第一章

* 1. 研究背景和意义
     1. 研究背景

互联网提升了传统行业的效率，改变了人们的生活方式，为用户带来了安全、便捷、舒适的体验。随着互联网的成熟、计算机信息网络的发展、智能终端的普及，移动互联网成为我们最依赖的媒体，通过它，我们不仅享受了随时随地的消息服务，而且完成了颇具创意的商业行为。出行APP就是其中之一。随着城市规模的扩大、城市人口的增加，人们出行的需求不断增大，像北京这样的大城市，每天会有高达200万次左右的打车行为发生，但每一个打车的用户都会发出“打车难”的感慨。其原因主要有：出租车仅能服务身旁可见区域，服务范围不足；单程远途或交通拥堵时接单可能导致司机亏损，司机无法盈利而拒载；城市公交车数量不足，车辆满载。在不增加出租车数量的前提下，出行APP在一定程度上解决了下班高峰期、道路拥堵期乘客一车难求，司机盈利等问题。

* + 1. 研究意义

1. NC399具备极强的应用价值。本文设计的NC399出行用户界面友好、美观，能在不同版本的Android系统上快速、稳定的运行。通过NC399，乘客可以预约出租车、私家车，掌握行车过程中的实时路线、路况及消费情况。使得出行打车简单便捷，消费透明实惠。
2. NC399中间件具备重复使用价值。本文实现的中间件构建了手机Android系统和乘客功能之间的联系，Android系统为中间件提供服务，乘客功能享受中间件提供的服务。其模块之间低耦合高内聚，方便功能模块的扩大，具备重复使用的价值。程序员可以在中间件的基础上二次开发，实现其他需求功能。
3. NC399的设计理念，开发过程具备参照价值。本文按软件设计的分层思想将NC399分成能够重复利用的中间件。根据面向对象设计思想，实现不同乘客功能模块。此外，NC399采用XXXXXXX技术。虽然NC399基于Android系统，但其设计理念，开发技术对软件的设计和开发具有参考价值。
4. NC399一定程度上解决南昌市民出行“打车难”。区别于传统电话叫车，NC399叫车可以准确定位乘客位置。使用乘客可以预约远处的出租车而不是仅仅局限于眼前，也可以通过和其他人拼车绿色出行、减少出行的花费。乘客预约之后可以在办公室、咖啡厅等待出租车，节省自己的工作、休闲时间。
   1. 国内外出行APP发展状况

2016年1月，我国发布了第37次CNNIC报告，从图1-1中可以看出，2015年底，我国人民使用手机上网的人数已经达到6.2亿，使用手机上网的人占到90.1%。从图1-2中可以看出，使用手机接入互联网的网民逐年增加。

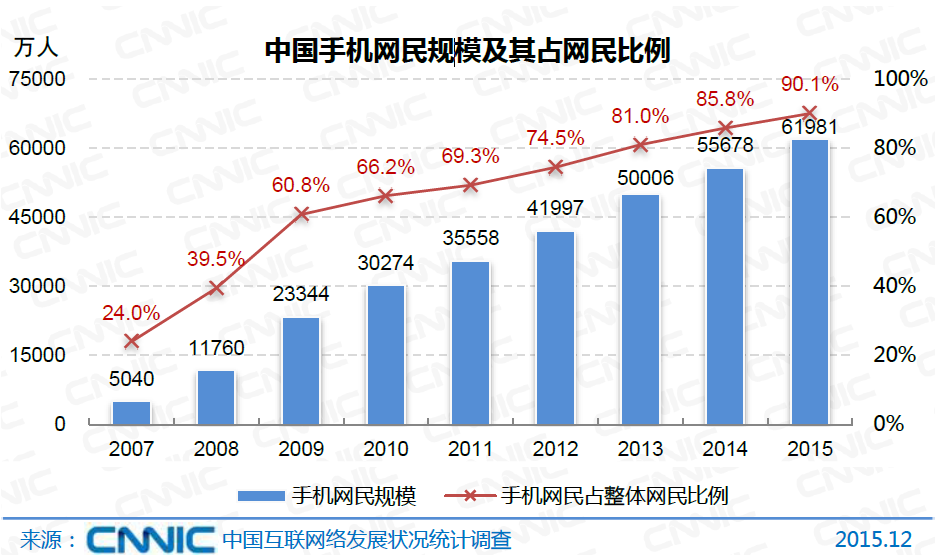


图1-1,中国手机网民规模及其占网民比例

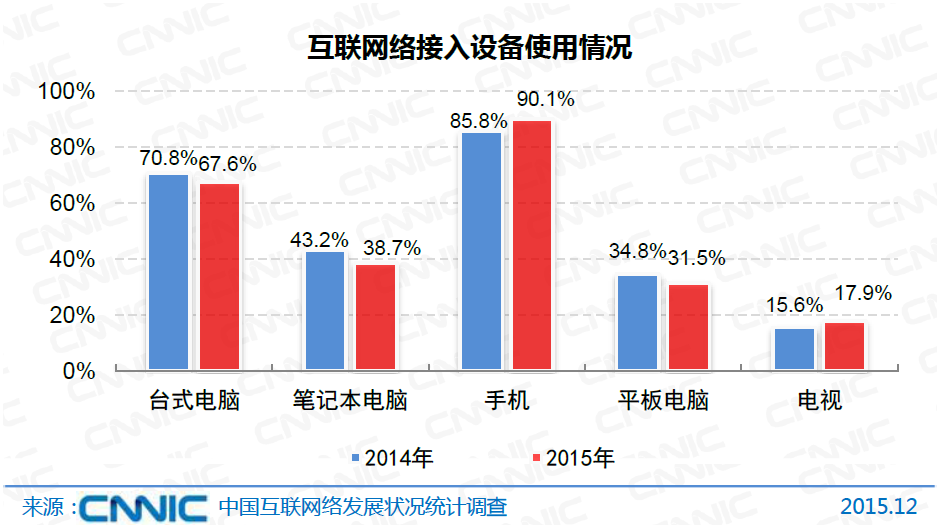


图1-2互联网络接入设备使用情况

乘着移动终端APP发展的顺风车，2010年8月，易到用车成功上线，成为我国首个打车APP，2012年2月摇摇招车APP上线，至此，中国手机打车市场悄然开启，APP数量按几何倍数增长，如大黄蜂、百米、滴滴打车、嘟嘟、快的打车等如图1-3所示。2013年，一线城市甚至二三线城市的用户使用打车软件叫车的频率越来越高，打车市场的竞争俞加强烈。随后快的收购大黄蜂，微信支付、支付宝分别接入嘀嘀打车和快的打车，快的和滴滴两者之间的市场竞争上升到白热化阶段。2015年2月,快的和滴滴宣布合并,携手对抗易到用车和国际出行APP权威Uber。

除了打车APP以外, 嘀嗒拼车，巴士拼车分别是出行O2O市场中专注于顺风车、公共出行两种细分市场的出行软件。2010年至今, 出行O2O市场经历了用户的考验进入全新的发展阶段。

