

长沙民政职业技术学院 毕业设计说明书

题目：汽车租赁系统的设计与实现

类型：

产品设计	工艺设计	方案设计
√		

学生姓名：刘华平

学号：1425123314

学院：软件学院

专业：软件开发

班级：软开 1433

学校指导教师：谢英辉

企业指导教师：谢英辉

年 月 日

摘 要

随着汽车工业的发展和汽车的普及,汽车租赁成为近年来兴起一个新行业。为了有效减轻企业各类工作人员的工作负担,规范企业的管理和经营行为,减少企业的经营成本,提高工作效率,开发了汽车租赁管理系统软件。

汽车租赁系统采用 C#作为前台开发工具, SQL Server 作为后台数据库。本系统是典型的管理信息系统,其开发主要包括后台数据库的建立和维护以及前端应用程序的开发两个方面。本系统的功能包括三大模块:基本信息的管理、客户会员管理、日常业务管理。可以对基本信息添加、修改、删除、查询;可以对客户会员添加、修改、删除,查询和查看客户会员信息,可以对汽车租赁业务进行管理主要包括汽车租赁、续赁、归还等到操作。

关键词: 基本信息管理, 客户会员管理, 汽车租赁管理 ;

目 录

【】 1.1 课题研究背景.....	3
1.3 系统实现的基本功能.....	4
2. 可行性分析.....	4
2.1 目标:	4
2.2 技术可行性分析:	5
2.3 经济可行性分析:	5
2.4 操作可行性分析:	5
2.5 系统开发环境需求:	5
3 需求分析.....	6
3.1 系统功能需求.....	6
3.2 数据流图.....	6
3.3 数据字典.....	8

4. 总体设计.....	11
4.1 系统的总体设计.....	11
4.4 数据库设计.....	14
将 E-R 图转换为关系模型.....	17
4.4.4 数据物理设计:	17
表 4-1 车辆信息表.....	18
5. 详细设计与代码实现.....	24
5.3.4. 用户管理模块的设计与实现.....	36
该模块主要是对用户的查看和删除等操作。	36
5.3.4.1 查看用户信息模块.....	36
6. 结论.....	43
参考文献:	44
致谢.....	45

1.1 课题研究背景

现代科学的发展,使计算机进入了几乎一切领域。从科学研究到工农业生产,从企业生产管理到家庭生活,各行各业都广泛地使用着计算机。当今社会的数据信息管理离不开计算机技术的支持。如何实现对数据信息快捷、有效、实用的管理,是软件开发业追求的目标。

今天,随着社会生产的发展,人民生活水平的提高,汽车租赁已成为一个极具市场潜力的行业,面对飞速发展的租赁市场,传统的人工租赁管理方式在进行繁多的业务信息处理时显得极不适应,因此,人们对于运用计算机来辅助协调和管理租赁工作的需求正在逐步提高。

随着社会物质条件的改善,生产生活节奏的加快,人们越来越意识到时间的重要性,在工作中,生活中越来越注重如何节省时间,提高效率。因为工作和生活的缘故,在必要的时候向汽车出租公司提出租赁车辆的服务要求,已成为大众广为认可和接受的行为和选择。在这种社会需求的强力带动下,汽车租赁市场出现了前所未有的一片繁荣景象。各个汽车租赁公司门庭若市,业务激增。然而在繁忙的景象背后,却也暴露出汽车租赁公司在处理业务信息时以人工方式为主所带来的诸如:效率低下,管理混乱,失误增多等诸多弊端。在实践中,汽车租赁行业的业主们逐渐意识到在租赁行业中借助计算机引入信息管理的重要性,并开始不断提高自身的素质和高科技在工作中的投入,以改善信息的管理,提高服务效率。

1.2 课题研究意义

汽车租赁管理就是对车辆信息和客户信息的管理,主要包括车辆的出租状态、车辆的基

本信息、车辆的投保情况、车辆的出厂情况、客户信息的管理等。由于这些过程间的关系复杂，有一对一的关系，一对多的关系，也有多对多的关系，所有这些工作使汽车租赁管理工作变得量大而又复杂，以前包括现在还有很多租赁公司采用人工管理，直接导致了错误的产生，服务质量的低下。汽车租赁管理系统的开发使得这一状况得到了根本的改善。这套管理系统包括涵盖了车辆租赁业务的各个方面。使得汽车租赁服务可以快速、规范地完成，节省了人力物力，提高了服务质量和经济效益。

1.3 系统实现的基本功能

汽车租赁管理系统针对的是汽车租赁公司在向客户提供出租车辆服务过程中面对的各种问题，对其业务信息进行自动化、信息化处理，使车辆租赁管理变得规范、快捷、高效。

基于各方面的研究，同时针对汽车租赁公司的实际情况，在金淑娟老师的悉心指导和帮助下，我查阅并应用数据库、管理信息系统、软件工程的一些基本原理和理论思想，以具有强大的应用程序开发功能和数据库开发功能的 myeclipse 为开发工具，设计开发了这套汽车租赁管理系统。在 Windows 环境下，此系统具有比较友好的界面和基本完善的功能，使用轻松方便，易于维护，适应性强。

本系统包括客户信息管理、车辆信息管理、汽车租赁归还管理、会员类型管理、会员信息管理、保险公司管理、汽车销售商管理等。具有添加、修改、删除、查询等功能，通过这些管理模块，可以使汽车租赁服务更加清晰，更加便捷，提高了服务质量，减少了操作人员的工作量。为规范操作流程，降低劳动强度使租赁业务走向标准化和科学化，设计开发了这套管理系统，它的使用将有利于提高租赁公司的劳动生产率，节约生产成本，增加经济效益，希望它能为业内人员的工作提供一定的帮助。

2. 可行性分析

2.1 目标:

随着汽车工业的发展和汽车的普及，汽车租赁成为近年来兴起的一个新兴行业。使用汽车租赁管理系统可以规范企业的管理和经营行为，可以减少企业的经营成本，提高工作效率，同时为规范操作流程，降低劳动强度使租赁业务走向标准化和科学化，设计开发了这套管理系统，它的使用将有利于提高租赁公司的劳动生产率，节约生产成本，增加经济效益，希望能为业内人士的工作提供一定的帮助。

2.2 技术可行性分析：

本系统后台数据库使用 SQL Server，开发语言使用 java。本系统采用目前比较流行的 ADO 数据库访问技术，并将每个数据库表的字段和操作封装到类中，从而成功地将面向对象的程序设计思想应用到数据库应用程序设计中。这也是本系统的特色和优势。C# 语言与数据库的结合是数据库发展的内在需求，也是数据库应用模式的具体实例。该系统上的汽车租赁过程和现实中通过汽车中介进行租赁汽车过程是极其相似的：就是把各种信息录入计算机中，以数据库的形式存储[1]

2.3 经济可行性分析：

随着汽车工业的发展和汽车的普及，汽车租赁成为近年来兴起的一个新兴行业。使用汽车租赁管理系统可以规范企业和管理和经营行为，减少企业的经营成本，提高工作效率。可使汽车租赁服务更加清晰，更加便捷，提高了服务质量，减少了操作人员的工作以及工作人员的数量，从经济角度来看有很大的发展潜力。

2.4 操作可行性分析：

此系统具有比较好的界面和基本完善的功能，使用轻松方便，操作简单，易于维护，适应性强，有较好的操作可行性。

2.5 系统开发环境需求：

开发工具：myeclipse, jdk1.7, Tomcat7.0;

数据库管理系统：sqlServer 2005

运行环境：Windows 7

3 需求分析

3.1 系统功能需求

本系统的使用对象是管理员。管理员部分的功能有管理员的登录，进入管理员相应的页面后可以看见管理员部分的主要功能有：

超级管理员管理功能：查看管理员、修改管理员个人信息、添加管理员和删除管理员；

一般管理员：查看用户信息、修改用户信息、添加用户和删除用户；

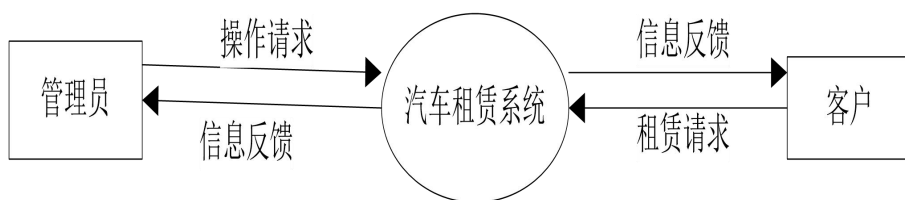
汽车管理：查询所有汽车租赁状况、添加汽车租赁记录、删除汽车租赁记录、修改汽车租赁状况、修改汽车租赁折扣、添加租赁汽车型号、查看租赁汽车型号和删除租赁汽车型号；

