**基于python房屋出租系统设计与实现**

姓 名：

学 号：

指导教师：

**xxxx年xx月**

**摘要**

本文介绍了基于Django和Vue.js的房屋出租系统的设计与实现。该系统旨在提供一个方便用户发布和搜索房屋出租信息的平台。系统采用了前后端分离的架构，前端使用Vue.js框架进行开发，后端使用Django框架。在系统设计方面，我们首先设计了数据库模型，包括房屋信息、用户信息和订单信息等。通过Django的ORM功能，我们可以轻松地与数据库进行交互，并进行数据的增删改查操作。为了提高系统的性能，我们还使用了缓存机制，将经常访问的数据进行缓存，减少数据库的访问次数。在前端开发方面，我们使用Vue.js框架搭建了一个响应式的用户界面。通过Vue.js的组件化开发方式，我们将页面拆分成多个可复用的组件，提高了代码的可维护性和重用性。同时，我们使用Vue Router进行页面路由管理，实现了单页面应用的效果。在系统功能方面，用户可以注册和登录账号，发布房屋出租信息，并进行搜索和筛选。我们使用Django的用户认证和授权系统，保证了用户信息的安全性。用户可以通过地理位置、价格和房屋类型等条件进行搜索，系统会返回符合条件的房屋列表。用户还可以查看房屋详情，预订房屋并进行订单管理。在实现过程中，我们注重系统的性能和安全性。通过使用缓存机制和优化数据库查询，提高了系统的响应速度。同时，我们对用户输入进行了合法性检查和数据验证，防止了潜在的安全漏洞。总结来说，基于Django和Vue.js的房屋出租系统设计与实现了一个功能完善、性能优越的平台，方便用户发布和搜索房屋出租信息。该系统的前后端分离架构和使用的技术栈使得系统具有良好的可维护性和扩展性。未来，我们可以进一步完善系统功能，如添加在线支付和评价系统，提升用户体验。。

**关键词**：Django; Vue; 系统设计;

**Abstract**

This article presents the design and implementation of a house rental system based on Django and Vue.js. The system aims to provide a platform for users to conveniently publish and search house rental information. The system adopts a frontend-backend separation architecture, with Vue.js framework for frontend development and Django framework for backend.In terms of system design, we first design the database models, including house information, user information, and order information, among others. With the ORM feature of Django, we can easily interact with the database and perform CRUD operations. To enhance system performance, we also utilize caching mechanism to cache frequently accessed data, reducing the number of database queries.For frontend development, we build a responsive user interface using Vue.js framework. Through the component-based development approach of Vue.js, we divide the pages into reusable components, improving code maintainability and reusability. Additionally, we employ Vue Router for page routing management, achieving a single-page application effect.In terms of system functionalities, users can register and login to their accounts, publish house rental information, and perform searches and filters. We utilize Django's user authentication and authorization system to ensure the security of user information. Users can search for houses based on criteria such as location, price, and house type, and the system will return a list of houses that meet the criteria. Users can also view house details, make reservations, and manage their orders.During the implementation process, we prioritize system performance and security. By utilizing caching mechanism and optimizing database queries, we enhance the system's responsiveness. Additionally, we perform validity checks and data validation on user inputs to prevent potential security vulnerabilities.In conclusion, the design and implementation of the house rental system based on Django and Vue.js deliver a feature-rich and high-performance platform for users to publish and search house rental information. The frontend-backend separation architecture and the chosen technology stack ensure good maintainability and scalability of the system. In the future, further improvements can be made, such as adding online payment and review systems, to enhance the user experience.

目 录

[1 绪论 5](#_Toc154651691)

[1.1 研究背景 5](#_Toc154651692)

[1.2 研究现状 6](#_Toc154651693)

[1.3 研究目标 7](#_Toc154651694)

[2 相关技术介绍 8](#_Toc154651695)

[2.1 Python语言 8](#_Toc154651696)

[2.2 Django框架技术 9](#_Toc154651697)

[2.3 Vue前端技术 11](#_Toc154651698)

[2.3.1 Vue框架 11](#_Toc154651699)

[2.3.2 Ant-Design-Vue组件 12](#_Toc154651700)

[2.4 Nginx负载均衡技术 13](#_Toc154651701)

[2.4.1 Nginx应用场景 13](#_Toc154651702)

[2.4.2 Nginx优势 13](#_Toc154651703)

[2.5 MySQL数据库 14](#_Toc154651704)

[3 需求分析 14](#_Toc154651705)

[3.1 系统功能需求分析 14](#_Toc154651706)

[3.1.1 业务需求分析 14](#_Toc154651707)

[3.1.2 数据需求分析 15](#_Toc154651708)

[3.2 系统非功能性需求分析 16](#_Toc154651709)

[3.2.1 系统处理能力需求 16](#_Toc154651710)

[3.2.2 可靠性需求 16](#_Toc154651711)

[3.2.3 可用性需求 17](#_Toc154651712)

[3.2.4 维修性需求 17](#_Toc154651713)

[3.2.5 环境适配需求 17](#_Toc154651714)

[4 系统设计 18](#_Toc154651715)

[4.1 设计原则 18](#_Toc154651716)

[4.1.1 阶段开发原则 18](#_Toc154651717)

[4.1.2 易用性原则 18](#_Toc154651718)

[4.1.3 业务完整性原则 18](#_Toc154651719)

[4.1.4 业务规范化原则 18](#_Toc154651720)

[4.1.5 可扩展性原则 18](#_Toc154651721)

[4.2 总体架构 18](#_Toc154651722)

[4.3 功能设计 19](#_Toc154651723)

[4.4 数据库设计 20](#_Toc154651724)

[4.4.1 设计规则 20](#_Toc154651725)

[4.4.2 表结构设计 20](#_Toc154651726)

[5 系统实现 23](#_Toc154651727)

[5.1 前台功能模块实现 23](#_Toc154651728)

[5.1.1 注册登录模块 23](#_Toc154651729)

