

学号：291400000

遼寧大學

毕业论文（设计）



题 目：____易道招车移动应用软件的设计与实现____

学 院：____信息学院____

专 业：____计算机科学与技术（软件方向）____

姓 名：____邓中华____

指导教师：____周老师____

完成日期：____2013 年 5 月 16 日____

摘 要

随着智能手机的普及，手机上的应用越来越多，打车应用却很少。有时乘客打车时需要等很长时间，尤其在上下班或雨雪天更难打车。同时，司机在寻找乘客时需要跑空车。实际上，导致此问题的根本原因是出租车与乘客分布不均匀。面对这样的问题，设计并实现一个招车软件，解决他们的问题，满足他们的需求，势在必行。

本文阐述了易道招车手机应用软件的设计、开发以及实现，该应用软件在 Android 操作系统上运行。该应用采用 C/S 结构，建立在 Internet 网络环境中。该应用软件分为 Android 客户端、服务器端、数据库和通信四大部分。客户端程序采用 Android 在 Eclipse 开发环境上编写；而服务器端程序使用 Java 语言在 MyEclipse 开发环境上编写；数据库采用 MySQL，使用 Navicat 工具对数据库进行管理。

Android 客户端包括登录，注册，设置，乘客模块，司机模块和 GPS 模块。登陆模块是首页，通过它可以进入注册、乘客和司机模块。注册模块为未注册的用户提供申请账号功能。设置模块提供用户信息管理功能。乘客和司机模块是利用百度地图 API 开发的，通过之间的交互完成招车功能。而乘客和乘客模块通过之间的交互完成拼车功能。GPS 模块负责经纬度的采集，为乘客和司机模块提供坐标信息，用来定位。

服务器端包括服务器程序和数据库工具模块。服务器程序通过数据库工具模块连接数据库，并完成初始化工作，准备应答客户端发来的请求。数据库存储用户信息以及打车过程信息。支持服务器程序对数据库的增、删、改、查等操作。通信模块分为客户端和服务端两个小部分。由于服务器发给客户端的数据比较多，需采用一种封装方法。服务器通过 Json 封装数据并通过 Http 协议发送给客户端，然后客户端再通过 Json 将收到的数据解封装。这样一来保证了数据的有效性。

在这期间，开发该系统所要做的需求分析、总体设计、Android 客户端界面和服务端程序的详细设计和测试等工作全部完成。实现的功能符合需求分析中的要求。调节出租车与乘客分布不均匀的现象，解决乘客打车时间长和司机跑空车的问题。满足乘客对打车、拼车和司机对载客的需求。

关键词：Android；Java；打车；地图；移动应用软件

Abstract

With the popularity of smart phones, there are more and more applications on the phone, however taxi applications rarely appear. Sometimes passengers need to wait a long time to take a taxi, especially when they go to work or get off work or in rain or snow. At the same time, drivers need to run for free when they look for passengers. In fact, the reason leading to the situation is the uneven distribution of taxis and passengers. Facing such problems, it is imperative to design and implement a software for taking a taxi which can solve their problems and meet their needs.

This paper elaborates the design, development and implementation of the Yidao mobile phone application. The application software runs on the Android operating system. It is built in the Internet environment using the C/S structure. The application software is divided into four parts including the Android client and server, database and communication. We use Android to write client programs on the Eclipse development environment, use Java language to write server programs on the MyEclipse development environment, use MySQL for database and use Navicat tools to manage the database.

Android client includes login module, registration module, setting module, passenger module, driver module and GPS module. Login module is the homepage, through which you can enter the registration module, passenger module and driver module. Registration module provides the function of applying account for the unregistered users. Setting module provides the management function for user's information. The passenger and the driver module are developed by Baidu Maps API. They complete taking a taxi by the interaction between the passenger and the driver module. The passenger and the passenger module complete carpooling by the interaction between them. GPS module is responsible for collecting longitude and latitude and providing coordinate information for the passenger and the driver module to locate.

The serve-side includes the server programs and the database tools module. The server program connects the database by means of the database tools module, then completes the initialization work and is ready to reply the requests from the client. The database that stores user information and taxi process information supports the server programs to add, delete, change and search on the database. Communication module is divided into two small parts of client and server. Since the server sending too much data to the client, we need to adopt a method of encapsulation. Server encapsulates data by Json and then sends data to the client by Http protocol. Then client de-encapsulate the received data by Json. So we can ensure the validity of the data in this way.

During this period, requirement analysis, overall design, the Android client interface, the detailed design and testing of server-side programs work all complete in the development of the system. The implemented functions conform to the demands in the requirement analysis. Yidao adjusts the phenomenon of uneven distribution between drivers and passengers and solves problems that passengers wait long and drivers run for free. Yidao also satisfies passengers to take a taxi and carpool and meets drivers' demands for passengers.

Key words: Android; Java; Taxi; Map; mobile application software

目 录

序 言	- 1 -
第 1 章 应用软件开发工具	- 2 -
1.1 Java 语言概述	- 2 -
1.2 Android 操作系统概述	- 2 -
1.3 Eclipse 概述	- 4 -
1.4 Tomcat 概述	- 4 -
1.5 MySQL 数据库概述	- 4 -
1.6 百度地图 API 概述	- 5 -
第 2 章 需求分析	- 6 -
2.1 用户需求及功能分析	- 6 -
2.2 业务流程	- 8 -
2.3 数据流分析	- 9 -
2.4 数据库概念模型的设计	- 11 -
第 3 章 总体设计	- 12 -
3.1 模块化程序设计	- 12 -
3.2 数据库设计	- 15 -
第 4 章 详细设计与实现	- 17 -
4.1 服务器端	- 17 -
4.2 Android 客户端	- 20 -
第 5 章 系统测试	- 27 -
第 6 章 总结	- 36 -
参考文献	- 37 -
致 谢	- 38 -

序 言

近些年来，手机硬件技术不断发展，从以前的只能打电话和发短信的低能手机，到现在的多功能智能手机。手机操作系统也经历几代的更替，从昔时称霸的诺基亚 Symbian 操作系统，到当下占领手机操作系统份额 68% 的 Google 公司的 Android 手机操作系统和苹果公司的 iOS 手机操作系统。

这些操作系统上的应用越来越多，目前已有 50 多万的应用，涉及生活、娱乐、教育等各个方面。如何才能满足人们日益膨胀的需求，IT 人需要走在用户的前面，因此，创新在 IT 领域是永恒的话题。

城市中有很多出租车，也有很多乘客，但是有时乘客打车时需要等很长时间，尤其在上下班时更难打车，想拼车，彼此不认识，更不知道别人的目的地，何谈拼车。同时，司机在寻找乘客时需要跑空车，费油，费时，殊不知此时其它地域有很多乘客但出租车很少。实际上，导致此问题的根本原因是出租车与乘客分布不均匀。面对这样的问题，设计并实现一个招车软件，解决他们的问题，满足他们的需求，势在必行。

根据尼尔森的报告，2013 年中国智能手机普及率高达 66%，用户主要分布在城市。为招车软件提供了大量潜在的用户群体。目前地图技术已经很成熟了，例如百度地图，谷歌地图都有开发工具包及 API 使用文档。为招车软件提供了软件支持。同时，目前智能手机都预装了 GPS 模块，定位快捷、方便，为招车软件提供了硬件支持。而且手机比其他智能设备更具有移动性，且把所有功能都集于一身的手机更能胜任。这样一来，在手机上开发一款打车应用能够满足人们迫切的需求。

第 1 章 应用软件开发工具

本应用软件以 Java 语言作为开发语言，使用 Eclipse 和 MyEclipse 作为开发工具，在 Eclipse 上开发 Android 程序，在 Myeclipse 上开发服务程序。服务器使用 Tomcat6.0.36，数据库使用 MySQL，使用 Navicat 对数据库进行图形化管理，并用 JDBC 方式与服务程序进行连接，同时使用 Json 封装数据并使用 Http 传输协议作为客户端和服务器的通信方式。以下对所使用技术进行详细说明。

1.1 Java 语言概述

Java 语言的风格与 C++语言非常相似。但 Java 是一个纯粹的面向对象的程序设计语言。它继承了 C++语言面向对象技术的核心，舍弃了 C++语言中容易引起错误的指针、运算符重载、多重继承等特性，增加了垃圾回收器功能用于回收不再被引用的对象所占据的内存空间，使得程序员不用再为内存管理而担忧。

Java 不同于一般的编译执行计算机语言和解释执行计算机语言。它首先将源代码编译成二进制字节码，然后依赖各种不同平台上的虚拟机来解释执行字节码。从而实现了“一次编译、到处执行”的跨平台特性。不过，每次的执行编译后的字节码需要消耗一定的时间，这同时也在一定程度上降低了 Java 程序的运行效率。

1.2 Android 操作系统概述

Android 一词本意是指“机器人”，当然也是众所周知的 Google 推出的开源手机操作系统。Android 操作系统采用 Linux 内核，号称是首个为移动终端打造的真正开放和完整的移动软件平台。Google 与 30 多家科技公司、手机公司组成“开放手机联盟”，共同研发。这将极大地降低新型手机设备的研发成本。图 1.1 为 Android 的 Logo。



图 1.1 Android 的 Logo

Android 系统架构分为应用层（Applications）、应用框架层（Application Framework）、系统运行库（C/C++库以及 Android 运行库）层（Library 和 Android Runtime）和 Linux 内核层（Linux Kernel）。系统架构如图 1.2 所示。

