

天津大学

软件工程综合实践

实践报告



学	院	智能与计算学部
专	业	软件工程
年	级	2020级
姓	名	梁益铭、郑志嘉、周圣喻
学	号	3020244108、3020244109、3020244111

2022 年 10 月 11 日

目 录

第一章	JDBC项目	1
1.1	项目需求	1
1.2	项目设计	7
1.3	功能优化	8
第二章	前端项目	10
2.1	项目需求	10
2.2	项目设计	12
2.3	功能优化	13
第三章	Servlet 项目	15
3.1	项目需求	15
3.2	项目设计	18
第四章	SpringBoot项目	26
4.1	项目需求	26
4.2	项目设计	26
第五章	功能创新	30
5.1	会员功能	30
5.2	“我的” 页面	32
5.3	评论功能	33
第六章	测试方法	36
6.1	后端接口测试	36
6.2	集成测试	36

第七章	开发流程	37
7.1	小组分工	37
7.2	代码仓库	37
7.3	开发过程中的问题及解决方法	39
第八章	实践心得	41

第一章 JDBC项目

1.1 项目需求

本项目要求使用javaSE+JDBC+MySql技术开发基于控制台的C/S结构应用程序，完成本阶段是为了掌握java与数据库的综合应用，为后续阶段打下基础。本阶段的主要目的是实现服务器代码与数据库的连接，完成一个简单的后端程序。

1.1.1 管理员入口

1.管理员登录，出现主菜单

```

-----
|                                     |
|                               饿了么后台管理系统                               |
|                                     |
|-----|
请输入管理员名称：
王磊
请输入密码：
123

===== 1.所有商家列表=2.搜索商家=3.新建商家=4.删除商家=5.退出系统 =====
请输入你的选择：

```

图 1-1 管理员登录

2.菜单项一：所有商家列表

```

===== 1.所有商家列表=2.搜索商家=3.新建商家=4.删除商家=5.退出系统 =====
请输入你的选择：
1

```

商家编号	商家名称	商家地址	商家介绍	起送费	配送费				
10001	万家饺子（软件园E18店）	沈阳市浑南区软件园E18楼1-121-3号				各种饺子	18.0	3.5	
10002	小锅饭豆腐馆（全运店）	沈阳市全运路126号	小锅套餐	15.0	3.0				
10003	麦当劳麦乐送（全运路店）	沈阳市全运路53号麦当劳	汉堡薯条	25.0	4.0				
10004	米村拌饭（浑南店）	沈阳市浑南区彩霞街26号	拌饭套餐	18.0	3.0				
10005	申记串道（中海康城店）	沈阳市中海康城社区48号	烤串炸串	20.0	4.0				
10006	半亩良田排骨米饭	沈阳市和平区三好街126号	排骨米饭	22.0	4.0				
10007	茶兮鲜果饮品（国际软件园店）	沈阳市和平区南京南街345号	甜品饮品	20.0	3.0				

图 1-2 商家列表

3.菜单项二：搜索商家

```

===== 1.所有商家列表=2.搜索商家=3.新建商家=4.删除商家=5.退出系统 =====
请输入你的选择：
2
是否需要输入商家名称关键词(y/n)：
y
请输入商家名称关键词：
饭
是否需要输入商家地址关键词(y/n)：
n
商家编号  商家名称  商家地址  商家介绍  起送费  配送费
10002     小锅饭豆腐馆（全运店）  沈阳市全运路126号  小锅套餐  15.0    3.0
10004     米村拌饭（浑南店）  沈阳市浑南区彩霞街26号  拌饭套餐  18.0    3.0
10006     半亩良田排骨米饭  沈阳市和平区三好街126号  排骨米饭  22.0    4.0

```

图 1-3 搜索商家

4.菜单项三：新建商家

```

===== 1.所有商家列表=2.搜索商家=3.新建商家=4.删除商家=5.退出系统 =====
请输入你的选择：
3
请输入商家名称：
米村拌饭（浑南店）
新建商家成功！商家编号为：10008

```

图 1-4 新建商家

5.菜单项四：删除商家

```

===== 1.所有商家列表=2.搜索商家=3.新建商家=4.删除商家=5.退出系统 =====
请输入你的选择：
4
请输入商家编号：
10008
确认要删除吗(y/n)：
y
删除商家成功！

```

图 1-5 删除商家

1.1.2 商家入口

1.商家登录，出现主菜单

```
-----
|                                     |
|                               饿了么后台管理系统                               |
|                                     |
|-----|
请输入商家编号：
10001
请输入密码：
123

===== 一级菜单（商家管理）1.查看商家信息=2.修改商家信息=3.更新密码=4.所属商品管理=5.退出系统=====
请输入你的选择：
```

图 1-6 商家登录

2.菜单项一：查看商家信息

```
===== 一级菜单（商家管理）1.查看商家信息=2.修改商家信息=3.更新密码=4.所属商品管理=5.退出系统=====
请输入你的选择：
1
|
商家编号：10001
商家名称：万家饺子（软件园E18店）
商家地址：沈阳市浑南区软件园E18楼1-121-3号
商家介绍：各种饺子
起送费：18.0
配送费：3.5
```

图 1-7 查看商家信息

3.菜单项二：修改商家信息

```

===== 一级菜单（商家管理）1.查看商家信息=2.修改商家信息=3.更新密码=4.所属商品管理=5.退出系统=====
请输入你的选择：
2

商家编号：10001
商家名称：万家饺子（软件园E18店）
商家地址：沈阳市浑南区软件园E18楼1-121-3号
商家介绍：各种饺子
起送费：18.0
配送费：3.5
是否修改商家名称(y/n)：
y
请输入新的商家名称：
万家饺子（软件园E19店）
是否修改商家地址(y/n)：
n
是否修改商家介绍(y/n)：
n
是否修改起送费(y/n)：
y
请输入新的起送费：
20
是否修改配送费(y/n)：
n

修改商家信息成功！
    
```

图 1-8 修改商家信息

4.菜单项三：更新密码

```

===== 一级菜单（商家管理）1.查看商家信息=2.修改商家信息=3.更新密码=4.所属商品管理=5.退出系统=====
请输入你的选择：
3

请输入旧密码：
123

请输入新密码：
234

请再次输入新密码：
234
|
修改密码成功！
    
```

图 1-9 更新密码

5.菜单项四：所属商品管理（进入二级菜单，进行商品的增删改查管理）

```

===== 一级菜单（商家管理）1.查看商家信息=2.修改商家信息=3.更新密码=4.所属商品管理=5.退出系统=====
请输入你的选择：
4

===== 二级菜单（食品管理）1.查看食品列表=2.新增食品=3.修改食品=4.删除食品=5.返回一级菜单 =====
请输入你的选择：
    
```

图 1-10 更新密码

1.1.3 商家二级菜单管理

1.二级菜单项一：查看食品列表

```

===== 二级菜单（食品管理）1.查看食品列表=2.新增食品=3.修改食品=4.删除食品=5.返回一级菜单 =====
请输入你的选择：
1
|食品编号  食品名称  食品介绍  食品价格
1          纯肉鲜肉（水饺）  纯肉馅饺子 16.0
2          玉米鲜肉（水饺）  玉米清香  18.0
3          虾仁三鲜（蒸饺）  三鲜馅饺子 22.0
4          素三鲜（蒸饺）  韭菜鸡蛋虾皮 15.0
5          角瓜鸡蛋（蒸饺）  角瓜鸡蛋  16.0
    
```

图 1-11 查看食品列表

2.二级菜单项二：新增食品

```

===== 二级菜单（食品管理）1.查看食品列表=2.新增食品=3.修改食品=4.删除食品=5.返回一级菜单 =====
请输入你的选择：
2
请输入食品名称：
牛肉饺子
请输入食品介绍：
牛肉饺子很好吃
请输入食品价格：
20
    
```

图 1-12 新增食品

3.二级菜单项三：修改食品


```

===== 二级菜单（食品管理）1.查看食品列表=2.新增食品=3.修改食品=4.删除食品=5.返回一级菜单 =====
请输入你的选择：
3
食品编号  食品名称  食品介绍  食品价格
1          纯肉鲜肉（水饺）  纯肉馅饺子 16.0
2          玉米鲜肉（水饺）  玉米清香  18.0
3          虾仁三鲜（蒸饺）  三鲜馅饺子 22.0
4          素三鲜（蒸饺）   韭菜鸡蛋虾皮      15.0
5          角瓜鸡蛋（蒸饺）  角瓜鸡蛋  16.0
8          牛肉饺子  牛肉饺子很好吃    20.0
请选择要更新食品编号：
8

食品编号：8
食品名称：牛肉饺子
食品介绍：牛肉饺子很好吃
食品价格：20.0
所属商家：10001
是否更新食品名称(y/n)：
y
请输入新的食品名称：
纯牛肉饺子
是否更新食品介绍(y/n)：
n
是否更新食品价格(y/n)：
n

修改食品成功！

