

湖南科技大学计算机科学与工程学院

综合实践实训课程设计报告

专业班级： 19 级计算机七班

姓 名： 杨向羽

学 号： 1901110222

指导教师： 曹步清

时 间： 2021.06.07—2021.06.20

地 点： 逸夫楼 416

指导教师评语：

成绩： 等级：

签名： 年 月 日

目录

一、实验题目	2
1.1、题目简介	2
(1) 题目名称.....	2
(2) 功能要求.....	2
1.2、需求分析	2
(1) 选题背景.....	2
(2) 总体规划.....	4
(3) 项目特色.....	5
二、实验目的	5
三、总体设计	5
3.1、总体架构	5
3.2、技术选型	7
3.3、项目研发步骤.....	7
3.4、小程序功能设计.....	8
四、详细设计	9
4.1、原型设计	9
4.2、数据库设计	14
4.3、流程设计	16
五、实验结果与分析.....	17
六、小结与心得体会.....	21
附源码.....	23

一、实验题目

1.1、题目简介

(1) 题目名称

网上订餐平台

(2) 功能要求

- 1) 门店管理
- 2) 菜品管理
- 3) 用户注册
- 4) 用户点餐、查看修改菜单
- 5) 模拟支付
- 6) 订单管理
- 7) 配送管理

1.2、需求分析

(1) 选题背景

就目前而言，美团、饿了么等相关订餐配送平台相继出现在广大人民群众的视野，与此同时，相关平台取得的成就恰恰反映了网上订餐平台迎合了现代人饮食娱乐的需求。无可厚非，现代人生活节奏快，没有较为充足的精力去认真学烹饪，节省烹饪时间专注于其它领域。

《2020-2021 年中国外卖行业发展研究报告》指出：疫情重塑外卖格局。疫情时期餐饮行业迎来了平台开放时代的全新红利期，加速数字化以及品牌化恰逢其时。而外卖行业的蓬勃发展，正在以时间交换打破空间边界，重塑着城市的隐形商业空间。疫情冲击，外卖用流量逆势增长。

2020 年中国餐饮业遭遇寒冬，全国餐饮收入近十年来首次下滑；
2020 年中国在线外卖市场规模 6646.2 亿元，同比增长 15%；
2020 年中国在线外卖用户规模 4.56 亿人，同比增长 7.8%。

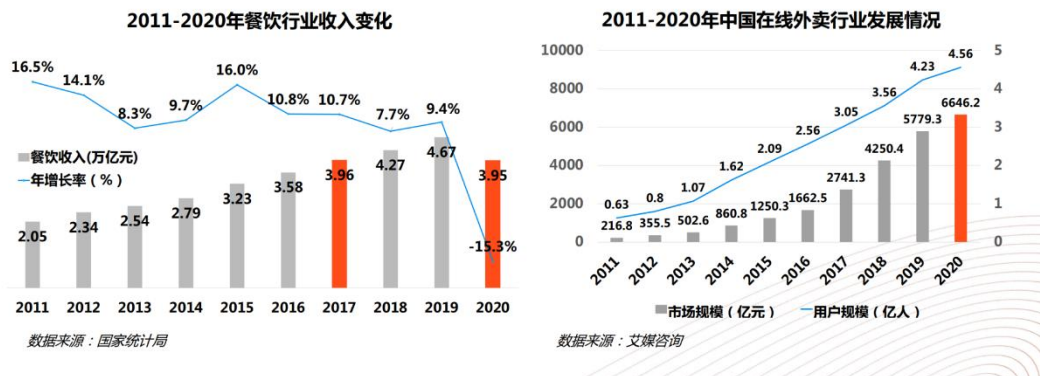


图 1 2011-2020 年餐饮业与在线外卖行业发展图

下沉市场崛起，60、70 用户激增：

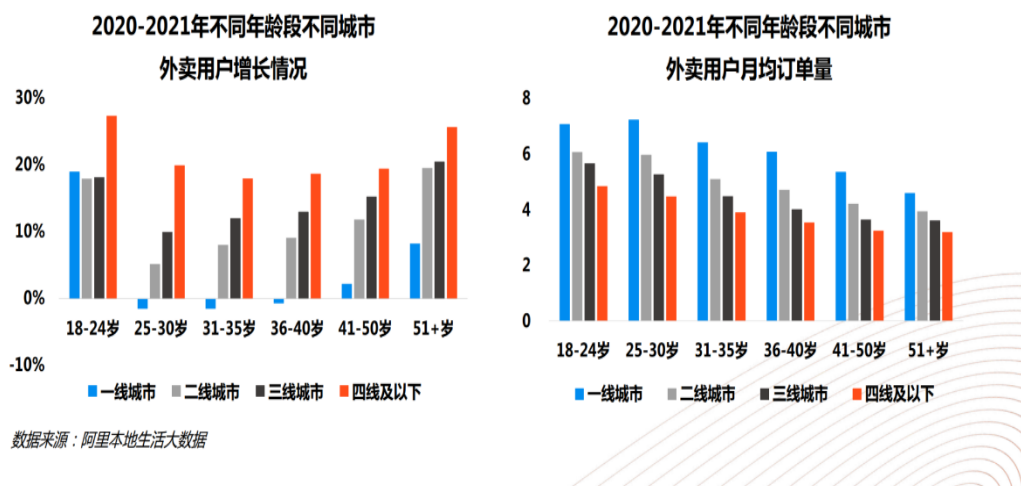


图 2 2011-2020 年在线外卖用户量图

餐饮外卖多样化，非正餐占比上升：

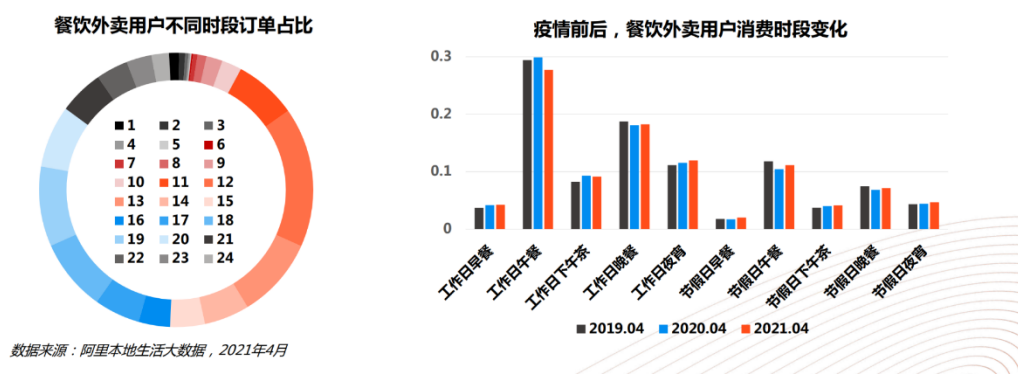
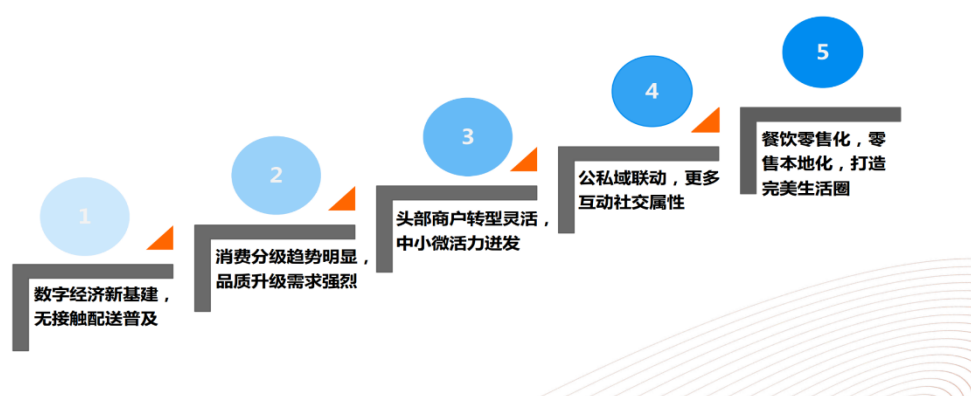


图 3 2011-2020 年餐饮业与在线外卖订单类型图

疫情倒逼餐营业数字化提速，疫情加速餐饮企业建设外卖渠道，增强自身抗风险能力，餐饮企业门店中堂食+外卖模式比例从 35%提高至 40%，纯外卖比例从 4%提高至 7%，并保持稳定。几乎全部品类订单均显著增加，齐总西式快餐、小吃快餐增幅显著，并保持持续增长。

