

综合课程设计报告

题目 自行车租赁收费系统设计与实现

二级学院 计算机科学与工程学院

专 业 软件工程

班 级 1130308030

学生姓名 邵小镔 学号 11303080307

指导教师 张小川 职称 教授

时 间 2016年9月

**目录**

[1. 概述 3](#_Toc18780)

[1.1开发背景 3](#_Toc2670)

[1.2 需求分析 3](#_Toc11445)

[1.2.1 需求综述 3](#_Toc32155)

[1.2.2 系统功能需求 4](#_Toc27432)

[1.2.3 其他非功能性需求 4](#_Toc11124)

[1.3 设计原则 4](#_Toc4772)

[2. 系统构成 5](#_Toc23560)

[2.1 系统架构概述 5](#_Toc23921)

[2.2 操作系统 5](#_Toc28968)

[2.3 数据库平台 5](#_Toc7197)

[2.4关键技术 6](#_Toc7974)

[2.4.1 关键技术概述 6](#_Toc10178)

[2.4.2 Spring技术 6](#_Toc16507)

[2.4.3 mybatis技术 7](#_Toc26518)

[2.4.4 射频卡技术 8](#_Toc17312)

[3. 系统流程 8](#_Toc3045)

[3.1 采购车辆 8](#_Toc27801)

[3.2 车辆回收 9](#_Toc29155)

[3.3 车辆维修 9](#_Toc12298)

[3.4 用户租借车辆 10](#_Toc32497)

[4. 功能设计 11](#_Toc23507)

[4.1 综合功能概览 11](#_Toc30553)

[4.1.1 后台功能 12](#_Toc8565)

[4.1.2 前台功能 12](#_Toc13189)

[4.2 功能模块详细用例图 13](#_Toc30940)

[4.2.1 登陆 14](#_Toc28724)

[4.2.2 注册 14](#_Toc17985)

[4.2.3 申办会员 14](#_Toc6478)

[4.2.4 个人信息维护 14](#_Toc17465)

[4.2.5 租赁历史信息查询 14](#_Toc11318)

[4.2.6 费用汇总 14](#_Toc31539)

[4.2.7 租车 14](#_Toc30159)

[4.2.8 意见建议反馈 14](#_Toc6643)

[5数据库设计 14](#_Toc20881)

[5.1 系统概念模型设计 14](#_Toc24263)

[5.1.1 系统实体描述 14](#_Toc31329)

[5.1.2 系统E-R图 14](#_Toc32655)

[5.2 数据表详细 14](#_Toc17159)

[用户表（RbUser） 15](#_Toc11976)

[管理员/工作人员表（Operator） 15](#_Toc31608)

[一卡通/会员信息表（RbCard） 16](#_Toc3647)

[自行车信息表（BikeInfo） 17](#_Toc12624)

[自行车型表（BikeType） 17](#_Toc9245)

[车辆租借记录表（RentalBikeHistory） 18](#_Toc11561)

[供应商表（Supplier） 19](#_Toc19802)

[赔偿记录表（CompensationRecord） 19](#_Toc23484)

[采购计划表（PurchasePlan） 19](#_Toc16614)

[采购计划批次表（PurchasePlanBikeBatch） 20](#_Toc2281)

[回收计划表（RecyclePlan） 20](#_Toc4362)

[维修计划表（MaintenancePlan） 21](#_Toc17433)

[维修记录表（MaintenanceRecord） 21](#_Toc17608)

[日财务表（DayFinancialSheet） 22](#_Toc16536)

[月财务表（MonthFinancialSheet） 23](#_Toc12613)

[年财务表（YearFinancialSheet） 23](#_Toc22999)

[意见及建议表（Suggestion） 24](#_Toc15236)

[仓库表（Storage） 25](#_Toc25278)

[6. 系统实现 25](#_Toc11518)

[6.1 系统实现总体描述 26](#_Toc24682)

[6.2 具体模块实现界面 26](#_Toc22991)

[6.2.1 登录界面（客户端） 26](#_Toc5140)

[6.2.2 登出界面（客户端） 26](#_Toc3484)

[6.2.3 租车界面（客户端） 26](#_Toc5360)

[6.2.4 查看租车记录界面（客户端） 26](#_Toc7262)

[6.2.5 个人信息界面（客户端） 26](#_Toc582)

[6.2.6 查看赔款界面（客户端） 26](#_Toc7851)

[7. 系统测试 26](#_Toc22292)

[7.1 系统测试综述 26](#_Toc23872)

[7.2 系统测试详细 27](#_Toc29687)

[7.2.1 登录模块测试用例（客户端） 27](#_Toc19921)

[7.2.2 首页测试用例（客户端） 27](#_Toc22834)

# 1.绪论

## 1.1开发背景

自行车租赁收费管理系统，凭借其价廉、便捷、健身、节能、环保等诸多优势，可以有效补充和延伸常规公交，成为全球不少国家和地区极力推广的公益性事业。经济便捷的自行车便成为广大人民的最佳选择。由此，提议开发一套完整的自行车租赁收费管理系统，以此解决广大人民群众对自行车需求、保管等方面的问题。在本系统中，自行车主要来自统一管理方的购买和闲置车辆回收等途径。用户可以凭借绑定的一卡通刷卡租借归还自行车，非常简单便利。该自行车租赁是为广大用户开放的。通过系统的统一管理，使得自行车的使用率达到最大化，并且能够满足用户自行车需求，降低了自己购买的成本以及方便了广大用户的旅游出行。

用户取车时刷卡或者登陆系统进行进行，系统进行记录，并记录取车时间，游客还车时重新刷卡或者登陆系统确认归还，根据收费标准收取费用。 系统自动记录数据，统计金额，打印报表，挂失出来等，管理使用简便快捷！

## 1.2 系统设计原则

(1)可靠性

用软件系统规模越做越大越复杂，其可靠性越来越难保证。应用本身对系统运行的可靠性要求越来越高，软件系统的可靠性也直接关系到设计自身的声誉和生存发展竞争能力。软件可靠性意味着该软件在测试运行过程中避免可能发生故障的能力，且一旦发生故障后，具有解脱和排除故障的能力。软件可靠性和硬件可靠性本质区别在于：后者为物理机理的衰变和老化所致，而前者是由于设计和实现的错误所致。故软件的可靠性必须在设计阶段就确定，在生产和测试阶段再考虑就困难了。

(2)健壮性

健壮性又称鲁棒性，是指软件对于规范要求以外的输入能够判断出这个输入不符合规范要求，并能有合理的处理方式。软件健壮性是一个比较模糊的概念，但是却是非常重要的软件外部量度标准。软件设计的健壮与否直接反应了分析设计和编码人员的水平。

(3)可修改性

要求以科学的方法设计软件，使之有良好的结构和完备的文档，系统性能易于调整。

(4)容易理解

软件的可理解性是其可靠性和可修改性的前提。它并不仅仅是文档清晰可读的问题，更要求软件本身具有简单明了的结构。这在很大程度上取决于设计者的洞察力和创造性，以及对设计对象掌握得透彻程度，当然它还依赖于设计工具和方法的适当运用。

1. 程序简便

(6)可测试性

可测试性就是设计一个适当的数据集合，用来测试所建立的系统，并保证系统得到全面的检验。

(7)效率性

软件的效率性一般用程序的执行时间和所占用的内存容量来度量。在达到原理要求功能指标的前提下，程序运行所需时间愈短和占用存储容量愈小，则效率愈高。

（8）标准化原则[1]

在结构上实现开放，基于业界开放式标准，符合国家和信息产业部的规范。

（9）先进性

满足客户需求，系统性能可靠，易于维护。

（10）可扩展性

软件设计完要留有升级接口和升级空间。对扩展开放，对修改关闭。

# 2.关键技术

## 2.1 关键技术概述

本系统采用了B/S架构，B/S结构即浏览器和服务器结构。它是随着Internet技术的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过WWW浏览器来实现，极少部分事务逻辑在前端(Browser)实现，但是主要事务逻辑在服务器端(Server)实现，形成所谓三层3-tier结构。B/S结构是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器（Browser），如Netscape Navigator或Internet Explorer，服务器安装Oracle、Sybase、Informix或 SQL Server等数据库。浏览器通过Web Server同数据库进行数据交互。 这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本(TCO)。

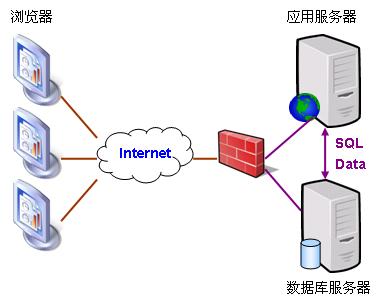


图1-1 b/s结构

## 2.2 Spring技术

spring是J2EE应用程序框架，是轻量级的IoC和AOP的容器框架，主要是针对javaBean的生命周期进行管理的轻量级容器，可以单独使用，也可以和Struts框架，ibatis框架等组合使用。

IoC(Inversion of Control)控制反转，对象创建责任的反转，在spring中BeanFacotory是IoC容器的核心接口，负责实例化，定位，配置应用程序中的对象及建立这些对象间的依赖。XmlBeanFacotory实现BeanFactory接口，通过获取xml配置文件数据，组成应用对象及对象间的依赖关系。

spring中有三种注入方式，一种是set注入，一种是接口注入，另一种是构造方法注入。

AOP面向切面编程

aop就是纵向的编程，比如业务1和业务2都需要一个共同的操作，与其往每个业务中都添加同样的代码，不如写一遍代码，让两个业务共同使用这段代码。

spring中面向切面变成的实现有两种方式，一种是动态代理，一种是CGLIB，动态代理必须要提供接口，而CGLIB实现是有继承。

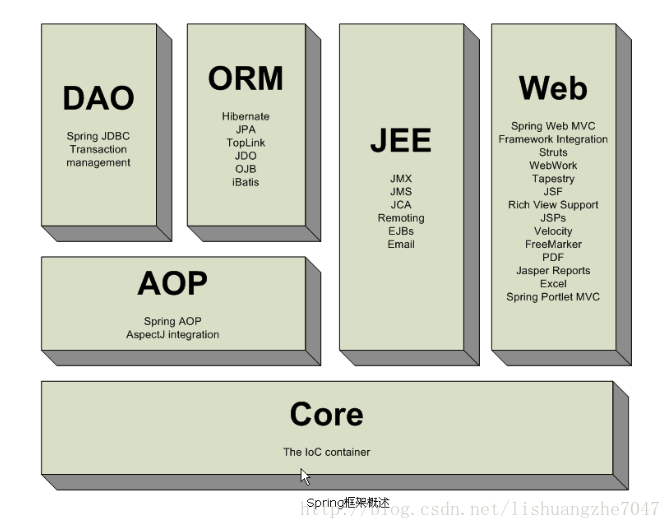


图2-1 spring结构

## 2.3 mybatis技术

MyBatis 是支持普通 SQL查询，存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML或注解用于配置和原始映射，将接口和 Java 的POJOs（Plain Old Java Objects，普通的 Java对象）映射成数据库中的记录。

每个MyBatis应用程序主要都是使用SqlSessionFactory实例的，一个SqlSessionFactory实例可以通过SqlSessionFactoryBuilder获得。SqlSessionFactoryBuilder可以从一个xml配置文件或者一个预定义的配置类的实例获得。用xml文件构建SqlSessionFactory实例是非常简单的事情。推荐在这个配置中使用类路径资源（classpath resource)，但你可以使用任何Reader实例，包括用文件路径或file://开头的url创建的实例。MyBatis有一个实用类----Resources，它有很多方法，可以方便地从类路径及其它位置加载资源。

