|  |  |  |  |
| --- | --- | --- | --- |
| **上海交通大学** | **文 档 编 号** | **产品版本** | **密级** |
| SJTU-IS-2016-06-22-5 | **V1.0** | **内部** |
| **项目名称： InstCar** | | **共 47 页** |

**XXX项目**

**详细设计说明书**

**(仅供内部使用)**

**文 档 作 者： \_\_\_\_\_\_\_\_包文天\_\_\_\_\_\_\_\_ 日期：2016/6/22**

**审 核 人 员： \_\_\_\_\_\_\_\_王科迪\_\_\_\_\_\_\_\_\_ 日期：2016/6/23**

**批 准 人 员： \_\_\_\_\_\_\_王科迪\_\_\_\_\_\_\_\_\_ 日期：2016/6/23**

****

**上海交通大学信息安全工程学院**

**版权所有 不得复制**

目录

[目录 2](#_Toc277146165)

[摘要 3](#_Toc277146166)

[版本历史 4](#_Toc277146167)

[项目小组 4](#_Toc277146168)

[1. 引言 5](#_Toc277146169)

[1.1 编写目的 5](#_Toc277146170)

[1.2 背景 5](#_Toc277146171)

[1.3 定义 5](#_Toc277146172)

[1.4 参考资料 5](#_Toc277146173)

[2. 系统架构/结构 5](#_Toc277146174)

[3. 用例设计 5](#_Toc277146175)

[3.1 系统角色 5](#_Toc277146176)

[3.2 用例索引 6](#_Toc277146177)

[3.3 用例描述 6](#_Toc277146178)

[3.3.1 用例1 /【U01-01-01-01办公管理】 6](#_Toc277146179)

[3.3.2 用例2 /【U01-01-01-02信件管理】 9](#_Toc277146180)

[4. 编程协定 9](#_Toc277146181)

[4.1 操作系统 9](#_Toc277146182)

[4.2 编程工具 9](#_Toc277146183)

[4.3 编辑、调试、联接程序 9](#_Toc277146184)

[4.4 编译工具 9](#_Toc277146185)

[4.5 模拟、仿真数据 9](#_Toc277146186)

[4.6 诊断、测试程序 9](#_Toc277146187)

[4.7 检测程序 9](#_Toc277146188)

[5. 数据结构 10](#_Toc277146189)

[5.1 物理结构 10](#_Toc277146190)

[5.2 库结构 10](#_Toc277146191)

[5.3 链路结构 10](#_Toc277146192)

[5.4 表结构 10](#_Toc277146193)

[5.5 内部存贮区分配 10](#_Toc277146194)

[6. 逻辑结构 10](#_Toc277146195)

[6.1 常量定义 10](#_Toc277146196)

[6.2 变量定义 10](#_Toc277146197)

[6.3 结构定义 10](#_Toc277146198)

[6.4 全局定义 11](#_Toc277146199)

[7. 源文件列表 11](#_Toc277146200)

# 摘要

摘 要:

本文档为项目详细设计说明书，本文将从系统架构，用例设计，编程协定，数据结构，逻辑结构等方面，详细阐述项目的设计方法。具体涉及模块的功能、性能、输入输出项、局部变量、设计方法、类图和顺序图、接口说明、存储分配、注释设计、限制条件、测试计划等。

本文同时给出了项目在编程方面的开发环境，测试环境，数据库配置环境，代码开发规范等，详细给出了本系统涉及的基本表的物理结构、表结构，也给出了类库架构和链路结构。另外，还详细列出了编程过程中使用的常量、变量、结构、全局定义和源文件列表。

通过对本文档的阅读，应能对整个系统所需实现的功能以及模块划分、实现和数据库的表结构有清晰的认识。

关键词:**InstCar，系统架构，用例设计，编程协定，数据结构，逻辑结构**

# 版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 日期 | 更改内容 | 版本 |
| 包文天 | 2016/6/10 | 项目详细设计说明书初稿 | V0.1 |
| 包文天 | 2016/6/22 | 增加模块使用图例，系统架构图，代码规范说明等 | V0.5 |
| 王科迪 | 2016/6/23 | 审核报告各部分 | V1.0 |

