**《共享单车管理系统的开发》**



**第八组**

**成员：曾胜坚 唐光宇 谷银珂**

# 1.问题分析

## 1.1背景

现如今，共享单车行业蓬勃发展，每天都有成千上万辆共享单车在城市中运行。这些共享单车需要一个高效的信息管理系统来帮助日常管理工作，因此设计一套科学、有效的单车信息管理系统具有现实的意义。

该系统投入运行后，可以有效减少公司日常运营的压力和开销。通过将单车的信息管理电子化和统一化，可以提高公司对单车相关事务的处理效率。例如，系统可以实时记录和管理单车的位置、状态和使用情况，帮助公司更好地分配和维护单车资源，避免资源浪费和损耗。同时，系统还可以自动生成报表和统计数据，为公司的决策和规划提供依据。

此外，设计的单车信息管理系统应该使用简单、直观，不给公司带来其他压力。系统应该具备友好的用户界面和操作流程，使员工能够轻松上手并高效地完成各项任务。同时，系统还应该具备稳定性和安全性，确保数据的准确性和保密性。

综上所述，开发一套单车信息管理系统具备社会可行性。通过科学、有效的信息管理，可以提升共享单车公司的运营效率和服务质量，促进共享单车行业的健康发展。

## 1.2需求分析

系统目标

本共享单车系统旨在为用户提供便捷的共享单车服务，同时为管理员提供高效的管理工具。系统需实现车辆信息管理、用户信息管理、订单管理、车辆调度与维修、报表与统计数据生成等功能。同时，为用户提供注册、借车、充值等操作界面，为维修人员提供维修任务的接收和完成等功能。

主要实现的功能如下:

（1）管理员车辆信息管理：这是一项基础功能，管理员可以通过系统对车辆信息进行添加、修改、查询和删除。这包括了车辆的基本信息如车型、颜色、编号等。

（2）管理员用户信息管理：同样，管理员也可以通过系统对用户信息进行添加、修改、查询和删除。这包括了用户的基本信息如用户名、密码、联系方式等。

（3）订单管理：管理员可以通过系统查询订单，取消订单，以及生成订单。这项功能涉及到订单的的状态管理，例如待接单、进行中、已完成等。

（4）车辆调度与维修：管理员可以对车辆进行调度和维修，系统会记录车辆的调度和维修历史。这项功能需要结合车辆的实时位置信息，以及维修人员的分布情况进行。

（5）报表与统计数据：管理员可以生成各类报表和统计数据，以便于分析运营情况和做出决策。这可能包括用户活跃度、订单量、车辆使用率等。

（6）用户注册与借车：用户可以通过系统完成注册，借车以及充值等操作。这项功能需要考虑到用户的操作便利性，以及账户安全等问题。

（7）维修人员维护：维修人员可以通过系统接收维修任务，完成维修工作，并记录维修历史。这项功能可以提高维修工作的效率，以及维修质量。

# 2.可行性研究

在进行项目的实际开发前，要进行可行性研究，目的是以最小的代价在短时间内确定软件项目是否值得开发，是否可以实现。对共享单车管理系统可以从经济可行性、技术技术可行性和社会可行性三个方面来论证。

## 2.1经济可行性

经济可行性的目的是估算开发成本，确定项目值得投资。共享单车管理系统的开发成本包括软件开发单车的工资、服务器和硬件设备的购买成本等。同时，系统的运营和维护成本也需要考虑。通过对市场需求和预期收入的分析，可以评估系统的经济可行性。

（1）现有的共享单车管理系统采用集中式管理不仅繁琐而且抗风险能力较差，导致共享单车的管理较为困难和麻烦。与此相比，采用这种分布式系统的成本和精力投入要低于现有系统，并且系统上手容易，操作简便。个人可以节省大量时间和精力，公司可以减少雇员资金和管理的成本，开发者可以更容易地维护系统和适应用户和技术的更新。

（2）软件开发周期不应拉的过长，也不能过短，周期过长容易导致软件与预想差距过大，进度拖拉，周期过短导致功能不全面。作为一个实验项目，开发周期应在八到九周比较合理，这样能够有足够的实际来完成规划，构思，实现。

（3）项目中所使用的开发工具软件（Java SE, IDEA Community, MySQL Community）等均为开源软件,降低了项目的经济压力，让开发单车能够专门投入于开发过程中。

