软件需求规格说明书

目录

[1 引言 1](#_Toc381109000)

[1.1 目的 1](#_Toc381109001)

[1.2 文档约定 1](#_Toc381109002)

[1.3 预期的读者和阅读建议 1](#_Toc381109003)

[1.4 产品的范围 1](#_Toc381109004)

[1.5 参考文献 1](#_Toc381109004)

[2 总体描述 1](#_Toc381109000)

[2.1 产品的前景 1](#_Toc381109001)

[2.2 产品特性 1](#_Toc381109002)

[2.3 用户类和特征 1](#_Toc381109003)

[2.4 运行环境 1](#_Toc381109004)

[2.5 设计和实现上的限制 1](#_Toc381109004)

[2.6 假设和依赖 1](#_Toc381109004)

[3 系统特征 1](#_Toc381109000)

[3.1 租车 1](#_Toc381109001)

[3.2 创建、浏览、修改、删除订单 1](#_Toc381109002)

[3.3 创建、浏览、修改车辆信息 1](#_Toc381109003)

[3.4 付费 1](#_Toc381109004)

[3.5 理赔 1](#_Toc381109004)

[4 外部接口需求 1](#_Toc381109000)

[4.1 用户界面 1](#_Toc381109001)

[4.2 硬件接口 1](#_Toc381109002)

[4.3 软件接口 1](#_Toc381109003)

[4.4 通信接口 1](#_Toc381109004)

[5 其它非功能需求 1](#_Toc381109000)

[5.1 性能需求 1](#_Toc381109001)

[5.2 安全设施需求 1](#_Toc381109002)

[5.3 安全性需求 1](#_Toc381109003)

[5.4 软件质量属性 1](#_Toc381109004)

[5.5 业务规则 1](#_Toc381109003)

[5.6 用户文档 1](#_Toc381109004)

[6 其它需求 1](#_Toc381109000)

[6.1 支持多种操作系统 1](#_Toc381109001)

[6.2 系统安装方便 1](#_Toc381109002)

[6.3 支持多种语言 1](#_Toc381109003)

[6.4 具有一定的容错性 1](#_Toc381109004)

[6.5 支持多种方式登陆 1](#_Toc381109003)

[6.6 找回密码功能 1](#_Toc381109004)

[6.5 具有一定的独立性 1](#_Toc381109003)

[6.6 抵制非法操作 1](#_Toc381109004)

1. 引言

1.1目的

软件需求规格说明描述了“汽车租赁系统（Car’s Rental Management System，CRMS）”2.0版本的软件功能型需求和非功能性需求。

编写此文档的目的是进一步定制软件开发的细节问题，希望能使本软件开发的工作更具体。此文档可以使客户、需求分析团队、软件开发团队以及软件测试团队等有一个初始的共同的理解，方便在以后软件开发团队和软件测试团队和需求分析团队不冲突，防止以后工作的低效。

这一文档计划由实现和验证系统正确功能的项目团队成员来使用。这个文档说明了各项功能需求、性能需求和数据要求等等问题，明确标明了各功能的实现过程，阐述了使用背景以及范围。提供给客户解决问题或者达到目标所需的条件和权能，提供一个度量和遵循的基准。

本需求规格说明书在于阐述开发者对项目商业需求的理解，项目的开发将会遵循并实现文档中的商业需求，在项目验收时，本说明将作为评估标准之一。

1.2文档约定

本文档按以下要求和约定进行书写：

1. 正文为小四字号，宋体字号，标题均粗。
2. 无特殊情况下，字体颜色均采用黑色。

1.3预期的读者和阅读建议

1.3.1客户

客户在阅读此软件需求规格说明书需要仔细对照需求分析人员得出的

需求，检查是否符合自己的要求。

一旦发现有出入的地方，需要及时与分析人员沟通并修改，发现的越早

越好，尽量避免开发人员开始开发以后再发现问题并修改。

1.3.2项目经理

项目经理在阅读此项软件需求规格说明书时，需要理解客户的总体需求，

并且根据客户的要求确定项目开发的总体进度和时间限制，根据任务，合理

有效的安排各个人员完成任务。

1.3.3开发人员

开发人员在阅读此项软件需求规格说明书时，需清晰的了解客户的需求，

并且根据项目经理的任务安排，及时有效的完成开发任务。在开发阶段，开

发人员也需要时常与分析人员和客户进行有效的沟通，方便项目以后的开发，

防止出现不同的理解，导致项目的重做。

1.3.4测试人员

测试人员在阅读此项软件需求规格说明书时需要仔细阅读客户的需求，

并且根据客户的需求来编写有效的用例来测试。

1.4产品的范围

汽车租赁系统（Car’s Rental Management System，CRMS）旨在拓展租车公司的业务，将租车业务推广成方便快捷的业务。

租车公司的业务目标是在半年内把开发此系统的成本收回，半年之后开始收益，并且把营业额逐步提升，提高公司的效益。

1.5参考文献

i、Karl E. Wiegers 著，刘伟琴、刘洪涛译的软件需求第二版（Software Requirements 2，清华大学出版社。

ii、 Karl Wiegers所著的Process Impact Internet Development Standards，version 1.3。

网址是：

www.processimpact.com/corporate/standards/PI\_intranet\_dev\_std.doc

iii、Christine Zambito所著的Process Impact Business Rules Catalog。

网址是：

www.processimpact.com/corporate/policies/PI\_business\_rules.doc

iv、Christine Zambito所著的Process Impact Internet Application User

Interface Standard，Version 2.0。

网址是：

[www.processimpact.com/corporate/standards/PI\_internet\_ui\_std.doc](http://www.processimpact.com/corporate/standards/PI_internet_ui_std.doc)

1. 总体描述

2.1 产品的前景

中国汽车租赁业于1989年起源于北京，进入2001年前后，掀起了第三轮的汽车租赁企业发展高峰，从此，国内汽车租赁行业有了大跨步的发展，从原来仅限在北京、上海、广州等大型城市的汽车租赁业务，发展到了中小城市、乃至县镇。