[5.1.2 首页模块 25](#_Toc154651730)

[5.1.3 详情页模块 26](#_Toc154651731)

[5.1.4 搜索模块 27](#_Toc154651732)

[5.1.5 用户中心模块 27](#_Toc154651733)

[5.2 后台管理模块实现 28](#_Toc154651734)

[5.2.1 用户管理 28](#_Toc154651735)

[5.2.2 分类管理 29](#_Toc154651736)

[5.2.3 标签管理 29](#_Toc154651737)

[5.2.4 房屋管理 29](#_Toc154651738)

[5.2.5 日志管理 29](#_Toc154651739)

[5.2.6 运营管理 29](#_Toc154651740)

[5.2.7 系统信息 29](#_Toc154651741)

[5.2.8 总览模块 30](#_Toc154651742)

[6 系统测试 30](#_Toc154651743)

[6.1 测试策略 30](#_Toc154651744)

[6.2 测试环境 33](#_Toc154651745)

[6.3 测试用例及结果 33](#_Toc154651746)

[6.3.1 后台功能测试用例 33](#_Toc154651747)

[6.3.2 前台功能测试用例 34](#_Toc154651748)

[6.4 测试总结 35](#_Toc154651749)

[7 总结与展望 36](#_Toc154651750)

[参考文献 37](#_Toc154651751)

# 绪论

## 研究背景

随着城市化进程的加速，人口流动性的增加，房屋租赁市场日益繁荣。特别是在一线城市，大量流动人口使得租房成为一种刚需。然而，传统的房屋租赁方式存在诸多问题，如信息不透明、租后管理困难、租客与房东权益难以保障等。这不仅给租房者带来不便，也给房东带来了诸多困扰。在此背景下，房屋出租系统的研究与应用显得尤为重要。

首先，房屋出租系统的研究是信息化时代的必然产物。随着互联网技术的飞速发展，人们的生活方式正在发生深刻变化。租房者期望更方便地查找和比较房源，房东希望更高效地管理和出租房屋。房屋出租系统利用先进的信息技术，能够实现房源信息的快速发布、查询和管理，极大地满足了市场需求。

其次，房屋出租系统的研究也是社会经济发展的必然要求。随着城市化进程的加速，人口流动性增强，租房市场日益庞大。然而，传统的租房方式存在诸多问题，如中介费用高昂、信息不透明、租后管理困难等。这些问题不仅给租房者带来不便，也给房东带来了诸多困扰。房屋出租系统能够解决这些问题，保障租客和房东的权益，推动租房市场的健康发展。

此外，房屋出租系统的研究还是推进城市化的重要举措。城市化不仅仅是人口的聚集，更是人们生活方式的转变。房屋出租系统能够提供更加便捷、高效的租房服务，帮助流动人口更快地融入城市生活，进一步推进城市化进程。

因此，房屋出租系统的研究背景主要体现在三个方面：信息化时代的必然产物、社会经济发展的必然要求以及推进城市化的重要举措。随着技术的不断进步和市场需求的不断变化，房屋出租系统将不断完善和发展，为人们提供更加便捷、高效的租房服务。

在未来，房屋出租系统有望实现更多功能，如智能匹配、信用评价、在线签约等，以满足市场的多样化需求。同时，随着数据安全和隐私保护意识的提高，房屋出租系统也需要加强数据加密和隐私保护措施，确保用户信息安全。总之，房屋出租系统的研究和发展是时代发展的必然趋势，对于推动租房市场的健康发展、提升人们的生活品质和推进城市化进程都具有重要意义。。

## 研究现状

随着互联网技术的快速发展，房屋出租系统已成为一个热门的研究领域。目前，国内外对于房屋出租系统的研究主要集中在以下几个方面：

一、房源信息发布与检索

在房源信息发布与检索方面，研究者们致力于提高系统的信息准确性和检索效率。例如，利用自然语言处理技术对房源描述进行文本分析，提取关键信息，提高信息检索的准确率。同时，采用数据挖掘和机器学习算法，对用户查询行为进行分析，实现个性化推荐，提高房源匹配度。

二、用户信用评价体系

在用户信用评价体系方面，研究者们关注如何建立有效的信用评价模型，保障租客和房东的权益。通过采集用户的交易记录、评价历史等数据，利用机器学习算法对用户信用进行评估。同时，引入区块链技术，实现信用数据的可追溯和不可篡改，提高信用评价的公正性和可信度。

三、智能合约与在线签约

在智能合约与在线签约方面，研究者们致力于实现自动化、智能化的合同管理。通过智能合约技术，将租房合同条款以代码形式写入区块链，实现自动执行和不可篡改。租客和房东可以在线上完成合同签署，简化签约流程，提高签约效率。同时，利用区块链的透明性和可追溯性，保障合同的公正性和安全性。

四、社区管理与服务

在社区管理与服务方面，研究者们关注如何构建一个健康、和谐的租房社区。通过引入社交网络技术，为租客和房东提供一个交流平台，增强社区的互动性和凝聚力。同时，提供物业报修、费用缴纳等在线服务，提高社区管理效率和服务质量。此外，利用大数据分析技术，对社区活动进行监测和分析，及时发现并解决潜在问题。

五、数据安全与隐私保护

在数据安全与隐私保护方面，研究者们致力于保障用户信息安全和隐私权益。通过采用加密技术和访问控制机制，对用户数据进行严格保护。同时，规范数据处理流程，确保用户数据的合法使用和存储。此外，加强对第三方应用的监管和认证，防止数据泄露和滥用。

综上所述，当前房屋出租系统研究现状呈现出多元化、智能化的发展趋势。在技术层面，不断引入新的信息技术和创新理念；在应用层面，注重解决实际问题和满足市场需求；在伦理层面，强调数据安全和隐私保护。未来随着技术的进步和社会需求的演变，房屋出租系统将不断演进和完善。