应用层上的应用程序采用 Java 语言编写，运行在虚拟机上。Google 一开始时就在 Android 系统上捆绑了一些核心应用，比如 E-mail、浏览器、短信、电话簿程序，等等。

应用框架层为软件开发人员提供 API 框架，使用这些框架来开发自己的应用，这样一来便简化了程序开发中的框架设计，但是必须遵守其框架的开发原则。

当开发人员使用 Android 应用框架时，Android 系统通过系统运行库来支持开发中使用的各个组件，为开发人员提供更好的服务。

Android 操作系统基于 Linux2.6 内核，如安全性、内存管理、进程管理、网络协议栈和驱动模型等都依赖于内核。Linux 内核也是硬件和软件栈之间的中间层。

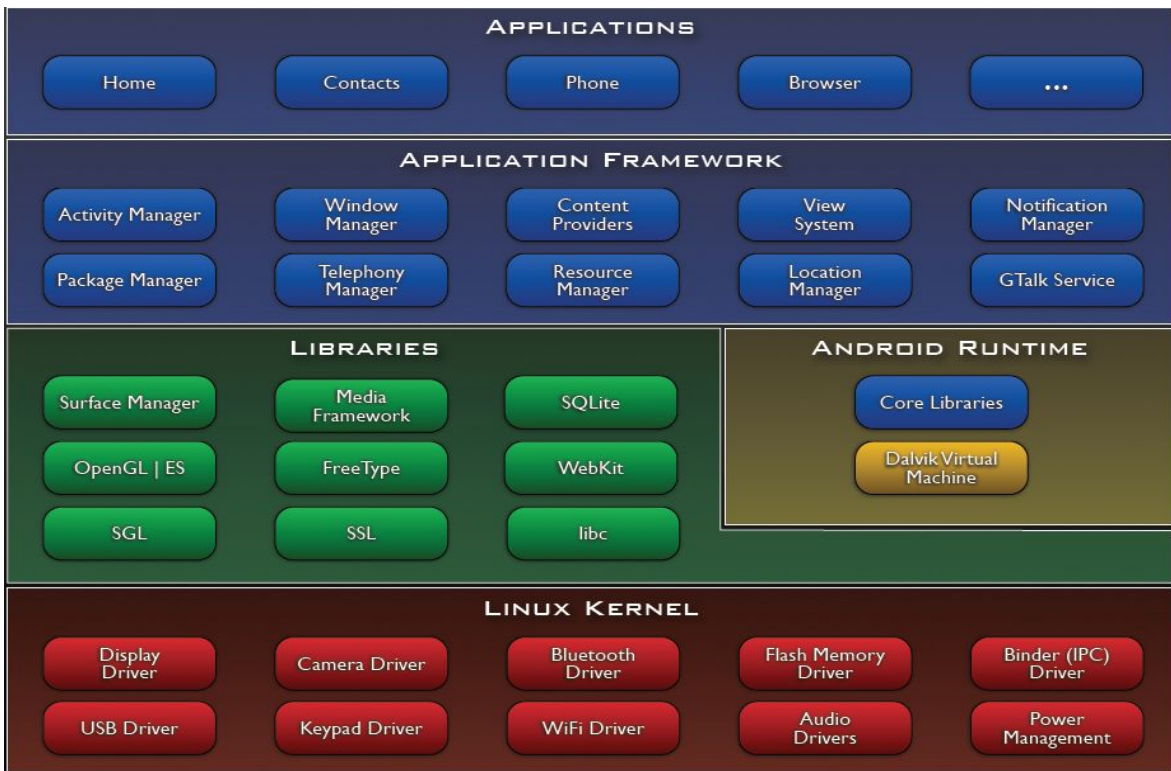


图 1.2 Android 系统架构

1.3 Eclipse 概述

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。它只是作为一个框架和一组服务，通过各个插件组件构建开发环境。Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Kit，JDK）。

虽然很多开发人员都愿意将 Eclipse 作为 Java 集成开发环境（IDE），但是 Eclipse 设计目的不仅仅是为 Java 设计的，而是面向所有开发语言而设计的，例如目前 Eclipse 还支持诸如 C/C++、JSP、PHP 等编程语言。Eclipse 还包括插件开发环境（Plug-in Development Environment，PDE），软件开发人员可以通过这个组件对 Eclipse 进行扩展，它允许开发人员构建与 Eclipse 环境无缝集成的工具。

1.4 Tomcat 概述

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器。Tomcat 是 Apache 软件基金会（Apache Software Foundation）的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。因为有 Sun 公司的积极参与和大力支持，最新的 Servlet 和 JSP 版本规范总是能在 Tomcat 中最先得到体现。Tomcat 不仅具有技术先进、性能稳定的特点，而且是免费的，深受 Java 开发人员的青睐并且获得到了大部分软件开发商的认可，成为目前主流的 Web 应用服务器。Tomcat 是一个轻量级应用服务器，适用于中小型系统和并发访问用户数量不大的场合。

Tomcat 具有很多优点，倍受广大开发人员的青睐，它在运行时所占用的系统资源小，易于扩展，支持负载平衡与邮件服务等，这些都是开发应用系统时常用的功能。并且 Tomcat 还在不断的改进和完善中，所有感兴趣并且有能力的开发人员都可以改动它或在其中加入自己开发的新功能。

1.5 MySQL 数据库概述

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前已经在 Oracle 公司旗下。关系型数据库将各类不同的数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样一来就加快了查询等操作的速度，并提高了数据管理的灵活性。与其他关系型数据库一样，MySQL 也采用数据库标准化的结构化查询语言（SQL）。MySQL 具有体积小、速度快、总体成本低等特点，尤其是开放源码这一特点，大部分中小型网站的开发都选择 MySQL 作为网站数据库。由于其性能卓越，搭配 PHP

和 Apache 可组成良好的开发环境。

1.6 百度地图 API 概述

百度地图 API 是为开发者免费提供的一套基于百度地图服务的应用接口，包括 JavaScript API、Web 服务 API、Android SDK、iOS SDK、定位 SDK、车联网 API、LBS 云等多种开发工具与服务，提供基本地图展现、搜索、定位、逆/地理编码、路线规划、LBS 云存储与检索等功能，适用于 PC 端、移动端、服务器等多种设备，多操作系统下的地图应用开发。

百度移动版地图 SDK，分为 Android 版与 Symbian 版，为移动设备地图应用开发提供基本地图、本地搜索、路线规划、定位等服务。若用户使用该套 SDK，需要先申请 key。

第 2 章 需求分析

2.1 用户需求及功能分析

随着智能手机的迅速发展，人们更希望将在电脑上能操作的功能移植到手机上，并且，手机还具有比电脑更好的移动灵活性。目前智能机有两大操作系统，分别为 Android 和 iOS。表 2.1 为 2012 和 2011 年手机操作系统的市场份额对比。

表 2.1 2012 和 2011 年手机操作系统的市场份额对比

操作系统	2012 年出货量 (万台)	2012 年市场 份额	2011 年出货量 (万台)	2011 年市场 份额	同比增长
Android	497.1	68.8%	243.5	49.2%	104.1%
iOS	135.9	18.8%	93.1	18.8%	46.0%
BlackBerry	32.5	4.5%	51.1	10.3%	-36.4%
Symbian	23.9	3.3%	81.5	16.5%	-70.7%
Windows Phone/Mobile	17.9	2.5%	9.0	1.8%	98.9%
Others	15.1	2.1%	16.3	3.3%	-7.4%
Total	722.4	100%	494.5	100%	46.1%

随着智能手机硬件性能的提高，手机应用软件也在不断的发展，应用数量每日剧增，目前所有手机操作系统的应用数量超过 50 万。IT 是需要创新的行业，要想创新就要走在用户的前面，为用户提供更好服务来满足人们日益扩大的需求。

城市中有很多出租车，也有很多乘客，但是有时乘客打车时需要等很长时间，尤其在上下班时更难打车，想拼车，彼此不认识，更不知道别人的目的地，何谈拼车。同时，司机在寻找乘客时需要跑空车，费油，费时，殊不知此时其它地域有很多乘客但出租车很少。实际上，导致此问题的根本原因是出租车与乘客分布不均匀。面对这样的问题，设计并实现一个招车软件，解决他们的问题，满足他们的需求。

本应用是基于 Android 操作系统开发的，设计目的是充分利用现有的计算机软、硬件资源，智能手机软、硬件资源和网络资源实现便于乘客打车，拼车和司机找乘客能在手机上完成。通过本应用上的地图显示，乘客和司机可以清楚的看到对方的位置和用户的手机号、姓名、信任度、目的地等，给用户提供更多有帮助的信息，有助于用户的选择。可以解决司机跑空车的问题和缩短了乘客的打车时间。打车成功后在应用地图上显示司机与乘客间的导航线路，便于司机能够快速找到乘客。打车完成后可以对乘客或司机进行评价，为其他用户提供有帮助的“信任度”信息，便于他人的选择。应用软件不仅要求有简单易学的操作界面，而且还要考虑到系统的安全性和稳定性，所以在注重界面美观、操作简易的基础上，还要特别注意对相关数据信息的储存。本文介绍了智能手机客户端和服务端端的详细设计与实现，根据用户的需求，该智能手机应用主要由以下功能组成：