# 项目小组

|  |  |  |  |
| --- | --- | --- | --- |
| 职务 | 姓名 | 学号 | 职责 |
| 组长 | 王科迪 | 5130369069 | 详细设计 程序编码 |
| 组员 | 秦瑞泽 | 5130369074 | 需求分析 程序编码 |
| 组员 | 包文天 | 5130369060 | UI设计 程序编码 |
| 组员 | 白晓宇 | 5130369070 | 程序测试 需求分析 |
| 组员 | 易泉 | 5130369075 | 程序测试 概要设计 |

# 引言

## 编写目的

本文档为说明文档中的纤细设计说明部分，旨在对InstCar项目各部分作系统和详细的说明。具体包括：项目系统架构，用例设计，编程协定，数据结构，逻辑结构等，以及模块的功能需求、流程、输入输出项、变量定义和接口等部分。

本文档的预期读者：

1.InstCar项目组成员，通过参考本详细设计说明书，应能对整个系统所需实现的功能以及模块划分、实现和数据库的表结构有清晰的认识，进而进行后续的代码编写和测试。

2.项目审核人员，本次课程设计中为组长王科迪。

3.项目评价人员，本次课程设计中为刘海涛老师。

## 背景

1. 项目名称 ：InstCar
2. 项目任务提出者 ：InstCar项目组
3. 项目任务开发者 ：InstCar项目组
4. 项目任务开发平台 ：Window7/8，macOS
5. 项目针对用户 ：长途旅客
6. 任务来源 ：软件设计课程设计
7. 交办单位 ：刘海涛老师
8. 承办单位 ：上海交通大学

## 定义

REST: (REpresentation State Transfer) 描述了一个架构样式的网络系统，Web 应用程序最重要的 REST 原则是，客户端和服务器之间的交互在请求之间是无状态的。

API：Application Programming Interface, 是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

Leancloud: LeanCloud 是国内领先的针对移动应用的一站式云端服务，BaaS 领域的先行者，专注于为应用开发者提供一流的工具和平台。

AngularJS: AngularJS是Google开发的纯客户端JavaScript技术的WEB框架,用于扩展、增强HTML功能,它专为构建强大的WEB应用而设计。

JQuery: JQuery是继prototype之后又一个优秀的Javascript库。它是轻量级的js库 ，它兼容CSS3，还兼容各种浏览器。

NodeJS：简单的说 Node.js 就是运行在服务端的 JavaScript。Node.js 是一个基于Chrome JavaScript 运行时建立的一个平台。Node.js是一个事件驱动I/O服务端JavaScript环境，基于Google的V8引擎，V8引擎执行Javascript的速度非常快，性能非常好。

