**分类号 论文选题类型 非师范类应用研究**

**U D C 编号**



**本科毕业论文（设计）**

**题 目 食堂点餐微信小程序的设计与实现**

**学 院 计算机学院**

**专 业 计算机科学与技术**

**年 级 2017级**

**学生姓名 刘玥滢**

**学 号 2017211741**

**指导教师 李波**

**二○二一年四月**

**华中师范大学**

**学位论文原创性声明**

本人郑重声明：所呈交的学位论文是本人在导师指导下独立进行研究工作所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

学位论文作者签名： 日期： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士学位论文评选机构将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

1、保密 □ ，在\_\_\_\_\_年解密后适用本授权书。

2、不保密 □。

（请在以上相应方框内打“√”）

学位论文作者签名： 日期： 年 月 日

导师签名： 日期： 年 月 日

**摘 要**

近年来，几大外卖app在商业领域独占鳌头。商家大多使用第三方平台进行营销推广。在这种趋势下，食堂也面临着营业线上化的挑战。然而，疫情防控导致第三方平台外卖员无法入校配送等原因，导致食堂加入外卖平台时，难以提供同等级的便捷服务。可见，研发一款专门针对食堂点餐的系统是有必要的。微信小程序无需额外下载软件的特点迎合了学生的需求。同时，微信平台自带大量用户群体，能起到和外卖平台相同的宣传效果。基于上述原因，本文设计并实现了一个食堂点餐微信小程序。该微信小程序采取C/S结构，使用SSM和Uni-app框架进行开发，以eclipse为开发工具，利用MySQL完成系统数据的存储工作。小程序提供了可按食堂名或菜品名搜索菜品并点餐、收藏喜欢的菜品、为菜品点赞点踩等功能。同时配有后台管理网页，提供了管理员审核订单并进行审核回复的功能。二者共同形成一个功能完备的食堂点餐系统。

**关键词：**微信小程序 SSM Uni-app

**Abstract**

In recent years, several takeout apps have taken the lead in the commercial field. As a result, most stores use the third-party platform for marketing. Under this trend, school canteens are facing the challenge of online business. However, due to the COVID-19 prevention and control, delivery riders from third-party platforms are unable to get into the school. This and other similar factors make it difficult for the canteen to provide convenient service at the same level. Obviously, it is necessary to develop a special ordering system for canteens. Wechat mini-program has the advantage that its users don’t need to download additional software, which meets the needs of students. At the same time, wechat platform has a large customer base, which can play the same publicity effect as takeout platform. Based on factors above, this paper designs and implements a wechat mini-program for canteen ordering. The mini-program adopts C/S structure, uses SSM and Uni-app framework for development, takes Eclipse as the development tool, and uses MySQL to complete the storage of system data. It provides the functions of searching and ordering dishes according to the canteen name or dish name, collecting favorite dishes and praising dishes, and so on. There is also a backstage management web page which provides administrators with the function of auditing orders and replying to them. Both of them make up a functional canteen ordering system.

**Key words：**WeChat Mini-program SSM Uni-app

**目　录**

摘要 1

关键词 1

Abstract 1

Key Words 2

第1章 绪论 5

1.1 研究背景 5

1.2 研究现状 5

第2章 需求分析和可行性分析 7

2.1 需求分析 7

2.1.1 功能性需求分析 7

2.1.2 非功能性需求分析 7

2.2 可行性分析 8

2.2.1 技术可行性 8

2.2.2 经济可行性 8

2.2.3 操作可行性 8

2.3 关键技术 9

2.3.1 SSM框架 9

2.3.2 MySQL数据库 9

2.3.3 微信小程序 9

2.3.4 Uni-app框架 10

第3章 系统设计 11

3.1 系统架构设计 11

3.2 系统模块设计 11

3.3 数据库设计 12

3.3.1 概念结构设计 12

3.3.2 逻辑结构设计 13

3.4 详细设计 17

3.4.1 用户点餐功能设计 17

3.4.2 用户订单功能设计 18

3.4.3 用户收藏功能设计 18

3.4.4 管理员订单功能设计 19

第4章 系统实现 21

4.1 连接数据库 21

4.2 数据库与持久层实现 22

4.3 自定义API接口 24

4.3.1 base.js 24

4.3.2 http.js 24

4.3.3 index.js 26

4.4 用户点餐及收藏功能 27

4.5 用户订单功能 28

4.6 管理员订单功能 29

4.7 系统截图 29

第5章 总结与展望 32

参考文献 33

致谢 35

**第1章 绪论**

**1.1 研究背景**

自诞生以来，迅速发展的网络逐渐为全世界人民构建起了新的生活方式。根据2021年CNNIC即中国互联网络信息中心发布的报告，截至2020年12月，我国网民的规模已达9.89亿，互联网的普及率高达70.4%[1]。高普及的网络加快了人们的生活方式的转变，由传统升级为现代。最为明显的即是人们可利用手机完成各类工作。5G技术带来的网速变革更是进一步推动了移动app的发展。

微信，作为中国最广为使用的社交平台之一，在2016年9月首次推出了微信小程序的测试。以“人-技术-场景”为核心的微信小程序迅速受到人们的青睐[2]。据2020年腾讯公司公开的数据，微信的月活跃账号已达12.1亿，为微信小程序的开发和应用提供了巨大的发展空间。微信小程序自身具备无需下载即可使用的优点，加上微信平台提供的支付方式，自上线以来就保持着良好的发展趋势。

传统的食堂点餐方式是用户线下到食堂窗口进行购买、取餐操作，而在疫情防控常态化的背景下，食堂窗口排队队伍与进食部分并无明显分区的现状导致人员容易大量聚集，存在极大的安全隐患。此外，食堂的主要用户群体是学生与教师，二者皆具有课后固定时间进餐的特性，导致点餐高峰期人满为患，浪费时间。

本食堂点餐系统完美解决了以上问题：用户可以根据实际情况提前点餐，用户可以通过友好的界面来灵活选择餐食，点餐体验也更好，并根据食堂管理员的审核回复按时取餐，避免了点餐时犹豫不决导致浪费他人时间，提升了点餐及取餐效率。在食堂无空桌的情况下，管理员可在订单中进行备注，用户取餐后直接离去，减少食堂内人数，保证在食堂进食的用户的健康安全。

**1.2 研究现状**

当下发展较为完备的点餐系统主要分为两大类：外卖平台和堂食系统。

（一）外卖平台主要为餐饮类实体店提供网上展示菜品及销售的途径。国内的主流外卖平台主要为饿了么、美团和大众点评，商家也大多选择在上述三家平台上进行营销。部分销售量较大的品牌，如麦当劳、肯德基、喜茶等，在平台上运营的同时，还具有配套的微信外卖小程序，且门店有专门雇用送餐外卖员。国外也有类似的成熟产品，app如针对外卖的Yelp app，网站如Doordash和Orderup都受到了消费者的青睐。此外，原本以网约车业务为主的Uber，在外卖业务上也取得了不错的成绩。根据官方公布的数据，以2021年3月的统计为基础，Uber在前一年的外卖业务总预定量高达520亿美元，同比增长率高达150%[3]。

（二）实体餐饮店大多配有针对堂食客户的微信扫码点餐系统。用户可以在到店后扫描桌面二维码，从公众号入口跳转至微信小程序，并在小程序中完成从点餐到结账的所有步骤[4]。其中，公众号同时起到小程序入口和后期推送活动的营销作用。

以上两种点餐系统都不适用于食堂。

首先，外卖平台和校方难以达成一致。外卖员随意进出校园对学生的安全造成了一定隐患，对外卖员身体健康的监管和对电动车在校内行驶的管理都是难以解决的问题。

其次，即使雇佣了仅负责校内送餐的外卖员，其工资低于其他外卖员，存在难以招工的问题。以我校为例，南湖校区食堂已有部分窗口入驻外卖平台，外卖员的欠缺使得配送速度远慢于校外外卖。

