

基于 Java 的汽车租赁系统的设计与实现

云贵全

(四川信息职业技术学院计算机工程系, 四川 广元 628017)

摘要: 当前我国已经跻身于世界汽车大国, 汽车租赁市场也日趋活跃, 一个好的汽车租赁管理系统对于汽车租赁公司而言, 可以大大降低工作量, 节约成本, 提高效率。主要从设计的角度来阐述一个汽车租赁系统的设计和实现, 能够为汽车租赁系统的开发带来一定的帮助。

关键词: 汽车租赁系统; 设计; 开发

Design and implementation of Java-based car rental system

YUN Gui-quan

(Department of Computer and Engineering Sichuan Information Technology College, Guangyuan 628017 Sichuan Province China)

Abstract: China has been among the world's car, the car rental market is also becoming increasingly active. A good car rental management system for car rental companies greatly reduces the workload, cost savings and improves the efficiency. From a design perspective, this paper elaborates the design and implementation of a car rental system, which brings some help for the development of the car rental system.

Key words: car rental system; design; development

0 引言

随着我国逐渐进入汽车时代, 汽车也越来越成为大家不可或缺的交通工具。但是由于种种原因, 会存在很多人的汽车闲置, 或者很多人没有车辆使用的情况。基于这种原因, 汽车租赁行业也就应运而生了。由于汽车本身是一种比较昂贵的商品, 因此在汽车租赁的过程中, 对汽车的管理和对客户的信息采集管理等, 都涉及到大量的数据处理, 如果使用传统的方法来管理, 将会带来效率低下、成本过高等问题, 一个好的汽车租赁管理软件则可以较好地解决这些问题。本文则是从软件开发的全面阐述一个汽车租赁系统的设计与实现, 以供大家参考。

1 系统分析与设计

系统分析作为系统开发的基础, 它是一个将客户的需求从具体到抽象的过程, 为后续环节的设计与开发指明方向。车辆租赁系统大致可以分为登录、系统管理、员工管理、车辆管理、客户管理、租赁管理和数据统计模块。车辆租赁系统的功能模块图大致如图 1 所示。

该系统主要有客户、管理员、验车员、出纳、经理

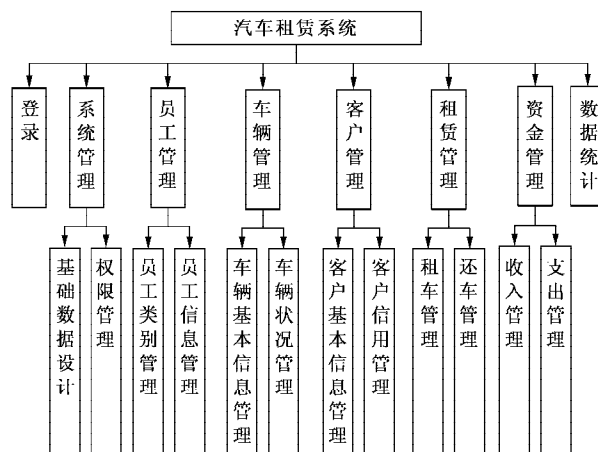


图 1 功能模块

这几个角色。其中管理员主要完成系统管理、员工管理、车辆管理中的车辆基本信息管理和客户管理中的车辆基本信息管理几个模块的管理工作; 验车员主要完成车辆管理中的车辆状况管理模块的管理