登录功能，乘客和司机可以通过手机号和密码登陆到服务器，进入及时打车地图界面。当手机号或密码错误时提示用户登录失败。

注册功能，用户通过使用自己的手机号作为唯一标识，其他信息可以自行填写，让用户输入两次密码，保证密码的正确性，用户可以选择身份，当用户选择司机身份时，必须填写车牌号，车型信息；当用户选择乘客身份时，不必填写以上内容。

打车功能，该功能完成乘客和司机间的打车过程，乘客通过地图上显示的司机位置和其目的地，还剩座位数，信任度等信息来选择适合自己的司机，选择后发出打车请求，等待司机的响应，当司机确认后，在应用地图上，司机和乘客间显示导航路线，便于司机找到乘客和乘客观察司机位置。打车结束后可以给对方评价。

拼车功能，在应用地图上，乘客间彼此可以看到对方的位置和目的地等信息，当乘客发现有其他乘客和自己的目的地相同时，可以通过“呼叫”功能与其联系，同意后乘客们可以一起打车，进行拼车过程。

司机列表功能，该功能的目的是在实现与地图界面相同的功能下，列表界面会比地图界面节省流量，但是地图界面的效果会比司机列表界面的好些。该功能只为乘客服务，司机不享有该服务，当打开司机列表时，会显示车牌号和离乘客自己当前位置的距离，点击列表中的每一项时，会显示更多该司机的信息。

个人信息设置功能，该功能提供对用户自己信息的管理，其中包括用户手机号，昵称，密码，车牌号，车型，身份的修改和显示当前用户信息。

设置功能，司机可以修改当前车内乘客数和乘客们的目的地，乘客可以修改自己的目的地。便于乘客们拼车，和乘客与司机间的搭车。

定位自己功能，当用户自己的图标不在当前地图上显示时，可以通过定位自己功能，找到自己的位置。

2.2 业务流程

业务流程图是一种用尽可能少、尽可能简单的方法来描述业务处理过程的方法，能帮助我们了解该业务的具体处理过程，发现和处理系统调查工作中的错误和疏漏，修改和删除原系统的不合理部分，在新系统的基础上优化业务处理流程。

根据用户的需求，系统业务流程图如图 2.1 所示：

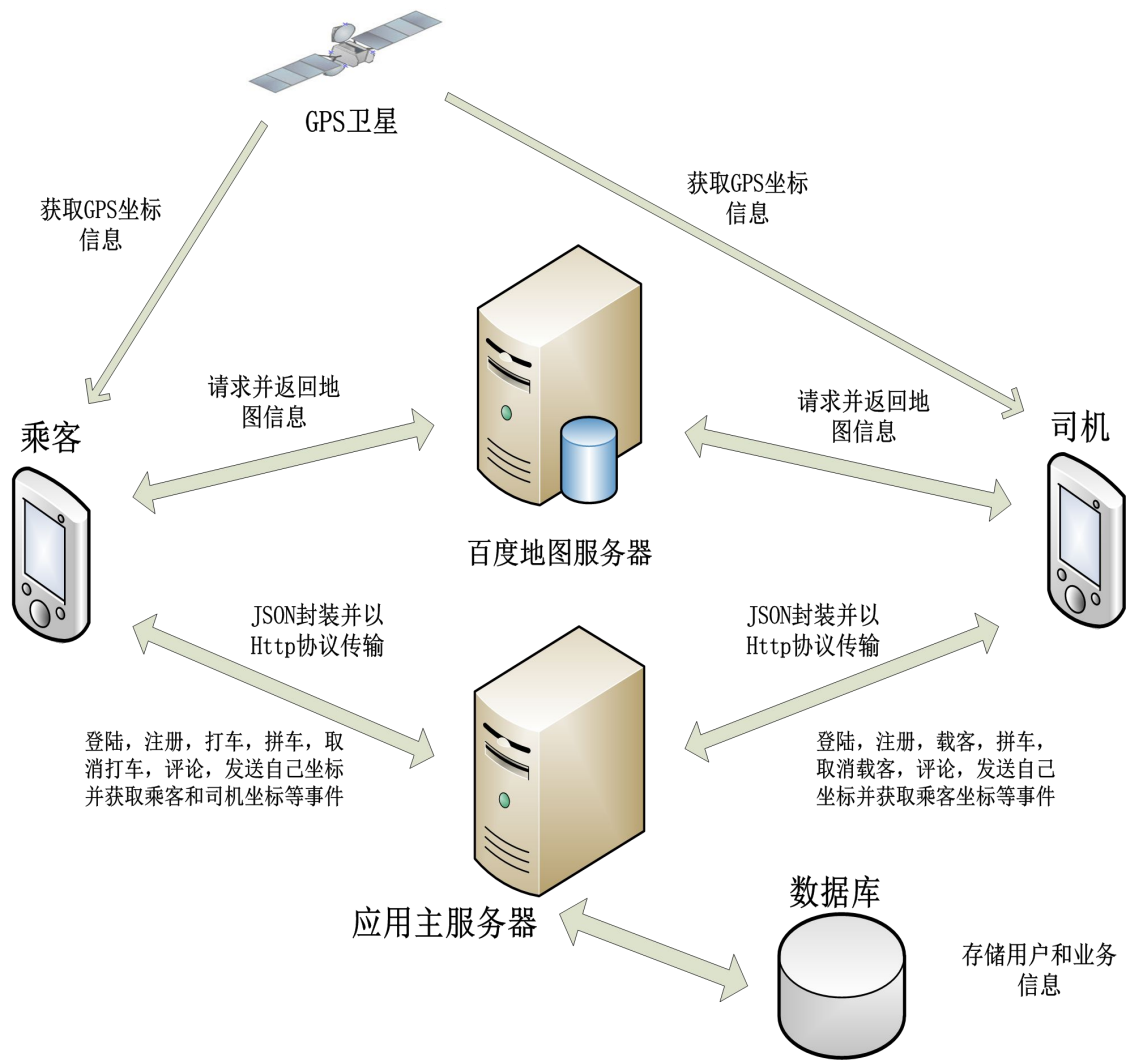


图 2.1 系统业务流程图

由图 3.1 可以直观地看出易道租车应用的工作流程：从乘客和司机登陆成功后开始，通过 GPS 模块获取 GPS 信息，访问百度服务器下载百度地图，进而可以进行打车，拼车等操作。

2.3 数据流分析

数据在软件系统中移动时，会经过一系列模块，它每经过一个模块后，会将被处理，修改后传给下一个模块。数据流图是一种图形化的技术，用来描绘信息流和数据从输入移动到输出的过程中所经受的变换。在数据流图中，没有任何具体的物理部件，它只是描绘数据在软件中流动和被处理的逻辑过程。通过数据流图，分析员描绘出对系统的认识或设想。

在本应用中，当用户首次使用时，通过自己的手机号注册一个账号，根据自己情况选择司机或乘客的身份，服务器接收请求后把新的用户信息存储到数据库中。乘客用户首先在 Android 客户端的登录对话框中输入登录所需的用户名和密码，向服务器发送登陆请求，服务程序接受请求后查找数据库，判断用户名和密码是否匹配，如果匹配成功则通过验证进入程序，之后乘客可以进行查看个人信息，修改个人信息和查看司机列表。司机与乘客一样，登陆成功后可以进行个人信息管理等操作。打车过程中，乘客首先向司机提出打车请求，打车中的过程信息存储在数据库中，由司机确认后完成打车。打车完成后双方给对方评价，记录在数据库中。在拼车过程中，由乘客向乘客提出拼车请求，拼车达成后共同打车。绘制出如图 2.2 的数据流图。通过数据流图，可进行进一步软件设计。