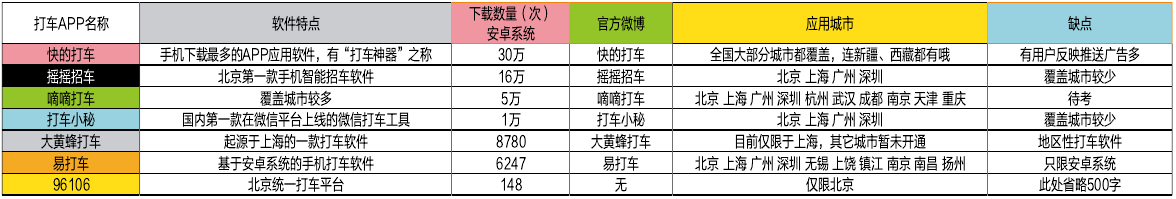


图1-3部分打车App

国外的出行O2O市场相对国内更加完善，不仅体现在打车服务上，其运营模式也非常丰富和多元，其打车及租车服务企业的商业模式和发展策略，能给我国打车APP发展带来启发。当前国外主流打车软件有Hailo、Uber、Lyft、Sidecar四家。

Uber：Uber是一款美国打车软件，其允许拥有中档汽车、豪华轿车或SUV等车型的车主加入Uber租赁服务。乘客通过手机软件Uber发送打车服务请求。相对其他类型的出租车服务，Uber提供的打车服务及拼车服务要价不菲。从成立以来Uber已经进入了121座城市，其中包括中国的北京，深圳，上海，广州四座城市。所以用户在国外旅行时，同样可以享受uber提供的叫车服务。

Hailo：Hailo的用户能够在手机地图上查看附近的出租车，可以通过若干条件如距离最近，5分钟车程之内等筛选出符合条件的出租车发送打车服务请求并等待车主确认，在车主确认用户的请求之后，你可以在手机APP上看到司机的详细信息，车子预计到达时间等。通过Hailo打车只需支付打车费用不用支付其他额外的费用，Hailo会根据和车主之间的协议向司机收取APP的使用费用。

Lyft：Lyft给用户提供私家车搭乘服务，它不仅是一个手机移动应用也是一个陌生人社交平台。用户在客户端上发布搭车请求，加入Lyft的车主如果在你附近并和你顺路，就会让你搭上他的顺风车，为其他人提供服务的车主也会有额外的费用收入。到达终点后，Lyft会提供一个比较合理的价格，但实际支付金额可用户和车主商量之后进行调整，Lyft从实际支付金额中抽取20%做为APP使用。Lyft试图在虚拟的网络世界建立一个真实的社交平台，这就意味着你在回家路上得到一个司机的同时，也许更得到了一个志趣相投的朋友。

Sidecar：用户通过手机移动APP发送打车服务请求，软件自动获取当前位置，用户输入目的地，愿意搭载你的Sidecar司机会马上将车开到你的位置。司机都是通过了Sidecar官方认证 的当地居民。Sidecar相比其他app，Sidecar可以查看别人乘坐这条路线所支付的金额，你可以根据其他乘客支付的金额来决定自己支付费用的多少，而且支付多少费用完全由个人意愿决定。Sidecar也允许司机自己定价，并鼓励他们在提供服务的同事供应一些个性化服务如音乐，充电，餐巾纸等。

第二章 基于Android平台的NC399客户端关键技术

本章主要介绍实现NC399的核心技术，本app在谷歌提供的Android SDK环境下开发，和服务器数据交互采用JSON的数据格式,和服务器通讯使用android-async-http框架，数据在手机端保存主要使用SharedPreferences存储。

2.1 JSON数据格式

JSON(JavaScript Object Notation)是目前最为常用的数据传输格式。JSON数据格式就是一个字符串，非常易于人阅读和书写，同时也便于计算机的解析。JSON数据格式和JavaBean之间转换较XML数据格式更加便利，所以在此与服务器的通讯使用JSON的数据格式。JSON数据主要有3种结构，分别是JSON对象、JSON数组、JSON对象和数组嵌套。下面将详细讲述JSON数据的这3种格式。

2.1.1 JSON对象

JSON对象这种格式可以使用键值对或者名值对来表示，如{key : value},键一般都是字符串类型，值则可以是字符串、布尔值、数值。JSON对象的具体格式如下图所示。

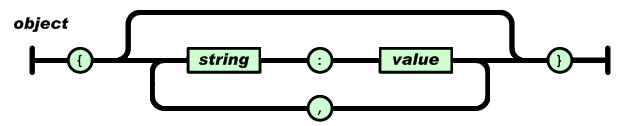


图 JSON对象格式

2.1.2 JSON数组

JSON数组的表示依然是键值对，键主要是字符串，值是一个数组，如：{“lists”:[1,2,3,4,5]}，具体如下图所示。

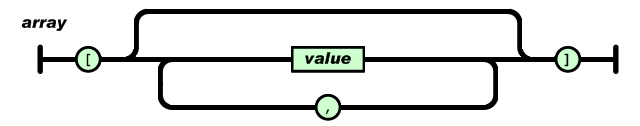


图 JSON数组格式

2.1.3 JSON对象和数组嵌套

复杂的JSON格式是通过嵌套来完成的，JSON对象中可以有JSON数组，而JSON数组中又有JSON对象，如：{

"total": 3,

"rows": [

{

"title": "百度",

"url": "http://www.baidu.com"

},

{

"title": "谷歌",

"url": "http://www.google.com"

},

{

"title": "优酷",

"url": "http://youku.com"

}

]

};

2.2 SharedPreferences数据存储

在Android平台中存储轻量级数据会使用SharedPreferences，SharedPreferences非常适合存储一些配置参数，SharedPreferences在Android中是以XML的形式存储的key-value键值对数据，文件的保存位置在/data/data/<应用包名>/shared\_prefs目录下。SharedPreferences对象能实现数据在本地的存储，而不支持数据的修改。如果需要对数据的修改和删除，那么需要用到Android平台的一个轻量级数据库SQLite。

2.3 android-async-http框架

在Android app开发时，如果需要访问网络，一般都是使用http协议接受和发送数据，Android原生提供了两个框架和网络进行HTTP通信，这两个框架分别是：HttpClient和HttpURLConnection。

谷歌提供的HttpClient和HttpURLConnection使用方式都很复杂，为了使开发者使用方便，有必要对这两个框架进一步封装，android-async-http就是封装好了的一个优秀框架。android-async-http框架封装好了HTTP请求，开发者使用此框架访问网络无需关心访问网络的具体细节，只需调用此框架暴露的API即可，网络的请求结果通过回调传递。AsyncHttpRequest框架通过线程池管理框架中开启的子线程，防止因线程开的过多而导致的OOM。

2.3.1 android-async-http框架的主要类介绍

AsyncHttpRequest类：此类的父类是Runnabler，被提交到线程池，然后执行网络请求。

AsyncHttpResponseHandler类：此类封装了网络请求的结果，网络请求的状态和结果通过此类回调传递值，此类的方法有onStart、onFinish、onSuccess、onFailure。onStart方法在网络请求的开始时执行，onFinish方法在网络请求完成时执行。onSuccess方法在网络请求成功时执行。onFailure在网络请求失败是执行。

AsyncHttpClient类，此类为核心类，使用HttpClient发起网络请求，里面提供了多种网络请求方法，如GET、POST、PUT、DELETE、HEAD等。

