# 1 引言

## 研究的背景

随着信息技术的飞速发展和互联网的普及，人们获取信息、分享创意和进行交流的方式发生了深刻的变革。在室内设计领域，设计师和业主对于高效便捷的设计作品展示、灵感获取以及互动交流平台的需求日益增长。传统的室内设计流程往往依赖于线下的沟通、实体作品集或分散的网络信息，这可能导致信息传递效率不高、优秀设计作品的传播范围有限、以及设计师与潜在客户之间缺乏有效的连接渠道等问题。

在这样的背景下，开发一个集设计作品展示、管理、分享与交流功能于一体的在线平台，具有重要的现实意义和应用价值。一方面，它可以为室内设计师提供一个集中展示个人才华、管理设计作品集、并与同行及潜在客户进行互动的专业空间，有助于提升设计师的个人品牌和业务拓展能力。另一方面，对于广大有装修需求的用户或设计爱好者而言，该平台可以作为一个丰富的灵感库，方便他们浏览多样的设计风格、学习设计知识、收藏喜爱的作品，并直接与设计师进行沟通，从而更有效地将设计理念转化为现实。

因此，本“基于SpringBoot的室内设计管理系统”旨在通过现代Web技术，构建一个功能完善、用户体验良好、安全可靠的在线平台，以满足室内设计领域多方用户的核心需求，促进设计资源的共享与创新，提升行业的信息化水平。

## 研究的现状

在线设计分享与管理平台在国内外均发展迅速。

国外以Pinterest、Houzz、Behance为代表，它们凭借海量用户与丰富资源，提供强大的图片展示、作品集管理、社交互动及多维度搜索功能。Houzz更整合了设计、装修与电商。这些平台常采用先进技术架构，注重用户体验与个性化推荐。学术界亦关注用户行为分析、图像识别等领域。

国内市场也涌现出“好好住”、“酷家乐”等优秀平台，它们借鉴国际经验并结合本土需求进行创新，如提供3D云设计、本地化服务对接等。移动端体验和社交功能得到强化，技术上广泛应用Java、Vue.js等主流Web技术，并探索AI、VR等新兴技术。国内科研机构也为相关系统开发提供了理论支持。

尽管如此，针对特定细分市场或特定用户群体的定制化设计管理平台仍有发展空间。许多现有平台可能功能过于庞杂，或者在某些特定需求（如小型工作室的作品管理、特定设计风格的社区构建）上未能完全满足。因此，开发一个功能聚焦、体验优化、技术架构现代化的室内设计管理系统，仍然具有研究和实践价值。

## 论文研究的主要内容

本论文主要围绕“基于SpringBoot的室内设计管理系统”的设计与实现展开研究。具体研究内容包括：

1. 系统需求分析: 深入分析系统的目标用户（普通用户、设计师、管理员）及其核心需求，明确系统的功能需求（如用户认证、素材管理、分类浏览、评论收藏、后台管理等）和非功能需求（如性能、安全、易用性等）
2. 系统详细设计与实现：针对各个核心功能模块，进行详细的功能设计和接口设计。具体阐述用户认证流程、素材上传与展示逻辑、评论与收藏功能的实现、后台管理功能的实现等关键环节，并展示部分核心代码。
3. 系统测试：设计测试用例，验证系统功能的正确性、稳定性和易用性，并对测试结果进行分析。

## 论文的组织结构

本论文一共分为五个章节，具体的内容组织如下：

1 引言

本章主要介绍本课题的研究背景与意义，分析国内外相关领域的研究现状，明确本文研究的主要内容和论文的整体组织结构。

2 系统可行性分析与系统关键技术

本章主要对系统的技术可行性、经济可行性和操作可行性进行分析。详细介绍系统开发所选用的后端Spring Boot技术栈、前端Vue.js技术栈、MySQL数据库以及其他相关关键技术。

3 系统需求分析

本章主要是对对系统的用户角色、核心用例进行分析，详细阐述系统的功能需求和非功能需求，为后续的系统设计提供依据。

4 系统的设计与实现

本章主要阐述系统的整体架构设计与实现、主要功能模块的划分，并进行数据库的概念设计和详细的表结构设计。

5 系统测试

本章介绍系统测试的方法和环境，设计并执行测试用例，对系统的主要功能进行测试，并对测试结果进行分析和总结。

# 2 系统可行性分析与系统关键技术

## 2.1 系统可行性分析

本“基于SpringBoot的室内设计管理系统”旨在构建一个高效的设计作品分享与管理平台。其可行性主要从技术和经济两方面评估。

技术可行性方面，系统后端采用Spring Boot、Spring MVC、Spring Data JPA及JWT，前端选用Vue.js、Vite、Pinia和Vue Router，数据库为MySQL。这些均为业界主流成熟技术，辅以标准开发工具与丰富的技术资源，确保了技术实现的可行性与低风险。

