目录

[摘 要 1](#_Toc483344044)

[第1章 绪论 2](#_Toc483344045)

[1.1 课题背景 2](#_Toc483344046)

[1.2 研究现状及发展趋势 2](#_Toc483344047)

[1.2.1 国外研究现状 2](#_Toc483344048)

[1.2.2 国内研究现状 3](#_Toc483344049)

[1.3 课题意义 3](#_Toc483344050)

[1.4 指导思想 4](#_Toc483344051)

[第2章 方案论证 5](#_Toc483344052)

[**2.1 开发方案** 5](#_Toc483344053)

[2.1.1 SSH开发框架 5](#_Toc483344054)

[2.1.2 SSM开发框架 5](#_Toc483344055)

[**2.2 可行性分析** 5](#_Toc483344056)

[2.2.1 经济可行性 5](#_Toc483344057)

[2.2.2 技术可行性 5](#_Toc483344058)

[2.2.3 操作可行性 7](#_Toc483344059)

[2.2.4 法律可行性 7](#_Toc483344060)

[**2.3 结论** 7](#_Toc483344061)

[第3章 系统的需求分析 8](#_Toc483344062)

[**3.1 需求描述** 8](#_Toc483344063)

[**3.2** **系统描述** 9](#_Toc483344064)

[**3.3 数据流分析** 11](#_Toc483344065)

[3.3.1 顶层数据流图 11](#_Toc483344066)

[3.3.2 一层数据流图 11](#_Toc483344067)

[3.3.3 二层数据流图 11](#_Toc483344068)

[**3.4 功能性需求分析** 12](#_Toc483344069)

[3.4.1 系统包图 12](#_Toc483344070)

[3.4.2 系统类图 13](#_Toc483344071)

[**3.5 非功能需求分析** 15](#_Toc483344072)

[第4章 过程论述 16](#_Toc483344073)

[4.1 架构设计 16](#_Toc483344074)

[4.1.1 架构设计目标 16](#_Toc483344075)

[4.1.2 系统架构设计 16](#_Toc483344076)

[4.1.2 系统网络拓扑结构设计 17](#_Toc483344077)

[4.2 系统功能设计 18](#_Toc483344078)

[4.2.1 系统活动图设计 18](#_Toc483344079)

[4.2.2模块功能设计 20](#_Toc483344080)

[4.3 数据库详细设计 22](#_Toc483344081)

[4.4 系统实现 25](#_Toc483344082)

[4.4.1 系统环境实现 25](#_Toc483344083)

[4.4.2 系统功能实现 25](#_Toc483344084)

[第5章 结果分析 32](#_Toc483344085)

[5.1 测试方法 32](#_Toc483344086)

[5.2 测试环境 32](#_Toc483344087)

[5.3 功能测试 33](#_Toc483344088)

[5.3.1 测试用例 33](#_Toc483344089)

[5.3.2 测试结果分析 34](#_Toc483344090)

[5.4 性能测试 35](#_Toc483344091)

[第6章 结论 37](#_Toc483344092)

[6.1 总结 37](#_Toc483344093)

[6.2 展望 37](#_Toc483344094)

[致 谢 39](#_Toc483344095)

[参考文献 40](#_Toc483344096)

[附 录 41](#_Toc483344097)

# 摘 要

从“网上订餐”到“网上约车”，从“共享单车”到“淘宝二手货”，我们已经在不知不觉中收获了“共享”的便捷，体验了“分享”的乐趣。在如今这个时代，“共享经济”不仅仅只局限于一种全新的经济模式，更是成为了一个时代的烙印。虽然各种租赁平台层出不穷，但是，存粹面向办公用品的租赁平台似乎尚未出现，所以我有了开发一个面向办公用品的租赁网站的想法。

本文设计且实现了基于J2EE的Web轻量级开发框架SSM(Spring、SpringMVC、Mybatis)的网上租赁平台，以面向对象的分析与设计思想，阐述了快租系统开发全过程，包括从可行性研究到系统发布的主要相关文档。本系统以Java、JavaScript、HTML、.JSP、Ajax、MySQL、Maven项目管理等为技术支撑，实现了在线租赁、、订单管理、租品管理等功能。

SSM框架是SSH的改良版，它不仅继承了SSH框架具有的易扩展性、易复用、移植性强等特点，而且还具有自己配置轻量级的优点，它将注解开发发挥到极致，且ORM实现更加灵活，SQL优化更简便。在加之引用Maven项目对象模型（POM），可以通过一小段描述信息来管理项目的构建，报告和文档的软件项目管理，使得项目构建十分方便，大大提高了项目开发效率。

关键词：J2EE，SSM框架，，Ajax,网上租赁平台

# 第1章 绪论

## 1.1 课题背景

互联网的出现和深度发展，极大改变了人们的生活水平和生活习惯，使得我们进入了一个前所未有的便捷时代。电子商务的迅猛发展就可以说是很好的见证：人们摆脱了传统交易给人的束缚，一个手机，一个软件就能实现任何时间、任何地点的轻松购物体验。电子商务取得的巨大成功，使我们开始将互联网和其他行业联系起来，于是开拓了一个全新的、众所周知的互联网+时代，其中就包括互联网+资源。

共享经济理论早已有之。传统社会里，朋友之间借一本书或者共享一条信息或者邻里之间互相借用东西，都是一种形式的共享。但这种共享受限于空间、关系两大因素，一方面，信息或者实物的共享要受限于空间的约束，只能仅限于个人所能触达的空间之内；另一方面，共享需要有双方的信任关系才能达成。2000年之后，随着互联网web2.0时代的到来，给共享经济带来了前所未有的发展机遇，一些实物共享平台如Uber、Airbnb顺势崛起，基于熟人和信任的旧式共享，逐渐向基于陌生人且存在物品使用权暂时转移的“共享经济”发展。

基于此，一条行之有效的新型资源分配方式——互联网+租赁看起来十分诱人。对于很多个人户初期成立的企业来说，经费的有限性或者办公物品调用的临时性等问题，决定了买一个或多个办公用品是对资源的不必要浪费，如果有一个平台能够提供完整的办公用品，并且他们只需支付远低于物品价格的少量租金，那么无论是对于出租者来说，还是对于承租者来说，都是有利的。本课题所设计和实现的内容，就是基于这样一个时代背景，开发出能够应用于办公物品出租中的租赁平台，使得资源能够得到合理的分配。

## 1.2 研究现状及发展趋势

1.2.1 国外研究现状

共享经济毫无疑问将成为社会服务行业内极其重要的一股力量。在住宿、交通，教育服务以及生活服务及旅游领域，优秀的共享经济企业不断涌现：从宠物寄养共享、车位共享到专家共享、社区服务共享及导游共享，甚至移动互联强需求的WiFi共享。新模式层出不穷，在供给端整合线下资源，在需求端不断为用户提供更优质体验。

共享WiFi的代表必虎WiFi，共享出行的代表UBER；共享空间代表Airbnb，面向全球的在线工作平台AAwork，共享资金价值代表Prosper，共享饮食的Eatwith等。2013年3月9日，《经济学人》杂志在其封面文章第一次详细描述了“共享经济”的场景后，共享经济的模式已在深深影响着人们的观念和生活。

1.2.2 国内研究现状

和欧洲Uber和Airbnb在市场占据的份额相比，中国共享经济的市场显得格外没有地位。从O2O到共享经济，中国的市场对于新兴模式的容忍性仿佛离不开大把大把的向里砸钱。中国的传统思想中似乎有着对“有形资产”的占有欲，这将共享经济在中国的发展牢牢绑在了异想天开的枷锁之上，所以，中国市场对共享经济的容忍性最开始于“无形资产”。如何改变这一现象，是我们急需思考和解决的问题。良好的教育、规范的法律和全民普及法律法规知识是解决问题的根本，合理有效的租赁平台则起到了画龙点睛的绝妙作用。本课题的主要研究目的也在于此。

## 1.3 课题意义

本课题设计并实现了一个网上办公用品的租赁网站，目前网站已进入试运行阶段。该网站的实现满足了个人用户在线租用办用品的需求，在功能和效率方面达到了预期效果，在很大程度上促进了办公资源的合理利用。具体来说意义如下：

1. 本系统具有较高的实用价值。

个人用户可根据自己的需求在线租用办公用品，只需支付少量租金。出租人可交付押金租用商品，商品归还且商品检查无误后全额退还押金；出租人也可免押金租用商品：提供本人身份证复印件，注明"仅限于租赁办公用品使用"；以拍照形式提交至网站，经过审核后，能享受免押金租赁。这样就可以有助于缓解承租人资金压力，避免不必要的浪费，同时合理有效的利用有限资源。

1. 本网站所采用的开发技术和设计方法可供其它同类网站参考。

在线租赁涉及多个因素，要确保商品在出租过程中不会发生错误，多人对同一物品租借不产生冲突等等，这是一个非常复杂的关系。系统采用了一种基于J2EE的新型Web轻量级开发框架——SSM框架，他是一个灵活度高、耦合度低的信息系统开发框架，可以为后续的Java系统开发提供较为容易的开发模式，也为后续的Java学员提供了非常好的技术平台【1】。

1. 本系统促进了共享经济信息化的发展。

共享经济平台作为连接供需双方的[纽带](http://baike.baidu.com/subview/621961/18000101.htm)，通过互联网工具完成租赁全过程，用户只需通过主流网站即可完成租赁全过程，促进了共享经济信的息化。其主要特点是，包括一个由第三方创建的、以[信息技术](http://baike.baidu.com/view/3226.htm)为基础的市场平台。

