MINI PROJECT

ON

FOOD ORDERING APPLICATION

Under the guidance of

➔ **Mr. Sharad Gupta**

Department of Computer Engineering & Applications Institute of Engineering & Technology

GLA UNIVERSITY
ESTD. 2010
ऋते ज्ञानान्न मुक्तिः

GLA
UNIVERSITY
MATHURA
Established vide U.P. Act 21 of 2010.

Submitted To:

Mr. Sharad Gupta

(Assistant Professor)

Submitted By:

Prajesh Srivastava

Suyash Batham

Akash Gupta

Mridul Srivastava

Mohd. Aqib

Department of Computer Engineering and Applications
GLA University, Mathura
17 km. Stone, NH#2, Mathura-Delhi Road, P.O. –
Chaumuhan Mathura – 281406

Declaration

I hereby declare that the work which is being presented in the Mini Project entitled "Food Ordering Application" is an authentic record of my own work carried out during the period from 1-07-2020 to 25-11-2020 under the supervision of the instructor "Mr. Sharad Gupta".

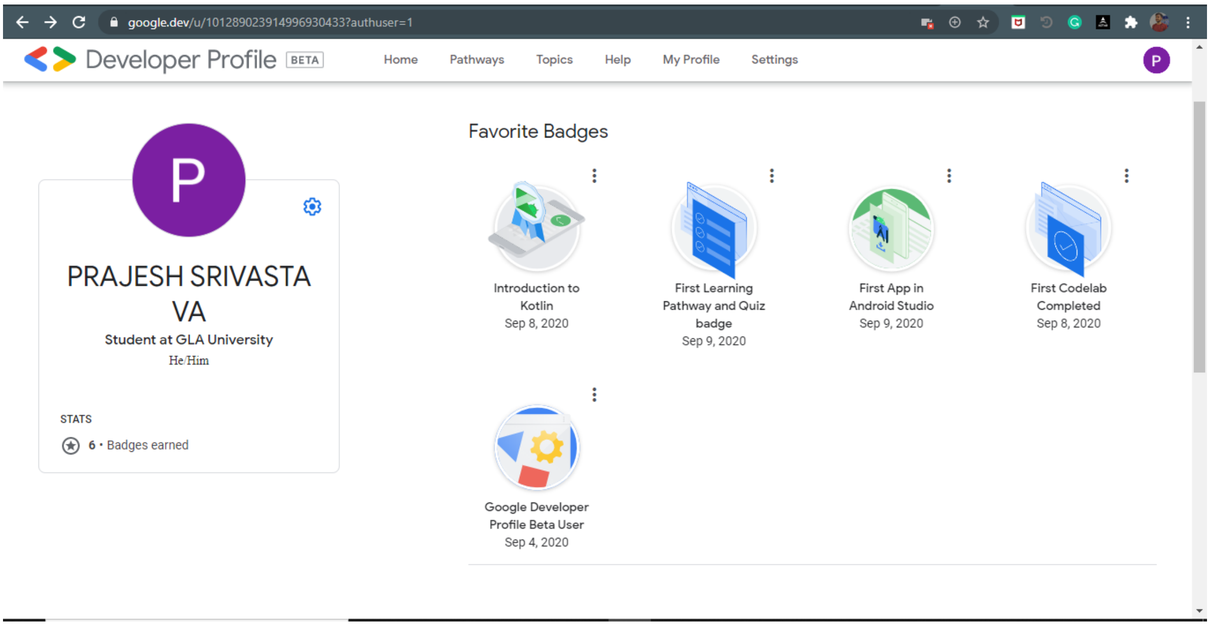
Date: 25-10-2020
Course: B.Tech (CSE)

Year: Third yr.

Semester: 5th

Certificates:

1-Prajesh Srivastava



PRAJESH SRIVASTAVA

Student at GLA University

He/Him

STATS

6 • Badges earned

Favorite Badges

Introduction to Kotlin
Sep 8, 2020

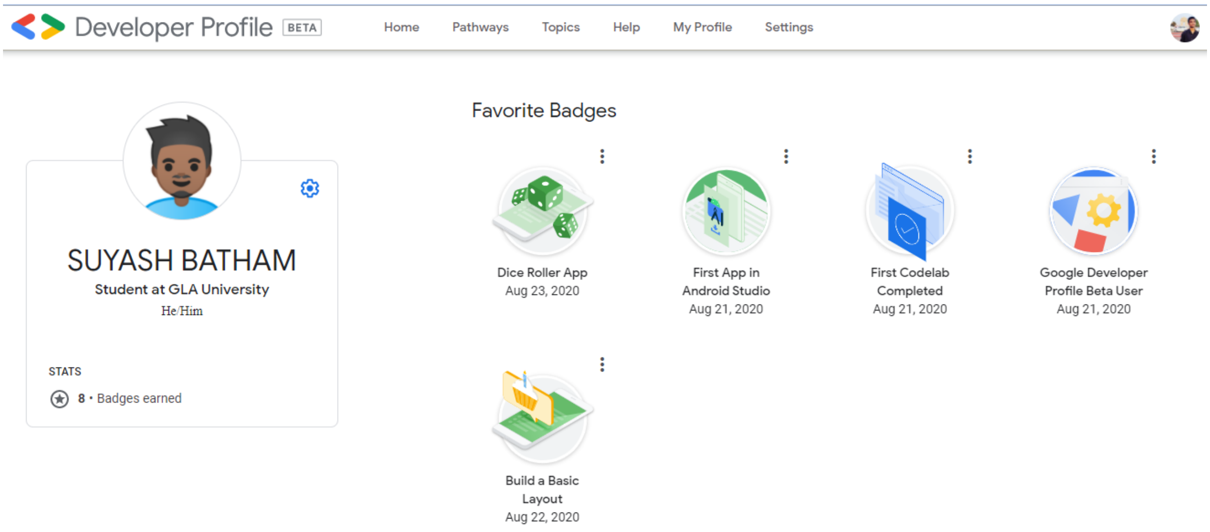
First Learning Pathway and Quiz badge
Sep 9, 2020

First App in Android Studio
Sep 9, 2020

First Codelab Completed
Sep 8, 2020

Google Developer Profile Beta User
Sep 4, 2020

2-Suyash Batham



SUYASH BATHAM

Student at GLA University

He/Him

STATS

8 • Badges earned

Favorite Badges

Dice Roller App
Aug 23, 2020

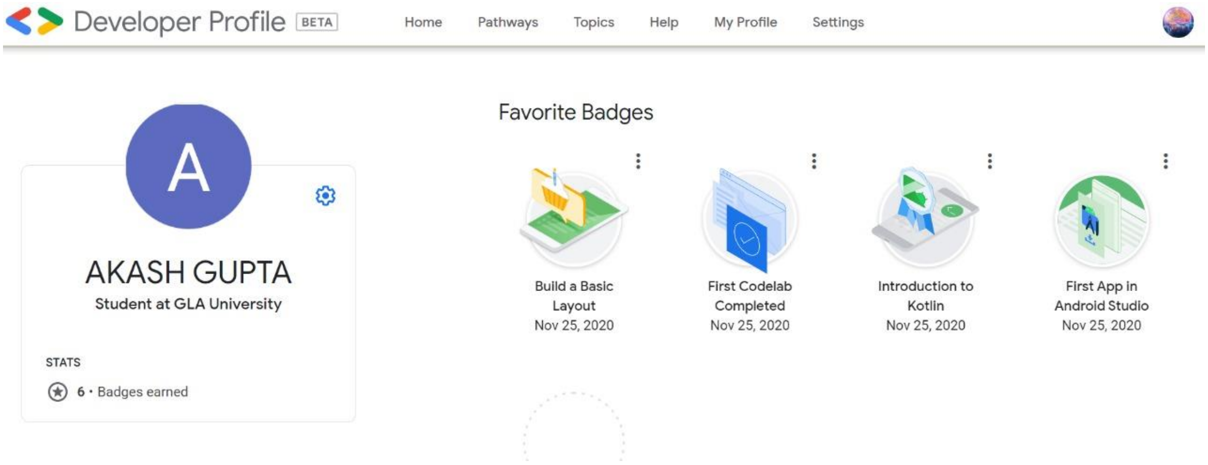
First App in Android Studio
Aug 21, 2020