订单管理：查看订单；

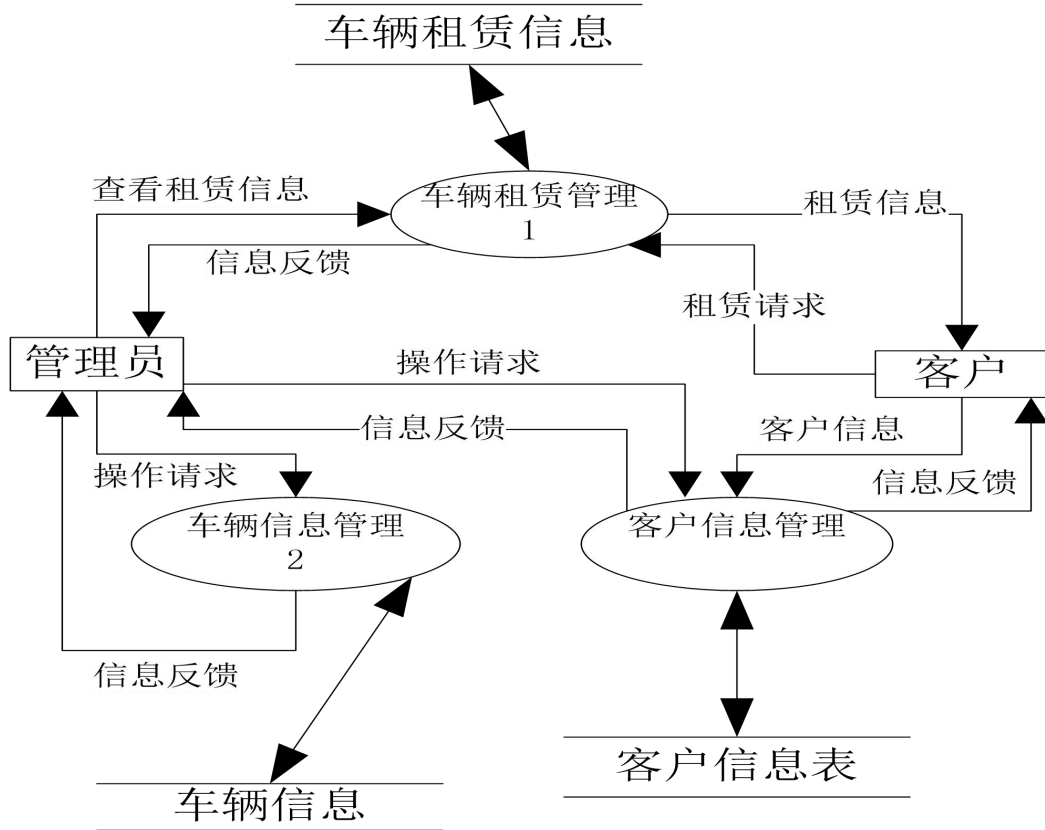
用户管理：查看用户、删除用户。

3.2 数据流图

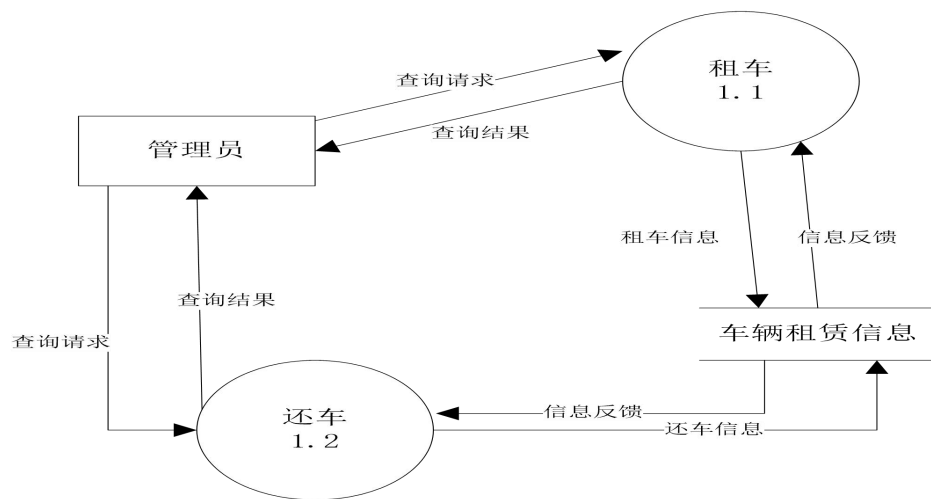
数据流图，简称 DFD，是 SA 方法中用于表示系统逻辑模型的一种工具，它以图形的方式描绘数据在系统中流动和处理的过程，由于它只反映系统必须完成的逻辑功能，所以它是一种逻辑模型。本系统具体数据流图如下[9]：



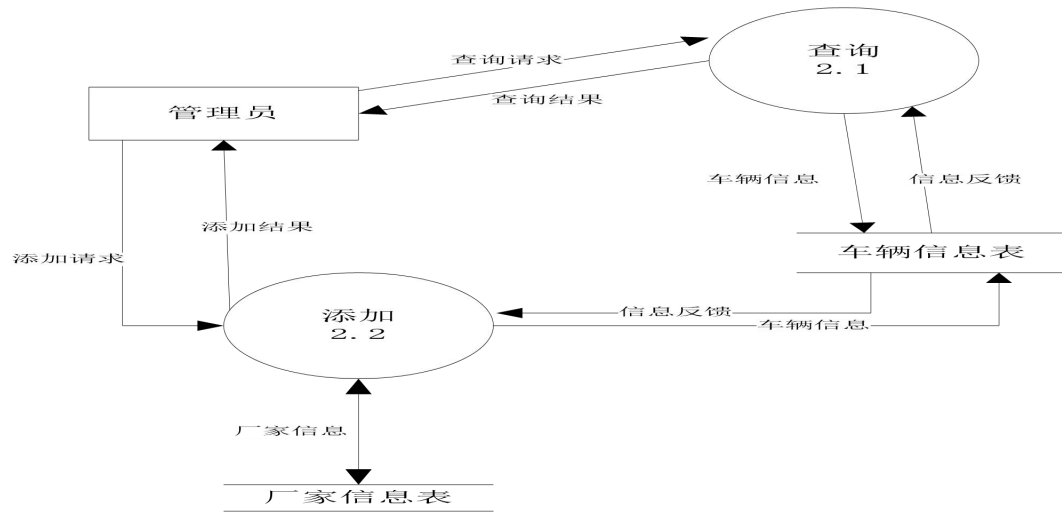
顶层数据流图



0层数据流图



一层数据流图



一层数据流图

3.3 数据字典

数据字典是关于数据信息的集合，它对数据流图中的各个元素做完整的定义与说明，是数据流图的补充工具。数据流图和数据字典共同构成的逻辑模型，没有数据字典的数据流图不严格，然而没有数据流图的数据字典也难发挥作用。只有两者一起，才能共同构成系统的规格说明。数据字典是开发数据库的第一步，本系统的数据字典。如下所系[9]:

名字: car

别名: 车辆信息

描述: 。汽车的入库、注销、租还

定义: car=车牌号+图片+车名+车辆颜色+发动机号+租赁价格+车辆类型+销售商+保险+状态+车架号

位置: 车辆管理, 汽车租赁, 车辆归还

b) 名字: admin

别名: 管理员信息

描述: 系统用户的名称

定义: admin=用户名+用户密码

位置：管理员信息

c) 名字：customer

别名：客户信息

描述：租赁汽车的人

定义：customer=客户编号+姓名+性别+年龄+身份证号+电话+工作单位+地址+
邮编+Email+驾驶证号+驾驶类型+驾龄+抵押证件

位置：用户信息

d) 名字：vips

别名：会员类型

描述：享有折扣的人

定义：vips=客户编号+姓名+性别+年龄+身份证号+电话+工作单位+地址+邮编
+Email+驾驶证号+驾临+抵押证件

位置：会员类型定义，会员信息，汽车租赁，车辆归还

e) 名字: company

别名: 汽车租赁公司

描述: 租车的企业

定义: company=公司名称+联系地址+联系电话+传真+Email+传真+联系人

位置: 车辆信息

f) 名字: lease

别名: 租赁业务信息

描述: 合同编号

定义: lease=汽车出租+汽车归还+汽车结算+合同编号

位置: 车辆租赁, 车辆归还

4. 总体设计

4.1 系统的总体设计

4.1.1 页面的设计

由于我采用了 B/S 结构, 摆脱了 C/S 传统的客户端和服务端一对一的关系, 同时我也注入了页面的显示, 用户和服务端打交道不再是传统的客户端代码和服务端代码的编写。本系统我实现了以下功能的页面设计。

汽车信息的查询和预定功能：此功能包含了汽车信息查询页面、查询结果显示页面、乘客信息填写页面和支付页面。

预定和退订功能：此功能包含了用户已预定的汽车的实现页面和退订的支付页面。

需先登录服务器后，才能对汽车信息进行增删查改等功能的操作。

关于这些功能我设计了以下的页面：

管理员登录功能：用来完成管理员的登录功能。

管理员管理功能：用来实现对管理员信息管理的提示页面。

汽车信息管理功能：用来实现对所有汽车信息进行增删查改等功能的提示页面。

订单管理功能：用来实现查看订单信息的提示页面。

用户管理功能：用来实现查看或删除用户信息的提示页面。

4.1.2 业务层的设计

业务层也就是功能实现的具体代码，B/S 结构中，业务层必须很好的实现表现层即页面显示代码和业务层之间代码的分离，这样方便以后的维护。

在业务层一般有实体层和业务逻辑处理层。同时也会有控制层，就是连接业务层和表现层的桥梁。

实体层：即是我们所做系统中需要的数据，我们将它们打包封装在一个类中，它没有任何实现代码，它的功能就是将业务数据映射到数据库中，在数据库中形成一张表来保存在系统中，生成我们所涉及到的一些业务数据。

业务逻辑处理层：具体的实现代码，是表现层所有功能实现的代码的集合，它能够根据页面上发送过来的请求来实现相应的功能，它是一个系统的核心，没有这一层，我们所有的操作都只是表象，所做的数据永远不能保存到数据库中。

控制层：连接表现层和业务逻辑层的桥梁。它的存在就是为了实现表现层和业务逻辑层的代码分离，降低他们的耦合性，方便以后更好的维护和功能的扩展。

用户端：

汽车信息查询：汽车保险，汽车租赁价格，汽车租赁状况，汽车类型，点击查询就能查找到相应的汽车信息。

汽车预定：在汽车信息查询的页面上，所查询到的汽车后都会有会有一个预定按钮，点击预定，转入到输入乘客信息和联系人信息，填写完之后就转入到了预定页面，确认无误后，完成了预定操作，相关的数据就已经更改。

汽车退订：用户登陆后，在首页栏上有“我的预定与退订”选项，在页面就会显示你所预定的汽车，在其后有一个“退订”操作，点击后会显示此汽车的详细信息，确认无误后，点击完成，相关的数据已经更改。

管理员：

汽车信息查询：管理员输入汽车的相关信息，点击查询就能查找到相应的汽车。

汽车信息删除：在查询页面的汽车信息后都有删除和修改操作，点击删除此汽车信息，更改数据库数据，完成删除。

修改汽车信息：在查询页面的汽车信息后都有删除和修改操作，点击修改，进入要修改的页面，填写要修改的详细信息后，确定无误后，点击完成，完成修改。

添加汽车信息：在管理员页面首页有一个增加汽车信息的选项，选中后转入要增加的汽车信息，填写汽车信息的信息后，提交，添加成功。

业务层设计中，我也主要是针对这些功能的描述来具体的实现它的功能。

4.2 系统业务流程设计

经过我的考虑和设计，系统流程如下：首先是由管理员登录此系统首页。登录成功后，该管理员就看查询、预定和退订汽车租赁，以及用户添加；

4.3 系统功能模块设计

此汽车租赁系统主要具有以下功能：

类型信息：包括汽车类型的添加、修改、删除；汽车保险类型的添加、修改、删除。具体信息包括分类名称、具体描述；

销售商信息：包括销售商信息的添加、修改、删除、查询、查看销售汽车。具体包括销售商的公司名称、联系地址、联系电话、传真、有正面吗、电子邮箱、联系人和备注；

车辆信息：包括车辆信息的查找、修改、删除、添加。具体信息包括车辆的 a 基本信息，有车牌号、车辆类型、车辆名称、车辆颜色、发动机号、销售商和状态；b 保险信息，有保险公司、保险单、开始日期、结束日期和保险种类；c 租赁价格信息，有押金、日租金、周末租金、月租金车辆描述；

客户信息：包括客户信息的查找、修改、添加、删除和加入会员。具体信息包括 a 基本信息，有客户编号、姓名、性别、年龄、身份证号、电话、工作单位、地址、邮编和 Email；b 驾驶证信息，有驾驶证号、驾驶类型、驾龄、发证日期和失效日期；c 抵押担保信息，有抵押证件、担保人、担保人身份证和担保人单位；

会员信息：包括会员信息的查找、修改、注销。具体信息包括 a 基本信息，有客户编号、姓名、性别、年龄、身份证号、电话、工作单位、地址、邮编和 Email；b 驾驶证号、驾驶类型、驾龄、发证日期和实效日期；c 抵押担保信息，有抵押证件、担保人、担保人身份证和担保人单位，d 会员信息，有会员类型和享受折扣；

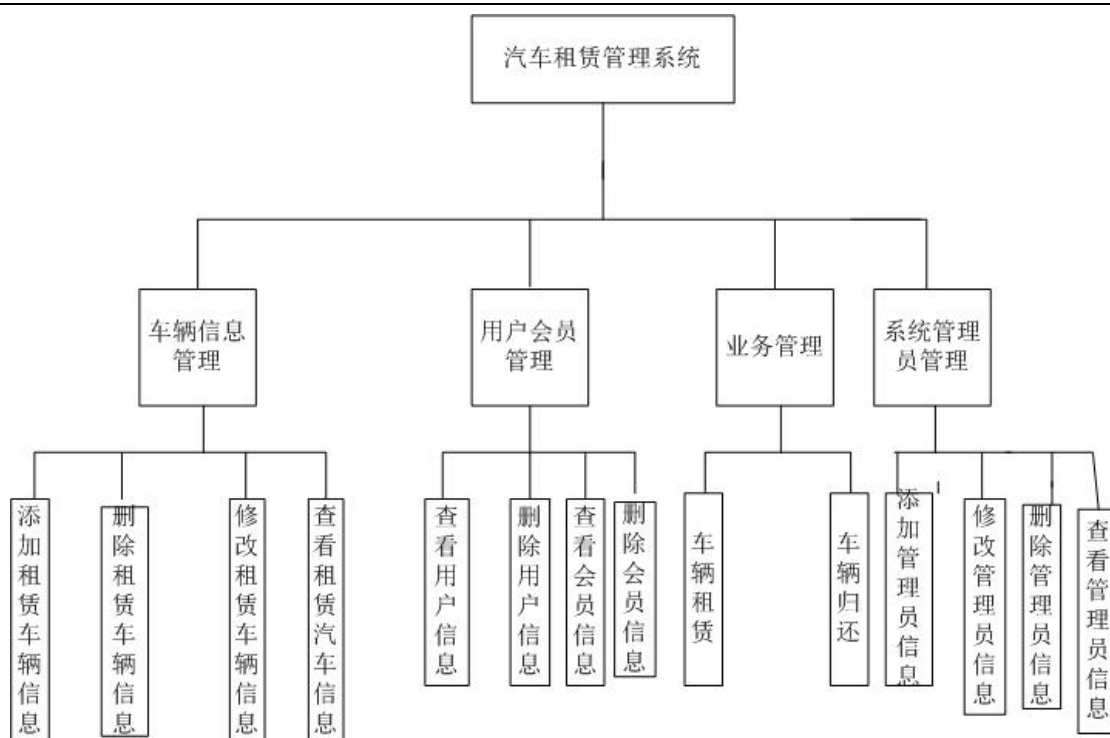
会员类型：包括会员类型的添加、修改和删除。具体信息包括类型名称和折扣率；

汽车租赁：包括汽车租赁信息的添加、修改、删除、租赁审核、续租和续租审核。具体信息包括合同编号、车牌号码、客户编号、租赁模式、工作日数、租车时间、还车时间、租车费用和享受折扣；