## 1.4 指导思想

本系统采用面向对象的指导思想对系统展开分析与设计。面向对象(Object Oriented,OO)是当前软件开发和分析中的主流分析思想，它是90年后软件开发方法的主流。面向对象的概念和应用已经不在局限于软件的设计与开发过程，它已经延伸到了其他很多领域。如数据库系统、交互式界面、应用结构、应用平台、分布式系统、网络管理结构、CAD技术、人工智能等领域。

# 第2章 方案论证

**2.1 开发方案**

2.1.1 SSH开发框架

SSH是 struts+spring+hibernate的一个集成框架，是目前比较流行的一种Web应用程序开源框架。其中使用Struts作为系统的整体基础架构，负责MVC的分离，在Struts框架的模型部分，控制业务跳转，利用Hibernate框架对持久层提供支持，Spring做管理，管理struts和hibernate。具体做法是：用[面向对象](http://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1)的分析方法根据需求提出一些模型，将这些模型实现为基本的Java对象，然后编写基本的DAO(Data Access Objects)接口，并给出Hibernate的DAO实现，采用Hibernate架构实现的DAO类来实现[Java](http://baike.baidu.com/item/Java/85979)类与数据库之间的转换和访问，最后由Spring做[管理](http://baike.baidu.com/item/%E7%AE%A1%E7%90%86)，管理struts和hibernate。

2.1.2 SSM开发框架

SSM（Spring+SpringMVC+MyBatis）框架集由Spring、SpringMVC、MyBatis三个开源框架整合而成，常作为数据源较简单的web项目的框架。

其中spring是一个轻量级的控制反转（IoC）和面向切面（AOP）的容器框架。SpringMVC分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。MyBatis是一个支持普通SQL查询，存储过程和高级映射的优秀持久层框架。

**2.2 可行性分析**

2.2.1 经济可行性

本次课题属于教学环节，不涉及商业利益，基本没有经济成本，无论选择哪种方案，在经济可行性分析上都是可行的。

2.2.2 技术可行性

传统的面向数据流的分析方法（ODA）虽然依然留存于现在的开发模型之中，但面向对象分析方法（OOA）却是现在主流的分析方法，在某种程度上已取代了面向数据流分析方法（ODA）,而作为一个基于J2EE的网站类系统，OOA似乎是更合理的选择。开发过程中所用到的开发工具：JDK1.8，MyEclipse2015，MySQL5.7，MySQL Workbench6.3等都是在网上可以找到的工具。系统的工作平台为现有主流浏览器，方便快捷，随时随地。同时，开发期限较为宽裕，预计可以规定期限内完成开发任务。技术方面可行性的分析主要是分析现有的技术条件是否能顺利完成开发工作，硬件，软件的配置能否满足开发者得需要，各类技术人员的数量，水平，来源等。使用的语言是JAVA和MySQL之间连接，这是大家还比较熟悉的内容，完全具有技术可行性。SSM框架和SSH框架现在也都已经发展的比较成熟，广泛应用与各个公司的开发过程中。以下是两种框架的比较：

1. Spring框架优点

轻量级的容器框架没有侵入性；使用IoC容器更加容易组合对象直接间关系，面向接口编程，降低耦合；Aop可以更加容易的进行功能扩展，遵循ocp开发原则；创建对象默认是单例的，不需要再使用单例模式进行处理

1. SpringMVC与Struts

他们的共同之处是都是数据javaweb层的开发框架，都是mvc模式的的经典产品，都实现了页面分离控制，但是两者还是有一定区别的。在开发中，人们更愿意使用SpringMVC而不是Struts。因为SpringMVC分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。在扩展和灵活性上更胜一筹；Struts的优势在于静态注入，插件机制和拦截器链，但是struts存在漏洞，经常会被作为攻击点进行冲击。相比更加安全简单的SpringMVC，开发者渐渐开发放弃了它。

1. Hibernate 与MyBatis

Hibernate与Mybatis都是流行的持久层开发框架，一句话概括：MyBatis 简单易上手；hibernate成熟，市场推广率高。MyBatis可以进行更为细致的SQL优化，可以减少查询字段。MyBatis容易掌握，而Hibernate门槛较高。更重要的是，mybatis提供了对应各种用途、功能的插件，而hibernate在这一方面是远远比不上mybatis的。Hibernate的DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。Hibernate对对象的维护和缓存要比MyBatis好，对增删改查的对象的维护要方便。Hibernate数据库移植性很好，MyBatis的数据库移植性不好，不同的数据库需要写不同SQL。Hibernate有更好的二级缓存机制，可以使用第三方缓存。MyBatis本身提供的缓存机制不佳。但是hibernat缺点很明确，如果涉及到多张关联表的调用时： 多表关联等比较复杂，使用的成本并不低；效率比较低，在大型项目中很少会使用到它，因为sql都是自动生成的，不太好进行人工的优化。

2.2.3 操作可行性

系统采用网页形式实现用户与数据库的交互，界面简洁友好，操作方便。用户只需对租赁流程和业务调查了解即可，并不不需掌握数据库等相关技术知识，大大提高了工作效率和操作可行性。

2.2.4 法律可行性

政府，无论是中央政府还是地方政府，一般都用法律规定组织可以做什么，不可以做什么。例如：《合同法》，《消费者权益保护法》，《专利法》，《反不正当竞争法》等对所有企业的行为或个人都做了限制。本课题属于实验教学，不涉及商业利益，不危害社会公共安全，不传播不良消息等，符合相关法律法规，在法律方面可行。

**2.3 结论**

根据上述分析得知，开发网上租赁系统不仅有良好的学习效益，而且还会产生更大的社会效益。该系统很具开发价值，它能整合大量的资源，达到资源的使用率最大化。由于个人在时间技术方面的局限性使得该系统有很多不健全的地方，希望大家能够给出意见或建议，我会在不足方面努力学习与改进。除了技术可行性以外，其他都可以不做考虑，由上面对技术可行性的比较可知，SSM是更为合适的开发方案，因此，系统在SSM的开发框架下进行开发。综上，该系统完全具备开发的各项基本条件，已经可以进行开发。

# 第3章 系统的需求分析

需求分析的结果是软件开发过程的基础，是软件工程中的一个关键过程，它关系到软件开发的成败以及软件的质量。软件需求分析的方法有很多，本章将从用户角度、租赁过程的业务流程图和数据流程、功能和非功能需求等方面对系统进行需求分析。

**3.1 需求描述**

网上租赁平台是一种新型资源利用平台，它能大大提高资源的利用效率，减少不必要的浪费或者是为资金短缺的人提供便利。网上租赁平台的用户一般分为两类：出租人，管理员，。通过互联网将两者联系起来，大大提高了工作效率。其需求描述如下：

* 可以在网站中浏览租品，可以注册成为会员，用会员身份登录后可以选择租品加入到购物车，出租人可以在购物车中租用看中的商品；成为会员的出租人可以改密，添加修改收货地址，可以操作自己的消费记录，订单，物流，合同等。租赁业务流程图如图3-1所示：

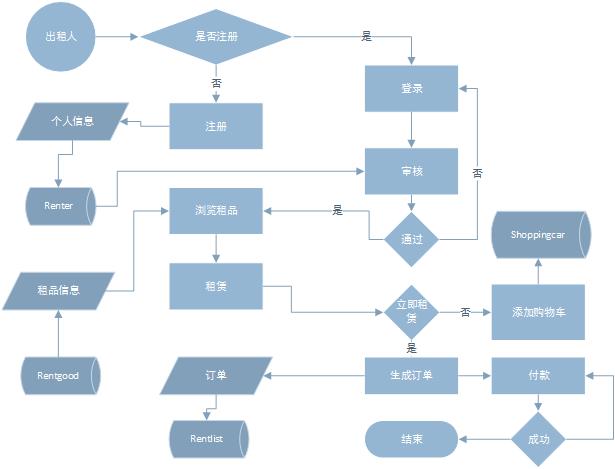


图3-1租赁业务流程图

* 管理员可以登录系统，对租品，新闻等进行添加、查看、删除、更新管理。租品管理业务流程图如图3-2示：

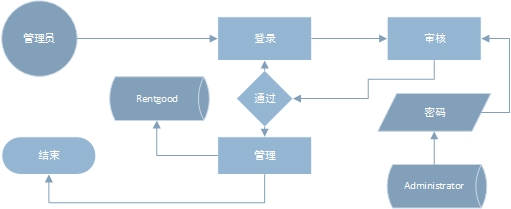


图3-2租赁管理业务流程图

* 1. **系统描述**

系统的需求获取之后就是对系统的总体分析，这也是一个十分重要的环节，如何提练系统的角色和用例是面向对象分析方法的重要节点，而系统用例和角色的正确提取，将直接影响整个系统的开发进度和系统的质量。从系统外界角度，暂时忽略掉系统的实现细节，采用基于用例的功能分析方法，是有这较高效率的分析方式。

1、管理员用户

管理员用户主要对系统的各种基础资源进行管理，包括租品信息、新闻活动、服务信息。管理员用户的用例图如3-3所示：

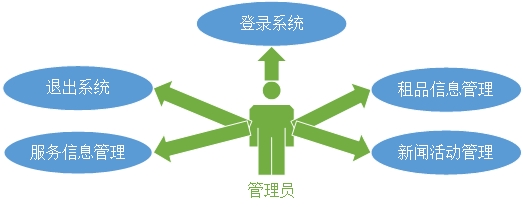


图3-3管理员用户用例图

对主要功能用例描述如下：

* 登录用例：管理员可根据自己已有的账户和密码登录
* 租品管理用例：对系统的租品信息进行增删改查操作
* 新闻活动管理用例：对系统的新闻活动信息进行增删改查操作
* 服务信息管理用例：对系统的服务信息进行增删改查操作

