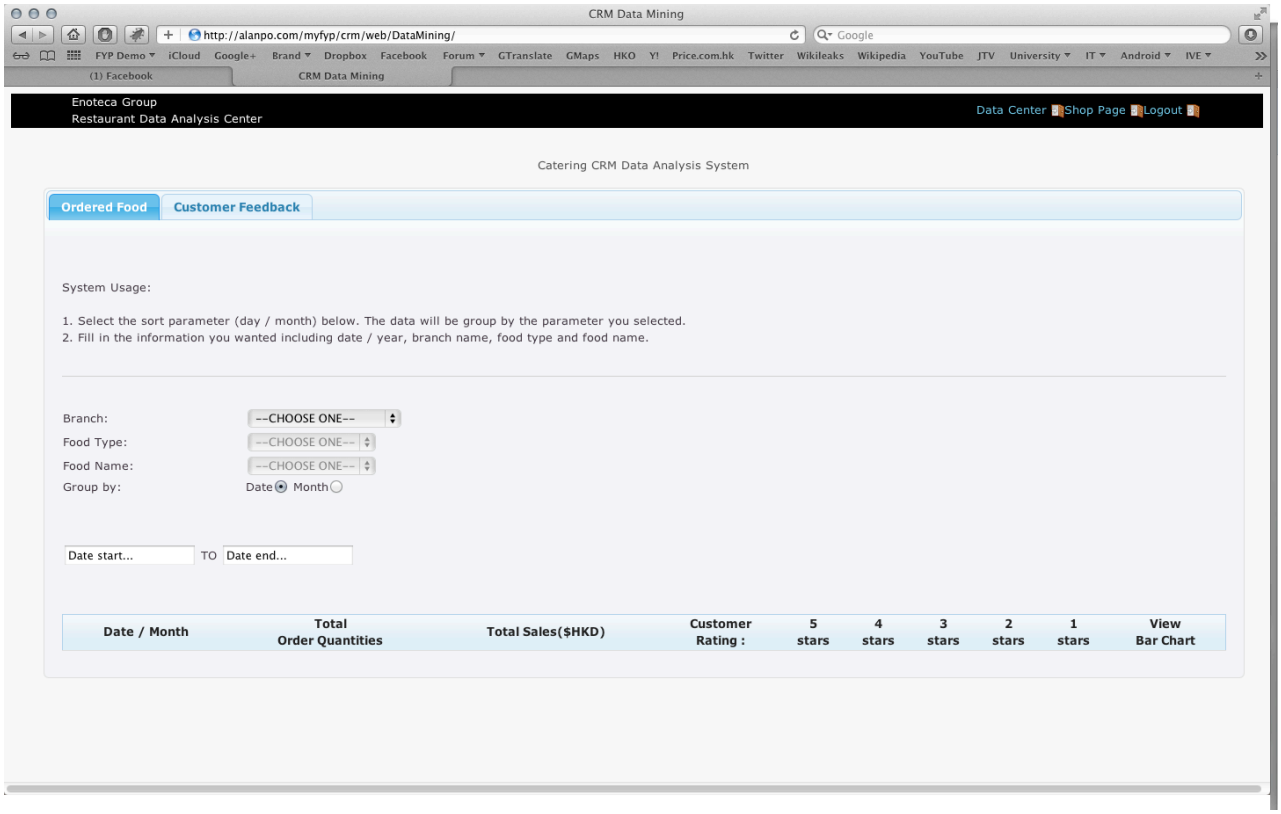


Figure 119. Website data analysis

This is the data center for the user to make analysis on the data in the database. When the user change the parameters on the website, the analysis data will also change.

2012-01-01 TO 2012-01-18

Date / Month	Total Order Quantities	Total Sales($HKD)	Customer Rating :	5 stars	4 stars	3 stars	2 stars	1 stars	View Bar Chart
2012-Jan-01	12	$ 696		3	0	0	1	2	View Graph
2012-Jan-02	7	$ 406		0	2	1	0	1	View Graph
2012-Jan-03	20	$ 1, 160		3	3	4	0	1	View Graph
2012-Jan-04	10	$ 580		1	1	1	1	1	View Graph
2012-Jan-05	16	$ 928		4	0	1	2	1	View Graph
2012-Jan-06	23	$ 1, 334		3	2	2	2	3	View Graph
2012-Jan-07	8	$ 464		0	2	1	1	0	View Graph
2012-Jan-08	25	$ 1, 450		1	2	2	6	2	View Graph
2012-Jan-09	12	$ 696		1	2	2	0	0	View Graph
2012-Jan-10	21	$ 1, 218		1	1	4	1	2	View Graph
2012-Jan-11	27	$ 1, 566		0	5	3	2	3	View Graph
2012-Jan-12	7	$ 406		0	1	2	0	1	View Graph
2012-Jan-13	28	$ 1, 624		1	3	3	3	2	View Graph
2012-Jan-14	12	$ 696		0	0	4	2	2	View Graph
2012-Jan-15	19	$ 1, 102		1	2	4	1	1	View Graph
2012-Jan-16	16	$ 928		0	6	0	1	2	View Graph
2012-Jan-17	7	$ 406		0	1	1	0	1	View Graph
2012-Jan-18	17	$ 986		1	0	3	1	2	View Graph

Download Report

Figure 120Website data analysis

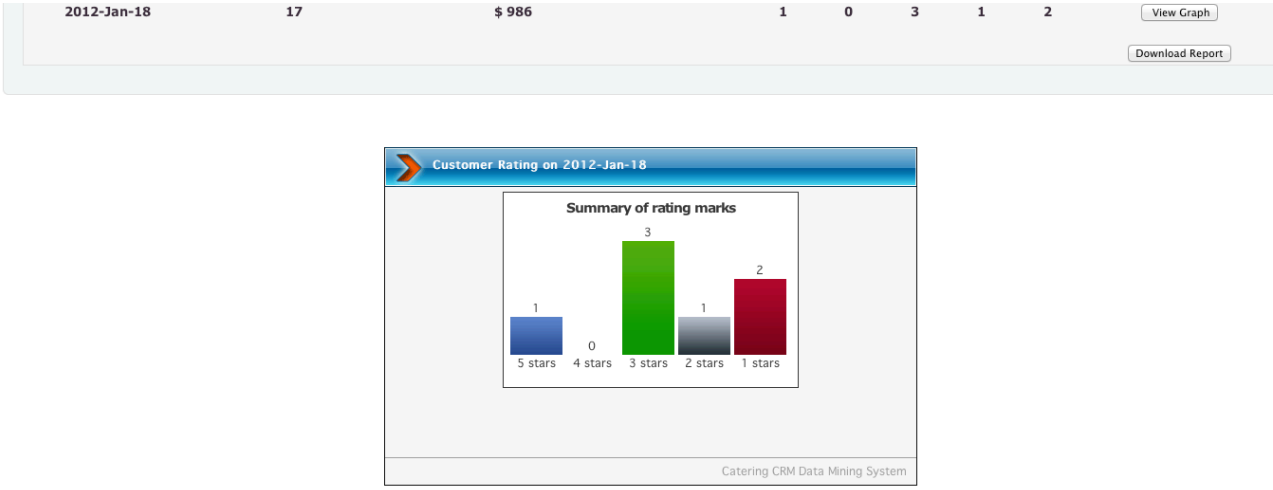
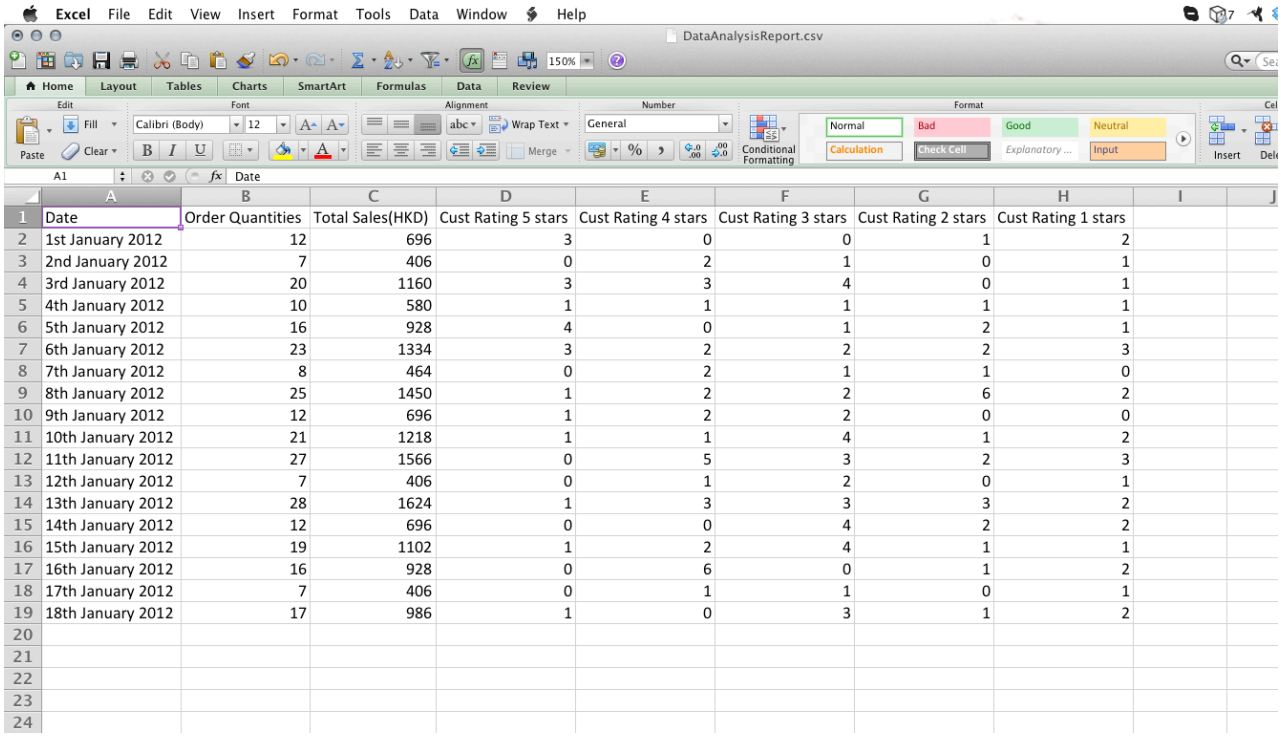


Figure 121. Website bar chart



	A	B	C	D	E	F	G	H	I	J
1	Date	Order Quantities	Total Sales(HKD)	Cust Rating 5 stars	Cust Rating 4 stars	Cust Rating 3 stars	Cust Rating 2 stars	Cust Rating 1 stars		
2	1st January 2012	12	696	3	0	0	1	2		
3	2nd January 2012	7	406	0	2	1	0	1		
4	3rd January 2012	20	1160	3	3	4	0	1		
5	4th January 2012	10	580	1	1	1	1	1		
6	5th January 2012	16	928	4	0	1	2	1		
7	6th January 2012	23	1334	3	2	2	2	3		
8	7th January 2012	8	464	0	2	1	1	0		
9	8th January 2012	25	1450	1	2	2	6	2		
10	9th January 2012	12	696	1	2	2	0	0		
11	10th January 2012	21	1218	1	1	4	1	2		
12	11th January 2012	27	1566	0	5	3	2	3		
13	12th January 2012	7	406	0	1	2	0	1		
14	13th January 2012	28	1624	1	3	3	3	2		
15	14th January 2012	12	696	0	0	4	2	2		
16	15th January 2012	19	1102	1	2	4	1	1		
17	16th January 2012	16	928	0	6	0	1	2		
18	17th January 2012	7	406	0	1	1	0	1		
19	18th January 2012	17	986	1	0	3	1	2		

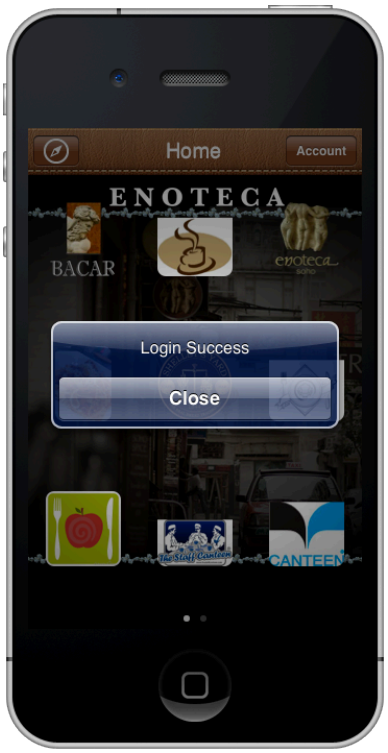
Figure 122. CSV report of data analysis

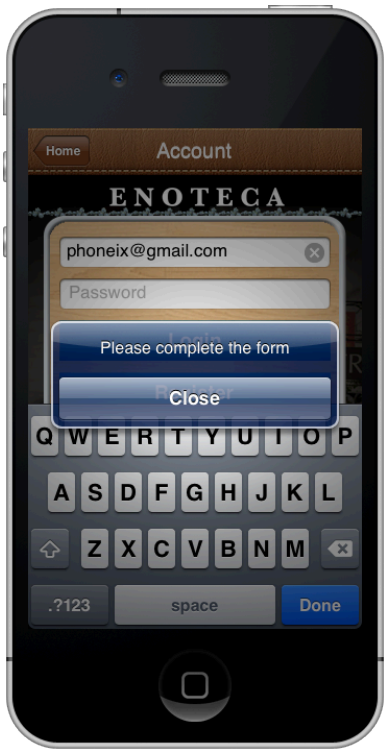
IMPLEMENTATION

iPhone Test Plan and Test Result

Login

Case ID	Description	Purpose	Expected Result	Match with expectation?
1	Enter the valid email and password	Test if user can login successfully	User login successfully	YES
2	Enter invalid email or invalid password	Test if invalid data can be used for login	Error message shown	YES
3	Email or password is missing	Test for error checking	Error message shown	YES





Register

Case ID	Description	Purpose	Expected Result	Match with expectation?
4	Enter the valid data	Test if new user can register successfully	User register successfully	YES
5	“Password” and “Retype Password” mismatch	Test if the checking is available	Error message “Password Mismatch” shown	YES
6	Invalid Email	Test the email checking	Error message “Invalid Email” shown	YES
7	Register with a registered email	To avoid an email being register for more than one time.	Error message shown	YES

Account
Register
Register
ENOTECA
Tai Man
phoneix@gmail.com
This email is already registered.
Close
2000-05-12

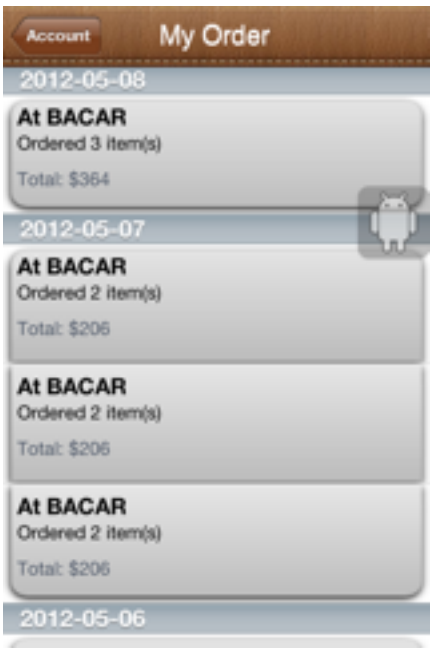
Home
Account
ENOTECA
Email
Password
Registered Successfully
Close

View Coupon Log

Case ID	Description	Purpose	Expected Result	Match with expectation?
8	After login, tried to view the coupon that have been used	Test if the user can check their used coupon	User can view their coupon	YES

View My Order

Case ID	Description	Purpose	Expected Result	Match with expectation?
9	After login, tried to view the order log	Test if the user can check their order	User can view their order	YES

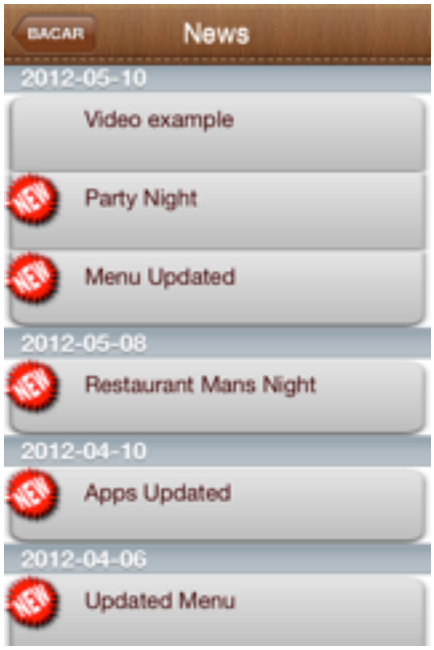


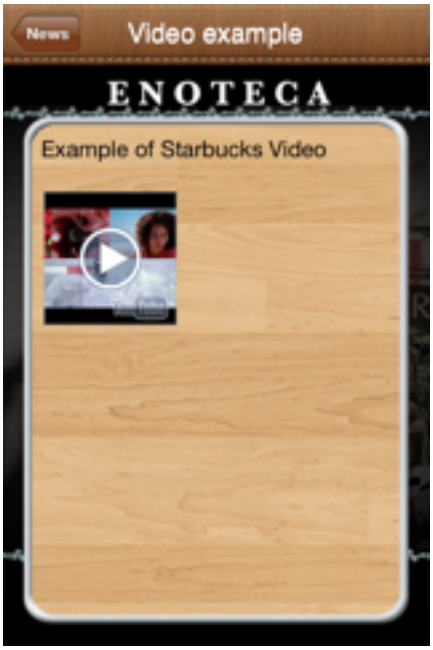
View Shop

Case ID	Description	Purpose	Expected Result	Match with expectation?
10	View different Restaurant	Different Restaurant should have different function provided.	User can view their order	YES

View News

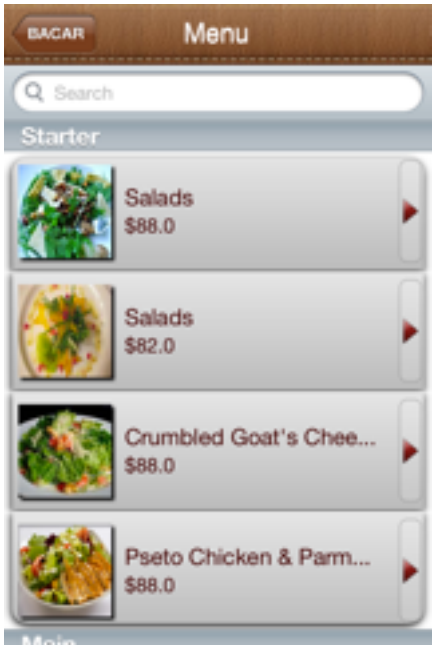
Case ID	Description	Purpose	Expected Result	Match with expectation?
11	View the news of Restaurant	Test if the application can get the news from server	User can view the news	YES





View Menu

Case ID	Description	Purpose	Expected Result	Match with expectation?
13	View the menu of Restaurant	Test if the application can get the menu from server	User can view the menu	YES
14	Search for “Cherry”	Test if the search function working properly	Only the food containing the word “Cherry” being shown	YES
15	Share to Facebook	Test if the sharing function work properly	The menu is shared to Facebook	YES


BACAR
Menu
Search
Starter
Salads
$88.0
Salads
$82.0
Crumbled Goat's Chee...
$88.0
Pseto Chicken & Parm...
$88.0

cherry
Cancel
Cherry Tomato Lingui...
$58.0
Cherry Tomato & Pest...
$108.0
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
?1
中
enter

Cancel
Send to Facebook
Cherry Tomato & Pesto From BACAR
Price: $108.0
香港中環蘇豪些利街2號
Image
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
?1
中
enter

Menu
Cherry Tomato &...
$ 108.0
Fresh tomato sauce, mozzarella, vine ripened cherry tomatoes & pesto

Phoneix Kwok
Cherry Tomato & Pesto From BACAR
Price: $108.0

香港中環蘇豪些利街2號

FYP's Photos

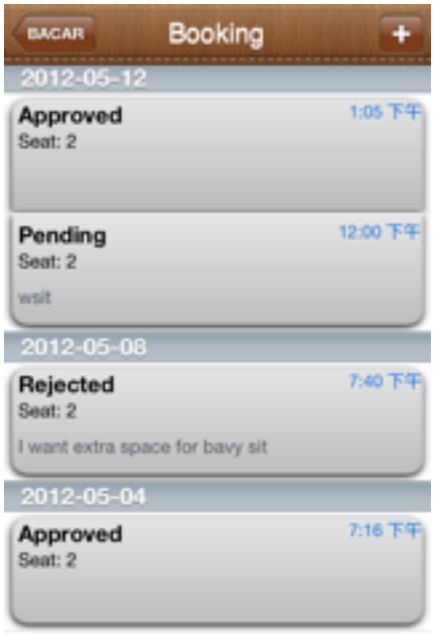
讚 · 留言 · 分享 · 於 23 秒前來自 FYP's ·

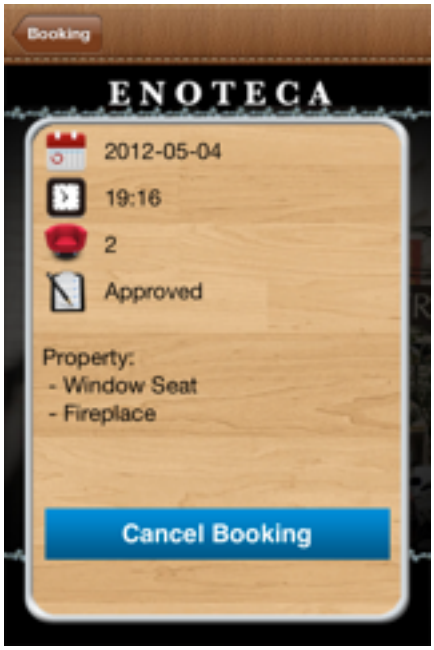
View Map

Case ID	Description	Purpose	Expected Result	Match with expectation?
16	View the location of the restaurant on map	Test if the application can get the shop data correctly	User can view the restaurant and the closest traffic on the map	YES
17	Find My Position	Check if the application can fid the user location	The location of user is shown	YES
18	Route to the restaurant location	Test if the routing is functioning properly	Show the path from user location to restaurant	YES

View Booking

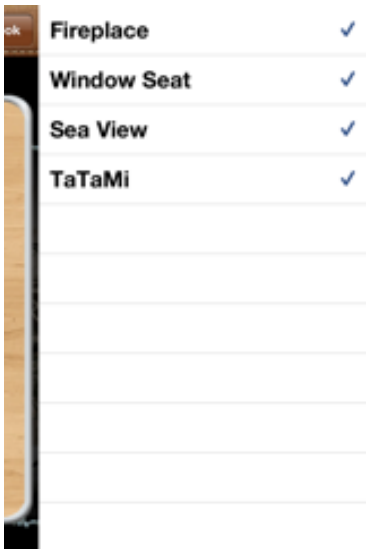
Case ID	Description	Purpose	Expected Result	Match with expectation?
19	View the booking	Show the booking that made by the user	List of booking shown	YES
20	Cancel Booking	A user should be able to cancel the booking	The booking is canceled	YES

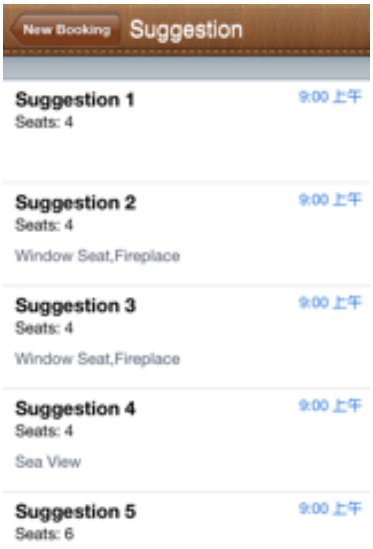


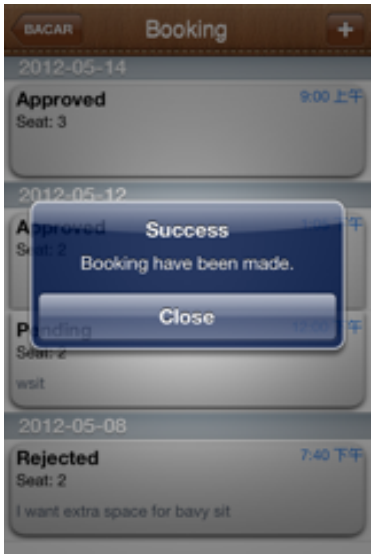


New Booking

Case ID	Description	Purpose	Expected Result	Match with expectation?
21	Click Submit without entering any data	To avoid the abuse of the system	Error message shown	YES
22	Select all the property	Test what happen if no table can fulfill the requirement	A list of suggestion is shown	YES
23	Booking at the time not in opening hour	To avoid booking at non opening hour	Error message should be shown	YES
24	Book with valid data	To test if user can book successfully	Booked successfully	YES