MyBatis的前身叫iBatis，本是apache的一个开源项目, 2010年这个项目由apache software foundation 迁移到了google code，并且改名为MyBatis。MyBatis是支持普通SQL查询，存储过程和高级映射的优秀持久层框架。MyBatis消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis使用简单的XML或注解用于配置和原始映射，将接口和Java的POJOs（Plan Old Java Objects，普通的Java对象）映射成数据库中的记录。

API接口层：提供给外部使用的接口API，开发人员通过这些本地API来操纵数据库。接口层一接收到调用请求就会调用数据处理层来完成具体的数据处理。

数据处理层：负责具体的SQL查找、SQL解析、SQL执行和执行结果映射处理等。它主要的目的是根据调用的请求完成一次数据库操作。

基础支撑层：负责最基础的功能支撑，包括连接管理、事务管理、配置加载和缓存处理，这些都是共用的东西，将他们抽取出来作为最基础的组件。为上层的数据处理层提供最基础的支撑。

## 2.4 射频卡技术

智能卡发展到今天，经历了磁卡、接触式IC卡、射频卡三个阶段。由于射频卡有接触IC卡无可比拟的优势，射频卡类的产品必然会取代接触IC卡类的产品，就像IC卡取代磁卡一样。是一个技术替代过程，不为人的意愿左右。智能卡在门锁中的应用也同样经历了三个阶段。磁卡门锁的传统市场已逐渐消失，接触式IC卡门锁的使用已经相当普遍，射频卡门锁目前全国的正处在上升期。

射频卡技术的七大优势

1、读取方便快捷

数据的读取无需光源，甚至可以透过外包装来进行。有效识别距离更大，采用自带电池的主动标签时，有效识别距离可达到30米以上；

2、数据容量大

数据容量最大的二维条形码（PDF417），最多也只能存储2725个数字；若包含字母，存储量则会更少；RFID标签则可以根据用户的需要扩充到数10K；

3、标签数据可动态更改

利用编程器可以向标签写入数据，从而赋予RFID标签交互式便携数据文件的功能，而且写入时间相比打印条形码更少；

4、更好的安全性

不仅可以嵌入或附着在不同形状、类型的产品上，而且可以为标签数据的读写设置密码保护，从而具有更高的安全性；

5、识别速度快

标签一进入磁场，解读器就可以即时读取其中的信息，而且能够同时处理多个标签，实现批量识别；

6、使用寿命长，应用范围广

其无线电通信方式，使其可以应用于粉尘、油污等高污染环境和放射性环境，而且其封闭式包装使得其寿命大大超过印刷的条形码；

7、动态实时通信

标签以与每秒50～100次的频率与解读器进行通信，所以只要RFID标签所附着的物体出现在解读器的有效识别范围内，就可以对其位置进行动态的追踪和监控。

# 3. 需求分析

## 3.1 需求综述

软件需求可以用若干种方法来表达，如通过输入、输出说明；使用代表性的例子；用规范化的模型。开发者应尽可能地使用模型的方式，因为这是表达复杂需求的精确和有效的方法。比如用统一建模语言（UML）来描述需求。 编写需求分析报告的要求

1. 无歧义性 对最终产品的每一个特性用某一术语描述；若某一术语在某一特殊的行文中使用时具有多种含义，那么应对该术语的每种含义做出解释并指出其适用场合。
2. 完整性 需求分析报告应该包括全部有意义的需求，无论是关系到功能的、性能的、设计约束的、还是关系到外部接口方面的需求；对所有可能出现的输入数据的响应予以定义，要对合法和非合法的输入值的响应做出规定；填写全部插图、表、图示标记等；定义全部术语和度量单位。
3. 可验证性 需求分析报告描述的每一个需求应是可以验证的。可以通过一个有限处理过程来检查软件产品是否满足需求。
4. 一致性 在需求分析报告中的各个需求的描述不能互相矛盾。
5. 可修改性 需求分析报告应具有一个有条不紊、易于使用的内容组织；没有冗余，即同一需求不能在需求分析报告中出现多次。
6. 可追踪性 每一个需求的源流必须清晰，在进一步产生和改变文件编制时，可以方便地引证每一个需求。 g．运行和维护阶段的可使用性 需求分析报告必须满足运行和维护阶段的需要。在需求分析报告要写明功能的来源和目的。

## 3.2 系统功能需求

自行车租赁收费管理系统着重于降低工作人员工作压力，减少繁重业务，实现用户友好界面，加强账务分析。该自行车系统主要要业务包括山地车和常用车出租业务，该系统的自行车包括有，普通轻便车，山地车两大类。其中轻便车分文男式和女式，山地车包括GIANT，美利达，蒙太奇。其中GIANT包括有680到770的车型，美利达和蒙太奇也有不同层次的车。由于这些车的性能不同，该车行通常会根据不同的车型来确定车的租赁。例如，轻便车通常为价格要便宜些。山地车的购入成本和维护成本偏高，租赁价格就要高一些。而顾客也包括两类。一类是会员客户，一类是非会员客户。会员客户租赁自行车有一定的优惠和折扣。管理员也可以根据实际需要设置会员的等级条件和优惠比例。

**系统主要参与人信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **名称** | **说明** | **系统操作要求** |
| UP-01 | 非会员用户 | 没有办理会员业务的顾客。这类顾客一般不会有优惠或折扣。对于非会员租赁顾客而言需要寄压身份证等有效证件！ | 1. 可以很方便的申请办理会员 2. 可以很方便快速找到相关车信息。 3. 租赁时，可以方便建立档案，押金以及寄压证件情况！ |
| UP-02 | 会员用户 | 办理会员业务的顾客。车行对这类顾客又一定优惠或折扣。 | 1. 可以方便快速查找相关产品信息！ 2. 租赁自行车时可以查看折扣信息！ |
| UP-03 | 超级管理员（老板） | 掌控全局，进行员工资料的管理，拥有最高权限 | 1. 方便进行人员的管理 2. 获得财务信息快捷 |
| UP-04 | 仓库管理员 |  |  |
| UP-05 | 服务部门 | 负责自行车的摆放以及布置效果。同时承担顾客的接待工作，受理顾客的业务 租车还车 | 1. 方便的对会员提供服务  2.帮会员帮查询信息  3.能通过系统生成用户统计报表上报管理者 |
| UP-06 | 财务部门 | 负责统计财务报表，员工工资发放以及 与供应商进行财务结算 | 方便的进行各种财务计算，定期将产生财务报表 |
| UP-07 | 采购维护部门 | 负责接受供应商提供的自行车产产品目录单。导入到系统中进行管理。通过供应商采购产品。同时应对自行车的损坏程度做记录，以便及时修理以及采购。 | 1 通过计算机能方便快捷的查看车行存车信息。 2. 能快捷的导入供应商的目录清单信息，进行比较 3. 采购回的车辆能方便的进行录入管理。 |
| UP-08 | 供应商 | 提供车辆 | 1. 快速结账 |

## 3.2.1 客户端用户管理

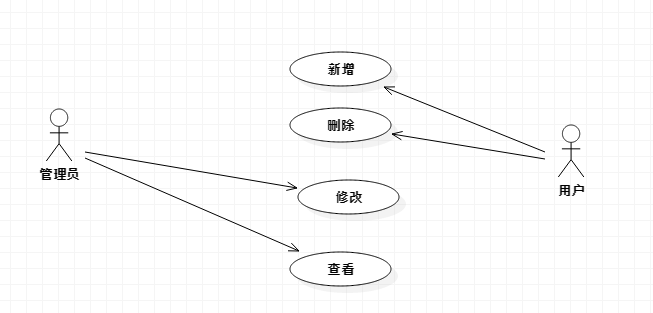


图3-1 客户端用户管理

用例描述：

|  |  |
| --- | --- |
| **用例名称** | 查询用户信息 |
| **执行者** | 用户 |
| **用例说明** | 用户查询个人信息 |
| **前置条件** | 用户登录客户端后进入个人信息界面 |
| **后置条件** | 显示用户信息 |
| **事件流** | |
| **基本事件流** | |
| 1.用户登录到系统  2.通过界面选择“个人信息”  3.在界面中查看用户个人信息 | |
| **扩展事件流** | |
| 1.用户登录失败（账号无效或密码错误），输出相应信息信息；  2.用户操作错误或不符合规定。。 | |
| **其它说明** | 无 |

## 3.2.2 管理员端用户管理

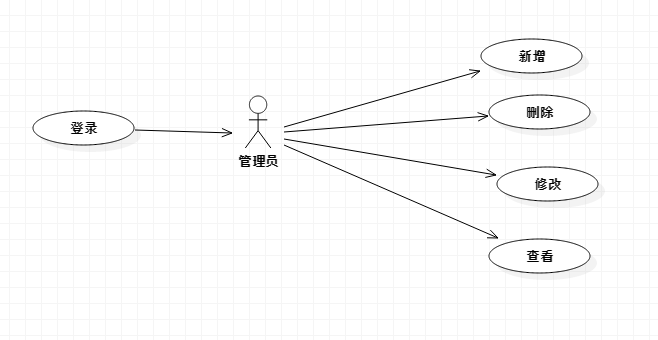


图3-2 管理员端用户管理

|  |  |
| --- | --- |
| **用例名称** | 查询用户信息 |
| **执行者** | 管理员 |
| **用例说明** | 管理员查询用户的信息 |
| **前置条件** | 系统管理员登陆后台系统管理网页 |
| **后置条件** | 显示用户信息 |
| **事件流** | |
| **基本事件流** | |
| 1.管理员登录到系统  2.通过管理界面选择“查询用户信息”  3.在用户列表中选择需要查看的用户或者输入用户呢称或帐号查看用户信息 | |
| **扩展事件流** | |
| 1.系统管理员登录失败（账号无效或密码错误），提示相应信息；  2.系统管理员操作错误或不符合规定，提示相应信息。 | |
| **其它说明** | 无 |

## 3.2.3 租赁历史管理

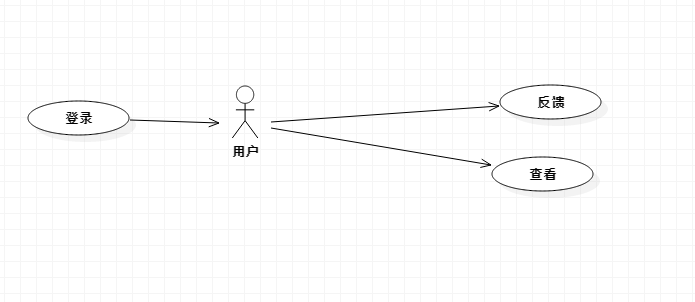


图3-3 租赁历史管理

|  |  |
| --- | --- |
| **用例名称** | 用户使用租赁历史信息 |
| **执行者** | 用户 |
| **用例说明** | 用户查询反馈租赁历史 |
| **前置条件** | 用户登陆 |
| **后置条件** | 反馈信息交由管理员审核 |
| **事件流** | |
| **基本事件流** | |
| 1.用户登录到系统  2.通过客户端页面选择租赁历史  3.填写反馈信息，点击提交  4.然后交由管理员审核反馈信 | |
| **扩展事件流** | |
| 1.用户登录失败（账号无效或密码错误），提示相应信息；  2.用户操作错误或不符合规定，提示相应信息。 | |
| **其它说明** | 无 |