First Codelab Completed
Aug 21, 2020

Google Developer Profile Beta User
Aug 21, 2020

Build a Basic Layout
Aug 22, 2020

3-Akash Gupta



A

AKASH GUPTA

Student at GLA University

STATS

6 • Badges earned

Favorite Badges

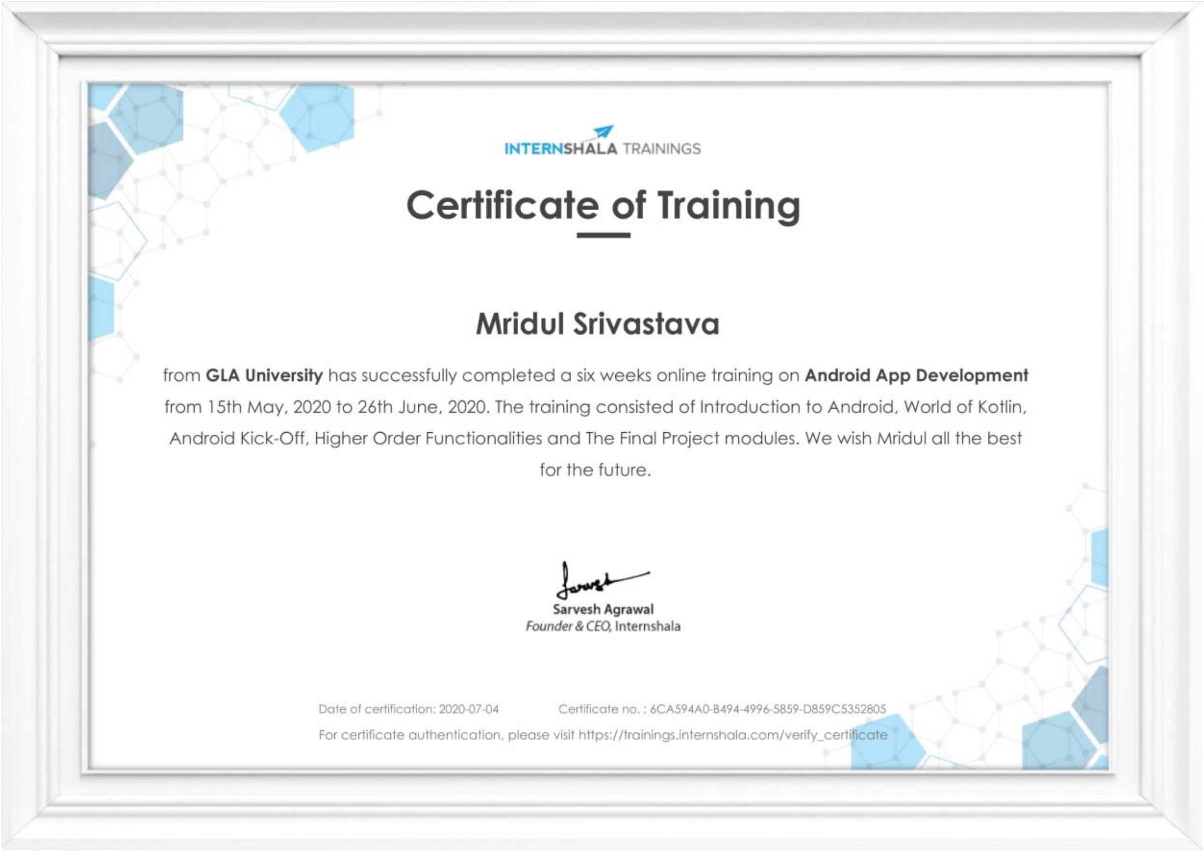
Build a Basic Layout
Nov 25, 2020

First Codelab Completed
Nov 25, 2020

Introduction to Kotlin
Nov 25, 2020

First App in Android Studio
Nov 25, 2020

4- Mridul Srivastava



INTERNSHALA TRAININGS

Certificate of Training

Mridul Srivastava

from **GLA University** has successfully completed a six weeks online training on **Android App Development** from 15th May, 2020 to 26th June, 2020. The training consisted of Introduction to Android, World of Kotlin, Android Kick-Off, Higher Order Functionalities and The Final Project modules. We wish Mridul all the best for the future.

Sarvesh Agrawal
Founder & CEO, Internshala

Date of certification: 2020-07-04   Certificate no. : 6CA594A0-B494-4996-5859-D859C5352805

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

ABSTRACT

The purpose of this was to build a food ordering client server application for GLA UNIVERSITY, which is a the online restaurant Hub.

For the customer, this application provides a view of current food information (category, name, image， price, description etc.) on the website and Android application. The customer can order food from these two platforms. For the administrator in restaurant, this application offers a series of operations to add, update, delete and query the information of food, food order and employees.

So far, all core functions were developed successfully and the progress of the project was most rewarding and generated an excellent experience in programming.

ACKNOWLEDGEMENTS

Firstly I would like to give a big appreciation to my mentor **Sharad Gupta**, for his patient guidance and support during conducting the whole this is.

Moreover, I would like to give my gratitude to other teachers, for the quality education. Finally I also need thank all the people whose names are not mentioned here for their help on study and life.

INTRODUCTION

With the rapid development of information technology, Android application have been increasing in recent years. Compared with the desktop application.

The advantage of the Android application:

- Mobile application is convenient to carry
- Global partnerships and large install base
- Powerful development framework
- Open marketplace for distributing apps

Based on the advantages of applications, I motivated myself to develop a project on Android application.

Due to the cheaper prices and delicious food, more and more people select to eat in this restaurant. Meanwhile, with the number of customers increasing.

Mobile Food Ordering Application is the key to solve this problem. Using this application, the customers need not go to the restaurant by themselves, but they can order the dishes through Android mobiles anywhere.

The Background Management platform in this application was designed for the administrator. The Administrator will be able to manage food dishes, dish orders and company employees here.

Android Framework

Android is known as a mobile operating system based on the Linux Kernel. It is mainly designed for touchscreen mobile devices such as smartphone and table computers. Meanwhile, the Android operating system is widely used in televisions, games consoles, digital cameras and other electronics because of its open and customizable features.

The source code for Android is available under free and open-source software licenses. It means that the device manufactures, wireless carriers and enthusiast developers can freely modify and distribute the software. Most Android devices ship with a combination of open source and proprietary.

Today, Android has become the most popular mobile OS, and is the leader in smartphone market in the world. More and more customers select mobile phones.

Linux Kernel

At the bottom of the layers is Linux Kernel. It provides a basic system functionality, such as memory management, device management etc. Also, it handles the things that Linux is good at such as networking.

✐ Libraries

On top of Linux Kernel is a set of libraries including open-source Web browser engine WebKit , SQLite database, libraries to play and record audio and video, SSL libraries and so on.

✐ Android Runtime

This section provides a key component called Dalvik Virtual Machine that is a kind of Java Virtual Machine designed for Android. The Android runtime also offers a set of core libraries: it enables developers to develop Android applications using the standard Java programming language.

✐ Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

✐ Applications

All the Android applications will be found at the top layer. The application written by the developer will be installed on this layer only.

APPLICATION DESCRIPTION

There are certain requirements the proposed application must fulfil to meet the objectives of the project.

The requirements to be achieved:

In Background Management Platform:

- Administrator can add and modify food categories.
- Administrator can add, modify and query food information.
- Administrator can add, modify and query employee information.
- Administrator can manage orders produced from the web application and Android application.

In the Website Public Page and Android Application:

- Customer can view food information, such as category, name, price, image, description and so on.
- Customer can order food.
- Customer can modify food item, food amount in Shopping Cart.
- Produce food order.

The requirements that should be achieved:

- The project supports internationalization, customer can select different language environments according to their real requirements.

The requirements nice to be achieved:

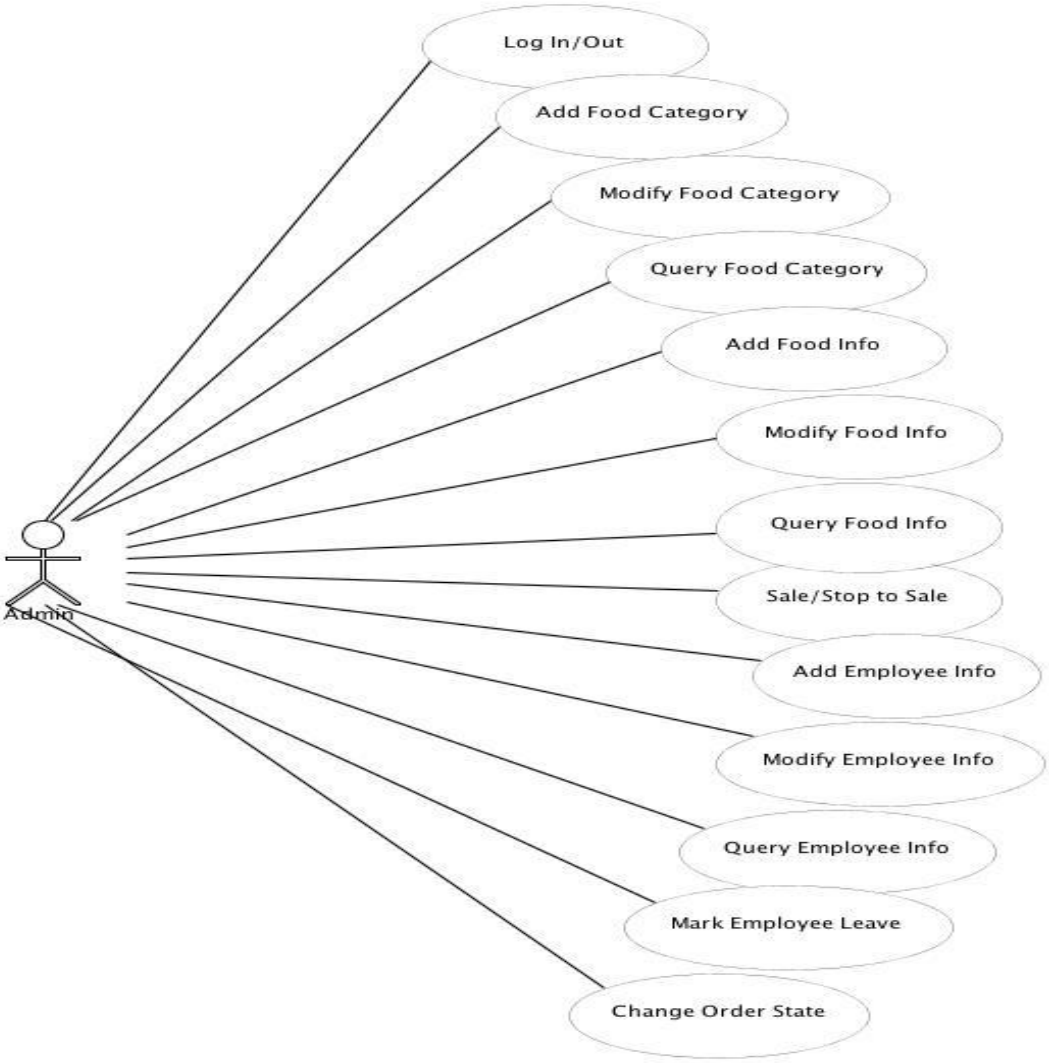
- Permission Management in Background Management Platform
- Query Dish Function in Website Public Page