参考资料

Leancloud官方文档

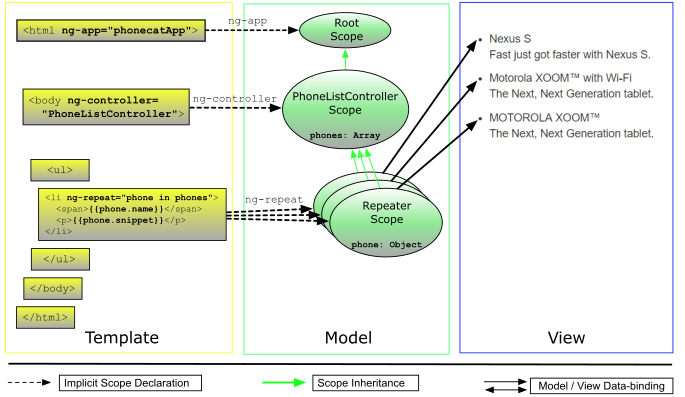
AngularJS开发者手册

Google

Wikipedia

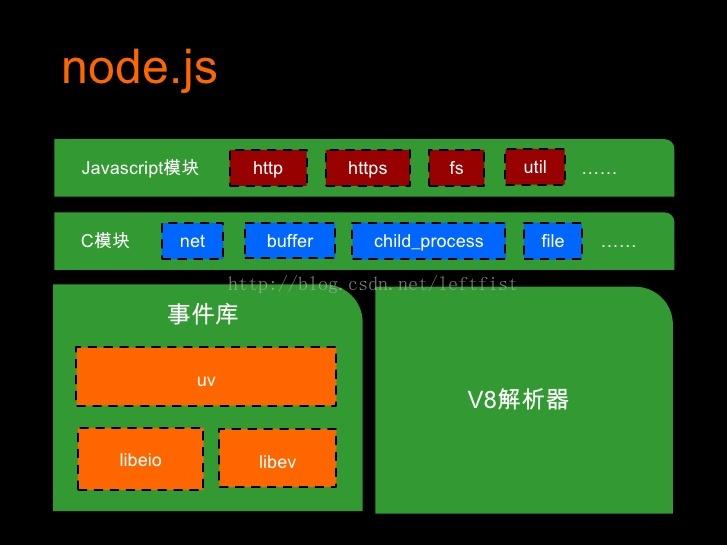
# 系统架构/结构

InstCar采用NodeJS作为后端框架，AngularJS+JQuery作为前端框架，Leancloud作为后端数据库。项目前端采用MVC架构，图例如下：



InstCar前端逻辑架构

后端NodeJS的逻辑框架如下：



NodeJS逻辑框架

# 系统模块设定

## 微信客户端模块说明

### 4.1.1模块描述

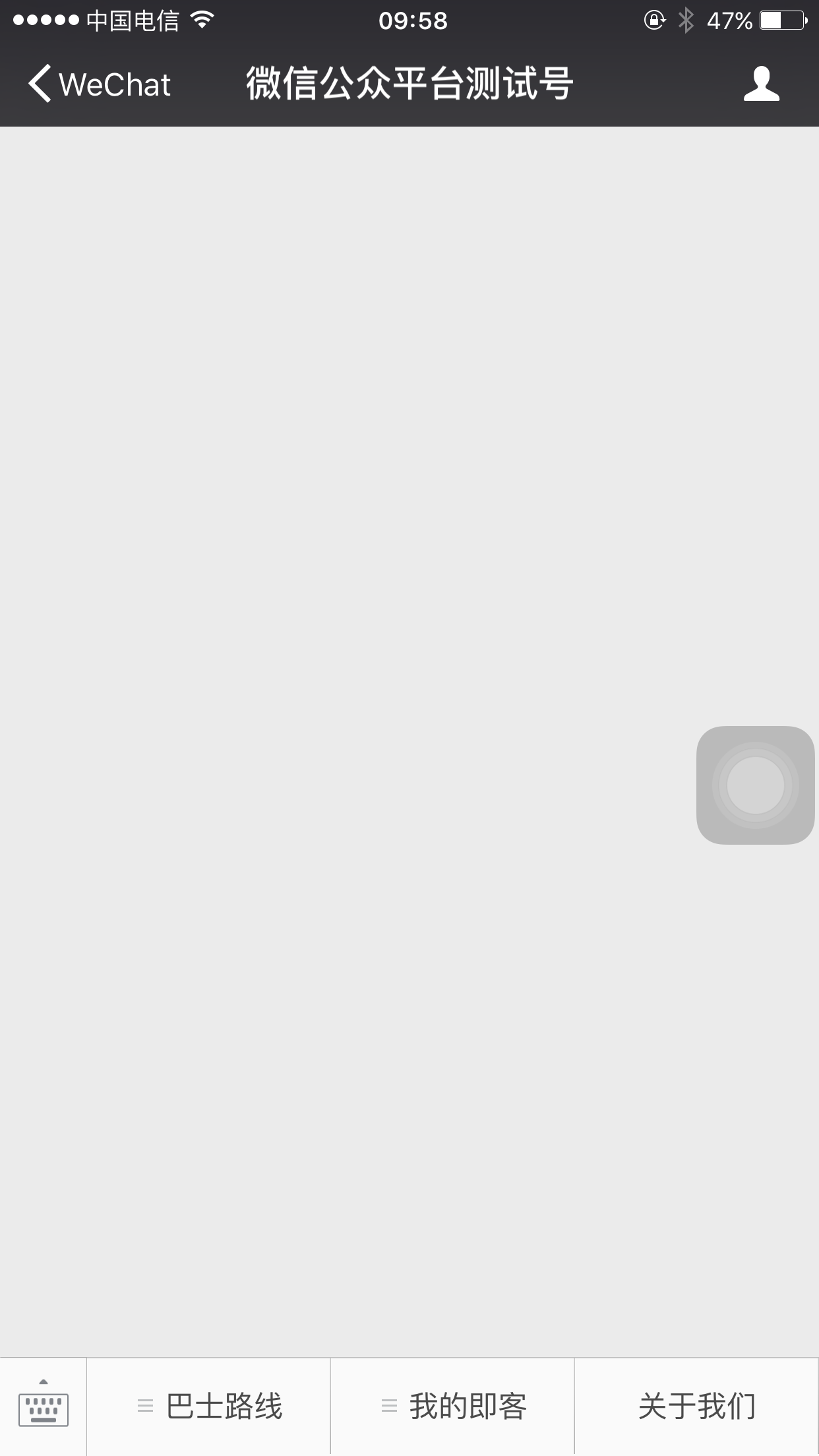
本模块为InstCar项目的用户界面，考虑到现在移动互联网的趋势，所以我们舍弃了传统的网页，而改用微信公众号作为实际的登录和操作界面。这样的设计一是对用户来说没有额外记录网址的负担，二是操作简易，与其他公众号平台的接口相似，简单易学。