第三，通过微信公众号扫码点餐的目的是方便商家后续推送活动，但食堂并不需要过多的营销，活动的推送反而会打扰学生的学习生活，造成负面影响。

考虑各方面因素，微信小程序与食堂点餐系统契合度更好，优点如下：

（一）开发周期短，运维成本低。微信小程序支持跨平台服务，无论是IOS还是安卓系统的手机，只要用户安装了微信即可享受服务。且微信小程序的开发周期远远短于普通app的开发周期，运营起来也更加方便。

（二）提前点餐、自行取餐。考虑到学生群体没有经济来源的特性，本系统不提供外卖服务。学生可在课间提前点餐，在下课后前往对应的食堂取餐。既规避了外卖员的种种隐患，又为学生节省了配送费。

**第2章 需求分析和可行性分析**

本章首先简要阐述了该系统的需求情况，然后对系统的可行性进行分析，最后，对本系统使用的关键技术进行说明。本章的分析均基于科学性、综合性和整体性的原则，保证业务需求分析的实用性。

**2.1 需求分析**

**2.1.1 功能性需求分析**

本系统为食堂提供了一个基于微信小程序的售卖平台，该平台主要分为两个大模块：用户微信小程序模块和食堂管理员网页模块。

（一）用户需求

在功能上，用户可通过注册、登录进入食堂点餐微信小程序，在首页可见所有食堂的菜品，在点餐界面可对食堂进行选择，并对选中的食堂内的菜品进行单独点餐。用户下单后，若食堂管理员审核通过该订单，则可虚拟支付，并在订单状态中查看该订单的详情状况。另外，用户可在个人界面对个人信息进行修改，也可将喜欢的菜品加入收藏，并在收藏界面中进行管理。

（二）食堂管理员需求

同样，注册与登录必不可少。进入网页后，管理员可选择进行食堂管理、管理员管理、菜品管理和订单管理。食堂管理需实现管理员对食堂的基本信息的操作。管理员管理需实现管理员对当前账号的账号及密码的操作。菜品管理需实现管理员对食堂菜品信息的增删查改操作。订单管理包括两部分。一部分是可回复的审核功能。另一部分是状态管理，用来修改订单具体状态。

**2.1.2 非功能性需求分析**

（一）易用性

本系统全部功能需划分明确、统一，操作简单明了，保证用户在进入系统后有良好体验，使用户无需额外引导即可按照习惯进行操作，顺利完成整个业务流程。在出问题时系统也会弹出简洁的提示或是跳转页面，整体而言具有较好的交互性。

（二）性能

从微信小程序页面响应的角度，用户端向服务端发送请求后，服务端处理请求并返回数据的时间不超过3秒，用户端接收到数据到完成加载的时间也不超过3秒。从后台管理网页响应的角度，在上传信息时，响应时间也不超过3秒。

（三）易扩展性

本系统的各功能模块相互独立，耦合度低，若未来遇到需求上的改变，可迅速对已有的部分功能改变，而不影响其它功能的使用。

**2.2 可行性分析**

**2.2.1 技术可行性**

软件方面，本文所实现系统使用的是目前使用最广泛的微信小程序。选用的语言主要是Java。作为一个当前非常成熟的编程语言，Java具有相当完备的技术参考文献和基数庞大的技术论坛规模，保证本系统开发的效率和开发过程的稳定性。在数据库方面，使用了轻量级的MySQL。比起Oracle数据库，MySQL对系统计算资源的需求较低，为系统提供了坚实的数据库支持。系统开发时使用的工具为eclipse和HBuilderX，二者均为稳定性较高的开发工具，具有高实用性和高使用度。总体而言，在软件方面，本系统的开发满足可行性需求。

硬件方面，本系统仅要求用户手机能够运行微信小程序即可。根据微信官网可知微信小程序适应的最低配置为四核1.5GHz CPU，内存1GB RAM+4GB ROM，安卓5.0及以上。考虑到目前手机版本迭代快，整体配置逐渐增高，本系统在硬件方面也是可行的。

**2.2.2 经济可行性**

本文所述系统是通过当前广泛使用的微信小程序平台及互联网技术实现的，前端、后端代码所使用的框架和MySQL数据库皆为免费开源内容，无需增加额外成本。开发前期阶段查阅文献资料所需经费由学校承担，开发过程中所购教程与书籍所需金额完全在开发人员承担范围内，开发使用的工具如eclipse、HBuilderX和微信小程序开发者工具均有免费版本。

微信小程序在开发完成并调试成功后，很少出现运行不稳定或出错的情况，因此维护成本远低于其他应用程序。而使用微信小程序最大的好处也是在引流时，可以使用微信平台提供的公众号、视频号和朋友圈等原生渠道，也无需增加其他成本。因此，本系统在经济上的可行性也具有保障。

**2.2.3 操作可行性**

本文实现的点餐系统在前端使用Uni-app提供的组件，严格遵照微信小程序界面简洁、易使用的原则进行布局，界面合理，无冗余部分。在操作方面，整体逻辑与现实点餐一致，功能明确，数据传输可靠，具有较高的操作可行性。

**2.3 关键技术**

**2.3.1 SSM框架**

SSM作为当今最为流行的框架，是Spring+ Spring MVC+ MyBatis的组合。

Spring是由Java EE开发完成的技术性框架，核心功能是AOP（面向切面）和IoC（控制反转）。可以实现将Java Bean对象移交至核心容器完成处理的工作，开发人员只需从容器里调用对象。

Spring MVC是由Spring提供的框架，用于Web应用的开发，使用的是典型的mvc架构（模型、视图和控制器）[5]。该框架将开发分为输入、输出和处理三个部分，利用DispatcherServlet作为 Web 应用的最高控制器，所有的HTTP请求都会根据Spring MVC提供的HandlerMapping分发到对应的Controller接口。Controller完成处理后会将数据返回给DispatcherServlet的前端控制器，最终显示在前端页面上。

MyBatis是一个良好的、高效的框架。其最广为人知的对JDBC的封装是通过底层Java的映射与反射实现的。开发人员需要配置Dao层（也称Mapper层）的接口和XML配置文件来进行数据库相关的操作[6]。

**2.3.2 MySQL数据库**

数据库可谓是系统的核心。其中，在开发人员之间最为流行的关系型数据库非MySQL莫属。其优点包括但不仅限于免费、体积小、支持高数据量的处理，且适用于如今最常用的数据库操作语言即标准SQL语言[7]。

**2.3.3 微信小程序**

微信小程序是由微信提供的一种无需下载即可使用的应用。选用以MINA（基于Java的网络通信应用框架）为基础的开发框架。其结构可简单分为逻辑层与视图层两部分，且逻辑层和视图层能够保证数据同步[8]。即在更新逻辑层内容时，视图层有自动响应的能力。

微信小程序常用的目录结构也不复杂，根目录通常包括utils目录和pages目录，分别存放工具类和各个详细页面，同时也包含三个必须放在根目录的主题文件，如图2-1所示。

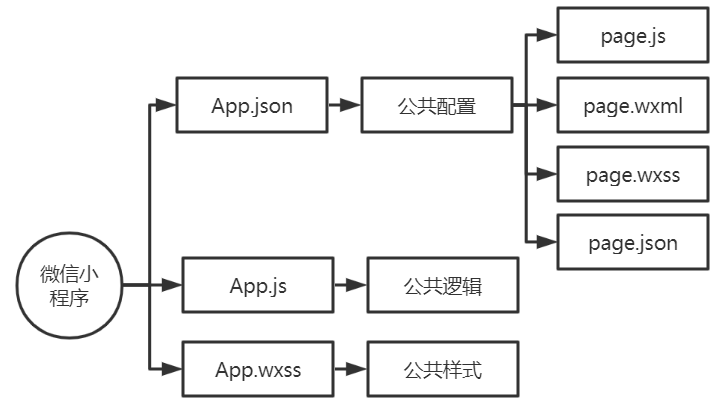


图2-1 微信小程序主体文件图

App.json负责小程序的公共配置。如小程序页面组成、页面跳转以及tab栏的设置。

App.js负责小程序的公共逻辑。包含生命周期的设置、各类事件处理函数和全局变量。

App.wxss负责小程序的公共样式。每个页面都可以调用公共样式实现公共布局的部分，也可以在pages目录中重写代码实现差异化布局的内容。

典型的Pages目录中会包含如图2-1所示的四个文件。其中page.js与page.json属于逻辑层，分别负责页面逻辑与页面配置；page.wxml与page.wxss则属于视图层，分别指定页面的元素结构和页面的具体样式[9]。

