****

**2021届毕业设计说明书**

世纪餐厅管理系统的设计与实现

院 、 部： 计算机与信息科学学院

学生姓名： 石玮

指导教师： 徐艳山 职称 讲师

专 业： 信息与计算科学

班 级： 信息1702

完成时间： 2021-5

第一指导教师：徐艳山

摘 要

当一家餐厅的规模足够大时，餐厅的管理就很难靠人工完成，为了更加有效率的对餐厅进行管理，不少餐厅老板都将信息化技术引入到餐厅管理当中，而世纪餐厅管理系统就是这样一款帮助管理者高效进行餐厅管理的系统，能有效提高员工之间的信息传递，帮管理者掌握餐厅状况，从而在面对突发状况时，能及时做出反应，商量好对策。

本文介绍的世纪餐厅管理系统，是以帮助管理者高效管理餐厅为主要目的，对员工信息的记录、餐厅的信息记录、如何记录对管理者有帮助的信息、餐厅产生突发状况时第一时间通知管理者进行了研究与探索。

目前我国的餐厅管理系统正处于起步阶段，通过对一些著名的餐厅管理系统的研究，分析其系统的功能设计，特点，并结合了对一些小型餐厅的需求调研，开发了这款适合大部分餐厅的管理系统，人员管理方面采用树状模式，老板管理、分配各员工权限，所有上级领导都可以对下属进行自己的权限分配与管理。此外还加入了流水记录功能，帮助分析流水的涨幅情况。

本系统采用传统B/S架构，使用JAVA语言进行开发，采用了目前流行的SpringBoot开发框架，该框架的好处就是自带集成Tomcat服务器，并且不需要繁琐的配置，节省用户学习成本。客户端为web应用，随时随地都能使用。自动集成SSM框架，运行效率更高。数据存储方面也使用了时下比较热门的MySQL关系型数据库，所有的表格的关系模式为二范式。Web页面采用了bootstrap架构，页面简洁，响应迅速。

本文主要介绍世纪餐厅管理系统是如何实现的，针对市场需求的调研，确定了餐厅管理系统有一定的市场需求。而本系统提供了一套默认的管理体系，客户可在此基础上进行自主扩展，本文主旨为如何在餐厅管理中应用信息数字化技术，来提高管理效率。通过JAVA的面相对象特性，将部门、员工、餐厅结合在一起，使其部门之间相互独立的同时，上下级之间的联系又密不可分，使每个人的分工明确 。

关键词 ：人事管理；SSM框架；信息数字化；二范式；面相对象

ABSTRACT

When the scale of a restaurant is large enough, the management of the restaurant is difficult to be completed manually. In order to manage the restaurant more efficiently, many restaurant owners introduce information technology into the restaurant management. Century restaurant management system is such a system to help managers manage the restaurant efficiently, which can effectively improve the information transmission between employees, Help managers grasp the situation of the restaurant, so that in the face of emergencies, they can make timely response and discuss the countermeasures. This paper introduces the century restaurant management system, which aims to help managers manage the restaurant efficiently. It studies and explores the record of employee information, restaurant information record, how to record helpful information for managers, and how to inform managers immediately when the restaurant has an emergency.

At present, the restaurant management system in our country is in its infancy. Through the research of some famous restaurant management systems, this paper analyzes the function design and characteristics of the system, and combined with the demand research of some small restaurants, develops this management system suitable for most restaurants. In terms of personnel management, the tree mode is adopted, and the boss manages and assigns the authority of each employee, All superior leaders can assign and manage their own authority to subordinates. In addition, the flow record function is added to help analyze the rise of flow.

The system uses the traditional B / S architecture, using java language for development, using the current popular springboot development framework, the advantage of the framework is that it has integrated Tomcat server, and does not need cumbersome configuration, saving user learning costs. The client is a web application, which can be used anytime and anywhere. Automatic integration of SSM framework, higher efficiency. In the aspect of data storage, the popular MySQL relational database is also used. The relational mode of all tables is two normal forms. The web page adopts bootstrap architecture, which makes the page concise and responsive.

This paper mainly introduces how to realize the century restaurant management system. According to the investigation of the market demand, it is determined that the restaurant management system has a certain market demand. This system provides a set of default management system, customers can expand on this basis, the main purpose of this paper is how to apply information digital technology in restaurant management to improve management efficiency. Through the face object characteristics of Java, the departments, employees and restaurants are combined together, so that the departments are independent of each other, and the relationship between the upper and lower levels is inseparable, so that everyone's division of labor is clear 。

Key words: personnel management; SSM framework; Information digitization; Two paradigms; Face object

目 录

[1 绪论 1](#_Toc30147)

[1.1选题背景 1](#_Toc30666)

[1.2国内外研究现状 1](#_Toc5650)

[1.2.1国内研究现状 1](#_Toc4564)

[1.2.2 国外研究现状 2](#_Toc4020)

[1.3设计内容和意义 2](#_Toc9168)

[1.3.1 设计内容 2](#_Toc23223)

[1.3.2 意义 3](#_Toc18176)

[1.4本文结构 3](#_Toc20247)

[2开发工具及相关技术简介 4](#_Toc4139)

[2.1 intelliJ IDEA 4](#_Toc16467)

[2.2 JDK1.8 4](#_Toc5982)

[2.4 Navcat premium 4](#_Toc22183)

[2.5 Visual Studio Code 4](#_Toc6847)

[2.6 Git 5](#_Toc6930)

[2.7 java 5](#_Toc31992)

[2.8 Spring Boot 5](#_Toc5267)

[2.9 Bootstrap 5](#_Toc9227)

[2.10 SSM架构 5](#_Toc2111)

[3系统分析 7](#_Toc9084)

[3.1 需求分析 7](#_Toc15139)

[3.2组织结构分析 7](#_Toc13637)

[3.3执行流程分析 8](#_Toc19466)

[4 系统设计 9](#_Toc24301)

[4.1功能模块设计 9](#_Toc11423)

[4.2数据库设计 10](#_Toc19628)

[4.3用户界面设计 14](#_Toc15497)

[4.3.1 登录界面 14](#_Toc31383)

[4.3.2 系统主界面 14](#_Toc24173)

[4.3.3 员工管理界面 15](#_Toc17434)

[4.3.4 部门管理界面 16](#_Toc24060)

[4.3.5 商品管理界面 16](#_Toc29768)

[4.3.6客户个人界面 17](#_Toc30585)

[4.4 本章小结 17](#_Toc1496)

[5系统实现 18](#_Toc5205)

[5.1 登录验证功能模块 18](#_Toc30653)

[5.2 查询管理模块 20](#_Toc11738)

[5.3 其他功能模块 26](#_Toc951)

[5.4本章小结 29](#_Toc3576)

[6 系统测试 30](#_Toc11867)

[6.1测试目的 30](#_Toc10162)

[6.2测试方法 30](#_Toc9308)

[6.3测试用例 30](#_Toc19391)

[7 总结 32](#_Toc14112)

[参考文献 33](#_Toc25429)

# 1 绪论

## 1.1选题背景

帮助管理者更加轻松的掌控大局，提高员工的工作效率，为此我开发了这款名为世纪餐厅的管理系统。

本系统基于Spring Boot框架开发，java语言编写，使得该应用程序能在各个平台上运行，且Spring Boot框架已集成Tomcat服务器，能做到安装即可用，且操作简单，大大降低管理者的学习成本。

Spring Boot框架是目前java开发的主流框架，其优点在于省去了繁杂的配置文件，内置servlet容器，只需java环境就能运行，便于客户使用的同时也使代码维护变得简单。

Web页面采用了HTML5技术，使用了thymeleaf来进行前端的数据展示，ajax、jQuery来进行与后台的交互,动态的展示数据库内容。

数据库方面采用流行的MySQL数据库，InnoDB存储引擎，对数据进行一个较为安全的管理，同时MySQL具有体积小用量大等优势。

## 1.2国内外研究现状

### 1.2.1国内研究现状

中国虽然是举世闻名的传统美食强国,拥有五千年的传统饮食文化和巨大的餐饮消费市场,随着中国人民的生活物质水平和其生活方式的巨大转变[2]。中餐因菜品的多样化和特色服务很难实现标准化管理[1]。餐饮行业已经具有巨大的创新性和投资市场,被誉为当代中国的一个重要黄金产业，但同样我们也应该清醒地看到，餐饮行业不仅在发展中面临着巨大的市场机遇；也正在面临着前无法预期的所未来挑战与考验。这些挑战主要来源于以下几方面 ：

