服务外包大赛A组中文—项目详细方案

Valkyrie小组

2016-8



任意门-基于无人机与沉浸式VR技术的实时信息采集系统

目录

项目概要 3

1. 引言 5

1.1 编写目的 5

1.2 背景 5

1.3 建设目标 5

1.4 建设原则 6

2. 需求分析 6

2.1 利益相关者 6

2.1.1司机： 6

2.1.2货主： 7

2.1.3平台运营方： 7

2.2 现有平台分析 7

2.2.1货车帮： 7

2.2.2运满满： 7

2.2.3物流小秘： 7

2.2.4沃车港： 7

2.2.5总结： 8

2.3 系统边界 8

2.4 功能性需求 9

2.5 非功能性需求 9

2.6 可行性分析 10

2.6.1技术可行性 10

2.6.2资源可行性 10

2.6.3法律可行性 10

2.6.4实用性和可拓展性 10

2.6.5经济可行性 10

3. 概要设计 11

3.1 系统设计目标和原则 11

3.2 系统总体设计 12

3.2.1 架构设计 12

3.2.2 运行环境 14

3.3功能模块结构 15

3.3 数据库设计 20

3.3.1 概述 20

3.3.2 逻辑结构设计 21

3.3.3 物理结构设计 22

3.3.4 数据结构与程序的关系 25

3.4 关键技术介绍 25

3.3.1 视频流分屏渲染技术 25

3.4.1 视觉同步技术 26

3.4.2 增强现实技术 27

3.4.3 主流程 29

4. 详细设计 29

4.1 包设计 29

4.2 类设计 31

4.3 模块设计 34

4.3.1 视觉同步模块 34

4.3.2 视频流接收模块 36

4.3.3 分频渲染模块 38

4.3.4 增强现实模块 39

4.4 关键类说明 40

4.4.1 DemoBaseActivity类说明 40

5. 接口设计 41

5.1 接口设计概述 41

5.2 用户接口 42

5.3 外部接口 42

5.3.1 软件接口 42

5.3.2 硬件接口 42

5.4 内部接口 43

6. 运行设计 43

6.1 运行模块组合 43

6.2 运行控制 44

6.3 运行时间 44

7. 系统出错处理设计 44

7.1 出错信息 44

7.2 补救措施 44

7.3 系统维护设计 44

8. 运行效果展示 44

9. 附录 45

附录一 商业计划方案 45

9.1.1市场目标概述 45

9.1.2项目优势及特点介绍 45

9.1.3利润来源简析 45

9.1.4投资和预算 49

9.1.5融资方案（资金筹措及投资方式） 49

9.1.6预算及投资报酬 50

9.1.7 ROI分析 50

# 项目概要

* **前言**

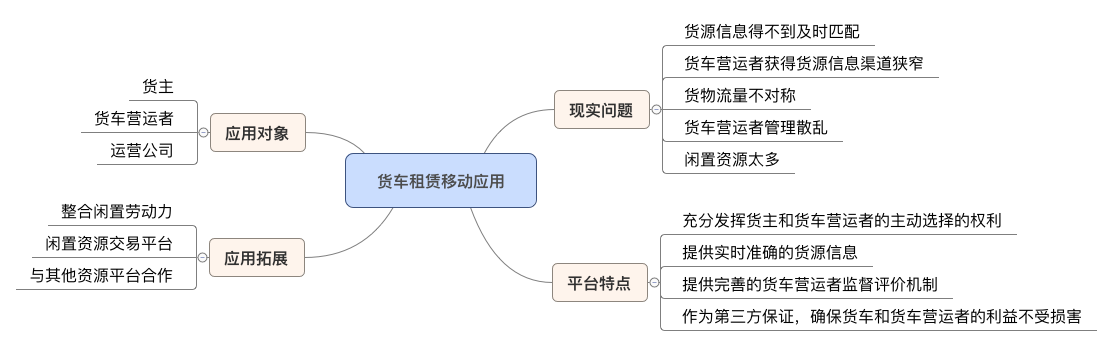
随着经济的不断发展，人民的生活水平不断提高，购买力也随之提高，社会资源被不断消耗，与此同时，消耗剩余的资源不断积累，成为闲置资源。如何将闲置资源进行整合处理，成为一个新的研究课题。

在货运行业中，各类车型可以承担的货物条件不同，不同类型车主的需求得不到满足，同时货源发布者无法找到合适的货车，更多条件下只能选择将就。同时在运输的过程中，存在着返程跑空车的流量不对等情况。这都造成了货车资源利用不高的问题。针对这一现象，本项目设计并开发了一款基于iOS的移动APP。旨在打造一个沟通货车司机与货主的平台，提高货车的利用率，同时满足货主的及时需求。同时规划发展，拓展业务，不断整合小型闲置资源，最终实现中小型闲置资源的整合，为社会作出贡献。

* **创意描述**

通过对货运行业和货运APP的分析，我们决定开发一款名为“国桥货运”的一款货运移动APP。“国桥”是我们的一个目标，我们希望借助我们的应用，可以在全国范围内搭起一座桥梁，将货运司机与货源信息沟通起来。最终实现闲置资源的统一管理。

通过对项目对前期分析，我们得出如下思维导图。我们对主要应用对象是货主、货车营运者和运营公司。我们发现现实生活中存在者货源信息得不到及时匹配等问题，因此，平台将着重结局这些问题，充分发挥货主货货车营运着的主动选择权利，同时提供完善的监督机制，保障货车司机和货主的权利不受侵害等，形成一个信息沟通的平台。将来将应用进行拓展，积极争取与其他平台的合作，最终实现中小型闲置资源的整合，为社会和谐作出贡献。



**图1 项目思维导图**

* **功能简介**

本项目的应用点相对专一，针对货主和货车运营者的行为进行分析，有针对性的对项目对功能进行设计。项目的功能主要有：发布货源、货车智能推荐、新货源提醒、用户行为记忆、偏好记忆、线路导航等。

1. 发布货源：

货主在平台注册后，便可在APP中发布货源信息，货源信息包括：起点、重点、货物类型、货物数量、货物参考照片、货主的联系方式等。货源信息必须尽可能完善，使得货车营运者能在得到货源信息后，可以根据自身货车的特性，对货车承载力进行准确判断，提高工作对效率。

1. 货车智能推荐：

货车运营者在平台注册后，司机会将可以满足运营要求的货车的信息在平台中进行发布，同时要求司机定期更新货车的车况。

因此，在系统中，已经记忆了一些货车的基本信息。当货主发布货源信息后，系统会将货源信息与货车信息、司机所在地等信息进行匹配，并根据匹配结果对货车进行排序，供货主进行主动选择。提高了货主的主动选择性。

1. 新货源提醒：

在货主发布货源信息后，系统会根据货源信息与货车信息、司机所在地等信息，进行推荐评分，不是所有货源都符合某个司机的运营能力。系统会根据评分结果将货源信息进行分批次的推荐，优先推荐给评分高的货车司机。目的就是让货源信息进行准确定位，第一时间送达符合需求的司机客户端，供司机进行选择。一方面使信息得到高效对接，另一方面提高了货主和货车司机的实用体验。

1. 偏好记忆：

系统中包涵货主和货车司机两类人群，系统将对这两类人群的行为进行记忆。

对于货主，系统会根据货主对司机对评分，对货车列表进行更新，同时记录货主的登陆地点，根据常出现地点，结合货车司机常用路线，进行货车列表更新等。

对于货车司机，每一个订单都会被系统记忆，同时每一个订单对信息也会被系统记录，根据这些订单信息，分析货车司机对接单喜好，优先匹配符合喜好的货源信息，进行较为个性化的定制推荐。

1. 路线导航：

系统将结合高德地图SDK，在货车导航的过程中，系统将借助高德地图对行车路线进行导航，选择高德地图是由于系统基于iPhone开发，iPhone都会自带高德地图APP,进行内部调用比较方便。同时高德地图也具有相对较高对准确性，可以为用户提供相对精确对路线信息，使司机在行车过程中少走弯路，节省资源。

# 引言

## 编写目的

项目解决方案说明书编写的目的是说明系统的整体架构以及程序模块的设计考虑，包括概要设计、详细设计、接口设计等内容，为今后软件编程和系统维护提供基础。本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。

## 背景

现在市面上管理调度货车的应用软件非常少，而且调度货车的应用也是一种新鲜的事物。新鲜事物代表着这方面还是空白市场，有着极大的市场等待我们去开发。

某市货车闲置时间较长，没有一个系统的方式来帮助这些货车司机达到更好的运营目的。为了提升本市货车运营的效率以及货车客户的便捷性，希望开发出智能货车移动应用，类似打车软件一样能够高效地帮助货车和其客户。

## 建设目标

货车分为营运货车和非营运货车两种。运营货车就是指为社会提供劳务，发生各种费用结算方式，收取报酬的货运车辆。非营业货车是指党政机关、企事业单位、社会团体自用或仅用于个人及家庭生活，不以直接或间接方式收取运费或租金的货车（包括客货两用车）。我们的项目应该是面向营运货车的，非营运货车进行运货是违法行为，所以在添加商户时需要有证件检查环节。

出租人将自己的租赁信息发布到网站上，比如：

* 4.2米厢式货车出租,货运物流搬家,24小时服务。
* 全城快速响应！24小时周到服务！欢迎来电咨询！
* 北京兄弟联盟租车家政服务。
* 联系电话 xxx。

点击以后会有详情描述、评价信息、预约记录和免费预约按钮，点击免费预约后可以在线填写预约信息，比如日期、时间、联系电话、手机号等。也可以选择直接拨打电话进行线下交流。从交易记录数来看，线下交流占多数。

调研结果表明，目前货车租赁系统十分不完善，以线下交易为主，存在不够灵活、信息闭塞等诸多弊端。线上交易必须使用电脑，不够便捷。即便使用了线上交易，填写的信息也只有交易日期等简单信息，不能填写货物质量、类型，不能自动定位以就近取车，更没有自动推荐车辆、分析路线的算法。因此，一个货车租赁的APP是十分必要的。

## 建设原则

这个部分主要描述了系统需要满足的基本条件：

1. 必须是手机端的APP；
2. 可以定位承租人（APP使用者）的位置，并通过地图显示附近可租赁地点；
3. 应用储存货车的信息足够多，比如货车的载货量、平均车速、适合路段等，方便路线规划算法的实现；
4. 尽量简化租赁流程（减少在APP上点击次数和页面数量），保留主要的信息；
5. 可以缓存用户浏览的信息，离线也可查看；
6. 可以记录用户的偏好。

# 需求分析

## 利益相关者

经过分析我们确定了系统的利益相关者主要包括运营公司、货车运营者和客户，然后对他们的价值分析如下。

### 2.1.1司机：

司机的人脉有限，会遇到接不到货的情况，通过我们的货车租赁应用可以快速的接到单。或者在运过单程以后，为了避免空车返程的情况出现，也可以使用我们的APP接到当地的单。对于司机来说，接单就是利润，我们的应用可以实现司机的快速接单。

### 2.1.2货主：

我们的APP提供给货主的利益主要有：

1. 更方便的下单，寻找并选择司机；
2. 价格透明，降低运货成本；
3. 运货可靠性更高，作为一个互联网平台，信誉可靠。

### 2.1.3平台运营方：

运营方的利益主要包涵以下几点：

1. 货主的消费，运营方有一定分成。
2. 司机的工资可以月结，流动资金有大量的利益。
3. 当APP扩展到一定规模后，广告费用是一笔收入。

## 现有平台分析

通过对利益相关者的分析，确定了他们的价值，然后需要对现有平台进行分析，确定现有平台是否已经满足了货车运营者以及客户的需求。平台对“货车帮”、“运满满”、“物流小秘”和“**沃车港**”进行相关调研，调研结果如下：

### 2.2.1货车帮：

司机感受是这样的：“我最先开始用的是货车帮，没办法，谁让人家宣传力度最大呢！在很多配货站和物流园区都能看到货车帮的海报，就连附近吃饭睡觉的地方也都有，后来上网搜了搜，感觉来头挺大的，就先开始用它。但用起来我却不开心，最大最大的原因是丑！没错…真的很丑（我用的是司机版，但是货主版样式也差不多）。司机版的图标就够丑了，可是界面…简直了。长期用手机的朋友们对手机上的软件审美肯定都有一定的高度，我也绝对相信很多朋友会因为软件的丑美来决定某一个软件是否有必要留在手机上。”

### 2.2.2运满满：

“那个时候我正好跑天津，回来的时候还没找到货，又不想空车返回来，就试了试这软件，首先要说，这个软件在操作上很好用，因为简单。主界面上只有三个选项，这种极简设计非常适合货车司机（货车司机高文化比例很低，操作步骤多或者逻辑性强的话相对麻烦）。总体来说印象还算不错，所以后来偶尔也在用这个软件。”

操作简单、使用方便是这个APP最大的优点，结合司机人群的特点，设计很人性化。

### 2.2.3物流小秘：

环节最完整、并且有人性化的播音。操作比较人性化。

### 2.2.4沃车港：

出现较晚，比较新的APP。界面最美观，简洁、操作简单、功能丰富。

### 2.2.5总结：

根据以上分析，我们对分析进行汇总，如下表：

**表1现有平台分析**

|  |  |  |
| --- | --- | --- |
| 现有平台 | 优点 | 缺点 |
| 货车帮 | 宣传投入多、效果好，网站规模较大 | 界面设计不注重美观，容易产生视觉疲劳 |
| 运满满 | 操作简单、使用方便，设计人性化 | 宣传投入较少，知名度低 |
| 物流小秘 | 环节最完整、并且有人性化的播音 | 宣传投入较少，知名度低 |
| 沃车港 | 界面最美观，简洁、操作简单、功能丰富 | 出生较晚，被其他竞争对手提前占领市场 |

## 系统边界

系统边界定义了系统跟外部的接口，确定了系统的范围，有助于我们确定用户需求。如图2所示。系统内部实现了关于订单、价格评估、附近货车等信息，同时系统将于获取高德地图、获取支付接口，提供进行在线支付功能、同时与公安系统合作，完善监督机制等，这些形成了系统的边界。

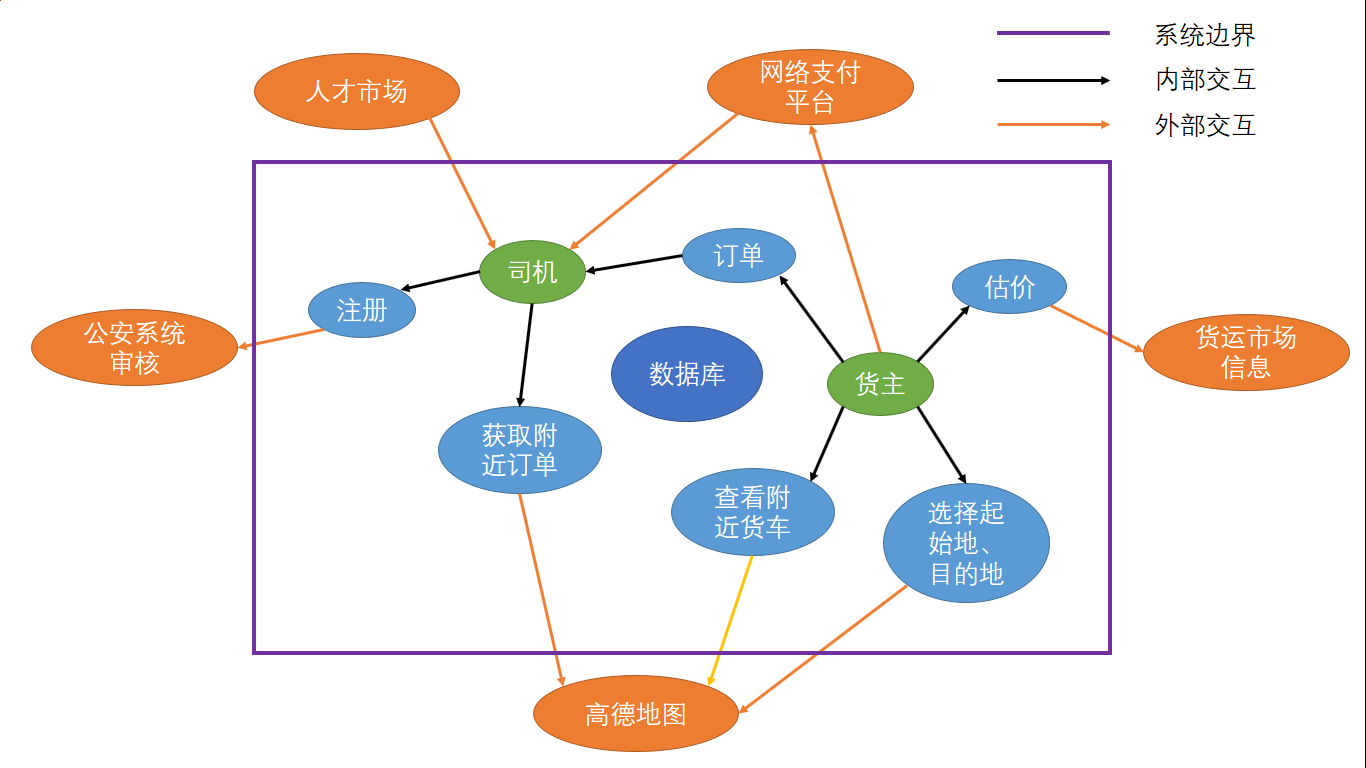


图2 系统边界

## 功能性需求

**表2 功能需求**

|  |  |
| --- | --- |
| 功能模块 | 需求说明 |
| 货源信息管理 | 可以查看、增加、修改、删除货源信息 |
| 个人信息管理 | 可以查看、增加、修改、删除司机或货主的个人信息 |
| 附近货车推荐 | 自动推荐附近的货车 |
| 货车评价管理 | 可以查看、增加、修改、删除对货车的评价信息 |
| 常用路线管理 | 管理经常使用的路线，方便再次使用 |
| 下单、匹配功能 | 货主下单、由司机抢单，并有匹配功能 |
| 在线付款 | 可以使用支付宝、微信支付等把运货款付给司机 |
| 选择目的地 | 可以方便的使用APP在地图上选择目的地坐标，司机可以收到坐标并了解目的地的精确位置 |
| 估价功能 | 在APP中选择了车型、载重等信息，可以自动估算运货价格 |