2、出租人用户

出租人可以浏览租品，可以注册和登录系统，可以管理自己的购物车、订单，可以支付在线和查询物流。期用例图如3-4所示：

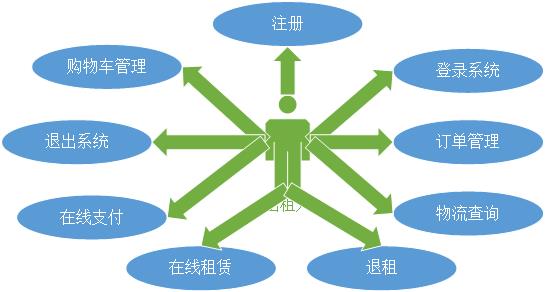


图3-4出租人用例图

主要功能用例图描述如下

* 注册用例：出租人可进行注册操作
* 订单关系用例：出租人可对自己的订单进行增删改查操作
* 物流查询用例：出租人可对自己租赁商品的物流信息进行查询操作
* 在线支付用例：出租人可对自己的订单进行在线支付操作
* 购物车管理用例：出租人可对自己的购物车进行增删改查操作
* 在线租赁用例：出租人可以执行租赁操。

3、系统用例图。

根据对出租人角色和管理员角色的单独分析，总结出系统的用例图。登录和退出系统是两者共有的，可以整合在一起。系统用例图如图3-5所示：

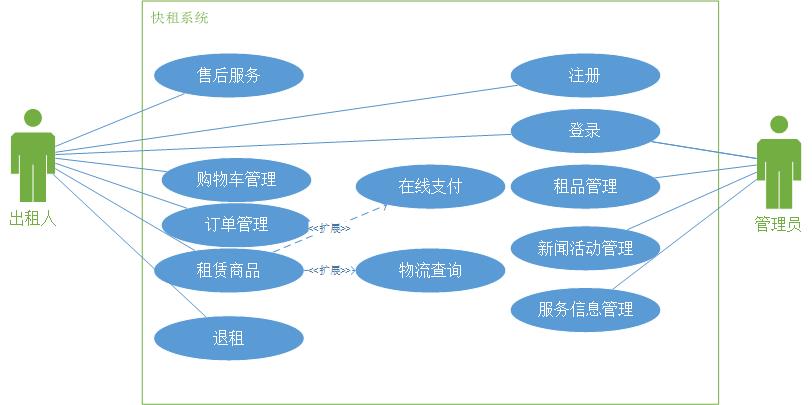


图3-5系统用例图

**3.3 数据流分析**

系统信息的载体就是数据，我们要都过系统表达的信息都是要通过底层的数据来实现。系统的主要处理对象是数据，因此，全面而准确的整理与分析系统的数据是非常重要的事情。数据流分析即是把数据在系统中的流走情况抽象出来，单从数据的流动情况方向来考察业务处理时的数据模式。数据流分析的作用主要是解决数据流通问题。

3.3.1 顶层数据流图

数据流程分析大部分是通过分层的数据流图来实现的。本系统的顶层数据流图如图3-6所示：



图3-6顶层数据流图

3.3.2 一层数据流图

在顶层数据流图的基础之上，继续分析数据流向，设计出系统的一层数据流图，入图3-7所示：

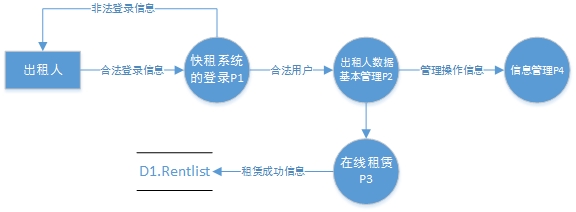


图3-7一层数据流图

3.3.3 二层数据流图

将一层数据流图继续分解，得到系统的二层数据流图如图3-8所示：

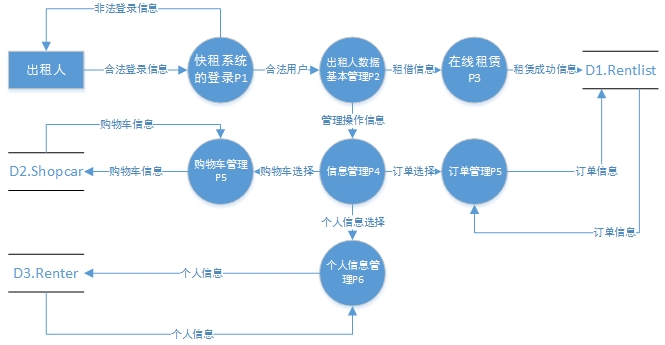


图3-8三层数据流图

**3.4 功能性需求分析**

功能需求清晰的表明了开发人员为系统设计的功能实现，用户可根据这些功能来完成任务。根据第3.2节的用例图和用例分析，可以推出系统的包图和类图。

3.4.1 系统包图

可以将系统分为三层：用户界面层、业务逻辑层、数据访问层，将各个层次中具有相同的公共部分提取出来：Service，Controller，这样得到系统包图如3-9所示

* Controller包用于处理HTML表单的请求，他只定义了一个方法，并这个方法来处理客户请求，他把用户的请求经过业务逻辑处理之后封装成一个Model，从而返回适当的模型和视图，这就是控制器所起到的作用。
* Service包是业务层，主要负责业务模块的逻辑应用设计。对客户的验证请求权限，向数据访问对象发送数据持久化操作的请求。设计Service时首先设计接口，其次是实现类。
* Dao包只是为了完成增删改查，他虽然可以完成各类查询，但是无论多复杂的查询，dao中只封装基本操作，以至于增删改查如何去实现，dao不去理会。
* Model包就是对应数据库的实现类他将从数据库查询到的完整对象封装成一个实体，供其它方法调用。
* Mapping包用于映射SQL语句，也已说是Mabatis的核心特征之一。他分为两个部分，一个部分是Mapper接口，另一个部分是Mapper映射文件。

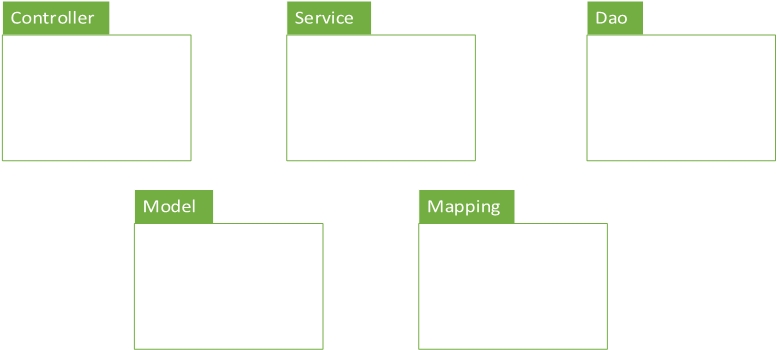


图3-9系统包图

3.4.2 系统类图

在面向对象的分析方法中，万物皆是对象。具有相同属性的对象形成类，达到了封装信息的功效，类里面包含了数据的说明和对数据的操作。类与类之间的关系表明了系统在运行过程中遵循的条用准则。下面将对系统的类作出详细的分析。介于篇幅有限，类图中只有每个属性，对属性点的操作在此做一定说明，类中增加对每个字段的get，和set方法，用于获取和更改类的属性值。

将相互关联的两个类提取出关键字生成一个接口，接口中拥有两个类的关键属性。依赖另一个类的类直接继承被依赖类的关键字，系同中的部分类的图如图3-10到3-15所示。



图3-10订单类图



图3-11收货地址类图



图3-12租品类图



图3-13合同类图



图3-13发票类图



图3-14评论类图



图3-15购物车类图

**3.5 非功能需求分析**

软件为满足用户业务需求以为的必须具有的除功能以外的需求称为非功能需求。包括系统性能，系统的适应性，维护下，可移植性、有效性和可扩充性等，下面对某些指标予以说明。

系统的可靠性、适应性和可维护性紧密联系在一起。系统应该支持故障和错误操作后的恢复，用户在运行过程中出错应在短时间内定位到问题所在。网络异常和异常中断应有容错机制。

除此之外，系统还要有较好的可维护性，可移植性，安全性、可靠性、互操作性、健壮性、易使用性、可重用性、可扩充性。

# 第4章 过程论述

## 4.1 架构设计

软件架构是一个系统的草图，它描述的对象是直接构成系统的抽象组件，各个组件之间的连接则明确和相对细致地描述组件之间的通讯，在实现阶段，这些抽象组件被细化为实际的组件，比如具体某个类或者对象【2】。也可以通俗的理解为系统的总体布局。它主要包含两个重要因素：整体到部分的最高层次划分；部署后的难以更改。

### 4.1.1 架构设计目标

正如同软件本身有其要达到的目标一样，系统架构也有要达到的目标。这一目标是架构设计者必须了解和掌握的，它直接关系着系统架构的成败和优良

一般而言，软件的架构设计要实现如下目的：

* R（Reliable）。可靠性是系统架构设计的基本要求，不可靠的架构不能开发出有效的系统软件。
* S（Secure）。软件系统承担着无比重要的商业价值，软件架构需要很高的安全性。
* E（Extensible）。计算机的发展速度超乎人的想象，技术更新日新月异，因此，系统在新技术出现时应当允许导入新技术，从而实现对对系统的扩展。
* M（MAIntainable）。软件系统的维护是为了排除现有错误和将新的软件需求反映到现有系统中去。一个易于维护的系统可以有效地降低软件维护成本。

