



云南财经大学信息学院

网上鲜花销售设计报告

学 期：_____ 2023-2024 学年第二学期 _____

课程名称：_____ JavaEE 应用开发 _____

班 级：_____ 智科 22-1 _____

学 号：_____ 202205002326、202205002344 _____

_____ 202205002324、202205002346 _____

姓 名：_____ 杨瑞霖、杨俊松、王明玥、舒雨扬 _____

2024 年 6 月 25 日

目录 Content

第 1 章 绪论	1
1.1 研究背景	1
1.2 项目简介	1
1.3 项目实例	3
第 2 章 相关技术	4
2.1 B/S 结构概述	4
2.2 前端主要技术	5
2.2.1 HTML5、CSS3、JavaScript 概述	5
2.2.2 Bootstrap 框架技术	6
2.2.3 Vue 单页面应用	6
2.3 后端主要技术	8
2.3.1 SpringMVC	8
2.3.2 Spring	9
2.3.3 MyBatis	9
2.4 其他技术	9
2.4.1 高德地图 api 接口	9
2.4.2 MD5 校验工具	10
2.4.3 MyBatis-Plus	10
2.4.4 Token 身份验证和权限管理技术	11
第 3 章 系统设计	12
3.1 系统概要	12
3.2 系统功能结构设计	13
3.2.1 系统功能总述	13
3.2.2 系统设计图	14
3.2.3 系统文件结构	15
3.3 后端设计	19
3.3.1 项目 Until 层	19
3.3.2 项目 Entity 层	21
3.3.3 项目 dao 层	22
3.3.4 项目 Service 层	24
3.3.5 项目 Controller 层	27
3.3.6 Interceptor（拦截器）	33
3.3.7 总结	34
3.4 前端设计	34
3.4.1 Layui	36
3.4.2 Bootstrap	37
3.4.3 Vue.js 2	39
3.5 数据库设计	46
3.5.1 需求分析	46
3.5.2 数据需求	47

3.5.3 事物需求	50
3.6 MySQL 设计	51
3.4.1 MySQL 环境	51
3.4.2 数据库表的设计	52
第 4 章 系统实现	54
4.1 前台界面实现	54
4.1.1 前台首页	54
4.1.2 前台商家	55
4.1.3 前台鲜花游览	55
4.1.4 前台系统公告	55
4.1.5 前台鲜花介绍	56
4.1.6 前台的登录和注册	56
4.2 后台界面实现	57
4.2.1 用户后台	57
4.2.2 用户个人中心及用户联系客服界面	57
4.2.3 商家后台	58
4.2.4 管理员后台	58
第 5 章 系统测试	60
5.1 系统功能测试	60
5.1.1 用户登录界面测试	60
5.1.2 后台登陆界面测试	61
5.1.3 用户个人中心上传图片测试	61
5.2 系统测试结论	62
参考文献	63
致谢	64

第 1 章 绪论

1.1 研究背景

花之美，花之雅，老少皆喜。春城昆明，花开遍地，乃花之都，远近闻名。随着昆明城内外人民对鲜花的需求日益上升，具有新样式，服务性能更强的鲜花销售平台将越来越受欢迎，它在鲜花销售的市场上发挥的作用会越来越明显。

当今使用鲜花销售平台的消费用户和商家，他们的需求相对以往，有所不同。作为消费用户，他们更渴望一个个人信息操作更为方便且详细，信息获取更为精细的平台；作为商家，他们急需一个能够快速更新旗下店面和鲜花信息的平台，避免所在旗下相关信息的滞后性，影响产品销量。而无论是消费用户还是商家，都希望销售平台的界面简洁美观，该有的信息一目了然，同时，也希望自己相关的信息的安全性得到保障。对此，本组所开发的鲜花销售平台的界面，首页便提供了花类和商家的信息，让消费用户快速定位所需信息；上方菜单栏额外提供不同的管理渠道，利于用户、商家以及管理员对自己信息的修改；而该平台的管理员，具有最高权限，可以对用户和商家的信息进行不定时检查与更新，保证了信息的安全性。

1.2 项目简介

本组所开发的平台除了基础的使用者登录、注册、信息浏览和个人信息修改等基本功能外，最大的亮点在前台浏览的功能之外，设置了后台管理功能，以权限不同来确保平台运行的稳定性。此外，该平台还具有在线客服的功能。

基础功能：

（1）用户登录与注册功能：该功能用于区别该用户是否在平台上注册过账号，若该用户还未注册，则点击“注册”，填写相关信息以进行注册。该平台的后台管理除了用户的登录与注册功能，还有商家与管理人员的登录与注册功能，便于二者进入后台管理进行相应操作。

（2）页面浏览功能：首页在简洁美观的前提下，直接把鲜花种类与商家展现出来，方便浏览；而页面上会有菜单栏，在菜单栏里可以进行更有针对性地信息浏览，也可以访问“个人中心”或者“后台管理”。

（3）个人中心：作为消费用户，可以在网页进行个人信息的修改与完善。不仅如此，“个人中心”里还有“我的地址”、“我的订单”等，可以查看自己最近购买鲜花的情况等信息。

特色功能：

（1）后台管理：用户、商家与管理人员都可以访问该页面，不同的是，商家只有访问“后台管理”才能添加“店铺”与“鲜花信息”，而一个商家若想入驻平台，需要向管理人员申请，自身是无法在“后台管理”内进行商家的添加操作。作为管理人员，“后台管理”便是他们用来修改、更新用户与商家信息的地方。管理人员是具有最高权限的后台管理者，用户和商家的信息都能进行操作，而一个商家的入驻也得经过管理人员的审核，避免非法商家的入驻。管理平台让管理人员更方便地对该平台进行维护。可以说，商家与管理人员，属于后

台管理部分，前台主要访客—用户，他们所看到的商品相关数据，由后台上传与修改。“后台管理”使得前台访客与后台人员的工作得以区分。

（2）在线客服：该网页用于管理人员对于用户和商家所提出的问题进行答疑。该网页能够使得在线的管理人员都能同时看到来自不同用户和商家最近提出的问题，管理人员可以选择某一提问者进行解答。“在线客服”使得该平台能够及时收到使用者对平台功能的使用的反馈，以便更好地在后期进行更新和维护。同时，解答提问者的问题，也能够让使用者感觉平台服务的贴心之处，利于口碑的打造。

1.3 项目实例

本项目已经过开发环境的严密调试后部署到生产环境。

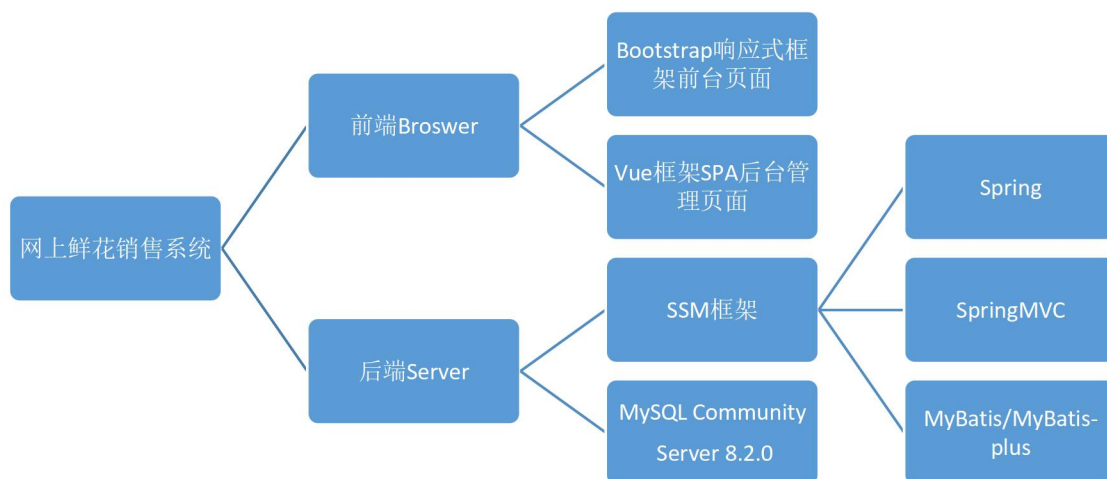
现发布鲜花销售系统项目 Release 1.0 版的链接

Release 1.0 版: <http://flower.rlsblog.cn> 环境: 腾讯云服务器、Linux 系统、宝塔面板、Tomcat9.0、MySQL5.7、Nginx1.21.0、PHP7.4、JDK1.8。

开发环境: Windows11 系统、Tomcat9.0.80、MySQL8.2.0、JDK19.0.2、IntelliJ IDEA 2022.3.2、maven3.9.6。

第 2 章 相关技术

网上鲜花销售平台采用了经典的前后端分离式的 Java web 项目结构，整体代码逻辑遵循 B/S 结构。其中，项目的前端主要采用了 Bootstrap 响应式框架设计的前台响应式布局和 Vue 框架设计的后台管理系统单页面应用技术。



2.1 B/S 结构概述

在 Java Web 项目中，B/S 结构通常由三个主要部分组成：浏览器（Browser）、Web 容器/服务器（Web Container/Server）和数据库服务器（Database Server）。

浏览器：用户通过浏览器访问 Web 应用程序，如 Google Chrome、Mozilla Firefox、Microsoft Edge 等。

Web 服务器：负责处理客户端请求，并返回响应。在 Java Web 项目中，常用的 Web 服务器有 Apache Tomcat、Jetty 等。Web 服务器

上部署了 **Java Web** 应用程序，负责处理客户端请求，并调用后端服务。

数据库服务器：用于存储和检索数据。在 **Java Web** 项目中，常用的数据库服务器有 **MySQL**、**Oracle** 等。数据库服务器通过 **JDBC** (**Java Database Connectivity**) 与 **Java Web** 应用程序进行通信，以实现数据的存储和检索。

这种结构中，用户通过浏览器访问 **Web** 应用程序，发送请求到 **Web** 服务器；**Web** 服务器接收到请求后，与数据库服务器进行交互，获取或更新数据；最后，**Web** 服务器将结果返回给浏览器，用户可以看到相应的页面。这种结构使得 **Java Web** 应用程序可以在任何支持 **Web** 浏览器的设备上访问，并且易于维护和扩展。

2.2 前端主要技术

2.2.1 HTML5、CSS3、JavaScript 概述

HTML5、**CSS3** 和 **JavaScript** 是构建现代 **Web** 应用程序的基石。**HTML5** 提供了结构和语义化的页面标记，**CSS3** 负责页面的样式和布局，而 **JavaScript** 则为页面添加交互性和动态效果。

HTML5 的基础知识包括了解基本的标签，以及语义化标签，以提高页面的结构和可读性。

CSS3 的基础知识包括掌握各种选择器的用法，如元素选择器、类选择器、ID 选择器等，并了解盒模型的概念和属性，以实现页面的样式和布局。**CSS3** 具有过渡和动画特性，如 **transition** 和 **animation** 属性，以实现页面元素的平滑过渡和动态效果。响应式设计也是 **CSS3**

的一个重要知识点，通过使用媒体查询和弹性布局等技术，可以适应不同设备的屏幕大小和分辨率。

JavaScript 的基础知识包括学习变量和数据类型，如字符串、数字、数组、对象等，以及基本的运算符和表达式。条件语句和循环是实现根据条件执行不同的代码块和重复执行某段代码的功能的关键。使用 **JavaScript** 处理页面上的事件，如点击事件、鼠标移动事件等，以实现交互性的网页效果。

2.2.2 Bootstrap 框架技术

项目前端的前台用户展示界面采用了 **Bootstrap** 响应式框架的设计，具备多端自适应能力，可以较好的展示页面内容。

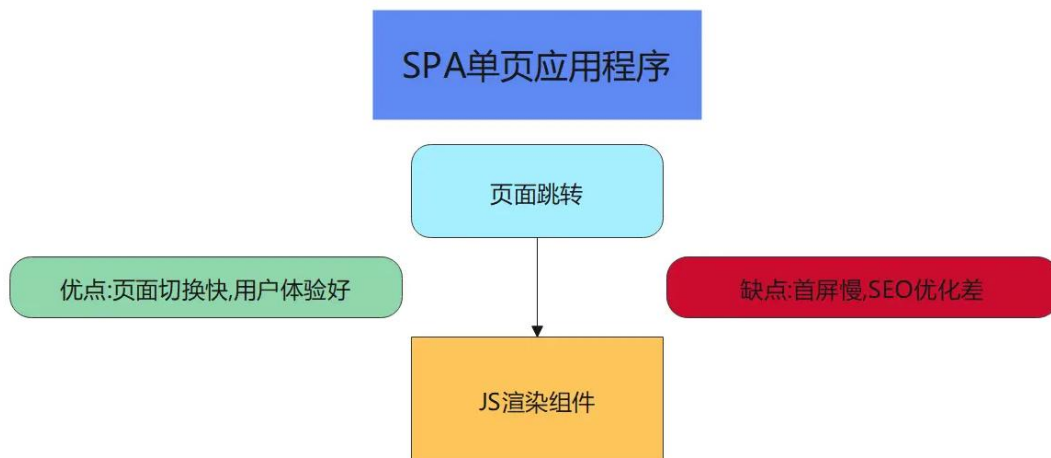
Bootstrap 是一个开源的前端开发框架，广泛应用于各种网站和 Web 应用程序的开发。它基于 HTML、CSS 和 JavaScript 构建，提供了丰富的组件和样式，使得开发者可以快速构建出美观、响应式的界面。

Bootstrap 的核心特点是其响应式设计，能够自适应不同的设备和屏幕尺寸，提供一致的用户体验。它还包含许多预设的样式和组件，例如导航栏、模态框、下拉菜单等，使得开发者可以快速实现各种设计和功能需求。此外，**Bootstrap** 还具有良好的可定制性和可扩展性，可以根据项目需求进行自定义和扩展。