图 4 新形势下在线外卖机遇趋势图

新形势下，外卖行业**新消费**·新场景·**新品牌**



就大学而言，也是个小型社区，相较于外校的餐馆，并没有将我校食堂与学生联系起来的具有我校特色的数字化订餐平台，基于数字化现代的时代背景，此项目旨在构建【校园食堂】—【配送员】—【学生】体系架构，利用现有开发技术与多媒体宣传技术实现网上订餐平台的建设与推广。

(2) 总体规划

1) 鉴于移动互联网的高速发展，项目团队采用【小程序】与【APP】为主要实现方式。另外，利用微信偌大的用户流量与支付系统，项目团队采用【小程序】优先原则，进行用户端与管理端的便携化实现。

2) 另外，考虑到 WEB 管理系统的强大功能，项目团队以 WEB 为后台主要实现方式，另外，进行前台简化实现，扩展移动端的功能，进行复杂功能的实现。

（3）项目特色

- 1) 疫情时期校园食堂数字化平台建设;
- 2) 校园食堂网上点餐、支付、配送、验收;
- 3) 后台数据可视化分析, 绑定用户粘性。

二、实验目的

基于此综合实训, 进行前期调研、题目选择、资料搜集、框架搭建、技术实现、功能测试等产品开发的多项工作, 将所学的理论知识应用到实际项目中。与此同时, 锻炼自身的资料搜集能力, 了解专业前沿技术知识, 在资料、项目、框架搜集的同时, 学习相关技术, 以项目促进学习。另外, 项目孵化与理论教育结合, 激发学生的创新性思维, 增强学生创造力与实际操作能力。作为新时代技术要求较高的专业——计算机相关专业, 理论与实践相结合显得尤为重要。另外, 校外部分产业发展工程教育来挖掘校内技术人才, 此次综合实训恰恰融合了工程教育的核心理念。

三、总体设计

3.1、总体架构

本项目以【小程序】为主, 与此同时, 进行管理后台的搭建工作, 实现多元系统对接, 尽可能较大程度方便校园学生使用。

为达到快速迭代的项目需求, 本多元系统【小程序、APP、WEB】均采用前后端分离技术架构, 针对不同的系统形式采用较为适合的框架(如 MVC 框架、MVP 框架等)。

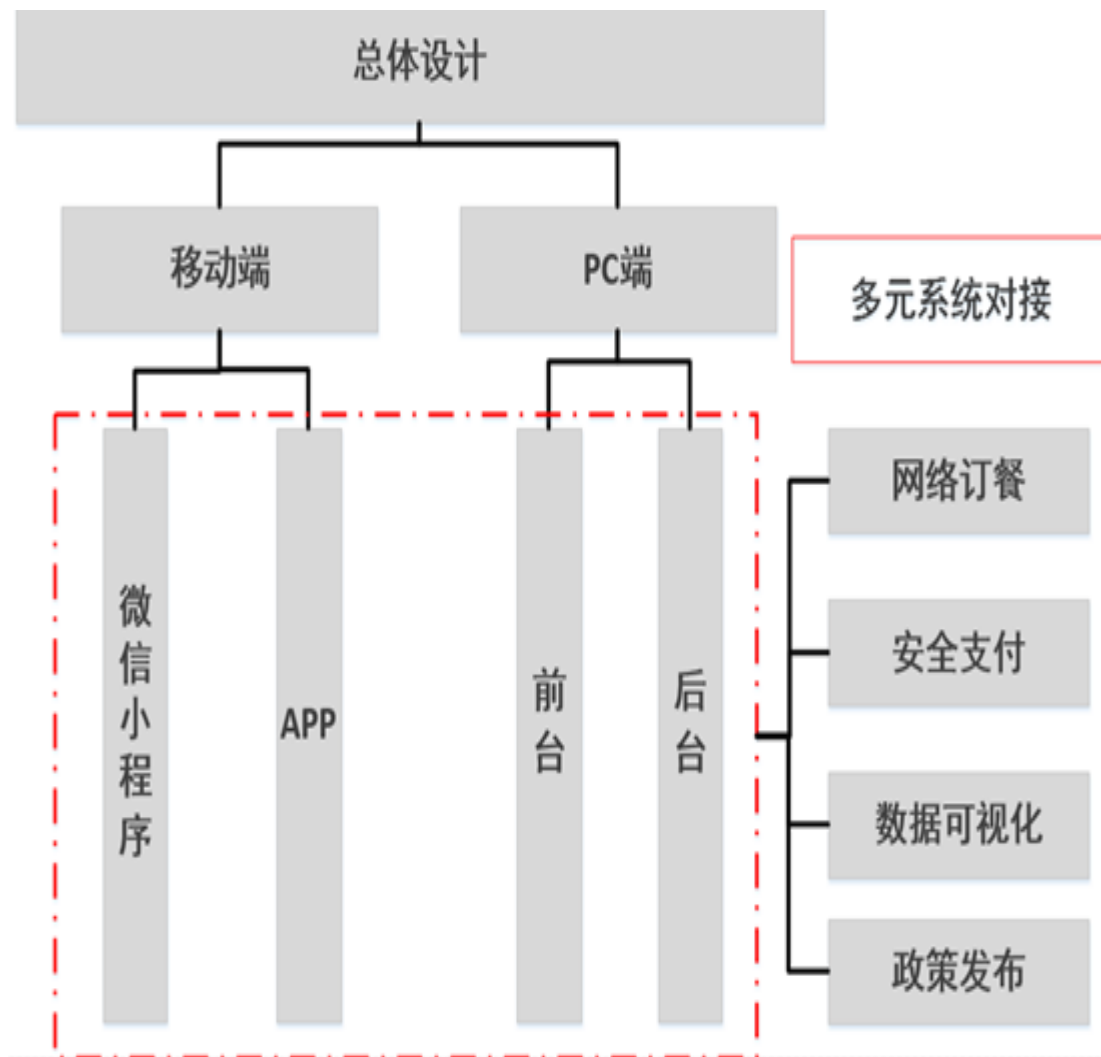


图 5 项目总体架构图

此项目涵盖网上订餐与配送、安全支付、数据可视化、激励政策发布等主要功能，利用现有微信 API 接口与我国云平台相关成熟框架，进行此项目的开发工作。

3.2、技术选型

基于目前前后端主流框架,前端选择VUE. JS、BOOTSTRAP、AJAX. JS、Python 可视化大屏等框架,利用其规范进行前端开发,与此同时,APP 的控件及实践需利用 XML、JAVA 相关知识进行研发工作。同时后端利用 SPRING 全家桶(springBoot、springCloud、springSecurity)、Mybatis (持久层框架)、云平台接口等技术栈,同时利用高效协作工具、版本管理 (GIT) 进行项目协作优化。