（1）人才的专业化程度不足导致管理困难:因餐饮业门坎较低，中国的大多数餐饮企业的老板都是白手起家，从小型餐馆做起，习惯了人工管理的居多，以致于当餐厅日益壮大后，出现管理困难的局面，并没有设计一套智能化的人员管理系统，所以我国的餐饮行业的管理方式无论是从观念意识、经营思想还是管理水平都有待专业化、系统化。

（2）信息传递效率差，信息获取滞后,许多中小型餐饮行业没有一套完善的、智能化的信息系统，导致信息获取滞后没有及时应对市场冲击带来的风险。人员的管理也没有一套系统的模式，导致出现事故时缺少应对方案；没有合理安排员工工作，员工之间的工作时间比例不平衡

### 1.2.2 国外研究现状

国外由于科技发展迅速，管理意识超前，加上起步早，已经形成了标准化、系统化的工作流程和人员管理。如KFC.必胜客等，从原材料采购、收银、点菜、财务、成本都已形成了一定的标准化模式，信息化。高科技程度高，能对菜品的选择进行科学的调控，剔除销量不好的产品。对热销产品进行总结，继续开发新的产品，给决策机构提供了很多素材，对制定相应的政策提供了帮助。高度系统化的人员管理，每个人的分工明确，使管理人员能及时下打任务指令到每个岗位，极大提高管理效率，减少经营成本。

## 1.3设计内容和意义

### 1.3.1 设计内容

本系统主要是为了帮助餐饮行业更好的进行人员管理而设计的一套管理系统，其主要功能侧重于人员管理，包括但不限于，员工信息管理（新增、修改、删除），餐厅的营收记录，菜品信息管理（新增、修改、删除）等。

服务端采用MVC三层架构：

（1）Model层：用来处理业务逻辑，比如添加世员工信息、上传头像、展示餐厅流水。这一层是比较重要的一层，任何动作都在这一层完成，业务逻辑越复杂，代码量越多。

（2）View层：将后台的数据、动作展示到页面上，是比较简单的一 层，代码量较少。

（3）Controller层：获取前端请求，返回给后台，并将后台的数据传 递给前台，其传输信息的工作，代码量较少，但也尤为重要。

数据库方面，根据调研所得到的信息建立本系统所需要的表格，表格关系模式采用二范式，部分采用三范式，BC泛式。并且加入索引使查询更加快捷。

界面设计采用菜单栏加标签页的形式，使整个操作在同一个页面进行，不用打开新的网页。在表格的上方添加条件搜索，每行后面添加编辑功能。网页还添加了换肤、全屏、锁定等小功能。

### 1.3.2 意义

如今餐饮行业在我国发展的蒸蒸日上，规模也越来越大，再加上人们的生活越来越智能化，一直依赖于人工管理餐厅是不可取的，不仅效率低下，成本也高，据中国产业信息网发布的《[2016-2022年中国餐饮行业市场专项调研及投资前景评估报告](https://www.chyxx.com/research/201508/334970.html)》显示，沿海城市大部分餐厅均已实现信息数字化管理，而不是采用传统的人工管理，大大的提高了工作效率和营业额。人工管理的方式已经限制了餐厅的营业额，并且随着餐厅规模的壮大，弊端愈加明显。

21世纪之后，信息技术的发展也推动了经济的快速增长[5]，信息技术的应用使得信息之间的传递变得更加迅速，能有效提升管理人员的监管效能，为管理人员提供方便快捷的信息交流平台，信息不再具有滞后性，这样就能大大提高餐厅管理的效率，腾出更多时间处理其他事务，从而提高营业额。可谓是一举多得。

## 1.4本文结构

世纪餐厅管理系统设计说明书，针对本管理系统的设计思维及实现方法作出了一个系统的介绍，共分为七个章节，整体结构如下：

第一章为绪论，主要介绍本人设计本系统在国内外的发展现状，说明本系统的研究意义，对餐厅的经营能起到什么作用，以及介绍本文的大致梗概。

第二章为本系统所用到的开发工具及相关技术的简单介绍，介绍了Java在开发web应用中的优势，介绍了SpringBoot框架的特点，介绍了为什么本系统要使用MySQL数据库，而不是其他数据库。

第三章为系统分析：根据对市面上各餐厅管理系统的功能调查以及我国餐厅数字化管理系统的覆盖率。确定了本系统的功能。

第四章为系统设计：根据系统分析所得出的功能需求，设计相应的功能模块，设计相应的数据库结构，并描述其工作流程。

第五章为系统实现：介绍本系统的各个功能模块的实现界面和实现方法。

第六章为系统测试：介绍了对整个系统的测试流程、测试目的、测试方法。

第六章为总结：说明本文的主要内容并总结本人在这次项目中遇到的困难及收获。

# 2开发工具及相关技术简介

本系统所用到的开发工具有intelliJ IDEA-2018， Navcat Premium，使用到的技术有SSM框架，Spring Boot框架。

## 2.1 intelliJ IDEA

IDEA 全称 IntelliJ IDEA，是一款java编程语言开发的应用软件。IDEA是业界被公认为最好的java开发工具之一，以其酷炫的ui，智能的代码提示、简洁的系统设置、强大的插件系统、JavaEE支持、各类版本控制工具(github、svn等)、CVS整合、代码自动排版等功能赢得了广大程序员的喜爱，相比于其他工具，IDEA运行代码的速度更快，debug更为方便，而且界面的布局也很合理让人感觉很舒服，对于刚上手开发的萌新也是较为友好的，每个功能的提示都很到位，还能自定义文件格式，每新建一个文件就能按照设定好的格式自动生成相应的内容，省去了很多公共的代码编写。

编码方面，有这非常方便的代码提示，能在程序员一边写代码一边提示可能需要用的类名方法等。并且能自动生成一些常用的代码，重写父类方法，帮助程序员更高效的进行开发。还具有maven项目管理工具，对项目进行维护，管理jar包版本。还具有Git工具能连接远程仓库对项目进行版本管理及控制。

## 2.2 Navcat premium

Navicat premium是一款数据库管理工具,其简结的ui设计深得大众喜爱，拥有独特的视觉化sql语句创建，根据表格能自动生成对应的sql语句以及er图，非常便捷。并且支持同时连接多个数据库，支持的数据库类型也非常多，还能直接操作数据库数据。能多开窗口，多线操作，实时提交回滚数据。也能运行sql文件，提示sql语句的错误。还能连接数据库的控制台，对数据库进行直接操作。

不止支持MySQL数据库的功能还支持大部分Oracle、SQLserver的功能，如事件，函数，检视等。不同的资料库的批次处理任务也可以设定排程并在指定的时间执行。其他功能包括汇入/汇出精灵、查询建立工具、报表产生器、资料同步、备份、工作排程及更多[19]

## 2.3 Spring Boot框架

Spring Boot基于Spring4.0设计，省去Spring繁琐的参数配置，做到了到手即用的特点，内置Tomcat服务器，只需启动Application应用类就能启动服务，帮助开发人员快速开发、快速整合、配置简化。并提供了一系列大型项目通用的非业务性功能。通过在YML文件进行一些系统配置例如服务器的端口、服务名称，除了对SpringBoot的一些注册中心、网关、路由等一些组件进行配置以外，还能对数据库驱动、数据库连接属性、Kafka、Redis等组件进行配置。

由于是基于Spring设计的，那么Spring所拥有的注解、依赖注入也是能用的，并且能支持更多类型的注解，功能更加强大。并且SpringBoot能将功能模块化，各功能模块之间互不干扰，做到了低耦合，高聚合。是目前最好用的开发框架之一。

## 2.4 SSM架构

SSM框架整合了spring MVC ，Spring和Mybatis三大框架，采用标准的Model 、View、Controller三层架构模式，使用Spring MVC来完成请求的转发和视图管理，spring实现业务对象管理，mybatis作为数据对象的持久化引擎，具体流程如下:

1. 前端发送请求到前端控制器，前端控制器接受请求后，会转给处理器映射器。
2. 处理器映射器收到请求后，根据其请求类型，返回对应的Handler给前端控制器。
3. 前端控制器收到Handler之后，在转给处理器适配器请求执行Handler。
4. 处理器适配器会根据Handler的类型分配给对应的处理器。
5. 处理器执行完毕后返回ModelAndView，原路返回传给前端控制器，前端控制器在发送给视图解析器解析ModelAndView。
6. 视图解析器解析完成后，返回视图给前端控制器，前端控制交给视图渲染器渲染视图后就返回给前端完成视图展示。