RequestParams类，此类封装了网络的请求参数，请求参数可以是字符串也可以是文件、字符流。

2.3.2 android-async-http框架的请求流程

android-async-http框架的请求流程如图所示。

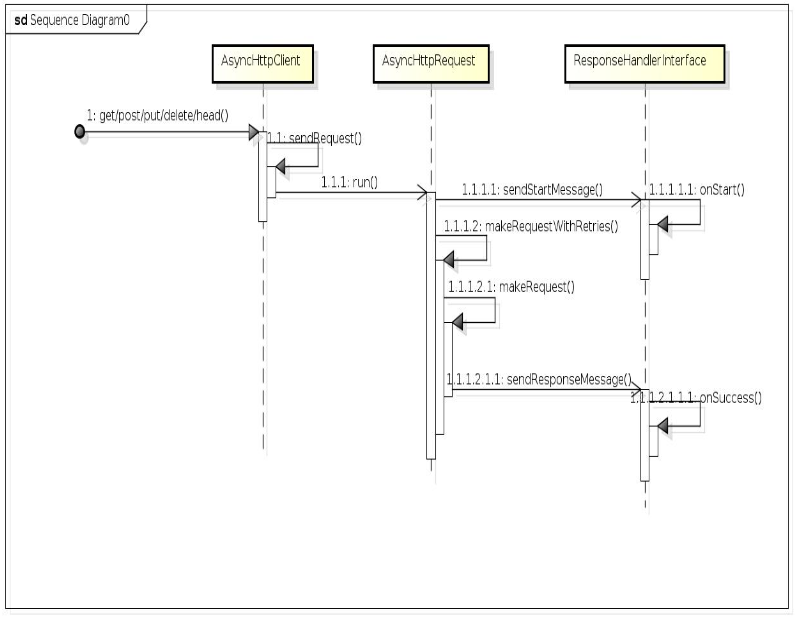


图 android-async-http框架的请求流程

1.调用AsyncHttpClient的get或post等方法发起网络请求

2.所有的请求都走了sendRequest，在sendRequest中把请求封装为了AsyncHttpRequest，并添加到线程池执行

3.当请求被执行时（即AsyncHttpRequest的run方法），执行AsyncHttpRequest的makeRequestWithRetries方法执行实际的请求，当请求失败时可以重试。并在请求开始，结束，成功或失败时向请求时传的ResponseHandlerInterface实例发送消息

4.基本上使用的都是AsyncHttpResponseHandler的子类，调用其onStart，onSuccess等方法返回请求结果

2.4 Android系统架构

Android系统架构使用分层次的系统架构，主要由Linux Kernel（linux内核）、Libraries（类库）、Android Runtime（系统运行时）、Application Framework（应用程序架构层）和Applications（应用程序层），具体的结果如图所示。下面将详细的讲述这几层。

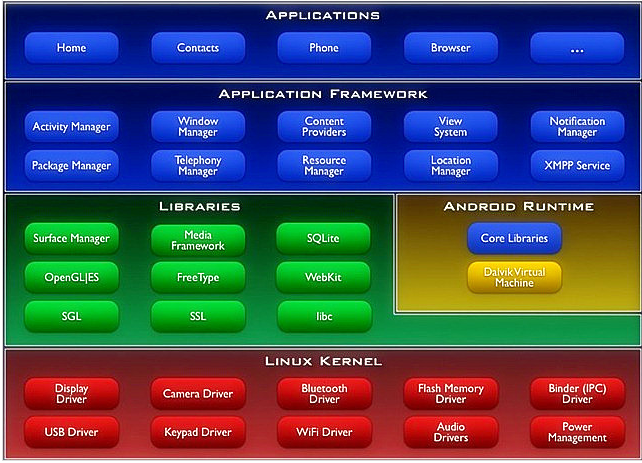


图 Android系统架构

2.4.1 Linux Kernel

Android操作系统是基于Linux操作系统之上的，通过Linux内核服务提供硬件驱动。Android 4.0(api 14)版本之前是基于Linux2.6内核，而之后的版本是使用更新的Linux3.x内核。Android系统对Linux内核进行了加强，增加了很多移动端的特有功能。如：低内存管理器LMK(Low Memory Keller)、匿名共享内存(Ashmen)。这些增强的功能使Android继承Linux内核的机制时，还进一步的提升了内存管理，增加了通信的安全。Android内核的主要的驱动有Android电源管理（Power Ma nagement）、低内存管理器（Low Memory Keller）、匿名共享内存（Ashmem）、日志（Android Logger）、定时器（Anroid Alarm）、物理内存映射管理（Android PMEM）、Android定时设备（Android Timed device）、Yaffs2文件系统、Android Paranoid网络，其具体的说明如下表。



|  |  |
| --- | --- |
| 驱动名称 | 说明 |
| Android电源管理（Power Ma nagement） | 针对嵌入式设备的，基于标准Linux电源管理系统的，轻量级的电源管理驱动 |
| 低内存管理器（Low Memory Keller） | 可以根据需要杀死进程来释放需要的内存。扩展了Linux的OOM机制，形成独特的LMK机制 |
| 匿名共享内存（Ashmem） | 为进程之间提供共享内存资源，同时为内核提供回收和管理内存的机制 |
| 日志（Android Logger） | 一个轻量级的日志设备 |
| 定时器（Anroid Alarm） | 提供了一个定时器用于把设备从睡眠状态唤醒 |
| 物理内存映射管理（Android PMEM） | DSP及其他设备只能工作在连续的物理内存上，PMEM用于向用户空间提供 连续的物理内存区域映射 |
| Android定时设备（Android Timed device） | 可以执行对设备的定时控制功能 |
| Yaffs2文件系统 | Android采用大容量的NAND闪存作为存储设备，使用Yaffs2作为文件系统管理大容量MTD NAND Flash；Yaffs2占用内存小，垃圾回收简洁迅速 |
| Android Paranoid网络 | 对Linux内核的网络代码进行了改动，增加了网络认证机制。可在IPV4，IPV6和蓝牙中设置，由ANDROID\_PARANOID\_NETWORK宏来启用此特性。 |

2.4.2 Libraries

系统类库大部分是用C++写的，此类库提供了很多功能供Android应用程序使用，主要的系统类库如下表所示。

表 android系统类库

|  |  |
| --- | --- |
| 系统类库名称 | 说明 |
| Surface Manager | 执行多个应用程序时，管理子系统的显示，另外也对2D和3D图形提供支持 |
| Media Framework | 基于PacketVideoOpenCore的多媒体库，支持多种常用的音频和视频格式的录制和回放，所支持的编码格式包括MPEG4，MP3，H264，AAC，ARM |
| SQLite | 本地小型关系数据库，Android提供了一些新的SQLite数据库API，以替代传统的耗费资源的JDBC API |
| OpenGL|ES | 基于OpenGL Web浏览器的软件引擎ES 1.0API标准实现的3D跨平台图形库 |
| FreeType | 用于显示位图和矢量字体 |
| WebKit | Web浏览器的软件引擎 |
| SGL | 底层的2D图形引擎 |
| Libc（bionic l ibc） | 继承自BSD的C函数库bionic libc，更适合基于嵌入式Linux的移动设备 |
| SSL | 安全套接层，是为网络通信提供安全及数据完整性的一种安全协议 |