  截至2006年底，中国13个中心城市，共有汽车租赁公司787家，租赁汽车44864辆，租赁站点446个，从业人员14682人，营业额20亿元。

2007年，中国汽车租赁市场经历了一场异常迅速的发展，各地的汽车租赁公司如雨后春笋纷纷出现，国内汽车租赁市场到2007年已拥有2000多家租赁企业，汽车租赁市场供租赁车接近15万辆，营业额超过100亿元。虽然其发展速度比西方汽车租赁 行业创始之初要快得多，但中国的汽车租赁市场仍处于起步阶段，绝大多数企业规模很小。

截至2008年6月底，中国13个中心城市的汽车租赁企业有800余家，但运营车辆仅4.5万辆，其中80%的企业运营车辆不足50辆。

中国汽车租赁行业，无论是从中国宏观经济的走向趋势还是微观的社会基础来看，都有着无比光明的发展前途。到2010年左右，随着中国信用体系的建立，市场经济的进一步完善，中国汽车租赁业将会有更大更快的发展。预计到2015年，中国汽车租赁市场的租赁车辆需求将达到30万至50万辆，营业收入将达到200亿元。

在这样的大背景下，使用汽车租赁系统可以规范企业的管理和经营行为，减少企业经营成本，提高工作效率。

2.2 产品特性

|  |
| --- |
| 租车 |
| 创建、浏览、修改、删除订单 |
| 创建、浏览、修改车辆信息 |
| 付费 |
| 理赔 |

2.3 用户类和特征

2.3.1 租车公司

租车公司既是客户也是用户。

所有需要用到该系统的租车公司都要购买汽车管理系统，租车公司在购买该系统后，

使用该系统管理公司的所有车辆。

所有购买该系统的租车公司每天都要使用该系统，主要是提供满足用户租赁需求的

车辆及人员。他们还需通过该系统计算每个月的平均收益。

2.3.2 管理员

管理员有两大类，为网络系统管理员和信息系统管理员。

管理员必须每天使用该系统，网络系统管理员，主要负责服务器系统的设计、安

装、配置、管理和维护工作，为内部网的安全运行做技术保障。

服务器是网络应用系统的核心，由系统管理员专门负责管理；信息系统管理员负

责具体信息系统日常管理和维护，具有信息系统的最高管理[权限](http://baike.baidu.com/view/562732.htm" \t "http://baike.baidu.com/_blank)。

2.3.3 第三方支付系统

第三方支付系统需要24小时不间断的使用该系统。

主要任务是负责租车公司和使用租车公司业务的人员之间的资金流通，保障用户

在转账时资金的安全性，以及用户账户、密码等私人信息的隐蔽性。

第三方支付系统涉及到使用租车公司业务的人员个人隐私，以及资金的周转，因

此，第三方支付系统要有很高的安全性。

2.3.4 保险公司

保险公司为每辆车都提供保险。

保险公司需要24小时不断的使用该系统，一旦所投保的车辆发生事故，保险公

司有义务在第一时间处理事故，赔偿被保险公司的经济损失。

2.3.5 租车人

租车人是汽车租赁系统的主要用户，租车人来自社会的各个阶层，而且来自不同

的年龄段，租车人不需要时时刻刻的登陆该系统，只有当租车人要用到租车公司所提

供的业务是时候，才需要登陆该系统。

租车人的主要责任是租赁汽车，并且保护好所租汽车。如果租车的同时需要司机，

则应该尊重司机。当所需的服务结束后，应该及时向第三方支付系统转账。

2.3.6 司机

司机属于租车公司的员工，他们每天上班前需要登陆系统。当司机被公司分配到任务时，司机不到随意的拒接。司机的主要责任是要提供给租车人良好的服务，以及安全的出行环境。

2.3.7 租车公司自己的系统

他的主要作用是储存车辆司机的信息。

2.3.8 计费系统

他的主要作用是计算租车人所要支付给租车公司的费用，并提交给第三方支付系统。

2.4 运行环境

2.4.1服务器：

|  |  |
| --- | --- |
| 服务端 | 操作系统:  Windows2003企业版  Windows2008企业版  SQL Server2005企业版 |
| 数据库 | MS-SQLServer2005企业版  MYSQL5.5 |
| 其他 | Tomcat6 jdk6 |

|  |  |
| --- | --- |
| PC端操作系统 | 操作系统:  Windows XP  Windows vista  Windows 7  Windows 8 |
| 移动端 | android:phone/平板  ios:iPhone/iPad |
| 其他 | MS-Office2003  MS-Office2007  MS-Office2010  MS-Office2013 |

2.4.2客户端：

注：原则上只要服务器运行环境良好，系统支持的用户数没有限制。

2.5 设计和实现上的限制

2.5.1 编程语言

因为该系统不是在基础设施上的项目，要使用面向对象的思想编程，在可以实现面

向对象的语言中，Java是目前我们最熟悉的，所以必须使用Java编程语言。

2.5.2 数据库管理软件

系统必须使用Java接口同关系型数据库管理软件建立连接。因为使用的是Java编

程。

2.6 假设和依赖

客户必须具备基本的计算机知识。

管理员必须经过一定的培训。

3 系统特征

3.1 租车

3.1.1 描述和优先级

租车人登陆系统，身份得到验证之后，他们就可以租车，并且可以要求送到公司内

指定的地点，也可以去公司取车。优先级为高。

3.2.2 刺激/响应序列

刺激：租车人请求租车，可以是一辆或多辆。

响应：系统向租车人咨询细节、付费方式和租车说明；

刺激：租车人请求改变所租的车。

响应：如果订单状态是“已接受”，则系统允许用户编辑以前的订单。

刺激：租车人请求取消订单。

响应：如果订单状态是“已接受”，则系统取消订单。

3.3.3 功能性需求

|  |  |
| --- | --- |
| 租车人下订单 | 登陆到“汽车租赁系统”的租车人可以通过该系统租车，租一辆或多辆都可以 |
| 租车人订单登记 | 系统将确认下订单的租车人的付费方式是从支付宝中扣除租车费的付费方式 |
| 租车人订单登记未成功 | 如果租车人没有从支付宝中扣除租车费的付费方式，那么系统将为租车人提供一些选择方案，租车人可以现在注册并继续进行租车，也可以租车后去租车公司提交现金，或者还可以退出“汽车租赁系统” |
| 租车人下订单的日期 | 系统将提示租车人输入租车日期 |
| 取车的方式 | 租车人将指定是去公司取车还是送到指定的地点 |
| 取车的地点 | 租车人如果要求送到指定的地点，那么租车人将提供一个有效的取车地点 |
| 取车的时间 | 系统将显示剩余的有效取车时间，租车人可以从显示的剩余时间中选一个时间。 |