## 3.2.4 租车流程-用户

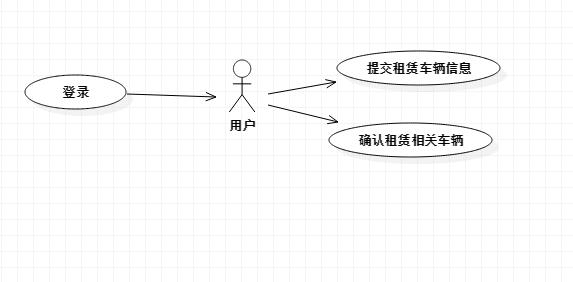


图3-4 租车流程-用户

|  |  |
| --- | --- |
| **用例名称** | 用户租车 |
| **执行者** | 用户 |
| **用例说明** | 用户键入车辆信息，进行租车 |
| **前置条件** | 用户登陆 |
| **后置条件** | 系统提示租车成功，服务人员解锁自行车 |
| **事件流** | |
| **基本事件流** | |
| 1.用户登录到系统  2.通过客户端页面选择租车  3.填写车辆信息，点击提交  4.系统反馈车辆是否可用  5.用户点击确认，进行租车。 | |
| **扩展事件流** | |
| 1.用户登录失败（账号无效或密码错误），提示相应信息；  2.用户操作错误或不符合规定，提示相应信息。 | |
| **其它说明** | 无 |

## 3.2.5 还车-用户

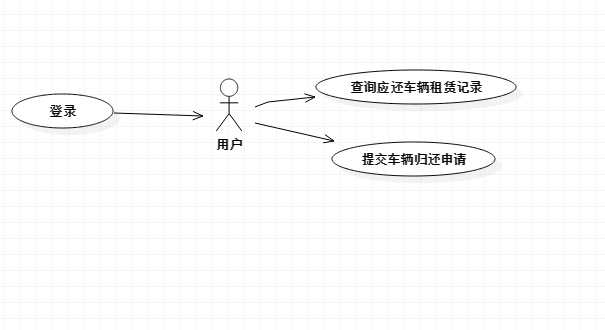


图3-5 还车-用户

|  |  |
| --- | --- |
| **用例名称** | 用户还车 |
| **执行者** | 用户 |
| **用例说明** | 用户查询应还车辆信息，进行还车申请 |
| **前置条件** | 用户登陆 |
| **后置条件** | 系统提示申请成功，服务人员验收自行车 |
| **事件流** | |
| **基本事件流** | |
| 1.用户登录到系统  2.通过客户端页面选择还车  3.查询车辆租赁历史信息，选择条目点击申请还车提交  4.系统反馈车辆还车申请已提交，等待管理人员验车  5.服务人员验收车辆。 | |
| **扩展事件流** | |
| 1.用户登录失败（账号无效或密码错误），提示相应信息；  2.用户操作错误或不符合规定，提示相应信息。 | |
| **其它说明** | 无 |

## 3.2.7 还车-管理员

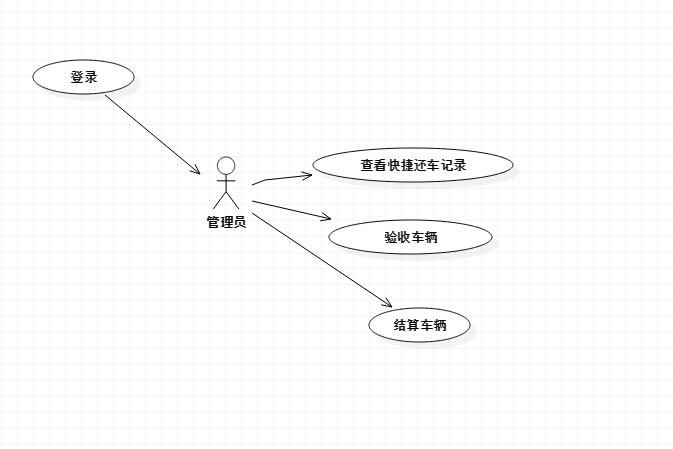


图3-6 还车-管理员

|  |  |
| --- | --- |
| **用例名称** | 管理员验车 |
| **执行者** | 管理员 |
| **用例说明** | 管理员查看快捷还车的信息，进行车辆结算 |
| **前置条件** | 管理员登录 |
| **后置条件** | 服务人员验收成功，服务人员锁定自行车 |
| **事件流** | |
| **基本事件流** | |
| 1.用户登录到系统  2.通过客户端页面选择还车  3.查询车辆租赁历史信息，选择条目点击申请还车提交  4.系统反馈车辆还车申请已提交，等待管理人员验车  5.服务人员验收车辆。 | |
| **扩展事件流** | |
| 1.管理员登录失败（账号无效或密码错误），提示相应信息；  2.管理员操作错误或不符合规定，提示相应信息。 | |
| **其它说明** | 无 |

## 3.3 其他非功能性需求

## 3.3.1 可靠性

系统需要保证会员数据尤其是充值信息以及寄压证件的安全性，系统能长时间稳定的运行

## 3.3.2 可用性

系统具备使用方便，系统交互简介可行，查询信息和统计方便工作人员和用户。管理人员能方便的查看报表。系统界面美观，按钮方便操作，系统的整体色调符合当前社会的审美观。

## 3.3.3 有效性

系统使用模块化设计，易于管理和维护为以后的升级做保障，数据库兼容不同的数据，以及录入数据库的快捷和方便。

## 3.3.4 可移植性

系统具备多平台使用，用户可通过手机查询用户信息，客户机使用触摸操作。系统预留硬件操作接口，为后续开发打下坚实的基础。

# 4.系统架构

## 4.1 系统架构概述

MVC(Module-View-Controller)是一种软件设计模式，其中M(Module)指业务模型，V(View)指用户界面，C(Controller)则是控制器，使用MVC的目的是将Module和View的实现代码分离，从而达到使同一个程序可以使用不同的表现形式的目的。

MVC模式将一个应用分为三层，即模型层、试图层、控制器层。每层之间相互独立，这大大提高了系统的维护性和可扩展性。MVC模式能够对Html、Css和JavaScript的进行完全控制。通常模型对象负责在数据库中存取数据，视图处理数据显示，控制器处理用户交互。采用MVC设计模式的优点如下：

1、高内聚。系统各个功能内部高度聚合，系统内部的设计更加的紧凑，逻辑性更强，不松散。

2、低耦合。系统的各个功能之间以组件的形式进行解耦和，确保各个模块之间相互独立。

3、易维护。由于面向对象的设计方式，各个功能的独立，是的代码维护起来非常方便，能够有效的降低成本。

4、可移植。系统整体是以Web应用插件的方式设计，方便系统应用与大多数的Web应用系统中。

## 4.2 操作系统

系统部署在Windows server 上，使用java语言编写，也可移植到Linux 和Unix 系统上。客户端设备为触摸显示装置，连接到部署的总平台上。每个区域可独立使用整套系统，也可以分布式部署在各个地区，实现分布式管理经营。

## 4.3 数据库平台

数据库采用mysql平台，MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 最流行的关系型数据库管理系统，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是一种关联数据库管理系统，关联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，它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，一般中小型网站的开发都选择 MySQL 作为网站数据库。

* Mysql是开源的，所以你不需要支付额外的费用。
* Mysql支持大型的数据库。可以处理拥有上千万条记录的大型数据库。
* MySQL使用标准的SQL数据语言形式。
* Mysql可以允许于多个系统上，并且支持多种语言。这些编程语言包括C、C++、Python、Java、Perl、PHP、Eiffel、Ruby和Tcl等。
* Mysql对PHP有很好的支持，PHP是目前最流行的Web开发语言。
* MySQL支持大型数据库，支持5000万条记录的数据仓库，32位系统表文件最大可支持4GB，64位系统支持最大的表文件为8TB。
* Mysql是可以定制的，采用了GPL协议，你可以修改源码来开发自己的Mysql系统。

## 4.3系统流程

系统流程图是描绘系统物理模型的传统工具。它的基本思想是用图形符号以黑盒子形式描绘系统里面的每个部件(程序、文件、数据库、表格、人工过程等），表达信息在各个部件之间流动的情况。