2.4.3 Android Runtime

Android Runtime包括两个部分，分别是Core Libraries(核心库)和Dalvik Virtual Machine(Dalvik虚拟机)。

核心库提供了Java5 API的很多功能，并为Android提供核心API,如：android.media。

Dalvik虚拟机是基于JVM,并且适当的优化以降低内存，Dalvik依赖于Linux内核，提供安全和异常管理功能，并能让垃圾自动回收。

2.4.4 Application Framework

Application Framework是我们开发Android开发的基础，很多应用都需要依靠该层实现其指定的核心功能，该层简化了开发者的使用，增加了组件的重用性，为开发者带来极大的便利，能够快速的开发应用程序。Application Framework主要由Activity Manager(活动管理器)、Window Manager(窗口管理器)、Content Provider(内容提供器)、View System(视图系统)、Notification Manager(通知管理器)、Package Manager(包管理器)、Telephone Manager(电话管理器)、Resource Manager(资源管理器)、Location Manager(位置管理器)、XMPP Service(XMPP服务)。

2.4.5 Application

Android系统不只是一个操作系统，系统还自带了很多应用程序，比如：电话拨号、浏览器、联系人、桌面。所有的应用程序都是用Java语音编写的，开发者也可以不使用系统自带的应用程序，可以根据实际需求制定开发应用程序。

2.5 本章小结

客户端和服务端的通讯的数据格式一般都使用JSON格式，本章着重的介绍了3种JSON数据格式。Android数据本地持久化使用SharedPreferences，SharedPreferences可以保持一些简单的配置信息，很方便的做到数据的持久化。Android系统架构主要是分层次的，Android系统的下一层作为上一层的基础，为Android应用程序的运行提供了支持。

**3基于Android平台的NC399出行的分析**

本章首先讲述用户使用NC399出行的整体功能，其次对本次开发的NC399出行做出较为详细的需求分析，包含NC399出行中打车模块、拼车模块、订单模块、个人信息模块的详细功能描述及实用性、复用性、响应时间等软件性能的分析。

3.1 NC399出行的系统阐述

NC399出行的服务器端用于存储与用户、司机相关联的订单信息包括用户信息、拼车信息、叫车信息。NC399出行的客户端需要提供友好的交互界面并能够兼容PC、Android、ios、windows phone这四个当前主流的系统。由于工作阶段主要负责基于Android的应用程序的实现，本次NC399出行实现只基于Android系统，以下是详细的功能模块、系统性能需求分析

* 1. NC399出行功能模块需求分析

NC399出行总共分为打车、拼车、订单、个人信息四个大模块，功能模块需求分析将对这四个模块进行详细的功能阐述。

3．2.1 NC399出行的主要功能

NC399出行致力于解决需要乘车出行的用户在任何地点、任何时间都能打上出租车并安全、准时到达目的地这一问题。用户使用NC399出行前需要进行注册（如图3-1所示），待身份核实之后登录NC399出行即可在侧滑栏中看见NC399出行的四大模块。四大模块可以在预约叫车、拼车，即时叫车、拼车，发表个人意见等方面提供支持。



图3-1注册流程图

3.2.2 NC399出行的功能模块

1、 拼车模块：拼车模块为用户提供拼车服务，可以发布拼车、加入/退出拼车、删除拼车。

（1） 发布拼车信息：用户点击“拼车”大厅的“发布拼车信息”，填写出发时间、出发地、目的地、拼车人数及必要说明如免费、AA、实际费用分摊等，填写完必要信息后发布拼车需求具体流程如图3-2所示。



图3-2 发布拼车流程图

（2）加入或退出拼车：用户搭车人数较少或者不想创建拼车订单时可以加入或者退出别人发起的拼车订单。通过查找定位到适合自己的拼车出行订单，进行加入或退出操作，加入拼车订单具体流程如图3-3所示。



图3-3 加入拼车订单流程图

用户也可以通过筛选，定位到已经加入的拼车，进入“拼车详情”页，点击右上角的“退出”完成拼车退出。

（3）拼车成员联系：加入拼车订单之后，用户可以查看拼车成员的具体信息，能够通过电话或者短信方式与拼车成员沟通出行细节，以下为查看拼车成员信息流程

1）点击拼车订单，进入“拼车详情”页；

2）点击要联系的拼车成员，系统弹出成员信息页面；

3）可以点击“私信给TA”，发短信联系；点击“语音通话”电话联系。

（4）踢出拼车：发起拼车的用户有权限踢出加入拼车的成员，具体踢出拼车流程如下：

1）点击用户自己发布的拼车记录，进入“拼车详情”页；

2）长按想要踢出的成员；

3）在弹出的“提示”页面，点击“确定”，即可踢出此成员。

（5）删除拼车：在“拼车”大厅中点击自己发布的拼车订单，进入“拼车详情”页，点击“删除拼车”完成拼车的删除，用户删除拼车后，拼车记录将在“拼车”大厅中消失。

2、叫车模块：叫车模块为用户提供发布叫车、取消叫车服务。

（1）发布叫车：用户发布“叫车”需求后，系统会将叫车订单推送给司机。用户点击左侧菜单的“叫车”,可以发起“现在用车”和“预约用车”需求。具体发布叫车流程如图3-4所示。



图3-4 发布叫车流程图

（2）取消叫车：用户误操作或者不想叫车可以打开左侧菜单栏中的“订单”，选择想取消的订单，进入后通过点击右上角的“取消订单”完成订单的取消。用户取消叫车后，司机端的订单将消失。

3.订单模块：订单模块可以查询用户所有叫车、拼车行为也能查看司机信息、评价司机

（1）联系司机：通过电话联系司机，沟通具体细节，司机接单后，在订单详情中通过“电话TA”与之联系。

（2）评价司机：对司机的服务给予评价，监督其服务质量，针对司机的服务可以进行5星的打分，点击“评价”后，司机将收到系统关于评价的消息推送。

4.用户模块：用户注册之后，可以通过用户模块对个人信息进行修改，包括登录密码、头像、昵称、邮箱、区域、生日、性别信息。

（1）修改基本资料：修改内容不涉及账号安全性的资料即修改基本资料具体流程如图3-5所示。



图3-5修改基本资料流程图

（2）登录密码的修改：登录密码涉及账户安全，与基本信息有一定的区别，具体修改流程如下：

1） 用户进入“个人中心”后点击“账号安全”；

2） 输入当前正在使用的密码；

3） 输入绑定手机号、手机验证码以及新的密码，点击“提交”后，完成密码的修改。

3.3 NC399出行非功能性需求分析

NC399出行在完成各模块功能的同时，要符合系统的响应时间，安全性，易用性，可维护性，可管理性和可移植性等非功能性的需求。本节将对这些非功能性的需求进行简单的阐述。

3.3.1 NC399出行的响应性能需求

　　　NC399出行运行在Android系统中，每天面对成百上千万用户的使用，所以我们首先需要考虑NC399出行的数据库端能不能承受千万级的并发访问，保证并发正常的情况下，我们需要让NC399出行的响应时间小于3秒，如果响应时间大于三秒手机会出现假死现象影响用户的使用体验，难以留住用户。

3.3.2　NC399出行的安全性需求

系统应具有在规定时间进行系统备份、提供异常日志、防止病毒的侵袭、数据校验等功能。同时上线系统、开发环境和测试环境一定要严格分开，软件功能的修改和添加必须经过严格的审核和记录，在系统设计时应采用分层结构使用三层以上分层结构，表现层用户输入的数据需要经过业务逻辑层处理之后才能对数据库进行操作，以免数据发生错误。此次设计关注NC399出行的客户端，客户端需要保证提供身份认证的用户才能够进行拼车、叫车。

