[1. 产品描述 2](#_Toc2141523007)

[1.1. 背景： 2](#_Toc1759558957)

[1.2. 产品名称 2](#_Toc1000063702)

[1.3. 定义 2](#_Toc1088649229)

[2. 产品需求概述 2](#_Toc186331977)

[2.1. 功能概述 2](#_Toc828437368)

[2.2. 运行环境 4](#_Toc734024177)

[2.3. 约束条件 4](#_Toc1913629433)

[3. 功能需求 4](#_Toc624630705)

[3.1. 附近车辆查看 4](#_Toc211913804)

[3.2. 预约车辆 4](#_Toc972861172)

[3.3. 借车 4](#_Toc1829964548)

[3.4. 还车确认 5](#_Toc1793021054)

[3.5. 车辆评估与报修 5](#_Toc934704621)

[3.6. 车辆信息查询 5](#_Toc1722020769)

[3.7. 车辆维护记录查询 5](#_Toc192828583)

[3.8. 意见反馈 5](#_Toc120665618)

[3.9. 登录系统 6](#_Toc487587945)

[3.10. 自身数据维护 6](#_Toc609042026)

[3.11. 车辆查找 6](#_Toc1856244989)

[3.12. 取出车辆 6](#_Toc1524408961)

[3.13. 故障车确认 6](#_Toc2033610421)

[3.14. 查看维修任务 7](#_Toc912367973)

[3.15. 设置车辆状态 7](#_Toc1244947780)

[3.16. 结束维修任务 7](#_Toc1381234216)

[3.17. 录入巡检任务 7](#_Toc1085526756)

[3.18. 查看用户信息 7](#_Toc1608448178)

[3.19. 查看用户借还车记录 8](#_Toc1530237157)

[3.20. 用户信息维护 8](#_Toc1745177026)

[3.21. 异常数据处理 8](#_Toc757848791)

[3.22. 系统维护 8](#_Toc1299711927)

[4. 流程分析 8](#_Toc1739216385)

[4.1. 流程1：借-还车-评价流程 9](#_Toc369924100)

[4.2. 流程2：巡检-维修流程 9](#_Toc152291981)

[4.3. 流程3：用户报修-巡检确认-维修流程 10](#_Toc680381966)

[4.4. 流程4：任务创建-分配流程 10](#_Toc556256078)

[5. 数据分析 10](#_Toc980729350)

[6. 限制条件 11](#_Toc1414406144)

[6.1. 开发周期 11](#_Toc322401863)

[6.2. 数据 11](#_Toc1605360055)

# 产品描述

## 背景：

城市公共自行车系统在运营和使用过程中仍然存在着一些痛点问题，这些问题一方面导致了管理维护成本高，效率低；另一方面用户使用过程中用户体验得不到持续提升。因此，借助移动互联网技术和平台，实现公共自行车系统运营-维护-使用的智能化、网络化和数据化，提升运营企业的管理水平，提高效率，降低成本；改善用户的使用体验，培养用户的忠诚度和粘性等具有非常重要的意义。

## 产品名称

产品暂定名称为：城市公共自行车助手V1.0，后文称“本软件”。

## 定义

(后面补充)

# 产品需求概述

## 功能概述

本软件主要面向2大类用户，即公共自行车使用者和公共自行车系统的管理维护者。前者我们定义为本软件的普通用户，后者定义为本软件的企业用户。因此，在需求上两者有所不同。普通用户需求偏向公共自行车的使用，重点在于借-还车流程的体验；而企业用户则侧重于运营，重点在于提高工作的便捷性和效率。除了上述两个直接用户外，由于是网络多用户系统，本软件还设置了管理员用户用来维护本软件系统的正常运行。