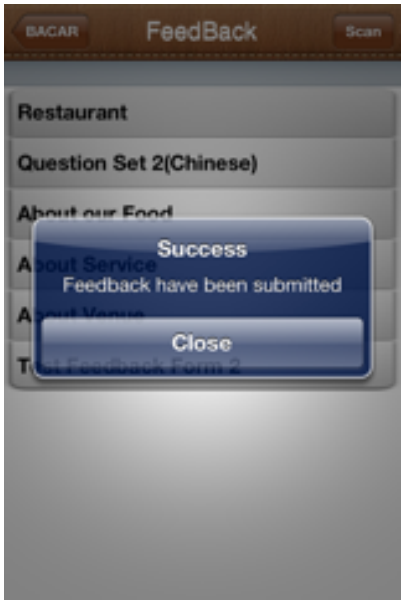






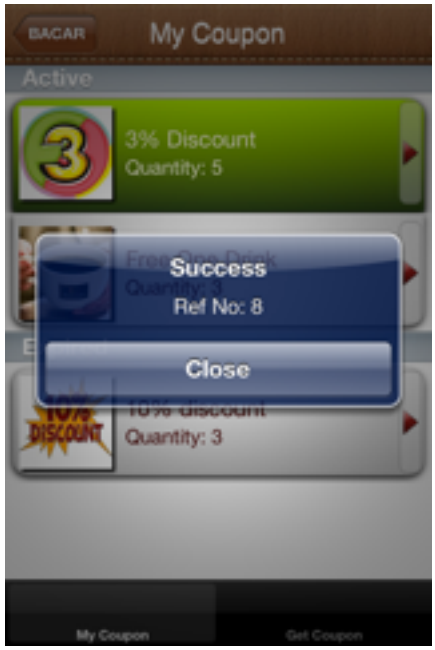
Feedback

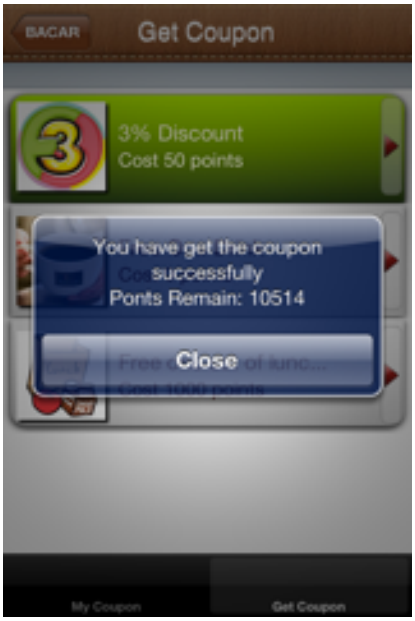
Case ID	Description	Purpose	Expected Result	Match with expectation?
25	Submit a feedback	Test if the application can submit the feedback	User can view the menu	YES
26	Receive a order from server	Test if the application can receive the order that made by a user for review	User can view the questionnaire for the order they have made	YES

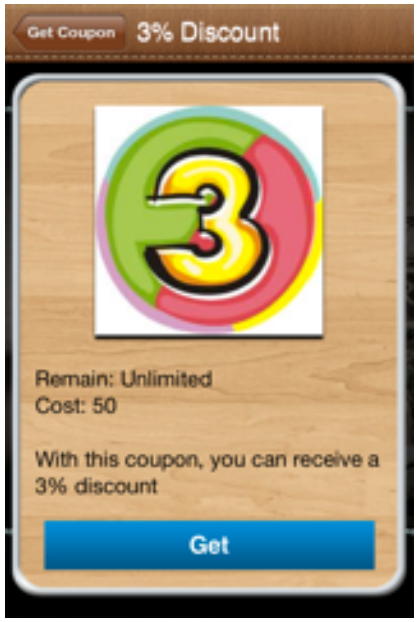


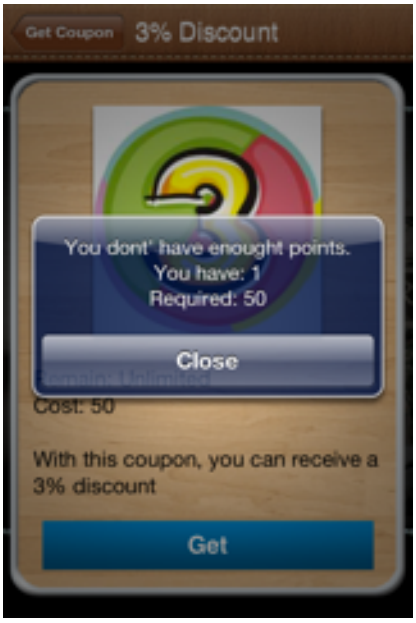
Coupon

Case ID	Description	Purpose	Expected Result	Match with expectation?
27	View the coupon	Show the coupon that user owned or they can get	List of coupon shown	YES
28	Use the coupon	Test if the function working properly	A reference no. should be shown, and the remain quantity shown be deduct by one	YES
29	Get the coupon with enough point	Test if the checking is valid	User should be able to get the coupon	YES
30	Get the coupon without enough point	Test if the checking is valid	User should not be able to get the coupon	YES



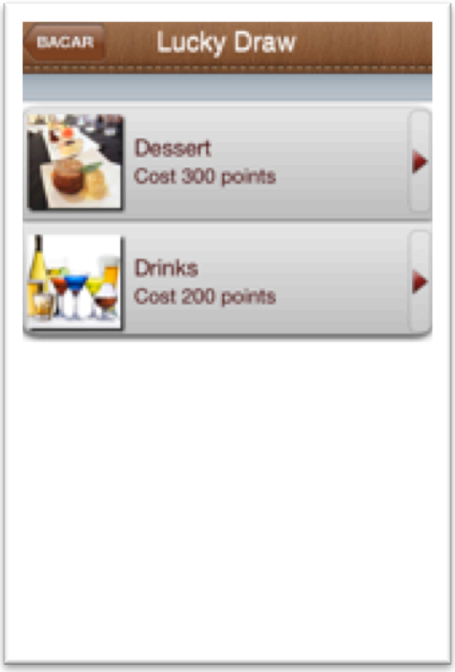

BACAR
Get Coupon
3% Discount
Cost 50 points
You have get the coupon
successfully
Ponts Remain: 10514
Close
My Coupon
Get Coupon


Get Coupon
3% Discount
Remain: Unlimited
Cost: 50
With this coupon, you can receive a
3% discount
Get

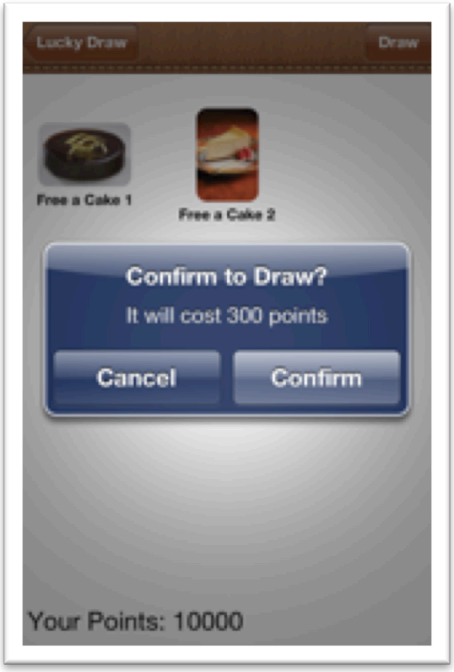

Get Coupon
3% Discount
You dont' have enought points.
You have: 1
Required: 50
Close
Cost: 50
With this coupon, you can receive a
3% discount
Get

Lucky Draw

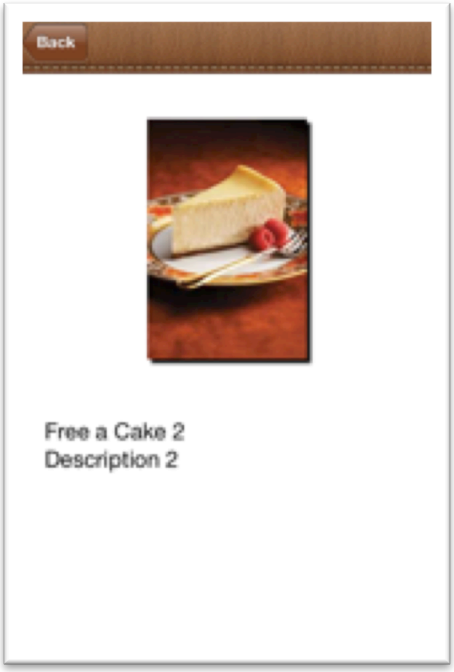
Case ID	Description	Purpose	Expected Result	Match with expectation?
31	View the lucky draw	Show the lucky draw that user can play	List of lucky draw shown	YES
32	Play the lucky draw with enough point	Test if the checking is valid	A random item is shown	YES
33	Play the lucky draw with enough point	Test if the checking is valid	User should be able to play lucky draw	YES
34	Use the item	Test if the checking is valid	A reference no. should be shown, and the remain quantity shown be deduct by one	YES



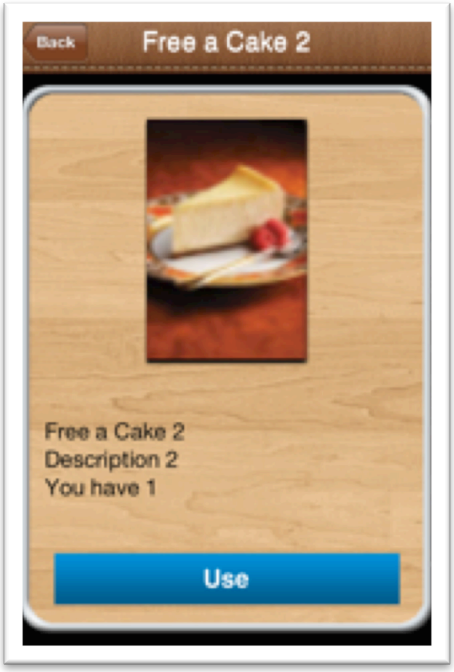
Lucky Draw
Draw
Free a Cake 1
Free a Cake 2
Confirm to Draw?
It will cost 300 points
Cancel
Confirm
Your Points: 10000

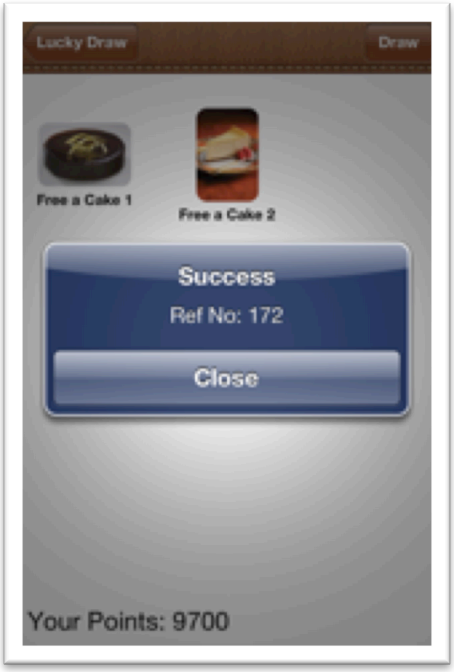

Lucky Draw
Draw
Free a Cake 1
Free a Cake 2
Confirm to Draw?
It will cost 300 points
Cancel
Confirm
Your Points: 10000

Back


Back
Free a Cake 2
Description 2

Lucky Draw
Draw
Free a Cake 1
Free a Cake 2
Your Points: 9700

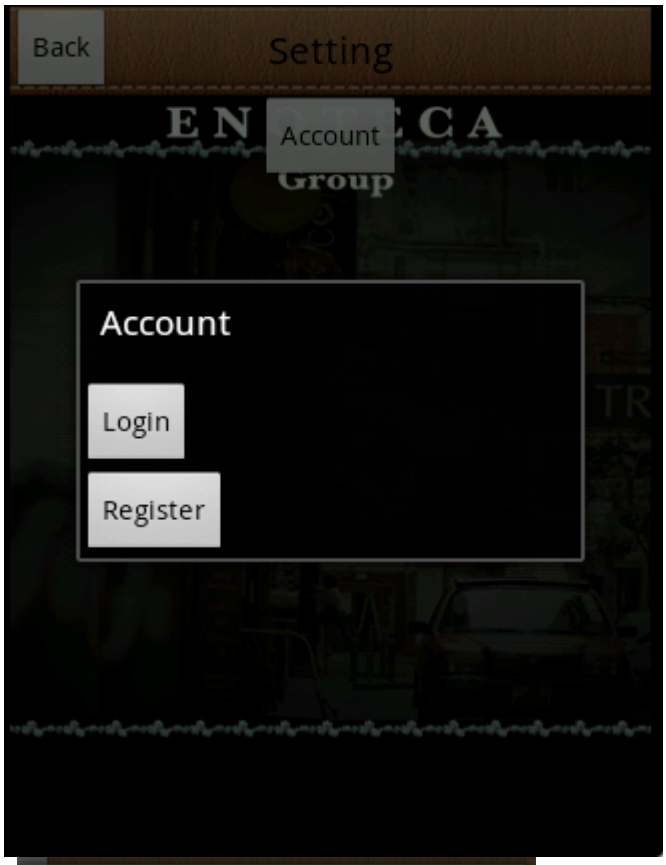

Back
Free a Cake 2
Free a Cake 2
Description 2
You have 1
Use

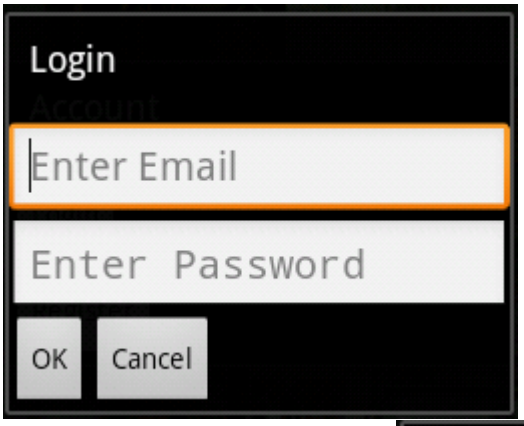

Lucky Draw
Draw
Free a Cake 1
Free a Cake 2
Success
Ref No: 172
Close
Your Points: 9700

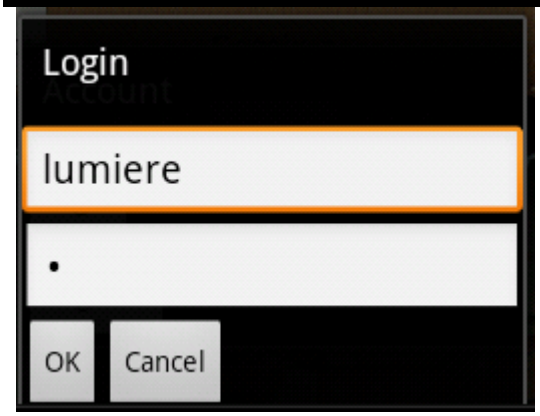
Test Plan and Test Result (Android)

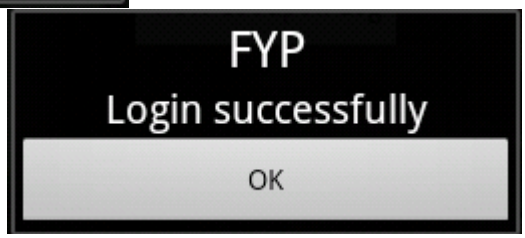
Login

Test Case ID	Test Case Description	Expected Result
1	Login with valid detail	Successful login



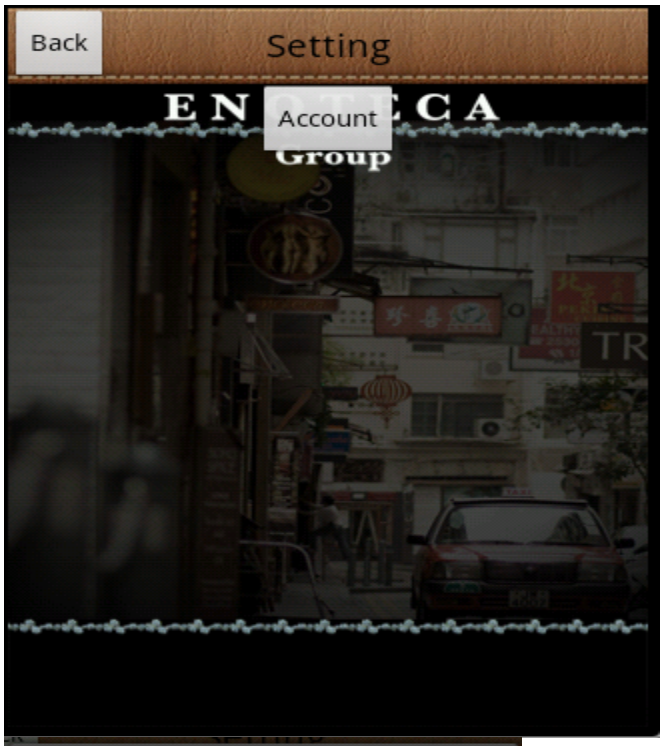


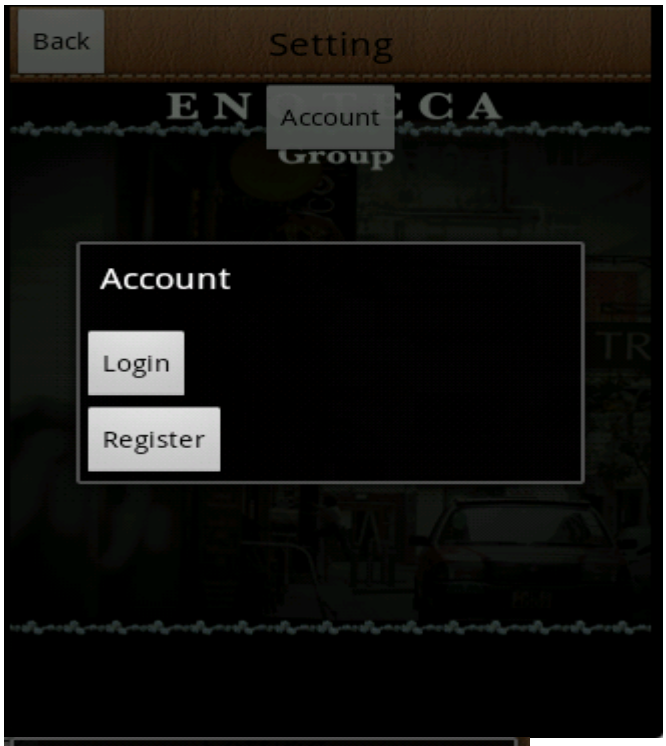


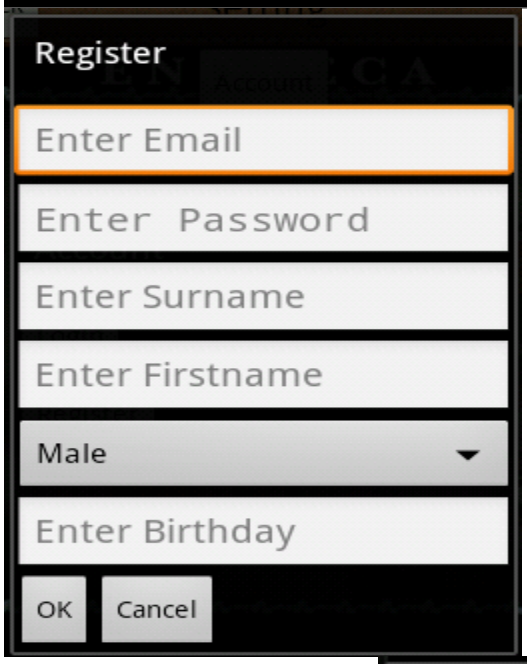


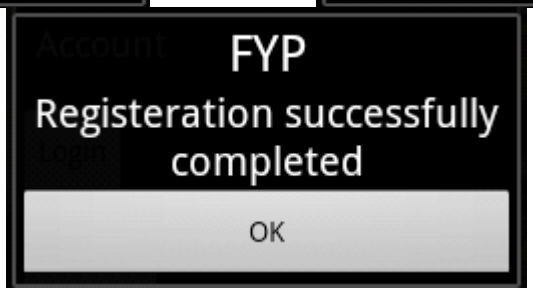
Register

Test Case ID	Test Case Description	Expected Result
2	Register with valid detail	Successful register



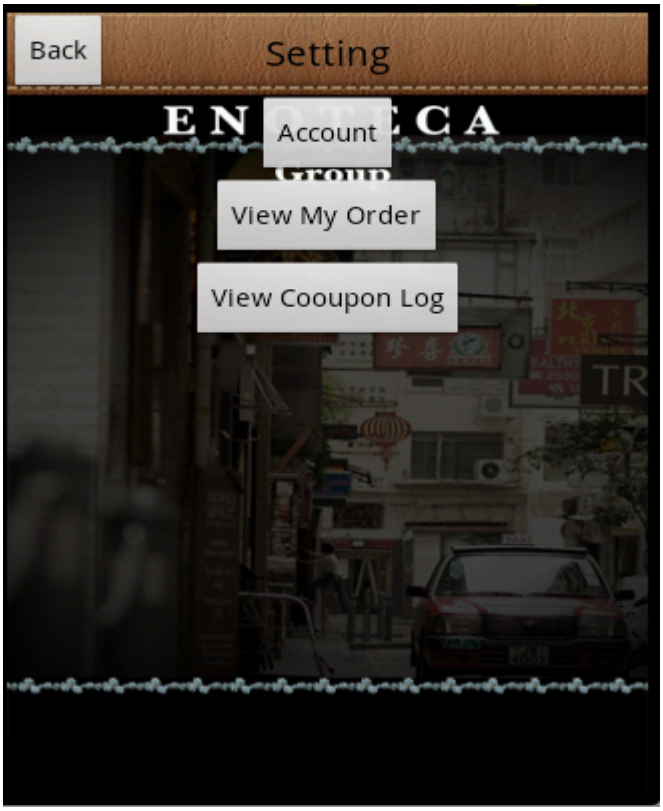


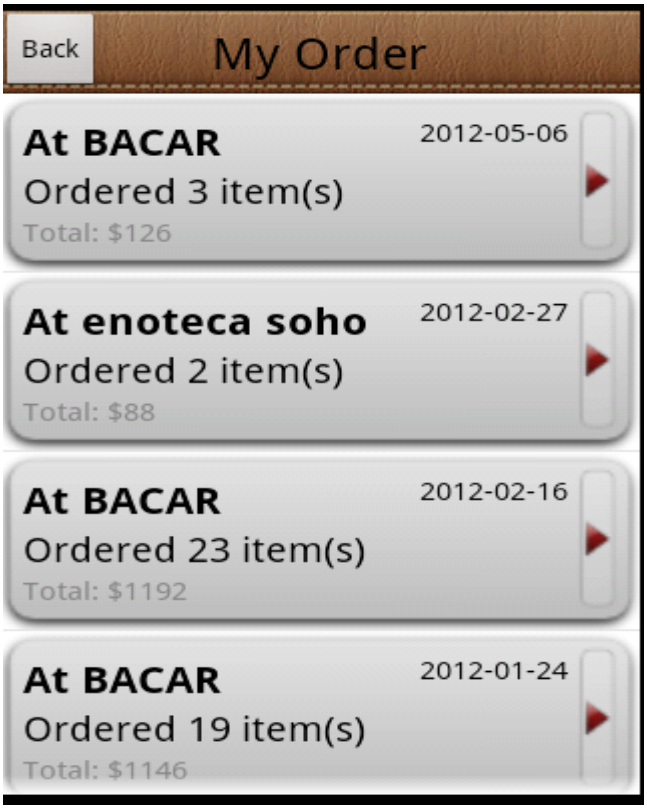


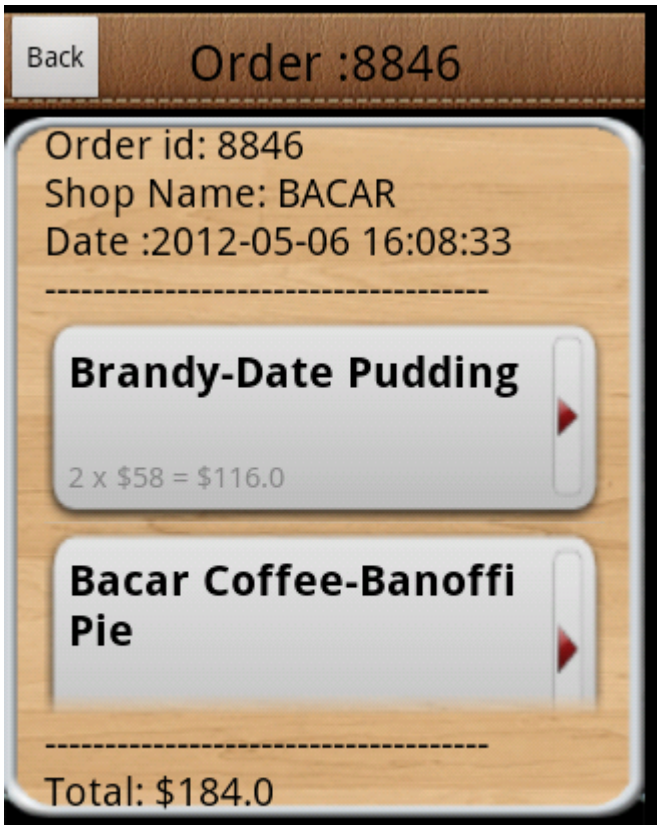


View My Order

Test Case ID	Test Case Description	Expected Result
3	After Login, show the order that user have ordered	Successful show order