这一套流程保证了各环节出错后能精准定位到出错的位置，还能及时中断流程，避免错误影响范围的扩大。

# 3系统分析

## 3.1 需求分析

（1）做为一个人员管理系统，最主要的功能则是员工信息的增删改查，要求实现对员工的基本信息如：性别、年龄、姓名、岗位、电话号码等做些简单的操作。

（2）要求显示餐厅的年、月、日收入与支出。

（3）要求能显示部门信息，能新增、删除、修改部门。

（4）要求能具有商品信息管理、商品信息查询、库存查询，一遍采购部门及时补充商品。

（5）要求有登录界面，但不开放注册功能，员工账号由上司开通，老板账号由开发者开通。

（6）要求能实时监控在线人数。

（7）要求各级领导的权限各不相同，权限与级别挂钩。

（8）要求上级能对下级的权限进行修改。

## 3.2组织结构分析

目前系统自定义的有一套组织结构模板，客户也可自主定义

本餐厅组织结构为：餐饮部、财务部、人力资源部、市场部

其结构图为：

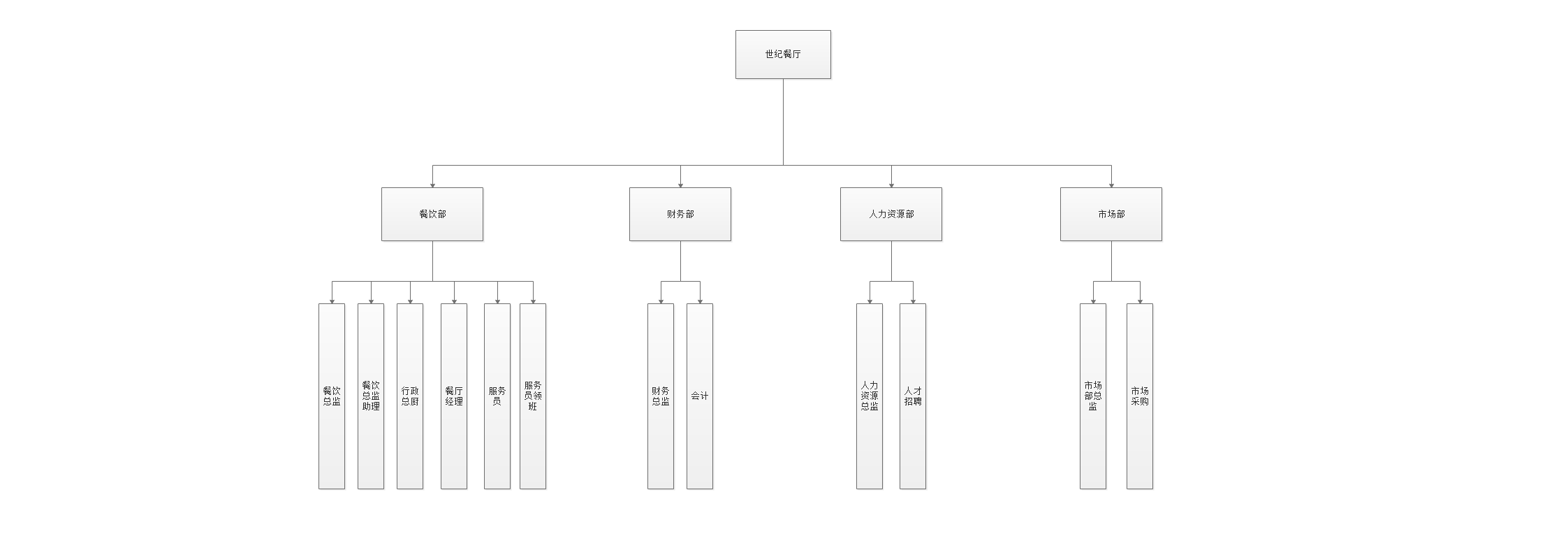


图 1 餐厅部门结构图

其中老板作为最高权限拥有者，自然是统领所有部门，在这之下分为了四大部门：餐饮部、财务部、人力资源部、市场部。这是系统默认的餐厅结构，后续可根据客户需求增加部门。考虑到本系统为餐厅管理系统，所以餐饮部下面分配的职位较多，其他部门的职位也可以根据客户需求进行添加

## 3.3执行流程分析

针对传统管理系统的流程分析，本人设计了如下的流程图：

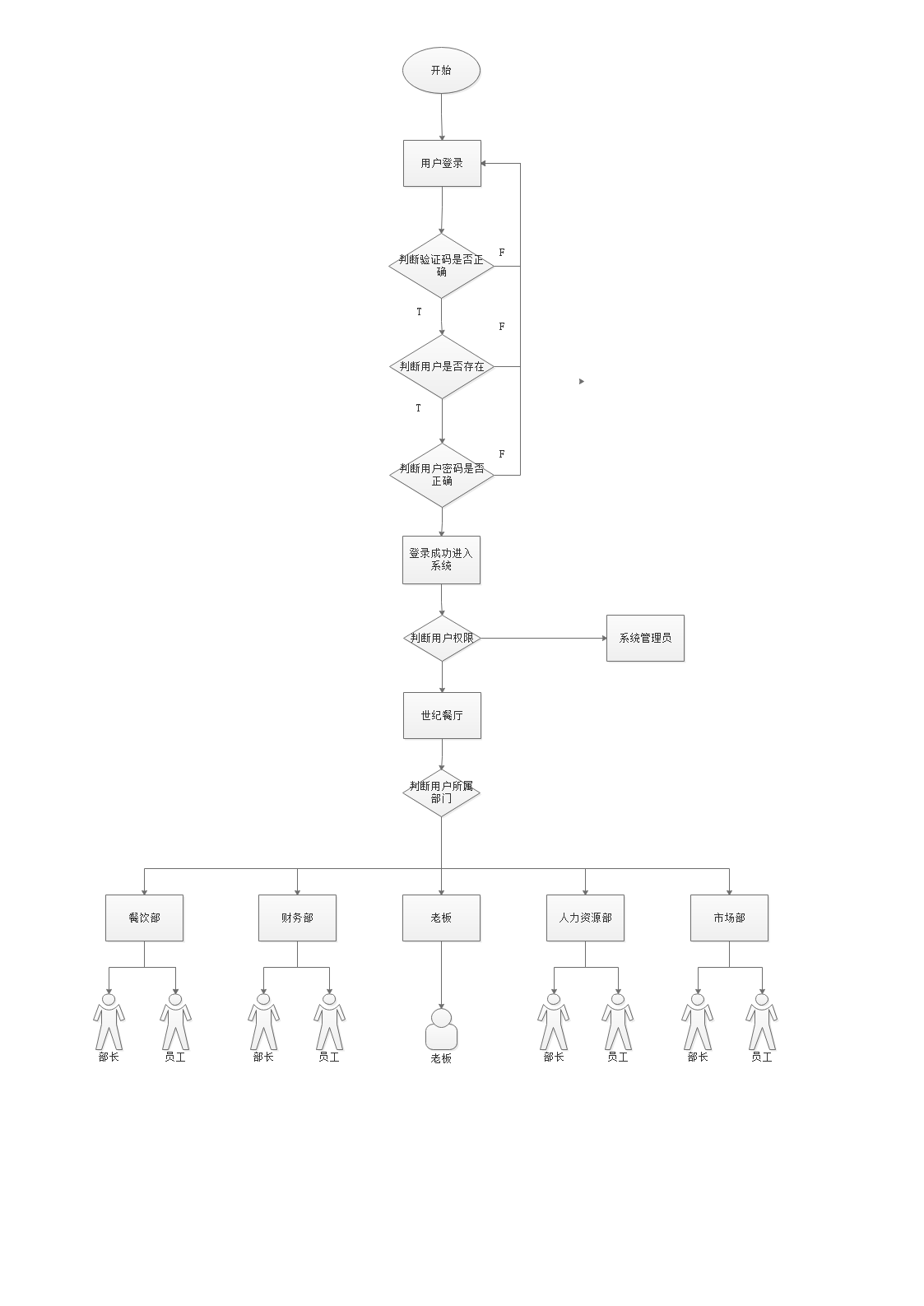


图 2 餐厅系统流程图

用户首先进入的就是登陆界面，要求必须填写用户名和密码才能登陆，并且还要填写验证码，先检验验证码是否填写正确，之后再根据用户填写的用户名和密码与数据库的数据进行比对，一致后还要根据用户的权限跳转到相应的页面，然后才能进行业务操作。

