# 一、背景分析

大学校园公用的自行车租赁服务是十分常见的。以往，这样的服务提供方式是通过人工登记，办理租赁手续等方式来完成的。这种传统的方式面临着手续繁琐而且效率低下的问题。随着互联网的兴起以及web技术的不断发展，我们的工作与日常生活受到了很大的改变，这极大的地方便了我们的工作与生活。在这样的背景下，我们可以借助于互联网的手段来改善我们传统的自行车租赁模式，从而减少成本，提高我们的工作效率。

在这次的课题中，我们是通过Java来完成课题的设计与开发，采用Navicat来作为数据的存储支持。通过采用Java的面向对象的编程方式，来对本校自行车管理系统进行设计与实现。该网站的主要功能是要代替传统自行车租赁方式，实现租赁自行车，归还自行车、自行车维修等功能，极大地提高了自行车租赁业务的效率，从而降低了人力成本和物力成本。

这次课题分析并且设计了自行车租赁后台管理系统的静态模型和动态模型，完成了自行车租用系统开发的分析、设计和实现的工作。本自行车租赁网站通过Web方式完成用户与系统的交互，系统的功能模块具体有自行车信息管理模块、租车信息管理模块、维修信息管理模块等。

# 二、系统规划

## 2.1系统目标分析

系统必须能减少工作人员的工作量，方便大学生借还自行车，提高效率，降低成本，由此提出以下基本要求。

1.系统设计原则：

（1）实用性：这是开发该系统的首要原则，系统的实用性是衡量信息系统的建设与否的基本标准之一，是系统为学校创造效益的重要保证。

（2）可维护性：系统要有较强的后台管理能力，

（3）易移植性和扩展性：新系统要能方便地移植，并且实现与外部系统的集成。

（4）针对性：该系统主要是为满足学校的具体需求和解决实际问题而开发的，经过调查了解而设计的。

2.功能目标：

针对现有问题及实际要求，提出如下系统的功能目标：

3.性能目标：

（1）准确可靠，要求各种数据准确无误，信息反映真实有效。

（2）安全：对于学生的个人信息，只有相关人员可以获取，其他人员无法得到数据。

（3）响应速度快、对用户友好：正常情况下，应保证系统的运行速度较快，方便用户的输入、操作简单。

## 2.2系统可行性分析

1. 技术可行性
   1. 开发工具：Eclipse
   2. 数据库坏境：Navicat
   3. 系统环境：Windows

系统采用SSH框架和Jquery Easyui进行开发，成熟的技术对系统提出的目标的支持程度高，技术优秀的开发平台加上优秀的开发人员和完善的开发坏境，为系统的顺利开发提供了有力的技术保证。

1. 经济可行性

成本：（1）主要就是在系统开发上所投入的一些固定成本，例如系统开发前期的准备、搜集资料、问卷调查等方面；（2）新系统开发成功后所产生的维护费用以及一些必要的技术支持工作。

效益：（1）首先主要是开发的系统发布后，工作人员在使用网站后的工作效率大大提高，工作时间也节省了，从而间接为大学生及学校创造了经济效益，减少了管理费用的支出，避免了不必要的人工失误；（2）再者无纸化的信息管理相当于为学校减少了一部分的费用支出例如：订单记录本、信息登记册等纸质文档的费用；（3）最后工作内容全都信息化，把所有操作记录都保存，有利于管理方式与体制的合理改革。

1. 运行可行性

（1）这套系统能帮助学校工作人员从繁杂的工作中解脱出来，能大大降低学校的人力、物力及财力的耗用；（2）系统移植性能良好，模块与模块之间相互独立，数据库具有保密性；（3）在开发时代码编写规范，另外系统对用户的权限进行了分层控制，不同层级的人员能在自己的权限范围内使用，