## 研究目标

房屋出租系统作为现代租赁市场的重要工具，其研究目标主要集中在以下几个方面：

一、提高租赁效率

房屋出租系统的首要目标是提高租赁的效率和便利性。通过技术手段，如利用大数据、人工智能等技术，对房源和租客进行精准匹配，缩短租赁周期，降低交易成本。同时，提供线上看房、选房、签约等服务，使房东和租客可以更加便捷地进行交易，提升租赁体验。

二、保障权益

保障房东和租客的权益是房屋出租系统的重要研究目标。通过建立完善的信用评价体系，对房东和租客的信用进行评估，减少不良交易的风险。同时，通过智能合约等技术，确保租赁合同的公正、合法和有效，避免合同纠纷。此外，系统还应提供纠纷调解、投诉处理等服务，维护租赁市场的公平和秩序。

三、优化资源配置

房屋出租系统的研究目标还包括优化资源配置，提高房屋使用效率。通过对租赁市场的数据分析，预测未来的租赁需求和供应变化，引导房东合理定价和调整出租策略。同时，利用大数据技术对房源进行智能分析和评估，为房东提供科学的房屋管理和维护建议，延长房屋使用寿命，提高房屋价值。

四、促进社区和谐

房屋出租系统不仅是一个交易平台，更是一个社区平台。通过社区管理功能，促进租客之间的交流和互动，增强社区凝聚力。同时，提供物业服务、社区活动等在线服务，提升租客的居住体验和归属感。此外，系统还应关注社区安全问题，加强对租房安全信息的收集和发布，提高社区安全防范意识。

五、推进技术创新

房屋出租系统是一个技术密集型的项目，需要不断推进技术创新和研究。在数据安全和隐私保护方面，加强数据加密和访问控制技术的研究和应用，确保用户信息安全。同时，探索新的技术应用，如区块链技术、物联网技术等，提升系统的智能化和自动化水平。此外，还需要关注新技术的发展趋势和前沿动态，为系统的持续升级和优化提供技术支持。

综上所述，房屋出租系统研究目标涵盖了提高租赁效率、保障权益、优化资源配置、促进社区和谐以及推进技术创新等多个方面。通过实现这些目标，房屋出租系统有望为现代租赁市场带来更大的便利和发展空间。

# 相关技术介绍

## Python语言

Python是一种高级编程语言，由Guido van Rossum于1991年创建。它以简洁、易读和可维护的代码而闻名，被广泛用于各种领域，包括Web开发、科学计算、人工智能和数据分析等。

Python的设计哲学强调代码的可读性和简洁性，这使得它成为初学者和专业开发人员的首选语言之一。它使用明确的语法和简单的语句结构，使得编写代码变得容易。Python采用了缩进方式来表示代码块，而不是使用大括号或关键字，这使得代码的结构更加清晰。

Python是一种解释型语言，这意味着在运行代码之前不需要进行显式的编译。开发人员可以直接编写和执行代码，这加快了开发过程的速度。Python还具有交互式编程的特性，可以在命令行中逐行执行代码，方便测试和调试。

Python具有丰富的标准库和第三方库，提供了大量的功能和工具，可以满足各种需求。标准库包含了各种模块，用于处理文件、网络通信、数据库连接、图形界面开发等。此外，Python还有众多的第三方库，例如NumPy、Pandas、Matplotlib和TensorFlow等，用于数据分析、科学计算和机器学习等领域。

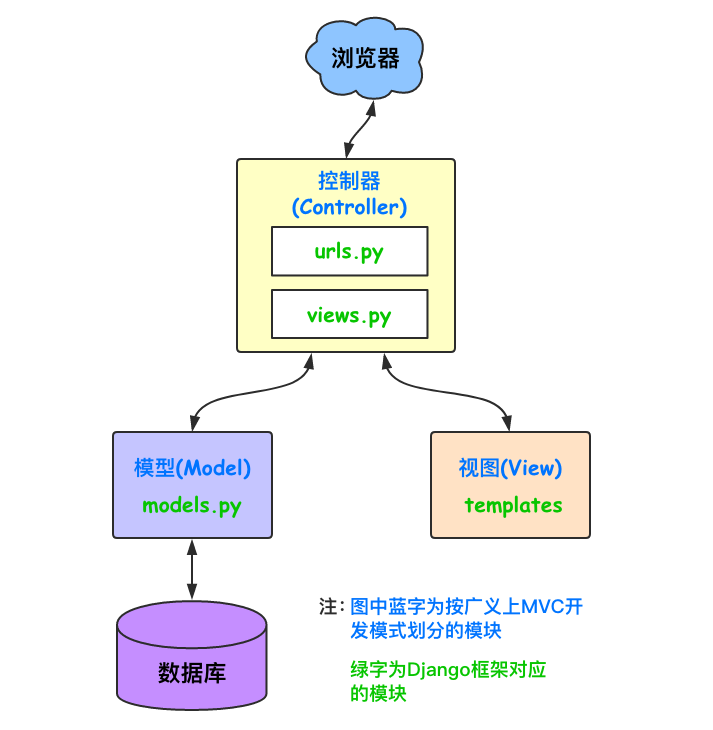
Python支持面向对象编程（OOP），这使得代码的组织和复用更加灵活和简单。开发人员可以定义类和对象，并使用继承、多态等概念来构建复杂的程序结构。

Python具有广泛的应用领域。在Web开发方面，Python的框架如Django和Flask提供了强大的工具和功能，用于构建高效和可扩展的Web应用程序。在科学计算和数据分析领域，Python的库如NumPy和Pandas提供了丰富的功能和高效的数据处理能力。在人工智能和机器学习领域，Python的库如TensorFlow和PyTorch成为了主流工具，用于构建和训练各种深度学习模型。

总之，Python是一种功能强大、易学易用的编程语言，适用于各种应用场景。它的简洁语法、丰富的库和活跃的开发社区使得它成为了众多开发者的首选语言。无论是初学者还是专业开发人员，都可以通过Python实现自己的创意和解决各种问题。