2.2.3 Vue 单页面应用

项目前端的后台管理系统使用 **Vue.js** 渐进式 JavaScript 框架的 SPA（单页面应用）设计。

通过 **Node.js** 的包管理器（**npm**）创建 **Vue** 的 SPA 项目。

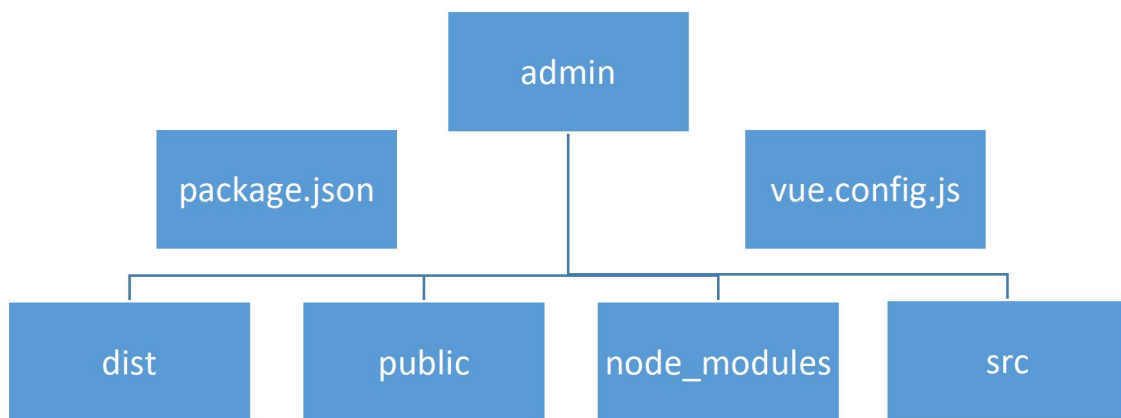


Vue.js 是一个用于构建用户界面的渐进式 JavaScript 框架，它提供了一种简单、灵活的方式来创建单页面应用（SPA）。SPA 是一种 Web 应用程序，只有一个主页面，并通过客户端路由来管理用户界面。

Vue.js 通过组件化的方式构建应用，使得开发人员可以轻松地创建可重用的组件，并利用 Vue 的响应式系统来管理数据和状态。Vue Router 是 Vue.js 的官方路由管理器，它与 Vue.js 深度集成，提供了强大的路由功能，使得开发人员可以轻松地实现页面导航和数据管理。

Vue.js 还支持与第三方库和框架的集成，如 Axios 用于数据请求和处理、ECharts 用于数据可视化等。通过这些技术的结合使用，开发人员可以快速构建出功能强大、性能卓越的单页面应用。

Vue 项目结构如下：



鲜花销售系统后台管理系统的部分主要文件架构

2.3 后端主要技术

鲜花销售系统后端使用 SSM 整合框架开发，使用 MVC 模型中的 Controller、Ajax 等与前端进行数据传输。

SSM 框架由 Spring 框架、SpringMVC 框架、MyBatis 框架组成。通过 SSM 框架的集成，可以快速构建出稳定、高效和可扩展的 Web 应用。Spring 提供了一整套的解决方案，包括事务管理、安全性、数据校验等；SpringMVC 使得 Web 开发更加简单高效；而 MyBatis 则提供了强大的数据库操作支持。

2.3.1 SpringMVC

SpringMVC 是一个基于 Java 的 Web 框架，用于构建单页面应用。它采用了 MVC 设计模式，将应用逻辑、数据和用户界面分离，使得应用更加易于维护和扩展。SpringMVC 简化了 Web 开发过程，通过注解驱动的请求处理、组件扫描和自动装配等技术，提高了开发效率。

SSM 框架整合中，SpringMVC 负责管理表现层和控制器。SpringMVC 是 Spring 框架的一个子模块。

2.3.2 Spring

Spring 是一个开源的 Java 平台，提供了一种全面的编程和配置模型，用于构建企业级应用。它是一个轻量级的容器，负责管理应用的对象、配置和生命周期。Spring 的核心是控制反转（IoC）和面向切面（AOP）编程，通过这些机制，开发者可以更轻松地组织和管理代码。

SSM 框架整合中，Spring 负责对象的创建与事务管理，Spring 可以管理持久层的 Mapper 对象和业务层的 Service 对象。即原本由 MyBatis 负责管理并实例化的 SqlSessionFactory 改由 Spring 负责配置和实例化。MyBatis 不再配置和维护数据库连接信息，数据库连接池由 Spring 负责配置与管理，数据库连接对象由 Spring 负责创建并管理。MyBatis 的 Mapper 代理也由 Spring 负责创建。

2.3.3 MyBatis

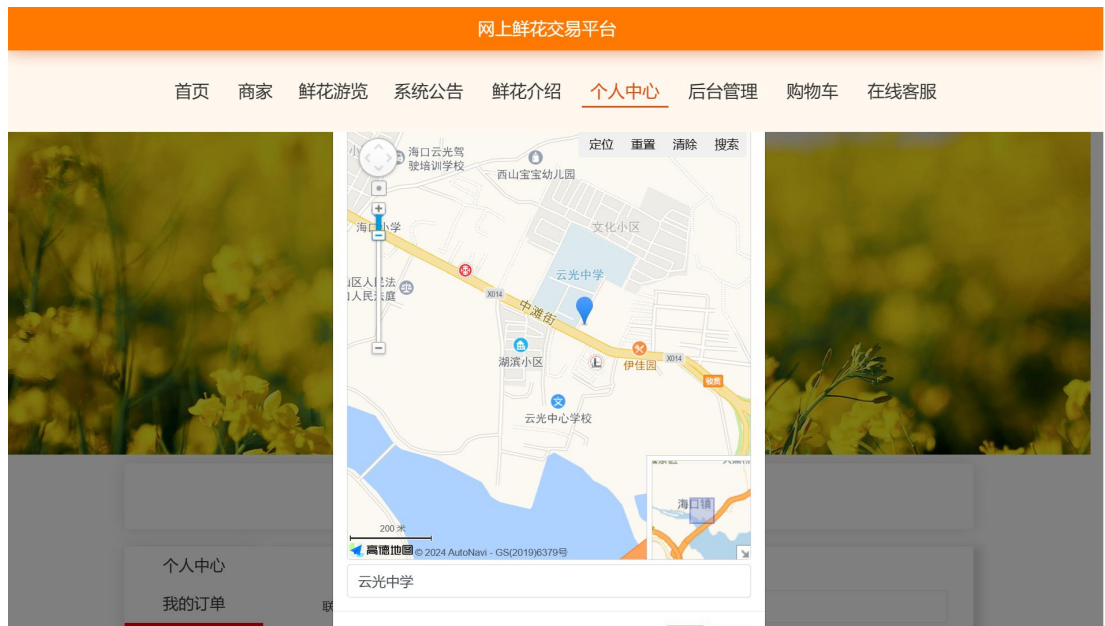
MyBatis 是一个持久层框架，用于将 Java 对象与数据库中的记录进行映射。它提供了 SQL 语句的映射功能，使得开发者可以直接编写 SQL 语句，而不需要手动编写 JDBC 代码。MyBatis 支持自定义查询、存储过程和高级映射，能够满足各种复杂的数据库操作需求。

SSM 框架整合中 MyBatis 负责与数据库的交互。

2.4 其他技术

2.4.1 高德地图 api 接口

用户的收货地址可通过在地图上选择定位，也可搜索位置。如下图所示：



2.4.2 MD5 校验工具

MD5 在项目中主要用于数据完整性和安全性。

数据完整性验证：MD5 是一种单向散列函数，它将任意长度的数据映射为固定长度的散列值。项目选择 MD5 对敏感数据进行加密，比如用户密码。

防止数据被恶意篡改：由于 MD5 是单向的，所以不能直接从散列值恢复出原始数据。这就意味着，如果有人试图修改数据，那么 MD5 值也会改变。

2.4.3 MyBatis-Plus

MyBatis-plus 是 MyBatis 的增强工具，在 MyBatis 的基础上只做增强不做改变，是为了简化开发和提高效率而生。

它通过提供丰富的功能和便捷的配置，使得开发者能够快速构建高效、稳定的应用程序。

MyBatis-Plus 的主要特点包括：支持自动代码生成，减少手写代码量；支持主键生成策略配置，提高数据插入效率；提供分页插件，

方便进行数据分页查询；支持全局通用注解配置，简化开发流程。

2.4.4 Token 身份验证和权限管理技术

鲜花销售系统的双端登录身份验证采用了 Token 令牌机制，而后台管理系统的角色权限则使用了 Token 的权限管理。

Token 验证技术是一种安全机制，用于验证用户身份并保护数据的完整性。它通过在客户端和服务端之间建立安全的令牌，确保只有授权用户才能访问特定的资源。令牌可以包含用户的身份信息和访问权限，并且只能由服务器进行验证。这种验证方式可以有效防止非法访问和数据篡改，提高应用程序的安全性。

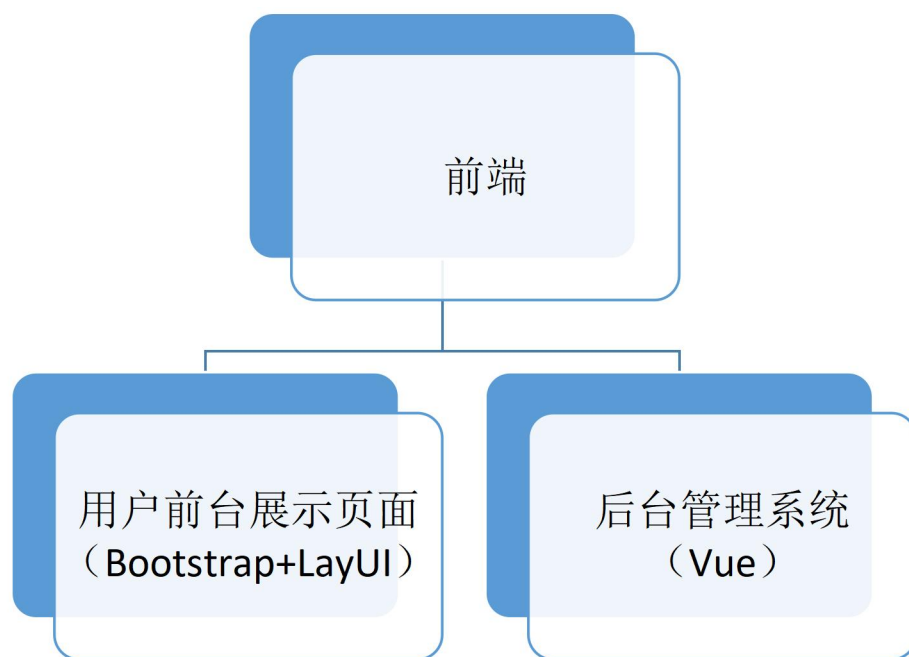
服务器将用户的 Token 相关信息存储于 MySQL 数据库中。

第3章 系统设计

3.1 系统概要

网上鲜花销售平台在系统设计上采用了前后端分离设计，确保了前后端之间的独立开发和高效协作。整体架构遵循 B/S 结构，同时，使用 Maven 构建项目并管理依赖为系统的可扩展性和可维护性提供了坚实的基础。

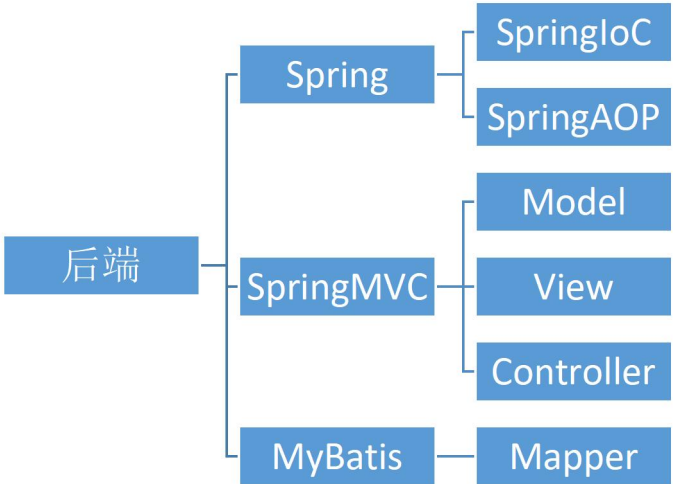
系统的前端部分，平台选用了 Bootstrap 响应式框架+LayUI 组件，实现了自适应的、响应式的网页布局，无论在 PC 端、平板还是手机上，都能提供出色的用户体验。同时，结合 Vue 框架，构建了高效、稳定的后台管理系统单页面应用。Vue 的组件化开发模式，大大提高了开发效率和代码可维护性。



系统前端设计

系统的后端主要部分使用了 SSM 框架来进行设计并进行集成开发。

通过 SSM 框架的集成，本系统具备快速构建稳定、高效和可扩展特性。**Spring** 提供了一套全面的解决方案，包括事务管理、安全性、数据校验等；**SpringMVC** 使得 Web 开发更加简单高效；**MyBatis** 则提供了强大的数据库操作支持。



系统后端设计

3.2 系统功能结构设计

3.2.1 系统功能总述

网上鲜花销售系统旨在为用户和商家提供一个安全、快捷、方便的线上交易平台。根据普通用户、平台商家、网站管理员的不同角色需求对系统的功能进行设计。

游客：在未注册登录时可以浏览系统的前台展示页面，包括首页、商家、鲜花游览、系统公告、鲜花介绍页面。该设计使得游客具有一部分的浏览权限，当游客进行购买、评论、咨询在线客服等操作时将跳转到系统前台的登录页面。未注册的用户可以进行注册后登录以获取完整的阅览权限。

用户：注册后的普通用户具有前台浏览与后台个人管理的权限。

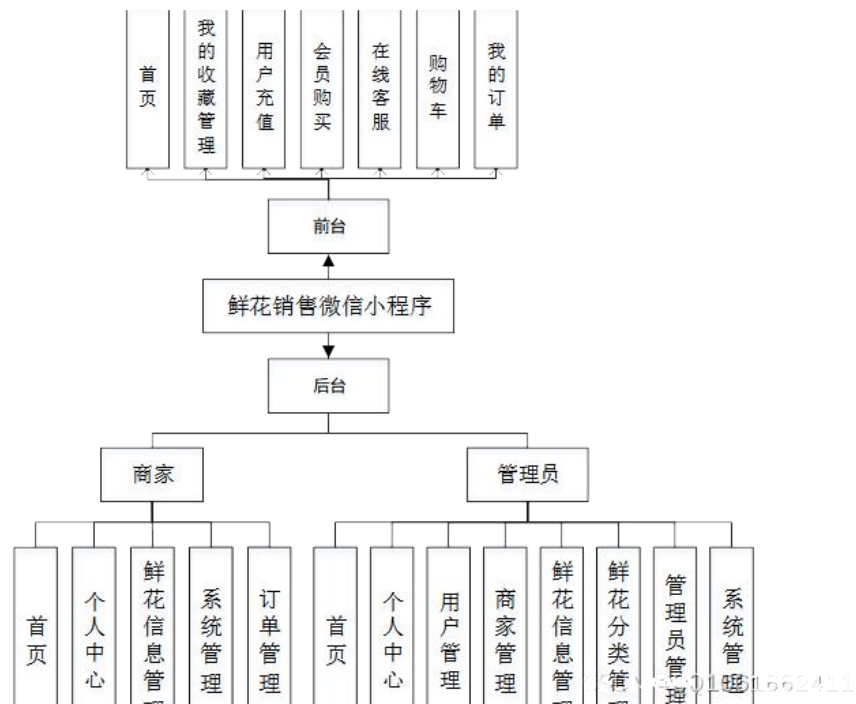
普通用户可以在系统前台的个人中心页面进行用户账号、密码、头像、姓名、性别等个人信息的变更、账户余额充值（当前未接入支付平台的 api），管理订单、收货地址、商品收藏等操作；普通用户可以将自己心仪的商品收藏或加入购物车，可在购物车页面查看；普通用户可以对商品、商家进行评论或评价，也可以咨询线上客服；注册后的用户也可以登入后台管理页面进行如上操作，不同的是，后台管理页面的设计更加简洁、方便。（用户测试账号：test@customer 密码：test）