### 4.1.2功能

微信用户端主要功能：

* + - 1. 用户登录
      2. 选择路线
      3. 申请路线
      4. 检票上车
      5. 乘客管理

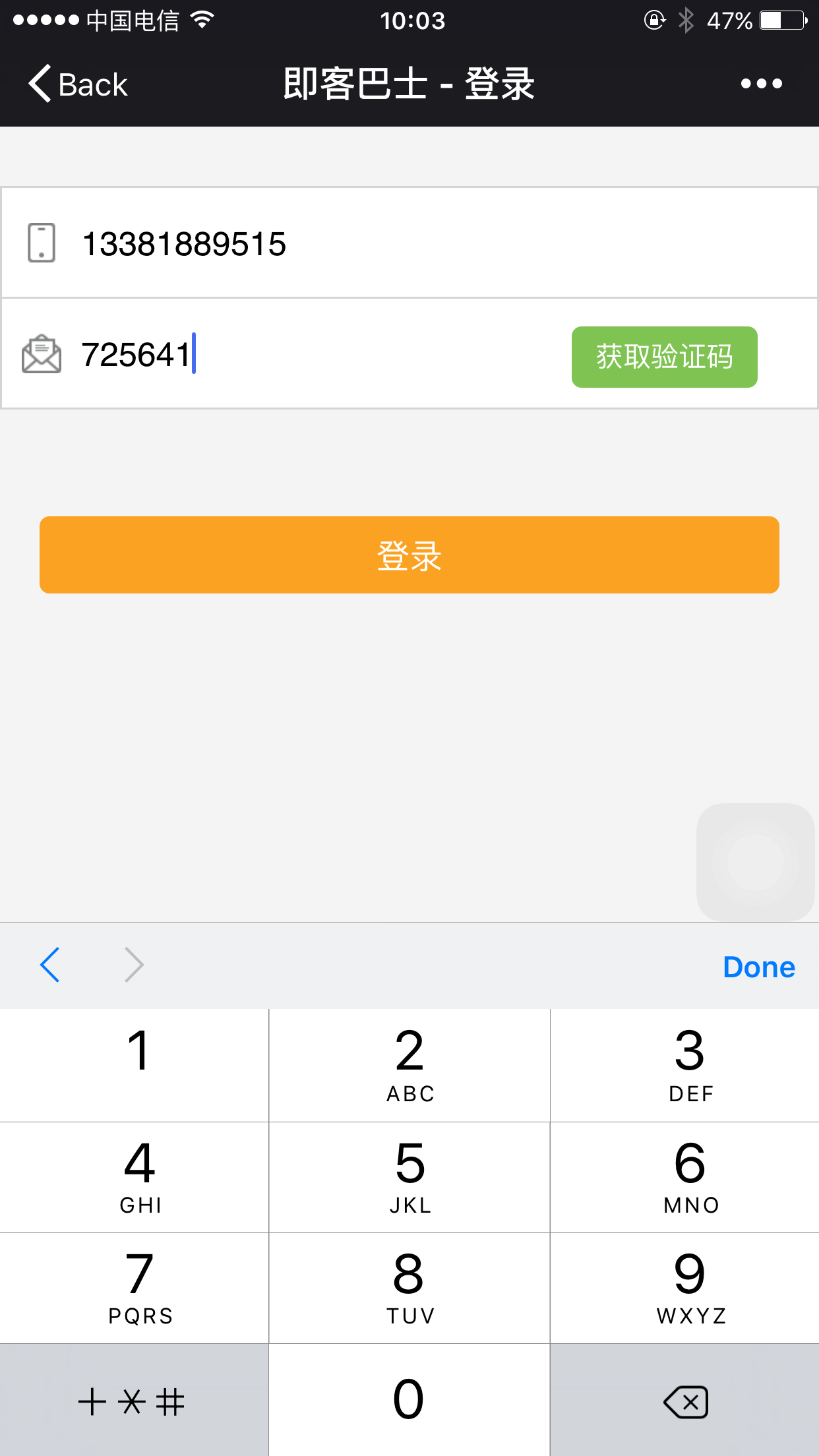