汽车归还：包括汽车归还信息的归还、修改、结算审核。具体信息包括合同编号、回车公里数。

系统用户信息：包括添加用户和删除用户，具体包括用户名、密码和确认密码。系统帮助信息：包含软件使用过程中的一些重要的，需要注意的使用方法和其他的一些必要的信息。

在系统功能分析的基础上，按结构化程序设计思想，分析的系统功能图如图所示：



4.4 数据库设计

4.4.1 数据库概念结构设计

数据库概念设计是产生反映实验成绩管理信息需求的数据库概念结构，即概念模式。反映其概念模式是用 E-R 模型。概念结构是独立于任何一种数据模型的信息结构。逻辑结构设计任务就是把概念结构设计阶段设计好的 E-R 图转换为与选用 DBMS 产品所支持的数据模型相符合的逻辑结构。在概念设计阶段中，要从用户的角度看待数据及处理要求和约束，产生一个反映用户观点的概念模式。然后再把概念模式转成逻辑模式[1]。

4.4.2 数据库逻辑结构设计

数据库的设计在一个项目中是一个非常重要的环节。数据是一切系统设计的基础，通俗地讲，数据库设计就像高楼大厦的根基一样，如果设计不合理，不完善，将在系统开发过程中，甚至到后期的系统维护，功能变更和功能扩充时，引起较多问题，严重时甚至要重新设计重做已完成的工作。