收稿日期: 2012-03-07

作者简介: 云贵全(1975-), 男, 在读高师硕士, 讲师, 主要从事软件技术类课程的教学与研究工作。

工作;出纳主要完成资金管理模块的工作;经理主要完成数据统计模块的工作;而客户则要贯穿整个租车的流程。

在整个租车的过程中主要是围绕租车和还车两个流程来完成的。租车流程如图2所示。

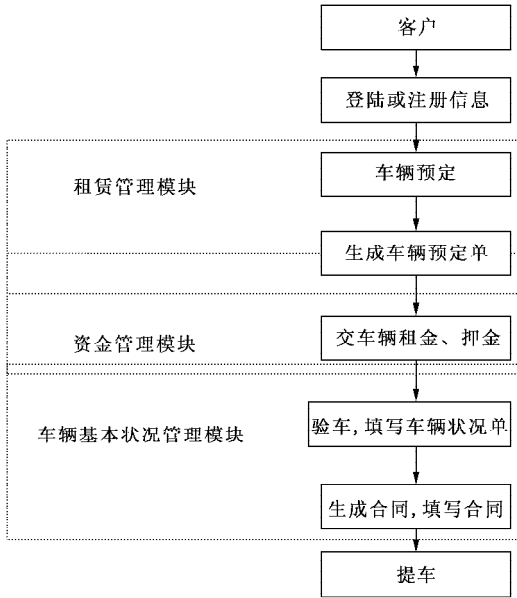


图2 租车流程

租车活动主要是由客户发起的。第一次租车的客户要注册自己的信息,然后登陆到系统,然后到租车管理模块中选择好自己想要租赁的车辆,然后进行预订,在预订完成后,客户就需要到租车公司的门店完成下面的一些操作。客户到了门店,首先由出纳接待,根据客户的车辆预订单收取车辆的租金和保证金,然后客户拿着预订单和收款凭证到验车员那里进行验车活动,验车员和客户共同验车,并在车辆状况管理模块中填写车辆状况并生成车辆状况单;验车员根据车辆状况单、客户车辆预订单和收款凭证,生成并打印车辆租赁合同,客户和租赁公司经理签字后,客户就可以提车了。

客户还车流程图如图3所示。

还车流程也是由客户发起的。客户开车到租赁门店,向验车员出示租车合同,验车员和客户共同验车,并生成车辆状况单。然后与之前的车辆状况单进行比对,如果车辆有损坏,则进行定损,然后生成定损单,然后再生成还车单;如果车辆没有损坏,则直接生成还车单。然后客户到出纳处,根据还车单,扣除车损金并暂扣车辆违纪保证金(车辆违纪保证金15日退还)后,向客户退还多余金额,并打印票据。然后整个还车活动结束。

下面对主要功能模块进行简单的介绍。

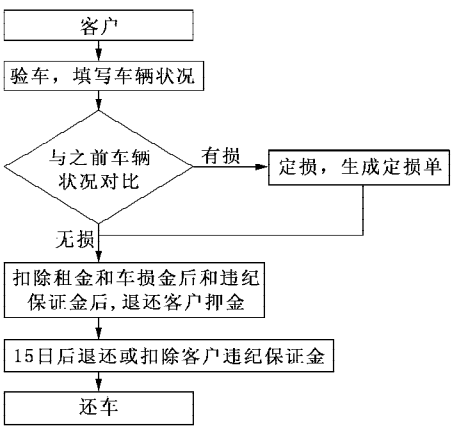


图3 还车流程

1.1 登录模块

对于系统中的所有用户都必须要进行登录才能够操作系统。每类型的用户登录后具有不同的权限。客户自己注册后即可登录;其它类型的成员则由管理员设定用户名和权限。

1.2 系统管理

系统管理模块主要是由管理员来完成。管理员登录成功后可以设定一些基础数据,如:车辆租金的算法、车损的算法等等数据。另外在系统管理模块中,管理员也可以对其它成员的权限进行管理,即可以设定哪些用户可以操作哪些模块。

1.3 员工管理模块

员工管理模块是由管理员来操作。当管理员登录后,则可以对员工的类别进行增删改查的操作。同时管理员在员工管理模块中也可以对员工的基本信息进行管理,即对员工进行增删改查的操作。

1.4 车辆管理

车辆管理模块中有两个小模块,分别为车辆基本信息管理和车辆状况管理。其中车辆基本信息管理由管理员来完成,主要是对车辆信息进行增删改查的操作。车辆状况管理则是由验车员来完成。当客户来租车或还车时都要对车辆进行查验,主要是检查车辆是否有损伤。然后生成验车单。在客户还车时,如果车辆有损伤,验车员就要对车辆定损,然后生成定损单,再生成还车单。