## 非功能性需求

非功能性需求是指软件产品为满足用户业务需求而必须具有且除功能需求以外的特性，包括系统的性能、可靠性、可维护性、可扩充性和对技术和对业务的适应性等。

下表描述了本系统中的非功能性需求：

**表3 非功能性需求**

|  |  |
| --- | --- |
| 需求名称 | 需求简介 |
| 性能 | 能够以 5 秒的最大响应时间，处理 50 个并发用户对本系统的访问 |
| 可靠性 | 在给定的时间内以及规定的环境条件下，软件系统能完成所要求功能的概率。 |
| 易使用性 | 指本系统功能的简易程度，也包括对系统的输出结果易于理解的程度 |
| 可扩充性 | 指软件系统能方便和容易地增加新功能 |
| 可维护性 | 指在软件系统中发现并纠正一个故障或进行一次更改的简易程度。可维护性取决于理解、更改和测试软件的简易程度。 |

## 可行性分析

可行性分析将从不同侧面出发，对系统对可行性进行判断。

### 2.6.1技术可行性

使用开发工具Xcode来进行APP前端的开发，用到的技术有swift。后台开发使用Node.js。

### 2.6.2资源可行性

* 时间：比赛时间在暑假，团队内所有成员都能把精力集中在比赛上。
* 人员：团队成员都具备充足的项目开发经验与较强的研究学习能力。
* 开发地点：学校提供了实验室供团队开发使用。

### 2.6.3法律可行性

货车分为营运货车和非营运货车两种。

运营货车就是指为社会提供劳务，发生各种费用结算方式，收取报酬的货运车辆。非营业货车：是指党政机关、企事业单位、社会团体自用或仅用于个人及家庭生活，不以直接或间接方式收取运费或租金的货车（包括客货两用车）。

我们的项目应该是面向营运货车的，非营运货车进行运货是违法行为，所以在添加商户时需要有证件检查环节。

### 2.6.4实用性和可拓展性

用户登录app之后，填写货物信息，之后可以在app的“咨询功能”下选定货车的类型，得到然后等待系统分配司机，得到司机的联系方式。司机登录app后填写自己最常用的行车路线之后，系统会根据用户与司机双方的情况完成匹配。司机会得到该订单的详细信息以及用户的详细信息。我们的程序在开发之初就将可扩展性作为一大目标。

### 2.6.5经济可行性

* SWOT分析
* **Strengths:**

1. 货运是一个极其庞大的市场。经过三十多年的发展，我国货运行业发展势头良好，但是信息化水平还有待提高。
2. 现在市面上管理调度货车的应用软件非常少，而且调度货车的应用也是一种新鲜的事物。新鲜事物代表着这方面还是空白市场，有着极大的市场等待我们去开发。

* **Weaknesses:**

1. 互联网解决不了流量不对称的情况。比如上海去的宿州货就是多，但是宿州回上海的货就是少。
2. 货车司机是一个仅次于风投需要靠吃人脉的职业，原因是他的生意绝大部分来自于他的人脉。没有足够的利益难以让他们使用APP。
3. 对于生产厂商、物流公司等他们已经有成熟的找车渠道，难以改变。

* **Opportunities:**

货运APP目前更多的是见到基于整车运输的APP，目前还有一个非常空白的地方，就是是零担运输货运的APP，在这个行业目前还未出现。

* **Threats:**

货运与打车的区别在于，货主并不一定会因为这个平台给的补贴多就使用它，因为，货主与货车司机之间存在人情关系、信任风险等问题，这是货运行业的特点。如何解决诚信、人才、服务质量、支付的问题是一大痛点。

# 概要设计

在本系统的需求分析阶段中，已经对本系统的功能需求做了详细的阐述。本阶段将会在需求分析阶段的基础上对本系统做进一步的概要设计，主要包括本系统的总体设计、操作处理流程设计和关键算法设计等。以上系统模块的设计将结合需求分析阶段的功能需求，把各模块间的关系给建立起来，从而完成整个系统的概要设计需求。另外，在下一阶段的详细设计中，本阶段的概要设计将作为参考，以方便完成整个系统的设计工作。

## 系统设计目标和原则

软件系统架构是一个系统的草图。直接构成系统的抽象组件作为软件架构描述的对象。各个组件之间的连接明确和细致地展现出组件之间是如何进行通讯的。软件架构图在整个项目中起到承上启下的作用，向上他承接业务目标，担负着为完成业务目标而进行大局规划的职责；向下引导技术决策，将面向业务的需求转向面向技术的软件架构设计方案，为后面的项目开发人员提供明确详实的指导和限制。

在制造代码阶段，这些抽象组件被具体为实际的组件，例如某个具体的类或者对象。在面向对象开发中，组件之间的联系通常是采用接口来实现。一般来说，一个好的系统架构应该达到如下目标：

1. 可靠性（Reliable）：软件系统对于用户的商业经营和管理来说非常重要，因此软件系统必须具备可靠性；
2. 安全性（Secure）：软件系统所承担的商业价值极高，同时系统要存储学校、老师、家长等相关个人信息，保护系统的数据安全非常重要；
3. 可伸缩性（Scalable）：当系统的用户使用率以及系统的用户数目增加时，软件系统必须保持合理的性能，这样才能适应用户市场扩展；
4. 可定制化（Customizable）：系统的功能必须依据用户群体的不同和市场需求的变化进行相应改变；
5. 可扩展性（Extensible）：系统的功能随时可能增加，在新技术出现的时候，一个软件系统应当允许导入新技术，从而对现有系统进行功能和性能的扩展；
6. 可维护性（Maintainable）：排除现有的错误并且可以将新的需求及时添加到现有系统中是软件系统维护的两个至关重要的组成部分。一个易于维护的软件系统可以有效地降低软件后期的技术成本投入。

为了完成上述目标，系统架构的设计必须满足一定的设计原则。本系统为了实现上述目标，在应用分层、功能职责、高内聚低耦合等方面进行要求：

1. 分离关注点：将应用分离为不同的功能，这些功能保持尽可能小的重叠。主要的好处是一个功能可以最小化和其他功能的依赖关系。另外，如果一个功能失败了，不会导致其他功能的失败，对于其他功能来说是独立的，使得应用更容易理解和设计，简化复杂系统的管理；
2. 职责单一：每一个组件或者是模块应该只有一个职责或者是功能，功能要内聚；
3. 最小知识原则：一个组件或者是对象不应该知道其他组件或者对象的内部实现细节。

## 系统总体设计

### 架构设计

软件体系结构是构建计算机软件实践的基础。与建筑师设定建筑项目的设计原则和目标，作为绘图员画图的基础一样，一个软件架构师或者系统架构师陈述软件构架以作为满足不同客户需求的实际系统设计方案的基础。

下面将从几个方面介绍本系统的架构设计。

* 整体架构

本系统分为展现层、控制层、业务逻辑层、数据处理层。整体架构如图3所示：

1. 展现层

由于客户端采用iOS开发，所以展现层将借助XML进行页面展示，同时管理员端是基于pc的开发，所以使用html5，css3和javascript的进行开发设计，使得页面适应不同大小显示设备的尺寸，保持用户能在单位屏幕内得到更多的信息。

1. 控制层

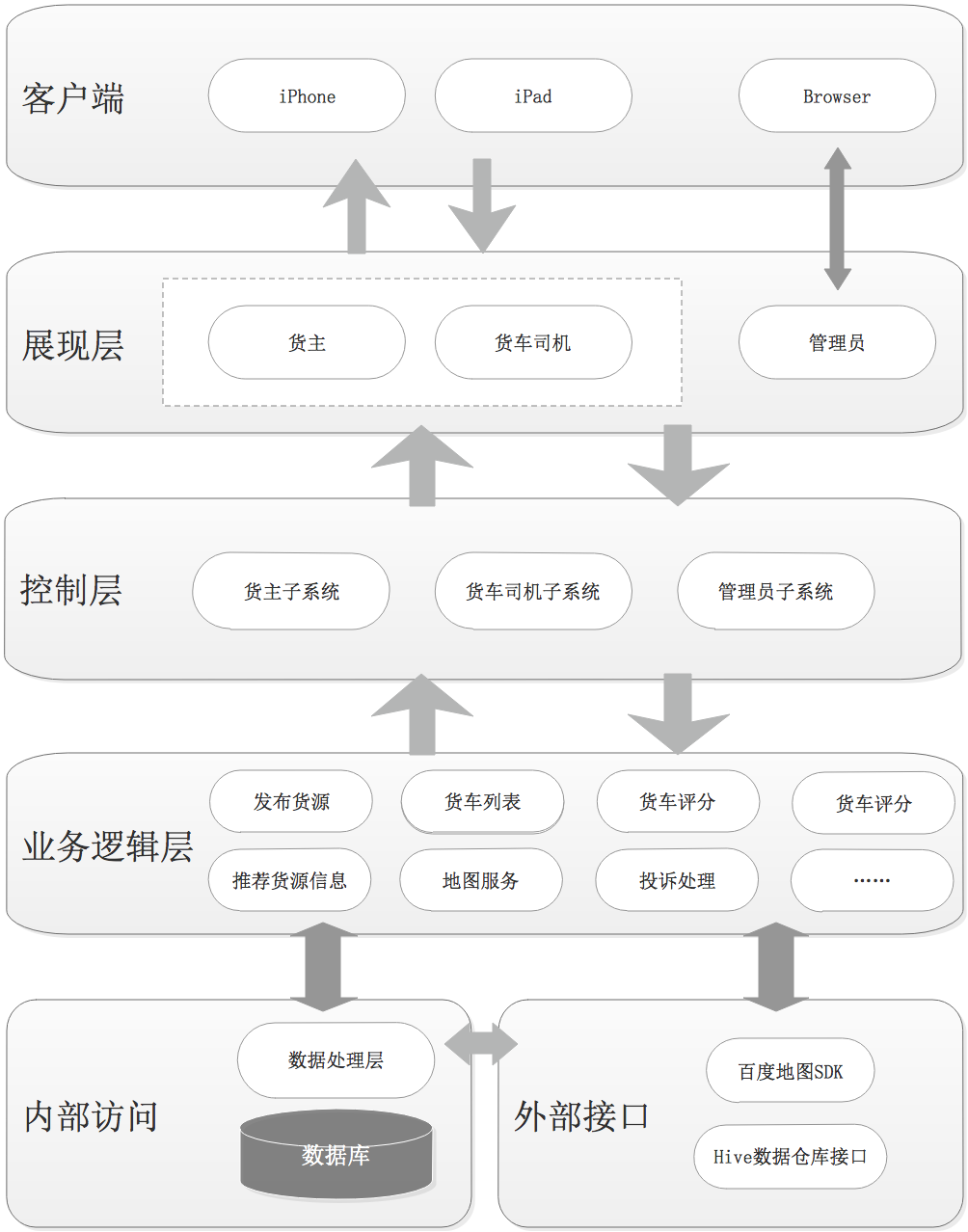
控制层主要接受来自展现层的用户请求，对请求进行分析判断之后，向服务器端发送业务请求，同时接受业务逻辑返回结果，将结果显示在展现层，其中要做多步数据验证，对异常进行判断处理，保证数据的正确性，以及尽可能多的规避异常情况。

1. 业务逻辑层

系统后台采用nodejs进行开发设计，nodejs具有易于开发，周期短，效率高等特点，能保证系统在短时间内尽快上线，同时进行完整的接口设计，以便在项目开发后期，为了应对高性能的要求，可能会做服务器的重构，届时只更新服务器实现方式即可，可以满足高效替换，便于二期开发的目的。

1. 数据处理层

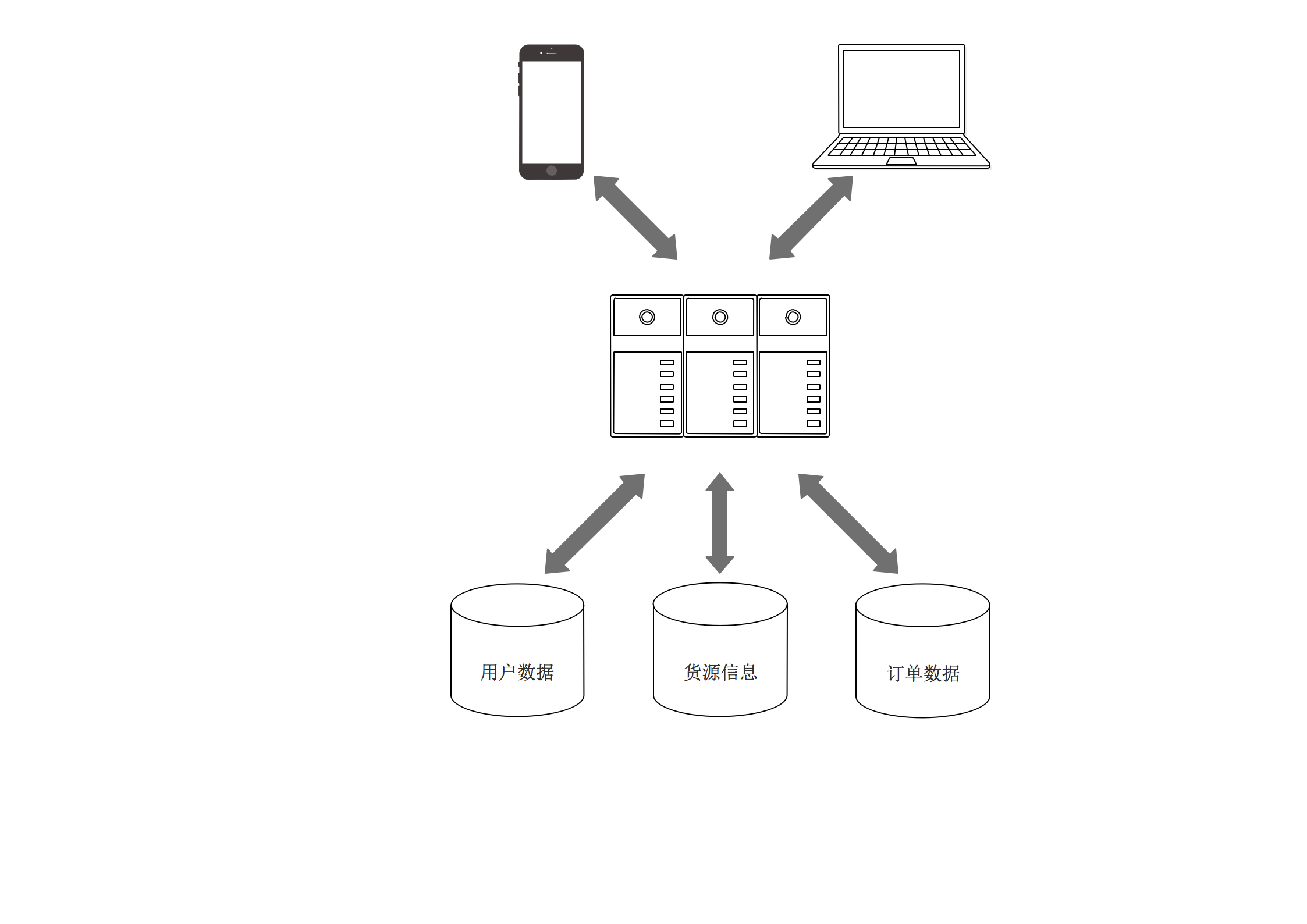
数据持久化方面，系统采用较为常用的mysql数据库进行数据存储，同时才从分布式存储的机制，将数据进行分割管理，保证数据的高效存取，提高数据的存取效率。



**图3 系统架构图**

* 部署架构

系统的部署架构如图4所示。客户端开发完成后，将在APP Store进行发布，服务器端将使用服务器集群，将系统的服务进行分发，采用负载均衡的思想，将服务器的压力降到最小，同时数据库将采用不同的数据库进行数据的分散存储，特定服务器上的数据库处理特定的数据，使得数据得到有效管理。