# 4 系统设计

## 4.1功能模块设计

根据需求分析，本系统设计了如下几个功能模块：

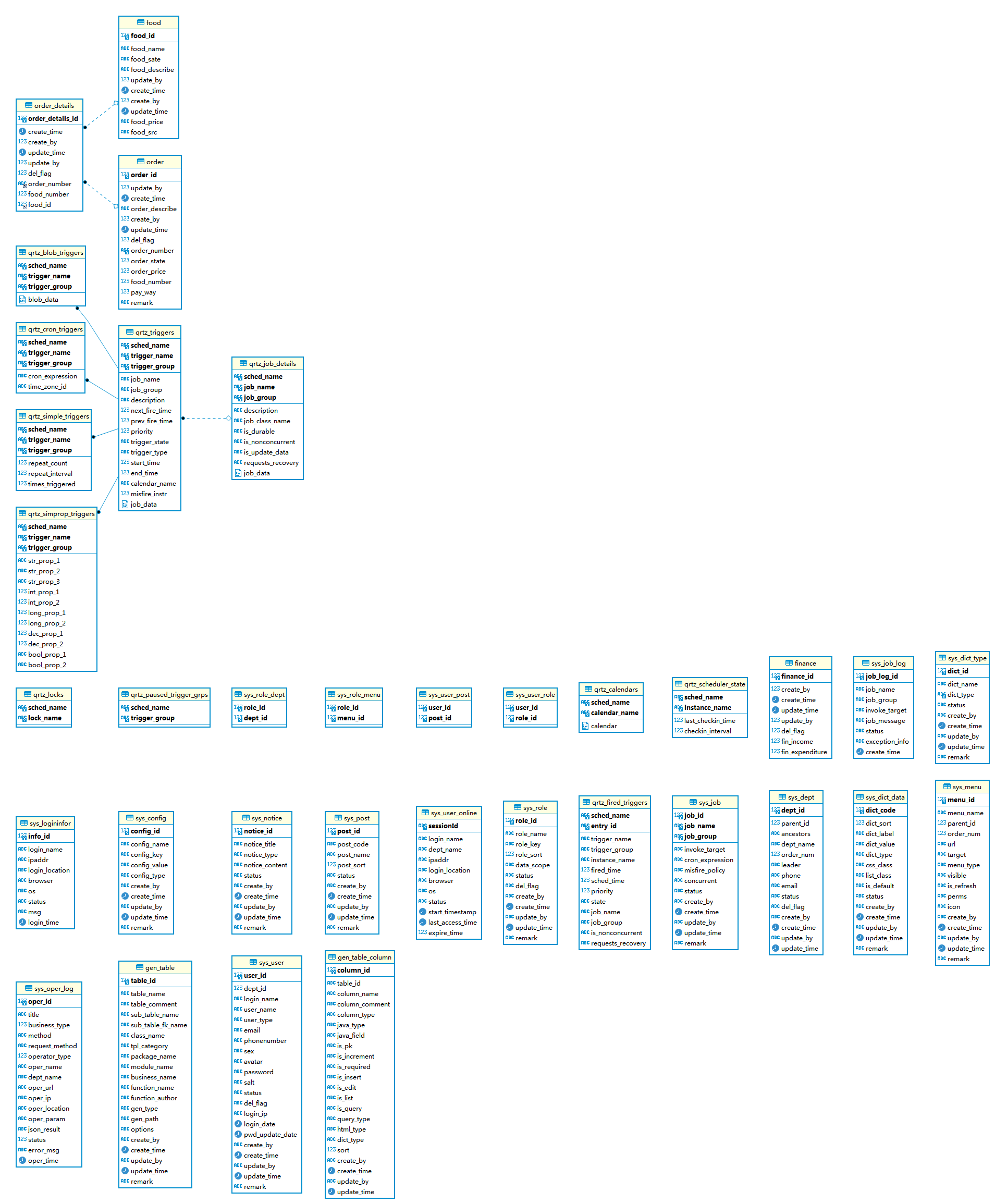
登录功能模块，能让客户登录系统界面，除了输入账号密码以外还要求客户输入验证码做人工判断，并且记录用户的登陆时间，长时间不进行操作会提示客户重新登录，登录成功后还要根据用户的权限跳转到对应的页面。

查询管理模块，用户登录后可以查询与自身权限对应的信息，并对其进行增加、更改、删除等操作，并且支持多选功能。

其他功能模块，为方便以后添加功能而存在的模块，目前包含对年月日流水的记录、记录在线人数等。目前实现的功能有：

1. 上传头像功能：用户可以本地上传图片当做头像，在个人信息界面进行此操作。
2. 网页换肤功能：能更改主界面的颜色主题。
3. 网页全屏、锁屏功能：能让主界面全屏显示，并带有锁屏功能，这样离开办公桌也能避免信息泄露。
4. Tab页的操作功能模块：能一键关闭其他Tab页、关闭此Tab页左边（右边）所有Tab页、全部关闭。
5. 流水记录功能模块：统计餐厅的年月日流水。
6. 登录日志：记录登录人的ip，浏览器类型，操作系统。
7. 在线人数统计：实时统计登录在线人数。

## 4.2数据库设计



4.2-1数据库er图

图4.2-1为本系统所用的数据库er图

以下为本系统所使用的主要表的字段设计

表4.2-1各表格公用字段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据类型 | 是否为空 | 默认值 |  |
| Create\_time | datetime | false |  | 创建时间 |
| Create\_by | Interger | false |  | 创建者 |
| update\_time | datetime |  |  | 更新时间 |
| update\_by | Interger |  |  | 更新者 |
| Del\_flag | Interger | false | 0 | 删除标识 |

表4.2-2订单表（order）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据类型 | 是否为空 | 默认值 |  |
| Order\_id | Interger | false |  | 订单id |
| order\_describe | String |  |  | 订单描述 |
| Order\_number | String | false |  | 订单编号 |
| Order\_price | double |  |  | 订单总价 |
| Order\_state | Interger | False | 0 | 订单状态 |
| Pay\_way | Interger |  |  | 支付方式 |

表4.2-3订单菜品详细表（order\_details）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据类型 | 是否为空 | 默认值 |  |
| Order\_details\_id | Interger | False |  | 主键 |
| Order\_number | String | False |  | 订单编号 |
| Food\_id | Interger | False |  | 菜品id |
| Food\_number | Interger |  |  | 菜品数量 |

表4.2-4菜单表（food）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据类型 | 是否为空 | 默认值 |  |
| food\_id | Interger | false |  | Id |
| food\_name | String |  |  | 菜名 |
| food\_state | Interger |  | 0 | 状态，是否售罄 |
| food\_describe | text | False |  | 菜品描述 |
| Food\_price | double |  |  | 价格 |

表 4.2-5用户信息表（sys-user）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据类型 | 是否为空 | 默认值 |  |
| user\_id | Interger | false |  | Id |
| Dept\_id | Interger | false |  | 部门id |
| user\_name | String |  |  | 用户名 |
| Login\_name | String |  | 0 | 登录账号 |
| email | String | False |  | 用户邮箱 |
| Phone\_number | String | false |  | 用户手机号 |
| Sex | Char |  |  | 用户性别 |
| Avatar | String |  |  | 头像路径 |
| Password | String |  |  | 密码 |
| Salt | String |  |  | 盐密码 |
| Status | Char |  |  | 账号状态 |
| Login\_ip | String |  |  | 最后登录ip |
| Login\_date | Datetime |  |  | 最后登录时间 |
| Pwd\_update | Datetime |  |  | 密码最后更新 |

表 4.2.6 用户在线记录表（sys\_user\_online

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据类型 | 是否为空 | 默认值 |  |
| session\_id | Interger | false |  | Id |
| Login\_name | String |  | 0 | 登录账号 |
| Dept\_name | String |  |  | 部门名称 |
| ipaddr | String | false |  | Ip地址 |
| Ip\_location | String |  |  | Ip登录地点 |
| browser | String |  |  | 浏览器类型 |
| os | String |  |  | 操作系统 |
| Status | String |  |  | 在线状态 |
| Start\_timestamp | Datetime |  |  | Session创建时间 |
| Last\_access\_time | Datetime |  |  | Session最后访问时间 |
| Expire\_time | Datetime |  |  | 超时时间 |

表 4.2.7 部门表（sys\_dept）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 数据类型 | 是否为空 | 默认值 |  |
| dept\_id | Interger | false |  | Id |
| parent\_id | Interger | false |  | 父部门id |
| ancestors | String |  |  | 祖级列表 |
| Dept\_name | String |  |  | 部门名称 |
| Order\_num | Integer |  |  | 显示顺序 |
| Leader | String |  |  | 负责人 |
| leader\_phone | String |  |  | 负责人电话 |
| Leader\_email | String |  |  | 负责人邮箱 |
| Status | Integer |  |  | 部门状态 |