商家：商家如想要入驻平台需要提前联系网站的管理员进行商家账号的注册。商家成功注册账号后可登入网上鲜花交易平台的后台管理系统完善商铺的信息、上架或下架商品、查看或回复商品、店铺的评论等操作。（商家测试账号：test@store 密码：test）

管理员：网站管理员具有网站的最高权限，可以对用户、商家账户进行任意操作，具有管理全部订单及其状态的权限（如已取消、发货等）。同时，管理员拥有系统管理的权限，可以更改前台展示页面的轮播图、发布系统公告，作为在线客服回复信息等。（管理员测试账号：test@admin 密码：test）

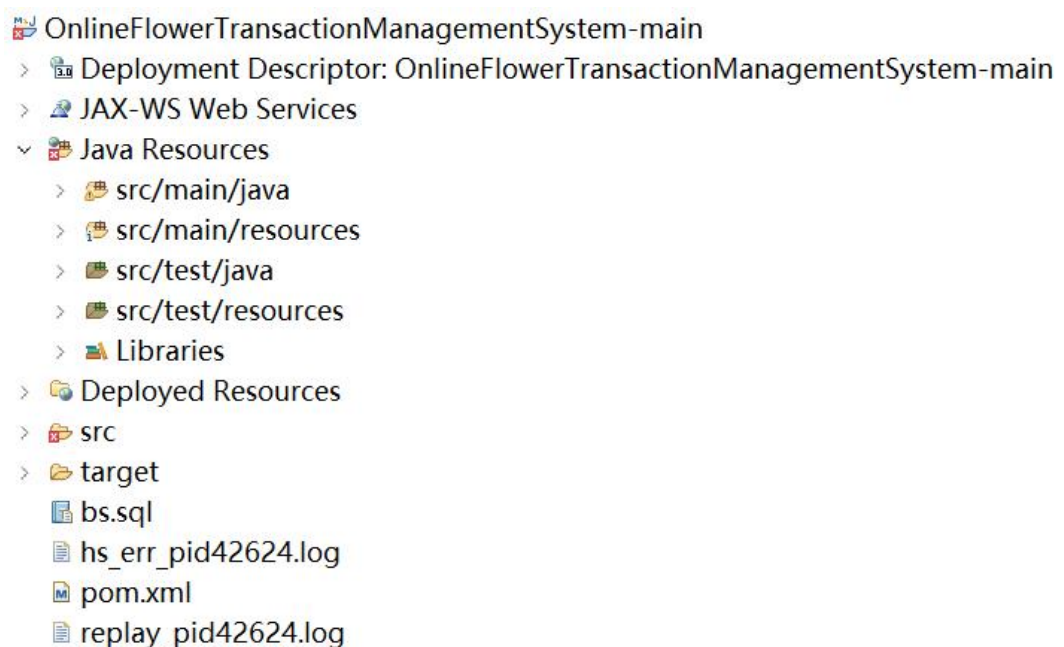
3.2.2 系统设计图

鲜花销售系统结构图如下：



3.2.3 系统文件结构

项目文件总览（使用 Maven 构建的 Java Web 项目）



其中各目录（或配置文件）的所存放的内容如下：

src/main/java	项目的 Java 源代码存放的目录
---------------	-------------------

src/main/resources	项目的资源文件存放的目录
src/main/webapp	项目的前端 web 应用根目录
src/test/java	Java 测试代码存放的目录
src/test/resources	测试相关的资源文件存放的目录
target	存放 Java 代码编译后的字节码文件
pom.xml	Maven 项目的核心配置文件

bs.sql 为 MySQL 数据库的建表文件，*.log 为日志文件。

src/main/java 下的各个包（package）的作用：

com.annotation	自定义注解@APPLoginUser、@IgnoreAuth、@LoginUser 等接口的 Java 源文件
com.config	MyBatis-Plus 的属性自动填充配置接口的实现类 MyMetaObjectHandler.java 文件
com.controller	用于存放 AddressController、CartController 等 SpringMVC 的控制器（Controller）类
com.dao	用于存放数据链路层的接口代码
com.dao.spring	用于存放 Spring 的部分 xml 配置文件
com.entity	存放数据库通用操作实体类的 Java 源代码文件如 AddressEntity.java、CartEntity.java 等。
com.entity.model	用于存放接收传参的实体类的 Java 源代码
com.entity.view	后端返回视图实体辅助类的 Java 源代码
com.entity.vo	手机端接口放回实体辅助类的 Java 源代码
com.interceptor	存放实现 HandlerInterceptor 接口的 AuthorizationInterceptor 类，用于 Token 验证
com.model.enums	MyBatis-Plus 的通用枚举类型接口 IEnum 的实现类 TypeEnum.java 文件
com.service	存放业务逻辑层 Service 的各种接口
com.service.impl	存放各个 Service 接口的实现类
com.util	用于存放各种工具类

src/main/resources 文件夹结构如图所示：

src/main/resources	doc 文件夹下存有
> doc	
> favicon_logosc	MyBatis-Plus 数据库的
> mapper	sys_user 建表的 SQL 文件。
> mybatis	
> spring	favicon_logosc 文件夹用于存
config.properties	网站的各种 favicon 图标。
log4j.properties	

mapper 文件夹下有 MyBatis 框架的映射配置文件。mybatis 文件夹下有 MyBatis 框架的 mybatis-config.xml 主要配置文件。spring 文件夹下存储 Spring 框架的 spring.xml 配置文件、SpringMVC 的 spring-mvc.xml 配置文件以及 Spring 整合 MyBatis 框架的 spring-mybatis.xml 配置文件。config.properties 是数据源的配置文件，有数据库的连接字信息。

log4j.properties 是 log4j 日志框架的配置文件。

以下是前端 webapp 的文件结构：

webapp	admin 文件夹为后台管理系统
> admin	
> front	根文件夹。front 为前台展示页面的
> resources	文件夹。resources 用于存放 Bootstrap
> upload	框架和 jQuery 框架的相关文件。upload
> WEB-INF	文件夹用于存放用户上传的头像，商
favicon.ico	
tiaozhuan.html	

家上传的图片等。favicon.ico 为前端使用的书签图标；tiaozhuan.html 将自动跳转到 front/index.html。

其中 admin 文件夹为 Vue.js 框架的 SPA（Single Page Application）单页面应用的根文件夹，是标准的 npm 构建的 Vue 项目结构。front

文件夹下有 Element UI 框架、LayUI 框架的组件文件 夹。

3.3 后端设计

3.3.1 项目 Util 层

util 层就是存放工具类的地方，对于一些独立性很高的小功能，或重复性很高的代码片段，可以提取出来放到 Util 层中。

1.BaiduUtil

作用：getCityByLonLat 方法通过使用 HttpClientUtils.doGet 方法获取 URL 用于根据经纬度获取省市信息，getAuth 方法用于获取 API 访问 token

2.CommonUtil

作用：用于生成随机字符串

3.FileUtil

作用：将文件数据转为字节数组形式，用于文件上传。

4.HttpClientUtils

作用：发起 HTTP Get 请求方法，请求 url 地址

5.JQPageInfo

作用：作为获取页面信息工具类，其中使用 Get，Set 方法获取了 Page（当前页码），Limit（每页显示的数据条数），Sidx（排序字段），Order（排序方式），Offset（查询偏移量）

6.Query

作用：

a).使用 GetPage，GetLimit 方法判断分页参数

b).使用 SQLFilter 工具类防止 SQL 注入

c).进行分页和排序

7.PageUtils

作用：为已经存在的 List 数据进行分页处理

```
public PageUtils(List<?> list, int totalCount, int pageSize, int  
currPage)
```

它负责实际的分页逻辑，首先，计算总记录数 **total**，然后根据当前页和每页大小确定分页的起始和结束索引。如果索引越界，则进行调整。接着，创建一个 **Page** 对象，设置其当前页码、每页大小，并将原始列表添加到 **Page** 对象中。然后，计算总页数（向上取整），并设置到 **Page** 对象的 **pages** 字段。最后，创建一个 **PageInfo** 对象，将 **Page** 对象传递给它，并返回 **PageInfo** 对象。**PageInfo** 包含了分页后的列表、总记录数、总页数等信息。以此来进行分页操作。

1.8SQLFilter

作用：作为 SQL 注入过滤工具类。防止 web 应用程序在接收相关数据参数时未做好过滤，将其直接带入到数据库中查询，导致攻击者可以拼接执行构造的 SQL 语句。

1.9MD5Util

作用：对字符串加密

1.10MPUtil

作用：表达式转化和字符串转换

1.11R

作用：R 工具类在 Java 开发中的作用是封装 HTTP 请求的响应结果，有统一格式、处理异常、封装数据和错误处理，提供了便捷的方法和标准的响应格式，方便开发者进行数据处理和错误处理。

1.12springContextUtil

作用：用来获取 Spring 容器管理的 bean。

1.13VaildatorUtils

Vaildator 校验类作用：使用 Validator 来验证 web 前端页面传过来的对象数据属性 是否符合标准要求预期。

3.3.2 项目 Entity 层

Entity:实体层，用于存放我们的实体类，与数据库中的属性值基本保持一致，实现 set 和 get 的方法。

例如 AddressEntity 中将数据库 Adress 表中对应的 id,userid,address,name,phone 这几个字段使用了 get,set 方法，以从数据库中获取相应数据。同理，其他的 Entity 类也是如此。

1.项目 Entity.Model

Model 作为传参实体类同样使用了 set,get 方法，不过 model 是使用 get,set 方法将前端更改的数据进行访问和修改数据库。

例如 AddressModle 中，使用 get,set 方法分别获取和设置 address,name,phone 等字段，进行对数据库的修改。同理，model 包中的其他 model 也是一个作用(分别对应其数据库进行前端返回的进行接收并修改数据库)

2.项目 Entity.view

view 里面存放的是对实体表的映射类(视图类)，是用来给前端用的。

即：有时候我们仅仅需要获得某一个表的几个字段，此时可以用 **view** 存储这几个字段。

例如将 **address** 表中的 **name**、**id** 存入 **address_view** 视图，这样在进行操作时只需调用视图表，就可以完成相对应的操作。

如 **Address View**：它继承 **Address Entity** 实体类，将 **Address Entity** 实体类中访问的数据传到 **View** 中，**View** 层在将 **Address Entity** 后端返回的数据进行封装。**Address View** 接受了 **Address Entity** 访问的数据库 **id**，**name** 等参数进行封装等待 **Controller** 层接收客户端的请求进行调用。同理，其他的 **View**（视图类）也是将其所继承的 **Entity** 传来的参数进行封装。

3.项目 Entity.VO

VO 层的主要作用是用于解耦数据传输和[数据逻辑](#)，使得各层的功能和职责更加清晰和明确。通常情况下，**Controller** 层从前端获取数据后，将其转化为 **VO** 对象，然后传递给 **Service** 层进行业务逻辑处理，最后再将处理结果通过 **VO** 对象返回给前端展示。在此项目中 **VO** 层负责手机端接口的数据库数据进行封装，提供给前端视图查看。

如：**Address VO** 中使用 **get,set** 方法对 **Address** 表中数据进行访问和修改，以等待客户端的动作进行对 **Address** 数据的使用。

3.3.3 项目 dao 层

DAO 层(数据访问层)主要是做数据持久层的工作，用来对应数据

库实体，负责与数据库进行联络的一些任务都封装在此。主要就是封装一些数据库的增删改查操作。

注：

SelectListView:wrapper 参数是一个条件构造器，用于构建查询条件，例如指定查询哪些字段，查询哪些条件等。**selectListView** 方法将这个条件构造器传递给 **baseMapper.selectListView** 方法，该方法会根据条件构造器生成对应的 SQL 语句，并执行查询操作。最后将查询结果转化为视图对象列表并返回。

SelectView:在数据库编程时关系很多表查询数据的情况。这时如果查不出来数据或者查出来的数据不正确（如：有重复数据）解决起来是比较麻烦也花时间。**SelectView** 就解决了这个问题，它可以分析 **Select** 语句，列出各个数据表的数据，并根据数据的关联情况在数据行上做区分

SelectVO:wrapper 构建查询后，调用 **baseMapper** 的 **selectVO** 方法根据条件查询出符合条件的单体实体类对象对应的 **VO**

SelectLiseVO:该方法的作用是根据传入的 **wapper** 对象查询符合条件的实体对象列表，并将其转化为对应的 **VO** 对象列表返回（**VO** 对象是对实体对象的封装，通常包含了实体对象的部分属性和一些相关业务信息）

AddressDao:负责连接数据库 **address** 表中的 **address,id,name** 等数据，其中 **AddressDao** 接口的

selectListVo,selectVo,selectListView,selectView 方法由

AddressServiceImpl 调用。同理 CartDao(连接购物车表), ChatDao (连接 Chat 表), DiscussShangjiaDao(商家评论表), DiscussxianhuayoulanDao(鲜花游览评论表), NewsDao(系统公告表), OrdersDao(订单表), ShangjiaDao(商家表), StoreupDao(收藏表), UserDao(用户表), XianhaufenleiDao(鲜花分类表), XianhuayoulanDao(鲜花游览表), YonghuDao(用户表)中的以上四种方法皆由其对应的 ServiceImpl 调用来进行数据的增删查改工作。

其中 TokenDao 只运用了 SelectListView 方法用于查询用户名和密码。(Token 是在客户端频繁向服务端请求数据, 服务端频繁的去数据库查询用户名和密码并进行对比, 判断用户名和密码正确与否, 并作出相应提示, 同时 Token 是服务端生成的一串字符串, 以作客户端进行请求的一个令牌, 当第一次登录后, 服务器生成一个 Token 便将此 Token 返回给客户端, 以后客户端只需带上这个 Token 前来请求数据即可, 无需再次带上用户名和密码。)