在本系统中主要流程图为采购车辆、车辆回收、车辆维修、用户租借车辆几个流程。

## 4.3.1 采购车辆

参与者：管理人员、工作人员

目标：管理人员/工作人员制定采购计划，从供应商获取车辆，编制进系统记录中。

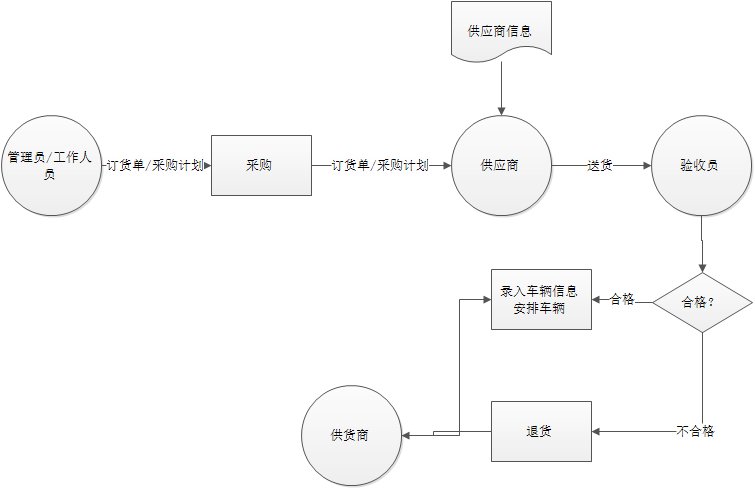


图4-1 采购车辆

## 4.3.2 车辆回收

参与人员：管理人员/工作人员

目标：管理人员/工作人员制定车辆的回收计划，验收人员进行车辆考察进行价格估算，符合标准车辆出售给回收机构。

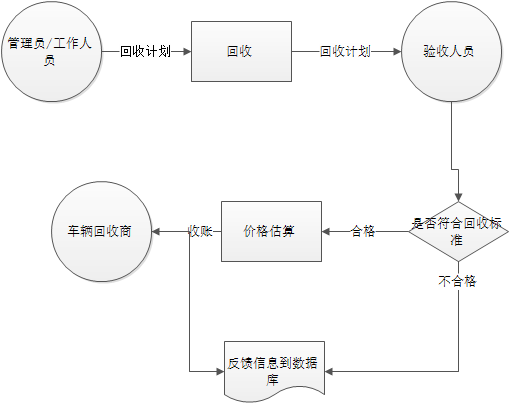


图4-2 车辆回收

## 4.3.3 车辆维修

参与人员：维修人员，管理人员，用户

目标：车辆维修由两个途径而来，一是用户归还的时候车辆受损进行赔偿和修理，二是管理人员/工作人员制定统一的车辆维修计划进行维修。

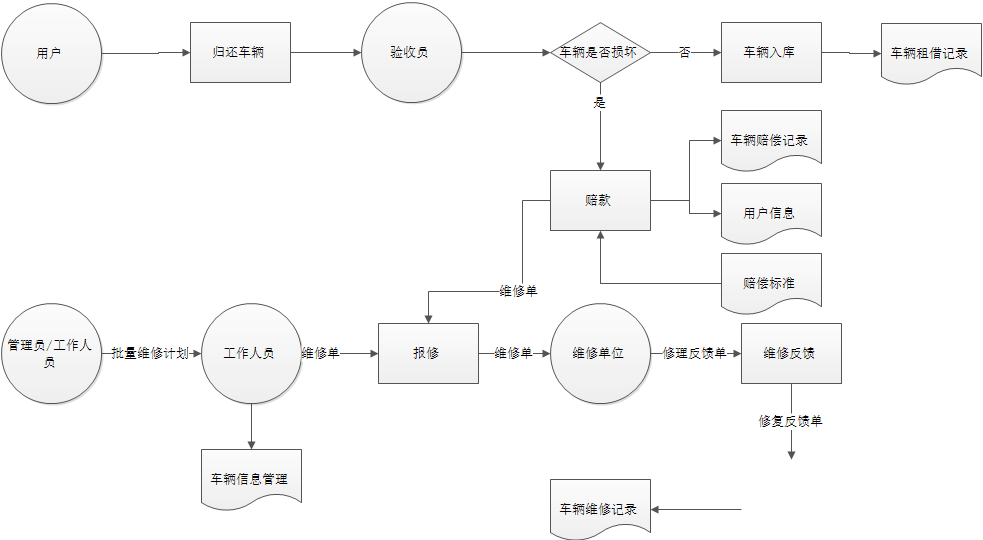


图4-3 车辆维修

## 4.3.4 用户租借车辆

参与人员：用户，服务人员

目标：用户租借车辆的过程为登陆系统或者使用一卡通进行快速登陆和租借车辆，绑定一卡通默认为会员账户。

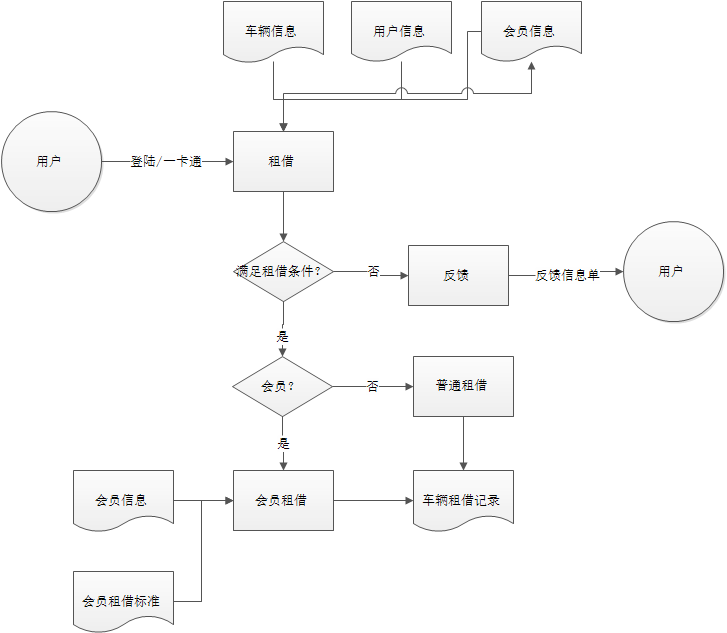


图4-4 用户租车

## 4.4 功能架构

## 4.4.1综合功能概览

## 4.4.1.1 后台功能

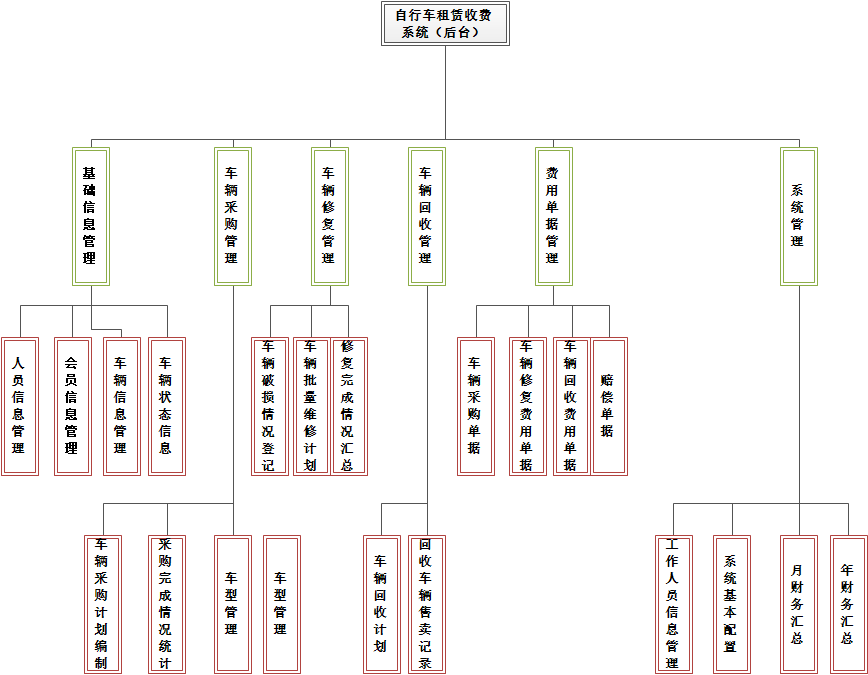


图4-4 后台功能

## 4.4.1.2 前台功能

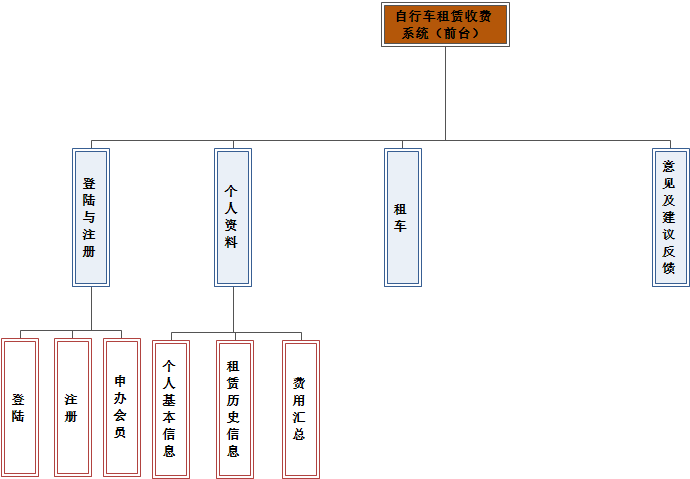


图4-4 前台功能

# 5数据库设计

## 5.1 系统概念模型设计

## 5.1.1 系统实体描述

通过前期对房屋租赁管理系统需求进行分析和总结，进入到数据库设计的环节——逻辑结构设计。逻辑结构设计是指将需求分析得到的用户的需求转换为现实信息世界的模型，它是数据库设计中的一个重要环节，不仅能够十分真实的反应现实世界，还能清楚的展示各事物之间的联系。因此概念结构设计在数据库设计环节中的主要任务是在需求分析阶段编写的需求文档的基础上，使用特定的方式将其转化为一个不依赖于任何机器的数据模型。最常用的概念模型是E-R模型，它是由实体、属性和联系三要素组成。

图5-1 er图图例



## 5.1.2 系统E-R图

E-R图即实体-联系图(Entity Relationship Diagram)，是指提供了表示实体型、属性和联系的方法，用来描述现实世界的概念模型。E-R方法:是“实体-联系方法”（Entity-Relationship Approach）的简称。它是描述现实世界概念结构模型的有效方法。

