**1410064117**

**10722**

**学校代码**

**学号**

**TP311**

**分类号**

**公开**

**密级**



**题目**

**基于JavaEE的微中介系统**

**设计与实现**

**JavaEE based micro mediation system**

**Design and Implementation**

**徐勇**

**作者姓名**

**软件工程**

**专业名称**

**工学**

**学科门类**

**指导教师**

**二○一八年五月**

**提交论文日期**

**成绩评定**

### 摘 要

众所周知，类是对实体的抽象，系统则是类的集合，实体之间的相互作用、影响形成了一个系统。一个优质的系统构型必定来源于生活，其主旨在于服务生活，微中介系统是一个免费的信息资源共享平台，源于生活，服务于生活。本次毕业设计我的选题为基于JavaEE的微中介服务系统的设计与实现，系统包括两大模块，即房屋模块，零工模块。房屋管理模块服务功能即房东发布个人房源、找租人员查找在线房源；零工管理模块服务功能即雇主发布自定义零工、用户查找在线零工。微中介系统的构型来源于日常生活，是一个真正意义上的房屋、零工信息共享平台。

本文将会详细阐述微中介系统的详细架构、设计以及实现过程。微中介系统设计遵循MVC设计典范。系统前端页面设计采用BootStrap前端框架进行，视图解析用的是ThemeLeaf模板技术；后台则采用时下最为火热的SpringBoot框架和MyBatis框架整合作为基础架构；数据持久化工具我则选用的是MySQL数据库。

**关键词**：微中介服务；SpringBoot；ThymeLeaf；MyBatis

# Abstract

As we all know, classes are the abstraction of entities, systems are collections of classes, and the interaction and influence between entities form a system. A high quality system configuration must come from life. The main purpose of the system is to serve life. The micro intermediary system is a free platform for sharing information resources. It comes from life and serves life. This graduation project my topic is the design and implementation of the micro intermediary service system based on JavaEE. The system includes two modules, namely, the housing module and the odd job module. Housing management module service function that the landlord releases the personal source, find the rental staff to find the online source; the zero work management module service function is the employer to release custom work, and the user to find online zero work. The structure of micro intermediary system comes from daily life, and it is a real sense of housing and odd jobs information sharing platform.

This article will elaborate on the detailed structure, design and implementation process of the micro mediation system. The design of the micro - mediator system follows the model of MVC design. The front end page of the system is designed with the BootStrap front-end framework. The view parsing is ThemeLeaf template technology; the backstage uses the most hot SpringBoot framework and the MyBatis framework as the basic framework; the data persistence tool I choose is the MySQL database.

**Keywords**: Intermediary; SpringBoot; ThemeLeaf; MyBatis

**目 录**

[摘 要 I](#_Toc20384)

[Abstract II](#_Toc29156)

[第一章 绪论 1](#_Toc7285)

[1.1研究背景 1](#_Toc32089)

[1.2研究现状 1](#_Toc15063)

[1.3 论文组织架构 1](#_Toc7185)

[第二章 需求分析 2](#_Toc6549)

[2.1 需求分析 2](#_Toc25489)

[2.1.1 用户需求 2](#_Toc19667)

[2.1.2 功能需求 2](#_Toc16347)

[2.2系统可行性分析 2](#_Toc3340)

[2.2.1经济可行性分析 2](#_Toc23182)

[2.2.2 功能可行性分析 3](#_Toc6907)

[2.2.3 技术可行性分析 3](#_Toc17994)

[2.3项目功能性分析 3](#_Toc2119)

[第三章 项目开发技术与工具 4](#_Toc6525)

[3.1 项目开发关键技术 4](#_Toc14216)

[3.1.1 B/S结构 4](#_Toc8914)

[3.1.2 JavaWeb的MVC架构 4](#_Toc8545)

[3.1.3前台页面技术简介 5](#_Toc29284)

[3.1.4 SpringBoot于MyBatis技术简介 5](#_Toc12155)

[3.1.5 MySQL简述 6](#_Toc25922)

[3.2 开发工具简述 6](#_Toc17523)

[第四章 系统总体设计 7](#_Toc3649)

[4.1 项目目标 7](#_Toc32743)

[4.2 项目设计原则 7](#_Toc28209)

[4.3 系统功能模块 7](#_Toc2127)

[4.4数据库设计 8](#_Toc16114)

[4.5 微中介系统实体关系图 11](#_Toc12360)

[4.6 系统用例图 12](#_Toc28667)

[第五章 项目详细设计与实现 15](#_Toc15217)

[5.1 代码设计与实现原则 15](#_Toc10322)

[5.2 系统公共功能模块 16](#_Toc26389)

[5.3 前台主界面的搭建 17](#_Toc262)

[5.3.1 导航栏模块 17](#_Toc15666)

[5.3.2 房源、零工模块 18](#_Toc1189)

[5.4 后台管理模块 21](#_Toc11417)

[5.4.1 导航栏模块 21](#_Toc9034)

[5.4.2 中左管理操作模块 21](#_Toc30527)

[5.4.3 中右视图相应模块 21](#_Toc2102)

[第六章 网站测试 22](#_Toc20478)

[6.1 软件测试及工具 22](#_Toc25872)

[6.1.1 软件测试 22](#_Toc2890)

[6.1.2 测试内容 22](#_Toc10302)

[总 结 24](#_Toc8875)

[致 谢 25](#_Toc32505)

[参考文献 26](#_Toc2303)

# 绪论

## 1.1研究背景

暑假实习的时候因为住宿不方便，需要自己租房，发现找房子是一项极其费力气的事情，虽然市场上有很多诸如58同城，链家网，安居客之类的房源信息共享平台，经过本人的亲身试用，发现其功能并不如广告宣传那样没有中间商，很多信息是委托中介公司发布，不仅时使租房流程变得复杂，而且也不实惠[1]。鉴于此，我就设想能不能开发一个系统做到真正意义上的剔除中间商，使房源信息得到共享，于是我就设想了微中介服务系统，微中介系统是一个免费的房源信息、零工信息共享平台。零工模块设计构想同房源模块，让人们把闲暇时光利用起来，变成零花钱。