Functional Description

Destine Food System is divided into two parts in terms of roles, the background for the administrator, the website foreground public pages and the Android application for user. Functions corresponding to each role will be introduced in detail in the following.

Background Management Platform

Figure in below is a use case diagram to illustrate the main functions in the background for the administrator. The detailed description of functions is as follows:


Log In/Out
Add Food Category
Modify Food Category
Query Food Category
Add Food Info
Modify Food Info
Query Food Info
Sale/Stop to Sale
Add Employee Info
Modify Employee Info
Query Employee Info
Mark Employee Leave
Change Order State
Admin

Log In/Out

Properties:

A Username

A Password

When the administrator connects the background management platform url , a log in interface will be displayed. The administrator needs to input correct username and password to log into the main page of the background management page. One admin username and password is pre-set when the application is initialized.

Add Food Category

Properties:

A Category Name

A Note

When the administrator clicks “Category Manage”, a list view of dish category will be displayed. On the bottom of the list view page, there are two buttons: “Add Category” and “Query”. While clicking the “Add Category” button, the page will be linked to a dish-adding interface, the administrator can add a new dish category here.

Query Food Category

Properties:

A Category Name

Also, clicking the “Query” Button, a dish category query interface will be displayed, here the administrator can query current existing dish category according to the category name.

Modify Food Category

Properties:

A Category Name

A Note

In addition, in the category list view page, the administrator can modify the existing dish category by clicking “Modify” button.

Add Food Info

Properties:

A Product Name

A Product Category

A Market Price

A Sell Price

A Product Code

A Product Style Image

A Product Description

When the administrator wants to add dishes, “Product Manage” just needs to be clicked; the right frame page will jump to a list view page of dishes. On the page, the detailed information of existing dishes will be shown, such as Product Code, Product Name, Category Belong, Sell Price, the state of sale, etc.

Modify Food Info

Properties:

A Product Name

A Product Category

Market Price

Sell Price

Product Code

Product Style Image

Product Description

The administrator can modify food information by clicking “Modify“ button.

Query Food Info

Properties:

Product Name

Product Category

Sell Price

Product Code

The administrator can query food information according to different conditions above.

Sale/Stop to sale

In addition, this project provides “On Sale” and “Stop Sale” functions. Using these functions, the administrator can manage the state of dish sale.

Add Employee Info

Properties:

- Login Account
- Login Password

Repeat Password

Employee Name

Employee Photo

Id Number

Birthday

Address

Telephone

Email (optional)

Degree (optional)

School (optional)

In the Employee Management module, the administrator can view the information of existing employees in the company by clicking “Employee Manage”. The functions of “Add“ and “Modify employee“ are also provided in the project. The employee accounts added can be used to log in this background management platform. In addition, according to the real request, when an employee leaves the company, the administrator can mark leave for this employee using “Mark Leave” function, it means that the employee account will be invalid to log in the background management platform.

Query Employee Info

Properties:

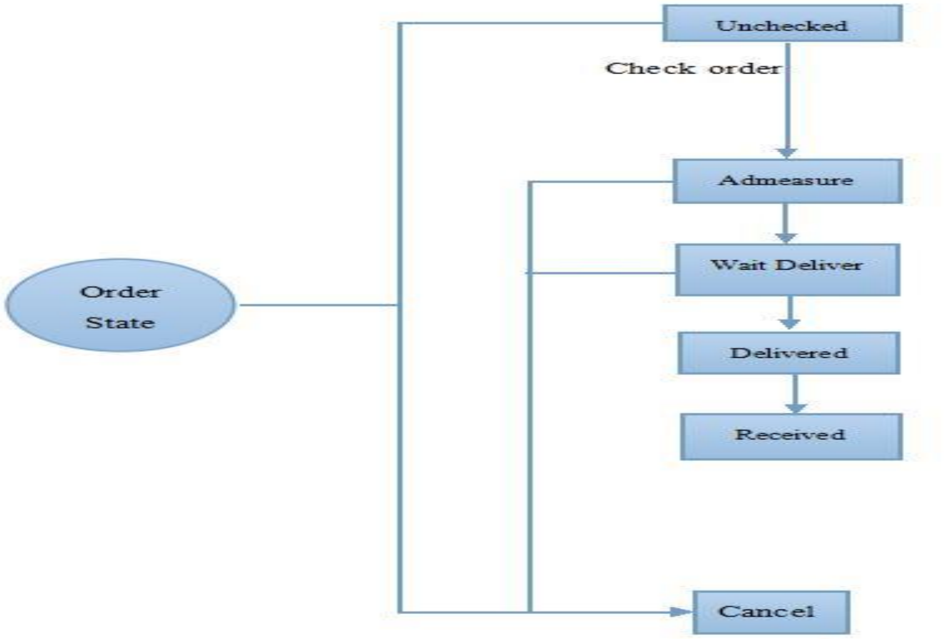
User Name

Real Name

By clicking the “Employee Query” link, the administrator can query company employees by the user name (log in account) or the employee’s real name.

Change Order States

Figure 6 is a use case diagram for the order flow to illustrate the flow of order state. When the customer purchases the dishes no matter what platform they are in (website or Android application). The order of dishes will be produced and showed in the background management platform.



Order state changing flow

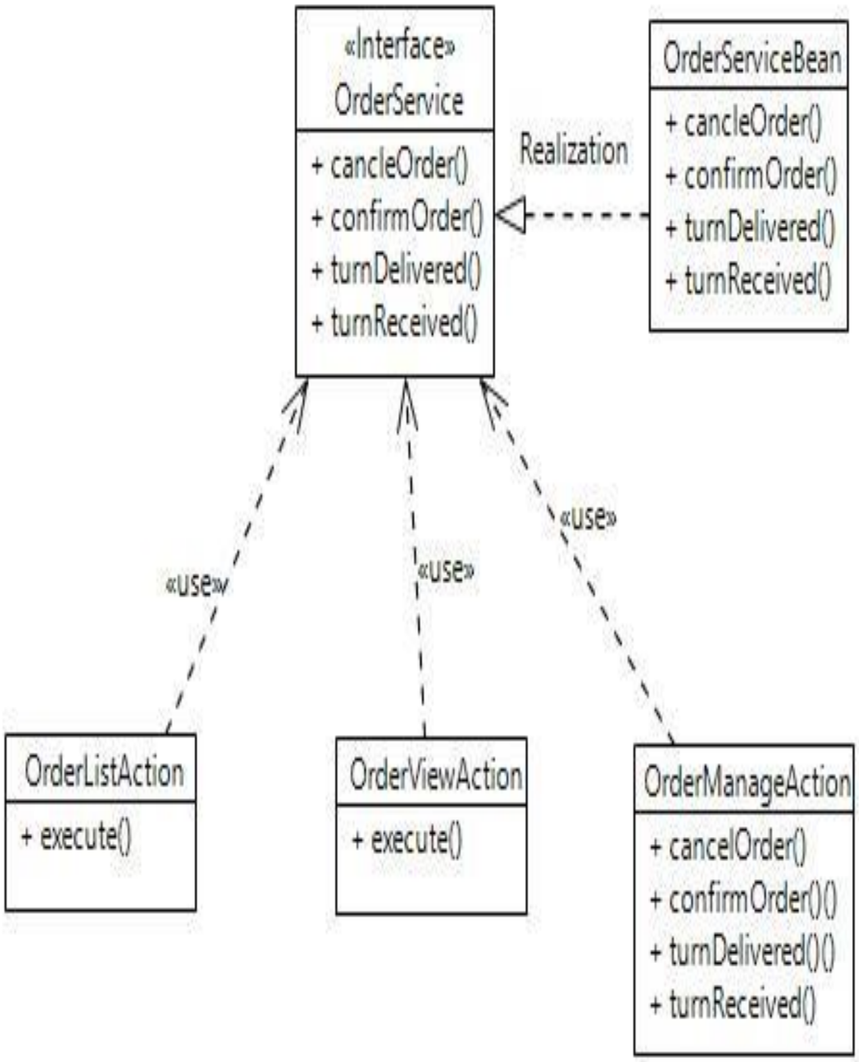
The dish order is divided into five states: Unexamined, Wait Deliver, Delivered, Received and Cancelled. The default state of produced order is Unexamined, the administrator can load the order to check the detailed information, for example, dish items, amount, price and customer information. After checking the Unexamined order, the administrator can decide to change the state of order, if the order is examined, the order state is changed to Wait Deliver, otherwise, the state is changed to Cancelled. As the simpler flow, the order state goes through Delivered, till Received.