```

图 1-13 修改食品

4.二级菜单项四：删除食品

```

===== 二级菜单（食品管理）1.查看食品列表=2.新增食品=3.修改食品=4.删除食品=5.返回一级菜单 =====
请输入你的选择：
4
食品编号  食品名称  食品介绍  食品价格
1          纯肉鲜肉（水饺）  纯肉馅饺子 16.0
2          玉米鲜肉（水饺）  玉米清香  18.0
3          虾仁三鲜（蒸饺）  三鲜馅饺子 22.0
4          素三鲜（蒸饺）   韭菜鸡蛋虾皮      15.0
5          角瓜鸡蛋（蒸饺）  角瓜鸡蛋  16.0
8          纯牛肉饺子  牛肉饺子很好吃    20.0
请选择要删除的食品编号：
8
确认要删除吗(y/n)：
y

删除食品成功！


```


图 1-14 删除食品

1.2 项目设计

1.2.1 数据库设计

本项目使用MySQL数据库，选择DataGrip作为开发工具。在数据库的设计上，需要创建管理员、食品和商家共三张表。

business		
 businessId	/* 商家编号 */	int
 password	/* 商家密码 */	varchar(20)
 businessName	/* 商家名称 */	varchar(40)
 businessAddress	/* 商家地址 */	varchar(50)
 businessExplain	/* 商家介绍 */	varchar(40)
 deliveryPrice	/* 配送费 */	decimal(5,2)
 startPrice	/* 起送费 */	decimal(5,2)

food		
 foodId	/* 食品编号 */	int
 foodName	/* 食品名称 */	varchar(30)
 foodExplain	/* 食品介绍 */	varchar(30)
 foodPrice	/* 食品价格 */	decimal(5,2)
 businessId	/* 所属商家编号 */	int

admin		
 adminId	/* 管理员编号 */	int
 adminName	/* 管理员名称 */	varchar(20)
 password	/* 密码 */	varchar(20)

图 1-15 JDBC项目数据表

1.2.2 后端设计

一. 开发环境

1. 开发工具：IDEA、DataGrip
2. 检查IDEA的jdk配置：jdk8
3. 检查IDEA的文件编码配置：utf-8
4. 检查MySQL配置：MySQL 8.0.30

二. 项目结构

本项目主要结构由上到下分为Entry层、view层、dao层：Entry即入口，包括商家和管理员两个入口，在入口接收到请求后调用view层，view层主要实现了一些对请求的进一步处理（例如询问是否按照地址查询），当请求明确之后，由view层调用dao层，dao层则是负责与数据库连接，对数据库进行操作。

除此之外的po文件夹里主要存放业务流中所需的持久对象，与数据表中的属性相对应，而util包中则存放一些常用的工具类。

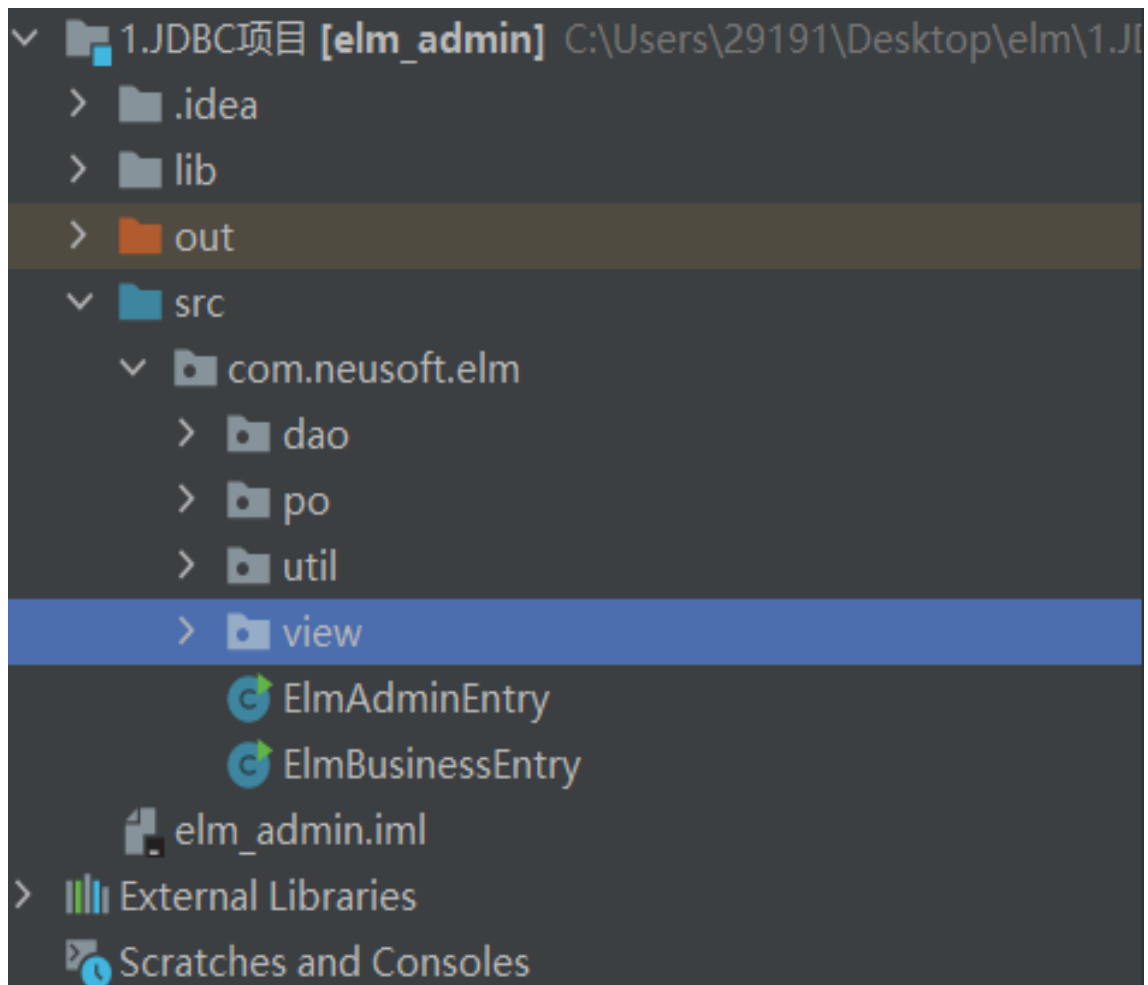


图 1-16 项目结构

1.3 功能优化

在管理员或商家登陆成功后，输入数字选项阶段，如果输入数字以外的类型，程序会发生崩溃。我们针对此问题进行了优化，如果输入数字以外的类型，会提示“请输入正确格式的数字选项”。

```

-----
请输入管理员名称：
admin
请输入密码：
123

----- 1.所有商家列表 2.搜索商家 3.新建商家 4.删除商家 5.退出系统-----
请输入你的选择
h
请输入正确格式的数字选项！

----- 1.所有商家列表 2.搜索商家 3.新建商家 4.删除商家 5.退出系统-----
请输入你的选择
1
商家编号 商家名称 商家地址 商家介绍 起送费 配送费
10001 必胜客 天津市津南区永旺梦悦城 西餐 20.0 9.0
10002 肯德基宅急送 天津市津南区天津大学北洋园校区 快餐 15.0 9.0
10003 绝味鸭脖 天津市西青区大寺镇 熟食 20.0 7.5
10004 熊家炸鸡 天津市津南区辛庄镇 炸鸡 20.0 0.0
10005 喜姐炸串 天津市津南区永旺梦悦城 炸串 20.0 4.8
10006 1 null null 0.0 0.0

```

图 1-17 项目一优化

第二章 前端项目

2.1 项目需求

前端项目是第二阶段，使用HTML+CSS+JS技术开发企业级静态网页，完成后续项目中前端部分的布局。使用HTML技术完成前端静态网页应用程序的实现，使用CSS技术对网页进行样式渲染，使用JavaScript语言制作网页特效。