## 4.3用户界面设计

### 4.3.1 登录界面

主打简约风格，图片素材取自网络

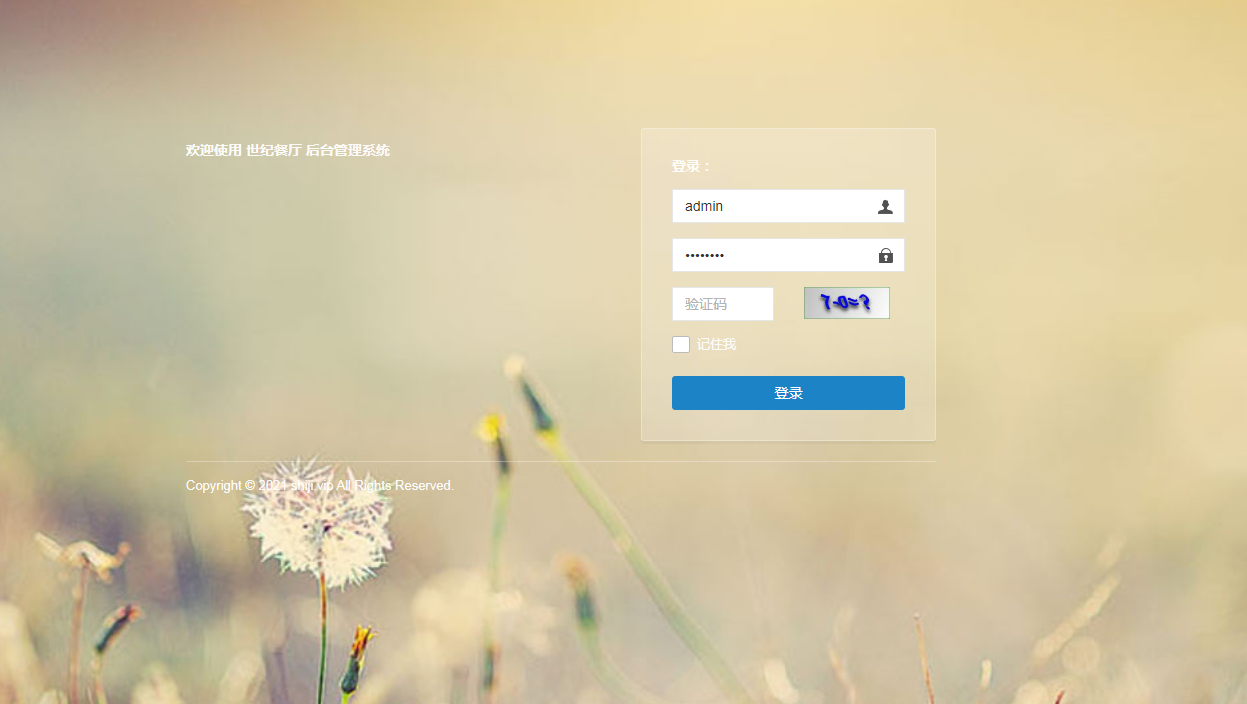


图 4.3.1-1

### 4.3.2 系统主界面

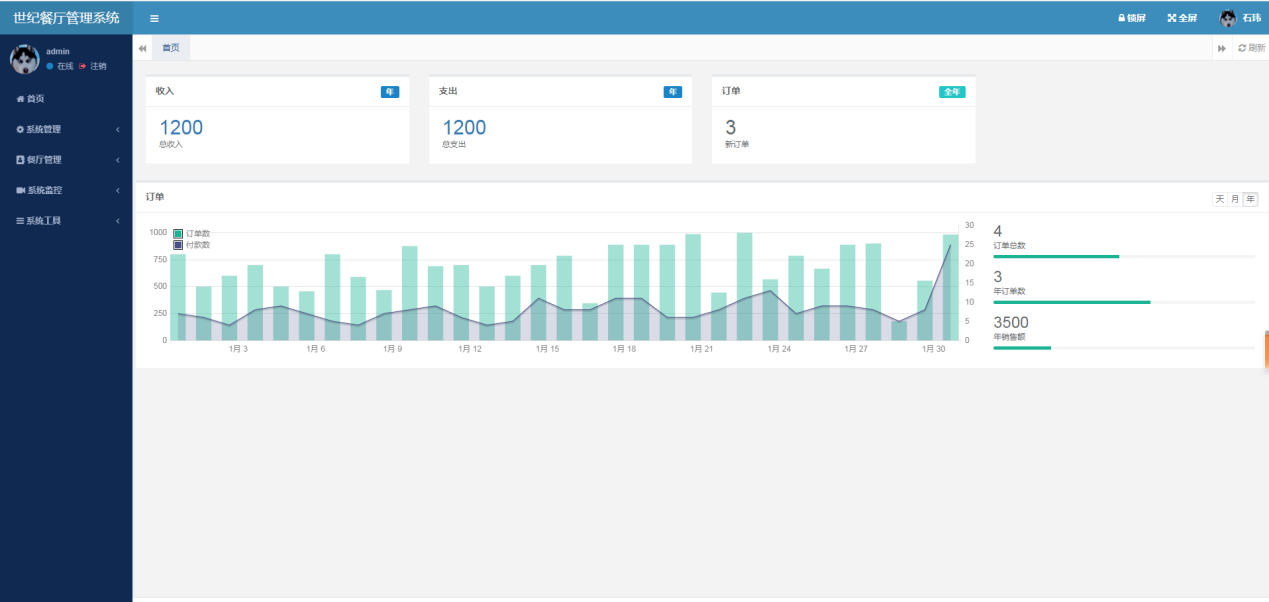


图 4.3.2-1

默认以蓝白为配色，提供有换肤功能，可供客户自己选择

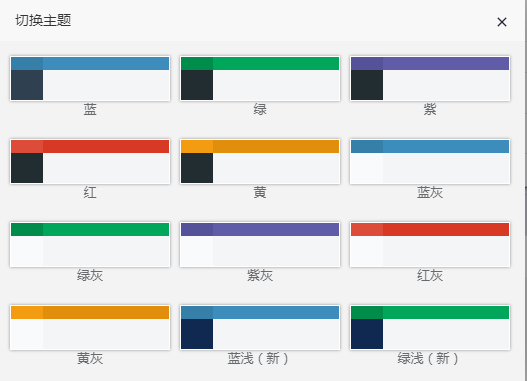


图 4.3.2-2

将主要的功能菜单放到左侧导航栏，右边区域则用来做数据显示，方便客户操作

### 4.3.3 员工管理界面

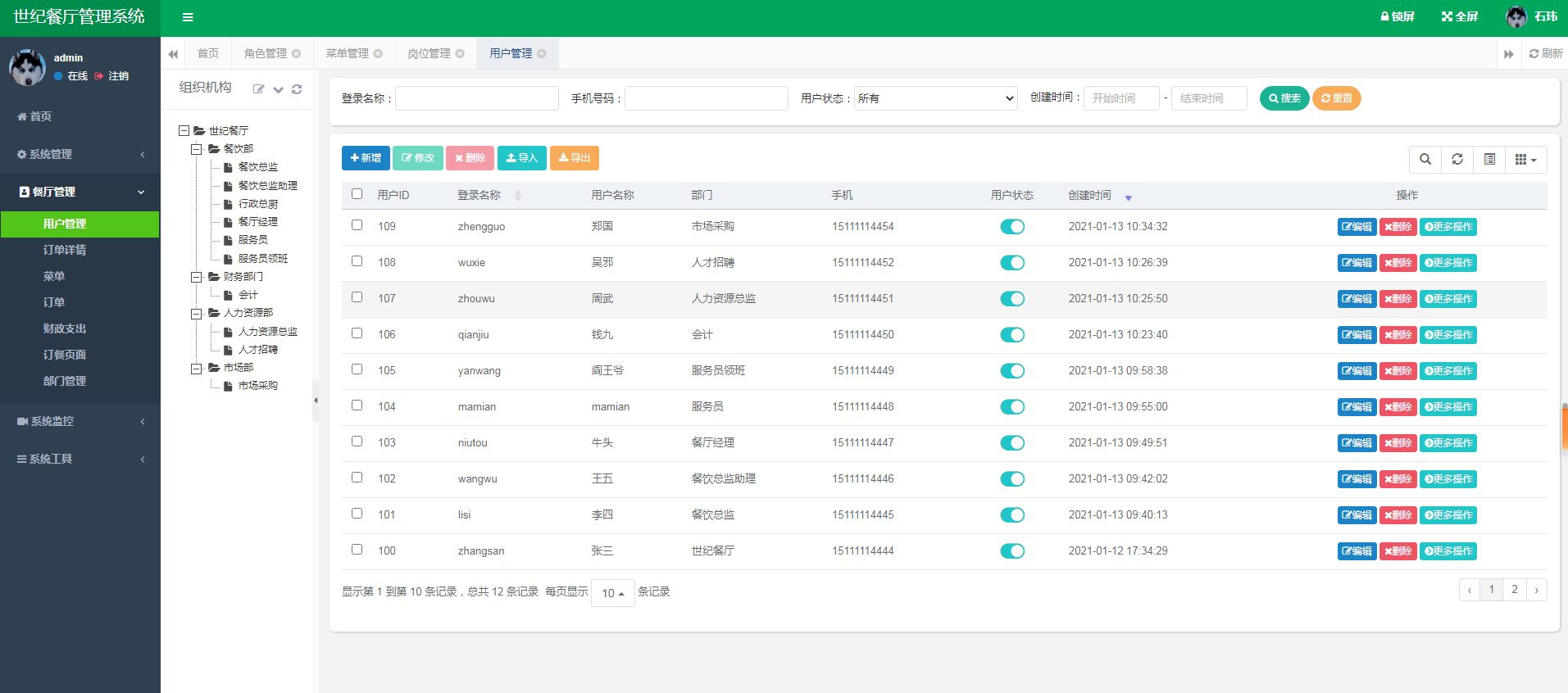


图 4.3.3-1

该界面为老板所能看到的界面，列举了全部门的员工信息，上方工具栏能人工搜索指定员工的个人信息，下方的工具按钮能进行员工信息的增删改查，而右侧的工具按钮则是能对对应的员工信息进行操作。

### 4.3.4 部门管理界面

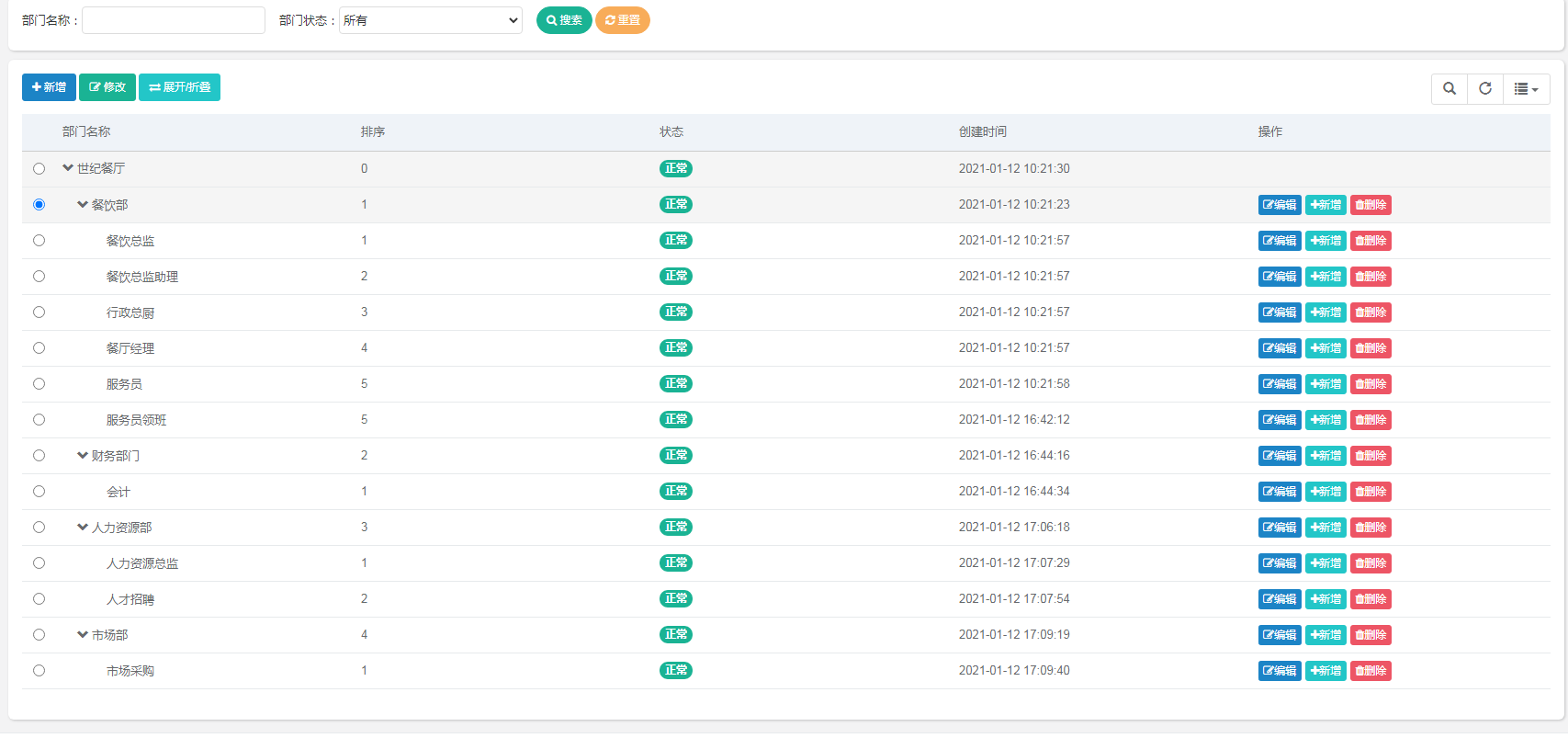


图4.3.4-1

该界面同样为老板所能看到的界面，列举了所有的部门，上方工具栏能人工搜索指定部门，下方的工具按钮能进行对部门的增删改查，而右侧的工具按钮则是能对对应的部门进行操作，同时能将部门下的子部门进行折叠，浏览起来更加简洁，不会因为部门过多而让人应接不暇。

### 4.3.5 商品管理界面



表4.3.5-1

该界面同样为所有员工都能看到的界面，列举了所有的菜品，上方工具栏能人工搜索指定菜品，下方的工具按钮能进行对菜品的增删改查，而右侧的工具按钮则是能对对应的菜品进行操作。

### 4.3.6客户个人界面



表4.3.6-1

客户能查看或修改自己的个人信息，还能上传头像

## 4.4 本章小结

本系统主要着重设计了对人员管理方面的功能，也注重页面的美观每个页面的样式都力求简洁漂亮，同时也让各功能之间的切换比较流畅，视图采用标签页的形式展现，可以一键关闭或者关闭部分标签页，而且标签之间的切换并不会刷新整个页面，对各个浏览器也进行了兼容，并且能全屏显示

