

基于 Django 的校园食堂订餐平台的设计与实现 开发文档

第 5 组（宋世龙、刘朝晖、孙昊、杨宏宇、闫家毓）

1 项目概述

随着互联网技术的高速发展和广泛应用，当今社会已全面进入信息时代。计算机技术为经济发展注入了强劲动力的同时，人们的日常生产生活习惯也随之改变，而且伴随着生活水平的提升，人们对自己的饮食选择投入了相较从前更多的时间和精力。紧张的学习和工作之余人们会选择通过享受美食获得身心愉悦。然而很多时候尤其是校园内会出现这样一种情况，老师和学生在午休下课去就餐时，会出现排队或没有位置的现象，还有的同学喜欢自己“宅”在寝室不出门。与市面上已经存在的外卖订餐平台模式相同，校内线上订餐也成为了一个方便快捷的选择。

本文从项目的技术路线、用户需求、总体和详细设计以及测试计划，简要说明了要设计和实现的校园食堂订餐平台是一款专为提升高校食堂订餐效率而设计的服务网站，旨在加快、简化同学们传统食堂就餐的进程和过程，提高在校师生的用餐体验。



Fig.1 线上订餐平台

2 拟定开发路线

核心技术支撑。Django 是一个高级 Python Web 框架，它以简洁、快速和可扩展的设计理念而闻名。它提供了一套完整的工具和库，用于帮助开发人员构建复杂且功能丰富的 Web 应用程序。1) Django 采用了 MVC(Model-View-Controller) 架构的变体，称为 MVT (Model-View-Template)。这种架构将应用程序的业务逻辑 (Model)、用户界面 (Template) 和请求处理逻辑 (View) 分开，使开发人员可以更好地组织和管理代码。2) Django 通过 URL 路由系统将请求映射到相应的视图函数上。视图函数处理请求并返回相应的 HTTP 响应。这种灵活的路由系统使得开发人员可以轻松定义和管理应用程序的 URL 结构。3) Django 提供了强大的表单处理功能，包括表单验证、数据清洗和错误处理。开发人员可以轻松创建和管理各种类型的表单，从简单的用户注册表单到复杂的数据输入表单。4) Django 自带一个功能强大的管理界面，使开发人员可以轻松管理后台数据和进行基本的 CRUD 操作。通过简单的配置，可以自动生成管理界面，减少了开发时间和工作量。5) Django 提供了一系列安全功能，包括防止常见的 Web 安全漏洞 (如跨站脚本攻击和跨站请求伪造)、密码哈希、用户认证和授权等。这些功能帮助开发人员构建安全可靠的 Web 应用程序。



Fig.2 Django 框架

MySQL 是一种开源的关系型数据库管理系统 (RDBMS)，它被广泛用于存储和管理结构化数据。1) MySQL 是一种关系型数据库，采用了表格的形式来组织和存储数据。数据以行和列的方式组织，每个表都由列定义，并使用主键和外键等关系建立表与表之间的关联。2) MySQL 支持标准的结构化查询语言 (SQL)，用于查询、插入、更新和删除数据。通过 SQL，开发人员可以对数据库执行各种操作，包括创建表、定义约束、查询数据和执行复杂的数据操作。3) MySQL 可在多个操作系统上运行，包括 Windows、Linux、Mac 等。这使得它成为一个灵活和可移植的数据库解决方案，可以适应不同的开发和部署环境。4) MySQL 提供了各种安全功能，包括用户认证、访问控制、数据加密和数据备份等。它允许管理员为数据库和表设置权限，并使用 SSL 加密连接来保护数据传输的安全性。



Fig.3 MySQL 数据管理系统

软硬件支持。PyCharm 是一款功能强大、易于使用的 Python 集成开发环境。它提供了丰富的工具和功能，帮助开发人员编写、调试和管理 Python 代码。无论是初学者还是专业开发人员，PyCharm 都是一个受欢迎的选择，可以提高开发效率和代码质量。

Tab.1 方法和工具

过程域	方法与工具
程序设计	PyCharm
系统分析与设计	Visio
配置管理	Visual Source Safe
制作文档	Microsoft Office 2021
...	

Tab.2 软硬件配置资源方案

资源名称	级别	详细配置	获取方式	费用
计算机	普通	Win 10/Mac 系统，可上网	已经存在	0
PyCharm	重要	Python 版本 3.8	已经存在	0
数据库	重要	关系型数据库 MySQL	已经存在	0
Java 开发环境	关键	JDK 1.8	已经存在	0
...				

项目实施计划。

小组分工。宋世龙，承担项目经理和 UI 设计师的角色；刘朝晖负责后端框架搭建与功能实现；孙昊，数据库管理员，数据库设计与实现；杨宏宇，前端工程师，负责前端框架搭建与数据交互；闫家毓担任测试工程师兼机动人员。

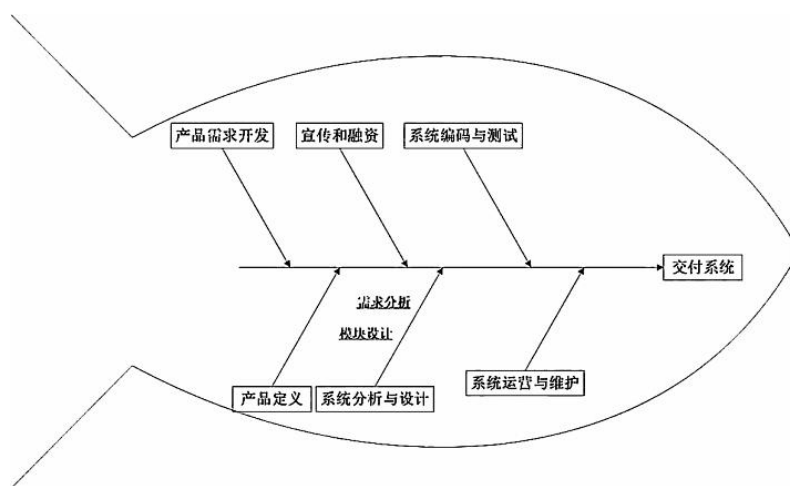


Fig.4 项目实施过程模型

3 需求分析

校园食堂订餐平台的系统用户主要分为两大类用户群体：在校师生和校内食堂平台的主要管理者。其中，在校师生为系统核心用户，核心需求为浏览并选择相应食堂的对应窗口的意愿菜品并下单。食堂以及平台管理人员核心需求为管理和维护用户信息和食堂/窗口/菜品信息。详细内容叙述如下：