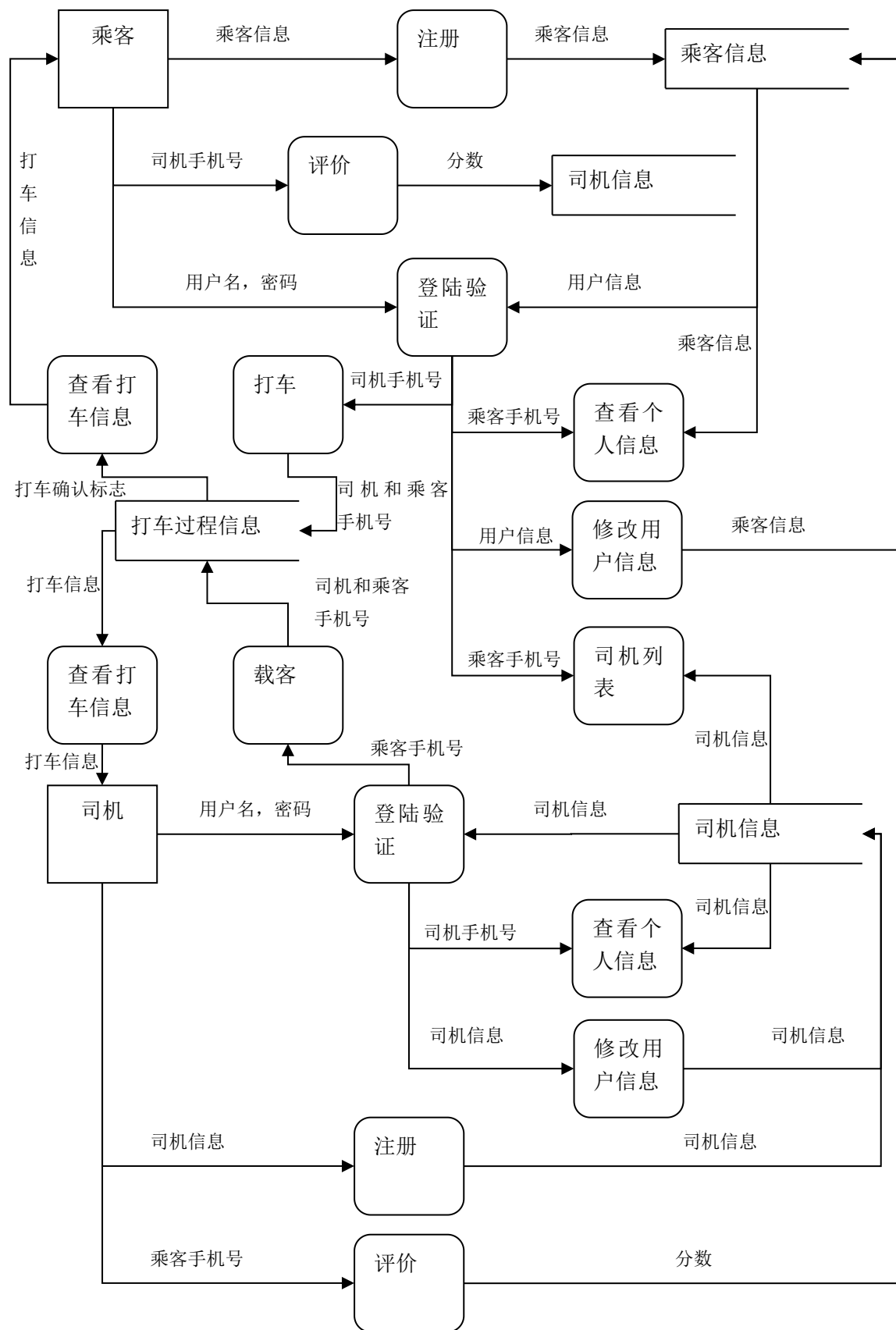


图 2.2 数据流图

2.4 数据库概念模型的设计

概念模型，也称信息模型，它是按用户的观点来对数据和信息建模，概念模型是现实世界到机器世界的一个中间层次。概念模型是数据库设计人员的有力工具，也是数据库设计人员和用户之间交流的语言，它清楚、准确地把用户的问题和需求反映出来。

概念模型的表示方法很多，其中 E-R 图（实体-联系图）的表示方法最为常用。在 E-R 图中包含了 3 中基本成分：实体、属性和关系。实体是具有一系列不同性质或属性的事物或概念。属性是实体的性质。联系是实体之间相互连接的方式，联系可以有属性。根据前边的用户需求及详细分析，可以建立 E-R 模型如图 2.3 所示，其中用户实体的属性如图 2.4 所示。

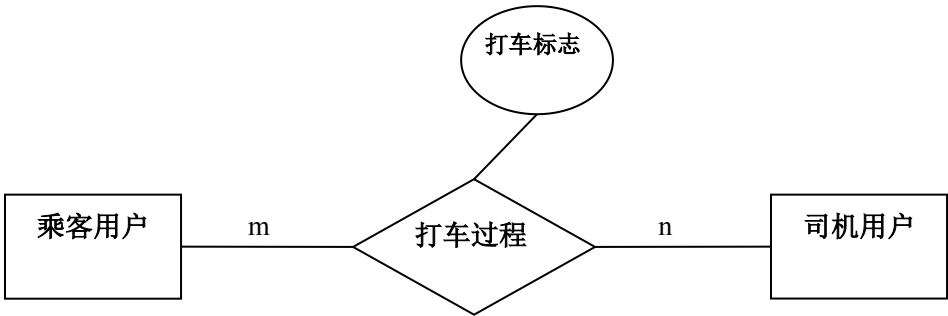


图 2.3 E-R 模型

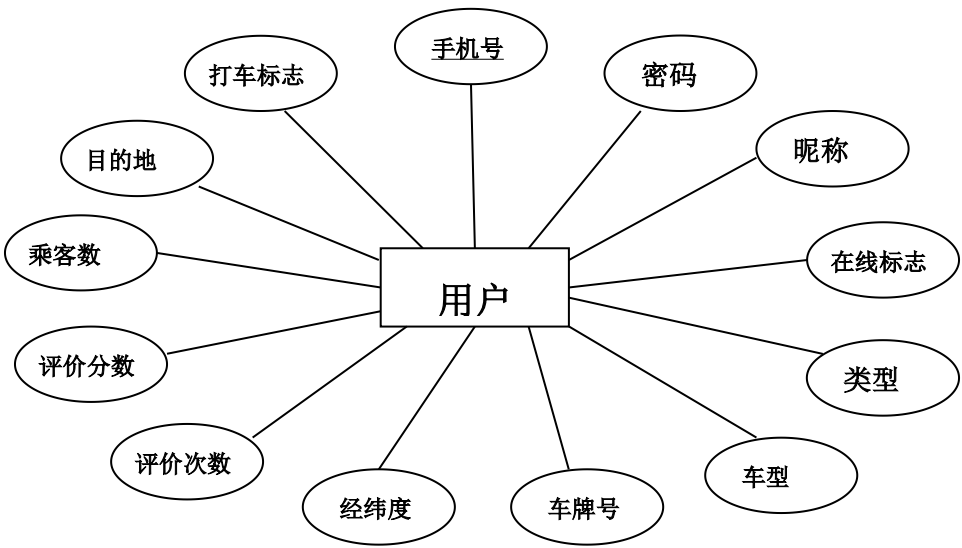


图 2.4 用户实体属性图

第 3 章 总体设计

3.1 模块化程序设计

本应用采用 C/S 结构，建立在 Internet 网络环境中。C/S 模式主要由客户应用程序(Client)、服务器管理程序(Server)组成。客户应用程序是系统中用户与数据进行交互的部件。服务器程序负责有效地管理系统资源，如管理一个信息数据库，其主要工作是当多个客户并发地请求服务器上的相同资源时，对这些资源进行最优化管理。本应用中，主服务器为服务器端，Android 应用为客户端。

易道租车应用包括四大模块：android 客户端，服务器程序，数据库和通信模块。Android 客户端包括登录，注册，设置，乘客界面，司机界面，GPS 模块和司机列表模块；服务器模块包括服务器程序和数据库工具模块。如图 3.1 所示。

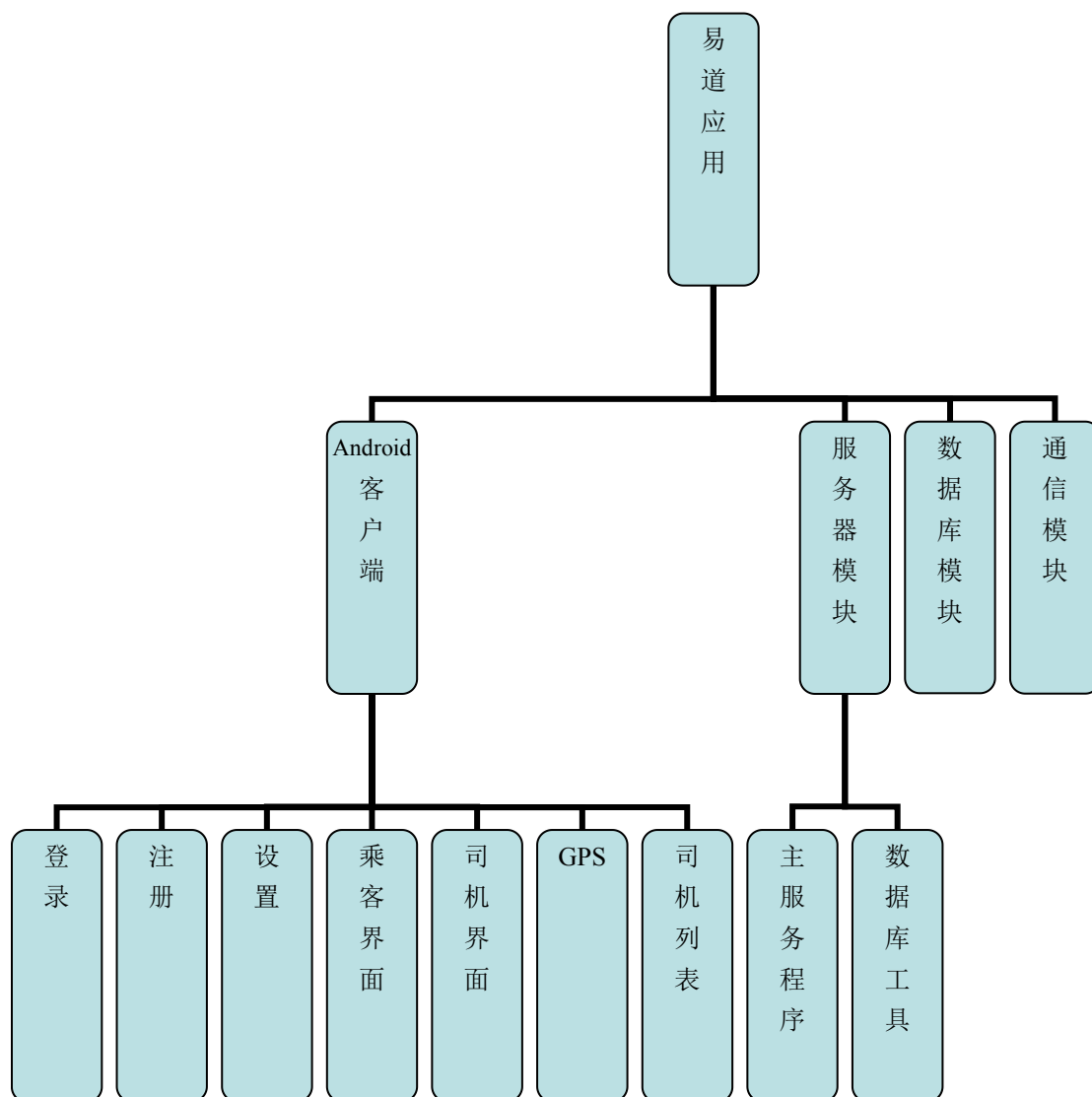


图 3.1 功能模块图

1、登录模块

用户运行易道招车应用软件，首先进入到登陆界面，在当前界面用户输入注册时用的手机号和密码进行登录，若登陆成功，则从当前页面跳转到地图定位页面，若登陆失败弹出对话框提示登录失败。当用户点击注册按钮是，则从当前界面跳转到注册会员界面。

