

企业级 Java 应用开发实践报告

课程代码：1302606

课程名称：企业级 Java 应用开发

课程类别：专业基础课

开课学期：2017 春季学期

编写单位：中国地质大学（北京）信息工程学院

项目负责人：门一凡

编写人：樊利豪、杜康晖、门一凡

指导教师：薛涛

提交班级：10041461 软件工程

提交时间：2017.6.2

目 录

| | |
|-----------------|----|
| 1 项目目的和任务 | 1 |
| 1.1 项目背景..... | 1 |
| 1.2 项目目标..... | 1 |
| 1.3 项目任务..... | 1 |
| 1.4 任务分工..... | 2 |
| 2 项目进度安排 | 2 |
| 3 系统需求分析 | 2 |
| 3.1 定义..... | 2 |
| 3.2 业务流程..... | 2 |
| 3.3 系统用例..... | 4 |
| 3.4 系统总体功能..... | 6 |
| 3.5 数据结构..... | 7 |
| 4 系统设计 | 7 |
| 4.1 系统架构设计..... | 7 |
| 4.2 系统功能设计..... | 8 |
| 4.3 数据库设计..... | 19 |
| 5 系统实现 | 20 |
| 6 项目总结 | 29 |

1 项目目的和任务

本项目的目的在于依托线上数据库进行数据管理，开发在线实时的点餐系统，形成一套完整的点餐到下单的流程，并通过设计与开发线上点餐系统，使开发者更加深刻的认识与学习 Java EE 的操作与使用，数据库的思想，实现一个较为完整的点餐系统。为将来参与企业级 Java 的开发、应用和研究奠定了基础。

本项目来源于中国地质大学（北京）信息工程学院软件工程专业项目实践，开发周期约为 35 天。

本项目由中国地质大学（北京）信息工程学院“饱了么”开发团队研发，可实现用户通过注册网址进行在线菜品点餐、下单，以及后台系统进行菜品、用户管理的功能。

1.1 项目背景

随着当代社会的信息科技爆炸式的发展，线上经营的方式给线下实体经营带来愈来愈大的冲击。为了提高一个实体经营者的利润与生存空间，开发一个线上的经营网站已经成为了现如今的经营者的必要手段。

开发一个线上经营的网站不仅可以提高实体经营者的工作效率，还可以拓宽来源，提高整体的服务水平和速度为规模化的经营打下坚实的基础。

饱了么的项目目标为全世界的适龄人口（5 岁~80 岁），以简洁的界面与灵活的操作方式吸引了累计超过 300 万用户的信赖与选择。本网站试用于就餐时及时下单与线外下单两种环境的应用。

1.2 项目目标

致力于更好更快、更简洁的点餐操作，减少服务器的压力与用户等待响应的的时间，并为饱了么拓宽宣传面，实现了更快捷的方式。

同时，项目致力于锻炼学生独立完成课题的工作能力及获取知识的能力，巩固 Java EE 基础知识，提高开发水平，掌握开发在线网站的方法及各种规范，为毕业后走上工作岗位打下良好基础。

1.3 项目任务

本项目为一个完整的、用户体验感极佳的在线点餐网站，对普通顾客而言，本项目能够满足注册账号，浏览菜品，搜索菜品，在线点餐，下单等基本操作；对管理员而言，本项目能够提供菜品的管理和用户的信息管理的功能，能够对饱了么网上订餐系统的管理员实现对与菜品的增删改查等操作。

普通顾客角度的任务：

1. 用户注册，登录，注销。
2. 用户个人信息管理
3. 浏览菜品：分页浏览，浏览菜品详情
4. 搜索菜品：实现关键字搜索
5. 加入购物车

6. 购物车管理：删除菜品，查看详情
7. 生成订单
8. 查看订单

管理员角度的任务：

1. 菜品的增删改查
2. 用户的信息查找
3. 正常的用户功能

1.4 任务分工

表 1 小组成员及分工

| 组员 | 组内角色 | 组内分工 |
|-----|------|---|
| 门一凡 | 组长 | 数据库访问与接口设计；用户信息管理；菜品信息管理；后台管理；过滤器模块开发；UI 设计与文档编写；菜品搜索；。 |
| 杜康晖 | 组员 | 菜品详情模块；分页浏览模块；搜索菜品模块；购物车模块的开发；日志管理模块与文档编写 |
| 樊利豪 | 组员 | 用户登录模块与注册模块开发；购物车模块；分页设计；数据库接口设计；UI 设计与文档编写 |

2 项目进度安排

本系统开发流程如表 2 所示

表 2 系统开发流程表

| 时间 | 工作安排 | 具体内容 |
|-----------|-----------|----------------------------|
| 5.3~5.5 | 数据库的设计与架设 | 设计数据库，完成数据库里具体的表的设计 |
| 5.6~5.10 | 数据库连接的实现 | 在 JavaEE 中实现数据库的接口，并与数据库连接 |
| 5.11~5.29 | 项目具体开发 | 开发普通顾客和管理员功能 |
| 5.30~6.2 | 文档编写 | 总结项目，编写文档 |

3 系统需求分析

3.1 定义

本系统是基于 Java EE 课程实践的项目开发系统，基于 Java 开发的 web 项目。项目名称为“饱了么”网上订餐系统，能够提供便捷流畅的订餐服务。系统提供两类用户使用：普通顾客能够通过浏览器注册、登录，能够在系统中订餐、访问个人中心等操作；管理员能够对菜品和用户进行管理。另外，本系统还提供搜索查询功能。

3.2 业务流程

本项目主要流程主要从用户身份出发辨识。普通顾客不用登录实现搜索、查看菜品信息，

注册登录后可以将菜品添加到购物车，能够去下单结算。普通顾客登录后可访问个人中心，查询或修改个人信息，查询历史订单等操作。管理员登录后分别进入菜品和用户管理模块，分别实现对菜品、用户的“增删改查”操作。

主要业务流程：

表 3 业务流程表

| 序号 | 业务逻辑 | 角色 |
|----|----------------------------|--------|
| 1 | 新用户注册：注册为用户，随后可以进行登录 | 顾客 |
| 2 | 登录：用户登录系统，进入主页 | 顾客/管理员 |
| 3 | 浏览菜品信息：用户不用登录可以直接浏览菜品和详情信息 | |
| 4 | 修改个人信息：用户可以进入信息中心进行查、改等操作 | 顾客 |
| 5 | 分页浏览菜品信息：首页、上下页、尾页 | |
| 6 | 加入购物车：用户可以需要的菜品加入到自己的购物车 | 顾客 |
| 7 | 查看购物车：用户可以查询自己的购物车并进行删改操作 | 顾客 |
| 8 | 生成订单：用户在订单页中填入信息确认下单 | 顾客 |
| 9 | 查询历史订单：用户可以按时间降序查询自己的所有订单 | 顾客 |
| 10 | 菜品管理：管理员对菜品信息进行增、删、改等操作 | 管理员 |
| 11 | 用户管理：管理员对顾客信息进行删、改、查等操作 | 管理员 |

系统业务流程图如下：

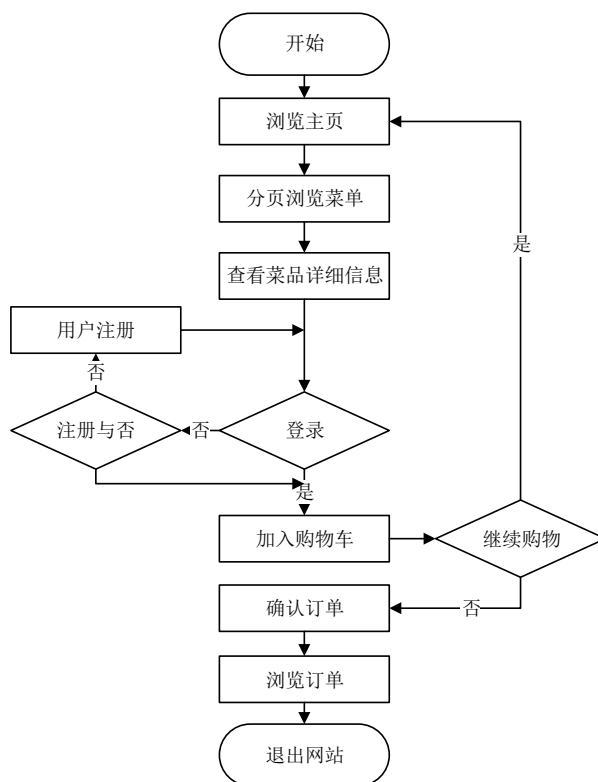


图 1 系统主业务流程图

3.3 系统用例

本系统为网上订餐平台，系统的主要参与者为顾客和管理员。

顾客的主要用例为登录、分页浏览菜品、查看菜品详情、添加菜品到购物车、查看购物车、生成订单、查看订单、修改个人信息等。管理员用户继承自顾客，除基础用例外，还能实现后台管理：可以管理菜品和管理用户。