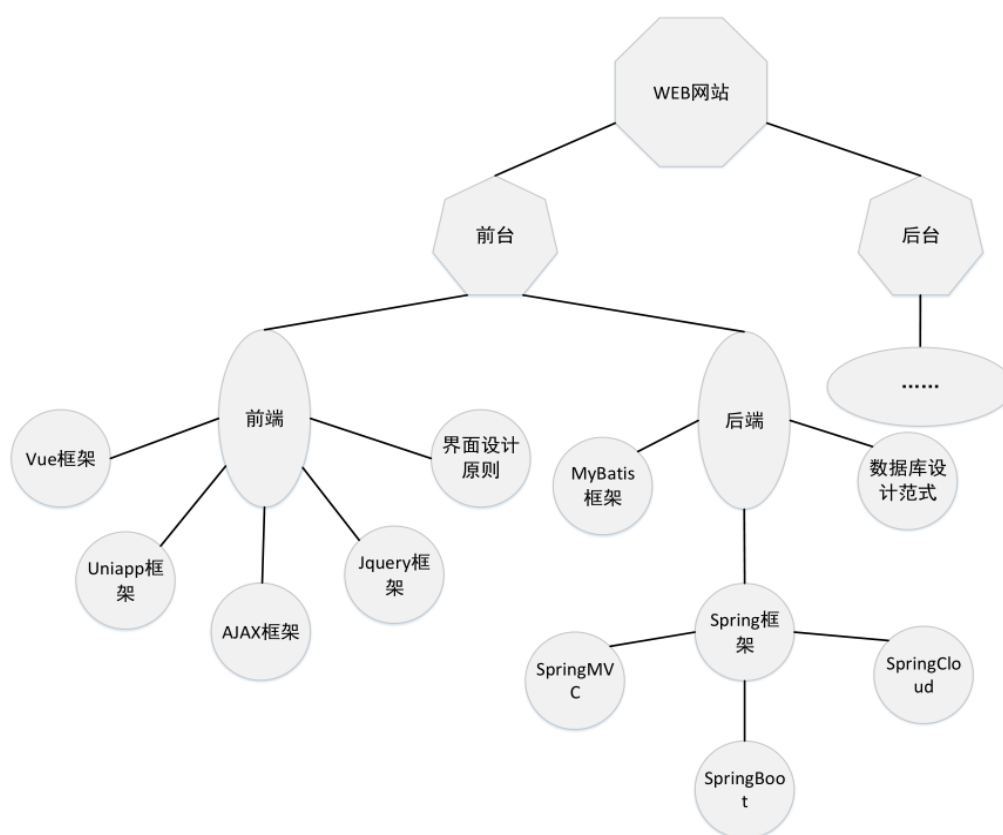


图 6 WEB 后台架构图

3.3、项目研发步骤

以小程序为例,主要分为调研分析、设计、开发、测试、推广五个流程,其中调研分析涉及需求分析、数据结构模型、需求优化与迭代计划,设计包含总体设计、原型设计、功能设计、数据库设计、流程设计,开发 git 等工具进行协作开发,把控开发进度,利用现有框

架简化项目开发，并进行优化。

3.4、小程序功能设计

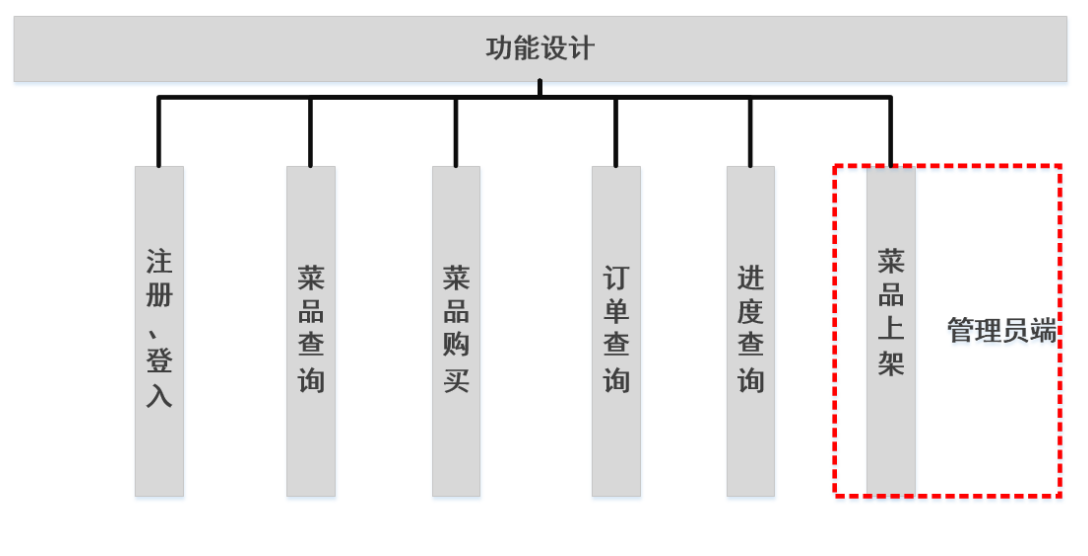


图 7 小程序总体功能设计图

目前小程序总体涵盖【注册登入】、【菜品查询】、【菜品购买】、【订单查询】、【进度查询】、【菜品上架】（管理员端）六大基本功能模块，后续将基于此进行模块功能的拓展与优化，如【安全支付】模块等。

【注册登入】：用户可通过手机号进行账号注册，或选择学籍信息进行录入（参照《到梦空间》激活账户方式），两种方式都可生成小程序的账户信息，而后可利用其进行小程序的登入操作；