## 1.2研究现状

在设想此系统的时候，我有去了解和使用市场上的一些同类软件和系统。发现许多同类系统并不如描述那样，没有中间商赚差价。相反有软件不仅插入到第三方甚至是成为了第四方，增加了交易流程，也增加了用户们的资金流失。

## 1.3 论文组织架构

论文的主要内容包括从开发平台选择到系统架构，再到此系统的需求分析，概要设计，详细设计，以及本人在开发遇到的一些问题，及经验总结。本文的章节目录如下：

第一章：简单阐述了本系统的开发意义以及论文的提纲列表

第二章：详细叙述了项目的需求分析，系统可行性分析。

第三章：介绍了微中介项目开发中使用的关键技术，开发平台环境。

第四章：详细的描述了微中介系统的概要设计，包括系统的零工管理、房屋管理模块的设计。

第五章：详细描述了微中介系统的零工管理模块、房源管理模块的实现过程。

第六章：进行微中介系统测试，并对系统主要模块测试用例表进行展示。

# 

# 需求分析

## 2.1 需求分析

需求分析是软件系统开发过程中极其重要的一环，也被称为软件需求分析、系统需求分析，或者需求分析工程等[2]。需求分析是复杂的双向交流过程，作为用户需准确表达自己的需求，对功能、性能等方面得要求。开发人员分析、研究用户要求转化为完整的需求定义，确定用户需求后撰写需求文档。

### 2.1.1 用户需求

微中介普通用户群体注册登录后可以通过此系统发布自己的房源，不用去贴招租启示更不用去挂到中介公司。需要租房的人群，只需要浏览这个网站，就可以查找房屋，满意的就可以和房主联系；雇主注册登录后可以通过本系统发布零工，零工描述由雇主自行发布，其包括层面更广，例如可以找人帮你跑腿、带外卖、洗衣服、做饭。需要找零工的人可以进入系统，浏览零工信息，满意就可以联系雇主。

### 2.1.2 功能需求

所有用户功能：登录、注销、修改密码

普通用户：发布房源、查找房源、发布零工、查找零工

系统管理员：管理用户（添加普通用户、删除普通用户），管理房屋（删除、修改房屋信息），管理零工（删除、修改零工信息）

超级管理员：管理所有用户（添加用户/管理员、删除用户/管理员、修改用户信息

## 2.2系统可行性分析

### 2.2.1经济可行性分析

从项目开发角度来说本项目基于Java EE的WEB项目的开发，有很多免费开源的框架，还有许多功能强大，简单实用且免费的开发环境，例如IDEA、MAVEN。并且此项目中用到的Spring Boot框架、MyBatis框架，ThemeLeaf模板，在网上有很多免费的学习、参考资料，尤其是在github上的众多免费开源项目都会给我们开发提供帮助。从项目使用价值来讲，此项目是一款免费的服务信息共享软件，可以达到服务生活的目的。

### 2.2.2 功能可行性分析

系统功能模块包括用户信息管理、房屋管理和零工管理三大模块：

（1）用户信息管理模块：用户信息管理，用户操作管理，用户操作包含基本的登录，注册，修改密码，以及对零工模块、房源模块的管理。

（2）房屋模块：管理房屋信息（发布个人房源、查找在线房源信息）。

（3）零工模块：管理零工信息（发布自定义零工、查找在线零工信息）。

### 2.2.3 技术可行性分析

微中介服务系统的整体架构遵循传统的MVC设计模式，后台使用时下极为火热的SpringBoot微框架与MyBatis持久层框架整合作为系统的基本架构[6]。数据库采用MySQL数据库进行数据持久化。前台采用BootStrap、JQuery、JavaScript技术整合进行页面开发，利用ThemeLeaf模板进行数据解析来进行数据展示。

### **2.3项目功能性分析**

微中介系统有两类角色，管理用户（微中介系统管理员和微中介系统超级管理员）和普通用户（发布房屋用户、发布零工用户、查找房屋用户、查找零工用户）。管理员用户功能模块包括登录、注销、修改密码、管理普通用户、管理零工、管理房屋等。超级管理员功能范围与权限高于系统管理员，可以对微中介系统中所有用户进行管理。普通用户功能模块包括账号注册、登录、修改密码、发布个人房源、查找在线房源、发布自定义零工、查看属于用户自己发布的房源信息、零工信息。

# 项目开发技术与工具

## **3.1 项目开发关键技术**

### 3.1.1 B/S结构

微中介服务系统一个基于B/S架构的系统[3]。那么B/S结构的系统是怎样的业务模式？

用户通过浏览器向服务器发出请求，系统服务器响应来自浏览器的用户请求，返回用户需要的信息。相比于C/S架构的固定业务模式，B/S架构的系统灵活性更高，B/S架构的不需要去开发客户端，浏览器就是客户端，用户通过浏览器操作就可以实现自身的所有需求。但B/S架构也有不利之处，那就是系统的服务器端将会承担大量的工作，业务响应，数据库操作都将由服务器端完成会造成功能弱化等问题。

B/S结构系统包括：显示逻辑层，运算逻辑层，数据逻辑层，三层是相对独立又相互关联，在这里我们在三个层面上做一个介绍：

显示逻辑层，其主要任务为系界面展示，也承担着客户端的任务，用户在显示逻辑层发出请求，以及得到来自服务器端的响应数据的展示结果。

2.控制逻辑层，这一层在系统中承担着从显示逻辑层接收用户的操作信息与数据，并进行业务逻辑处理，返回数据回到显示逻辑层。

3.数据表示层，其主要任务是响应控制逻辑层的业务逻辑进行数据库操作，包括对数据库的查询，修改，更新和删除等功能，并将操作结果返回给控制逻辑层。

### 3.1.2 JavaWeb的MVC架构

微中介系统采用的是软件工程中的非常经典的MVC架构模式，它具有解耦合，降低生命周期成本，易于重用等优点[4]。模型（Model）是实体实例化对象，用于用户输入数据和数据库返回数据的暂取和暂存；控制器（Controller）是用于接收前台数据并配合模型（JavaBean，即Model）进行逻辑业务处理，并返回数据；视图（View）用于接收用户自浏览器输入的数据并发送到控制器，接收从控制器返回的数据并进行解析、展示。完整的MVC流程如下：视图向控制器发送请求数据，控制器接收视图发送来的数据配合模型在控制器层完成逻辑业务的处理，然后将处理的数据返回给视图进行数据展示。如图3.1所示MVC结构图。