在校师生（以下简称为顾客）需注册个人信息，如用户名/学号和密码，填写用户名/学号和密码即可登录使用并进入系统主界面。系统主界面首先展示顾客所在校区的全部食堂信息，选择食堂并点击进入可浏览该食堂内全部营业窗口信息，选择窗口并点击进入可获得该窗口内部全部菜品信息。选中菜品添加至购物车栏，选好下单。顾客可修改自己当前订单信息，如取消/删除订单，可查看自己的历史订单。顾客可维护个人信息，如修改用户名和密码，以及地址信息。

食堂以及平台管理人员（以下简称管理员）输入用户名和密码登录系统。管理员可管理和维护平台用户信息以及食堂/窗口/菜品信息，如创建食堂，新增窗口，修改窗口，删除菜品等等。

除此之外，校园食堂订餐平台还需具备信息安全功能，可以保障顾客的隐私以及其账户安全。通过该应用，顾客可以更加方便快捷地订购食堂菜品，同时管理员也可以管理食堂菜品以及订单信息，方便管理食堂人员构成以及顾客订餐业务。

4 概要设计

校园食堂订餐平台主要包括六大模块，分别为食堂/窗口/菜品模块、顾客订单模块以及系统角色及其地址模块。订餐平台整体结构、功能和系统数据库设计展示如下。订餐平台功能概要设计。