【菜品查询】：可利用多项选择器进行地点、时间段的菜品查询；

【菜品购买】：点击对应菜品可进行详细信息的查询，可点击【订购】按钮控件进行菜品购买操作，并让用户具有修改默认配送信息的操作，订购后订单记录增加一条；

【订单查询】：用户可以通过订单查询功能查询当前订单，可修改订单的状态，如待收货、已收货、已取消等；

【进度查询】：此功能可以将配送进度以可视化方式（如进度条）展示给用户，便于用户知晓配送进度；

【菜品上架】：小程序利用角色权限管理原则，暂时分为普通用

户、管理员两种角色，普通用户可进行基础订餐操作，管理员可进行菜品上架操作。菜品上架功能后续可结合嵌入式技术便于食堂人员进行菜品信息录入。

四、详细设计

4.1、原型设计

原型设计，又称界面设计，利用 Axure RP 软件进行原型图的绘制操作，包括界面、事件、跳转等，延伸需求，进行数字化需求成果的实现。

小程序的原型设计如下：

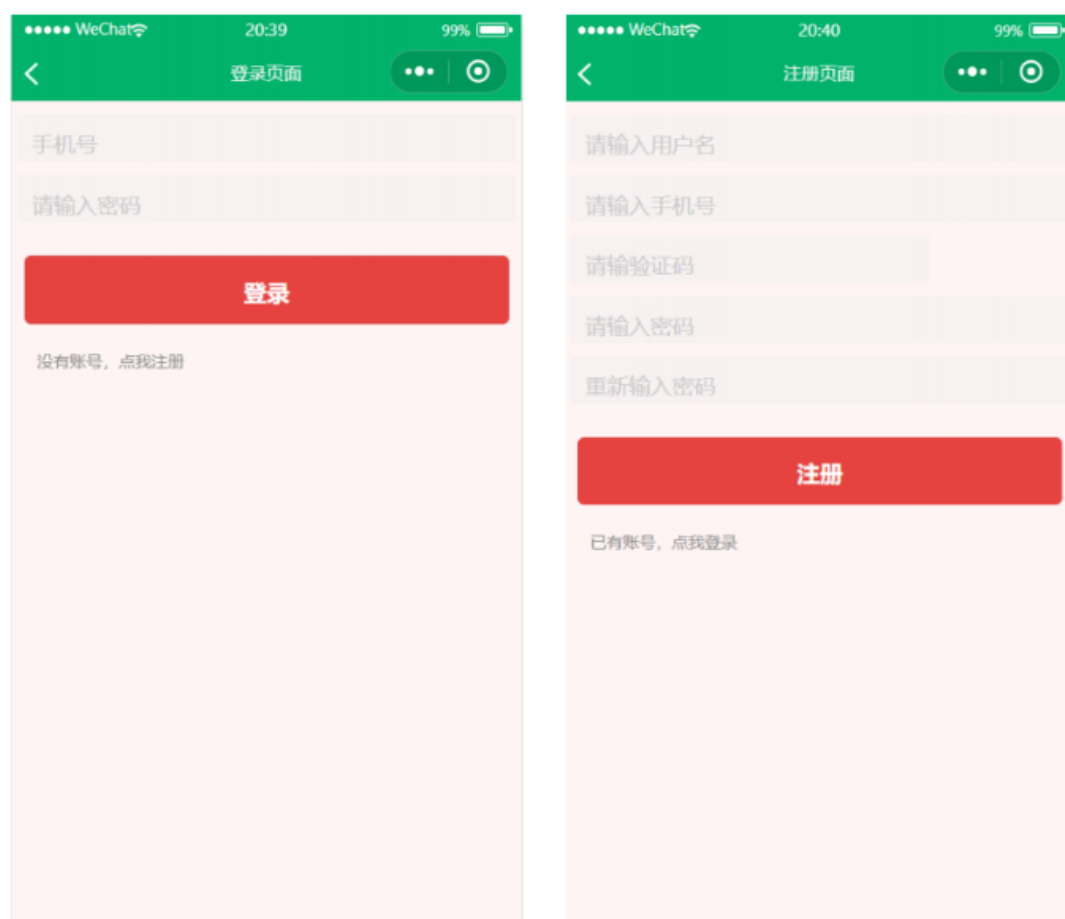


图 8 登入注册原型设计图



图 9 小程序菜品查询原型设计图

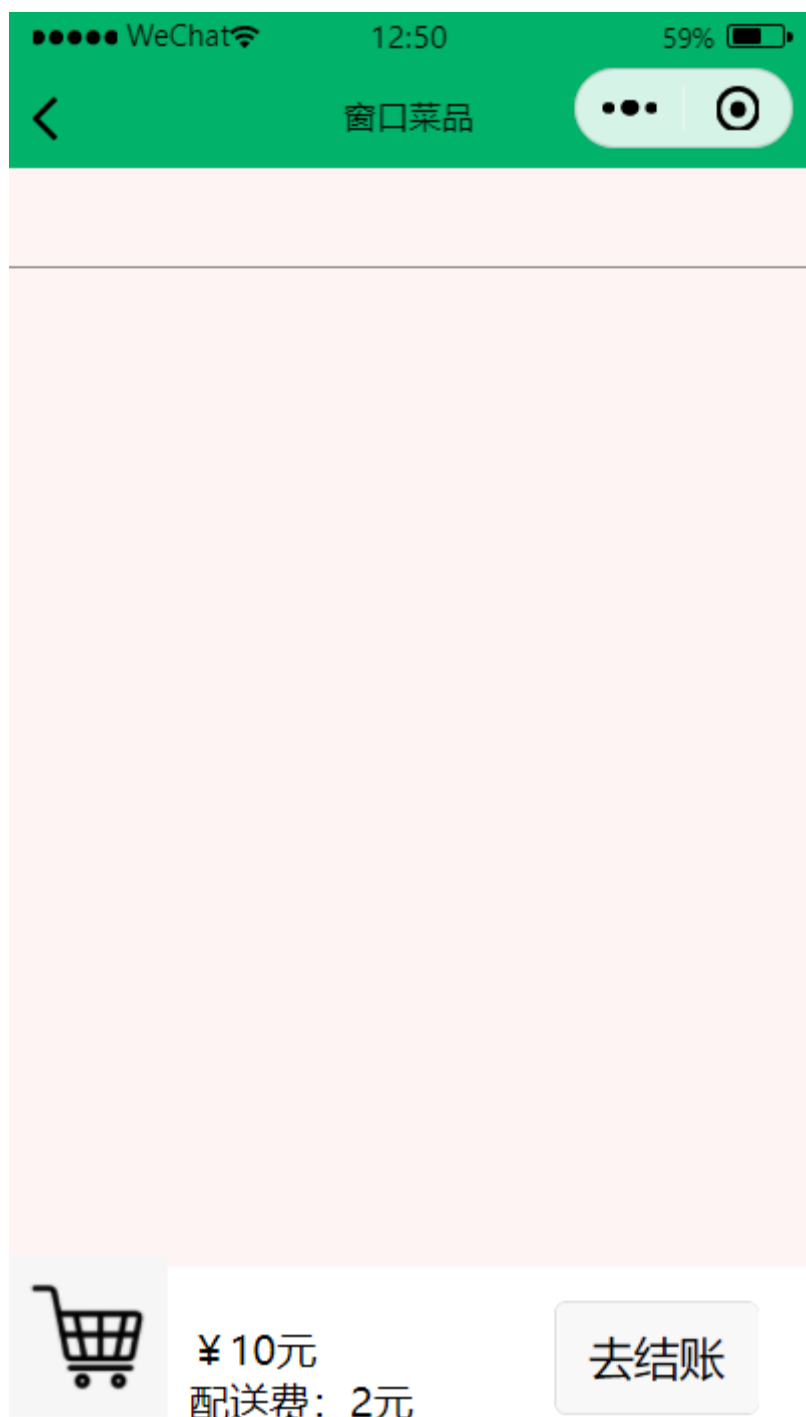


图 10 小程序菜品购买原型设计图



图 11 小程序支付提示原型设计图



图 12 小程序订单查询原型设计图



图 13 小程序个人信息原型设计图

4.2、数据库设计

后端采用 Mysql 数据库，利用 VISIO 绘图软件进行数据库表的绘制，包括 E-R 图、UML 类图。

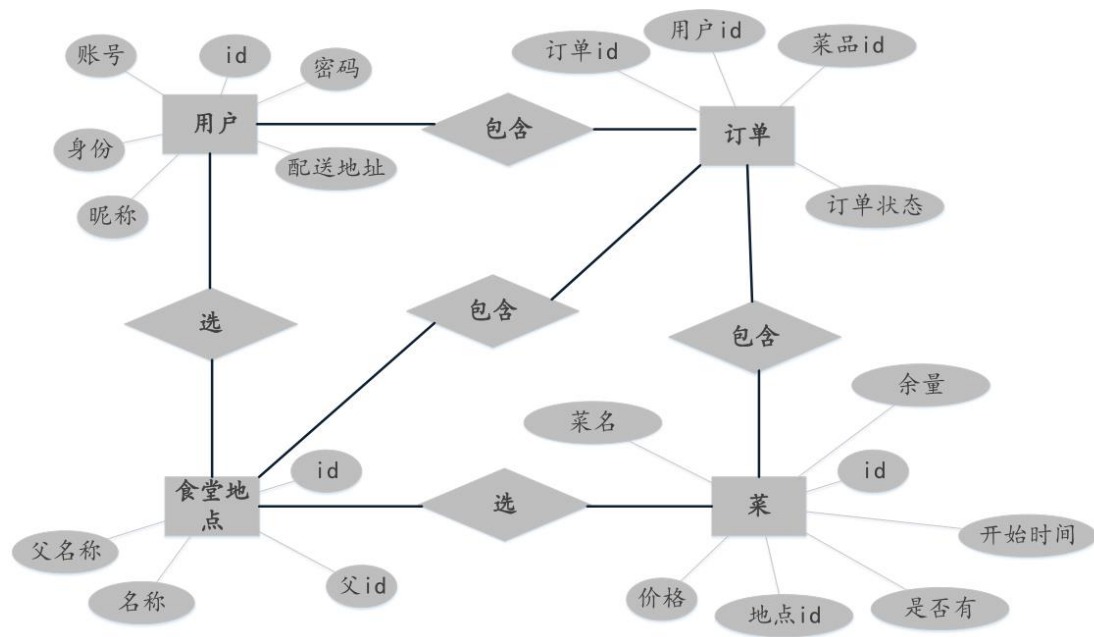


图 14 小程序数据库 E-R 图

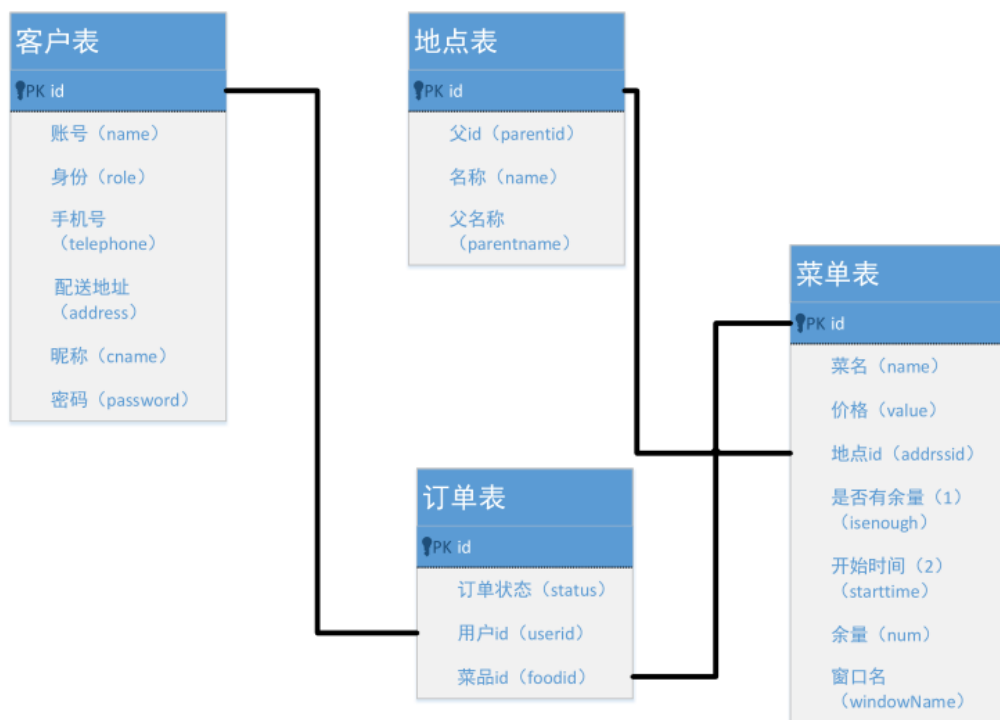


图 15 小程序数据库 UML 类图

4.3、流程设计

本项目采用前后端分离架构，前端基于微信开发者工具进行实现，利用微信 api 与 vue.js 架构进行实现，后端主要利用 java 语言进行开发，使用 SpringBoot 架构。此次项目是小程序的入门之作，还未使用持久层框架（Mybatis 框架），故代码可能会有些冗余。本人主要是进行 java 后端的开发，因流程较多，只展示部分流程操作。