## Django框架技术

Django是一个流行的Python Web应用程序开发框架，它提供了一套强大的工具和功能，用于快速构建高效和可扩展的Web应用程序。Django的设计目标是使开发人员能够专注于应用程序的逻辑而不是底层的技术细节。



Django框架采用了MVC（Model-View-Controller）的架构模式，但它更倾向于使用MTV（Model-Template-View）的模式。这种模式将应用程序的逻辑分为三个主要部分：模型（Model）、模板（Template）和视图（View）。

模型（Model）是与数据库交互的组件，它定义了数据的结构和行为。Django使用ORM（对象关系映射）来处理数据库操作，开发人员可以使用Python代码来定义模型类，每个类对应一个数据库表。ORM提供了简洁的API，使得数据库的操作变得简单和直观。

模板（Template）是用于呈现数据的组件，它定义了Web页面的结构和样式。Django的模板语言使用简单且易于理解的语法，开发人员可以在模板中使用变量、循环和条件语句等来动态生成页面内容。模板还支持继承和模板标签等功能，提供了灵活和可重用的页面设计方式。

视图（View）是处理用户请求的组件，它接收用户的请求并返回相应的响应。视图函数是Django中的核心概念，开发人员可以编写视图函数来处理不同的URL请求。视图函数可以访问模型的数据，并通过模板来呈现页面。Django还提供了一套强大的URL映射机制，可以将URL与视图函数进行关联，使得请求能够正确地路由到相应的视图函数。

除了MVT架构，Django还提供了许多其他功能和组件，以简化Web应用程序的开发过程。其中包括：

1. 表单处理：Django提供了表单处理的功能，开发人员可以轻松地创建和验证表单数据。这使得用户交互和数据提交变得简单和安全。

2. 用户认证和授权：Django提供了内置的用户认证和授权系统，开发人员可以轻松地管理用户的登录、注册和权限控制。

3. 后台管理：Django自带了一个功能强大的后台管理界面，开发人员可以通过简单的配置来管理应用程序的数据和功能。

4. 缓存机制：Django提供了缓存机制，可以缓存数据库查询结果、页面片段或其他计算结果，以提高应用程序的性能和响应速度。

5. 国际化和本地化：Django支持多语言和多时区的应用程序开发，开发人员可以轻松地将应用程序本地化为不同的语言和区域设置。

6. 安全性：Django具有内置的安全机制，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）和密码哈希存储等。

7. 第三方库和插件：Django拥有庞大的生态系统，有大量的第三方库和插件可供使用，可以扩展框架的功能和特性。

总之，Django是一个功能强大、灵活且易于使用的Web应用程序开发框架。它提供了丰富的工具和功能，使得开发人员能够快速构建高质量的Web应用程序。无论是小型网站还是大型企业级应用，Django都是一个理想的选择。

## Vue前端技术

### Vue框架

Vue是一套用于构建用户界面的渐进式框架。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。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

组件化开发即是对某些可以进行复用的功能进行封装的标准化工作。组件一般会内含他的内部UI元素、样式和JS逻辑代码，它可以很方便的在应用的任何地方进行快速的嵌入。组件内部可以使用其他组件来构成更复杂的组件。

组件化特点：

（1）组件是对逻辑的封装，不限于图形元素

即我们可以把if做成组件，按一个倒计时做成组件，把一段动画做成组件，把数据架构做成组件，而这些并不能成为控件，这些组件在不同的系统中可以做完美的嵌入。

（2）高内聚、低耦合

前端的组件化开发，可以很大程度上降低系统各个功能的耦合性，数据相互独立，并且提高了功能内部的聚合性。这对前端工程化及降低代码的维护来说，是有很大的好处的，内部结构密封，不与全局或其他组件产生影响，特别是针对逻辑复杂的功能能够进行拆分，更好排查问题。

（3）具备单个可移植性

组件可以提供很好的提升代码的可重用性（而非可复制性），如果有其他项目需要该组件可以直接引入使用，而不是拷贝代码，拷贝资源等。

（4）极大提高开发效率

组件划分后，组件的开发不受其他业务影响，可以多个组件并行开发，加快开发进度。在多人团队里，每个人只负责自己的业务模块，他对业务功能的增删改查，都只限定在自己的这个业务模块里，不会影响其他人的业务，他代码质量的好坏也只会影响到自己的业务模块；如果有新人的加入，可以直接分配组件进行开发，而非需要熟悉整个项目，可以从一个组件的开发使新进人员比较快速熟悉项目、了解到开发规范；对测试来说，也十分方便，大部分情况下，我们只需要着重测试修改过的业务组件即可，而不用老是进行全部回归测试。

（5）功能重用的基石

业务组件类似一个个积木一样，我们可以用积木搭建出不同的房子，同理我们也可以创建多个不同的APP。我们只需要维护好每个组件，需要用到该组件的功能时，一建引用集成就可以了。

### Ant-Design-Vue组件

Ant Design Vue是蚂蚁金服Ant Design官方推荐的Vue 版UI组件库，它其实是 Ant Design 的 Vue实现，组件的风格与 Ant Design 保持同步，组件的html结构和css样式也保持一致。 用下来发现它的确称得上为数不多的完整的 VUE 组件库与开发方案集成项目。Ant Design Vue 是使用Vue实现的遵循 Ant Design 设计规范的高质量UI组件库，用于开发和服务于企业级中后台产品。特性提炼自企业级中后台产品的交互语言和视觉风格。