下面采用场景描述的方法来说明用户的需求。

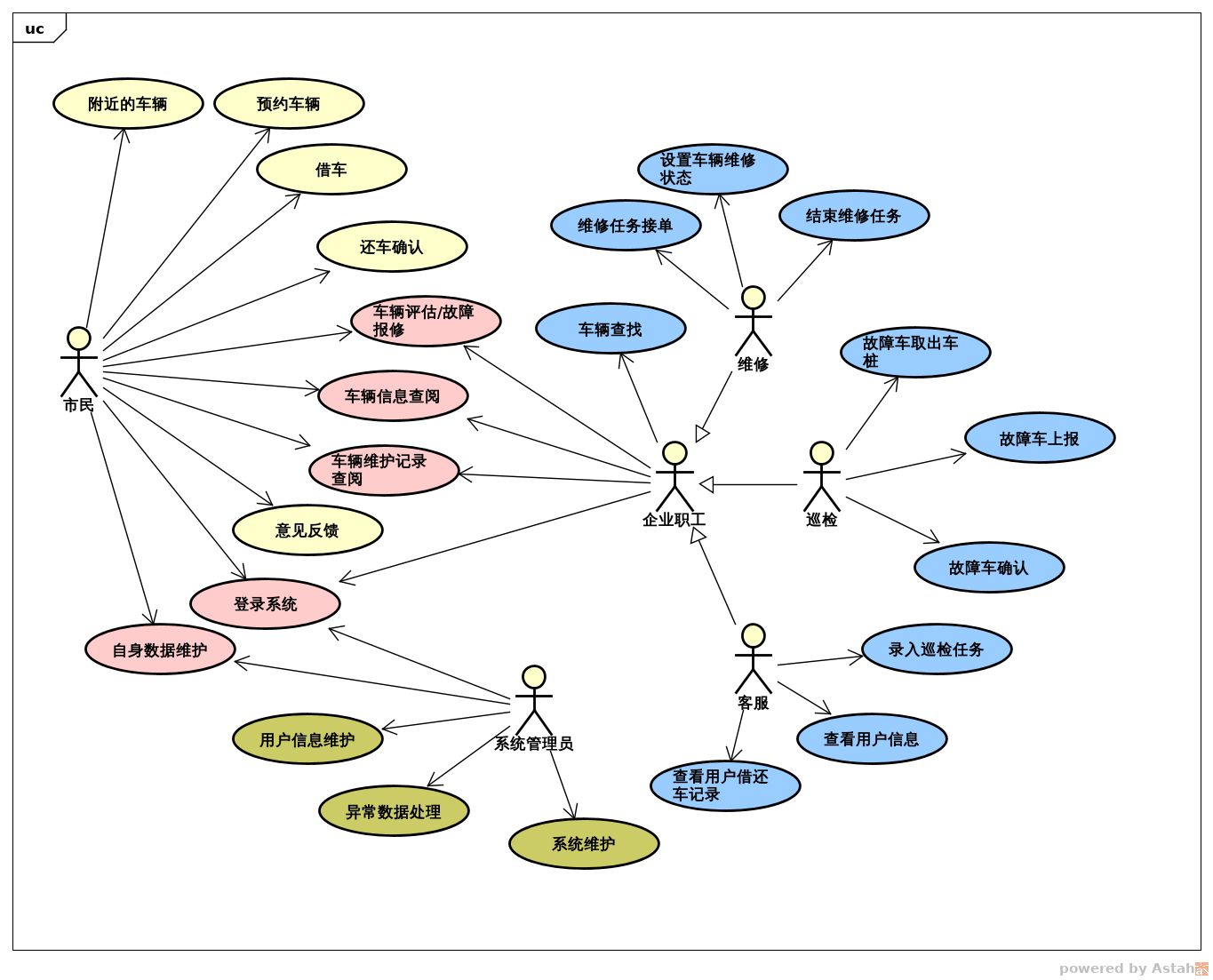
假设李明是本软件的普通用户，当他需要出行的时候，会打开手机端的APP。App启动后他就能够看到自己附近的公共自行车站点及车辆的相关信息，这时他会根据自己的实际情况选择去哪个站点借车，但他发现周围自行车数量已经不多，为了保证自己的出行不被耽误，他通过App选择了一辆状态良好的自行车进行了预约。预约成功后，他立即出门去对应的站点取车出行。到达站点后，找到自己预约的自行车后，通过刷卡或者二维码扫描的方法借出。在顺利的借出自行车后，他又拿出App查找了一下目的地附件的公共自行车站点，选了一个距离目的地比较近而且比较空闲的站点，确认了以后开始骑行。半个小时以后，到达了目的站点，将车子锁入车桩后，手机App立刻收到了还车成功的提示，从而完成了一次绿色出行。此外，李先生还可以利用app客户端对此次骑行给予评价，对车子的状况进行评判等操作。

