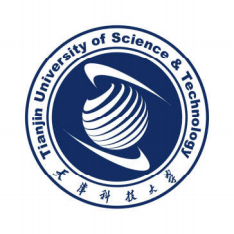


毕业设计说明书

(2023届·本科)



姓 名：

学 号：

指导教师：

**基于SpringBoot与Vue的校园订餐系统**

**Campus Meal Ordering System Based On SpringBoot and Vue**

**专 业：**

**姓 名：**

**指导教师姓名：**

**申请学位级别：**

**论文提交日期：**

**学位授予单位：**

摘 要

随着信息时代互联网技术的快速发展、人民经济水平与生活质量追求的提高，人们对在网上消费享受方便快捷的服务愈发享受，在饮食上外卖的使用也愈发频繁，而在校园中莘莘学子为了能有更多的时间用来学习对外卖这种餐饮方式亲赖有加。而校园作为学子们踏入社会前的最后一座花园，与大众社会还是有不少不同，为了解决这些不同给学生教职工带来的不便，本次毕业设计选题为校园订餐系统。本设计旨在设计和开发一款高效、便捷、安全的校园订餐系统，以提高学生和教职工的餐饮服务体验。

本次设计首先对校园订餐系统的需求进行了分析和调研，了解用户对于订餐系统的期望和需求。在此基础上，系统架构和功能进行了设计，包括用户注册、登录、个人信息、菜单浏览、购物车、支付、订单查询等功能。同时，为了更好地保障用户信息的安全，采用了加密技术对用户密码和支付信息进行保护。

在系统开发过程中，采用了前后端分离的开发方式，使用了Spring Boot框架和Vue.js框架进行开发，同时还使用了MySQL数据库和Redis缓存技术优化系统性能。在测试阶段，使用了JUnit和Postman等测试工具对系统进行了功能测试和性能测试，确保系统稳定性和用户体验。

最终，本设计开发的校园订餐系统得到了良好的用户反馈，用户可以方便地选择菜品、下单、支付、查询订单等操作，同时也提高了餐饮服务的效率和质量。本论文的研究成果可以为其他类似系统的设计和开发提供参考和借鉴，也为校园生活中的餐饮服务提供了更好的解决方案。

关键字：校园外卖 ；订餐系统；Spring Boot; Vue.js

**ABSTRACT**

As the rapid development of internet technology in the information age and the increasing pursuit of economic and living standards, people are increasingly enjoying the convenience and efficiency of online consumption, including the frequent use of food delivery services. In the campus, students rely heavily on food delivery to save time for study. However, as a unique environment, the campus has its own characteristics that require a customized solution. Therefore, this graduation project focuses on the design and development of an efficient, convenient, and secure campus food ordering system to improve the dining experience of students and faculty.

Firstly, the project analyzed and investigated the requirements of campus food ordering system, and understood users' expectations and needs. Based on this, the system architecture and functions were designed, including user registration, login, personal profile, menu browsing, shopping cart, payment, and order inquiry. To ensure the security of user information, encryption technology was applied to protect user passwords and payment information.

During the development process, the project adopted the front-end and back-end separation development approach, using Spring Boot framework and Vue.js framework, and MySQL database and Redis caching technology to optimize system performance. In the testing phase, JUnit and Postman were used to conduct functional and performance tests to ensure system stability and user experience.

Finally, the campus food ordering system developed by this project received positive feedback from users. Users can easily select dishes, place orders, make payments, and check order status, thereby improving the efficiency and quality of dining services. The research results of this project can provide reference and inspiration for the design and development of similar systems, and provide a better solution for dining services in campus life.

**Keywords:** Campus takeout; Ordering system; Spring Boot; Vue.js

目 录

[第一章 引言 1](#_Toc132320049)

[第一节 研究背景及意义 1](#_Toc132320050)

[第二节 校园订餐系统国内外发展对比与借鉴 2](#_Toc132320051)

[第三节 文章组织结构 3](#_Toc132320052)

[第二章 社会背景调研与可行性分析 5](#_Toc132320053)

[第一节 社会背景调研 5](#_Toc132320054)

[第二节 校园订餐系统的现状与前景 6](#_Toc132320055)

[第三节 校园订餐系统的可行性分析 6](#_Toc132320056)

[第三章 系统开发环境及技术 8](#_Toc132320057)

[第一节 系统开发环境简介 8](#_Toc132320058)

[第二节 系统开发相关技术和工具 8](#_Toc132320059)

[第四章 系统总体分析设计 11](#_Toc132320060)

[第一节 系统整体思路及架构 12](#_Toc132320061)

[第二节 模块功能基于权限介绍 12](#_Toc132320062)

[第五章 校园订餐系统实现 15](#_Toc132320063)

[第一节 数据库设计 15](#_Toc132320064)

[第二节 登录注册模块的设计实现 19](#_Toc132320065)

[第三节 订餐系统功能模块的设计实现 21](#_Toc132320066)

[第六章 系统功能测试 24](#_Toc132320067)

[第七章 总结 32](#_Toc132320068)

[参考文献 34](#_Toc132320069)

[致 谢 36](#_Toc132320070)

第一章 引言

# 第一节 研究背景及意义

进入21世纪以来，随着信息技术的发展，人民经济水平与生活质量的提高，移动互联网技术和电子商务普及开来，以外卖为代表，大大方便了人民的生活需求，给人民带来了十分方便的服务。根据国家统计局发布的《中国统计年鉴2021》数据，2019年全国居民互联网普及率为61.2%，而2020年居民互联网普及率为64.5%。又根据CNNIC（中国互联网信息中心）发布的《中国互联网发展状况统计报告（2021年6月）》，截至2021年6月，中国互联网用户规模达到了9.93亿，互联网普及率为70.4%。预计到2022年，中国居民互联网用户规模将继续增长。如此庞大的互联网用户群体，可见我国居民对互联网服务的需求之高，以外卖为例，根据艾瑞咨询发布的《2021年中国外卖市场研究报告》显示，2020年中国外卖市场交易规模达到6912.2亿元，同比增长18.9%。其中，大学校园是外卖市场的重要组成部分。在众多互联网用户当中，大学生这一群体尤显突出。据国家信息中心发布的《2019年中国大学生网络消费报告》显示，2019年中国大学生网络消费总额达到2.2万亿元，同比增长22.8%，其中网络购物、外卖订餐、在线旅游等成为大学生消费热点。可见大学生对于网络消费的习惯与依赖，校园订餐系统作为其中的一种形式也受到了越来越多的关注和使用。根据美团外卖发布的《2021全国高校外卖消费报告》显示，截至2021年6月，美团外卖在全国范围内已经覆盖了2000多所高校，日订单量达到160万单。校园订餐系统的不断完善和发展，越来越多的学生和教职工开始使用订餐系统进行点餐，同时这也促进校园订餐系统用户数量的逐年增长。综上所述，校园订餐系统随着移动互联网技术和电子商务的普及出现和发展，其引发的经济效益肉眼可见，大学生作为一种稳定而又流动的消费群体，现今虽然已有美团饿了么等外卖巨头，但校园订餐系统依旧有着不小的发展空间。

现今随着国家对教育的重视，国内各所大学都进入飞速发展阶段，在校本科生研究生人数年年新高，据统计:

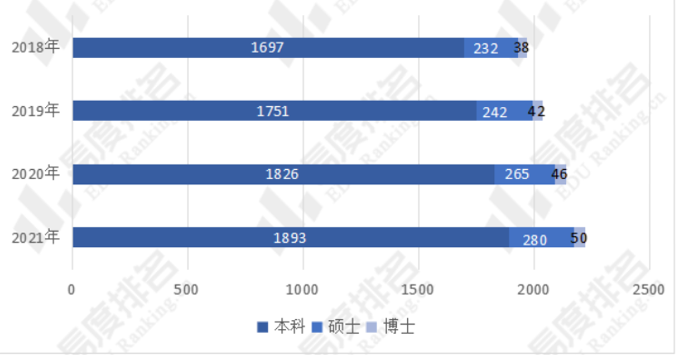


图 1.1 在校学生人数增长图

如此数量的学生因校园关系饮食等消费均在校内，与之相对的我国有上万座校内食堂，然而近些年因食堂内餐饮种类单一、大锅饭冷热不均、口味差、下课时间统一食堂拥挤等，随着校园订餐的出现，越来越多的学生不选择食堂就餐而是网上订餐。根据2021年美团外卖发布的《2021全国高校外卖消费报告》，截至2021年6月，美团外卖在全国范围内已经覆盖了2000多所高校，其中在校学生网上订餐人数逐年增长，已经超过食堂就餐人数。具体来说，报告显示，2020年高校外卖消费订单量同比增长超过80%，在校学生网上订餐人数占比为60.9%，而食堂就餐人数占比为39.1%。这表明，在校学生网上订餐已经成为了主流消费方式，而传统的食堂就餐方式则逐渐被取代。

使用校园订餐系统在网上订餐，学生与职工既可以品尝校内食堂食品，也可以通过外卖配送品尝校外的各种美食，大大丰富了学生与职工们的饮食品类与口味，同时打通了校内校外餐饮商铺的竞争通道，使其通过在价格与质量方面的有序竞争给学生和职工提供更优质的餐饮服务。同时，通过增加餐饮渠道的方式，降低了校内早中晚饭点时的交通拥挤，为学生和职工们减少出行压力同时节约了时间。

当然，校园订餐系统的出现在方便优惠了学生与教工们的同时，也造成了新的问题与矛盾，如食品垃圾在食堂外的肆意丢弃、外卖丢失等。为学生与教工提供更优质的服务的同时去金莲解决这些问题，就是校园订餐系统发展的方向。

# 第二节 校园订餐系统国内外发展对比与借鉴

校园订餐系统在中国和国外的发展情况存在些许差异，有以下四种主要差异。

第一种，发展时间差异。

校园订餐系统在国外的发展比在国内早。早在2004年，美国就有了第一个校园订餐系统Tapingo（后被Grubhub收购），而在国内，校园订餐系统的出现要晚得多，大约在2010年左右才开始逐渐普及。

第二种，使用方式差异。

国内校园订餐系统的使用方式相对简单，一般通过手机APP点餐、支付，然后到指定地点自取或送餐上门。而在国外，许多校园订餐系统则采用在线预约、送餐服务等更为完善的服务体系，包括在学校内设立配送站点或者与外部餐饮商合作等，以提高服务质量。

第三种，市场规模差异。

国内的校园订餐市场规模较大，由于中国大学生数量庞大，且普遍三餐甚至多惨，用餐需求量大，所以校园订餐系统的应用范围较广。相比之下，国外各个国家内，人口基数较少，学生数量不够多，校园订餐市场相对较小，但也在不断发展壮大。

第四种，支付方式差异。

国内的校园订餐系统支持多种支付方式，包括支付宝、微信支付、银行卡等，用户支付习惯相对成熟，支付体验也更加便捷。而在国外，许多校园订餐系统则使用校园卡或信用卡等相对传统的支付方式，但也有部分系统开始尝试使用移动支付等新型支付方式。

在校园订餐系统的发展上，国内可以借鉴国外的经验，保持自身优点的同时完善服务体系，以提高服务质量和用户体验。如校内设立统一配送站点，与监管部门和外部餐饮上合作，提供多种美食的同时加强食品质量和安全方面的监管，优化订餐网站与app，美化界面、加快运行速度、完善反馈渠道以此提高服务质量。