经济可行性方面，系统依托Java、Spring Boot、Vue.js、MySQL等开源技术栈，显著降低了软件采购成本。开发与测试可在个人计算机完成，初期部署成本可控。主要成本为人力与时间，对于毕业设计或个人项目而言，经济上可接受，并具学习与实践价值。

## 2.2 系统关键技术

### 2.2.1 Spring Boot框架介绍

Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。在本系统中，Spring Boot 作为后端服务的核心框架，承担了：

1. 自动配置：根据项目依赖自动配置大部分常用组件，减少了XML配置。
2. 添加依赖：简化了Maven依赖管理，如用于Web开发`spring-boot-starter-web`，用于数据持久化`spring-boot-starter-data-jpa`，用于安全控制`spring-boot-starter-security`。
3. 内嵌服务器：内置Tomcat服务器，使得应用可以打包成可执行的JAR文件直接运行。
4. 项目骨架：提供了快速构建RESTful API、集成数据库、实现安全认证等功能的基础。

### 2.2.2 Spring MVC 框架介绍

Spring MVC是一个基于Java的实现了MVC设计模式的请求驱动类型的轻量级Web框架。它通过一套注解，使得Java类成为处理请求的控制器，方法成为处理请求的处理器。例图2-2-1为MVC模式，图2-2-2为MVC设计思路图

****

图2-2-1 MVC模式图



图2-2-2 MVC设计思路图

### 2.2.3 Spring Data JPA介绍

Spring Data JPA 是 Spring Data 项目中用于简化基于JPA（Java Persistence API）的数据访问层开发的一个模块。JPA本身是Java EE规范中用于对象关系映射（ORM）的标准。 在本系统中，Spring Data JPA用于：

1. 对象关系映射 （ORM）：将Java实体类（Entity）与数据库表进行映射，使得开发者可以用面向对象的方式作数据库。
2. Repository接口：通过定义继承自的接口（如JpaRepository UserRepository， MaterialRepository），自动获得了一系列标准的CRUD（创建、读取、更新、删除）作以及分页和排序功能，极大减少了DAO层的编码工作。
3. 自定义查询：支持通过方法名约定或使用注解`@Query`编写JPQL或原生SQL来实现复杂的自定义查询。
4. 事务管理：与Spring的事务管理无缝集成，通过注解`@Transactional`方便地管理数据库事务。

### 2.2.4 Vue.js 前端框架介绍

Vue.js 是一套用于构建用户界面的渐进式JavaScript框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。 在本系统中，Vue.js用于：

1. 构建用户界面（UI）：通过其声明式渲染和组件系统，高效地构建交互式的用户界面。
2. 组件化开发: 将前端页面拆分为可复用的组件，提高了代码的组织性和可维护性。
3. 数据绑定与响应式更新: 实现了数据到视图的双向绑定或单向数据流，当数据变化时视图自动更新。

### 2.2.5 MySql数据库

MySQL 是一个广泛使用的开源关系型数据库管理系统（RDBMS），以其高性能、高可靠性和易用性而闻名。 在本系统中，MySQL作为主要的持久化存储方案：

* 1. 存储所有核心业务数据，包括用户信息、素材信息、分类、评论、收藏记录等。
  2. 通过Spring Data JPA进行数据作，利用其事务支持保证数据一致性。

### 2.2.6 其他技术

1. Axios： 一个基于Promise的HTTP客户端，用于前端发起对后端API的异步请求。
2. Lombok： 一个Java库，通过注解自动生成getter/setter、构造函数等样板代码，简化JavaBean的开发。
3. RESTful API 设计： 系统前后端交互采用RESTful风格设计API接口，使得接口语义清晰、易于理解和使用。

## 2.3 本章小结

本章主要对系统所涉及到的关键技术包括SpringBoot，MVC框架，SpringDataJPA模块，VUE框架，MySql数据库等做了简单的介绍，这些关键技术的选用和整合，为“基于SpringBoot的室内设计管理系统”的成功实现提供了坚实的技术保障。