**图4 部署架构图**

### 运行环境

系统的硬件运行环境如下表所示：

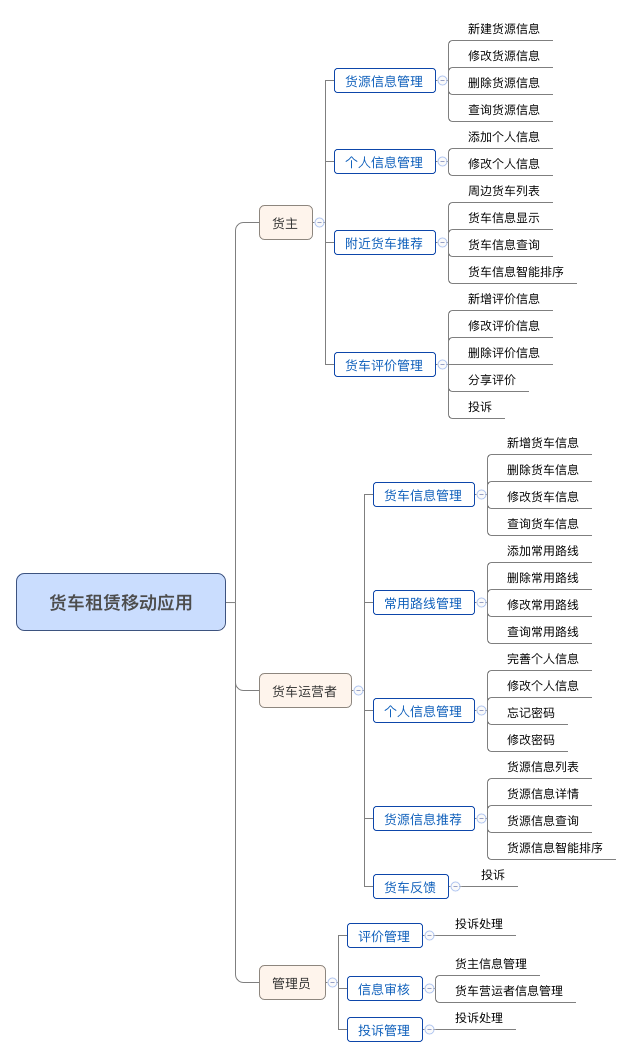
|  |  |
| --- | --- |
| 手机端 | |
| 机型 | iPhone，iPad |
| 操作系统 | iOS7及以上 |
| 业务系统部署环境配置 | |
| CPU | Intel Xeon E5-2683 v3 |
| 内存 | 32GB |
| 硬盘 | 2TB |
| 操作系统 | Windows Server 2008 R2 |
| 数据库 | MySQL 56 |

## 3.3功能模块结构

功能模块是指数据说明、可执行语句等程序元素的集合，它是指单独命名的可通过名字来访问的过程、函数、子程序或宏调用。功能模块化是将程序划分成若干个功能模块，每个功能模块完成了一个子功能，再把这些功能模块总起来组成一个整体。以满足所要求的整个系统的功能。

### 3.3.1总体结构

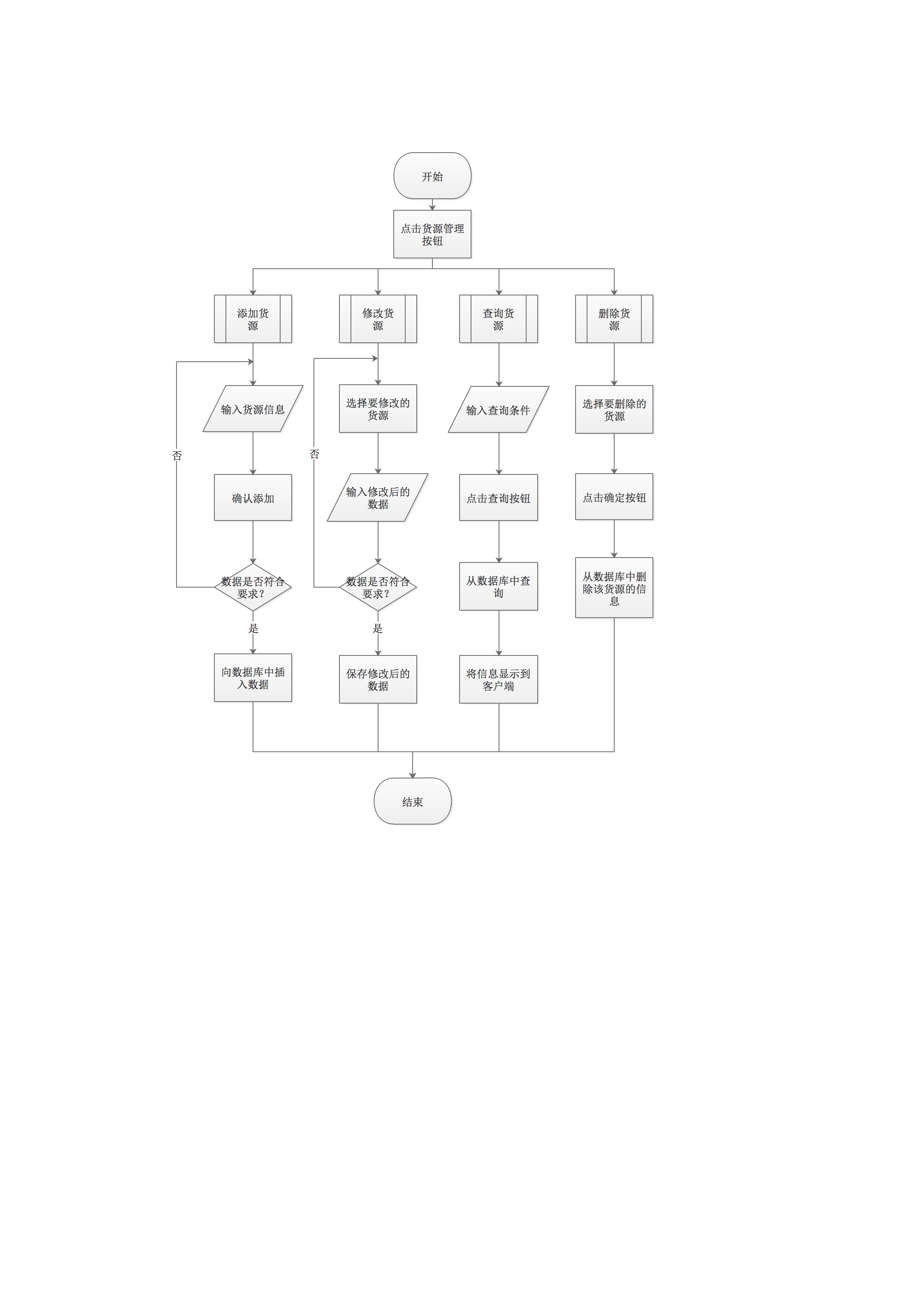
根据前面对于需求对分析，可以得到如下功能分解图，列举了货主、货车运营者和平台运营方对于系统的一些期望，如图所示。