# 第三节 文章组织结构

本节用于说明全文的组织结构，文章主要介绍了该订餐系统设计与实现的整体思路与方法，并适当介绍了相关的技术与专业基础，读者可根据个人水平有选择的阅读内容，以下是关于文章的整体结构说明。

第一章引言部分，主要介绍了我国互联网发展现况老龄化现状以及校园订餐行业的发展与国内外对比，对读者初步了解校园订餐系统提供一定基础；第二章为社会背景调研和可行性分析，详细研究调查了我国校园餐饮的现状和问题，以及网上订餐的现状与前景。第三章为系统开发环境及技术，说明了该订餐系统的开发环境与运行环境，并对该平台开发过程中应用到的专业知识进行了简要介绍，本章仅作为了解该订餐系统的应用技术，不作为对相关专业知识的学习内容，若读者对相关知识已熟悉可不必阅读该内容；第四章为校园订餐系统总体分析设计，从前端、后端、数据库三个方面上对该订餐系统做出概览和模块分解，使订餐系统的结构体系清晰明了；第五章为订餐系统具体实现，对主要功能模块：登录注册模块、食品大厅模块、购物车模块和订单模块进行分析实现，每个部分都给出了详细实现思路和实现方法；第六章系统测试，展示了该订餐系统的实现效果；第七章总结，本章介绍该订餐系统综合技术要点，对不完善或实现不好的部分给出预计解决思路，以及订餐系统总体上可进行改进的方向。

第二章 社会背景调研与可行性分析

# 第一节 社会背景调研

1. 传统校园餐饮模式发展与现状

我国传统的校园餐饮模式主要是校内食堂与小卖部为学生提供统一制式食物，同时部分校园周围临近建有餐馆，学生在食堂或餐馆就餐或打包到宿舍就餐。就餐时间集中，主要在早晨6-9点，中午11-14点，傍晚5-8点，除此之外的时间点食堂和周边餐馆大多关门休整，无法为学生提供服务，同时，这三个就餐时间点也是学生的上课下课时间点，学生人流集中，经常造成拥挤现象。

在传统的校园餐饮模式中，食堂作为学生的主要餐饮选择，多数是学校承包给校外专营食堂的公司，食品出货量稳定且庞大，食堂通常具有较为上游的原材料供应链。与此同时，食堂往往具有统一且成体制的管理模式，人员流动较少。

在传统的校园餐饮模式中，校外周边餐馆作为学生的餐饮补充，在食堂制式如流水线般食品外给学生提供个性化差异化的选择，多数为小吃、烧烤、火锅及传统家庭菜肴。这些处于学校辐射范围内的餐馆往往聚集一处或多处，形成大学城，为学生在繁重学业外提供些生活上的烟火味。

这种以食堂为主，周边餐馆补充的传统餐饮模式为学子们提供饮食已经很久，可以说从校园建成便存在，在现今信息时代下，该模式存在着僵硬固化的现象，种种问题在电子商务的出现后显现，亟需改变。

二、以食堂为主的传统餐饮模式弊病

首先，食堂模式提供餐饮品类单调。

食堂为了满足庞大学生群体的食品需求量，在供餐时只能选择如大锅菜大锅煮等做饭方式，虽然有着巨大的出餐量，但能够使用大锅方式制作的菜品数量稀少，可供学生选择较少。同时又受大锅做法与厨师手艺的影响，菜品时常出现咸淡不均、过重过淡等口味问题，又因学生就餐时间幅度较长，后来的学生吃到的食品已经过了趁热的时间，对学生的就餐体验与身体健康有不小的负面影响。

然后，食堂占据主要模式地位，周边餐馆竞争力不足，形成垄断现象

理论上，拥有上游供应链且出货量大，食堂原材料成本较低，同时拥有大体量公司化管理，人力成本低，食堂所提供的食品即便做不到物美也可以做到价廉。然而现实中，校园内食堂的价格却居高不下，甚至比周边餐馆还要贵，与学校食堂类似的工地食堂给工人提供的是量大低廉的食品，校园食堂普通的两素一荤都能够有着10元以上的售价，学生甚至是有没有劳动收入的依附性群体。造成这种现象的主要原因便是缺少竞争，食堂在校园这个较为封闭的环境中具有垄断地位，这一地位在近几年疫情封校期间尤为明显。校园周边餐馆数量少体量小，难以与大体量的食堂进行强竞争，造成这种垄断现象。

除此之外，传统校园餐饮模式中还有着管理混乱，校内校外人员相互勾结等问题，这些都降低了学生的餐饮体验，而随着信息时代的发展，电子商务的出现，与互联网接轨的校园订餐系统的出现便有效的解决或缓解了这些问题与矛盾。

# 第二节 校园订餐系统的现状与前景

中国互联网络信息中心（CNNIC）最新发布的第51次中国互联网络发展状况统计报告显示，截至 2022 年 12 月，我国网民规模为 10.67 亿，同比增加 3.4%，互联网普及率达 75.6%。其中30岁以下人群占比34.7%，而学生中使用互联网的人群相对学生总数占比几乎100%。2019年中国大学生网络消费总额达到2.2万亿元，同比增长22.8%，可见学生群体对网络消费的习惯程度之高。

在校内餐饮方面的网络消费主要体现为外卖等网上校园订餐。而网络订餐，拓宽了学生的就餐渠道的同时，也丰富了学生可选择的菜品种类。与传统校园餐饮模式中的周边餐馆相比，传统周边辐射范围不超过2km，而网络订餐，可辐射的范围可达20甚至30km最远。报告显示，2020年高校外卖消费订单量同比增长超过80%，在校学生网上订餐人数占比为60.9%，而食堂就餐人数占比为39.1%。显然，在网络订餐的冲击下，传统模式中的食堂就餐占比大幅缩小，辐射范围的扩大极大的增加了学生们饮食的选择，有效打击了其旧有的垄断地位。个体而言，食堂仍旧是体量最大的主要提供者，但就整体而言，占比少于50%的食堂已经难以利用垄断地位肆意抬价，大量的商户可以与食堂形成有效的强竞争关系，健康的竞争关系可以给消费者学生带来更多更优的餐饮体验。

而学生群体的餐饮辐射范围的扩大，对整体经济而言也是起到了明显的帮助，提高了学生经济的流动范围，刺激辐射范围内消费增长。而校园订餐系统的发展完善便是能够在为学生提供更优质的服务体验的同时增大辐射范围，拉动更大范围内的消费增长。

# 第三节 校园订餐系统的可行性分析

1. 技术可行性

该校园订餐系统的开发主要由登录注册模块、食品大厅模块、购物车模块和订单模块四个模块构成。

为了开发的效率以及后续的扩展性，本设计采用前后端分离的方式完成设计。

对本系统的每个模块均设计相应的前端展示界面以及后端的逻辑数据编写，为完成本设计，本设计主要以Java为编程语言。笔者对Spring、SpringMVC、Mybatis框架有进行系统性完善的学习，最终使用SpringBoot框架对其进行整合使用，以Mysql数据库为后台设计四个模块的后端设计。SpringBoot为笔者快速构建应用，并可以自动化配置来方便开发，通过该框架提供的规范笔者完成对本订餐系统各个模块后端的开发与编写，实现数据的增删改查、转发渲染以及系统的逻辑运行。

前端部分，笔者粗略学习了Vue框架，设计上以Ruo-yi框架的前端界面为模板进行修改设计，最终完成订餐系统界面的展示。

综上所述，本订餐系统设计在技术方面基本理论知识均已具备，开发过程中需要学习补充部分第三方知识，总体设计难度中等，具有良好的学习价值和进步空间，适合作为本阶段的开发实践。

1. 经济可行性

本订餐系统基于SpringBoot和Vue框架开发，硬件需求仅有开发时常见的电脑以及其最后部署时所需的一台云服务器。软件开发环境均使用免费社区版或学生版进行开发。

总体软硬件设备所需经济较低，在能承受范围内，可以进行开发实践。

1. 操作可行性

本订餐系统旨在为校内学生和教职工提供网上订餐服务，用户仅需在网页上挑选心仪的菜品加入购物车，下单结算即可完成订餐服务。平台接收到订单后会将订单推送给商家，商家完成菜品后交由配送人员送于约定地点，操作简单，本系统使用者为学生群体和教职工，具备基本的互联网使用以及网络消费能力，能够完成本操作。

第三章 系统开发环境及技术

# 第一节 系统开发环境简介

1. Java

Java是一种面向对象的编程语言，具有可移植性、安全性、高效性、面向对象等特点，Web应用程序、移动应用程序、企业应用程序等领域得到了广泛应用。本设计便是Web应用程序，笔者使用该语言完成本系统的设计开发。关于Java具体安装及配置步骤这里不再赘述、笔者开发过程基于的版本号为jdk1.8。

1. IDEA

IDEA（IntelliJ IDEA）是一款由JetBrains公司开发的集成开发环境，主要用于Java语言开发。IDEA具有丰富的功能和特性，包括代码智能提示、自动代码重构、版本控制、调试等。其优秀功能以及在编写代码时提供的便利让他被称为业界最强大的IDE，帮助开发者提高开发效率和代码质量

IDEA有社区(免费)版，专业版，笔者这里使用的便是专业版IDEA。使用其完成代码的后端编写。版本号为2022.3.2

三、 WebStom

WebStom同样是一款由JetBrains公司开发的集成开发环境，主要用于Web前端开发，支持HTML、CSS、JavaScript等前端技术。和IDEA相同，WebStom也具有智能代码提示、语法高亮、代码重构、版本控制、调试等功能 。使用其进行前端开发十分方便。笔者这里使用的版本是2022.3版本。

四、 开发硬件环境

PC，Windows11，64位操作系统，CPU为i7-8750h，显卡型号为GTX1060-6G。

五、运行环境

该应用程序的运行环境操作系统为Windows7及更高版本，支持Chrome、Edge、Firefox等主流浏览器，不建议使用IE浏览器访问。

# 第二节 系统开发相关技术和工具

一、B/S架构

本系统是Java Web项目，而B/S架构与Web技术之间有着深切的关系。B/S 体系结构，也就是浏览器与服务器体系结构，是在互联网技术发展的基础上，对 C/S 体系结构进行了修改和完善。在这个体系结构中，用户的工作接口是由浏览器进行的，很少的业务逻辑由前端（Browser）来完成，而主要的交易逻辑则 由服务器端（Server）来完成。在此基础上，对系统的主要功能进行了集成，使 得系统的开发、维护和使用更加方便。在客户端上安装了浏览器，服务器上安装 数据库[3]。浏览器与数据库之间的数据交互是由 Web 服务器完成的，从而极大地 减少了客户机负载，减少了系统的维护和更新费用，减少了用户的总费用。

B/S 结构采用分布式处理技术，不仅可以改善系统的运行效率，而且可以大 幅度地减少资源的消耗和费用。现在，B/S 体系结构已经被广泛地应用，它也使 得客户机变得更加简单。B/S 结构具有更高的普遍性、不依赖于环境，与 C/S 相 比， B/S 结构具有更好的开发和维护能力，可以大大降低系统的开发和维护费 用。B/S 采用三个层次架构，即浏览器、服务器、中间件，通过互动和相互连接 来实现系统的功能和任务。