**2.3.4 Uni-app框架**

Uni-app支持代码跨平台运行。

若使用Uni-app进行代码开发，开发人员可实现完成一套代码即可发布到不同平台的小程序中，如微信、支付宝、百度和钉钉[10]。

即使不进行跨平台开发，Uni-app也是一款非常便利的开发框架，优点如下：

1. 开发工具HBuilderX自带Uni-app开发环境，无需额外配置环境。
2. 学习成本低，页面文件（数据绑定、事件处理等）遵从Vue规范，在接口能力与组件使用方面则与微信小程序原生开发的规范相同，易于上手。
3. 减少冗余工作，页面仅需完成page.vue文件，编译后会自动生成微信原生开发的page.js、page.json、page.wxml和page.wxss文件。

**第3章 系统设计**

**3.1 系统架构设计**

本食堂点餐系统采用了C/S架构模式，包括后端与前端两大部分。系统前端包括用户进行食堂点餐的微信小程序页面和食堂管理员进行订单管理的网页页面，与用户进行直接交互。系统后端负责接收前端传来的请求，完成对应的逻辑功能、进行相应独立计算后，返回用户请求的数据到客户端。

**3.2 系统模块设计**

本系统由两大模块组成：用户端和食堂管理端，以用户端为主，食堂管理端为辅。模块图如图3-1所示。

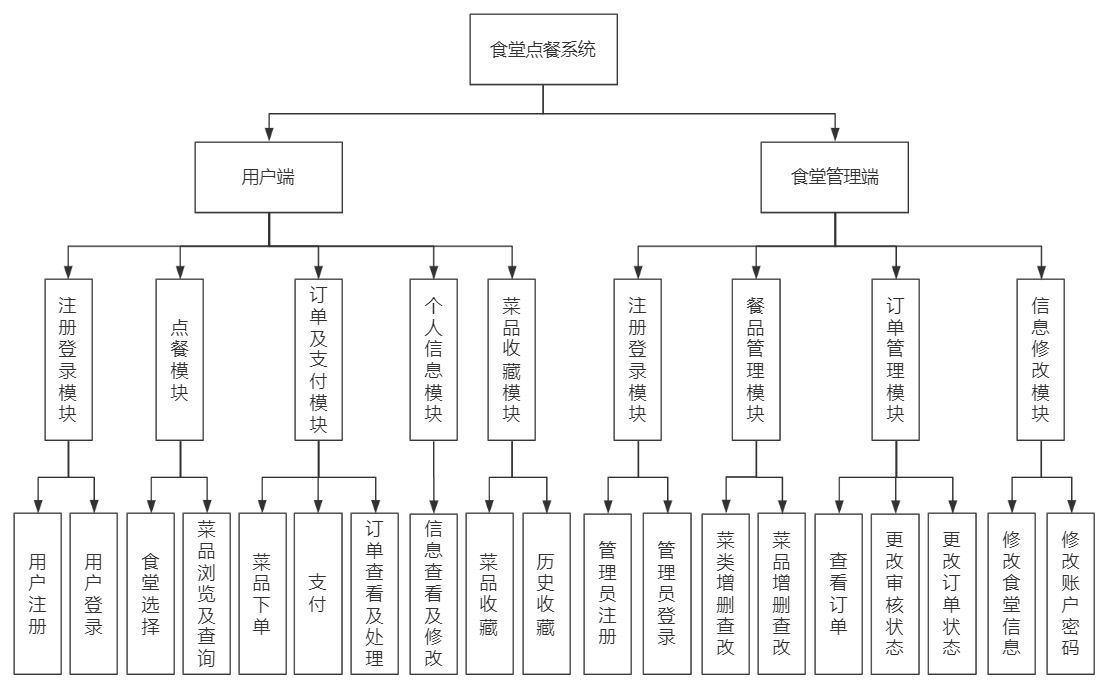


图3-1 系统模块设计图

对于小程序端，注册登录模块是后续功能的基础。用户的点餐、收藏、和修改个人信息操作都以登录为前提。点餐模块是本系统的核心模块，包括选择食堂和正式点餐两部分，点餐部分包括菜品的浏览、查询及选择。订单及订单支付模块必须在用户完成点餐后进行。最主要的功能是查看订单后续状态。个人信息模块独立于点餐模块和订单模块，用户在登录后即可进入个人信息模块，其功能包括查看或修改用户名、密码、联系方式、头像等。菜品收藏模块在点餐模块的浏览菜品子功能上进行延伸。

对于食堂管理端，注册登录模块同样使用账号密码的方式实现用户的注册和登录功能，其他功能均需要在用户已登录的情况下实现。菜品管理模块涉及菜类管理和具体菜品管理模块，分别实现对二者的增删查改功能。订单管理模块包括两部分的修改。首先完成审核状态的修改。订单审核通过后，即可对具体状态进行修改。信息修改模块则包含食堂信息修改和用户信息修改两部分，独立于菜品管理模块和订单管理模块。

在完成上述对系统功能的划分后，各个模块相对独立，逻辑清晰，在系统开发过程中降低了开发的难度、提升了开发的效率，同时方便后续对各个模块进行独立测试。

**3.3 数据库设计**

根据本系统的模块划分，易知本系统的主要模块均与数据库的存储有密切关系。因此，本文在明确系统数据之间的具体联系后，对系统的数据库完成相关设计。

**3.3.1 概念结构设计**

在概念结构设计部分，本文以系统功能为依据，以系统相关的实体为对象进行分析[11]，构建出E-R模型（实体联系模型）。

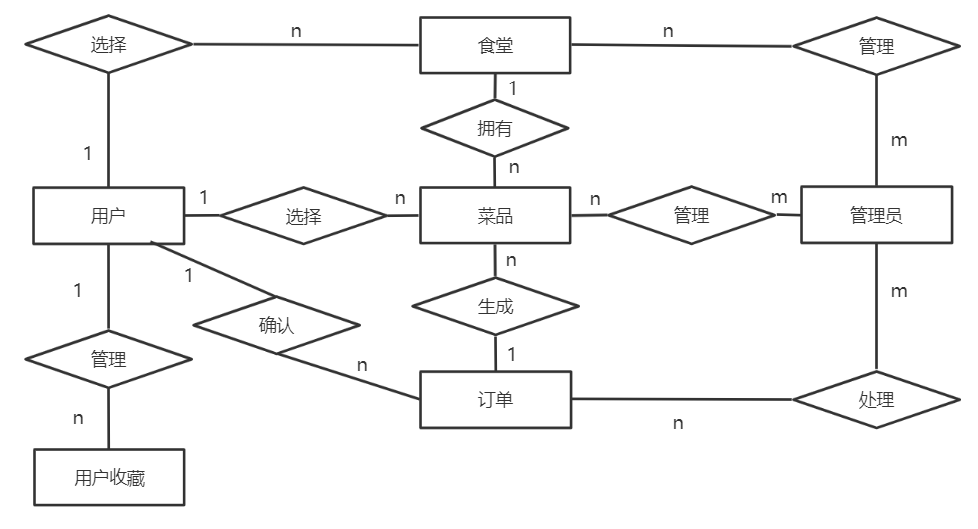


图3-2 系统模块设计图

在本系统中，实体设置为：管理员、用户、食堂、菜品、订单和用户收藏。六个实体各有不同的属性，实体间也有不同的关系。

用户指点餐用户，根据每个点餐用户可选择多个食堂进行点餐的逻辑关系，可知用户与食堂之间为一对多的关系。用户在选择食堂后，可以对食堂内的多种菜品进行选择，因此用户与菜品、食堂与菜品间均为一对多的关系。每名用户可确认多个订单进行下单，每个订单不可由多名用户进行确认。而每个订单都可包括一个或多个用户选择的菜品，因此用户与订单间、订单与菜品间均为一对多的关系。同时，用户可收藏多种菜品并对不同的收藏进行管理，可知用户与用户收藏之间也是一对多关系。管理员指食堂管理员，可对一个或多个食堂进行管理，而每个食堂也可以由多个管理员进行管理，因此二者为多对多的关系。管理员可管理多个菜品，每个菜品可被不同管理员进行管理，所以管理员与菜品间也是多对多的关系。同理，用户的订单可以被不同管理员进行审核和处理，同一管理员可处理不同的用户订单，为多对多。