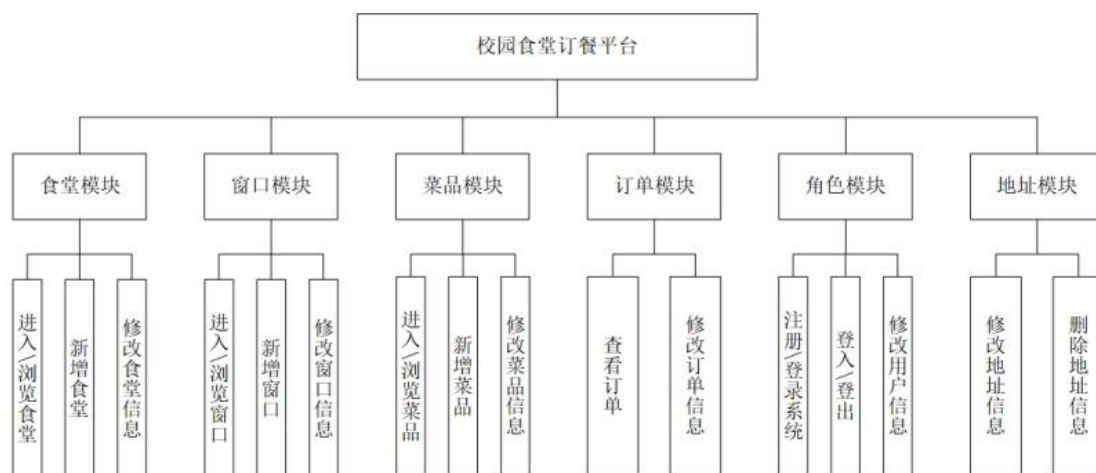


Fig.5 系统功能总体设计

系统数据库简要 ER 图展示如下。

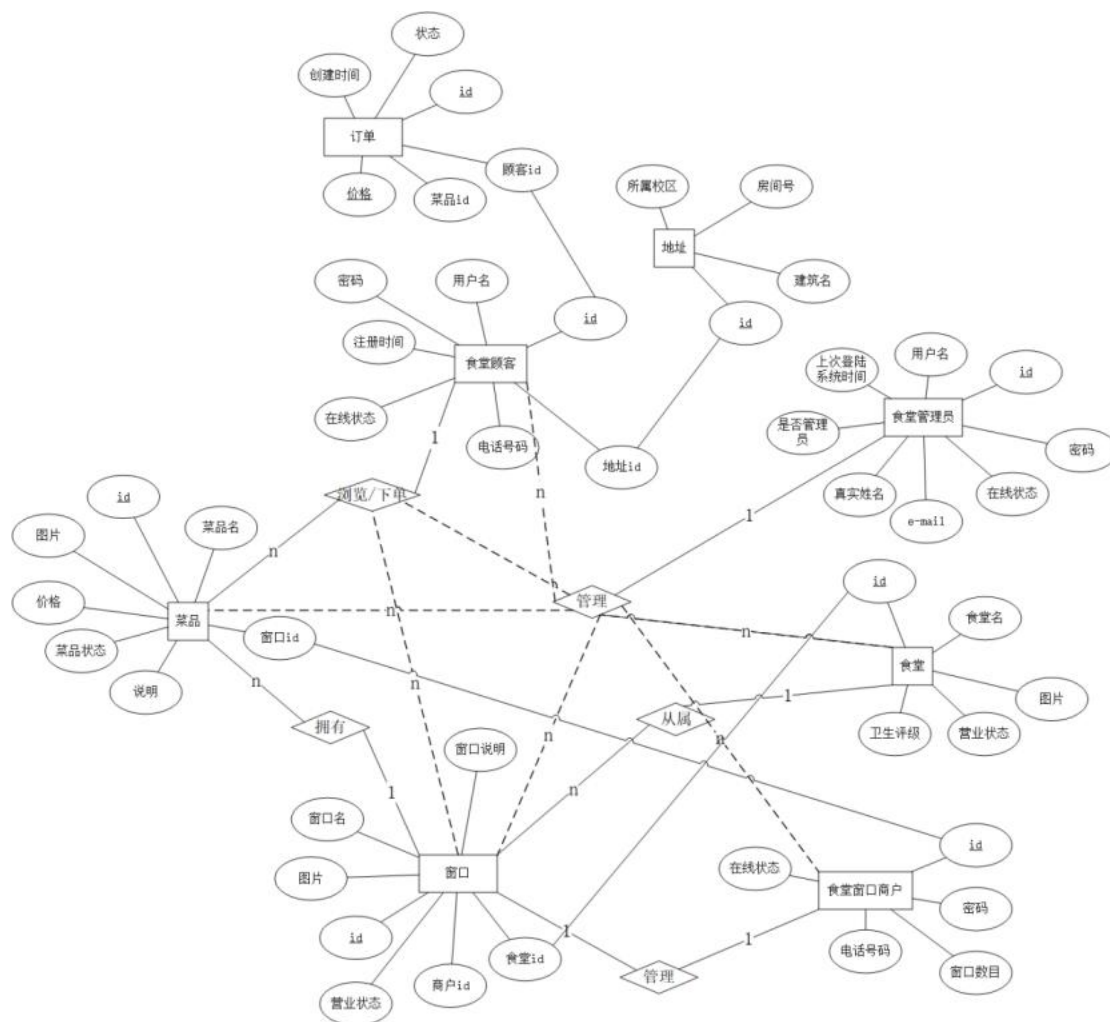


Fig.6 系统数据库简要 ER 图

订餐平台顾客和管理员的简要用例图展示如下。

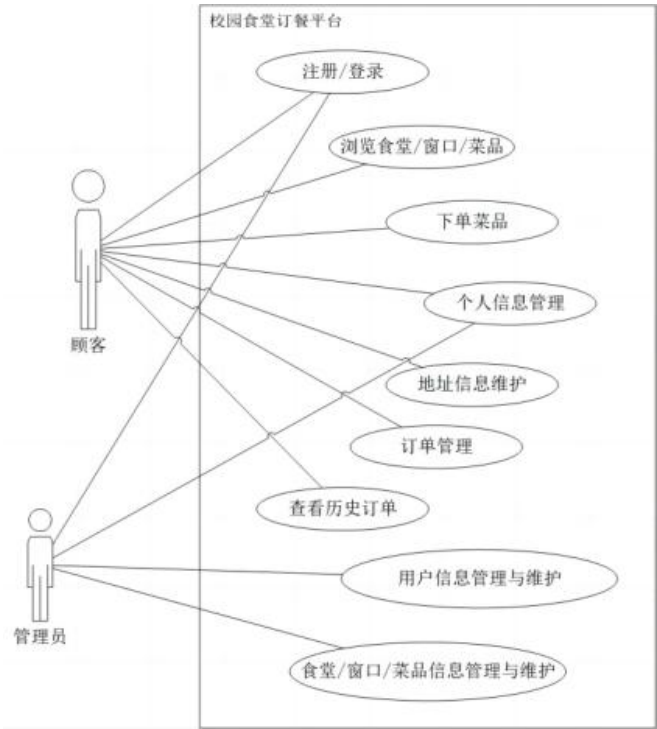


Fig.7 系统简洁用例图

5 详细设计

5.1 前端页面设计

前端页面设计与实现。

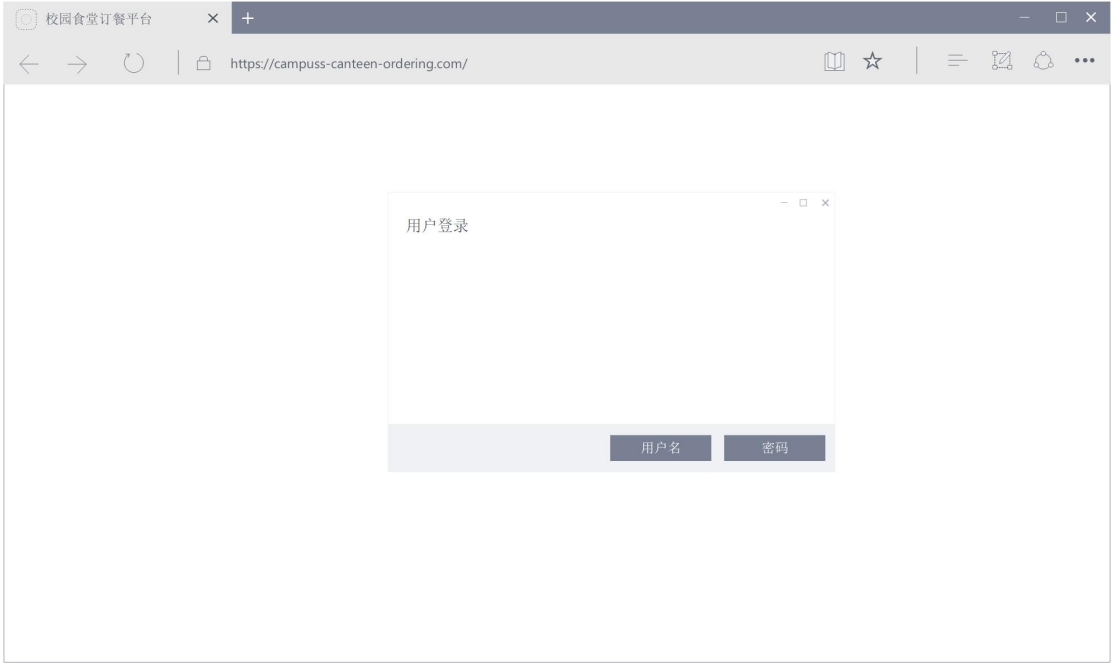


Fig.8 (a) 用户注册/登录界面设计

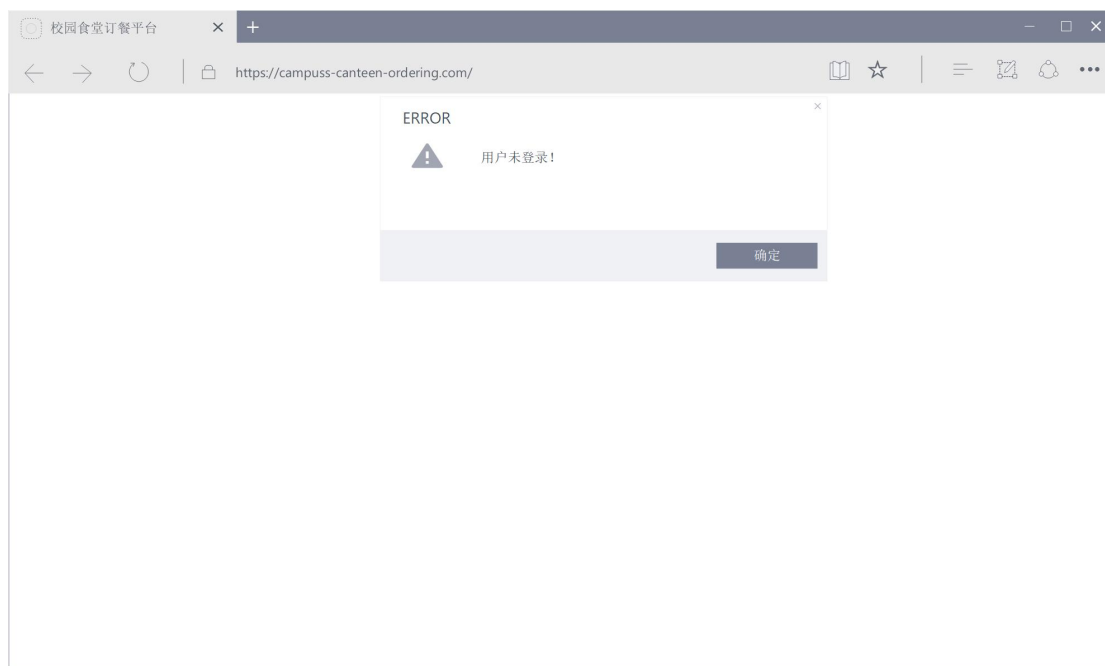


Fig.8 (b) 用户注册/登录界面设计



Fig.9 用户主界面设计



Fig.10 管理员主界面设计

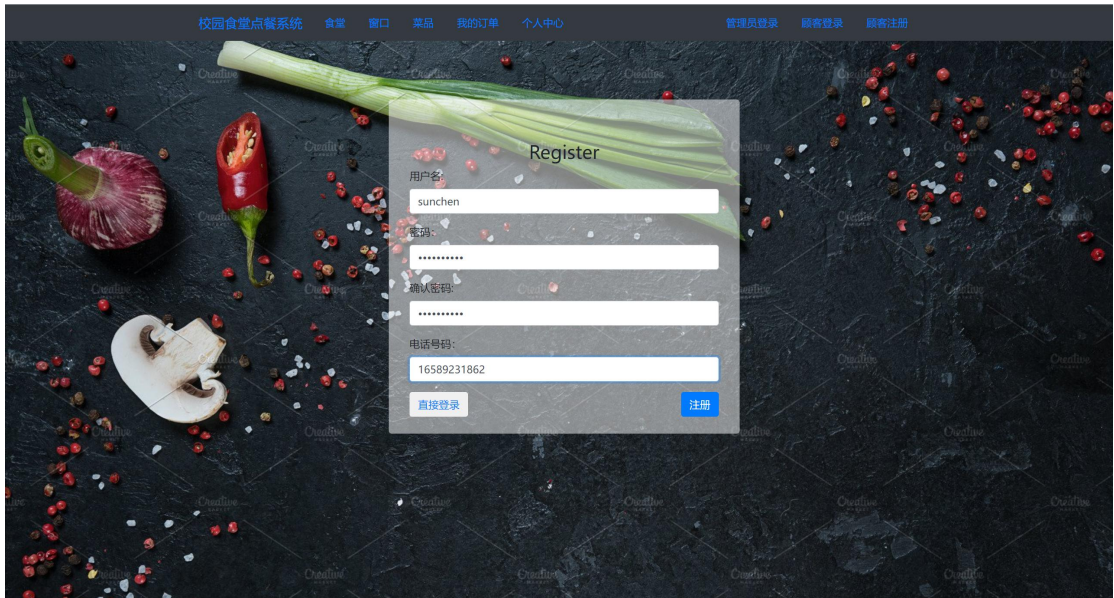


Fig.11 用户登录操作界面

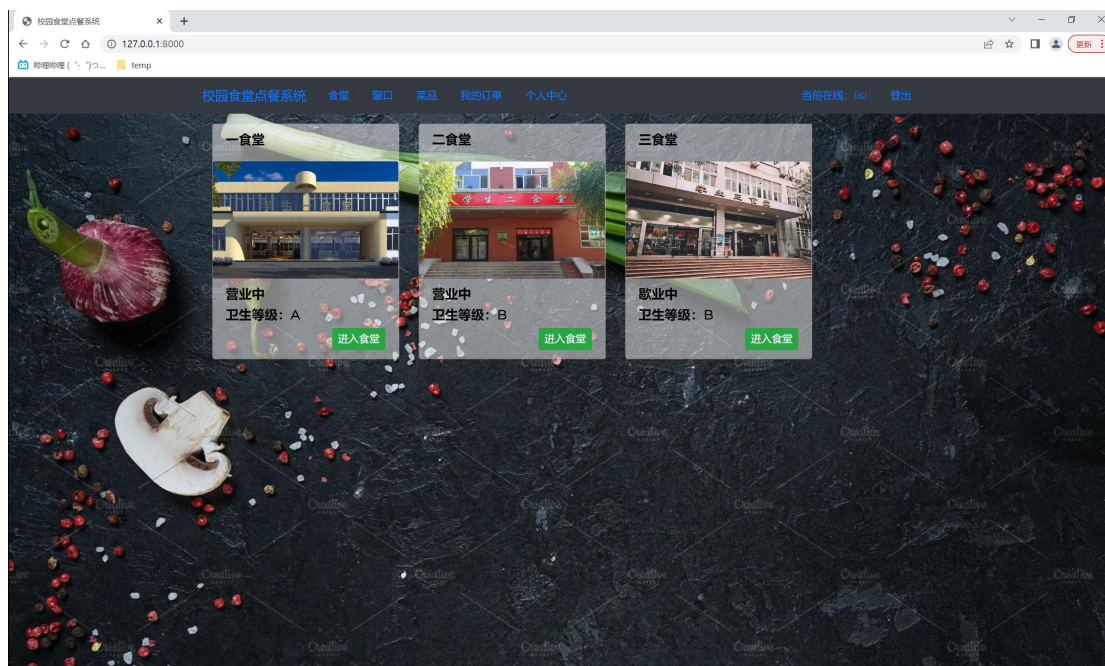


Fig.12 (a) 用户主界面

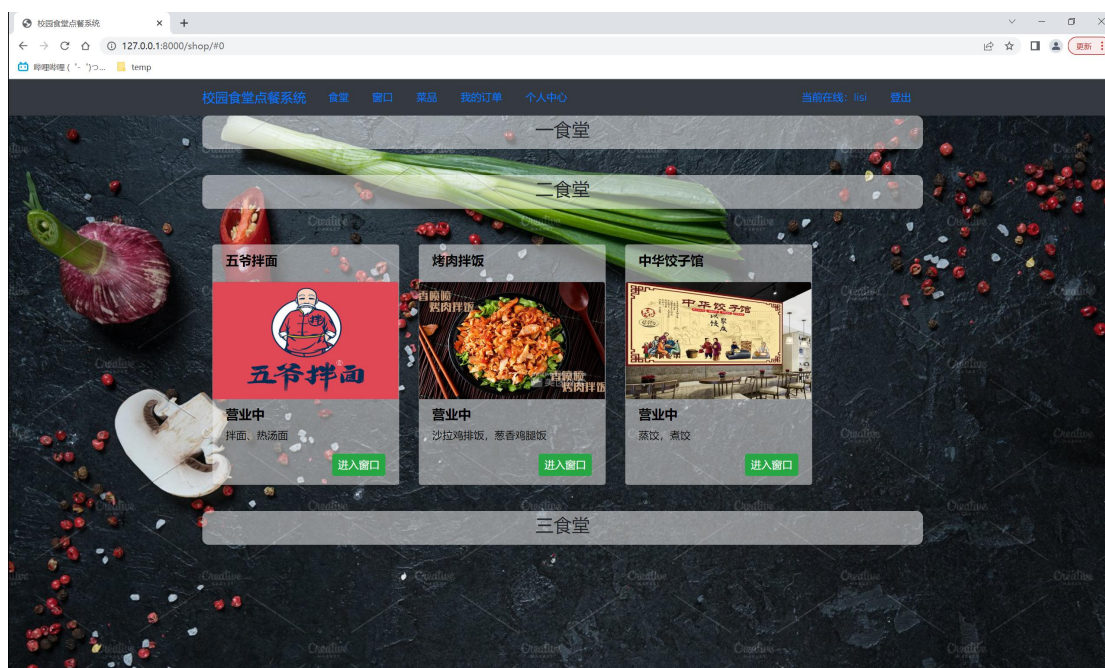


Fig.12 (b) 用户主界面

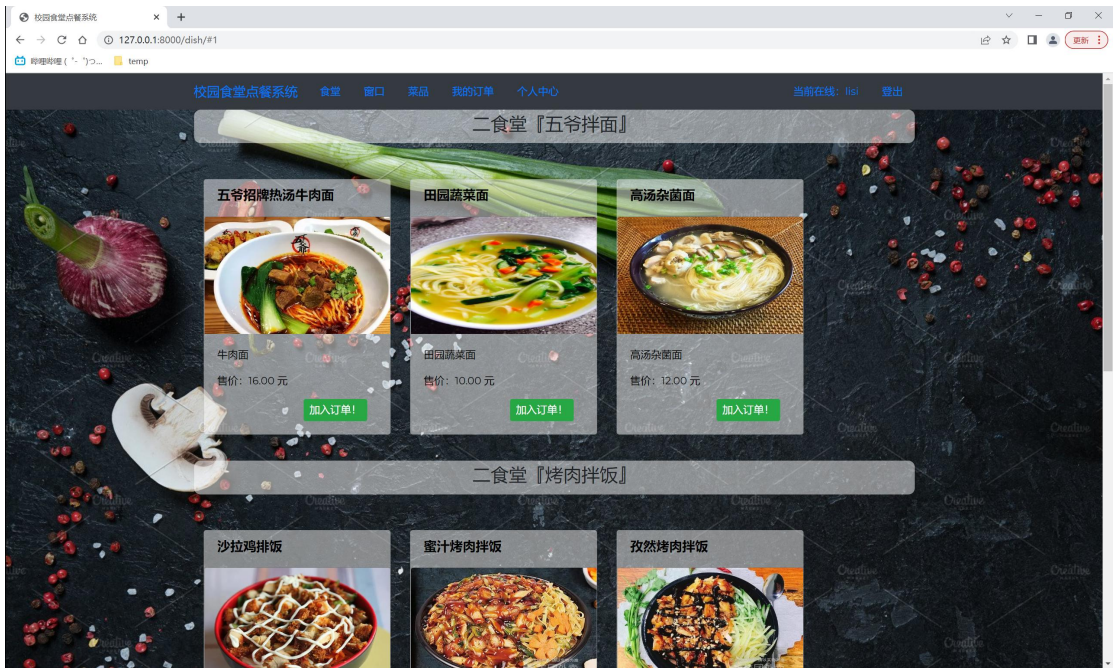