浏览器端：指用户所用的浏览器，是指使用者在网页上的一个动作界面，由 使用者透过浏览器向服务器提出要求，再将资料处理后呈现在使用者面前。

服务器端：提供数据的服务，它对数据进行运算，再将数据送回到中间层， 并在接口上显示出结构。

中间件：介于浏览器和服务器之间的一种应用，其主要作用是完成特定的功 能，接收请求并向服务器端发送请求，再将结果反馈给用户。在浏览器和服务端 之间，使用中间件进行交互。

二、Java Web简介

Java Web 是一种技术堆栈，它使用 Java 技术来处理有关的网络技术栈。网 络由两个部分组成：网络服务器和网络客户端。HTML、CSS、JavaScript 是客户 机中常用的程序。Java 在服务端有很多的应用，例如 Servlet、 JSP、第三方框 架等等。Java 技术为 Web 开发提供了强有力的支持。

Spring Boot是一种基于Spring框架的快速开发框架，它提供各种自动配置和依赖管理，简化了应用程序的开发流程，使得开发人员可以更加专注于业务逻辑的实现而不是繁琐的调试各种配置文件。同时，Spring Boot内置了多种Web容器，包括Tomcat、jetty和Undertow等，使得程序可以直接通过main方法启动应用程序，避免了繁琐的部署过程。

MyBatis是一种基于Java语言的持久化框架，它通过XML文件或注解的方式，将数据库操作和Java对象映射起来，提供了一种简单、灵活和高效的数据访问方式。MyBatis可以与各种关系型数据库（例如MySQL、Oracle、SQL Server等）相兼容，支持事务管理、缓存机制、分页查询等功能。

Spring Boot可以集成MyBatis框架，提供更简单和快速的数据访问方式。Spring Boot提供了MyBatis Starter，可以自动配置MyBatis框架和相关的依赖库，避免了繁琐的配置过程。同时，Spring Boot还提供了许多有用的开发工具和插件，例如Spring Data JPA、Spring JDBC等，可以进一步提高数据访问的效率和质量。

总之，Spring Boot和MyBatis是两种常用的Java开发框架，它们可以相互结合使用，提供一种快速和高效的开发方式，使笔者更为轻松便捷的完成本系统的设计。

三 、Spring Security 简介

Spring Security 是一个安全性架构，它可以为基于 Spring 的企业应用系统提 供一个声明性的安全存取控制方案。它提供一套 Bean，可以在 Spring 应用环境 中进行配置，实现了对应用系统的声明式安全存取控制，从而降低了对企业系统 安全性控制的大量重复代码。

Spring Security 是以 J2EE 企业应用程序为基础的综合安全服务。尤其是在企 业软件项目中，采用了领先的 J2EE 解决方案—— Spring 框架。Spring 安全对于 Web 安全的支持很大程度上取决于 Servlet 过滤器。在应用程式处理此要求前， 这些筛选器会拦截访问请求，并执行特定的安全性处理。Spring 安全提供了几个 过滤器，可以拦截 Servlet 请求，并把请求传递到验证和存取决定管理器来处理， 提高了安全性。您可以根据您的需求选择合适的过滤器来保护您的应用。

如果已经使用了 Servlet 筛选器并让它工作，那么您需要在 Web 应用的 web. xml 中使用< filter>和元素来配置这些组件。尽管这样可以工作，但在使用依赖注 入的情况下，它并不适合。

FilterToBeanProxy 是一种专门的 Servlet 过滤器，不需要自己做太多的工作， 只需要把自己的工作交给下面的 Spring 应用程序中的 Bean [9]。与其它 Servlet 过滤器类似，受委派 Bean 也可以使用 javax.servlet. filter 接口，但是这个接口是 用 Spring 配置来配置的，而非 web. xml。

事实上，由 FilterToBeanProxy 代理提供的 Bean 可以是 javax.servlet. filter 的任何一个实现方式。这可以是 Spring 安全中的任意一种，也可以是你自己建 立的。但就像本书中所说的，Spring 安全需要至少四个，甚至更多。

人们之所以会选择 Spring Security，是因为它们经常被 J2EE Servlet 和 EJB 规范所吸引。如果您改变了服务器环境，您需要在新的目标环境中做很多工作， 天津科技大学 2022 届本科生毕业设计 5 并为您的应用系统重新设定安全性。

Spring 安全解决了上述问题，同时还提供了许多其它的安全功能，这些功能 都是可以被指定的。

1. MysSQL简介

MySQL 是瑞典 MySQL AB 公司开发的关系式数据库管理系统。MySQL 是目前最受欢迎的关系数据库管理系统[2]，它是 WEB 中最好的一种应用程序。

MySQL 是一种把数据存储在不同表格中的关系式数据库，它能加快处理速 度，增强灵活性。

SQL 是 MySQL 中最常见的一种标准语言，它被用来访问数据库。MySQL 应用的是双重授权策略，分为社区版本和商用版本，因为 MySQL 具有体积小、 天津科技大学 2022 届本科生毕业设计 7 速度快、总体拥有成本低、特别是开源的特性，所以大多数中小站点都将 MySQL 作为其数据库。

MySQL 和其它大型数据库（比如 Oracle、SQL Server）相比有着自己的缺 点，但这并没有降低 MySQL 的流行程度。MySQL 为普通用户和中小公司所提 供的能力是足够的，并且因为它是开源的，所以它可以极大地减少整体的所有权。

Java 后端程序要与 MySQL 数据库进行连接，需要用到 JDBC 驱动。

JDBC: Java 数据库连接（JDBC），是 Java 语言中用于定义用户程序是怎样 存取数据库的一种规范方式，它为用户提供了查询和更新数据库数据的方式。 JDBC 是一个以关系为导向的数据库。JDBC 是一个“低级”的接口[3]，即用来 对 SQL 命令进行直接调用。在这一点上，它的性能非常好，并且比其他的数据 库连接 API 更容易使用，但是它还被设计成一个可以在上面创建更高层次的接 口和工具的基本接口[4]。先进的接口是一个“易于使用”的接口，它使用了一个 更易于理解和更加便利的 API，这些 API 会在后台转化成底层接口，比如 JDBC 这样的性能更高的低级接口。

在关系数据库的"对象-关系"的关联中，表中的每一行都与类的每个个体实 例相对应，而每个列的数值则与执行个体的属性相对应[5]。这样，编程人员就可 以直接操纵 Java 对象；在代码中，会自动产生访问数据所需要的 SQL 调用。另外，也可以提供更加复杂的映射，比如把多个表格中的数据合并到 Java 类中。

由于对 JDBC 的关注与日俱增，为了让编程变得简单，越来越多的开发者 开始采用 JDBC 的工具。使用 JDBC 进行数据库的访问过程如下：

1.数据库的加载（注册）

2.连接

3.SQL 语句的执行

4.对结果集合的处理

5.关闭数据库连接

第四章 系统总体分析设计

# 第一节 系统整体思路及架构

本校园订餐系统是一个基于Spring Boot框架的Java Web项目，旨在方便学生和教职工在校园内进行订餐，其系统大体思路和架构如下图所示：

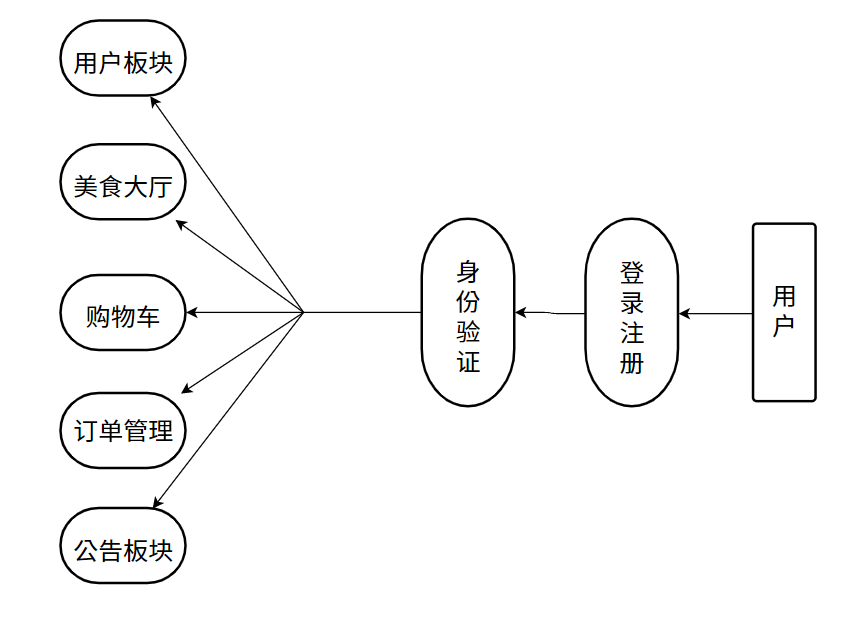


图4-1 系统框架图

用户通过浏览器访问系统网页端，首先进行注册，填写必要的部分用户信息，完成注册后使用注册时填写的账号密码登录本系统，系统服务器端接收到用户的账号密码信息后进行身份验证，验证通过后用户即可使用本系统，体验本系统的各个功能，本系统主要功能有展示和修改用户信息的用户板块、浏览挑选美食的美食大厅、挑选出心仪食品后暂时存放的购物车、记录操作订单信息的订单管理以及发布系统通告的公告板块。

# 第二节 模块功能基于权限介绍

基于用户身份的不同，不同权限的用户在使用本系统的各个功能模块时可以进行的操作也有所不同，大致操作如下图所示：

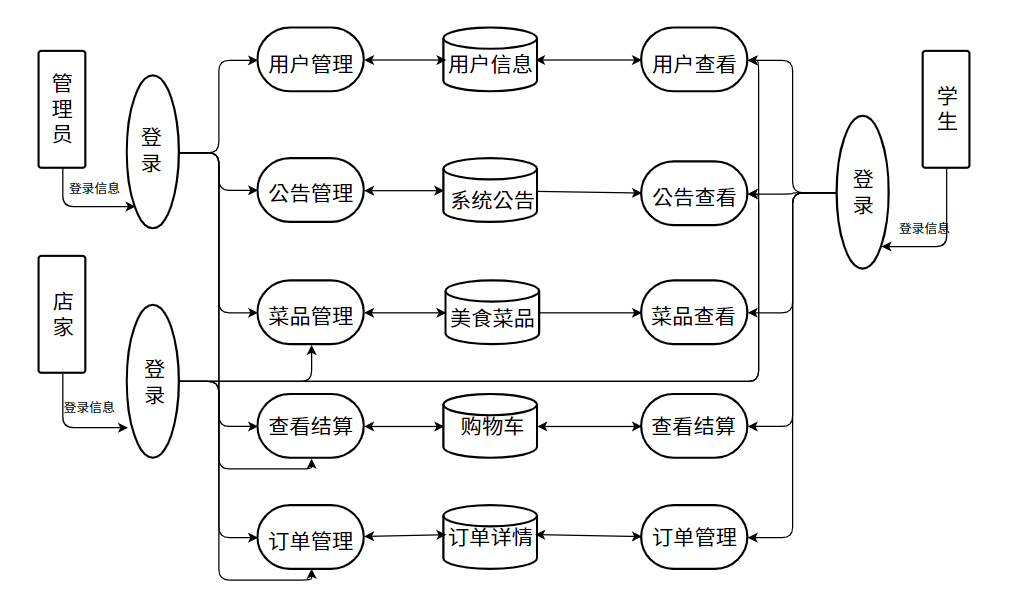


图 4.2 在校学生人数增长图

用户的身份总共分为三种，分别是管理员、店家和学生。这三种身份对应着三种不同的权限。

一、用户信息模块

在用户信息模块中，学生和店家都只具有对个人信息的查看修改权限，而管理员在具有对个人信息的查看权限的同时，还具有对所有用户权限的修改权限以及各个用户账号的停用权限，可以对所有的账号进行停用甚至是删除，也可以对每个账号的个人信息进行修改。