张师傅是运维人员，他的工作是对所负责的公共自行车站点进行巡检，检查站点和车辆的情况。当他发现受损的自行车后，会利用手机上App扫描受损的自行车，在简单的选择好自行车的故障情况后将信息提交给系统。陈师傅是维修人员，他同样可以利用手机App查看自己负责区域内的公共自行车受损情况，确定损坏的公共自行车的位置，并对其进行检修。对于简单的故障可以进行在站的维修，对于复杂的故障则必须运回专门的维修站进行维修。

小王是公共自行车公司的热线服务人员，收到了热心市民的电话。在电话里市民反映了受损的公共自行车情况，以及找到的自行车车身编号。小王通过浏览器接入到后台中，将该信息记录下来，并根据所处的位置区域指定给负责的巡检人员进行核实处理。

通过上述例子，可以看到，不同用户通过移动设备或者浏览器连接到同一个信息系统平台上，通过各自的角色不同，发布或者处理不同的数据，实现了公共自行车的使用和维护等功能。

整个系统的需求可以用一张典型的用例图来进行说明，在该图中企业用户进行了细化。



## 运行环境

系统需要运行在移动互联网平台上，使用的服务器必须能够被公众访问，用户必须使用智能手机安装对应的App软件。

## 约束条件

无

# 功能需求

## 附近车辆查看

* 用户：普通用户
* 功能描述：查看附件的车辆，根据用户手机GPS数据或者用户在地图上指定的位置来进行区域查找（附近1公里内）的公共自行车站点，显示各个站点并显示每个站点，状态及各自的自行车数量。点击站点可以查看该站点的每一辆公共自行车数据。
* 交互要求：电子地图

## 预约车辆

* 用户：普通用户
* 功能描述: 通过车辆查看功能选中一辆自行车，在界面中通过点击预约功能进行预约。预约后车桩将自行车锁定，其他人无法借出，预约时间为10分钟，时间截止未被借出的自行车会被自动解除预约并扣除费用。已经预约自行车的用户无法再借出其他自行车；预约一经确认不可取消。
* 交互要求:提醒用户预约无法撤销。

## 借车

* 用户：普通用户
* 功能描述: 通过刷自行车上的二维码进行借车操作。用户在界面上点击“借车”按钮后，打开摄像头，用户此时需扫描车身上的二维码，确认后，App与服务器交互进行验证并打开锁止器，用户取出自行车，开始计时。用户也可以使用绑定的IC卡进行借车，此时无需App操作，借车成功后App会收到服务器的通知。
* 交互要求: 需要验证用户的状态，信用及押金。成功后给予借车成功的通知，如果失败返回失败原因。失败的原因包括：信用不足，账户被禁用，账户余额不足，已借出其他车辆，已预约其他车辆等。

## 还车确认

* 用户：普通用户
* 功能描述: 在用户将自行车安装到锁止器（车桩）上以后，系统会将车辆的信息通过网络传回到服务器上，服务器根据数据判断借车人并将还车结果通知到借车者的App账户上。如果是刷卡借车，还必须要刷卡更新一下IC卡内的状态。
* 交互要求: 结果要实时推送到用户App端，如果App端不在线，最多保留24小时。

## 车辆评估与报修

* 用户：普通用户
* 功能描述: 还车后可以对所借的车辆进行状态评估，供其他用户或者维护人员参考。主要操作过程是选中一条借还记录，然后选择评估按钮，进入界面填写反馈意见或者直接选择车辆状态类型（良好，不好骑，车锁故障，车篮故障，车胎故障，链条故障等）。
* 交互要求: 无