系统的用例图如下：

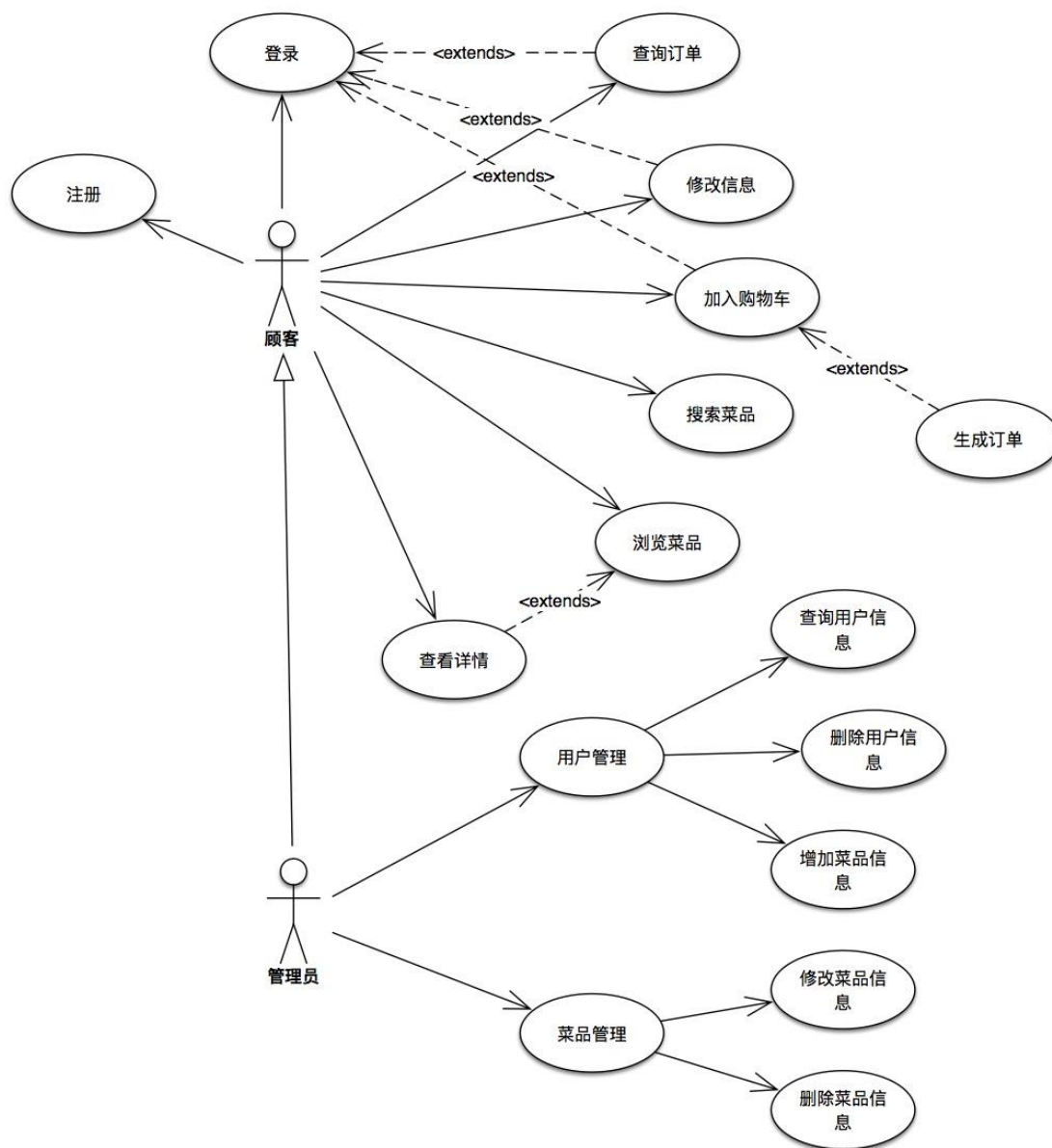


图 2 系统用例图

主要用例描述如下：

表 4 登录用例描述

| | |
|--------|--|
| 用例名称 | 登录 |
| 用例描述 | 输入手机号码和密码登录网站 |
| 参与者 | 顾客/管理员 |
| 前置条件 | 进入网站 |
| 后置条件 | 网站登录状态更新 |
| 基本操作流程 | 顾客通过输入注册的用户名和密码登录； 管理员通过系统给定的账号密码登录 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 加入购物车、查看订单 |

表 5 加入购物车用例描述

| | |
|--------|---|
| 用例名称 | 加入购物车 |
| 用例描述 | 用户通过分页浏览菜单、查看菜品详细信息， 将喜爱的菜品加入购物车 |
| 参与者 | 顾客或管理员 |
| 前置条件 | 登录 |
| 后置条件 | 更新当前用户购物车信息 |
| 基本操作流程 | 用户登录后，通过分页浏览菜单、查看菜品 详细信息，选定数量、点击加入购物车，将 菜品加入购物车 |
| 被包含的用例 | 无 |
| 被扩展的用例 | 生成订单 |

表 6 浏览菜品用例描述

| | |
|--------|---|
| 用例名称 | 浏览菜品 |
| 用例描述 | 用户分页浏览菜单上的菜品 |
| 参与者 | 顾客或管理员 |
| 前置条件 | 进入网站 |
| 后置条件 | 无 |
| 基本操作流程 | 顾客或管理员已登录或未登录，通过翻页浏 览菜单上的菜品，点击上下、首尾页浏览菜 品信息 |
| 被泛化的用例 | 无 |
| 被扩展的用例 | 查看详情信息 |

表 7 搜索菜品用例描述

| | |
|------|------------------|
| 用例名称 | 搜索菜品 |
| 用例描述 | 用户通过搜索栏搜索想要的菜品信息 |
| 参与者 | 顾客或管理员 |
| 前置条件 | 进入网站 |

| | |
|--------|---------------------------------------|
| 后置条件 | 显示搜索结果分页菜品信息 |
| 基本操作流程 | 用户进入网站后，直接选择使用搜索栏输入关键词，即可搜索出想要的菜品分页结果 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 无 |
| 被扩展的用例 | 无 |

表 8 添加菜品用例描述

| | |
|--------|---|
| 用例名称 | 添加菜品 |
| 用例描述 | 管理员向数据库中添加新菜品 |
| 参与者 | 管理员 |
| 前置条件 | 登录为管理员 |
| 后置条件 | 分页中菜品信息添加 |
| 基本操作流程 | 管理员登录网站后，进入菜品管理页，选择添加菜品信息，输入对应信息后，确认添加到数据库菜品信息中 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 无 |
| 被扩展的用例 | 无 |