图3.1 MVC三层结构图

### 3.1.3前台页面技术简介

我将说说为什么微中介系统采用BootStrap来写前端界面[5]。BootStrap是十分好用的工具包，它封装了丰富的CSS、HTML样式, 并且带有漂亮美观的案例，他的作者是Mark Otto和Jacob Thornton，来自Twitter的两位设计师。BootStrap的使用不论是新人菜鸟还是资深开发者都十分容易上手。当然页面效果设计必定离不开JQuery、JavaScript技术，在这里就不赘述了。

### 3.1.4 SpringBoot于MyBatis技术简介

为什么我会选用SpringBoot与MyBatis框架整合来作为系统的基本架构呢？实习的时候公司用的核心技术是SpringBoot微框架、Spring Cloud分布式微服务[6]。使用了Spring Boot技术后我就在也不想去使用老版的SSM、SSH架构了，配置又多，又繁杂。说到这里就不得不说一说Spring Boot开发者遵循着的核心思想了，Spring Boot的核心理念是“习惯优于配置”，远离了繁琐的配置文件，让开发变得“So Easy”，同时也极大程度的提高了开发效率，也就是大家说的敏捷开发。时下最为火热的数据持久层工具当属MyBatis框架了，源于其强大的SQL语句动态拼接和SQL语句可控性的特征。Spring Boot与MyBatis的强强联合已被大多数Java开发者所热衷，逐渐形成新的“SSM”架构模式。

### 3.1.5 MySQL简述

说到关系型数据库，时下最流行莫过于MySQL数据库了，它对数据管理的灵活性，以及它使用的是标准的SQL语言，都是它被广大开发者热衷的原因[7]。并且MySQL 软件体积小、速度快、[总体拥有成本](https://baike.sogou.com/v60488729.htm" \t "_blank)低，开放源代码更是吸引来了一大批铁粉，中小型网站的开发者大都选择 MySQL作为[数据库](https://baike.sogou.com/v7927370.htm" \t "_blank)管理工具。

## **3.2 开发工具简述**

微中介服务系统的开发环境选择如下：

环境要求：JDK 1.8，Windows 10。

开发平台：开发工具为IDEA，用于依赖导入、jar包管理、项目管理的Maven[8]。

开发语言：HTML超文本传输语言、Java语言、JavaScript语言。

# 系统总体设计

## **4.1 项目目标**

在需求分析中确定了微中介系统的功能实现目标，本系统开发完成后，应满足普通用户发布个人房源、查找在线房源、发布自定义零工、查找在线零工的主要功能，以及管理用户对系统的管理功能。此外微中介系统在开发过程中还应遵循以下原则：

（1）所有界面要秉承布局合理，界面设计美观、大方的原则。

（2）操作功能模块简捷、明确，使用简单。

## **4.2 项目设计原则**

微中介系统是一款应用于浏览器端的Web应用，开发过程中需遵从一定的原则：

（1）统一性原则: 开发进度需按照统一的进度开发计划，代码结构与业务逻辑结构均要从长远可维护的教的去实现[9]。

（2）成熟性原则：开发系统所选用的技术与控件需采用在该行业中成熟、稳定和具有一定先进性。

## **4.3 系统功能模块**

微中介系统的角色分为三类：超级管理员、管理员、普通用户。

公共操作模块，即微中介系统中所有角色都会使用的功能，包括登录，注销，修改密码。系统用户角色模块分为两部分，即前台用户操作模块、后台系统管理模块。

前台用户服务页面职能是为用户提供一个优秀的信息共享平台来达到房屋、零工资源的发布或获取，核心功能模块包括发布房源，查找房源，发布零工，查找零工，查看或修改用户自己发布的房屋、零工信息；

后台为系统管理页面，其职能为管理用户、房屋、零工信息，剔除无用或无效的的信息来源，达到更好的服务于用户的目的，其核心功能模块包括浏览房屋、零工信息，删除无效或者过期的房源、零工记录。微中介系统总功能模块图如图4.1所示。