3.3.3 NC399出行的易用性需求

在移动互联网时代，不同细分市场的移动应用数见不鲜，设计复杂没有特点的APP几乎都无法在市场立足，简便易用是一个优秀APP应该具有的基本特征。结合UI设计、人机交互研究、用户操作app的行为分析等知识，NC399出行的用户界面应该统一风格，使用让人舒适温暖的色调，使其变得友好、简洁、美观、清晰、操作简便。当然NC399出行在突出其特点的同时也与当前市场主流APP保持比较一致的风格，这样用户只需要花费非常短的时间就能熟练使用该应用。具体的界面采用扁平化设计，一个界面只需突出一两个功能，显得美观、大方。

3.3.４ NC399出行的可维护性需求

系统维护包括对系统上线之后出现的各种问题进行解决、系统功能的添加和修改及系统不同版本的维护。首先，系统经过测试成功上线，用户使用时还是可能会出现各种问题如：系统响应时间长、数据错误、无法准定位边远地区等，这需要开发人员对系统进行优化维护，只有在系统维护性高即模块划分清晰、日志文件完整的情况下才能快速有效的调整系统。其次，应用系统会定时增加新功能不断完善完善自己，给用户带来极致的体验，这将涉及系统功能的添加和修改、系统版本控制这两方面的问题，需要设计模块的分类、良好的代码让新旧版本平稳过度、新老功能完美兼容。这些都是都是应用程序后期的维护工作，所以应用程序的可维护性在非功能性需求中非常重要，必须要满足基本的问题维护、功能添加、版本过度等问题。

3.3.5 NC399出行的可移植性需求

NC399出行采用分层设计思想，中间件与前端采用接口交互，虽然只是基于Android平台，但中间模块代码可以跨平台移植，具有非常强的复用性，为其他移动应用提供参考。

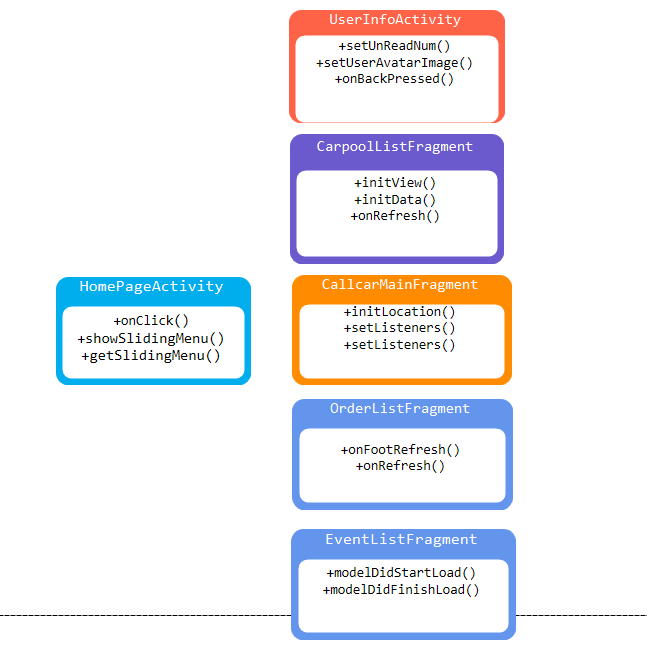
3.4本章小结

本章开始时概述了NC399出行的总体需求，然后将NC399出行功能性需求分成叫车、拼车、订单、用户四大模块进行详细的需求分析，最后从可移植性、可维护性、安全性、易用性、响应性能五个方面简单描述系统的非功能性需求。

**4．基于Android系统的NC399出行的设计与实现**

本章集合上章的需求分析,对拼车、打车、个人信息及订单四大功能模块使用类图等图形工具进行设计。类图能够将每个模块中类与接口的关系清晰展现，有助于模块内部实现的理解。

4.1导航栏的设计与实现



4.2 拼车模块的设计与实现

4.2.1 拼车模块主页面的数据显示实现

用户进入HomePageActivity点击SlidingMenu的Button进入相应的界面。当点击拼车时，HomePageActicvity中的Fragment切换成CarpoolListFragment，CarpoolListFragment的父类是CommonFragment。CommonFragment也是所有Fragment的父类，里面封装所有子类的共同的方法。当切换到CarpoolListFragment时，客户端会向服务端请求网络数据，查询所有的拼车信息。CarpoolListFragment通过调用mCarpoolDto.getCarpoolList(seatRemain, departTime, qFromAddr, qToAddr, offset + "", pageSize + "", "")方法，并传递对应的参数访问网络。网络的请求结果通过实现TTModelDelegate<CarpoolDto>接口回调传递，在回调方法public void modelDidFinishLoad(CarpoolDto model)中能得到网络请求的结果。服务端返回的数据封装在model中，通过model.mCarpoolListResponeModel.retData.getCarpoolList()方法获取拼车列表，获取拼车列表之后调用mCarpoolAdapter.notifyDataSetChanged()通知ListView刷新数据，让界面重绘显示最新数据。

4.2.2 拼车模块的的拼车信息筛选实现

在拼车页面点击筛选，弹出一个对话框，如图所示，



图 拼车筛选弹框

用户可以根据需求筛选所需要的拼车信息，点击筛选按钮执行的逻辑如下：

if(mFilterDialog == null) {

mFilterDialog = new FilterDialog(getActivity());

}

mFilterDialog.registerListener(new FilterDialog.DlgListener() {

@Override

public void confirm() {

seatRemain = "";

departTime = "";

qFromAddr = "";

qToAddr = "";

seatRemain = mFilterDialog.mRemain;

if (!TextUtils.isEmpty(mFilterDialog.mStartTimeEditText.getText().toString())) {

String startTime = mFilterDialog.mStartTimeEditText.getText().toString();

startTime = startTime.replaceAll("-", "").replaceAll(" ", "").replaceAll(":", "");

departTime = startTime + "00:";

}

if (!TextUtils.isEmpty(mFilterDialog.mEndTimeEditText.getText().toString().trim())) {

String endTime = mFilterDialog.mEndTimeEditText.getText().toString().trim();

endTime = endTime.replaceAll("-", "").replaceAll(" ", "").replaceAll(":", "");

if (TextUtils.isEmpty(departTime)) {

departTime = ":" + endTime + "00";

} else {

departTime = departTime + endTime + "00";

}

}

qFromAddr = mFilterDialog.mStartPlaceEditText.getText().toString();

qToAddr = mFilterDialog.mEndPlaceEditText.getText().toString();

currentState = STATE\_REFRESH;

isFilter = true;

showLoading();

mPullToRefreshListView.setRefreshing();

mCarpoolDto.getCarpoolList(seatRemain, departTime, qFromAddr, qToAddr, "0", pageSize + "", "");

}

});

4.3 叫车模块的实现

4.3.1 叫车模块地图的显示

本模块通过集成百度地图SDK,获取百度地图提供的服务。在CallCarMainFragment的布局文件中添加<com.baidu.mapapi.map.MapView

android:id="@+id/bmapView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:clickable="true"/>

再在CallCarMainFragment的onCreate方法中添加一下代码

mMapView = (MapView) getView().findViewById(R.id.bmapView);

mBaiduMap = mMapView.getMap();

mLocationClient = new LocationClient(getActivity().getApplicationContext());

mLocationClient.registerLocationListener(myListener);