除此以外，系统架构还需达到的目标有：可升缩性、客户体验、市场时机。

### 4.1.2 系统架构设计

根据架构设计的原则和本系统的工作原理，我们可以为系统设计出如图4-1所示的架构设计模型。

用户通过浏览器发送请求，请求传给前端控制器（DispatcherServlet）。前端控制器请求委托，请求委托会传给处理器/控制器，由控制器/处理器进行处理，控制器调用业务对象访问模型（Model），模型（Model）返回模型数据。处理器/控制器得到模型（Model）返回的数据后给前端控制器（DispatcherServlet），前端控制器（DispatcherServlet）将从处理器/控制器得到的Model交给视图层进行视图渲染，数据被渲染之后回到前端控制器，在由前端控制器返回给浏览器，到此，整个流程结束，完成一个任务。

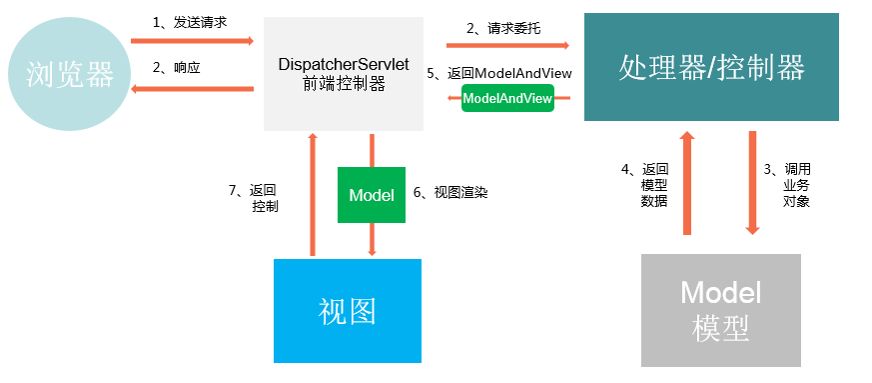


图4-1系统架构图

### 4.1.2 系统网络拓扑结构设计

网上租赁系统由部署了数据库的数据库系统服务器和部署了应用服务中间件与应用软件的服务器组成客户机通过网络与服务器相连。系统网络拓扑结构图如图4-2所示：

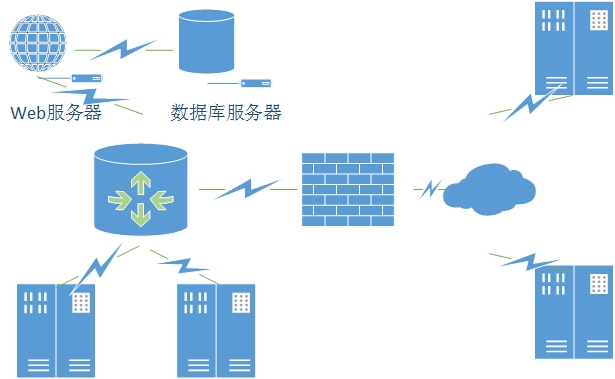


图4-2系统网络架构图

## 4.2 系统功能设计

本系统主要有三个角色权限，分别是非会员用户、会员用户和系统管理员用户。

* 非会员用户：可以浏览网站的部分内容，能够对租品进行查询操作，能注册成为会员。
* 会员：是系统的主要服务对象。可以查询租品；对租品进行在线租赁；可以管理自己的订单；可以管理收货地址；可以对已租租品进行评价；可以退货；可以申请售后服务。
* 管理员：是系统后台数据的主要管理对象，能管理租品、新闻活动。
* 合同：系统可以自动生成租赁合同，作为双方的法律约束文件。

系统整体功能图如图4-3所示：

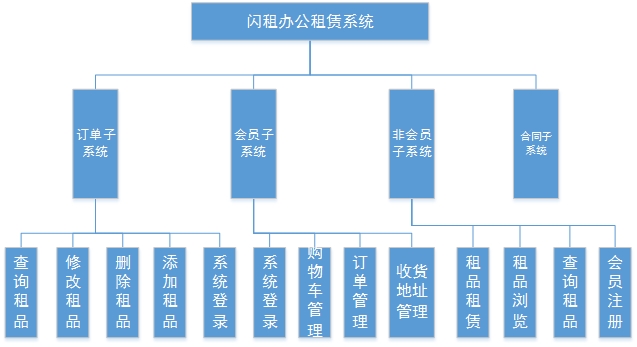


图4-3系统功能模块图

### 4.2.1 系统活动图设计

闪租办公租赁系统的角色有：管理员、非会员用户、会员用户。所以系统大致可分为管理员模块、会员模块、非会员模块。他们都有属于自己的功能子模块和业务流程，下面将针对各个角色给出相应的活动图。

1. 管理员

管理员进入系统，然后对系统进行的活动可分流成管理租品和管理新闻活动；管理租品接着分流为添加租品、修改租品信息、查看租品信息、删除租品信息；管理活动分流为添加新闻活动、修改新闻活动信息、查看新闻活动信息、删除新闻活动信息；各自合流到首页；最后合流为退出系统。管理员活动图如图4-4所示：

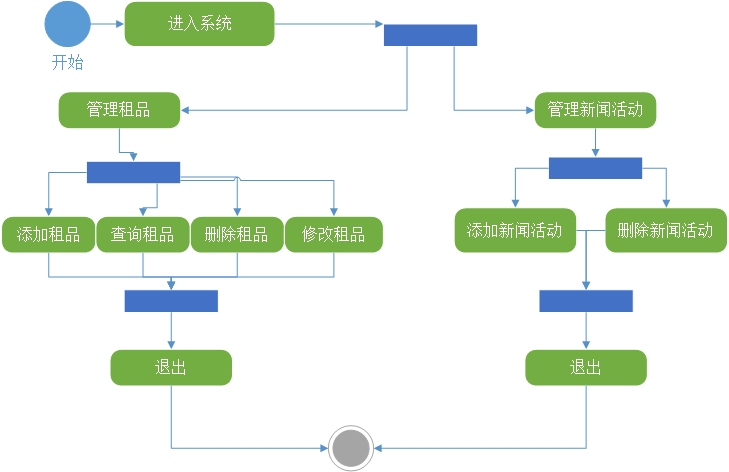


图4-5管理员活动图

1. 会员用户

会员首先登录；登录之后可进行租赁、购物车管理、订单管理和退租活动；租赁后确认订单，然后付款；购物车管理科分流恩增删改购物车；订单管理分流为查删订单；地址管理分流为增删改地址信息。其活动图如图4-6所示

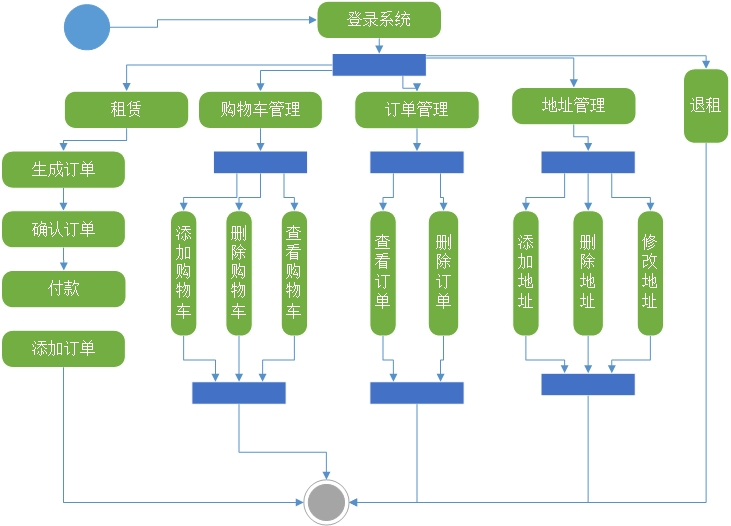


图4-6会员活动图

1. 非会员用户

非会员进入系统后可以进行浏览租品和注册会员等活动。其活动图如图4-7所示

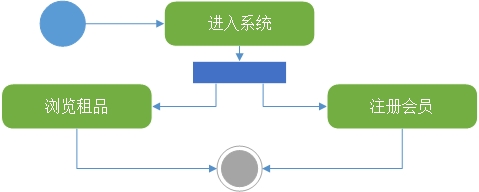


图4-7非会员活动图

### 4.2.2模块功能设计

业务流程是为达到特定的价值目标而由不同的人分别共同完成的一系列活动。活动之间不仅有严格的先后顺序限定，而且活动的内容、方式、责任等也都必须有明确的安排和界定，以使不同活动在不同岗位角色之间进行转手交接成为可能。【】由于系统的子模块较多，所以只对系统的主要功能模块采用UML建模方式进行顺序图建模。

1. 登录子模块

本系统用到登录子系统的主要有两个角色权限：会员用户，管理员。用户在登录前需要选择自己的角色所对应的登录权限，系统根据角色权限验证其相应的登录权限，如果登录信息正确，系统跳转到和登录角色相对应的功能界面。其顺序图如图4-8所示：

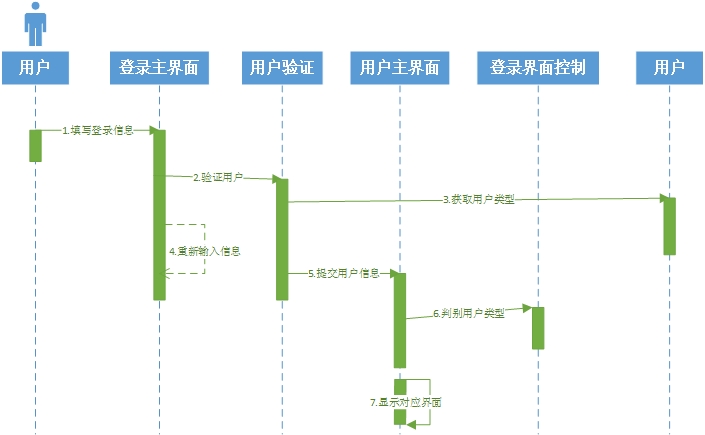


图4-8登录顺序图

1. 注册子模块

注册子模块的主要角色对象是非会员用户，非会员进入到系统主界面之后可以提交成为会员的申请，系统将申请传给注册界面，用户正确填写相应信息之后系统将会员信息提交给后台数据库。顺序图如图4-9所示：