本小节利用E-R图对现实生活中的实体及其之间的联系进行了基础的抽象，并描述了实体、联系和属性三要素中的前两要素，关于实体型的属性将在数据库的逻辑结构设计中具体体现。

**3.3.2 逻辑结构设计**

本系统基于MySQL 5.7来进行数据存储。

根据数据表与实体型一一对应的原则，本系统包括的实体表有：管理员表、用户表、食堂表、菜品表、订单表、用户收藏表、token表。

实现多对多关系的联系表在此不做赘述，下面仅对实体表的设计进行描述：

1. 管理员表设计如下表所示：

表3-1 管理员表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 允许为空 | 是否主键 | 备注 |
| 1 | Gid | bigint | 否 | 是 | 管理员id |
| 2 | username | varchar | 否 | 否 | 管理员名 |
| 3 | password | varchar | 否 | 否 | 管理员密码 |
| 4 | role | varchar | 否 | 否 | 角色 |
| 5 | addtime | timestamp | 否 | 否 | 新增时间 |

管理员表用来存储管理员的相关信息。其中id互不相同，作为主键。其他字段均不允许为空。

1. 用户表设计如下表所示：

表3-2 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 允许为空 | 是否主键 | 备注 |
| 1 | id | bigint | 否 | 是 | 学生id |
| 2 | Spassword | varchar | 否 | 否 | 密码 |
| 3 | Sname | varchar | 否 | 否 | 姓名 |
| 4 | Sgender | varchar | 是 | 否 | 性别 |
| 5 | Sphone | varchar | 否 | 否 | 手机号 |
| 6 | Spic | varchar | 是 | 否 | 头像图地址 |
| 7 | Saddtime | timestamp | 否 | 否 | 新增时间 |

用户表用于存储用户的信息。用户id互不相同，作为主键。其中，头像图地址既可以为网址，也可为静态资源内的地址作默认头像。

1. 食堂表设计如下表所示：

表3-3 食堂表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 允许为空 | 是否主键 | 备注 |
| 1 | STid | bigint | 否 | 是 | 食堂id |
| 2 | STnum | bigint | 否 | 否 | 食堂编号 |
| 3 | STarea | varchar | 否 | 否 | 食堂地址 |
| 4 | STname | varchar | 否 | 否 | 食堂名称 |
| 5 | STphone | varchar | 否 | 否 | 联系电话 |
| 6 | STtype | varchar | 否 | 否 | 卫生等级 |

食堂表用来存储食堂的相关信息，食堂id为主键。其中，食堂编号不可改变，为自增的固定编号。食堂名称也互不相同，作为用户在微信小程序端检索菜品的标准之一。

1. 菜品表设计如下表所示：

表3-4 菜品表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 允许为空 | 是否主键 | 备注 |
| 1 | Cid | bigint | 否 | 是 | 菜品id |
| 2 | STnum | bigint | 否 | 否 | 食堂编号 |
| 3 | STname | varchar | 否 | 否 | 食堂名 |
| 4 | Cname | varchar | 否 | 否 | 菜品名 |
| 5 | Ctype | varchar | 否 | 否 | 菜类 |
| 6 | Cweight | varchar | 否 | 否 | 菜品份量 |
| 7 | Cpic | varchar | 否 | 否 | 图片地址 |
| 8 | Cprice | decimal | 否 | 否 | 菜品价格 |
| 9 | Cstock | int | 否 | 否 | 菜品库存 |
| 10 | Cdesc | longtext | 是 | 否 | 菜品描述 |
| 11 | thumbnum | bigint | 是 | 否 | 点赞数 |
| 12 | dislikenum | bigint | 是 | 否 | 点踩数 |

菜品表用来存储菜品的相关信息，菜品id为主键。通过食堂编号和食堂名实现菜品与食堂之间的从属关系。本系统不设置单独的菜类表，仅在菜品表内设置菜类字段。除菜品描述、点赞数和点踩数外，其他字段皆不可为空。

1. 订单表设计如下表所示：

表3-5 订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 允许为空 | 是否主键 | 备注 |
| 1 | Oid | bigint | 否 | 是 | 订单id |
| 2 | addtime | timestamp | 否 | 否 | 新增时间 |
| 3 | Onum | bigint | 否 | 否 | 订单编号 |
| 4 | STnum | bigint | 否 | 否 | 食堂号 |
| 5 | STname | varchar | 否 | 否 | 食堂名 |
| 6 | Cname | varchar | 否 | 否 | 菜品名 |
| 7 | Cnumber | int | 否 | 否 | 菜品数量 |
| 8 | Cpic | varchar | 否 | 否 | 图片地址 |
| 9 | price | decimal | 否 | 否 | 金额 |
| 10 | Sid | bigint | 否 | 否 | 用户id |
| 11 | VerifyType | varchar | 否 | 否 | 审核状态 |
| 12 | feedback | varchar | 是 | 否 | 审核回复 |
| 13 | Pstatus | varchar | 否 | 否 | 支付状态 |
| 14 | Ostatus | varchar | 否 | 否 | 取餐状态 |

订单表用来存储订单的相关信息。审核状态存储订单当前的审核状态，所有订单在生成后审核状态默认为0，即未审核，管理员可设置审核状态为1审核通过或2审核不通过。审核状态不可为空，但审核回复可为空。

1. 用户收藏表设计如下表所示：

表3-6 用户收藏表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 允许为空 | 是否主键 | 备注 |
| 1 | CollectID | bigint | 否 | 是 | 收藏id |
| 2 | Sid | bigint | 否 | 否 | 用户id |
| 3 | Cname | varchar | 否 | 否 | 菜品名 |
| 4 | Cid | bigint | 否 | 否 | 菜品id |

用户收藏表用来存储用户收藏菜品的相关信息，每个收藏都有不同的收藏id，作为主键。通过用户id、菜品id分别与用户表、菜品表建立关系。

1. Token表设计如下表所示：

表3-7 token表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 允许为空 | 是否主键 | 备注 |
| 1 | Tid | bigint | 否 | 是 | Token id |
| 2 | Sid | bigint | 否 | 否 | 用户id |
| 3 | Sname | bigint | 否 | 否 | 用户名 |
| 4 | token | bigint | 否 | 否 | Token参数 |
| 5 | addtime | timestamp | 否 | 否 | 新增时间 |
| 6 | expiratetime | timestamp | 否 | 否 | 过期时间 |

Token表的作用是辅助java过滤器判定用户是否登录以及登录是否过期，以Token id为主键。其中token参数和新增时间由微信小程序端获取并存储，开发时设置的失效时长加上新增时间则为过期时间，存储入数据库中。

**3.4 详细设计**

前述三小节分别对系统架构、系统模块和数据库的设计进行了分析与解释，本节将对系统的重要功能进行详细分析，并结合流程图阐述相关的思路。

**3.4.1 用户点餐功能设计**

用户点餐功能是本系统最重要的功能之一。

用户在完成注册并成功登录后，会跳转至首页，点击小程序下方固定导航栏中的“食堂菜品”按钮后，跳转至正式的点餐界面。在点餐界面中，若用户在搜索栏里搜索特定的食堂，则以卡片形式显示该食堂内的菜品概览图及菜品名；若用户不指定特定的食堂，则以卡片形式显示全部食堂的全部菜品概览图及菜品名。