Class Hierarchy

Background Management Platform

The Background Management Platform part is architected with Struts2 framework, Spring framework and JPA framework, and it makes use of classical concept known as MVC, short for model, view and controller. This inside structure will be analysed in detail in the following.

the administrator can change the state of dish order by using these methods.


«Interface»
OrderService
+ cancleOrder()
+ confirmOrder()
+ turnDelivered()
+ turnReceived()
Realization
OrderServiceBean
+ cancleOrder()
+ confirmOrder()
+ turnDelivered()
+ turnReceived()
«use»
«use»
«use»
OrderListAction
+ execute()
OrderViewAction
+ execute()
OrderManageAction
+ cancelOrder()
+ confirmOrder()()
+ turnDelivered()()
+ turnReceived()

Implement this method to jump to a JSP page.

- cancleOrder():

  Cancel the dish order.

- confirmOrder():

  Change order state from “Unexamined” to “Wait Deliver”

- turnDelivered();

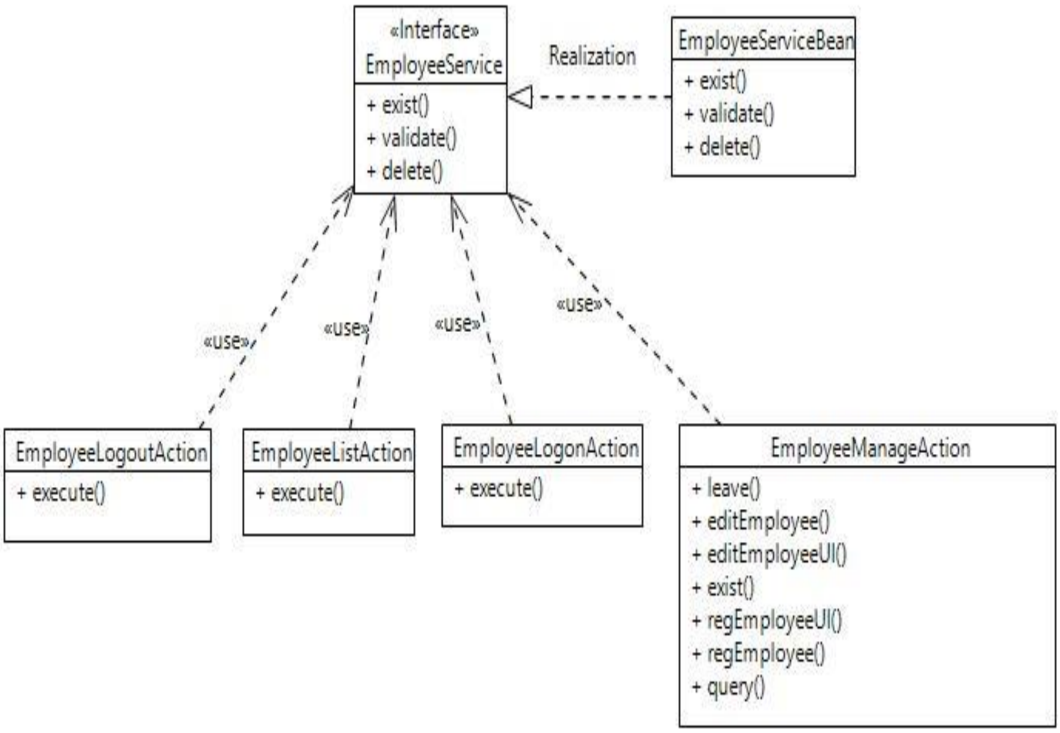
  Change order state from “Wait Deliver” to “Delivered”

- turnReceived():

  Change order state from “Delivered” to “Received”

Employee Module:

The administrator can make a series of operations shown in Figure 10 regarding employees. Also, employees can use their own function given in the system, such as log in, log out and so on.



Employee Module Controller Structure

✐ execute():

Implement this method to jump to a JSP page.

✐ editEmployee():

Modify the current existing employee information.

✐ editEmployeeUI():

Jump to modify employee information page.

✐ exist():

Measure whether the employee account exists or not when registering.

✏ leave():

Mark an employee in leave situation when an employee leaves company.

✎ query():

Query an employee according to the account name or employee’s real name.

✎ regEmployee():

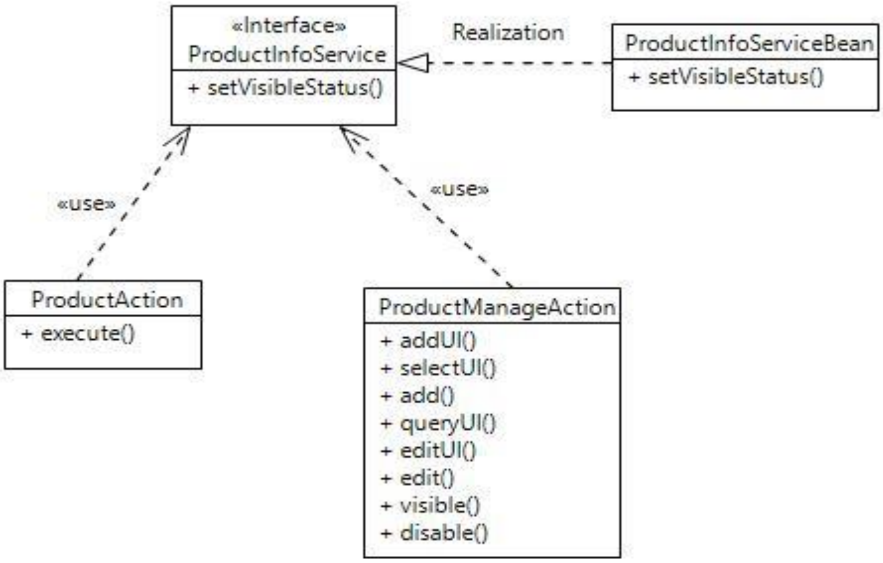
Employee registers.

✎ regEmployeeUI():

Jump to employee register page.

Product Module:

In the product module, the administrator can control the related operations for dishes.



Product Module Controller Structure

- execute():

  Implement this method to jump to a JSP page.

- disable():

  Stop to sale some dish.

- add():

  Add dishes.

- edit():

  Edit current existing dish.

- editUI():

  Jump to edit dish page.

- queryUI():

  Jump to query dish UI

- selectUI():

  Select dish category for dish when adding a dish.