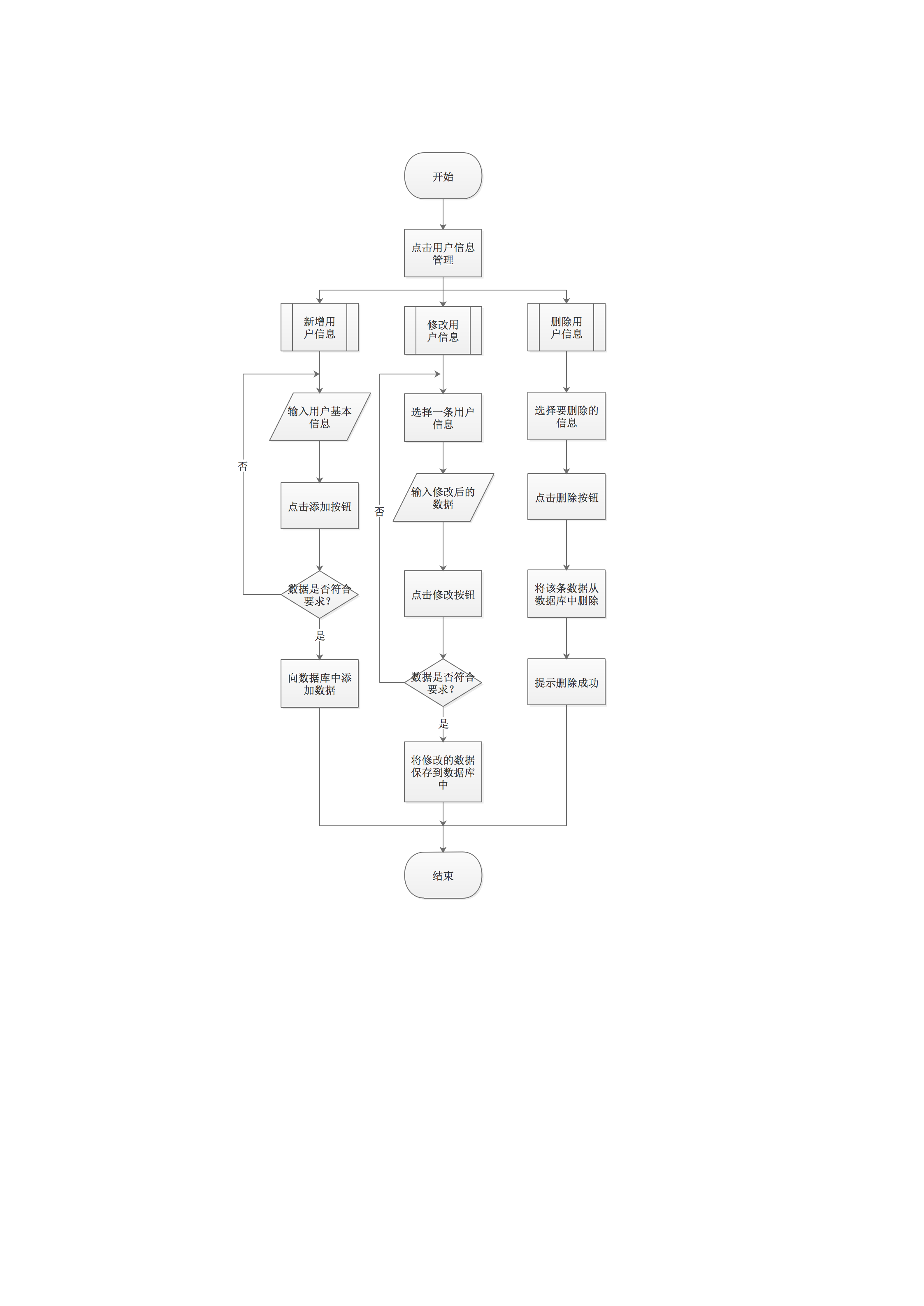
**图15 模块划分图**

### 3.3.2货主子系统

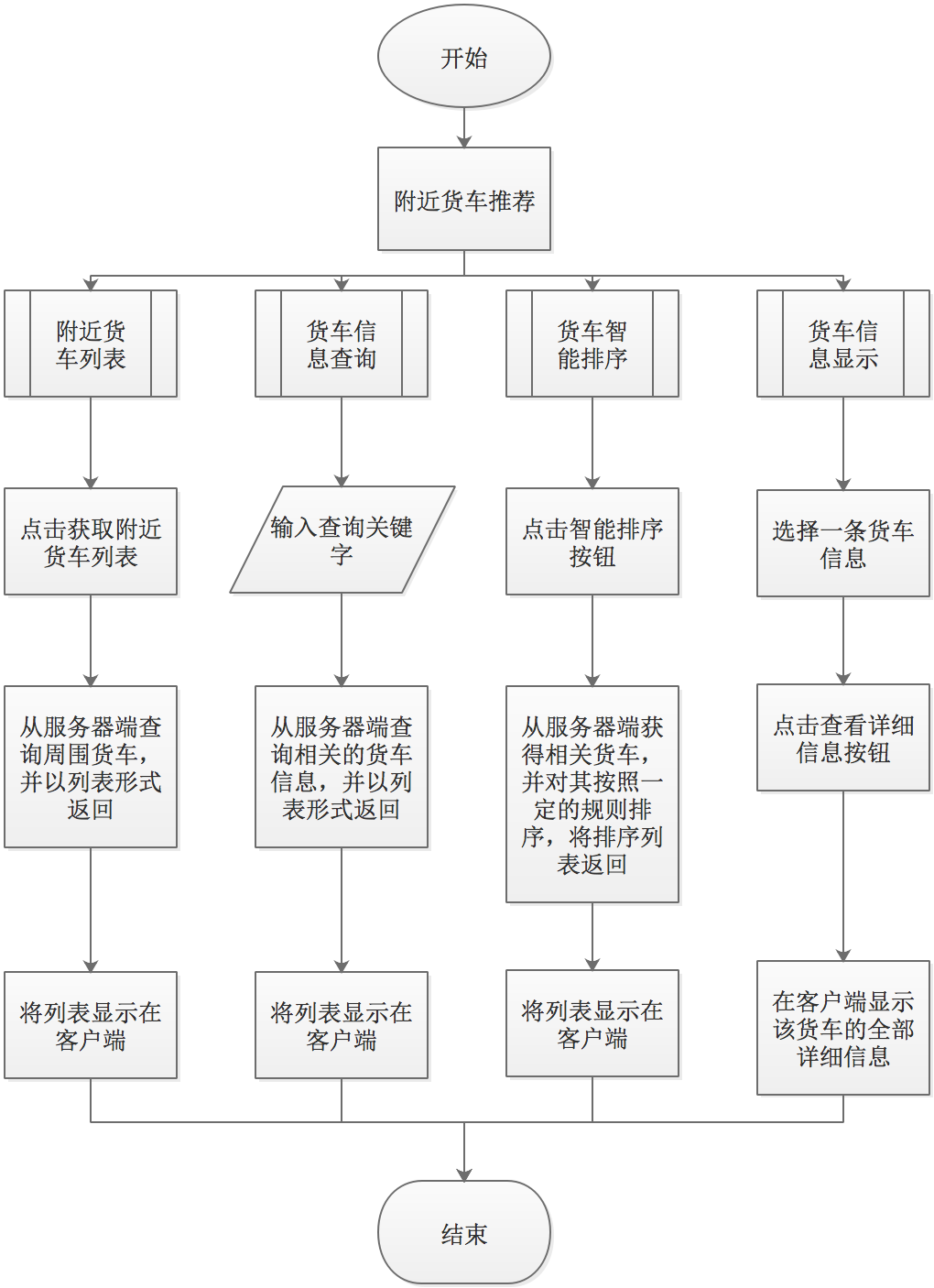
货源信息管理



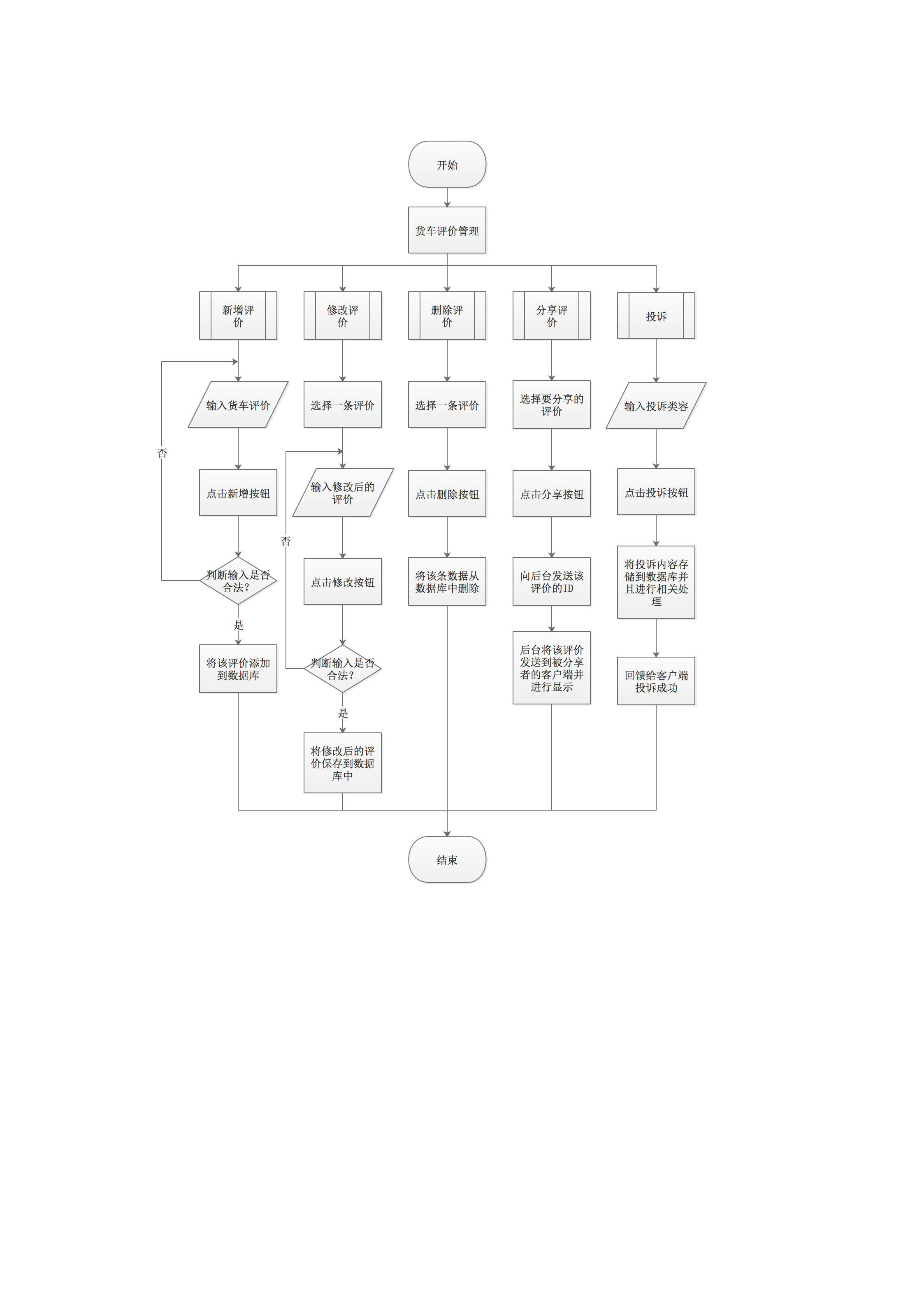
个人信息管理



附近货车推荐

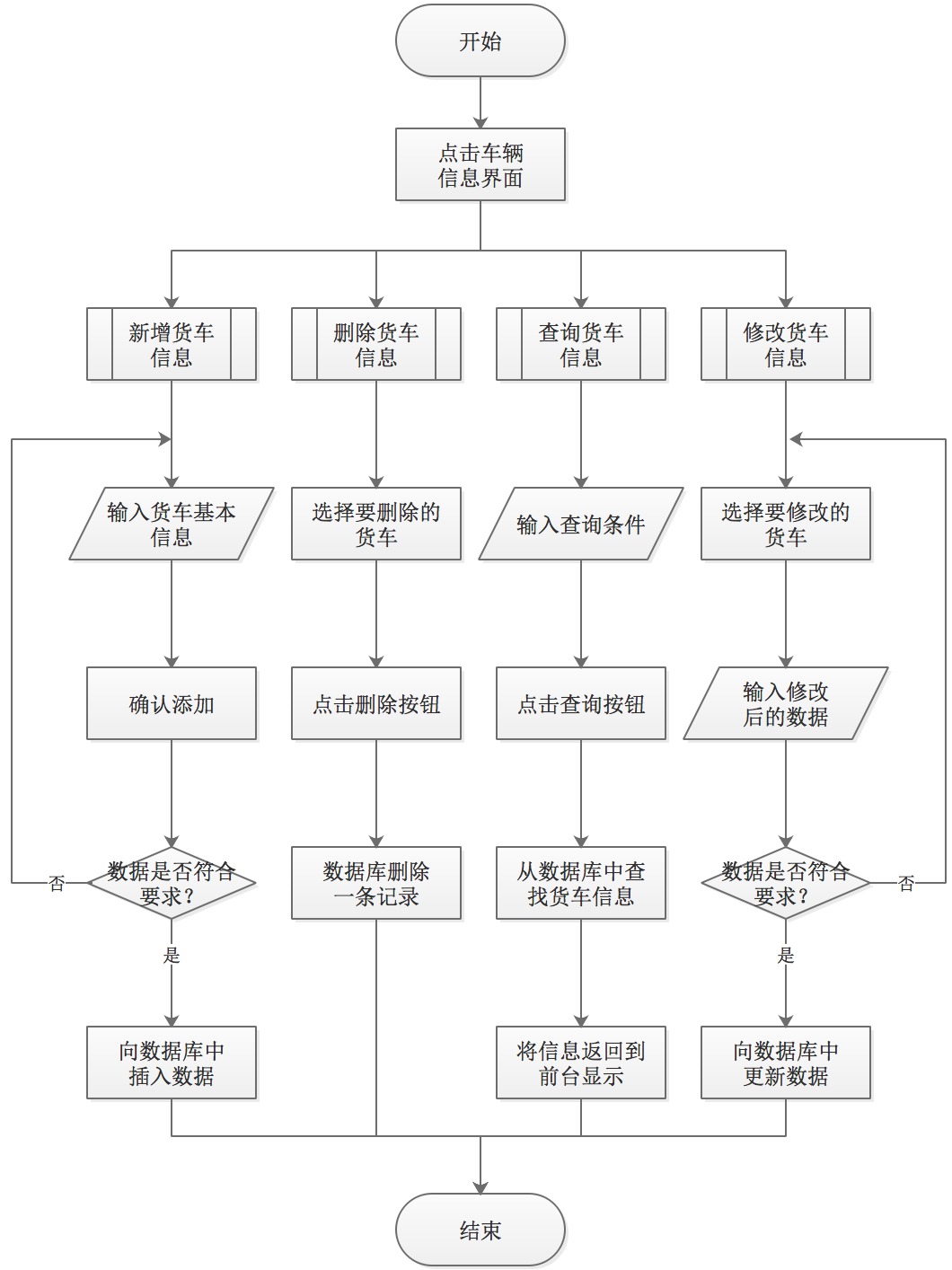


货车评价管理

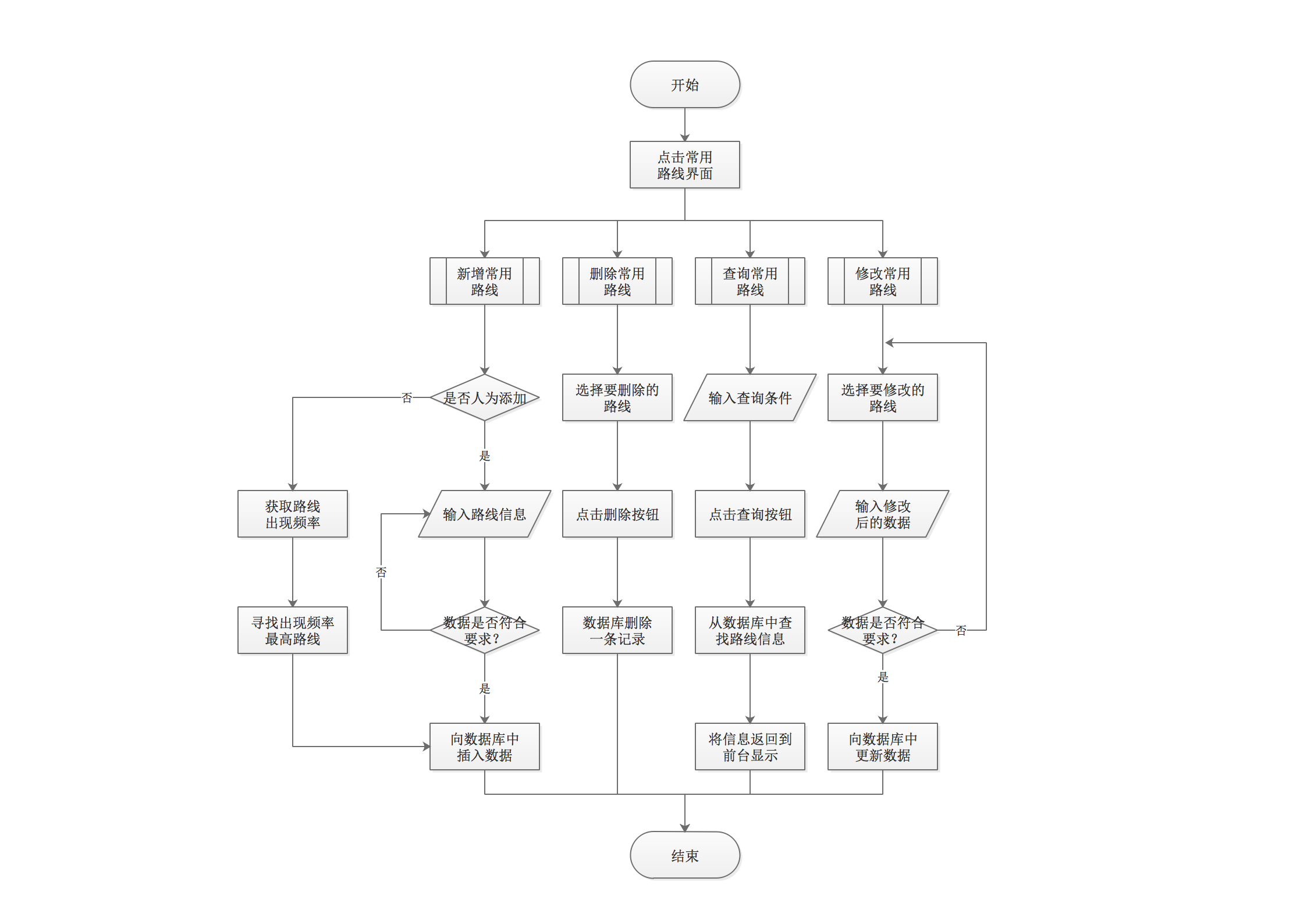


### 3.3.3货车运营者子系统

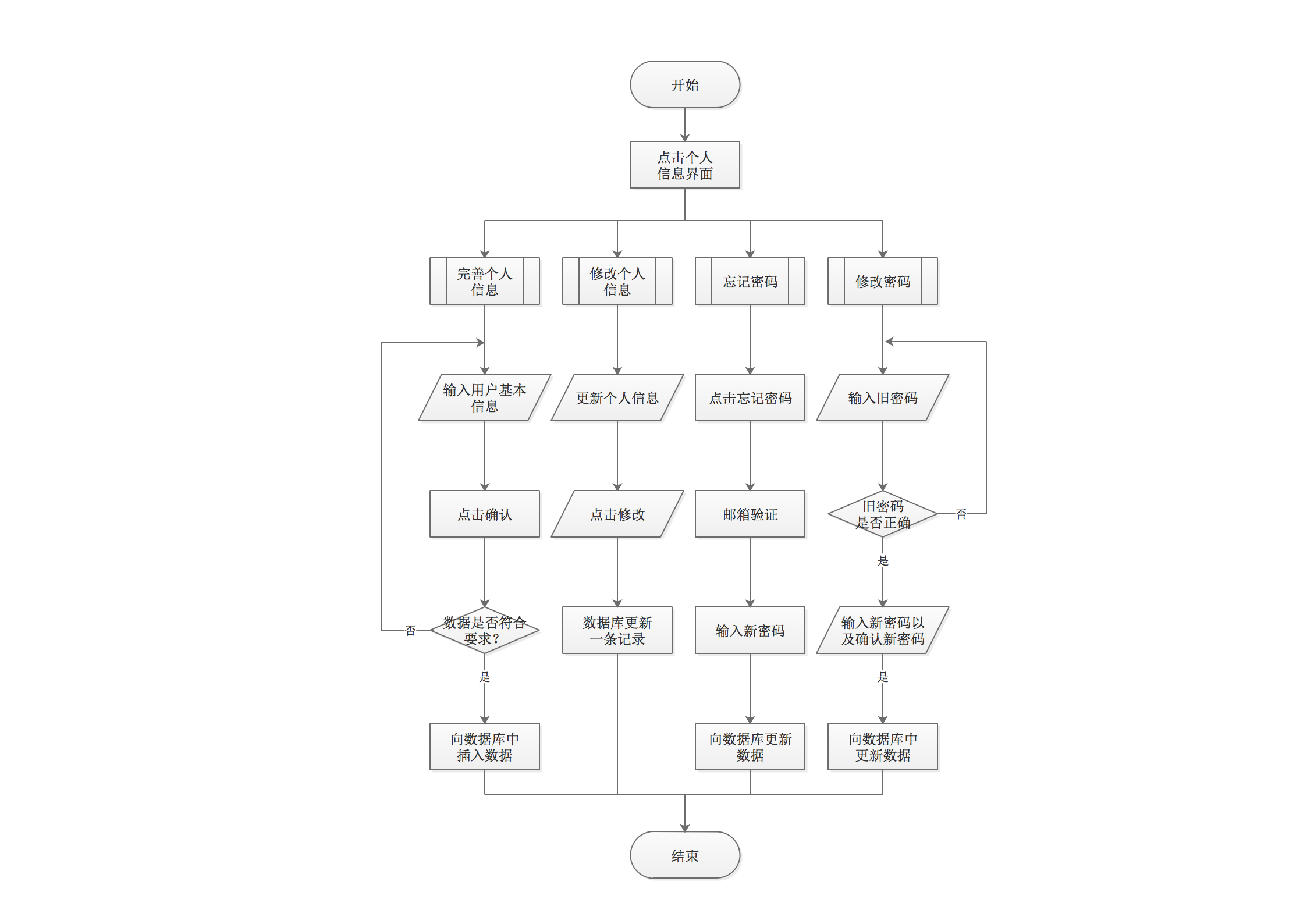
货车信息管理



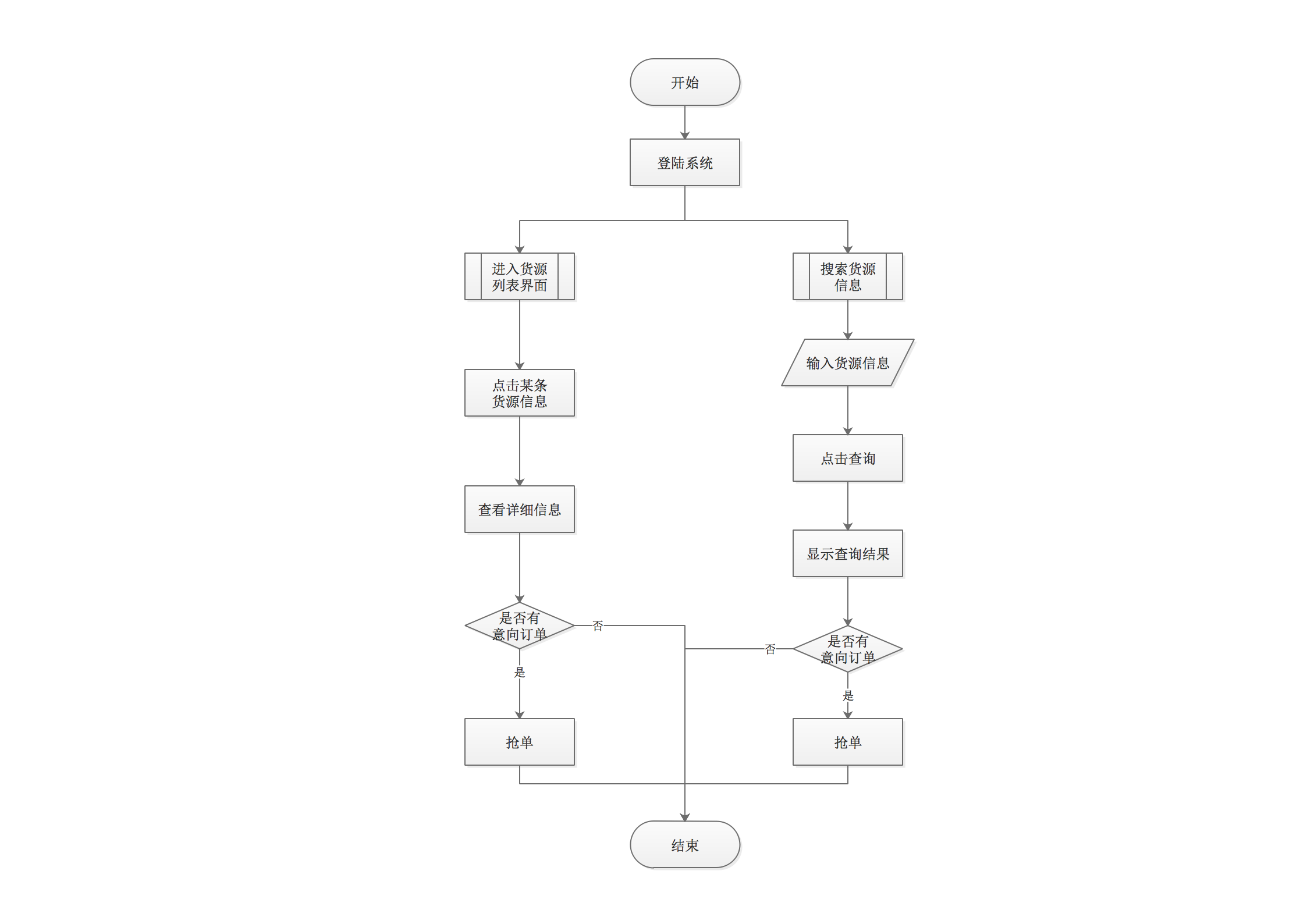
常用路线管理



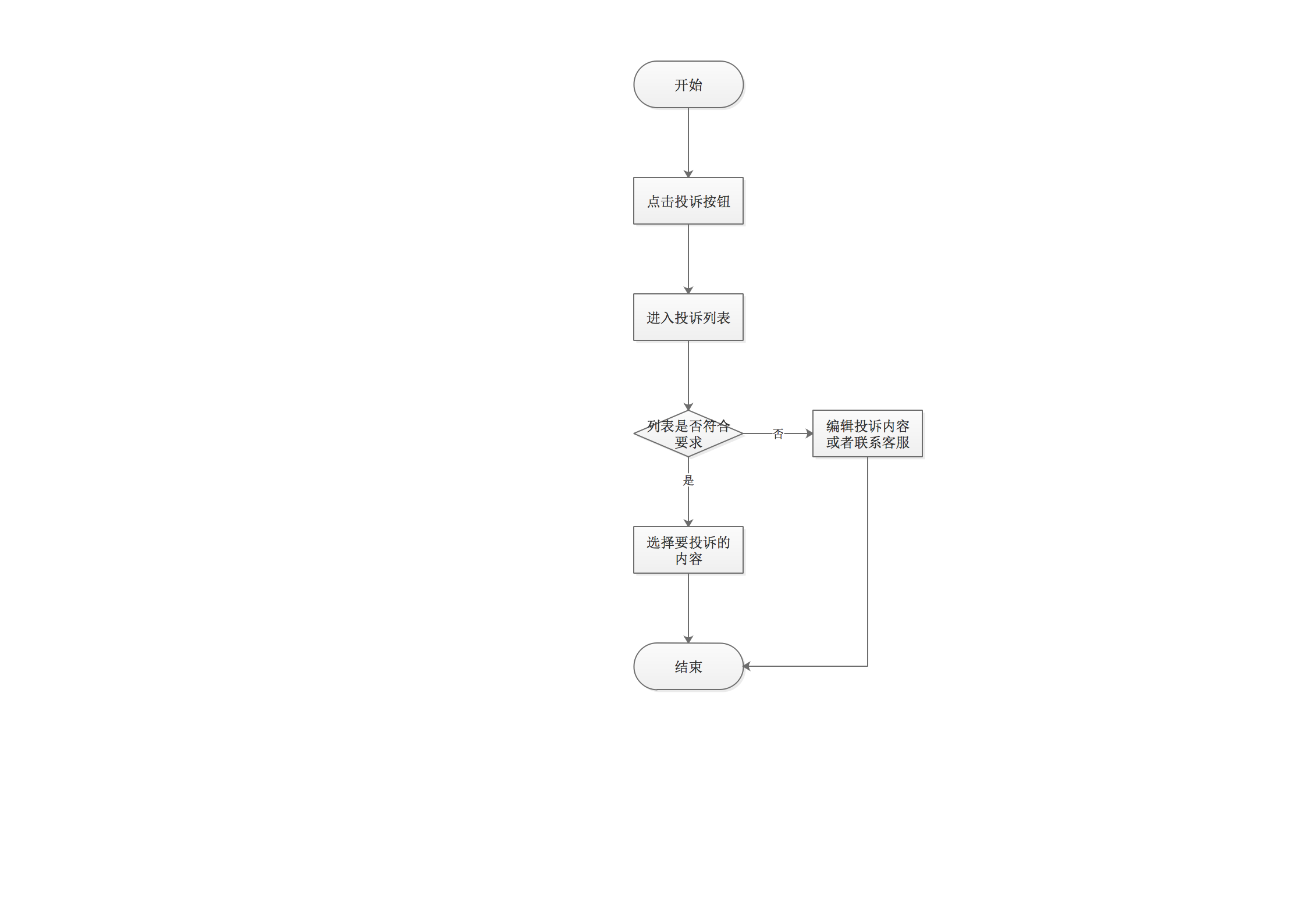
个人信息管理



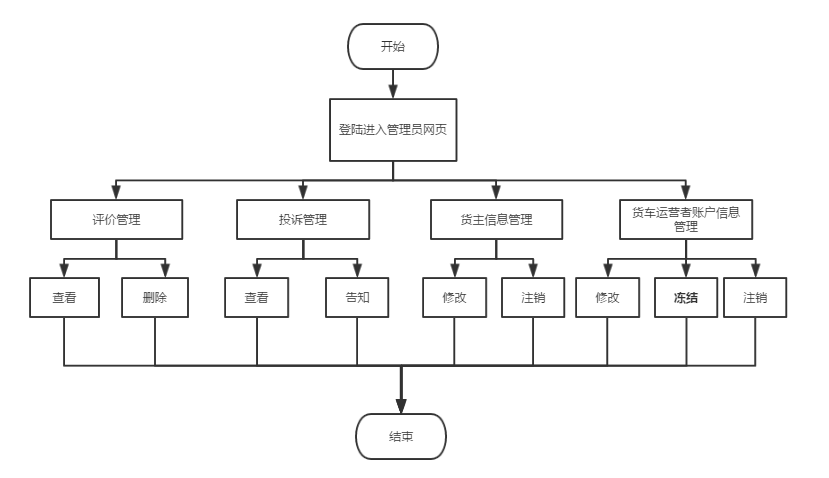
货源信息管理



货车反馈



### 3.3.4管理员子系统



## 数据库设计

### 概述

本系统采用MySQL5.6 数据库，轻量、快速，且功能强大，移植性较好。

系统数据库设计遵循如下原则：

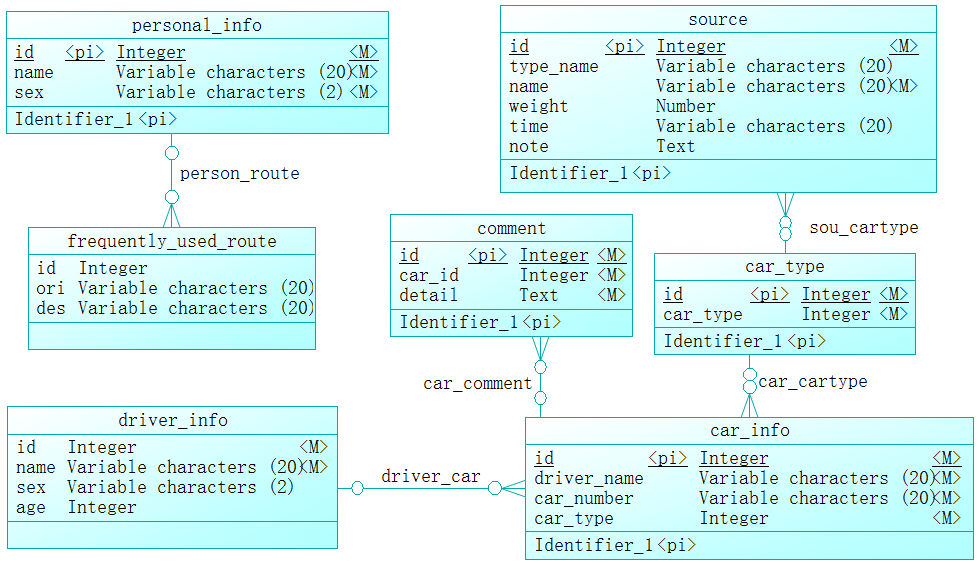
1. 命名规范

|  |  |  |
| --- | --- | --- |
| 命名项 | 命名规范 | 备注 |
| 表名 | 数据表名称必须以有特征含义的单词或缩写组成，中间以用大写分割词语。表名称不能用双引号包含 | 关系表的表名为 “表名 1 + to+ 表名 2”的形式， 或表名1/2 采用缩写的形式 |
| 字段名 | 主键都设为 ID， 其他字段名称必须用字母开头，采用有特征含义的单词或缩写，不能用双引号包含 | 外键名为被引用的键所在表名/缩写+Id |

1. 数据类型

|  |  |  |
| --- | --- | --- |
| 数据类型 | 选择准则 | 备注 |
| 字符型 | 固定长度的字串类型采用 CHAR，长度不固定的字串类型采用 VARCHAR。避免在长度不固定的情况下采用 CHAR 类型 | 如果在数据迁移时出现以上情况，则必须使用 trim()函数截去字串后的空格 |
| 数字型 | 数字型字段尽量采用 INT 类型 | 默认长度为 11 |
| 日期和时间 | 由数据导入或外部应用程序产生的日期时间类型采用 DATETIME 类型 | 无 |

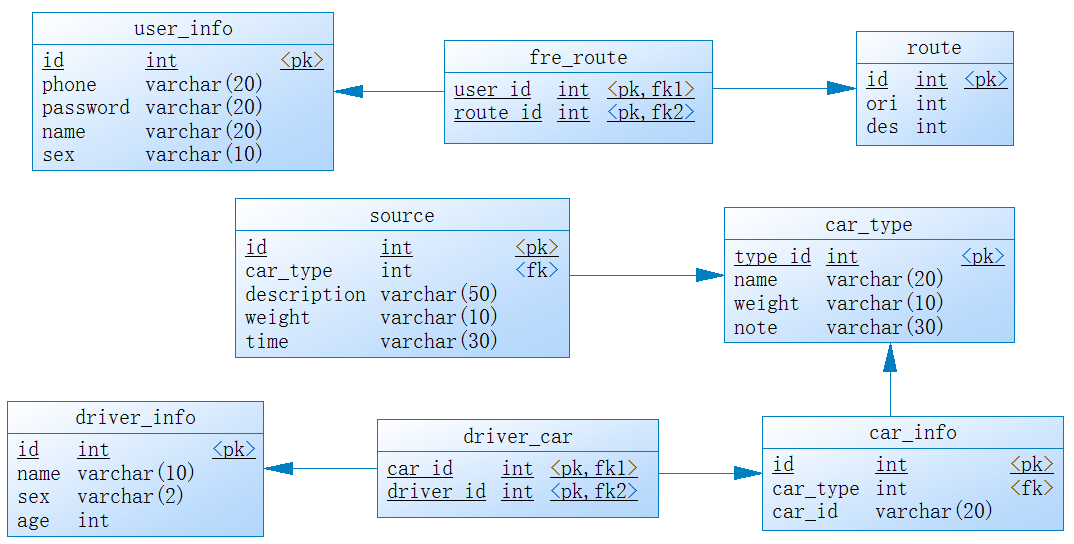
### 逻辑结构设计



**图21 系统逻辑结构设计图**

|  |  |
| --- | --- |
| Personal\_info | 用户个人信息，包括姓名、性别等 |
| Frequently\_used\_route | 常用路线信息，包括起点、终点 |
| Source | 货源信息，包括需要的车辆类型、重量、时间、备注信息 |
| Car\_type | 车辆的类型信息，比如小型车、中型车 |
| Car\_info | 车辆信息，包括司机姓名、车牌号等 |
| Comment | 评论信息 |
| Driver\_info | 司机信息 |

### 物理结构设计



**图22 系统物理结构图**

表信息

|  |  |  |  |
| --- | --- | --- | --- |
| user\_info 用户基本信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | Primary key | 用户唯一标识符 |
| phone | Varchar(20) | Not null | 手机号码，用作用户登录名 |
| password | Varchar(20) | Not null | 登录密码 |
| name | Varchar(20) |  | 用户姓名 |
| sex | Varchar(10) |  | 用户性别 |

|  |  |  |  |
| --- | --- | --- | --- |
| fre\_route 用户常用路线表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| user\_id | int | primary key,  foreign key | 用户标识符 |
| route\_id | int | primary key,  foreign key | 路线标识符 |

|  |  |  |  |
| --- | --- | --- | --- |
| route 路线表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | primary key | 路线标识符 |
| ori | varchar(30) | not null | 路线起点 |
| des | varchar(30) | not null | 路线终点 |

|  |  |  |  |
| --- | --- | --- | --- |
| car\_type 车辆类型信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| type\_id | int | primary key | 类型标识符 |
| name | varchar(20) | not null | 类型名 |
| weight | varchar(10) |  | 最大载重 |
| note | varchar(30) |  | 备注信息 |

|  |  |  |  |
| --- | --- | --- | --- |
| driver\_info 司机信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | primary key | 司机标识符 |
| name | varchar(10) | not null | 司机姓名 |
| sex | varchar(2) |  | 司机性别 |