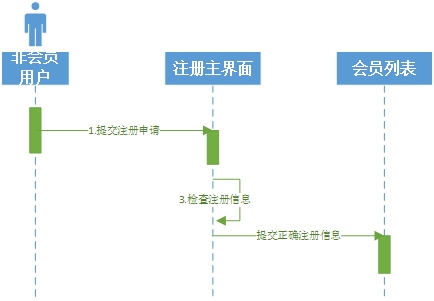


图4-9注册时序图

1. 租赁子模块

租赁模块的主要服务角色是会员用户。会员在主页中找到自己中意的租品给系统发送租赁请求，然后进入到租品选择界面选择要租机的租品型号等，确认租赁后生成租赁订单供用户查看，用户确认无误后可以付款，完成租赁。其顺序图如图4-10所示：

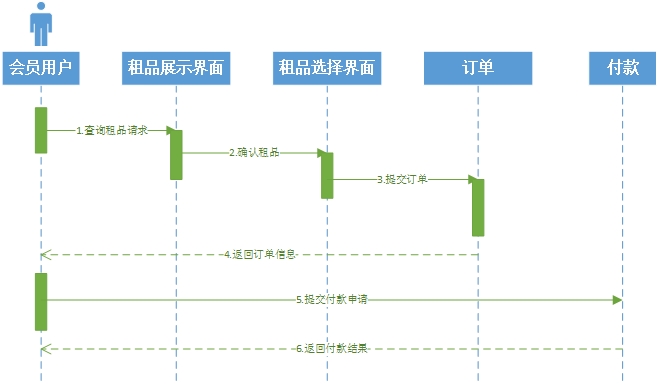


图4-10租赁顺序图

1. 添加租品子模块

添加租品模块的服务角色是管理员。管理员提交添加租品申请后系统几行入到租品添加主页，管理员填写相应的租品信息，经系统检测正确后提交到后台数据库。其顺序图如图4-11所示：

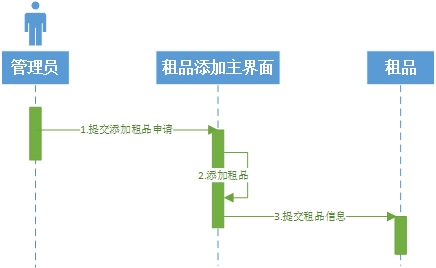


图4-11租品添加顺序图

## 4.3 数据库详细设计

根据第三章的需求分析，我们已经得出了用户的关系模式，并为其设计了详细的实体类图，将类图映射到数据库，实际上可理解为数据库的概要设计到逻辑设计的一个映射。同时，结合数据流图的分析，对闪租系统数据表进行详细设计。系统的主要设计的表有：租客信息表、合同表、租品信息表、管理员表、购物车表、订单表、评论表、发票表、销售记录表、租品细节表。

租客信息表用于存储会员的注册信息，主要内容为：电话号码、邮箱、密码、账户余额。具体如表4-1所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 字段名 | 类型 | 长度 | 主键 | 非空 |
| 电话号码 | phonenumber | vachar | 11 | Yes | Yes |
| 密码 | password | vachar | 16 | no | no |
| 邮箱 | email | vachar | 20 | No | no |
| 账户余额 | account | double |  | No | no |

表4-1租客信息表

购物车表用于存储会员有意愿租赁的租品。具体如表4-2所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 字段名 | 类型 | 长度 | 主键 | 非空 |
| 购物车编号 | idshoppingcar | vachar | 20 | yse | yes |
| 租客电话 | tb\_renter\_phonenumber | vachar | 11 | yes | yes |
| 租品数量 | amount | int | 11 | no | No |
| 租期 | tenancy | int | 11 | no | no |

表4-2购物车表

订单信息表主要用于存会员已经租赁或者正在租赁的商品。具体如表4-3所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 字段名 | 类型 | 长度 | 主键 | 非空 |
| 编号 | idrent | Vachar | 20 | Yes | Yes |
| 数量 | amount | Int | 11 | No | No |
| 期数 | periodnumber | Int | 11 | No | No |
| 总金额 | money | Float |  | No | No |
| 滞纳金 | overduefine | Float |  | No | No |
| 支付状态 | paystatus | Vachar | 40 | No | No |
| 租赁状态 | rentstatus | Vachar | 40 | No | No |
| 起租时间 | renttime | DATETIME |  | No | No |
| 支付时间 | costtime | DATETIME |  | No | No |
| 退租时间 | sendbacktime | DATETIME |  | No | No |
| 电话号码 | tb\_renter\_phonenumber | Vachar | 20 | Yes | Yes |

表4-3订单信息表

评论表主要用于存储用户对租品的评论。具体如表4-4所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 字段名 | 类型 | 长度 | 主键 | 非空 |
| 标号 | idreview | Vachar | 20 | Yes | Yes |
| 内容 | details | Text |  | No | No |
| 时间 | reviewtime | Datatime |  | No | No |
| 货物编号 | tb\_rentgoods\_idrentgoods | Vachar | 20 | Yes | Yes |
| 电话 | tb\_renter\_phonenumber | Vachar | 20 | Yes | Yes |

表4-4评论表

货物信息表主要存储货物的信息。其具体如表4-5所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 字段名 | 类型 | 长度 | 主键 | 非空 |
| 租品编号 | idrentgoods | Vachar | 20 | Yes | Yes |
| 租品名 | commodityname | Vachar | 100 | No | No |
| 单期租金 | rentalprice | Float |  | No | No |
| 数量 | amount | Int | 10 | No | No |
| 押金 | antecedentmoney | Folat |  | No | No |
| 描述 | describle | Text |  | No | No |
| 图片 | image | Blog |  | No | No |
| 查询次数 | selecttimes | Int | 11 | No | No |

表4-5货物信息表

管理员信息表主要存储管理员的信息，由于系统目前本身并不是针对管理员开发的，所以只有简单的管理员编号和密码两个字段，具体如表4-6所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 字段名 | 类型 | 长度 | 主键 | 非空 |
| 管理编号 | idadministrator | Vachar | 20 | Yes | Yes |
| 密码 | password | Vachar | 16 | No | No |

表4-6管理员表

合同表主要用于存储出租人和承租人的合同内容，具体如表4-7所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 字段名 | 类型 | 长度 | 主键 | 非空 |
|  | idcontract | Vachar | 20 | Yes | Yes |
|  | details | Text |  | No | No |
|  | responsiblepeople | Vachar | 45 | No | No |
|  | time | Datatime |  | No | No |
|  | tenant\_idtenant | Vachar | 20 | Yes | Yes |
|  | tb\_renter\_phonenumber | Vachar | 20 | Yes | Yes |
|  | tb\_tenant\_idtenant | Vachar | 20 | Yes | Yes |

表4-6合同信息表

## 4.4 系统实现

### 4.4.1 系统环境实现

系统的实现建立在需求分析和系统设计之上，在MyEclipse 2016 CI和MySQL的开发工具之上实现快租系统的开发。操作系统采用Windows2007；发布服务器采用Tomcat8.0；开发和编译环境采用Myeclipse2016+jdk1.8；CPU采用Intel core i5；网络环境无特别要求；浏览器采用主流浏览器。

### 4.4.2 系统功能实现

1. 系统登录功能模块的实现

用户通过URL进入到系统之后，显示的是系统的欢迎页面，用户可以在页面的左上角选择登录点击登陆后会出现一个登录界面，用户根据自己的类型选择要登录的类型。选择结束后，用户在登录界面输入电话号码，系统会自动检测输入号码的正确性；之后用户输入密码，系统检查密码格式；若密码和电话号码格式正确用户点击登录后系统检查用户名和密码是否匹配。一切无误后系统会根据各自账户的类型跳转到各自的欢迎页面。登录页面的具体实现如图4-12所示：