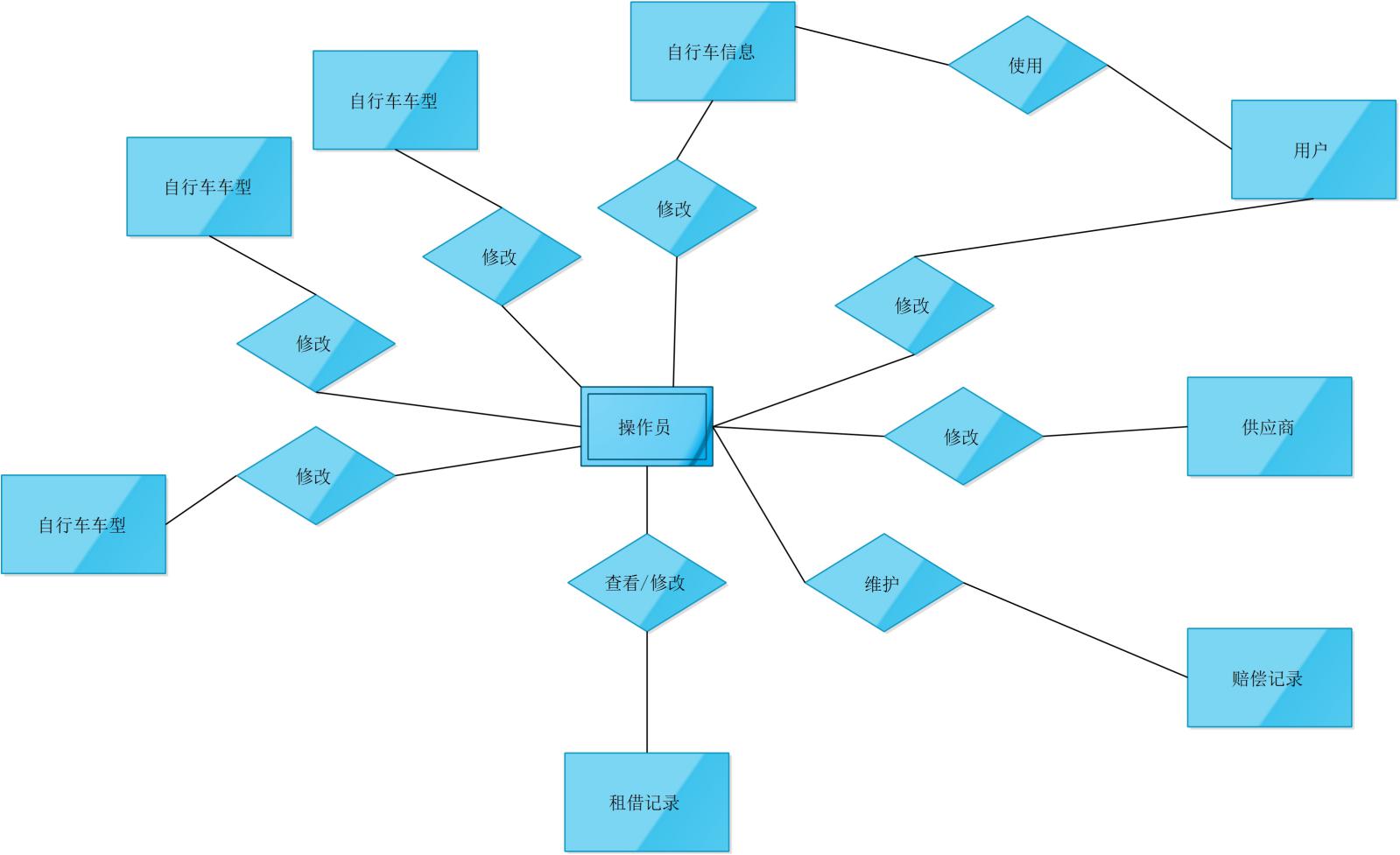


图5-2 er图

## 5.1.3 系统类图

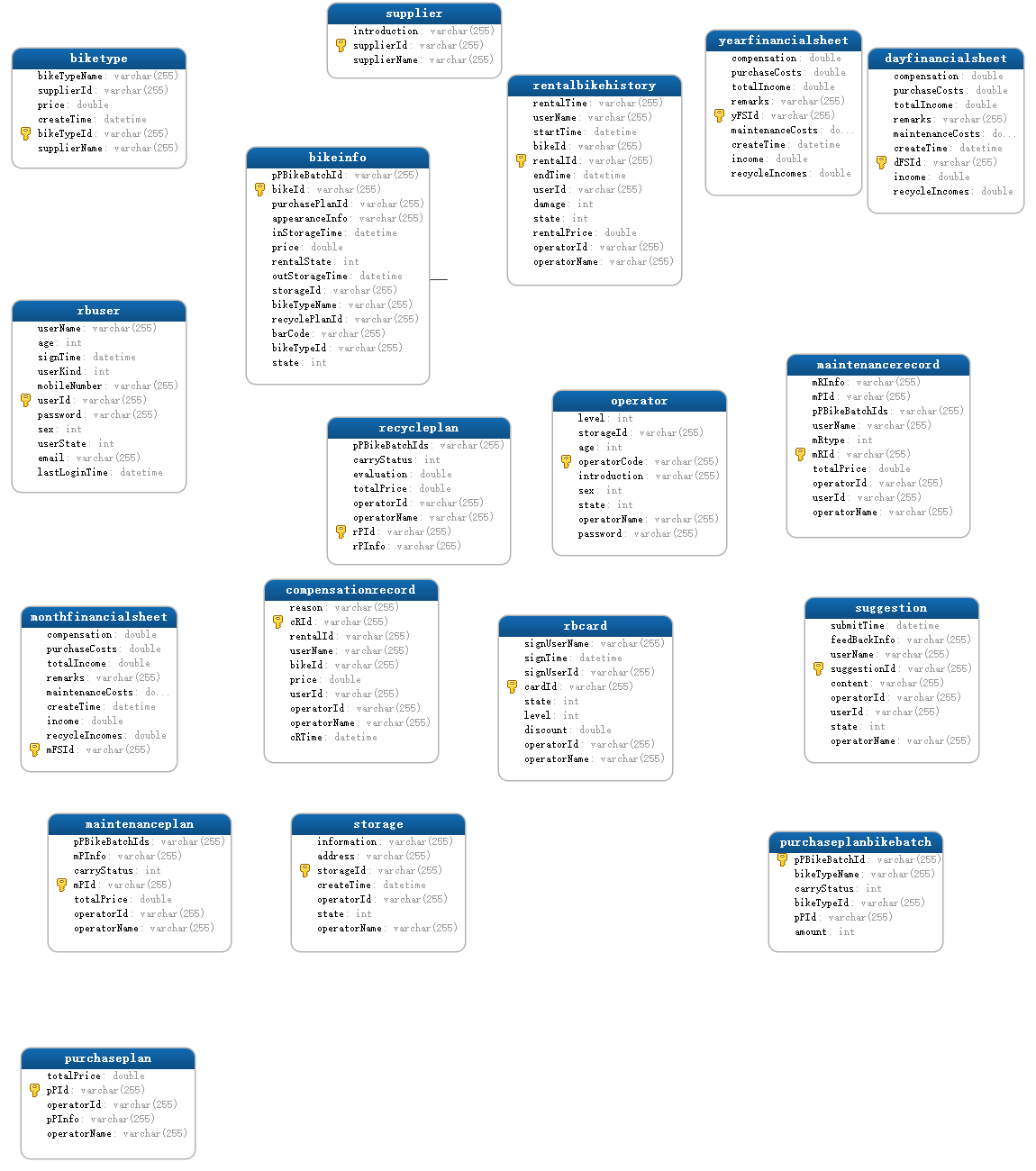


图5-3 类图

## 5.2 数据库表详细

## 5.2.1 数据库表设计综述

数据库就是一个系统，表就是它的对象，而字段即是它的属性。一个确保数据库事务正确执行的四个基本要素是：

(1) 原子性。基本表中的字段是不可再分解的。  
(2) 原始性。基本表中的记录是原始数据（基础数据）的记录。  
(3) 演绎性。由基本表与代码表中的数据，可以派生出所有的输出数据。  
(4) 稳定性。基本表的结构是相对稳定的，表中的记录是要长期保存的。

类型缩写对应的类型全称：

I—int,Date-Datetime,D-double,NC-NChar, S-smallint,T-Text,L—long,C—Char,F—Float,

S---String

## 5.2.2用户表（RbUser）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| userId | 用户ID | C | 20 | 默认:年+序号 Key |
| userName | 姓名 | C | 20 |  |
| userKind | 用户类型 | I | 20 | 0:普通用户，1：vip会员 |
| userState | 用户状态 | I | 2 | 0-禁用，1-启用 |
| mobileNumber | 手机号 | C | 20 |  |
| email | 邮箱地址 | C | 30 |  |
| password | 密码 | C | 20 | 使用手机号和邮箱登陆或者一卡通刷卡 |
| sex | 性别 | I | 1 | 0-男，1-女 |
| age | 年龄 | I | 2 |  |
| lastLoginTime | 最后登录时间 | Date | 255 |  |
| signTime | 注册时间 | Date | 255 |  |

## 5.2.3管理员/工作人员表（Operator）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| operatorCode | 人员编号 | C | 20 | 默认:年+序号 |
| operatorName | 人员姓名 | C | 20 |  |
| sex | 性别 | I | 1 | 0-男，1-女 |
| age | 年龄 | I | 2 |  |
| password | 密码 | C | 20 |  |
| state | 状态 | I | 2 | 0-禁用，1-启用 |
| level | 人员等级 | I | 2 | 0-超级管理员，1仓库管理员，2-工作人员，3-自行车归还验收员 |
| introduction | 人员职权简介 | C | 255 |  |
| storageId | 人员所属仓库 | C |  | 关联仓库表 |

## 5.2.4一卡通/会员信息表（RbCard）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| cardId | 卡号 | C | 20 | 默认:年+序号 |
| signTime | 绑定时间 | C | 20 |  |
| signUserId | 绑定用户ID | C | 20 |  |
| signUserName | 绑定用户名 | C | 255 |  |
| level | 会员等级 | I | 2 | 分为多个等级，每个等级对应的折扣不同，办理的会员价格也不同 |
| discount | 会员折扣 | D | 2 | 用于计算价格时候的折扣，超级管理员可以修改 |
| operatorId | 操作人员 | C |  | 关联操作员表 |
| operatorName | 操作人员姓名 | C |  |  |
| state | 会员状态状态 | I |  | 0-禁用，1-启用 |

## 5.2.5自行车信息表（BikeInfo）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| bikeId | 自行车编号 | C | 20 | 默认:年+序号 |
| purchasePlanId | 采购计划 | C | 20 |  |
| pPBikeBatchId | 采购计划批次表 | C | 20 |  |
| recyclePlanId | 回收计划 | C |  |  |
| bikeTypeId | 车型 | C | 20 |  |
| bikeTypeName | 车型名 | C |  |  |
| appearanceInfo | 外观信息 | C | 255 | 记录自行车外观信息 |
| storageId | 所属仓库 | C |  |  |
| inStorageTime | 入库时间 | Date |  |  |
| outStorageTime | 出库时间 | Date |  |  |
| price | 租车价格（小时） | D |  |  |
| state | 车辆状态 | I | 2 | 0-禁用，1-启用 |
| rentalState | 租借状态 | I | 2 | 0-未租用，1-正在租借中 |
| barCode | 条形码 | C | 20 | 用户后续开发使用 |

## 5.2.6自行车型表（BikeType）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| bikeTypeId | 自行车类型编号 | C | 20 | 默认:年+序号 |
| supplierId | 供应商 | C | 20 |  |
| supplierName | 供应商名称 |  |  |  |
| bikeTypeName | 车型名称 | C | 255 |  |
| createTime | 创建时间 | Date |  |  |
| price | 购买价格 | D |  |  |

## 5.2.7车辆租借记录表（RentalBikeHistory）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| rentalId | 租借流水号 | C | 20 | 默认:年+序号 |
| bikeId | 租借车辆编号 | C | 20 |  |
| userId | 租借人 | C | 20 |  |
| userName | 租借人姓名 | C |  |  |
| operatorId | 验收人员 | C | 20 | 关联操作员表 |
| operatorName | 验收人员姓名 | C |  |  |
| startTime | 开始时间 | Date |  |  |
| endTime | 结束时间 | Date |  |  |
| rentalTime | 出租时间 | C |  |  |
| rentalPrice | 出租消费 | D |  |  |
| state | 租赁状态 | I | 2 | 0-未结算，1-已结算 |
| damage | 车辆受损 | I | 2 | 0-没有损坏，1-车辆有破损 |

## 5.2.8供应商表（Supplier）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| supplierId | 供应商编号 | C | 20 | 默认:年+序号 |
| supplierName | 供应商名称 | C | 255 |  |
| introduction | 供应商简介 | C | 255 |  |

## 5.2.9赔偿记录表（CompensationRecord）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| cRId | 赔偿记录编号 | C | 20 | 默认:年+序号 |
| bikeId | 自行车编号 | C | 20 |  |
| userId | 用户编号 | C | 20 |  |
| userName | 用户名 | C | 20 |  |
| rentalId | 租借记录流水号 | C | 20 |  |
| price | 赔偿金额 | D | 8 |  |
| reason | 赔偿缘由 | C | 255 |  |
| operatorId | 验收人员 | C |  | 关联操作员表 |
| operatorName | 验收人员姓名 | C |  |  |
| cRTime | 赔偿制定时间 | Date |  |  |

## 5.2.10采购计划表（PurchasePlan）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| pPId | 采购计划流水号 | C | 20 | 默认:年+序号 |
| pPInfo | 采购计划详情 | C | 255 |  |
| operatorId | 工作人员 | C | 20 |  |
| operatorName | 姓名 | C | 20 |  |
| totalPrice | 总金额 | D | 20 |  |

## 5.2.11采购计划批次表（PurchasePlanBikeBatch）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| pPBikeBatchId | 采购计划车型批次流水号 | C |  | 默认:年+序号 |
| pPId | 采购计划流水号 | C | 20 |  |
| bikeTypeId | 采购计划车型 | C | 20 | 关联车型表 |
| bikeTypeName | 车型名称 | C | 255 |  |
| amount | 采购数量 | I | 20 |  |
| carryStatus | 执行情况 | I |  | 0-计划作废，1-已完成，2-回收失败，3-回收成功 |

## 5.2.12回收计划表（RecyclePlan）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| rPId | 回收计划流水号 | C | 20 | 默认:年+序号 |
| rPInfo | 回收计划详情 | C | 255 |  |
| operatorId | 工作人员 | C | 20 |  |
| operatorName | 姓名 | C | 20 |  |
| pPBikeBatchIds | 批次 | C | 255 | 多个批次逗号隔开 |
| evaluation | 估价 | D | 20 | 验收员估价 |
| totalPrice | 总金额 | D | 20 |  |
| carryStatus | 执行情况 | I |  | 0-未回收，1-处理中，2-已回收 |