# 5系统实现

## 5.1 登录验证功能模块

当客户在网页上输入账号、密码、验证码后，前端通过from表单提交将数据传输给后端，后端通过springboot框架将自动识别前端传来的Json数据中的账号、密码与验证码，首先先检验验证码是否正确,使用shiro实现验证码的生成与检验，错误则直接返回错误信息，若正确，则将账号、密码与数据库的员工表进行比对，如果账号密码有不存在、账号或密码错误、账号已被删除、则返回提示信息给前端，前端接收到后展示给客户,密码错误5次锁定账号10分钟，若账号与密码正确则根据用户的权限等级跳转到对应的页面

页面跳转方面使用的则是thymeleaf框架，通过Springboot解析前端传来的url，得到页面的名称，thymeleaf将自动拼接为xx.html，并在系统静态资源目录下找到该页面展示出来

实现该功能所用到的配置文件与代码：

SysLoginController.java:

*/\*\*  
 \* 登录验证  
 \*   
 \** ***@author*** *ruoyi  
 \*/*@Controller  
**public class** SysLoginController **extends** BaseController  
{  
 @GetMapping(**"/login"**)  
 **public** String login(HttpServletRequest request, HttpServletResponse response)  
 {  
 *// 如果是Ajax请求，就返回json格式的字符串。* **if** (ServletUtils.*isAjaxRequest*(request))  
 {  
 **return** ServletUtils.*renderString*(response, **"{\"code\":\"1\",\"msg\":\"没有登录或登录超时。则提示客户请重新登录\"}"**);  
 }  
  