3.2 创建、浏览、修改、删除订单

3.2.1 描述和优先级

租车人登陆系统，身份得到验证之后，他们就可以创建、浏览、修改、删除订单。

优先级居中。

3.2.2 刺激/响应序列

刺激：租车人请求浏览订单。

响应：如果有“已接受”或者“已完成”的订单，则系统允许用户浏览订单。

刺激：租车人请求修改订单。

响应：如果有“已接受”的订单，则系统允许用户修改订单。

刺激：租车人请求删除订单。

响应：如果有“已接受”或者“已完成”的订单，则系统允许用户删除订单。

3.3.3 功能性需求

|  |  |
| --- | --- |
| 租车人创建订单 | 租车人在订单完成后，通过系统确认订单完成，订单将正式创建 |
| 租车人创建订单的时间 | 系统将提醒订单人输入时间 |
| 租车人未确认订单 | 若租车人在规定时间内未确认订单，那么系统将自动将该订单视为确定过了的订单 |
| 租车人浏览订单 | 登陆到“汽车租赁系统”的租车人可以通过该系统浏览“已接受”或者“已完成”的订单 |
| 租车人浏览订单未成功 | 如果租车人没有“已接受”或者“已完成”的订单，那么系统将通知租车人没有可以浏览的订单 |
| 租车人浏览订单的时间 | 系统将自动记录租车人浏览订单的时间 |
| 租车人修改订单 | 登陆到“汽车租赁系统”的租车人可以通过该系统修改“已接受”的订单 |
| 租车人修改订单未成功 | 如果租车人输入的信息是非法信息（系统所不支持的信息），那么系统将不允许租车人修改，并提醒租车人输入了非法信息 |
| 租车人修改信息的时间 | 系统将自动记录租车人修改订单的时间 |
| 租车人删除订单 | 登陆到“汽车租赁系统”的租车人可以通过该系统删除“已接受”或者“已完成”的订单 |
| 租车人删除订单的时间 | 系统将自动记录租车人删除订单的时间 |

3.3 创建、浏览、修改车辆信息

3.3.1 描述和优先级

租车公司登陆系统，身份得到验证后，租车公司可以创建、浏览、修改车辆信息。

租车人登陆系统，身份得到验证后，租车人可以浏览车辆信息。优先级高于创建、浏览、

修改、删除订单，但是低于租车，优先级较高。

3.3.2 刺激/响应序列

刺激：租车公司请求创建车辆信息。

响应：系统允许租车公司创建信息。

刺激：租车公司请求浏览车辆信息。

响应：如果已有车辆信息，则允许租车公司浏览车辆信息。如果没有车辆信息，系

统则提示租车公司创建车辆信息。

刺激：租车公司请求修改车辆信息。

响应：如果已有车辆信息，则允许租车公司修改车辆信息。如果没有车辆信息，系

统则提示租车公司创建车辆信息。

刺激：租车人请求浏览车辆信息

响应：如果已有车辆信息，则允许租车公司浏览车辆信息。如果没有车辆信息，系

统则提示租车人没有可浏览的车辆信息。

3.3.3 功能性需求

|  |  |
| --- | --- |
| 租车公司创建车辆信息 | 登陆到“汽车租赁系统”，租车公司可以创建车辆信息 |
| 租车公司创建车辆信息的时间 | 租车公司每完成一辆汽车信息的创建，系统将提醒租车公司输入创建的日期 |
| 租车公司浏览车辆信息 | 登陆到“汽车租赁系统”，租车公司可以浏览车辆信息 |
| 租车公司浏览车辆信息失败 | 如果租车公司尚未创建车辆信息，则浏览车辆信息失败，并且系统提示租车公司创建车辆信息 |
| 租车公司浏览车辆的时间 | 系统自动记录租车公司浏览车辆的时间 |
| 租车公司修改车辆信息 | 登陆到“汽车租赁系统”，租车公司可以修改车辆信息 |
| 租车公司修改车辆信息失败 | 如果租车公司尚未创建车辆信息，则修改车辆信息失败，并且系统提示租车公司创建车辆信息 |
| 租车公司修改车辆信息的时间 | 系统自动记录租车公司修改车辆的时间 |
| 租车人浏览车辆信息 | 登陆到“汽车租赁系统”，租车人可以浏览车辆信息 |
| 租车人浏览车辆信息失败 | 如果租车公司尚未创建车辆信息，则浏览车辆信息失败 |
| 租车人浏览车辆的时间 | 系统自动记录租车人浏览车辆的时间 |

3.4 付费

3.4.1 描述和优先级

租车人在租车结束后，登陆“汽车租赁系统”,通过身份验证后，就可以通过系统

进行付费。优先级低。

3.4.2 激励/响应序列

激励：租车人请求付费。

响应：系统允许租车人付费。

3.4.3 功能性需求

|  |  |
| --- | --- |
| 租车人付费 | 保险公司请求理赔 |
| 租车人付费失败 | 如果租车人没有从支付宝中扣除租车费的付费方式，那么系统将为租车人提供一些选择方案，租车人可以现在注册并继续进行付费，如果其他方案都不能付费成功，则系统提示租车人付费失败。租车人需到租车公司支付现金 |
| 租车人付费时间 | 系统自动记录租车人付费时间 |