2.1.1 首页功能

主要功能

显示点餐分类信息

动作

- 1.点击点餐分类小图片，跳转到商家列表页面
- 2.点击下方菜单栏中的“订单”，跳转到历史订单页面

2.1.2 商家列表页面功能

主要功能

显示商家列表信息

动作

点击某个商家，跳转到此商家的详细信息页面

2.1.3 商家详细信息页面功能

主要功能

显示商家详细信息及所属食品信息

动作

点击“去结算”按钮，跳转到确认订单页面

2.1.4 确认订单页面功能

主要功能

- 1.显示订单信息
- 2.选择送货地址

动作

1. 点击送货地址，跳转到送货地址列表页面
2. 点击“去支付”按钮，跳转到支付页面

2.1.5 在线支付页面功能

主要功能

显示订单信息及订单明细信息

动作

无

2.1.6 登录页面功能

主要功能

用户登录

动作

1. 点击“登陆”按钮，跳转到上一个页面
2. 点击“去注册”按钮，跳转到注册页面

2.1.7 注册页面功能

主要功能

注册新用户

动作

点击“注册”按钮，跳转到登陆页面

2.1.8 历史订单页面功能

主要功能

显示用户历史订单信息

动作

点击某个历史订单，可以对订单明细信息进行显示和隐藏

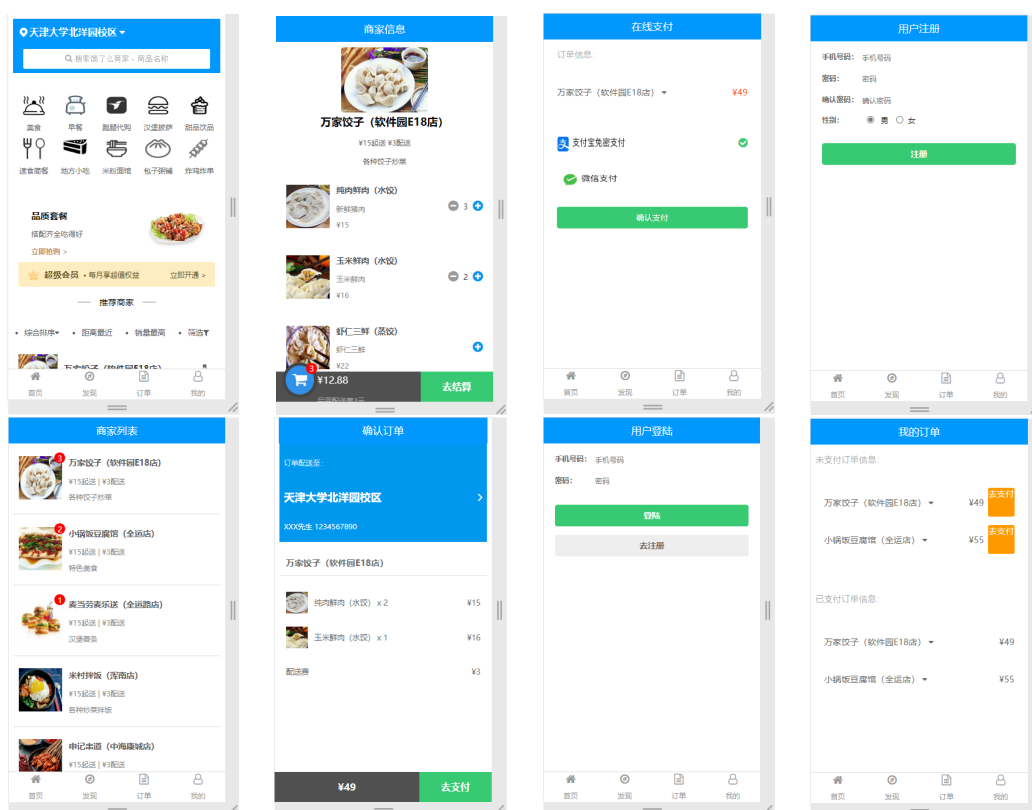


图 2-1 前端页面实现结果

2.2 项目设计

2.2.1 移动端网页设计

问题

移动端屏幕尺寸多变，不像PC端屏幕那样统一。这样就造成在移动端显示网页时，由于尺寸问题，会出现显示不下网页，从而出现横向滚动条现象。为了解决这个问题，出现了viewport（屏幕视口）的概念。

解决方案

使用viewport解决上述问题。先将网页放入layout viewport中，然后在将layout viewport等比例缩小到ideal viewport中。这样就能保证：无论什么样的网页，都能在手机屏幕上显示，而且没有横向滚动条。

2.2.2 弹性布局

概念：主轴和侧轴

弹性布局中的一个重要概念：主轴与侧轴：弹性盒子中默认存在两根轴，一个是水平方向的主轴，一个是垂直方向的侧轴。

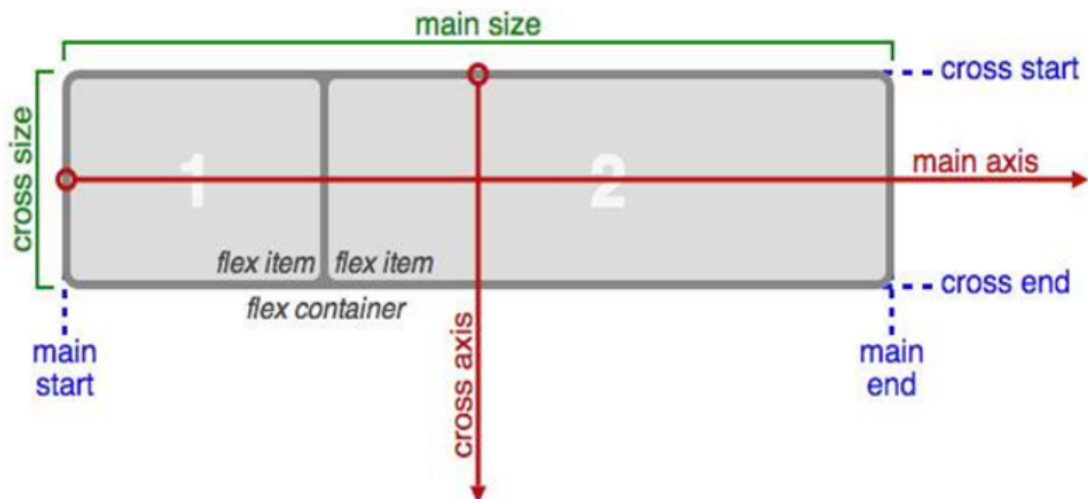


图 2-2 主轴和侧轴

设计方法

- 1.使用flex-direction属性设置主轴方向。
- 2.使用flex-wrap:wrap让子元素自动换行。
- 3.设置align-items样式。常用值有三种：flex-start（上对齐）、flex-end（下对齐）、center（居中）。
- 4.使用flex给予子元素分配空间，从而让每个子元素所占空间不一致。

2.2.3 视口尺寸

视口：在PC端，指浏览器的可视区域；在移动端，指Viewport 中的 Layout Viewport。视口尺寸常用的有以下2个：vw: 1vw 等于视口宽度的1% vh: 1vh 等于视口高度的1% 实际应用中，可以使用vw，实现元素宽度和高度成比例自动缩放。

2.2.4 边框盒子模型

CSS3之前的盒子模型，可以说是content-box型盒子，即宽和高为内容。CSS3之后新增了一的盒子模型，可以说是border-box型盒子，即宽和高为边框。那么，使用box-sizing属性可以设置盒子模型类型。

2.3 功能优化

对于课程提供的前端代码，商家列表底部会被遮挡，我们对此进行了优化：将margin-bottom改为padding-bottom并调整了一些参数完成优化。



图 2-3 项目二优化

第三章 Servlet 项目

3.1 项目需求

本项目要求使用VUE+Servlet+AJAX技术，开发前后端分离的饿了么Web应用程序。主要设计参照“饿了么官网网页版”制作，并且仅关注点餐业务线功能，“饿了么官网”中的其它功能暂不涉及。

本项目需要完成的功能大致分为十一个页面：

3.1.1 首页功能

主要功能

显示点餐分类信息

动作

- 1.点击点餐分类小图片，跳转到商家列表页面
- 2.点击下方菜单栏中的“订单”，跳转到历史订单页面

3.1.2 商家列表页面功能

主要功能

- 1.根据点餐分类显示商家列表信息
- 2.如果处于登录状态，那些需要查询购物车中是否有此商家的食品。如果有，在页面上显示食品数量

动作

点击某个商家，跳转到此商家的详细信息页面

3.1.3 商家详细信息页面功能

主要功能

显示商家详细信息及所属食品信息，并自动计算总价

动作

- 1.点击某食品的 + 按钮，食品数量加 1
- 2.点击某食品的 - 按钮，食品数量减 1
- 3.点击“去结算”按钮，跳转到确认订单页面

4.如果没有登录，那么上述三个动作会自动跳转到登录页面

3.1.4 确认订单页面功能

主要功能

- 1.确认订单信息是否正确
- 2.选择送货地址

动作

- 1.点击送货地址，跳转到送货地址列表页面
- 2.点击“去支付”按钮，跳转到支付页面

3.1.5 在线支付页面功能

主要功能

显示订单信息及订单明细信息

动作

无（点餐业务流程到此结束）

3.1.6 送货地址列表页面功能

主要功能

显示当前用户的送货地址信息

动作

- 1.点击某个送货地址，跳转回确认订单页面
- 2.点击“新增”按钮，跳转到新增送货地址页面
- 3.点击“编辑”按钮，跳转到送货地址编辑页面
- 4.点击“删除”按钮，删除此送货地址