1. 法律可行性

本软件是专用软件，具有较强的针对性。在我们的编码和设计中，除了参阅相关的资料，不涉及已存在的设计专利、不侵犯国家、学校的相关制度，因此具法律可行性。

# 三、系统分析

## 3.1范围定义

范围定义阶段回答这样一个问题：“这个项目看起来是否值得”，为了回答这个问题我们定义了项目的范围以及触发该项目的可见的问题、机会和指令。

1. 列出问题和机会

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 问题、机会或指示的简要描述 | 紧急程度 | 可见性 | 优先权 | 建议的方案 |
| 1、校园公共自行车没有可用的管理系统进行统一的管理 | ASAP | 高 | 1 | 新开发 |
| 2、校园公共自行车编号不够规范，管理困难 | 1个月 | 中等 | 2 | 在新系统开发后，进行统一编号 |
| 3、自行车站点分布不够合理，不能有效的解决学生就近租借自行车的需求 | 1个月 | 中等 | 2 | 在新系统开发后，进行统一的站点安排 |
| 4、自行车的维护工作不规范，不能及时的得到维修处理和获取自行车当前的状况 | 1个月 | 中等 | 2 | 通过新系统进行及时更新处理 |
| 5、手工安排站点值班人员过于麻烦 | 1个月 | 中等 | 2 | 新开发 |
| 6、自行车借出和归还的数据不对称 | 1个月 | 中等 | 1 | 新开发。额外的收益可能会增加其紧急程度 |
| 7、工作人员的工作有穿插，职责不够明确 | 1个月 | 中等 | 2 | 新开发。在新系统中加入权限设定 |
| 8、信息安全性问题 | 1个月 | 高 | 1 | 新系统 |

表3-1 问题陈述

2、项目的初步范围

（1）系统数据：车辆信息，用户信息，借还车记录，维修信息，站点信息等

（2）业务过程：车辆管理，人员管理，借、还车，保修申请等

（3）接口对象：用户信息系统等

3.2问题分析

在这个阶段要解决两个问题：“提出的问题真的值得解决问题吗？”，“值得构造一个新系统吗？”，为此，我们必须充分研究和理解问题领域并全面分析其中存在的问题、机会和约束条件。

1. 问题领域研究

采用上下文图分析系统如何与它周围的世界交互，并说明系统的输入和输出。

1. 问题和机会分析

问题、机会、目标和约束矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 因果分析 | | | 系统改进目标 | |
| 问题或机会 | 原因和结果 | | 系统目标 | 系统的约束条件 |
| 1.全手工操作借还车，人力、时间浪费 | 1  2  3  . | 统计借车信息困难，浪费时间  人工统计大量数据易出错，难以核对数据，查询数据  无法及时与维修人员交接上，设备维修处理延迟 | 1.处理一个借还车记录的时间减少30%  2.查询数据时间减少70%  3合理分配个站点的车辆时间减少80% | 1.不会增加值班人手  2新系统必须在12月7日前运行  3新系统成本不能超过6000元 |