1.5 客户管理

客户管理分为客户基本信息管理和客户信用管理两个子模块。客户基本信息管理由管理员来完成,主要是对客户信息进行增删改查的操作。客户信用管理主要是对客户信用级别的调整,由管理员完成。

1.6 租赁管理

租赁管理是本系统的主要模块,分为租车管理

和还车管理两个子模块。租车管理由客户发起,实现一个流程,如图2所示。客户首先在该系统的网站上登录(如果首次使用该系统,则要注册),然后进入租车模块中选择自己喜欢的车辆进行租车预订,系统生成租车预订单(该预订单的有效期为1~3天,由管理员设定),这样客户就发起一个租车的流程。同时,当客户和验车员对车辆检验结束后,根据验车单、预收费单生成租车合同,经理打印出租车合同(一式两份)与客户一起签订租车合同,客户拿着租车合同到车辆管理员处才可以开车离开。还车管理则是客户开车到租车门店,由验车员进入还车模块,开启还车流程,还车流程如图3所示。

1.7 资金管理

资金管理分为两个子模块,收入管理和支出管理。当客户开启租车流程后,系统将生成车辆预订单,出纳员根据客户的车辆预订单收取客户的车辆租金和押金。然后出纳打印相应收据给客户,并生成预收费用的单据提交该验车员,以便验车员好开启验车流程。当客户来还车时,出纳员根据验车员的验车记录单和定损单,计算出客户将要支付的实际费用,然后在客户押金中扣除实际费用和一定的违纪保证金(一般15日后退还客户),将多余的钱退还给客户,并打印发票,租车活动结束。支出管理模块主要功能是对客户的违纪保证金进行管理,如果客户在租车的过程中,有违反交通法规的行为,则将在违纪保证金中扣除违纪罚款,如果客户没有违纪记录,则不扣任何金钱,15日之后客户到门店对违纪保证金进行最后结算。

1.8 数据统计

数据统计主要是实现门店经理数据查询、统计等功能。主要实现每月的营业额统计、车辆使用率统计等。

2 数据库设计

数据库设计是该系统的核心,数据库设计的好坏直接决定了系统是否能符合客户的需求。在这个系统中总共需要创建十多个数据库表,如客户信息表、员工信息表、基础数据表、员工类型表、权限表、车辆预订表、租车表、还车表、车辆状况表等等。这些数据库表都是来源于系统的需求,对于需要持久化的数据统统的做成数据库表,在数据库的设计中一定注意至少符合3NF。可以使用PowerDesigner来进行数据库的设计和验证,在这里就不再详述。

3 代码设计

由于该系统是使用B/S模式来开发的,Java语言是一个比较好的选择。使用Java语言可以利用

现有的市面上比较流行的框架来实现系统,如SSH框架。通过使用SSH框架可以很好的对系统进行分层,如图4所示。

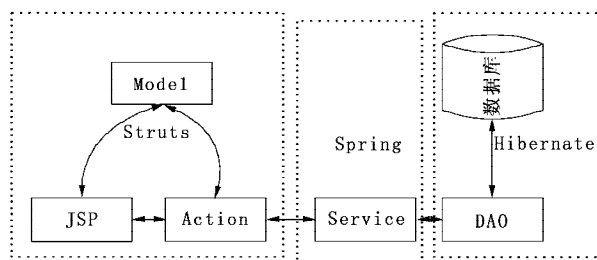


图4 SSH 分层模型

3.1 连接数据库

对于SSH连接数据库的方式可以使用配置式,如下所示:

```

<bean id="dataSource"
      class="org.apache.commons.dbcp.BasicDataSource" >
    <property name="driverClassName"
        value="com.microsoft.sqlserver.jdbc.
SQLServerDriver" >
    </property>
    <property name="url"
        value="jdbc:sqlserver://localhost:
1433;databaseName=XXX" >
    </property>
    <property name="username" value="
XXX" > </property>
    <property name="password" value="
" > </property>
</bean>