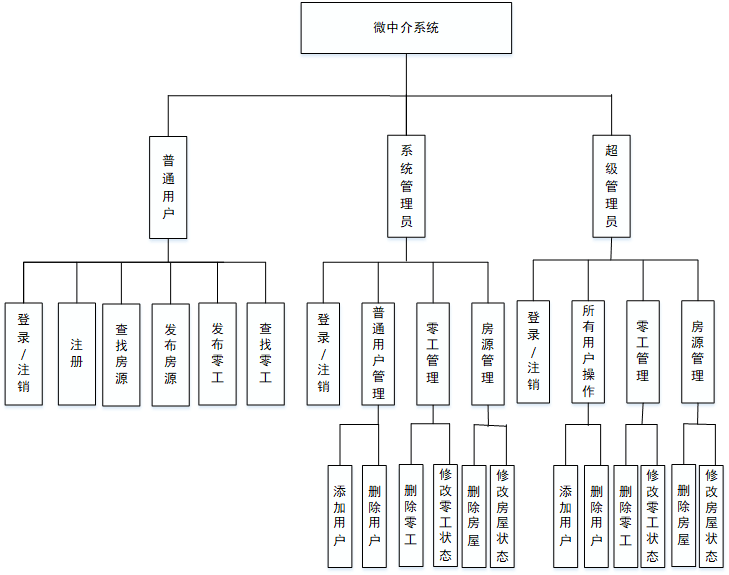


图4.1 微中介系统总功能模块图

## **4.4数据库设计**

因为这是我第一次独立搭建项目，很多方面缺乏经验，一开始还没有考虑好业务逻辑就开始写代码，需求分析做的不够细致，也没设计数据库表结构，系统代码写到一半就写不下去了，才发现业务逻辑实现不了，数据库结构满足不了项目需求，就只好推倒写了一半的项目重头再来。我说这么多无非就是想说明数据库设计在一个系统开发过程中十分重要，是系统设计的灵魂和核心[10]。不成熟的或者存在缺陷的数据库设计会致使一个项目开发寸步难行，拖延项目开发计划，甚至会造成项目的失败。在本系统的实现过程中，跌跌撞撞，但使我提高了认知，设计出合理的数据库表关系结构是开发系统中至关重要的一步。设计数据库的时候需结合当前系统的需求和业务流程，仔细斟酌实体之间的对应关系[11]。遵循数据库设计的要求，微中介服务系统表结构以及表关系结构设计如下：

用户表：包括了用户ID、用户名、用户密码、用户类型。用户表信息表4-2所示。

表4-2 用户表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 是否主键 | 说明 |
| u\_id | int | 11 | 否 | 是 | 用户ID |
| User\_name | varchar | 30 | 否 | 否 | 用户名 |
| pass\_word | varchar | 100 | 否 | 否 | 用户密码 |
| u\_type | int | 11 | 否 | 否 | 用户类型 |

房屋表：包括了房屋ID（h\_id）、房主姓名（hostname）、房屋价格(price)、房屋面积(area)、房屋信息描述(descripe)、房屋地址(address)、联系电话(phone)、房屋状态(state)、发布房屋的用户ID(u\_id)。房屋表信息如表4-3所示。

表4-3房屋表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 是否主键 | 说明 |
| h\_id | int | 11 | 否 | 是 | 房屋ID |
| hostname | varchar | 30 | 否 | 否 | 房主姓名 |
| price | double | (null) | 否 | 否 | 房屋价格 |
| area | double | (null) | 否 | 否 | 房屋面积 |
| descripe | varchar | 300 | 否 | 否 | 房屋描述 |
| address | varchar | 100 | 否 | 否 | 房屋地址 |
| phone | varchar | 11 | 否 | 否 | 联系方式 |
| state | tinyint | 1 | 否 | 否 | 房屋状态 |
| u\_id | Int | 11 | 否 | 否 | 用户ID |

零工表：包括了零工ID、雇主姓名、零工薪水、联系电话、零工描述、零工地址、零工状态、零工发布时间、发布者ID。零工表信息如图4-4所示。

表4-4 零工表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 是否主键 | 说明 |
| w\_id | int | 11 | 否 | 是 | 零工ID |
| w\_name | varchar | 30 | 否 | 否 | 雇主姓名 |
| price | double |  | 否 | 否 | 零工薪水 |
| phone | varchar | 11 | 否 | 否 | 联系电话 |
| descripe | varchar | 300 | 否 | 否 | 零工描述 |
| address | varchar | 200 | 否 | 否 | 零工地址 |
| state | tinyint | 1 | 否 | 否 | 零工状态 |
| u\_id | int | 11 | 否 | 否 | 发布者ID |
| releasetime | varchar | 30 | 是 | 否 | 发布时间 |

图片表：包括了图片p\_id、图片存储地址路径p\_url、属于房屋ID、用户ID。图片表信息如表4-5所示。

表4-5 图片表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 是否主键 | 说明 |
| p\_id | int | 11 | 否 | 是 | 图片ID |
| p\_url | varchar | 100 | 否 | 否 | 图片url |
| h\_id | int | 11 | 是 | 否 | 房屋ID |
| u\_id | int | 11 | 是 | 否 | 用户ID |

城市表：包括了城市ID（id）、父城市ID(parentid)、城市全名(areaname)、城市名缩写(shortname)、经纬度(lng,lat)，城市表结构如表4-6所示。

表4-6 城市表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 是否主键 | 说明 |
| id | int | 11 | 否 | 是 | ID |
| areaname | varchar | 50 | 否 | 否 | 地名 |
| parentid | int | 11 | 否 | 否 | 父ID |