## 车辆信息查询

* 用户：普通用户，企业用户
* 功能描述: 根据车辆的编码查询车辆的基本信息。基本信息的数据暂定为：车辆二维码，产地，生产厂家，出厂日期，当前位置，当前状态等
* 交互要求: 有可能需要调用相机扫描自行车上二维码。位置状态等信息可通过按钮操作二次显示。

## 车辆维护记录查询

* 用户：普通用户，企业用户
* 功能描述: 查询指定车辆的维修保养记录。
* 交互要求: 记录按照时间倒序显示。

## 意见反馈

* 用户：普通用户
* 功能描述: 市民通过界面提交合理化建议，维修建议等信息。
* 交互要求: 界面提供意见类别信息供用户选择（例如：计费投诉，流浪车，调度建议等）。

## 登录系统

* 用户：普通用户，企业用户，管理员用户
* 功能描述: 提供用户名和密码，进入系统。
* 交互要求: 不同类别的用户数据存储在不同的表格中，因此不同类别的用户不能互相登录，用户登录后需要返回给用户登录凭据，供后续功能使用。

## 自身数据维护

* 用户：普通用户，企业用户和管理员用户
* 功能描述: 维护用户的基本信息，例如用户名，密码，联系方式等。
* 交互要求: 无

## 车辆查找

* 用户：企业用户
* 功能描述: 给定车辆的编码，定位到车辆的位置信息。如果在桩状态，给出站点和桩号；如果是维护保养，给出维护保养的站点；如果是借出状态，给出骑行的用户信息以及借出站点信息。如果用户App允许共享GPS信号，则给出车辆的GPS地址。
* 交互要求: 无

## 取出车辆

* 用户：企业用户-巡检
* 功能描述: 在管理界面上扫码一次取出自行车。主要用户检查车辆状况。确认状况良好后还入桩中。否则进行故障车确认。
* 交互要求: 扫码一次。

## 故障车确认

* 用户：企业用户-巡检
* 功能描述: 确定车辆存在故障，无法服务时，并标记其故障类型，提交故障数据，同时设置车辆状态信息为维修中。如任务来源于普通用户，则反馈信息给用户。
* 交互要求: 扫码一次。

## 查看维修任务

* 用户：企业用户-维修人员
* 功能描述: 用户登录后可以查看归属于自己的维修任务清单，用户可以选中一条查看自行车信息。
* 交互要求: 无

## 设置车辆状态

* 用户：企业用户-维修人员
* 功能描述: 对自行车扫码后自动开始维修任务，自动设置自行车的状态为维修中。
* 交互要求: 需要扫码

## 结束维修任务

* 用户：企业用户-维修人员
* 功能描述: 对正在进行的维修任务进行确认，反馈维修的结果，有两种。一种是维修完毕，设置自行车状态为可用，重新入桩或入站；另一种是无法当场修复，则设置自行车状态为进站维修，将其运回维修站。
* 交互要求: 无