开箱即用的高质量 Vue 组件。具有如下优势：

（1）共享Ant Design of React设计工具体系。

（2）支持环境现代浏览器和 IE9 及以上。

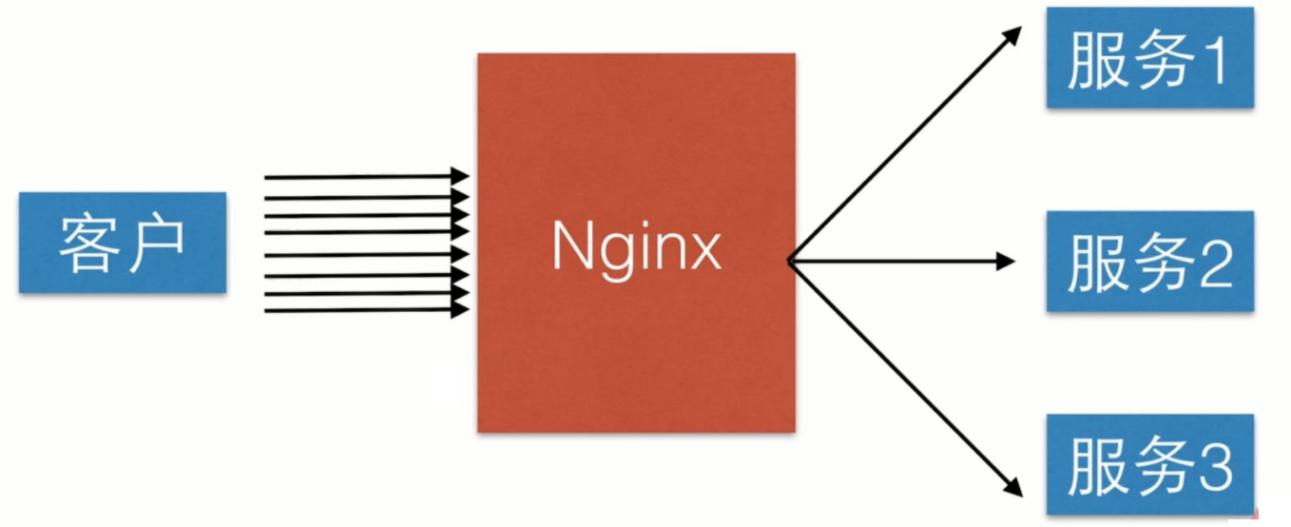
（3）支持服务端渲染。

众所周知，Ant Design 作为一门设计语言面世，经历过多年的迭代和积累，它对 UI 的设计思想已经成为一套事实标准，受到众多前端开发者及企业的追捧和喜爱，也是React开发者手中的神兵利器。Ant-design-vue 能够让 Vue 开发者也享受到 Ant Design 的优秀设计。

Ant-design-vue 是 Ant Design 的 Vue 实现，组件的风格与 Ant Design 保持同步，组件的 html 结构和 css 样式也保持一致，真正做到了样式 0 修改，组件 API 也尽量保持了一致。

## Nginx负载均衡技术

Nginx是一款自由的、开源的、高性能的HTTP服务器和反向代理服务器；同时也是一个IMAP、POP3、SMTP代理服务器；nginx可以作为一个HTTP服务器进行网站的发布处理，另外nginx可以作为反向代理进行负载均衡的实现。



Nginx负载均衡

### Nginx应用场景

**场景一：应用于高访问量的业务**

如果您的应用访问量很高，您可以通过配置监听规则将流量分发到不同的服务器上。

**场景二：横向扩张系统**

可以根据业务发展的需要，通过随时添加和移除服务器，来扩展应用系统的服务能力，适用于各种Web服务器和App服务器。

**场景三：消除单点故障**

当其中一部分服务器发生故障后，负载均衡会自动屏蔽故障的服务器，将请求分发给正常运行的服务器，保证应用系统仍能正常工作。

**场景四：同城容灾 （多可用区容灾）**

为了提供更加稳定可靠的负载均衡服务，当主可用区出现机房故障或不可用时，负载均衡仍然有能力在非常短的时间内切换到另外一个备可用区恢复服务能力；当主可用区恢复时，负载均衡同样会自动切换到主可用区提供服务，保证服务依然正常运行。

### Nginx优势

1. 解决并发压力，提高应用处理性能（增加吞吐量，加强网络处理能力）；
2. 提供故障转移，实现高可用；
3. 通过添加或减少服务器数量，提供网站伸缩性（扩展性）；
4. 安全防护：负载均衡设备上做一些过滤，黑白名单等处理。
5. Nginx支持页面热部署，不用重启服务器，前端升级更无缝。
6. Nginx很好的匹配了本项目中子系统多、服务器部署量大、并发量大、运行效率要求高等特点，因此选用Nginx承担系统部署中的负载均衡工作。

## MySQL数据库

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MySQL是一种关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型和大型网站的开发都选择 MySQL 作为网站数据库。

# 需求分析

## 系统功能需求分析

### 业务需求分析

房屋出租系统是一个涉及众多方面的综合性系统，需要全面深入地分析和了解各个业务模块的需求，才能实现一个符合用户期望的房屋出租系统。本文首先对房屋出租系统的业务需求进行详细的分析，以便更好地满足用户需求。

1. 房屋管理

房屋管理是房屋出租网站系统最基本和最主要的功能之一，包括房屋信息录入、修改、删除、推荐、分类等等。房屋信息的录入可以采用图形化界面或Excel表格录入等形式，方便管理员进行批量操作。在房屋信息的管理过程中，必须要支持房屋图片的上传和浏览，帮助用户更好地了解房屋。

2. 用户管理

用户管理是房屋出租网站系统与用户交互的重要方式，需要支持用户个人信息、浏览记录、等级特权、积分纪录等等。用户可以通过注册、登陆、修改个人资料等方式进行用户管理，同时房屋管理可以通过用户管理界面进行统计和管理。

3. 数据统计

数据统计是房屋出租网站系统的另一个重要功能，它可以提供企业管理者了解网站运营数据的综合分析和报告。数据统计主要包括关键指标的图表分析、用户行为分析、浏览习惯分析等等。通过对数据统计的分析，管理员可以更好地了解网站运营情况，制定更好的经营策略。

总之，房屋出租网站系统的业务需求非常多且复杂，本文只列举了关键部分。为了实现一个好用、易用、高性能、安全可靠的房屋出租网站系统，还需要深入研究和了解各个业务领域的需求，以便提供更好的解决方案。