数据库表关系图如图4.7所示。

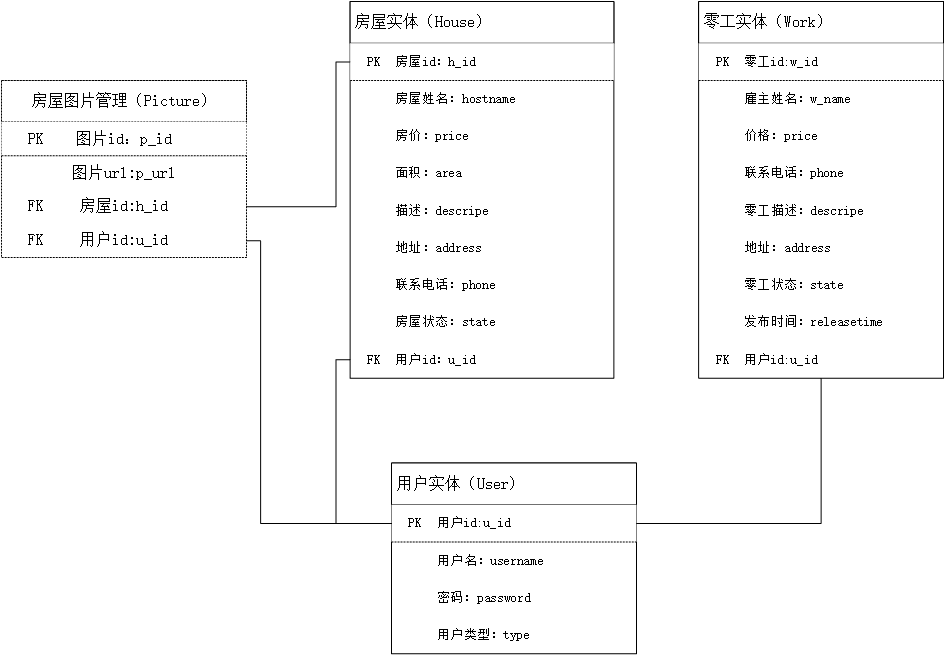


图4.7 数据库表关系结构

数据库关系结构中有四张基表分别是用户表（User）,房屋实体表（House）,零工实体表（Work）,房屋图片管理表（Picture）。用户表的主键为用户id(u\_id),用户表为其他表的基本表，房屋实体表主键为房屋id（h\_id）,外键为用户id（u\_id）关联用户表主键（u\_id）,房屋图片管理表主键id为图片id(p\_id),外键为房屋id(h\_id)关联房屋实体表房屋id（h\_id）;零工实体表主键为零工id(w\_id)，外键为用户id(u\_id),关联用户表用户id（u\_id）。

## **4.5 微中介系统实体关系图**

微中介服务系统有四类实体实体对象:零工实体对应零工对象、房屋图片实体对应房屋图片对象、房屋实体对应房屋对象、用户实体对应用户对象。用户实体与房屋实体之间存在关联关系，一个用户对应零条到多条房屋实体信息；房屋实体与图片实体存在关联关系，房屋实体可能对应零张到多张图片；一房屋实体与图片实体存在关联关系，一个用户对应零条到多条零工实体信息。系统实体E-R图如图4.8所示。

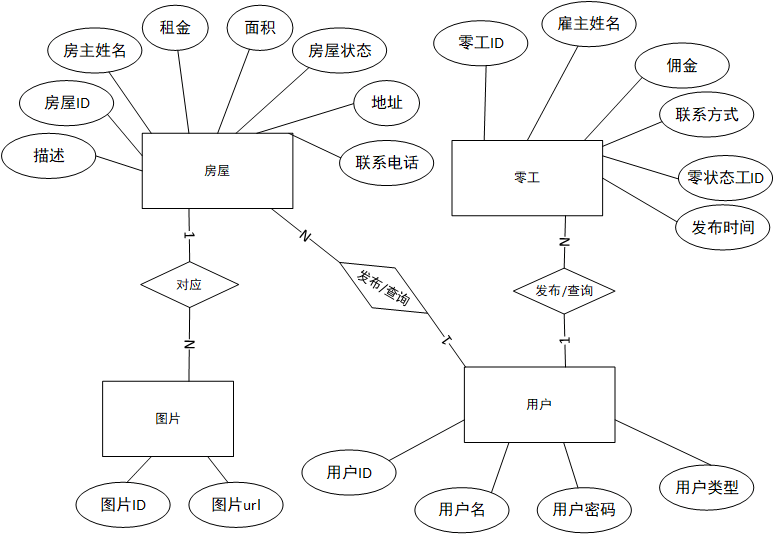


图4.8 系统E-R图

## **4.6 系统用例图**

用户进入本系统后在未登录之前可以进行注册操作，注册成功后得到用户名。用正确的用户名密码登录后可以修改密码，发布个人房源，查找在线房源，发布自定义零工，查找在线零工，查看自己发布的房源、零工信息，修改房源、零工状态。普通用户用例图如图4.9所示。



图4.9 用户用例图

系统管理员用户登陆后可以对房源信息、零工信息、普通用户信息进行管理，以及退出和修改密码的功能。其用例图如图4.10所示。



图4.10 系统管理员用例图

系统管理员其功能权限范围是超级管理员的子集，超级管理员拥有管理管理员用户的功能权限。微中介系统超级管理员用例图如图4.11所示。



图4.11 超级管理员用例图

# 项目详细设计与实现

## **5.1 代码设计与实现原则**

代码设计原则：

随着时代变化、技术进步，在软件开发过程中，对我们提出了新的要求，不论是复杂繁大的项目还是简单的小项目我们都应秉循六大原则[12]。即：

(1)单一职责原则

(2)里氏替换原则

(3)依赖倒置原则

(4)接口隔离原则

(5)迪米特法则

(6)开闭原则

微中介系统各模块代码设计与实现：

微中介系统模块均遵循MVC三层结构，用户模型对应User类，User类中定义了用户的属性：用户名、密码、用户类型，用户属性的Get、Set输入输出方法，以及构造函数和toString方法。用户类业务实现、逻辑关系处理封装在UserController中，业务完成后由UserController将处理后的数据返回给视图。由MyBatis反向代理进行数据持久层操作，SQL语句配置在MyBatis配置文件中。

House类实体对应房屋实体，House类中定义了房屋实体的属性，用于数据输入输出的Get、Set方法，无参、有参的构造函数，toString方法。房屋实体类的业务实现、逻辑关系处理操作封装在控制层HouseController中，业务完成后由HouseController将处理后的数据返回给视图。

零工模块对应实体类Work,Work类中定义了用户的属性，用于数据输入输出的Get、Set方法，无参、有参的构造函数，toString方法。零工实体类的业务实现、逻辑关系处理操作封装在控制层WorkController中，业务完成后由WorkController将处理后的数据返回给视图。

## **5.2 系统公共功能模块**

本系统公共功能模块包括登录、注销、修改密码。

登录模块详述如下：普通用户进入微中介系统后，点击登录进入登录页面，本系统注册功能设置在登陆模块下面，点击注册按钮跳转进入注册页面，填写信息，验证用户信息成功后，进入登录页面，由控制层（Controller）接收数据进行业务处理，验证用户名与密码是否与数据库中对应的用户信息匹配，验证不符提示“您输入的用户名不存在”或“您输入的用户密码错误”并返回登录页面，验证通过则将用户信息写入session，返回微中介系统前台主页。系统管理用户不能注册，只能由超管添加，管理用户验证用户名密码成功后则跳入微中介后台管理页面。登录模块流程图如图5.1所示。