4.4.3 数据库 E-R 如下：

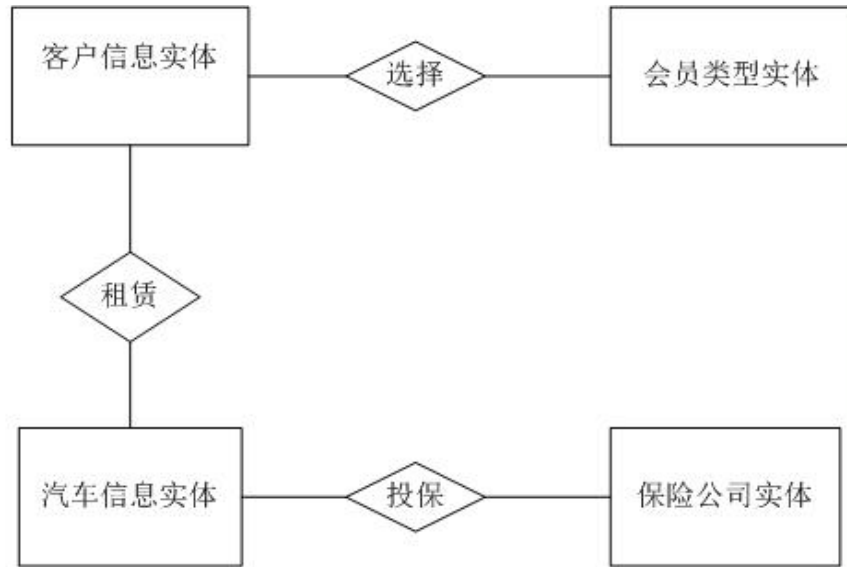


图 4-3-3 各实体之间关系 E-R 图

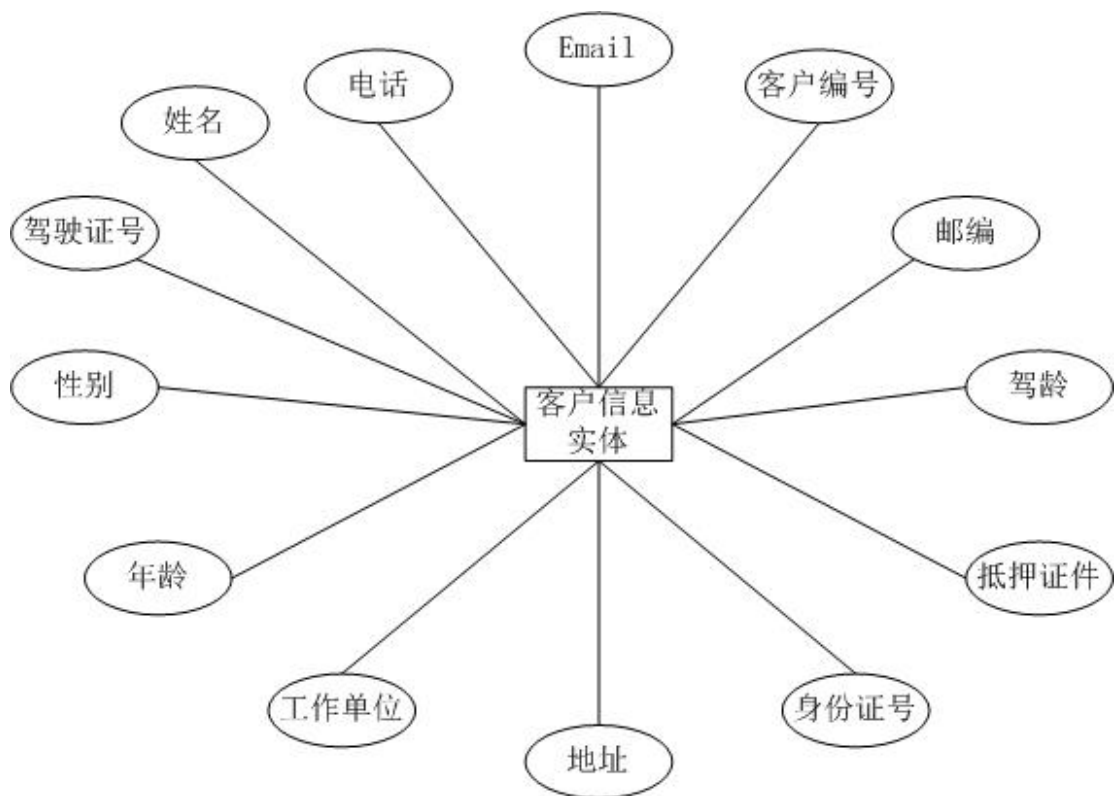


图 4-3-3 客户信息实体图



图 4-3-3 类型信息实体图

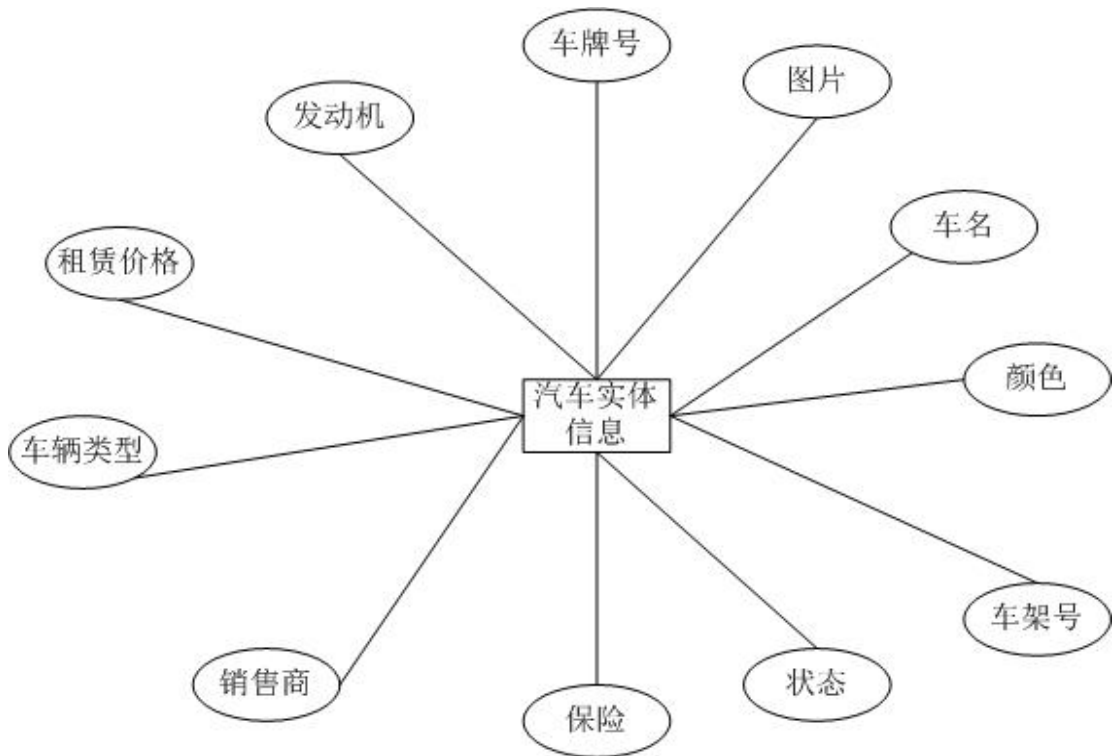


图 4-3-3 车辆信息实体图

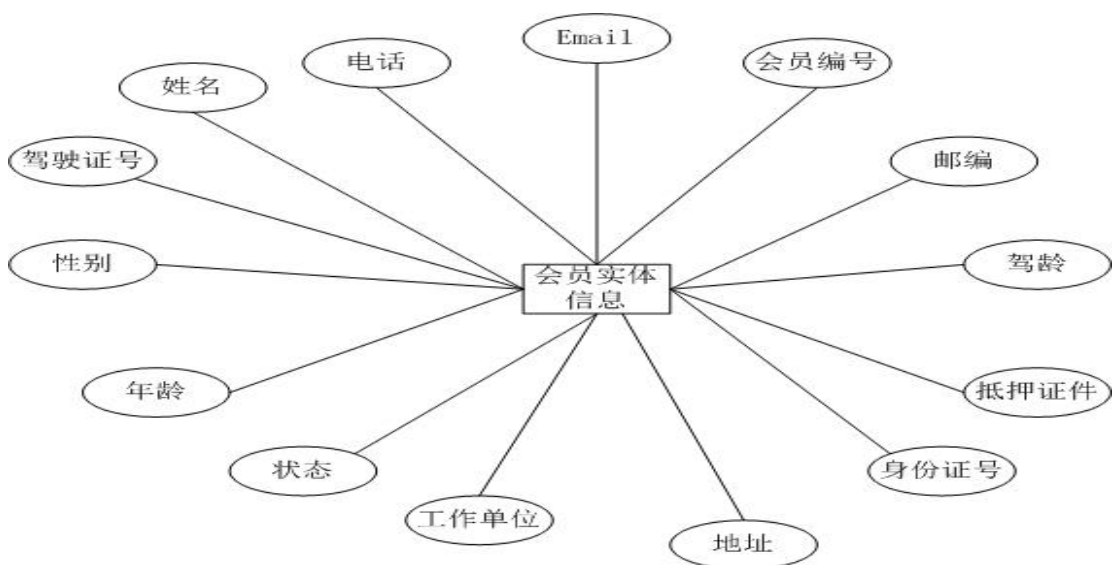


图 4-3-3 会员类型实体图

将 E-R 图转换为关系模型

管理员表（登录名，真实姓名，密码，联系电话，邮箱）

用户表（登录名，真实姓名，密码，性别，证件号码，联系电话，邮箱，会员类型）

汽车信息表（车牌号，编号，车架，租赁价格，保险信息，折扣）

销售商信息表（公司名称，联系地址，联系电话，Email，传真，邮编，联系人，备注）

4.4.4 数据物理设计：

数据库在物理设备上的存储结构和存取方法称为数据库的物理结构，它依赖于给定的计算机系统。为一个给定的逻辑数据模型选取一个最合适应用要求的物理结构的过程，就是数据库的物理设计。

根据数据逻辑结构设计，本系统主要有如下 7 张表：

管理员表（car_adimn）：用于存放管理员用户的数据记录。

用户表（car_user）：用于存放用户的基本信息。

销售商表（car_producers）：用于存放销售商基本信息。

汽车信息表（car_car）：用于存放租赁汽车的信息。

订单表（car_orders）：用于存放用户预定汽车的基本信息。

类型表（car_type）：用于存放类型的信息。

会员表（car_vip）：用于存放会员信息。

以上 7 个数据表的字段说明如几个表所示。

表 4-1 车辆信息表

字段名	类型	宽度	是否为空
id	整型	10	不为空
carNo	字符串型	50	可以
typeId	字符串型	50	可以
picture	字符串型	50	可以
carName	字符串型	50	可以
carColour	字符串型	50	可以
carPrice	整型	10	可以
carRemark	整型	10	可以



engineType	字符串型	50	可以
producersId	整型	10	可以

表 4-2 客户信息表

字段名	类型	宽度	是否为空
id	整型	10	不为空
name	字符串型	50	可以
sex	字符串型	50	可以
address	字符串型	50	可以

phone	字符串型	50	可以
email	字符串型	50	可以
identityCode	字符串型	50	可以
loginId	字符串型	50	不为空

表 4-3 用户信息表

字段名	类型	宽度	是否为空
id	整型	10	有
username	字符串型	50	有
userpwd	字符串型	50	无

表 4-4 公司信息表

字段名	类型	宽度	索引
id	整型	10	不为空
company	字符串型	50	可以
telephone	字符串型	50	可以
address	字符串型	50	可以

表 4-5 类型信息表

字段名	类型	宽度	是否为空
id	整型	10	不为空
typeName	字符串型	50	可以

typeId	整型	10	可以
description	字符串型	50	可以

表 4-6 租赁信息表

字段	类型	宽度	是否为空
id	整型	10	不为空
contractNo	字符串型	50	可以
carNo	字符串型	50	可以
leasMode	字符串型	50	可以
leaseTime	字符串型	50	可以