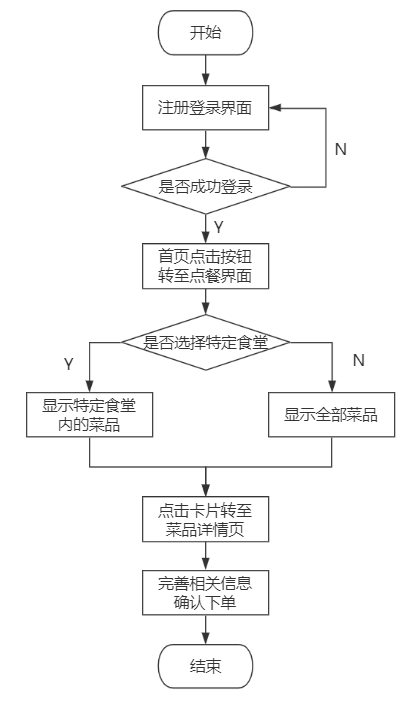


图3-2 用户点餐功能流程图

用户点击卡片跳转至菜品详情页，详情页将具体展示菜品的名称、份量、价格、描述信息、点赞数和点踩数。其中，点击点赞或点踩按钮都可实时改变点赞数或点踩数。用户点击右下角“订餐”按钮后，将显示具体订餐信息页面，需完成填写菜品数和取餐时间（现取或指定时间）后确认下单，完成整个点餐流程。如图3-2。

**3.4.2 用户订单功能设计**

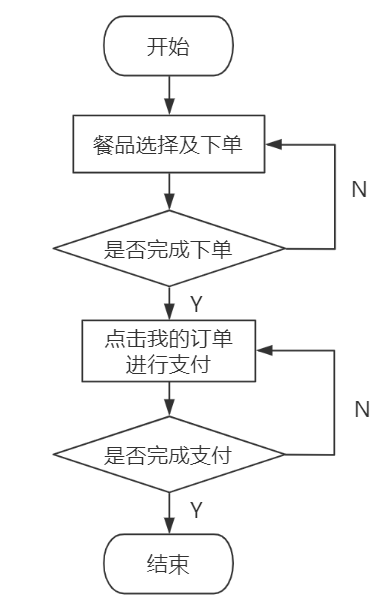
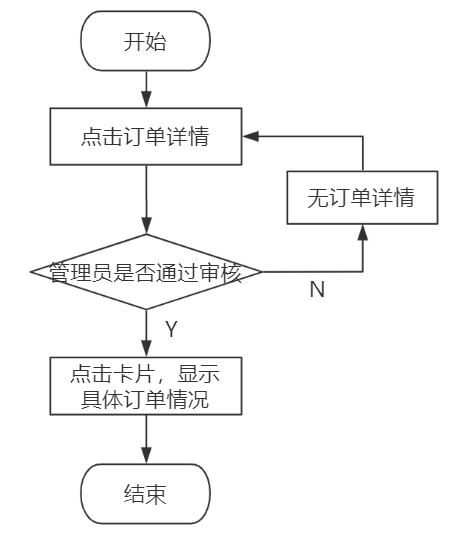
 

图3-3 用户订单功能流程图1 图3-4 用户订单功能流程图2

用户订单功能也在本系统中起到关键作用，其存在目的是便于用户查看订单内容及订单状态，未支付订单将在“我的订单”栏中显示，已支付且已通过审核订单在“订单状态”栏中显示。

该功能以用户已登录且已完成下单步骤为前提，否则“我的订单”部分显示为空。用户完成支付且管理员通过审核后，“订单状态”栏将以卡片形式显示当前订单，点击卡片后显示订单详情。

需注意，根据必须通过审核才有订单状态的逻辑，订单审核情况仅在“我的订单”中的详情情况里显示，不在“订单状态”的详情情况显示。

**3.4.3 用户收藏功能设计**

用户收藏功能具体分为添加收藏和管理收藏两部分，入口分别为菜品详情页和个人收藏页。

点击“我的”按钮跳转至个人界面，再点击“个人收藏”按钮即可看到以卡片形式显示的已收藏菜品；若无收藏，则显示为空。点击卡片后跳转至菜品详情信息页，可直接进行下单。在“个人收藏”页面，可点击垃圾桶按钮删除收藏。流程图如图3-5所示。

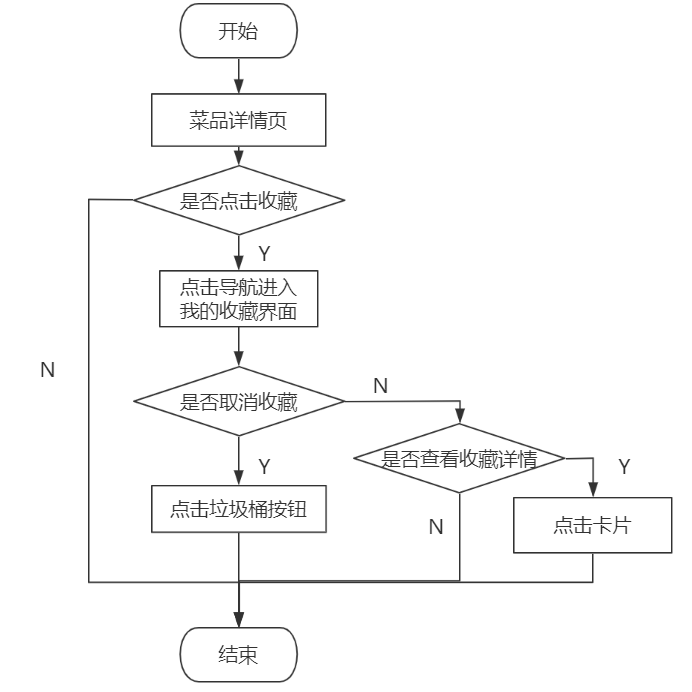


图3-5 用户点餐功能流程图

**3.4.4 管理员订单功能设计**

本文实现的系统以用户使用的微信小程序端为主，管理员使用的后台网页为辅，但管理员对订单的相关操作仍是系统中重要的一部分。以3.4.2节所述用户订单功能为例，只有管理员将订单审核状态改为已审核，订单才能够显示在用户界面上，若无管理员的相关操作，将会出现逻辑上的缺失，进而导致设计不完整。

管理员对订单的操作为查看、审核与修改状态。

网页以表单形式向管理员展示用户订单的相关信息，如订单号、订单菜名、菜品数量、订单金额与用户支付状态等。审核订单功能除了设置订单审核通过、审核不通过以外，还包括留言回复功能。更改订单状态功能则是基于下拉选择组件的选取状态功能，包括备餐中、待取餐和已取餐三个状态。管理员订单功能流程图如下。