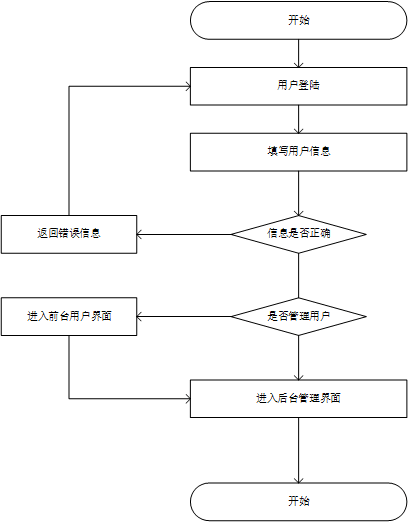


图5.1 登陆模块流程图

注销：清除session中保存的当前用户信息，界面返回系统前台主页。

修改密码：用户点击页头用户信息模块，弹出下拉菜单，点击“修改密码”，进入更改密码页面，提交新密码成功后，清除session中的用户信息，页面跳转到登录页面。

## **5.3 前台主界面的搭建**

微中介系统前台主界面分为三大模块即导航栏模块，房屋零工模块，页脚模块。

### 5.3.1 导航栏模块

导航栏模块主要功能是实现页面导航，有返回主页，导航到零工房屋操作模块，以及用户登陆等功能性按钮。用户登录后则显示用户登陆信息，没登录信息则显示登陆按钮，用户登录后可通过下拉按钮进行修改密码，查看自己发布的零工、房源信息，注销账户等功能。用户登录时序图如图5.2所示。

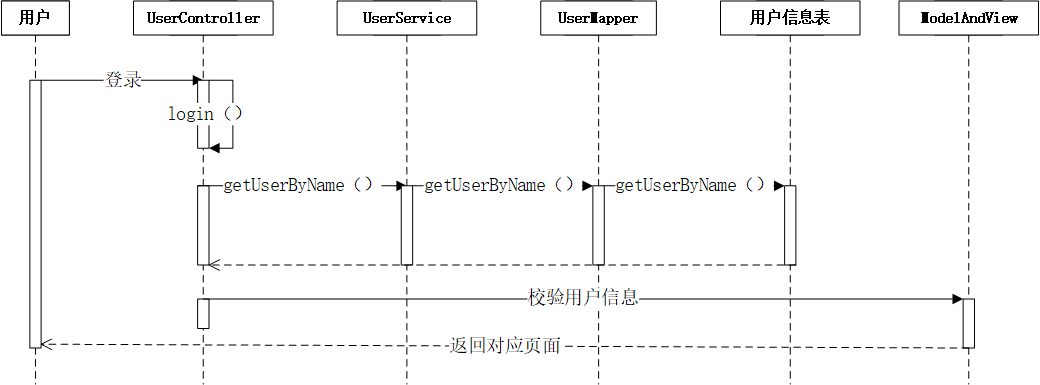


图 5.2 用户登录时序图

核心代码如下所示：

<li class="py-2 d-none d-md-inline-block dropdown" th:if="${session.user!=null}">

//session 值不为空显示  
 <a class="nav-link dropdown-toggle" style="margin-top: -5px" href="http://example.com" id="dropdown03"  
 data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" th:inline="text">  
 欢迎您，  
 [[${session.user.userName}]]<span class="caret"></span></a>  
 <div class="dropdown-menu" aria-labelledby="dropdown03">  
 <a class="dropdown-item " href="/#">我的房源</a>  
 <a class="dropdown-item " href="/#">我的零工</a>  
 <a class="dropdown-item " href="/uppassword">修改密码</a>  
 <a class="dropdown-item" href="/logout">退出登录</a>  
 </div>  
</li>  
<li class="py-2 d-none d-md-inline-block dropdown" style="margin-top: 5px" th:if="${session.user == null}"><a

//session 值为空显示  
 href="/login">登录</a>  
</li>

### 5.3.2 房源、零工模块

房源、零工操作模块有四大功能模块即：我要招租（发布房源信息），我要找租（查找房源信息模块），发布零工（发布零工信息模块），查找零工（查找零工信息模块）。点击“我要招租”跳转进入房源信息发布页面，用户输入房屋相关信息后点击“添加”按钮，操作成功返回成功页面。发布零工模块逻辑与发布房屋信息模块类同，在此以零工发布模块为例，零工发布模块时序图如图5.3所示。

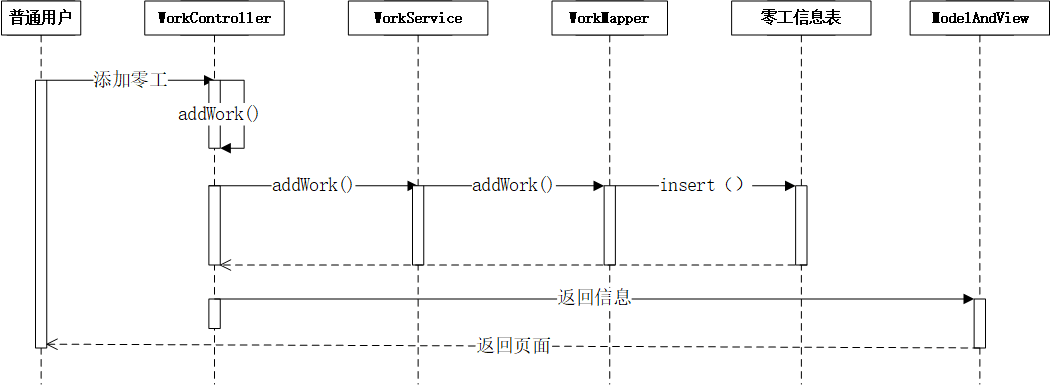


图5.3 零工发布模块时序图

在此以零工发布模块核心源码为例展示：

@RequestMapping(value = "/addwork")  
public ModelAndView addWork(Work work, HttpSession session) throws Exception {  
 /\* 当前时间\*/  
 Date day = new Date();  
 SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
 /\* session中获取用户uid\*/  
 User user = (User) session.getAttribute("user");  
 work.setuId(user.getuId());  
 work.setState(true);  
 work.setReleasetime(df.format(day));  
 int num = workService.addWork(work);  
 mv = new ModelAndView("success");  
 mv.addObject("msg", "添加了" + num + "条");  
 return mv;  
}