CommonDao:作为通用接口提供一些通用的数据库访问方法, 有 getOption,getFollowByOption,sh,remindCount,selectCalm,selectGroup 和 selectValue 方法, 这些方法都接受 params 的参数, 这个参数又包含一些查询参数, 用于过滤结果。

ConfigDao:作为配置接口

3.3.4 项目 Service 层

由于 Dao 里面存放了对表操作的方法, entity 类中存放了映射表

的 View，VO 类。

Service 的作用就是将 Dao 和 entity 类整合起来，进行再次封装，封装成一个方法，我们调用这个方法，就实现了对某个表的增删改查操作。

Service 是服务层的接口，定义了服务层的方法和业务逻辑，是业务逻辑的抽象层。Service 层的主要作用是对外提供服务，将业务逻辑和数据访问分离，提高代码的可维护性和可扩展性。

以 AdressService 为例：

PageUtils queryPage(Map<String, Object> params);

（根据参数 params 查询并返回分页数据）

这个方法用于分页查询，它接收一个 Map 类型的参数 params，这个参数通常包含查询条件、分页参数等信息。该方法的返回值类型为 pageUtils，pageUtils 是一个自定义的分页工具类，可以用于封装分页查询的结果。该方法的实现通常包括查询总记录数、计算分页参数、执行分页查询等操作。

**PageUtils queryPage(Map<String, Object>
params,Wrapper<ChuxiaodengjiEntity> wrapper);**

（根据参数 params 和条件 wrapper 查询并返回分页数据）

这段代码是一个通用分页查询方法，用于查询符合条件的 Entity 实体对象并返回分页结果。其中，params 参数是一个 Map 类型，用于接收查询条件，wrapper 参数是一个 MyBatis-Plus 的查询条件封装类，用于构建查询条件。运用 PageUtils 工具类封装分页查询结

果，它会根据传入的页码、一页显示的条数等参数自动进行分页，并返回分页结果。该方法的返回值类型为 `PageUtils`，表示查询结果的分页信息。

```
List<AddressView> selectListView(Wrapper<AddressEntity>  
wrapper);
```

（ `wrapper` 查询并返回 `View` 类型的列表。）

这段代码定义了一个方法 `selectListView`，它的返回值是一个 `List<AddressView>`，参数是一个泛型为 `AddressEntity` 的 `wrapper` [对象](#)。

`selectListView` 是用来从数据库中查询符合条件的[地址](#)视图[列表](#)的方法。`wrapper` [对象](#)可以用来设置查询条件。返回的 `List<AddressView>` 则是符合条件的地址视图[列表](#)，可以用来展示在界面上或者进行后续处理。

```
AddressView selectView(@param("ew")(Wrapper<AddressEntity>  
wrapper);
```

（根据条件 `wrapper` 查询并返回一个 `View` 类型的对象）

```
selectListVO(Wrapper<ChongwulingyangEntity> wrapper);
```

（根据条件 `wrapper` 查询并返回 `VO` 类型的列表）

```
selectListVO(Wrapper<ChongwulingyangEntity> wrapper):
```

（根据条件 `wrapper` 查询并返回 `VO` 类型的列表）

VO 与 View 的做的不同工作

View 层查询完数据后，数据库表和表之间的外键约束会非常密

切,这样一来,我们数据库表中存储的数据大多都是其他表的主键 ID,并不会存储具体的信息。

这样一来,我们数据库查询之后得到了一堆其它表的主键 ID,但这并不是我们想要的结果,所以我们需要在业务层再去根据这些主键 ID 去各自对应的数据库表中查询用户想要的信息,然后在返回,查询到的数据我们通常会使用 VO 类来进行接收,查询完毕,我们将 VO 返回给前端展示,这才算是一个业务流程真正完成。

ServiceImpl 是 **Service** 接口的实现类,实现了 **Service** 层的方法和业务逻辑。**ServiceImpl** 层的主要作用是实现 **Service** 层的接口,处理具体的业务逻辑,调用数据访问层的方法,完成数据的增删改查等操作。

以 **AddressServiceImpl** 为例:

其中实现了以上讲述的 **AddressService** 定义的 **selectListView** 等方法和业务逻辑。

同理本项目所写的 **Service** 以及对应的 **ServiceImpl** 皆是同一个运作方式。

3.3.5 项目 Controller 层

Controller 层意义:接收客户端的请求,然后调用 **Service** 层接口控制业务逻辑,获取到数据,传递数据给页面。

Controller 层具体实现以 **AddressController** 为例

1. 获取并返回后端列表

@RequestMapping("/page")注解标记,表示当接收到对应的 URL

请求时会执行该方法。

在方法中：

使用 `@RequestParam` 注解将请求参数映射到 `Map params` 中，同时将请求参数映射到 `AddressEntity address` 中。

通过 `HttpServletRequest request` 获取当前请求的信息。

判断当前用户的角色是否为"管理员"，如果不是管理员，则将 `address` 对象的 `userid` 属性设置为当前用户的 `userId`。

创建一个 `EntityWrapper` 对象 `ew`，用于构建查询条件。

调用 `addressService` 的 `queryPage` 方法，传入参数 `params` 和构建好的查询条件 `ew`，获取分页数据结果 `PageUtils`。

使用 `R.ok().put("data", page)` 返回一个响应结果，其中 `"data"` 对应的值为查询到的分页数据 `page`。

该方法的作用是根据请求参数进行地址信息分页查询，并根据当前用户角色设置相应的查询条件。最终以统一的响应格式返回查询结果。

2. 获取并返回前端列表

该方法与上一个方法相同

3. 获取并返回列表

`@RequestMapping("/lists")` 注解标记，表示当接收到对应的 URL 请求时会执行该方法。

在方法中：

直接将请求参数映射到 `AddressEntity address` 中。

创建一个 `EntityWrapper` 对象 `ew`，用于构建查询条件。

调用 `EntityWrapper` 对象 `ew` 的 `allEq` 方法，将 `address` 对象中非空的属性作为查询条件，通过 `MPUtil.allEQMapPre` 方法将查询条件转换为 `Map`。

调用 `addressService` 的 `selectListView` 方法，传入构建好的查询条件 `ew`，获取查询结果。

使用 `R.ok().put("data", addressService.selectListView(ew))` 返回一个响应结果，其中 `"data"` 对应的值为查询到的数据结果。

该方法的作用是根据地址信息进行查询，并返回查询结果。

3. 获取查询并返回查询结果

该方法直接将请求参数映射到 `AddressEntity address` 中。

创建一个 `EntityWrapper<AddressEntity>` 对象 `ew`，用于构建查询条件。

调用 `EntityWrapper` 对象 `ew` 的 `allEq` 方法，将 `address` 对象中非空的属性作为查询条件，通过 `MPUtil.allEQMapPre` 方法将查询条件转换为 `Map`。

调用 `addressService` 的 `selectView` 方法，传入构建好的查询条件 `ew`，获取查询结果 `AddressView`。

使用 `R.ok("查询地址成功").put("data", addressView)` 返回一个响应结果，其中 `"data"` 对应的值为查询到的 `AddressView` 对象。

该方法的作用是根据地址信息进行查询，并返回查询结果。通过返回包含 `"查询地址成功"` 消息的响应，提供了查询操作成功的提示信息。

息。

4. 获取后端详情并返回结果

`@RequestMapping("/info/{id}")` 注解标记，表示当接收到对应的 URL 请求时会执行该方法

使用 `@PathVariable` 注解将 URL 中的 `{id}` 路径变量映射到方法的参数 `Long id` 中。

调用 `addressService` 的 `selectById` 方法，传入参数 `id`，根据 `id` 值查询对应的地址信息。

使用 `R.ok().put("data", address)` 返回一个响应结果，其中 `"data"` 对应的值为查询到的地址对象。

该方法的作用是根据 `id` 值查询地址信息，并以统一的响应格式返回查询结果。

5. 获取前端详情并返回结果

此方法与方法 4 相同

6. 前端 Id 信息传入后端进行保存

`@RequestMapping("/save")` 注解标记，表示当接收到对应的 URL 请求时会执行该方法。

在方法中：

使用 `@RequestBody` 注解将请求体中的 JSON 数据映射到 `AddressEntity address` 对象中。

生成一个唯一的 `id`，将当前时间戳与随机数相结合作为 `id`。

获取当前请求的 `HttpServletRequest` 对象，并从中获取当前用户

的 `userId`。

如果 `address` 的 `isdefault` 属性为"是"，则将该用户之前的默认地址设置为"否"，以保证只有一个默认地址。

将 `address` 对象的 `userid` 属性设置为当前用户的 `userId`。

调用 `addressService` 的 `insert` 方法，将 `address` 对象插入到数据库中。

使用 `R.ok()` 返回一个表示操作成功的响应结果。

该方法的作用是将前端传递的地址信息保存到数据库中，并返回一个表示操作成功的响应结果。

7.更新后端数据

与 6 中保存数据的方法大体相同，区别在于

调用 `addressService` 的 `updateById` 方法，根据 `address` 对象的 `id` 属性，更新对应的数据库记录。这里采用的是全部更新的方式。

该方法的作用是根据前端传递的地址信息进行更新操作，并返回结果

8.获取默认地址

该方法通过获取当前请求的 `HttpServletRequest` 对象，并从中获取当前用户的 `userId`。

创建一个 `Wrapper` 对象 `wrapper`，通过 `EntityWrapper().eq("isdefault", "是").eq("userid", request.getSession().getAttribute("userId"))` 设置查询条件。

调用 `addressService` 的 `selectOne` 方法，传入查询条件 `wrapper`，

查询满足条件的一条默认地址记录。

使用 `R.ok().put("data", address)` 返回一个响应结果，其中 "data" 对应的值为查询到的默认地址对象。

该方法的作用是根据当前用户的 `userId` 查询该用户的默认地址信息

9. 获取前端请求删除 id

具体操作方法与方法 6，7 大体 2 相同，不同在于将 `ids` 转换为 `List` 类型，并调用 `addressService` 的 `deleteBatchIds` 方法，批量删除对应 `id` 的地址记录。

该方法的作用是根据前端传递的 `id` 数组批量删除地址记录

10. 获取记录数量

使用 `@PathVariable` 注解将 URL 中的 `{columnName}` 和 `{type}` 路径变量分别映射到 `String` 类型的变量 `columnName` 和 `type` 中。

获取当前请求的 `HttpServletRequest` 对象，并从中获取当前用户的 `userId` 和角色信息。

将 `columnName` 和 `type` 以及其他请求参数封装到 `Map map` 中。

如果 `type` 等于 "2"，则根据请求参数 `remindstart` 和 `remindend` 计算出提醒的起始日期和结束日期，并将其格式化后放入 `map` 中。

创建一个 `EntityWrapper` 对象 `wrapper`，用于构建查询条件。

根据请求参数 `remindstart` 和 `remindend`，如果不为空，则分别设置大于等于和小于等于的查询条件。

如果当前用户角色不是 "管理员"，则添加 `userid` 等于当前用户

userId 的查询条件。

调用 addressService 的 selectCount 方法, 传入查询条件 wrapper, 获取符合条件的记录数量。

使用 R.ok().put("count", count) 返回一个响应结果, 其中 "count" 对应的值为符合条件的记录数量。

该方法的作用是根据提供的条件查询符合条件的记录数量

3.3.6 Interceptor（拦截器）

用于在请求处理的不同阶段进行拦截和处理的机制

这段代码是一个拦截器 Interceptor 类 AuthorizationInterceptor, 用于在 JavaWeb 应用中进行权限验证和用户身份认证。

@Component 注解: 表明 AuthorizationInterceptor 是一个 Spring 组件, 会被 Spring 容器进行管理。

HandlerInterceptor 接口: AuthorizationInterceptor 实现了 HandlerInterceptor 接口, 用于拦截处理程序执行链的方法。

preHandle 方法: 该方法是拦截器的预处理方法, 在请求处理之前执行:

设置支持跨域请求的相关响应头信息。

通过反射获取请求处理方法上的 IgnoreAuth 注解, 判断是否需要忽略权限验证。

从请求头中获取 token 信息。

如果请求方法不需要权限验证, 则直接放行。

根据 token 信息从 tokenService 获取 TokenEntity 对象, 其中包含

用户信息。

如果 token 验证通过，将用户信息存储到 session 中，表示用户已登录，返回 true 继续请求处理。

如果 token 验证不通过，返回 401 错误提示“请先登录”。

IgnoreAuth 注解：用于标记不需要权限验证的方法。

TokenService：通过@Autowired 注解注入 TokenService，用于处理 token 相关逻辑。

R.error(401, "请先登录")：返回一个包含错误码和提示信息的响应。

这段代码实现了在请求处理前对用户权限进行验证和身份认证的功能，通过拦截器实现统一的权限控制逻辑，确保只有登录用户才能访问需要权限的接口。

3.3.7 总结

流程：Controller 层（负责传递前端数据）与前端交互，将前端传来的数据通过调用 Service 层的方法进行处理，Service 层（负责业务模块的逻辑应用设计）调用 Dao 层（负责与数据库进行交互）中的方法，其中调用的参数是使用 Entity 层进行传递的，实现了整个网页的前端与后端交互运行。

3.4 前端设计

我们的前端设计主要用 HTML 定义页面的整体结构，再结合 CSS 的库和框架来美化页面，引用 JavaScript 库和框架使网页动起来，实现交互。注：因 js 与 jsp 不同（JSP 页面本质上是一个 Servlet 程序，

第一次访问 JSP 页面时(运行 Tomcat 服务器后在浏览器地址栏输入路径), Tomcat 服务器会将此 JSP 页面翻译成为一个 Java 源文件, 并对其进行编译成为.class 字节码文件(一个.java, 一个.class), 需要服务器。), js 通常运行在前台, 即运行于用户浏览器的, 通常情况下不需要后台服务器的支持, 混合在 HTML 中的 JavaScript 脚本程序直接被浏览器解析执行, 以提高页面的美观(主要是 bootstrap、jquery (主要引入在 resources 文件)、layui 等的 CSS 和 js)。所以我们还需要用 jquery、ajax 实现交互 (Ajax(Asynchronous Javascript And XML) 是一种在无需重新加载整个网页的情况下, 能够更新部分 网页的技术)。我们项目用 jquery 的 ajax 是\$的一个方法——ajax 调用这个方法并传参实现 ajax 数据交互。用\$.ajax()实现前端向后端穿数据

W3C 是万维网联盟, 这个组成是用来定义标准的。规定了一个网页是由三部分组成, 分别是:

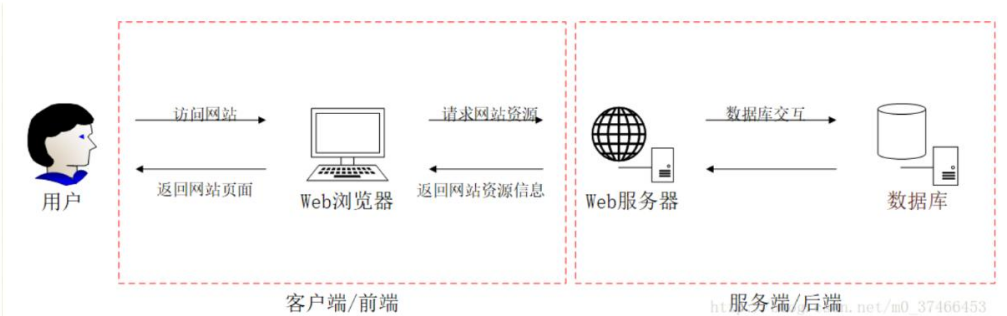
结构 (页面元素和内容): 对应的是 HTML 语言,HTML 优点 1.开发过程中, 前端和后端脱离, 交互通过 JSON 传输来实现 2.跨平台能力更强, 依托于浏览器的支持 3.是后台数据接口能够得到复用 HTML 缺点 1.开发难度大, 考虑浏览器的兼容性 2.页面请求过多 3.属于后加载, 无法被爬虫爬到 4.接口代码需要新增很多 5.无法直接显示 java 实体类对象, 需要转换为 json 格式

表现 (页面元素的外观、位置等页面样式如颜色、大小等): 对应的是 CSS 语言 (是通过选择器 (Selectors) 来选择 HTML 元素, 并为这些元素应用样式规则 (Rules)。由一个选择器和一组样式声明组成。

选择器指定了要应用样式的 HTML 元素，样式声明定义了要应用的样式属性和值。)

行为（交互效果）：对应的是 JavaScript 语言（是一种弱类型脚本语言（动态性解释性语言），其代码源不需经过编译，而是由浏览器内置的 JavaScript 引擎解释运行，实现业务逻辑和页面控制如交互性的网页功能、动态内容、表单验证、动态验证（决定功能））利用<script>标签将其代码嵌入至 HTML 文件中，外部 JavaScript 文件：将 JavaScript 代码放在外部文件中，并通过<script>标签引入到 HTML 文件中。这样可提高代码的可维护性和重用性。常见的 JavaScript 库和框架：如 Jquery、React、Vue.js 等。这些库和框架可以加速开发过程并提供更高级的功能。

web: (WWW world wide web) 即全球广域网，也称为万维网，它是一种基于超文本和HTTP的、全球性的、动态交互的、跨平台的分布式图形信息系统。是建立在Internet上的一种网络服务，为浏览者在Internet上查找和浏览信息提供了图形化的、易于访问的直观界面，其中的文档及超级链接将Internet上的信息节点组织成一个互为关联的网状结构。能够通过浏览器访问的网站。



注：1、先导入相关库的资源

3.4.1 Layui

LayUI 是一个基于 JavaScript 的前端开发框架，旨在简化 Web 界面的设计和开发。它使用了简洁友好的 API，并提供了丰富的组件和样式，使开发者可以快速构建美观、响应式的网页界面。LayUI 注重简洁易用和定制性

使用步骤：1、引入 LayUI 库：在 HTML 文件中，引入 LayUI 库的相关文件，包括`layui.js`和`layui.css`。可以从 LayUI 官方网站或其余网站下载最新版本的 LayUI 库，然后将这些文件复制到项目中。我们项目是从其它网站引用的：

```
/** layui-v2.5.6 MIT License By https://www.layui.com */
```

2. 初始化 LayUI：在 JavaScript 代码中，使用`layui.use()`方法来初始化 LayUI 并加载需要使用的模块。我们项目以 chat.html 页面举例

```
layui.use(['layer', 'element', 'http', 'jquery'], function() {
```

```
    var layer = layui.layer;
```

```
    var element = layui.element;
```

```
    var http = layui.http;
```

```
    var jquery = layui.jquery;
```

```
    app.init();
```

```
});
```

3. 使用 LayUI 组件：在`layui.use()`方法的回调函数中，使用 LayUI 提供的各种组件来构建界面和功能。

4. 编写 HTML 结构：根据需求，编写相应的 HTML 结构，可以使用 LayUI 提供的 HTML 标签和 CSS 类来构建界面。

```
<script src="../../layui/layui.js"></script>（我们采用的外联）
```

3.4.2 Bootstrap

（前端 UI 响应式框架，屏幕尺寸的变化，让页面有不同的布局效果，界面跟随屏幕尺寸变化而变化），Bootstrap 是基于 HTML、CSS、

JavaScript 的前端框架。该框架已经预定义了一套 CSS 样式和样式对应的 JS 代码。开发人员只需要编写 HTML 结构，添加 bootstrap 固定的样式就可以轻松完成指定效果的实现。Bootstrap 注重响应式布局和社区支持。

使用步骤：1）本地引入（必须先下载）a.引入 bootstrap.css 或 bootstrap.min.css

b.引入 JS（如果不使用插件，不需要引入 JS）bootstrap.js 或 bootstrap.min.js。注意：bootstrap.js 依赖于 jquery.js，在引入它之前必须先引入 jquery.js。

（2）CDN 引入（不需要下载）（推荐使用这种方法引入）

(<https://bootcss.cn>)引入 CSS 和 JS

我们使用的是 Bootstrap v3.3.6，在 (<http://getbootstrap.com>)引入的，然后 bootstrap 必须需要至少一个布局容器，才能为页面内容进行封装和方便的样式控制。这个布局容器就相当于一个画板，我们所需要的样式和组件都在上面进行使用。我们项目使用 div 作为布局容器。

JQuery：是一个高效、精简并且功能丰富的 JavaScript 工具库。它提供的 API 易于使用且兼容众多浏览器，这让诸如 HTML 文档遍历和操作、事件处理、动画和 Ajax 操作更加简单。

使用步骤：1、本地引入：将 jQuery 下载下来，然后导入项目中，使用 script 标签进行引用；我们项目是下载的：jQuery v3.3.1

CDN 引入：使用远程 CDN 资源库在线引入，避免了文件下载

作用：1.选择器（基本、层级、过滤、表单）2.工具：实现相关方法、事件、动画等 3.ajax：执行一个异步的 HTTP 请求等等

2、对库和框架的使用方法：外链式：我们项目将 CSS 和 JavaScript 代码保存在独立的外部文件中，CSS 外链式：采用的<link>标签引用 CSS 文件至 HTML 文件中，js 外链式：通过 src 属性将文件属性引入到 HTML 文件中。

例如：<link rel="stylesheet" href="../../layui/css/layui.css">

<script src="../../layui/layui.js"></script>

系统前端的体验是现阶段重点考虑的一个问题，较好的人机交机，能有效提升用户的体验，所以我们选用比较成熟的 Bootstrap 来做页面开发，使用 JQuery 来进行异步请求处理，数据存储方面本系统数据库则使用 My SQL。本系统目前主要利用这些技术和工具来开发网站，主要实现的系统功能：一是鲜花线上销售管理系统前台用户注册之后，根据注册账户登陆系统，浏览商品并进行购买，并可以在个人中心中查询查询订单等；二是鲜花线上销售管理系统后台管理用户登陆系统后，可以实现像包含用户信息管理、商品类别管理、商品管理、订单管理等以及网站新闻管理等信息管理的功能。鲜花线上销售管理系统是借助 Internet 移动互联网的应用技术,实现了资源的共享,借助于网络平台各种形式的商品展示窗体,让网上客户能够更易地接受这个网上商城的交易网站,并且打破以往网上购买商品的局限性,缩短了用户寻找商品的难度和时间。

3.4.3Vue.js 2

前端的文件：**webapp**（链接后台 **vue.js**）：

admin：**dist** 文件夹（**vue.js** 编辑输出）：编译输出后或者压缩后的代码，最终发布版本的压缩代码；**Public** 文件夹:公开代码；

front：**css**（**Cascading Style Sheets** 层叠样式表，美化网页，主用 **layui**，解决是内容与表现相分离的问题）：**bootstrap.min.css**：（使用响应式设计）这是 **Bootstrap** 框架的主要样式文件，它包含了整个框架的样式定义，用于构建网页界面的外观和布局。这个文件是压缩和优化过的，适合直接在生产环境中使用。将这个文件嵌入到网页中，以应用 **Bootstrap** 的样式（引入 **bootstrap** 样式时 **layui** 表单样式会被影响，需要加入 **css** 设置）。**Common.css**:网页设计常用的基本 **css** 控制，商品展示网页嵌套文字横线设计代码 **Style.css**:样式位置，大小（长宽）设计等，如导航栏，轮播图，商品推荐样式，底部栏，留言，首页新闻图等 **Theme.css**：相关板块背景、颜色设置。

Elementui：**fonts**：字体颜色大小样式等（**maven**）

Img：用于存放网页所用的图片背景、图标等

Js:使网页动起来，常见的 **JavaScript** 库和框架（插件）：如我们使用的 **bootstrap.min.js** 是经过压缩的 **JavaScript** 文件，其中的空格、注释和不必要的字符都已被删除，从而减小了文件大小。这个文件通常用于生产环境，以减少加载时间和网络带宽的占用。在开发阶段我们使用 **bootstrap.js** 以便进行调试，然后在将网站部署到生产环境时切换到 **bootstrap.min.js** 以获得更好的性能。这种做法可以在开发和生产环境之间取得平衡，同时兼顾了可读性和性能。（可实现地址选择时模态

框、选项卡菜单及面板。首页轮播图)；**config**:调整 Vue CLI 生成的项目的默认配置(轮播图配置,指定部署应用时的基本 URL 网页地址。等)、**Jquery**: 异步请求处理，其中 jQuery Ajax 是 jQuery 提供的一个功能强大的异步通信方法，它允许网页与服务器之间进行数据交换，而无需重新加载整个页面。通过使用 Ajax，开发者可以实现更加流畅和响应式的用户体验，减少不必要的网络传输和页面加载时间（网页前端刚输入完数据，后端就验证完了，客户端发出请求无需响应就可以进行第二次请求）。、**utils**: 是一个轻量级、功能丰富的 JavaScript 实用工具库，旨在帮助开发者简化常见的代码编写任务，提高开发效率。该项目以模块化设计和易用性为核心，提供了多种实用的辅助函数，可广泛应用于前端和后端 JavaScript 环境中。按需引入 Utils.js 提供了许多实用的功能，例如：字符串操作：包括格式化、比较、截取等，有助于字符串的处理。数组操作：提供了便利的数组过滤、查找、合并等功能，使数组操作更加简单。日期时间：提供格式化日期、计算差值等方法，处理日期和时间更便捷。对象操作：包括对象的深拷贝、合并等，便于处理复杂的数据结构。类型判断：准确地检测变量类型，为条件判断提供依据。数学运算：包含一些基本的数学操作，如随机数生成、最大最小值查找等。页面跳转，生成订单；这些功能可以在日常开发中广泛应用于数据处理、逻辑控制等多个环节，有效提升开发速度和代码质量。、**validate**: 验证库，提供了一种简单易用的方式来验证输入数据的有效性。通过使用 validate.js，可以快速地创建复杂的表单验证规则，以确保应用程序接收到的数据是正确的。

如验证码功能等、**Vue.js**：帮助开发者构建交互式的用户界面。它的设计目标是通过简单的 **API** 和高效的数据绑定，让开发者更容易地构建可维护的应用程序（我们项目用它来验证用户注册时邮箱、手机号码、身份证等数据是否正确）。**Vue.js** 的核心特性包括：**1. 响应式数据绑定**：**Vue.js**（**admin** 那个板块搭建后端时，使用的是整个 **Vue.js** 框架，借助 **load.js** 搭建）可以轻松地将数据与 **DOM** 元素进行绑定，当数据发生变化时，相应的 **DOM** 元素也会自动更新（比如用户商家聊天，用户评论界面）。**2. 组件化开发**：**Vue.js** 支持组件化开发，开发者可以将应用程序拆分成多个独立的组件，每个组件都有自己的状态和逻辑。**3. 简单的 API**：**Vue.js** 提供了简单易用的 **API**，开发者可以很容易地理解和使用它。**4. 虚拟 DOM**：**Vue.js** 使用虚拟 **DOM** 来提高性能，它会将 **DOM** 的更新操作优化成批量操作，从而减少 **DOM** 操作的次数。

等。这些库和框架可以加速开发过程并提供更高级的功能。

layui:1.简化界面开发：**LayUI** 提供了丰富的组件、样式和布局，使界面开发变得简单和直观。通过引入 **LayUI**，并使用其简洁易用的 **API**，开发者可以通过少量的代码实现复杂的界面效果。这极大地简化了界面的开发流程，降低了开发的复杂性和难度。

--**2. 提高开发效率**：**LayUI** 的设计目标之一是提高开发效率。它采用了模块化的开发方式，将各个组件和功能拆分成独立的模块。开发者可以根据自己的需求选择性地引入和使用这些模块，避免了加载不必要的组件和功能，提高了页面加载速度。此外，**LayUI** 的简单易用的

API 和丰富的组件库，使开发者能够快速开发出功能完备、美观的网页界面。

-- 3. 响应式布局支持：LayUI 提供了响应式布局的支持，使页面能够自动适应不同的屏幕尺寸和设备。开发者可以使用 LayUI 提供的响应式的布局组件和工具，根据不同的设备，自动调整页面的布局和显示效果。这使得开发的网页在不同终端上都能够有良好的显示效果，提供更好的用户体验。

-- 4. 提供丰富的组件和样式：LayUI 提供了一系列常用的界面组件，如表单、表格、弹窗、导航、菜单等等。这些组件都经过精心设计和优化，具有良好的交互体验和视觉效果。此外，LayUI 还提供了多种样式和主题供选择，开发者可以根据实际需求，自由搭配和定制界面风格。

-- 5. 可定制性：LayUI 提供了开放的扩展机制，允许开发者根据自己的需求进行二次开发和定制。开发者可以根据具体项目的需求，在 LayUI 的基础上进行扩展和定制，实现个性化的界面和功能。这使得 LayUI 具有更强的灵活性和可扩展性，能够适应各种不同的项目需求。

modules:相关已有的库。收货地址。http: 网络请求接口扩展，layui: 省市区联动选择器，tinymce: 文本编辑器

pages (web 页面设计):用 html (“超文本标记语言” 超文本: 超越了文本的限制，比普通文本更强大。除了文字信息，还可以定义图片、音频、视频等内容。除了能看到一些文字，同时也有大量的图片展示; 有些网页也有视频，音频等。这种展示效果超越了文本展示的限制。