综上所述，共享单车信息管理系统在经济上是可行的。

## 2.2技术可行性

技术可行性要对项目的功能和限制条件进行分析，目的是确定项目是否能实现。一般要包括开发风险和技术风险。

在开发风险方面，由于使用的大部分软件都是开源的，且软，硬件要求都不高，所以开发的费用较低，本系统使用的技术都比较成熟，故开发风险较低。

公司单车信息管理系统不受限于系统，可跨平台运行，系统使用的技术大都比较成熟，出错的概率较低，性能出色。在开发技术方面选择Java SE作为实现语言，包括JDBC（java数据库编程），java web在内的所用技术都比较成熟，可用用于本项目，同时使用结构化方法进行开发，在技术层面是完全可行的。

## 2.3社会可行性

现如今，共享单车行业蓬勃发展，每天都有成千上万辆共享单车在城市中运行。这些共享单车需要一个高效的信息管理系统来帮助日常管理工作，因此设计一套科学、有效的单车信息管理系统具有现实的意义。

该系统投入运行后，可以有效减少公司日常运营的压力和开销。通过将单车的信息管理电子化和统一化，可以提高公司对单车相关事务的处理效率。例如，系统可以实时记录和管理单车的位置、状态和使用情况，帮助公司更好地分配和维护单车资源，避免资源浪费和损耗。同时，系统还可以自动生成报表和统计数据，为公司的决策和规划提供依据。

此外，设计的单车信息管理系统应该使用简单、直观，不给公司带来其他压力。系统应该具备友好的用户界面和操作流程，使员工能够轻松上手并高效地完成各项任务。同时，系统还应该具备稳定性和安全性，确保数据的准确性和保密性。