点击“我要找租”按钮，进入房屋信息浏览页面，此页面实现了按条件（要查找房屋的价格区间，面积区间，地址）查询房屋信息和分页功能，分页使用PageHelper插件实现[13]。用户浏览信息，找到合适的房屋信息，就可以电话联系房东。零工查找功能与之类似，在此以房屋查找模块为例，房屋查找模块时序图如图5.4所示。

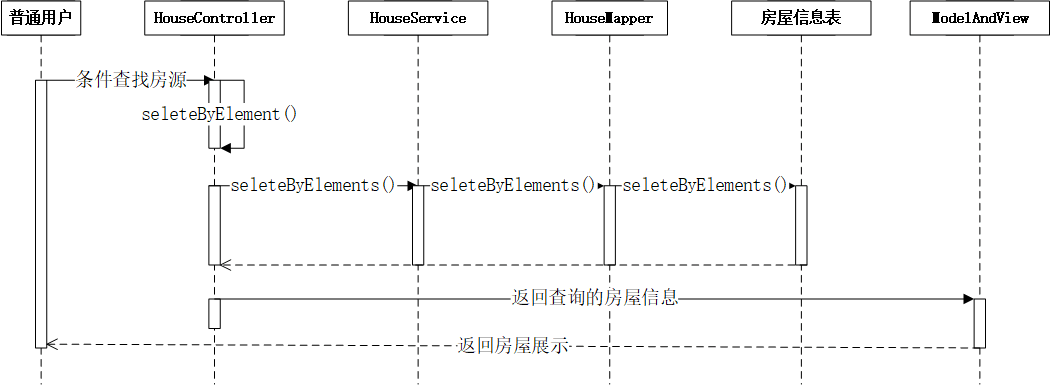


图 5.4 房屋查找模块时序图

在此以房源信息查找源码为例展示：

@RequestMapping(value = "/selectHouseByelments")  
public ModelAndView seleteByElement(House house, Double maxarea, Double maxprice, Integer pageNum, HttpSession session) {  
 LOGGER.info(house.toString());  
 LOGGER.info("页码：{}",pageNum);  
 User user = (User) session.getAttribute("user");  
 if (user.getuType() >= Manger) { // 后台管理逻辑  
 Page<?> page=PageHelper.startPage(pageNum, 8); //用PageHelper进行分页  
 List<House> houselist = houseService.seleteByElements(house, maxarea, maxprice);  
 PageInfo<?> pagehelper=page.toPageInfo();   
 mv = new ModelAndView("m\_houselist");  
 mv.addObject("houselist", houselist);  
 mv.addObject("pagehelper",pagehelper); //把视图信息加入mv  
 } else { //前台普通用户逻辑  
 Page<?> page=PageHelper.startPage(pageNum, 8);  
 List<House> houselist = houseService.seleteByElements(house, maxarea, maxprice);  
 PageInfo<?> pagehelper=page.toPageInfo();  
 mv = new ModelAndView("seclecthouse");  
 mv.addObject("houselist", houselist);  
 mv.addObject("pagehelper",pagehelper);  
 }  
 return mv; //返回模型视图  
}

## **5.4 后台管理模块**

后台管理模块有两种角色：系统管理员、超级管理员。后台管理页面分为四大模块即导航栏模块、中左管理操作模块、中右视图展示模块，底部页脚模块。

### 5.4.1 导航栏模块

本模块和大多数系统功能模块类似，用于展示管理用户信息，以及用户的操作功能（修改密码、退出登录）。

### 5.4.2 中左管理操作模块

本模块有三大模块：零工管理模块、房源管理模块、用户管理模块。

用户管理模块功能包括：

添加用户：系统管理员权限仅限于添加普通用户，超级管理员具有系统角色最高权限可以添加任意类型（超级管理员、系统管理员、普通用户）的用户，添加用户初始密码默认为“666666”。

用户列表：此模块的操作角色是系统管理员和超级管理员，系统管理员权限仅限于管理普通用户，超级管理员管理所有用户。此模块用于用户信息的展示。

房源管理模块功能包括：

房源列表：房源信息展示，删除房源操作。

零工管理模块功能包括：

零工列表：零工信息展示，删除零工信息操作。

批量删除失去状态的零工：删除所有状态完毕的零工信息。

### 5.4.3 中右视图相应模块

本模块主要用于对左边操作列表的响应与视图展示。

# 网站测试

## **6.1 软件测试及工具**

### 6.1.1 软件测试

软件测试的目的通俗的来讲，就是看你开发的系统能不能按期望目标那样运行起来，功能是否实现，以及你实现的功能是否存在缺陷等问题[14]。进行系统测试不仅是验证系统是否符合需求文档实现对应功能，还必须检查系统各模块的正确性和健壮性。在测试之前需针对系统运行过程和真实应用环境进行测试用例设计，并严格按测试计划进行测试[15]。

### 6.1.2 测试内容

登陆模块测试：针对微中介系统中的超级管理员，系统管理员，普通用户设计了相对应的测试用例，进行登陆测试。登陆模块测试用例表如表6-1所示。

表 6-1 登陆模块测试用例表

|  |  |
| --- | --- |
| 用例名称 | 登陆模块测试 |
| 测试项目 | 测试登录功能 |
| 前置条件 | 用户通过浏览器进入系统 |
| 预期结果 | 普通用户登陆验证成功后进入前台页面，管理用户登录验证成功后进入后台主页。 |
| 操作步骤 | 普通用户进入系统，点击登录按钮，进入登陆信息录入页面，输入用户名和密码，提交表单。  管理用户（系统管理员、超级管理员）进入系统，点击登录按钮，进入登录信息录入页面，输入用户名密码，提交表单。 |
| 测试结果 | 普通用户登录验证后进入前台主页；管理用户登陆信息验证后进入微中介后台管理主页。 |