标记语言：由标签构成的语言。XML 就是标记语言，由一个一个个的标签组成，HTML 也是由标签组成。HTML 中的标签都是预定义好的，运行在浏览器上并由浏览器解析，然后展示出对应的效果。)，由 html 标签构成<html>标签表示定义整个 html 文档<head>包含文档的元数据(关于文档的信息如标题和引用的样式)<title>定义文档标题<body>包含实际的页面内容等；使用 Layui 这款轻量级前端框架，它提供丰富的组件和易于使用的 API。在使用它时，先引入 layui 的 CSS 和 js 文件。layui 常用内置方法加载模块(核心)layui.use([mods],function()){}, 用于异步加载模块并执行相应的回调函数，按需加载，提高页面性能。通过灵活地结合 Layui 的组件和自定义函数，创建符合需求的交互式前端应用。

列举一个 chat (聊天界面)先在<style>标签设计页面布局样式,用<div>标签定义文档中的分区或节,用于容纳网页中的文本模块、图像模块,从而实现网页的规划和布局。将文档样式分割为独立、不同的部分,并用 id, class 来标记,使其更有效。其中 id 用于标识单独的唯一元素,在页面中只能出现一次,不能重复使用。class 用于元素组,可反复使用,与 CSS 一起使用。然后用<script>标签(脚本标签)嵌入脚本代码,用于在页面加载时或页面中的特定位置执行脚本。其作用是实现动态内容:用于动态生成、修改或删除 HTML 元素,从而实现页面内容的动态更新和交互。比如通过 js 脚本,用户可动态地向页面添加新的聊天,用户发表聊天语言后,页面会实时更新显示最新的聊

天内容。处理用户输入与服务器通信，可通过 Ajax、API 等技术与服务
器进行通信，从而实现数据的异步加载、更新和交换。再用

layui.use(['layer' , 'element' , 'http' , 'jquery'],function(){}):layer(用
于创建弹出层、提示框和对话框的模块，用于展示信息和用户进行交
互)element(用于处理页面元素、导航栏和选项卡的模块，用于构建多
页面应用)http()jquery(DOM 库)，引入这四个库

config（add 用户注册界面，detail 用户信息界面 list? 首页轮播图管
理界面）discusshangjia（add 商家注册界面，detail 商品评论区界面
list 商家评论表）discussxianhuayoulan（鲜花游览评论表）

login（用户登录界面）

home（首页界面）

news（detail 系统公告首页，list 公告详情）

shangjia（add 商家上传信息页面 center 商家个人中心 detail 商家介绍
list 商家搜索界面，register 商家注册账号）

shop-address:（add 收货地址添加界面 list 收货地址详情页面 update
收货地址修改界面）

Shop-cart: 用户购物车详情页面

Shop-order: confirm 用户下单界面，list 用户我的订单界面

Shop-recharge: 用户支付界面

Storeup: 用户我的收藏界面

Users: add 后台管理员注册界面，detail 管理员详细信息界面，list
管理员管理界面

Xianhuafenlei: 后台管理员对鲜花分类的界面, detail 详情信息面

Xiahuajieshao: 鲜花大体介绍页面

Xianhuayoulan: add 商家上传鲜花详情的页面 detail 用户看到的某一类鲜花商品详情页面, list 用户浏览所有鲜花商品的页面

Yonghu: add 用户注册界面 center 用户个人中心界面 detail 用户首页详情界面 list 用户账号搜索界面 register 用户首次登录网站注册界面

xznstatic: 存放静态资源。

Resources:Bootstrap、jQuery 资源文件

Upload:图片

3.5 数据库设计

3.5.1 需求分析

背景:

随着互联网技术的发展,电商市场已经进行的如火如荼,网上购物已经是趋势

网上购物系统的开发可以提供更多的便利.网上购物弥补了传统购物的一些局限性。而且这对于管理人员来说,利用购物系统能轻松地管理大量的商品。

我设计的网上购物系统的数据库开发存储,其可以存储销售商需要卖出的商品信息用户的所有信息.方便用户根据自己的喜好,浏览

自己喜欢的东西. 若要想购买商品, 通过注册成为用户进行购买, 登陆后就可以购物了。商店的各种商品都进行了详细地分类, 可以轻松地找到想要的商品。同时买到自己喜欢的东西后就可以去结算. 通过填写自己的相关信息, 以便下订单, 从而完成购物的流程。

现在准备建立网上购物系统。要把销售商的商品、用户的信息等基本信息进行管理。确保有足够的安全性。

3.5.2 数据需求

初次登陆的用户可以进行浏览系统公告页面和商品游览页面, 登陆成功后可以按照商品的分类进行查看商品。或按照自己的需求收藏, 购买商品。商家可以提供自己销售的商品的详细信息, 并可以进行修改。以便让用户可以查看商品的信息。

当用户确认购买的商品后, 在付款成功的条件下, 销售商就立即给会员邮寄商品, 等待会员查收。

数据需求, 如下表

数据名称	数据类型	长度	列名
记录编号	整型		Id
创建时间	时间型		Addtime
用户编号	字符型		userid
电话	字符型	200	Phone
地址	字符型	200	Adress
是否默认地址	字符型	200	Isdefault
商品表名	字符型	200	Tablename
商品编号	字符型		Goodid
商品图片	字符型	200	Picture
购买数量	整形		Buynumbel
单价	浮点型		Price
会员价	浮点型		Discountprice
管理员编号	整型		Adminid

提问	文本型		Ask
回复	文本型		Reply
是否回复	整型		Isreply
关联表编号	整型		Refid
用户名	字符型	200	Nickname
评论内容	文本型		Content
公告标题	字符型	200	Title
简介	文本型		Introduction
内容	文本型		Content
订单编号	字符型	200	Orderis
总价	浮点型		Total
折扣总价	浮点型		Discounttotal
支付类型	整型		Type
状态	字符型	200	Status
商家账号	字符型	200	Sahngjaizhanghao
密码	字符型	200	mima
店铺名	字符型	200	Dianpuming
店铺地址	字符型	200	Dianpudizhi
联系方式	字符型	200	Lianxifashi
联系人	字符型	200	Lianxiren
商家图片	字符型	2000	Shangjiatupian
经营范围	字符型	200	jingyingfanwei
商家介绍	文本型		Shsangjiajieshao
余额	浮点型		Money
收藏编号	整型		Refid
角色	字符型	200	Role
密码	字符型	200	Token
鲜花分类	字符型	200	Xianhuafenlei
鲜花名称	字符型		Xianghaumingcheng
鲜花用途	字符型		Xianghauyongtu
赠送对象	字符型		Zengshongduixiang
订单图片	字符型		Tupian
相关花语	字符型		Xiangguanhuayu
商品详情	文本型		Shangpinxianqing
最近点击时间	时间型		Clicktime
点击次数	整型		Clicknum
用户账号	字符型	200	Yonghuzhanghao
性别	字符型	200	Xingbie
年龄	整型		Nainling
头像	字符型	200	Touxiang

数据结构

数据结构名	说明	组成
-------	----	----

地址	记录收货的基本信息	用户名、用户编号、电话、地址，是否为默认地址
订单	记录用户的购物信息	用户编号、商品编号、订单编号、商品名称，商品表名，商品图片，购买数量，价格，折扣价，总价格，折扣总价格，支付类型，状态，地址
购物车	存放用户需要购买的商品	用户编号、商品编号、商品名称、商品数量、单价，商品表名，图片，会员价
管理员	记录管理员的基本信息	管理员编号、管理员姓名、密码、联系电话
在线客服	记录用户评论和商家回复信息	用户编号，管理员编号提问，回复，是否回复
配置文件	系统公告背景轮播图	配置参数名称，配置参数值，编号
商家评论	记录商家页面评论信息	关联表编号，用户编号，用户表名，评论内容，回复内容
鲜花评论	用于记录各类花的评论	关联表编号，用户编号，用户表名，评论内容，回复内容
系统广告	记录管理员发布的公告	标题，简介，图片，内容
商家	记录商家信息	商家账号，密码，店铺名，店铺地址，联系方式，联系人，商家图，经营范围，商家介绍，余额
收藏	记录用户收藏商品信息	用户编号，收藏编号，收藏图片，
Token	用于验证登录信息	用户编号，用户名，表名，角色，密码，新增时间，过期时间
管理员（user）	记录管理员信息	用户编号，用户名，表名，角色，密码，新增时间，过期时间
鲜花分类	记录鲜花类	鲜花分类
鲜花游览	记录鲜花商品具体信息	鲜花名称，鲜花分类，鲜花用途，赠送对象，图片，相关花语，商品详情，商家账号，店铺名，店铺地址，联系方式，联系人，最近点击时间，点击次

		数，价格
用户	记录用户信息	用户账号，密码，用户姓名，性别，年龄，头像，用户手机，余额

3.5.3 事物需求

(1) 在注册用户信息管理部分。要求：

- a 可以查看注册用户信息。
- b. 可以对注册会员信息进行添加及管理的操作。

(2) 在商品信息管理部分。要求：

- a 可以浏览商品信息。
- b. 可以对商品信息进行维护, 包括添加及删除的操作。

(3) 在购物车信息管理部分，要求：.

- a 可以浏览购物车信息。
- b. 可以对购物车信息进行维护, 包括添加及删除的操作。

(4) 在订单信息管理部分，要求：

- a 可以浏览订单信息。
- b. 可以对订单信息进行维护，包括添加及删除的操作。

(5) 在管理员信息管理部分。要求：

- a 可以浏览管理员信息
- b. 对管理员信息进行维护，包括添加及删除的操作。

(6) 在商品组信息管理部分，要求：

- a 显示当前商品组信息。
- b. 对商品组信息进行维护操作，包括添加及删除的操作。

(7) 在管理员一权限管理部分:要求:

- a 可以浏览管理员与其对应的管理权限。
- b. 对管理员一权限信息进行更新。

(8) 在权限信息管理部分, 要求:

- a 可以浏览和购买操作相关的管理权限。
- b. 对权限信息进行更新。

2. 概念结构设计

E-R 图见附件 1。

3.6 MySQL 设计

3.4.1 MySQL 环境

项目选择 JDK19+MySQL Community Server 8.2.0+Tomcat9.0 作为项目的运行时环境。

数据库选择远程云服务器上的 MySQL8.2 服务器, 确保了数据的同一性和持久性。同时 MySQL8.2 有以下优点:

性能提升: MySQL 8.2 在性能方面做了大量的优化, 包括查询优化、存储引擎优化和 CPU 优化等。这使得 MySQL 8.2 可以更高效地处理大量数据, 提高系统的吞吐量和响应速度。

更好的安全性: MySQL 8.2 引入了新的密码哈希算法, 增强了用户认证的安全性。同时, 还增加了更多的安全功能, 如审计功能和更强的加密算法等, 以保护数据库的安全。

更好的可扩展性: MySQL 8.2 支持更多的数据类型和存储引擎, 使得数据库可以更好地适应不同的应用场景。同时, MySQL 8.2 还提

供了更好的水平扩展和垂直扩展能力，可以轻松地扩展数据库的性能和容量。

3.4.2 数据库表的设计

数据库表的设计采用切面设计，实现各种功能。

网上鲜花销售平台的主数据库为 `flower_transaction`，另有一数据库 `mybatis-plus` 用于测试 `MyBatis-Plus` 框架的功能。其中数据库

名	自动递...	修改日期	数据长度	引擎	行	注释
address	17046...	2024-01-07 13:44:...	16 KB	InnoDB	6	地址
cart	17046...	2024-01-07 13:44:...	16 KB	InnoDB	2	购物车表
chat	17046...	2024-01-07 15:26:...	16 KB	InnoDB	27	在线客服
config	10	2024-01-07 01:25:...	16 KB	InnoDB	3	配置文件
discussshangjia	16206...	2024-01-08 20:43:...	16 KB	InnoDB	6	商家评论表
discussxianhuayoulan	17045...	2024-01-08 20:44:...	16 KB	InnoDB	3	鲜花游览评论表
news	16203...	2024-01-07 02:07:...	16 KB	InnoDB	2	系统公告
orders	17046...	2024-01-07 13:44:...	16 KB	InnoDB	22	订单
shangjia	17047...	2024-01-08 20:50:...	16 KB	InnoDB	6	商家
storeup	17045...	2024-01-07 01:53:...	16 KB	InnoDB	1	收藏表
token	18	2024-01-08 20:48:...	16 KB	InnoDB	17	token表
users	5	2024-01-08 20:52:...	16 KB	InnoDB	4	用户表
xianhuafenlei	16203...		16 KB	InnoDB	6	鲜花分类
xianhuayoulan	16203...	2024-01-09 00:58:...	16 KB	InnoDB	5	鲜花游览
yonghu	17047...	2024-01-08 20:25:...	16 KB	InnoDB	7	用户

`flower_transaction` 的结构如下：

共计 15 张表，以实现地址、购物车、在线客服、配置文件、商家评论、鲜花游览评论、系统公告、订单、商家、收藏、`token`、管理员、鲜花分类、鲜花游览、用户的数据储存以及各种增删查改（`CRUD`）操作。因部分涉及到隐私原因，仅展示部分数据表的结构。完整建表 `SQL` 语句可在源代码中查看。

在线客服数据表的结构如下：

id	addtime	userid	adminid	ask	reply	isreply
1620619954805	2021-05-10 12:12:34	11	(Null)	你哈哦	(Null)	0
1620620026297	2021-05-10 12:13:45	11	(Null)	你你你	(Null)	0
1704562943137	2024-01-07 01:42:22	1703240622194	(Null)	你好	(Null)	0
1704563668442	2024-01-07 01:54:28	1703240622194	1	(Null)	您好, 请问有什么问题吗?	0
1704563811103	2024-01-07 01:56:50	1703240622194	(Null)	如何充值系统	(Null)	0
1704563873694	2024-01-07 01:57:53	1703240622194	1	(Null)	充值系统在个人中心喔~可	0
1704563875451	2024-01-07 01:57:55	1703240622194	(Null)	这都不会	(Null)	0
1704563919621	2024-01-07 01:58:38	1703240622194	1	(Null)	!	0
1704563951254	2024-01-07 01:59:10	1703240622194	(Null)	你这花保熟吗	(Null)	0
1704563989168	2024-01-07 01:59:48	1703240622194	1	(Null)	不保熟	0
170459994325	2024-01-07 11:59:53	11	(Null)	我是傻逼	(Null)	0
1704608841326	2024-01-07 14:27:21	11	2	(Null)	我是客服yangshuai	(Null)
1704608859155	2024-01-07 14:27:38	1703240622194	(Null)	请问是人工客服吗?	(Null)	0
1704609279415	2024-01-07 14:34:38	1704609170027	(Null)	? ? ?	(Null)	0
1704609329641	2024-01-07 14:35:28	1704609170027	1	(Null)	啊?	0
1704609329670	2024-01-07 14:35:29	1704609170027	(Null)	能不能上门安装	(Null)	0
1704609365628	2024-01-07 14:36:05	1704609170027	1	(Null)	部分快递支持上门服务的	(Null)
1704612243338	2024-01-07 15:24:02	1704605952168	(Null)	hello world	(Null)	0
1704612262796	2024-01-07 15:24:22	1703240622194	1	(Null)	是的	(Null)
1704612295454	2024-01-07 15:24:54	1704605952168	1	(Null)	System.out.println("你好"	0
1704612324489	2024-01-07 15:25:23	1704605952168	1	(Null)	<h3>HelloWorld</h3>	0
1704612376859	2024-01-07 15:26:16	1704605952168	(Null)	cout<<"hello world";	(Null)	1