# 3 系统需求分析

## 3.1 系统用例图

根据系统的需求分析，决定采用B/S模式开发系统，系统用例图如图3-1所示：



图3-1 系统用例图

## 3.2 系统功能需求

基于对用户角色和使用场景的分析，本系统的主要功能需求如下：

1. 用户认证与个人中心管理：提供用户注册和登录功能。用户登录后可以查看和编辑个人基本资料（如头像、昵称、简介），并修改登录密码。用户还可以方便地查看自己上传的素材列表和收藏的素材列表。
2. 素材展示与交互管理：系统应能公开展示所有已审核通过的装修素材，支持分页浏览。用户可以根据分类、最新发布、热门程度（如按浏览量或收藏数排序）、素材名称等方式进行筛选和排序。提供关键词搜索功能，方便用户快速定位感兴趣的素材。素材详情页应展示素材的完整图片、详细描述、所属分类、标签、上传者信息、当前的浏览量和收藏数，以及用户评论列表。认证用户可以对素材进行评论、回复他人评论，以及收藏或取消收藏自己感兴趣的素材。每次查看素材详情时，系统应自动增加该素材的浏览量。
3. 素材上传与设计师管理：能够上传新的装修素材。上传时需提供素材标题、详细描述、至少一张主图片，并选择素材所属的分类、填写相关标签和选择许可证类型。新上传的素材默认为“待审核”状态，需要管理员审核后才能公开显示。设计师应能管理自己已上传的素材，包括编辑素材信息和删除自己的作品（具体删除权限根据业务规则设定）。
4. 分类浏览：系统公开展示所有装修材料分类，用户可以点击分类查看该分类下的所有素材。
5. 后台管理功能（管理员）：

* 用户管理：管理员可以查看系统内所有用户的列表，支持分页和按条件（如用户名、邮箱、角色、状态）搜索。管理员有权修改用户的基本信息、角色（如将普通用户提升为设计师或管理员）以及账户状态（如启用或禁用账户）。
* 素材管理：管理员可以查看平台上的所有素材列表，包括各种状态（待审核、已批准、已拒绝）的素材，并支持分页和多条件筛选。核心功能是素材审核，管理员可以批准待审核的素材使其公开可见，或拒绝不符合要求的素材并填写拒绝理由。管理员还有权限编辑或删除平台上的任何素材。

## 3.3 系统性能需求

本系统作为用户获取设计灵感和设计师展示作品的平台，其运行的稳定性、操作的便捷性和数据的安全性至关重要。

* 1. 较高的稳定性与高效性：系统作为用户日常访问和设计师工作依赖的平台，必须保证长时间稳定运行，最大限度地减少故障时间。如果系统频繁出现故障或响应缓慢，将严重影响用户体验和平台的信誉。因此，系统设计应充分考虑后端服务的健壮性和前端交互的流畅性，确保在高并发访问（根据预期设定，例如支持50-100个并发用户）下依然能提供快速的响应（例如，主要API接口平均响应时间低于500ms，页面加载时间在3秒内）。
  2. 易操作性：系统的用户界面（UI）和用户体验（UX）设计应力求简洁直观、符合用户操作习惯。无论是普通用户浏览素材，设计师上传作品，还是管理员进行后台管理，都应能通过清晰的导航和简单的操作步骤快速完成任务。减少用户的学习成本，提高操作效率，是系统易用性的核心要求。
  3. 较高的安全性：
     + 用户数据保护：用户的个人信息（特别是密码）必须进行加密存储，防止数据泄露。
     + 访问控制：系统需建立完善的权限控制机制，确保不同角色的用户只能访问其被授权的功能和数据。例如，普通用户不能访问后台管理接口，设计师只能管理自己的素材。

## 3.4 本章小结

本章详细分析了“基于SpringBoot的室内设计管理系统”的各项需求。首先通过系统用例图明确了不同用户角色的核心交互场景和系统边界。其次，详细阐述了系统需要实现的各项功能需求，覆盖了用户认证、个人中心、素材管理与交互、分类管理以及后台管理等关键模块。最后，对系统的性能、易用性和安全性等非功能需求提出了明确的要求。这些需求分析将作为后续系统设计与开发工作的重要依据和衡量标准。

# 4 系统的设计与实现

## 4.1 系统功能分布图

我的本系统一共有用户认证与个人中心、素材中心、后台管理系统三个功能，具体分布如图4-1所示：

图示

AI 生成的内容可能不正确。

图4-1 系统功能分布图

## 4.2 数据库设计

本文档描述了“装修分享平台”后端应用的数据库设计。该数据库旨在:存储和管理用户信息、装修材料信息、分类、评论以及用户收藏核心数。据根据一共建立了~~~~五张表。数据库采用关系型数据库模型。