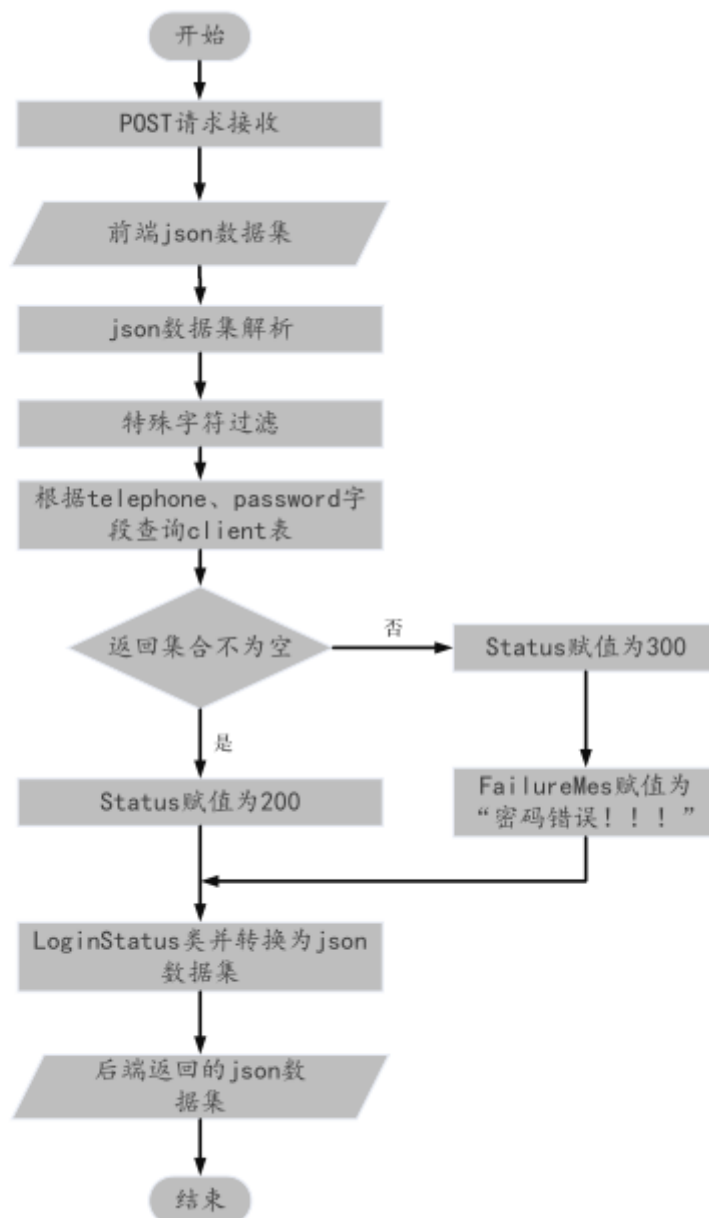


图 16 小程序后端登入功能流程图

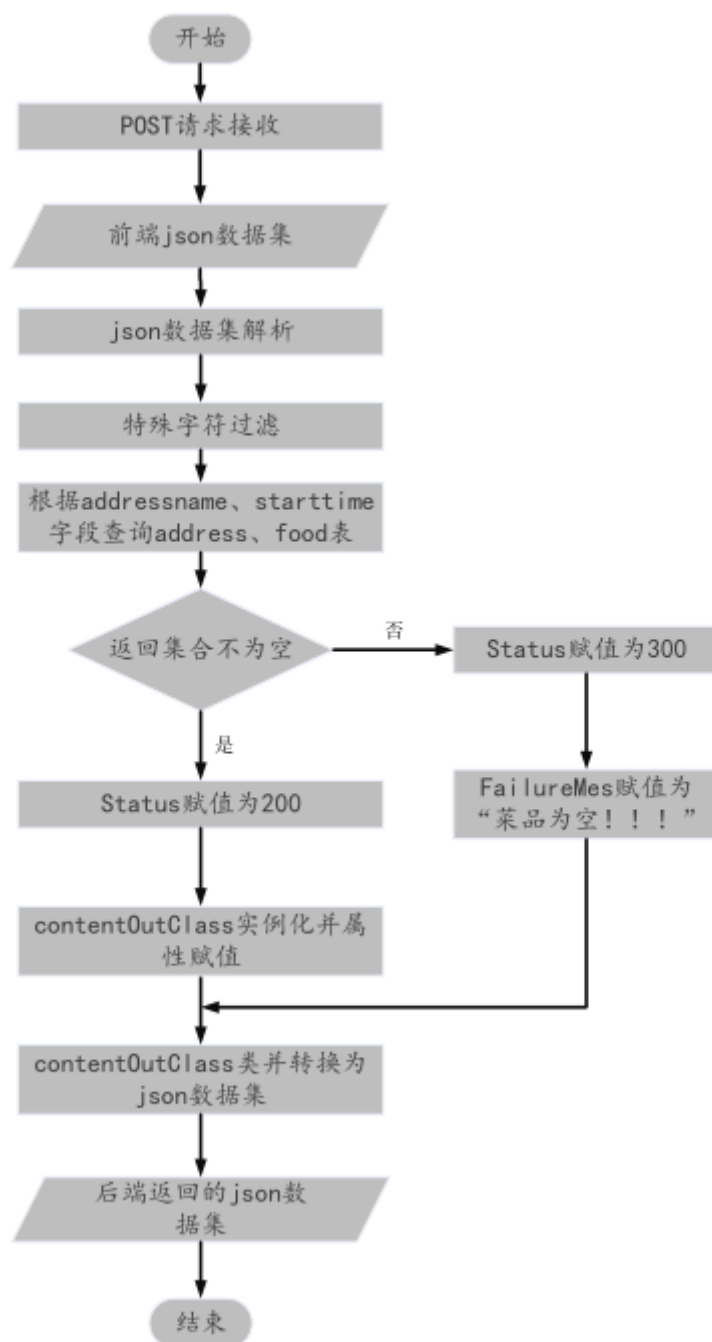


图 17 小程序后端菜品查询功能流程图

在实际开发中利用异常类进行错误预警，实现用户友好功能。

五、实验结果与分析

利用前后端分离实现整个订餐平台 DEMO 的实现，包括前端界面

开发、数据库建立、后端程序开发，基本上实现注册登入、菜品查询、订单查询三大基础部分功能，其余功能模块还在进行开发中。

效果图如下：



图 18 小程序角色选择界面图



图 19 小程序登入失败界面图



图 20 小程序主界面多项选择图



图 21 小程序菜品查询界面图

详细代码请参考源码。

六、小结与心得体会

此次综合实训，我是打算初探微信小程序，入手微信小程序，方便日后移动端的开发。作为移动互联网强悍的霸主，微信小程序依靠

着雄厚的用户流量、丰富的 api、微信支付对接，微信小程序拥有较为广阔的市场。此次项目，添加了自己的创新元素，为校园市食堂的数字化转型做出贡献，从而加快智慧校园的前进步伐。与此同时，作为此项目的主负责人，不仅了解了微信小程序的开发流程、架构等，还锻炼了自己的实践动手能力和团队合作能力，提前适应公司软件开发流程，并强化自己的创新性思维能力。

从前期的资料搜集、数据分析、团队讨论到后期的协作开发、进度管理、版本管理，自己的综合能力、交流能力得到较大提升。此外，从专业技能角度来看，此次综合实训将实践与理论结合，将 Internet 与 web 编程的理论知识应用到实际项目中，大大提升自己的实操能力。在项目研发过程中，虽然遇到技术难点，但前后端积极讨论与协商，协定接口，商量实现方式，当我们真正将小程序运行起来时，那一刻的成就感把所有的疲惫都驱散了。

小程序研发成果这块，从程序逻辑、数据库结构、界面美化、性能方面都有较大的优化空间，在界面美化方面，小程序的整体界面显得粗糙不美观，与前端协商后，决定寻找相关界面的资源或进行原创设计，另一方面，此版本的交互并没有按照用户至上的原则进行设计，此方面还需好好优化一下。关于数据库结构方面，数据库设计时未考虑全面，字段冗余问题突出。同时打算运用字典进行数据库结构的优化，并参考优秀数据库结构作品，进行学习与实践操作。程序逻辑这块，在开发中仅注重逻辑、功能的实现，并没有进行代码整理与逻辑优化，并未使用 Mybatis 框架，代码整体显得臃肿。其次，后端 springboot 注解未深入学习，只在入门阶段，效率和编码上较为逊色。其次，还需多与团队成员、前辈、老师进行沟通与讨论，便于项目的调优与问题的解决。在设计方面，可能前期设计不太充分，导致后期的更改频繁。