```

通过配置Spring的配置文件,可以较好地配置数据源的连接,然后通过与Hibernate的集成来实现对数据库的操作。

3.2 数据的基本操作

数据的基本操作是通过Hibernate来实现的,通过Hibernate可以为每一个数据库表生成一个DAO(数据访问对象)类和POJO类。在DAO类中主要是实现对数据库表的基本操作,如“增删改查”。

```

public class CarInfoDAO extends HibernateDaoSupport {

    private static final Logger log = LoggerFactory
        .getLogger( CarInfoDAO.class );

    .....

    public void save( CarInfo transientInstance ) {
        log.debug( "saving CarInfo instance" );

```

```

try {
    getHibernateTemplate ( ) . save ( transi-
entInstance) ;
    log. debug( " save successful" ) ;
} catch ( RuntimeException re) {
    log. error( " save failed" , re) ;
    throw re;
}
}
.....

```

3.3 业务逻辑代码

业务逻辑代码由 Spring 来完成,主要是连接 Hibernate 和 Struts。Spring 的操作主要是通过 DI (依赖注入) 来完成,如:

```

public class ShowCarBO implements IShowCarBO
{
    private CarInfoDAO carInfoDAO;
    public void setCarInfoDAO ( CarInfoDAO
carInfoDAO) {
        this. carInfoDAO = carInfoDAO;
    }
    public CarInfoDAO get CarInfoDAO ( ) {
        return carInfoDAO;
    }
    .....
}

```

在上述代码中的 carInfoDAO 对象不显示的实例化,有 Spring 的配置文件来实现实例化,如:

```

< bean id = " carInfoDAO " class = " com. dao.
CarInfoDAO " >
    < property name = " sessionFactory " >
        < ref bean = " sessionFactory " / >
    < /property >
< /bean >
< bean id = " transcationManager" class = "
org. springframework. orm. hibernate3. HibernateTransac-
tionManager" >
    < property name = " sessionFactory " ref = "
sessionFactory" / >
< /bean >
< bean id = " showCarBoTarget" class = "
com. bo. ShowCarBO" >
    < property name = " carInfoDAO " ref = "
carInfoDAO " / >
< /bean >
< bean id = " showCarBo" class = " org.

```

```

springframework. transaction. interceptor. Transaction-
ProxyFactoryBean" >
    < property name = " target " ref =
" showCarBoTarget" / >
    < property name = " proxyInterfaces
" value = " com. bo. IShowCarBO" / >
    < property name = " transactionManag-
er" ref = " transcationManager" / >
    < property name = " transactionAt-
tributes" >
        < props >
            < prop key = " add* " > PROPAG-
ATION_REQUIRED < /prop >
            < prop key = " * " > PROPAGA-
TION_SUPPORTS ,readOnly < /prop >
        < /props >
    < /property >
< /bean >

```

3.4 视图层代码

视图层则通过 Struts 来完成。Struts 主要是一个处理视图层的框架,它可以较好地与 Spring 集成,其集成方式也是通过配置文件来完成的。如:

```

< bean id = " showCarAction" class = " com. ac-
tion. ShowCarAction" >
    < property name = " showCarBo" ref =
" showCarBo" / >
< /bean >

```

对于 Struts 本身也是一个 MVC 框架,如图 4 的 SSH 分层模型所示,其中的 Action 类似于 JavaEE 中的 Servlet,充当了一个控制层的角色,实现页面间的跳转,在这里就不再详述。

4 结束语

本汽车租赁系统是基于 Java 语言实现的 B/S 系统,这类系统有着部署简便、维护简单等特点。该系统主要以汽车租赁的流程来实现客户的租赁活动,操作简单、容易开发。同时以 SSH 框架来实现系统,增加了系统的可维护性、扩展性,也为系统开发的管理带来了方便。

参 考 文 献:

- [1] 邓玉娇. 图书管理系统设计与实现[J]. 软件, 2011(2) : 36 - 38.
- [2] 林开荣. 汽车租赁企业经营分析及管理系统设计[J]. 山东轻工学院学报: 自然科学版, 2008(1) : 95 - 98.
- [3] 付更丽. SOA - SSH 分层架构的设计与应用[J]. 计算机技术与发展, 2010(1) : 74 - 77.

责任编辑: 刘新影