表3-2因果分析

3.3需求分析

为充分理解“用户需要什么？想从一个新系统中得到什么？”。我们采用用例的建模工具表述业务需求，对该系统必须处理的业务场景和时间进行建模。

用例词汇表

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用例描述 | 参与者 |
| 登录 | 该用例描述后台管理人员进行后台操作必须的步骤，登录过后，才能进行相应的操作 | 值班人员、维修人员、车辆调拨人员、超级管理员 |
| 查看、修改我的信息 | 该用例描述后台管理人员在后台可以查看个人信息并且修改个人信息 | 值班人员、维修人员、车辆调拨人员、超级管理员 |
| 修改密码 | 该用例描述后台管理人员在后台可以修改密码 | 值班人员、维修人员、车辆调拨人员、超级管理员 |
| 查看我的借车记录 | 该用例描述值班人员可以查看自己借出的信息 | 值班人员 |
| 查看我的还车记录 | 该用例描述值班人员可以查看自己归还的信息 | 值班人员 |
| 提交借车信息 | 该用例描述值班人员提交一条借车信息（该车必须是已经归还的） | 值班人员（主要操作） |
| 提交还车信息 | 该用例描述值班人员提交一条还车信息（该车必须是借出的） | 值班人员（主要操作） |
| 提交维修信息 | 该用例描述值班人员提交一条维修信息 | 值班人员 |
| 接收维修信息 | 该用例描述维修人员接收一条维修信息 | 维修人员 |
| 提交维修处理信息 | 该用例描述维修人员对车辆维修之后提交的一条信息 | 维修人员 |
| 查询车辆信息 | 该用例描述车辆调拨人员查询每个站点的车辆 | 车辆调拨人员 |
| 调拨车辆 | 该用例描述车辆调拨人员从某个站点调拨车辆到另一个站点 | 车辆调拨人员 |
| 录入值班人员信息 | 该用例描述超级管理员增加值班人员的信息，并且对值班人员的权限进行设定 | 超级管理员 |
| 录入车辆信息 | 该用例描述超级管理员对增加的车辆进行编号并且录入车辆的相关信息 | 超级管理员 |
| 安排值班 | 该用例描述超级管理员对值班人员进行值班安排 | 超级管理员 |
| 查询值班信息 | 该用例描述超级管理员对值班的信息进行查询 | 超级管理员 |

用例图：

用例描述：

|  |  |  |  |
| --- | --- | --- | --- |
| 用例名称 | 提交借车信息 | | 用例类型：  业务需求： |
| 用例ID |  | |
| 优先权 | 高 | |
| 来源 | 需求 | |
| 主要业务参与者 | 值班人员 | | |
| 其他参与者 | 站点、租借人 | | |
| 描述 | 该用例描述值班人员提交一条借车信息。一旦信息提交，则该车辆不能被再次借出。 | | |
| 前置条件 | 该车必须是已经归还的 | | |
| 触发器 |  | | |
| 典型事件过程 | 参与者动作 | 系统响应 | |
| 第1步：值班人员提供他的资料信息以及车辆租借信息和租借同学的信息。 | 第2步：系统验证所需的所有信息都提供了之后做出响应。  第3步：系统根据以前记录的资料验证值班人员的资料信息。  第4步：对于租借的每辆自行车，系统验证自行车的编号。  第5步：对于租借的每辆自行车，系统验证自行车的可用性，即是否已经被借出。  第6步：对于每辆可用的自行车，系统决定将该自行车租借给该同学。  第7步：一旦处理了租借的自行车，该自行车则属于借出状态，无法再进行租借。  第8步：系统检查值班人员账号的状态。  第9步：系统记录租借信息，然后将租借信息上传到总数据库。 | |