还需加强项目开发能力与技术选型能力。

附源码

Java 后端部分代码:

```
/* 启动类 */
package com.example.demo;

import java.util.ArrayList;
import java.util.List;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import
org.springframework.boot.autoconfigure.http.HttpMessageConverters;
import
org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration
;
import org.springframework.context.annotation.Bean;
import org.springframework.http.MediaType;
import org.springframework.http.converter.HttpMessageConverter;
import com.alibaba.fastjson.serializer.SerializerFeature;
import com.alibaba.fastjson.support.config.FastJsonConfig;
import
com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter;

@EnableAutoConfiguration(exclude={DataSourceAutoConfiguration.class})
@SpringBootApplication(scanBasePackages={"Controller"})
public class ReservationSmallappApplication {

    public static void main(String[] args) {
        SpringApplication.run(ReservationSmallappApplication.class,
args);
    }
    @Bean
    public HttpMessageConverters fastJsonHttpMessageConverters(){
```



```

        FastJsonHttpMessageConverter fastConverter = new
FastJsonHttpMessageConverter();

        List<MediaType> supportedMediaTypes = new ArrayList<>();
        supportedMediaTypes.add(MediaType.APPLICATION_JSON);
        supportedMediaTypes.add(MediaType.APPLICATION_JSON);
        supportedMediaTypes.add(MediaType.APPLICATION_ATOM_XML);
        supportedMediaTypes.add(MediaType.APPLICATION_FORM_URLENCODED);
        supportedMediaTypes.add(MediaType.APPLICATION_OCTET_STREAM);
        supportedMediaTypes.add(MediaType.APPLICATION_PDF);
        supportedMediaTypes.add(MediaType.APPLICATION_RSS_XML);
        supportedMediaTypes.add(MediaType.APPLICATION_XHTML_XML);
        supportedMediaTypes.add(MediaType.APPLICATION_XML);
        supportedMediaTypes.add(MediaType.IMAGE_GIF);
        supportedMediaTypes.add(MediaType.IMAGE_JPEG);
        supportedMediaTypes.add(MediaType.IMAGE_PNG);
        supportedMediaTypes.add(MediaType.TEXT_EVENT_STREAM);
        supportedMediaTypes.add(MediaType.TEXT_HTML);
        supportedMediaTypes.add(MediaType.TEXT_MARKDOWN);
        supportedMediaTypes.add(MediaType.TEXT_PLAIN);
        supportedMediaTypes.add(MediaType.TEXT_XML);
        fastConverter.setSupportedMediaTypes(supportedMediaTypes);

        FastJsonConfig fastJsonConfig = new FastJsonConfig();

        fastJsonConfig.setSerializerFeatures(SerializerFeature.PrettyFormat);
        fastConverter.setFastJsonConfig(fastJsonConfig);
        HttpMessageConverter<?> converter =fastConverter;
        return new HttpMessageConverters(converter);
    }

}

```

```
/* controller 类 */
```

```
package Controller;
```

```
import java.io.IOException;
```

```
import java.util.Enumeration;
```

```
import java.util.HashMap;
```

```
import java.util.Map;
```

```
import java.util.UUID;
```

```
import javax.servlet.http.Cookie;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
import org.springframework.http.HttpStatus;
```

```
import org.springframework.http.ResponseEntity;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.RequestBody;
```

```
import org.springframework.web.bind.annotation.RestController;
```

```
import com.alibaba.fastjson.JSONObject;
```

```
import Model.DatebaseMain;
```

```
import Model.contentOutClass;
```

```
import Model.loginDatabase;
```

```
import Model.loginStatus;
```

```
import Model.orderOutClass;
```

```
import Model.registStatus;
```

```
@RestController
```

```

public class controllerMain {
    private static loginDatabase logindatabase=new loginDatabase();
    private static DatabaseMain databaseMain=new DatabaseMain();

    /*
     * @RequestMapping("/") public String view(){ return "login"; }
     */
    //接收 login 的 get 请求, 传送 uid, password 字段的 json 数据集, 返回 status
    json 数据集, 成功 status=200, 反之 300
    @PostMapping("/canteen/visitor/login")
    public loginStatus loginFun(@RequestBody(required=false) JSONObject
    jsonParam,HttpServletRequest request, HttpServletResponse response) {

        System.out.println(jsonParam);
        loginStatus loginstatus=new
loginStatus(databaseMain.canLogin(logindatabase,

jsonParam.getString("user_telephone"),jsonParam.getString("user_passwor
d"),request,response));
        return loginstatus;
    }

    @PostMapping("/canteen/visitor/regist")
    public registStatus contentFun(@RequestBody(required=false)
JSONObject jsonParam) {
        registStatus registstatus=new
registStatus(databaseMain.canRegist(logindatabase,

jsonParam.getString("user_name"),jsonParam.getString("user_telephone"),
jsonParam.getString("user_password")));
        return registstatus;
    }
}

```

```

    @PostMapping("/canteen/visitor/windows")
    public contentOutClass menuMain(@RequestBody(required=false)
JSONObject jsonParam) {
        System.out.println(jsonParam);
        contentOutClass contentoutClass=new contentOutClass();
        contentoutClass.setFlag(datebaseMain.canMenu(contentoutClass,
jsonParam.getString("canteen"),
            jsonParam.getString("tier"),
jsonParam.getString("time"))));
        return contentoutClass;
    }

    @PostMapping("/canteen/visitor/finishOrder")
    public orderOutClass finishOrderDisplay(@RequestBody(required=false)
JSONObject jsonParam,HttpServletRequest request) {
        orderOutClass orderoutClass =new orderOutClass();
        orderoutClass.setFlag(datebaseMain.candisplayOrder(orderoutClass,
request, 1));
        return orderoutClass;
    }

    @PostMapping("/canteen/visitor/nofinishOrder")
    public orderOutClass
NofinishOrderDisplay(@RequestBody(required=false) JSONObject
jsonParam,HttpServletRequest request) {
        orderOutClass orderoutClass =new orderOutClass();
        orderoutClass.setFlag(datebaseMain.candisplayOrder(orderoutClass,
request,0));
        return orderoutClass;
    }
}

```

```

/* model 包 */
/* DatabaseMain 类 */
package Model;

```

```

import java.io.File;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Enumeration;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Objects;
import java.util.UUID;
import java.util.List;

import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

import org.apache.tomcat.util.http.fileupload.disk.DiskFileItem;
import org.springframework.http.HttpStatus;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.PreparedStatementCreator;
import org.springframework.jdbc.core.PreparedStatementSetter;
import org.springframework.jdbc.datasource.DriverManagerDataSource;
import org.springframework.jdbc.support.GeneratedKeyHolder;
import org.springframework.jdbc.support.KeyHolder;
import org.springframework.web.multipart.MultipartFile;
import org.springframework.web.multipart.MultipartHttpServletRequest;
import org.springframework.web.multipart.commons.CommonsMultipartFile;

public class DatabaseMain {
    private DataSource datasource;
    //数据库配置
    public DatabaseMain() {

```

```

        DriverManagerDataSource dataSource =new
DriverManagerDataSource();

dataSource.setUrl("jdbc:mysql://localhost:3306/reservation?useUnicode=t
rue&characterEncoding=utf8&useSSL=false&allowPublicKeyRetrieval=true&se
rverTimezone=GMT%2B8");

        dataSource.setUsername("root");//用户名
        dataSource.setPassword("root");//密码
        dataSource.setDriverClassName("com.mysql.cj.jdbc.Driver");
        this.datasourcesource=dataSource;
        this.jdbcTemplate=new JdbcTemplate(dataSource);
    }
    private String tableName = ""; // table name

    private String sqltext = "";
    //绑定数据源
    private JdbcTemplate jdbcTemplate ;

    public void setTableName(String name) {
        this.tableName = name;
    }

    public String getTableName() {
        return this.tableName;
    }

    public boolean canLogin(loginDatabase logindatabase, String Tele,
String Password,HttpServletRequest request, HttpServletResponse
response) {
        boolean canlogin=false;
        //Boolean canlogin = false; // can login
        // 1、获取提交的值
        logindatabase.setTelephone(Tele);
        logindatabase.setPassword(Password);
        // 2、查值
        this.tableName = " client ";