| age | int |  | 司机年龄 |

|  |  |  |  |
| --- | --- | --- | --- |
| car\_info 车辆信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | primary key | 车辆标识符 |
| car\_type | int | foreign key | 车辆类型 |
| car\_id | varchar(20) | not null | 车牌号 |

|  |  |  |  |
| --- | --- | --- | --- |
| driver\_car 司机车辆对应表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| driver\_id | int | primary key,  foreign key | 司机id |
| car\_id | int | primary key,  foreign key | 车辆id |

|  |  |  |  |
| --- | --- | --- | --- |
| source 货源信息表 | | | |
| 字段名 | 字段类型 | 约束条件 | 字段含义说明 |
| id | int | primary key | 货源标识符 |
| car\_type | int | foreign key | 所需车辆类型 |
| description | varchar(50) |  | 货源描述 |
| weight | varchar(10) |  | 重量 |
| time | varchar(30) |  | 时间 |

### 数据结构与程序的关系

服务器程序在对数据进行展示的时候，需对数据库数据结构，也就是数据表进行查询：在数据展示的过程中都需要对数据库中的所有表，进行联合查询。

物理数据结构主要用于各模块之间函数的信息传递。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。出错信息将送入显示模块中。

## 关键技术介绍

### 3.4.1 视技术

视

景

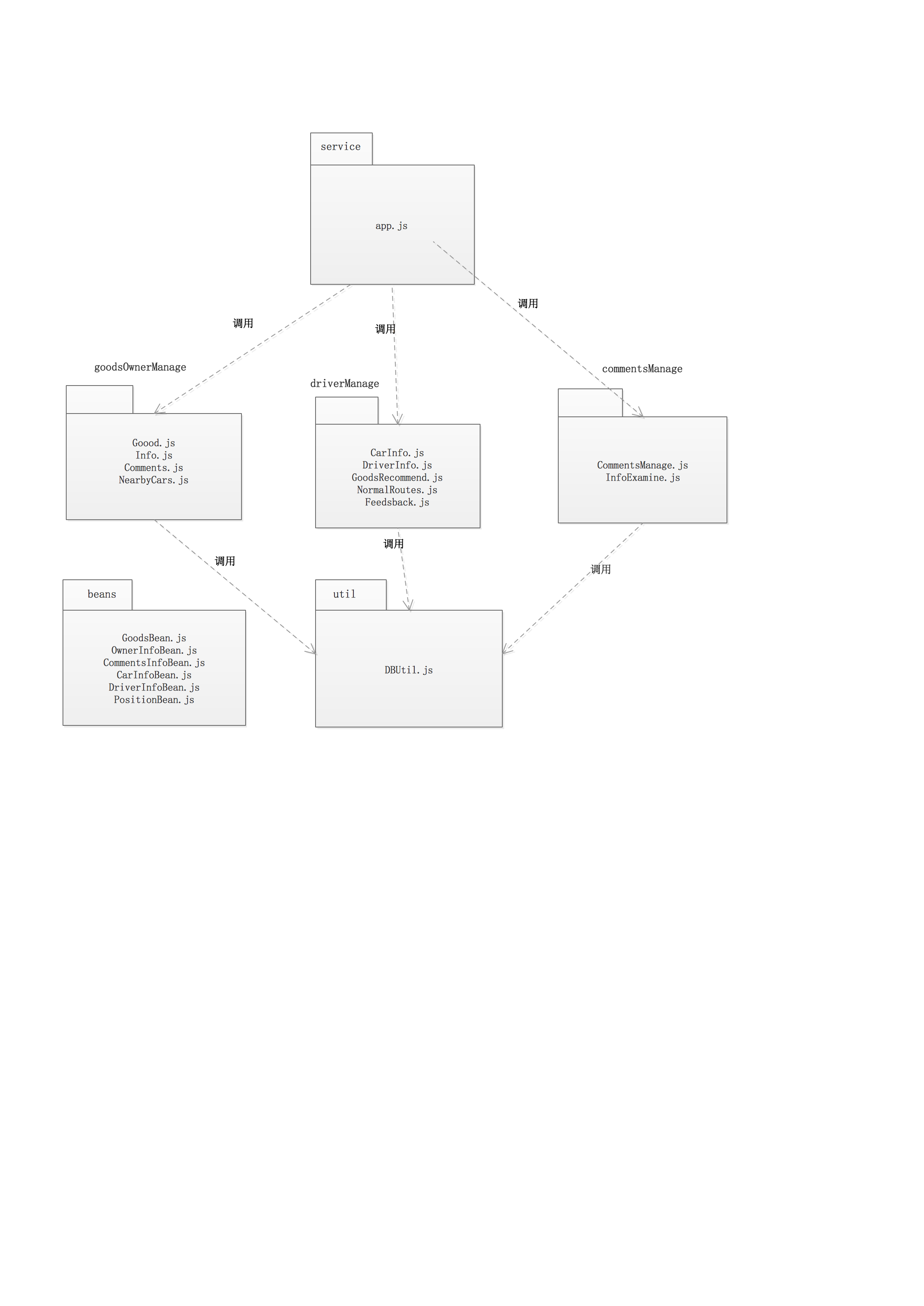
# 详细设计

该部分是基于概要设计编写的详细设计说明书。详细描述了本项目的框架、模块划分、模块结构和模块设计。项目的开发基于此说明书进行。基于此项目主要难点集中于无人机实时视频流的渲染以及用户头部动作的分析，所以我们的重点放在了类的详细设计上。

## 包设计

本系统中包含了大量的类， 因此也会存在大量的事物，为了能更加有效地进行整合、 生产宏观模型，就需要对其分组，就是本系统中的包设计。

下图是本系统中的包设计、 包间的依赖关系、 包中类设计：



**图29 项目包图**

**表5 包设计说明**

|  |  |
| --- | --- |
| 包名 | 设计说明 |
| GoodsOwnerManage | 包含货主业务逻辑处理类 |
| DriverManage | 包含运货司机业务逻辑类。 |
| commentsManage | 包含评论业务逻辑类 |
| beans | 包含bean对象，对数据进行分装 |
|  |  |
| service | 包含服务器相关类 |

## 类设计

本系统共有 5 个包，25个类。本小节以包为单位，对每个包所含有的类及类与类间的调用关系进行说明。

1. GoodsOwnerManage：包含货主业务逻辑处理类

|  |
| --- |
| goodsOwnerManage相关类说明 |
| 类名 | 设计说明 |
| Goods.js | 处理货源的增删改查 |
| Info.js | 处理货主信息的管理 |
| NearbuCar.js | 附近车辆管理 |
| Comments.js | 货主评论管理 |

1. DriverManage：包含运货司机业务逻辑类

|  |
| --- |
| DriverManage相关说明 |
| 类名 | 设计说明 |
| CarInfo.js | 处理货车信息的增删改查 |
| DriverInfo.js | 处理司机信息的增删该查 |
| GoodsRecomend.js | 向司机推荐合适的货源 |
| NormalRoutes.js | 管理司机的常用路线 |
| FeedBack.js | 管理货主对司机的反馈 |

1. commentsManage：包含评论业务逻辑类

|  |  |
| --- | --- |
| CommentsManage相关说明 |  |
| 类名 | 设计说明 |
| CommentsMan.js | 用于管理员处理评论 |
| InfoExamine.js | 管理员处理投诉反馈信息 |

1. beans：包含bean对象，对数据进行分装

|  |  |
| --- | --- |
| beans |  |
| 类名 | 设计说明 |
| GoodsBean | 封装与货物有关的属性 |
| OwnerInfoBean | 封装与货主信息有关的属性 |
| CommentsInfoBean.js | 封装与评论有关的属性 |
| CarInfoBean | 封装与货车信息有关的属性 |
| DriverInfoBean | 封装与司机有关的属性 |
| PositionBean | 封装与位置有关的属性 |

1. service：包含服务器相关类
2. util：包含工具类，如对数据库进行封装 ：
3. Goods.js

该类主要负责货主货源方面的操作，具体包括货主添加货源，删除货源，修改货源，查询货源。

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名称 | 参数 | 返回值 | 备注 |
| addGoods | Goods | Boolean | 添加该货主货源 |
| deleteGoods | ID | Boolean | 删除该货源 |
| updateGoods | Goods | Boolean | 更新原来而货源 |
| queryGoods | String | list<Goods> | 查询符合条件的货源 |