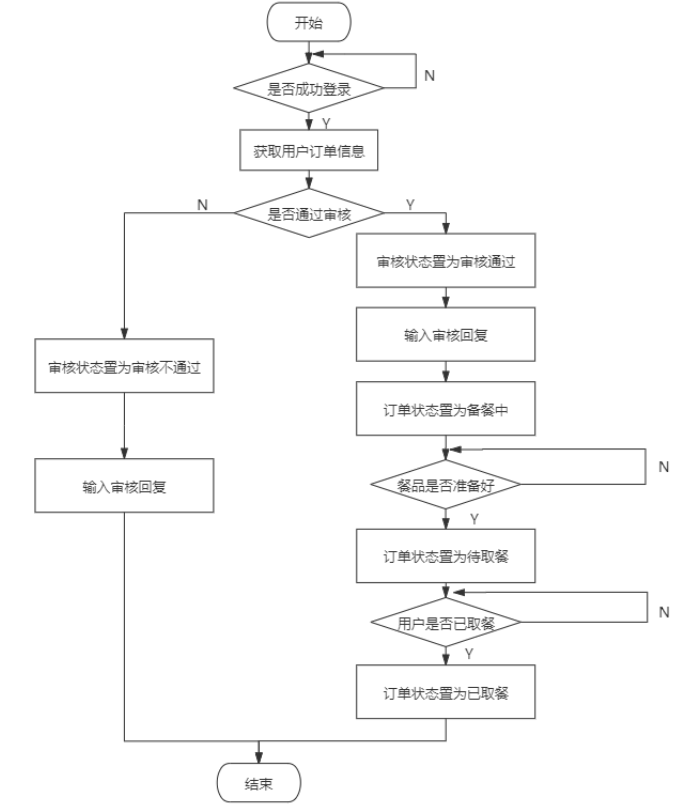


图3-6 订单功能流程图

**第4章 系统实现**

前面的章节已经介绍并分析了系统的需求和可行性，完成了对系统的基础设计。通过上述内容，本系统的主要脉络和实现方法已经明确，下面将挑选本系统的主要功能进行说明，相关运行截图统一于4.7节中展示。

**4.1 连接数据库**

本文设计并实现的功能都以数据库操作为基础。其中，成功连接数据库这一步骤至关重要。本系统采用的数据库为MySQL，后端框架为SSM框架。在创建项目初期，pom.xml（项目对象模型）中添加依赖后即可自动下载数据库连接驱动包等需要用到的jar包[12]，添加MySQL依赖的代码如下：

<!--https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.7.41</version>

</dependency>

之后在config.properties文件中进行如下配置：

jdbc\_url=jdbc:mysql://127.0.0.1:3306/shitangdiancan?useUnicode=true&characterEncoding=UTF-8

jdbc\_username=root

jdbc\_password=123456

其中，jdbc\_username和jdbc\_password分别是数据库账户和数据库密码。在jdbc\_url中，shitangdiancan为数据库名。

添加useUnicode=true&characterEncoding=UTF-8配置的理由是，MySQL数据库采用的是gbk编码模式，而项目使用的是utf-8编码模式，二者并不一致。如此配置后，数据库存数据时会按照utf-8模式把数据拆成字节码，然后使用gbk模式将字节码重新编码，存入数据库中；从数据库中读取数据的过程则相反，先依照gbk模式解码，再按照utf-8模式重新编码成数据，交还给发送了数据请求的客户端。

最后需在spring-mybatis.xml中对连接池等进行设置。

具体配置如下：

<bean name="dataSource" class="com.alibaba.druid.pool.DruidData

Source" init-method="init" destroy-method="close">

<property name="url" value="${jdbc\_url}"/>

<property name="username" value="${jdbc\_username}"/>

<property name="password" value="${jdbc\_password}"/>

<!-- 初始化连接大小 -->

<property name="initialSize" value="0"/>

<!-- 连接池最大使用连接数量 -->

<property name="maxActive" value="20"/>

<!-- 连接池最大空闲 -->

<property name="maxIdle" value="20"/>

<!-- 连接池最小空闲 -->

<property name="minIdle" value="0"/>

</bean>

此处使用了阿里巴巴公司开源的Druid数据库连接池，避免因频繁建立和销毁连接引起的时间浪费和性能开销，保证创建连接的速度和连接对象的使用率[14]。

**4.2 数据库与持久层实现**

本系统实现的根本是数据库的CRUD操作，即增加(Create)、检索(Retrieve)、更新(Update)和删除(Delete)。本系统使用的SSM架构中，MyBatis专门用于与数据库相关的持久层操作。

首先需要完成实体类的建立，实体型的每一个属性都与实体类中的成员数据一一对应。以用户表User表为例，表中有id字段，那么对应的实体类文件UserEntity.java文件中，public class User类里必须有public int id且类型与数据库设计的字段类型一致。而为了实现具体的读取和更新功能，User类里也必须有相应的get和set方法，如：

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

之后需要完成的是Mapper层。该层也称Dao层，是由MyBatis生成的提供数据库数据持久化操作的部分。实体层（Model层）并不与数据库直接交互，因此开发时需要在mapper文件夹内使用MyBatis提供的XML映射器对SQL语句进行映射，即需要完成SQL映射文件。

用户表相关的部分代码如下。

<mapper namespace="com.dao.UserDao">

<!-- 删除用户 -->

<delete id="deleteUser" resultType="com.entity.UserEntity">

delete from user where id = #{id}

</delete>

<!-- 根据id获取用户信息 -->

<select id="findUserById" parameterType="int" resultType

=" com.entity.UserEntity">

select \* from user where id = #{id}

</select>

</mapper>

上述删除映射中，额外使用了resultType来设置从这条语句中返回结果的类全限定名。整体语言规范遵从MyBatis官方文档，本系统的SQL映射文件中使用到的几个顶级元素为insert映射插入语句、update映射更新语句、select映射查询语句和delete映射删除语句。

除此以外，仍需配置Dao层的接口方法，也就是对应的Dao.java文件。以User表为例，一共需要完成UserEntity.java实体类，UserDao.xml映射文件和UserDao.java接口三部分。

对于Dao层的接口文件，需保证方法名与xml文件中定义的id相同，输入或输出的参数类型也与xml文件中定义的一致。

仍以User表为例，UserDao.java的部分代码如下：

public interface UserDao {

public void addUser(User user);

public void deleteUser(int id);

public void updateUser(User user);

public User findUserById(int id);

}

其中，deleteUser与UserDao.xml中delete标签内的id名完全一致，findUserById也与xml文件中select标签内的id名一致。

**4.3 自定义API接口**

微信小程序端功能的实现离不开对数据库的操作，但如果在每个需要请求数据的页面文件中都添加request等请求，最终代码将非常繁琐。因此，本文实现的系统采用自定义API接口的方式，将页面与服务器进行信息交互的操作写在api文件夹内的base.js，http.js和index.js文件中，完成接口的封装并export，最后在main.js文件内进行接口挂载。之后，在需要进行数据交互的文件内，可在文件首import接口后，通过vue对象调用接口方法，完成请求。

**4.3.1 base.js**

在三个接口文件中，base.js完成的是最基础的url请求，将地址赋给url，代码如下：

const base = {

url : "http://localhost:8080/shitangdiancan/",

}

export default base

**4.3.2 http.js**

在http.js中，代码实现通用的Uni-app网络请求，首先import base from './base'来方便调用url，之后完成request方法，对url、请求方式（GET, POST, PUT, DELETE）、成功时调用的方法和失败时调用的方法进行定义。

以如下判断请求是否成功的代码为例。若成功则传递数据至客户端，若失败则判断权限后跳转页面并跳出提示框：

if (statusCode === 200) {

// http协议，200表明该请求被成功地完成，所请求的资源发送到客户端[15]

var rs = response.data;

if (rs.code === 0) {

// 请求成功后返回数据

resolve(response.data);

}

else if (rs.code == 401) {

// 401表示请求未授权也就是还没有登录，跳转到登录界面

uni.navigateTo({

url: '../login/login'})

} else {

uni.showToast({

// showToast显示消息提示框

title: rs.msg,

icon: 'none',

duration: 2000 //持续时间

});

}

} else {

uni.showToast({

title: "接口执行异常",

icon: 'none',

duration: 2000

});

}

}

**4.3.3 index.js**

在index.js中完成详细功能请求的封装，因为需要用到base.js和http.js内的数据或方法，也要在文件首部添加import http from './http'和import base from './base'代码。代码多出采用ES6标准新增的箭头函数。以登录校验和查询功能的代码为例：

export const auth = () => {

let token = uni.getStorageSync("token");

// 从缓存获取token，未登录前token取到是空

if (!uni.getStorageSync("token")) {

uni.navigateTo({ //跳转页面

url: '../login/login'

})

return

}

}

export const page = (tableName, data) => {

return http.request({

url: `${tableName}/page`,

method: 'GET',

data

});