| 替代事件过程 | 替代第2步：值班人员没有提供租借所需的所有信息，通知值班人员并提示重新提交。  替代第3步：如果提供的值班人员信息与原先记录不同，则验证那个记录是最新的，然后相应的修改值班人员信息。  替代第4步：如果值班人员提供的自行车信息与站点的自行车信息都不匹配，则通知值班人员并查明情况。  替代第5步：如果租借的自行车不可用，则提示值班人员。  替代第8步：如果值班人员的账号状态不良，记录租借信息，把它设为挂起状态，通知超级管理员其账号状态，以及租借信息被挂起的原因。终止用例。 | | |
| 结论 | 当值班人员确认租借信息时，该用例结束。 | | |
| 后置条件 | 租借信息被记录下来，租借出自行车为不可用状态。 | | |
| 业务规则 | 租借车辆时和该同学的租借记录相对应，如果出现不良租借记录，则不提供租借。 | | |
| 实现约束和说明 | 为值班人员提供后台操作页面。 | | |
| 假设 | 在日常报告中提示未及时归还的车辆。 | | |
| 开放问题 | 需要确定值班人员。 | | |
| 用例名称 | 提交还车信息 | | 用例类型：  业务需求： |
| 用例ID |  | |
| 优先权 | 高 | |
| 来源 | 需求 | |
| 主要业务参与者 | 值班人员 | | |
| 其他参与者 | 站点、还车人 | | |
| 描述 | 该用例描述值班人员提交一条还车信息。一旦信息提交，则该车辆可以被再次借出。 | | |
| 前置条件 | 该车必须是已经借出的 | | |
| 触发器 |  | | |
| 典型事件过程 | 参与者动作 | 系统响应 | |
| 第1步：值班人员提供他的资料信息以及车辆归还信息和归还同学的信息。 | 第2步：系统验证所需的所有信息都提供了之后做出响应。  第3步：系统根据以前记录的资料验证值班人员的资料信息。  第4步：对于归还的每辆自行车，系统验证自行车的编号。  第5步：对于归还的每辆自行车，系统验证自行车的可用性，即是否是被借出的。  第6步：对于每辆借出的自行车，系统决定将该自行车和该同学是否对应。  第7步：一旦处理了归还的自行车，该自行车则属于可借状态。  第8步：系统检查值班人员账号的状态。  第9步：系统记录归还信息，然后将归还信息上传到总数据库。 | |
| 替代事件过程 | 替代第2步：值班人员没有提供归还所需的所有信息，通知值班人员并提示重新提交。  替代第3步：如果提供的值班人员信息与原先记录不同，则验证那个记录是最新的，然后相应的修改值班人员信息。  替代第4步：如果值班人员提供的自行车信息与站点的自行车信息都不匹配，则通知值班人员并查明情况。  替代第5步：如果归还的自行车不是原自行车，则提示值班人员。  替代第8步：如果值班人员的账号状态不良，记录归还信息，把它设为挂起状态，通知超级管理员其账号状态，以及归还信息被挂起的原因。终止用例。 | | |
| 结论 | 当值班人员确认归还信息时，该用例结束。 | | |
| 后置条件 | 归还信息被记录下来，归还的自行车为可用状态。 | | |
| 业务规则 | 归还车辆时和该同学的归还记录相对应，如果出现不良归还记录，则不提供租借。 | | |
| 实现约束和说明 | 为值班人员提供后台操作页面。 | | |
| 假设 | 在日常报告中提示未及时归还的车辆。 | | |
| 开放问题 | 需要确定值班人员。 | | |