系统公告数据表结构如下：

id	addtime	title	introduction	picture	content
1620372349849	2023-12-22 15:25:49	鲜花上新啦	菊花上新	http://platform.privaterh	<p>菊花上新</p>
1620372396617	2024-01-09 01:26:35	系统更新	系统更新	http://platform.privaterh	<p>系统更新请输入新网址

订单数据表部分结构如下：

id	addtime	orderid	tablename	userid	goodid	goodname	picture	buynumber	price	disc
1620373076774	2021-05-07 15:37:56	2021571537563493230	xianhuayoulan	11	1620372824492	向日葵	http://platform.privaterh	1	55	
1620373109834	2021-05-07 15:38:29	20215715382971188417	xianhuayoulan	11	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1620373121881	2021-05-07 15:38:41	2021571538415963965	xianhuayoulan	11	1620372869977	菊花	http://platform.privaterh	4	54	
1620373151367	2021-05-07 15:39:11	2021571539111030017	xianhuayoulan	11	1620372824492	向日葵	http://platform.privaterh	1	55	
1620373159934	2021-05-07 15:39:19	2021571539196420119	xianhuayoulan	11	1620372705300	月季	http://platform.privaterh	1	13	
1620373169846	2021-05-07 15:39:29	2021571539294859883	xianhuayoulan	11	1620372598896	向日葵	http://platform.privaterh	1	99	
1620373211212	2021-05-07 15:40:10	2021571540106392298	xianhuayoulan	11	1620372536411	99朵玫瑰	http://platform.privaterh	35	999	
1620395900758	2021-05-07 21:58:20	2021572158199614980	xianhuayoulan	11	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1620395900759	2021-05-07 21:58:19	2021572158199355153	xianhuayoulan	11	1620372705300	月季	http://platform.privaterh	1	13	
1620395932524	2021-05-07 21:58:52	2021572158523909067	xianhuayoulan	11	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1620396706094	2021-05-07 22:11:45	2021572211459578437	xianhuayoulan	11	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1620616804381	2021-05-10 11:20:04	2021510112041326471	xianhuayoulan	11	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1620616807995	2021-05-10 11:20:07	2021510112071609708	xianhuayoulan	11	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1620616813444	2021-05-10 11:20:12	2021510112012524379	xianhuayoulan	11	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1660897746093	2022-08-19 16:29:05	2022819162955905857	xianhuayoulan	1660843983170	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1660897790834	2022-08-19 16:29:49	2022819162949759083	xianhuayoulan	1660843983170	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1703255363832	2023-12-22 22:29:15	2023122222292194910	xianhuayoulan	1703240622194	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1704565077228	2024-01-07 02:17:56	2024172175649012759	xianhuayoulan	1703240622194	1620372536411	99朵玫瑰	http://platform.privaterh	1	999	
1704565118893	2024-01-07 02:18:38	2024172183820119588	xianhuayoulan	1703240622194	1620372668819	绿萝	http://platform.privaterh	1	33	
1704565166593	2024-01-07 02:19:26	2024172192624587811	xianhuayoulan	1703240622194	1620372598896	向日葵	http://platform.privaterh	2	99	
1704601146450	2024-01-07 12:19:06	2024171219570522576	xianhuayoulan	1703240622194	1620372536411	99朵玫瑰	http://platform.privaterh	35	999	
1704606291457	2024-01-07 13:44:50	2024171344514724971	xianhuayoulan	1704605952168	1620372536411	99朵玫瑰	http://platform.privaterh	6	999	

id	addtime	userid	refid	tablename	name	picture
1620618308207	2021-05-10 11:45:07	11	1620372536411	xianhuayoulan	99朵玫瑰	http://platform.privaterh
1704563583040	2024-01-07 01:53:02	1703240622194	1620372536411	xianhuayoulan	99朵玫瑰	http://platform.privaterh

收藏数据表部分结构如下：

第 4 章 系统实现

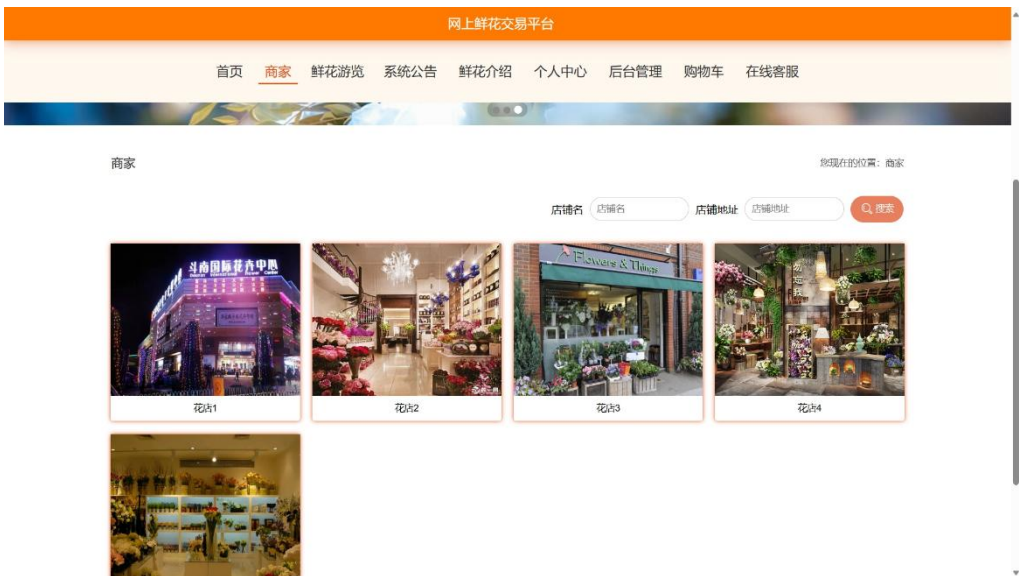
鲜花销售系统的前端可分为前台展示界面和后台管理系统两个部分，分别使用 Bootstrap 和 vue 框架实现。

4.1 前台界面实现



4.1.1 前台首页

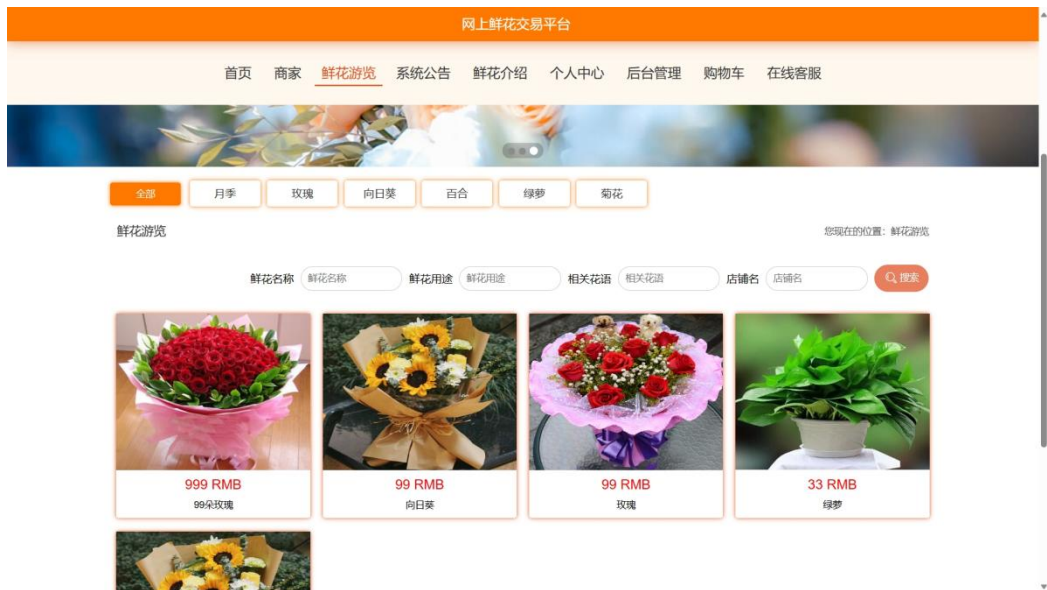
在这里呈现鲜花分类的图片和商家图片（都以图片链接的形式呈现），通过在首页点击心仪的花或商家查看到相关的商家和花的信息。