}

其中，请求方式有以下四种。

GET为请求指定的资源。

POST为提交数据给指定资源。可能会导致新资源的创建或已有资源的更改。

PUT为上传新信息到指定的服务器资源位置上，不会改变数据种类等信息。

DELETE为请求删除服务器所标识的资源。

上述查询功能仅需向服务器请求资源，无需对资源进行修改或删除，因此使用的是GET方式。此外，本系统中实现创建和更新功能主要使用的是POST方式和PUT方式。

**4.4 用户点餐及收藏功能**

本节将展示用户点餐功能的界面图，并在对基于Uni-app的页面文件结构进行说明后，对本功能的实现进行解释说明。

用户在打开微信小程序并完成登录后，可在小程序下方固定导航栏中点击“食堂菜品”进入点餐界面，菜品以卡片形式显示在界面上，参考4.7节展示的图4-1。点击卡片可进入菜品详情页，点击订餐后即可进入点餐详情页，如图4-2所示。菜品详情页由图4-3展示，用户收藏页面由图4-4所示。

根据Uni-app官方文档展示的使用原则，每个页面都应在/pages文件夹下有单独文件夹，文件夹内为对应Vue文件。如，菜品浏览界面由/pages/shitangdiancan下的list.vue文件实现。

Vue文件分为template，script和style三个组成部分。

在template中，以Vue的语法和微信小程序的标签形式使用组件。

如4.7节图4-1所示，顶栏标题和下方固定导航tabbar组件为小程序内置的多数页面公用的固定组件，在根目录下进行定义即可，而搜索栏、标签选择组件和卡片显示组件则需要在当前Vue文件中进行描述。

在script中，通过数据绑定的方式和template共同完成组件的功能，在template中使用的方法和以插值表达式方式使用的变量都应在script内进行定义和描述，与后端数据的交互，比如调用自定义的API接口，也在script内完成。

在style中则仅对样式进行描述，为纯前端内容。

以食堂菜品详情页为例，前后端之间的数据流如下：

在详情页面对应的detail.vue文件中，template描述列表组件内显示{{detail.caipin}}数据，script通过调用4.3节所述的API接口向服务器提交GET参数的请求，后端SSM架构的Controller层接收请求，Controller层通过@Autowired注解自动装配，向Service层请求视图。Service层完成相应的逻辑判断（此处为查询功能因此仅需传参即可），部分serviceImpl.java代码如下： @Override

public Caipin queryCaipinByCid(int Cid) {

//根据菜品id检索,是数据库内主键

Caipin caipin = caipinDao.queryCaipinByCid (Cid);

return caipin;

}

之后再调用Dao层的java文件。根据4.2节所述，Dao层的java文件是xml文件中的操作按id映射成的方法。数据库根据对应xml文件中的操作，将菜品详情即菜品表中的数据返回给详情页，至此，全部步骤已完成，运行截图参考4.7节中的图4-3食堂菜品详情界面图。

本系统的其他功能的实现均与上述过程大同小异，如图4-2所示点餐详情页，在总体流程上与菜品详情的流程并无不同，差异仅在于在vue文件中使用了不同的组件，script调用的请求为POST请求，数据库操作是insert而非select等。

**4.5 用户订单功能**

根据4.4节所述，用户提交订单后，会在订单表中插入一条订单记录。订单表是用户订单功能实现的基础，微信小程序端“我的订单”和“订单状态”界面的数据请求均是向订单表查询数据。

订单界面部分截图参考4.7节图4-5和图4-6。

图4-5所示界面的orderdetail.vue文件使用了Uni-app提供的swiper组件，即滑块容器视图，实现了轮播图模块。

轮播图模块是微信小程序开发中最常用的模块之一，通常利用v-for迭代数组来实现。在此简述其参数作用及数据绑定方式：在template的swiper标签内，indicator-dots表示是否显示原点指示器，circular表示是否循环滑动，interval为轮播的间隔。部分代码如下：

<swiper-item v-for="(item,index) in swipers" :key="index">

<image :src="item"></image>

</swiper-item>

同样，v-for所需数组在script中有定义this.swipers=[]。但由于本系统设计的数据库中，每个菜品的图片仅有一张，此处的轮播图仅为单图无限循环。

**4.6 管理员订单功能**

管理员订单管理界面图参考4.7节所示图4-7。

管理员网页的后端部分与微信小程序端使用的是同一套后端代码，仅在View层有所不同。微信小程序使用的是Uni-app实现前端内容，管理员网页使用的是Vue，由于Uni-app是使用Vue.js开发的架构，二者核心思想均为组件化和数据驱动，差距仅为标签使用上的不同，如微信小程序端使用的是<view>标签，而非<div>。

此处主要使用的组件为输入框和选择栏selector组件。以选择栏为例，在methods中绑定事件，在选择内容时触发selectChange(item)，对value进行赋值 this.value = item.value，并在用户点击提交后，将整体内容存放在订单对象中，向服务器发送POST请求。通过后端Controller层映射到Service层中。之后，调用Dao层中的方法，从而完成对数据库内容的更新。若取消，则不对服务器发送更新请求，直接跳转页面回到上一页面。