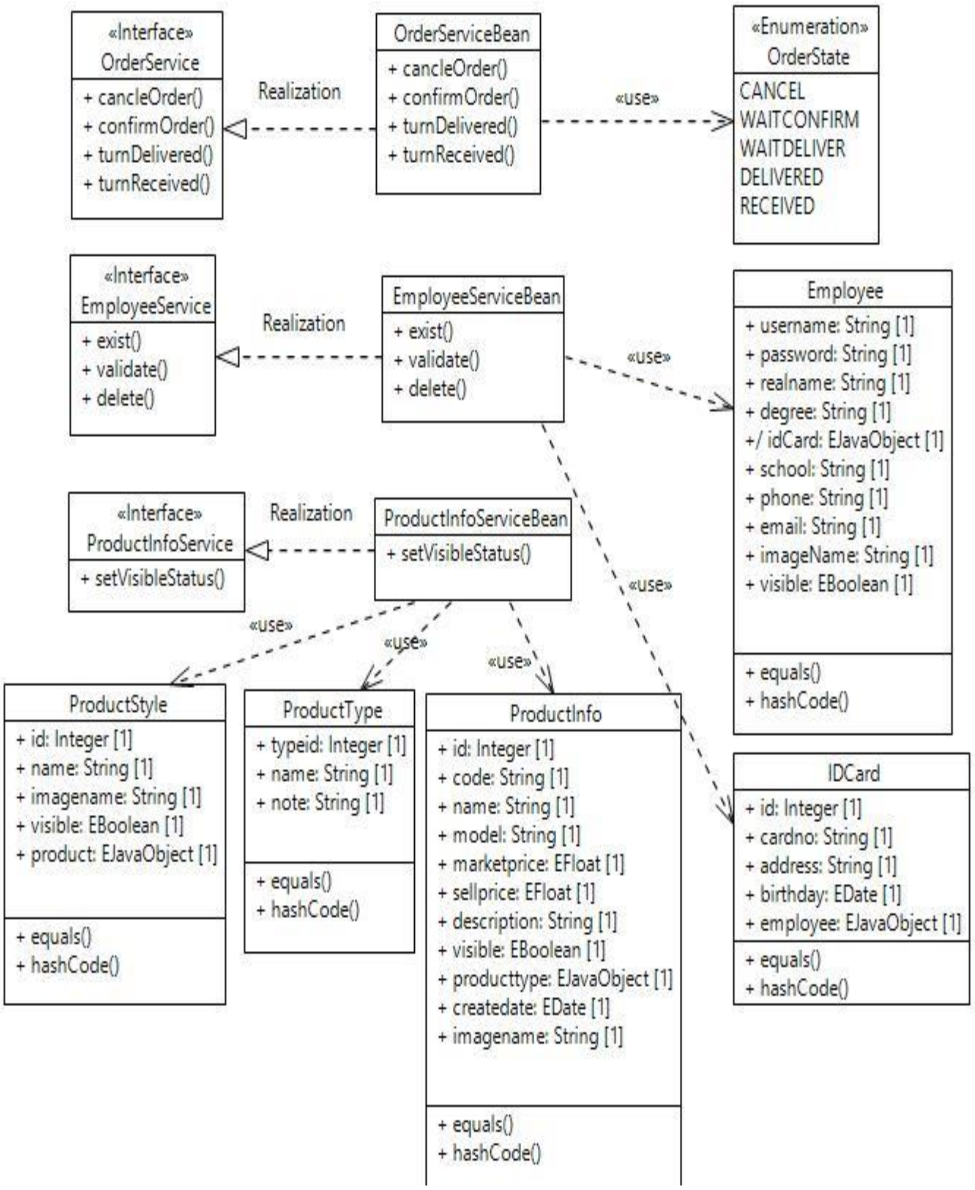
- visible():

  Start to sell a dish corresponding to stop to

The view layer mainly takes in charge of JSP display pages in the foreground.

Model Class Diagram

These POJO classes in the diagram in Figure 12 define their own entities in this project meanwhile building a mapping with relative tables in the database.



Module Structure

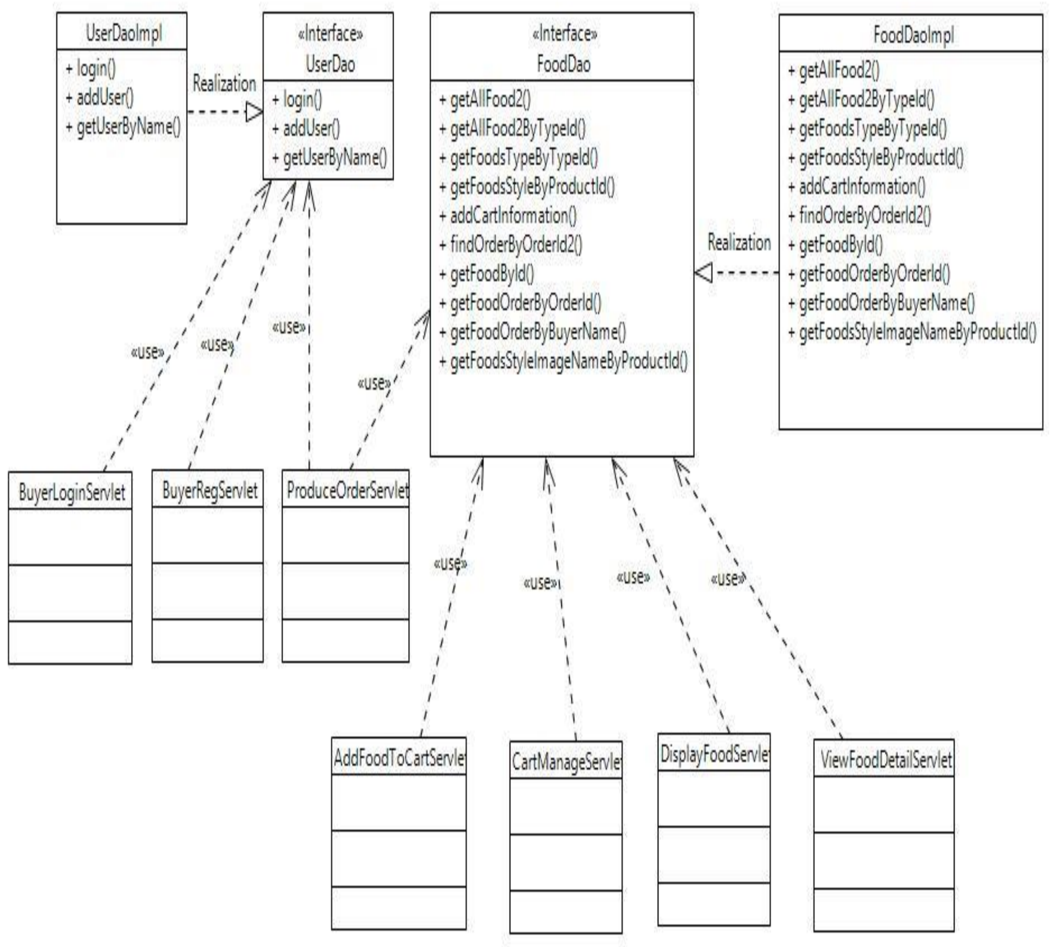
As a consequence, each row of POJO attribute is relative to corresponding attribute in tables (Object/Relatinal Mapping).

Website Foreground Public Page and Android Application

The Website Foreground Public Page part and the Android Application still make use of MVC structure. Compared with the Background Management platform, the technology used for the controller layer is a servlet.

Controller Class Diagram

Controller layer in Website Foreground Public page is servlet. The main methods called by servlet are described below.



Controller structure in Website Foreground Public Page

- getAllFood2():

Get all dishes information for displaying on the website public page.

- getAllFood2ByTypeId():

According to type id, get all dishes belonging to some category.

- getFoodsTypeByTypeId():

Get all dish categories in order to display dishes according to different categories.

- getFoodsStyleByProductId():

According to dish id, find dish image.

- addCartInformation():

Add dish amount and dish id to shopping cart.

- findOrderByOrderId():

Find order by order id, return type is list.

- findOrderByOrderId2():

Find order by order id, return type is OrderItem, an entity bean contains some properties of dish item.

- getFoodById():

Get Food Entity according to food id.

- getFoodOrderByOrderId():

  Get food order according to order id.

- getFoodOrderByBuyerName():

Find food order according to buyer’s name.

✏ getFoodsStyleImageNameByProductId():

Fetch some dish image name according to dish id.

✏ Login():

Customer logs in the website when purchasing dishes.

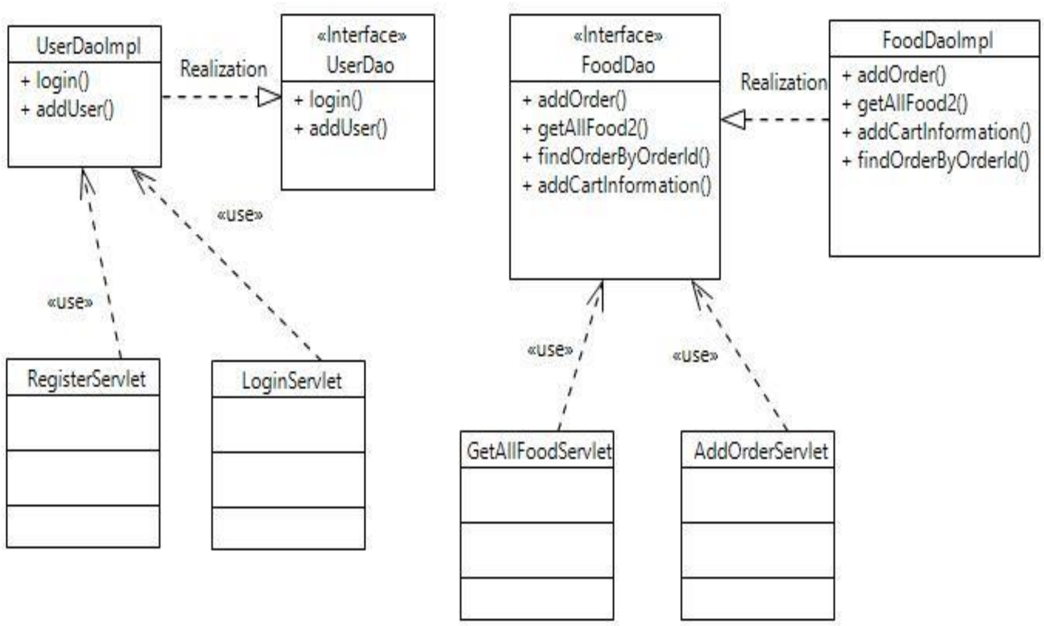
✏ addUser():

Customer registers in to the website.