房源、零工模块发布、查找测试：设计了普通用户测试用例，进行发布、查找测试。房源发布模块测试用例如表6-2所示。

表 6-2 房源发布模块测试用例表

|  |  |
| --- | --- |
| 用例名称 | 房源发布模块 |
| 测试项目 | 房源发布功能测试 |
| 前置条件 | 普通用户进行登录信息验证后，进入系统主页面。 |
| 预期结果 | 用户点击发布房源按钮，进入房源信息填写页面，填写房源信息，提交后，弹出添加房源成功提示。 |
| 操作步骤 | 用户点击发布房源按钮，进入房源信息填写界面，根据提示填写房屋信息，提交房屋图片，提交后显示房源添加成功。 |
| 测试结果 | 用户按提示填写房屋信息，提交后显示添加一条房屋信息。 |

兼容性测试：不同浏览器测试，浏览器不同版本测试。浏览器版本兼容测试用例表如表6-3所示。

表 6-3 浏览器版本兼容测试用例表

|  |  |
| --- | --- |
| 用例名称 | 浏览器版本兼容测试 |
| 测试项目 | 测试本系统在各个版本的浏览器上能否正常运行。 |
| 前置条件 | 计算机上安装多个版本的浏览器（谷歌，搜狗，qq浏览器，IE浏览器） |
| 预期结果 | 系统在各个版本上可以正常运行，完成业务功能。 |
| 操作步骤 | 启动服务器端后，在各个浏览器上输入“127.0.0.1:8088”，进入系统，进行系统功能测试。 |
| 测试结果 | 系统在各个版本上正常运行，能正确完成所有业务功能。 |

# 总 结

在老师和同学的帮助和指导下，第一次真正意义上的写了一个系统，微中介服务系统，在期间遇到了好多问题，认识到了自身的知识体系的缺陷，也学习了好多新东西。我就谈谈在项目开发中遇到的一些问题，开始的时候我就按着自己设想建了数据库，没有进行可行性分析就急于开始写代码，写到中途，发现数据库难以支持项目的设计想法，无奈只能把数据库重构，因为数据库重构势必会引起代码大幅改动，所以整个项目又回到了起点，功亏一篑。又从头开始，第二次我精心进行了数据库关系研究，确定了数据库的基本关系，大大提高了项目开发效率，但仍有很多不足之处。此外通过这次毕业设计是我深刻认识到我的语言基础还是略显浅薄，对前端知识掌握程度严重欠缺。

# 致 谢

“执手相看泪眼，竟无语凝噎”，记得四年前，第一次踏入师院校园的时候情形，整个师院笼罩在萧瑟的秋雨里，秋雨虽凉，却不敌莘莘学子入学的热情浓郁。白驹过隙，四年时光匆匆而去，又是一年毕业季，首先对四年来一直默默为我们付出的老师致以衷心的感谢，老师谢谢你们对我们的辛勤付出，辛苦了。感谢学校给了我四年不一样的青春，让我可以在这片乐土上快乐的生活，收获了知识。感谢校领导对我们的关心和支持。感谢我的指导老师对我的悉心教导，在论文撰写过程中遇到困难时为我解疑答惑，给我了很多新的解决问题思路。谢谢你投入了诸多的心血和精力在我的身上，在此报以真诚的谢意，谢谢您。就快要说再见了，我亲爱的同学们，感谢你们的陪伴同行；要说再见了，我敬爱的老师们，谢谢你们的谆谆教诲；就快要说再见了，我亲爱的母校。但愿走的时候师院也下一场雨来送我。

# 参考文献

1. 王彰红.基于B/S的信息资源共享平台的研究[J].现代图书情报技术,2004(11):70-73.
2. 徐赛华.软件需求分析研究[J].吉林师范大学学报(自然科学版),2006(01):104-105+110.
3. 刘敏娜,魏浩.基于B/S架构的电子商务网站的设计与实现[J].软件工程师,2015,18(11):42-44.
4. 张晓丽,路杨.基于MVC模式的Web OA系统的设计与实现[J].计算机技术与发展,2012,22(08):63-66.
5. 舒后,熊一帆,葛雪娇.基于Bootstrap框架的响应式网页设计与实现[J].北京印刷学院学报,2016,24(02):47-52.
6. 张峰.应用SpringBoot改变web应用开发模式[J].科技创新与应用,2017(23):193-194.
7. 林元元.JDBC连接MySQL数据库的方法浅析[J].长沙通信职业技术学院学报,2009,8(01):27-30.
8. 江日念,林霞,乔德新.Maven在Java项目中的引入及应用[J].电脑知识与技术,2013,9(21):4842-4847+4851.
9. 王消奇.计算机软件数据库设计的原则之研究[J].电脑知识与技术,2017,13(11):17-18.
10. 刘念祖. 数据库设计中的基本原则[A]. 四川联合大学.数据库研究进展97——第十四届全国数据库学术会议论文集（上）[C].四川联合大学:,1997:6.
11. 梁一平. 一种多关系数据挖掘中元组ID传播的设计与实现[A]. 中国通信学会青年工作委员会.2009年中国高校通信类院系学术研讨会论文集[C].中国通信学会青年工作委员会:,2009:5.
12. 赵鹏,孙玫,赵洁.面向对象程序设计的代码规范探析[J].江苏教育学院学报(自然科学),2013,29(01):16-18+92.
13. 高文鹏.Java Web分页技术与实现[J].电脑编程技巧与维护,2010(08):80-82.
14. Padmaraj Nidagundi,Leonids Novickis. Introduction to Lean Canvas Transformation Models and Metrics in Software Testing[J]. Applied Computer Systems,2016,19(1).
15. Oksana Petunova,Solvita Bērziša. Test Case Review Processes in Software Testing[J]. Information Technology and Management Science,2017,20(1).