### 数据需求分析

数据需求分析是一份系统设计的重要环节，它主要是明确系统应该支持怎样的数据关联操作，主要涉及需求调研、数据收集与整理、数据审视与分类等方面。在本文中将围绕房屋出租网站系统设计与实现来探讨数据需求分析的具体实现。

一、需求调研

首先，我们要对房屋出租网站的业务特点进行调研，明确它主要涉及到哪些数据关联。通过查阅房屋出租网站资料和同类型网站的特点，我们可以列出以下常见数据关联：

1.用户关联：包括用户的注册信息、个人中心信息、个人中心信息包括用户的基本信息、地址、记录、评论等。

2.房屋关联：包括房屋信息、分类信息、品牌信息、评价信息等。

这些数据关联是房屋出租网站的基本需求，所以在系统设计之前，必须要明确它们之间的关系和操作。

二、数据收集与整理

一旦明确了数据关联，我们就可以开始收集和整理数据了。首先，我们需要收集用户信息，包括用户的身份信息、注册信息、地址信息以及各类监控信息（例如：登录信息）等。

其次，我们需要收集房屋信息，包括房屋名称、库存、价格、类型的相关信息。

最后，我们需要收集房屋信息，包括房屋基本信息、用户信息及网站管理信息等。

在整理数据的过程中，我们需要理清数据之间的关联和操作方式，然后建立数据表以及设计合适的字段和外键等。

三、数据审视与分类

最后，我们需要对数据进行审视和分类。首先是唯一标识符的设计，比如用户ID、房屋ID、评论ID等。其次是房屋网站中的分类功能，在房屋相关数据中，分类功能是必不可少的。最后是对数据的分类，分别将不同类型的数据归类到不同的数据表中，确保数据表之间没有重复的数据。

在房屋出租网站中，因为有大量的交互，所以要在数据处理中学会自然观察并对不合理或冲突的操作进行预判。同时，我们需要考虑系统的数据量和数据安全问题，以便优化数据查询效率并保护用户数据的安全性。

综上所述，数据需求分析在系统设计中极其重要，不光是为了明确数据关联、建立数据表和确定字段、外键等，更是为了确保系统能够正确处理数据并达到业务目标。

## 系统非功能性需求分析

### 系统处理能力需求

系统需要具备大规模、高并发的处理能力，在云服务部署的情况下，系统处理能力总需求为：

1. 支持并行同时开展不同类型的数据检索100次/秒，支持并行接收、查看检索结果≥100；
2. 并发量大于100次/秒时，系统响应时间≤1秒。

### 可靠性需求

软件可靠性是指软件产品在规定条件下和规定时间内完成规定功能的能力。 由于功能的持续能力具有概率统计特性， 可靠性也可以定义为产品在规定条件下和规定时间内完成规定功能的概率。为保证数据库检索工作的高效完成，系统软件和数据库需要有较高的可靠性：

1.后台服务系统持续性要求：7\*24小时不间断；

2.系统有效工作时间≥99%；

3.系统不间断运行30天，运行效率不会明显降低；

4.系统故障平均间隔时间≥300天。

### 可用性需求

软件可用性是对软件系统易学、易用、用户满意等方面的评价。系统用户设计管理人员、技术人员、社会人员等，人员素质参差不齐，需要软件具有较高的可用性：

1.系统人机交互界面友好，便于使用；

2.对于PC端业务处理软件，具有相关专业知识的用户在接收10个工作日以内的培训后，90%的用户能够独立使用。

### 维修性需求

软件的维修性是指软件产品在规定的条件下和规定的时间内， 按规定的程序和方法进行维修时，保持或恢复到规定状态的能力。在发生故障时，软件应具备较高的可维修性，保障普查工作顺利进行：

1.软件系统后台服务具备自动故障恢复功能，单个部署节点发生故障时，自动恢复时间<15分钟，手工恢复时间<2小时；

2.软件系统支撑各项功能的在线更新、迭代、升级和扩充，新功能开发测试完成后，更新集成时间≤5个工作日，且升级过程中，用户日常使用不受版本更新和维护的影响。

### 环境适配需求

软件系统环境适配包括对于硬件环境的适配和软件环境的适配。考虑到参与本系统的人员有自主可控软硬件环境的情况，软件的环境适应性包括：

1.能够在自主可控的CPU硬件设备中部署和运行，性能能够达到设计要求；

2.支持各种主流数据库，可以在主流数据库中正常运行并保持性能；

3.B/S架构软件具有良好的浏览器兼容性，支持Windows平台以及Linux操作系统下的各类主流浏览器。

**购买完整文章，可联系客服微信： Lengqin1024**

# 系统测试

系统测试是关注系统的外部特性。它的主要对象是进行系统测试的测试工程师。该系统所包含的测试用例范围包括随机抽查信息平台的所有功能测试用例、环境测试用例、性能测试用例以及UI测试用例等。

## 测试策略

（1）集成测试

按照模块上下集关系，进行从上到下或者从下到上的集成测试方法进行集成测试，单元测试与集成测试主要考虑功能性测试。同时也对各个模块或者集成模块进行非功能性的抽样测试。

（2）系统测试

对整合系统进行整合测试，主要测试系统的整体功能和全部非功能性的需求。

（3）验收测试

验收测试首先进行正规性的测试，由技术人员模拟各用户环境，以用户的身份进行应用测试工作。然后进行非正规测试alpha测试和bate测试。

Alpha测试

由开发人员模拟用户进行测试，允许对需求做修改工作。

Bate测试

alpha测试后将产品提交给特定用户进行测试，冻结系统需求。

（4）功能测试

功能测试是保障测试软件各个功能模块正确、逻辑正确。对测试对象的功能测试应侧重于所有可直接追踪到用例或业务功能和业务规则的测试需求。测试的目标是核实数据的接受、处理和检索是否正确，以及业务规则的实施是否恰当。功能测试的主要参考为类似于功能说明书之类的文档。