2、注册模块

新用户安装易道招车应用后，运行该软件，进入登陆界面，点击注册按钮就进入注册界面了，用户在此界面那要求填写个人信息，如果用户没按要求填写信息，会弹出相应的提示，当用户填写完整后，点击提交，若注册成功，弹出一个对话框提示用户注册成功，此时并将用户的个人信息存放在数据库中。若注册失败，弹出对话框提示用户注册失败，此时用户点击取消按钮，可以重新注册。

3、设置模块

实现用户修改自己信息的功能。当用户登录成功后，点击 MENU 键，在弹出的 MENU 菜单中选项“个人信息”，便可进入用户信息设置界面，当用户点击某一项信息的修改按钮时，会弹出一个对话框，用户在此对话框中输入新的信息，点击确定按钮，如果修改成功会弹出提示，告诉用户修改成功，此时修改后的信息也显示在相应的项中，如果修改失败，这提示用户修改失败。当用户修改完信息后，点击返回，这返回到地图定位界面。

4、乘客界面

乘客界面是一个地图界面，通过 GPS 模块获取自己的经纬度，并定位乘客自己的位置，同时显示司机和其他乘客用户的坐标位置。当点击司机用户图标时，弹出一个对话框，显示司机的昵称，手机号，车牌号，车型，信誉度等信息和三个按钮，分别是“打车”，“呼叫”和“取消”；当点击“打车”时，客户端向服务器发送打车请求，并且当前地图界面中只显示该司机和其他乘客的位置信息，等待该司机的答复。如果该司机应答了乘客的打车请求，那么会显示司机到乘客之间导航线路信息；如果该司机没有应答乘客的请求，并且应答了别的乘客，那么乘客界面就重新回到初始界面，显示所有司机和乘客的位置信息，打车请求被取消。当点击“呼叫”按钮时，会调启 android 打电话界面，并呼叫司机；当点击“取消”按钮时，对话框消失，返回之前界面。

当点击 MENU 键时，会显示出 5 个按钮，分别是“定位自己”，“打车”，“设置”，“个人信息”和“司机列表”。下面分别介绍一下它们的功能。当点击“定位自己”按钮时，乘客界面会以自己位置为中心调整地图显示；当点击“打车”按钮时，如果当前显示司机到乘客的导航路线时，会弹出一个对话框，显示“确定重新打车？”，当点击“确定”按钮时，又弹出一个对话框，显示“是否进行评价？”，当点击“是”按钮时，弹出评价界面，评价后返回到初始地图界面。这几个对话框中都有“取消”按钮，点击后返回对应的之前界面；当点击“设置”按钮时，弹出修改目的地的界面，确定后返回；当点击“个人信息”按钮时弹出修改个人信息界面，乘客可以修改自己的基本信息。当点击“司机列表”时，显示司机的列表，包括司机的车牌号和距离乘客自己的距离。当点击列表中的每一项时，会显示该司机的具体信息。

5、司机界面

司机界面也是一个地图界面，通过 GPS 模块获取自己的经纬度，并定位司机自己的位置，当没有乘客向司机自己发出打车请求时，司机界面会显示所有乘客用户位置，当有一个或多个乘客向司机自己发出打车请求时，那么司机界面会显示这些乘客的位置，并等待司机的确认。当点击乘客用户图标时，弹出一个对话框，显示乘客的昵称，手机号，信誉度等信息和三个按钮，分别是“载客”，“呼叫”和“取消”；当点击“载客”时，客户端向服务器发送载客请求，成功后会显示司机到乘客之间导航线路信息；当点击“呼叫”按钮时，会调启 Android 打电话界面，并呼叫乘客；当点击“取消”按钮时，对话框消失，返回之前界面。

当点击 MENU 键时，会显示出 4 个按钮，分别是“定位自己”，“载客”，“设置”和“个人信息”。下面分别介绍一下它们的功能。当点击“定位自己”按钮时，司机界面会以自己位置为中心调整地图显示；当点击“载客”按钮时，如果当前显示司机到乘客的导航路线时，会弹出一个对话框，显示“确定重新载客？”，当点击“确定”按钮时，又弹出一个对话框，显示“是否进行评价？”，当点击“是”按钮时，弹出评价界面，评价后返回到初始地图界面。这几个对话框中都有“取消”按钮，点击后返回对应的之前界面；当点击“设置”按钮时，弹出修改乘客数和目的的界面，确定后返回；当点击“个人信息”按钮时弹出修改个人信息界面，司机可以修改自己的基本信息。

6、GPS 模块

通过 GPS 模块，获取 GPS 经纬度信息，并且实时更新。将经纬度信息传送给司机和乘客界面。

7、主服务程序

服务程序是由 Java 编写，是 Servlet 程序，运行在 Tomcat 服务器上，可以同时响应大量用户的访问，负责响应并处理所有客户端发送过来的请求，例如登陆，注册，设置，打车或载客等请求，代替用户对数据库进行增、删、改、查等操作。

8、数据库工具

数据库工具由 Java 编写，是服务程序与数据库之间的桥梁。负责连接数据库（包括定义数据库用户名，密码，数据库驱动，数据库的地址，定义 SQL 语句执行的对象，定义查询返回的结果集合），增、删、改、查的底层支持和资源释放等操作。

9、通信模块

通信模块是客户端和服务端之间的桥梁，负责客户端与服务端之间的信息交互，数据是使用 Json 封装，并通过 Http 协议传输。

10、数据库模块

数据库使用的是 MySQL 数据库，是一个关系型数据库管理系统，由于其体积小、速度快、总体拥有成本低，一般中小型网站的开发都选择 MySQL 作为网站数据库。数据库存储了用户信息和打车过程信息。

11、司机列表模块

当乘客登陆成功后，首选进入地图定位界面，此时，点击 MENU 键，在弹出的菜单选项中点击“司机列表”按钮，就会弹出一个列表，此时列表列出了在线司机的车牌号和离当前用户的距离，更方便用户选择司机，同时也为用户省去一些流量。当用户点击列表的每一个 Item 时，会弹出一个对话框，此对话框显示了司机的基本信息和信任度，当用户想打此车时，用户可以点击呼叫按钮联系司机，如果不想则点击取消，返回列表，可以继续查看其他司机信息。

3.2 数据库设计

本应用软件使用 MySQL 数据库。首先创建数据库实例 carservice，根据系统的具体要求，由数据库概念模型设计阶段得到的 E-R 图可得到系统所需要的数据表。表 3.2、3.3 为程序主要功能所需数据库表，说明书部分所使用的表将在后边介绍，这

里不再赘述。

表 3.2 用户信息表 (userinfo)

列名	类型	字段含义
id	Int	自动增长，作为主键
phonenum	varchar(32)	手机号码，作为主键
password	varchar(32)	用户密码
name	varchar(32)	用户昵称
online	Int	用户是否在线
type	Int	用户类型
longitude	Int	经度
latitude	Int	纬度
carnum	varchar(32)	车牌号
cartype	varchar(32)	车型
appraisetimes	Int	评价次数
appraisescore	Double(11, 2)	评价分数
destination	varchar(32)	目的地
sit	varchar(32)	乘客数
success	varchar(32)	打车标志

表 3.3 打车过程表 (trade)

列名	类型	字段含义
phonenum_p	varchar(32)	乘客手机号，作为主键
phonenum_c	varchar(32)	司机手机号，作为主键
success	varchar(32)	打车标志

第 4 章 详细设计与实现

4.1 服务器端

所有服务器端的程序都是在 MyEclipse 下完成编写与测试的,其中包括主服务程序,对数据库进行操作的工具类以及数据封装类。服务器端程序结构如图 4.1 所示。下面分别详细介绍。