Fig.12 (c) 用户主界面

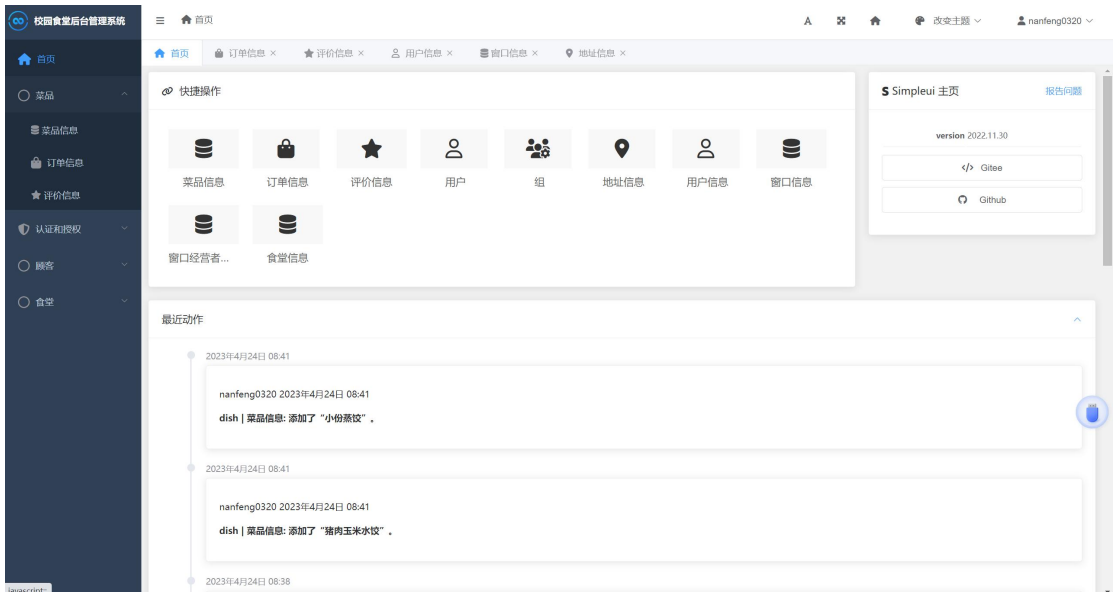


Fig.13 管理员主界面

5.2 数据库设计

数据库设计与实现。

用户密码保护。项目综合使用 PBKDF2（Password-Based Key Derivation Function）算法和 SHA256 算法两种方法对用户密码进行加密操作。

PBKDF2 通过哈希算法进行加密。因为哈希算法是单向的，可以将任何大小的数据转化为定长的“指纹”，而且无法被反向计算。另外，即使数据源只改动了一丁点，哈希的结果也会完全不同。这样的特性使得它非常适合用于保存密码，因为我们需要加密后的密码难以被解密，同时也能保证正确校验每个用户的密码。但是哈希加密可以通过字典攻击和暴力攻击破解。PBKDF2 添加了一个被称为密码加盐（salt）的机制。salt 是一个添加到用户的密码哈希过程中的一段随机序列。这个机制能够防止通过预先计算结果的彩虹表破解。每个用户都有自己的 salt，这样的结果就是即使用户的密码相同，通过加盐后哈希值也将不同。为了校验密码是否正确，我们需要储存 salt 值。salt 通常和密码哈希值一起存放在账户数据库中，或者直接存为哈希字符串的一部分。格式如下：

```
<algorithm>${<iterations>}${<salt>}${<hash>}
```