重要函数说明：

1. NearbuCar.js

该类主要负责向货主推荐合适的车辆，具体包括：附近货车列表，货车查询，货车智能排序，货车信息显示。；

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名称 | 参数 | 返回值 | 备注 |
| getNearCars | Position | List<Car> | 得到附近货车信息 |
| queryCars | String | List<Car> | 查询符合条件的货车 |
| sortCars | List<Cars> | List<Car> | 对货车列表智能排序 |
| ShowCar | Null | Car | 将该条货车信息显示 |

1. GoodsRecommends.js

该类主要完成向司机推荐货源的操作，主要包括：货源信息列表，货源信息查询，货源信息显示，货源信息智能排序。

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名称 | 参数 | 返回值 | 备注 |
| getGoodsList | Position | List<Goods> | 得到货源信息列表 |
| queryGoosd | String | List<Goods> | 查询符合要求的货源 |
| sortGoods | List<Goods> | List<Goods> | 对货源信息进行智能排序 |
| showGood | Null | Goods | 显示货源信息 |

1. NormalRoutes.js

该类主要对货车司机常用货车路线进行管理，具体包括：添加常用路线，删除常用路线，修改常用路线。

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名称 | 参数 | 返回值 | 备注 |
| addRoutes | Routes | Boolean | 添加常用路线 |
| updateRoutes | Routes | Boolean | 更新常用路线 |
| deleteRoutes | ID | Boolean | 删除常用路线 |

1. DBUtil.js

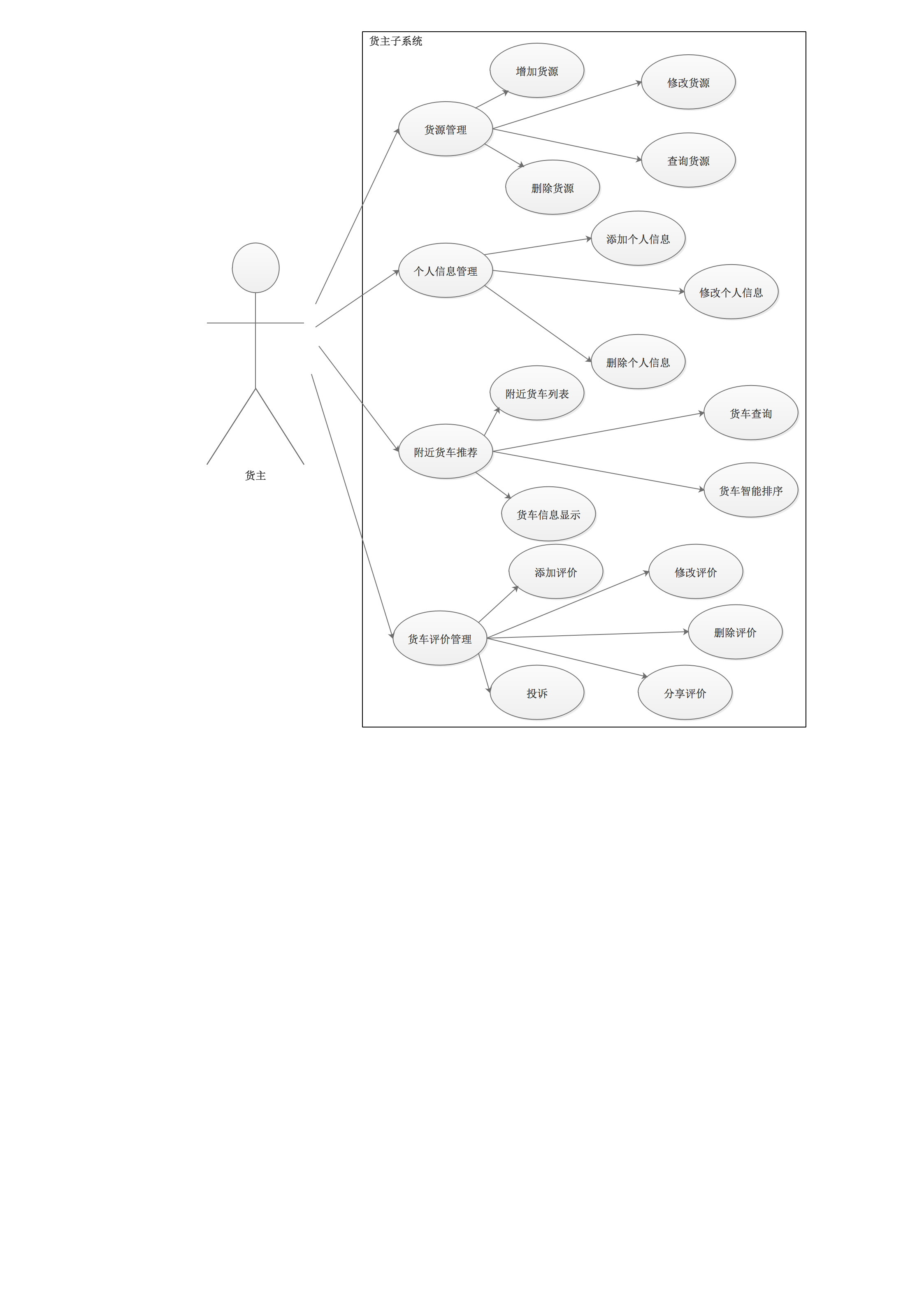
|  |  |  |  |
| --- | --- | --- | --- |
| 方法名称 | 参数 | 返回值 | 备注 |
| save | Object | Boolean | 保存对象到数据库中 |
| update | Object | Boolean | 更新数据库中的对象 |
| query | String | List<Object> | 查询符合要求的对象 |
| delete | Object | Boolean | 从数据库中删除对象 |

## 模块设计

本上。

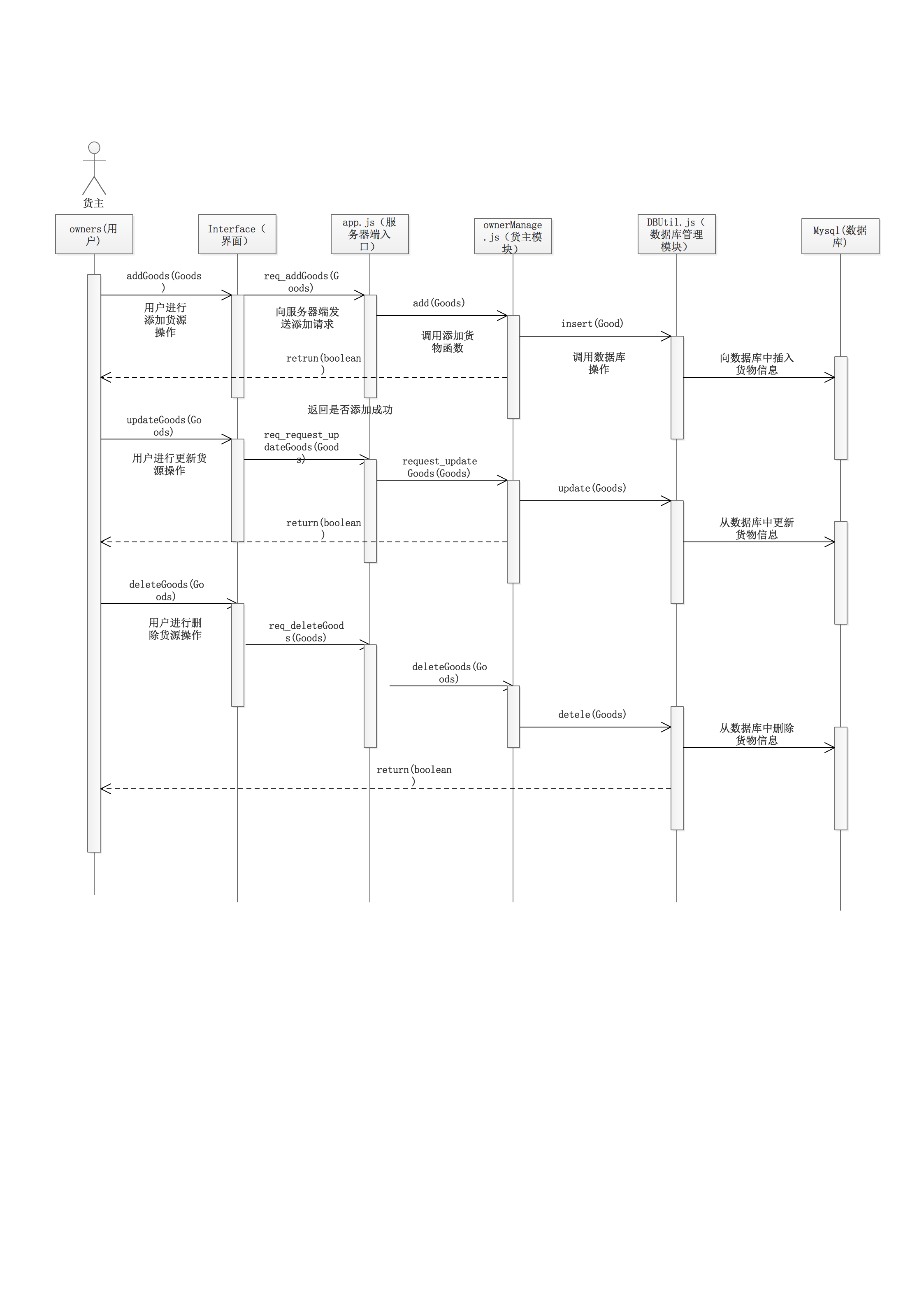
本模块主要是对货主所使用的功能实现。主要包括四大模块：货源信息管理，个人信息管理，附近货车推荐，货车评价管理。

货主子系统



### 4.3.1货源信息管理

货主进入“货主管理”模块，使用货源信息管理功能，可以添加自己运货的需求，输入货物种类，货物出发地，货物目的地，运货时间，期望价格等等基本信息。货源信息成功添加之后，运货司机变可以了解到货主的运货需求。货主还可以修改已发布的信息，删除发布信息，查询发布的信息等等。



1. 添加货源信息

货主填写完货物重量，货物类型，运货时间，出发地，目的地之后，点击提交。App调用addGoods(Goods)方法向服务器发送添加货物的请求。服务器收到该请求的路由“/goods/add”之后传递给相应的路由并且进行处理，将该条信息添加到数据库中，并且向客户端返回添加的结果，成功或失败。

1. 修改货源信息

货主首先选择要修改的信息，修改完要修改的属性之后，向服务器端发送“/goods/update”请求，服务端接受该请求并传递给相应的路由，调用更新该信息的方法updateGoods(Goods)更新该条数据。

1. 查询货源信息

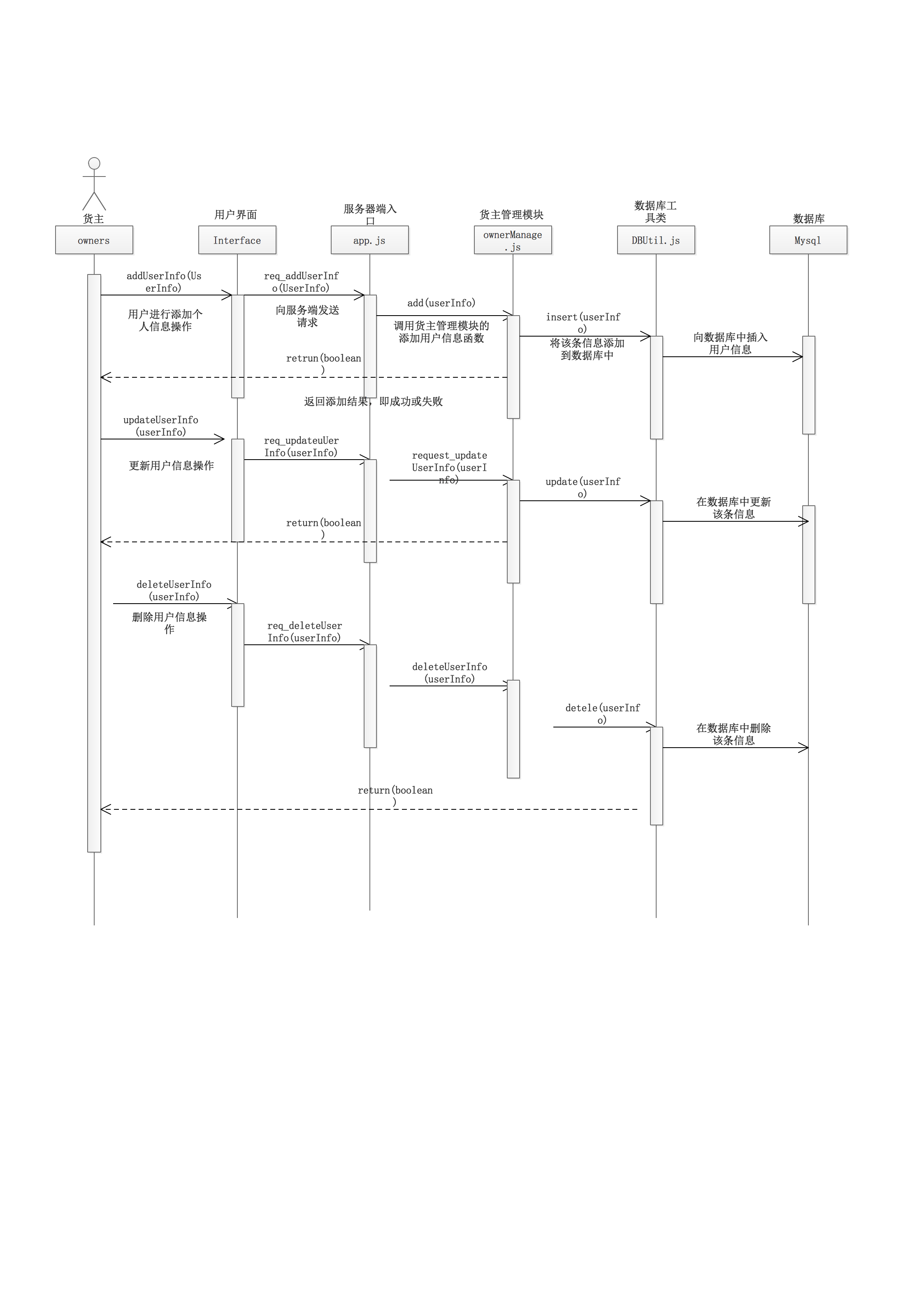
货主输入查询的关键字，如“6吨、海淀区”，向服务端发送“/goods/query”请求，服务端接受请求之后将该请求传递给相应的路由，该路由中调用查询货源的方法，queryGoods(String)方法，查询符合要求的货源，并将该货源以列表的形式返回给客户端并且显示。

1. 删除货源信息

货主选择要删除的货源，获得该货源的ID，向服务端发送“/goods/delete”请求，并且传递给服务端货源ID，服务端获得该请求并且将该请求传递给相关的路由，该路由调用deleteGoods(String)方法从数据库中删除该条记录。并且将删除结果返回给客户端并且显示。

### 4.3.2个人信息管理

货主进入“个人信息管理”模块，管理维护自己的基本信息。可以添加货主的信息，比如姓名，性别，联系方式等等。可以添加多个货主的信息，适应实际的运货需求。货主可以修改基本信息，删除基本信息。



1. 添加个人信息

货主输入姓名，性别，联系方式的个人信息之后，点击“添加”按钮，向服务端发送“/info/add”请求，服务端接受该请求之后，将该请求传递给相应的路由，该路由调用addInfo(Info)方法，将该用户的个人信息添加到数据库中，并向客户端返回添加结果。

1. 修改个人信心

货主选择要修改的信息，点击“修改”按钮，此时向服务端发送“/info/update”请求，服务端接受到该请求之后，将该请求传递给相应的路由，该路由调用修改方法updateInfo(Info)方法，将该修改后的个人信息保存到数据库中，并且将修改的结果返回给客户端。

1. 删除个人信息

用户选择要删除的个人信息，获得该信息的ID，点击删除按钮，向服务端发送“/info/delete”请求，服务端接受该请求并且将该请求传递给相应的路由，该路由调用deleteInfo(String)方法，将该用户的信息从数据库中删除。