## 录入巡检任务

* 用户：企业用户-客服人员
* 功能描述: 根据用户电话，输入需要维修的车辆等信息。根据车辆的地址，分配给负责的巡检人员。
* 交互要求: 无

## 查看用户信息

* 用户：企业用户-客服人员
* 功能描述: 给定客户的编号，联系电话或者姓名等资料，调出用户的其他信息。
* 交互要求: 无

## 查看用户借还车记录

* 用户：企业用户-客服
* 功能描述: 根据用户的编号，查看该用户的借还车记录。
* 交互要求: 无

## 用户信息维护

* 用户：系统管理员
* 功能描述: 创建-修改-删除用户账户，维护账户的信息资料，维护账户的属性以及安全等级。
* 交互要求: 无

## 异常数据处理

* 用户：系统管理员
* 功能描述: 处理数据库中基本表以及各关系表中的异常数据。
* 交互要求: 无

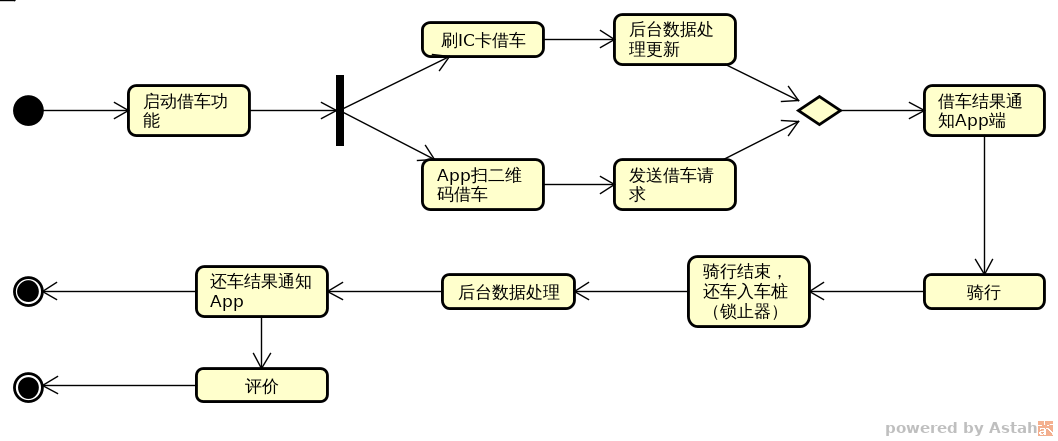
## 系统维护

* 用户：系统管理员
* 功能描述: 进行一些与数据安全等有关系的操作，以及一些需要定期运行的任务。待后续进行完善。
* 交互要求: 无

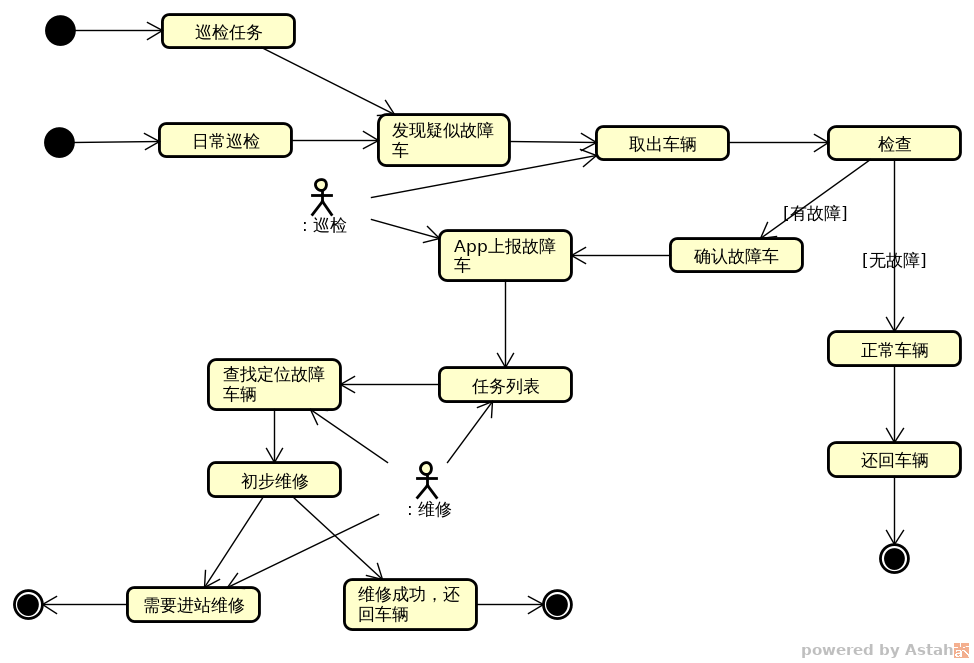
# 流程分析

此处对涉及的主要流程进行分析和说明。

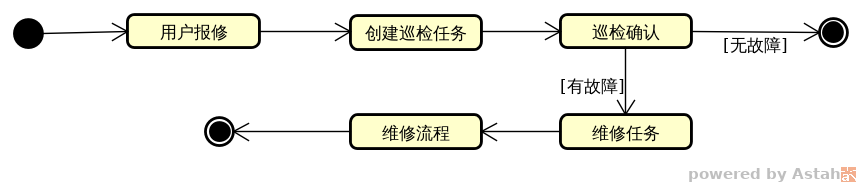