二、系统公告模块

系统公告模块中，学生和店家都只具有对公告的查看权限，而管理员用户可以对所有的公告进行管理。管理员用户可以增添发布新公告，同时也可以删除修改以后的公告，同时还可以在保留公告的同时将该公告关闭，形成暂存的效果。

三、美食大厅模块

美食大厅模块中，学生仅可以查看各个菜品的商品信息以及对商品的挑选添加购物车，而商家和管理员用户可以增添删除商品，也可以对商品的图片、描述、价格进行修改，商家仅可以管理自己发布的商品，而管理员可以管理所有商家发布的商品。

四、购物车模块

购物车模块，学生、店家还有管理员都可以对个人进行使用，都可以对其中的商品进行增删以及结算，改模块中三种身份权限没有不同，都只具备对个人购物车的管理操作。

五、订单管理模块

订单管理模块，学生仅能查询到个人的订单信息，而商家可以查询到该商家发布商品的所有订单信息但查不到其他商家发布商品的订单信息，而管理员用户则是可以查到所有人的订单信息。

第五章 校园订餐系统实现

# 第一节 数据库设计

在Web项目中，数据库是一个非常重要的组成部分，它承载着系统的数据和信息，支持系统的业务逻辑和功能实现，是系统的核心组件之一。数据库作为Web项目的数据存储和管理平台，承载着整个系统的数据和信息，其中包括了用户信息、业务数据等。

数据库设计是系统设计的基础，系统产生的所有需要持久化存储的信息都要存储在数据库中。数据库设计是系统开发中极其重要的一环，数据库设计的墓地便是建立系统所需要的数据库，从系统需求和实际出发，简历符合系统所需的数据库以满足系统和用户的需求。对整个系统而言，系统的运作便是通过对数据库中的数据进行存取，再经过系统的逻辑运算，最终将实现的功能结果渲染到识图界面上，没有数据库整个系统便是一个空架子。由于数据库应用系统的复杂，为支持相关程序运行，数据库设计也变得异常复杂，因此，合格的数据库设计不能一蹴而就，需要反复设探寻，精益求精。

数据库的设计原则主要包括一对一原则、独特命名原则和双向使用原则等 [1]。良好的数据库设计可以有效的节约资源，提高系统的运行效率，减少系统故障。

一对一原则：一对一原则即在数据库中两个实体之间存在唯一的关联关系，其中一个实体记录只能对应另一个实体几里路，通常这种关系使用外键来实现。一对一关系通常来说并不是必须的，一些情况下可以将一对一关系合并成一个表用来简化数据模型和提高系统性能，一对一关系由于存在关联查询，可能会对系统性能造成影响，而使用一对一关系则可以避免数据冗余和数据不一致问题，使用唯一索引保证一对一关系，则可以避免重复记录和错误数据。在系统性能和系统安全性之间进行权衡，就是在数据库设计时对一对一关系使用的考量。

独特命名规则：独特命名规则是指在软件开发中为了避免命名冲突和提高代码的可读性而采用的一种命名规则，独特命名规则要求每个标识符都具有唯一的命名，以避免命名冲突和混淆。命名时，标识符名称要简洁明了以易于理解和记忆，尽量避免过长、复杂的命名方式以保证代码的可读性。同时命名也不能过于简洁到abcd这种程度，要有一定的语义化，要能够表达其所代表的含义，这样才能避免混淆和误解，同时，同一数据库中的实体元素的命名要有一致性，使用相同的命名规则，便于开发人员的理解和维护。当然，命名也要具有唯一性，不能不同的元素起相同的名称，如有类似意义，可采用命名空间、前缀等方式进行区分。独特命名规则在软件开发中十分重要，它提高代码的可读性和可维护性，使得哪怕是不同的开发人员对同一项目进行开发时也可以更快的理解数据库设计从而进入开发。

本系统基于MySQL数据库进行数据管理，数据库设计如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | user\_id | bigint | 20 | 否 | 主键ID |
| 2 | user\_name | varchar | 30 | 否 | 用户账号 |
| 3 | nick\_name | varchar | 30 | 否 | 用户昵称 |
| 4 | email | varchar | 30 | 是 | 用户邮箱 |
| 5 | phonenumber | varchar | 11 | 是 | 手机号码 |
| 6 | sex | char | 1 | 是 | 用户性别 |
| 7 | avatar | varchar | 100 | 是 | 头像地址 |
| 8 | password | varchar | 100 | 是 | 密码 |
| 9 | status | char | 1 | 是 | 账号状态 |
| 10 | del\_flag | char | 1 | 是 | 删除标志 |
| 11 | login\_ip | varchar | 128 | 是 | 最后登录IP |
| 12 | login\_date | datetime | 0 | 是 | 最后登陆时间 |
| 13 | create\_by | varchar | 64 | 是 | 创建者 |
| 14 | create\_time | datetime | 0 | 是 | 创建时间 |
| 15 | update\_by | varchar | 64 | 是 | 更新者 |
| 16 | update\_time | datetime | 0 | 是 | 更新时间 |
| 17 | remark | varchar | 500 | 是 | 备注 |

表 5.1 sys\_user角色表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | order\_id | int | 10 | 否 | 主键ID |
| 2 | product\_id | int | 10 | 是 | 菜品ID |
| 3 | product\_name | varchar | 20 | 是 | 菜品名 |
| 4 | product\_price | decimal | 10,2 | 是 | 价格 |
| 5 | product\_picture | varchar | 100 | 是 | 菜品图片地址 |
| 6 | description | varchar | 255 | 是 | 详情 |
| 7 | buyer\_id | bigint | 10 | 是 | 购买人ID |
| 8 | buyer\_name | varchar | 20 | 是 | 购买人 |
| 9 | create\_time | datetime | 0 | 是 | 创建时间 |
| 10 | business\_id | bigint | 10 | 是 | 商家ID |
| 11 | business\_name | varchar | 30 | 是 | 商家 |
| 12 | product\_flavour | int | 5 | 是 | 口味 |
| 13 | order\_status | int | 5 | 是 | 订单状态 |
| 14 | shop\_num | int | 10 | 是 | 商品数量 |

表 5.2 order\_info订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | product\_id | int | 10 | 否 | 主键菜品ID |
| 2 | product\_name | varchar | 20 | 是 | 菜品名 |
| 3 | product\_price | decimal | 10,2 | 是 | 价格 |
| 4 | product\_picture | varchar | 100 | 是 | 菜品图片地址 |
| 5 | description | varchar | 100 | 是 | 详情 |
| 6 | buyer\_name | varchar | 20 | 是 | 购买人 |
| 7 | create\_time | datetime | 0 | 是 | 创建时间 |
| 8 | business\_id | bigint | 10 | 是 | 商家ID |
| 9 | business\_name | varchar | 30 | 是 | 商家 |
| 10 | product\_flavour | int | 5 | 是 | 口味 |

表 5.3 product\_info美食表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | shop\_cart\_id | int | 10 | 否 | 主键ID |
| 2 | product\_id | int | 10 | 是 | 产品ID |
| 3 | product\_name | varchar | 20 | 是 | 产品名 |
| 4 | product\_price | decimal | 10,2 | 是 | 价格 |
| 5 | product\_picture | varchar | 100 | 是 | 图片 |
| 6 | description | varchar | 255 | 是 | 详情 |
| 7 | buyer\_id | bigint | 10 | 是 | 购买人ID |
| 8 | buyer\_name | varchar | 20 | 是 | 购买人 |
| 9 | create\_by | varchar | 64 | 是 | 创建者 |
| 10 | create\_time | datetime | 0 | 是 | 创建时间 |
| 11 | update\_by | varchar | 64 | 是 | 更新者 |
| 12 | update\_time | datetime | 0 | 是 | 更新时间 |
| 13 | remark | varchar | 255 | 是 | 备注 |
| 14 | business\_id | bigint | 10 | 是 | 商家ID |
| 15 | business\_name | varchar | 30 | 是 | 商家 |
| 16 | product\_flavour | int | 5 | 是 | 口味 |
| 17 | shop\_num | int | 10 | 是 | 商品数量 |

表 5.4 shop\_cart购物车表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | notice\_id | int | 4 | 否 | 主键公告ID |
| 2 | notice\_title | varchar | 50 | 否 | 公告标题 |
| 3 | notice\_type | char | 1 | 否 | 公告类型 |
| 4 | notice\_content | longblob |  | 是 | 公告内容 |
| 5 | status | char | 1 | 是 | 公告状态 |
| 6 | create\_by | varchar | 64 | 是 | 创建者 |
| 7 | create\_time | datetime | 0 | 是 | 创建时间 |
| 8 | update\_by | varchar | 64 | 是 | 更新者 |
| 9 | update\_time | datetime | 0 | 是 | 更新时间 |
| 10 | remark | varchar | 255 | 是 | 备注 |

表 5.5 sys\_notice公告表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | role\_id | bigint | 20 | 否 | 主键身份ID |
| 2 | role\_name | varchar | 30 | 否 | 身份 |
| 3 | role\_key | varchar | 100 | 否 | 身份权限 |
| 4 | status | char | 1 | 否 | 角色状态 |
| 5 | del\_flag | char | 1 | 是 | 删除标志 |
| 6 | create\_by | varchar | 64 | 是 | 创建者 |
| 7 | create\_time | datetime | 0 | 是 | 创建时间 |
| 8 | update\_by | varchar | 64 | 是 | 更新者 |
| 9 | update\_time | datetime | 0 | 是 | 更新时间 |
| 10 | remark | varchar | 255 | 是 | 备注 |

表 5.6 sys\_role身份表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | menu\_id | bigint | 20 | 否 | 主键权限ID |
| 2 | menu\_name | varchar | 50 | 否 | 权限名称 |
| 3 | parent\_id | bigint | 20 | 是 | 父权限ID |
| 4 | path | varchar | 200 | 是 | 路由地址 |
| 5 | component | varchar | 255 | 是 | 组件路径 |
| 6 | query | varchar | 255 | 是 | 路由参数 |
| 7 | create\_by | varchar | 64 | 是 | 创建者 |
| 8 | create\_time | datetime | 0 | 是 | 创建时间 |
| 9 | update\_by | varchar | 64 | 是 | 更新者 |
| 10 | update\_time | datetime | 0 | 是 | 更新时间 |
| 11 | remark | varchar | 255 | 是 | 备注 |

表 5.7 sys\_menu菜单权限表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | role\_id | bigint | 20 | 否 | 主键身份ID |
| 2 | menu\_id | bigint | 20 | 否 | 主键菜单ID |

表 5.8 sys\_role\_menu身份权限表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 列名 | 数据类型 | 长度 | 允许空 | 说明 |
| 1 | user\_id | bigint | 20 | 否 | 主键用户ID |
| 2 | role\_id | bigint | 20 | 否 | 主键身份ID |