3.1.7 新增送货地址页面功能

主要功能

添加新的送货地址

动作

点击“保存”按钮，添加新的送货地址，并跳转到送货地址列表页面

3.1.8 编辑送货地址页面功能

主要功能

编辑送货地址

动作

点击“更新”按钮，保存编辑后的送货地址，并跳转到送货地址列表页面

3.1.9 登录页面功能

主要功能

用户登录

动作

- 1、点击“登陆”按钮，进行登陆业务处理。如果登陆成功，跳转到上一个页面
- 2、点击“去注册”按钮，跳转到注册页面

3.1.10 注册页面功能

主要功能

注册新用户

动作

点击“注册”按钮，进行新用户注册。注册成功后，跳转到登陆页面

3.1.11 历史订单页面功能

主要功能

显示用户历史订单信息

动作

点击某个历史订单，可以对订单明细信息进行显示和隐藏

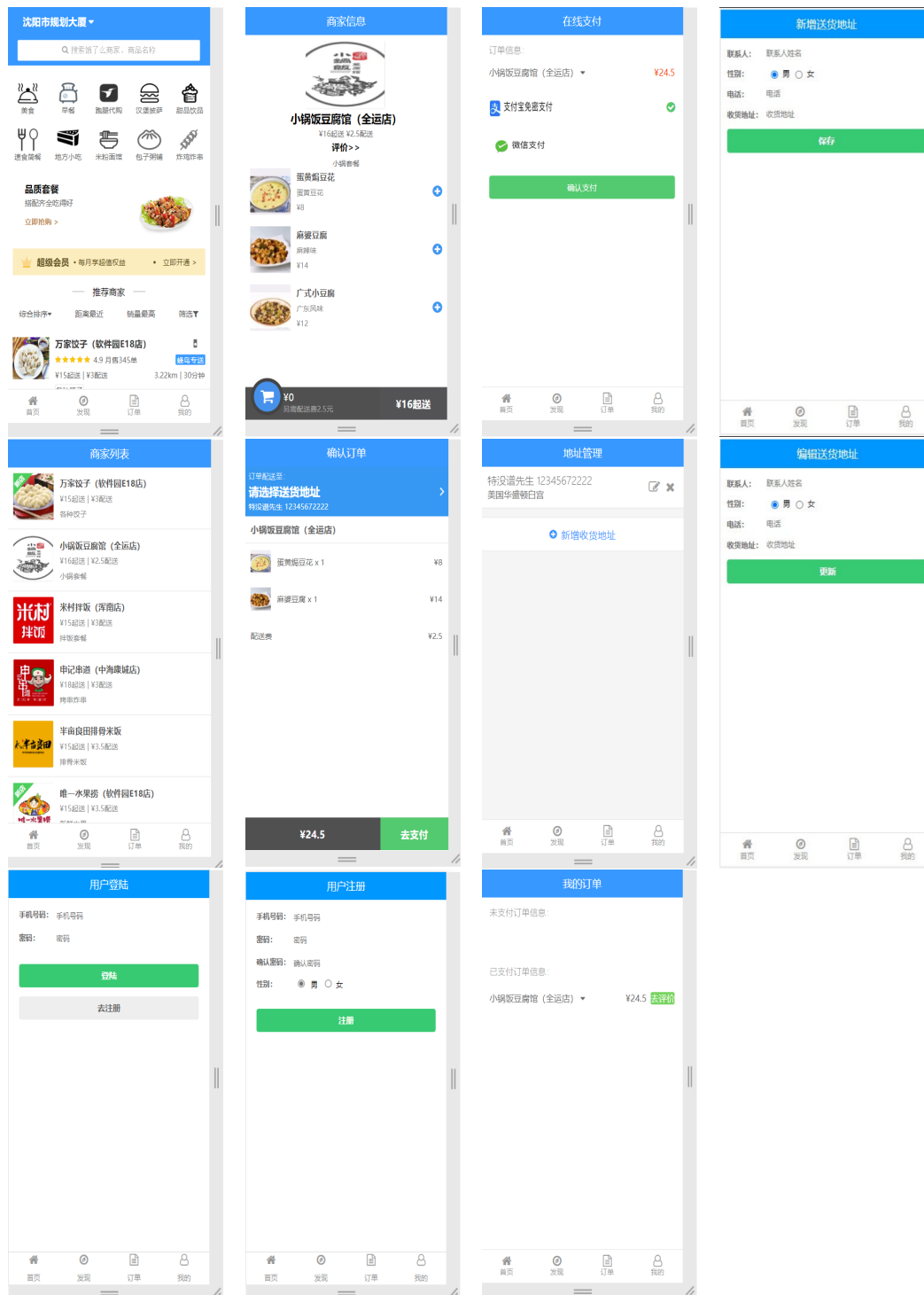


图 3-1 VUE前端页面实现结果

3.2 项目设计

3.2.1 VUE前端设计

首先安装cnpm和VueCli开发环境，使用VSCode工具进行项目开发。

接着搭建VueCli模板工程、添加依赖及配置文件。添加font-

awesome与axios依赖。添加图片到src的assets中。在src目录下添加common.js文件，用于添加一些公用方法。在工程根目录下添加vue.config.js文件，用于配置前端的端口号。添加main.js文件，用于判断每个页面是否需要登录才可以使用。添加App.vue文件，里面设置全局的共通样式。

最后实现各个组件：主页、商家列表、商家详细信息、订单、支付、添加修改用户地址、历史订单、登录、注册和底部导航栏共同组件。

3.2.2 数据库设计

本项目使用MySQL数据库，选择DataGrip作为开发工具。在数据表的设计上，需要创建商家表、食品表、购物车表、送货地址表、订单表、订单明细表、用户表共七张表。

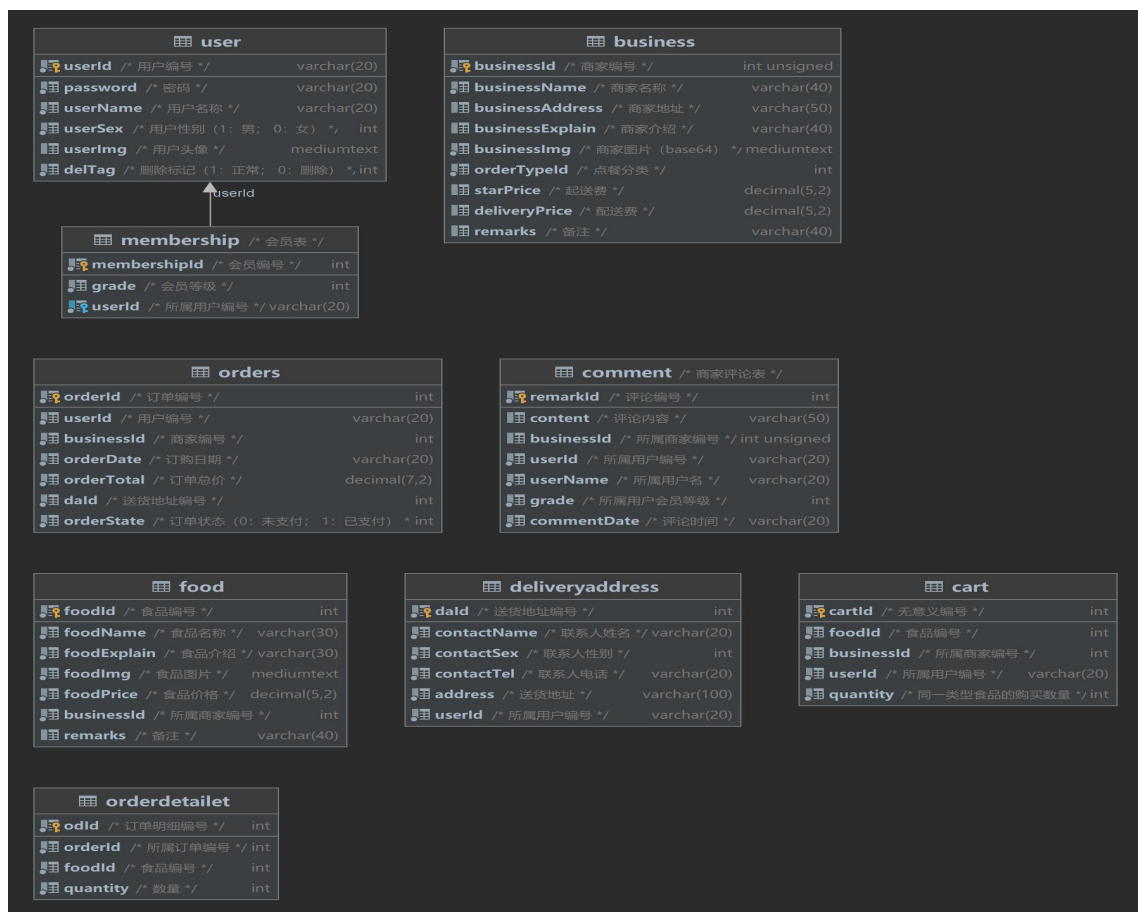


图 3-2 Servlet项目数据表

3.2.3 Servlet后端设计

一. 开发环境

1. 开发工具：IDEA、DataGrip

2. 检查IDEA的jdk配置: jdk8
3. 检查IDEA的tomcat配置: tomcat 9.0.65
4. 检查IDEA的文件编码配置: utf-8
5. 检查MySQL配置: MySQL 8.0.30