```

```

        SQLmodel sqlmodel=new SQLmodel(this.tableName);
        ArrayList<String> field = new ArrayList<String>();
        ArrayList<String> conditionList = new ArrayList<String>();
        field.add(" id ");
        conditionList.add(" telephone ");
        conditionList.add(" password ");
        this.sqltext=sqlmodel.Select(field, conditionList);
        /* System.out.println(this.sqltext); */
        List<Map<String, Object>> list =
this.jdbcTemplate.queryForList(this.sqltext,Tele>Password);
        if (list.isEmpty()==false) {
            canlogin = true;
            //加 cookie
            Cookie cookie = new
Cookie("userId",list.get(0).get("id").toString());
            cookie.setHttpOnly(true);
            cookie.setPath("/");
            cookie.setDomain("localhost");
            response.addCookie(cookie);
        }
        this.tableName = "";
        return canlogin;
    }

    public boolean canRegist(loginDatabase logindatabase, String
name,String Tele, String Password) {
        boolean canlogin=false;
        //Boolean canlogin = false; // can login
        // 1、获取提交的值
        /*
            * logindatabase.setTelephone(Tele);
logindatabase.setPassword>Password);
            */
        // 2、查值
        this.tableName = " client ";
        SQLmodel sqlmodel=new SQLmodel(this.tableName);
        ArrayList<String> field = new ArrayList<String>();

```

```

        ArrayList<String> conditionList = new ArrayList<String>();
        field.add(" id ");
        conditionList.add(" telephone ");
        this.sqltext=sqlModel.Select(field, conditionList);
        /* System.out.println(this.sqltext); */
        List<Map<String, Object>> list =
this.jdbcTemplate.queryForList(this.sqltext,Tele,Password);
        if (list.isEmpty()==true) {
            field.clear();
            conditionList.clear();
            this.sqltext=sqlModel.Insert(6);
            int
flag=this.jdbcTemplate.update(this.sqltext,name,"student",Tele,"",name,
Password);
            if (flag!=0) {
                canlogin = true;
            }
            //加 cookie

        }

        this.tableName =
            "";
        return canlogin;
    }

    public boolean canMenu(contentOutClass contentoutClass , String
area,String floor, String time) {
        boolean canMenu=false;
        try {

            timeFormat timeformat=new timeFormat();
            float timeInt=timeformat.timeformat(time);

            int addressId=0;
            SQLmodel sqlModel=new SQLmodel(" dininghall ");
            ArrayList<String> field = new ArrayList<String>();

```



```

        ArrayList<String> conditionList = new ArrayList<String>();
        field.add(" id ");
        conditionList.add(" name ");
        conditionList.add(" parentname ");
        this.sqltext=sqlModel.Select(field, conditionList);
        List<Map<String, Object>> list =
this.jdbcTemplate.queryForList(this.sqltext,floor,area);
        addressId=(int)list.get(0).get("id");

        field.clear();
        conditionList.clear();
        sqlModel=new SQLModel(" food ");
        field.add(" id ");
        field.add(" name ");
        field.add(" value ");
        field.add(" windowName ");
        conditionList.add(" addressid ");
        conditionList.add(" isenough ");
        this.sqltext=sqlModel.Select(field, conditionList);
        this.sqltext+=" AND starttime <= ? ";
        System.out.println(this.sqltext);

        List<Map<String, Object>>
foodList=this.jdbcTemplate.queryForList(this.sqltext,addressId,1,timeIn
t);
        for (Iterator iterator = foodList.iterator();
iterator.hasNext();) {
            Map<String, Object> map = (Map<String, Object>)
iterator.next();
            ArrayList<contentInterClass> temlist=new
ArrayList<contentInterClass>();
            temlist=contentoutClass.getMenuArr();
            int rank=temlist.size();
            int Windindex=-1;
            for (int i=0;i<temlist.size();i++) {

```

```

if(temlist.get(i).getTitle().equals(map.get("windowName").toString()))
{
    temlist.get(i).setId(rank);
    List<Map<String, String>>
teminterList=temlist.get(i).getSubArr();
    Map<String,String> temMap=new HashMap<String,
String>();
    temMap.put("imgDesc",map.get("name").toString());

temMap.put("imgSrc","/Images/menu/"+map.get("id").toString()+".jpg");
    temMap.put("value", map.get("value").toString());
    teminterList.add(temMap);
    temlist.get(i).setSubArr(teminterList);
    break;
}
}
if(Windindex== -1) {
    contentInterClass contentinterClass=new
contentInterClass();

contentinterClass.setTitle(map.get("windowName").toString());
    contentinterClass.setId(rank+1);
    temlist.add(contentinterClass);
    List<Map<String, String>>
teminterList=temlist.get(rank).getSubArr();
    Map<String,String> temMap=new HashMap<String, String>();
    temMap.put("imgDesc",map.get("name").toString());

temMap.put("imgSrc","/Images/menu/"+map.get("id").toString()+".jpg");
    temMap.put("value", map.get("value").toString());
    teminterList.add(temMap);
    temlist.get(rank).setSubArr(teminterList);
}
contentoutClass.setMenuArr(temlist);
}

```

```

        canMenu=true;
    } catch (Exception e) {
        // TODO: handle exception
    }

    return canMenu;

}

public boolean candisplayOrder(orderOutClass
orderoutClass,HttpServletRequest request,int status) {
    boolean candisplay=false;
    try {

        String cookieName="userid";
        int clientId=0;
        Cookie[] cookies=request.getCookies();
        for (Cookie cookie : cookies) {
            if(cookieName.equals(cookie.getName())){
                clientId=Integer.valueOf(cookie.getValue());
            }
        }

        SQLmodel sqlmodel=new SQLmodel(" orderlist ");
        ArrayList<String> field = new ArrayList<String>();
        ArrayList<String> conditionList = new ArrayList<String>();

        field.add(" id ");
        field.add(" foodid ");
        conditionList.add(" userid ");
        conditionList.add(" status ");
        this.sqltext=sqlmodel.Select(field, conditionList);
        List<Map<String, Object>> list =
this.jdbcTemplate.queryForList(this.sqltext,clientId,status);

```

```

        ArrayList<orderClass> temList=new ArrayList<orderClass>();
        for (Iterator iterator = list.iterator(); iterator.hasNext();) {
            orderClass orderclass=new orderClass();
            Map<String, Object> map = (Map<String, Object>)
iterator.next();
            sqlmodel=new SQLmodel(" food ");
            field.clear();
            conditionList.clear();
            field.add(" name ");
            field.add(" addressid ");
            field.add(" value ");
            field.add(" windowName ");
            conditionList.add(" id ");
            this.sqltext=sqlmodel.Select(field, conditionList);
            System.out.println(this.sqltext);
            List<Map<String, Object>> foodlist =
this.jdbcTemplate.queryForList(this.sqltext,Integer.valueOf(map.get("id
").toString()));
            sqlmodel=new SQLmodel(" dininghall ");
            field.clear();
            conditionList.clear();
            field.add(" name ");
            field.add(" parentname ");
            conditionList.add(" id ");
            this.sqltext=sqlmodel.Select(field, conditionList);
            List<Map<String, Object>> addressname =
this.jdbcTemplate.queryForList(this.sqltext,Integer.valueOf(foodlist.ge
t(0).get("addressid").toString()));

            orderclass.setFoodname(addressname.get(0).get("parentname").toString()+
" > "+addressname.get(0).get("name").toString()+" > "+
foodlist.get(0).get("windowName").toString()+">>"+foodlist.get(0).get("
name").toString());
            orderclass.setValue(foodlist.get(0).get("value").toString());
            temList.add(orderclass);
        }

```

```

        orderoutClass.setData(temList);
        candisplay=true;
    } catch (Exception e) {
        // TODO: handle exception
    }
    return candisplay;
}
}

```

微信小程序前端代码

```

<!--pages/menu/menu.wxml-->
<view>
    <!-- 轮播图开始 -->
    <swiper class="banner" autoplay interval="2000" circular next-
margin="20rpx" indicator-dots indicator-color="#3AC48C" indicator-
active-color="#fff">
        <swiper-item>
            <image src="/Images/11.jpg"></image>
        </swiper-item>
        <swiper-item>
            <image src="/Images/12.jpg"></image>
        </swiper-item>
        <swiper-item>
            <image src="/Images/13.jpg"></image>
        </swiper-item>
        <swiper-item>
            <image src="/Images/14.jpg"></image>
        </swiper-item>
    </swiper>
    <!-- 轮播图结束 -->

    <!-- 滚动选择器 -->
    <view>