这些是用于存储用户密码的组件，由美元符号字符分隔，包括：哈希算法、算法迭代次数、随机 salt 和生成的密码哈希。该算法是 Django 可以使用的许多单向哈希或密码存储算法之一。迭代次数描述了算法在哈希上运行的次数。salt 是使用的随机序列，hash 是单向哈希算法的结果。

SHA 算法是美国的政府规范算法，SHA 在很多安全协定中广为运用，包含 TLS 和 SSL、PGP、SSH、S/MIME 和 IPsec，曾被视为是 MD5 加密算法的后继者。SHA 家族的五个算法，分别是 SHA-1、SHA-224、SHA-256、SHA-384 和 SHA-512，由美国国家安全局（NSA）规划，并由美国国家规范与技能研究院（NIST）发布。后四者有时并称为 SHA-2。SHA-2，名称来自于安全散列算法 2（英语：Secure Hash Algorithm 2）的缩写，一种密码散列函数算法标准，是 SHA-1 的后继者。SHA-2 族有着更多位的输出哈希值，破解难度更大，能够提供更高的安全性，目前尚未出现对 SHA-2 有效的攻击。

系统数据库部分字段设计。

顾客地址表。

```
address_id  district  building  room
```

用户信息表。

```
id  password  last_login  is_superuser  username  first_name  last_name
email  is_staff  is_active  date_joined
```

食堂信息表。

canteen_id canteen_name canteen_photo sanitation_level canteen_active

顾客信息表。

customer_id customer_name customer_passwc customer_tel customer_status create_time address_id

菜品信息表。

dish_id dish_name dish_detail dish_price dish_photo dish_active shop_id

订单信息表。

order_id order_price order_status order_time customer_id dish_id

窗口信息表。

shop_id shop_name shop_detail shop_photo shop_active canteen_id manager_id

商户信息表。

manager_id manager_name manager_passwo manager_tel manager_status manage_shop_num

5.3 业务逻辑设计

业务逻辑设计与实现。食堂/窗口/菜品模块面向两个用户群体，分别是顾客和管理员。其中，顾客可以浏览食堂/窗口/菜品并在相应的食堂/窗口下单。管理员可以浏览和查阅食堂/窗口/菜品并对其进行修改和维护操作。