本模块对应的界面如下：



1.用户登录



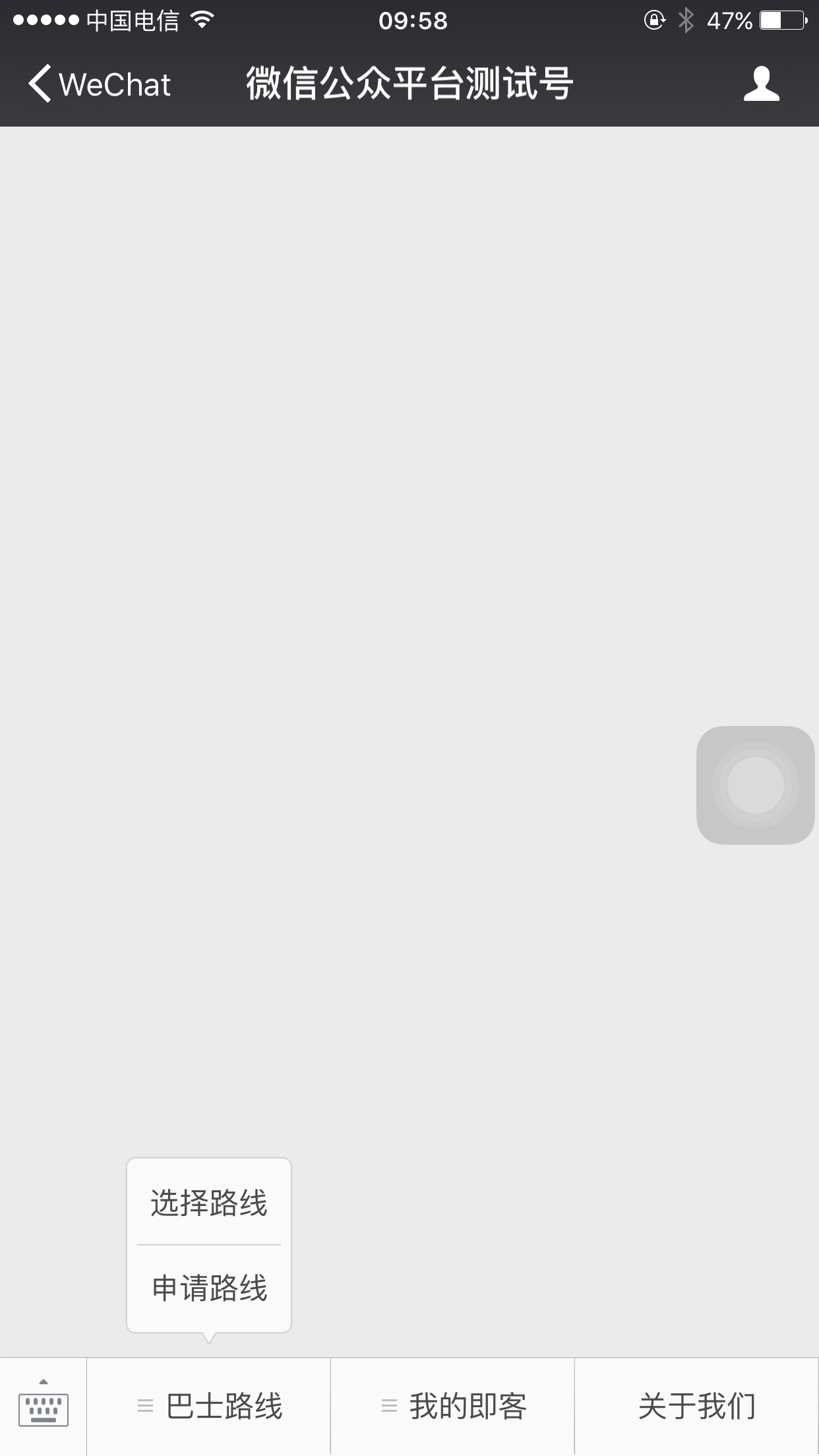
点击获取验证码，输入得到的验证码如下：



点击登录，再次返回主页面，即可进行选择路线即可出发啦~

2.选择路线