```

```

        <picker mode="multiSelector" range="{{multiSelectorItems}}"
bindchange='multiSelectorChange'>
            <view>*当前选择: {{multiSelector}}</view>
        </picker>
    </view>

    <view>
        <button bindtap="clickMe">查询</button>
    </view>

    <!-- <text id="remain-position">当前剩余位
置:{{remainPosition}}</text> -->
    <!-- catchtap="getButtonWindowsList" -->
    <view hidden="{{hiddenName}}">
        <view class="main">
            <!-- 左边导航栏 -->
            <scroll-view scroll-y="true" scroll-into-view="{{leftId}}"
class="left">
                <view wx:for="{{menuArr}}" wx:key="*this" data-
myid="{{item.id}}" id="left{{item.id}}" class="scroll-view-item
{{lifeActiveNum==item.id?'active':''}}" bindtap="leftClick">
                    {{item.title}}</view>
            </scroll-view>

            <!-- 右边 -->
            <scroll-view scroll-y="true" class="right" scroll-with-
animation="true" bindscroll="rightScrollTop" scroll-into-
view="{{rightId}}">
                <view wx:for="{{menuArr}}" wx:key="*this"
id="right{{item.id}}" class="scroll-view-item rightblock ">
                    <view class="right_title" >{{item.title}}</view>
                    <view class="right-flex" >
                        <view class="content" wx:for="{{item.subArr}}"
wx:key="*this">
                            <navigator
url="../orderitem/orderitem?id={{item.window_id}}" class="windows-eat-
item" hover-class="windows-eat-item-hover" >

```

```

        <image src="{{item.imgSrc}}"
mode="widthFix"></image>
        <text>{{item.imgDesc}}</text>
    </navigator>
</view>
</view>
</view>
</scroll-view>
</view>
</view>

<!-- 窗口信息显示开始 -->
<!-- <view class="video-wrap">
    <view class="windows-eat" wx:for="{{windowsList}}"
bindtap="select">
        <navigator url="../orderitem/orderitem?id={{item.window_id}}"
class="windows-eat-item" hover-class="windows-eat-item-hover" >
            <image src="{{item.window_picture}}" class="windows-eat-
image"></image>
            <view class="info">
                <text class='yuzhong'>{{item.window_num}}号窗口</text>
                <text class='info-name'>{{item.window_name}}</text>
            </view>
        </navigator>
    </view>
</view> -->
<!-- 窗口信息显示结束 -->

</view>

<view style="position:fix;bottom:0;left:0;width:100%">
    <tabbar class="tabbar" id="myTabbarId" default-show-
index="{{ defaultShowIndex }}"
bind:myTabbarEvent="popBaitiaoView"></tabbar>
</view>

```

```

// pages/menu/menu.js
var Value;
Page({
  /**
   * 页面的初始数据
   */
  data: {
    multiSelectorItems: [
      ['一区', '七区'],
      ['一楼', '二楼', '三楼'],
      ['11:30-12:00', '12:00-12:30', '12:30-13:00', '13:00-14:30']
    ],
    canteen_class: '',
    remainPosition: 20,
    defaultShowIndex: 1,
    hiddenName: true,
    menuArr: {},
    leftId: "left0",
    rightId: "right0",
    lifeActiveNum: 0,
    heightArr: []
  },
  clickMe: function(e){

    // this.setData({
    //   hiddenName: !this.data.hiddenName
    // })
    // if (this.data.time == "时间段 11:30-12:00") {
    //   times = "1";
    // } else if (this.data.time == "时间段 12:00-12:30") {
    //   times = "2";
    // } else if (this.data.time == "时间段 12:30-13:00") {

```



```

//   times = "3";
// } else if (this.data.time == "时间段 13:00-13:30") {
//   times = "4";
// } else if (this.data.time == "时间段 13:30-14:00") {
//   times = "5"
// }
const that = this;
wx.request({
  url: 'http://localhost:8080/canteen/visitor/windows',
  data: {
    canteen: Value[0],
    tier: Value[1],
    time: Value[2]
  },
  method: 'POST',
  header: {
    'Content-type': 'application/json' // 默认值
    //'Content-Type': 'application/x-www-form-urlencoded'
  },
  success(res) {
    console.log(res);
    if (res.data.flag) {
      that.setData({
        menuArr: res.data.menuArr,
        // remainPosition: res.data.data,
      })
      console.log(this.data.menuArr);
      // getApp().globalData.time_id = res.data.errorMsg
    }
  }
});
leftClick(e) {
  this.setData({
    lifeActiveNum: e.target.dataset.myid,

```

```

        leftId: "left" + e.target.dataset.myid,
        rightId: "right" + e.target.dataset.myid
    })

},
multiSelectorChange: function (e) {
    let arrayIndex = e.detail.value;
    let array = this.data.multiSelectorItems;
    let value = new Array();
    for (let i = 0; i < arrayIndex.length; i++) {
        let k = arrayIndex[i];
        let v = array[i][k];
        value.push(v);
    }
    Value=value;
    this.setData({
        multiSelector: value
    });
},

// 获取窗口列表
getButtonWindowsList: function () {
    let times;
    if (this.data.time == "时间段 11:30-12:00") {
        times = "1";
    } else if (this.data.time == "时间段 12:00-12:30") {
        times = "2";
    } else if (this.data.time == "时间段 12:30-13:00") {
        times = "3";
    } else if (this.data.time == "时间段 13:00-13:30") {
        times = "4";
    } else if (this.data.time == "时间段 13:30-14:00") {
        times = "5"
    }
    wx.request({

```

```

url: 'http://localhost:8888/canteen/visitor/windows',
data: {
  canteen: this.data.canteen,
  tier: this.data.tier,
  time: times
},
method: 'POST',
header: {
  'Content-type': 'application/json' // 默认值
  //'Content-Type': 'application/x-www-form-urlencoded'
},
success(res) {
  console.log(res);
  if (res.data.flag) {
    that.setData({
      windowsList: res.data.objects,
      remainPosition: res.data.data,
    })
    getApp().globalData.time_id = res.data.errorMsg
  }
}
});
},

windowsEat: function () {

},

popBaitiaoView(e) {
  console.log(e)
  this.setData({
    defaultShowIndex: e.detail.activeIndex.showIndex
  })
  if (this.data.defaultShowIndex == 0) {
    wx.reLaunch({
      url: '/pages/admin/admin',
    })
  }
}

```

```

    }
    if (this.data.defaultShowIndex == 1) {
      wx.switchTab({
        url: '/pages/menu/menu',
      })
    }
  },
  /**
   * 生命周期函数--监听页面加载
   */
  onLoad: function (options) {

  },

  /**
   * 生命周期函数--监听页面初次渲染完成
   */
  onReady: function () {
    let _this = this
    setTimeout(() => {
      let initArr = [0]; //初始数组
      let initNum = 0; //初始数值
      const query = wx.createSelectorQuery()
      query.selectAll('.rightblock').boundingClientRect()
      query.selectViewport().scrollOffset()
      query.exec(function (res) {
        console.log(res[0]);
        res[0].map(val => {
          initNum += val.height; //实现高度的累加
          initArr.push(initNum) //初始数值加进数组中
        })
        console.log(initArr); //拿到每一个 height 存起来
        _this.setData({
          heightArr: initArr
        })
      })
    })
  }
}

```

```

    }, 300)
  },
  // 右边滚动事件
  rightScrollTop(e) {
    let st = e.detail.scrollTop;
    let myarr = this.data.heightArr;
    for (let i = 0; i < myarr.length; i++) {
      if (st >= myarr[i] && st < myarr[i + 1] - 10) {
        this.setData({
          leftId: "left" + i,
          lifeActiveNum: i
        })
        return;
      }
    }
  },

  /**
   * 生命周期函数--监听页面显示
   */
  onShow: function () {
    this.getWindowsList();
  },

  /**
   * 生命周期函数--监听页面隐藏
   */
  onHide: function () {

  },

  /**
   * 生命周期函数--监听页面卸载
   */
  onUnload: function () {

```

```
    },

    /**
     * 页面相关事件处理函数--监听用户下拉动作
     */
    onPullDownRefresh: function () {

    },

    /**
     * 页面上拉触底事件的处理函数
     */
    onReachBottom: function () {

    },

    /**
     * 用户点击右上角分享
     */
    onShareAppMessage: function () {

    }
  })
})
```