 **return "login"**;  
 }  
  
 @PostMapping(**"/login"**)  
 @ResponseBody  
 **public** AjaxResult ajaxLogin(String username, String password, Boolean rememberMe)  
 {  
 UsernamePasswordToken token = **new** UsernamePasswordToken(username, password, rememberMe);  
 Subject subject = SecurityUtils.*getSubject*();  
 **try** {  
 subject.login(token);  
 **return** success();  
 }  
 **catch** (AuthenticationException e)  
 {  
 String msg = **"用户名或密码错误"**;  
 **if** (StringUtils.*isNotEmpty*(e.getMessage()))  
 {  
 msg = e.getMessage();  
 }  
 **return** error(msg);  
 }  
 }  
  
 @GetMapping(**"/unauth"**)  
 **public** String unauth()  
 {  
 **return "error/unauth"**;  
 }  
}

## 5.2 查询管理模块

该模块功能主要为客户对数据库的操作，后台连结数据库方面采用JDBC驱动，搭配Mybaits的逆向工程插件使用能自动根据数据的表格生存对应的实体类、mapper接口和mapper.xml文件。前端发送请求后，后台的controller层接收请求，根据请求的类型调用service层的方法，service处理好请求数据后，调用对应的mapper接口，通过mybatis辅助进对数据的增删改查。

实现该功能所需要的部分代码：

SysUserController.java

@Log(title = **"用户管理"**, businessType = BusinessType.***EXPORT***)  
@RequiresPermissions(**"system:user:export"**)  
@PostMapping(**"/export"**)  
@ResponseBody  
**public** AjaxResult export(SysUser user)  
{  
 List<SysUser> list = **userService**.selectUserList(user);  
 ExcelUtil<SysUser> util = **new** ExcelUtil<SysUser>(SysUser.**class**);  
 **return** util.exportExcel(list, **"用户数据"**);  
}  
  
@Log(title = **"用户管理"**, businessType = BusinessType.***IMPORT***)  
@RequiresPermissions(**"system:user:import"**)  
@PostMapping(**"/importData"**)  
@ResponseBody  
**public** AjaxResult importData(MultipartFile file, **boolean** updateSupport) **throws** Exception  
{  
 ExcelUtil<SysUser> util = **new** ExcelUtil<SysUser>(SysUser.**class**);  
 List<SysUser> userList = util.importExcel(file.getInputStream());  
 String operName = ShiroUtils.*getSysUser*().getLoginName();  
 String message = **userService**.importUser(userList, updateSupport, operName);  
 **return** AjaxResult.*success*(message);  
}  
  
@RequiresPermissions(**"system:user:view"**)  
@GetMapping(**"/importTemplate"**)  
@ResponseBody  
**public** AjaxResult importTemplate()  
{  
 ExcelUtil<SysUser> util = **new** ExcelUtil<SysUser>(SysUser.**class**);  
 **return** util.importTemplateExcel(**"用户数据"**);  
}  
  
*/\*\*  
 \* 新增用户  
 \*/*@GetMapping(**"/add"**)  
**public** String add(ModelMap mmap)  
{  
 mmap.put(**"roles"**, **roleService**.selectRoleAll().stream().filter(r -> !r.isAdmin()).collect(Collectors.*toList*()));  
 mmap.put(**"posts"**, **postService**.selectPostAll());  
 **return prefix** + **"/add"**;  
}  
  
*/\*\*  
 \* 新增保存用户  
 \*/*@RequiresPermissions(**"system:user:add"**)  
@Log(title = **"用户管理"**, businessType = BusinessType.***INSERT***)  
@PostMapping(**"/add"**)  
@ResponseBody  
**public** AjaxResult addSave(@Validated SysUser user)  
{  
 **if** (UserConstants.***USER\_NAME\_NOT\_UNIQUE***.equals(**userService**.checkLoginNameUnique(user.getLoginName())))  
 {  
 **return** error(**"新增用户'"** + user.getLoginName() + **"'失败，登录账号已存在"**);  
 }  
 **else if** (StringUtils.*isNotEmpty*(user.getPhonenumber())  
 && UserConstants.***USER\_PHONE\_NOT\_UNIQUE***.equals(**userService**.checkPhoneUnique(user)))  
 {  
 **return** error(**"新增用户'"** + user.getLoginName() + **"'失败，手机号码已存在"**);  
 }  
 **else if** (StringUtils.*isNotEmpty*(user.getEmail())  
 && UserConstants.***USER\_EMAIL\_NOT\_UNIQUE***.equals(**userService**.checkEmailUnique(user)))  
 {  
 **return** error(**"新增用户'"** + user.getLoginName() + **"'失败，邮箱账号已存在"**);  
 }  
 user.setSalt(ShiroUtils.*randomSalt*());  
 user.setPassword(**passwordService**.encryptPassword(user.getLoginName(), user.getPassword(), user.getSalt()));  
 user.setCreateBy(ShiroUtils.*getLoginName*());  
 **return** toAjax(**userService**.insertUser(user));  
}  
  
*/\*\*  
 \* 修改用户  
 \*/*@GetMapping(**"/edit/{userId}"**)  
**public** String edit(@PathVariable(**"userId"**) Long userId, ModelMap mmap)  
{  
 List<SysRole> roles = **roleService**.selectRolesByUserId(userId);  
 mmap.put(**"user"**, **userService**.selectUserById(userId));  
 mmap.put(**"roles"**, SysUser.*isAdmin*(userId) ? roles : roles.stream().filter(r -> !r.isAdmin()).collect(Collectors.*toList*()));  
 mmap.put(**"posts"**, **postService**.selectPostsByUserId(userId));  
 **return prefix** + **"/edit"**;  
}  
  
*/\*\*  
 \* 修改保存用户  
 \*/*@RequiresPermissions(**"system:user:edit"**)  
@Log(title = **"用户管理"**, businessType = BusinessType.***UPDATE***)  
@PostMapping(**"/edit"**)  
@ResponseBody  
**public** AjaxResult editSave(@Validated SysUser user)  
{  
 **userService**.checkUserAllowed(user);  
 **if** (StringUtils.*isNotEmpty*(user.getPhonenumber())  
 && UserConstants.***USER\_PHONE\_NOT\_UNIQUE***.equals(**userService**.checkPhoneUnique(user)))  
 {  
 **return** error(**"修改用户'"** + user.getLoginName() + **"'失败，手机号码已存在"**);  
 }  
 **else if** (StringUtils.*isNotEmpty*(user.getEmail())  
 && UserConstants.***USER\_EMAIL\_NOT\_UNIQUE***.equals(**userService**.checkEmailUnique(user)))  
 {  
 **return** error(**"修改用户'"** + user.getLoginName() + **"'失败，邮箱账号已存在"**);  
 }  
 user.setUpdateBy(ShiroUtils.*getLoginName*());  
 **return** toAjax(**userService**.updateUser(user));  
}

Mybaits-config.xml

*<?***xml version="1.0" encoding="UTF-8"** *?>***<!DOCTYPE configuration  
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-config.dtd"*>***<**configuration**>  
   
 <**settings**>  
 <**setting name="cacheEnabled" value="true"** /> <**setting name="useGeneratedKeys" value="true"** /> <**setting name="defaultExecutorType" value="REUSE"** /> <**setting name="logImpl" value="SLF4J"** /> </**settings**>  
   
</**configuration**>

ISysUserService,java

**public** List<SysUser> selectUserList(SysUser user);  
  
**public** List<SysUser> selectAllocatedList(SysUser user);  
  
**public** List<SysUser> selectUnallocatedList(SysUser user);  
  
**public** SysUser selectUserByLoginName(String userName);  
  
**public** SysUser selectUserByPhoneNumber(String phoneNumber);  
  
**public** SysUser selectUserByEmail(String email);  
  
**public** SysUser selectUserById(Long userId);  
  
**public** List<SysUserRole> selectUserRoleByUserId(Long userId);  
  
**public int** deleteUserById(Long userId);  
  
**public int** deleteUserByIds(String ids);  
**public int** insertUser(SysUser user);  
  
**public boolean** registerUser(SysUser user);  
  
**public int** updateUser(SysUser user);  
  
**public int** updateUserInfo(SysUser user);

## 5.3 其他功能模块

针对用户个人界面本系统实现了用户自传头像的功能，代码如下：

*/\*\*  
 \* 通用文件上传请求  
 \*/*@PostMapping(**"/common/upload"**)  
@ResponseBody  
**public** AjaxResult uploadFile(MultipartFile file) **throws** Exception  
{  
 **try** {  
 String filePath = RuoYiConfig.*getUploadPath*();  
 String fileName = FileUploadUtils.*upload*(filePath, file);  
 String url = **serverConfig**.getUrl() + fileName;  
 AjaxResult ajax = AjaxResult.*success*();  
 ajax.put(**"fileName"**, fileName);  
 ajax.put(**"url"**, url);  
 **return** ajax;  
 }  
 **catch** (Exception e)  
 {  
 **return** AjaxResult.*error*(e.getMessage());  
 }  
}