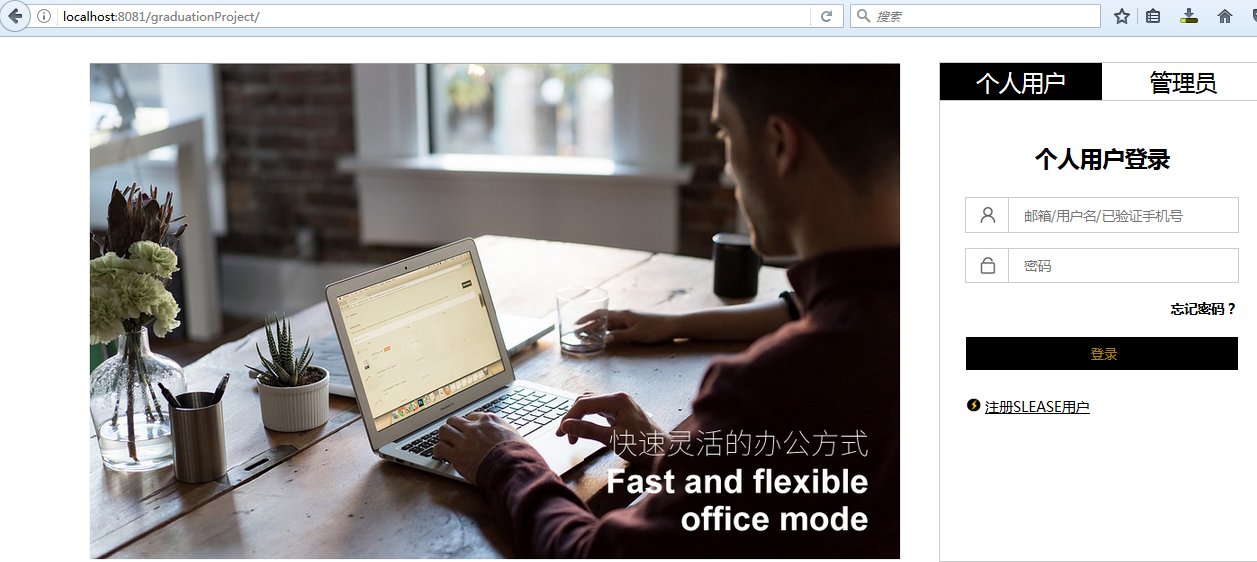


图4-12系统登录主界面

用户验证的关键代码如下图所示。用户验证的实现过程是在链接数据库成功之后匹配输入的用户名和密码，匹配成功后进入相应的国恩功能界面。

public boolean login throws SQLException {

DbConnection;//用于连接数据库，匹配账号和密码操

Stringsql=”select\*from tb—user where phoenenumber=“”+password =”””；∥获取用户名和密码

if() {

String message=”用户名为”+username+”登陆了系统”；

String sqs=”insert into tb\_fizhi(操作，时间)values(…+message+l'’，…+time+…)”；／／ 监控登录记录

ResultSet rs=be．getRs(sql)；

return true；//符合登录条件

}

else

return false；//#登录失败

}

此外，系统也能满足非会员用户的注册，用户在regist.jsp中填写相应的用户信息，输入的信息存在于一个表单之中；系统提交表单之后后台会对数据的合法性进行验证，如果数据有非法则会返回到regist．jsp并弹出提示；如果数据验证通过则会在RenterMapper类增加一名新的学生用户。

注册的流程是放用户输入合法的用户名和密码之后，系统对用户进行检测，如果用户已存在，则用户需要重新注册。如果相关信息检测通过，则用户名注册成功。其关键代码如下：

public class reg\_servlet extends HttpServlet{

if(gongneng．trim()．equals(“reg”)){∥获取注册用户名等信息

if(usemame．equals(“”)Nrealname．equals(“”,””)Npwd．equals(“”,””)llpassword．equals(…’)mail．equals(””)llsex．equals(…’)lladdress．equals(…’)[[telnumber．equals(…’)){∥判断用户名，密码等信息是否为空

//如果为空则提示用户重新输入相关必填信息

}

else{

//判断用户名是否存在，若存在提示用户名已经存在，请重新注册

}

else

newuser．newuserO；//用户注册成功，并且存储新用户

}

注册页面设计如图4-13所示

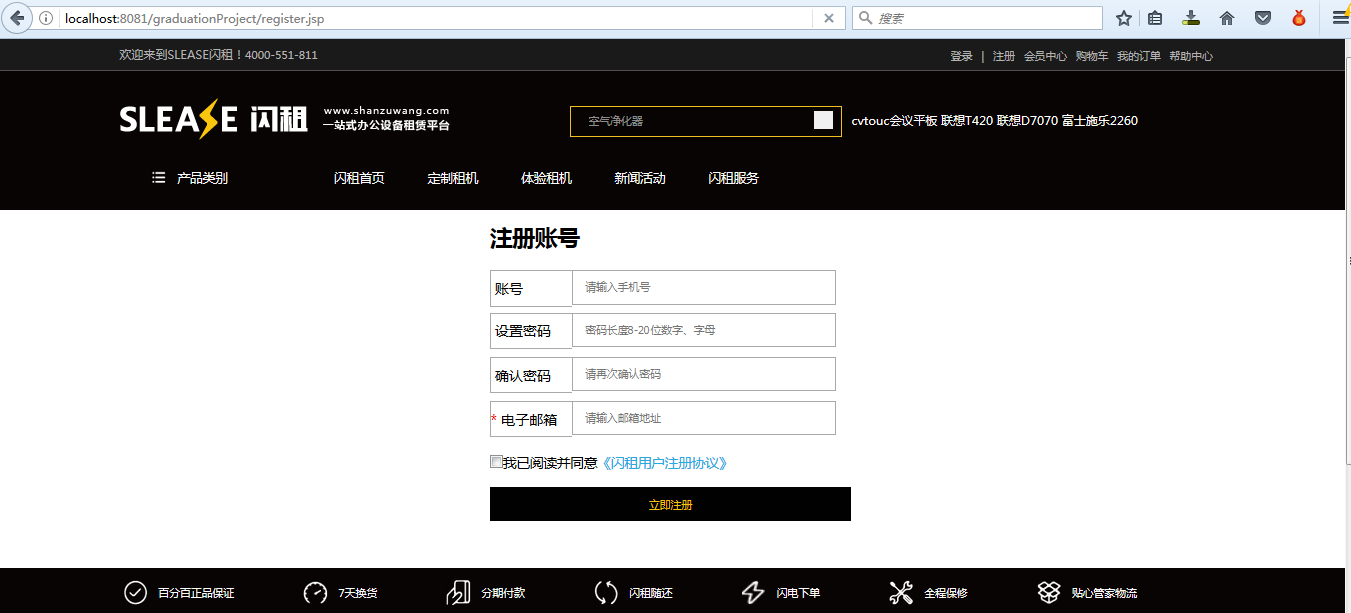


图4-13注册界面

1. 租赁模块的设计与实现

租赁过程的实现并不是一步完成，需要好几步操作。登录的会员首先在物品栏中选中需要租赁的商品，系统执行addController/add.do()方法跳转到add.jsp页面中确定要租赁物品的型号，数量，起租日期；用户选择好要租赁物品的详细信息后点击立即下单，系统执行ensureController/ensure.do()方法跳转到rentlist.jsp页面，然后生成租赁合同，如此继续向后执行，直到确认付款，完成租赁。系统的关键代码如下：

@Controller("addController")

@RequestMapping("/addController")//注解，表明下面的类是一个controller

**public** **class** AddController {

**private** AddServiceI addService;

**public** AddServiceI getAddService() {

**return** addService;

}

@Autowired//注入addService带来的值

**public** **void** setAddService(RegistServiceI registService) {

**this**.registService = registService;

}

@RequestMapping("/add")

**public** String add(HttpServletRequest request){

Rentgoods rentgoods=**new** Rentgoods();//定义租品对象，返回他的值

rentgoods=registService.selectByPrimaryKey(renter);//通过主键查询

**return** rentgoods;//返回查找的内容

}

}

租赁过程的界面设计如图4-14到4-16所示



图4-14租品信息确认图

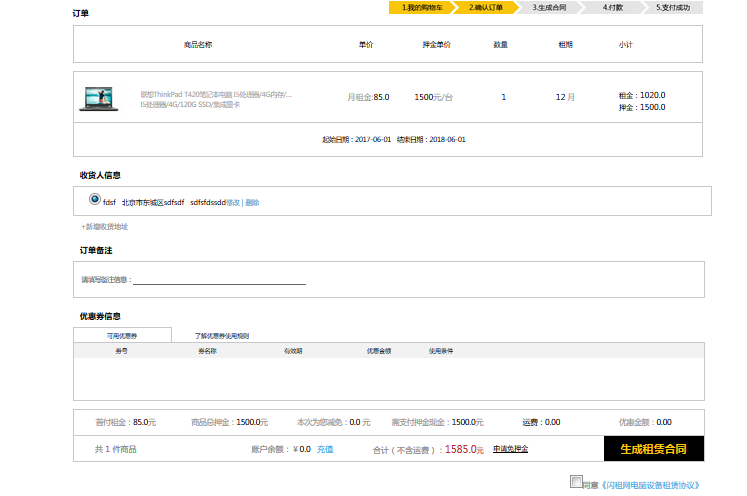


图4-15生成合同图



图4-16支付界面

1. 收货地址管理的设计与实现

根据前面的需求分析和概要设计，现对用户地址的管理作出详细的功能测试用户点击我的地址之后触发addController/addadress()方法，系统跳转到addadress.jsp页面，用户开始添加详细内容，确认保存后，系统执行submit()方法，将用户需要添加的地址信息录入到数据库。部分代码如下：

<form role="form" id="form" class="form-horizontal" method="post" action="verifysaveAddress" theme="simple">

<div style="border: 1px solid #c8c8c8;">

<div class="myProfile">

<input id="adderssId" name="id" type="hidden">

<ul style="padding-top:0.2rem;" class="myProfile3">

<li><span style="margin-left: 35px;font-size: 15px">地址管理</span></li>

<li><span style="font-size: 14px;margin-left: 101px">收货人：</span><input id="shipname" class="myProfile1" name="shipname" type="text"></li>

<li><span style="font-size: 14px;margin-left: 87px">邮  编：</span><input id="zip" class="myProfile1" name="zip" type="text"></li>

<li><span style="font-size: 14px;margin-left: 87px">详细地址：</span><input id="shipaddress" class="myProfile1" name="shipaddress" type="text"></li>

<li><span style="font-size: 14px;margin-left: 87px">手机号码：</span><input id="phone" class="myProfile1" name="phone" type="text"></li>

<li><input class="myProfile7" id="chk" name="sex" type="checkbox"><span style="font-size: 11px">设置为默认收货地址</span>