这部分是InstCar项目逻辑最复杂的部分。首先，在主页中点击选择路线选项：



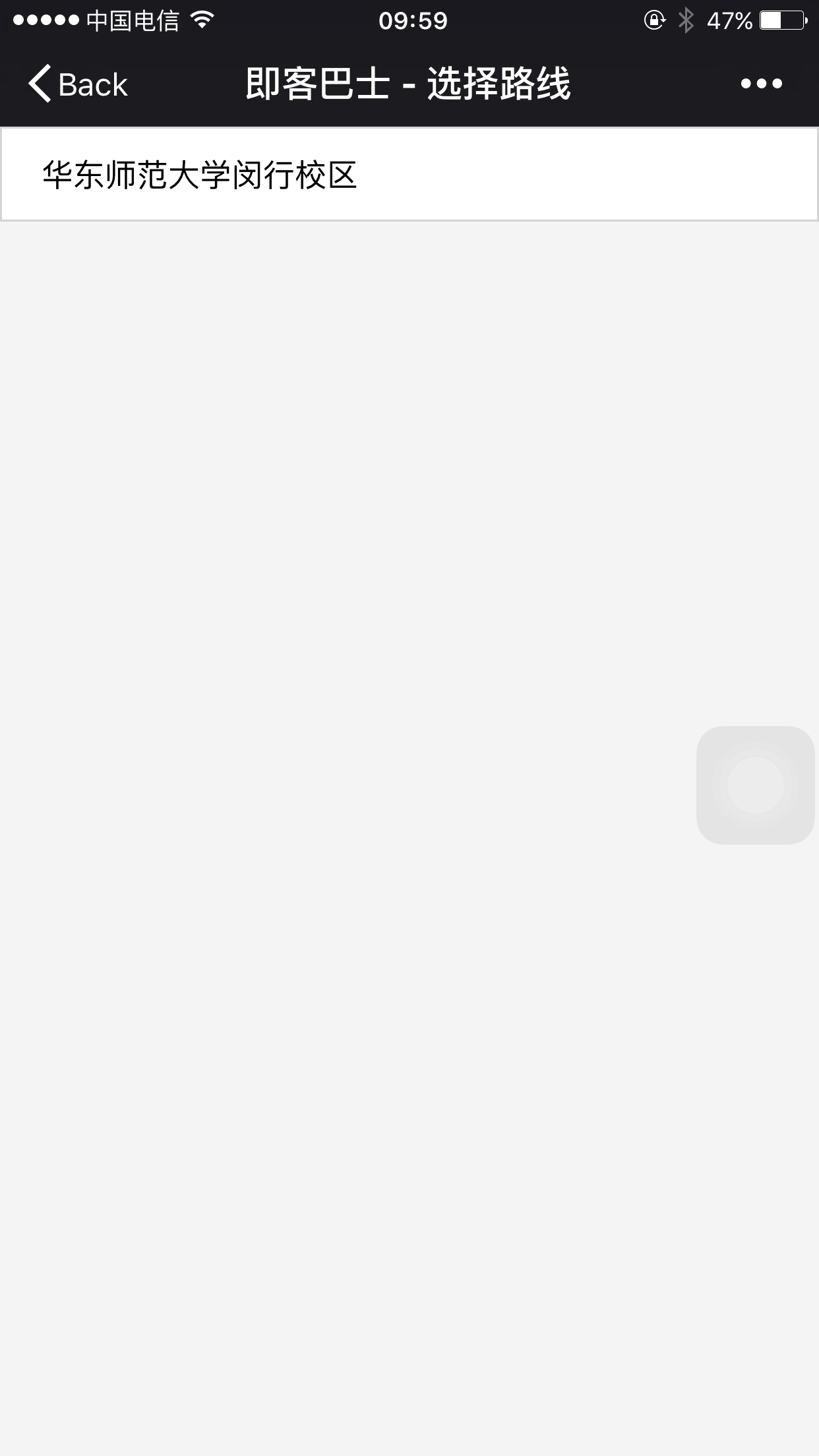
进入选择路线界面，如下：



选择起点，如下：



现在我们选择上海作为始发地，之后会要求我们选择更加具体的出发地址