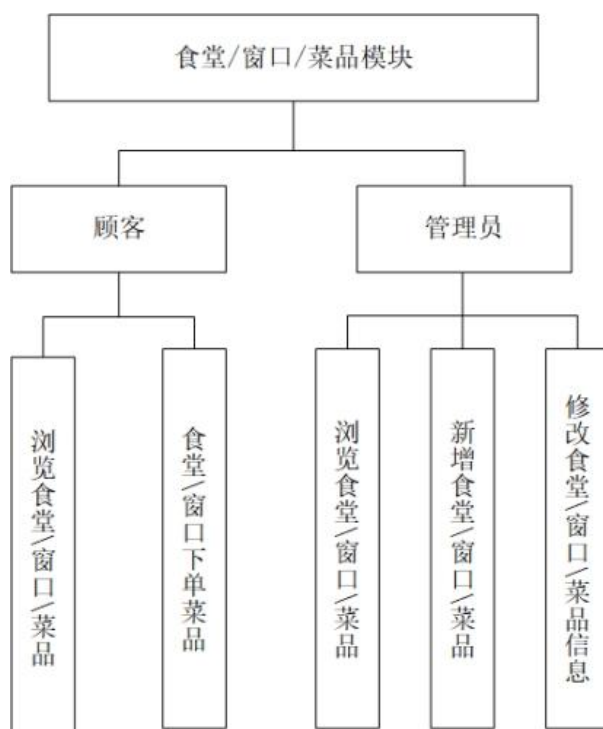


Fig.14 食堂/窗口/菜品模块功能详细设计

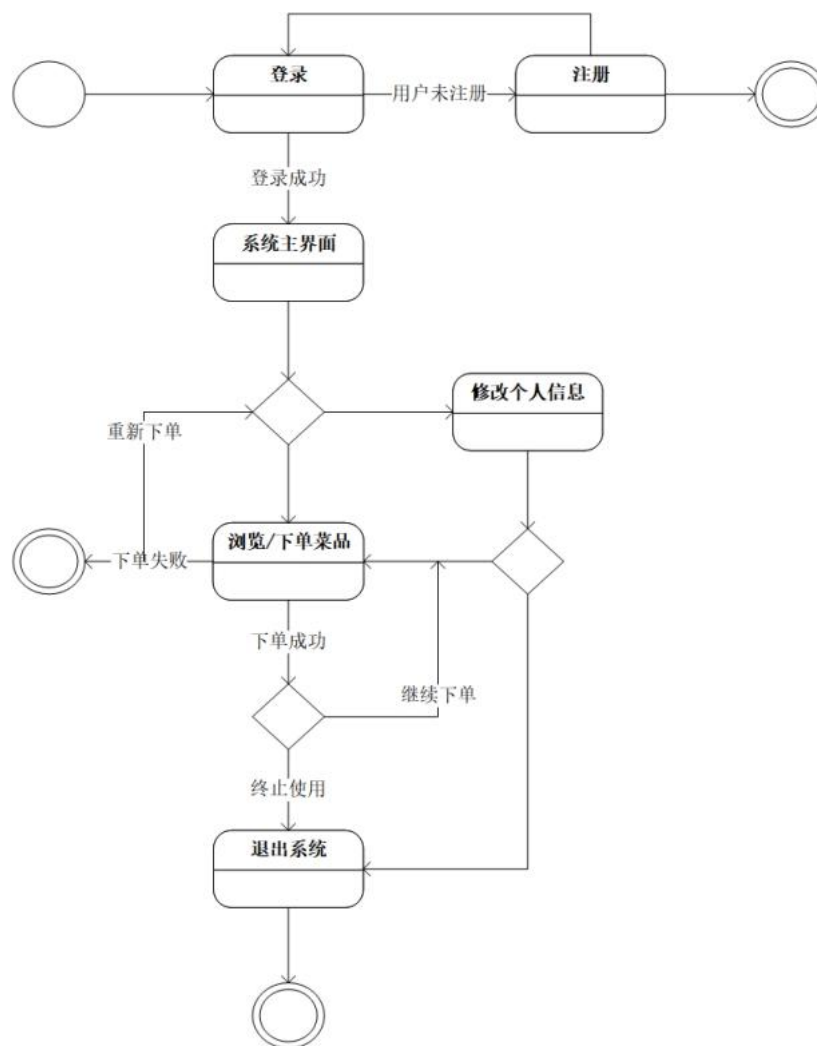


Fig.15 用户核心操作状态图

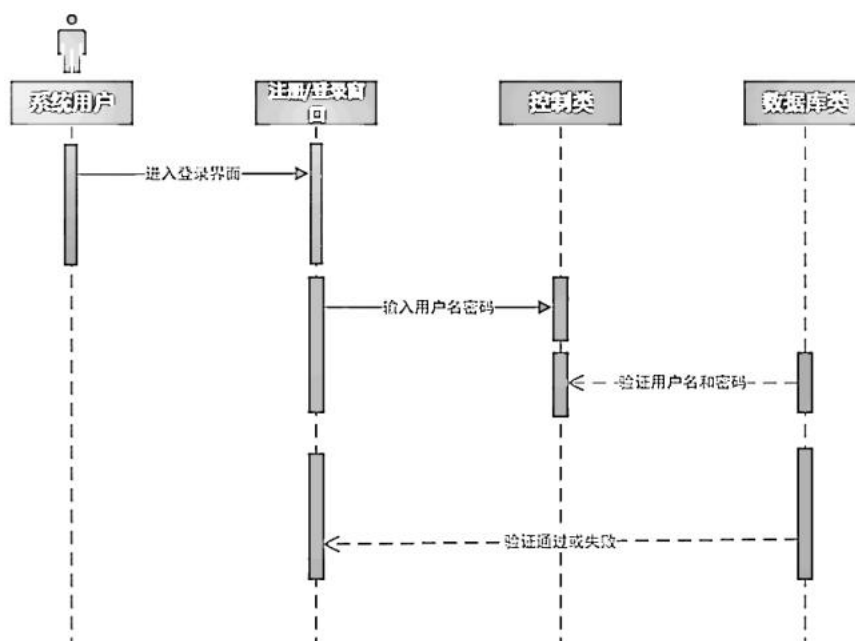


Fig.16 用户注册/登录时序图

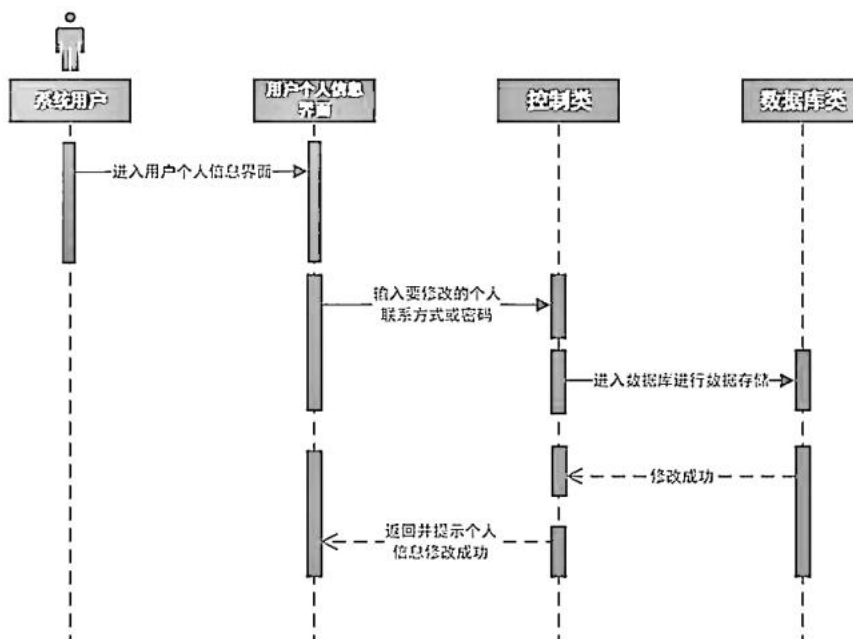


Fig.17 用户修改个人信息时序图

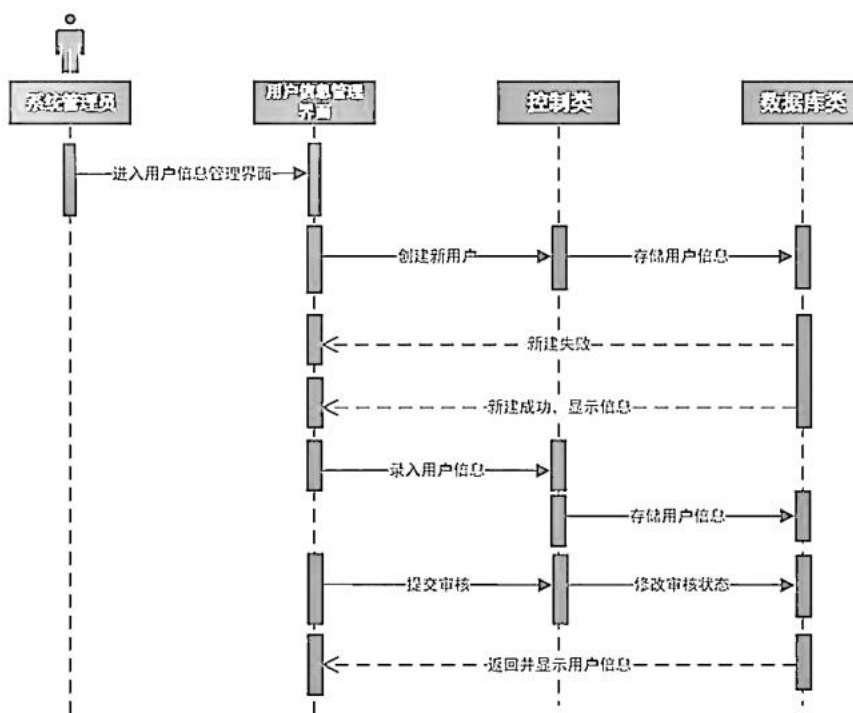


Fig.18 系统管理员部分操作时序图

6 测试计划

测试环境部署流程。测试环境搭建的中流程大致可以总结为：准备工作、安装、配置和试运行或者环境准备，包部署，配置修改。

Tab.3 单元测试计划

单元测试范围	代码单元输入和期待输出测试、边界值的测试、异常测试	
单元测试方法	用例设计测试	
单元测试环境	Mockito 示例	
测试辅助工具	EMMA 分析工具	
测试完成准则	所有的包和类的覆盖率必须达到 80%以上	
将产生的文档	单元测试用例，测试报告等	
单元测试任务 / 优先级	进度	人员与工作描述
单元测试用例设计	20 天	设计测试用例
评审	20 天	对测试结果审核
...		

Tab.4 集成测试计划

集成测试范围	提供一个可以测试使用的版本，让用户和开发人员同时看到相同的功能	
集成测试方法	利用一系列的工具有，方法和规则，做到快速的构建开发代码，自动的测试化，来提高开发代码的效率和质量	
集成测试环境	Android Studio	
测试辅助工具	IBM Rational Team Concert	
测试完成准则	集成测试报告结果符合标准值	
将产生的文档	集成测试用例，测试报告等	
集成测试任务/优先级	进度	人员与工作描述
自动构建	20 天	RTC 服务器提供构建结果报告
自动单元测试	20 天	每一次的构建，都可以把这些检查工作自动的进行一遍测试
...		

Tab.5 接口测试用例说明

接口 A 的函数原型	登录	
输入/动作	期望的输出/相应	实际情况