3.4 系统总体功能

本系统为实现网上订餐系统，系统从用户角色的考虑可分为顾客订餐和后台管理两部分，对于不同的用户角色功能各异。

顾客的主要需求为网上订餐，其主要功能为登录注册、分页浏览菜品信息、查看菜品详情、添加菜品到购物车、生成订单、访问个人中心、查看历史订单等。

管理员主要进行后台管理，能够实现用户管理和菜品管理。用户管理主要包括查看用户信息、删除用户信息，菜品管理包括对菜品信息的添加、修改、删除等操作。

系统的总体功能图如下：

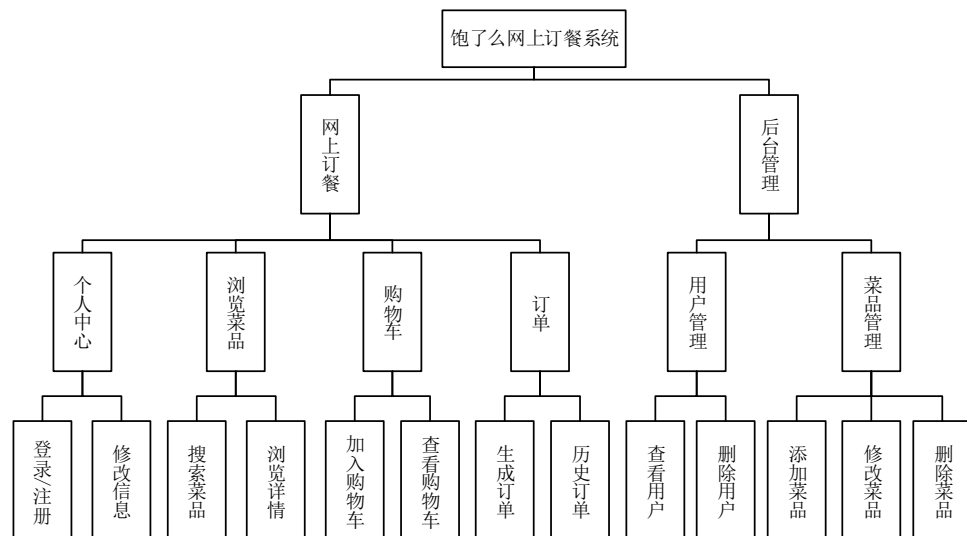


图 3 系统总体功能图

3.5 数据结构

本系统数据字典中数据结构设计如下：

表 9 数据结构表

| 数据结构名 | 含义说明 | 组成 |
|-------|--------------|-------------------------------|
| 菜品 | 定义了一个菜品的详细信息 | 菜品 ID、菜品名称、图片、菜品价格、菜品折扣价、描述信息 |
| 用户 | 系统用户的主要信息 | 用户名、密码、邮箱 |
| 订单 | 用户生成的订单信息 | 订单号、用户名、时间、地址、电话、支付状态、总额、数量 |
| 订单详情 | 订单中的每一项详细信息 | 订单号、菜品 ID、数量、金额 |

4 系统设计

4.1 系统架构设计

本系统架构主要分为表现层、逻辑层、数据库层，逻辑层分为业务层与数据访问层，其主要功能如下：

1. 表示层：显示网站的界面效果，实现简单的逻辑控制，同时将复杂处理提交给业务层
2. 业务层：封装了实际业务逻辑，包含数据验证，事物处理，权限处理等业务相关操作，是整个应用系统的核心。
3. 业务数据访问层：一个针对具体应用系统的专属层，它为业务层提供与数据源交互的最小操作方式，仅仅是业务层需要的数据访问接口，业务层完全依赖业务数据访问层所提供的服务。
4. 数据访问层：为数据源提供一个可供外界访问的接口。

系统架构图如下：

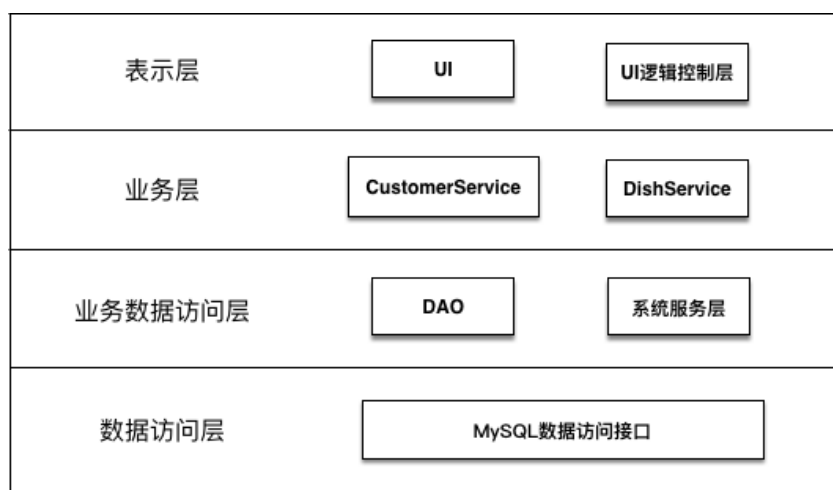


图 4 系统架构图

4.1.1 系统开发采用的技术手段

- ☒ HTML5、CSS3、JavaScript
- ☒ J2EE web 程序开发技术
- ☒ SQL 数据库存储技术

4.1.2 系统开发环境

- ☒ Java 1.8
- ☒ Tomcat 8
- ☒ MySQL 数据库
- ☒ Eclipse Neon.3
- ☒ Windows 10 / macOS 10.11.6 操作系统

4.1.3 系统部署实施环境

(1)系统部署主要采用数据库服务器：

- ☒ Windows 10、macOS、Linux 操作系统或以上版本
- ☒ MySQL 5.0 或以上版本

(2)系统部署主要采用客户机：

- ☒ Windows Xp 或以上版本
- ☒ Chrome v 58.0.3029.110 或其他浏览器

4.2 系统功能设计

4.2.1 用户模块

用户模块主要负责用户的登陆与注册，用户需要使用用户名、密码、邮箱注册，登录时仅凭注册时的用户名和密码即可完成登陆。

1. 流程图

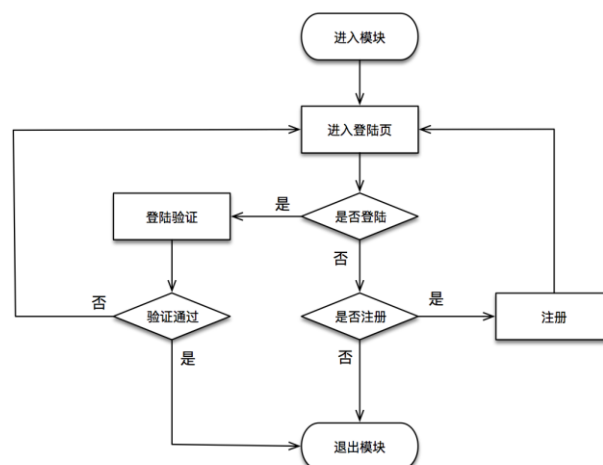


图 5 用户模块流程图

2. 类

表 10 用户模块类表

| | |
|------------|-----------------------|
| 实体类 | Customer |
| 数据库操作类 | JDBCUtil |
| 数据访问接口（接口） | ICustomerDAO |
| 数据访问接口类（类） | ICustomerDAOMySQLImpl |
| 业务逻辑类 | CustomerService |
| 工厂类 | DAOFactory |

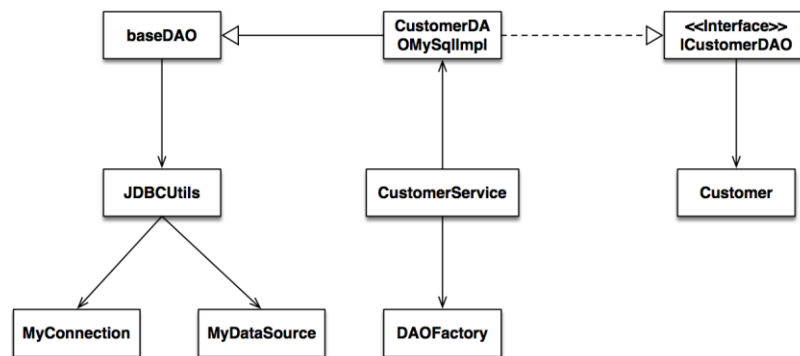


图 6 用户模块类图

3. 主要接口和类的方法

表 11 用户模块主要类和接口说明

| 接口 / 类名称 | 函数名称及参数 | 返回类型 | 完成功能 |
|-------------------|-------------------------------------|-----------|---------------------------------------|
| 接口 ICustomerDAO | addCustomer(Customer customer) | int | 传递一个用户的信息，返回值为增加的用户数量，1 表示成功，0 表示失败。 |
| | removeCustomer(String string) | int | 根据用户的用户名删除用户，返回值为删除的用户数量，1 为成功，0 为失败。 |
| | modifyCustomer(Customer customer) | int | 根据用户的用户名修改用户数据，修改的用户数量，1 为成功，0 为失败。 |
| | findCustomers() | ArrayList | 查找所有用户 |
| | findCustomer(String string) | Customer | 查找指定用户名的用户信息 |
| 类 CustomerService | isExistCustomer(String username) | boolean | 判断用户是否存在 |
| | validateCustomer(Customer customer) | boolean | 验证用户密码是否匹配 |

| | | | |
|-----------------------------|---|------------|--|
| 类 JDBCUtils | getConnection() | Connection | 读取配置文件中有关连接数据库的信息，为连接数据库提供服务，连接数据库。 |
| | free(ResultSet rs,PreparedStatement ps,Connection conn) | void | 释放资源。 |
| 类 Customer | getUsername() | String | 获取用户名 |
| | setUsername(String username) | void | 设置用户名 |
| | getEmail() | String | 获得用户邮箱 |
| | setEmail(String username) | void | 设置用户邮箱 |
| | getPassword() | String | 获得用户密码 |
| | setPassword(String password) | void | 设置用户密码 |
| 类 BaseDAO | MappingObj(ResultSet rs,Class clazz) | Object | 处理 sql 数据库与 Eclipse 之间数据类型的转换 |
| | findobjs(String sql,Class clazz) | ArrayList | 从数据库中查找满足条件的所有条目并返回 |
| | modify(String sql,Object[] params) | int | 实现对数据库数据的增删改操作 |
| | findObj(string,object[],class) | object | 从数据库中查找满足条件的所有条目并返回 |
| 类 ICustomerDAO MySqlImpl | 同接口 ICustomerDAO | | 继承 baseDAO 类，并且实现接口 ICustomerDAO 中的各函数 |