综上所述，开发一套单车信息管理系统具备社会可行性。通过科学、有效的信息管理，可以提升共享单车公司的运营效率和服务质量，促进共享单车行业的健康发展。

# 3. 面向对象分析

## 3.1 建立用例模型

### 业务需求：

一、用户注册与登录

用户在平台上注册，提供有效身份证件、手机号码、密码等个人信息。

系统对用户提交的信息进行审核，审核通过后，用户成为平台会员。

用户通过手机号码和密码登录系统，查询车辆信息、发起订单、充值等操作。

二、车辆借还与充值

用户在平台上查询可用车辆，选择目标车辆并发起借车请求。

系统确认车辆借车请求，更新车辆状态为“借出”。

用户在借车期间可以使用车辆，还车时通过系统发起还车请求。

系统确认车辆还车请求，更新车辆状态为“可借”。

用户可以在平台上为账户充值。

三、订单管理

用户发起订单请求，系统生成订单编号并记录订单信息。

系统更新订单状态为“已完成”。

用户可以查询订单状态，了解订单进展。

四、车辆调度与维修

管理员通过系统查看车辆位置，根据车辆位置进行调度。

管理员可以为车辆分配维修任务，系统推送给维修人员。

维修人员接收维修任务，查看车辆故障并进行维修。

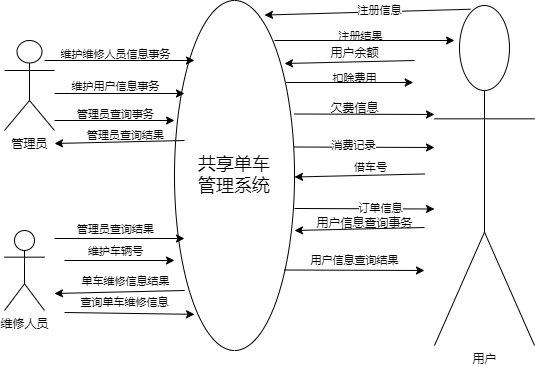
维修人员完成维修后，系统更新维修历史记录。

五、报表与统计数据

系统自动生成各类报表和统计数据，如用户活跃度、订单量、车辆使用率等。

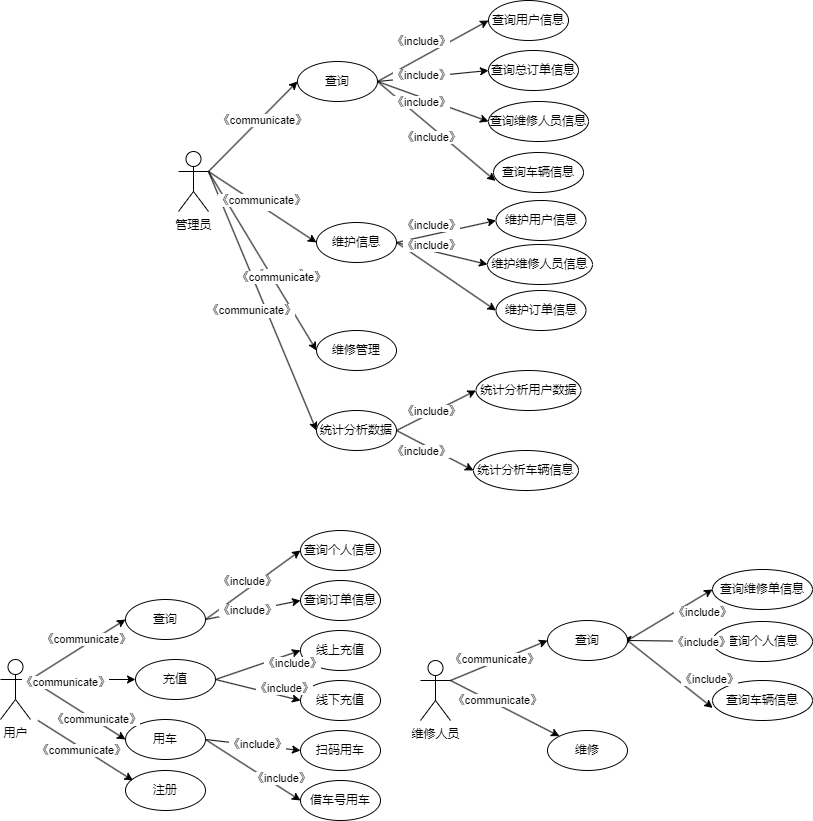
管理员可以通过报表分析运营情况，为运营决策提供数据支持。

根据上文所述业务需求，经过需求分析后可以得到以下用例图：

****

**图3-1 单车管理系统用例图**

进一步进行分解之后可得以下用例图

****

**图3-2 单车管理系统用例图（分解）**

以下是对用例图的用例说明：

**表3-1 “用户注册”用例规约**

|  |  |
| --- | --- |
| 用例名称： | 用户注册 |
| 用例ID： | U1 |
| 参与者： | 用户 |
| 用例说明： | 用户在平台上注册成为会员 |
| 前置条件： | 用户需要提供有效身份证件、手机号码、密码等个人信息 |
| 主事件流： | 1. 用户打开注册页面，输入个人信息  2. 系统对用户提交的信息进行审核  3. 审核通过后，用户成为平台会员 |
| 备选事件流： | 审核不通过，系统提示用户重新提交信息 |
| 异常事件流： | 用户输入信息不完整，系统提示用户完善信息 |
| 后置条件： | 用户注册成功，可以登录系统进行操作 |

**表3-2 “用户登录”用例规约**

|  |  |
| --- | --- |
| 用例名称： | 用户登录 |
| 用例ID： | U2 |
| 参与者： | 用户 |
| 用例说明： | 用户使用手机号码和密码登录系统 |
| 前置条件： | 用户已经注册成为平台会员 |
| 主事件流： | 1. 用户输入手机号码和密码，点击登录按钮  2. 系统验证用户信息，允许用户登录  3. 用户登录系统，可以进行车辆查询、订单查询等 |
| 备选事件流： | 用户输入的手机号码或密码错误，系统提示用户重新输入 |
| 异常事件流： | 用户忘记密码，系统提供找回密码的功能 |
| 后置条件： | 用户成功登录系统，可以进行操作 |

**表3-3 “车辆借还”用例规约**

|  |  |
| --- | --- |
| 用例名称： | 车辆借还 |
| 用例ID： | V1 |
| 参与者： | 用户 |
| 用例说明： | 用户借用和归还车辆 |
| 前置条件： | 用户已经登录系统 |
| 主事件流： | 1. 用户在平台上查询可用车辆  2. 用户选择目标车辆并发起借车请求  3. 系统确认借车请求，更新车辆状态为“借出”  4. 用户在借车期间可以使用车辆  5. 用户通过系统发起还车请求  6. 系统确认还车请求，更新车辆状态为“可借” |
| 备选事件流： | 无 |
| 异常事件流： | 车辆状态异常，系统提示用户联系管理员处理 |
| 后置条件： | 用户成功借还车辆，系统更新车辆状态 |