✏ getUserByName():

Query the user by name in order to validate whether the user name exists or not.

Customers do operations in Android Application, application sends request with data to server, and then servlet in server will handle data and response data to android application. Therefore, in Android Application, the controller layer is still servlet. Also, the main methods in servlet will be given as follow.



Controller structure in Android Application

- login():

  Customer logs in the Android application platform to purchase dishes.

- addUser():

  Customer registers in the Android application before purchasing dishes.

- addOrder():

  Insert customers deliver information to the database.

- getAllFood2():

  Return all dishes information from website.

- findOrderByOrderId():

Get some order according to order id.

✐ addCartInformation():

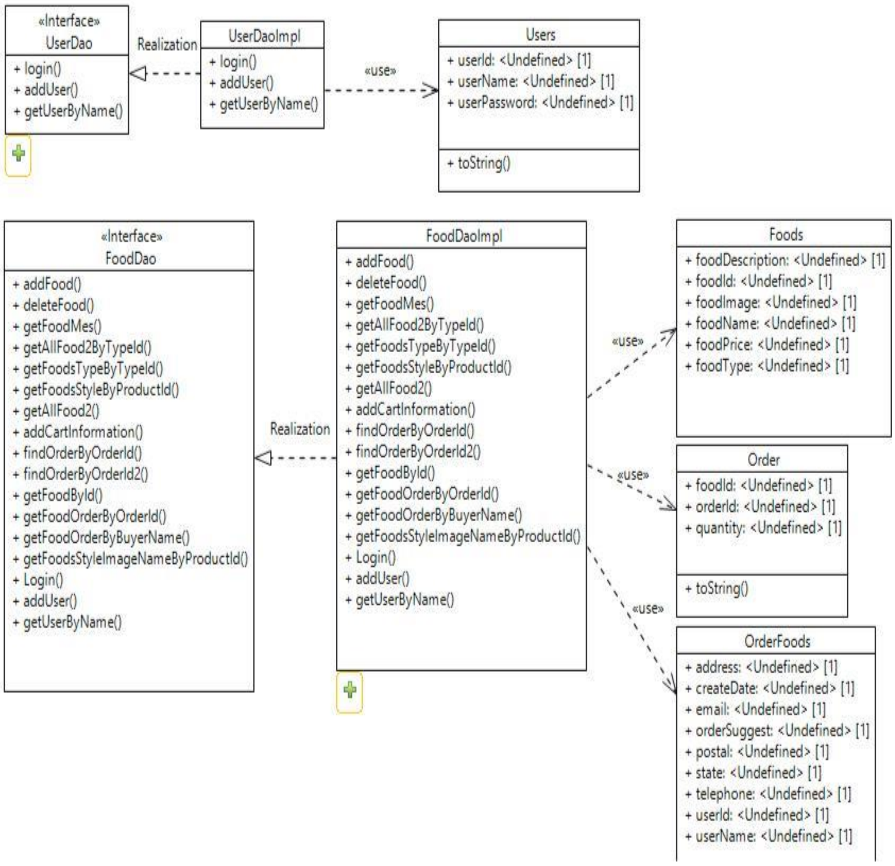
Insert customers’ order detailed dishes information to database.

View Class Diagram

The view layer on the Website Foreground Public Page still includes some JSP page to display on the foreground public pages. In the Android Application, the view layer mainly takes in charge of the layout xml files.

Model Class Diagram

Below is the module structure in Android Application. The entities in theses POJO classes also build a mapping with relative table in the MySql database.


«Interface»
UserDao
+ login()
+ addUser()
+ getUserByName()
Realization
UserDaoImpl
+ login()
+ addUser()
+ getUserByName()
«use»
Users
+ userId: <Undefined> [1]
+ userName: <Undefined> [1]
+ userPassword: <Undefined> [1]
+ toString()
«Interface»
FoodDao
+ addFood()
+ deleteFood()
+ getFoodMes()
+ getAllFood2ByTypeId()
+ getFoodsTypeByTypeId()
+ getFoodsStyleByProductId()
+ getAllFood2()
+ addCartInformation()
+ findOrderByOrderId()
+ findOrderByOrderId2()
+ getFoodById()
+ getFoodOrderByOrderId()
+ getFoodOrderByBuyerName()
+ getFoodsStyleImageNameByProductId()
+ Login()
+ addUser()
+ getUserByName()
Realization
FoodDaoImpl
+ addFood()
+ deleteFood()
+ getFoodMes()
+ getAllFood2ByTypeId()
+ getFoodsTypeByTypeId()
+ getFoodsStyleByProductId()
+ getAllFood2()
+ addCartInformation()
+ findOrderByOrderId()
+ findOrderByOrderId2()
+ getFoodById()
+ getFoodOrderByOrderId()
+ getFoodOrderByBuyerName()
+ getFoodsStyleImageNameByProductId()
+ Login()
+ addUser()
+ getUserByName()
«use»
Foods
+ foodDescription: <Undefined> [1]
+ foodId: <Undefined> [1]
+ foodImage: <Undefined> [1]
+ foodName: <Undefined> [1]
+ foodPrice: <Undefined> [1]
+ foodType: <Undefined> [1]
«use»
Order
+ foodId: <Undefined> [1]
+ orderId: <Undefined> [1]
+ quantity: <Undefined> [1]
+ toString()
«use»
OrderFoods
+ address: <Undefined> [1]
+ createDate: <Undefined> [1]
+ email: <Undefined> [1]
+ orderSuggest: <Undefined> [1]
+ postal: <Undefined> [1]
+ state: <Undefined> [1]
+ telephone: <Undefined> [1]
+ userId: <Undefined> [1]
+ userName: <Undefined> [1]

Also, every POJO class is an entity bean containing some attributes, building a mapping with relative tables in the database.

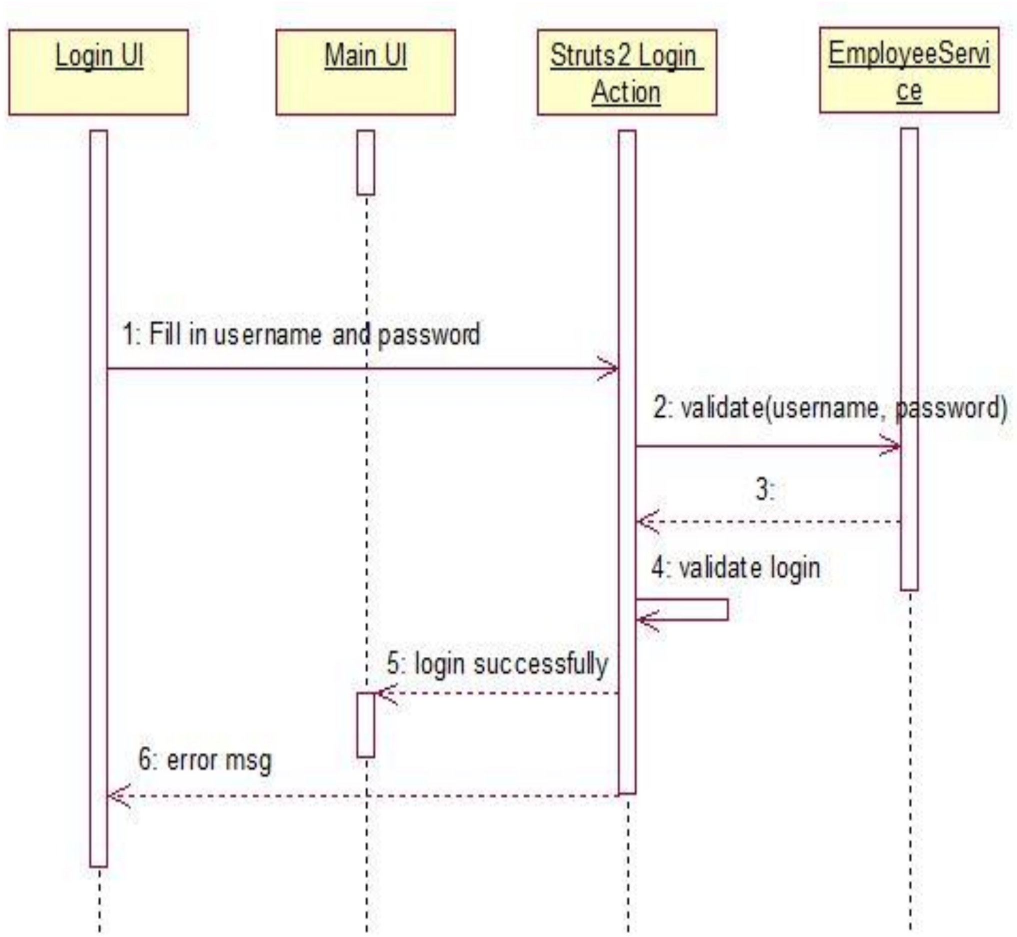
Sequence Diagram

The sequence diagram of main modules in Background Management Platform shows here. Since Website Foreground Public Page and Android Application have the same purchase sequence logic, so the purchase sequence in these both platforms will be analysed in this chapter as well.