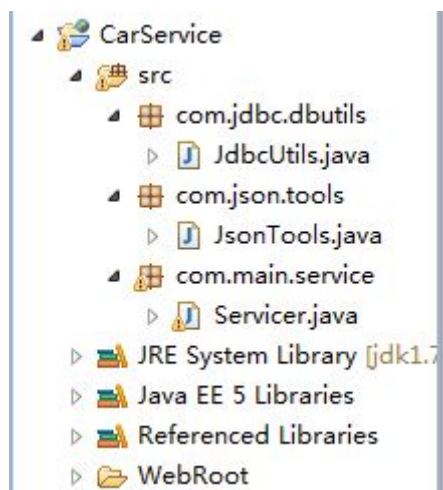


图 4.1 服务器端程序结构

4.1.1 主服务程序的设计

主服务程序是一个 Servlet 类,命名为 Servicer,主要负责接收客户端发来的 Http 请求,并且根据不同的类型请求进行相应的处理, Servicer 类图如图 4.2。

Servicer
jdbcutils connection
Servicer() init() destroy() doGet (HttpServletRequest, HttpServletResponse) doPost (HttpServletRequest, HttpServletResponse)

图 4.2 Servicer 类定义

当 Servicer 程序运行时,首先要进行初始化,在 init() 方法中创建数据库操作

类的对象，并获取对数据库的链接。然后服务程序等待客户端发来 Http 请求。当接收到 Post 方式请求后，服务程序会调启 doPost() 方法，当接收到 Get 方式请求后，服务程序会调启 doGet() 方法，在 doGet() 方法中又调用 doPost()，所以把所有的功能都集中写在了 doPost() 方法中，而 doGet() 只起到了承上启下的作用。在接收到的 Http 请求中有一个字段 action_flag，它是请求类型的标志，服务程序就是根据它的值来进行相应处理的。

action_flag 的值有 15 种，换言之，服务器接受 15 种请求，如表 4.1 所示。

表 4.1 服务器接收的请求类型

action_flag	请求内容	action_flag	请求内容
position_p	乘客修改自己坐标，和获取司机的坐标	cancelpick	处理司机取消原有载客请求
position_c	司机修改自己坐标，和获取乘客的坐标	driver_setting	处理司机修改乘客数和目的地请求
position_pf	处理查询拼友	selfinfo	处理个人信息请求
login	处理登陆请求	appraise	处理评价请求
signin	处理注册请求	settings	处理用户设置请求
call	处理乘客打车请求	carlist	处理列表请求
cancelcall	处理乘客取消原有打车请求	passenger_setting	处理乘客修改目的地请求
pick	处理司机载客请求		

下面以 login 请求为例介绍一下服务器程序的工作流程，接收到请求后，提取出 action_flag 的值为 login，进而提取出 Http 请求中的用户名和密码，调用 JdbcUtils 中的 loginFind() 方法进行访问数据库，如果用户名和密码匹配，服务器会返回给客户端一个登陆成功标志，如果用户名和密码不匹配，服务器会返回给客户端一个登陆失败标志。

4.1.2 数据库操作工具类的设计

建一个专门进行数据库操作的 JdbcUtils 类，当主服务程序 Servicer 启动后，对 JdbcUtils 类进行初始化，首先创建这个连接类的对象，然后进行数据库操作，完成操作后，再使用该对象断开数据库连接。JdbcUtils 类图如图 4.3。

JdbcUtils
USERNAME PASSWORD DRIVER URL connection pstmt resultSet
JdbcUtils() getConnection() updateByPreparedStatement(String, List<Object>) findMoreResult(String, List<Object>) findSimpleResult(String, List<Object>) loginFind(String, List<Object>) releaseRes()

图 4.3 JdbcUtils 类定义

本应用使用 MySQL 数据库，使用 JDBC 进行连接。创建连接的代码如下。

```

public JdbcUtils() {
    try {
        Class.forName(DRIVER);
        System.out.println("注册驱动成功!!! ");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}

//定义获得数据库的链接
public Connection getConnection() {
    try {
        connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```

        return connection;
    }

```

JdbcUtils 类中还封装了对数据库基本操作的方法，其中负责增、删、改的方法是 updateByPreparedStatement ()，提供查询功能的有负责查询多条记录的 findMoreResult ()，负责查询单条记录的 findSimpleResult () 以及负责登陆查询的 loginFind () 方法。

4.1.3 服务器端数据封装的设计

负责对从服务器端到客户端传输数据的封装的类是 JsonTools。其中的 createJsonString () 方法接受两个参数，key 用来标识 value，把它们按照 Json 形式封装，当客户端接受这个数据时同样用 Json 方式解封装，还原数据，JsonTools 类图如图 4.4。

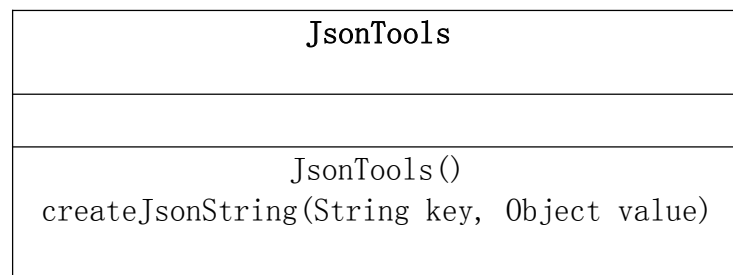


图 4.4 JsonTools 类定义

4.2 Android 客户端

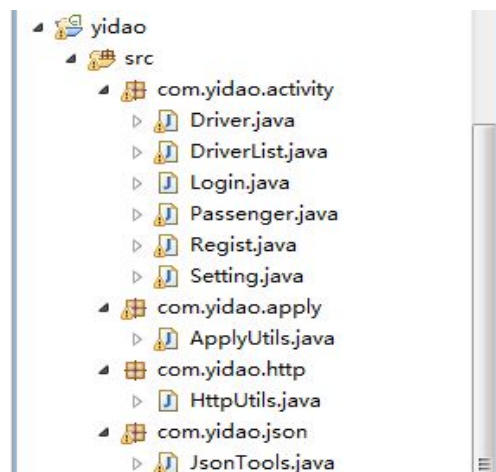


图 4.5 客户端程序结构

所有客户端程序都是在 Eclipse 下完成编写与测试的，其中包括 6 个 Activity 和 3 个自己封装的 Java 工具类，如图 4.5 所示，下面分别详细介绍。

4.2.1 请求工具类的设计

请求工具类 (ApplyUtils) 的设计目的是把所有客户端对服务器的请求处理方法都封装在一个类中，其他类在使用时，可以直接调用，而不必在每个类中都声明一个同样的方法，避免了代码的重复，同样也屏蔽掉了上层对底层的操作，便于开发和维护。

该工具类中包含 12 请求处理接口，其中包括登录申请接口 ApplyLogin(String phonenum, String password)，打车请求处理接口 ApplyCall(String phonenum_p, String phonenum_c) 和评价申请接口 ApplyAppraise(String phonenum, int score) 等，这里不一一列举，它们会在后续介绍的模块中使用，再详细介绍。

下面的是登录申请接口的具体实现。

```
public static int ApplyLogin(String phonenum, String password) {  
    String path =  
    "http://192.168.1.103:8080/CarService/servlet/Service?action_flag=login&  
    phonenum=" + phonenum + "&password=" + password;  
    String jsonstring = HttpUtils.getJsonContent(path);  
    String result = JsonTools.getJsonString(jsonstring, "log");  
    if(result.equals("passenger")) {  
        return 1;    //表示乘客登陆成功  
    }  
    if(result.equals("driver")) {  
        return 2;    //表示司机登陆成功  
    }  
    return 0;        //表示登录失败  
}
```

4.2.2 登录和注册模块的设计与编写

当应用开启后，首先进入登录界面（Login Activity），界面中有两个文本域（EditText）分别是用户名（username）和密码（password），还有两个按钮（Button）分别是登录（login）和注册（Register）。

在登录按钮下设置监听器，当点击登录按钮后，执行以下操作，获取用户名和密码信息，并分别保存在 username_value 和 password_value 局部变量中，获取方法以用户名为例，String username_value=username.getText().toString()。如果用户名和密码的值都不为空，那么调用请求工具类中登录申请接口。

```
int ApplyLogin(String phonenum, String password);
```

当返回的值为 1 时表示乘客登录成功，当返回 2 时表示司机登陆成功，登陆成功后进行界面跳转，进入地图界面，并在意图中添加手机号码信息，实现代码以司机为例。

```
Intent intent_Passenger = new Intent(Login.this, Driver.class);  
intent_Passenger.putExtra("phonenum", username_value);  
startActivity(intent_Passenger);
```

当返回 0 时表示登录失败，弹出登录失败对话框。如果用户名和密码有一个以上为空时，提示用户输入相应的内容。

在注册按钮下设置监听器，当点击注册按钮后，界面跳转到注册界面。实现跳转的主要代码如下。

```
Intent intent = new Intent(Login.this, Regist.class);  
startActivity(intent);
```

当用户首次使用本应用时，需要注册一个账号，再登陆界面点击注册按钮后进入注册界面（Regist Activity），注册界面的实现与登陆界面相似，但是比登陆界面多了些功能，例如需要输入两次密码，来确认密码是否正确，以及用户要选择乘客或司机身份，当用户选择乘客身份时，车牌号和车型的文本域不可输入，当用户选择司机身份时，必须输入车牌号和车型。

在提交按钮设置监听器，当点击提交按钮后，获取用户名，密码和昵称等信息，并调用 ApplyUtils 中的 int ApplySignin(String phonenum, String password, String name, int type, String carnumber, String cartype)，当返回 0 时表示注