百度地图就能显示出来了，如图所示。



图 百度地图

4.3.2叫车模块叫车的实现

点击CallCarMainFragment的叫车按钮就进入CallcarCreateActivity，在此页面能够选择起点和终点并规划路线，选择好起点和终点之后点击叫车按钮则可以叫车，并生成叫车订单，叫车的实现主要逻辑如下：if (mIsAppoint && TextUtils.isEmpty(mTxtStartTime.getText())) {

Toast.makeText(this, "请选择出发时间", Toast.LENGTH\_SHORT).show();

} else if (TextUtils.isEmpty(mPathInfo.getStartPos().getName())

|| TextUtils.isEmpty(mPathInfo.getStartPos().getAddress())

|| TextUtils.isEmpty(mPathInfo.getStartPos().getLatitude())

|| TextUtils.isEmpty(mPathInfo.getStartPos().getLongitude())) {

Toast.makeText(this, "请输入正确的起点位置", Toast.LENGTH\_SHORT).show();

} else if (TextUtils.isEmpty(mPathInfo.getEndPos().getName())

|| TextUtils.isEmpty(mPathInfo.getEndPos().getAddress())

|| TextUtils.isEmpty(mPathInfo.getEndPos().getLatitude())

|| TextUtils.isEmpty(mPathInfo.getEndPos().getLongitude())) {

Toast.makeText(this, "请输入正确的终点位置", Toast.LENGTH\_SHORT).show();

}else{

CallingCarWaitingDlg.DlgListener listener = new CallingCarWaitingDlg.DlgListener() {

@Override

public void send() {

mProgressDlg.setMessage("正在叫车");

mProgressDlg.show();

if (TextUtils.isEmpty(mEtPrice.getText().toString().trim())) {

mPriceString =0 + "";

} else {

mPriceString= mEtPrice.getText().toString().trim();

}

if (mType == CallCarType.APPOINT\_CARPOOL) {

mCarpoolIdString = "";

}

if(!mIsAppoint){

mStartTimesString ="";

}

mcallcardto.createOrder(CallcarTypeNumber, mStartTimesString,"",mPriceString,

mPathInfo.startPos.getAddress(),mPathInfo.startPos.getName(),mPathInfo.startPos.getLongitude(),mPathInfo.startPos.getLatitude(),

mPathInfo.endPos.getAddress(),mPathInfo.endPos.getName(),mPathInfo.endPos.getLongitude(),mPathInfo.endPos.getLatitude(),

mPathInfo.getDistance()+"",mDriverId,mCarpoolIdString);

}

};

CallingCarWaitingDlg dlg;

if (mIsAppoint) {

dlg = new CallingCarWaitingDlg(CallcarCreateActivity.this, mPathInfo, mTxtStartTime.getText().toString().trim());

} else {

dlg = new CallingCarWaitingDlg(CallcarCreateActivity.this, mPathInfo);

}

dlg.registerListener(listener);

}

4.4订单的实现

在主页点击订单按钮，HomePageActivity中的Framgment则替换成OrderListFragment，一进入OrderListFragment，通过调用mOrderDto.loadOrderList(pageNumber)方法获取服务端的所有的订单条目，通过ListView来显示所有的订单，显示订单OrderAdapter适配器的主要逻辑如下：

OrderModel orderModel = mOrderList.get(position);

holder.mOrderIdTextView.setText(orderModel.getOrderId());//设置订单id

if (orderModel.getStatus().equals("3") || orderModel.getStatus().equals("4")) {

holder.mExpireImageView.setVisibility(View.VISIBLE);

} else {

holder.mExpireImageView.setVisibility(View.GONE);

}

//设置订单状态

if(orderModel.getStatus().equals("0") || orderModel.getStatus().equals("4")) {

holder.mOrderStatusTextView.setText("未接单");

} else if (orderModel.getStatus().equals("-1")) {

holder.mOrderStatusTextView.setText("交易关闭");

} else if (TextUtils.isEmpty(orderModel.getDriverScores()) || orderModel.getDriverScores().equals("null")) {

holder.mOrderStatusTextView.setText("未评价");

} else {

holder.mOrderStatusTextView.setText("已评价");

}

holder.mStartPlaceTextView.setText(orderModel.getFromEdited());//设置起始点

holder.mEndPlaceTextView.setText(orderModel.getToEdited().trim());//设置终点

//设置距离

if (TextUtils.isEmpty(orderModel.getDistance()) || orderModel.getDistance().equals("null")) {

holder.mDistanceTextView.setText("距离0公里");

} else {

holder.mDistanceTextView.setText("距离" + (Integer.parseInt(orderModel.getDistance()) / 100 / 10.0) + "公里");

}

//订单时间

SimpleDateFormat format = new SimpleDateFormat("yyyyMMddHHmmss");

SimpleDateFormat newFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm");

4.5 个人信息模块的实现

个人信息模块主要的功能是展示用户信息，该模块的主要功能有查看推送信息、修改资料、账户安全、常见问题、关于。如图所示。



图 个人信息

4.5.1 消息推送功能的实现

消息推送功能能够查看服务端推送下来的信息，比如公司活动、优惠活动等。推送消息模块实现的主要逻辑如下：

AnnouncementModel announcementModel = new AnnouncementModel();

DbModel dbModel = null;

String count = "";

try {

dbModel = announcementModel.findDbModelBySQL("select count(\*) from com\_hand\_alt399\_jpush\_model\_AnnouncementModel where read = '0'");

if (dbModel != null) {

count = dbModel.getString("count(\*)");

}

} catch (Exception e) {

Log.e("399", "表还没建好");

}

Log.e("399", "消息未读数：" + count + "条");

if (TextUtils.isEmpty(count)) {

mUnreadNumTextView.setVisibility(View.INVISIBLE);

} else if (count.equals("0")) {

mUnreadNumTextView.setVisibility(View.INVISIBLE);

} else {

mUnreadNumTextView.setText(count);

}

}

4.5.2 修改资料功能的实现

资料的修改功能能够实现昵称的修改、用户头像的修改、生日的修改、邮箱的修改。主要的逻辑如下：

mUserInfoDto.enrichUserinfo("", mNicknameEditText.getText().toString().trim()

, AltUserCache.shareInstance().getUserModel().getMobilePhone()

, mEmailEditText.getText().toString().trim()

, AltUserCache.shareInstance().getUserModel().getSex()

, mCityEditText.getText().toString().trim()

, newdate, "", "");

**5.效果展示**

本章先说明NC399出行的功能测试和性能测试情况，再对NC399出行的研发结果做出截图展示。

5.1测试

测试小组通过大量的测试用例对NC399出行进行了模块测试、集成测试及系统测试。保证每个模块的运行能够正常、可靠，不同模块能够相互合作、兼容。

5.1.1功能测试

功能测试可以分模块进行，开发完一个模块即可测试一个模块。测试人员根据需求文档，每个功能模块的具体流程，采用边界值分析法设计不同特征的测试用例，通过黑盒测试保证NC399出行的功能模块能够达到用户的需求。

5.1.2 性能测试

在NC399出行的性能测试方面，采用Monkey工具进行测试。monkey测试就是利用socket通讯的方式来模拟用户的按键输入，触摸屏输入，手势输入等，看设备多长时间会出异常。当Monkey程序在模拟器或设备运行的时候，如果用户出发了比如点击，触摸，手势或一些系统级别的事件的时候，它就会产生随机脉冲，所以可以用Monkey用随机重复的方法去负荷测试开发的软件【互联网百度百科】。无限重复可以产生百万级事件，所以通过monkey测试的软件具有非常强的健壮性。