Back
Order :8846
Order id: 8846
Shop Name: BACAR
Date :2012-05-06 16:08:33

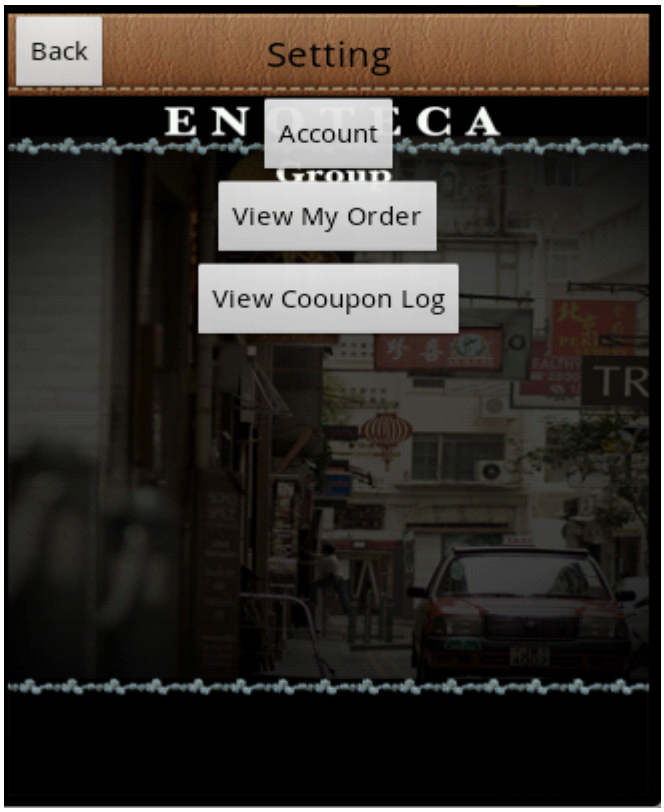
Brandy-Date Pudding
2 x $58 = $116.0
Bacar Coffee-Banoffi Pie

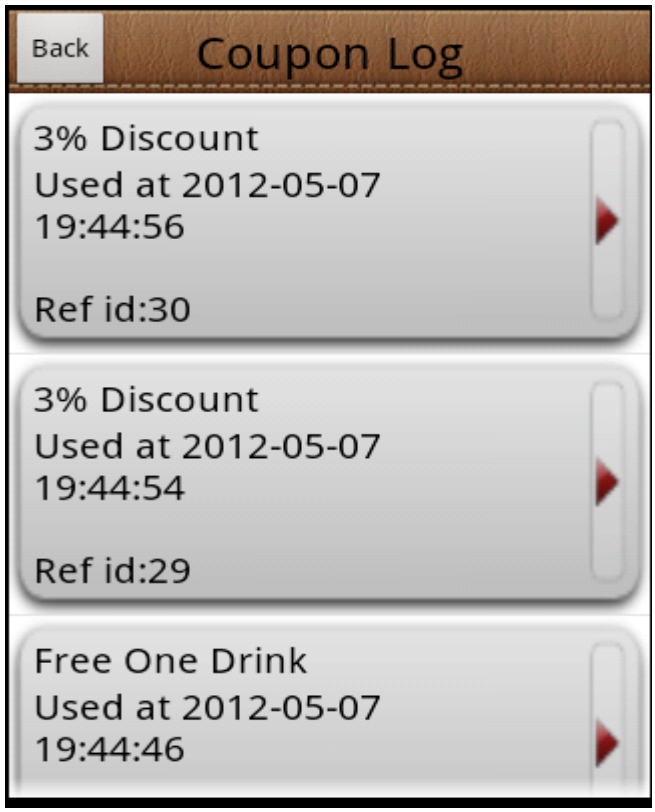
Total: $184.0

Back
Brandy-Date
58
with fresh whipped cream

View Coupon used

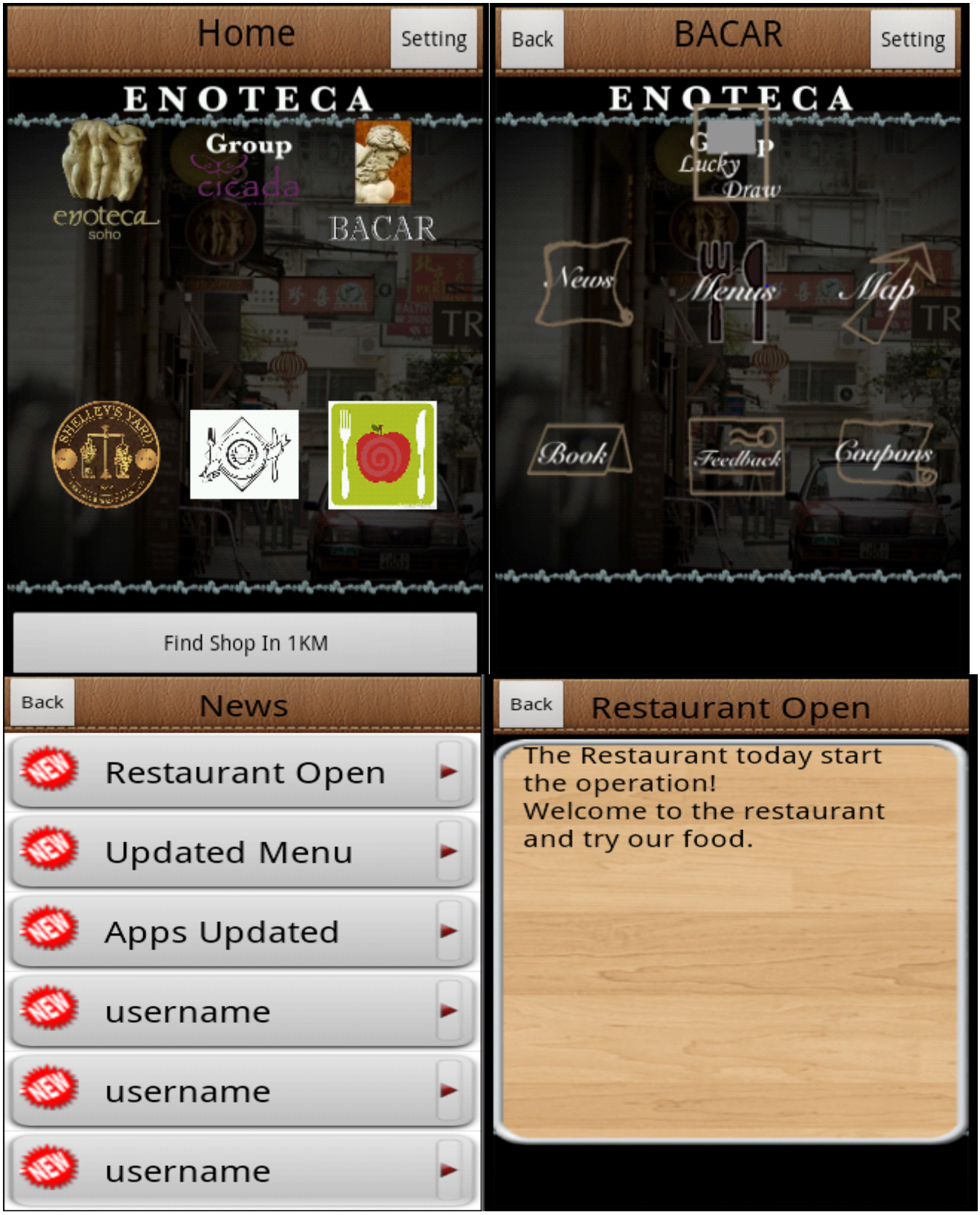
Test Case ID	Test Case Description	Expected Result
3	After Login, show the coupon that user have used	Successful show coupon





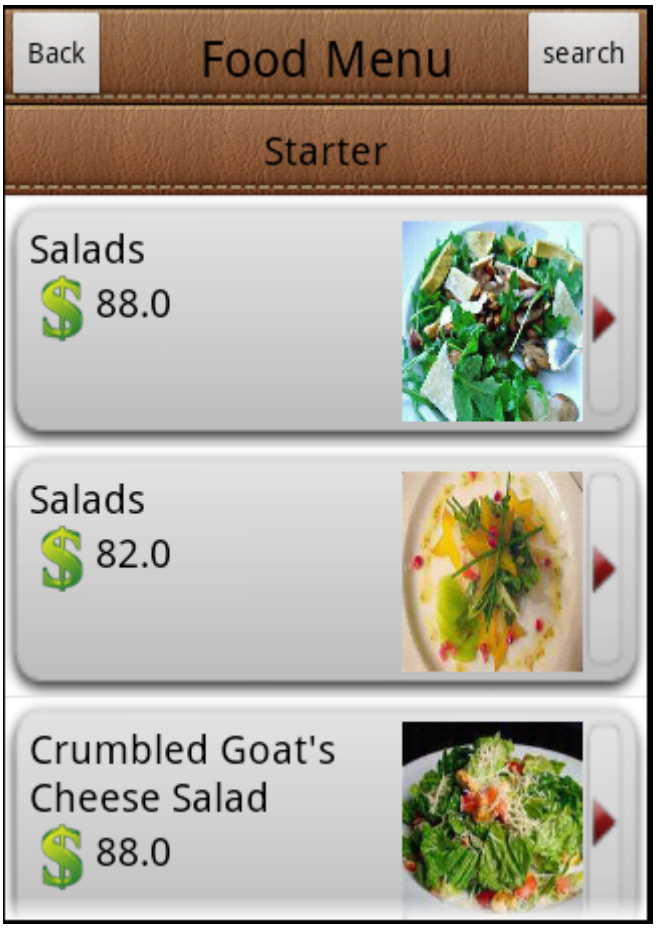
View news

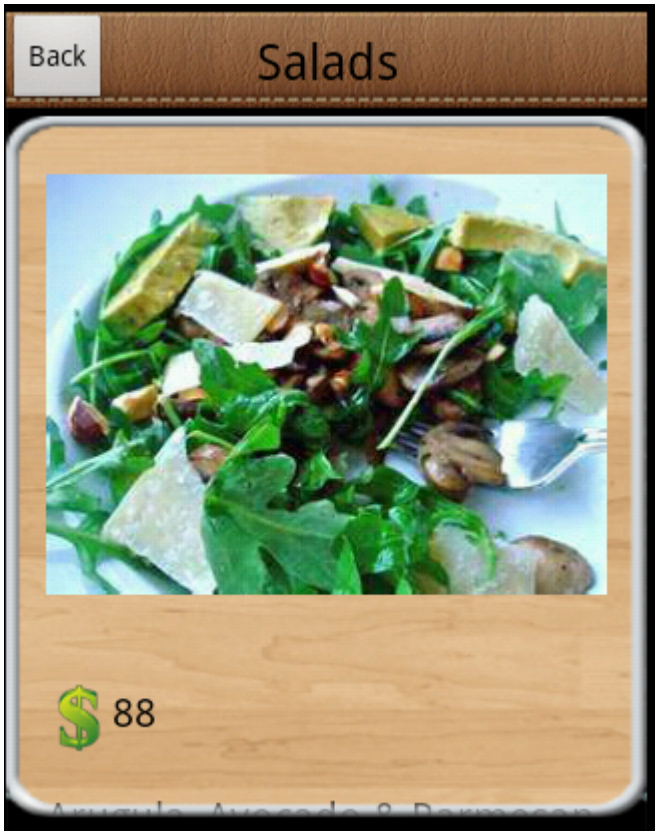
Test Case ID	Test Case Description	Expected Result
4	Show the news of the restaurant	Successful show coupon



View Menu

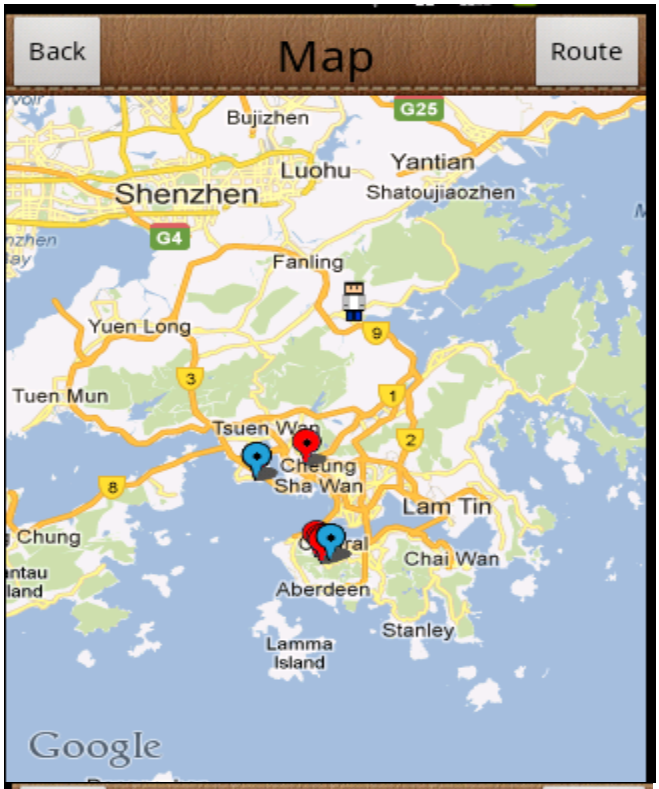
Test Case ID	Test Case Description	Expected Result
5	Show the food menu of the restaurant	Successful show food menu

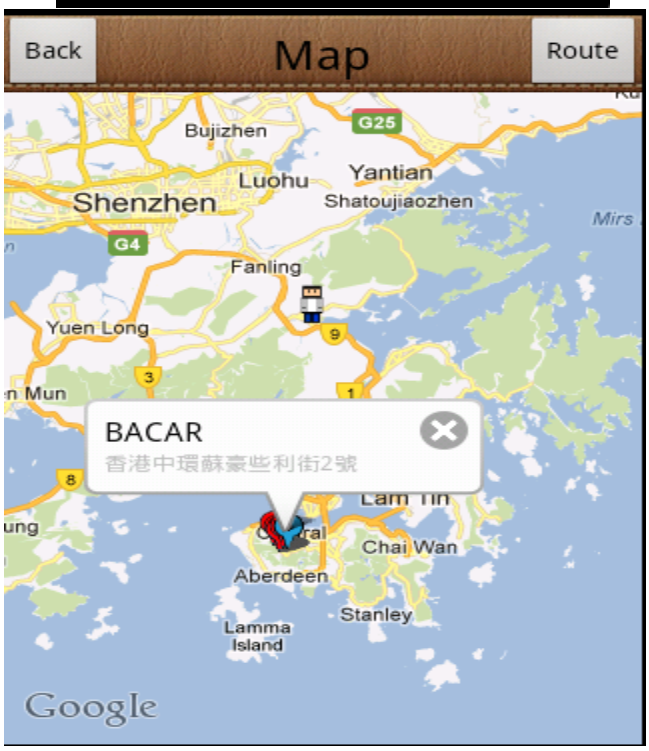


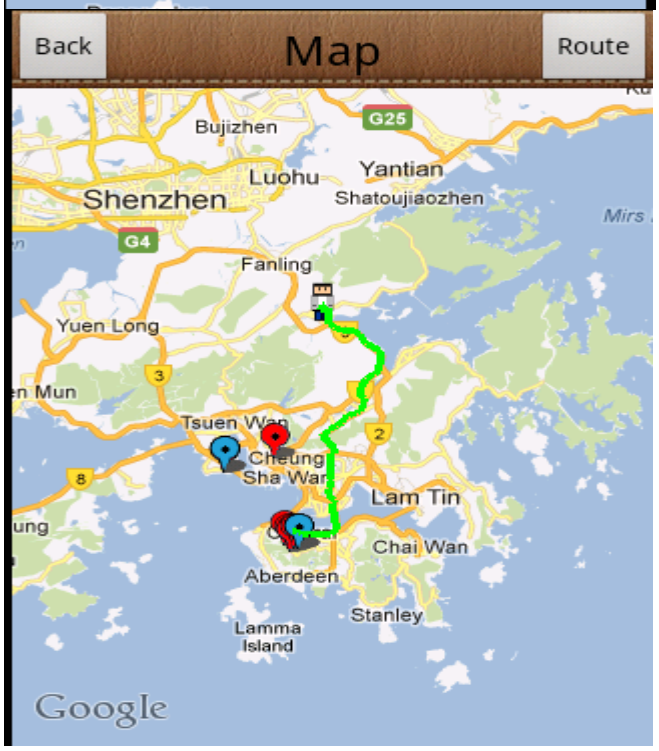

Back
Salads
88

View Map

Test Case ID	Test Case Description	Expected Result
6	Show the restaurant on the map	Successful show the restaurant

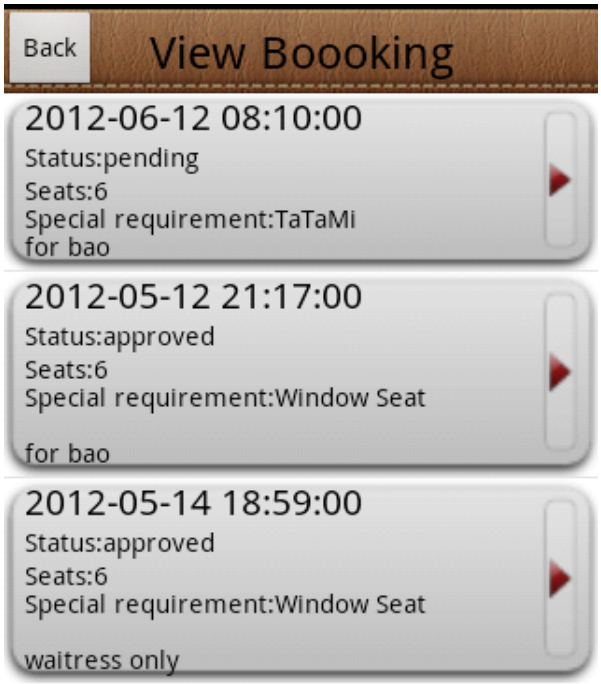






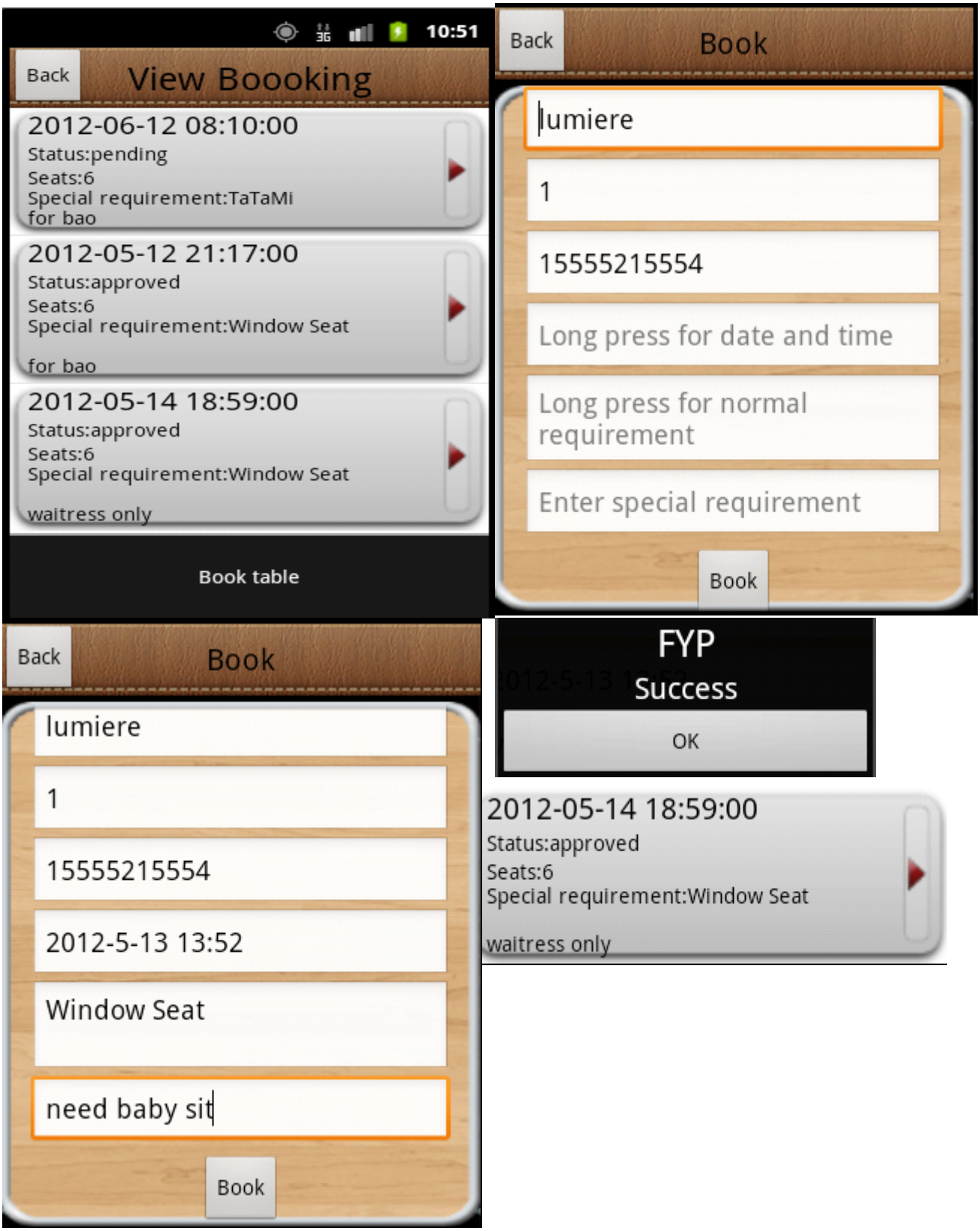
View Book

Test Case ID	Test Case Description	Expected Result
7	Show the booking user have requested	Successful show the booking



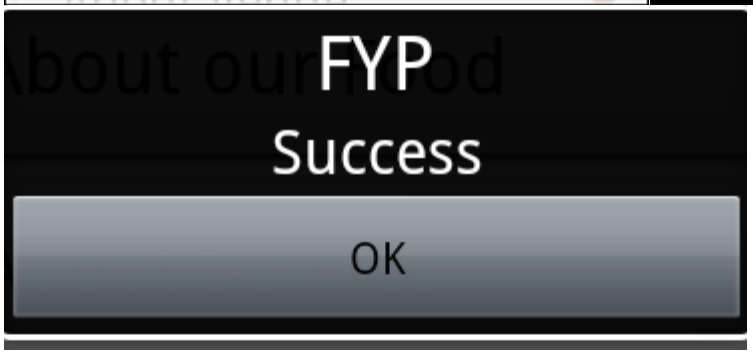
Booking

Test Case ID	Test Case Description	Expected Result
8	Book the table of the restaurant	Successful book the table