册失败，并弹出注册失败对话框；当返回 1 是表示注册成功，并弹出注册成功对话框，点击确定后，跳转到登陆界面。登陆和注册的流程图如图 4.6 所示。

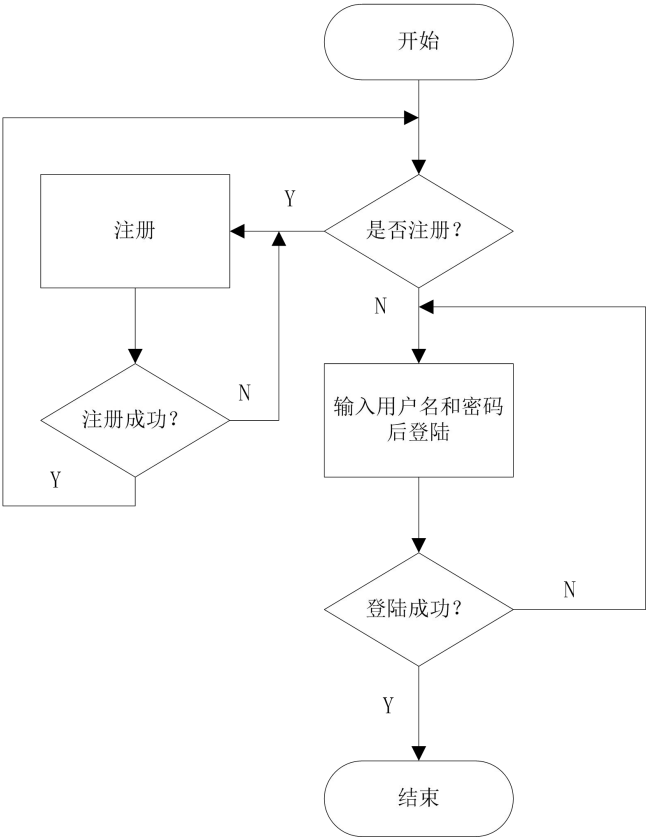


图 4.6 登陆和注册模块流程图

4. 2. 3 地图模块的设计与编写

地图模块分为乘客和司机两个界面，主要负责打车，载客，拼车和搭车功能。乘客与司机地图界面相似，但是乘客界面比司机界面复杂些。这里只介绍乘客地图界面（Passenger Activity）的设计与实现。

本应用使用的是百度地图 API，使用前需要申请并注册百度地图 key，下载百度地图 API 的工具包，并加载。

乘客地图界面中有几个重要的对象必须创建，其中包括 MapView，BMapManager，MapController，LocationManager 和 List<Overlay>等。下面分别介绍一下它们的作用。MapView 是地图对象，用来显示百度地图的控件，以及所有地图上的其他控件都是在其上添加。BMapManager 是百度地图的引擎，用来加载百度地图 key 以及其他核心功能。MapController 是百度地图的控制着，用于控制地图的缩放（zooming）和

平移 (panning)。LocationManager 是位置管理工具，负责 GPS 定位方面的功能。List<Overlay>是覆盖物的列表，用于在地图上显示自定义信息，包括图片，文字等，同时在这些信息上可以添加点击事件。

乘客地图界面的流程图如图 4.7 所示。乘客登陆成功后，乘客地图界面接受登陆界面传来的电话号码信息，实现代码是，`Intent intent = getIntent();phonenum = intent.getStringExtra("phonenum")`，然后在乘客地图界面中创建 MapView，BMapManager，MapController，LocationManager 和 List<Overlay>等对象，并初始化。

乘客地图界面中设有 4 个线程，除了主线程外有寻找司机线程，寻找拼友线程和刷新地图线程。寻找司机线程和刷新线程共享一个存储司机信息的列表 List<Map<String, Object>> list，寻找拼友线程和刷新线程共享一个存储乘客信息的列表 List<Map<String, Object>> list_pf。

主线程负责控制 Activity 的生命周期，初始化时调用 onCreate()，销毁时调用 onDestroy()，重新载入时调用 onResume()，暂停时调用 onPause()方法。寻找司机线程负责每 5 秒钟向服务器发送自己的坐标并接受发回来的司机信息，检查 list 是否正在被使用，如果没有被使用，则清空 list 并将新获取到的司机信息添加到 list 中。同样地，寻找拼友线程负责每 5 秒钟向服务器发送请求并接受发回来的乘客信息，检查 list_pf 是否正在被使用，如果没有被使用，则清空 list_pf 并将新获取到的乘客信息添加到 list_pf 中。刷新图层线程每 6 秒检查一下 list，看是否正被使用，如果没有被使用，读取 list 中的司机信息，并以覆盖物图层形式添加到地图界面上，在地图上相应位置显示司机图标。然后检查 list_pf，看是否正被使用，如果没有被使用，读取 list_pf 中的乘客信息，并以覆盖物图层形式添加到地图界面上，在地图上相应位置显示乘客图标。

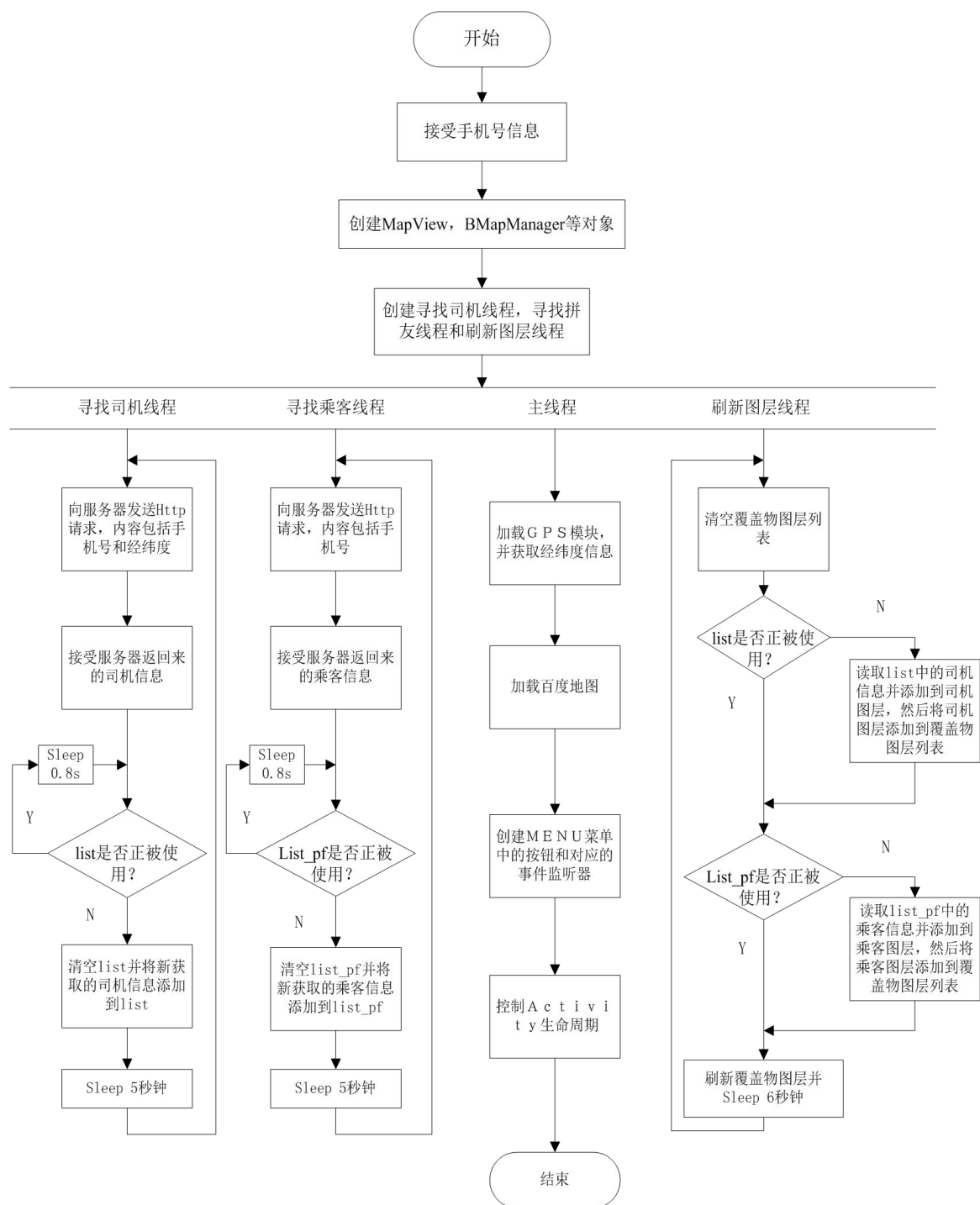


图 4.7 乘客地图模块流程图

4.2.4 司机列表模块的设计与编写

只有乘客具有司机列表 (DriverList Activity) 功能, 而司机不具有此功能, 司机列表的设计目的在打车体验不受较大的影响下, 节约乘客的流量。司机列表的开启方式是在乘客地图界面点击 MENU 键后点击“司机列表”按钮后进入的, 乘客地图