<br>

<input value="保存" class="myProfile4" type="submit"></li>

</ul>

</div>

收货地址管理的界面设计如图4-17所示：



图4-17地址管理功能图

1. 租品添加模块的功能实现

租品管理是管理员的权限，主要时针对对租品的增删改查。由于论文的篇幅有限，在此只针对对租品的添加做了详细的设计实现。

租品的添加主要是管理员对数据库的词条增加的一个过程。就添加租品而言，系统给出addgoods.jsp页面，管理员根据页面的提示填写相应的租品信息，完成信息的填写之后点击提交按钮，前端将form表单的内容存在request中供后台数据库提取，之后controller提取request中的内容，把他交给service，service在将从controller中的数据移送到dao中，最后完成数据词条的添加。具体代码流程如下：

@Controller("Controller")//指明接下来的类为控制类

@RequestMapping("/Controlle)//为前端指明类的调用路径

public class Controller类名{

private RentgoodsSerrvice rentgoodsService;

//定义一个服务属性，以便调用Service类

rentgoodsService的get方法；

@ Autowired

//为rentgoodsService自动注入

rentgoodsService的set方法；

@RequestMapping("/regist")；//为前端指明方法的调用路径

将得到的字符转换成一个对象并传给Service的方法；

}

@Service("registService")//注明一下类是一个Service类

public class Service类名{

private RentgoodsMapper rentgoodsMapper;

//定义一个dao的属性值，以便调用dao

rentgoodsMapper的get方法；

@ Autowired//为rentgoodsService注入内容

rentgoodsService的set方法；

@Override//指明下面的方法为重写的接口类的方法

public void insertSelective(Rentergoods record) {

rentgoodsMapper.insertSelective(record);

//将租品对象传到dao中

}

}

# 第5章 结果分析

在对快租系统的设计与实现基础之上，本章将对系统的结果进行详细分析，也即软件测试。软件测试是软件质量保证的一个手段，是为了发现并指出软件中存在的缺陷过程，这个过程指明和标注了问题存在的正确位置，详细记录导致问题出现的操作步骤，及时存储当时的错误状态，通过以上内容的组合，测试后问题能够准确在现【】。

## 5.1 测试方法

系统测试的方法一般涉及测试下列几个功能指标：性能质量、安全质量、可靠性质量、易用性质量，主要采用功能性测试和系统性测试相结合的方法。功能性测试的方法能够验证系统是否能满足需求分析所定义的软件功能，以得出系统是否满足客户需求的目的；系统的性能测试时为了确保系统的安全性、应用性和健壮性。

软件测试的方法多种多样，可以从不同的角度进行分类：从是否需要执行被测软件的角度分成静态测试和动态测试；从是否对系统的内部结构和具体实现算法的角度来看，可以划分成白盒测试和黑盒测试；从软件测试的的策略和过程角度，可以分为单元测试、集成测试、确认测试、系统测试和验收测试。其中，黑盒测试忽略程序内部的结构特性，依靠程序功能需求规格说明书，在只知道程序功能的情况下进行；白盒测试就与程序的内部结构，检查内部操作是否按规定执行。本文基于SSM的网上租赁平台程序复杂，系统内部结构关联性强，百合测试诸多不便，因此主要采用黑河测试的方法。

## 5.2 测试环境

本文对系统的测试主要是确定系统的运行能力，确保系统满足初期开发功能，并寻找系统的不足之处。具体测试环境如表6-1所示：

|  |  |  |
| --- | --- | --- |
| 测试系统 | 基于SSM的租赁网站 | 测试硬件环境：  普通笔记本电脑 |
| SSM应用服务器 | Tomcat8.0 |
| 测试前端 | 主流浏览器 |
| 数据库服务器 | MySQL5.7 |
| 测试网络环境 | 无特殊要求 |
| 测试方法 | 黑盒测试/并发性能测试 |

## 5.3 功能测试

### 5.3.1 测试用例

本文只要采用黑盒测试的测试方法，对闪租系统进行功能性测试。测试的主要模块是：会员登录模块、游客注册模块、会员租赁模块、地址管理模块、进行了测试，具体如表6-2至表6-所示：

表6-2游客注册模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 游客注册功能测试 |
| 用例ID | T1 |
| 基本描述 | 游客注册主要是要判断后台数据库是否存在将要注册的用户 |
| 测试方案 | 先用未注册的账户进行注册，然后用注册过的用户进行注册，反馈结果 |
| 测试数据 | 输入电话18629974397；输入电话18629974397 |
| 测试结果 | 第一次注册成功，系统会员记录中添加了一条记录；  再次输入注册失败，会员中已经存在该用户。 |
| 测试人 | 阳乾隆 |

表6-3会员登录模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 会员登录功能测试 |
| 用例ID | T2 |
| 基本描述 | 用户名和密码匹配并且存在于数据库中能成功登录；  用户名和密码不匹配或者数据库中不存在登录失败。 |
| 测试方案 | 先用账户和密码匹配且存在于数据库中的数据测试；然后用账户和密码不匹配的数据测试；最后用数据库中不存在的进行测试。 |
| 测试数据 | 18629974397,9999999.；18629974397,1561263216；18629974698，9999999.。 |
| 测试结果 | 第一次登录成功；后两次登录失败 |
| 测试人 | 阳乾隆 |

表6-4会员租赁模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 会员租赁模块功能测试 |
| 用例ID | T3 |
| 基本描述 | 会员通过租赁说明一步步完成租赁，未付款的订单无法完成交易 |
| 测试方案 | 在操作过程中不付款 |
| 测试数据 | 点击未支付 |
| 测试结果 | 返回网站首页 |
| 测试人 | 阳乾隆 |

表6-5地址管理模块测试用例

|  |  |
| --- | --- |
| 用例名称 | 地址管理模块功能测试 |
| 用例ID | T4 |
| 基本描述 | 用户在输入收货信息时要输入正确的手机号码格式 |
| 测试方案 | 先输入正确的电话号码格式，在输入错误的格式 |
| 测试数据 | 18629974397；fghj1852465 |
| 测试结果 | 正确保存；提示输入信息错误 |
| 测试人 | 阳乾隆 |

### 5.3.2 测试结果分析

本文对会员登录模块、游客注册模块、会员租赁模块、地址管理模块等几个模块进行测试。图6-1和6-2分别系统游客注册的编辑和注册成功的测试结果图。下图6-3登录成功的的测试结果图。下图6-4和图6-5分别是系统用户租赁的测试结果图。根据上述几个图测试结果表明：功能测试验中各模块之间的协作还算成功，在系统的功能性能方面和安全性方面都基本上达到了预期的需要效果。本文通过对系统的功能进行测试，发现系统较好的完成了功能需求，满足了用户操作要求，实现了系统预期的易用性，稳定性等目标。



图6-1游客注册页面

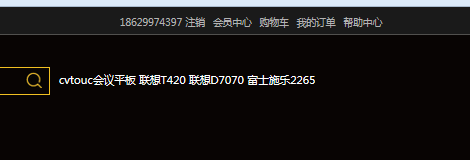


图6-3登录成功



图6-4租赁过程



图6-5租赁过程

## 5.4 性能测试

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。[负载测试](http://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E6%B5%8B%E8%AF%95)和[压力测试](http://baike.baidu.com/item/%E5%8E%8B%E5%8A%9B%E6%B5%8B%E8%AF%95)都属于性能测试，两者可以结合进行。通过[负载测试](http://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E6%B5%8B%E8%AF%95)，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。[压力测试](http://baike.baidu.com/item/%E5%8E%8B%E5%8A%9B%E6%B5%8B%E8%AF%95)是通过确定一个系统的瓶颈或者不能接受的性能点，来获得系统能提供的最大服务级别的测试。

本文对快租系统进行性能测试的主要目标是为了对系统的可靠性，正确性和效率作出有效验证。可靠性，正确性和效率在软件系统性能评价起着很重要作用，三者测试主要是通过对软件系统在网络或者其他测试环境下的运作。具体测试结果如下图6-6所示。

# 第6章 结论

## 6.1 总结

随着互联网的飞速发展，人们的生活变得越来越丰富。与此同时，资源的有限性牢牢禁锢着我们对资源的利用，如何利用现有的手段提高资源的实用效率已成为人们迫切需要考虑的问题。互联网+的概念为我们解决问题提供了一个良好的思路，于是便有了此次课题的开发灵感。

文中从题目的开发背景开始详细的论述了系统的开发过程，包括可行性分析、需求分析、概要设计、详细设计、系统测试。可行性分析从技术可行性开始，之后又详细的论述了经济可行性、技术可行性以及两个方案的对比得出开发框架。需求分析主要从系统的数据流、系统包图、系统重要类图进行详细论述。过程论述详细论述了系统的设计与实现，包括架构设计、功能设计数据库设计。而系统的分析方法主要采用黑盒测试的测试方法，设计出了详细的测试用例，在测试用例的基础上对系统进行了功能性测试，达到了预期效果。

通过几个月的毕业设计，终于完成了网上租赁网站的全部工作。此次开发我采用了较为新颖的SSM开发框架，改变了以往的做法，用注入和注解的方式来简化程序，使各层的代码分布在不同的Java文件中，提高了代码利用率。在技术选择上，我选用了当前流行的SSM框架，实现了典型的MVC三层结构以及J2EE四层模型，提高了系统的内聚性，降低了系统的耦合性。另外，我还使用了当前十分流行的Ajax技术实现了对前台页面的控制，以及与服务器的异步交换功能。最后，根据系统的需求分析和总体设计，对系统进行了详细编程实现。最终将Spring事物、SpringMVC模式、MyBatis技术、Ajax技术、AOP、Maven项目管理技术和ORM工具用到系统的开发中去，实现了此次课题，较好的完成了预期目标。