userName	字符串型	50	可以
returnTime	字符串型	50	可以

表 4-7 会员信息表

字段名	类型	宽度	是否为空
id	整型	10	不为空
typeName	字符串型	50	可以
discount	双精度	10	可以

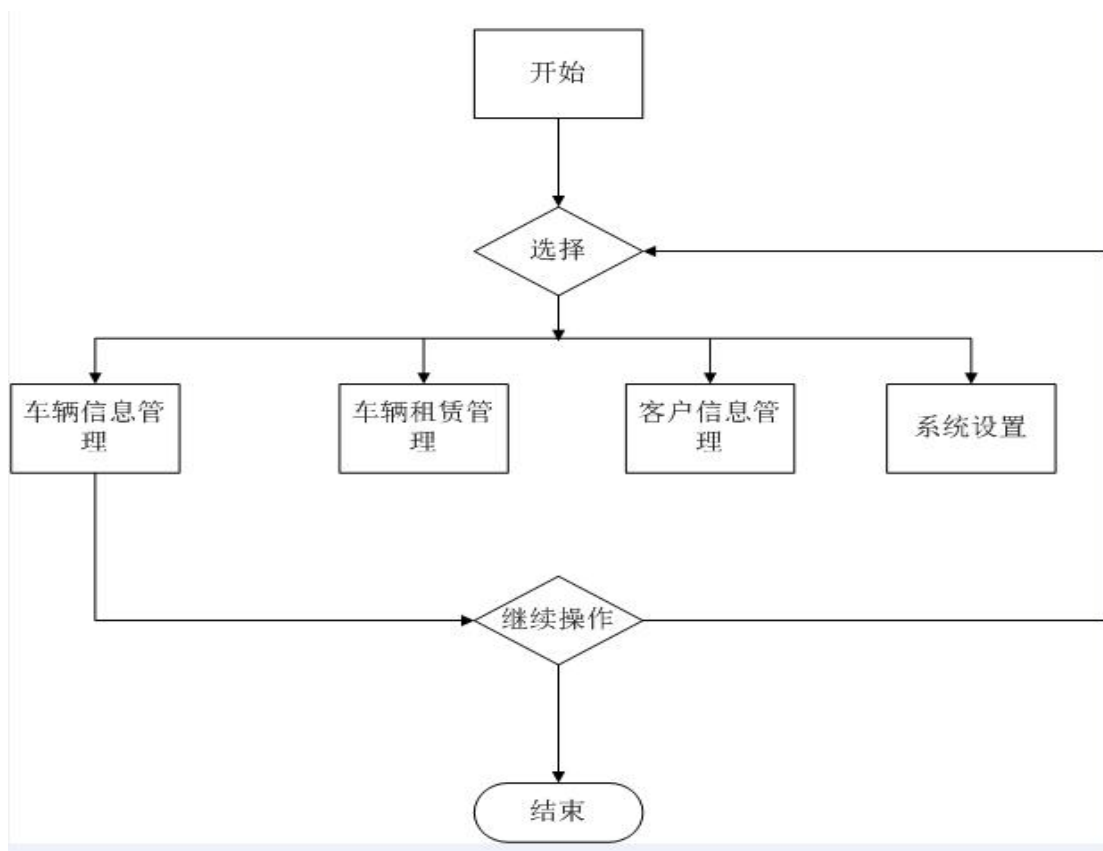
5. 详细设计与代码实现

以系统的总体设计为基础, 现在可以开始系统的实现工作, 系统是包括编码和测试, 关于测试, 将在下一张具体说明。本章主要叙述代码的实现。

编码就是把软件设计结果翻译成用某种程序设计语言书写的程序。座位软件工程的一个阶段, 编码是对设计的进一步具体化。因此, 程序的质量只要取决于软件设计的质量。而所选用的程序设计语言的特点及编码风格也将对程序的可靠性、可读性、可测试性和可维护性也将产生深远的影响。

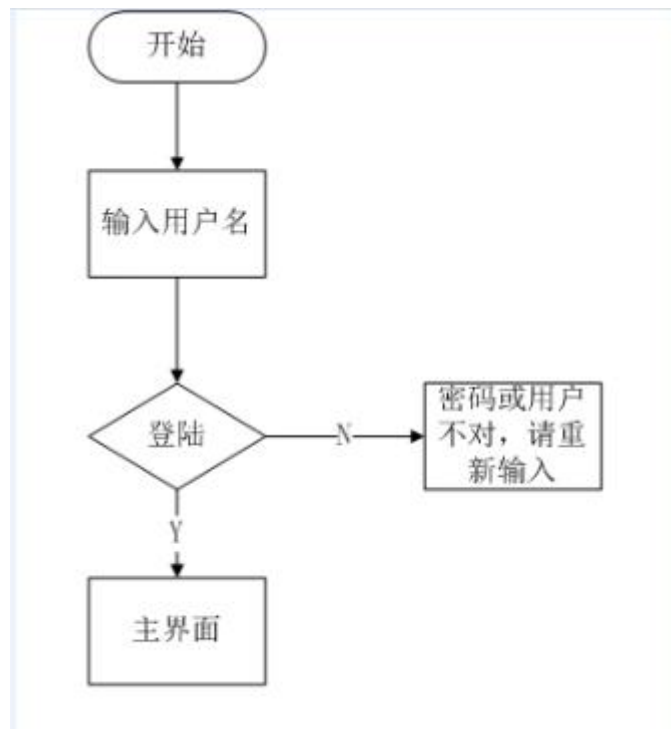
5.1 主控程序流程设计

本系统为实现汽车租赁业务的信息化而设计, 起主控程序流程图:



5.2 后台登陆模块的设计与实现

为了提高数据的安全性, 系统设置了防止非法人员进入系统的功能。系统在开始运行的时候, 首先进入用户登录界面, 再由操作人员输入用户名和密码后吗, 系统进行判定进入系统的合法身份, 此后, 方可正式进入该系统。如果用户名输入有错, 系统将显示“用户名或密码错误”提示。程序流程图如下:



用户登录流程图

功能描述：管理员成功登陆后台，直接进入到后台管理的界面，可以对管理员、汽车租赁、订单和用户进行增删查改等操作。

界面如下：



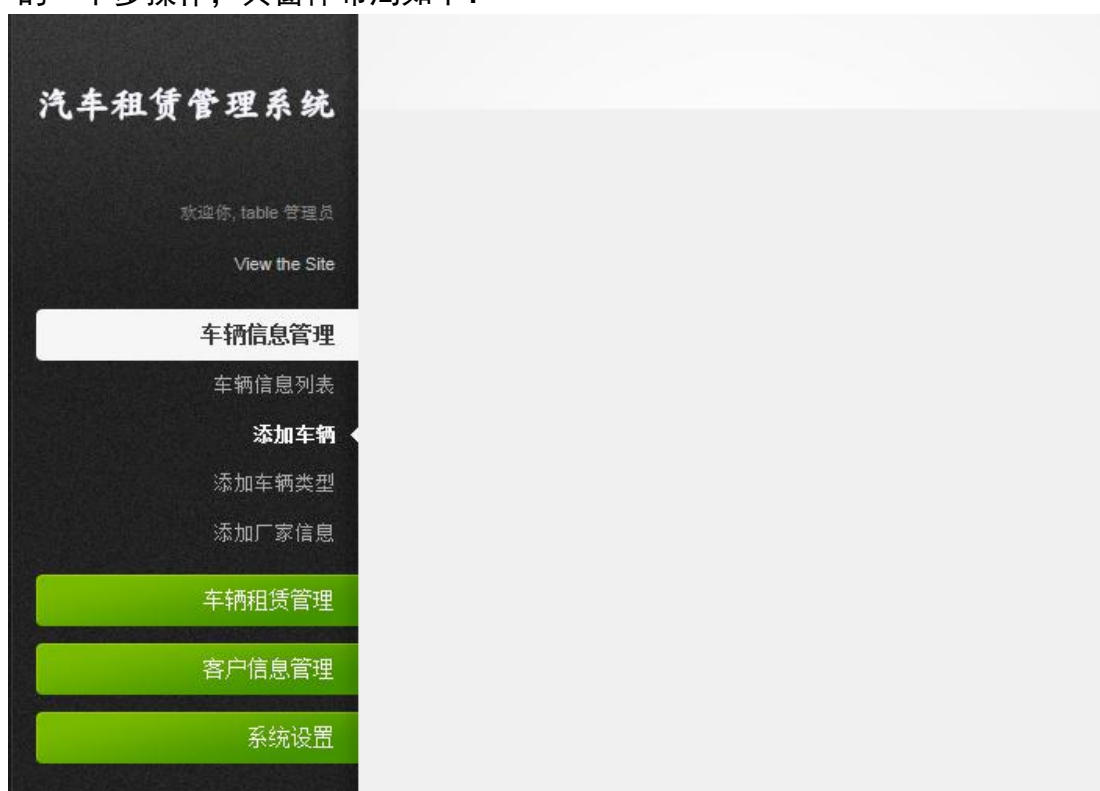
主要代码如下：

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

```
protected void Submit1_ServerClick(object sender, EventArgs e)
{
    BLL.Admin admin_BLL = new BLL.Admin();
    Model.Admin admin_M = new Model.Admin();
    admin_M.LoginId = LoginId.Value.Trim();
    admin_M.Password = password.Value.Trim();
    admin_M = admin_BLL.judgeAdminInfo(admin_M);
    if (admin_M.Name != null)
    {
        Response.Redirect("index.aspx");
    }
    else
    {
        lblMsg.Text = "用户名或密码错误！";
    }
}
}
```

5.3 主界面设计

用户登录成功后，即进入系统主界面。主界面的外观设计应大方美观，布局合格，方便用户的一下步操作，其窗体布局如下：



5.3.1 管理员管理模块的设计和实现

管理员进入管理员管理这一页面主要是对管理员进行增、删、改、查等操作。

5.3.1.1 查看管理员模块

界面如下：

查询管理员信息集合									
编号	真实姓名	登录名	密码	性别	年龄	地址	电话	身份证号	操作
2	张尹	admin	123	男	23	贵州财经学院	15329000213	522124198801241215	删除
3	刘皓	123	123	男	22	贵州财经学院	15829000851	522124198801531536	删除