界面给司机列表界面通过意图传递手机号和经纬度三个参数。司机列表界面接受三个参数后，调用 ApplyUtils 中的 ApplyCarList() 方法, 向服务器发送查找司机信息请求，服务器会返还所有其周围的司机的信息，包括手机号，昵称，经纬度，信任度等。然后根据经纬度分别计算每个司机与自己的距离，实现方法如下。

```
private double Comput_distance(int lat_a, int lng_a, double lat_b, double lng_b)
{
    double radLat1 = (lat_a * Math.PI / 180.0);
    double radLat2 = (lat_b * Math.PI / 180.0);
    double a = radLat1 - radLat2;
    double b = (lng_a - lng_b) * Math.PI / 180.0;
    double s = 2 * Math.asin(Math.sqrt(Math.pow(Math.sin(a / 2), 2)
        + Math.cos(radLat1) * Math.cos(radLat2)
        * Math.pow(Math.sin(b / 2), 2)));
    s = s * EARTH_RADIUS;
    s = Math.round(s * 10000) / 10000;
    return s;
}
```

最后在司机列表界面显示一个列表，每条中显示司机的车牌号和距离。点击每条记录后弹出一个对话框，显示司机的详细信息。

第 5 章 系统测试

系统测试就是为了发现错误而执行程序的过程。在这里将采用黑盒测试，模拟用户的正常使用过程。采用如图 3.1 所示的结构进行测试。

选择一台性能好的服务器作为主应用服务器，在主应用服务器中配置 Java 环境并安装 Tomcat 和 MySQL 数据库，把 Java 服务器程序部署到 Tomcat 上，并将系统的数据库信息添加进去，在多个 Android 手机上安装易道应用程序。本文介绍司机与乘客间打车的运行过程和乘客与乘客间的及乘客与司机间的拼车过程。在手机上点击“易道”应用图标，如图 5.1 所示。会进入易道首界面，有两个功能，分别是登陆和注册，图 5.2 所示



图 5.1 易道图标



图 5.2 登陆界面

如果用户首次使用本应用，那么需要注册一个账号，点击“注册”按钮进入注册界面，如图 5.3 所示。输入正确的内容后点击“提交”按钮，会弹出注册成功提示框，如图 5.4 所示；如果输入的内容有误，例如手机号码已经存在，会弹出注册失败对话框，如图 5.5 所示。



图 5.3 注册界面



图 5.4 注册成功界面



5.5 注册失败界面

当用户输入的信息不正确时，会提示用户信息不正确，无法登录，如图 5.6 所示。如果用户输入正确的用户名和密码登录则会进入两种界面，分别是司机和乘客两种界面，根据手机号码查询该用户是乘客还是司机身份后进入不同界面。如图 5.7 和 5.8 所示。

在乘客界面中除了显示自己的位置外，还显示乘客和司机的位置。蓝色圆点代表乘客自己，蓝色小人代表乘客，黑色汽车代表司机。



图 5.6 用户信息不正确



图 5.7 乘客登陆成功界面

在司机界面中同样除了显示自己的位置外，还只显示乘客的位置，而不显示其他司机的位置。同样地，蓝色圆点代表司机自己，蓝色小人代表乘客。

乘客登陆成功后，想打车时，选择其周围较近的司机图标，点击司机图标后，弹出一个对话框该司机的信息，可以看到司机的手机号，信任度，当前车上乘客数及他们的目的地，再根据乘客自身情况选择合适的司机，当找到合适的司机后点击“打车”按钮，发出打车请求，如图 5.9 所示。



图 5.8 司机登陆成功后界面



图 5.9 司机详细信息界面

乘客选择一个司机后，点击“打车”按钮发出打车请求后，在 15242044196 这个司机界面会显示如图 5.10 所示。图中两个蓝色小人表示有两名乘客请该司机提出了打车请求。司机可以根据他们的目的地选择其中的一个或多个乘客。



图 5.10 有乘客向司机请求打车



图 5.11 司机选择乘客

如果该司机选择了 15242044192 这个乘客，如图 5.11 所示。点击“载客”按钮后，地图上回出现司机到乘客的导航路线，便于司机找到乘客。如图 5.12 所示。

这时乘客界面也显示出冲司机到乘客间的导航路线，便于乘客发现司机的位置。如图 5.13 所示。



图 5.12 乘客打车成功



图 5.13 司机载客成功

当打车结束后乘客点击 MENU 键，弹出菜单，点击“打车”按钮后会弹出是否要重新打车对话框，选择“是”后会有弹出是否进行评价对话框。如图 5.14-5.17 所示。



图 5.14 MENU 菜单



图 5.15 重新打车对话框



图 5.16 是否评价提示框



图 5.17 评价对话框

同样地，当载客结束后，司机点击 MENU 键，弹出菜单，点击“载客”按钮，重新载客。如图 5.18。后续与乘客相似，这里不再赘述。



图 5.18 司机 MENU 菜单



图 5.19 乘客详细信息

当乘客想拼车时，乘客可以在地图上其周围的乘客图标，如果该乘客的目的地与自己的一致，可以点击呼叫按钮与其联系，一起打车，节省费用，实现拼车功能。如图 5.19 所示。

为了节省流量，乘客可以使用司机列表代替地图来打车，乘客点击 MENU 后，点击“司机列表”会进入，司机列表界面，如图 5.20 所示。



图 5.20 司机列表界面

下面测试一下其他功能，司机和乘客界面相似。

当司机载客后，他要修改当前车上的乘客数和他们的目的地，点击 MENU，如图 5.18，点击“设置”按钮，会弹出设置对话框，如图 5.21。



图 5.21 司机设置对话框



图 5.22 修改个人信息界面

当用户想修改个人信息时，再点击 MENU 键后，点击“个人信息”按钮进入个人信息界面，如图 5.22 所示，列举两个修改界面，如图 5.23 和 5.24 所示。



图 5.23 修改昵称对话框



图 5.24 修改身份对话框

第 6 章 总结

经过五个月的设计和开发，实现了易道招车手机应用软件。该应用软件在 Android 操作系统上运行。采用 C/S 结构，建立在 Internet 网络环境中。该应用软件分为 Android 客户端、服务器端、数据库和通信四大部分。调节出租车与乘客分布不均匀的现象，解决乘客打车时间长和司机跑空车的问题。为乘客用户提供两种招车界面，一个是利用百度地图开发的地图界面，另一个是司机列表界面。乘客用户通过地图显示的其他乘客的位置和目的地信息，来选择是否与其拼车。司机是被动的，当有乘客向其提出打车请求后，司机从这些乘客中选择合适的乘客。打车成功后在应用地图上显示司机与乘客间的导航线路，便于司机能够快速找到乘客。打车完成后可以对乘客或司机进行评价，为其他用户提供有帮助的“信任度”信息便于他人的选择。

在这期间，开发该系统所要做的需求分析、总体设计、Android 客户端界面和服务端程序的详细设计和测试等工作基本完成。实现的功能基本符合需求分析中的需求。

通过易道招车应用软件的开发，熟悉业务流程和软件的总体设计是至关重要的。编码只是整个开发中的一小部分，而整体的业务流程和需求分析才是至关重要的，只有这样编出的软件才是优秀的。所以，作为一名合格的计算机软件专业的人才，一定要具备规划和设计能力。

此招车应用软件仍存在许多不足，但是通过该应用的设计与实现，基本掌握了软件开发流程，了解了在开发中每个环节容易出现的问题以及如何解决的方法。作为开发人员不仅思考如何实现软件功能，也要从用户角度思考如何才能使软件更友好，这样才能编程好的软件。

参考文献

- [1]张海藩. 软件工程导论. 北京:清华大学出版社, 2008
- [2] (美) 埃克尔 (Eckel, B.) 著;陈昊鹏译. Java 编程思想. 北京:机械工业出版社, 2007
- [3]王珊, 萨师煊. 数据库系统概论. 北京:高等教育出版社, 2010
- [4] 杨丰盛. Android 应用开发揭秘. 北京:机械工业出版社, 2010
- [5]谢希仁. 计算机网络. 北京:电子工业出版社, 2008
- [6]郑莉. Java 语言程序设计. 北京:清华大学出版社, 2005
- [7] (美) 坎皮恩 (Campioni, M) . Java 语言导学. 北京:机械工业出版社, 2004
- [8]严蔚敏, 吴伟民. 数据结构. 北京:清华大学出版社, 1997
- [9]汤小丹. 计算机操作系统. 西安:西安电子科技大学出版社, 2007

致 谢

自从12年12月份开始设计需求分析，到13年3月份易道应用软件基本编码结束，按照需求，所有功能基本实现，再到5月初毕业论文的完成。我付出了很多，同时也收获了很多。当初我接到此课题时，心理有些担心，怕不能完成，辜负老师同学对我的期望。但是通过导师的精心指导再加上自己的不懈努力，最后我终于不负众望开发了一款招车软件。在这里我要感谢的人很多。

首先要感谢我最敬佩的周老师。周老师日常工作繁多，但是在我的毕设和论文指导方面，重来没有耽误过。半年来我们每周都会见一次面，检查我一周的工作量并指导我如何改进，他给了我很多的帮助，帮我找学习材料，帮我分析软件的需求，解决我开发中遇到的问题，当我碰到棘手问题急躁不安时，周老师总是激励我，鼓励我。如果没有周老师的指导，我要完成这么复杂的毕设和论文，可以想像，是多么困难的一件事情。

然后还要感谢大学四年来所有教过我，帮助过我的老师，他们是我的良师益友，给我的计算机专业技能打下了坚实的基础；同时还要感谢我的室友以及所有帮助和支持我的同学们，是他们给了我足够的勇气。

最后，要感谢我的母校——辽宁大学，感谢她四年来对我的辛勤栽培，使我受益终生。

邓中华

2013年5月 于沈阳