3.5 理赔

3.5.1 描述和优先级

保险公司登陆系统，身份得到验证之后，保险公司就可以进行理赔。优先级低。

3.5.2 激励/响应序列

激励：保险公司请求理赔。

响应：系统允许保险公司进行理赔。

3.5.3 功能性需求

|  |  |
| --- | --- |
| 保险公司理赔 | 登陆到“汽车租赁系统”，保险公司进行理赔 |
| 保险公司理赔失败 | 造成事故发生的原因不再保险公司理赔范围内 |
| 保险公司理赔时间 | 系统自动记录租车人付费时间 |

4 外部接口需求

4.1 用户界面

4.1.1设计风格

“汽车租赁系统”的屏幕界面将遵照PROCESS IMPACT INTERNET APPLICATION

INTERFACE STANDARD版本2.0，产品系列的风格以简约为主不要过于冗余的按键。

4.1.2局限因素

用户多数是手机用户，所以屏幕布局要考虑到手机屏幕的局限性和手机的性能，

电脑客户端的用户考虑电脑的性能。

4.1.3快捷键

需要手势快捷键，并且可以自定义手势快捷键。

4.1.4外来网页

系统对每个显示的外来HTML网页都提供帮助链接，并且结束如何使用这些网页。

4.1.5系统操作

对系统的操作不仅可以使用键盘输入和触控完成，对有障碍的用户还可以通过语音

完成。

4.1.6其他

具有扫描二维码的功能。

4.2 硬件接口

硬件接口现在还没有确定

4.3 软件接口

4.3.1 租车公司内部系统

4.3.1.1 “汽车租赁系统”通过程序界面向“租车公司内部系统”获

得可租车辆的信息、司机的信息和其他的信息帮助用户做决定。

4.3.1.2 “租车公司内部系统”通过程序界面像用户推送信息。

4.3.1.3 “租车公司内部系统”通知“汽车租赁系统”某辆车被出租

时，程序界面将其从目录中删除。

4.3.1.4 “汽车租赁系统”通过程序界面向“租车公司内部系统”发

送租车订单的相应信息，租车公司内部系统确认订单是否有效。

4.3.2 第三方支付系统

4.3.2.1 “第三方支付系统”通过程序界面和“计费系统”通信完成

付费。

4.3.3 保险理赔系统

4.3.3.1 “保险理赔系统”通过程序界面和“租车公司内部系统”通信，

完成事故的损失鉴定。

4.3.3.2 “保险理赔系统”通过程序界面和“第三方支付系统”通信，

完成赔偿金额的收取或发放。

4.3.4 计费系统

4.3.4.1 用户通过程序界面查看“计费系统”中的金额，通过“第三方支付系统”