3.4逻辑设

3.4.1数据建模

数据建模有助于在建模过程中确定更全面的业务词汇，并且构造的速度快，其具体具体步骤包括：

1、获取实体

2、上下文数据模型

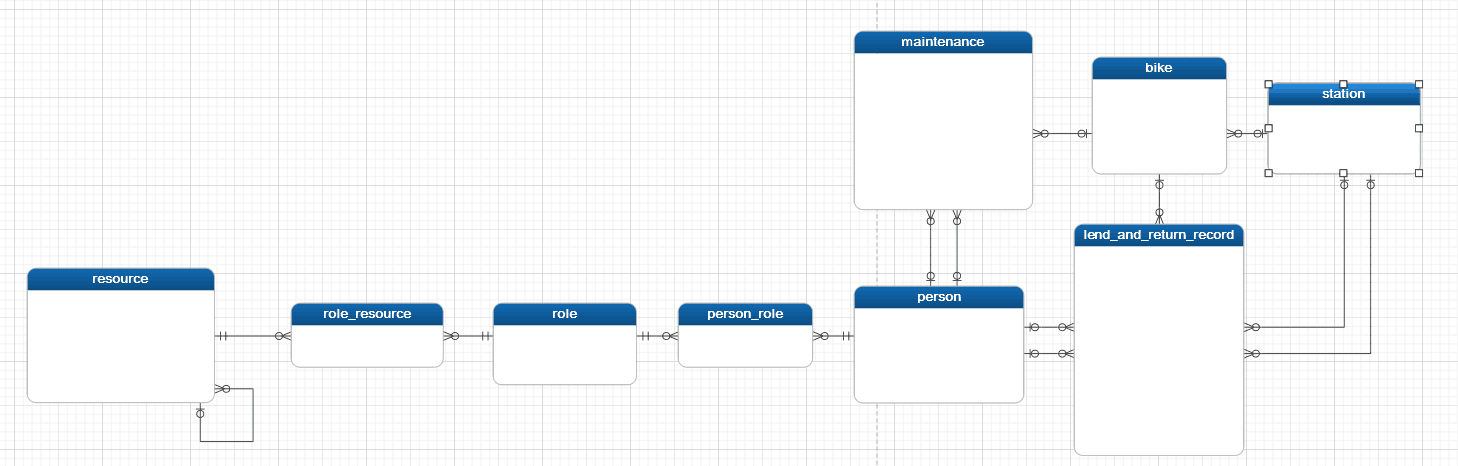
3、基于键的数据模型

4、具有完整属性的数据模型

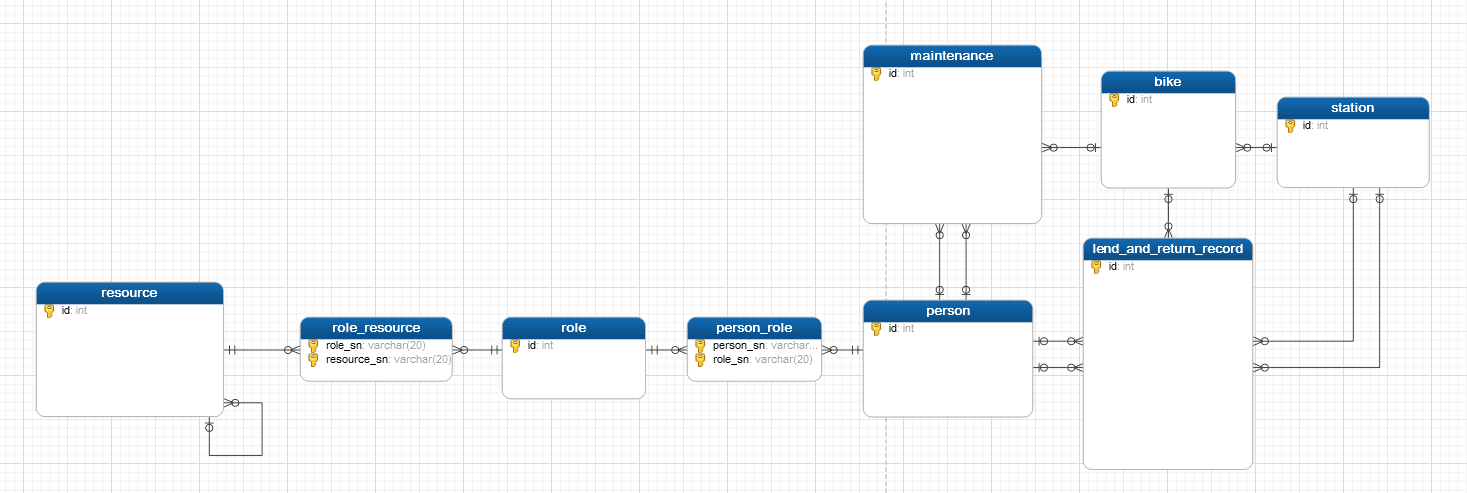
5、分析数据模型使之规范化

自行车管理系统项目的基本实体：

上下文数据模型



基于键的数据模型



具有完整属性的数据模型

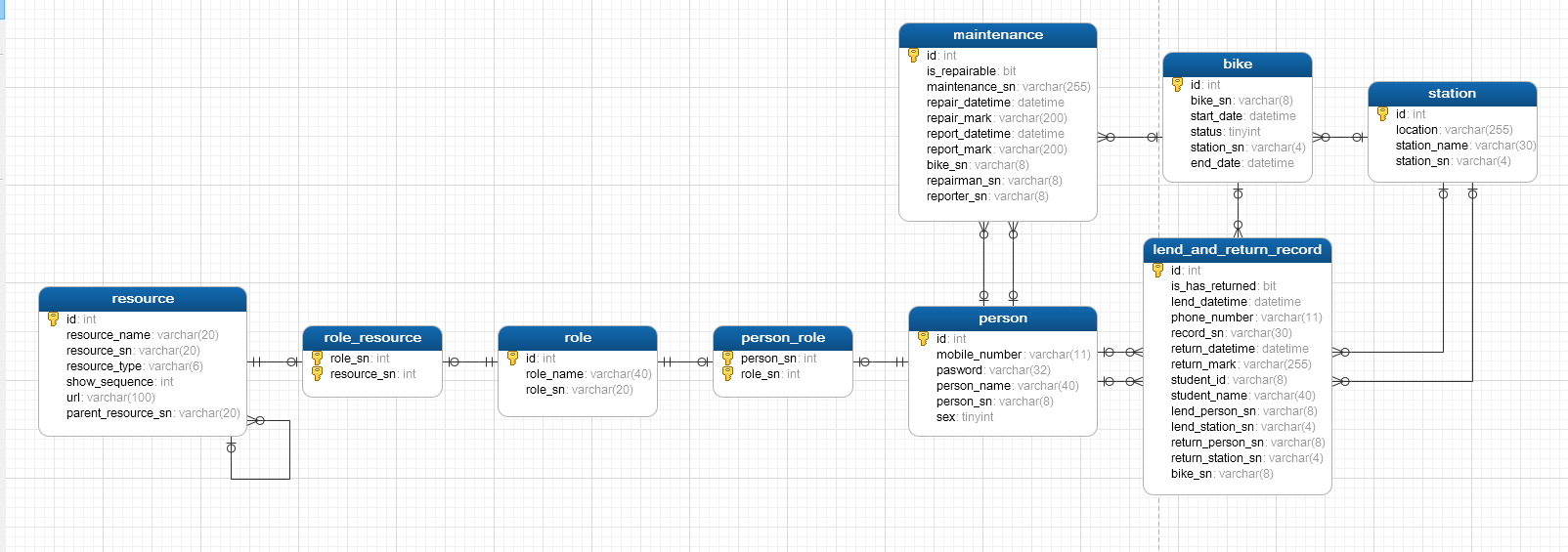


图3-具有完整属性的数据模型

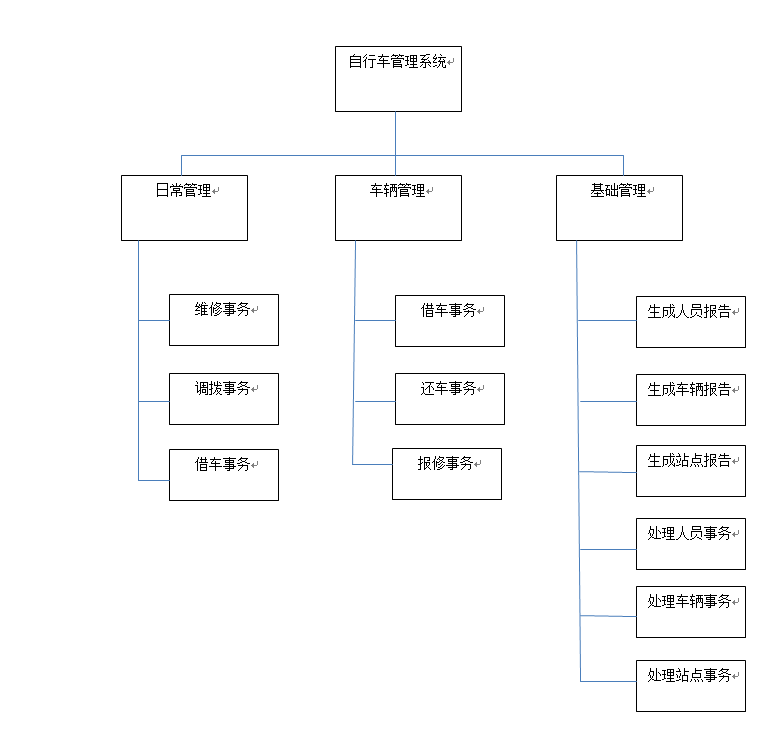
3.4.2过程建模

为组织和记录数据的结构和流向，记录系统的“过程”和/或由系统的“过程”实现的逻辑、策略和程序，我们主要采用系统分析的过程模型，即数据流图。建模步骤如下：

1. 构造系统上下文数据流图，记录项目的初始范围，即环境模型。
2. 绘制功能分解图以显示系统自顶向下的功能分解图。