4.2.2 菜品模块

菜品模块主要负责对于菜品的显示，根据用户访问的页数和每页的大小动态的加载不同的数据

1. 流程图

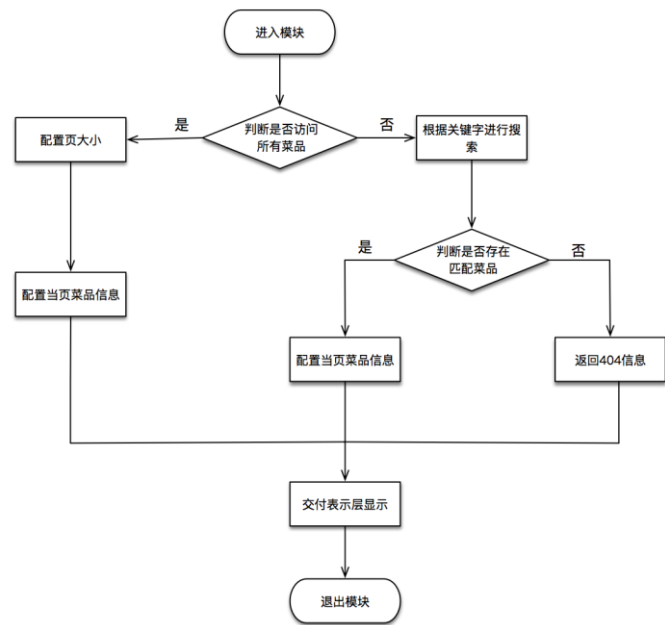


图 7 菜品模块流程图

2. 类

表 12 菜品模块类表

| | |
|------------|-------------------|
| 实体类 | Dish |
| 数据库操作类 | JDBCUtil |
| 数据访问接口（接口） | IDishDAO |
| 数据访问接口类（类） | IDishDAOMySqlImpl |
| 业务逻辑类 | DishService |
| 工厂类 | DAOFactory |

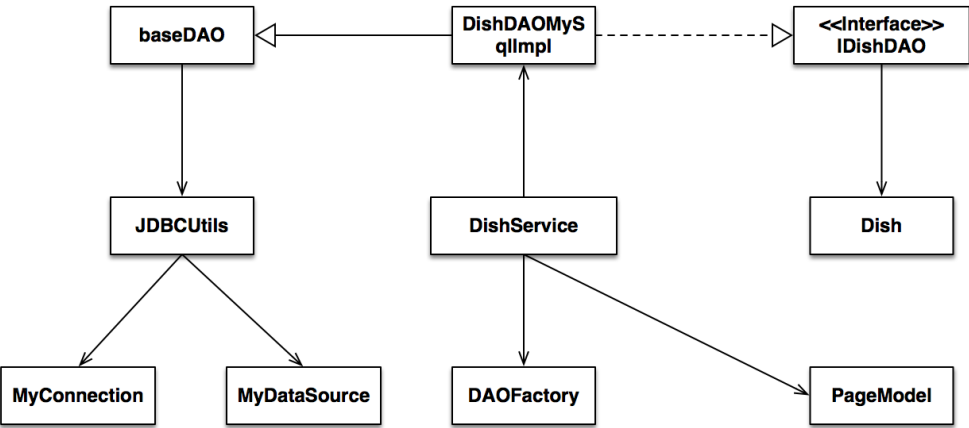


图 8 菜品模块类图

3. 主要接口和类的方法

表 13 用户模块主要类和接口说明

| 接口 / 类 名称 | 函数名称及参数 | 返回类型 | 完成功能 |
|---------------|---|-----------------|--|
| 接口 IDishDAO | addDish(Dish dish) | int | 传递一个菜品的信息，返回值为增加的菜品的数量，1 表示成功，0 表示失败。 |
| | removeDish(int dishid) | int | 根据菜品的 id 删除菜品，返回值为删除的用户数量，1 为成功，0 为失败。 |
| | modifyDish(Dish dish) | int | 根据菜品的菜品 id 修改用户数据，修改的菜品数量，1 为成功，0 为失败。 |
| | findDishes() | ArrayList | 查找所有菜品 |
| | findDish(int dishid) | Dish | 查找指定菜品 id 的菜品信息 |
| 类 DishService | getDishdao() | IDishDAO | 获取当前对象中 Dish 数据访问层接口 |
| | setDishdao(IDishDAO dishdao) | void | 设置当前对象 Dish 的数据访问层接口 |
| | findDish4PageList(int pageNO,int pageSize) | PageModel<Dish> | 获取正常显示的菜品的信息，以 PageModel 的形式返回 |
| | findDish5PageList(int pageNO,int pageSize,String str) | PageModel<Dish> | 获取搜索结果的菜品的信息，以 PageModel 的形式返回 |
| | getTotalDishes() | int | 返回菜品总数 |
| | getPerDishes(String str) | int | 返回符合关键词条件的菜品总数 |
| 类 JDBCUtils | getConnection() | Connection | 读取配置文件中有关连接数据库的信息，为连接数据库提供服务，连接数据库。 |
| | free(ResultSet rs,PreparedStatement ps,Connection conn) | void | 释放资源。 |
| 类 Dish | getDishID () | int | 获取菜品编号 |
| | setDishID(int dishID) | void | 设置菜品编号 |
| | getName () | String | 获得菜品名称 |
| | setName(String name) | void | 设置菜品名称 |

| | | | |
|-------------------------|---------------------------------------|-----------|-------------------------------------|
| | getPrice() | float | 获取菜品原价 |
| | setPrice(float price) | void | 设置菜品原价 |
| | getDescription() | String | 获取菜品描述 |
| | setDescription(String description) | void | 设置菜品描述 |
| | getImgURL() | String | 获取菜品图片 URL |
| | setImgURL(String imgURL) | void | 设置菜品图片 URL |
| | getDiscount() | float | 获取菜品优惠价 |
| | setDiscount(float discount) | void | 设置菜品优惠价 |
| | picSize(String pxl) | String | 获取指定大小的图片信息 |
| 类 BaseDAO | MappingObj(ResultSet rs, Class clazz) | Object | 处理 sql 数据库与 Eclipse 之间数据类型的转换 |
| | findobjs(String sql, Class clazz) | ArrayList | 从数据库中查找满足条件的所有条目并返回 |
| | modify(String sql, Object[] params) | int | 实现对数据库数据的增删改操作 |
| | findObj(string, object[], class) | object | 从数据库中查找满足条件的所有条目并返回 |
| 类 IDishDAO MySQLImpl | 同接口 IDishDAO | | 继承 baseDAO 类, 并且实现接口 IDishDAO 中的各函数 |

4.2.3 购物车模块

购物车模块主要负责临时存储用户想要购买的商品信息, 在用户下单前可进一步确认和查看详细情况。

1. 流程图

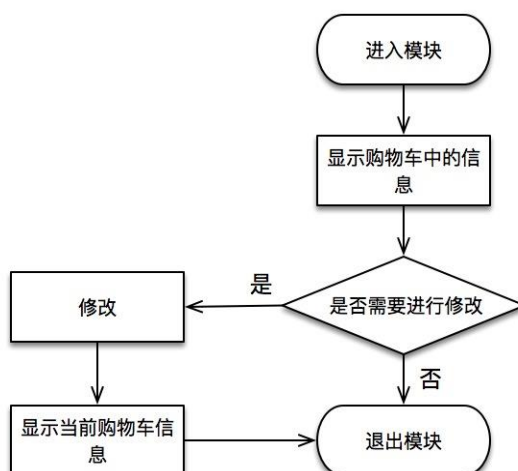


图 9 购物车模块流程图

2. 类图

表 14 购物车模块类表

| | |
|------------|-------------------|
| 实体类 | CartItem, Dish |
| 数据库操作类 | JDBCUtil |
| 数据访问接口（接口） | IDishDAO |
| 数据访问接口类（类） | IDishDAOMysqlImpl |
| 业务逻辑类 | DishService |
| 工厂类 | DAOFactory |