4.1.2 前台商家

在这个界面通过搜索商家名称和商家地址来搜索想要的商家，点击商家图片查看商家信息，也可进行评论该商家。

4.1.3 前台鲜花游览

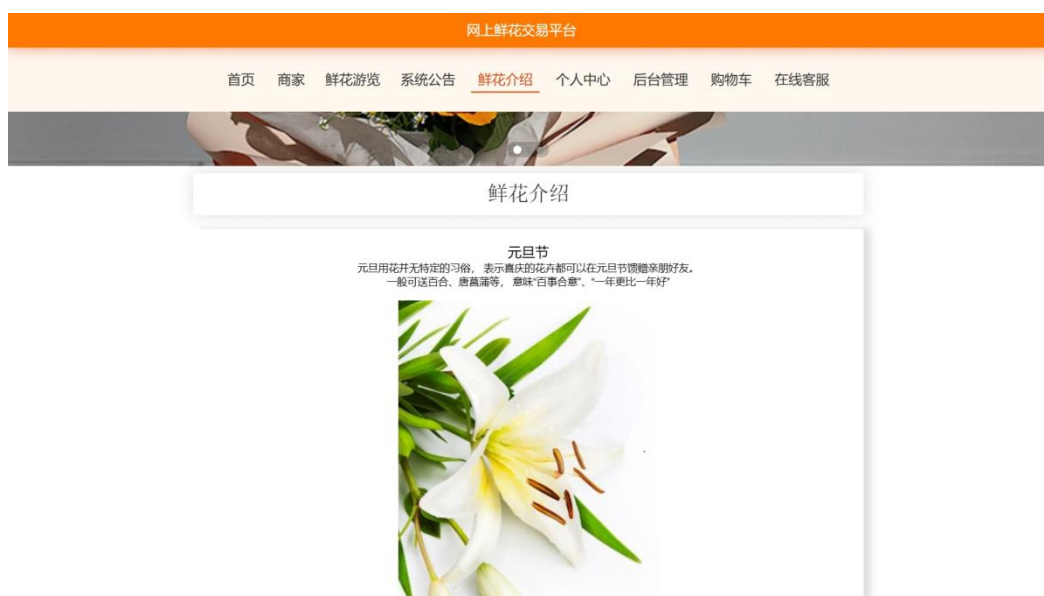


该界面可阅览鲜花图片来寻找心仪的鲜花，点击图片后可查看鲜花信息。也可在该界面直接搜索鲜花进行查看。

4.1.4 前台系统公告



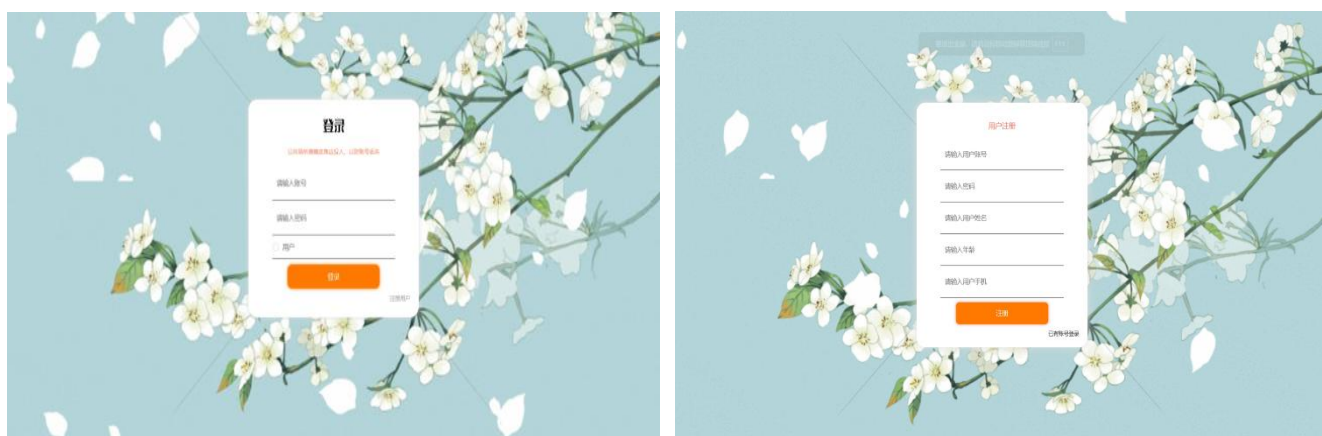
可在该界面查看由网站管理发布的公告。



4.1.5 前台鲜花介绍

在这里可以查看不同鲜花的介绍包括（用途，寓意，使用场景，花的特点等等）。

4.1.6 前台的登录和注册

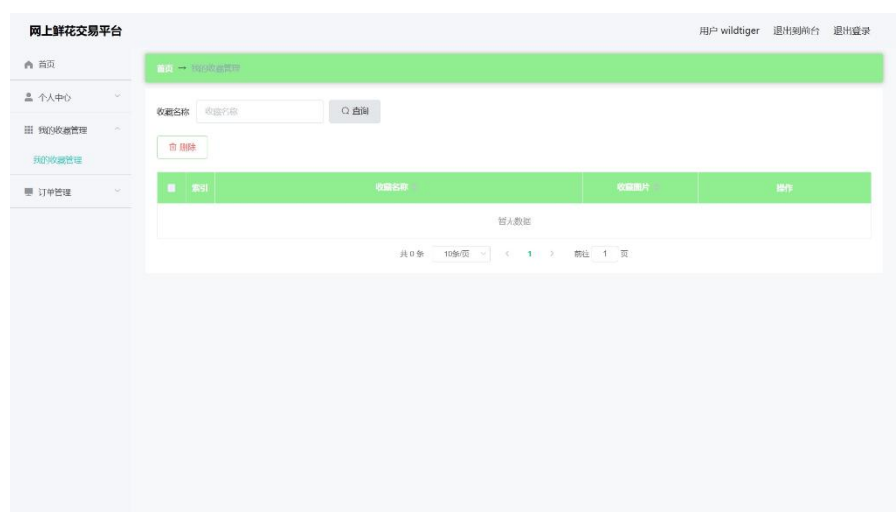


在这两个界面实现用户的登录和注册。用户在未登录前访问鲜花游览、个人中心、购物车等页面将自动跳转到登录页面。

注册用户的信息将保存在远程数据库服务器上，我们保证不会将个人信息泄露给任何第三方。

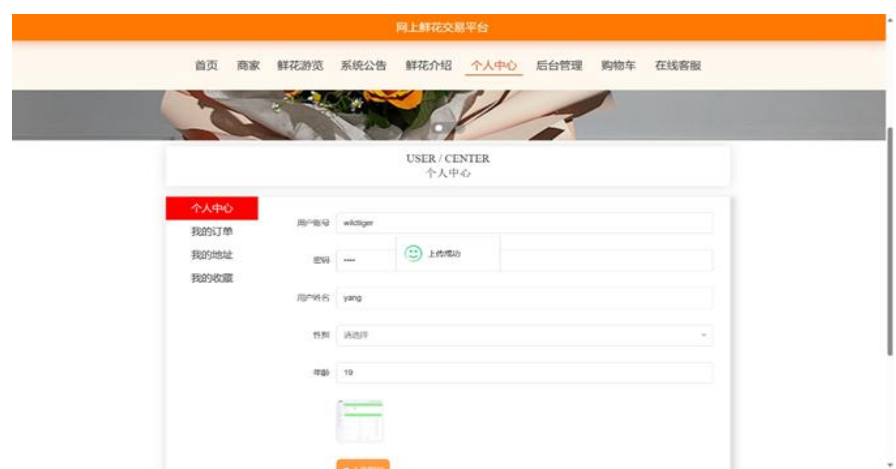
4.2 后台界面实现

4.2.1 用户后台



这里可以查看个人信息并进行修改，也可在这里查看并搜索自己的收藏。

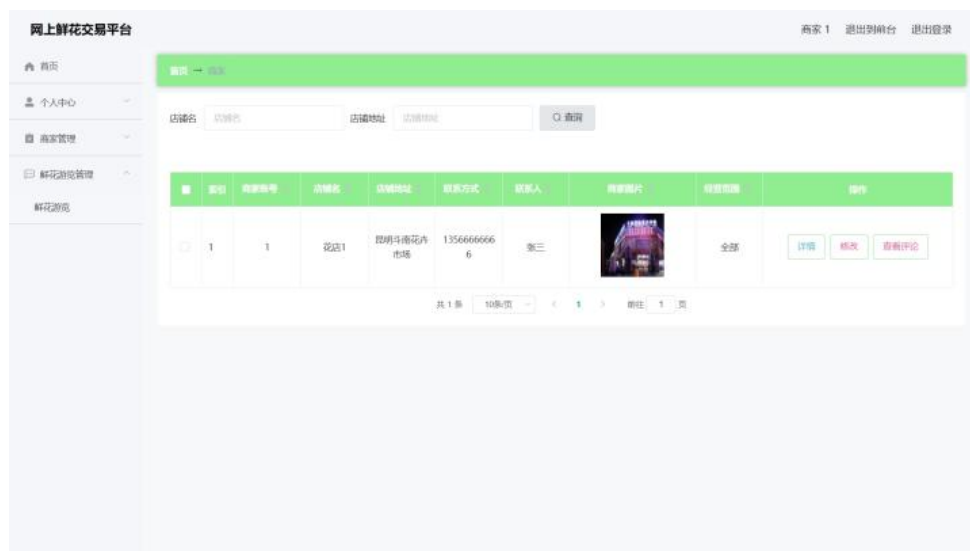
4.2.2 用户个人中心及用户联系客服界面



在这里可以上传自己的照片等信息。



这里可以与在线客服聊天，对商家，花的质量，平台等进行反馈（该评论将反馈到管理员，由网站管理员进行回复）。



4.2.3 商家后台

通过商家登录后进入后台管理进入到该界面；该界面可由商家自己查看和修改自己店铺信息，并对该商店的花的种类进行修改。



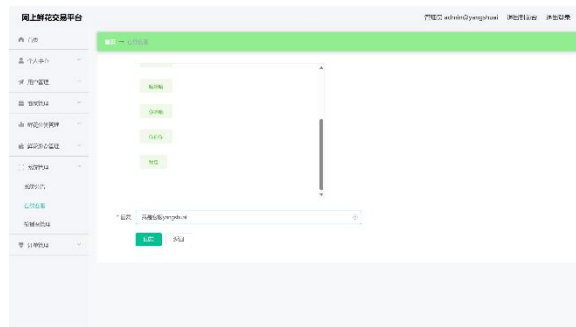
4.2.4 管理员后台

在登录界面选择管理员登录后即可进入管理员后台进行管理

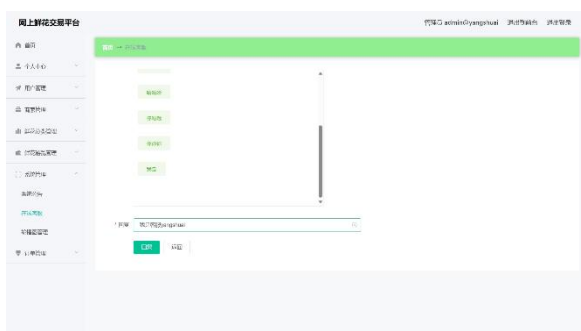
!! 以下是管理员后台可实行的操作（包括但不限于）!!



进行订单处理



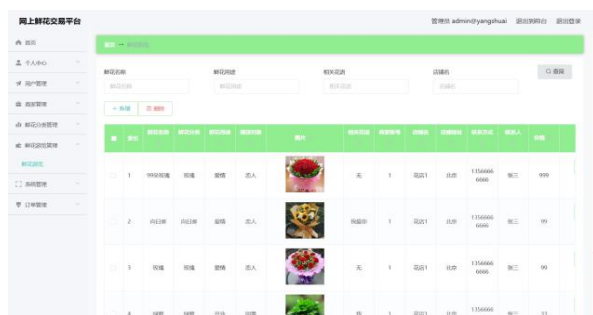
后台商家管理



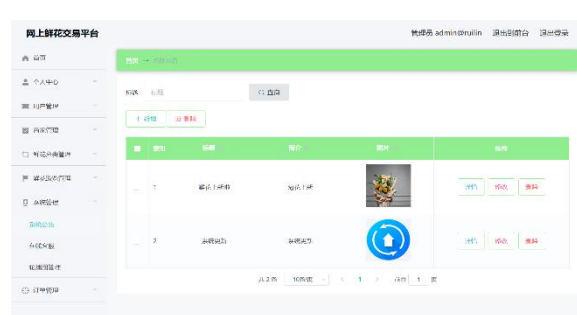
管理员管理在线客服



鲜花分类管理



鲜花浏览界面后台管理



发布系统公告

第 5 章 系统测试

进行系统测试是 Java Web 项目开发过程中至关重要的一步。

系统测试可以验证项目的功能是否按照需求规格说明书中所定义的要求进行工作。通过执行各种测试用例，可以确保项目的各个功能模块正常运行，并且可以检测到潜在的错误和缺陷。此外，系统测试还可以评估项目的性能，并确保在负载较大时系统仍然具有良好的响应时间和稳定性。

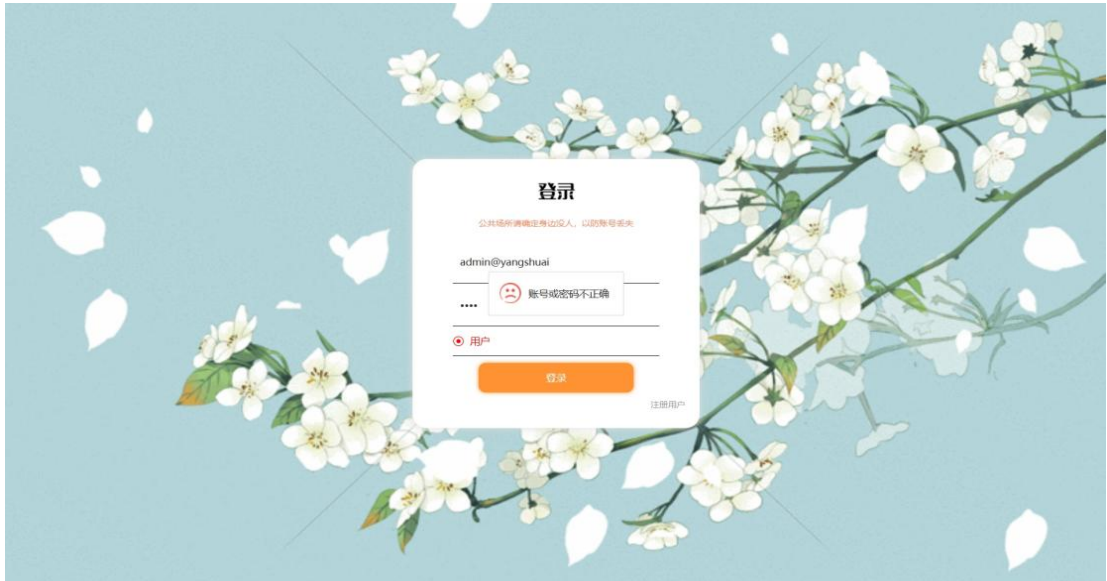
系统测试可以帮助发现潜在的问题和错误，以确保项目在正式投入使用前具备稳定和可靠的运行环境。通过模拟实际使用场景，可以发现和解决可能导致系统崩溃、数据丢失或其他故障的问题，从而提高系统的稳定性和可靠性。

系统测试还可以评估项目的安全性，包括对用户数据的保护和防止未经授权的访问。通过进行安全测试，可以发现潜在的漏洞和安全风险，并采取相应的措施进行修复和加固，以确保系统的安全性。

系统测试是项目交付前的最后一道质量保证关口。通过进行系统测试，可以确保项目的质量达到预期标准，并满足用户的需求和期望。这有助于避免后期出现高成本的修复和维护工作，并提高项目的用户满意度。

5.1 系统功能测试

5.1.1 用户登录界面测试



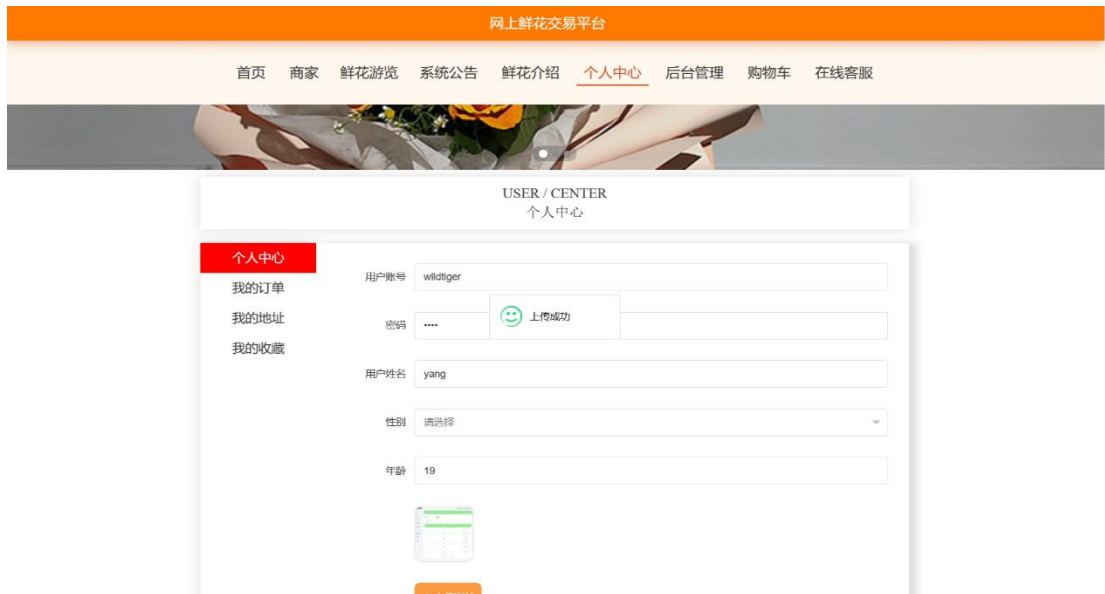
当选则不同的用户群体进入登录界面，会根据数据库中存有的数据来判断用户名及密码是否正确，以上所说的条件中但凡有一项不符合都会出现“账号或密码不正确”的提示框，同时若登录前台用户的身份为“商家”或“管理员”也会出现该提示框。

5.1.2 后台登陆界面测试



操作方法及原理同用户登录界面

5.1.3 用户个人中心上传图片测试



在用户个人界面上传图片时会反馈是否上传成功（以上是上传成功画面）

5.2 系统测试结论

经过较为完善的系统测试，验证鲜花销售系统的代码具有完整性，健壮性。

系统在功能、性能 and 安全性方面均达到了预期要求。代码质量高，符合编程规范，无明显逻辑错误或安全漏洞。测试过程中，系统运行稳定，响应速度快，用户体验良好。综上所述，该代码系统是一套成熟可靠的解决方案，值得信赖。

参考文献

- [1]郭煦.Java Web 程序设计与项目案例[M].北京：清华大学出版社，2023;2-1.
- [2]杨子鑫,余小兰. 基于 SSM 的农产品销售系统的设计与实现[J]. 现代信息科技,2021,5(15):33-35+39.
- [3]庄丽君.Bootstrap 响应式 Web 设计应用研究——以玩具商城网站设计为例[J].现代信息科技,2023,7(22):96-99.
- [4]论 MVC 架构设计及其应用
<https://www.cnblogs.com/znjy/p/15642672.html>.
- [5]SpringMVC 深入理解
<https://www.cnblogs.com/verlen11/p/5521536.html>.
- [6]MyBatis-Plus 快速入门 <https://baomidou.com/pages/24112f/>
- [7]Vue.js Introduction <https://vuejs.org/guide/introduction.html>
- [8]Spring Guides <https://spring.io/guides/>

致谢

在完成这份设计报告的过程中，项目组得到了许多人的帮助和支持，在此谨以第一作者代表项目组向他们表示衷心的感谢。

首先，我要感谢我的导师陈振兴副教授以及尹传娟讲师，他们严谨的学术态度和深厚的专业知识给予了我很大的启发。

最后，我要感谢项目组的同学们，他们在代码设计的过程中和实验阶段给予了我许多的帮助和支持。