### 4.3.3附近货车推荐

货主进入“附近货车推荐”模块，帮助用户快速找到自己需要的货车。

1. 附近货车列表

货主可以获得附近货车的信息，以列表的形式展现给用户。列出用户附近的货车车辆，具体信息包括：车型，位置，载重，车牌号等等。用户点击其中一条即可查看详细信息。

1. 货车信息查询

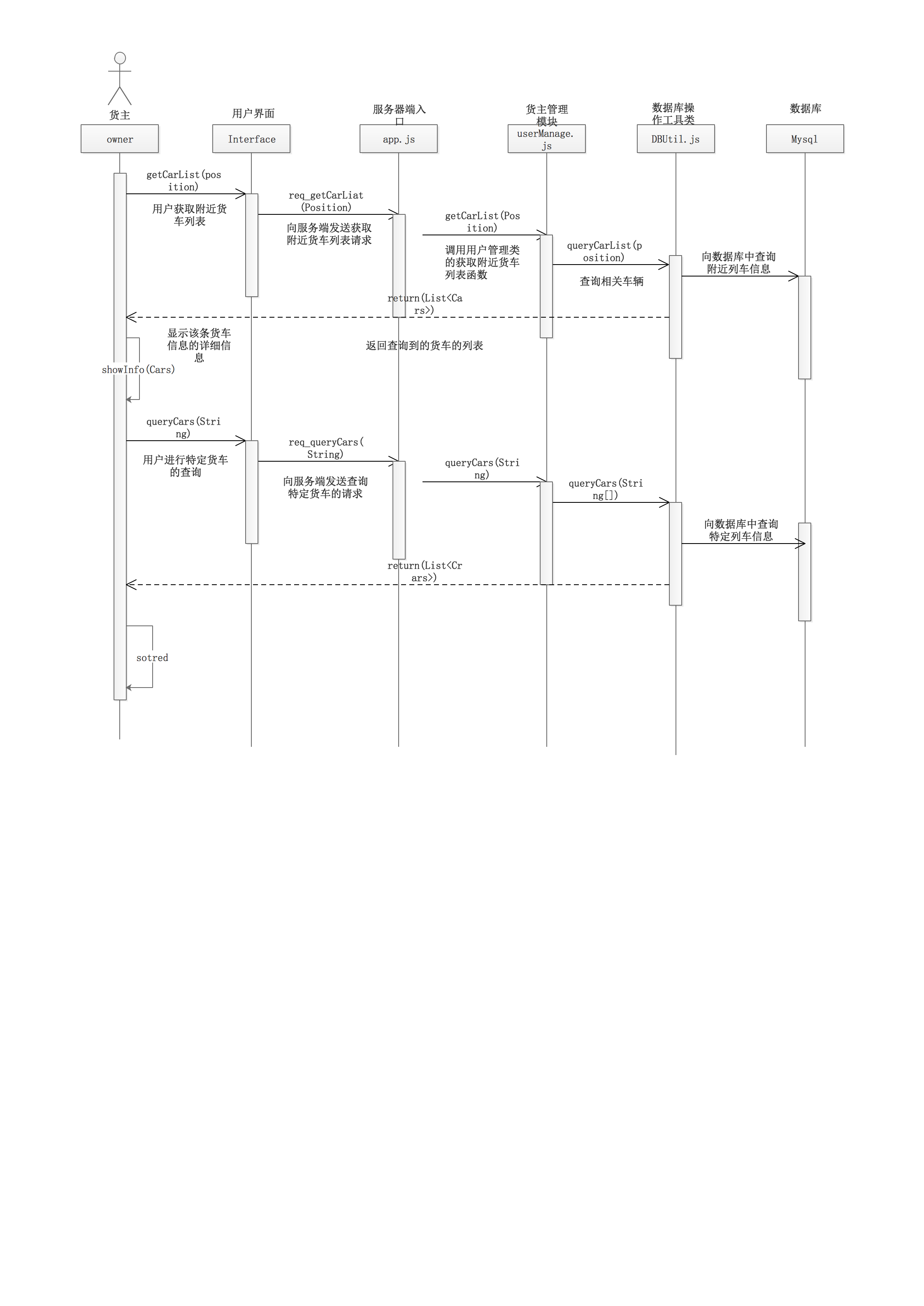
货主可以查询自己需要的货车，输入查询关键字，比如“面包车、海淀区”，app就会给用货主查询海淀区的面包车，并将所有的查询结果以列表的形式逐渐返回给用户，用户下拉，便可获得更多。如果用户点击一条，便显示这条货车的基本信息。

1. 货车智能排序

用户点击“智能排序”，便可以获得自己需要的货车列表，比如适合自己的货物重量的车型，适合自己运送货物种类的货车条件等等。

1. 货车信息显示

用户点击“货车信息显示”，便可以显示该条货车的所有详细信息，比如车牌号，司机姓名，货车类型，货车现在位置等等。



1. 附近货车列表

货主点击“获取附近货车”按钮，获取货主此时的位置坐标，此时向服务端发送“/nearby/getCars”请求，服务端接受该请求之后将该请求传递给相应的路由，该路由调用getNearbyCars（Position）方法，查询出附近的列车信息，以列表的形式返回给客户端并且显示。

1. 货车信息查询

货主输入要查询的条件，输入关键字，比如“小面包，海淀区”，此时向服务端发出“/nearby/queryCars”请求，服务端接受该请求之后将其传递给相关的路由，该路由调用queryCar(String)方法，查询符合要求的货车信息，比如“在海淀区运货的小货车”，然后将该信息已列表的形式返回给客户端并显示。

1. 货车智能排序

货主点击“智能排序”按钮，此时app调用我们根据实际情况设计的排序算法，对获取的货车进行排序，列出最适合该货主的货车列表，将排序后的列表重新显示在客户端。

1. 货车信息显示

用户点击某一条货车信息，此时跳转到一个新的页面，该页面专门用来展示该货车的信息，会将该货车的全部详细信息按格式显示，例如，货车号，货车司机姓名，货车类型，货车载重，货车位置等等。

### 4.3.4货车评价管理

货主进入“评价管理”模块。货主可以对评价信心进行管理。

1. 添加评价

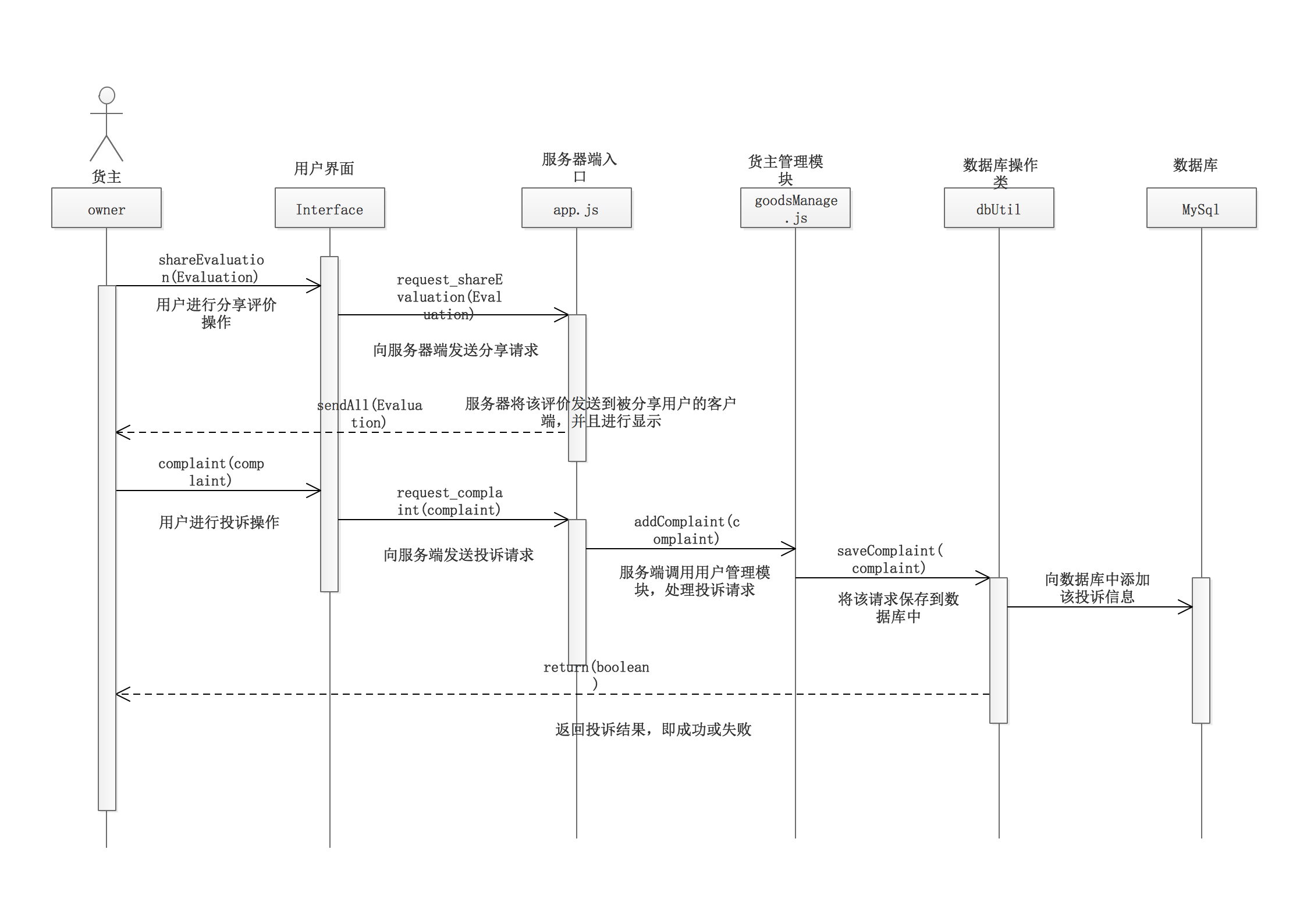
货主在完成一次交易之后，可以对此次运货提供评价。比如对货车司机的服务评分，对运货进行描述，是否守时，价格是否合理等等。这些评价可以让app参考对该司机进行评分。

1. 删除评价
2. 修改评价
3. 分享评价

货主可以将自己的评价分享给自己的好友，其作用相当于给好友推荐司机，如果此评价是正向的，相当于将该好友推荐该司机。如果评价负向，相当于给好友提示，不要雇佣该司机。

1. 投诉

货主可以对该次运货行为进行投诉，投诉将发给后台，后台将对该评价进行核实。如果投诉属实，将对该司机账号进行处理。



1. 添加评价

货主输入对该次运货的司机评价信息，点击提交。此时向服务端发送“/comments/add”的请求，服务端接受请求之后将其转发给相应的路由，该路由调用addComments(comments)方法，将该评价信息保存在数据库中，并且将添加结果返回给客户端。

1. 修改评价

货主完成对评价信息的修改，点击保存信息，此时向服务器发送“/comments/update”请求，服务端接受该请求之后，将该请求传递给相应的路由，该路由调用函数updateComments(Comments)将该信息从数据库中更新。并将更新结果返回给客户端。

1. 删除评价

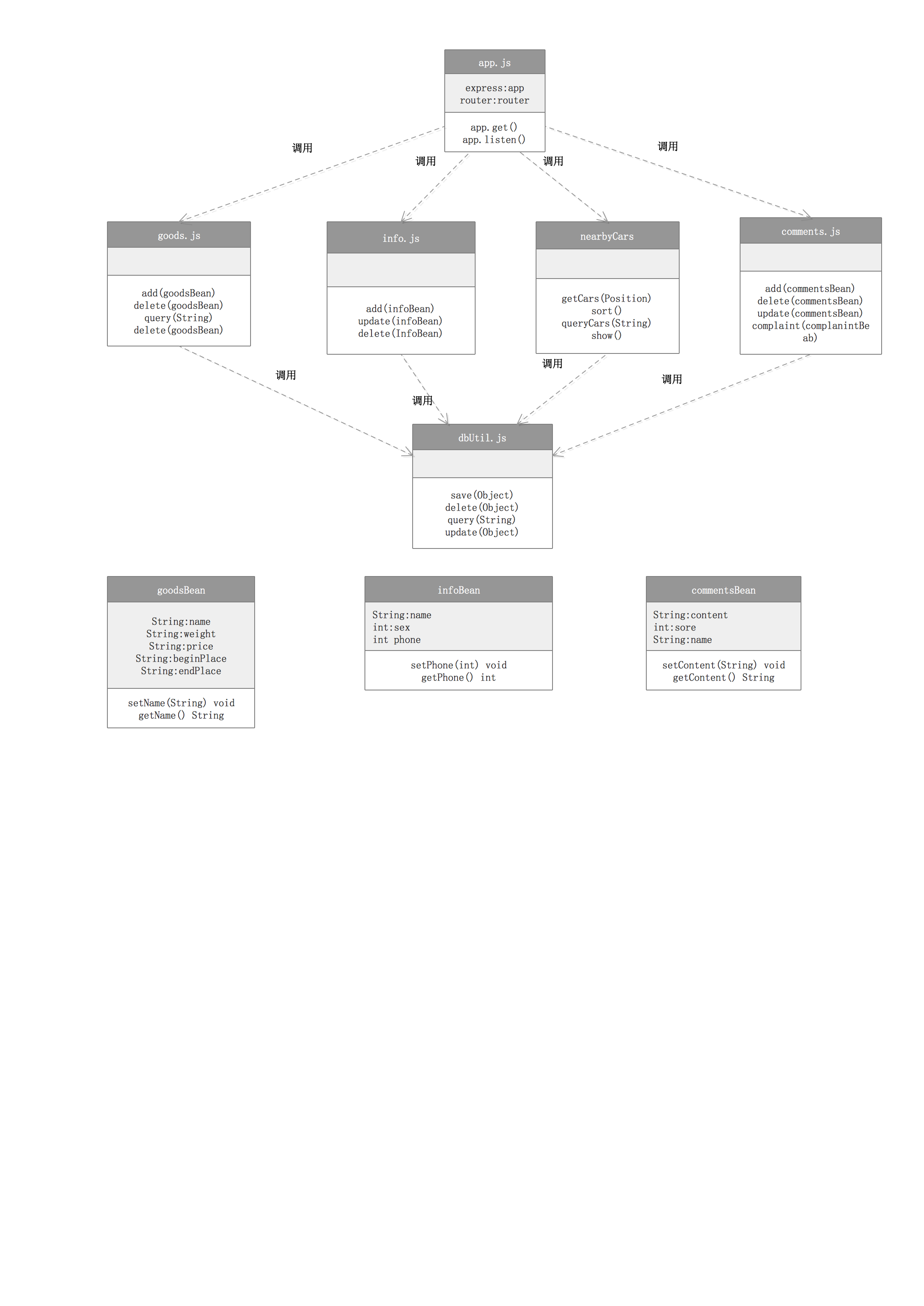
货主选择一条自己的评价，此时获取该条评价的ID，此时向服务器端发送“/comments/delete”请求，服务器端接受到该请求之后，将该请求转发给相应的路由，该路由调用函数deleteComments(String)删除该条评价，并将删除结果返回给客户端并显示。

1. 分享评价

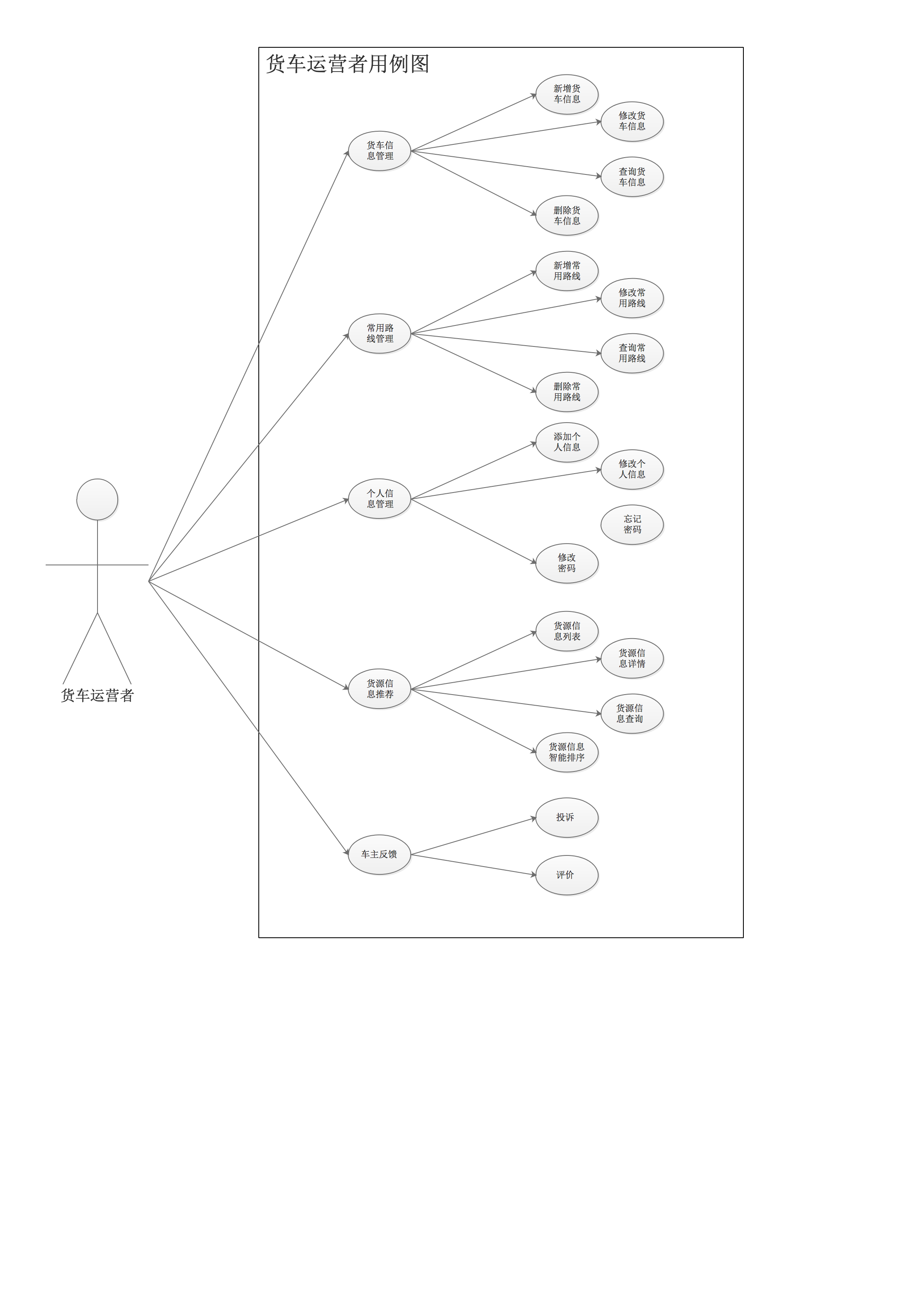
货主选择一条评价，app获取该条评价的ID，此时向服务器端发送“/comments/share”请求，服务器端接收到该请求之后将其转发给相应的路由，该路由调用函数shareComments(ID)方法，首先从数据库中查询出该条评论的所有类容，并将该条信息发送给需要被分享的对象。