Background Management Platform

There are five main modules, such as Log In Module, Category Management Module, Product Info Management Module, Employee Management Module and Order Management Module. The sequence diagram will be given separately below.

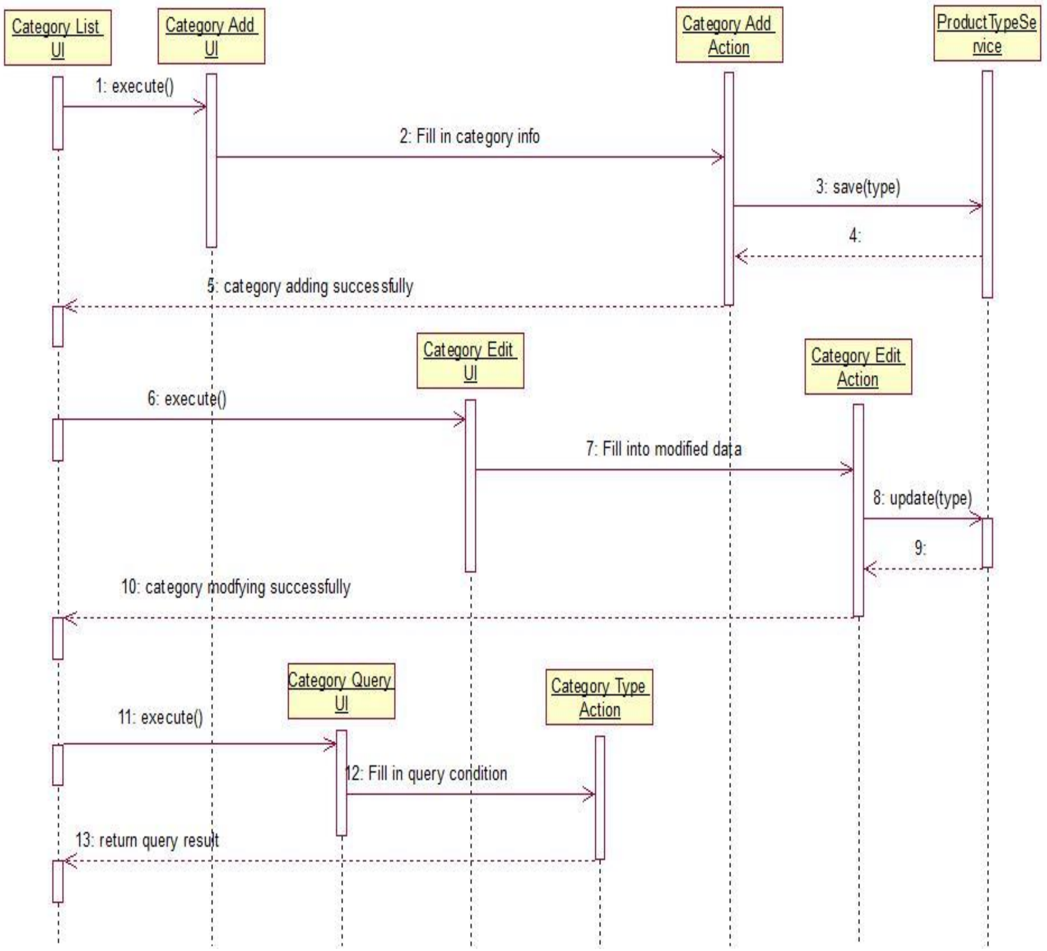
The administrator needs to login first before using the Background Management Platform. An account username and password for the owner of company will be pre-setted when the project is initialized.



Background Management Platform Login

The administrator fills in the username and password in the related fields and presses the login button. If the login is successful, the url will direct to the Background Management Platform main UI, otherwise, the user will be notified with an error message.

Application provides function to allow customers to view dishes according to different categories. Therefore, administrator needs to add to categories first.



Product Category Management

To add dish to display on the website or Android, the administrator should add dish categories first, in order that display dishes according to different categories.

1. Add Category:

In the Product Management, by clicking “Category Manage”, a list view page of current existing dishes will be shown. By clicking “Add Category” button, the right frame will jump to the category adding UI. Required data is filled in, then it is confirmed, and a dish will be added successfully. At the same time, the right frame will jump back to the category list view page, the new added dish will be displayed on this page.

2. Modify Category:

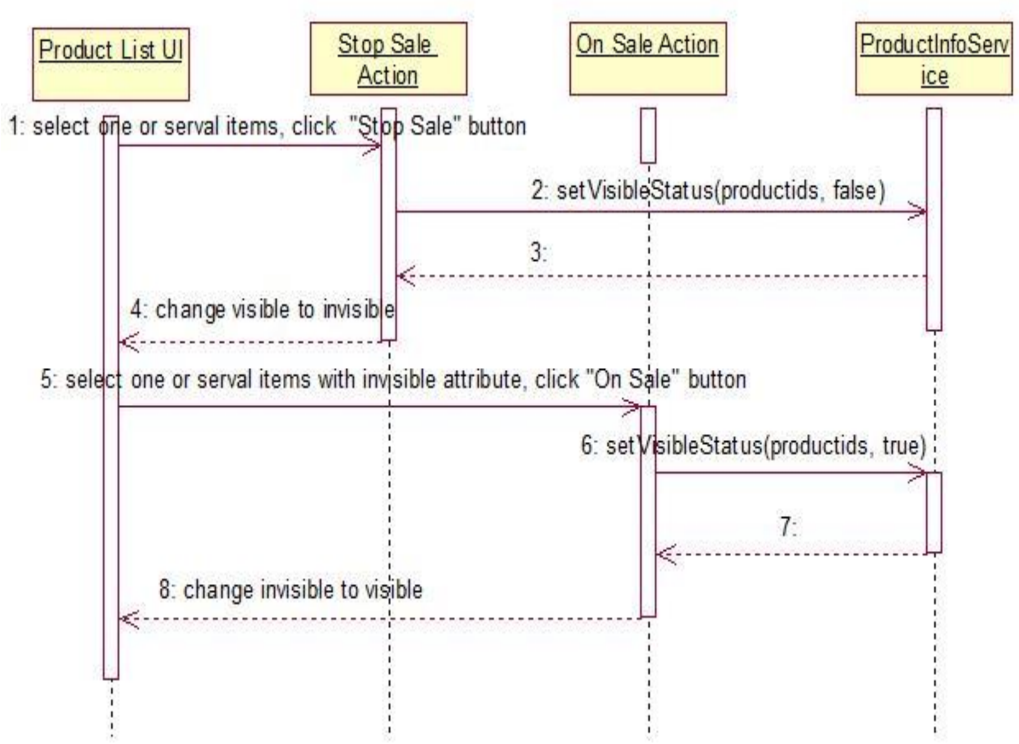
On the category list view page, every category item provides a “Modify” button, by using this button, the administrator can modify the existing dish information.

3. Query Category:

In addition, there is a “Query” button. This button can help the administrator to query some dish category among lots of categories.

Product Info Management

After adding categories, administrator can add dishes belong to different categories in Product Management module.


Product List UI
Stop Sale Action
On Sale Action
ProductInfoService
1: select one or serval items, click "Stop Sale" button
2: setVisibleStatus(productids, false)
3:
4: change visible to invisible
5: select one or serval items with invisible attribute, click "On Sale" button
6: setVisibleStatus(productids, true)
7:
8: change invisible to visible

Product Information Management

In the Product Management, there is a similar logic to that in Product Category Management, such as adding, modifying and querying. Therefore, no detailed description is given here.