表 5.9 sys\_user\_role用户身份表

# 第二节 登录注册模块的设计实现

本模块是整个系统的入口，也是用户使用本校园订餐系统的开始。用户只有通过本模块完成个人信息的录入注册后再通过本模块进行登录才可以使用本系统。

登录界面如图所示：

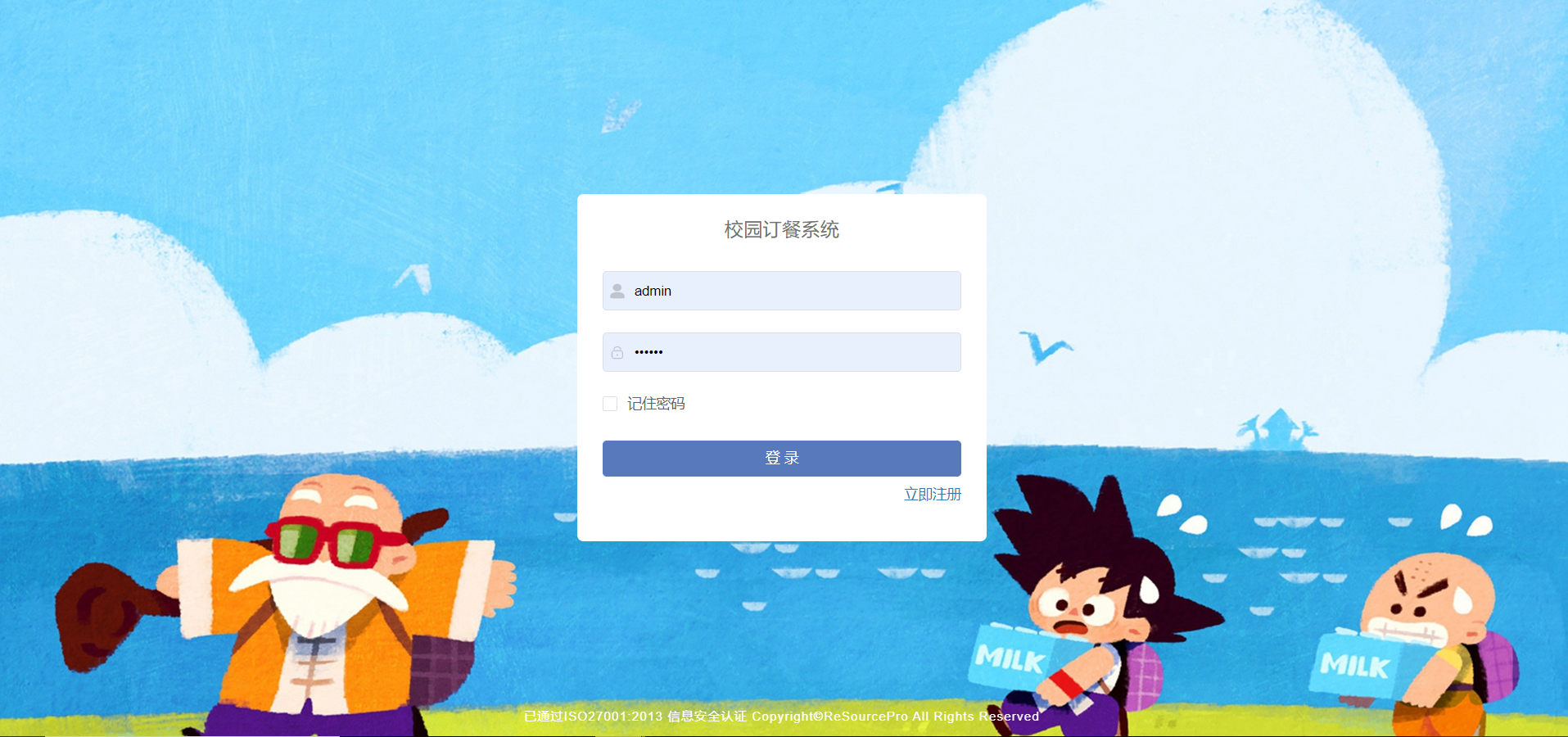


图 5.10 登录界面图

注册界面如图所示：

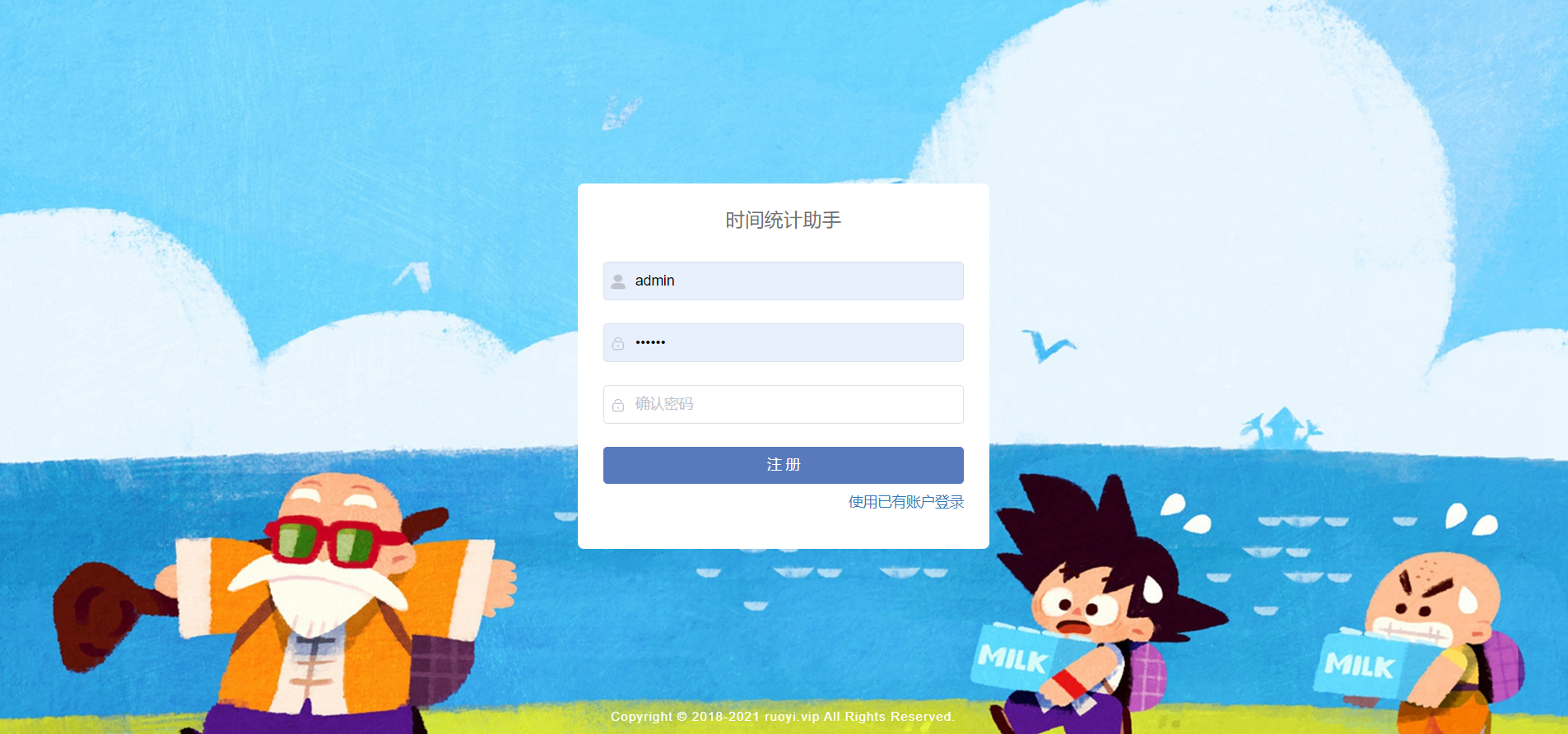


图 5.11 注册界面图

本模块主要由以下几个部分组成：

前端页面，前端页面：包括登录页面和注册页面，主要负责用户输入用户名、密码等信息，并进行相应的表单验证和提交。

前端控制器：前端控制器：负责前端页面和后端API接口之间的数据交互，包括发送请求、接收响应等操作。

后端API接口：提供登录和注册相关的服务接口，用于接收前端发送的请求，并返回相应的响应结果。

后端控制器：即controller控制器层，负责处理前端发送的请求，并调用相应的业务逻辑进行处理，例如验证用户名密码、创建用户账号等操作。

业务逻辑层：即service服务层，负责实现登录注册相关的业务逻辑，例如对用户输入的信息进行校验、进行用户身份验证等操作

持久层：根据数据库定义用户数据模型，并对数据库中的用户信息进行增删改查等操作。

本模块实现分为前端后端两个部分。

前端部分：

编写登录和注册页面：基于Ruoyi-Vue框架提供的页面模块在Vue.js框架中编写登录和注册页面，包括表单和按钮等元素。

表单验证：通过Vue.js框架提供的表单验证功能，对用户输入的用户名和密码进行校验，确保输入的格式正确。

发送请求：基于Ruoyi-Vue的路由规则填写路由配置，与后端API相应，使用axios库向后端发送登录和注册请求，传递用户名和密码等参数。

处理响应：处理后端返回的响应，如果登录或注册成功，将用户信息保存到浏览器cookie中缓存，并跳转到系统首页。

后端部分：

编写后端API接口：基于SpringBoot框架提供登录和注册相关的接口，将接收到前端发送的登录注册请求和参数信息传递给控制器层，待控制器层完成登录请求的工作后将处理结果相应给前端。获取路由信息：

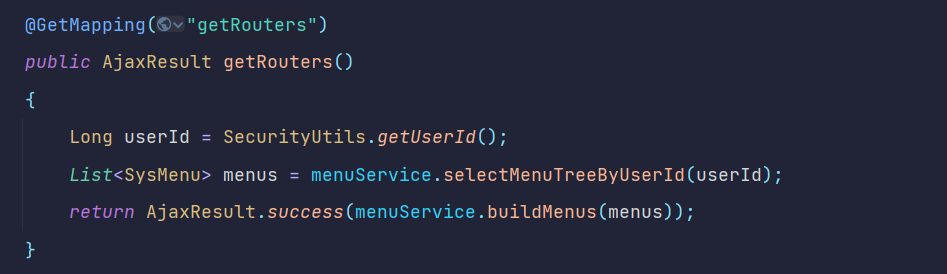


图 5.12 获取路由代码图

编写控制器：编写controller，使用service层提供的服务方法处理登录和注册的请求。对于登录请求，根据用户名和密码查询数据库，如果验证成功，返回成功的响应；对于注册请求，将用户信息保存到数据库中。

配置安全认证：使用Spring Security框架编写安全控制来保障系统的安全性，在控制器中配置安全认证，包括用户密码加密和会话管理等功能。Spring Security不拦截登录请求，在登录请求系统进行完数据库中的身份验证后会将封装用户的角色信息交由Spring Security 框架进行权限管理，然后跳转到首页。

编写持久层：编写关于用户信息的持久层，对数据库中用户信息表的进行增删改查，基于Mybatis使用mapper文件映射相应的SQL语句完成操作。这是接口文件和对应的mapper文件中的SQL语句例子：