二. 开发技术

本项目为JavaWeb项目，在后端使用Servlet技术，Tomcat作为容器进行项目开发。在JavaWeb工程的搭建方面，本项目使用基于Servlet的简易MVC架构，能够很好地解决后端与前端地交互。在响应前端请求方面，使用Jackson将java对象或集合转换为json对象或数组后，返回前端。

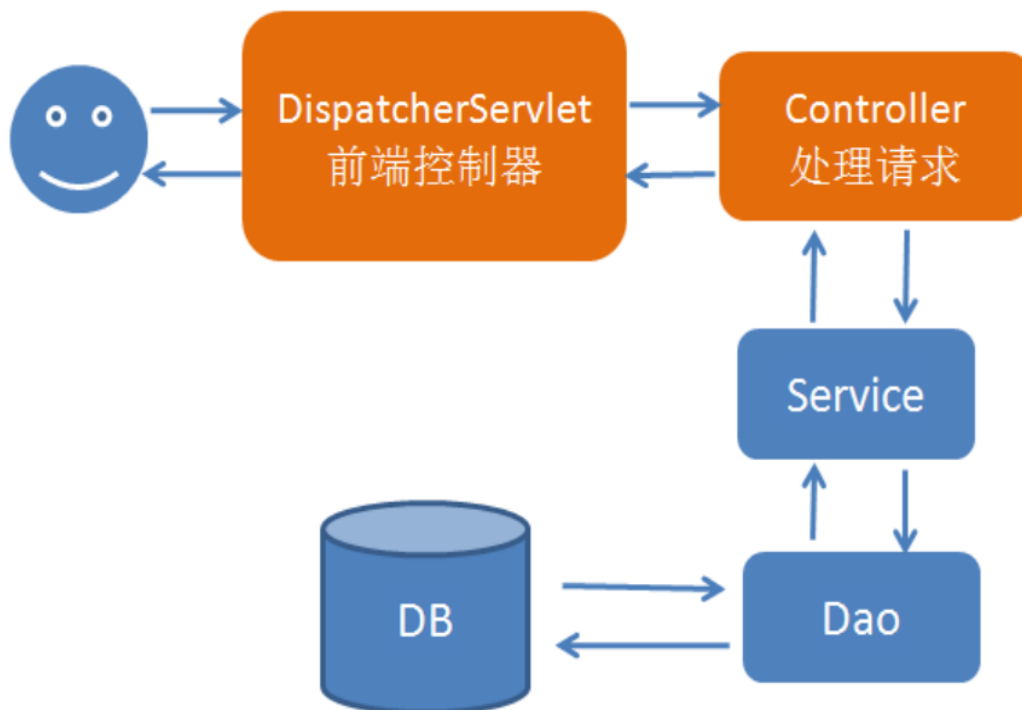


图 3-3 Servlet简易MVC架构

三. 跨域问题解决

本项目使用CORS解决前后端分离开发模式时的跨域问题，它允许浏览器向跨源服务器，发出XMLHttpRequest请求，从而克服了AJAX只能同源使用的限制。实现CORS通信的关键是服务器，只要服务器实现了CORS接口，就可以跨源通信。

四. 事务处理解决

本项目将事务的处理放在Service层：

1.Connection的创建与销毁要放在service层。

2.为了保证在同一次请求处理的线程中， service层和dao层都共用同一个Connection对象，需要将Connection对象放入ThreadLocal中。service层和dao层使用的Connection对象一律从ThreadLocal获取。

3.dao层不再处理异常， dao层产生的异常将直接抛给service层进行处理。

4.dao层 负责 关闭PreparedStatement和ResultSet， service层 负责 关闭Connection。

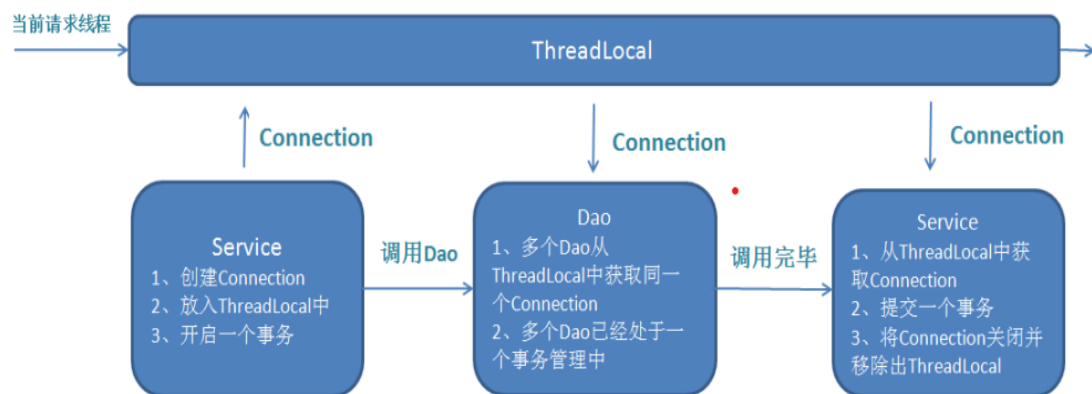


图 3-4 线程池

五. 项目结构

本项目主要结构由上到下分为Controller层、Service层、Dao层三层架构：Controller层主要负责与客户端进行交互，接收并响应前端请求；Service层主要负责处理需要实现的业务逻辑；Dao层主要负责对于数据库的操作。

其他软件包主要存放一些功能性代码， framework包主要搭建DispatcherServlet前端控制器， filter包主要解决跨域问题， po包主要存放业务流中所需的持久对象，与数据表相对应， util包中存放一些常用的工具类。

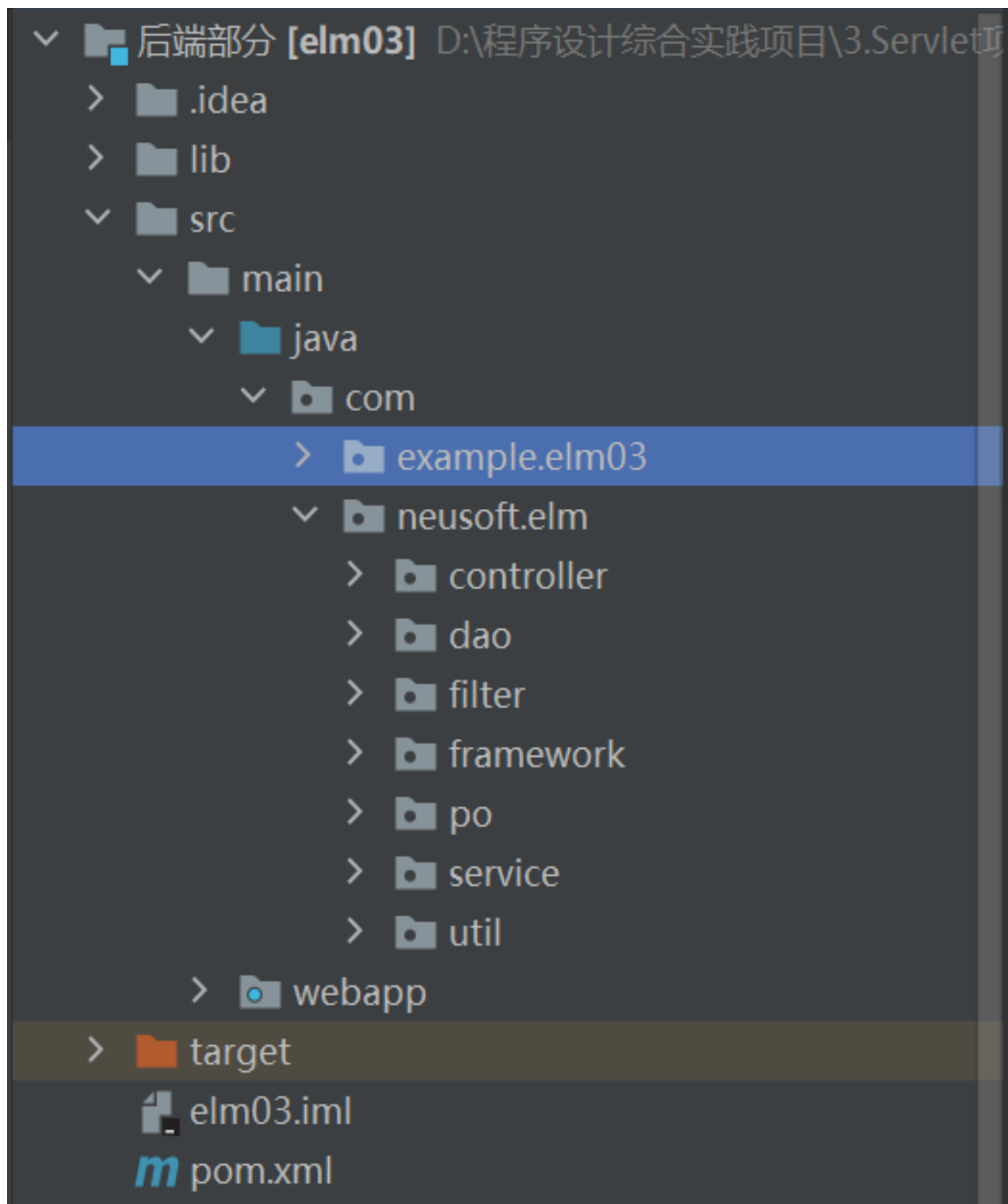


图 3-5 项目结构

六. 服务器接口API

1. Business

1.1 BusinessController/listBusinessByOrderId

参数: orderId

返回值: business数组

功能: 根据点餐分类编号查询所属商家信息

1.2 BusinessController/getBusinessById

参数: businessId

返回值: business对象

功能: 根据商家编号查询商家信息

2. Food

2.1 FoodController/listFoodByBusinessId

参数: businessId

返回值: food数组

功能: 根据商家编号查询所属食品信息

3. Cart

3.1 CartController/listCart 参数: userId、businessId (可选)

返回值: cart数组 (多对一: 所属商家信息、所属食品信息)