## 6.2 展望

经过自己的努力，在一系列的研究、设计工作之后，一个基于SSM的网上租赁网站开发完成，但由于时间和能力有限，开发出的系统并不完美，还有很大的改进空间。

1. 租赁过程实际是一个相当复杂的过程，其背后涉及的知识理论和技术要求远不止如此。占有欲则是我们积蓄考虑和解决的重要课题。如何通过技术完善机制也应当纳入问题的研究之中。
2. 本系统属于B/S结构，这本身就还应继续研究。现在移动应用盛行，并且其已成为抢占市场的重要工具。开发出面向移动端的客户端似乎具有不小的诱惑力。
3. 系统的功能并不完整。关于电子商务类的网站开发实际上是需要庞大的数据支撑的，但由于人员和技术有限，并不能完成一个电商网站所需要的全部功能，。这一点是系统的最大缺陷。
4. 开发环境和部署环境简陋。本次开发的硬件环境只是普通的笔记本电脑。他不像服务器可以承载较为大量的客户访问，在访问人数超出一定个数之后，系统可能会出现卡顿、死机、服务器崩溃等严重问题。这是开发者不愿看到的。

# 致 谢

毕业设计的完成也就意味着大学生涯的即将结束。从一无所知的懵懂少年到胸有成竹，对未来充满信心的有志之士，首先，我要感谢北华大学为我提供优良的学习环境，让我对知识的渴望有了现实的依托，“崇德尚学、自强力行”将长长久久地回响在我耳际！其次，我要感谢计算机科学与技术学院这四年来的培育、滋养，在我们可爱的老师的辛勤教育之下，将莘莘学子培育成可造之材！

在毕业论文完成之际，我要向我的指导老师曹宏宇表示深深地感谢！在我毕业设计的开发期间，老师给予了无私的关怀和帮助，虽然自己并没有提出什么建设性问题，老师还是不厌其烦的耐心解释。在论文执写期间，老师将我们叫到他的办公室，仔仔细细的为我们讲解论文的写作格式，写作内容。清晰的为我们安排了写作流程。

大恩不言谢，唯有在今后的学习中牢记指导和教诲，继续努力，不断进步！

另外，我还要感谢我619可爱的室友们在毕设开发过程中给予的帮助，他们将自己的项目经验孜孜不倦的传授给我，这对此次的项目开发起到了非常大的作用，祝他们在今后的生活中事事顺心，在工作中平步青云，在各自的领域里熠熠生辉！

# 参考文献

1. 杨群.基于SSM的高校排课系统的研究与应用[D].

# 附 录

软件使用说明书

FLASHRENT系统在能满足用户需求之前需要相应的软件和硬件要求。由于此次开发采用的是PC机，所以在此仅对PC机的使用作出如下说明：

首先为电脑安装MySQL服务器、Tomcat服务器、JDK1.8、Myeclipse2016等相关软件。然后在配置好Myeclipse之后为，将开发项目部署到Tomcat服务器中。到此完成了项目的部署工作。

在完成项目的部署之后，系统的用户即会员用户和管理员就可以利用系统完成各自的业务了。

用户在浏览器中输入进入网站的URL，具体内容为localhost：Tomcat端口号/graduationProject，回车后进入系统的欢迎页面。页面的右上角有一各登录按钮，已注册过的会员和管理员可以点击登录进入到系统完成各自的业务，其他用户只能浏览网页或注册成为会员。

管理员进入系统后在操作页面中选择要处理的业务，根据页面文字提示即可完成相关业务。

会员登录之后可以完成租赁、信息管理等业务。租赁过程为在租品页中点击想要租赁的商品，进入租赁信息填写界面，完成信息填写之后可选择加入购物车或立即下单操作，根据自己意向选择想要的操作，在相关向导下即可完成租赁操作。至于其他业务在此不作详细描述，本系统采用的是B/S架构，操作简单，界面友好，会上网和识字的用户基本都能准确完成利用系统完成业务。

系统详细代码

SSM框架的精髓在于其位置文件，也就是配置文件的存在大大简化了系统的代码数量，提高了代码的利用率，下面是主要配置文件的详细代码

pom.xml配置文件：

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>augusyoung.graduationProject</groupId>

<artifactId>graduationProject</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>graduationProject Maven Webapp</name>

<url>http://maven.apache.org</url>

<properties>

<!-- spring版本号 -->

<spring.version>4.0.2.RELEASE</spring.version>

<!-- mybatis版本号 -->

<mybatis.version>3.2.6</mybatis.version>

<!-- log4j日志文件管理包版本 -->

<slf4j.version>1.7.7</slf4j.version>

<log4j.version>1.2.17</log4j.version>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<!-- 表示开发的时候引入，发布的时候不会加载此包 -->

<scope>test</scope>

</dependency>

<!-- spring核心包 -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-oxm</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- mybatis核心包 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>${mybatis.version}</version>

</dependency>

<!-- mybatis/spring包 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

<!-- 导入java ee jar 包 -->

<dependency>

<groupId>javax</groupId>

<artifactId>javaee-api</artifactId>

<version>7.0</version>

</dependency>

<!-- 导入Mysql数据库链接jar包 -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.30</version>

</dependency>

<!-- 导入dbcp的jar包，用来在applicationContext.xml中配置数据库 -->

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.2.2</version>

</dependency>

<!-- JSTL标签类 -->

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- 日志文件管理包 -->

<!-- log start -->

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>${log4j.version}</version>

</dependency>

<!-- 格式化对象，方便输出日志 -->

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>fastjson</artifactId>

<version>1.1.41</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${slf4j.version}</version>

</dependency>

<!-- log end -->

<!-- 映入JSON -->

<dependency>

<groupId>org.codehaus.jackson</groupId>

<artifactId>jackson-mapper-asl</artifactId>

<version>1.9.13</version>

</dependency>

<!-- 上传组件包 -->

<dependency>

<groupId>commons-fileupload</groupId>

<artifactId>commons-fileupload</artifactId>

<version>1.3.1</version>

</dependency>

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.4</version>

</dependency>

<dependency>

<groupId>commons-codec</groupId>

<artifactId>commons-codec</artifactId>

<version>1.9</version>

</dependency>

</dependencies>

<build>

<finalName>graduationProject</finalName>

</build>

</project>

Spring-mybatis.xml配置文件

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.springframework.org/schema/p"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.1.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.1.xsd

http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd">

<!-- 自动扫描 -->

<context:component-scan base-package="gproject" />

<!-- 引入配置文件 -->

<bean id="propertyConfigurer"

class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">

<property name="location" value="classpath:jdbc.properties" />

</bean>

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"

destroy-method="close">

<property name="driverClassName" value="${driver}" />

<property name="url" value="${url}" />

<property name="username" value="${username}" />

<property name="password" value="${password}" />

<!-- 初始化连接大小 -->

<property name="initialSize" value="${initialSize}"></property>

<!-- 连接池最大数量 -->

<property name="maxActive" value="${maxActive}"></property>

<!-- 连接池最大空闲 -->

<property name="maxIdle" value="${maxIdle}"></property>

<!-- 连接池最小空闲 -->

<property name="minIdle" value="${minIdle}"></property>

<!-- 获取连接最大等待时间 -->

<property name="maxWait" value="${maxWait}"></property>

</bean>

<!-- spring和MyBatis完美整合，不需要mybatis的配置映射文件 -->

<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="dataSource" ref="dataSource" />

<!-- 自动扫描mapping.xml文件 -->

<property name="mapperLocations" value="classpath:gproject/mapping/\*.xml"></property>

</bean>

<!-- DAO接口所在包名，Spring会自动查找其下的类 -->

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="basePackage" value="gproject.dao" />

<property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"></property>

</bean>

<!-- (事务管理)transaction manager, use JtaTransactionManager for global tx -->

<bean id="transactionManager"

class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

<property name="dataSource" ref="dataSource" />

</bean>

</beans>

Spring-mvc.xml配置文件

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.springframework.org/schema/p"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.1.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.1.xsd

http://www.springframework.org/schema/mvc

http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd">

<!-- 自动扫描该包，使SpringMVC认为包下用了@controller注解的类是控制器 -->

<context:component-scan base-package="gproject.controller" />

<!--避免IE执行AJAX时，返回JSON出现下载文件 -->

<bean id="mappingJacksonHttpMessageConverter"

class="org.springframework.http.converter.json.MappingJacksonHttpMessageConverter">

<property name="supportedMediaTypes">

<list>

<value>text/html;charset=UTF-8</value>

</list>

</property>

</bean>

<!-- 启动SpringMVC的注解功能，完成请求和注解POJO的映射 -->

<bean class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter">

<property name="messageConverters">

<list>

<ref bean="mappingJacksonHttpMessageConverter" /> <!-- JSON转换器 -->

</list>

</property>

</bean>

<!-- 定义跳转的文件的前后缀 ，视图模式配置-->

<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<!-- 这里的配置我的理解是自动给后面action的方法return的字符串加上前缀和后缀，变成一个 可用的url地址 -->

<property name="prefix" value="/" />

<property name="suffix" value=".jsp" />

</bean>

<!-- 配置文件上传，如果没有使用文件上传可以不用配置，当然如果不配，那么配置文件中也不必引入上传组件包 -->

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<!-- 默认编码 -->

<property name="defaultEncoding" value="utf-8" />

<!-- 文件大小最大值 -->

<property name="maxUploadSize" value="10485760000" />

<!-- 内存中的最大值 -->

<property name="maxInMemorySize" value="40960" />

</bean>

</beans>

Jdbc.properties文件

driver=com.mysql.jdbc.Driver

url=jdbc:mysql://localhost:3306/gproject?useUnicode=true&characterEncoding=UTF8&zeroDateTimeBehavior=convertToNull&serverTimezone=GMT&useSSL=true

username=root

password=9999999.