3. 确定系统事件响应或用例清单并证实系统必须提供响应的业务事件。
4. 增加事件处理过程（每个用例一个）到分解图中以进一步划分功能。
5. 为每一个事件过程绘制一个事件图。
6. 通过合并事件图构造一个或者多个系统图。
7. 对需要进一步处理细节的事件过程构造基本图。

上下文数据流图是一个包括了一个且仅有一个过程，其是对系统的过程进行 一个大致的概括，因此有时也可以称之为环境模型。根据系统的整体操作绘制一 个关于该项目的大致的上下文数据流程图如下：

整个系统有繁琐复杂的各种操作，根据功能分解图可以知道系统大概会进行的一些流程：



一个功能分解流图

一个系统的用例通常很多，这对于系统设计人员构造一个完整的响应所有业务事件的系统来说是必要的，列出部分用例如下：

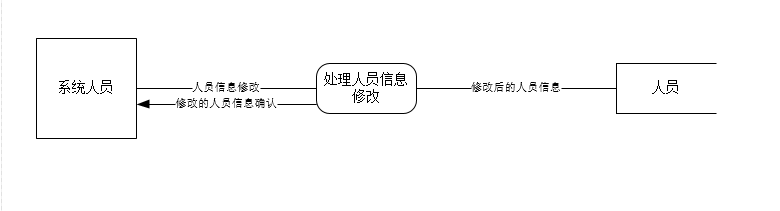
|  |  |  |  |
| --- | --- | --- | --- |
| 参考者/外部代理 | 事件（或用例） | 触发器 | 响应 |
| 用户 | 修改密码 | 密码修改 | 生成“用户修改确认”  修改数据库中的“用户” |