**表3-4 “订单管理”用例规约**

|  |  |
| --- | --- |
| 用例名称： | 订单管理 |
| 用例ID： | O1 |
| 参与者： | 用户 |
| 用例说明： | 用户查询订单信息 |
| 前置条件： | 用户已经登录系统 |
| 主事件流： | 1. 用户发起订单请求，系统生成订单编号并记录订单信息  2. 系统更新订单状态为“进行中”  3. 用户可以查询订单状态，了解订单进展 |
| 备选事件流： | 无 |
| 异常事件流： | 订单状态异常，系统提示用户联系管理员处理 |
| 后置条件： | 订单状态更新，用户可以查询订单信息 |

**表3-5 “车辆调度与维修”用例规约**

|  |  |
| --- | --- |
| 用例名称： | 车辆调度与维修 |
| 用例ID： | D1 |
| 参与者： | 管理员 |
| 用例说明： | 管理员对车辆进行调度和维修管理 |
| 前置条件： | 管理员已经登录系统 |
| 主事件流： | 1. 用户发起订单请求，系统生成订单编号并记录订单信息  2. 系统更新订单状态为“进行中”  3. 用户可以查询订单状态，了解订单进展 |
| 备选事件流： | 无 |
| 异常事件流： | 车辆故障严重，系统提示管理员联系维修人员处理 |
| 后置条件： | 车辆调度和维修管理完成，系统更新相关记录 |

**表3-6 “报表与统计数据”用例规约**

|  |  |
| --- | --- |
| 用例名称： | 报表与统计数据 |
| 用例ID： | B1 |
| 参与者： | 管理员 |
| 用例说明： | 生成各类报表和统计数据 |
| 前置条件： | 管理员已经登录系统 |
| 主事件流： | 1. 系统自动生成各类报表和统计数据  2. 系统自动生成各类报表和统计数据，如用户活跃度、订单量、车辆使用率等  3. 管理员可以通过报表分析运营情况，为运营决策提供数据支持 |
| 备选事件流： | 无 |
| 异常事件流： | 无 |
| 后置条件： | 系统生成报表和统计数据，管理员可以进行运营分析和决策 |

**表3-7 “管理基本信息”用例规约**

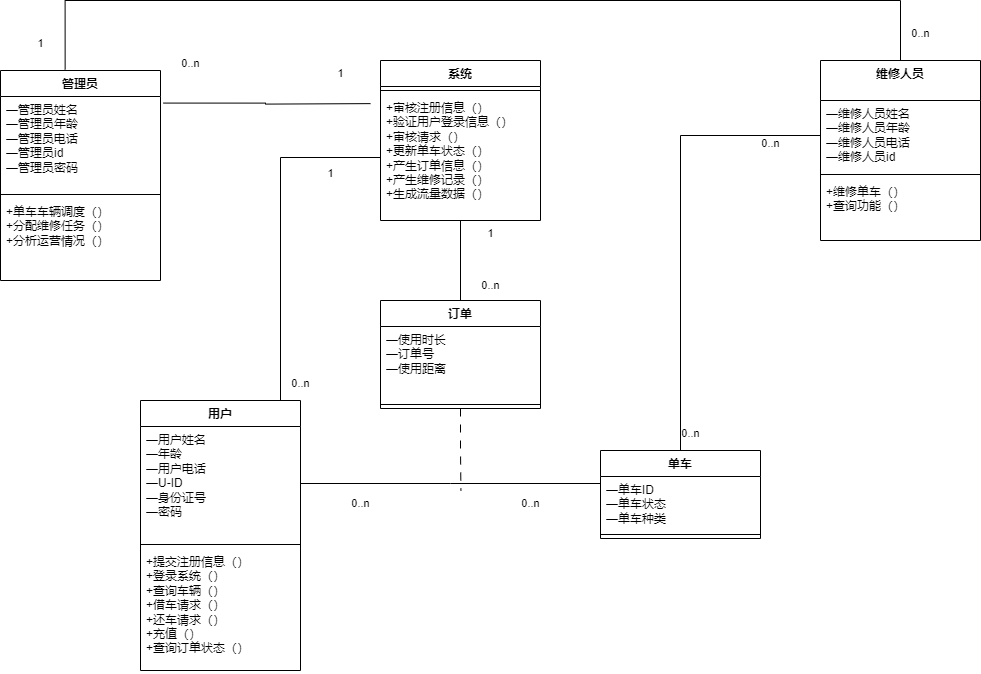
|  |  |
| --- | --- |
| 用例名称： | 管理基本信息 |
| 用例ID： | M1 |
| 参与者： | 管理员 |
| 用例说明： | 对单车的基本信息进行管理 |
| 前置条件： | 无 |
| 主事件流： | 1. 添加单车信息  （1）输入单车的基本信息  （2）选择添加按钮  （3）若该单车信息存在，执行A1  （4）若该单车信息不存在，执行（3）  （5）单车信息添加至数据库  2. 修改单车信息  （1）从下拉列表中选中要修改的单车  （2）输入修改的信息，更新至数据库  3. 删除单车信息  （1）从列表中选中要删除的单车信息  （2）点击删除按钮删除信息，更新至数据库中  4. 查询单车信息  （1）根据编号找到单车信息  （2）得到查询结果 |
| 备选事件流： | 无 |
| 异常事件流： | A1：提示“单车信息存在” |
| 后置条件： | 更新后的单车信息显示在列表中 |