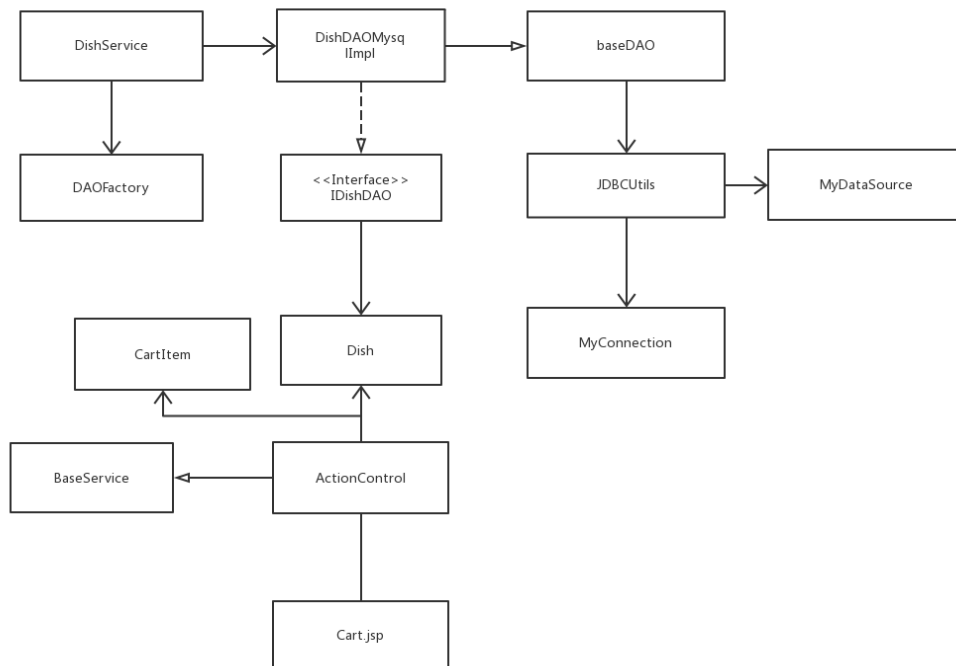


图 10 购物车模块类图

3. 主要接口和类的方法

表 15 购物车模块主要类和接口说明

| 接口 / 类名称 | 函数名称及参数 | 返回类型 | 完成功能 |
|-----------------|------------------------|------|---|
| 接口 ICustomerDAO | addDish(Dish dish) | int | 传递一个用户的信息，返回值为增加的菜品的数量，1 表示成功，0 表示失败。 |
| | removeDish(int dishid) | int | 根据菜品的菜品编号删除菜品，返回值为删除的菜品的数量，1 为成功，0 为失败。 |
| | modifyDish(Dish dish) | int | 根据菜品的菜品编号名修改菜品数据，修改的菜品数量，1 为成功，0 为失败。 |

| | | | |
|-------------------|---|-----------------|-------------------------------------|
| | findDishes() | ArrayList | 查找所有菜品。 |
| | findDish(int dishid) | Dish | 查找指定菜品编号菜品信息。 |
| | findDishes(String sql,Object[] params) | ArrayList | 查找对应的 SQL 语句的菜品。 |
| | getTotalDishes(String strsql) | Int | 返回所有菜品的数量。 |
| | findMaxDish() | Dish | 返回最大菜品编号的菜品信息 |
| 接口 DishService | findDish4PageList | PageModel<Dish> | 进行菜品信息的分页。 |
| | findDish5PageList | PageModel<Dish> | 进行搜索出来的菜品信息的分页。 |
| | getTotalDishes() | int | 返回所有菜品的总数量。 |
| | getPerDishes(String str) | int | 返回指定的菜品名称的总数量。 |
| 类 JDBCUtils | getConnection() | Connection | 读取配置文件中有关连接数据库的信息，为连接数据库提供服务，连接数据库。 |
| | free(ResultSet rs,PreparedStatement ps,Connection conn) | void | 释放资源。 |
| 类 Dish | getDishID | int | 获取菜品编号 |
| | setDishID (int dishID) | void | 设置菜品编号 |
| | getName() | String | 获得菜品名称 |
| | setName(String name) | void | 设置菜品名称 |
| | getPrice() | float | 获得菜品原价 |
| | setPrice(float price) | void | 设置菜品原价 |
| | getDescription() | string | 获得菜品介绍 |
| | setDescription(String description) | void | 设置菜品介绍 |
| | getImgURL() | string | 获得图片地址 |
| | setImgURL(String imgURL) | void | 设置图片地址 |

| | | | |
|---------------|---------------------------------------|-----------|-------------------------------------|
| | getDiscount() | float | 获得菜品现价 |
| | setDiscount(float discount) | void | 设置菜品现价 |
| 类 BaseDAO | MappingObj(Result Set rs,Class clazz) | Object | 处理 sql 数据库与 Eclipse 之间数据类型的转换 |
| | findobjcs(String sql,Class clazz) | ArrayList | 从数据库中查找满足条件的所有条目并返回 |
| | modify(String sql,Object[] params) | int | 实现对数据库数据增删改操作 |
| | findObj(string,object[],class) | object | 从数据库中查找满足条件的所有条目并返回 |
| CartItem | getUsername() | string | 得到用户名 |
| | setUsername | Void | 设置用户名 |
| | getId() | int | 获取菜品 ID |
| | setId(int id) | Void | 设置菜品 ID |
| 类 DishDAOImpl | 同接口 IDishDAO | | 继承 baseDAO 类, 并且实现接口 IDishDAO 中的各函数 |

4.2.4 订单结算模块

订单模块是由 JSP 页面和 Servlet 联动实现的, 当用户购物车为空时, 直接跳转回购物车, 不允许用户生成订单。若购物车有商品, 则按照顺序预览, 填写个人信息, 生成订单, 然后跳转至用户的订单页查看。

1. 流程图

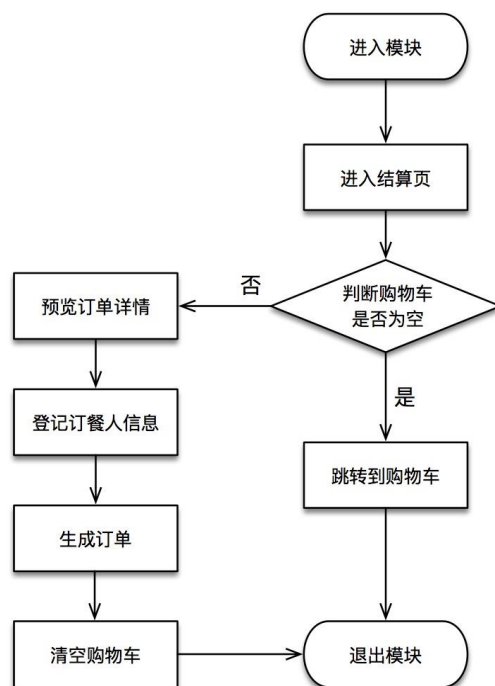


图 11 订单结算流程图

2. 类图

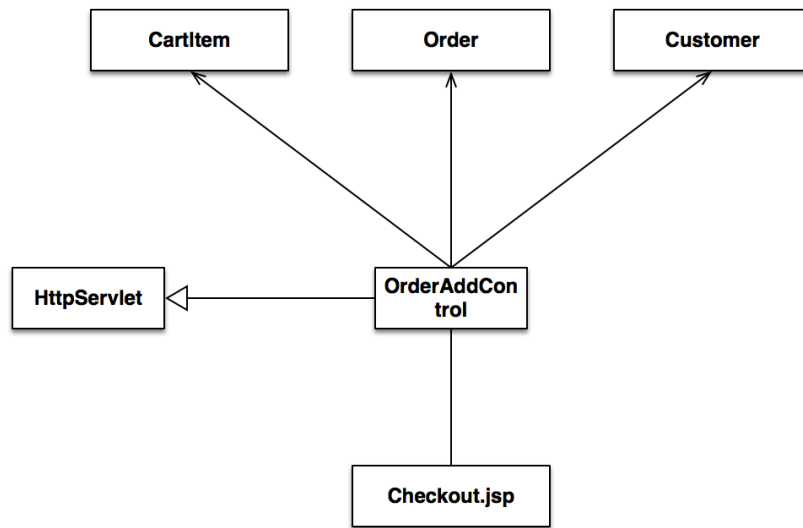


图 12 订单结算类图

3. 主要接口和类的方法

表 16 订单结算模块主要类和接口说明

| 接口 / 类名称 | 函数名称及参数 | 返回类型 | 完成功能 |
|-----------------------------|--|-----------------|---------------------|
| Servlet: OrderAddControl | OrderAddControl() | OrderAddControl | 该 Servlet 的构造函数 |
| | doGet(HttpServletRequest request, HttpServletResponse response) | void | 处理购物车中的商品，并且生成订单 |
| | doPost(HttpServletRequest request, HttpServletResponse response) | void | 使用 doGet 处理 post 请求 |
| 实体类: CartItem | getUsername() | String | 获取当前项目所属的用户 |
| | setUsername(String username) | void | 设置当前项目所属的用户 |
| | getId() | int | 返回菜品的 id |
| | setId(int id) | void | 设置菜品的 id |
| 实体类: Customer | setUsername(String username) | void | 设置用户名 |
| | getUsername() | String | 获取用户名 |