其主要代码如下：

```
public List<Model.Admin> QueryAdminAll()
{
    String strSql = "select * from Admin";
    SqlCommand cmd = new SqlCommand(strSql, Model.DBHelp.con);

    List<Model.Admin> list = new List<Model.Admin>();

    //判定数据库
    if (Model.DBHelp.con.State == ConnectionState.Closed)
    {
        Model.DBHelp.con.Open();
    }

    //创建数据库数读取对象
    SqlDataReader sdr = cmd.ExecuteReader();
```

```
//判断是否存在数据
while (sdr.Read())
{

    //创建一个管理员实体类对象
    Model.Admin admin_M = new Model.Admin();

    //给管理员对象的 姓名属性 赋值
    admin_M.Id = int.Parse(sdr["Id"].ToString());
    admin_M.Name = sdr["name"].ToString();
    admin_M.Age = sdr["age"].ToString();
    admin_M.Phone = sdr["phone"].ToString();
    admin_M.Sex = sdr["sex"].ToString();
    admin_M.LoginId = sdr["loginId"].ToString();
    admin_M.Password = sdr["password"].ToString();
    admin_M.IdentityCode = sdr["IdentityCode"].ToString();
    admin_M.Address = sdr["Address"].ToString();

    //将查询出来的对象添加到管理员实例 添加到 泛型集合中
    list.Add(admin_M);
}

if (Model.DBHelp.con.State == ConnectionState.Open)
{
    Model.DBHelp.con.Close();
}
//返回管理员对象
return list;
}

public Model.Admin QueryAdmin(Model.Admin admin)
{
    String strSql = string.Format("select * from Admin where id = {0}", admin.Id);
    SqlCommand cmd = new SqlCommand(strSql, Model.DBHelp.con);
    //创建一个管理员实体类对象
    Model.Admin admin_M = new Model.Admin();

    //判定数据库
    if (Model.DBHelp.con.State == ConnectionState.Closed)
    {
        Model.DBHelp.con.Open();
    }
}
```

```
//创建数据库数读取对象
SqlDataReader sdr = cmd.ExecuteReader();
//判断是否存在数据
if (sdr.Read())
{
    //给管理员对象的 姓名属性 赋值
    admin_M.Id = int.Parse(sdr["Id"].ToString());
    admin_M.Name = sdr["name"].ToString();
    admin_M.Age = sdr["age"].ToString();
    admin_M.Phone = sdr["phone"].ToString();
    admin_M.Sex = sdr["sex"].ToString();
    admin_M.LoginId = sdr["loginId"].ToString();
    admin_M.Password = sdr["password"].ToString();
    admin_M.IdentityCode = sdr["IdentityCode"].ToString();
    admin_M.Address = sdr["Address"].ToString();
}

if (Model.DBHelp.con.State == ConnectionState.Open)
{
    Model.DBHelp.con.Close();
}
//返回管理员对象
return admin_M;
}
```

5.3.1.2 添加管理员个人信息

界面如下：



主要代码如下：

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        BLL.Admin admin_BL1 = new BLL.Admin();
        Model.Admin admin_M = new Model.Admin();
        admin_M.Address = txtAddress.Text;
        admin_M.Age = txtAge.Text.Trim();
        admin_M.IdentityCode = txtIdentityCode.Text;
        admin_M.LoginId = txtLoginId.Text;
        admin_M.Name = txtName.Text;
        admin_M.Password = txtPassword.Text;
        admin_M.Phone = txtPhone.Text;
        admin_M.Sex = txtSex.Text;

        Boolean coo = admin_BL1.insertAdmin(admin_M);
        if (coo)
        {
            Response.Redirect("AdminList.aspx");
        }
        else
        {
            Response.Write("<script>alert('添加管理员信息失败！请重试。')</script>");
        }
    }
}
```



}

}

5.3.2 查询汽车租赁模块的设计和实现

该模块主要是汽车租赁信息的增、删、查、改等操作以及查看汽车型号和删除汽车型号。

5.3.2.1 查询所有汽车租赁信息模块

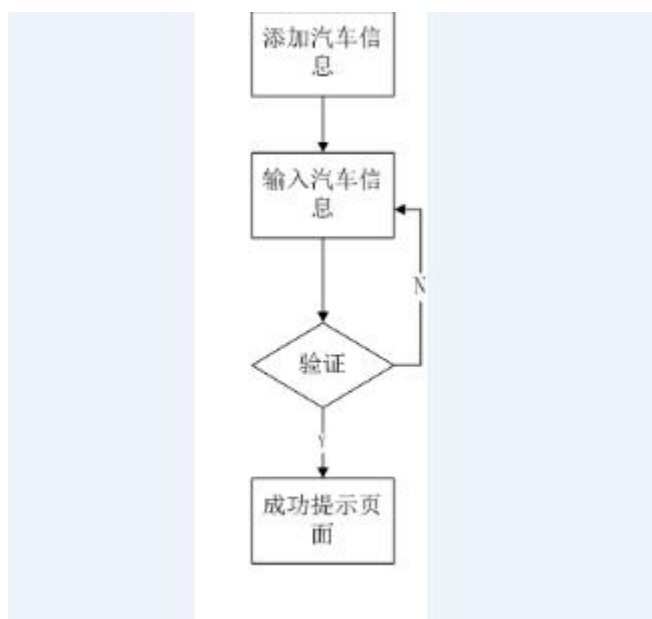
为了想出更多让企业盈利的方法，企业一个月要结算一次帐，看看本月租车的效益如何，看看客户更趋向租什么样的汽车，查查那些汽车在租车行业根本就没什么必要存在，就果断的不在购买这种车来租赁，而购买那些人们更喜欢租的汽车来租赁，会大大的提高企业的盈利。

界面如下：

查询车辆信息集合											
车辆编号	生产厂家	所属类型	车牌号	车辆名称	图片	颜色	发动机号	出租价格	出租状态	编号	删除
1	奔驰汽车厂	跑车	贵A 9999	奔驰		黑色	1.8T 204马力 14	3000.00	未出租	编辑	删除
2	美国AMG公司	中大型	贵A 1234	奔驰TT		白色	1.8T 184马力 L4	2000.00	未出租	编辑	删除
3	美国AMG公司	中大型	贵A 1314	皇冠		黑色	3.0L 231马力 V6	3400.00	未出租	编辑	删除

5.3.2.2 添加租赁汽车信息模块

为了成功的添加租赁汽车的信息，必须通过汽车信息验证，如果输入汽车租赁编号在数据库中不存在，则添加成功，否则饭后本页面重新输入。如果添加失败，则重新输入租赁汽车信息。具体程序流程图如下：



增加租赁汽车信息流程图

界面如下：

添加汽车租赁记录

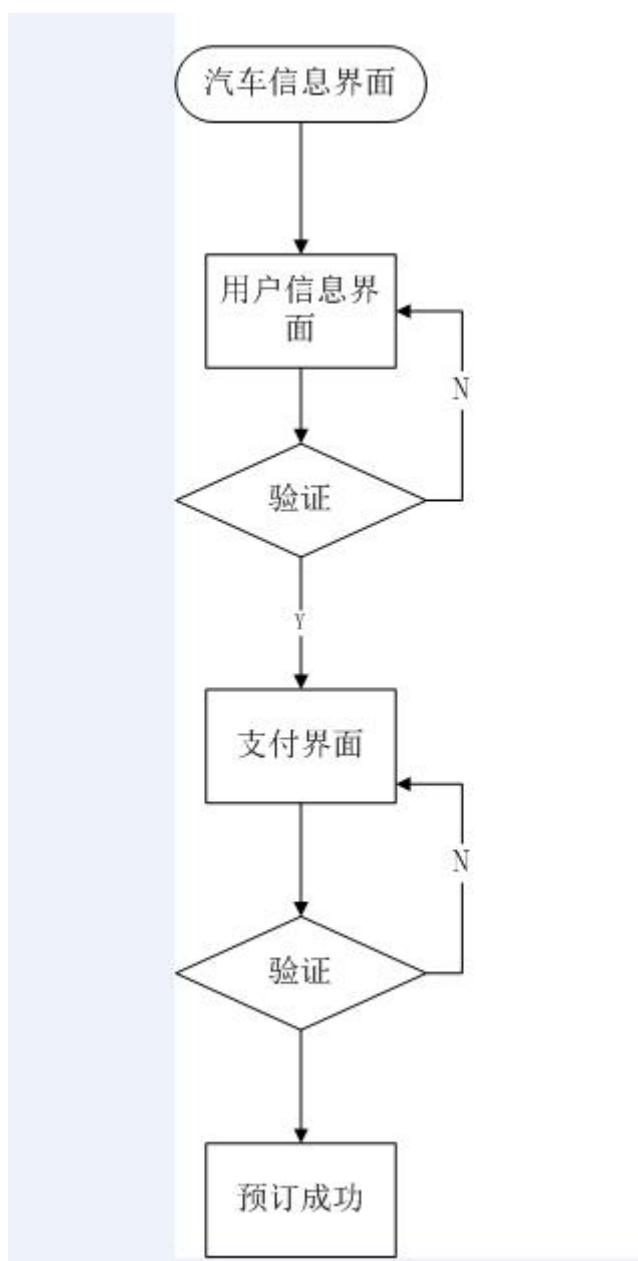
请选择汽车:	奔驰
请选择用户:	小王
期限:	
预付金额:	
应付金额:	
租赁订单状态:	
<button>确认添加</button>	

5.3.3 订单管理模块的设计与实现

该模块主要是对用户订单的查看与删除操作。

5.3.3.1 查看汽车预订信息模块

为了方便客户租赁方便，特意为客户开发了一个预定汽车的功能，方便客户在有时间和想租车外行的意愿，提高汽车租赁效率和服务质量。



界面如下：

查询汽车租赁信息									
订单编号	车型名称	出租人姓名	出租时间	出租期限	已付押金	结算金额	订单状态	编辑	删除
1	奔驰	小王	2011/4/28 17:16:10	7天	10000.00	5000.00	未结算	编辑	删除
2	奥迪TT	小何	2011/4/28 17:16:44	2天	5000.00	3000.00	未结算	编辑	删除
3	皇冠	小张	2011/4/28 17:17:19	1天	3000.00	1500.00	已结算	编辑	删除

5.3.4. 用户管理模块的设计与实现

该模块主要是对用户的查看和删除等操作。

5.3.4.1 查看用户信息模块

界面如下：

查询用户信息集合									
用户编号	VIP类型	真实姓名	年龄	身份证	性别	地址	电话	邮箱	删除
1	普通用户	小王	22	5221241988014231	男	贵州省遵义市	15829020965	11@yahoo.com.cn	删除
2	VIP1	小何	31	522123198010113427	女	贵州贵阳	15823309654	123@yahoo.com.cn	删除
3	VIP3	小张	25	522124198612150124	男	贵州贵阳	13984210090	asdf@qq.com	删除

主要代码如下：