（5）UI测试

UI测试指测试用户界面的风格是否满足客户要求，文字是否正确，页面美工是否好看，文字、图片组合是否完美，背景是否美观，操作是否友好等。

用户界面 (UI) 测试用于核实用户与软件之间的交互。UI测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能。另外，UI测试还可确保UI中的对象按照预期的方式运行，并符合行业的标准。包括用户友好性，人性化，易操作性测试。

（6）性能测试

性能测试主要测试软件的性能，包括负载测试，强度测试，容量测试等。

（7）安全性测试

安全性主要体现在以下两个方面：

应用程序级别的安全性，包括对数据或业务功能的访问。

系统级别的安全性，包括对系统的登录或远程访问。

应用程序级别的安全性，可确保在预期的安全性情况下，用户只能访问特定的功能或用例，或者只能访问有限的数据。

系统级别的安全性

可确保只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访问。比如输入管理员账户，检查其密码是否容易猜取，或者可以从数据库中获得。

（8）故障转移和恢复测试

故障转移和恢复测试指当主机软硬件发生灾难时候，备份机器是否能够正常启动，使系统是否可以正常运行。

故障转移和恢复测试可确保测试对象能成功完成故障转移，并能从导致意外数据损失或数据完整性破坏的各种硬件、软件或网络故障中恢复。如突然主机程序发生死机，备份机器是否能够启动，确保系统能够正常运行，从而不影响用户使用。

（9）安装测试

安装测试的目的就是确保该软件在运行环境下进行安装、升级等都能顺利进行。

（10）测试管理

软件测试的主要目的在于发现软件存在的错误(bug)，对于如何处理测试中发现的错误，将直接影响到测试的效果。在实际测试过程中，对于每个错误，都需要进行测试、确认、修复、验证等管理过程：

（1）错误跟踪管理

为了正确跟踪每个功能项错误的处理过程，将软件测试中发现的每个错误作为一条条记录输入错误跟踪管理系统。正确设计每个错误的包含信息的字段内容和记录错误的处理信息的全部内容。字段内容应该包括测试软件名称、测试版本号、测试人名称、测试事件、测试软件和硬件配置环境、发现软件错误的类型、错误的严重等级、详细步骤、必要的附图、测试注释等。处理信息包括处理者姓名、处理时间、处理步骤、错误记录的当前状态。正确的数据库权限管理是错误跟踪管理系统的重要考虑要素，一般要保证对于添加的错误不能从数据库中删除。

（2）软件错误的状态

新信息(New)：测试中新报告的软件缺陷；

打开 (Open)：被确认并分配给相关开发人员处理；

修正(Fixed)：开发人员已完成修正，等待测试人员验证；

拒绝(Declined):拒绝修改缺陷；

延期(Deferred): 不在当前版本修复的错误，下一版修复；

关闭(Closed)：错误已被修复；

未处理(Reopen)：错误未处理；

（3）Bug管理的一般流程

测试人员提交新的Bug入库，错误状态为New。高级测试人员验证错误，如果确认是错误，分配给相应的开发人员，设置状态为Open。如果不是错误，则拒绝，设置为Declined状态。开发人员查询状态为Open的Bug，如果不是错误，则置状态为Declined；如果是Bug则修复并置状态为Fixed。不能解决的Bug，要留下文字说明及保持Bug为Open状态。对于不能解决和延期解决的Bug，不能由开发人员自己决定，一般要通过某种会议（评审会）通过才能认可。测试人员查询状态为Fixed的Bug，然后验证Bug是否已解决，如解决置Bug的状态为Closed，如没有解决置状态为Reopen。

（4）软件错误流程管理要点

为了保证错误的正确性，安排有丰富测试经验的测试人员验证发现的错误是否是真正的错误，书写的测试步骤是否准确，可以重复。每次对错误的处理都要保留处理信息，包括处理姓名、时间、处理方法、处理意见、Bug状态。拒绝或延期错误不能由程序员单方面决定，应该由项目经理，测试经理和设计经理共同决定。错误修复后必须由报告错误的测试人员验证后，确认已经修复，才能关闭错误。加强测试人员与程序员的交流，对于某些不能重复的错误，可以请测试人员补充详细的测试步骤和方法，以及必要的测试用例。

## 测试环境

为了保证软件版本的控制，本文建立三个环境，分别是：开发环境，测试环境以及运行环境。

**开发环境**

**测试环境**

**运行环境**

开发环境: 软件产品开发工作所用的环境。

测试环境：软件测试工作所用的环境。

运行环境：软件运行的环境。

软件在各个环境中的迁移：

当软件经过开发完毕，将软件产品移植到测试环境进行测试，这样测试和开发工作可以相互独立，互不影响。

当软件测试完成发现错误，开发人员在开发环境中修改错误，修改好后，打成数据包，传输到测试环境进行回归测试。

当软件决定运行时，将软件从测试环境移植到运行环境，供用户使用。

## 测试用例及结果

### 后台功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块名称** | **测试内容** | **并发数** | **状态** | **平均耗时(秒)** |
| 登录首页 | 1）管理员登录到后台首页 | 100 | 正常 | 0.218 |
| 2）前端并发刷新后台首页 |
| 用户管理列表 | 1）查看用户列表 | 100 | 正常 | 0.582 |
| 2）新增一个用户 |
| 3）删除一个用户 |
| 4）查询用户关键词 |
| 日志管理 | 1）刷新操作日志列表 | 100 | 正常 | 0.877 |
| 2）刷新错误日志列表 |
| 3）刷新登录日志列表 |
|  |  |  |
| 分类管理 | 1）新增一个分类 | 100 | 正常 | 0.697 |
| 2）刷新分类列表 |
| 3）删除一个分类 |
| 4）更新一个分类 |
| 标签管理 | 1）新增一个标签 | 100 | 正常 | 0.233 |
| 2）刷新标签列表 |
| 3）删除一个标签 |
| 4）更新一个标签 |
| 评论管理 | 1）刷新评论列表 | 100 | 正常 | 0.605 |
| 2）删除一条评论 |
| 总览 | 1）刷新总览页面 | 100 | 正常 | 0.572 |
| 房屋管理 | 1）刷新房屋列表 | 100 | 正常 | 0.339 |