功能: 根据用户编号查询此用户所有购物车信息

根据用户编号和商家编号, 查询此用户购物车中某个商家的所有购物车信息

3.2 CartController/saveCart

参数: userId、businessId、foodId

返回值: int (影响的行数)

功能: 向购物车表中添加一条记录

3.3 CartController/updateCart

参数: userId、businessId、foodId、quantity

返回值: int (影响的行数)

功能: 根据用户编号、商家编号、食品编号更新数量

3.4 CartController/removeCart

参数: userId、businessId、foodId (可选)

返回值: int (影响的行数)

功能: 根据用户编号、商家编号、食品编号删除购物车表中的一条食品记录

录

根据用户编号、商家编号删除购物车表中的多条条记录

4. DeliveryAddress

4.1 DeliveryAddressController/listDeliveryAddressByUserId

参数: userId

返回值: deliveryAddress数组

功能: 根据用户编号查询所属送货地址

4.2 DeliveryAddressController/getDeliveryAddressById

参数: daId

返回值: deliveryAddress对象

功能: 根据送货地址编号查询送货地址

4.3 DeliveryAddressController/saveDeliveryAddress

参数: contactName、contactSex、contactTel、address、userId

返回值: int (影响的行数)

功能: 向送货地址表中添加一条记录

4.4 DeliveryAddressController/updateDeliveryAddress

参数: daId、contactName、contactSex、contactTel、address、userId

返回值: int (影响的行数)

功能: 根据送货地址编号更新送货地址信息

4.5 DeliveryAddressController/removeDeliveryAddress

参数: daId

返回值: int (影响的行数)

功能: 根据送货地址编号删除一条记录

5. Orders

5.1 OrdersController/createOrders

参数: userId、businessId、daId、orderTotal

返回值: int (订单编号)

功能: 根据用户编号、商家编号、订单总金额、送货地址编号向订单表中添加一条记录,

并获取自动生成的订单编号,

然后根据用户编号、商家编号从购物车表中查询所有数据, 批量添加到订单明细表中,

然后根据用户编号、商家编号删除购物车表中的数据。

5.2 OrdersController/getOrdersById

参数: orderId

返回值: orders对象 (包括多对一: 商家信息; 一对多: 订单明细信息)

功能: 根据订单编号查询订单信息, 包括所属商家信息, 和此订单的所有订单明细信息

5.3 OrdersController/listOrdersByUserId

参数: userId

返回值: orders数组 (包括多对一: 商家信息; 一对多: 订单明细信息)

功能：根据用户编号查询此用户的所有订单信息

6. User

6.1 UserController/getUserByIdByPass

参数：userId、password

返回值：user对象

功能：根据用户编号与密码查询用户信息

6.2 UserController/getUserById

参数：userId

返回值：int（返回行数）

功能：根据用户编号查询用户表返回的行数

6.3 UserController/saveUser

参数：userId、password、userName、userSex

返回值：int（影响的行数）

功能：向用户表中添加一条记录

七. 业务流程

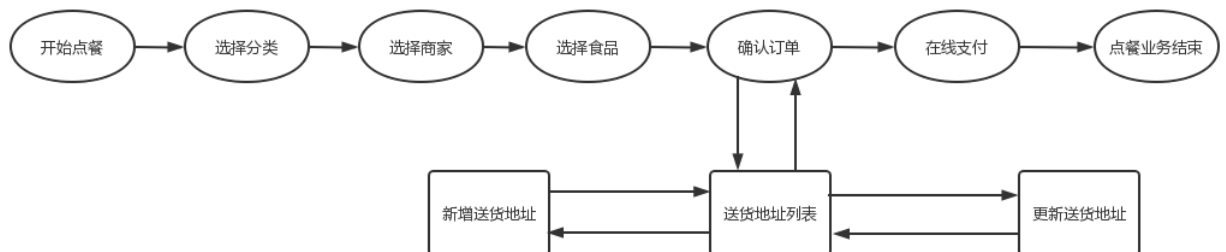


图 3-6 业务流程图

第四章 SpringBoot项目

4.1 项目需求

本项目主要使用VUE+SpringBoot+AJAX技术，开发前后端分离的Web应用程序。由于本项目中的前端VUE代码和数据库结构与第三阶段Servlet项目一致，且项目需求与第三阶段一致，只是将服务器端代码由Servlet升级为Springboot，所以该项目仅介绍服务器端。

4.2 项目设计

4.2.1 SpringBoot后端设计

一. 开发环境

1. 开发工具：IDEA、DataGrip
2. 检查IDEA的jdk配置：jdk8
3. 检查SpringBoot配置：SpringBoot 2.7.3
4. 检查maven构建工具的配置：maven3
5. 检查IDEA的文件编码配置：utf-8
6. 检查MySQL配置：MySQL 8.0.30

二. 开发技术

本项目为JavaWeb项目，在后端使用Springboot技术，使用注解开发，基本不使用xml配置文件，在运行时，直接运行application中的main方法即可转到web页面，输入项目路径访问即可，而不是通过发布项目，启动tomcat，debug运行等一系列操作。

三. 项目结构

本项目主要结构由上到下分为Controller层、Service层、Mapper层三层架构：Controller层主要负责与客户端进行交互，接收并响应前端请求，并将请求返回到Service层，Service层主要负责处理需要实现的业务逻辑，最后Service层调用Mapper层，Mapper层主要负责编写Mysql语句对数据库进行操作。

其他的包主要存放一些功能性代码，po包主要存放业务流中所需的持久对象，与数据表中的属性相对应，util包中主要存放一些常用的工具类。

四. 服务器接口API

1.Business

1. 1BusinessController/listBusinessByOrderTypeId

参数: orderId

返回值: business数组

功能: 根据点餐分类编号查询所属商家信息

1.2.BusinessController/getBusinessById

参数: businessId

返回值: business对象

功能: 根据商家编号查询商家信息

2.Food 2.1 FoodController/listFoodByBusinessId

参数: businessId

返回值: food数组

功能: 根据商家编号查询所属食品信息

3.Cart

3.1 CartController/listCart

参数: userId、businessId (可选)

返回值: cart数组 (多对一: 所属商家信息、所属食品信息)

功能: 根据用户编号查询此用户所有购物车信息根据用户编号和商家编号, 查询此用户购物车中某个商家的所有购物车信息

3.2 CartController/saveCart

参数: userId、businessId、foodId

返回值: int (影响的行数)

功能: 向购物车表中添加一条记录

3.3 CartController/updateCart

参数: userId、businessId、foodId、quantity

返回值: int (影响的行数)

功能: 根据用户编号、商家编号、食品编号更新数量

3.4 CartController/removeCart

参数: userId、businessId、foodId (可选)

返回值: int (影响的行数)

功能: 根据用户编号、商家编号、食品编号删除购物车表中的一条食品记录

根据用户编号、商家编号删除购物车表中的多条条记录

4.DeliveryAddress

4.1 DeliveryAddressController/ listDeliveryAddressByUserId

参数: userId

返回值: deliveryAddress数组

功能: 根据用户编号查询所属送货地址

4.2 DeliveryAddressController/getDeliveryAddressById

参数: daId

返回值: deliveryAddress对象

功能: 根据送货地址编号查询送货地址

4.3 DeliveryAddressController/saveDeliveryAddress

参数: contactName、contactSex、contactTel、address、userId

返回值: int (影响的行数)

功能: 向送货地址表中添加一条记录

4.4 DeliveryAddressController/updateDeliveryAddress

参数: daId、contactName、contactSex、contactTel、address、userId

返回值: int (影响的行数)

功能: 根据送货地址编号更新送货地址信息

4.5 DeliveryAddressController/removeDeliveryAddress

参数: daId

返回值: int (影响的行数)

功能: 根据送货地址编号删除一条记录

5.Orders

5.1 OrdersController/createOrders

参数: userId、businessId、daId、orderTotal

返回值: int (订单编号)