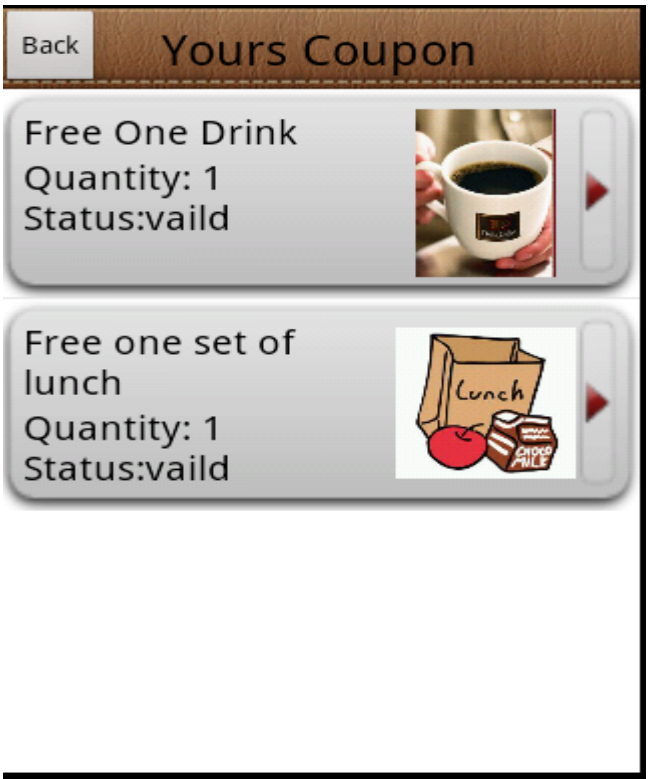
Feedback

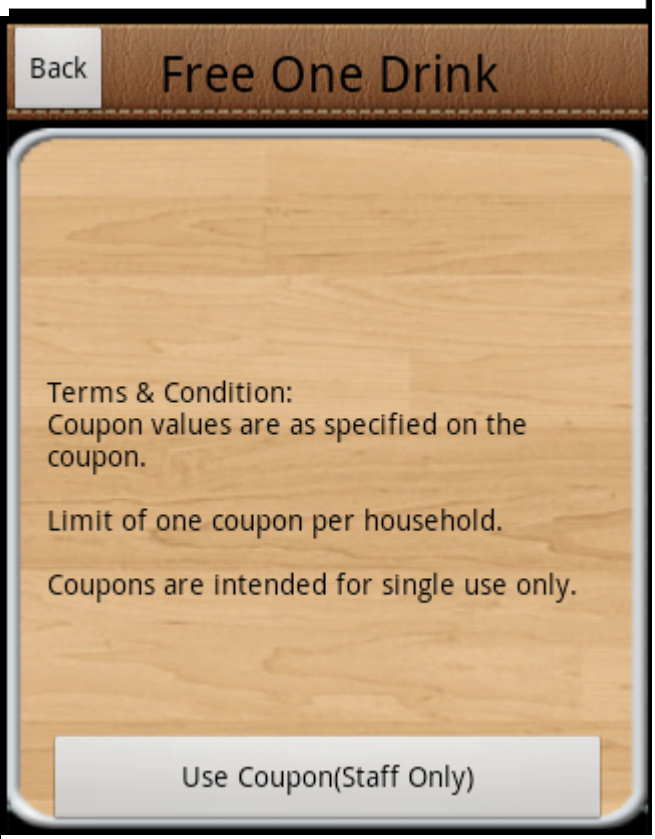
Test Case ID	Test Case Description	Expected Result
9	Submit a feedback to restaurant	Successful submit the feedback

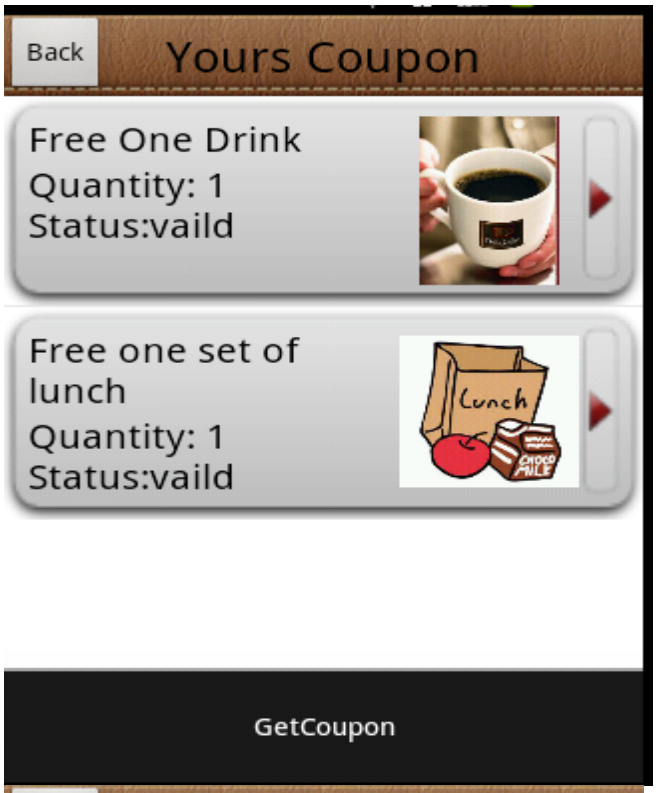


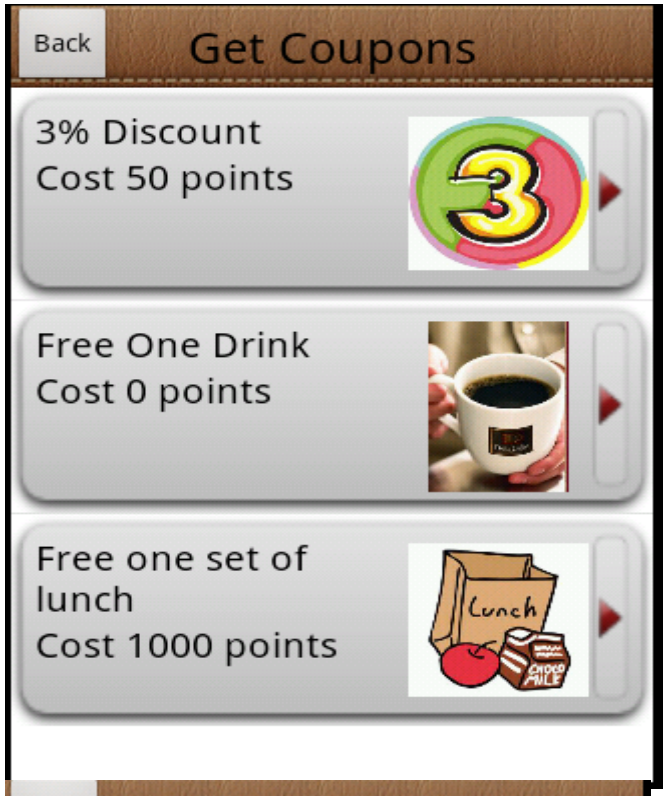
Coupon

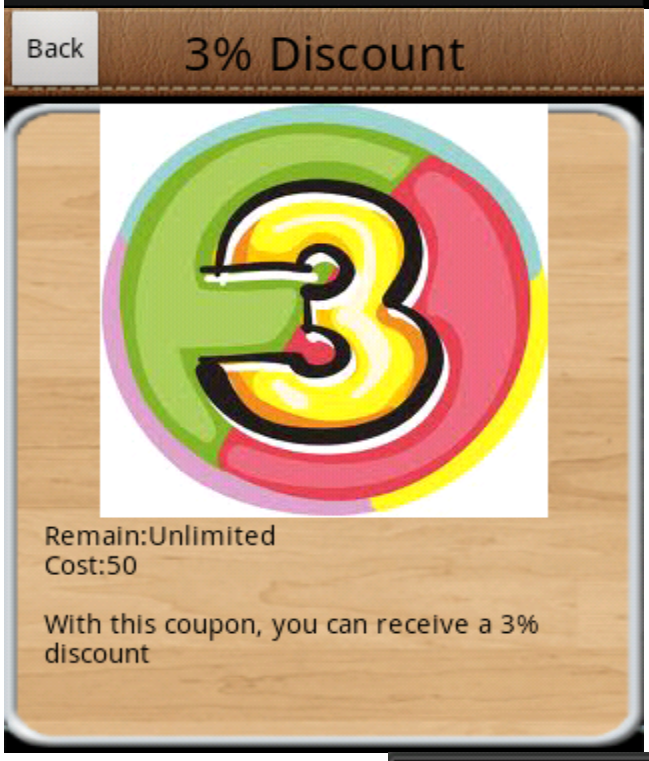
Test Case ID	Test Case Description	Expected Result
10	View the coupon that user can get or use	Successful view or get the coupon

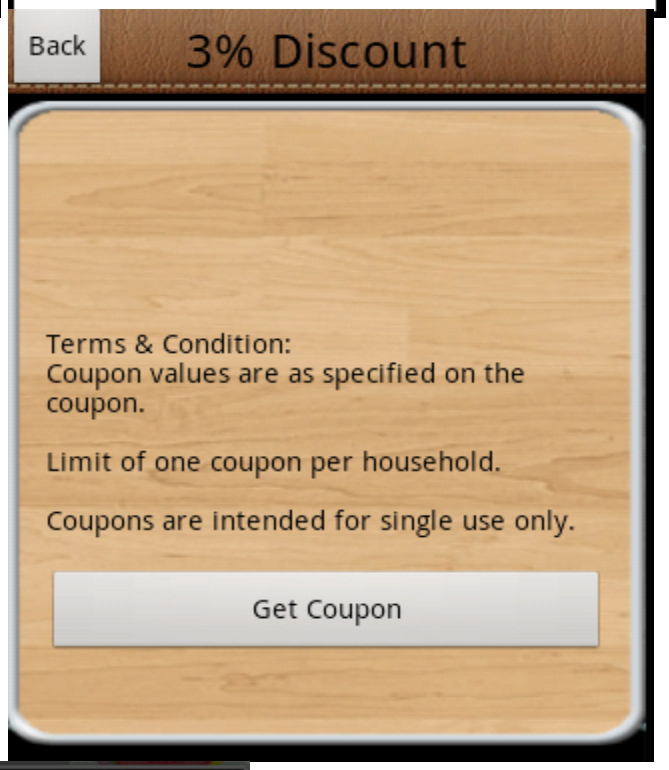


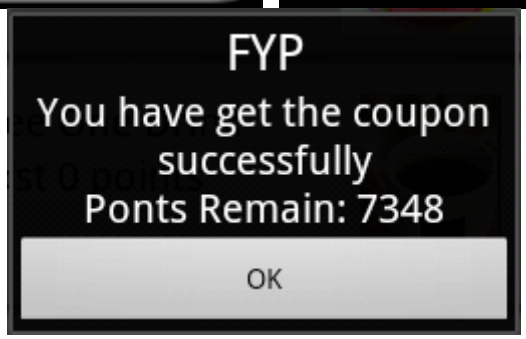



Back
Yours Coupon
Free One Drink
Quantity: 1
Status:vaild
Free one set of lunch
Quantity: 1
Status:vaild
Lunch
GetCoupon


Back
Get Coupons
3% Discount
Cost 50 points
Free One Drink
Cost 0 points
Free one set of lunch
Cost 1000 points
Lunch

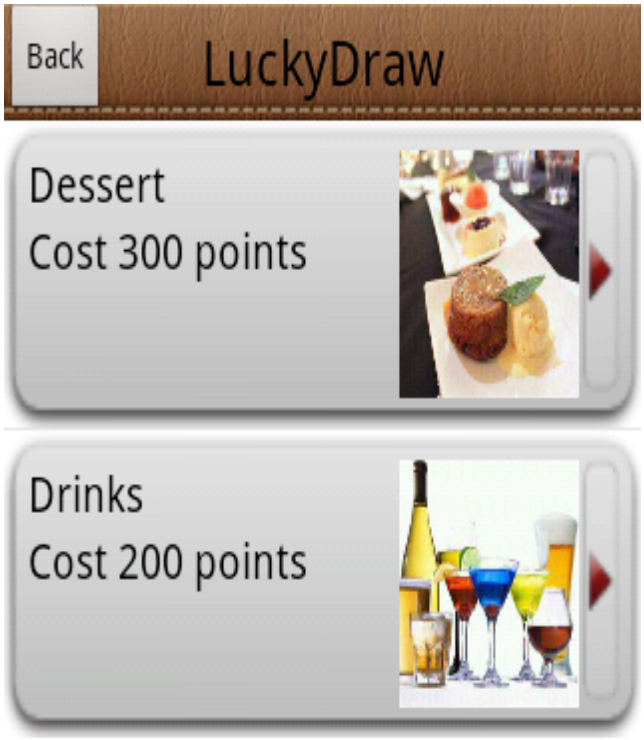

Back
3% Discount
Remain:Unlimited
Cost:50
With this coupon, you can receive a 3% discount

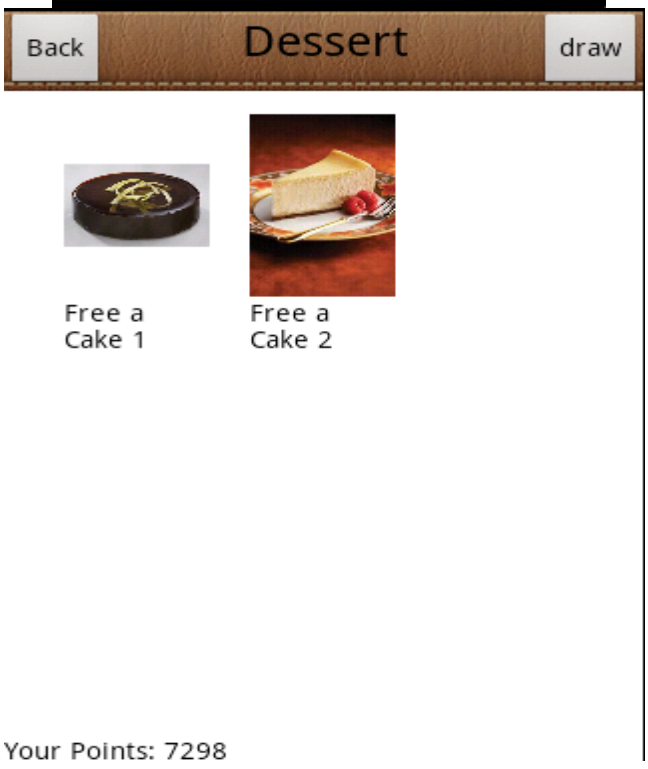

Back
3% Discount
Terms & Condition:
Coupon values are as specified on the coupon.
Limit of one coupon per household.
Coupons are intended for single use only.
Get Coupon

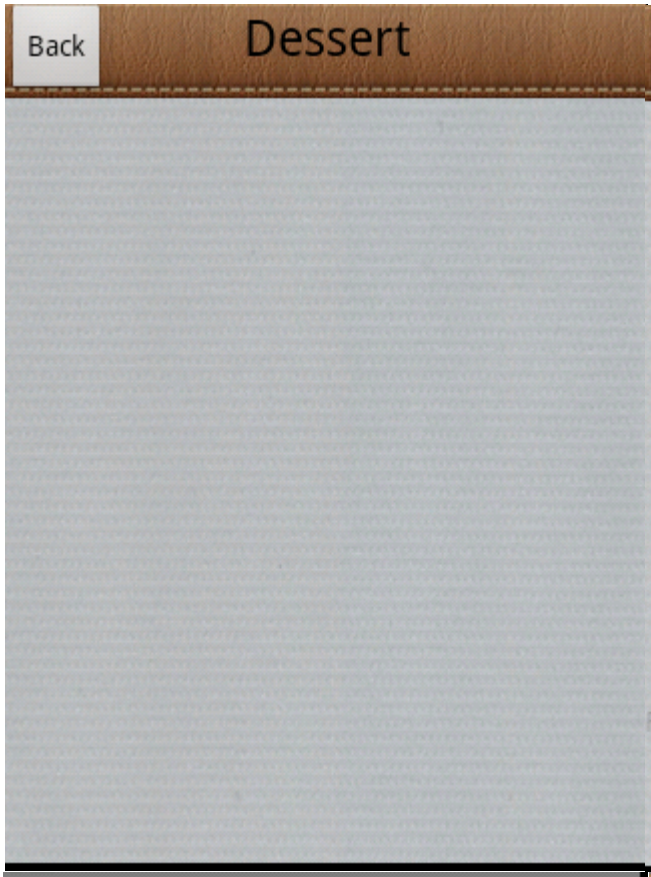

FYP
You have get the coupon successfully
Ponts Remain: 7348
OK

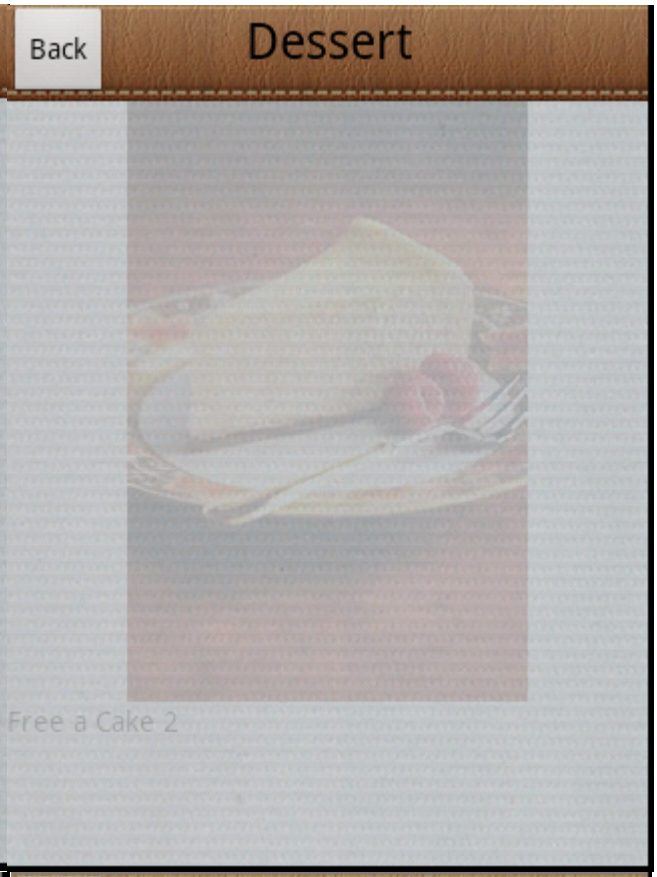
Lucky Draw

Test Case ID	Test Case Description	Expected Result
10	View the lucky draw prize or draw an item	Successful draw an item






Back
Dessert


Back
Dessert
Free a Cake 2

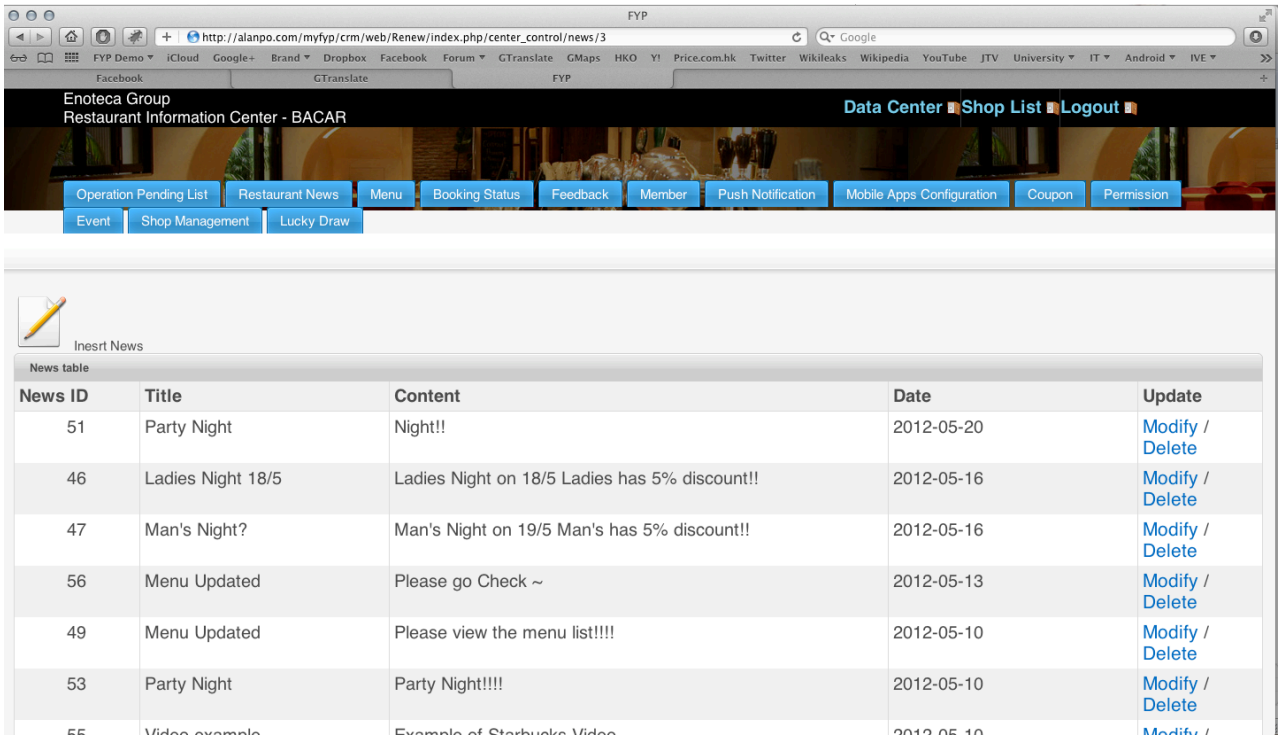
Back
Dessert
Free a Cake 2

Back
Free a Cake 2
Free a Cake 2
You have 2

Test Plan and Test Result (Website)

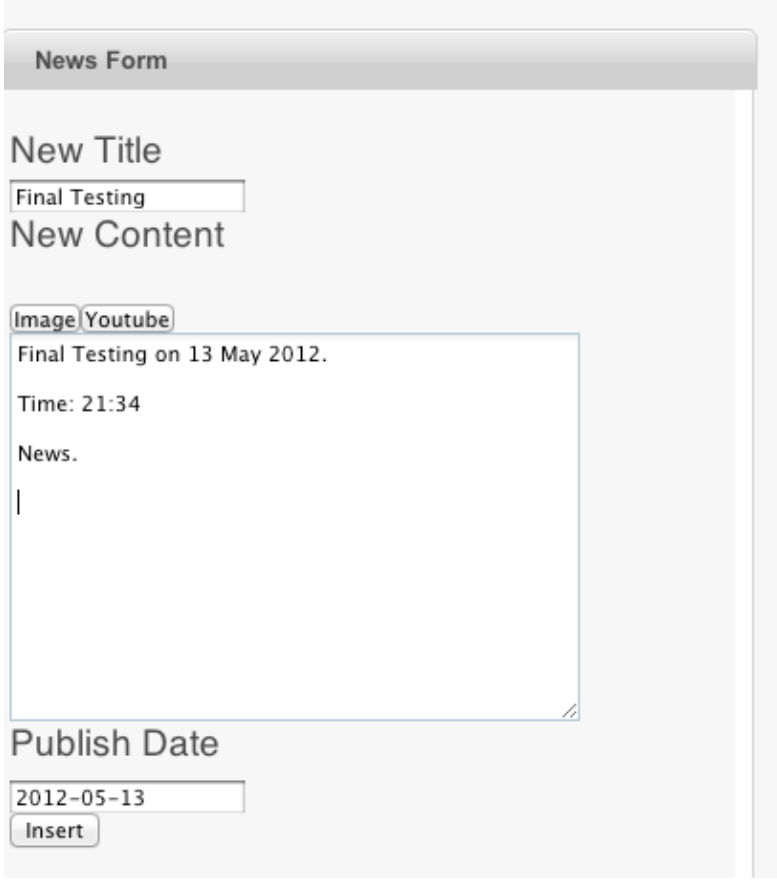
View Restaurant E-news

Test Case ID	Test Case Description	Expected Result
1	View Restaurant E-news	Successful view e-news



New restaurant news

Test Case ID	Test Case Description	Expected Result
2	Insert new Restaurant E-news	Successful insert e-news

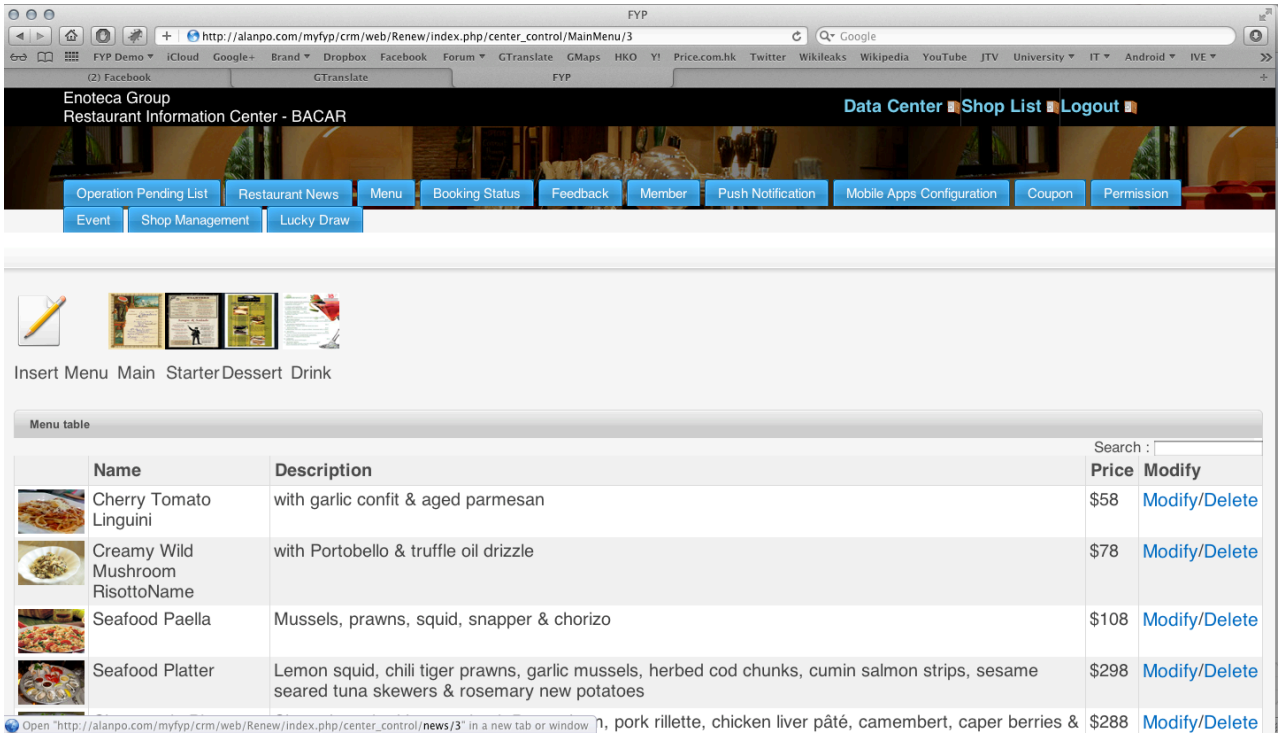


News table

News ID	Title	Content	Date	Update
51	Party Night	Night!!	2012-05-20	Modify / Delete
46	Ladies Night 18/5	Ladies Night on 18/5 Ladies has 5% discount!!	2012-05-16	Modify / Delete
47	Man's Night?	Man's Night on 19/5 Man's has 5% discount!!	2012-05-16	Modify / Delete
56	Menu Updated	Please go Check ~	2012-05-13	Modify / Delete
57	Final Testing	Final Testing on 13 May 2012. Time: 21:34 News.	2012-05-13	Modify / Delete
58	News Testing	Final testing on server function	2012-05-13	Modify / Delete