## 5.2.13维修计划表（MaintenancePlan）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| mPId | 维修计划流水号 | C | 20 | 默认:年+序号 |
| mPInfo | 维修计划详情 | C | 255 |  |
| operatorId | 工作人员 | C | 20 |  |
| operatorName | 姓名 | C | 20 |  |
| pPBikeBatchIds | 批次 | C | 255 | 多个批次逗号隔开 |
| totalPrice | 总金额 | D | 20 | 反馈信息 |
| carryStatus | 执行情况 | I |  | 0-未维修，1-处理中，2-已维修 |

## 5.2.14维修记录表（MaintenanceRecord）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| mRId | 维修记录流水号 | C | 20 | 默认:年+序号 |
| mRInfo | 维修详情 | C | 255 |  |
| operatorId | 工作人员 | C | 20 |  |
| operatorName | 姓名 | C | 20 |  |
| mPId | 维修计划流水号 | C | 20 | 如果是用户损坏，没有该项 |
| pPBikeBatchIds | 批次 | C | 255 | 多个批次逗号隔开，如果是用户损坏，没有该项 |
| mRtype | 维修类型 | I |  | 0-批量维修，1-用户损毁维修 |
| userId | 用户id | C |  | 如果是维修计划的，没有该项 |
| userName | 用户名 | C |  | 同上 |
| totalPrice | 总金额 | D | 20 | 反馈信息 |

## 5.2.15日财务表（DayFinancialSheet）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| dFSId | 月财务表Id | C | 20 | 默认:年+序号  每月自动数据库生成 |
| createTime | 时间 | Date |  |  |
| income | 收入 | D |  |  |
| purchaseCosts | 采购费用 | D |  |  |
| maintenanceCosts | 维修费用 | D |  | 只包含维修计划 |
| compensation | 用户赔偿 | D |  |  |
| recycleIncomes | 回收收益 | D |  |  |
| totalIncome | 总收益 | D | 20 | 收入-维修+回收-采购 |
| remarks | 备注 | C | 255 | 特殊情况记录 |

## 5.2.16月财务表（MonthFinancialSheet）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| mFSId | 月财务表Id | C | 20 | 默认:年+序号  每月自动数据库生成 |
| createTime | 时间 | D |  |  |
| income | 收入 | D |  |  |
| purchaseCosts | 采购费用 | D |  |  |
| maintenanceCosts | 维修费用 | D |  | 只包含维修计划 |
| compensation | 用户赔偿 | D |  |  |
| recycleIncomes | 回收收益 | D |  |  |
| totalIncome | 总收益 | D | 20 | 收入-维修+回收-采购 |
| remarks | 备注 | C | 255 | 特殊情况记录 |

## 5.2.17年财务表（YearFinancialSheet）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| yFSId | 月财务表Id | C | 20 | 默认:年+序号  每月自动数据库生成 |
| createTime | 时间 | D |  |  |
| income | 收入 | D |  |  |
| purchaseCosts | 采购费用 | D |  |  |
| maintenanceCosts | 维修费用 | D |  | 只包含维修计划 |
| compensation | 用户赔偿 | D |  |  |
| recycleIncomes | 回收收益 | D |  |  |
| totalIncome | 总收益 | D | 20 | 收入-维修+回收-采购 |
| remarks | 备注 | C | 255 | 特殊情况记录 |

## 5.2.18意见及建议表（Suggestion）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| suggestionId | 建议/建议流水号 | C | 20 | 默认:年+序号 |
| content | 意见/建议详情 | C | 255 |  |
| state | 审查状态 | I |  | 0-未处理，1-已处理 |
| operatorId | 工作人员 | C | 20 |  |
| operatorName | 工作人员姓名 | C | 20 |  |
| userId | 用户Id | C | 20 |  |
| userName | 用户名 | C | 255 |  |
| submitTime | 提交时间 | Date |  |  |
| feedBackInfo | 反馈信息 | C | 255 | 反馈信息 |

## 5.2.19仓库表（Storage）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **英文名** | **中文名** | **数据类型** | **长度** | **备注** |
| storageId | 仓库编号 | C | 20 | 默认:年+序号  分为多个仓库，每个仓库都有管理人员 |
| address | 地址 | C | 255 |  |
| state | 状态 | I |  | 0-禁用，1-启用 |
| createTime | 创建时间 | Date |  |  |
| operatorId | 管理人员 | C | 20 |  |
| operatorName | 姓名 | C |  |  |
| information | 具体信息 | C | 255 |  |

# 系统实现

## 6.1 系统实现总体描述

## 6.1.1 实现理念

代码风格是编码的一个重要方面，它直接影响代码的质量和阅读性。收支管理子系统在编码实现的过程中采用统一模板编程。尽量统一代码的风格。同时代码的实现的过程中尽量采用健壮灵活的编码方式。使用面向对象的方式进行编程。

## 6.1.2 软件环境

java开发环境：Eclipse-mar、 jdk 1.8

web页面开发环境：Sublime Text 2.0

图形绘制：Edraw、StartUML

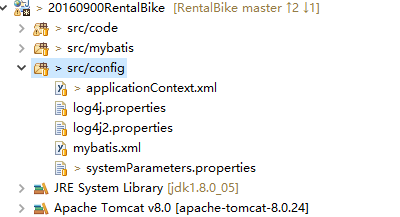
图形处理：Photoshop cc

数据库：mysql 5.5

## 6.1.3 代码实现

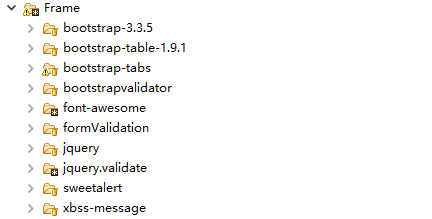
* java代码层次

分为分为四层：数据entity层、数据库访问DAO层、业务服务层service、界面服务层contraller。数据模型层负责实体的定义、数据访问DAO层服务数据的持久化操作、业务服务层负责业务逻辑处理、界面服务层负责界面数据显示与处理。

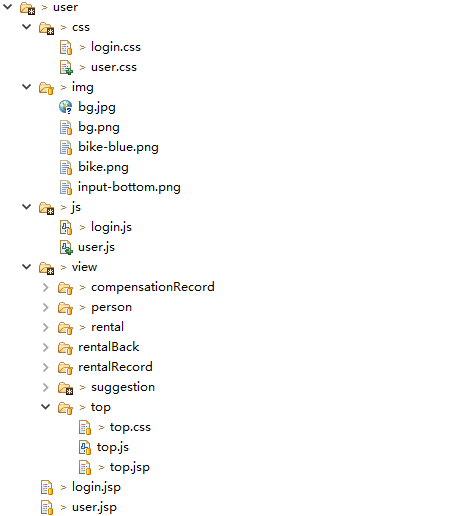


* 前台框架层次

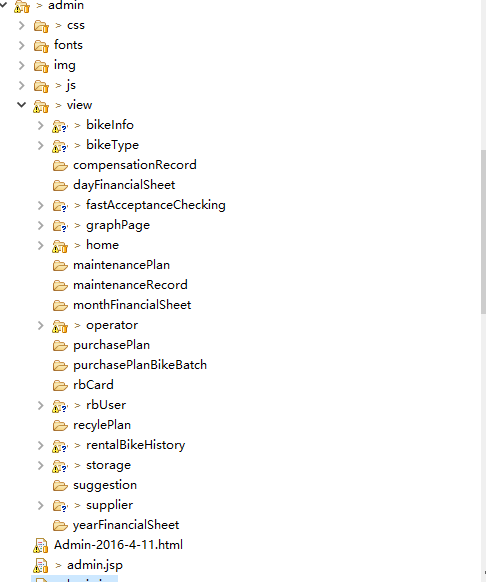
采用bootstrap来作为页面的主要框架，搭配bootstrap-table、bootstrap-tabs等形成一个完整的前台页面服务框架