1. 投诉

货主点击“投诉”按钮，输入投诉的基本信息，此时向服务器端发送“/comments/complaint”请求，服务器端接受该请求后将其转发给相应的路由，该路由调用函数complaint将该投诉信息保存在数据库中等待审核人员审理，若审核结果属实，则对投诉的司机账号进行处理，比如降低其信誉度等级等待。



货车运营者子系统



## 关键类说明

# 接口设计

## 接口设计概述

接口是提供给其他模块或者系统使用的一种约定或者规范。因此接口必须要保证足够的稳定性和易用性。下表描述了设计接口的基本要求：

|  |  |
| --- | --- |
| 需求名称 | 需求描述 |
| 稳定性 | 接口必须相对稳定，否则将导致接口的使用者和提供者为了适应新接口而不断修改接口的实现，可能重复进行无用功，严重时影响整个软件开发进度。 |
| 易用性 | 采用面向对象思想。 |
| 规范性 | 主要是接口设计的代码规范，这是最基本的要求。同时考虑 C 接口命名污染的问题。 |
| 可移植性 | 对于需要在多平台实现的接口需要考虑接口本身的可移植性，因此最少使用对于系统依赖的类型作为接口的参数类型或者返回值类型 |
| 鲁棒性 | 接口需要有适度的鲁棒性，主要是指能够在多种情况下接口都能实现统一的效果，不会随着调用者传入的参数的变化而导致接口的输出出现违背接口语义的情况出现。 |
| 安全性 | 接口定义时需要严格限制参数的读写权限，如果只能是只读的参数一定要设置成 const，以免出现非法使用。 |
| 兼容性 | 这是接口扩充的原则，必须保证同一个接口实现后向兼容前一版本的使用。扩充的同类接口也能兼容老接口的实现。 |

## 用户接口

本系统提供可视化的操作方式，不提供命令控制语句进行输入控制，从而用户只需要使用鼠标进行命令操作。用户主要通过窗体、控件等可视化元素进行交互。考虑到本系统的特性，用户界面应符合以下设计规范：

|  |  |
| --- | --- |
| 要求 | 说明 |
| 友好性 | 界面直观、对用户透明，效果炫丽 |
| 交互性 | 通过鼠标点击进行操作，简单易懂 |

## 外部接口

### 软件接口

使用高德地图SDK，实现地图的现实和地理信息的获取。

### 硬件接口

与电子罗盘、GPS、wifi等模块交互，实现地理定位。

## 内部接口

设计内部接口时，各模块之间主要采用函数调用，函数传递，返回值的方式进行信息传递。



# 运行设计

## 运行模块组合

客户机程序在有输入时启动接收数据模块，通过各模块之间的调用，读入并对输入进行格式化。在接收数据模块得到充分的数据时，将调用网络传输模块，将数据通过网络送到服务器，并等待接收服务器返回的信息。接收到返回信息后随即调用数据输出模块，对信息进行处理，产生相应的输出。

服务器程序的接收网络数据模块必须始终处于活动状态。接收到数据后，调用数据处理/查询模块。

## 运行控制

运行控制将严格按照各模块间函数调用关系来实现。在各事务中心模块中，需对运行控制进行正确的判断，选择正确的运行控制路径。

在网络传输方面，客户机在发送数据后，将等待服务器的确认收到信号，收到后，再次等待服务器发送回答数据，然后对数据进行确认.服务器在接到数据后发送确认信号，在对数据处理，访问数据库后，将返回信息送回客户机，并等待确认。

## 运行时间

在软体的需求分析中，对运行时间的要求为必须对做出的操作有较快的反应。网络硬件对运行时间有最大的影响，当网络负载量大时，对操作反应将受到很大的影响。其次是服务器的性能，这将影响对数据库访问时间即操作时间的长短，影响加大客户机操作的等待时间，所以必须使用高性能的服务器。

# 系统出错处理设计

## 出错信息

在错误发生时，给出出错的原因。

## 补救措施

系统遭到恶意攻击，或是中毒以后，导致系统内部数据紊乱，用户信息失真，因此，应定期对系统数据进行自动备份，以便数据丢失时，能第一时间恢复，减少损失；另外，要定期对整个系统进行纸制的备份，以免出现自然灾害而导致所有数据的丢失，自动制定自动还原点；

在网络传输方面，可考虑建立一条成本较低的后备网络，以保证当主网络断路时的据之通信。

## 系统维护设计

采用模块化的设计，方便维护。

# 运行效果展示

无人机获取实时的视频流，进行分屏渲染，显示到手机上，通过VR眼镜观看，效果十分逼真。

# 附录

## 附录一 商业计划方案

### 9.1.1市场目标概述

国桥货运APP将会面向所有使用智能手机的用户。它具有直接面向用户生活实际的特点，给想要运货的人一个平台，利用闲置的货车或者类似于兼职的闲置劳动力，达到多方共赢的局面，并且增加社会的就业率。

国内智能手机市场正在飞速增长，中国手机用户人数即将突破10亿，因此手机APP的应用推广只增不减。而这种实际的和用户生活相关联的APP将会逐步成为用户的新宠。以这种思想的国桥货运的目标就是为中国的手机应用市场提供一种更加实际，时尚的生活方式，并且合理利用闲置资源提高就业率。国桥货运有望在未来三年成为应用市场用户量最大、活动最密集的APP。

### 9.1.2项目优势及特点介绍

国桥货运发现了中国互联网用户现在以及未来相当长的一段时间的用户需求特点，及时的为他们的提供这种服务，进行了很好的市场调查与定位。

困扰司机的头疼问题是空载率、黄牛费。据不完全统计，司机从起点到终点卸货后空返率的概率达40%，黄牛费间接推高运输成本。

对货主来说，找货同样是难点，找寻同条线路的返程车尤其困难，该线路费用较低，往往落到黄牛手上。

此外，诚信记录缺失，导致交易效率低下。由于货运主力95%都是个体，个体轨迹不能被追踪，交易记录无从记载，每笔交易都成为陌生交易，需“从头开始”。

项目小组成员研究发现，现在在这个领域中还没有开发类似APP的定位关于闲置资源的利用，可以说是一个相当大的商机。

### 9.1.3利润来源简析

APP具有很好利润来源，并且这种利润来源的方式并不影响用户使用APP。

主要的利润有以下几个方面：

1. 互推互换。

商场不是你死我活，而是大家互利共赢。，做到你中有我，我中有你，而APP之间用户的交换也是经常会遇到的。比如有两款货车相关的APP，因为两者之间都有司机的共同属性，可以采取资源互换的方式来增加APP应用的用户群，也可以在彼此的APP中内置对方的链接。如果在资源不对等的情况下，也可以通过一部分交换，一部分售卖的模式来做。

1. 借助第三方的数据。

我们都用过微信来打车，买电影票；又或者是从支付宝入口团购了一顿火锅，预定了一次周边游等等。我们不是通过打车APP、电影票APP直接购买需要的东西，而是刚好在微商或者支付宝上看到了，并且如果购买的商品还便宜的话，就会毫不犹豫的下手。因为我们平时的购买行为，微信和支付宝成功的将我们导向了打车软件、 电影票APP等第三方软件，并成功变现。导量的先决条件是APP要拥有比其他平台更大的用户量，简单来说就是APP之王，我们拥有用户量并且可以操纵用户，国内BAT都在打造自己的超级APP。这不禁应了中国的一句话叫人多力量大，各家都在积累自己的群体。因此随着我们用户群体的扩大，采用这种方式盈利是有效而可靠的。

1. 内容营销。

近两年，随着o2o的势头影响，线下渠道市场受到了越来越多的APP开发者重视，这一点特别表现在以校园为市场的产品中，类似超级课程表这样的校园兼职类APP。而我们的国桥货运也正是运用这一点，在学生搬宿舍时，可以给予一些比校园内运货车更好的优惠政策。而且一些学过车有驾照的学生不仅可以巩固车技，也可以通过这个APP赚取一些外快，就像一个小社团。但是商家进入到校园的难点在哪里呢？中国的高校一般是禁止商业宣传性质活动的，最基本的一个活动场地可能都很难拿下来，所以一般商家进入到学校都是通过找第三方外包商来做，而这么做的成本是相当高的。这个时候在学校拥有校园团队的APP开发商就自然成了一个不错的选择。一方面是校园团队本身就是用户，上架可以直接影响他们，另一方面商家的产品结合APP开发商具有共同的推广需求，联合推广成本肯定比外包公司低。

1. 免费增值服务

我们在国桥APP实现一定的积分政策，通过积分区间来确定会员的等级，0-1000为svip，1000-3000为mvip，3000-6000为vip，6000-10000为vvip。不同的等级优惠政策打折力度不同。目的是让用户通过APP应用养成了一定的使用习惯，随着积分的增长，用户也就越难以割舍。

积分主要从个方面获得：

* 每日签到
* 运货交易额
* 绑定银行卡，手机号
* 活跃度，每年的交易量，前几名有特殊的奖励

1. 出售广告位

启动页、banner图，还有一些边边角角都有可能成为APP的广告位。

随着我们积累的用户数，进行大数据分类，可以对人群进行细分，根据广告购买者的意愿对象进行投放。如果公司APP产品线多而且用户量大时，我们可以自己去操作广告平台。

当然广告投放量要控制在一定的而数量，并且尽量让其以最自然的方式出现。比如和做货车有关的广告商合作，在用户完成运货订单后，我们可以做一个简单的30秒用户游戏体验，让用户自己开一辆货车模拟运货过程。

1. 奖励与提成

|  |  |  |  |
| --- | --- | --- | --- |
| **货车类型** | **星级** | **种类** | **星级** |
| 低栏车 | 1 | 纯平板 | 1 |
| 高地板 | 2 |
| 高拦车 | 2 | 半封闭 | 1 |
| 全封闭 | 2 |
| 挂车 | 3 | 各类用于装载油料气体液体等专用罐车 | 4 |
| 冷冻加温装备的厢式机动车 | 3 |
| 清障，清扫，起重，装卸，升降搅拌，挖掘，推土，压路等各种专用机动车，车内装有固定仪器设备的用于监视消防运钞医疗，电视转播雷达X光等机动车 | 2 |
| 专门用于牵引集装箱箱体的集装箱拖头 | 1 |
| 厢式车 | 4 | 仓栏 | 2 |
| 普通厢式 | 1 |
| 海关监管 | 3 |
| 集中箱车 | 5 | 干货 | 1 |
| 杂货 | 1 |
| 液体 | 3 |
| 冷藏 | 3 |
| 散货 | 1 |
| 汽车 | 2 |
| 牲畜 | 3 |
| 兽皮 | 1 |
| 全挂车 | 2 |
| 半挂车 | 1 |
| 专用车 | 6 | 特种 | 3 |
| 商务 | 2 |
| 运输 | 1 |
| 军用 | 3 |
| 冷藏车 | 7 | 微型 | 1 |
| 小型 | 2 |
| 中型 | 3 |
| 大型 | 4 |
| 特种车 | 8 | 环卫 | 1 |
| 工程 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 货物价值 | 星级 | 司机车龄 | 星级 |
| 0-10000 | 1 | 3~5 | 1 |
| 10000-50000 | 2 | 6~8 | 2 |
| 50000-150000 | 3 | 9~10 | 3 |
| 150000-300000 | 4 | 11~13 | 4 |
| 300000-600000 | 5 | 14~16 | 5 |
| 600000-1200000 | 6 | 17~19 | 6 |
| 1200000-3000000 | 7 | 20~ | 4 |
| 3000000-6000000 | 8 |  |  |

注：星级评定是 货车类型+种类+货车价值+司机年龄

* 账号激活日起7日内：完成2单订单至少奖励50元，奖励取最高值发放。根据星级不同奖励金额会在50元的基础上，每多一星加5元。
* 平日每单司机可得到的金额为:
* 业务费80%+用户加价+附加费（过路过桥费）-0.5元（保险费）\*1-1.77
* 奖励到账时间：新人周期结束后统一发放。

1. 换肤增值模式

场景更新、附属功能更新、主题更新，向用户二次推送。

### 9.1.4投资和预算

国桥货运的投资主要用来组建app的网络公司跟项目后期运营的费用，当然也包括项目中期的启动部分，但是绝大部分是为了用于开展业务方面的费用app在开发结束到运行开始以及到开始盈利估计总共需要766,500的投资。

* 预算评估

表2-8 预算评估表

|  |  |
| --- | --- |
| 费用项目 | 方案预算 |
| 建设成本(RMB) | 研发办公室建设 150,000 + 5名开发人员工资 8,000 RMB/月 \* 5 \* 4 + 2名测试人员6,000 RMB/月 \* 4 \* 2 + 服务器 20,000 + 托管费用平均 3,500 RMB/年 \* 3 = 364,500 |
| 维护成本(RMB) | (固定维护人员工资 8000 \* 2 \* 12 RMB/年 ) \* 3 = 192,000 |
| 宣传成本(RMB) | 线上宣传（广告，人员工资等）20,000 \* 3 + 线下宣传活动（场地，物资，人员等） 50,000 \* 3 = 210,000 |
| 前三年的总投(RMB) | 766,500 |

### 9.1.5融资方案（资金筹措及投资方式）

项目的融资方案以种子资金为主风险投资为辅。种子资金可以分期投入资金，并且投资者等投资风险将由项目人员全力承担。种子资金投入的资金量可以较小，作为项目的启动资金，项目后期将会继续通过第三方融资来获得资金，种子资金获得的投资回报率也将会比风险投资要少。

风险投资同样也可以分期注入，但是投资风险将与项目小组共同承担，但是投资回报率将会比种子资金加大。融资具体的方案会进行具体洽谈以确定。

### 9.1.6预算及投资报酬

网站项目需要766,500资金，投资者的投资报酬可以通过一下几个方式获得，一种是等到网站盈利开始后通过分成来获得利润；第二种是通过运营成功成立公司，获得相应的股份，根据价值将股份退出，获得投资金额与股份金额的差价；第三种是公司通过融资上市，投资者的股份可以卖掉赚取高额的利润。

### 9.1.7 ROI分析

* 价值回报表

|  |  |
| --- | --- |
| Year 1 | 第一年投资APP的制作，通过一些个人和小型公司的合作测试。并且争取同当地政府取得合作，扩大宣传，合理利用闲置货车，闲置劳动力，而这些闲置资源可以是长达几年的，也可以短至一小时，从而增加就业率。和通信公合作，用户注册时赠送流量等方式来提高用户注册量；和广告商合作，出售广告位，插入链接等方式获取一定的利润；和其他的APP合作，资源共享，共同盈利。在合作中宣传国桥货运APP，在线下可以张贴广告等方式推广我们的应用。 |
| Year 2 | 第二年要通过扩大宣传推广与服务优化。逐步扩大租车人群，和大型货车公司合作，深入到校园内，随时随地拉货。 |
| Year 3 | 第三年，对已经相对成熟的服务进行维护升级，从用户的反馈中发展新的业务和功能。注册用户增加到10万以上，保持与已有合作伙伴的的合作，并不断寻求卖家进入市场，回收成本。 |

* 基础假设

表2-9 基础假设

|  |  |
| --- | --- |
| 基础假设 | 值 |
| 第1年人数 | 20000 |
| 系统使用率 | 80% |
| 活跃用户率（/天） | 50% |
| 初始广告交易笔数 | 100 |
| 广告交易笔数增长率（/年） | 20% |
| 初始一次性广告投入（元/次） | 1500 |
| 一次性广告投入费用增长率（/年） | 5% |
| 税率 | 20% |
| 机会成本率(WACC) | 12% |

* 年交易量分析

表2-10 年交易量分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 第0年 | 第1年 | 第2年 | 第3年 |
| 与其他APP合作 | 1 | 2 | 4 | 7 |
| 注册用户数量 | 16，000 | 32，000 | 64，000 | 112，000 |
| 广告交易笔数 |  | 100 | 120 | 144 |
| 广告固定收益 |  | 150,000 | 189,000 | 238,140 |
| 基线收入 | -350,000 | 214,000 | 317,000 | 462,140 |
| 最初建设投入 | 200,000 | 144,000 | 144,000 | 144,000 |
| 维护升级和宣传 |  | 60,000 | 60,000 | 60,000 |
| 净收入 |  | 10，000 | 113,000 | 258，140 |
| 资本折旧 |  | 40 | 452 | 1032.56 |
| 收益 |  | 9960 | 112,548 | 257,107.4 |
| 税后净收益 |  | 7,968 | 90,038.4 | 205,686 |
| 折旧转回补偿 |  | 40 | 452 | 1,032.56 |
| 税后现金流 | -200,000 | 8,008 | 90,490.4 | 206,718.5 |

* 增量现金流

表2-11增量现金流

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 第0年 | 第1年 | 第2年 | 第3年 |
| 现金流净增长 | -200,000 | 8,008 | 90,490.4 | 206,718.5 |
| 净现值(NPV) | 212904.74 |  |  |  |
| 机会成本率(WACC) | 12% |  |  |  |
| 3年内部收益率(IRR) | 17% |  |  |  |

* 投资回收期

表2-12 投资回收期分析表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 第0年 | 第1年 | 第2年 | 第3年 |
| 现金流净增长 | -200,000 | 8,008 | 90,490.4 | 206,718.5 |
| 累计现金流 |  | -191,992 | -101,501.6 | 105,216.9 |
| 投资回收期(PBP) |  | of year 2 | 🡺 | 6 |

* 分析总结

通过以上对解决方案的成本分析、收入分析、投资回报率的分析计算易得出以下结论：

* 净现值 NPV 为 212904.74，大于0；
* 内部收益率(IRR)17%，大于机会成本率WACC；
* 到了第三年积累的现金流就达到了 10.5万元。

综上所述，此解决方案是可行的。