## 3.2 建立对象模型

根据上述用例模型以及业务功能需求可以得出涉及对象有：

单车类，维修人员类，系统类，管理员类，用户类，订单类

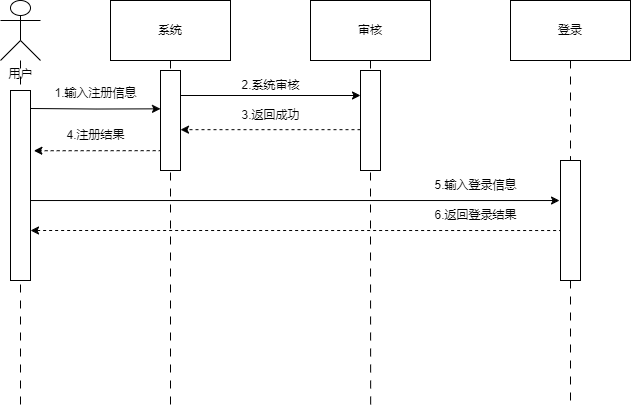
以下是根据这些类及其中关系建立的类图：



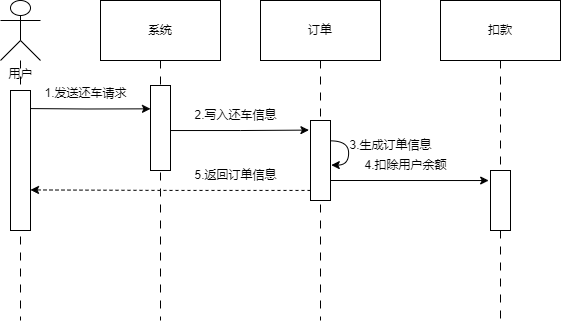
**图3-3 单车管理系统类图**

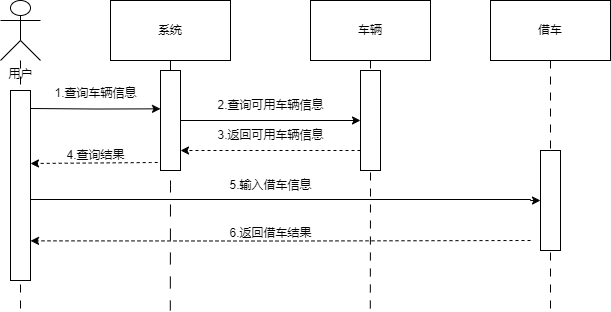
## 3.3 建立交互模型

以上所分析建立的对象模型，以及用例模型，可以进一步得出系统交互逻辑，以顺序图的形式展示如下（仅展示部分用例的顺序图）：



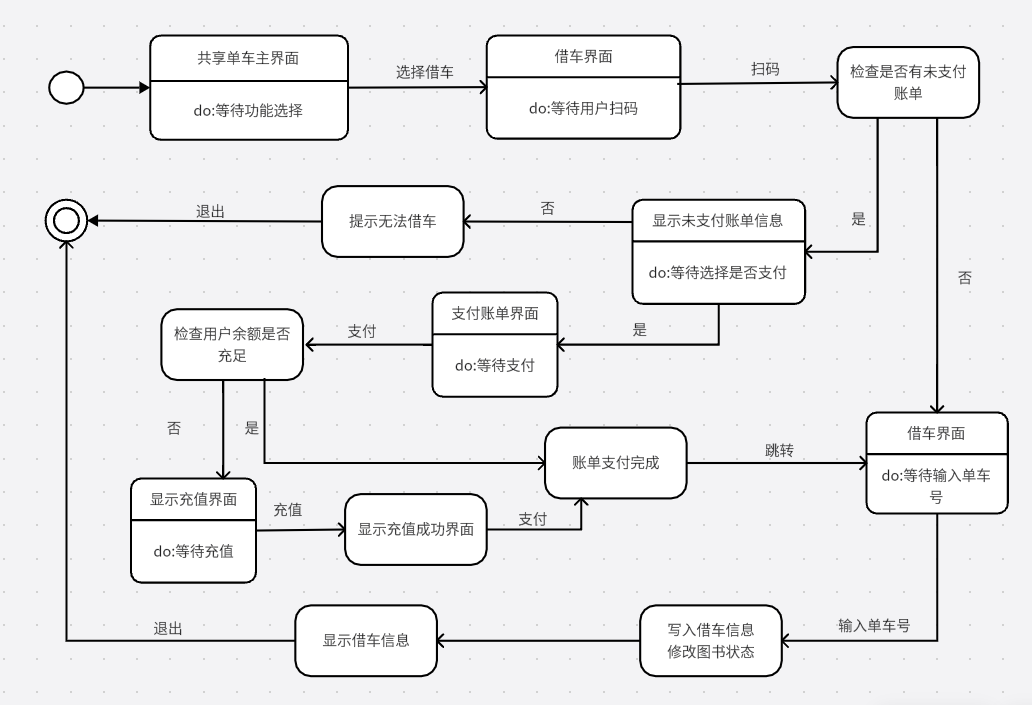
**图3-4 用户注册登入用例的顺序图**



**图3-5 还车用例的顺序图**

**图3-6 借车用例的顺序图**

根据上述对象模型以及用例顺序图可以建立状态图来描述系统的状态以及引起系统状态转换的事件来表示系统的行为。



**图3-7 单车管理系统状态图**

通过对系统流程的分析可以得出以下系统活动图:

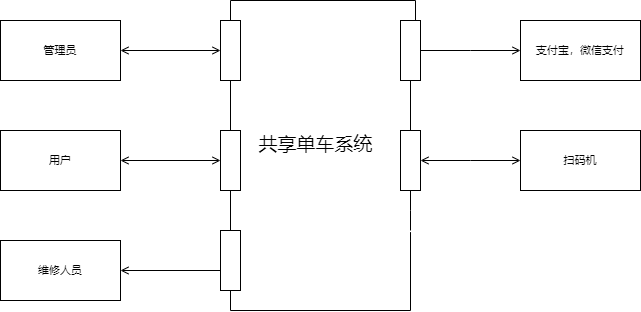
图示, 示意图

描述已自动生成

**图3-8 单车管理系统活动图**

# 4. 面向对象设计

## 4.1环境模型设计



**图4-1 单车管理系统环境模型**

## 4.2系统体系设计

本软件体系架构简单，主要遵循两个设计原则：单一职责原则，开放-封闭原则。