| 值班人员 | 录入借车信息 | 新借车信息 | 生成“信息提交确认”在数据库中创建“借车记录” |
| 值班人员 | 发出报修信息 | 新报修记录 | 生成“报修信息确认”在乎数据库中创建“维修” |
| 值班人员 | 录入还车信息 | 新还车信息 | 生成“信息提交确认”在数据库中创建“还车记录” |
| 维修人员 | 录入维修信息 | 新维修信息 | 生成“信息提交确认”在数据库中创建“维修记录” |
| 系统管理员 | 查询车辆借还的历史记录 | 车辆借还查询 | 生成“车辆借还历史” |
| 系统管理员 | 修改站点信息 | 站点信息修改 | 生成“站点目录修改确认”修改数据库中的“站点” |

部分用例表

为进一步在分解图中划分功能，结合功能分解图和用例表可以得出系统的各类事件的事件图。这些事件图就是对系统事件的输入、输出和数据存储交互的一个更加详尽的描述。

1. 处理人员信息修改事件

在这个事件中涉及到数据库信息的提取，用户需要提交修改界面所要求填写的内容，将内容填写完整后才能提交，弹出提示确认修改，用户需要再次确认后才提交至数据库。



2、

四、系统设计

4.1物理设计

4.2系统界面设计

4.3系统安全性设计

五、结束语