Stop or start selling

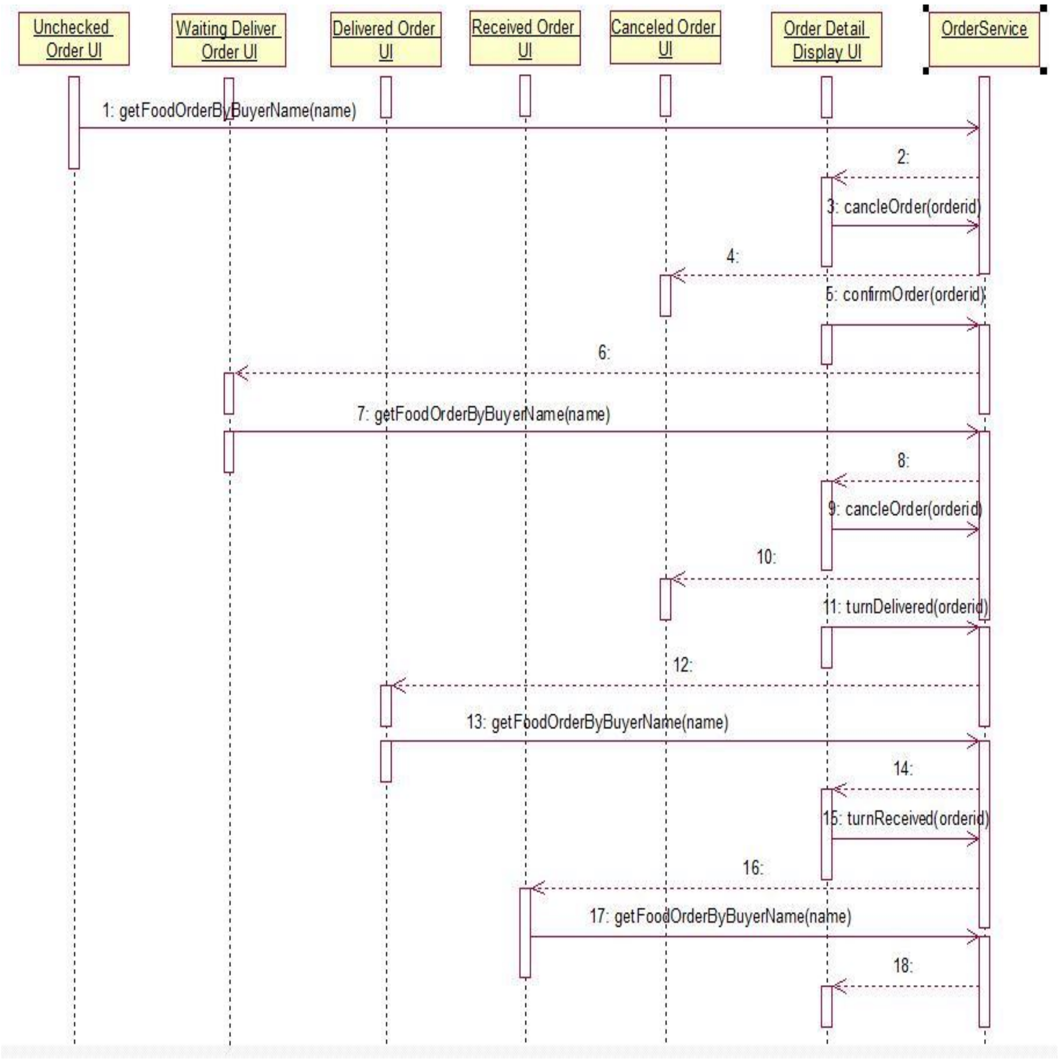
On the product list view page, there are two buttons: On Sale and Stop Sale. By using these two button, the administrator can control the sale state of food dish in order to display or remove a food dish on the website foreground public page or the Android application.

Employee Management

The Employee Management also has a similar logic to that referred above. The Mark Leave logic is also the same with the change sale state of food dishes. The employee accounts added by the administrator are used to log in and make use of the background management platform. Once the administrator marks an employee's leave, the employee account will not log in and use this platform any more.

Order Management

As mentioned in use case diagram for order flow, figure 19 demonstrates the entire ordering progress.

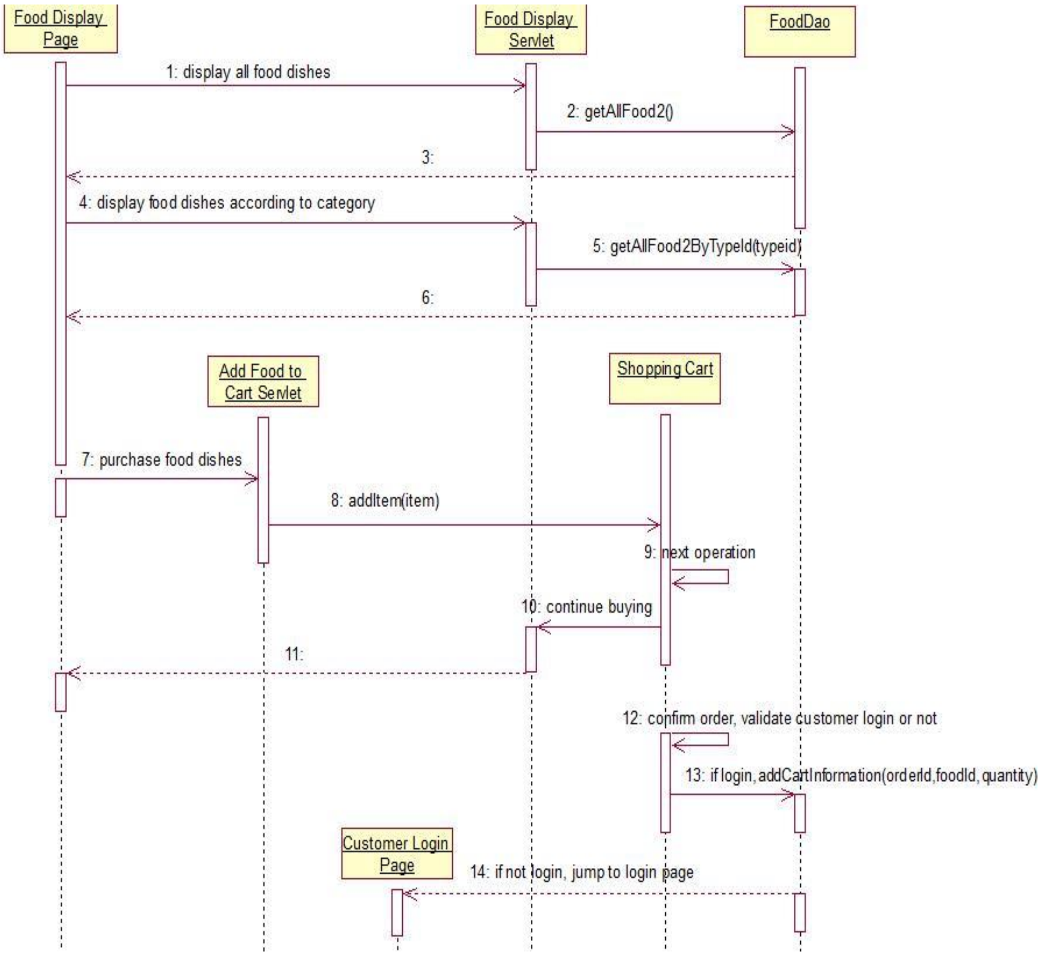


Order Management

The default state of produced order is unchecked, and the administrator can decide to change the state to waiting deliver or cancel. For waiting deliver state order, the administrator can change state to delivered or cancel. But for delivered order, the state cannot be changed any more.

Website Foreground Public Page and Android Application

Purchasing function is the core function in this application. So the purchase sequence diagram will be analysed as follow.


Food Display Page
Food Display Servlet
FoodDao
1: display all food dishes
2: getAllFood2()
3:
4: display food dishes according to category
5: getAllFood2ByTypeId(typeid)
6:
Add Food to Cart Servlet
Shopping Cart
7: purchase food dishes
8: addItem(item)
9: next operation
10: continue buying
11:
12: confirm order, validate customer login or not
13: if login, addCartInformation(orderId,foodId, quantity)
Customer Login Page
14: if not login, jump to login page

Customer Purchase sequence diagram

In the Website platform, the customer can view the information of food dishes without logging in. Before purchasing dishes, the customer needs to log in. The dish items purchased by the customer will be added to the shopping cart temporarily. In the shopping cart, the customer can modify the number of dishes and delete the dish items purchased. Moreover, the customer can select to continue to buy or confirm the order.

In the Android Application, the purchase flow is almost the same as that in the Website platform. The biggest difference is that in the Android Application, the customer must register a valid account to launch the application to do the further operations.

cancelled. In the Order Manage module, this project provides five order state links, “Order Unexamined”, “Order Wait Deliver”, “Order Delivered”, “Order Received”, “Order Cancelled”. In each link, there are the orders of related state. In addition, the “Order Query” link is used to query the order according to Order Code Or Buyer Name.

CONCLUSIONS

This project contains three parts, Background Management Platform, Website Public Page and Android Application. The Background Management Platform was implemented with Struts2, Spring and JPA framework. Website Public Page was achieved with Servlet/JSP and JavaBean. The Android framework was used in the Android Application.

This project was a typical combination between a website and an Android application. The aim of the project was to help the restaurant owner to improve the efficiency of managing, meanwhile, help the customer to purchase dishes in different platforms easily.

By now, the core function of this project has been implemented. The owner and employees in the company can manage dishes and handle dish orders and so on. On the public page, customers can view dish information and purchase dishes. Also, the customer can order dishes from the Android platform.

Developing the application made it possible to learn and practice the whole processes of agile development with S2SJ frameworks as well as Android framework.

Main Challenges in Developing:

✏ Struts2, Spring, JPA frameworks Integration

The Background Management Platform was achieved with Struts2 framework, Spring framework and JPA framework, the first challenge was how to integrate these three frameworks. For this task, the framework version, jar packages and configuration files needed to be considered and selected. Via checking related books and videos, the challenge was overcome.

✏ Android Network Programming

Because the project was a combination between a website and an Android application, the second challenge was how to transfer data between the website and the Android application. In order to overcome this challenge, the knowledge about the Android Network Programming was learned alone by means of different ways.