在上文提到的四个类的基础上增加了一个数据库类，每个类的职责单一，每个类有且仅有一个使其发生变化的原因，防止因职责的变化而削弱或抑制这个类完成其他职责的能力，同时对修改类是封闭的，对扩展类是开放的。

下面列出了各个业务的模块

1. 用户注册与登录模块

- 用户注册功能

- 用户登录功能

- 用户信息审核功能

2. 车辆借还与充值模块

- 车辆查询功能

- 借车功能

- 还车功能

- 充值功能

3. 订单管理模块

- 订单生成功能

- 订单状态更新功能

- 订单查询功能

4. 车辆调度与维修模块

- 车辆位置查询功能

- 车辆调度功能

- 维修任务分配功能

- 维修历史记录功能

5. 报表与统计数据模块

- 报表生成功能

- 统计数据生成功能

- 数据分析功能

6. 数据库模块

- 用户信息存储功能

- 车辆信息存储功能

- 订单信息存储功能

- 维修记录存储功能

每个模块都有明确的功能和职责，符合单一职责原则。同时，如果需要对功能进行扩展或修改，可以通过扩展已有的模块来实现，符合开放-封闭原则。

## 4.3数据管理子系统设计

数据库设计就是根据业务系统的具体需求，结合我们所选用的DBMS（数据库管理系统），为这个业务系统构造出最优的数据存储模型。并建立数据库中的表结构以及表与表之间的关联关系的过程。使之能有效的对应用系统中的数据进行存储，并可以高效的对已存储的数据进行访问。本系统采用MySQL 8.03作为数据库实现，数据库名：单车信息管理系统；标志：hrdatabase。