功能: 根据用户编号、商家编号、订单总金额、送货地址编号向订单表中添加一条记录, 并获取自动生成的订单编号,

然后根据用户编号、商家编号从购物车表中查询所有数据, 批量添加到订单明细表中,

然后根据用户编号、商家编号删除购物车表中的数据。

5.2 OrdersController/getOrdersById

参数: orderId

返回值: orders对象 (包括多对一: 商家信息; 一对多: 订单明细信息)

功能: 根据订单编号查询订单信息, 包括所属商家信息, 和此订单的所有订单明细信息

5.3 OrdersController/listOrdersByUserId

参数: `userId` 返回值: `orders`数组 (包括多对一: 商家信息; 一对多: 订单明细信息)

功能: 根据用户编号查询此用户的所有订单信息

6.User

6.1 UserController/getUserByIdByPass

参数: `userId`、`password`

返回值: `user`对象

功能: 根据用户编号与密码查询用户信息

6.2 UserController/getUserById

参数: `userId`

返回值: `int` (返回行数)

功能: 根据用户编号查询用户表返回的行数

6.3 UserController/saveUser

参数: `userId`、`password`、`userName`、`userSex`

返回值: `int` (影响的行数)

功能: 向用户表中添加一条记录

五. 业务流程

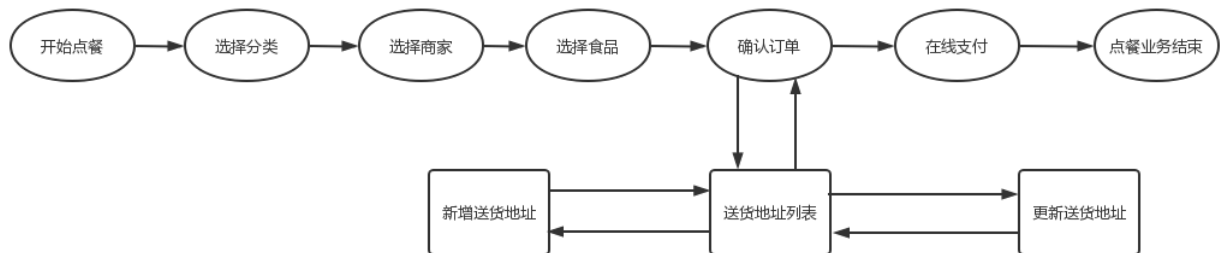


图 4-1 业务流程图

第五章 功能创新

在第四阶段完成后，我们意识到该项目还有许多值得改进和优化的地方，因此我们基于第四阶段的SpringBoot项目添加了一些新功能。

5.1 会员功能

5.1.1 功能需求

在首页，我们发现有一个注册会员入口，而这个功能在课程中并未实现，于是我们首先选择添加一个会员功能。用户可以选择注册成为不同等级的会员从而享受到不同程度的折扣。用户在支付订单页面可以查看打折后的价格，点击支付以支付优惠价。

5.1.2 功能设计

用户在首页可以点击“立即开通”可以进入会员注册界面，有三种会员可以选择，分别是白银会员、黄金会员和钻石会员。每种会员的价格不同，折扣力度也不同。点击“去注册”可以发送注册请求。注册请求分为3种情况：

- 1.如果用户不是会员，则会注册为选择的会员；
- 2.如果用户已经是会员且选择的会员比原本的会员等级高，则会升级会员；
- 3.如果用户已经是会员但选择的会员等级低于原本的会员等级，则会注册失败并返回错误信息。

5.1.3 数据库设计

为了实现会员功能，需要在数据库中新增一张会员表，具体属性和与其他表的关联关系如图所示：



图 5-1 会员表

5.1.3.1 接口文档

1.Membership

1.1 MembershipController/getMembershipById(post方法)

参数: user

返回值: grade会员等级

功能: 查询当前用户的会员等级

1.2 MembershipController/saveMembership(post方法)

参数: userId、grade

返回值: 数据库改变的行数

功能: 为一个用户注册会员

1.3 MembershipController/updateMembership(post方法)

参数：userId、grade

返回值：数据库改变的行数

功能：为一个用户升级会员

5.1.4 实现结果



图 5-2 会员功能实现结果

5.2 “我的” 页面

5.2.1 功能需求

在网页下方的导航栏有一个“我的”按钮，在原来的实现中点击这个按钮并不会跳转到任何页面，于是我们决定实现一个“我的”页面。

5.2.2 功能设计

“我的”页面最上面会显示用户姓名、ID和会员等级，若未注册会员则默认为普通会员。在用户信息的下面，添加“我的地址”和“注册会员”按钮，分别可以路由到编辑地址和注册会员页面。

5.2.3 接口文档

2.MyProfile

2.1 MembershipController/getMembershipById(post方法)

参数: user

返回值: grade会员等级

功能: 查询当前用户的会员等级

5.2.4 实现结果

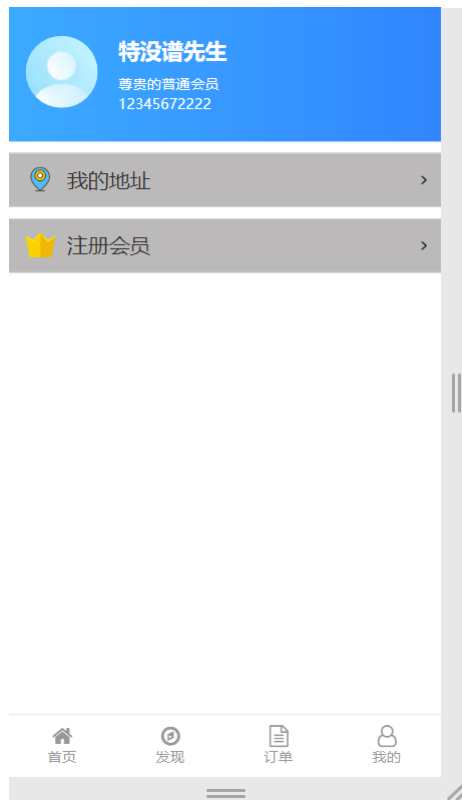


图 5-3 “我的” 页面实现结果

5.3 评论功能

5.3.1 功能需求

在日常使用外卖APP时，我们经常会查看用户对商家的评价，于是我们打算实现一个商家评论功能方便用户查看一个商家的评价

5.3.2 功能设计

用户在支付了一个订单之后，在订单页面可以点击“去评价”，用户进入

评价页面可以编写评价，点击发布即可成功发布。点击商家信息界面中的“评价”按钮可以进入商家评价页面查看自己和其他人的评价，每条评价会显示评价人姓名、会员等级、评价时间和评价内容。

5.3.3 数据库设计

为了实现评论功能，需要在数据库中添加一张评论表，具体属性和其他表的关联关系如图所示：

comment /* 商家评论表 */		
remarkId	/* 评论编号 */	int
content	/* 评论内容 */	varchar(50)
businessId	/* 所属商家编号 */	int unsigned
userId	/* 所属用户编号 */	varchar(20)
userName	/* 所属用户名 */	varchar(20)
grade	/* 所属用户会员等级 */	int
commentDate	/* 评论时间 */	varchar(20)

图 5-4 “我的” 页面实现结果

5.3.4 接口文档

3.Comment

3.1 CommentController/listCommentByBusinessId

参数：businessId

返回值：commentArr

功能：得到评论数据库的所有内容

3.2 CommentController/saveComment

参数：userId、userName、businessId、content、grade

返回值：数据库改变的行数

功能：保存一个评论

5.3.5 实现结果



图 5-5 评论员功能实现结果

第六章 测试方法

6.1 后端接口测试

本项目选择Postman作为测试工具，模拟前端的请求来进行后端接口的测试。下面的例子为对于新开发的评论功能中，由商家编号列出所有对于此商家评价信息接口的测试。

选择Post方法，请求时的参数采用x-www-form-urlencoded选项，对应信息头-application/x-www-form-urlencoded，会将表单内的数据转换为键值对。可以看到Http状态码为200OK，说明成功响应，最终后端响应的结果是一个包含所需信息的json对象。

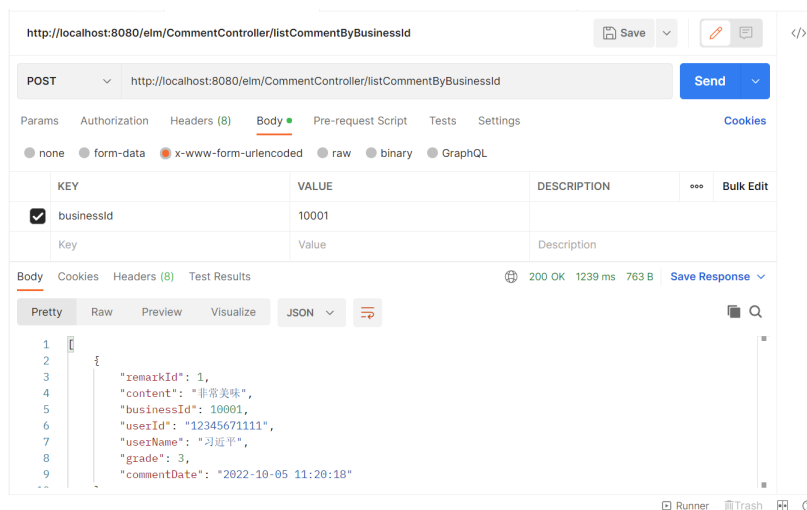


图 6-1 提交记录

6.2 集成测试

小组约定，当前后端分别完成本阶段的代码编写之后，各自将代码推到github上，再统一拉取做集成测试，在集成测试中，前后端合作debug，对于出现的bug先判断是前端还是后端的差错，再由对应成员做调试修改。