查看流水方面最主要的是对前端页面的操作，根据点击年月日来显示对应的流水，也进行了简单的操作显示，代码如下：

**$**(**"#btnDay"**).click(**function** () {  
 **$**(**"#btnMonth"**).removeClass();  
 **$**(**"#btnMonth"**).addClass(**"btn btn-xs btn-white"**);  
 **$**(**"#btnYear"**).removeClass();  
 **$**(**"#btnYear"**).addClass(**"btn btn-xs btn-white"**);  
  
 **$**(**"#btnDay"**).removeClass();  
 **$**(**"#btnDay"**).addClass(**"btn btn-xs btn-white active"**);  
  
 **$**.ajax({  
 **url**: **"order/OrderStatistics?flag=0"**,  
 **type**: **"post"**,  
 success: **function** (result) {  
 **$**(**"#orders"**).text(**"日订单数"**);  
 **$**(**"#sale"**).text(**"日销售额"**);  
 **$**(**"#orderStatistics"**).text(result.orderPrices == **null** ? 0 : result.**orderPrices** )  
 **$**(**"#ordersOfDate"**).text(result.orders == **""** ? 0 : result.orders )  
  
 }  
  
 }  
 )  
  
})  
  
**$**(**"#btnMonth"**).click(**function** () {  
 **$**(**"#btnDay"**).removeClass();  
 **$**(**"#btnDay"**).addClass(**"btn btn-xs btn-white"**);  
 **$**(**"#btnYear"**).removeClass();  
 **$**(**"#btnYear"**).addClass(**"btn btn-xs btn-white"**);  
  
 **$**(**"#btnMonth"**).removeClass();  
 **$**(**"#btnMonth"**).addClass(**"btn btn-xs btn-white active"**);  
  
 **$**.ajax({  
 **url**: **"order/OrderStatistics?flag=1"**,  
 **type**: **"post"**,  
 success: **function** (result) {  
 **$**(**"#orders"**).text(**"月订单数"**);  
 **$**(**"#sale"**).text(**"月销售额"**);  
 **$**(**"#orderStatistics"**).text(result.orderPrices == **null** ? 0 : result.**orderPrices** )  
 **$**(**"#ordersOfDate"**).text(result.orders == **""** ? 0 : result.orders )  
  
 }  
  
 }  
 )  
  
})  
  
**$**(**"#btnYear"**).click(**function** () {  
 **$**(**"#btnMonth"**).removeClass();  
 **$**(**"#btnMonth"**).addClass(**"btn btn-xs btn-white"**);  
 **$**(**"#btnDay"**).removeClass();  
 **$**(**"#btnDay"**).addClass(**"btn btn-xs btn-white"**);  
  
 **$**(**"#btnYear"**).removeClass();  
 **$**(**"#btnYear"**).addClass(**"btn btn-xs btn-white active"**);  
  
 **$**.ajax({  
 **url**: **"order/OrderStatistics?flag=2"**,  
 **type**: **"post"**,  
 success: **function** (result) {  
 **$**(**"#orders"**).text(**"年订单数"**);  
 **$**(**"#sale"**).text(**"年销售额"**);  
 **$**(**"#orderStatistics"**).text(result.orderPrices == **null** ? 0 : result.**orderPrices** )  
 **$**(**"#ordersOfDate"**).text(result.orders == **""** ? 0 : result.orders )  
  
 }  
  
 }  
 )  
  
})

## 5.4本章小结

登录功能模块实现了用户的登录与权限验证，查询管理模块实现了对数据库的增删改查操作，其他功能模块实现的有头像上传，流水记录，

# 6 系统测试

## 6.1测试目的

为了检测系统是否存在某些漏洞，导致应用程序无法正常运行，需要开发者对各功能模块做一个简单测试，保证项目上线后用户的良好体验，避免造成不必要的财产损失

## 6.2测试方法

使用了swagger工具对各功能模块的控制层接口做了简单测试，通过填入测试用例的数据确保每个接口都能返回正确的message，响应值为200，测试完成之后在登录系统测试各功能之间是否有相互冲突的情况。

## 6.3测试用例

表6.3-1 测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用户 | 用户身份 | 是否被删除 | 测试情景 | 测试情况 |
| admin | 超级管理员 | false | 登录系统管理员界面 | 成功登录 |
| shiwei | 店长 | false | 登录客户界面 | 页面跳转失败（已解决） |
| zhangsan | 员工 | false | 登录员工界面 | 成功登录 |
| zhengguo | 员工 | true | 测试被删除员工能否登录 | 登陆失败 |
| Admin | 超级管理员 | False | 测试员工的增删改查 | 操作成功 |
| Zhangsan | 员工 | False | 测试头像上传功能 | 操作成功 |
| Ruoyi | 员工 | False | 测试员工的权限，能否修改权限以外的员工信息 | 操作失败 |

# 7 总结

本系统是针对现在社会中小型餐饮行业落后人工管理现象设计的，该系统主要完成了对商品信息，员工信息，会员信息还有财务信息的数字化，让管理者更加得心应手的管理自己的企业。

这是本人第一次完成度比较高的项目，期间遇到许多困难，通过查资料，问前辈都一一解决，这些经历让我意识到经验不足是多么致命的缺点，经验老道的解决问题不拖泥带水，而我自己解决的问题往往会留下许多后遗症。

在分析设计过程中，主要是在分析系统的功能设计上耗费了大量的时间，对很多组织结构和业务流程分析得不够彻底，也忽略了很多细节性问题。

同时,我也在生活中很关注这种软件的使用情况,它的发展可以更加人性化，以目前的科技，完全可以省去在前台点菜的步骤，客人进入餐厅，完全可以通过平板电脑进行在线点菜，厨房也可以在第一时间为客人准备，这样的处理方式现在很多大型的餐饮企业已经实现，但是我个人认为可以推广到中小型企业，一旦这个得到实现，加上这样的一个系统，完全可以在餐饮行业的竞争中取得很大的优势。

# 参考文献

[1]张志清.国内餐饮电子管理系统现状浅析［EB/OL］

<https://wenku.baidu.com/view/66e03156f311f18583d049649b6648d7c1c708eb.html>

[2]石洮焕.餐饮管理系统文献综述［EB/OL］

<https://wenku.baidu.com/view/85018b340875f46527d3240c844769eae109a33c.html>

[3] 周志明.深入理解java虚拟机[M].北京:机械工业出版社,2011

[4]BoCong-Deng.详细Spring Boot教程之入门［EB/OL］

<https://blog.csdn.net/DBC_121/article/details/104383089>

[5]中国产业信息网.[2016-2022年中国餐饮行业市场专项调研及投资前景评估报告](https://www.chyxx.com/research/201508/334970.html)[R],2016

[6]Craig Walls. Spring in Action[M].Greenwich:Manning Publications,2018

[7] 郝佳.Spring源码深度解析（第2版）[M].北京:人民邮报出版社,2020

[8] 黄文毅.Spring MVC+MyBatis快速开发与项目实战[M].北京:清华大学出版社,2019

[9]陈光剑.Spring Boot 开发实战[M].北京:机械工业出版社,2018

[10] 杨恩雄.Spring Boot 2+Thymeleaf企业应用实战[M].北京:电子工业出版社,2018

[11] 程韶建.深入实践SpringBoot[M].北京:机械工业出版社,2018

1. Joshua Bloch.Effective Java: Programming Language Guide (Java Series)[M]；Massachusetts:Addison-Wesley,2001

[13] Bruce Eckel.Thinking in Java[M].New York:Pearson,2006

[14] Marth Deinum.Spring Boot 2 Recipes[M].State of California:Apress,2019

[15] Craig Walls.Spring Boot in Action[M],Greenwich:Manning Publications,2016

[16]杨开振.深入浅出MyBatis技术原理与实战[M].北京:电子工业出版社,2016

1. \_北冥有鱼\_.学会BootStrap这一篇文章就够了［EB/OL］

https://blog.csdn.net/yuhuiling/article/details/77444665

[18] 费希利(著),冯宇晖，贾文峰(译).SQL基础教程[M].北京:人民邮电出版社,2009

[19] Baron(著),王小东(译).高性能MySQL[M].北京:电子工业出版社,2010

[20] 泰德维尔(著),蒋芳(译).界面设计模式[M].北京:电子工业出版社,2013