对上一节进行分析，得到用户，单车，管理员，维修人员，订单，报表六个实体类。将各个实体类分别映射成数据库中的关系模式。

根据系统需求，可以设计以下数据库表：

**实体：**

用户，单车，管理员，维修人员，订单，报表。

**关系：**

1. 管理员与车辆：管理员能够添加、修改、查询、删除车辆信息，所以存在“管理”关系。

2. 管理员与用户：管理员能够添加、修改、查询、删除用户信息，所以存在“管理”关系。

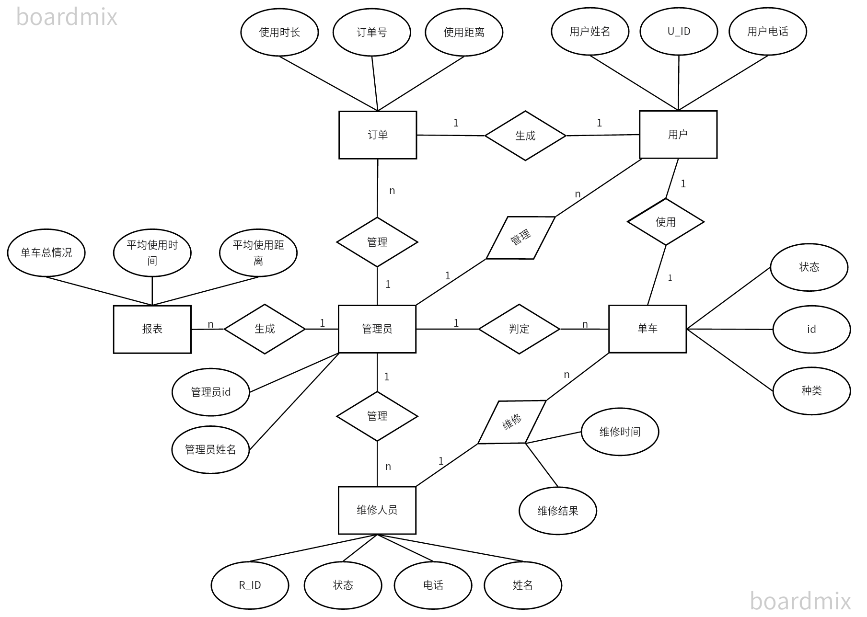
3. 管理员与订单：管理员能够查询订单、取消订单、生成订单，所以存在“管理”关系。