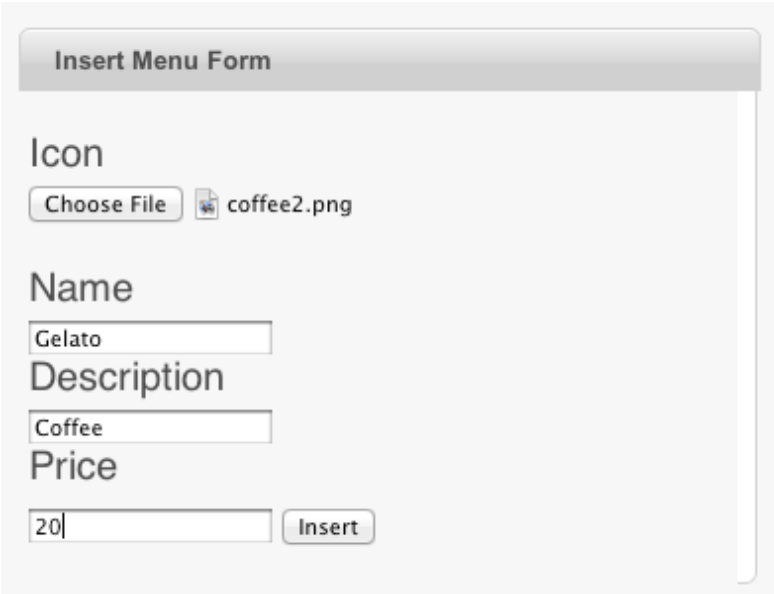
View E-Menu

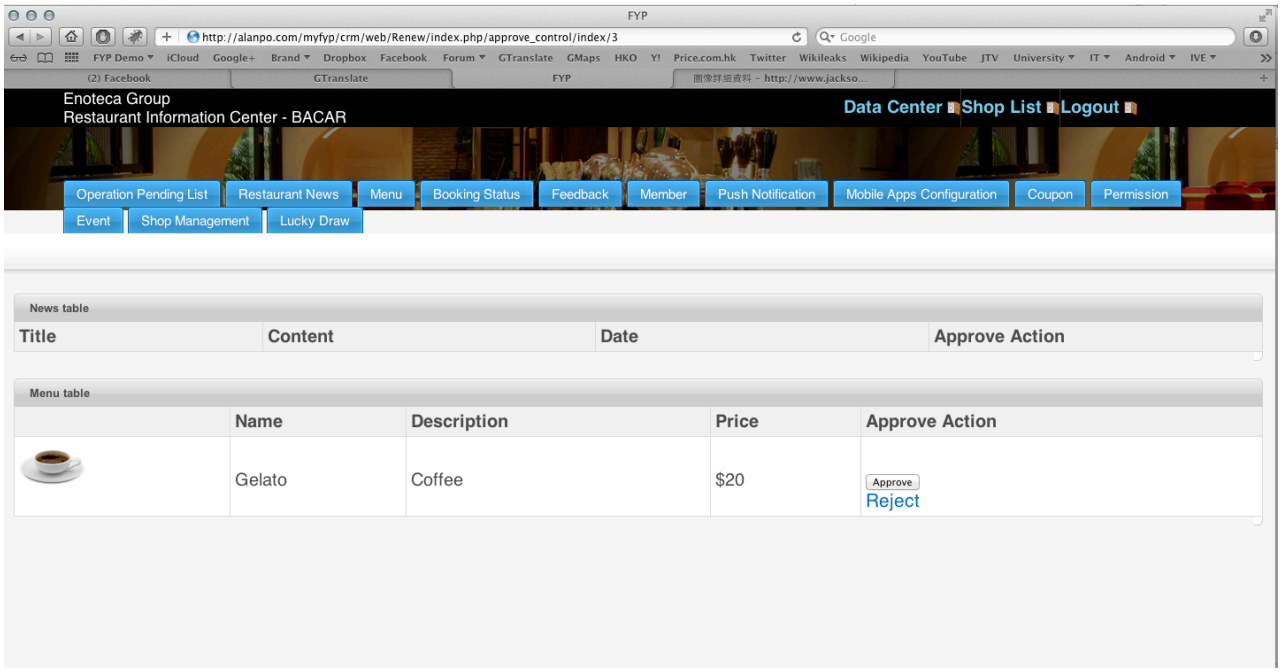
Test Case ID	Test Case Description	Expected Result
3	View E-menu	Successful view E-menu



New restaurant food

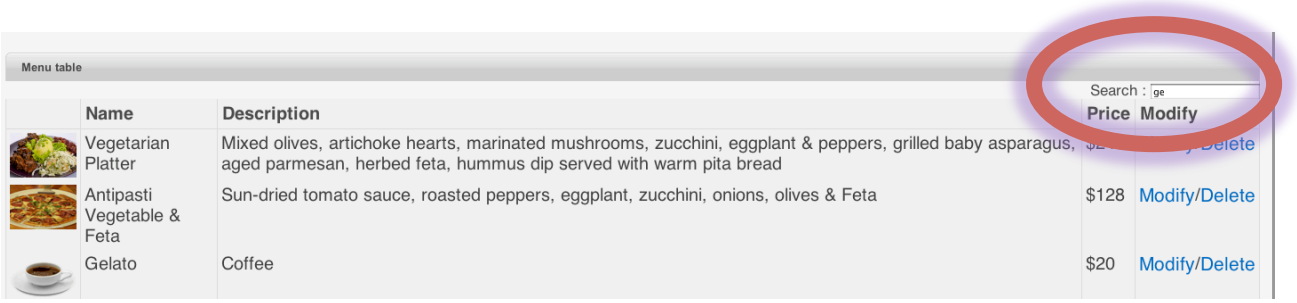
Test Case ID	Test Case Description	Expected Result
4	Create New Food E-menu	Successful create Food





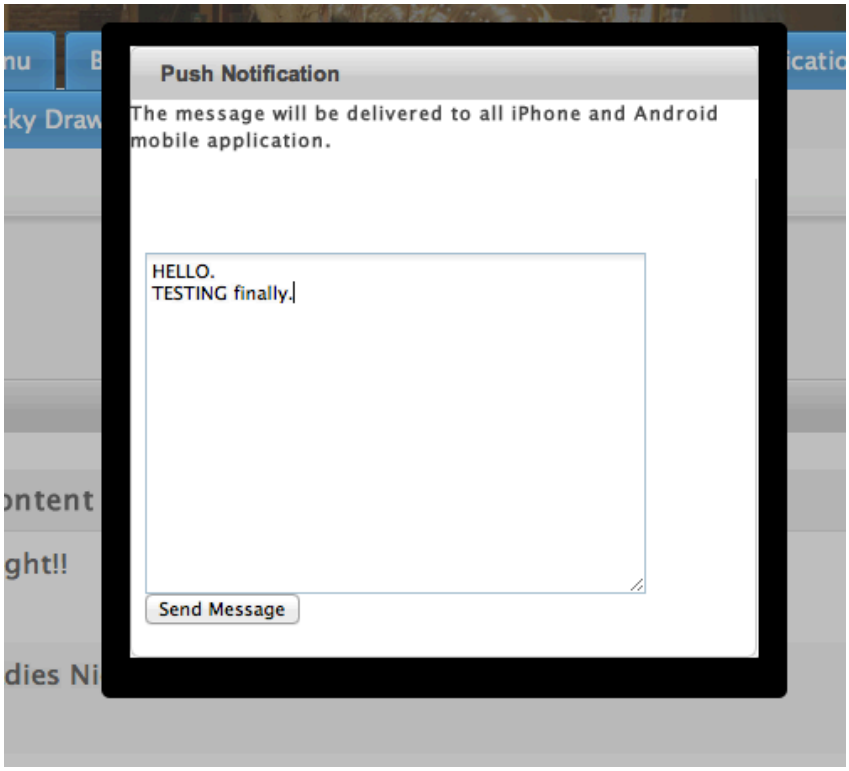
Search menu food

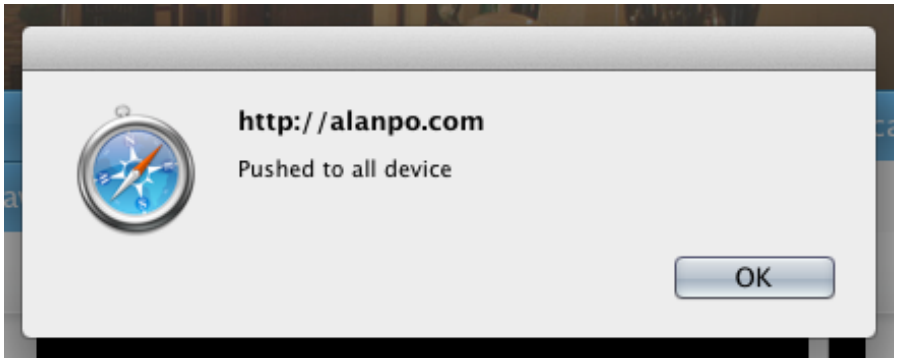
Test Case ID	Test Case Description	Expected Result
4	Search e-menu item	Successful search e-menu item



Send push notification

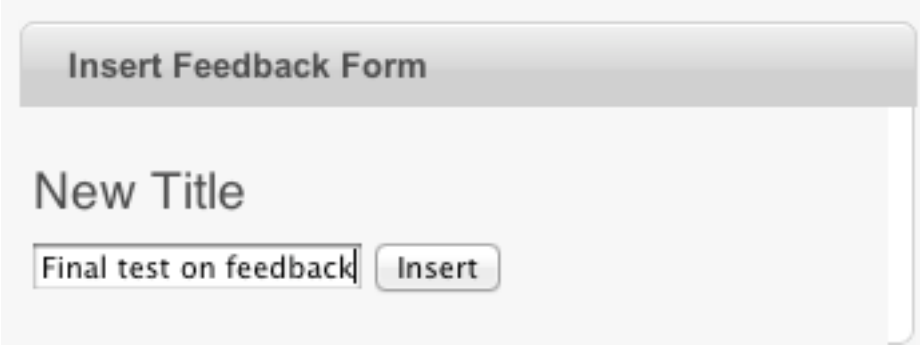
Test Case ID	Test Case Description	Expected Result
5	Send push notification to mobile	Successful send the message to mobile

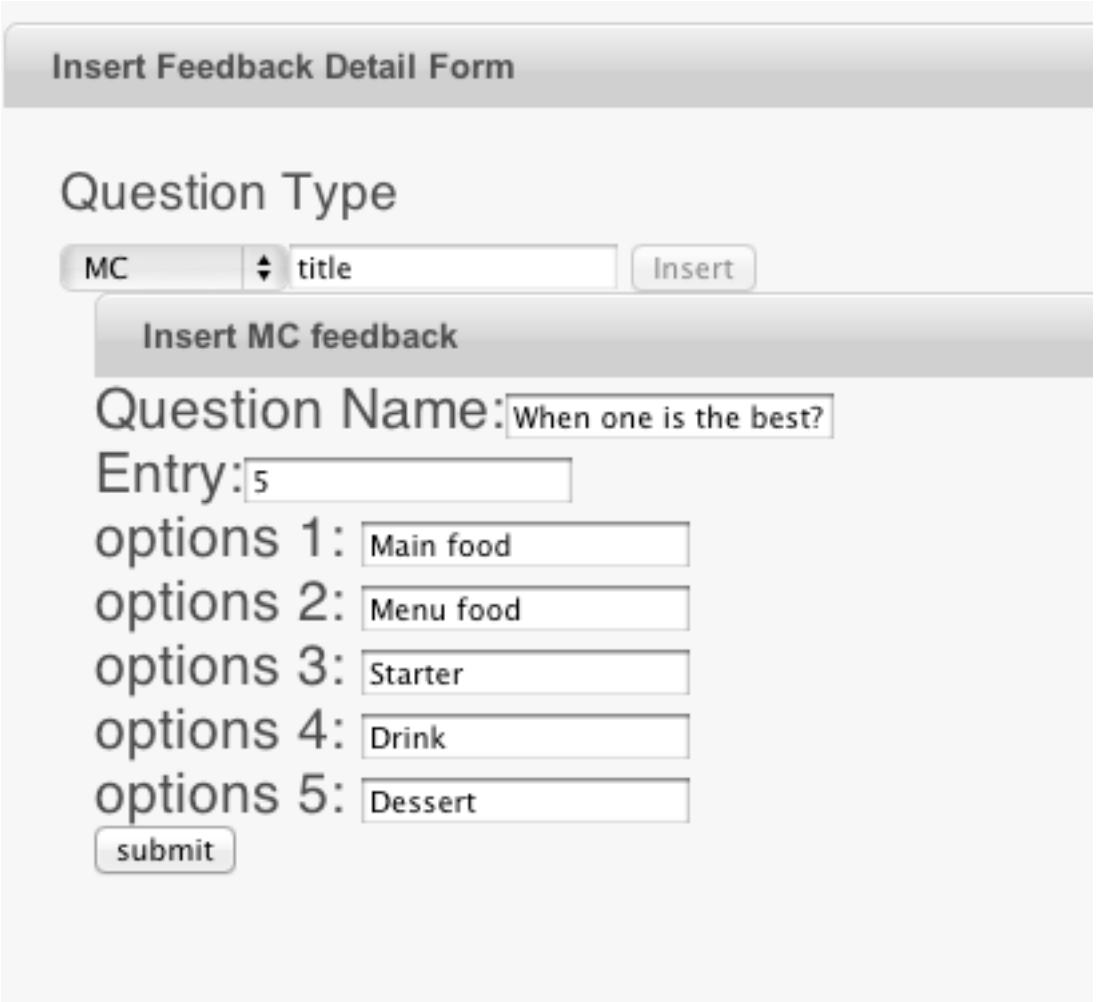




Insert new feedback

Test Case ID	Test Case Description	Expected Result
6	New feedback	New feedback added





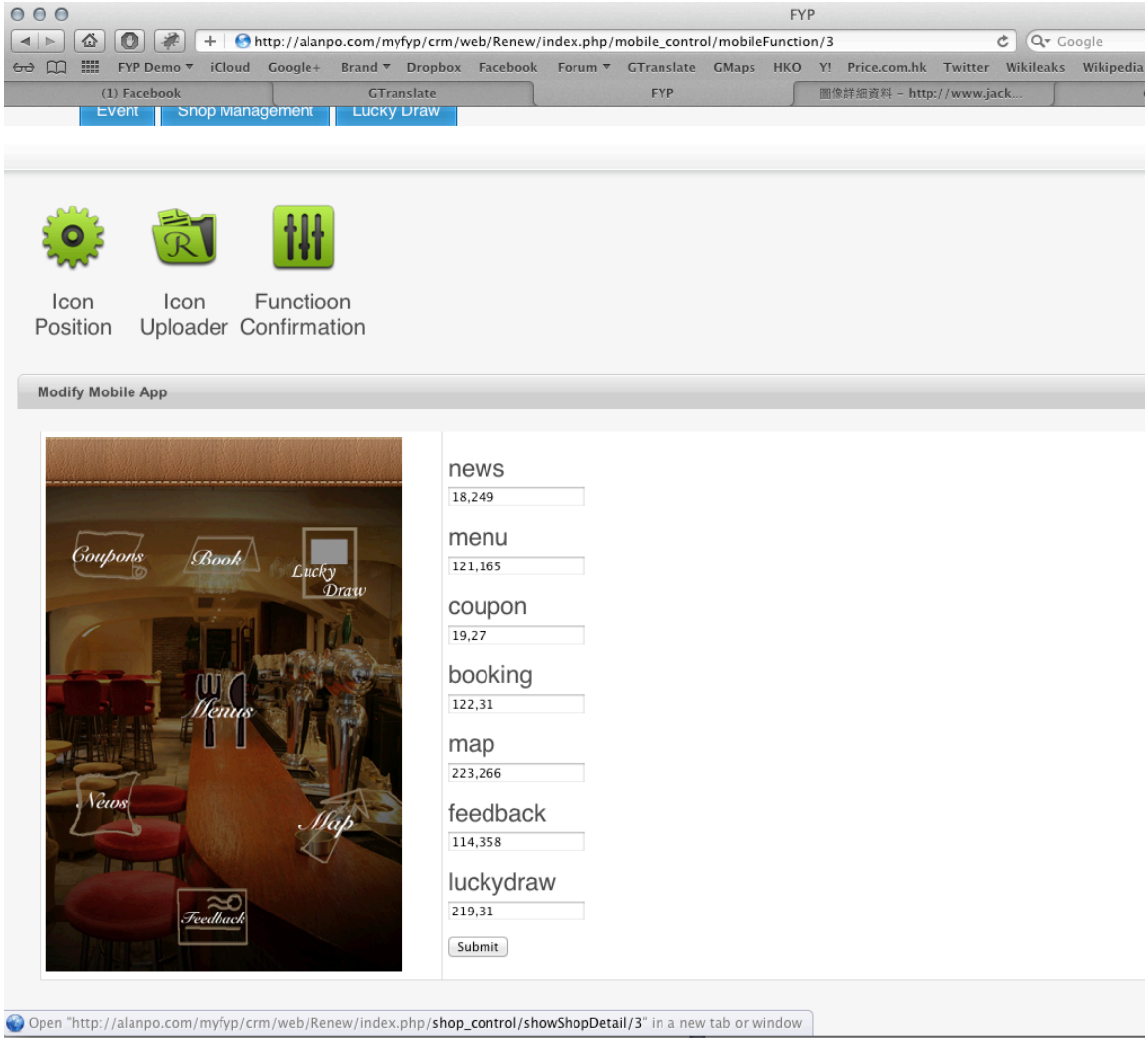
Inesrt Feedback Detail

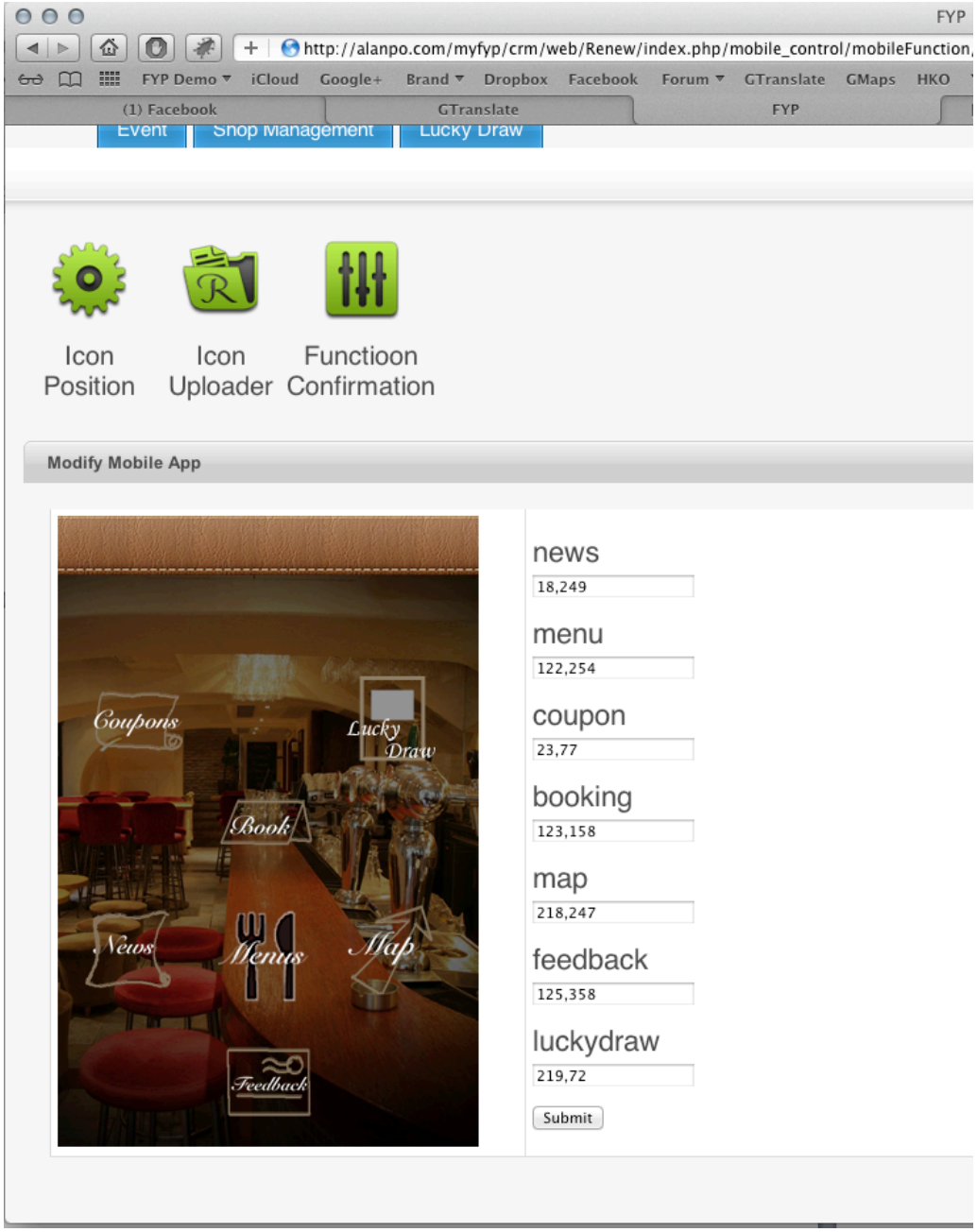
News table

Feedback ID	Feedback Question	Feedback Type	Delete
13	When one is the best?	MC	Delete question

Modify mobile app icon

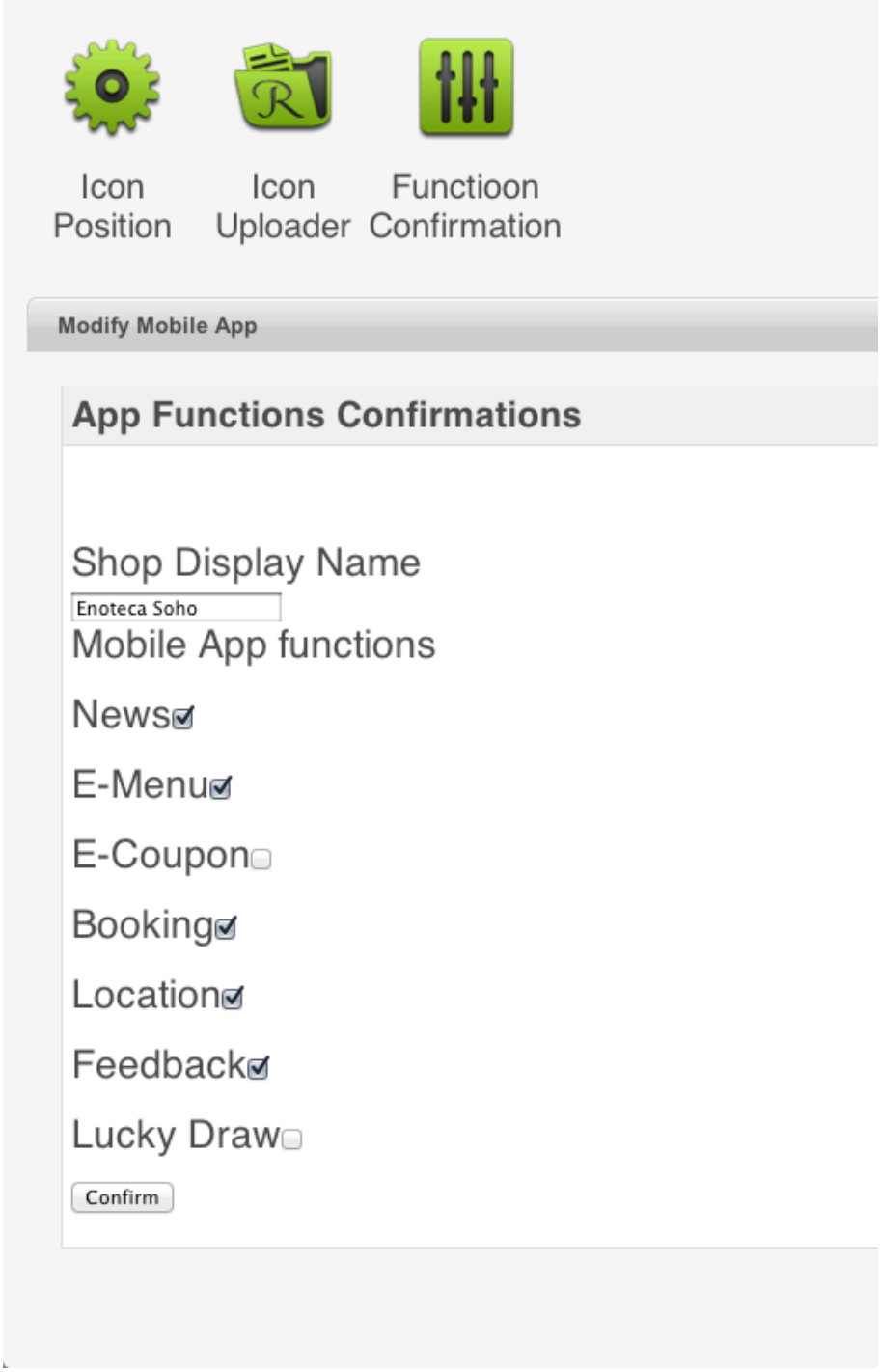
Test Case ID	Test Case Description	Expected Result
7	Modify Mobile app icon position	Icon position modified successfully