典型值...	返回正确结果,数据库中更新用户信息正确.	与期望输出结果一致
在正确数据类型下输入用户名和密码		
边界值...	无法返回正确结果且数据库中无数据更新。	与期望输出结果一致
正确的用户名和空白的密		

码		
异常值...	无法返回正确结果数据库中的无数据更新	与期望输出结果一致
正确的用户名和错误密码		
接口 B 的函数原型	注册	
输入/动作	期望的输出/相应	实际情况
典型值...		
在正确的数据类型下输入账号密码和用户身份	返回正确结果, 数据库中更新用户信息正确.	与期望输出结果一致
边界值...	无法返回正确结果且数据库中无数据更新。	与期望输出结果一致
已存在的正确用户和密码		
异常值...	无法返回正确结果且数据库中无数据更新。	与期望输出结果一致
错误的用户名类型		
...		

Tab.6 功能测试用例描述

功能 A 描述	用户登录功能	
用例目的	检测用户能否正常进行登录	
前提条件	用户登录接口正常	
输入/动作	期望的输出/相应	实际情况
示例：典型值...		
正确数据类型的用户名及用户相应的密码	用户登录成功	与期望输出一致
示例：边界值...		
正确的用户名及空白的密码	请输入密码	与期望输出一致
示例：异常值...		
正确的用户名及错误的密码	密码错误	与期望输出一致
功能 B 描述	用户注册功能	
用例目的	测试用户能否正常的注册账号密码	
前提条件	注册接口正常	
输入/动作	期望的输出/相应	实际情况
典型值...		
在正确的数据类型下输入账号密码和用户身份	注册成功	与期望输出一致

边界值...	注册失败，用户已存在	与期望输出一致
已存在的正确用户和密码		
异常值...	用户名不符合规则	与期望输出一致
错误的用户名类型		

7 项目总结

餐饮业原本就是一种个性化、多样化的服务产业，随着网络技术的发展进步和普及应用，方便、快捷、个性化的网上订餐不断深入人们的生活，尤其在生产生活和学习工作节奏加快和时间观念加强的群体中网上订餐成为他们的喜爱方式。人们通过上网，对菜品进行预订，这样不仅节约了时间加速了流程，也为食堂的订餐管理工作提供了方便。本文描述的校园订餐平台将在校师生以及食堂管理者两类群体纳入一个系统，将会使得传统相关业务更加方便快捷，简洁高效。