| | | | |
|------------|------------------------------|-----------|------------|
| | getEmail() | String | 获得用户邮箱 |
| | setEmail(String username) | void | 设置用户邮箱 |
| | getPassword() | String | 获得用户密码 |
| | setPassword(String password) | void | 设置用户密码 |
| 实体类: Order | getOrderID() | String | 获取订单 ID |
| | setOrderID(String orderID) | void | 设置订单的 ID |
| | getTime() | Timestamp | 获取订单时间戳 |
| | setTime(Timestamp time) | void | 设置订单时间戳 |
| | getCount() | int | 返回总菜品数量 |
| | setCount(int count) | void | 设置总菜品数量 |
| | updateCount() | void | 更新总菜品数量 |
| | getUsername() | String | 获取订单所属用户 |
| | setUsername(String username) | void | 设置订单所属用户 |
| | getAddress() | String | 获取用户收货地址 |
| | setAddress(String address) | void | 设置用户收货地址 |
| | getTel() | String | 获取用户手机号 |
| | setTel(String tel) | void | 设置用户手机号 |
| | getTotalPrice() | float | 获取订单总价 |
| | updateTotalPrice() | void | 根据菜品更新订单总价 |

| | | | |
|--|--------------------------------------|----------------------|----------|
| | setTotalPrice(float totalprice) | void | 设置订单总价 |
| | getPayStatus() | String | 获取订单支付状态 |
| | setPayStatus(String payStatus) | void | 设置订单支付状态 |
| | getItems() | ArrayList<OrderItem> | 获取菜品信息 |
| | setItems(ArrayList<OrderItem> items) | void | 设置菜品信息 |

4.3 数据库设计

4.3.1 概念结构设计

本次概念结构设计采用 E-R 图来表示。概念结构中涉及到的实体有：用户、菜品、订单、订单项。每个实体的属性如下：

- 1) 用户：用户名（主码）、密码、邮箱
- 2) 菜品：菜品编号（主码）、菜品名、原价、描述、图片、折扣价
- 3) 订单：订单号（主码）、下单时间、数量、总价、支付状态、地址、联系电话、用户名（外键）
- 4) 订单项：数目、价格、（菜品编号（外键）、订单号（外键））（主码）

实体之间的关系描述如下：

用户与订单：1: n

订单与订单项：1: n

订单项与菜品：n: 1

逻辑结构模型设计如下：

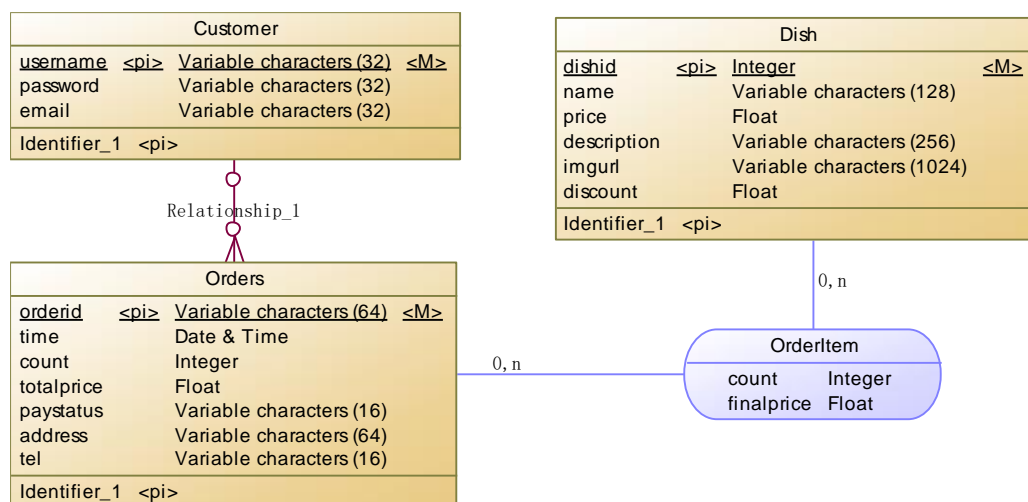


图 13 概念模型

4.3.2 逻辑结构设计

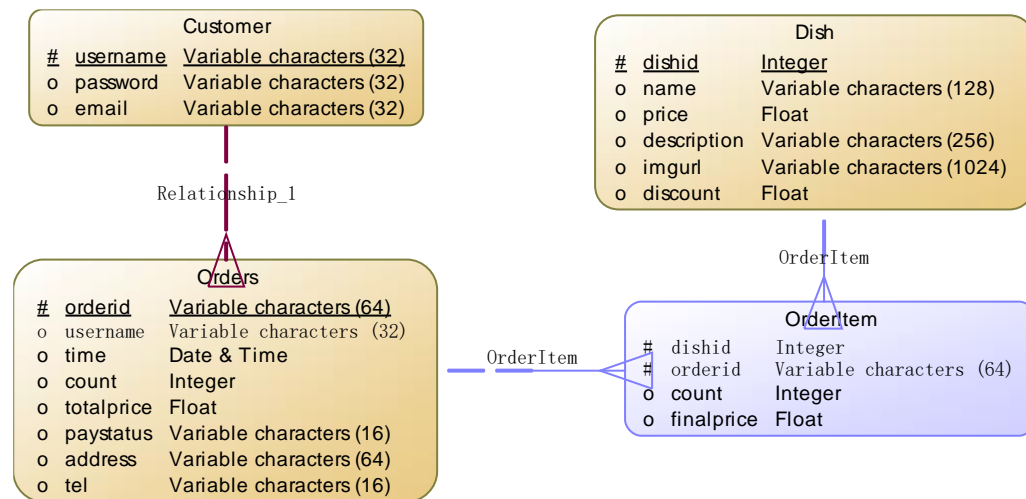


图 14 逻辑模型

4.3.3 物理结构设计

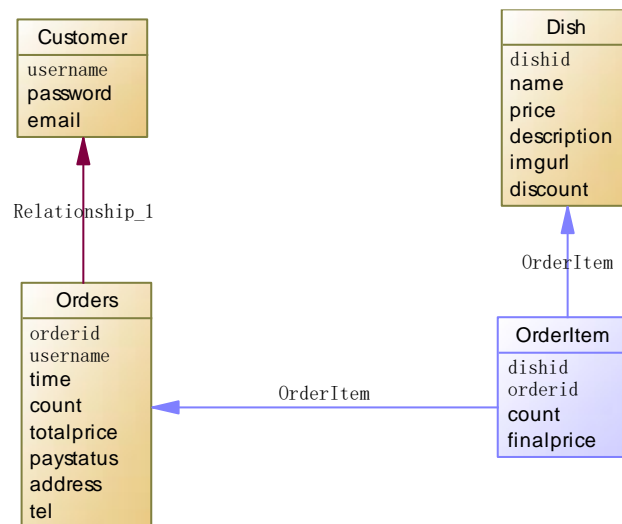


图 15 物理模型

5 系统实现

5.1 主页设计

秉承生饱了么简洁美观大方的时尚观念，服务至上客户第一的经营理念，这里出售的不仅是美食，还有优质的服务与极佳的客户体验；在每一个菜品后面都有一个庞大的团队精心为您准备。



图 16 主页

5.2 登录页面设计

用户点导航栏的登录按钮后，即可进入登陆页面，输入用户名和密码，即可登录，登陆后跳转到首页。



The login page features a header with a '请登录' (Please Login) button on the left and '我的购物车' (My Cart) and '请登录' (Please Login) links on the right. Below the header is a navigation bar with links: '首页' (Home), '购物' (Shopping), '我的订单' (My Orders), '修改信息' (Modify Information), and '联系作者' (Contact Author). A search bar labeled '搜索美味' (Search for Delicacies) is also present. The main content area is divided into two sections: '登录到您的账户' (Login to your account) and '注册新用户' (Register new user). The login section has input fields for '用户名' (Username) and '密码' (Password), followed by a '登录' (Login) button. The registration section has input fields for '用户名' (Username), '邮箱' (Email), and '密码' (Password), followed by a '注册' (Register) button. An 'OR' button is placed between the two sections.