这些数据表均可以反映在图4-2的数据关系图中，如图4-2所示：



图4-2 数据库关系图

以上每一个表表示在数据库中的表，这些表的介绍如下：

表4-1是`Users`一张用户表

描述：存储平台注册用户的信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 主键 | 外键 | 允许空 | 描述 |
| 1 | id | BIGINT | 是 |  | 否 | 用户唯一标识符 |
| 2 | username | VARCHAR(50) |  |  | 否 | 用户名 (唯一) |
| 3 | email | VARCHAR(100) |  |  | 否 | 用户邮箱 (唯一) |
| 4 | password | VARCHAR(255) |  |  | 否 | 用户密码 (加密存储) |
| 5 | avatar | VARCHAR(255) |  |  | 是 | 用户头像 URL |
| 6 | bio | TEXT |  |  | 是 | 用户个人简介 |
| 7 | role | ENUM('USER','ADMIN') |  |  | 是 | 用户角色 (用户/管理员) |
| 8 | status | ENUM('ACTIVE','BLOCKED') |  |  | 是 | 用户账户状态 (活跃/禁用) |
| 9 | created\_at | TIMESTAMP |  |  | 是 | 账户创建时间 |
| 10 | updated\_at | TIMESTAMP |  |  | 是 | 账户最后更新时间 |

表4-1 Users（用户表）

表4-2是` categories `一张装修材料分类表

描述：用来存储装修材料的分类信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 主键 | 外键 | 允许空 | 描述 |
| 1 | id | BIGINT | 是 |  | 否 | 分类ID |
| 2 | name | VARCHAR(50) |  |  | 否 | 分类名称 |
| 3 | description | TEXT |  |  | 是 | 分类描述 |
| 4 | icon\_url | VARCHAR(255) |  |  | 是 | 分类图标 URL |
| 5 | color | VARCHAR(20) |  |  | 是 | 分类代表颜色 |
| 6 | sort | INT |  |  | 是 | 排序字段 |
| 7 | created\_at | TIMESTAMP |  |  | 是 | 创建时间 |
| 8 | updated\_at | TIMESTAMP |  |  | 是 | 更新时间 |

表4-2 categories 表（装修材料分类表表）

表4-3是`materials` 装修素材表

描述：用来存储用户分享的装修素材信息。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 主键 | 外键 | 允许空 | 描述 |
| 1 | id | BIGINT | 是 |  | 否 | 素材ID |
| 2 | title | VARCHAR(100) |  |  | 否 | 素材标题 |
| 3 | description | TEXT |  |  | 是 | 素材详细 |
| 4 | image\_url | VARCHAR(255) |  |  | 否 | 素材主图片 URL |
| 5 | thumb\_url | VARCHAR(255) |  |  | 是 | 素材缩略图 URL |
| 6 | category\_id | BIGINT |  | 是 | 是 | 所属分类ID (参照 categories.id) |
| 7 | user\_id | BIGINT |  | 是 | 是 | 上传用户ID (参照 users.id) |
| 8 | views | INT |  |  | 是 | 浏览次数 |
| 9 | favorites | INT |  |  | 是 | 收藏次数 |
| 10 | tags | JSON |  |  | 是 | 素材标签 |
| 11 | license | VARCHAR(20) |  |  | 是 | 许可证信息 |
| 12 | status | VARCHAR(255) |  |  | 否 | 素材状态 |
| 13 | reject\_reason | TEXT |  |  | 是 | 拒绝理由 |
| 14 | created\_at | TIMESTAMP |  |  | 是 | 创建时间 |
| 15 | updated\_at | TIMESTAMP |  |  | 是 | 最后更新时间 |

表4-3 materials 表 (装修素材表)

表4-4 favorites 表 (收藏记录表)

描述: 记录用户收藏装修素材的行为，实现用户与素材之间的多对多关系。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 主键 | 外键 | 允许空 | 描述 |
| 1 | id | BIGINT | 是 |  | 否 | 收藏记录唯一标识符 |
| 2 | user\_id | BIGINT |  | 是 | 否 | 收藏用户ID (users.id) |
| 3 | material\_id | BIGINT |  | 是 | 否 | 被收藏素材ID (materials.id) |
| 4 | created\_at | TIMESTAMP |  |  | 是 | 收藏时间 |

表4-3 favorites 表 (装修素材表)

表4-5 是comments 表 (评论表)

描述: 存储用户对装修素材的评论。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 主键 | 外键 | 允许空 | 描述 |
| 1 | id | BIGINT | 是 |  | 否 | 评论ID |
| 2 | content | TEXT |  |  | 否 | 评论内容 |
| 3 | user\_id | BIGINT |  | 是 | 否 | 发表用户ID (users.id) |
| 4 | material\_id | BIGINT |  | 是 | 否 | 所属素材ID (materials.id) |
| 5 | parent\_id | BIGINT |  | 是 | 是 | 父评论ID (comments.id, 用于回复) |
| 6 | created\_at | TIMESTAMP |  |  | 是 | 创建时间 |
| 7 | updated\_at | TIMESTAMP |  |  | 是 | 更新时间 |

表4-5 comments 表 (评论表)

# 5 系统模块功能的实现