## 流程1：借-还车-评价流程



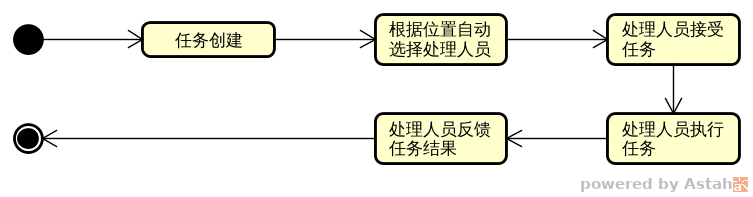
## 流程2：巡检-维修流程



## 流程3：用户报修-巡检确认-维修流程

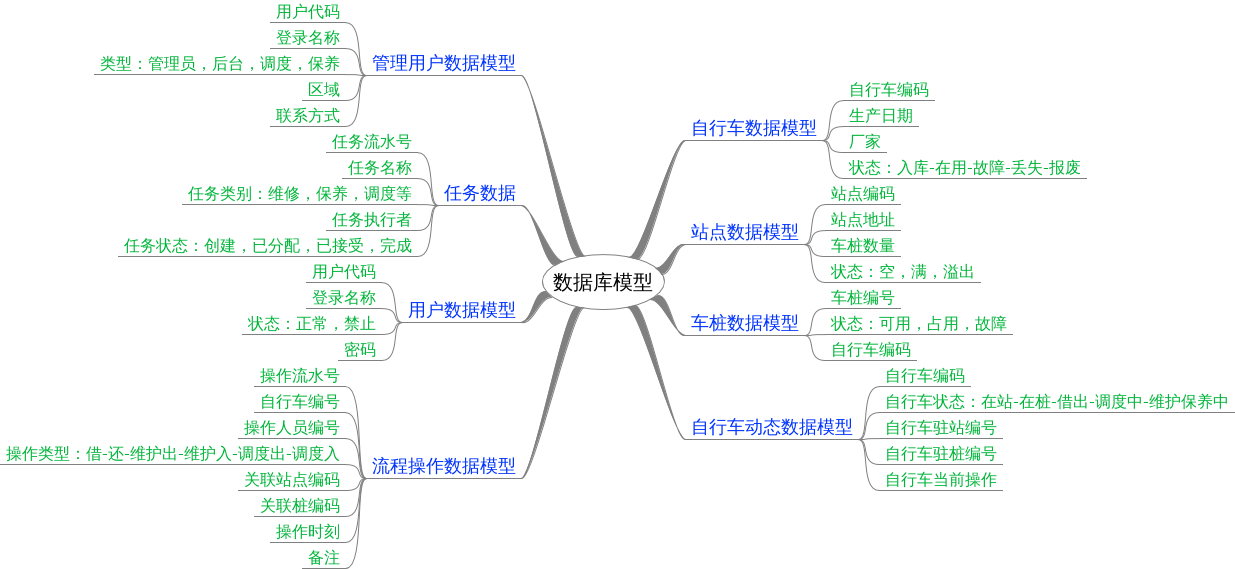


## 流程4：任务创建-分配流程



# 数据分析

待完善。以下是简要的数据模型设计：



# 限制条件

限制条件为研发本软件的约束行为，在进行系统设计和开发的时候必须要考虑到这里的约束条件。

## 开发周期

第一期开发周期为3周。本系统采用滚动式设计与开发的方法（参考**极限编程**），首先开发最核心的功能，然后随着需求的一步步明确，逐步滚动开发到整个系统。中间预计会进行多次迭代开发，从数据模型设计到软件系统模块都有可能会发生改变。开发周期3周为第一个可供迭代版本的开发完成时间。

因为时间有限，而数据模型和流程方面都需要做许多的用户调研，时间明显不够，因此本周期重点先进行客户端的开发，服务器后台和数据库进行配合；下一阶段再重点进行服务器后台和数据库的设计开发。

## 数据

系统测试需要一定的数据的支持，在目前情况下可以使用fake数据；另一个方法是从现有的数据中选取一部分导入到系统中来进行测试，届时视开发的进度和难易程度予以取舍。