图 17 登陆页面

5.3 注册页面设计

点击登录页面的同时，若是没有账户，即可在登陆的旁边使用注册页面，用户名、邮箱、密码后，验证是否注册成功，如果注册成功就会显示提示。



The registration page is similar to the login page but includes an additional '注册' (Register) button in the registration section. A modal dialog box is displayed in the center, showing the message 'localhost:8080 显示：注册成功+请登录！' (localhost:8080 displays: Registration successful + Please login!) with a '确定' (Confirm) button. The background shows the same navigation and search elements as the login page, with the registration form fields filled with example data: '5054' for username, '123@qq.com' for email, and '*****' for password.

图 18 注册页面

当注册失败，会跳转到提示页面，提示用户重新注册。



图 19 提示注册失败页面

5.4 分页浏览菜单页面

点击导航栏的菜单，即可进入分页浏览菜单页面，这边展示是所有的菜品，通过分页，将每页的菜品控制在最多 12 道。



图 20 分页浏览菜单页面

5.5 菜品详细信息页面

点击上述页面中的图片或者菜品名称，即可进入该菜品的详细信息页面，能够看到菜品的描述和菜品的价格及许多有趣的事情。

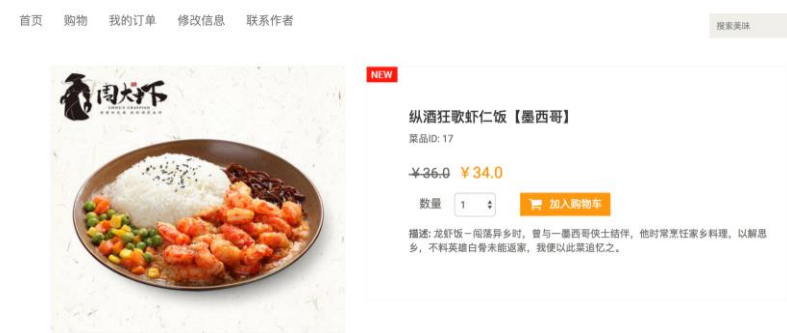


图 21 查看菜品详情信息页面

5.6 购物车页面

点击各个页面的添加到购物车按钮,即可把菜品添加到购物车,在首页添加的时候每点击一次购物车的按钮,及可以把对应菜品放入购物车一个,在菜品详情页面可以一次性加入多个物品,可以删除菜品。