4. 管理员与车辆：管理员能够对车辆进行调度和维修，系统能够记录车辆的调度和维修历史，所以存在“调度”和“维修”关系。

5. 管理员与报表/统计数据：管理员能够生成报表和统计数据，所以存在“生成”关系。

6. 用户与订单：用户能够完成系统注册、共享单车的借车、充值事务，所以存在“完成”关系。

7. 维修人员与车辆：维修人员维护单车事务，所以存在“维护”关系。





**图4-2单车管理系统ER图**

# 5. 面向对象实现和测试

## 5.1 面向对象实现

由于系统暂未实现，故而先给出部分类的设计：

**User**: 属性包括身份证件、手机号码、密码等，方法包括注册、登录、查询车辆信息、发起订单、充值等操作。

**Vehicle**: 属性包括车辆状态，方法包括确认借车请求、确认还车请求。

**Order**: 属性包括订单编号、订单状态，方法包括生成订单、更新订单状态、查询订单状态。

**Administrator**: 属性包括管理员权限，方法包括查看车辆位置、分配维修任务。

**MaintenancePersonnel**: 属性包括维修任务，方法包括接收维修任务、进行维修、更新维修历史记录。

**Report**: 方法包括生成各类报表和统计数据，如用户活跃度、订单量、车辆使用率等。

**System**: 包括系统自动生成报表和统计数据的方法。

## 5.2 面向对象测试

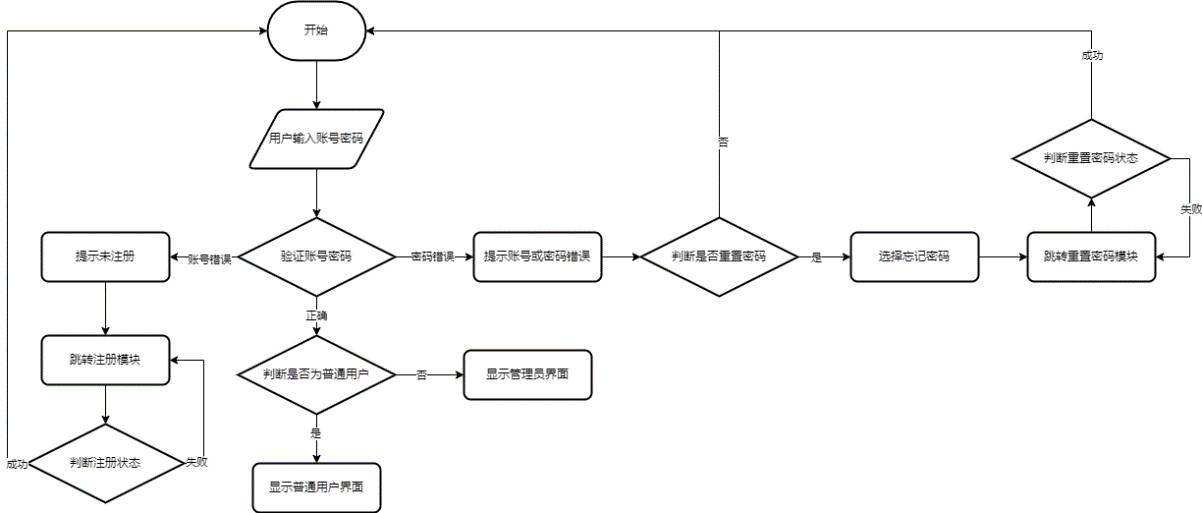
### 1. 测试范围

本次对该软件的测试范围包括借车模块，用户登录注册模块。

### 2. 测试的种类

考虑到本系统较为简单，且目前暂未实现，故而本文采用的测试种类有单元测试，白盒测试。

### 3. 测试流程图



**5.1用户登录与注册模块测试流程图**

图示

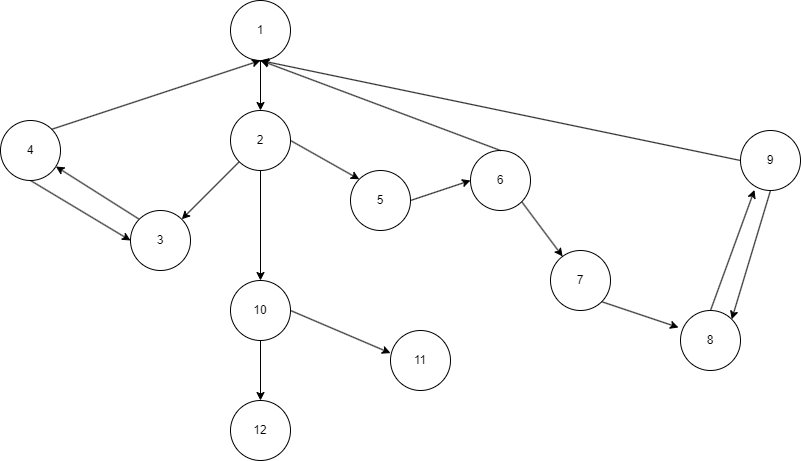
描述已自动生成

**5.2借车模块测试流程图**

图示

描述已自动生成

**5.3借车模块测试流程样例图**

****

**5.4用户注册登录模块测试流程样例图**

### 4. 软件测试过程

白盒测试，单元测试（基本路径测试，用户注册登录模块）

路径一:1-2-10-12

路径二:1-2-3-4-2...

路径三:1-2-5-6-2...

路径四:1-2-5-6-7-8-9-2...

这一小结采用动态分析的白盒测试使用路径测试对用户注册登录模块进行测试。

根据系统实现，当数据1被正确录入后将会显示在用户信息栏中，此时可以进行用户登录，登入成功将会显示登录界面。

### 5. 软件测试总结

测试总结的是问题，是风险，是经验，是如何改进，是分析测试过程中的问题，并针对项目进行分析从而找出解决方案。也可以说用于分析并总结这些缺陷，将总结出来的经验用于指导下一次的测试用例设计。在每个版本测试完毕后，要进行测试总结，做一些比例的总结、缺陷严重级别及比例的总结，单车工作效率的总结，还有最重要的是风险的平复，对下一测试版本的建议等。 本系统在软件编写代码时已经经过了多次实验，导致在软件测试时几乎找不出什么错误。在本章的软件测试中已经对几个主要的模块进行了测试，没有找出什么错误，表明在现阶段中的软件应用没有太大的问题需要去修改。