5.2 拼车模块效果展示

拼车模块分为发布拼车信息、加入或退出拼车、拼车成员联系、踢出拼车及删除拼车六个功能，以下对这六个功能模块进行截图展示

5.2.1发布拼车信息

打开应用 点击拼车进入拼车模块 点击发布拼车信息进入创建新拼车页面 点击创建创建新拼车单 点击叫车成功创建拼车订单，点击我已有车允许别人拼车并跳转至拼车大厅 点击订单查看订单详细信息，具体页面展示如图5-1所示。

应用在填写拼单信息时可进行地图选择或手动输入目的地、出发地如图5-2所示

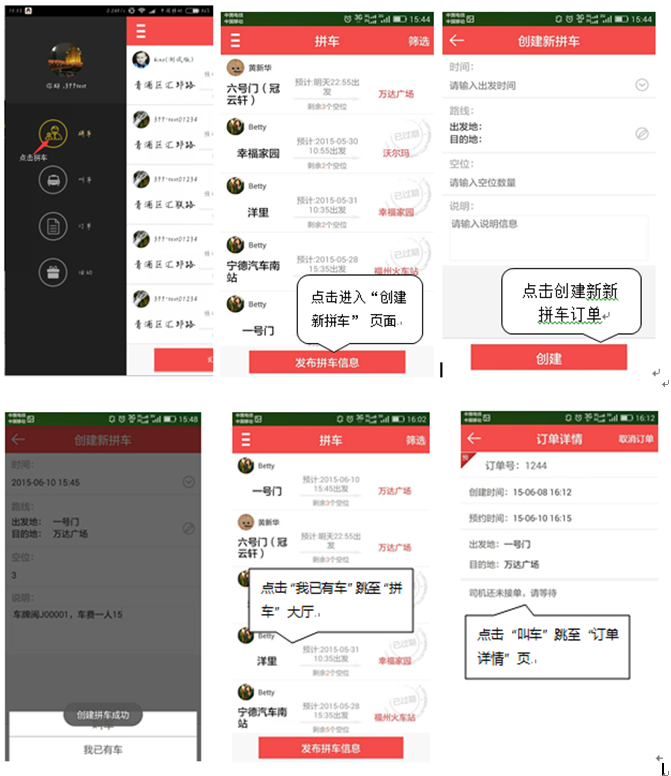


图5-1 拼车模块效果图

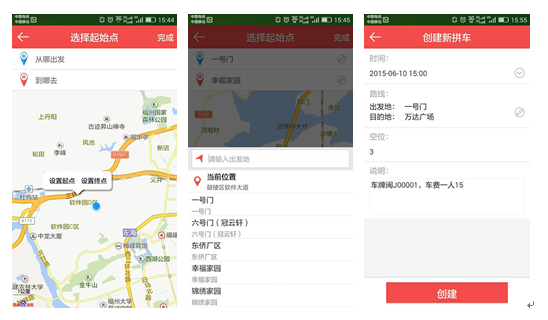


图5-2 信息输入界面

5.2.2 加入或退出拼车

1.加入拼车：在拼车大厅页面点击右上角筛选进入条件筛选页面，输入筛选条件点击拼车进入拼车详情页，然后在地图中选择上车地点，具体效果展示如图5-3所示。

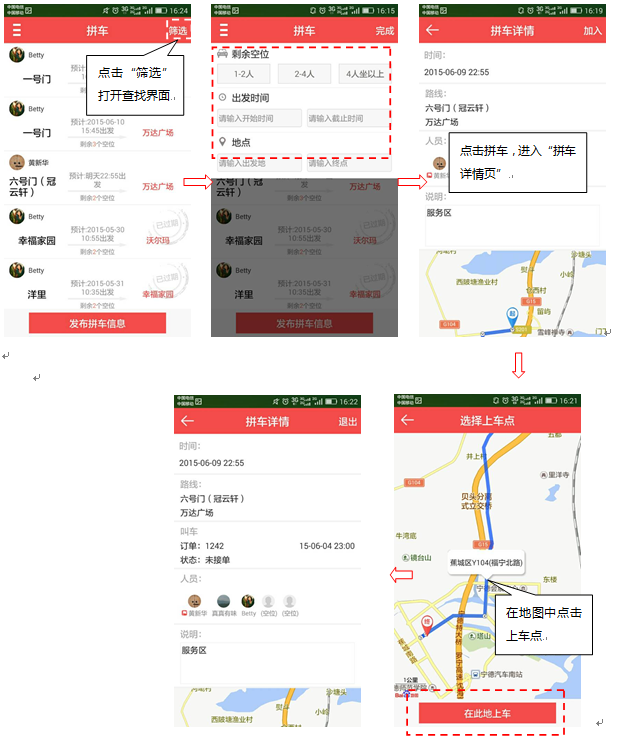


图5-3 加入拼车效果图

（2）退出拼车：进入拼车订单详情点击右上角退出即可退出拼车，如图5-4所示。



图5-4 退出拼车效果图

5.2.3拼车成员联系：进入拼车详情页面点击要联系的拼车成员，进入该成员的详细资料页面，如图5-5所示。

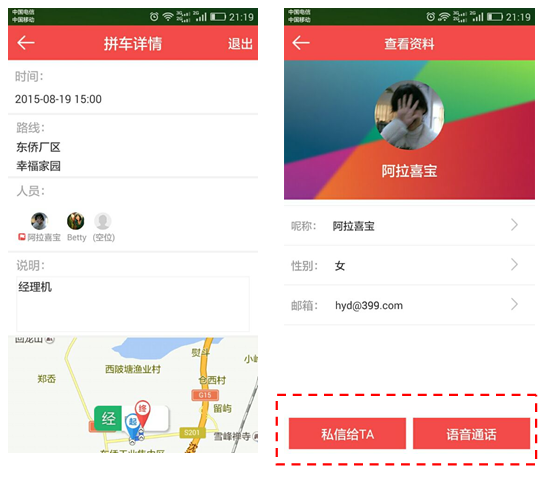


图5-5 联系拼车成员效果图

5.2.4踢出拼车：进入自己发起的拼车订单详情，长按想要踢出的成员，点击确定将该拼车用户踢出，效果如图5-6所示。



图5-6 踢出拼车效果图

5.2.5删除拼车:用户进入自己发布的拼车订单详情页面，点击删除拼车，完成拼车的删除，如图5-7所示。



**图5-7 删除拼车效果图**

5.3叫车模块效果展示

用户进入叫车模块出现现在用车和预约用车两个选项，如图5-8所示。

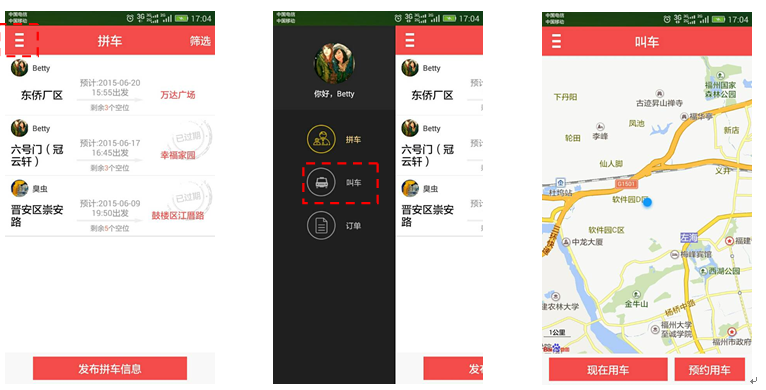