FYP
http://alanpo.com/myfyp/crm/web/Renew/index.php/mobile_control/mobileFunction
FYP Demo
iCloud
Google+
Brand
Dropbox
Facebook
Forum
GTranslate
GMaps
HKO
(1) Facebook
GTranslate
FYP
Event
Shop Management
Lucky Draw
Icon Position
Icon Uploader
Functioon Confirmation
Modify Mobile App
Coupons
Lucky Draw
Book
News
Menus
Map
Feedback
news
18,249
menu
122,254
coupon
23,77
booking
123,158
map
218,247
feedback
125,358
luckydraw
219,72
Submit

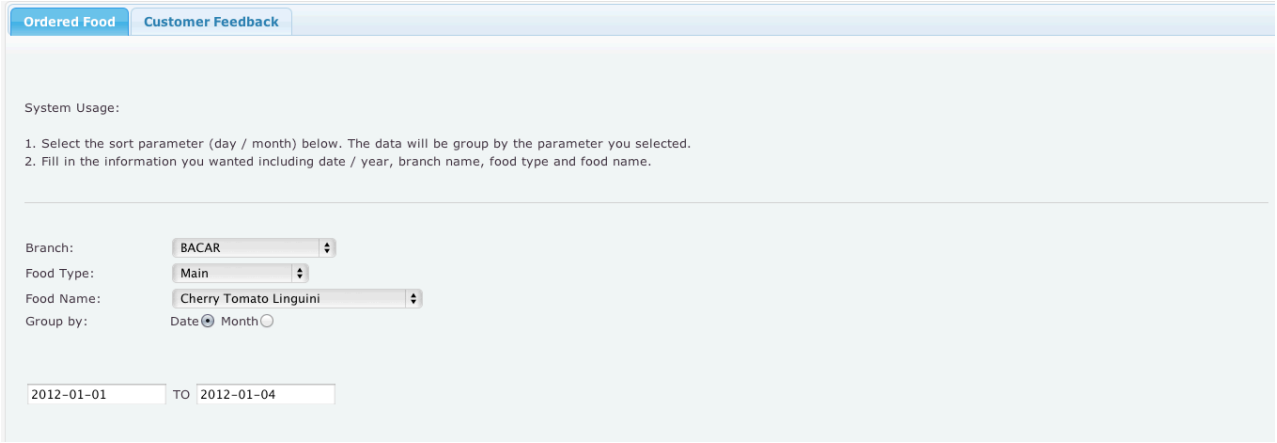
Modify mobile app function

Test Case ID	Test Case Description	Expected Result
8	Modify Mobile app function	Function modified successfully



Data analysis

Test Case ID	Test Case Description	Expected Result
9	Data analysis	Data shown successfully



Date / Month	Total Order Quantities	Total Sales($HKD)	Customer Rating :	5 stars	4 stars	3 stars	2 stars	1 stars	View Bar Chart
2012-Jan-01	12	$ 696		3	0	0	1	2	View Graph
2012-Jan-02	7	$ 406		0	2	1	0	1	View Graph
2012-Jan-03	20	$ 1, 160		3	3	4	0	1	View Graph
2012-Jan-04	10	$ 580		1	1	1	1	1	View Graph

Download Report

Ordered Food | Customer Feedback

System Usage:

1. Select the sort parameter (day / month) below. The data will be group by the parameter you selected.
2. Fill in the information you wanted including date / year, branch name, food type and food name.

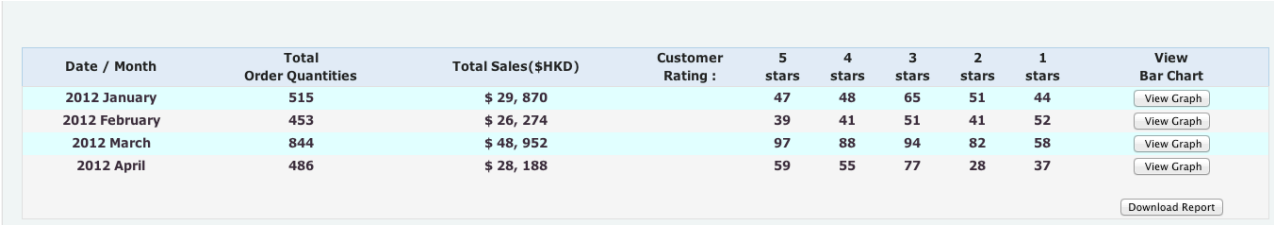
Branch: BACAR
Food Type: Main
Food Name: Cherry Tomato Linguini
Group by: Date ○ Month ◉

From January 2012 to May 2012

Date / Month	Total Order Quantities	Total Sales($HKD)	Customer Rating :	5 stars	4 stars	3 stars	2 stars	1 stars	View Bar Chart
2012 January	515	$ 29, 870		47	48	65	51	44	View Graph
2012 February	453	$ 26, 274		39	41	51	41	52	View Graph
2012 March	844	$ 48, 952		97	88	94	82	58	View Graph
2012 April	486	$ 28, 188		59	55	77	28	37	View Graph

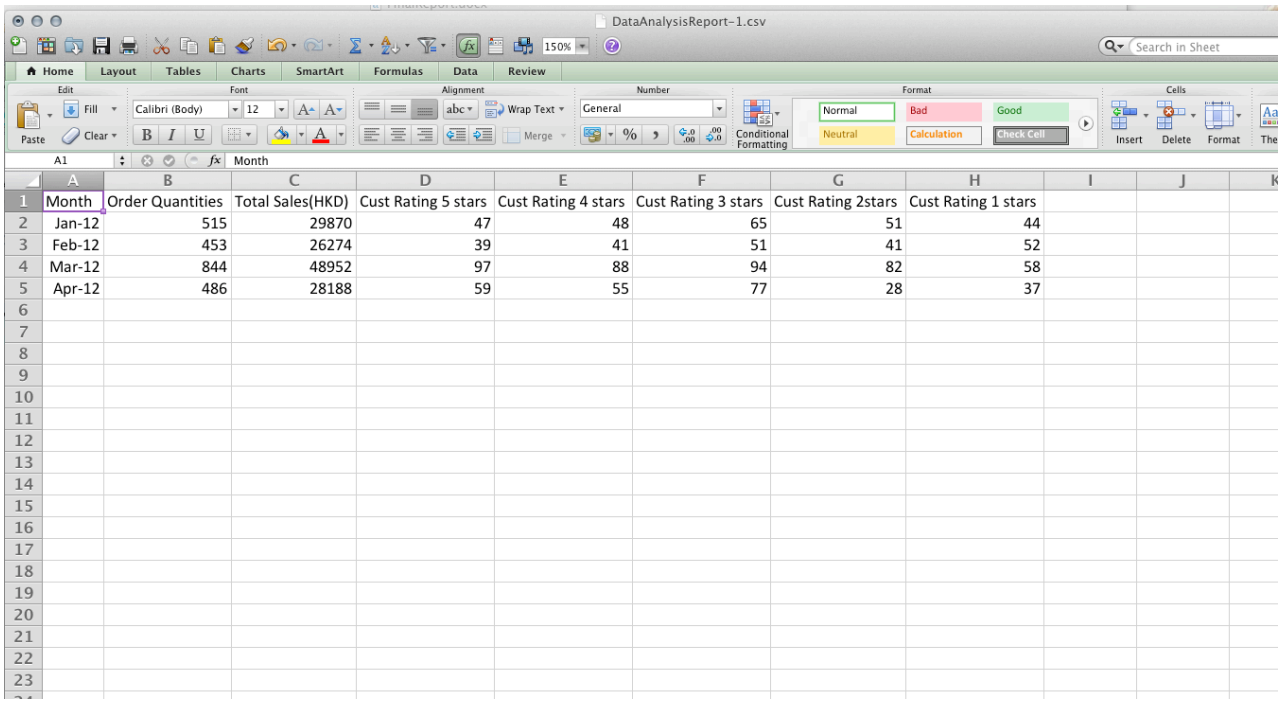
Download Report

Test Case ID	Test Case Description	Expected Result
10	Data analysis csv report	Report downloaded successfully



Date / Month	Total Order Quantities	Total Sales($HKD)	Customer Rating :	5 stars	4 stars	3 stars	2 stars	1 stars	View Bar Chart
2012 January	515	$ 29, 870		47	48	65	51	44	View Graph
2012 February	453	$ 26, 274		39	41	51	41	52	View Graph
2012 March	844	$ 48, 952		97	88	94	82	58	View Graph
2012 April	486	$ 28, 188		59	55	77	28	37	View Graph

Download Report



Month	Order Quantities	Total Sales(HKD)	Cust Rating 5 stars	Cust Rating 4 stars	Cust Rating 3 stars	Cust Rating 2stars	Cust Rating 1 stars
Jan-12	515	29870	47	48	65	51	44
Feb-12	453	26274	39	41	51	41	52
Mar-12	844	48952	97	88	94	82	58
Apr-12	486	28188	59	55	77	28	37

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.

CentOS
The Community ENTerprise Operating System

CodeIgniter®

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 lean 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

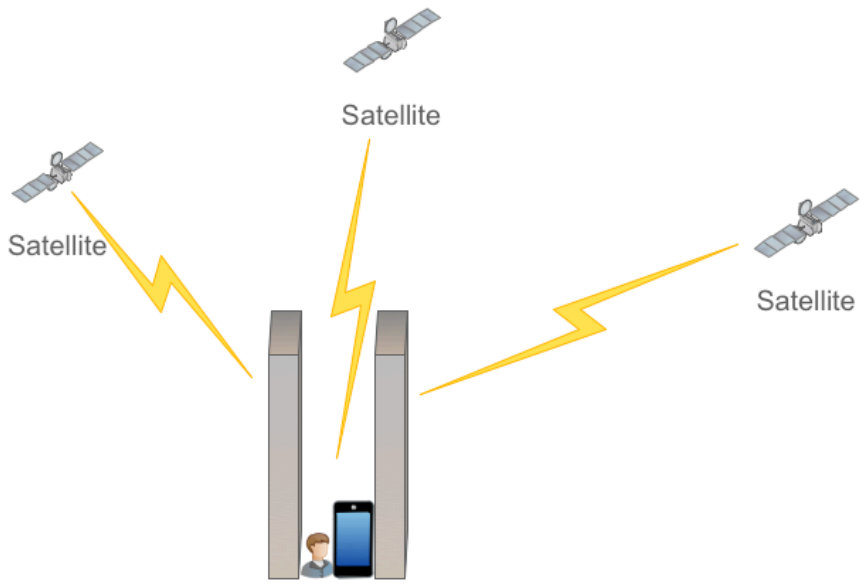
When we set up the server, sometimes we need to face some unknown problem or warning in Linux. We need to fix the problems in order to install the service to provide the stable server.

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.

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://plugins.jquery.com/

http://www.webtoolkit.info/ajax-file-upload.html

http://developers.facebook.com/

http://docs.oracle.com/javase/1.5.0/docs/api/

http://www.kavoir.com/2009/02/php-drop-down-list.html

http://www.webdeveloper.com/forum/showthread.php?t=172201

http://javascriptcalendar.org/javascript-date-picker.php

http://stackoverflow.com/questions/125113/php-code-to-convert-a-mysql-query-to-csv

http://snipplr.com/view/2234/

APPENDICES

A1. HKITAA Completion

hkbsdc.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

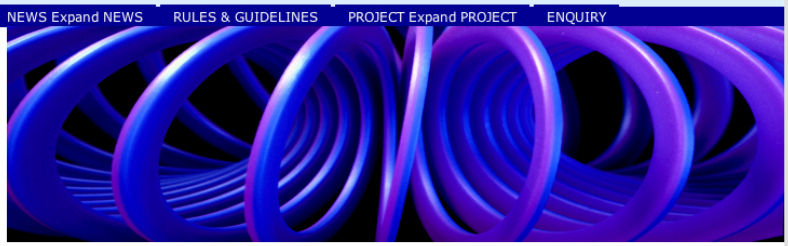
2012/05/25
Final Presentation

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

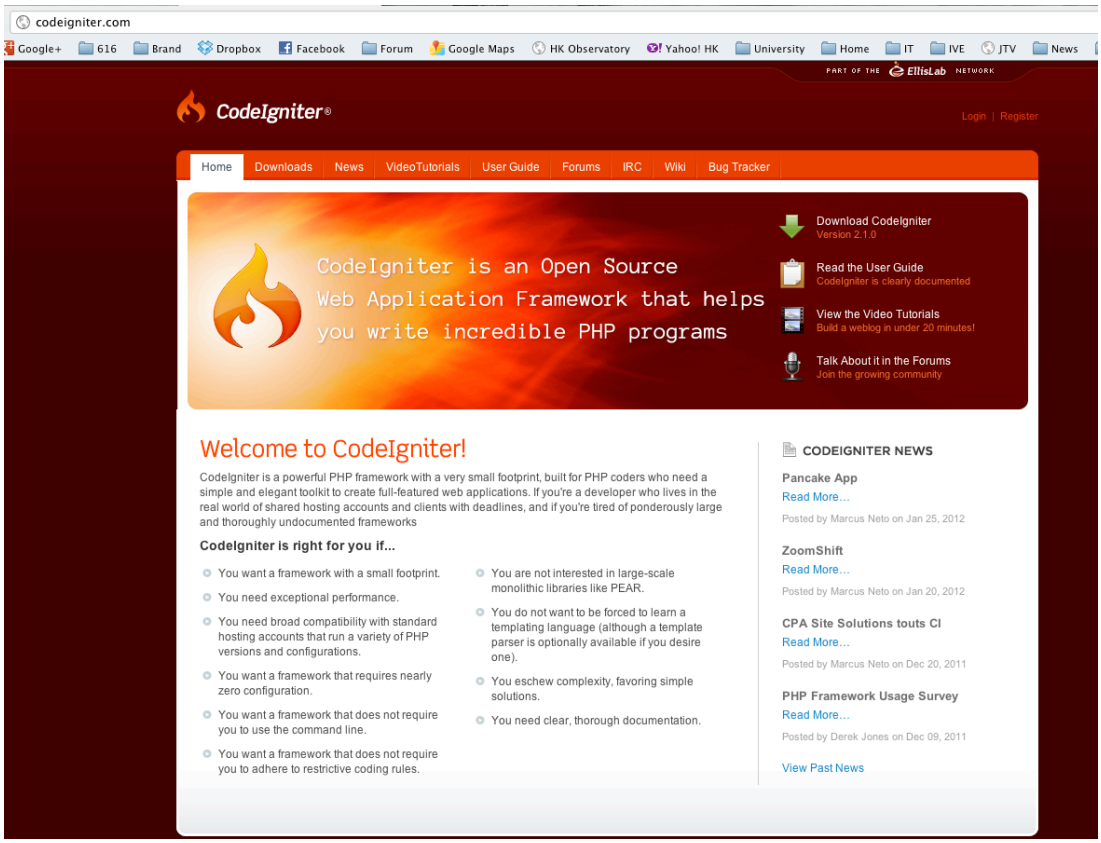
Phase I
Participants should submit a 2,000 words report

Phase II
Participants should deliver a presentation with live demo

Key Important Dates

30 Sep2011	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

A2. PHP Framework



A3. Facebook Developer

A4. JQuery

jQuery() - jQuery API

api.jquery.com/jQuery/

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

jQuery | UI | Mobile | Meetups | Forum | Blog | About | Donate

jQuery — write less, do more.

Download | Documentation | Tutorials | Bug Tracker | Discussion

JQUERY API

Search jQuery

jQuery API

- New or Changed in 1.7
- Raw XML API Dump
- Dynamic API Browser
- jQuery API Book

Browse the jQuery API

- All
- + Ajax
- Attributes
- Callbacks Object
- Core
- CSS
- Data
- Deferred Object
- Deprecated
- Dimensions
- + Effects
- + Events
- Forms
- Internals
- + Manipulation
- + Miscellaneous
- Offset
- + Plugins
- + Properties
- + Selectors

jQuery()

Contents:

Categories: Core

- jQuery(selector [, context])
  - jQuery(selector [, context])
  - jQuery(element)
  - jQuery(object)
  - jQuery(elementArray)
  - jQuery(jQuery object)
  - jQuery()
- jQuery(html [, ownerDocument])
  - jQuery(html [, ownerDocument])
  - jQuery(html, props)
- jQuery(callback)
  - jQuery(callback)

jQuery(selector [, context]) — *Returns: jQuery*

Description: *Accepts a string containing a CSS selector which is then used to match a set of elements.*

jQuery(selector [, context]) — version added: 1.0

selector A string containing a selector expression

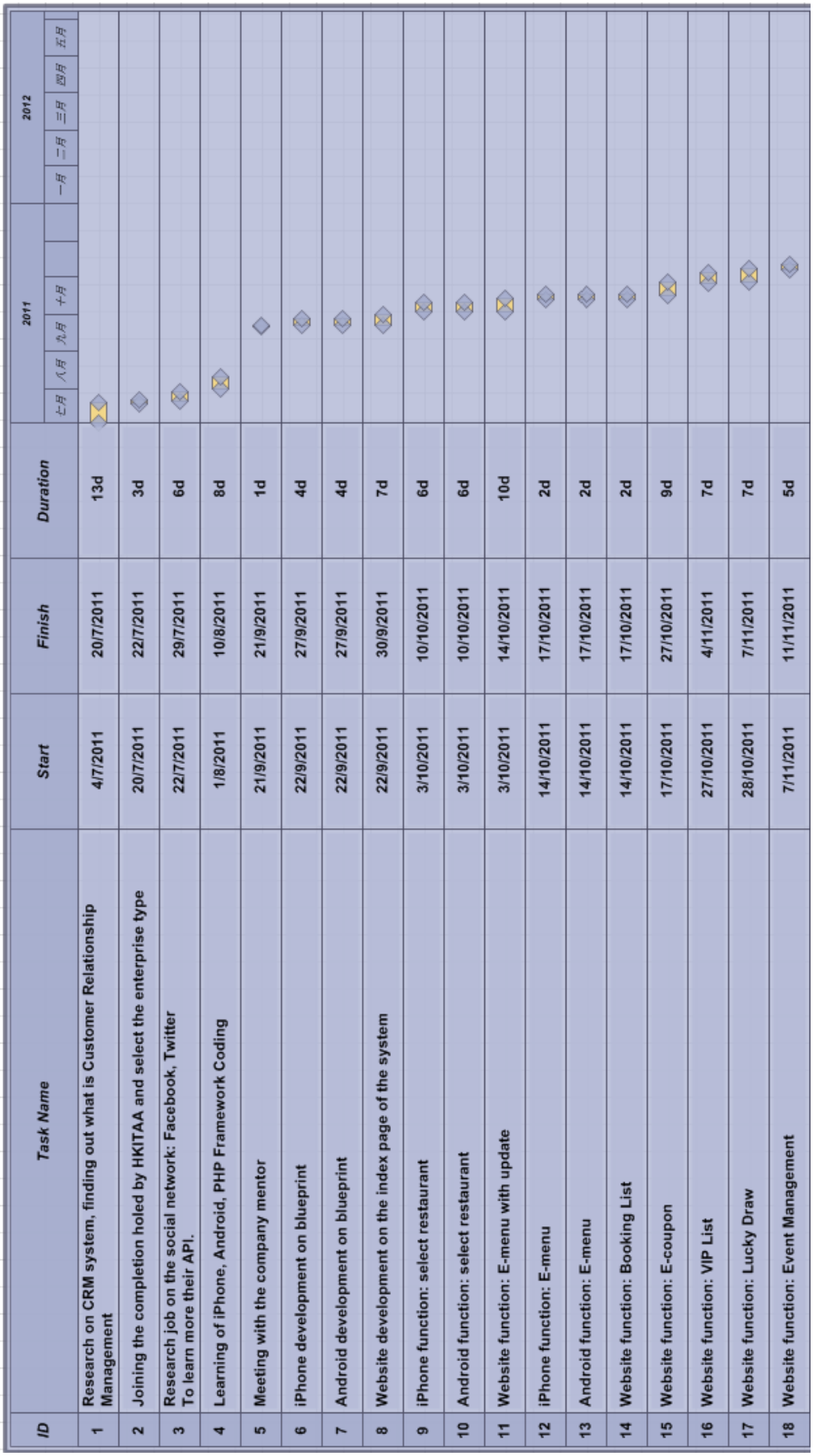
context A DOM Element, Document, or jQuery to use as context

jQuery(element) — version added: 1.0

element A DOM element to wrap in a jQuery object.