**4.7 系统截图**

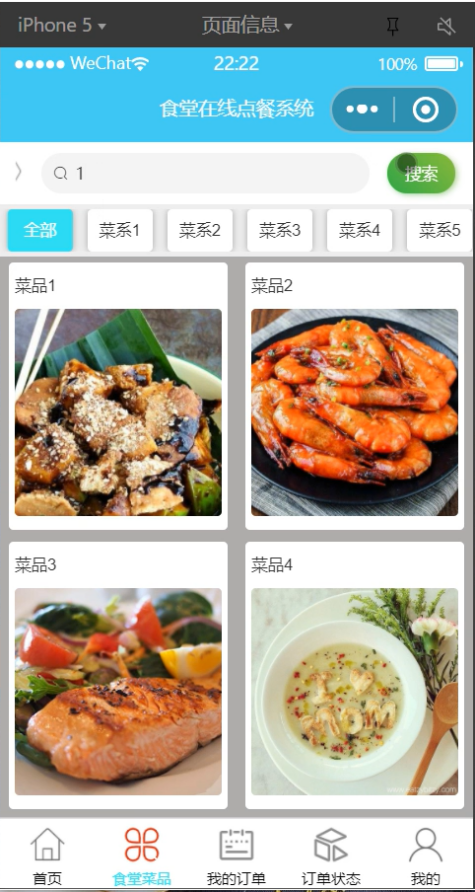
 

图4-1 用户点餐功能界面图 图4-2 用户点餐详情界面图

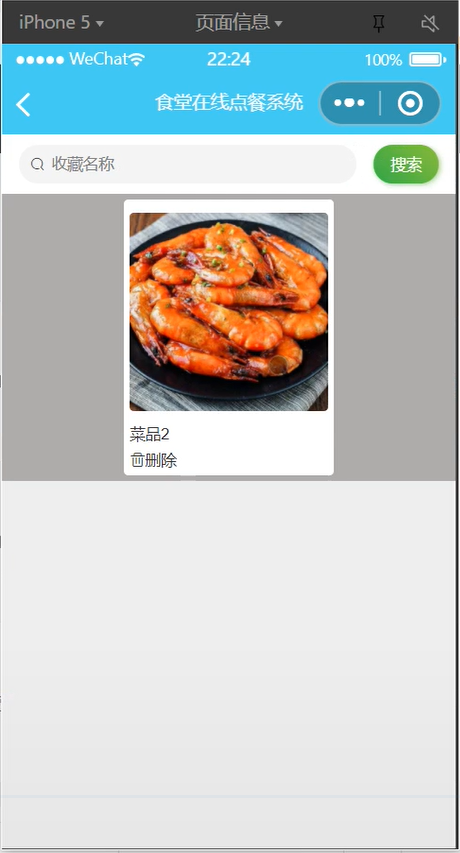
 

图4-3 食堂菜品界面图 图4-4 个人收藏界面图

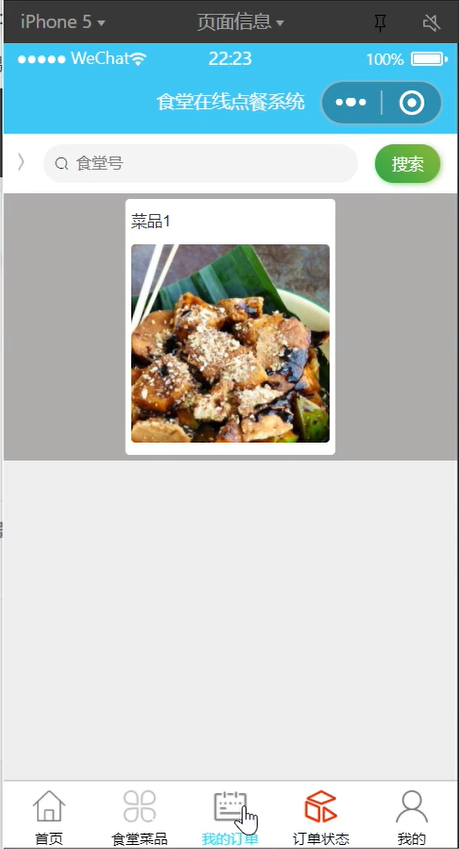
 

图4-5 订单状态浏览界面图 图4-6 订单状态详情界面图

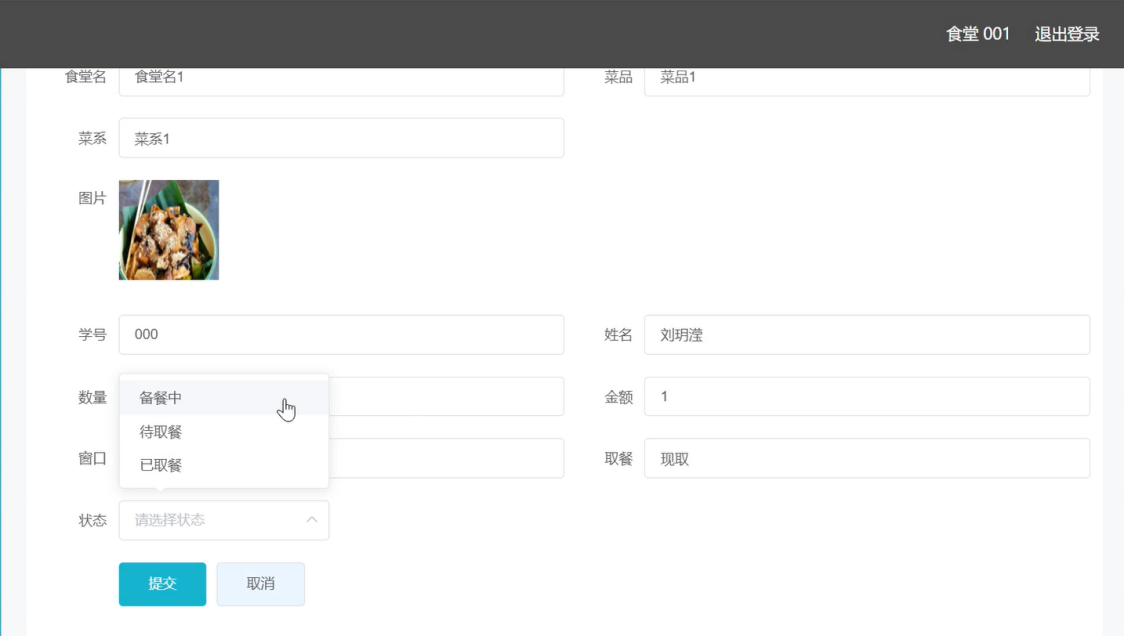


图4-7 管理员端订单管理界面图

**第5章 总结与展望**

本文设计并实现了一个食堂点餐微信小程序。系统的后端采用SSM框架，微信小程序端采用Uni-app框架，结合MySQL数据库提供的数据支持，共同完成了食堂点餐系统的功能。本系统在概念和实现上有以下特点：

（1）与到食堂排队点餐的方式不同，用户可根据个人情况备注取餐时间，食堂管理员的审核回复功能则保证用户可及时了解情况，保证点餐效率最大化。

（2）采用支持跨平台开发的Uni-app。比起繁杂的微信小程序的原生开发，代码复用性更佳。除了发布在微信小程序端，还可后续进行条件编译等操作发布到支付宝小程序和其他平台上，提高开发效率。

（3）开发食堂点餐小程序，比加盟第三方平台更适合食堂运营。无需针对不同的手机操作系统进行多套代码的编写，也无需购置额外的硬件设施，同时也避免了第三方平台的抽成，降低了食堂的成本；使用时也无需额外下载软件，符合学生群体的喜好。

但是，本文讨论的食堂点餐系统仍有改进的空间，如用户支付方面由于未能申请到权限仅做了虚拟支付。在未来的时间内，应继续进行多方面的学习和完善，争取最大程度满足用户需求。

**参考文献：**

[1].第46次《中国互联网络发展状况统计报告》发布[J].中国广播,2020(11):54.

[2]赵一燊.社会网络嵌入理论视域下微信小程序的场景建构及其影响研究[D].南京大学,2020.

[3]中国青年报社.网约车外卖创佳绩 Uber盯上新“蛋糕”[EB/OL].(2021-04).https://baijiahao.baidu.com/s?id=1696983663568204746&wfr=spider&for=pc.

[4]王慧媛.扫码点餐系统的设计与分析[J].电子技术与软件工程,2020(23):50-51.

[5]张翼.微信公众平台点餐系统设计与研究[J].电子制作,2015(02):89.

[6]亓雪冬,韩立峰.基于MVC的Web框架设计与应用[J].微型电脑应用,2021,37(03):4-6.

[7]Maven.《MyBatis官方文档》[EB/OL].(2020-10)[2020-10-07]. <https://mybatis.org/mybatis-3/zh/index.html>.

[8]李艳杰.MySQL数据库下存储过程的设计与应用[J].信息技术与信息化,2021(01):96-97.

[9]李哲,周灵.微信小程序的架构与开发浅析[J].福建电脑,2019,35(12):66-69.

[10]刘玉佳.微信“小程序”开发的系统实现及前景分析[J].信息通信,2017(01):260-261.

[11]DCloud.《Uni-app开发者文档》[EB/OL].https://uniapp.dcloud.io/.

[12]王珊,萨师煊.数据库系统概论[M].北京:高等教育出版社,2014.

[13]Yuxiang Hou. The design and implementation of the framework for Spring+SpringMVC+MyBatis in the development of Web application[A]. Institute of Management Science and Industrial Engineering: Computer Science and Electronic Technology International Society,2019:6.

[14]彭影.优化Java数据库访问效率的策略研究[J].计算机产品与流通,2020(02):21.

[15].10种常见的HTTP状态码解析[J].电脑知识与技术(经验技巧),2019(11):87-88.

致 谢

首先，我要特别感谢我的导师李波老师对我的细心指导，帮助我顺利了完成大学期间的最终作业。从对题目的选择到对系统实现方式的推荐；从对系统不足之处的指点再到论文的完成与修改，李波老师都以严谨的态度和专业的水平为我提出了实际意见。还要感谢张宇学姐和桂黎铭学长，在我遇到细节处问题的时候为我提供了他们的经验，并且在我生病期间提供了很大的支持和鼓励。

同时，要感谢计算机学院其他老师四年来对我的培养和指正，让我在大学期间找到努力的方向，指导我完成大学期间所有的学习任务与工作，体验了实习生活中的种种考验，并获得了升学的机会。更要感谢我的父母，供养我读书十余载，成为我最安心的后盾。在此，衷心向各位说声谢谢，大家辛苦了！

最后，再次感谢大家四年来的陪伴，我将以此为起点重新出发，继续努力！