* user用户页面代码层次



* admin管理员页面代码层次



## 6.2 客户端（客户机）模块实现

## 6.2.0 主界面模块的实现

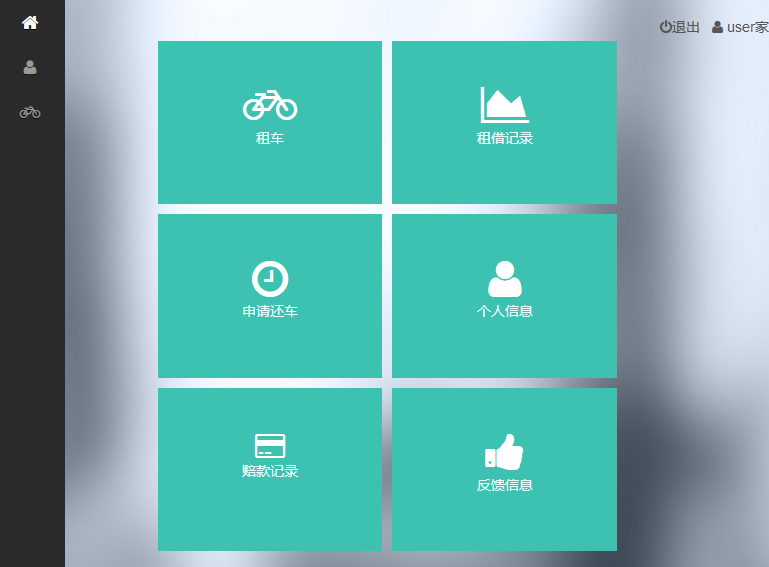


图6-1

## 6.2.1 登录模块的实现

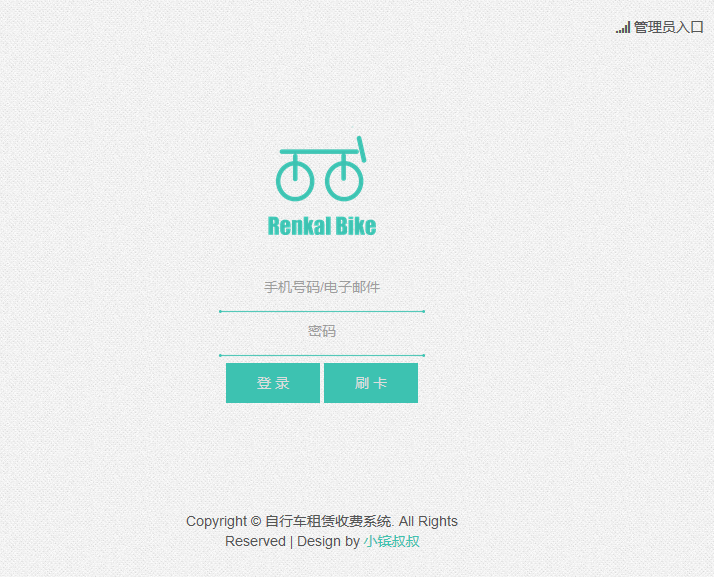


图6-2

## 6.2.3 租车模块的实现

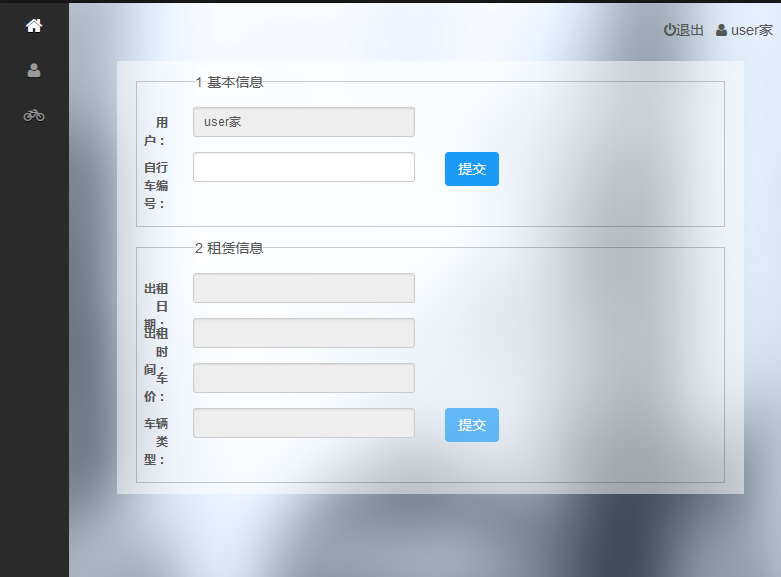


图6-3

## 6.2.4 查看租车记录模块的实现



图6-4

## 6.2.5 个人信息模块的实现

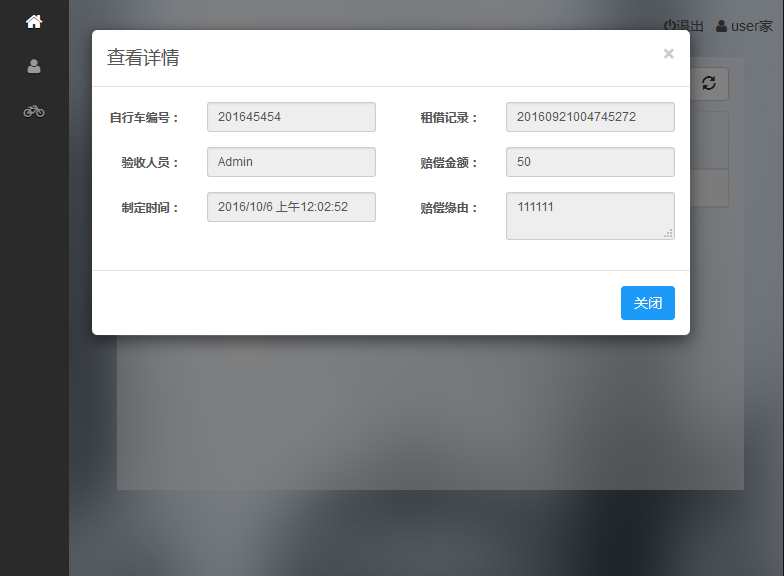


图6-5

## 6.2.6 查看赔款模块的实现



图6-6



## 6.2.7 申请还车模块的实现

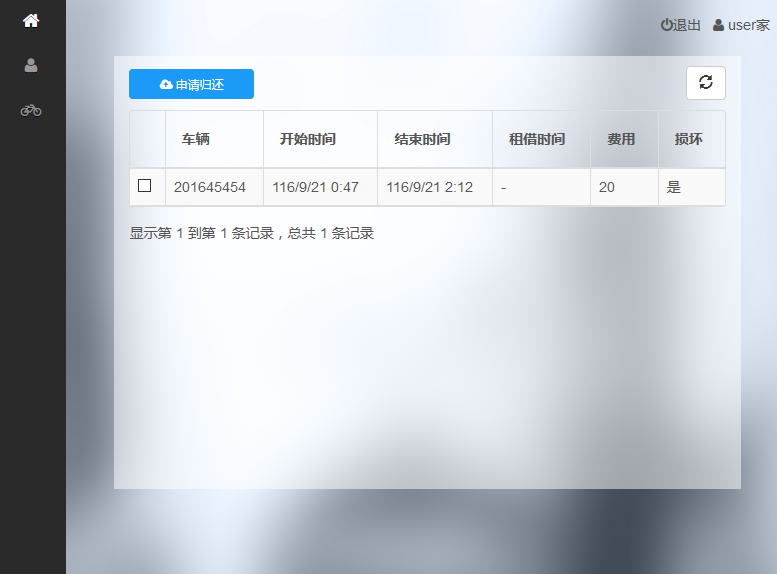


图6-7

## 6.2.8 反馈信息模块的实现



图6-8

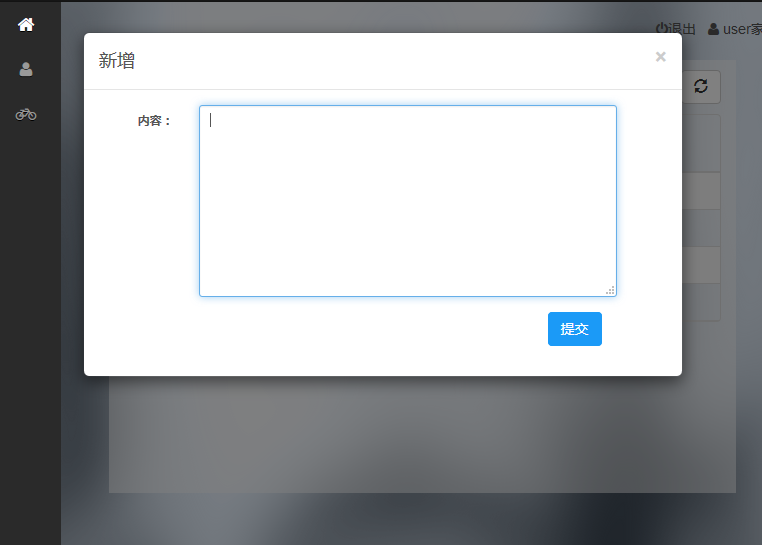


图6-9

## 6.3 客户端（PC端）模块实现

## 6.3.1 主界面的实现

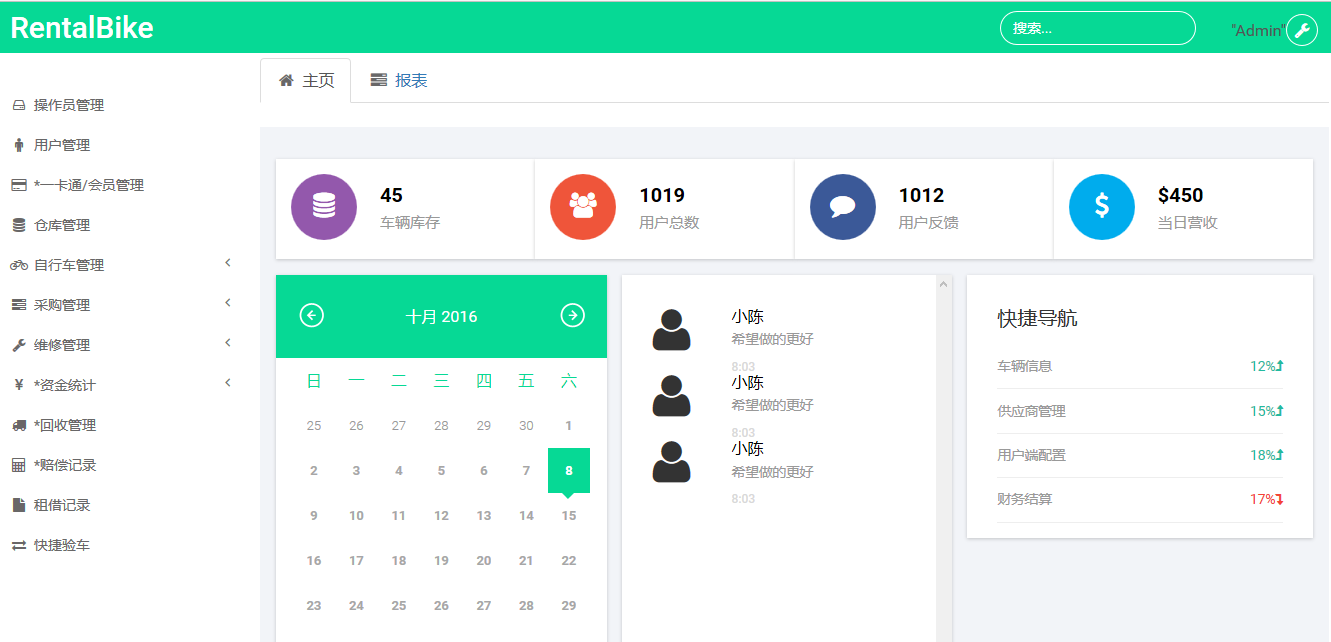


图6-10

## 6.3.2 报表模块的实现



图6-11

## 6.3.3 登录的实现

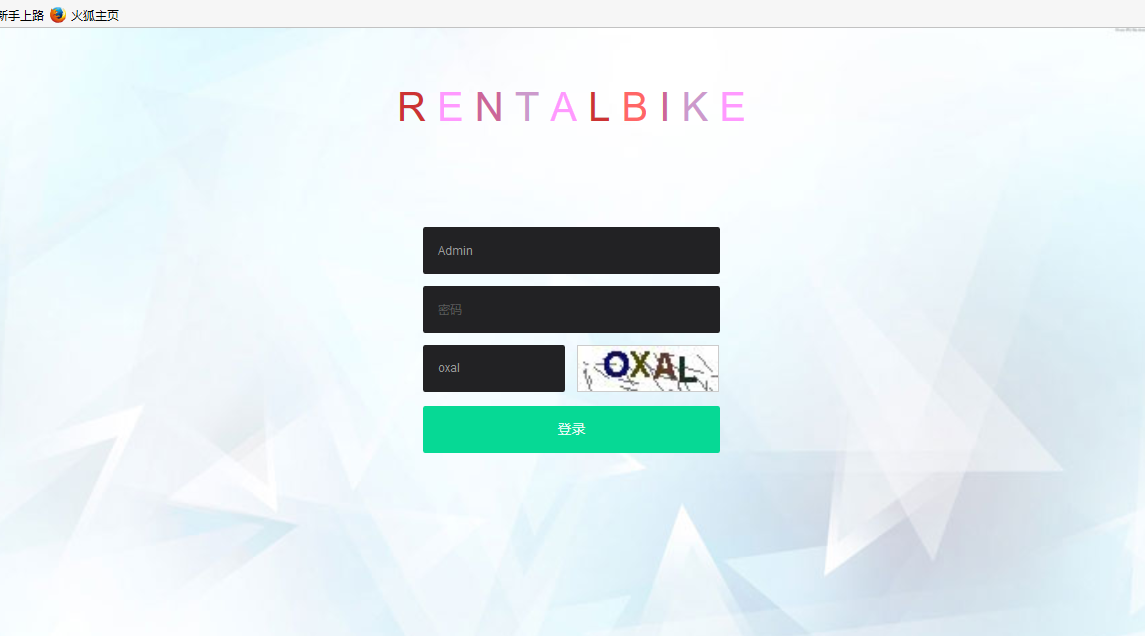


图6-12

## 6.3.4 操作员管理模块的实现



图6-13

## 6.3.5 仓库管理模块的实现



图6-14

## 6.3.6 自行车信息管理模块的实现



图6-15

## 6.3.6 快捷验车管理模块的实现



图6-16

# 系统测试

## 7.1 系统测试综述

使用人工或自动的手段，来运行或测试某个系统的过程。其目的在于检验它是否满足规定的需求或弄清与其审计结果之间的差别。

软件测试为了发现并指出软件系统缺陷即bug。软件中的bug指的是软件中（包含程序文档）不符合用户需求的问题。

我们测试一个软件的第一件事情，通常是搭建软件测试环境，简单的说，软件测试环境就是软件运行平台的，即软件、硬件和网络的集合：测试环境=软件+硬件+网络。

搭建测试环境有如下几个要点。

1. 真实，尽量模拟用户的真实使用环境，
2. 干净，测试环境中尽量不要安装其他无关的软件，
3. 无毒，测试环境没有中毒
4. 独立，测试环境和开发环境独立

测试用例，英文名为 Test Case，缩写TC，指的是在测试执行之前设计一套详细的测试方案，包括测试环境，测试步骤，测试数据和预期结果：测试用例 = 输入 + 输出 +　测试环境。 其中，“输入”包括测试数据和操作步骤；“输出”指的是期望结果；“测试环境”指的是系统环境设置。