表 5.13 mapper接口代码图

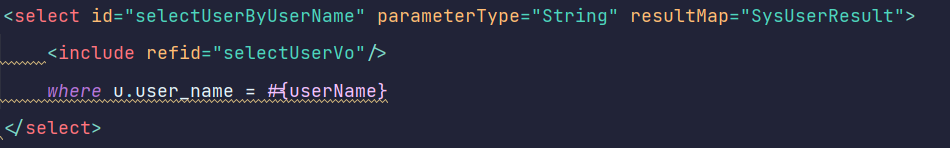


表 5.14 mapper文件SQL图

登录注册模块是校园订餐系统中的一个重要组成部分，其实现需要考虑到系统的安全性、稳定性和用户体验等方面，因此需要严格按照系统设计规范和开发流程进行实现。除了上述部分之外，登录注册模块还需要考虑到用户数据的保护和隐私问题，例如密码加密、数据备份等措施。对于这些，得益于Java对于web应用的强势性，现在这些细致但模板化的功能框架已经为我们编写好了，笔者这里便是按照Ruoyi-Vue框架提供的功能模板进行配置即可实现。从而能将更多的精力放在整个系统的逻辑研发上。

# 第三节 订餐系统功能模块的设计实现

如果说登陆注册模块是本系统的入口，那么功能模块便是本系统的业务体现，用来实现本系统为用户提供的选餐购餐以及查看个人信息、公告通知、订单记录服务。

各个功能模块实现的具体流程和登陆注册功能的实现相同，都大致可分为设计前端界面、编写前端控制器、适配后端API接口、编写后端控制器、细分功能操作编写服务层、使用Mybatis编写持久层与MySQL数据库进行交互实现对数据库的增删改查。实现的过程也都大同小异，这里便不再细讲每一部分的具体实现，笔者主要讲述各功能在系统运行层面上的实现。

个人中心模块用于用户信息的展示和修改。本模块需要展示用户的基本信息，并提供修改功能，例如修改昵称、头像等。使用Vue.js等前端框架来实现用户信息的展示和修改界面，通过调用后端API接口来实现用户信息的获取和更新操作。

与个人中心模块相对应，对于登陆者的身份，如果是管理员，则会多出一个管理个人信息的用户管理模块

用户管理模块需要提供用户信息的管理功能，例如新增、编辑、删除用户等。使用Vue.js等前端框架来实现用户信息的展示和管理界面，通过调用后端API接口来实现用户信息的增删改查操作。除却对用户信息的增删改查，更为重要的便是对用户身份权限的管理，用户在使用本系统时，是无法自己成为店家乃至管理员的。商家身份理应需要向系统平台方提供相应的营业安全执照并提交成为商家的申请。最后平台方确认店家资质后由管理员进行权限变更，管理员在系统中管理模块中便可以完成对于普通用户的身份变更。由于商家对应的主体是店铺而非店家个体，而学生身份对应的是学生个体，所以两者在系统中的功能上各有残缺，考虑到实际情况，这两种身份被设置为互斥关系，同一账号不可同时拥有两个身份，如果管理员将一个账号设置了学生+商店的身份，系统便会抛出异常提示。

不同身份对应不同权限，这一点在本系统的各个模块均有体现。

通知公告模块需要提供通知公告的发布和管理功能，例如新增、编辑、删除通知公告等。使用Vue.js等前端框架来实现通知公告的展示和管理界面，通过调用后端API接口来实现通知公告的增删改查操作。该模块中，学生和店家权限能够进行的只有查的操作，增删改操作仅有管理员权限才可以进行。

美食大厅模块要提供菜品信息的管理功能，例如新增、编辑、删除菜品等。使用Vue.js等前端框架来实现菜品信息的展示和管理界面，通过调用后端API接口来实现菜品信息的增删改查操作。菜品展示中包含由菜品的名称、价格、商家、图片等信息。学生身份仅能进行查的操作，增删改的操作需要有管理员权限或者商家权限才可以完成，同时，管理员权限高于商家权限，可以对所有商品进行增删改操作，而商家权限只能对商家自己发布的商品进行增删改。美食大厅模块最重要的便是点餐功能，点餐功能和购物车模块形成联动，用户在心仪商品后点击加入购物车后便会将该商品暂存于购物车中，该条购物信息便会保存在数据库的shop\_cart表中，并向前端返回添加成功的提示。

购物车模块要提供购物信息的管理功能，例如编辑、删除、结算购物。使用Vue.js等前端框架来实现购物车的展示和管理界面，通过调用后端API接口来实现购物信息的增删改查操作。本模块中并没有权限区分，所有身份所能使用到的功能是一样的，用户在美食大厅挑选好的食物会将食物信息和用户信息整合存放在shop\_cart购物车表中，再将shop\_cart中的整合数据展示在购物车界面上。每次登录后点击购物车后，都会从购物车表中根据用户id为条件检索出所有购物车记录展示在购物车界面上，实现购物车中的购物清单持久存在。用户勾选结算商品填写商品口味和数量后点击结算按钮后，便会将这些商品制成订单数据后存入order\_info表中，同时从购物车表中根据用户id和菜品id将相应记录删除。返回到前端界面上将结算商品记录展示删除。

订单管理模块同样也是提供订单信息的管理功能，如新增、编辑、删除订单记录。使用Vue。js等其阿奴但框架实现订单信息的展示和管理界面，通过调用后端API接口来实现订单信息的增删改查操作。本模块中通过用户id和商店id以及身份id检索订单表中的数据。管理员身份能够增删改查所有订单记录，学生身份仅能对跟用户id相同的订单记录进行操作，商店仅能对跟商店id相同的订单记录进行操作。在商家的订单界面中，可以对本店收到到的订单进行配送，商家点击配送后，订单表中该条订单的状态变成配送中，该订单对应的学生在收到商品后，对该订单点击确认收货，该订单状态变成完成。

各个模块中权限的区别是基于数据库实现。数据库中，对用户有独特的用户id，同时根据身份拥有不同的身份id，而根据身份id的不同又对应有不同的权限id，根据权限id来给各个用户分配各种权限。根据菜品id和商店id共同保证美食大厅中没有相同的菜品。根据个人id、商店id、身份id，统计订单信息，给不同身份展示相应权限的订单统计结果。