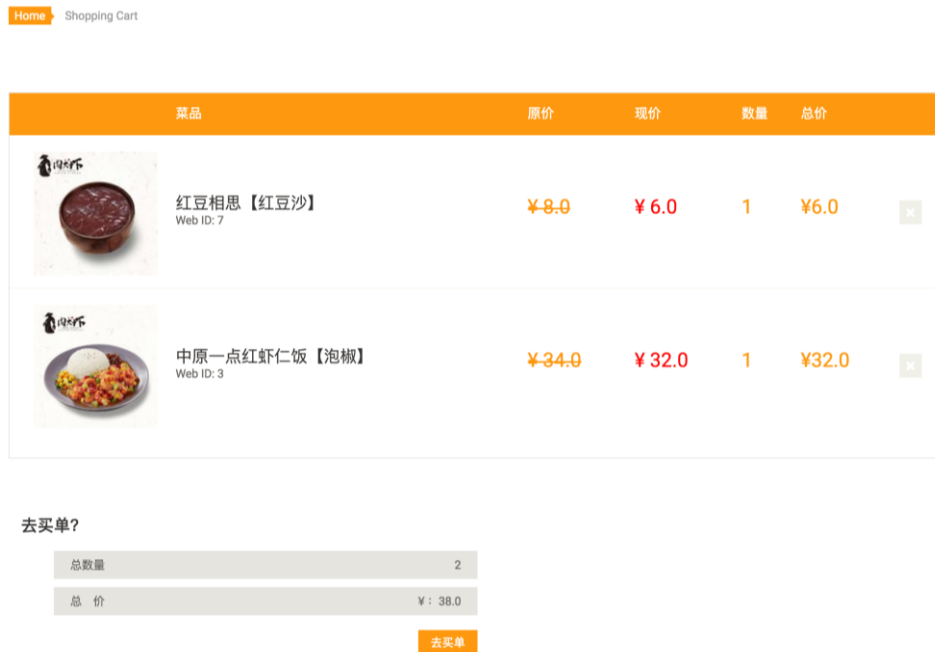


图 22 购物车页面

5.7 订单页面

点击购物车页面的前往支付,就可以生成订单,订单页面如下,可以看到订单号、订单金额、以及订单状态,如果没有付款,可以前往付款页面。

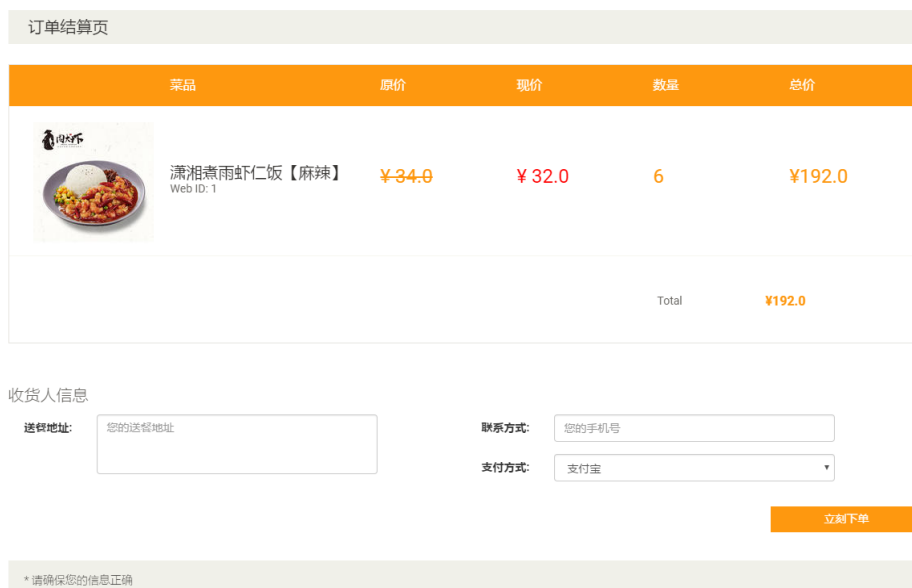


图 23 订单页面

5.8 搜索页面

点击右上角的搜索按钮以后，可以根据文字框中所属的内容进行搜索，可以看到搜索出来的菜品，若是没有对应的信息也会出现提示



图 24-1 搜索页面 a（以虾仁为例）



图 23-2 搜索页面 b（未找到相关信息页面）

5.9 后台管理

后台管理页面是一个可视化的管理页面，管理员在登录之后，看到的界面是与用户不一样的，具体分为用户管理与菜品管理

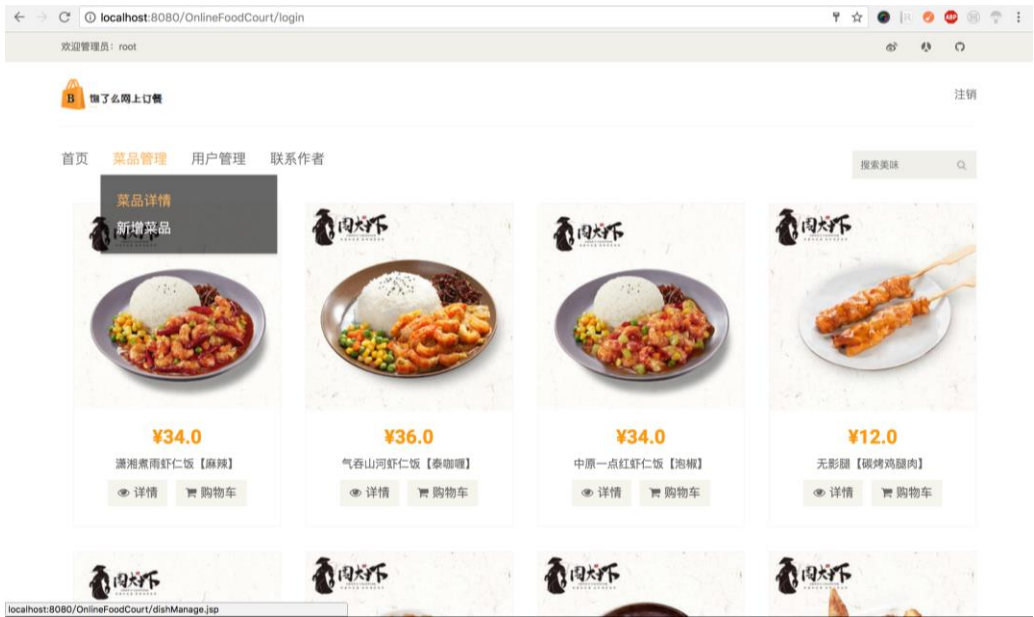


图 25 管理员登录等界面



图 26 菜品管理页

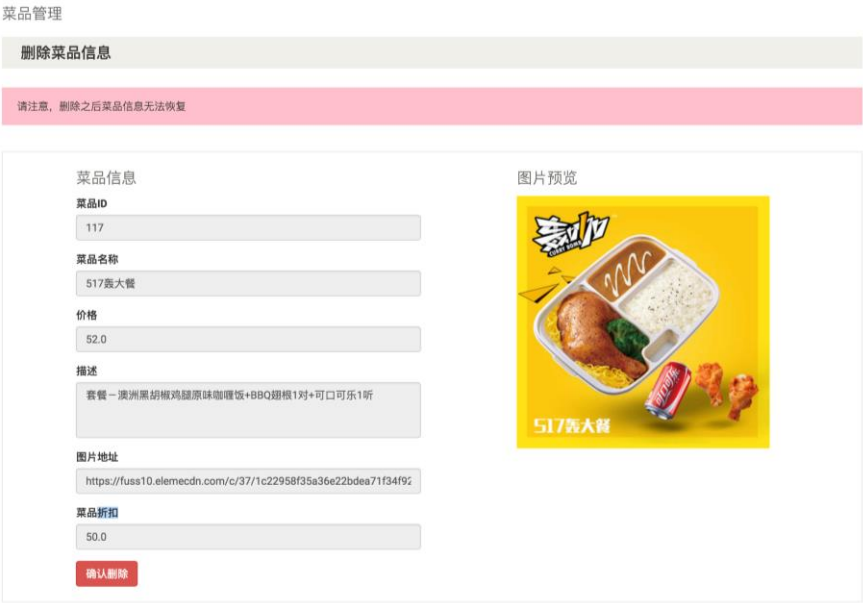


图 27 菜品删除页面

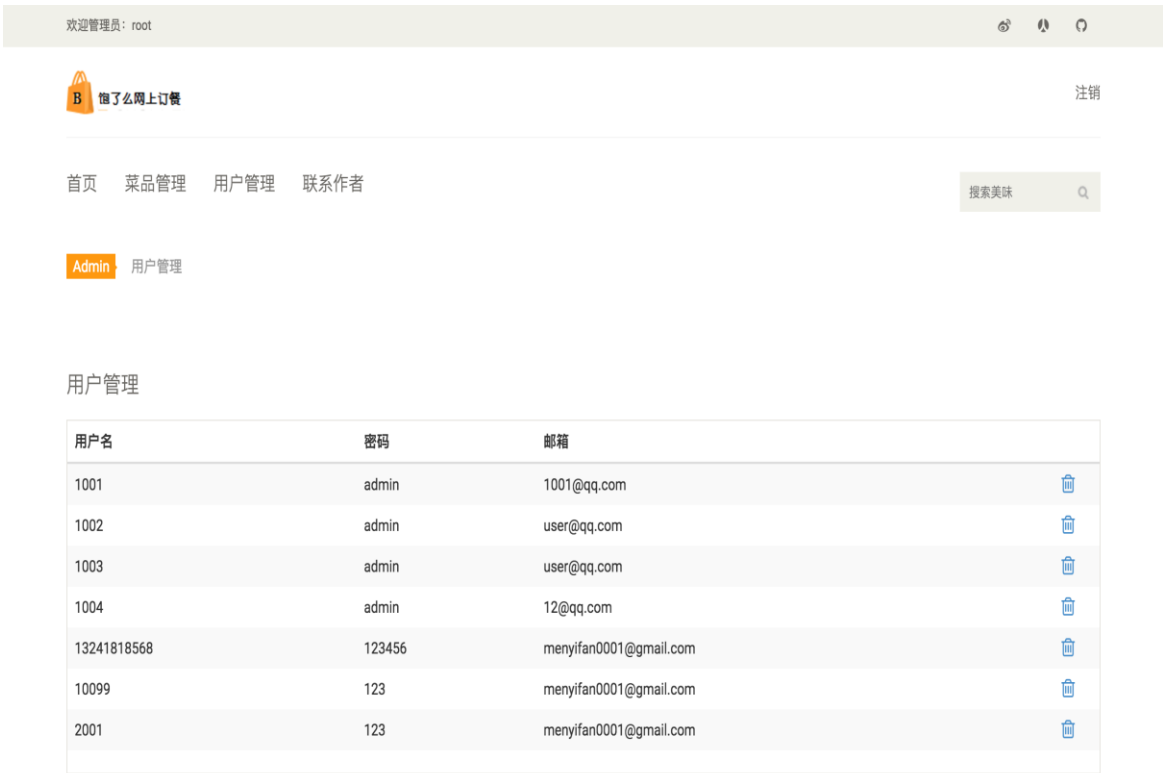


图 28 用户信息查看页

用户管理

删除用户信息

请注意，删除之后用户信息无法恢复

用户信息

用户名

13241818568

密码

123456

昵称

menyifan0001@gmail.com

确认删除

图 29 用户删除页面

菜品管理

修改菜品信息

请注意，修改之后将无法恢复原信息

菜品信息

菜品ID

1

菜品名称

潇湘煮雨虾仁饭【麻辣】

价格

34.0

描述

龙虾饭一拍案，手起刀落，拆头去壳取线的整只龙虾跃然入锅，滚油中红辣青麻绕一圈，铁勺荡起盛盘，咸鲜麻辣。

图片地址

https://fuss10.elemecdn.com/1/aa/c50c5db007533aeb6bbee1ee9d

菜品折扣

32.0

确认修改

图片预览




图 30 菜品信息修改页面

5.10 用户中心

用户中心为用户提供了查看自己的历史订单和修改个人信息的功能



图 31 用户个人信息



图 32 用户历史订单

6 项目总结

1. 学会团队的合作

本次的饱了么的项目的成功完成，离不开本次团队的完美的配合，各个参与者都抱着不做到最好誓不罢休的精神去进行合作开发。

- 1) 本次项目开发使用了 github 作为开发的后台仓库，每一个开发者在完成了对自己仓库中的代码进行了修改之后再传回给 github 进行保存。本次的项目进行了 128 次的提交，共包含五个分支，分别为 master 分支、dev 分支以及三个组员各一支。组员们在各自的分支开发，再合并到 dev 分支，当调试稳定之后最终提交到 master 分支作为正式发布的版本。
- 2) 在团队合作中要进行代码的合并，就要求整个团队进行事先的商量与每个人之间的配合，如何对变量进行命名，如何给一个函数命名，都要遵循一个统一的命名规范，才能更好的进行协作。
- 3) 在团队项目开发中，最重要的不是一个人低头闷声发大财，而是整个项目的人

坐在一起及时的沟通交流，在项目开始时就想好各个类和接口的设计，不会导致后期的牵一发而动全身的尴尬境地。

2. 不断挑战自己的极限

本次的饱了么线上点餐网站的设计目的是为了给饱了么项目组开发在线产品，拓宽其宣传面提高服务的水平与效率，遇到了如下的挑战：

- 1) 在界面的选择与设计的方面，为了饱了么的“简洁，高效，美观”的理念，本次的项目有很大一部分的精力放在了对于界面的设计上面，在页面设计上面力求简单美观大方，采用了行业中流行的 **Bootstrap** 前段开发框架。
- 2) 因为本次的项目有着比较具体的客户需求，本项目组在开发的过程中充分考虑了用户的心理和习惯，力求网站的交互性强，用户体验感好。
- 3) 考虑到日后上线的因素，本项目组力求做出的所有页面里不出现逻辑错误，并提高了系统的容错率，保证不会因为用户的操作而导致网页的崩溃。
- 4) 网站元素的复用，将数个重复使用的元素，比如 **HEADER** 和 **FOOTER** 变成复用的元素，减少重复代码，并使得细节修改更加简洁。
- 5) 添加了搜索功能，遇到了在页面间传递中文元素导致乱码的问题，最后查阅相关文档解决了问题。

3. 仍未实现的问题

作为一个准备上线的网站，本项目组认为饱了么订餐网站仍存在着如下的技术问题尚未解决：

- 1) 支付问题。目前的项目只能实现生成订单为止，并不能实现用户支付的功能。
- 2) 推荐问题。在用户搜索未出现结果时，本项目组希望实现根据用户喜好推荐菜品的功能

4. 发布

本项目最终发布于 [Github](#)（点击后可跳转）。