编写测试用例时应注意4W，why（为什么要写测试用例），when（什么时候写的测试用例），who（有谁来写测试用例），what（根据什么写测试用例）。

测试的分类：

1. 根据软件测试阶段可分为单元测试、集成测试、系统测试、验收测试，其中验收测试又可分为阿尔法测试和贝塔测试。
2. 从是否需要执行被测试软件的角度来看，可分为静态测试和动态测试。
3. 从测试是否针对系统内部结构和具体实现算法的角度来看，可分为黑盒测试和白盒测试。黑盒测试又可分为功能测试和性能测试。其中功能测试包括逻辑功能测试、界面测试、易用性测试、安装测试、兼容测试；性能测试包括一般性能测试、稳定性测试、负载测试、压力测试。

## 7.2 客户端测试

## 7.2.1 登录模块测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 7.2.1 | | 版本号 | | V 1.0 |
| 测试环境 | 火狐浏览器 | | | | |
| 用例名称 | 用户登录 | | | | |
| 前提条件 | 自行车租赁管理系统服务器已经启动 | | | | |
| 测试步骤 | 1. 打开火狐浏览器，输入网址 2. 准确填写登录信息，完成后，点击登录 3. 系统提示注册成功 | | | | |
| 输入数据 | 登录名、登录密码 | | | | |
| 预期输出 | 点击登录按钮，系统提示登录成功，到达客户端主界面 | | | | |
| 实际输出 | 同预期输出相同 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 邵小镔 | 设计日期 | | 2016-9-10 | |
| 测试人 | 邵小镔 | 测试日期 | | | 2016-9-11 |
| 再测试人 |  | 再测试日期 | | |  |
| 问题修改摘要 |  | | | | |
| 修改人 |  | | 修改日期 | |  |

## 7.2.2 租车测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 7.2.2 | | 版本号 | | V 1.0 |
| 测试环境 | 火狐浏览器 | | | | |
| 用例名称 | 租车业务 | | | | |
| 前提条件 | 自行车租赁管理系统服务器已经启动 | | | | |
| 测试步骤 | 1.打开火狐浏览器，输入网址，点击租车  2.准确填写车辆信息，完成后，点击提交  3.系统提示车辆是否可用  4.如果可用就输出用户的信息  5.用户确认信息后，点击提交  6.系统提示操作成功，等待服务员对车解锁 | | | | |
| 输入数据 | 车辆信息 | | | | |
| 预期输出 | 点击租车提交按钮，提示操作完成 | | | | |
| 实际输出 | 同预期输出相同 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 邵小镔 | 设计日期 | | 2016-9-10 | |
| 测试人 | 邵小镔 | 测试日期 | | | 2016-9-11 |
| 再测试人 |  | 再测试日期 | | |  |
| 问题修改摘要 |  | | | | |
| 修改人 |  | | 修改日期 | |  |

## 7.3 管理员端测试

## 7.3.1 登录模块测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 7.3.1 | | 版本号 | | V 1.0 |
| 测试环境 | 火狐浏览器 | | | | |
| 用例名称 | 管理员登录 | | | | |
| 前提条件 | 自行车租赁管理系统服务器已经启动 | | | | |
| 测试步骤 | 1. 打开火狐浏览器，输入网址 2. 准确填写登录信息，完成后，点击登录 3. 系统提示注册成功 | | | | |
| 输入数据 | 登录名、登录密码、验证码 | | | | |
| 预期输出 | 点击登录按钮，系统提示登录成功，到达管理员主界面 | | | | |
| 实际输出 | 同预期输出相同 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 邵小镔 | 设计日期 | | 2016-9-10 | |
| 测试人 | 邵小镔 | 测试日期 | | | 2016-9-11 |
| 再测试人 |  | 再测试日期 | | |  |
| 问题修改摘要 |  | | | | |
| 修改人 |  | | 修改日期 | |  |

## 7.3.2 用户管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 7.3.2 | | 版本号 | | V 1.0 |
| 测试环境 | 火狐浏览器 | | | | |
| 用例名称 | 用户开户，新增用户 | | | | |
| 前提条件 | 自行车租赁管理系统服务器已经启动 | | | | |
| 测试步骤 | 1.打开火狐浏览器，输入网址，正确登录  2.点击用户管理  3.点击新增  4.准确填写用户信息，完成后，点击提交  5.系统提示信息 | | | | |
| 输入数据 | 用户基本信息 | | | | |
| 预期输出 | 点击新增用户弹窗的提交按钮，提示操作成功 | | | | |
| 实际输出 | 同预期输出相同 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 邵小镔 | 设计日期 | | 2016-9-10 | |
| 测试人 | 邵小镔 | 测试日期 | | | 2016-9-11 |
| 再测试人 |  | 再测试日期 | | |  |
| 问题修改摘要 |  | | | | |
| 修改人 |  | | 修改日期 | |  |

## 7.3.3 自行车管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 7.3.2 | | 版本号 | | V 1.0 |
| 测试环境 | 火狐浏览器 | | | | |
| 用例名称 | 新增自行车 | | | | |
| 前提条件 | 自行车租赁管理系统服务器已经启动 | | | | |
| 测试步骤 | 1.打开火狐浏览器，输入网址，正确登录  2.点击车辆信息管理  3.点击新增  4.准确填写车辆信息，完成后，点击提交  5.系统提示信息 | | | | |
| 输入数据 | 车辆基本信息 | | | | |
| 预期输出 | 点击新增车辆弹窗的提交按钮，提示操作成功 | | | | |
| 实际输出 | 同预期输出相同 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 邵小镔 | 设计日期 | | 2016-9-10 | |
| 测试人 | 邵小镔 | 测试日期 | | | 2016-9-11 |
| 再测试人 |  | 再测试日期 | | |  |
| 问题修改摘要 |  | | | | |
| 修改人 |  | | 修改日期 | |  |

## 7.3.4 自行车类型管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 7.3.2 | | 版本号 | | V 1.0 |
| 测试环境 | 火狐浏览器 | | | | |
| 用例名称 | 新增自行车类型 | | | | |
| 前提条件 | 自行车租赁管理系统服务器已经启动 | | | | |
| 测试步骤 | 1.打开火狐浏览器，输入网址，正确登录  2.点击车辆类型信息管理  3.点击新增  4.准确填写车辆类型信息，完成后，点击提交  5.系统提示信息 | | | | |
| 输入数据 | 车辆类型基本信息 | | | | |
| 预期输出 | 点击新增车辆类型弹窗的提交按钮，提示操作成功 | | | | |
| 实际输出 | 同预期输出相同 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 邵小镔 | 设计日期 | | 2016-9-10 | |
| 测试人 | 邵小镔 | 测试日期 | | | 2016-9-11 |
| 再测试人 |  | 再测试日期 | | |  |
| 问题修改摘要 |  | | | | |
| 修改人 |  | | 修改日期 | |  |

## 7.3.4 仓库管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 7.3.2 | | 版本号 | | V 1.0 |
| 测试环境 | 火狐浏览器 | | | | |
| 用例名称 | 新增仓库 | | | | |
| 前提条件 | 自行车租赁管理系统服务器已经启动 | | | | |
| 测试步骤 | 1.打开火狐浏览器，输入网址，正确登录  2.点击仓库信息管理  3.点击新增  4.准确填写仓库信息，完成后，点击提交  5.系统提示信息 | | | | |
| 输入数据 | 仓库基本信息 | | | | |
| 预期输出 | 点击新增仓库弹窗的提交按钮，提示操作成功 | | | | |
| 实际输出 | 同预期输出相同 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 邵小镔 | 设计日期 | | 2016-9-10 | |
| 测试人 | 邵小镔 | 测试日期 | | | 2016-9-11 |
| 再测试人 |  | 再测试日期 | | |  |
| 问题修改摘要 |  | | | | |
| 修改人 |  | | 修改日期 | |  |

## 7.3.4 快捷验车管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试用例编号 | 7.3.2 | | 版本号 | | V 1.0 |
| 测试环境 | 火狐浏览器 | | | | |
| 用例名称 | 检验用户归还的车辆 | | | | |
| 前提条件 | 自行车租赁管理系统服务器已经启动 | | | | |
| 测试步骤 | 1.打开火狐浏览器，输入网址，正确登录  2.点击快捷验车管理  3.选择条目，点击验车  4.准确填写车辆受损情况，完成后，点击提交  5.系统提示信息 | | | | |
| 输入数据 | 车辆受损情况 | | | | |
| 预期输出 | 点击归还信息弹窗的提交按钮，提示操作成功 | | | | |
| 实际输出 | 同预期输出相同 | | | | |
| 问题描述 | 无 | | | | |
| 设计人 | 邵小镔 | 设计日期 | | 2016-9-10 | |
| 测试人 | 邵小镔 | 测试日期 | | | 2016-9-11 |
| 再测试人 |  | 再测试日期 | | |  |
| 问题修改摘要 |  | | | | |
| 修改人 |  | | 修改日期 | |  |

# 项目总结

通过这次综合课程设计，我收获了很多，获得了很多人的帮助，心中非常的感激帮助过我的人。从这个过程中认识到，学了技术就应该使用到实际生活中，那里才是它应该发光的地方，并且，知识如果不用，就会发霉，最终到达腐朽的地步。当然我在这个过程中也深刻理解到自身的不足，广大世界的知识我还只是摄取了极小的一部分，今后还需要勤奋努力，向更高的山峰攀登。

# 致谢

历时一个多月的时间终于完成自行车租赁收费系统的设计与实现，在系统开发过程中遇到了无数的困难和阻碍，均在张小川老师、王森老师以及同学朋友的细心帮助下冲破一道道难关。在做系统调研的时候，广大社会人士也给予了我巨大的帮助，在此向帮助和指导过我的各位老师表示最衷心的感谢。

# 参考文献

[1] 冯燕奎, 赵德奎. JSP实用案例教程[M] 清华大学出版社, 2004, 5: 70-100

[2] 王家华 软件工程[M] 东北大学出版社2001年3月303页

[3] 王宜贵 软件工程[M] 北京：机械工业出版社，2002：20-79

[4]傅鹏，殷旻昊.基于Structs+Spring+Hibernate+Ajax技术的科研管理系统设计[J].软件导刊,2009, 8(1):135-136.

[5]李宁,李战怀.基于黑盒测试的软件测试策略研究与实现[J].计算机应用研究2009,26(3):924-926.

[6]陆惠思，《软件工程》，电子工业出版社

[7]《数据库系统概论》，高等教育出版社。

[8]李贤华，基于JSP技术的大型网上购物系统的设计与实现[J]. 计算机与现代化. 2008

[9]贺松平．基于MVC模式的B/S架构的研究及应用［Ｄ］．武汉：华中科技大学

[10] 萨师煊，王珊.《数据库系统概论》.高等教育出版社,2002年2月