A5. Project Plan

ID	Task Name	Start	Finish	Duration
1	Research on CRM system, finding out what is Customer Relationship Management	4/7/2011	20/7/2011	13d
2	Joining the completion holed by HKITAA and select the enterprise type	20/7/2011	22/7/2011	3d
3	Research job on the social network: Facebook, Twitter To learn more their API.	22/7/2011	29/7/2011	6d
4	Learning of iPhone, Android, PHP Framework Coding	1/8/2011	10/8/2011	8d
5	Meeting with the company mentor	21/9/2011	21/9/2011	1d
6	iPhone development on blueprint	22/9/2011	27/9/2011	4d
7	Android development on blueprint	22/9/2011	27/9/2011	4d
8	Website development on the index page of the system	22/9/2011	30/9/2011	7d
9	iPhone function: select restaurant	3/10/2011	10/10/2011	6d
10	Android function: select restaurant	3/10/2011	10/10/2011	6d
11	Website function: E-menu with update	3/10/2011	14/10/2011	10d
12	iPhone function: E-menu	14/10/2011	17/10/2011	2d
13	Android function: E-menu	14/10/2011	17/10/2011	2d
14	Website function: Booking List	14/10/2011	17/10/2011	2d
15	Website function: E-coupon	17/10/2011	27/10/2011	9d
16	Website function: VIP List	27/10/2011	4/11/2011	7d
17	Website function: Lucky Draw	28/10/2011	7/11/2011	7d
18	Website function: Event Management	7/11/2011	11/11/2011	5d



19	Website function: Shop Management	14/11/2011	1/12/2011	14d
20	Website function: Mobile Configuration Management	12/12/2011	20/12/2011	7d
21	Website function: Push Notification	16/1/2012	16/1/2012	1d
22	Website function: Data Analysis	6/2/2012	13/4/2012	50d
23	Website function: Export of Data Analysis	13/4/2012	18/4/2012	4d
24	Website function: Permission Control	18/4/2012	20/4/2012	3d
25	iPhone function: select restaurant	17/10/2011	27/10/2011	9d
26	iPhone function: e-Menu	27/10/2011	4/11/2011	7d
27	iPhone function: e-News	28/10/2011	7/11/2011	7d
28	iPhone function: Lucky Draw	7/11/2011	11/11/2011	5d
29	iPhone function: Map Location	14/11/2011	1/12/2011	14d
30	iPhone function: Feedback	12/12/2011	20/12/2011	7d
31	iPhone function: Find restaurant	16/1/2012	26/1/2012	9d
32	Android function: select restaurant	6/2/2012	13/2/2012	6d
33	Android function: e-Menu	14/11/2011	1/12/2011	14d
34	Android function: Lucky Draw	14/11/2011	12/12/2011	21d
35	Android function: e-News	12/12/2011	20/12/2011	7d
36	Android function: Map Location	12/12/2011	12/1/2012	24d

37	Android function: Feedback	16/1/2012	9/3/2012	40d
38	Debug of the system	2/4/2012	4/5/2012	25d
39	Final Modify of the system	2/5/2012	7/5/2012	4d
40	Final Report	1/5/2012	11/5/2012	9d
41	Installation Guide and User Guide	1/5/2012	11/5/2012	9d
42	Installation Guide and User Guide	1/5/2012	11/5/2012	9d

A6. Job Division

Members	Job in this group project
CHAN Chun Wa	• Android Application development • Android Application maintenance • MQTT Service for the push notification on Android • Data Synchronization between Server and Android • Co-operation with other members' job
CHU Kai Hang	• Website development: Booking List, Modify E-News, Modify E-Menu, Modify E-Coupon, Lucky Draw Management, Event Management, Shop Management Pending List Management • Web service development • Web maintenance • Database Management • Co-operation with other members' job
KWOK Chun Hin	• iPhone Application development • iPhone Application maintenance • Data Synchronization between Server and iPhone • Co-operation with other members' job
TSANG Tsz Lok	• Website development: Push Notification on website, Permission Control, Data Analysis Center of Order Good, Export Data Analysis on CSV Data Analysis on Feedback Mobile Application Configuration Center • Web service development • Linux Server build up and maintenance • Database Management • Co-operation with other members' job

A7. Logsheet

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 01
Date: 06 JUL 2011 **Time:** 14:00	**Duration of meeting:** 90min
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
-NIL-	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
-NIL-	
Topics discussed in the meeting	
Develop Facebook Application: Facebook Page for company Research Work: Social Network in internet that collect user information - TSANG Tsz Lok CRM product, Database system nowadays- CHU Kai Hang iPhone App to collect user information from the iPhone - KWOK Chun Hin Android App to collect user information from the mobile - CHAN Chun Wa Remarks: getting data from address book on user mobile Target social enterprise type:	

1. Catering management 2. Manufacturing Company 3. Insurance Service Provider
Supervisor's comment on student's work
-NIL-
Plan of work before the next meeting
Research work

Students Signatures:	**Supervisor(s) Signature:**
---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 02
Date: 13 JUL 2011 **Time:** 11:15	**Duration of meeting:** 75mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
Research work in Facebook, Android mobile, ios mobile that receiving user information like address book, friends list.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
-NIL-	
Topics discussed in the meeting	
Confirm target social enterprise type: 1. Catering management 2. Manufacturing Company 3. Insurance Service Provider	
Supervisor's comment on student's work	
Good work	
Plan of work before the next meeting	
Research work	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 03
Date: 20 JUL 2011 **Time:** 11:30am	**Duration of meeting:** 60mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
Confirmation: CRM for Catering Management	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
-NIL-	
Topics discussed in the meeting	
Find out the promotion management methods in Facebook. (Also for new social networking: Google plus) The research work includes Electrical coupon design. [TSANG Tsz Lok & CHU Kai Hang] Make the Electrical coupon use 1 time only for membership. [All] Lucky drew activity on mobile and website. [All] Research for nowadays hotel, restaurant application that related to catering management on mobile and social network. Look deeply and try for their functions and mobility for reference to our project. [TSANG Tsz Lok] Research for booking system for membership or general people. [KWOK Chun Hin] Database Manager [KWOK Chun Hin] VIP tables reserve and queuing system, with floor plan editor. [CHAN Chun Wa] GPS location routing and push notification on mobile. [CHAN Chun Wa, KWOK	

Chun Hin]	
Supervisor's comment on student's work	
Good work	
Plan of work before the next meeting	
Next Meeting: 27 JUL 2011 (Thu) 11:30am	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 04
Date: 28 JUL 2011 **Time:** 16:00	**Duration of meeting:** 90 mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
Find out the promotion management methods in Facebook. The research work includes Electrical coupon design. Research for nowadays hotel, restaurant application that related to catering management on mobile and social network. Look deeply and try for their functions and mobility for reference to our project. Database Management	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
GPS location routing and push notification on mobile. (continuing) PHP Webpage integrating with Facebook user information (continuing)	
Topics discussed in the meeting	
Integration of android and iPhone. Proposal design for this project and company. Server Management and server manager which control the php server and others need.	
Supervisor's comment on student's work	

Good work	
Plan of work before the next meeting	
Next Meeting: 03 AUG 2011 (Wed) 11:00am	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 05
Date: 3 AUG 2011 **Time:** 11:00	**Duration of meeting:** 90 mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
Collect customer information in Facebook. Push notification in Android mobile and iPhone.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Proposal design for this project and company. Server Management and server manager, which control the php server and others need.	
Topics discussed in the meeting	
Continue the Proposal and should be finished it before 8 AUG 2011. Supervisor review student job in this project.	
Supervisor's comment on student's work	
Good work. And wait for 8 AUG 2011 to apply the application of ITAA Software Development Competition.	
Plan of work before the next meeting	
Next Meeting: 11 AUG 2011 (Thu) 11:00am	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 06
Date: 11 AUG 2011 **Time:** 11:00	**Duration of meeting:** 60 mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
Collect customer information in Facebook. Push notification in Android mobile and iPhone. Linux Cent OS 5.6 Server is built and installed all needed service.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Proposal design for this project and company. (Continuing)	
Topics discussed in the meeting	
Continue the Proposal and should be finished it before early SEP 2011. Supervisor review student job in this project.	
Supervisor's comment on student's work	
Good work. And still wait for the reply of the application of ITAA Software Development Competition.	
Plan of work before the next meeting	
Next Meeting: 26 AUG 2011 (Fri) 11:00am	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 07
Date: 26 AUG 2011 **Time:** 11:00	**Duration of meeting:** 60 mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
Linux Cent OS 5.6 Server is built and installed all needed service.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Proposal design for this project and company. (Continuing)	
Topics discussed in the meeting	
Continue the Proposal and should be finished it before early SEP 2011. Supervisor review student job in this project.	
Supervisor's comment on student's work	
Good work. And still wait for the reply of the application of ITAA Software Development Competition.	
Plan of work before the next meeting	
Next Meeting: 7 SEP 2011 (Wed)	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 08
Date: 15 SEP 2011 **Time:** 10:30am	**Duration of meeting:** 90 mins

Project Title:
ITAA Software Development Competition (July 2011- May 2012)
Customer Relationship Management System (CRM) in Catering Management

Work done since last meeting

All the demonstration work to the company mentor:

- Android push notification, collect user habits, collect user data, location routing, e-couple, company news
- iOS push notification, company news, e-couple, collect user information and habits, company news
- Data synchronization with our database
- Website for customer feedback, Facebook page, report generation

State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)

Topics discussed in the meeting

The company mentor confirmed our system and mobile application design, and said that our group had given a good presentation and well preparation.

The privacy problem of the user data collection on mobile devices and Facebook. We think that we should only collect the user basic information but not the contact list. All the data we collect should keep it in a safety place. They thought that we should get the user permission before collecting their information.

We suggested that after we have developed the application, they could use it for their customer to test the usefulness of the application and report generation. Also, to collect customer opinions on the application.

Supervisor's comment on student's work	
Presentation is good. Demonstrations is good.	
Plan of work before the next meeting	
Next meeting : 21 SEP 2011 3:30PM	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 09
Date: 21 SEP 2011 **Time:** 15:30	**Duration of meeting:** 90mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
We will adopt using cloud technology to start our CRM project. We will create all the function in the iOS and android application first, make a application as a full template including all the function. After we finish developing the iOS Application and Android Application, we will build up a website to let the end user create a xCode project with our iOS application and android application. By using cloud technology, we can let the user to edit what function, GUI, customer feedback that they need and then generate they project.	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	
Next meeting: 28 SEP 2011	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 10
Date: 28 SEP 2011 **Time:** 16:20	**Duration of meeting:** 75mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
Queuing function in iOS and android for table reservation.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
New function will added to the project: - Queuing function on mobile for table reservation - Server side (company) may need a php to view the table reservation and confirm the reservation; push notification would be send to notice the customer. - Review the customer feedback function	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	
Next meeting: 7 Oct 2011 in HKDI/LWL at 2:30 pm	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 11
Date: 12 OCT 2011 **Time:** 14:30	**Duration of meeting:** 60mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
- Queuing function on mobile for table reservation	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
- Server side (company) may need a php to view the table reservation and confirm the reservation; push notification would be send to notice the customer. - Review the customer feedback function	
Topics discussed in the meeting	
• Continue the process in iOS and Android development • Begin work on the web-view side. • Web function including: Data-mining Mobile function and information maintenances iOS and Android Project or App web generation • Adapt cloud computing technology in iOS and Android Project web generation.	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	

Next meeting: 19 Oct 2011 in B314(IVE-TY)	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 12
Date: 19 OCT 2011 **Time:** 14:30	**Duration of meeting:** 90mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
- iOS Application development - Android Application development - Website for administration user	
Topics discussed in the meeting	
• Using PHP Framework CodeIgniter for website development. • Initial Report Blueprint • Preparation of presentation for company	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	
Next meeting: 26 Oct 2011 in B312(IVE-TY)	

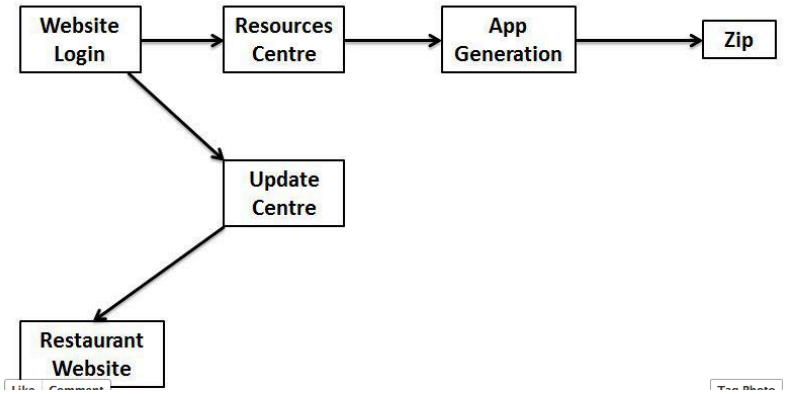
Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 13
Date: 26 OCT 2011 **Time:** 2:30pm	**Duration of meeting:** 90mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) Customer Relationship Management System (CRM)	
Work done since last meeting	
Built up PHP Framework service Website Login and Push notification to smart devices.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
- Website - iPhone Application - Android Application	
Topics discussed in the meeting	
• We will create Web Resource Center. • We will create Website for Restaurant.	



Supervisor's comment on student's work	
Good work	
Plan of work before the next meeting	

Students Signatures:	**Supervisor(s) Signature:**
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 14
Date: 2 NOV 2011 **Time:** 14:30	**Duration of meeting:** 90mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) renamed to Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
- Built up PHP Framework environment - Following the initial report - Preparation for the presentation to the company mentor	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
- iOS Application development should be continue - Android Application development should be continue - Website for administration user	
Topics discussed in the meeting	
• Mobile devices need to support multi restaurant • Mobile devices basic function need to be finish • Re-confirm the functions of mobile app: E-menu, News, Map Location, Coupon, Feedback, Booking	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	

Next meeting: 9 Nov 2011 in B314(IVE-TY)	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 15
Date: 9 NOV 2011 **Time:** 14:30	**Duration of meeting:** 90mins
Project Title: ITAA Software Development Competition (July 2011- May 2012) renamed to Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
- Built up PHP Framework environment - Basic database built up. - Mobile app basic functions complete and connected to the database.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
- Mobile app need to continue develop and modify - Website need to continue development	
Topics discussed in the meeting	
• Mobile devices need to let the user select restaurant before using any of the functions. • Website should start developing the app configuration center. • Functions in the app configuration center: modify app icon, push notification, icon location on the mobile, functions enabler	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	

Next meeting: 16 Nov 2011 in B314(IVE-TY)	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 16
Date: 16 NOV 2011 **Time:** 14:30	**Duration of meeting:** 90mins

Project Title:
ITAA Software Development Competition (July 2011- May 2012)
renamed to Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)

Work done since last meeting

- Mobile devices can let the user select restaurant before using any functions.
- App configuration center partially finished: modify app icon, push notification.

State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)

- App configuration center functions: icon location on the mobile, functions enabler.

Topics discussed in the meeting

- Mobile apps: Android and iPhone should share the same data in the database and also the json file in the server. Emenu and News should be divided into different group of data.
- Website should finish the modify the icon location and functions enabler functions.

Supervisor's comment on student's work

Good work.

Plan of work before the next meeting

Next meeting:

23 Nov 2011 in B314(IVE-TY)

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 17
Date: 23 NOV 2011 **Time:** 15:00	**Duration of meeting:** 75mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
- Mobile devices sharing the same data. - Mobile apps E-Menu and News division finished. - Website functions: icon location and functions enabler.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
• Website start to develop the restaurant information center. • E-coupon on the mobile app need user login. • Mobile app needed to provide the user register function.	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	
Next meeting: 30 Nov 2011 3:00pm	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 18
Date: 30 NOV 2011 **Time:** 15:00	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
- Restaurant Information Center function: restaurant e-menu completely shown.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• Restaurant information Center function: location, booking confirm, create coupon.	
Topics discussed in the meeting	
• Restaurant Information Center function: restaurant e-menu, location, booking confirm, create coupon. • Mobile app continues development and strength the function.	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	
Next meeting: 7 DEC 2011 3:00pm	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 19
Date: 7 DEC 2011 **Time:** 15:00	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
- Restaurant Information Center function: restaurant e-menu completely shown.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• Restaurant information Center function: location, booking confirm, create coupon.	
Topics discussed in the meeting	
• Restaurant e-menu need to include the modify, delete function • Restaurant e-menu need to be confirm. • Restaurant e-menu can let the user upload the new image.	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	
Next meeting: 20 DEC 2011 3:30pm	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 20
Date: 20 DEC 2011 **Time:** 15:30	**Duration of meeting:** 60mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
- Restaurant Information Center function modifying e-menu.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• Restaurant information Center functions: location, booking confirm, create coupon.	
Topics discussed in the meeting	
• Finish the following function in the Christmas holiday: - Restaurant location can be editable on website with Google map API - Restaurant booking confirmation and rejection - Restaurant create coupon	
Supervisor's comment on student's work	
Good work.	
Plan of work before the next meeting	
Next meeting: 30 JAN 2011 14:30	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 21
Date: 30 JAN 2012 **Time:** 14:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
Restaurant information Center functions: location, booking confirm, create coupon.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
- Restaurant Information Center - Mobile app The above needed to be continue to develop and modify	
Topics discussed in the meeting	
• The user interface need to be improved or maybe change the whole set of user interface to another jQuery, color, or design a new set of user interface. • The user interface of the mobile app should be improved, the theme is boring and the table view need to make 3D view with shadow. • Preparation work for the Interim report and Interim presentation.	
Supervisor's comment on student's work	
Fair work in the holiday.	
Plan of work before the next meeting	

07 FEB 2012 14:00	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 22
Date: 07 FEB 2012 **Time:** 14:00	**Duration of meeting:** 75mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
Restaurant information Center functions: location, booking confirm, create coupon.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• User interface improvement of Website • User interface improvement of Mobile App	
Topics discussed in the meeting	
• Presentation powerpoint needed for tomorrow presentation • Demonstrations technic • Demonstration functions	
Supervisor's comment on student's work	
We should well prepare for the functions in the demonstration process and a localhost server need to be used instead of using the web server.	
Plan of work before the next meeting	
14 FEB 2012 11:30	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 23
Date: 14 FEB 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
Interim Presentation and Demonstration	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
1. For the website, the mobile application center and the app configuration center will be combined together. 2. Mobile app booking information can be partially auto complete by getting the information from the mobile devices by the user. 3. The user interface of Date and Time picker on Android should be improved.	
Supervisor's comment on student's work	
Including the website and mobile app, the user interface should be re-designed with user-friendly issue.	
Plan of work before the next meeting	
21 FEB 2012 11:30	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 24
Date: 21 FEB 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• The mobile application center and the app configuration center will be combined together. • Mobile app booking information can be partially auto complete by getting the information from the mobile devices by the user.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
4. The user does not need to choose restaurant, the restaurant information should related to the user's session. When the user login, the server will load information belongs to his account. 5. Mobile app the news's effect with the image "NEW" in the corner of the icon. 	
Supervisor's comment on student's work	
Good work	
Plan of work before the next meeting	

6 MAR 2012 11:30	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 25
Date: 6 MAR 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• Mobile app the news's effect with the image "NEW" in the corner of the icon.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
1. Mobile App image should contain shadow effect. 2. Mobile App feedback rating should add emo icon. 3. Mobile App feedback needed to be configurable on the website.	
Supervisor's comment on student's work	
Good work	
Plan of work before the next meeting	
15 MAR 2012 11:30	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 26
Date: 15 MAR 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• Mobile App image should contain shadow effect. • Mobile App feedback rating should add emo icon. • Mobile App feedback needed to be configurable on the website.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
4. Mobile App functions' icon should be redesigned 5. Start Working in data analysis	
Supervisor's comment on student's work	
Good work	
Plan of work before the next meeting	
20 MAR 2012 11:30	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 27
Date: 20 MAR 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• Mobile App functions' icon redesigned	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• Data Analysis sub-system	
Topics discussed in the meeting	
6. Data Analysis including two parts: - Order Food with customer rating - Feedback received from mobile app	
Supervisor's comment on student's work	
Good work	
Plan of work before the next meeting	
27 MAR 2012 11:30	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 28
Date: 27 MAR 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• Data Analysis center	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• Data Analysis should be modified	
Topics discussed in the meeting	
7. Data Analysis: - data can be group by Monthly or Daily - auto change when switch the button monthly or daily	
Supervisor's comment on student's work	
Good work	
Plan of work before the next meeting	
06 APR 2012 14:00 Meeting with a catering company CAFÉ DECO GROUP to listen to their opinions.	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 29
Date: 06 APR 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• Data Analysis center	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
We gain opinions from the company, they suggest we finish the following items: • China market is also important, the mobile app can support 微博 • Suggest another time slot for booking if unavailable • Reject booking for special event • Require feedback to customer if the booking is unsuccessful • Booking reminder before booking time • Try to avoid the people to book for a lot of timeslot in the same day • Waiting list for booking • How long need to wait for this booking, how many tables in the restaurant • Switch to another restaurant. • One officer to handle the program of booking. • Export the data analysis to excel file (cvs format) • Deadline for booking not available	

- Under membership we can store the preference
- Integrate to reservation or pos system? (Simulate) web services
- Special requirement system. Window sit, child seat,
- Different restaurant request different environment and situation.
- Detail message for the push notification, link up to booking or coupon.
- Map for MTR exit that is close to restaurant. To auto search the location of the close restaurant.
- Any special way to keep this app on the iPhone.
- 1 restaurant can introduce another restaurants.
- What is the target of our app?
- Menu design problem? No database in the market. Need to approval workflow.
- Expire date on the coupon.
- Use the term VIP instead of membership.
- Eng, Chinese, Simple Chinese, mainland ISSUE!
- Feedback form is tailor made from website.
- Comment requires username.
- Birthday event, redeem marks.

Supervisor's comment on student's work	
Work hand in the coming month	
Plan of work before the next meeting	
11 APR 2012 14:30 in IVE	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 30
Date: 11 APR 2012 **Time:** 14:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• China market is also important, the mobile app can support 微博 • Suggest another time slot for booking if unavailable • Reject booking for special event • Require feedback to customer if the booking is unsuccessful • Booking reminder before booking time • Try to avoid the people to book for a lot of timeslot in the same day • Waiting list for booking • One officer to handle the program of booking.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• Export the data analysis to excel file (cvs format) • Deadline for booking not available • Under membership we can store the preference • Integrate to reservation or pos system? (Simulate) web services • Special requirement system. Window sit, child seat, • Different restaurant request different environment and situation. • Detail message for the push notification, link up to booking or coupon. • Different restaurant request different environment and situation. • Detail message for the push notification, link up to booking or coupon.	

- Map for MTR exit that is close to restaurant. To auto search the location of the close restaurant.
- Any special way to keep this app on the iPhone.
- 1 restaurant can introduce another restaurants.
- What is the target of our app?
- Menu design problem? No database in the market. Need to approval workflow.
- Expire date on the coupon.
- Use the term VIP instead of membership.
- Eng, Chinese, Simple Chinese, mainland ISSUE!
- Feedback form is tailor made from website.
- Comment requires username.
- Birthday event, redeem marks.

Topics discussed in the meeting
Continue to work
Supervisor's comment on student's work
Work hand.
Plan of work before the next meeting
17 APR 2012 14:30 in IVE

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 31
Date: 17 APR 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• Export the data analysis to excel file (cvs format) • Deadline for booking not available • Under membership we can store the preference • Integrate to reservation or pos system? (Simulate) web services • Special requirement system. Window sit, child seat, • Different restaurant request different environment and situation. • Detail message for the push notification, link up to booking or coupon. • Different restaurant request different environment and situation. • Detail message for the push notification, link up to booking or coupon. • What is the target of our app?	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• Map for MTR exit that is close to restaurant. To auto search the location of the close restaurant. • Any special way to keep this app on the iPhone. • Menu design problem? No database in the market. Need to approval workflow. • Expire date on the coupon. • Use the term VIP instead of membership.	

- Eng, Chinese, Simple Chinese, mainland ISSUE!
- Feedback form is tailor made from website.
- Comment requires username.
- Birthday event, redeem marks.

Topics discussed in the meeting

Continue to work

Supervisor's comment on student's work

Work hand.

Plan of work before the next meeting

24 APR 2012 11:30 in IVE

Students Signatures:	**Supervisor(s) Signature:**
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 32
Date: 24 APR 2012 **Time:** 11:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
• Map for MTR exit that is close to restaurant. To auto search the location of the close restaurant. • Any special way to keep this app on the iPhone. • Menu design problem? No database in the market. Need to approval workflow. • Expire date on the coupon. • Use the term VIP instead of membership. • Eng, Chinese, Simple Chinese, mainland ISSUE! • Feedback form is tailor made from website.	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
• Comment requires username. • Birthday event, redeem marks.	
Topics discussed in the meeting	
Continue to work	
Supervisor's comment on student's work	
Work hand.	