向公司付费。

4.3.4.2 对信用较高的老客户提供延时付款的功能。

4.4 通信接口

4.4.1 “汽车租赁系统”将向用户注册的手机发送短信，以确认订单的价格的等详细

信息。

4.4.2 “汽车租赁系统”将向顾客发送电子邮件，来向用户提供某一周期内的报表，

并且推送公司近期的一些活动。

5. 其它非功能需求

5.1 性能需求

5.1.1 在当地时间早上8点到10点这一段高峰时间，系统能适应15000个用户，平均

每个用户的会话时间将持续10分钟。

5.1.2系统所生成的所有web页面，通过速率为50KBps的调制解调器在不超过7秒的

范围内都能全部下载下来。

5.1.3 用户提交了查询之后，对查询的响应时间不超过5秒，在此时间内要将查询结

果显示在屏幕上。

5.1.4 用户提交了订单之后，系统对订单的响应不超过3s，在此时间内要向用户界面

发送确认消息，并且在20s内发送短信到用户手机确认订单。

5.2 安全设施需求

5.2.1 强制控制使用系统的人数不超过20000人次

5.2.2所在地区若出现恶劣天气，管理员暂时关闭系统。

5.3 安全性需求

5.3.1 所有涉及功能信息或个人身份信息的网络事务，都要按照128位进行加密操作。

5.3.2 除了浏览租车公司基本信息和车辆的基本信息外，用户要登录到“汽车租赁系

统”才能进行其他的操作。

5.3.3 顾客的登陆受到计算机系统访问控制策略的限制。

5.3.4 租车公司的管理人员只有经过系统管理员的授权之后才能对自己公司的相关信

息进行编辑。

5.3.5 系统只允许顾客浏览自己的相关信息和历史订单，而不允许其浏览其他顾客的

信息和历史订单。

5.4 软件质量属性

5.4.1 可用性：“汽车租赁系统”针对所有用户提供24小时不间断使用。

5.4.2 健壮性：在订单得到确认或者取消之前，用户和系统的连接中断，那么用户可通

过“汽车租赁系统”恢复不完整的订单。

5.4.3 易用性：“汽车租赁系统”面对的用户群是拥有汽车驾驶证的人，年龄跨度较大，

所以要求软件可操作性强，上手难度低。

5.4.4 可移植性：“汽车租赁系统”可以在Android和iOS平台下使用，并且在Windows

和OS系统中都能够使用。

5.5 业务规则

5.5.1租车公司的管理人员只有经过系统管理员的授权之后才能对自己公司的相关信

息进行编辑。

5.5.2系统只允许顾客浏览自己的相关信息和历史订单，而不允许其浏览其他顾客的信

息和历史订单。

5.5.3 注册用户只能使用一个手机号码，一个手机号码只能对应一个绑定账号。

5.5.4 一切除了付款退款外的资金流动都需要管理员授权。

5.6 用户文档

5.6.1 免责说明

5.6.2 使用说明书

5.6.3 在线帮助

5.6.4 资费标准

5.6.5 使用技巧

1. 其他需求

6.1支持多种操作系统

当今市场上，各种品牌的手机层出不穷，或外形大气美观，或拍照功能卓越，用户

选的眼花缭乱的时候，也于无形中给我们的系统提出了一个必须满足的需求，那便是支

持多种操作系统。顾客使用的手机型号各不相同，所适用的操作系统也各不相同。我们

的app必须具有在各种操作系统上使用的包容性，app才会有市场，并被广泛使用，而

不是仅仅局限于一小部分群体，同时，只有达到此种便捷程度的系统，才能真正为用户

和公司解决实际生活中的问题。

6.2系统安装方便，易于维护

顾客和使用我们系统的租车公司的管理员大多不是技术“大牛”，不具备掌握复杂

繁琐的安装程序和操作指令的能力，所以，我们的app绝对不能操作复杂，难以维护，

否则，对公司而言，不仅不能提高效率，还会给公司带去大笔维护成本，对顾客而言，

安装程序复杂，多少会产生烦躁心理，加上同类产品的竞争，那么app必然难以提高用

户量，长此以往，我们的系统便毫无市场可言。所以，我们的系统必须安装方便，易于

维护。

6.3支持多种语言

要做好一个app，我们的眼光不能仅仅定位在国内，而是要放眼国际，把不同肤色、

不同大洲，不同语言的人们一起使用我们的app作为终极目标。就近的来说，支持不同

语言对初来乍到的留学生而言也是莫大的便捷，人生地不熟的他们既能快速地解决出行

问题，也能在异国他乡找到久违的亲切感，更有利于将我们的系统推出国门，走向国际

化。所以，支持多种语言是一种趋势。

6.4出现错误时，启动应急机制，具有一定的容错性

任何一个系统不会在一开始就十分完美，没有任何错误，应对这些bug，除了不断

更新我们的系统之外，也必须建立容错机制和应急机制。当用户或租车公司进行操作，

系统出现错误时，app仍能以简易版形式提供服务，即最主要功能必须确保不会轻易崩

溃，同时，必须启动应急纠错机制，通过发送错误报告，后台紧急处理，以求在最短的

时间内解决错误，从而获得更高的用户体验。当我们的用户体验远高于同类app时，我

们所具备的高竞争力自然不言而喻。

6.5支持多种方式登录，例如qq登录，微博登录

部分新用户想要使用我们的系统，却没有账号，却又懒于申请账号，填写资料的各

种麻烦，或者在某些紧急情况下，用户无法使用账号登录，支持其他方式登录便会给用

户带来便捷，同时通过与其他社交软件，网站的合作，可以提高系统的曝光率和分享指

数，从而扩大用户人群。

6.6拥有安全系数较高的找回密码功能

一旦涉及密码，便会涉及隐私和财产问题，这些数据都是高级机密，是我们系统需

要启动高级防护机制的区域，必须慎重，否则，一旦泄露，所出现的损失便是我们无法

承受的。因此，我们的系统必须拥有较高的安全系数。此外，找回密码必然存在密码丢

失现象，那么找回密码功能必然就有一定风险，但用户难免有忘记密码的时候，如果缺

少此功能，用户就难以再次使用我们的系统，所以，此项功能确实是必须的。

6.7具有一定的独立性

app代码之间必须具有一定的独立性。试想，如果系统各个代码模块相互关联，不

可分割，一旦有一处出现了错误，那么整个系统必然全盘崩溃，而且也很难找出到底是

哪里出了错误，这也就等同于我们开发的系统已经完全残废，之前所有心血付诸东流。

反之，如果我们系统的各块代码之间具有较高的独立性。那么，即使一处出现错误，我

们也可以保留正确的代码，并通过测试快速找出错误的部分，进行修复。这种独立性避

免了一处错误而导致整个app崩溃的情况，也易于找出错误，进行维护。

6.8抵制非法操作，并启动举报机制

遵守法律使我们进行开发的首要准则，不符合法律的开发行为必会让人自食恶果。

我们在自己遵守法律的同时，还要履行一定的监督义务。所以，我们的app必须抵制非

法操作，一旦发现违法操作，必须通过举报机制，坚决举报，确保操作环境与支付环境

的绝对安全，也确保交易和订单的绝对纯洁性。

7 附录1（术语表）

术语表

A

activity diagram（活动图） 一种分析模型，它显示了系统的动态视图，方法是描绘从一个活动到另一个活动的流。活动图与流程图(flowchart)相似。

actor(参与者) 扮演特定角色的一个人、一个软件系统或一个硬件设备，他们与系统交互以达到某一有用目的。参与者也称作“用户角色（user role）”。

analysis requirements（分析需求） 包括这样一些过程：将需求信息分成各种类别；评估需求是否达到了期望的质量；以不同的形式表示需求；从高层需求衍生出详细的需求；写上需求优先级；等等。

architecture（体系结构） 软件系统的结构，包括组成系统的软件组件和硬件组件，这些组件之间的接口和关系，以及对其他组件可见的组建行为。

assumption（假设） 在缺乏证据或明确的知识的情况下，被认为是真（true）的陈述。

B

baseline，requirements（基线，需求） 一个时间快照，表示针对某一产品的特定版本达成一致意见的、经过评审的、并得到批准的需求集。

business requirements（业务需求） 构建产品的组织或获得产品的客户的高层业务目标。

business rule（业务规则） 定义或约束业务某些方面的政策、原则、标准或规则。

C

cardinality（基数） 逻辑上与另一个对象或数据实体的实例相关的某个指定对象或数据实体的实例数。范例有一对一、一对多、多对多。

class（类） 描述了具有公共特性和行为的一个对象集，一般情况下，与业务或问题域中的真实世界条目（人、地点或东西）相对应。

class diagram（类图） 一种分析模型，它展示了一组系统类或问题域类及其关系。

constraint（约束） 设计和构造产品时，开发人员进行有效选择时必须强行接受的限制条件。

context diagram（关联图） 一种分析模型，它在很高的抽象层次上对系统进行了描绘。关联图识别与系统交互的系统外部的对象，但它并不展示系统的内部结构或行为。