最后，为了展示本系统在校园中的使用情况，本系统开批了订单统计板块，根据订单表中所有的记录统计数量，展示使用本系统进行的订单数量。

第六章 系统功能测试

本章将展示本系统各功能模块的运行结果，测试各种情况下系统的运行结果。本章主要展示前端界面上的可视化展示，以图像的方式呈现检测结果。

一、登录注册测试

登录界面如图5-10所示

注册界面如图5-11所示

登录中界面如图



图6.1 登录中测试图

注册中界面如图



图6.2 注册中测试图

注册成功界面如图



图6.3 注册成功测试图

登录成功跳转首页界面如图



图6.4 首页测试图

二、功能模块测试

个人中心测试结果如图

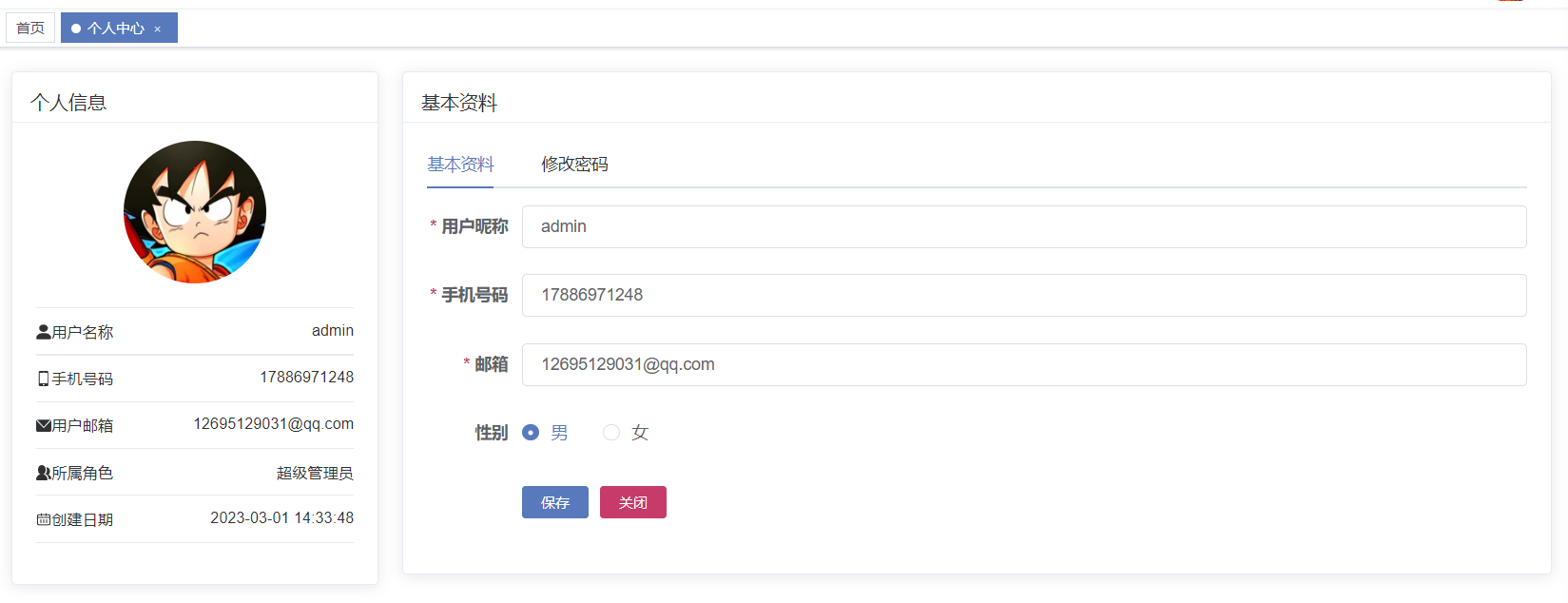


图6.5 个人中心测试图

用户管理界面如图

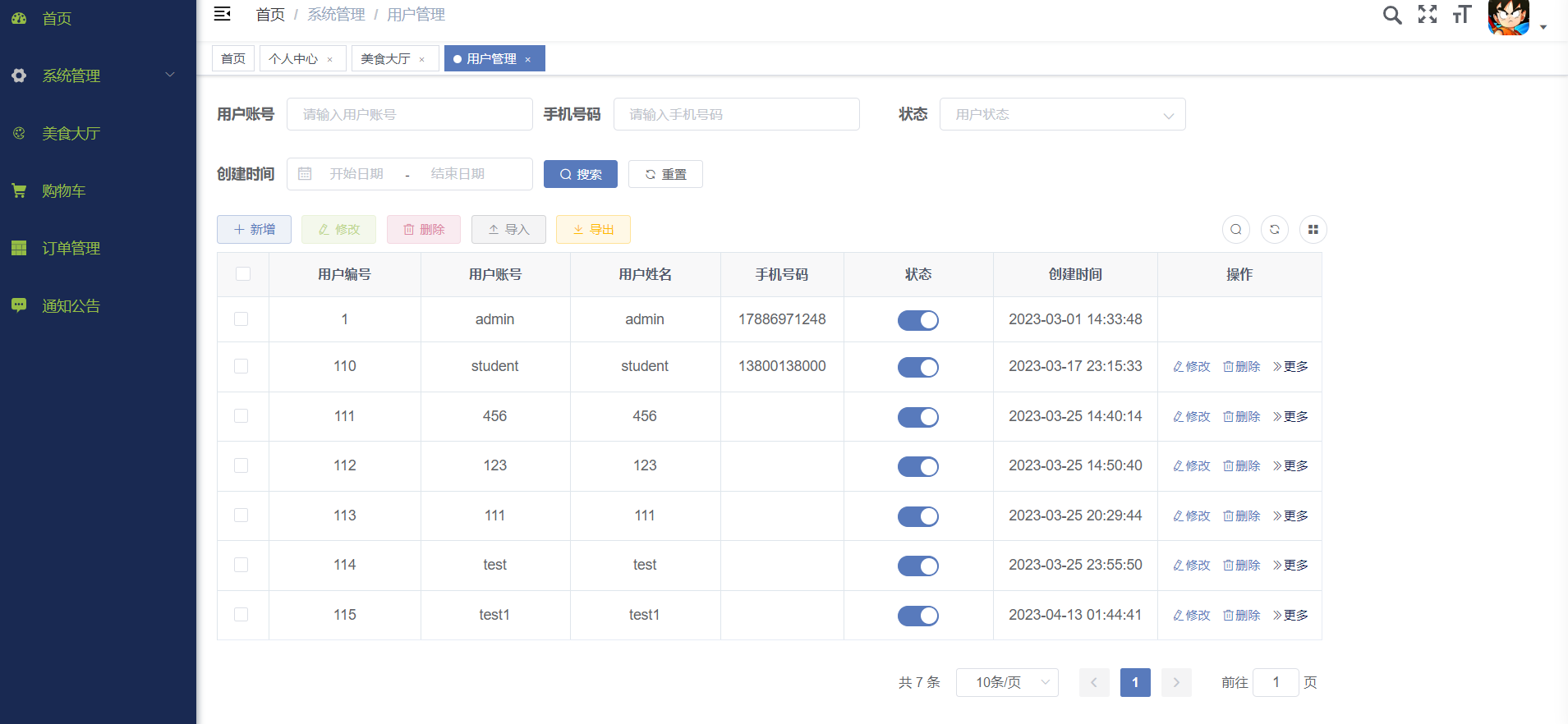


图6.6 用户管理测试图

美食大厅管理员界面如图



图6.7 美食大厅管理员测试图

美食大厅商店界面如图

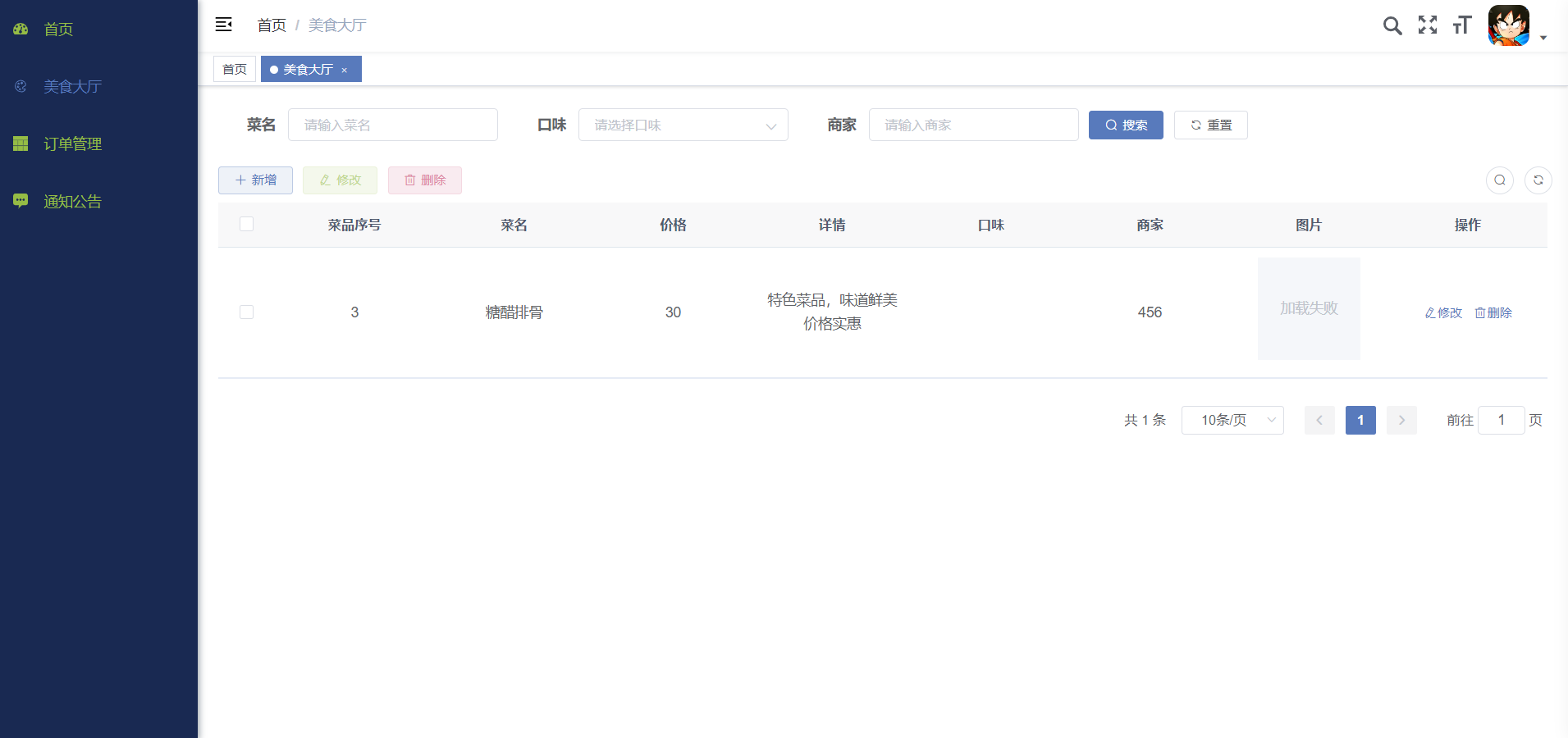


图6.8 美食大厅商店测试图

美食大厅学生界面如图



图6.9 美食大厅学生测试图

可以观察到每条食品信息右端根据不同的身份有不同的选择按钮，学生仅有加入购物车，管理员可以加入购物车也可以修改删除菜品，商店无法加入购物车

加入购物车成功如图



图6.10 加入购物车测试图

购物车界面如图

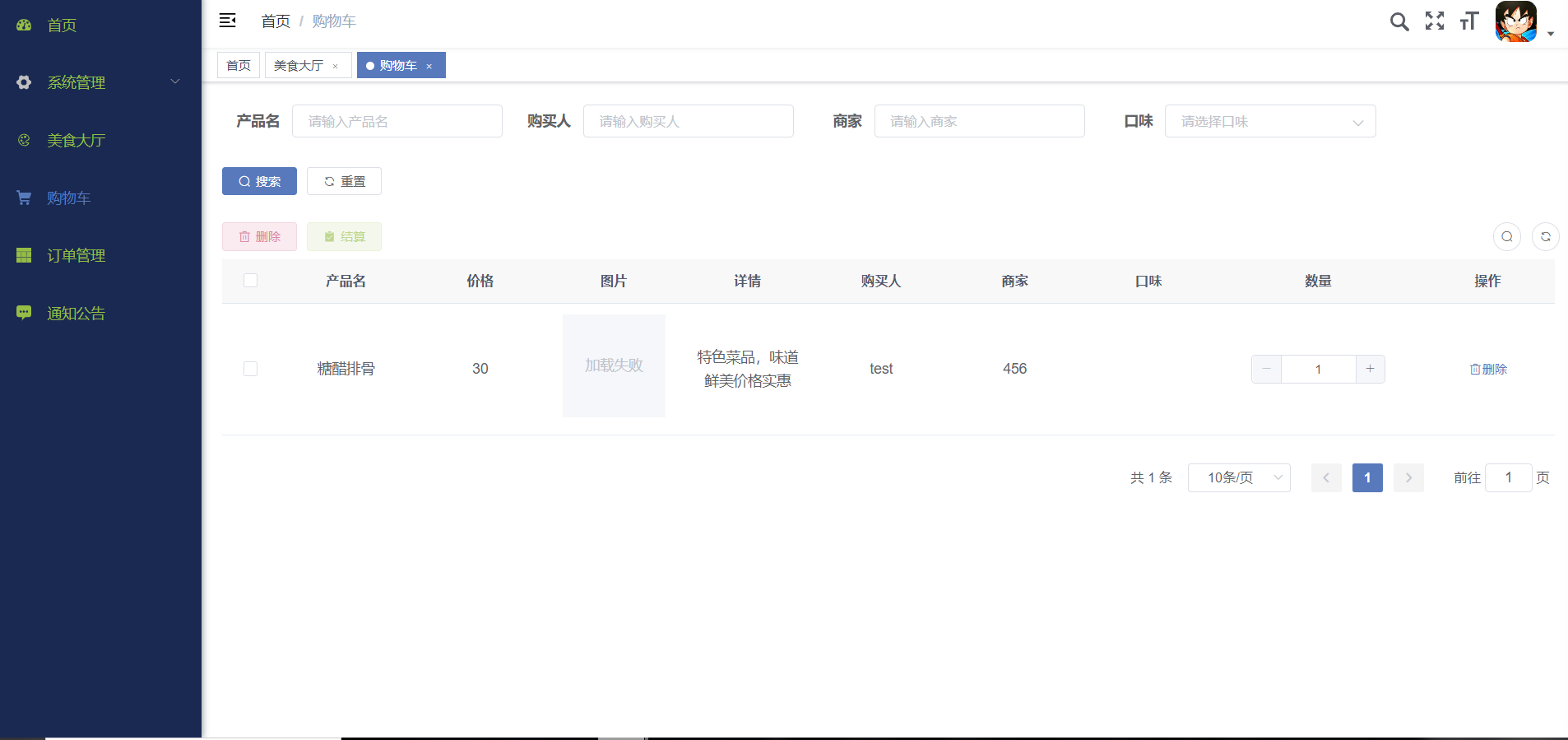


图6.11 购物车测试图

购物车删除菜品如图



图6.12 购物车删除测试图

购物车结算如图



图6.13 购物车结算测试图

订单管理管理员界面如图

图6.14 订单管理管理员测试图

订单管理店铺456界面如图

图6.15 订单管理店铺456测试图

订单管理店铺123界面如图



图6.16 订单管理店铺123测试图

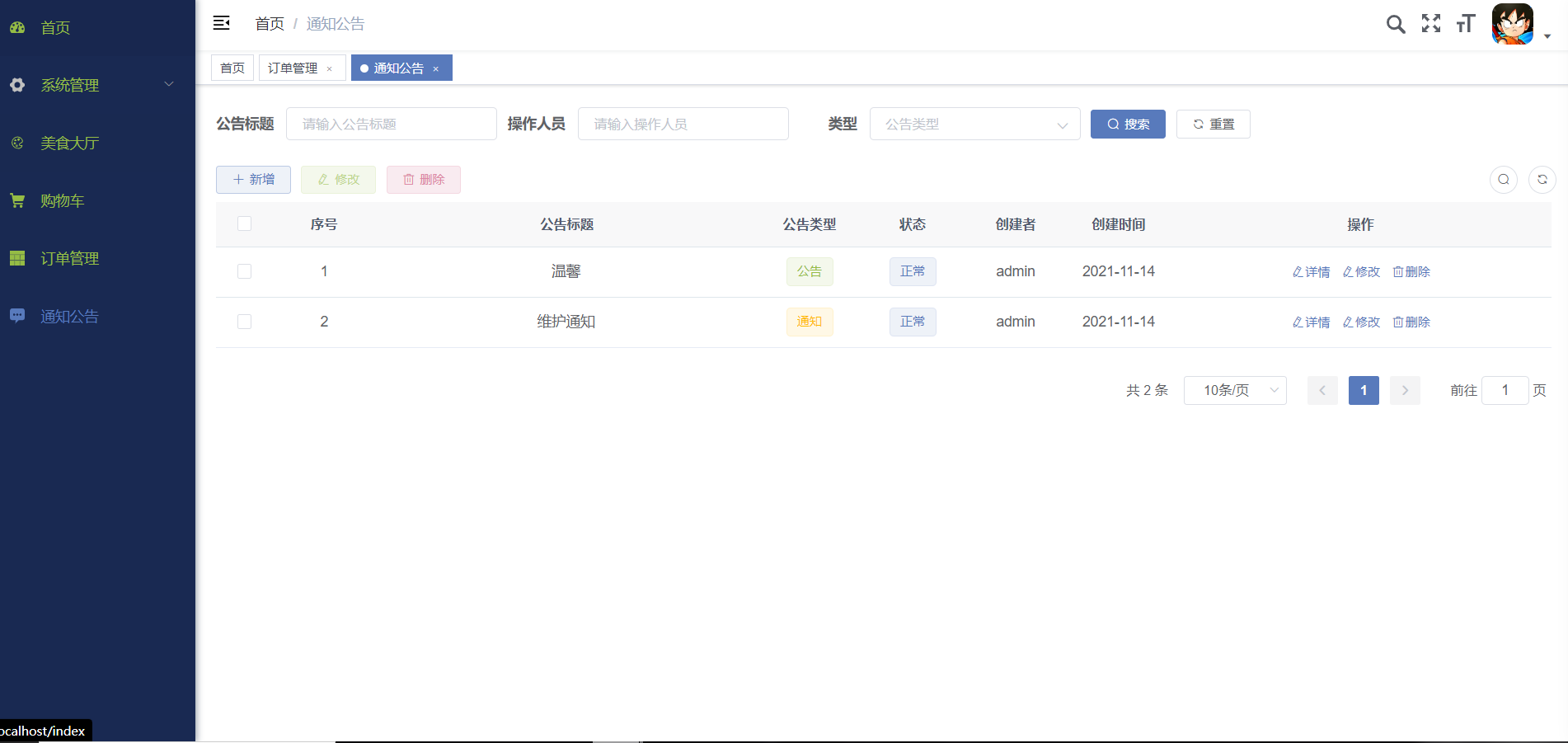
可以看出，管理员可以看到所有的订单记录，而商铺仅能看到本商铺发布菜品的订单记录。

公告板块店铺界面如图。



图6.17 公告板块测试图

公告板块管理界面如图。

 图6.18 公告板块管理界面测试图

公告板块修改界面如图。



图6.19 公告板块修改界面测试图

第七章 总结

校园订餐系统是一种基于互联网技术的新型订餐服务模式，它将传统的餐饮服务模式与现代化的互联网技术相结合，为广大校园师生提供便捷、快速、安全、优质的订餐服务。校园订餐系统的开发和应用，具有广泛的社会和经济意义。

校园订餐系统可以促进校园餐饮服务的现代化和智能化，随着信息时代的发展，餐饮行业智能化是必然趋势。传统校园餐饮模式结构僵化，活性不够，如一潭死水，食堂占据垄断地位，有着更低的成本却没有更低的价格，给没有收入的学生群体带来不小的经济压力。周围仅有大学城可以给学生提供其他选择，但小户经营体量小，成本也很难降下来，无所与食堂形成有效竞争，最终只会劣币驱逐良币。引入互联网后的校园订餐系统扩大了学校的辐射范围，能够将方圆更远距离内的大量餐馆引入学生的餐饮选择中，不仅丰富了学生们的选择，更可以跟食堂形成强有力的竞争，促进食品安全发展的同时还可以降低学生们的经济压力，让莘莘学子可以将更多的经济精力投入学业，为国家发展做出贡献。

为符合专业要求，也为了总结自身所学，笔者开发了本订餐系统。

本订餐系统是Web项目，而Java语言的开发环境在Web项目上有着得天独厚优势，是故笔者选择Java语言，基于SpringBoot和Vue框架，配合Ruoyi-Vue、Spring-Security框架进行系统开发。

Spring系列框架为Web项目开发整合提供了基于Java语言的全栈式解决方案，大量方便的接口能够帮助笔者快速高效的开发Web项目。SpringBoot框架更是一种优秀的整合框架，其中整合了Spring所有的功能同时还整合Mybatis和Tomcat等，并能够自动化配置，避免了繁琐复杂的配置部署过程，配合maven的依赖管理，让开发人员可以将全部的精力投入项目逻辑的开发上，而Ruoyi-Vue则是一款成熟的管理系统框架，提供了系统开发的常用功能模板，也将各种框架之间的配置整合简化，这种对Web项目开发几乎全方面的支持便是Java在Web项目开发地位的证明。

本人在对本系统开发的过程中，系统性的学习力JavaWeb、Spring、SpringMVC、SpringBoot，对于JavaWeb项目的开发流程有较深的理解，最终完成了本系统的开发，设计开发了系统的视图层、控制层、服务层、持久层和数据设计。最终本系统完成了登陆注册、个人中心、用户管理、美食大厅、购物车、订单管理、公告通知等功能，基本设计实现了订餐系统。

本文从该订餐系统的设计背景、技术依赖、功能实现、系统测试等方面对该系统加以介绍。系统的各个功能已经经过数次优化研发，通过研读相关论文、技术博文，为Web系统研发完成了一定程度技术积累，使我最终能够完成平台的开发。

由于本人学识能力有限，开发技术经验不够充分，时间准备不够充足，在开发过程中出现了部分问题尚未解决且大多数功能还有许多可扩展空间，使之部分后续研究并未实现：首先，界面设计不足，界面重复性较高，美观性不足，界面较为简陋；再者，菜品图片因为路径原因和对框架的不熟悉，导致无法显示在界面上；设计中的评价功能未能实现，菜品缺少学生的点评，反馈性不足；同时，学生使用互联网大多是通过手机进行访问，本系统网页未进行移动端适配，由于成本问题，学艺不精等原因未能成功将项目部署在服务器上。总之，目前本校园订餐系统仅是身为初学者的笔者的一次尝试，它还不够完善，功能有些缺失，实用性不够强，留待笔者以后技术经验积累充足后优化完善。

此次毕业设计的校园订餐系统开发对我而言是一笔非常宝贵的财富，是本人大学生活中首次独自从构思、设计、开发、测试、总结完成的一个较大的设计。在此次设计开发期间，我初步整合了大学期间所学的知识要点，能较为本质的理解计算机给笔者呈现的图像文字以及各种操作，从而实现设计的功能，同时也接触到了论文中更加广阔的视角以及深刻的问题思考角度和种种解决问题的方法，也在网络博文中学习了不少新鲜的知识，除此之外，我也领悟到，对于一个大型的项目，仅仅是代码的开发时不够的，没有实现的设计阶段，代码开发便会如同无头苍蝇一般研发混乱，即便是笔者做了自认为充足的设计准备，依旧在研发中遇到与设计效果不服、无法实现设计功能从而更改设计的状况，最终经过对研发成果的测试以及修改完善，才做出了较为完成的作品。一个项目的成功，是开发者学识、经验在时间上的积累，在次我受益良多。