### 前台功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块名称** | **测试内容** | **并发数** | **状态** | **平均耗时(秒)** |
| 登录 | 1）用户登录到前台首页 | 100 | 正常 | 0.594 |
| 2）并发刷新前台首页 |
| 注册 | 1）填写信息注册账号 | 100 | 正常 | 0.382 |
| 搜索 | 1）顶部输入内容并Enter | 100 | 正常 | 0.227 |
| 2）顶部留空并Enter |
| 3）刷新搜索页面 |
|  |  |  |
| 首页 | 1）刷新首页 | 100 | 正常 | 0.677 |
| 2）点击一个分类查看结果 |
| 3）点击一个标签查看结果 |
| 4）点击一个单项跳转详情 |
| 详情页 | 1）多次刷新详情页 | 100 | 正常 | 0.653 |
| 2）点击收藏 |
| 3）点击添加心愿单 |
| 4）点击评论按钮并刷新 |
| 5）发送一条评论 |
| 6）评论点击最新或最热 |
| 7）点击喜欢一条评论按钮 |
| 咨询页 | 1）点击咨询按钮 | 100 | 正常 | 0.601 |
| 2）点击确定 |
| 用户中心 | 3）刷新我的收藏界面 | 100 | 正常 | 0.587 |
| 4）刷新我的心愿单界面 |
| 5）刷新资料界面 |
| 6）刷新安全界面 |
| 7）刷新推送设置界面 |
| 8）刷新消息界面 |
| 通知中心 | 1）点击通知中心icon | 100 | 正常 | 0.217 |

## 测试总结

从后台并发性能测试用例结果看，该系统可以支持到100并发需求，按在线用户和执行事务比例10：1算，可满足1000人在线需求。后台管理功能的性能亦达到非功能性需求所定义的指标。从事务高并发时服务器资源占用的情况来看，资源瓶颈大部分在应用服务器的CPU时间及数据库服务器的磁盘IO。说明在同等数据量及服务器配置的情况下，提高应用服务器的CPU配置，提高数据库服务器的磁盘IO能力，能够进一步提升系统性能。

# 总结与展望

本文使用 python的django框架设计了基于分层架构的房屋出租系统项目，完成了数据库和模型的设计，并且实现了注册、登陆、新增房屋、评论、详情和收藏点赞等基础功能和后台管理功能。在系统中也着重阐述了使用 django作为开发框架的原因，相比于 flask 和requests 繁琐复杂的配置，django更加轻量化更加方便，同时又比较稳定，极大地减少了开发人员的工作量，并且减少了开发周期。然后通过测试用例对不同的场景和数据进行了测试，通过测试发现该系统具有较高的性能，满足了社会对于房屋出租的需要。

本文对于房屋出租系统的应用研究提供了一套行之有效的解决方案，但是随着研究的深入，了解到还有许多研究方向与优化方案在本文中没有涉及到，包括：

（1）用户界面的优化。本文对于前端界面的设计较为简单，用户界面还不够美观，未来可以尝试使用更多 CSS 样式对网页进行美化，优化用户体验

（2）使用 vite全页面静态化技术，在服务端完成 html，css 的加载工作， 渲染成纯静态html文件，然后将这些文件直接以静态资源的方式部署到内容分发网络CDN上。这样的做法可以极大程度上提升响应系统速度，也更利于网站的稳定性与安全性。

（3）引入智能算法推荐技术。系统目前还不具有数据挖掘等算法的部分，在后续的开发中笔者会努力学习这部分知识，完善该系统，对房屋出租数据进行有效的利用。

参考文献

[1]何为,迟文恒. 服务端研发知识图谱[M].机械工业出版社:信息科学与技术丛书, 201812.430.

[2]杨开振. 深入浅出Django 2.x[M].人民邮电出版社:, 201808.442.

[3]古凌岚,张婵,罗佳. 计算机系统化项目开发教程[M].人民邮电出版社:, 201802.313.

[4]肖睿,喻晓路,朱微,张超,戴立坤. Web应用设计及实战[M].人民邮电出版社:大数据开发实战系列, 201801.238.

[5]李宗花,朱林. 软件工程原理与实践[M].南京大学出版社:, 202008.252.

[6]陆惠恩,褚秋砚. 软件工程[M].人民邮电出版社:, 201712.284.

[7]肖睿,郭泰,王丁磊. SSH框架企业级应用实战[M].人民邮电出版社:大数据开发实战系列, 201801.344.

[8]付森,石亮,吴起立,刘冰. MySQL开发与实践[M].人民邮电出版社:, 201408.261.

[9]陈陆扬. Vue.js前端开发快速入门与专业应用[M].人民邮电出版社:, 201702.207.

[10]陈晓兵. Web前端开发技术[M].南京大学出版社:, 202008.252.

[11]杨洋,刘全. 软件系统分析与体系结构设计[M].南京东南大学出版社:, 201710.207.

[12]王昊,刘友华. 信息系统分析与设计[M].南京大学出版社:, 202109.323.

[13]张永奎. 数据库原理与设计[M].人民邮电出版社:, 201909.260.

[14]佟伟光,郭霏霏. 软件测试[M].人民邮电出版社:, 201501.281.

[15]郑钢,贺亚涛,尤胜涛. 大型网站服务器容量规划[M].人民邮电出版社:201608.204.

[16]姚庆涛.浅析计算机网络应用安全[J].网络安全技术与应用,2023(05):169-171.

[17]熊威.python编程基础课程类和对象的教学设计[J/OL].中国教育技术装备:1-4

[18]齐研科,杨颖,贺喜,蔡雨耕,赵家骏,王一先. 软件开发系统及方法[P]. 重庆市：2023-05-02.

[19]朱若榕.人机交互软件界面设计的重要性[J].文化产业,2023(12):153-155.

[20]刘德山,金百东. 设计模式深入研究[M].人民邮电出版社:, 201407.222.