customer（客户） 一类项目涉众，他们请求、付款、、选择、规定、使用或接受某一产品产生的输出。

D

data dictionary（数据字典） 有关对问题域重要的数据元素、结构和属性的定义的集合。

data flow diagram（数据流图） 一种分析模型，它描绘了过程、数据集合、端点以及它们之间的流，这种流表现了业务过程或软件系统的行为特点。

E

elicitation requirements（获取需求） 通过面谈、专题讨论会、工作流分析和任务分析、文档分析和一些其他机制，从各种来源中确认软件需求或系统需求的一种过程。

entity（实体） 收集或存储有关其数据的业务域中的一个条目。

entity-relationship diagram（实体-关系图） 一种分析模型，它确认了一对实体之间的逻辑关系。

event（事件） 系统环境中发生的触发或刺激，系统对此作出响应，例如功能行为或状态的变更。

external interface requirement（外部接口需求） 对软件系统和用户、另一个软件系统或硬件设备之间接口的描述。

F

feature（特性） 逻辑上相关的一组功能性需求集，为用户提供了某一能力，并使业务目标得以满足。

flowchart（流程图） 一种分析模型，它按照过程或程序的逻辑，显示了过程步骤和判定点。流程图与活动图（activity diagram）相似。

functional requirement（功能性需求） 对在某些特定条件下系统将展示的必需的功能或行为的陈述。

G

gold plating（镀金） 指定或构建到产品中的没必要或过分复杂的功能。

H

horizontal prototype（水平原型） 软件系统用户界面的部分实现或可能的实现。此原型用于评估软件系统的易使用性，也用于评估需求的完整性和正确性。也称为“行为原型（behavioral prototype）”或“模型（mock-up）”。

I

includes relationship（包含关系） 一种结构，这种结构把多个用例中重复出现的若干步骤提取出来作为单独的子用例，需要的时候，高层用例（或“调用”用例）可以调用此子用例。

N

nonfunctional requirement（非功能性需求） 对软件系统必须展示的特性或特点的描述，或软件系统必须遵照的约束，非功能性需求不同于可观察到的系统行为。

O

object（对象） 类的具体实例，类包括一组数据属性集和对这些属性执行操作的列表。

P

postcondition（后置条件） 描述用例成功完成之后系统状态的一种条件。

precondition（前置条件） 用例开始之前必须满足的条件或系统必须达到的一种状态。

procedure（步骤） 对完成某个指定活动所要执行的动作路径的一步一步的描述，描述了这一活动是如何完成的。

process（过程） 达到某一指定目的所执行的活动序列。“过程描述（process description）”是将这些活动的定义编写成文档。一个过程可以包括一个或多个步骤（procedure）。

Q

quality attribute（质量属性） 一种非功能性需求，描述了系统的质量或特性。例如包括有易使用性、可移植性、可维护性、完整性、有效性、可靠性、健壮性。质量属性需求描述了软件产品达到要求的特性的程度，而不是产品行为。

R

requirement（需求） 描述了客户需要或目标，或者描述了为满足这种需要或目标，产品必须具有的条件或能力。需求是这样一种特性，要求产品必须为涉众提供价值。

requirement attribute（需求属性） 有关需求的描述性信息，它丰富了需求的定义，超越了预期的功能陈述。例如包括有需求的来源、需求创建的理由、需求的实现优先级、需求的拥有者、实现需求的产品版本号和需求的当前版本号。

requirements allocation（需求分配） 把系统需求分配给各种构架子系统和组件的过程。

requirements analyst（需求分析员） 项目团队中的一种角色，主要负责与涉众代表协同工作，以便对项目需求进行获取、分析、编写规格说明、确认和管理。也可以称为业务分析员（business analyst）、系统分析员（system analyst）、需求工程师（requirements engineer）、或简称分析员（analyst）。

requirements development（需求开发） 一种过程，包括定义项目范围、确认用户类和用户代表、并获取、分析、编写规格说明和确认需求等，需求开发的产品是需求基线，它定义了所要构建的产品。

requirements engineering（需求工程）需求工程领域包括与理解产品必需的能力和属性相关联的项目生存期的所有活动。需求工程包括需求开发和需求管理。是系统工程和软件工程的一个分支学科。

requirements management（需求管理） 对一组已定义的产品需求的管理过程，跨越整个产品开发过程和产品使用寿命。包括跟踪需求状态、管理需求变更和需求规格说明的版本，并对其他项目阶段和工作产品的单个需求加以跟踪。

requirements traceability matrix（需求跟踪矩阵） 演示单个功能性需求和其他系统制品之间逻辑关系的一张表，这些系统制品包括其他功能性需求、用例、构架和设计元素、代码模块、测试用例和业务规则。

S

scope（范围） 当前项目将实现的最终产品前景中的某一部分，这一范围在项目范围内和项目范围外绘制了一个边界。

software development life cycle（软件开发生存周期） 对软件产品进行定义、设计、构建和验证的活动序列。

software requirements specification（软件需求规格说明） 软件产品的功能需求和非功能需求的集合。

specification（规格说明） 将系统需求以结构化的、共享的和可管理的形式编写成文档的过程，同样，产品也要经过这一过程。

stakeholder（涉众） 积极参与项目的一个人、小组或组织，受产品结果的影响，或影像产品的结果。

state-transition diagram（状态转换图） 一种分析模型，它展示了系统中对象在其生存期内所经过的状态顺序，以响应所发生的特定事件；或者展示了系统作为一个整体，它所可能存在的状态。状态图与状态转换图相似。

system requirement（系统需求） 包含多个子系统的产品的最高层需求，这些子系统可以全部是软件，也可以既有软件又有硬件。

T

tracing（跟踪）或可追溯（traceability） 定义一个系统元素（用例、功能性需求、业务规则、设计组件、代码模块、测试用例，等等）和另一个系统元素之间逻辑连接的过程。