Plan of work before the next meeting	
2 MAY 2012 15:30 in IVE	
Students Signatures: ---------- (TSANG Tsz Lok) ---------- (CHU Kai Hang) ---------- (KWOK Chun Hin) ---------- (CHAN Chun Wa)	**Supervisor(s) Signature:** ---------- (PO Lun Alan) ---------- ( )

Hong Kong Institute of Vocational Education (Tsing Yi)

HD in Computer Systems Administration

Final Year Project (ITD3309) - Log Sheet

Students Names: TSANG Tsz Lok, CHU Kai Hang, KWOK Chun Hin, CHAN Chun Wa	**Sheet no:** 33
Date: 5 MAY 2012 **Time:** 15:30	**Duration of meeting:** 90mins
Project Title: Hong Kong Business Software Development Competition (HKBSDC) - Customer Relationship Management System (CRM)	
Work done since last meeting	
State which part of the last plan not yet finished (leave it blank if you finished all the work of the last plan)	
Topics discussed in the meeting	
We should start prepare the final report, final demo.	
Supervisor's comment on student's work	
Work hand and good luck.	
Plan of work before the next meeting	

Students Signatures:	Supervisor(s) Signature:
---------- (TSANG Tsz Lok)	---------- (PO Lun Alan)
---------- (CHU Kai Hang)	---------- ( )
---------- (KWOK Chun Hin)	
---------- (CHAN Chun Wa)	

PROGRAM LISTING

```
<?php
	session_start();
	  header("Cache-Control: no-cache, must-revalidate");
	  include "include/database_connect.inc.php";
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>
	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
	<title>CRM Data Mining</title>
	<link rel="stylesheet" type="text/css" href="css/cupertino/jquery-ui-1.8.11.custom.css"/>
	<link type="text/css" href="fyp.css" rel="stylesheet" />
	<script type="text/javascript" src="js/jquery/jquery-1.5.1.min.js" ></script>
	<script type="text/javascript" src="js/jquery/jquery-ui-1.8.18.custom.min.js"></script>
	<script type="text/javascript" src="js/order_control.js"></script>
	<script type="text/javascript" src="js/feedback_control.js"></script>
	<script type="text/javascript" src="js/common.js"></script>
	<script language="javascript" >
			$(document).ready(function() {
				now = new Date();
				jsnow = new Date();
				now.setTime('<?php echo time()*1000;?>');
				update_current_time();
				$('#tabs').tabs({
					show: function(event, ui){}
				}).find('.ui-tabs-nav').sortable({
					axis:'x'
				});
				$('#tabs').tabs({
					show: function(event, ui) {
						if($('#tabs>div:not(.ui-tabs-hide)').attr('id')=='tabs1') {
```

```
$('#OrderData>tbody>tr').remove();
$('#OrderDateStart').attr('value','Please
Select...');
$('#OrderDateEnd').attr('value','Please
Select...');
$('#emenu_food>option').remove();
$('#emenu_type>option').remove();
$('#shop_id').attr('value','');

$('#emenu_type').attr('disabled','disabled');

$('#emenu_food').attr('disabled','disabled');
$('#emenu_type').append('<option>--
CHOOSE ONE--</option>');
$('#emenu_food').append('<option>--
CHOOSE ONE--</option>');
$('#Chart>tbody>tr>td').remove();
$('#Chart>tbody>tr').append('<td><div
id="BarChart"></div></td>');
}
if($('#tabs>div:not(.ui-tabs-
hide)').attr('id')=='tabs2') {
$('#Chart>tbody>tr>td').remove();
$('#Chart>tbody>tr').append('<td><div
id="BarChart"></div></td>');
}
}
});
$('#sort').live('change',function() {
$('#OrderHead>tbody>p').remove();
$('#OrderData>tbody>tr').remove();
$('#Chart>tbody>tr>td').remove();
$('#Chart>tbody>tr').append('<td><div
id="BarChart"></div></td>');
```

```
                                        orderCreate();
                                    });
                            }); //End document.ready
        </script>
          <link                    rel="stylesheet"                    type="text/css"
href="http://alanpo.com/myfyp/crm/web/Renew/application/views/NewCss/grid.css"
media="screen" />
                <link                    rel="stylesheet"                    type="text/css"
href="http://alanpo.com/myfyp/crm/web/Renew/application/views/NewCss/styles.css"
media="screen" />

            </head>
    <body>
      <div id="header">
    <!-- Header. Status part -->
    <div id="header-status">
        <div class="container_12">
            <div class="grid_8">

                <font color="white">Enoteca Group <br/>Restaurant Data Analysis Center

                </font>
            </div>



     <div class="grid_4">
      <a                 href="http://alanpo.com/myfyp/crm/web/Renew/index.php/login_control"
id="logout">
                Logout
              </a>
              <a
href="http://alanpo.com/myfyp/crm/web/Renew/index.php/center_control/news/3"
               id="logout">
                Shop Page
```

```
            </a>
            <a href="http://alanpo.com/myfyp/crm/web/DataMining/" id="logout">
              Data Center
            </a>
          </div>
        </div>
        <div style="clear:both;"></div>
    </div> <!-- End #header-status -->
    <!--- menu bar --->


        <table width="1440" border="0" align="center" cellpadding="0" cellspacing="0" >
        <tr>
                <td width="1440" height="20" class="head"></td>
                </tr>
                <tr>
                <td width="1440" align="right" class="body"><p>Catering CRM Data Analysis System</span><br />
    <!--        <span >System current time: </span><span id="current_system_time"> </span><br />
                <br /></p>
                <br />
    -->                 <table width="95%" border="0" align="center">
                        <tr>
                                <td>
                    <div id="tabs"> <!--Do not remove this div tag. It is used for jQuery tab meuns. -->
                        <ul>
                            <li><a href="#tabs1">Ordered Food </a></li>
                            <li><a href="#tabs2">Customer Feedback</a></li>
```

```
                    </ul>
                <br />
                    <div id="tabs1"><br />
                        <p id="abc" align="left">
                        System Usage:<br /><br />
                        1. Select the sort parameter (day / month) below. The data will be group by the parameter you selected.<br />
                        2. Fill in the information you wanted including date / year, branch name, food type and food name.<br />

                                                                                <br />
                                                                                <hr /><br />

<!--                    <p align="left">Sort the data by : <select id="sort">
                                <option value="1">Day</option>
                                <option value="2">Month</option>
                            </select>
                        </p>
                        </p>
  -->                    <table id="OrderHead" border="0" width="50%">
                        <thead>
                        </thead>
                        <tbody>
                        </tbody>
                        </table>
                        <table id="OrderHeadSort" border="0" width="62%">
                        <thead>
                        </thead>
                        <tbody>
                        </tbody>
                        </table>

                        <br /><br />
```

```
                    <table id="OrderData" border="0" class="ui-widget ui-widget-
header" width="100%">
                        <thead>
                            <th>Date / Month</th>
                            <th>Total<br />Order Quantities</th>
                            <th>Total Sales($HKD)</th>
                            <th>Customer<br />Rating : </th>
                            <th>  5<br />stars  </th>
                            <th>  4<br />stars  </th>
                            <th>  3<br />stars  </th>
                            <th>  2<br />stars  </th>
                            <th>  1<br />stars  </th>
                            <th>View<br />Bar Chart</th>
                        </thead>
                        <tbody class="ui-widget-content">
                        </tbody>
                    </table>
                </div>
                 <div id="tabs2">
                 <table id="FeedbackHead" width="100%" >
                        <thead>
                        </thead>
                        <tbody>
                        </tbody>
                    </table>
                    <table id="FeedbackData" class="ui-widget ui-widget-content"
width="100%">
                        <thead>
                        </thead>
                        <tbody>
                        </tbody>
                    </table>

                </div>
```

```
                                        </div> <!--Do not remove this div tag. It is used for
jQuery tab meuns. -->
                                    </td>
                            </tr>
                    </table>
            <BR /><BR />
            <table id="Chart" align="center">
             <tbody>
                    <tr><td><div id="BarChart"></div></td></tr>
              </tbody>
            </table>
            </tr></table>
        </body>
</html>
```

```
//
//  viewBookingViewController.m
//  FYP
//
//  Created by Kwok Chun Hin on 9/10/11.
//  Copyright 2011 __MyCompanyName__. All rights reserved.
//

#import "viewBookingViewController.h"

@implementation viewBookingViewController
@synthesize booking_id;

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle
*)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
        self.view.backgroundColor = [UIColor clearColor];

        UIImageView *table = [[[UIImageView alloc]
initWithFrame:CGRectMake(15, 40, 290, 380)] autorelease];
        table.image = [UIImage imageNamed:@"newsBg.png"];
        table.contentMode = UIViewContentModeScaleToFill;

        [self.view addSubview:table];
        [self.view sendSubviewToBack:table];
    }
    return self;
}

- (void)didReceiveMemoryWarning
{
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}

- (void) loadData: (NSString *) _booking_id
{
    booking_id = _booking_id;
    NSArray * item = [Common
                      jsonReaderWithURL:[NSString
stringWithFormat:@"%@/json/getBooking.php", [config getServerURL]]
                      WithParam:[NSString
stringWithFormat:@"booking_id=%@",
                                 _booking_id]];

    if ([[item valueForKey:@"error"] isEqual:@"1"]) {
        [Common msgbox:@"Please try again later.\nIf the problem
persists, try logout and login again." WithTitle:@"Authentication
Error!"];
```

```
        //[self.navigationController popViewControllerAnimated:YES];
    }  else if ([item count] > 0){
        // date
        NSString *date = [[item objectAtIndex:0] valueForKey:@"date"];

        UIImageView *img_date = [[[UIImageView alloc]
initWithFrame:CGRectMake(30, 50, 32, 32)] autorelease];
        img_date.image = [UIImage imageNamed:@"date.png"];
        img_date.contentMode = UIViewContentModeScaleToFill;
        [self.view addSubview:img_date];

        [self.view addSubview:[Common newLabelWithText:date posX:75
posY:55 width:300 height:32 WithFontSize:17 WithColor:[UIColor
blackColor]]];

        // time
        NSString *time = [[item objectAtIndex:0] valueForKey:@"time"];

        UIImageView *img_time = [[[UIImageView alloc]
initWithFrame:CGRectMake(30, 85, 32, 32)] autorelease];
        img_time.image = [UIImage imageNamed:@"time.png"];
        img_time.contentMode = UIViewContentModeScaleToFill;
        [self.view addSubview:img_time];

        [self.view addSubview:[Common newLabelWithText:time posX:75
posY:90 width:300 height:32 WithFontSize:17 WithColor:[UIColor
blackColor]]];

        // seat
        NSString *seats = [[item objectAtIndex:0] valueForKey:@"seats"];
        UIImageView *img_seat = [[[UIImageView alloc]
initWithFrame:CGRectMake(30, 120, 32, 32)] autorelease];
        img_seat.image = [UIImage imageNamed:@"seat.png"];
        img_seat.contentMode = UIViewContentModeScaleToFill;
        [self.view addSubview:img_seat];

        [self.view addSubview:[Common newLabelWithText:seats posX:75
posY:125 width:300 height:32 WithFontSize:17 WithColor:[UIColor
blackColor]]];

        // status
        int statusInt = [[[item objectAtIndex:0] valueForKey:@"status"]
intValue];
        NSString *status;
        switch (statusInt) {
            case 0:
                status=@"Pending";
                break;
            case 1:
                status=@"Approved";
                break;
            case 2:
                status=@"Rejected";
                break;
            case 4:
```

```
                status=@"Waiting your confirm";
                break;
            case 5:
                status=@"Confirmed";
                break;
            default:
                break;
        }

        if (statusInt == 4) {
            // btn Login
            UIButton *btnConfirm = [UIButton
buttonWithType:UIButtonTypeCustom];
            [btnConfirm addTarget:self
                         action:@selector(confirm:)
               forControlEvents:UIControlEventTouchUpInside];
            [btnConfirm setTitle:@"Confirm"
forState:UIControlStateNormal];
            [btnConfirm.titleLabel setFont:[UIFont
fontWithName:@"Helvetica-Bold" size:19]];
            [btnConfirm setBackgroundImage:[UIImage
imageNamed:@"buttonBg.png"] forState:UIControlStateNormal];
            btnConfirm.frame = CGRectMake(30, 280, 260, 39);

            [self.view addSubview:btnConfirm];
        }
        UIImageView *img_status = [[[UIImageView alloc]
initWithFrame:CGRectMake(30, 155, 32, 32)] autorelease];
        img_status.image = [UIImage imageNamed:@"status.png"];
        img_status.contentMode = UIViewContentModeScaleToFill;
        [self.view addSubview:img_status];

        [self.view addSubview:[Common newLabelWithText:status posX:75
posY:160 width:300 height:32 WithFontSize:17 WithColor:[UIColor
blackColor]]];

        //NSString *content = [NSString stringWithFormat:@"Date:
%@\nTime: %@\nSeats: %@\nStatus: %@", date, time, seats, status];
        NSString *content = @"";
        NSArray *propertyArray = [[item objectAtIndex:0]
valueForKey:@"property"];
        if ([propertyArray count] > 0) {
            content = [content stringByAppendingFormat:@"\nProperty:"];
            for (NSString *property in propertyArray) {
                content = [content stringByAppendingFormat:@"\n - %@",
property];
            }

            UITextView *tv;

            tv = [Common newTextViewWithText:content posX:20 posY:180
width:300 height:200 WithFontSize:17 WithColor:[UIColor blackColor]];

            [self.view addSubview:tv];
        }
```

```
    }
}

-(void) confirm: (id)sender {
    NSMutableDictionary *temp = [[NSMutableDictionary alloc] init];
    [temp setObject:booking_id forKey:@"booking_id"];

    [temp setObject:[config getCompanyID] forKey:@"company_id"];
    if ([Common getUserDefaultStringForKey:@"member_id"] &&[Common
getUserDefaultStringForKey:@"member_password"]) {
        [temp setObject:[Common getUserDefaultStringForKey:@"member_id"]
forKey:@"member_id"];
        [temp setObject:[Common
getUserDefaultStringForKey:@"member_password"]
forKey:@"member_password"];
    }

    [temp setObject:[Common getDeviceID] forKey:@"device_id"];

    NSArray *result = [Common jsonReaderWriterWithURL:[NSString
stringWithFormat:@"%@/json/booking.php", [config getServerURL]]
WithDictionary:temp];

    if ([[result valueForKey:@"error"] isEqual:@"0"]) {
        UIAlertView* dialog = [[UIAlertView alloc] init];
        [dialog setDelegate:self];
        [dialog setTitle:@"Success"];
        [dialog setMessage:@"Booking have been confirmed"];
        [dialog addButtonWithTitle:@"Close"];
        [dialog show];
        [dialog release];

        [self.navigationController popViewControllerAnimated:YES];
    } else if ([[result valueForKey:@"error"] isEqual:@"1"]) {
        [Common msgbox:@"Please try again later.\nIf the problem
persists, try logout and login again." WithTitle:@"Authentication
Error!"];
    } else {
        [Common errorWithNetworkConnection];
    }

    [temp release];
}

- (IBAction) cancelBooking:(id)sender
{
    /*
    NSString *tokenID = [Common getDeviceID];
    if (!tokenID)
        tokenID = @"unknown";
     */
```

```
    NSMutableDictionary *temp = [[NSMutableDictionary alloc] init];
    [temp setObject:booking_id forKey:@"booking_id"];
    [temp setObject:@"cancel" forKey:@"action"];

    [temp setObject:[config getCompanyID] forKey:@"company_id"];
    if ([Common getUserDefaultStringForKey:@"member_id"] &&[Common
getUserDefaultStringForKey:@"member_password"]) {
        [temp setObject:[Common getUserDefaultStringForKey:@"member_id"]
forKey:@"member_id"];
        [temp setObject:[Common
getUserDefaultStringForKey:@"member_password"]
forKey:@"member_password"];
    }

    [temp setObject:[Common getDeviceID] forKey:@"device_id"];

    NSArray *result = [Common jsonReaderWriterWithURL:[NSString
stringWithFormat:@"%@/json/booking.php", [config getServerURL]]
WithDictionary:temp];

    if ([[result valueForKey:@"error"] isEqual:@"0"]) {
        UIAlertView* dialog = [[UIAlertView alloc] init];
        [dialog setDelegate:self];
        [dialog setTitle:@"Success"];
        [dialog setMessage:@"Booking have been cancelled"];
        [dialog addButtonWithTitle:@"Close"];
        [dialog show];
        [dialog release];

        [self.navigationController popViewControllerAnimated:YES];
    } else if ([[result valueForKey:@"error"] isEqual:@"1"]) {
        [Common msgbox:@"Please try again later.\nIf the problem
persists, try logout and login again." WithTitle:@"Authentication
Error!"];
    } else {
        [Common errorWithNetworkConnection];
    }
    /*
    if ([Common jsonWriterWithURL:[NSString
stringWithFormat:@"%@/json/bookingCancel.php", [config getServerURL]]
WithDictionary:temp]) {
        UIAlertView* dialog = [[UIAlertView alloc] init];
        [dialog setDelegate:self];
        [dialog setTitle:@"Success"];
        [dialog setMessage:@"Booking have been cancelled"];
        [dialog addButtonWithTitle:@"Close"];
        [dialog show];
        [dialog release];

        [self.navigationController popViewControllerAnimated:YES];
    } else {
        [Common errorWithNetworkConnection];
    }
    */
    [temp release];
```

```
}

#pragma mark - View lifecycle

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view from its nib.
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)int
erfaceOrientation
{
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}

@end
```

```
public class CouponDetailActivity extends Activity {

    /**
     * Called when the activity is first created.
     */
    Drawable a;
    TextView menuTitle;
    TextView text;
    ImageView image;
    Button getCouponButton;
    Button useCouponButton;

    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        if (ShopMenuActivity.checkInternetConnection()) {
            android.fyp.push.ActivityLevel.forthLevel();
            final Bundle bundle = getIntent().getExtras();
            setContentView(R.layout.coupondetail);
            RelativeLayout rl = (RelativeLayout)
findViewById(R.id.coupondetail_relativelayout1);
            rl.setBackgroundDrawable(ShopMenuActivity.getBg());
            RelativeLayout rl2 = (RelativeLayout)
findViewById(R.id.coupondetail_relativelayout2);
            rl2.setBackgroundDrawable(ShopMenuActivity.getBar());
            menuTitle = (TextView)
findViewById(R.id.coupondetail_title);
            menuTitle.setText(bundle.getString("coupon_title"));
            text = (TextView) findViewById(R.id.CouponText);
            image = (ImageView) findViewById(R.id.CouponImage);
            getCouponButton = (Button)
findViewById(R.id.coupondetail_getButton);
            a =
LoadImageFromWebOperations(bundle.get("img_url").toString());
            image.setImageDrawable(a);
            if (bundle.containsKey("button")) {
                getCouponButton.setOnClickListener(new
View.OnClickListener() {

                    public void onClick(View arg0) {

getCoupon(Integer.parseInt(bundle.getString("coupon_group_id")));
                    }
                });
            } else {
                getCouponButton.setVisibility(View.INVISIBLE);
                useCouponButton = (Button)
findViewById(R.id.coupondetail_useButton);
                useCouponButton.setVisibility(View.VISIBLE);
                useCouponButton.setOnClickListener(new
View.OnClickListener() {

                    public void onClick(View arg0) {
                        useCoupon(bundle.getString("coupon_group_id"));
```

```
                    }
                });
            }
            text.setText(bundle.getString("text"));
            //text.setText("Remain:" + bundle.getString("qty") + "\n");
            //text.setText(text.getText() + "Cost:" +
bundle.getString("cost") + "\n");
            //text.setText(text.getText() + "\n");
            //text.setText(text.getText() +
bundle.getString("coupon_description") + "\n");
            //text.setText(text.getText() + "\n\n\n\n\n\n\n\n\n\n\nTerms &
Condition:\n"+bundle.getString("terms_condition") + "\n");
        } else {
            Bundle bundle = new Bundle();
            bundle.putString("message", "Please connect network!");
            Intent dialog = new Intent(this,
android.fyp.push.Dialog.class);
            dialog.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

            dialog.putExtras(bundle);
            startActivity(dialog);
        }
    }

    private Drawable LoadImageFromWebOperations(String url) {
        InputStream is = null;
        Drawable d = null;
        Bitmap bm = null;
        String[] temp = url.split("/");
        try {
            if (ShopMenuActivity.checkInternetConnection()) {
                is = (InputStream) new URL(url).getContent();
                bm = BitmapFactory.decodeStream(is);
                File file = new File("/data/data/android.fyp/",
temp[temp.length - 1]);
                if (!file.exists()) {
                    OutputStream outStream = null;
                    outStream = new FileOutputStream(file);
                    bm.compress(Bitmap.CompressFormat.PNG, 100,
outStream);
                    outStream.flush();
                    outStream.close();
                }
                bm.recycle();
                System.gc();
                d = Drawable.createFromPath("/data/data/android.fyp/" +
temp[temp.length - 1]);
            } else {
                Bundle bundle = new Bundle();
                bundle.putString("message", "Please connect network!");
                Intent dialog = new Intent(this,
android.fyp.push.Dialog.class);
                dialog.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

                dialog.putExtras(bundle);
```

```
                startActivity(dialog);
            }
        } catch (Exception ex) {
            ex.printStackTrace();
            d =
getResources().getDrawable(R.drawable.photo_not_available);
            return d;
        }
        return d;
    }

    public void toMenu(View v) {
        finish();
    }

    @Override
    protected void onPause() {
        super.onPause();
        Push.setC4(this);
        PushService.setPaused(true);
        /*
         * if (!first) { paused = true; }
         */
    }

    @Override
    protected void onResume() {
        super.onResume();
        Push.setC4(this);
        PushService.setPaused(false);
        //paused = false;
    }

    @Override
    protected void onStart() {
        super.onStart();
        Push.setC4(this);
        PushService.setPaused(false);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        System.out.println("Destory");
        if (a != null) {
            a.setCallback(null);
        }
        a = null;
    }

    private void getCoupon(int couponGroupID) {
        HttpClient client = new DefaultHttpClient();
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        HttpPost request = new HttpPost();
        String content;
```

```
        String message = "";
        JSONObject object = new JSONObject();
        try {
            object.put("member_id", ShopMenuActivity.getMemberID());
            object.put("company_id", getString(R.string.CompanyID));
            object.put("action", "getCoupon");
            object.put("shop_id",
String.valueOf(ShopMenuActivity.getCurrentShop().getShop_id()));
            object.put("member_password",
ShopMenuActivity.getMemberPassword());
            object.put("coupon_group_id",
String.valueOf(couponGroupID));
            params.add(new BasicNameValuePair("data",
object.toString()));
            request.setEntity(new UrlEncodedFormEntity(params,
HTTP.UTF_8));
            request.setURI(new URI(ShopMenuActivity.host +
"/coupon.php"));
            content = client.execute(request, new
BasicResponseHandler());
            object=new JSONObject(content);
            message = object.getString("error_msg");
        } catch (Exception ex) {
            message = "Fail";
            ex.printStackTrace();
        }

GetCouponActivity.refresh(Secure.getString(this.getContentResolver(),
Secure.ANDROID_ID), this);
        ViewCouponActivity.refresh();
        Bundle bundle = new Bundle();
        bundle.putString("message", message);
        Intent dialog = new Intent(this, android.fyp.push.Dialog.class);
        dialog.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        finish();

        dialog.putExtras(bundle);

        this.startActivity(dialog);
    }

    private void useCoupon(String couponGroupID) {
        HttpClient client = new DefaultHttpClient();
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        HttpPost request = new HttpPost();
        String content;
        String message = "";
        String title="";
        JSONObject object = new JSONObject();
        try {
            object.put("member_id", ShopMenuActivity.getMemberID());
            object.put("company_id", getString(R.string.CompanyID));
            object.put("action", "useCoupon");
            object.put("shop_id",
```

```
String.valueOf(ShopMenuActivity.getCurrentShop().getShop_id()));
            object.put("member_password",
ShopMenuActivity.getMemberPassword());
            object.put("coupon_group_id", couponGroupID);
            params.add(new BasicNameValuePair("data",
object.toString()));
            request.setEntity(new UrlEncodedFormEntity(params,
HTTP.UTF_8));
            request.setURI(new URI(ShopMenuActivity.host +
"/coupon.php"));
            content = client.execute(request, new
BasicResponseHandler());
            object=new JSONObject(content);
            if(object.getString("error").equals("0")){
            message = "Ref No: "+object.getString("coupon_id");
            title="Success";
            }else{
            message = object.getString("error_msg");
            title="Fail";

            }
        } catch (Exception ex) {
            message = "Fail";
            title="Fail";
            ex.printStackTrace();
        }
        ViewCouponActivity.refresh();
        Bundle bundle = new Bundle();
        bundle.putString("title", title);
        bundle.putString("message", message);
        Intent dialog = new Intent(this, android.fyp.push.Dialog.class);
        dialog.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

        dialog.putExtras(bundle);

        this.startActivity(dialog);
    }
}
```

```
public class CouponLogBaseAdapter extends BaseAdapter {

    private LayoutInflater layouInflater;
    private List<HashMap<String, Object>> list;
    private Context context;

    public CouponLogBaseAdapter(Context context, List<HashMap<String,
Object>> list) {
        this.context = context;
        this.list = list;
        layouInflater = LayoutInflater.from(context);
    }

    @Override
    public int getCount() {
        return list.size();
    }

    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
    public View getView(final int position, View convertView, ViewGroup
parent) {
        convertView = layouInflater.inflate(R.layout.couponitem, null);
        convertView.setBackgroundResource(R.drawable.listselector);
        ImageView image = (ImageView)
convertView.findViewById(R.id.couponitemImage);
        image.setVisibility(View.INVISIBLE);
        TextView title = (TextView)
convertView.findViewById(R.id.couponitemTitle);

title.setText(list.get(position).get("coupon_title").toString());
        TextView desc = (TextView)
convertView.findViewById(R.id.couponitemDesc);
        desc.setText("Used at
"+list.get(position).get("used").toString()+"\n\n"+"Ref
id:"+list.get(position).get("coupon_id").toString());
        return convertView;
    }
}
```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package android.fyp.coupon;

import android.content.Context;
import android.fyp.R;
import android.graphics.drawable.Drawable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.TextView;
import java.util.HashMap;
import java.util.List;

/**
 *
 * @author lumiere
 */
public class GetCouponBaseAdapter extends BaseAdapter {

    private LayoutInflater layouInflater;
    private List<HashMap<String, Object>> list;
    private String current;
    private Context context;

    public GetCouponBaseAdapter(Context context, List<HashMap<String,
Object>> list) {
        this.context = context;
        this.list = list;
        layouInflater = LayoutInflater.from(context);
    }

    @Override
    public int getCount() {
        return list.size();
    }

    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    @Override
    public View getView(final int position, View convertView, ViewGroup
parent) {
```

```
        convertView = layouInflater.inflate(R.layout.couponitem, null);
        convertView.setBackgroundResource(R.drawable.listselector);
        ImageView image = (ImageView)
convertView.findViewById(R.id.couponitemImage);
        image.setImageDrawable((Drawable)
list.get(position).get("img"));
        TextView title = (TextView)
convertView.findViewById(R.id.couponitemTitle);

title.setText(list.get(position).get("coupon_title").toString());
        TextView desc = (TextView)
convertView.findViewById(R.id.couponitemDesc);
        desc.setText("Cost "+list.get(position).get("cost").toString()+"
points");
        return convertView;
    }
}
```

END OF REPORT