第七章 开发流程

7.1 小组分工

小组开发采用严格的前后端分离开发模式，前后端分别由不同的组员进行编写并分别测试，每当一个阶段整体完成之后，再做统一的集成测试。

下面是组内成员的具体分工：

周圣喻

主要负责前端部分。负责饿了么前端界面、VUE前端代码编写以及新功能前端的开发。

梁益铭

主要负责后端部分和项目测试。负责JDBC项目，Servlet和Springboot项目的编写、前后端连接测试以及新功能后端的开发。

郑志嘉

主要负责后端部分和数据库连接。负责JDBC项目，Servlet和Springboot项目的编写以及数据库连接管理。

7.2 代码仓库

小组使用github进行远程仓库管理，再结合git实现项目的合作编写，每个人完成自己的部分后，首先将远程仓库的代码拉取到本地仓库，解决冲突后再将修改后的代码推送到github上，完成一整个阶段后再统一做集成测试。

仓库分为两个分支，main分支存储的是课程项目的部分，advance分支存储的是新功能开放的过程，整个项目小组总共提交了上百次，下面是github上的一些统计数据：

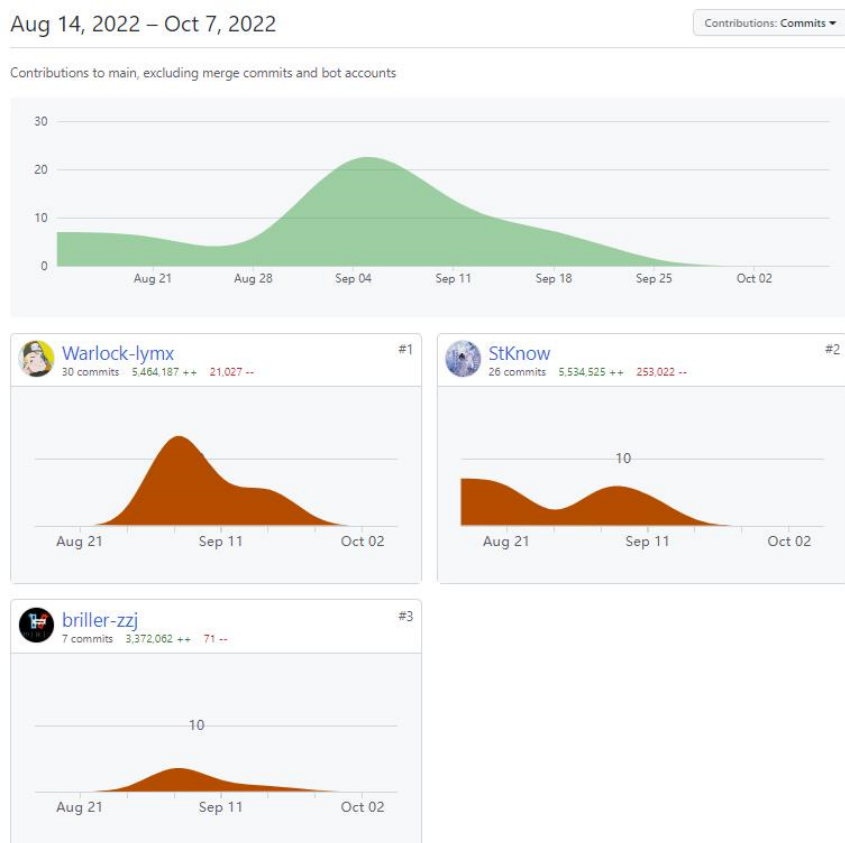


图 7-1 提交记录

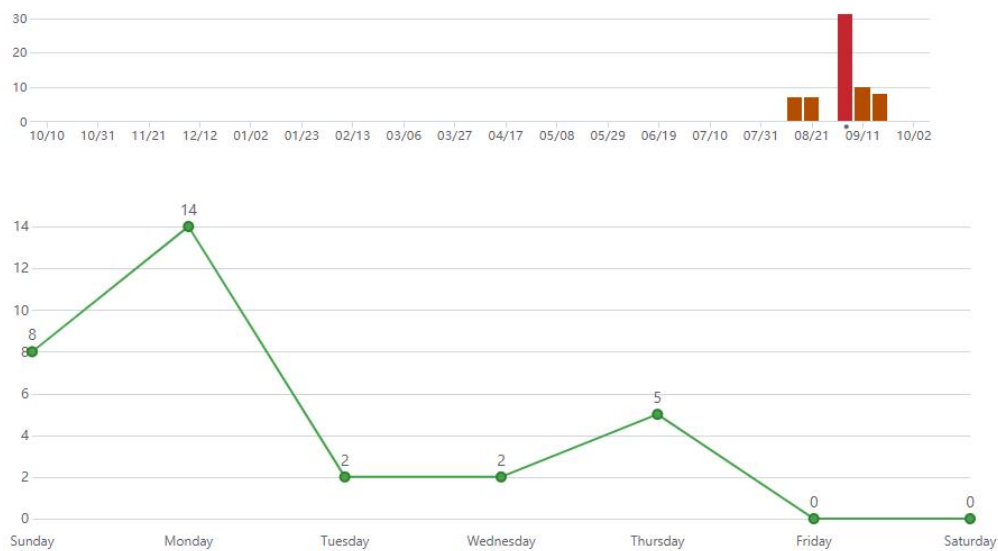


图 7-2 提交统计

图 7-3 main分支提交记录图 7-4 advance分支提交记录

7.3 开发过程中的问题及解决方法

解决方法：通过错误提示发现此版本依赖库中路径少一个.cj。

解决方法：通过CSS中的margin属性指定外边距，从而使页面显示达到预想的效果。

问题：VUE框架搭建出错，版本号不对应，导致按照课件编写的网页无法正常显示页面。

解决方法：卸载VUE，重新按照视频安装对应的版本。

问题: router路由重定向出错, 点击“去注册”按钮跳转到登录页面。

解决方法：经检查，在main.js中判断是否登录的条件出错，筛选“注册”页面未成功。将筛选条件修改正确解决问题。

问题：第四部分后端项目中sql语句编写错误。

解决方法：通过postman测试发现接口出现错误，通过后端报错解决问题。

问题：添加会员功能之后，在下订单部分下方价格部分小数点位数显示出现问题。

解决方法：通过Number(value).toFixed(2)方法限制显示的小数点位数。

第八章 实践心得

郑志嘉

在本次实践中，小组三个人几乎完整地完成了是整个项目的编写，采用前后端分离的开发模式也更加符合当下真实开发的情境，这次实践项目做下来让我收获颇丰，最重要的就是对于实际开发流程有了深刻的了解。除此之外，由于此次开发我担任的是后端开发以及数据库管理，这让我对JDBC有了更好的掌握，也学习到了Servlet和Springboot框架的开发流程。另外，我对于前端的开发也有了一定的了解，对前后端结合的方式也更有把握，这对我以后的开发工作有非常重要的帮助。总而言之，本次实践经历非常有意义，不论是对我的团队协作能力还是开发水平都有显著的提升。

周圣喻

在本次综合实践中，我主要负责的是前端部分的编写。我之前几乎没有系统地学习过前端相关的知识，这对我来说是一个非常大的挑战。我在实践过程中跟随着课程讲解边学边做，获得了非常大的收获，现在已经可以独立编写一个网页的前端部分。除此之外，这也是我第一次以团队合作的方式做一个项目。在团队一起完成项目时，小组成员之前相互讨论，相互进步，我也体验到团队合作的方式，了解到如何多人同时进行一个项目。总的来说，我在本次实践中收获巨大，进步许多，代码能力和团队合作能力有了显著提升。

梁益铭

在本次实践中，我主要负责后端的开发。由于之前没有接触过前后端分离项目的开发，在一步步学习完成整个项目后非常有成就感，对于软件开发有了一定程度的理解，为以后的学习工作打下了坚实的基础。由浅到深，由JDBC项目到Servlet项目，最后再到SpringBoot项目，一点点的深入学习使我对整个开发框架有了十分详细的了解。在后续新功能的开发中，进一步将所学的知识加以实践，与前端队友合作完成接口的制定，进行前后端分离的开发过程，最终成功将自己所想的功能一一实现，受益匪浅。