U

use case（用例） 描述了执行者与系统之间逻辑上相关的可能交互集，系统的输出为执行者提供了价值。用例可以包含多个场景。

use-case diagram（用例图） 一种分析模型，它确认了与系统交互以达到有用目的的执行者，和每一个执行者将执行的各种用例。

user（用户） 直接或间接（例如，使用来自系统的输出，但并不亲自产生这些输出）与系统交互的客户，也称为“最终用户（end user） ”。

user class（用户类） 系统的一组用户，他们具有相似的特征和系统需求。当与系统交互时，用户类的成员起执行者的作用。

user requirement（用户需求） 用户通过系统必须能够达到的用户目的或任务，或者陈述了用户对系统质量的期望。

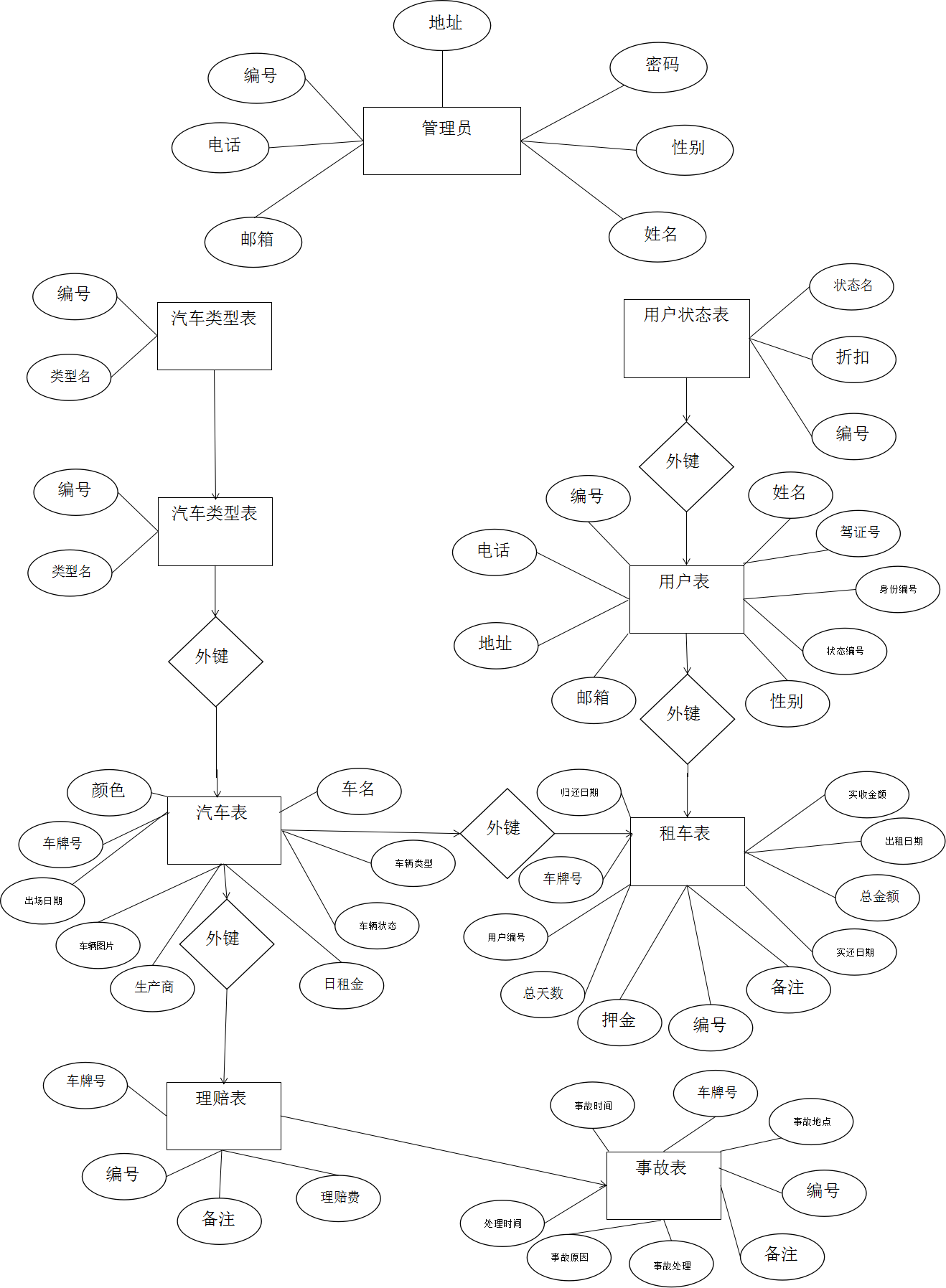
V

vision（前景） 有关新系统的最终目的和形式的一种长期的战略性概念。

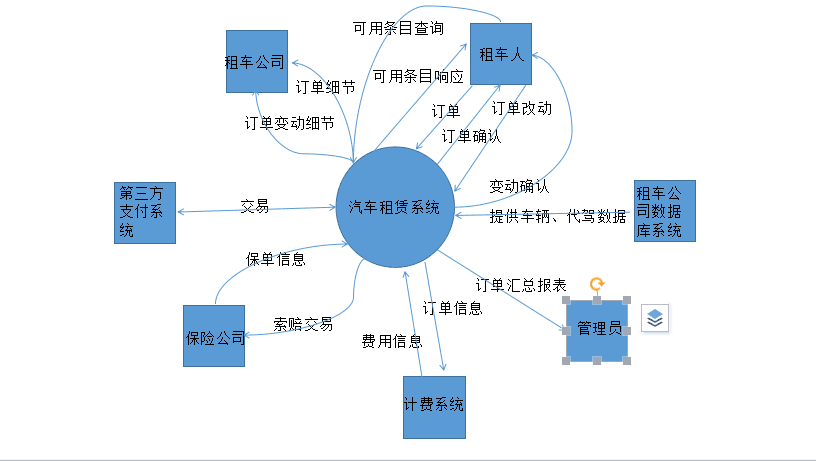
vision and scope document（前景与范围文档） 提出了新系统业务需求的一种文档，包括产品前景陈述和项目范围描述。