```
public Model.User QueryUser(Model.User user)
```

```
{
```

```
string strSql = string.Format("SELECT * FROM Users WHERE id = {0}",
user.Id);

//创建数据库命令对象

SqlCommand cmd = new SqlCommand(strSql, Model.DBHelp.con);

//创建用户实体类对象

Model.User user_M = new Model.User();

//判断数据库

if (Model.DBHelp.con.State == ConnectionState.Closed)

{

    Model.DBHelp.con.Open();

}

//创建数据库读取对象

SqlDataReader sdr = cmd.ExecuteReader();
```



//判断数据

```
if (sdr.Read())
```

```
{
```

```
    user_M.Id = int.Parse(sdr["id"].ToString());
```

```
    user_M.Name = sdr["name"].ToString();
```

```
    user_M.Age = sdr["age"].ToString();
```

```
    user_M.VipId = int.Parse(sdr["vipId"].ToString());
```

```
    user_M.Sex = sdr["sex"].ToString();
```

```
    user_M.Address = sdr["Address"].ToString();
```

```
    user_M.Phone = sdr["phone"].ToString();
```

```
    user_M.Email = sdr["email"].ToString();
```

```
    user_M.IdentityCode = sdr["identityCode"].ToString();
```

```
    user_M.Password = sdr["password"].ToString();
```



```
}
```

```
if (Model.DBHelp.con.State == ConnectionState.Open)
```

```
{
```

```
    Model.DBHelp.con.Close();
```

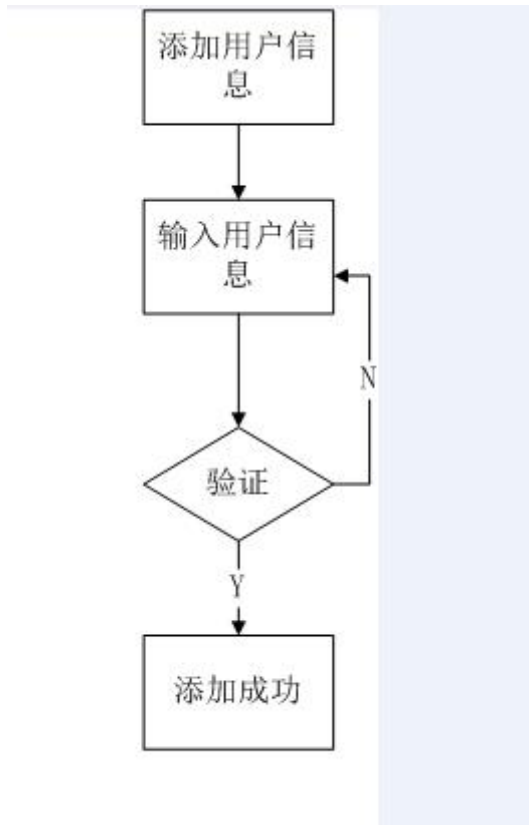
```
}
```

```
return user_M;
```

```
}
```

5.3.4.2 添加用户信息模块

用户信息添加系统流程图如下



界面如下：



The form is titled '添加用户信息' (Add User Information) in bold black text at the top center. It contains several input fields and a dropdown menu, all with light gray backgrounds and white text. The fields are arranged vertically: '真实姓名:' (Real Name), '请选择用户的VIP等级:' (Please select the user's VIP level) with a dropdown menu showing '普通用户' (General User), '密码:' (Password), '身份证:' (ID Card), '性别:' (Gender), '年龄:' (Age), '联系电话:' (Contact Phone), '联系地址:' (Contact Address), and '邮箱:' (Email). At the bottom left, there is a blue button with white text labeled '确认添加' (Confirm Add).

6. 结论

经过 2 个多月的时间吗，汽车租赁管理系统基本开发完成。在本次毕业设计中，我应用 B/S 结实现了企业对汽车租赁信息的查询和租赁汽车的一些功能,实现了管理员登录，管理员查询汽车租赁信息，增加租赁汽车信息，删除租赁汽车信息，使得汽车租赁智能化，使汽车租赁行业发展前景越来越好。

通过毕业论文的设计与开发，我们学到了很多知识，也掌握了一些开发工具的使用，充实了自己，为自己以后能得到更大的提高打了一个好的基础。同时，我们也知道自己存在很多的不足之处。我相信，在今后的学习中，一定会加强学些，总结经验，为今后的工作学习中能去的优异的成绩而做好充分的准备！

由于个人水平有限，在设计思路和技巧方法上可能比较愚钝，在开发语言的掌握上可能相对肤浅，所以反映到系统中难免存在许多的不足之处，殷切希望各位老师批评、指正。

参考文献:

- [1]丁宝康.数据库使用教程.清华大学出版社, 2004 年
- [2]汪诗林等.数据结构算法与应用.机械工业出版社, 2005 年
- [3]王小科 吕双.java 从入门到精通.清华大学出版社, 2008 年。
- [4]冯博琴等.面向对象分析与设计.机械工业出版社, 2003 年。
- [5] 刁成嘉, 刁奕: UML 系统建模与分析设计.北京:机械工业出版社, 2008 年。
- [6] 瞿中, 金文标.李伟生.Java 面向对象程序设计(第 3 版).北京: 清华大学, 2008 年。
- [7]曹祖圣, 吴明哲.java 程序设计经典。北京: 科学出版社, 2004 年。
- [8]宣小平.java 数据库系统开发实例导航。上海人民邮电出版社, 2003 年。
- [9]张海潘.软件工程.北京人民邮电出版社, 2002 年。

致谢

非常感谢我的指导老师老师对我的支持和帮助以及 XXX 等老师的热心指导非常感谢在系统和论文上帮助过我的同学。谢谢你们的支持和鼓励。