图5-8 叫车模块效果图

5.3.1现在用车：在叫车模块点击现在用车模块，可以选择叫车和坐他的车，叫车如图5-9所示，坐他的车如图5-10所示。

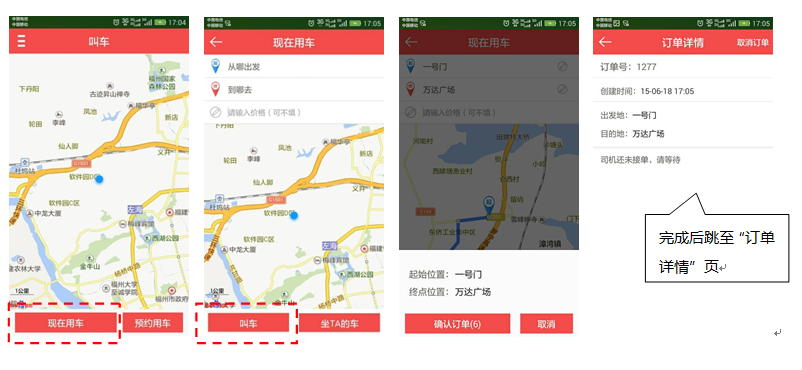


图5-9 及时叫车效果图



图5-10 坐他的车效果图

5.3.2预约用车：在叫车模块点击预约用车，可以进行车辆预约，如图5-11所示

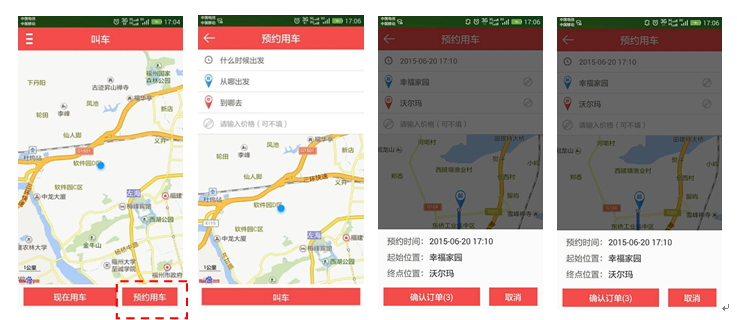


图5-11 预约用车效果图

5.4订单模块效果展示

5.4.1取消叫车：订单模块中点击想取消的订单进入订单详情页面，点击订单详情页右上角的取消即可取消叫车，如图5-12所示。

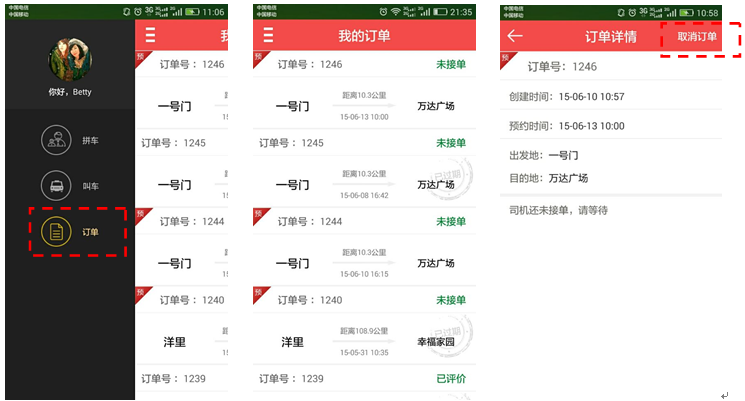


图5-12 取消叫车效果展示

5.4.2评价订单：订单模块设置评价功能，增强用户之间的互动。如图5-13所示

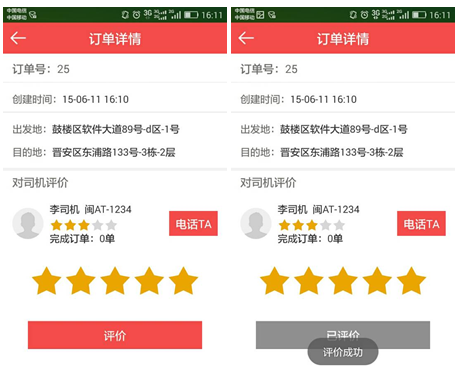


图5-13评价订单效果图

5.5个人信息模块效果展示

按图5-14流程进入个人信息模块

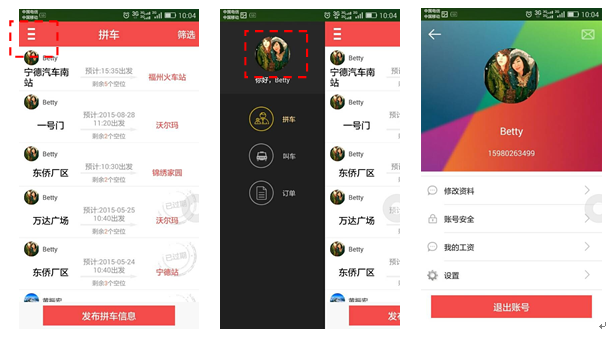


图5-14 进入个人信息模块流程图

5．5.1修改文字资料

个人信息模块提供修改文字资料功能，具体效果如图5-15所示

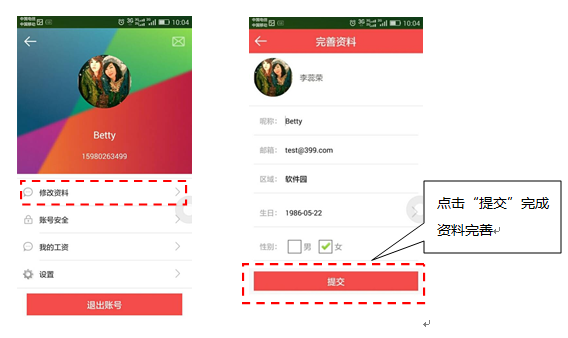


图5-15 修改文字资料效果

5．5.2更改头像

个人信息模块提供更改头像功能，具体效果如图5-16所示



图5-16 更改头像效果图

5.5.3修改密码

个人信息模块提供修改密码功能，具体效果如图5-17所示。

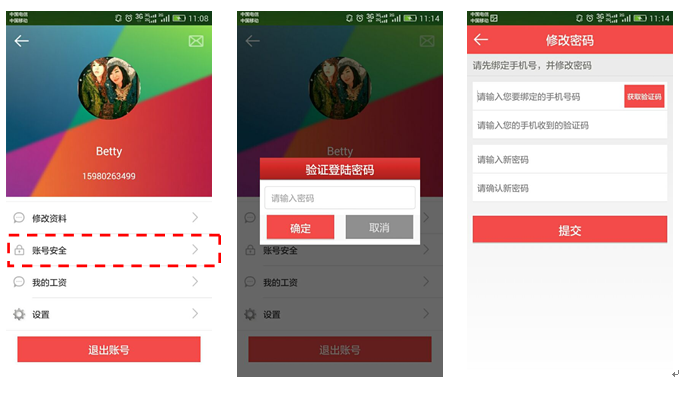


图5-17修改密码效果图

5.6本章小结

本章先对NC399出行的测试情况作出简述，然后通过展示拼车、叫车、叮当及个人信息四大模块的效果图来详细表现NC399出行的实现。