# 参考文献

1. 李大虎.软件开发中数据库设计理论的实践探究[J].计算机产品与流通,2019(12):22.
2. 薄志强.基于SSM 框架的网上商城系统的设计与实现[D].西安电子科技大学,2020.DOI:10.27389/d.cnki.gxadu.2020.002269.
3. 姜洋. 国际学术会议网站管理系统的设计与实现[D].电子科技大学,2013.
4. 聂维,张忠,刘小豫.网上订餐系统的设计与开发[J].信息与电脑(理论版),2019,31(20):56-58.
5. 丁杰. 某企业招聘管理系统的设计与实现[D].北京邮电大学,2013.
6. 张巧岭.校园订餐系统设计[J].电子世界,2021(20):162163.DOI:10.19353/j.cnki.dzsj.2021.20.065.
7. Taylor Scott. Campus dining goes mobile: Intentions of college students to adopt a mobile food-ordering app[J]. Journal of Foodservice Business Research,2021,24(2).
8. Sam Field. Student Perspective of On-Campus Foodservice[J]. Foodservice Equipment & Supplies,2017,70(7).
9. Ning Ning Li,Xiao Fang Zhang,Yuan Fei Wang,Ran Zhang. Design and Implementation of Campus Dining Application Based on Android[J]. Applied Mechanics and Materials,2014,3207(556-562).
10. 潘钰婷. 高校新型餐饮空间发展与设计研究[D].浙江大学,2021.DOI:10.27461/d.cnki.gzjdx.2021.002573.
11. 李新锦,鲁志萍,刘苑如.基于微信小程序的校园点餐系统的设计[J].电脑知识与技术,2020,16(07):50-51+56.DOI:10.14004/j.cnki.ckt.2020.0756.
12. Faezirad Mohammadali,Pooya Alireza,NajiAzimi Zahra,Amir Haeri Maryam. Preventing food waste in subsidy-based university dining systems: An artificial neural network-aided model under uncertainty.[J]. Waste management & research : the journal of the International Solid Wastes and Public Cleansing Association, ISWA,2021,39(8).
13. Engineering; Recent Studies from Shenyang University of Technology Add New Data to Engineering (Designing a Flexible Catering System for High-speed Railway Considering Departure Time Selection and Time Deadline Constraints)[J]. Journal of Mathematics,2020.
14. 张健.餐饮系统的设计与实现[J].信息与电脑(理论版),2021,33(08):135-138.
15. 芝诺比娅·巴洛,迈克尔·K·斯通,王子舟.生命系统论和可持续领导力框架下的制度变革——以学校餐饮系统改革为例[J].世界教育信息,2019,32(14):28-38.
16. 杜卉婷. 基于WebService的智慧餐饮服务系统的研究与设计[D].湖北大学,2017.
17. 刘卓. 基于某航空公司C/S模式的餐饮管理信息系统的设计与实现[D].电子科技大学,2014.
18. 田君.校园周边餐饮企业外卖O2O模式发展对策研究——以天津某大学城为例[J].统计与管理,2017(03):141-142.DOI:10.16722/j.issn.1674-537X.2017.03.048.
19. 陈美含,崔美玉.校园餐饮电子商务贸易平台建设的可行性研究[J].科技资讯,2018,16(17):1-2.DOI:10.16661/j.cnki.1672-3791.2018.17.001.
20. 湖北省食品药品监督管理局、湖北省教育厅《关于进一步加强校园及周边餐饮服务食品安全监督管理工作的通知》[J].中国食品,2017(04):170-171.
21. 仝好林,谢超峰,孙源江.基于食品质量安全的高校餐饮外卖物流配送模式研究[J].食品安全质量检测学报,2018,9(20):5478-5481.
22. 余强,陈异子.基于服务设计理念的校园餐饮无人配送系统设计研究[J].工业设计,2021(02):26-27.
23. 魏晓旭. 基于Android平台无线校园餐饮系统的设计与研究[D].吉林大学,2016.
24. 郭建卫.基于SpringBoot+Mybatis+Vue的学生德育量化考核系统设计与实现[J].现代信息科技,2023,7(01):18-22.DOI:10.19850/j.cnki.2096-4706.2023.01.004.
25. 王以伍,舒晖.基于SpringBoot+Vue前后端分离的高校实验室预约管理系统的设计与实现[J].现代计算机,2023,29(01):114-117.

致 谢