8 附录2（图表）

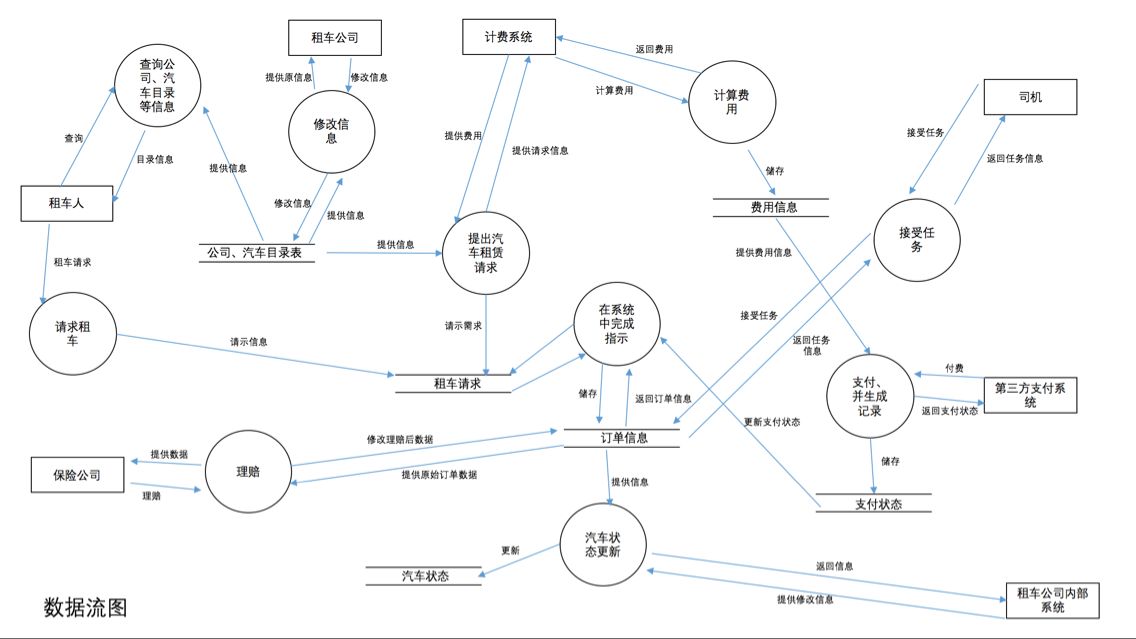
8.1 E-R图



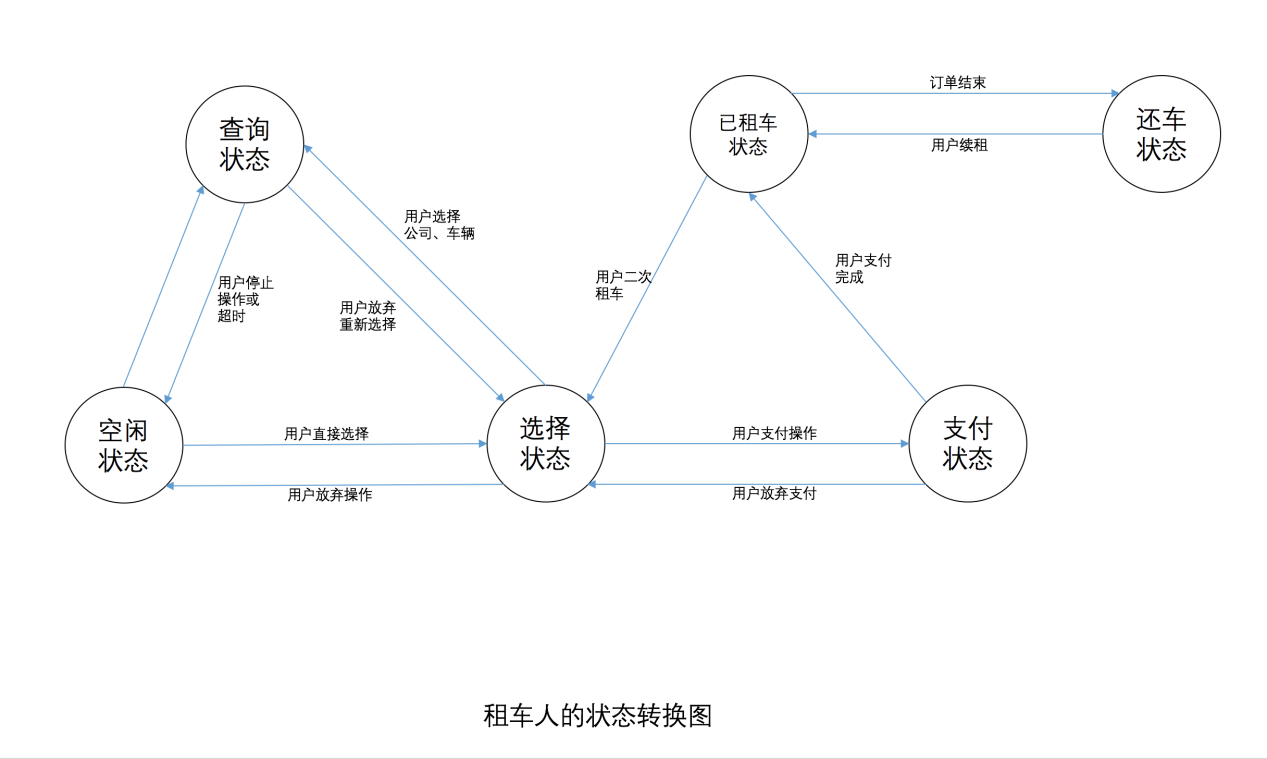
8.2系统关联图



8.3数据流图



8.4租车人的状态转换图



8.5被租车人的状态转换图