，由于数据库没有将全部数据导入，所以上海地区现在只有一个始发地，如下：



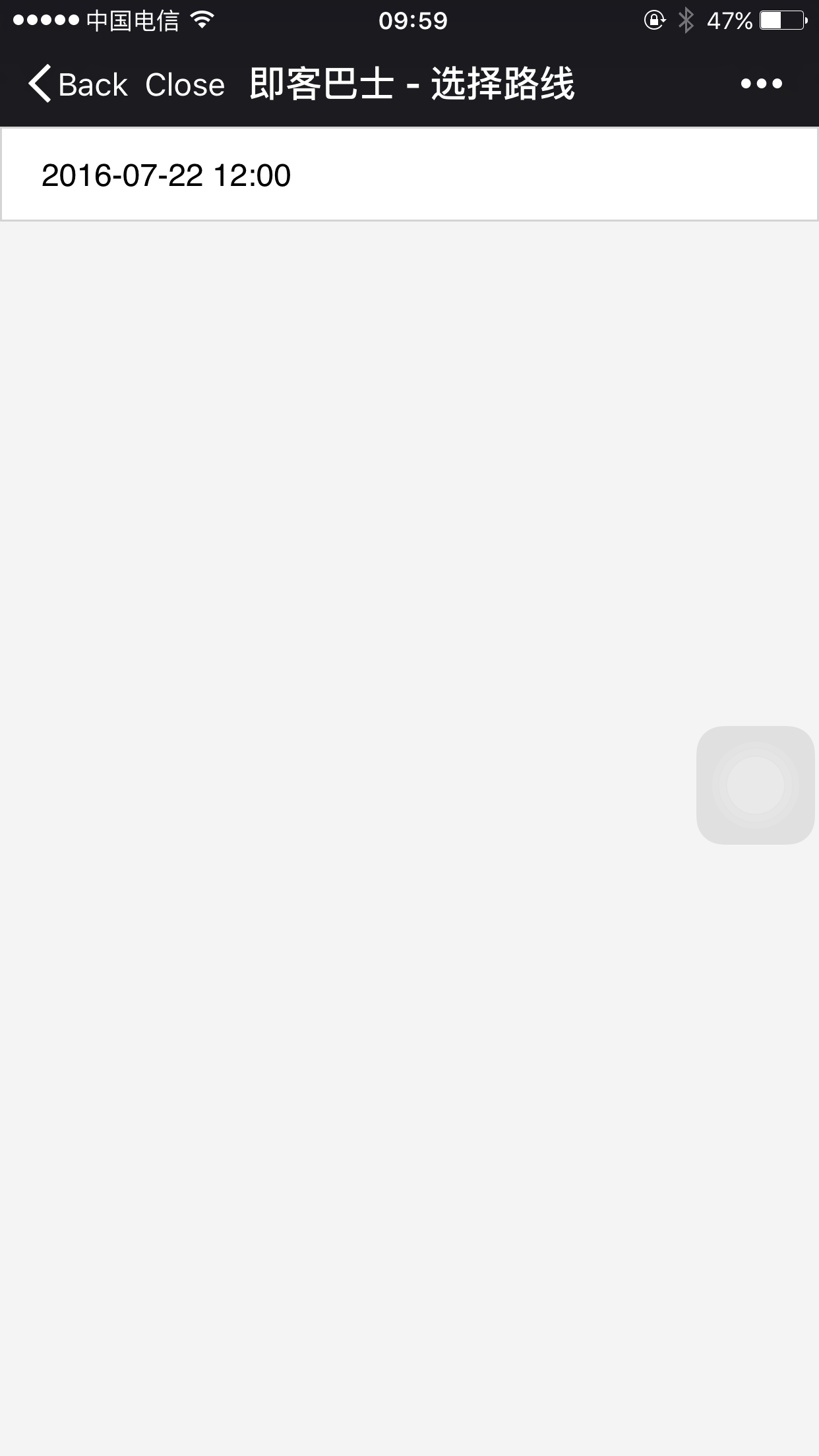
我们先愉快的选择华师作为出发点，然后需要选择目的地，如下：



经过如前步骤，我们选择交大作为目的地~



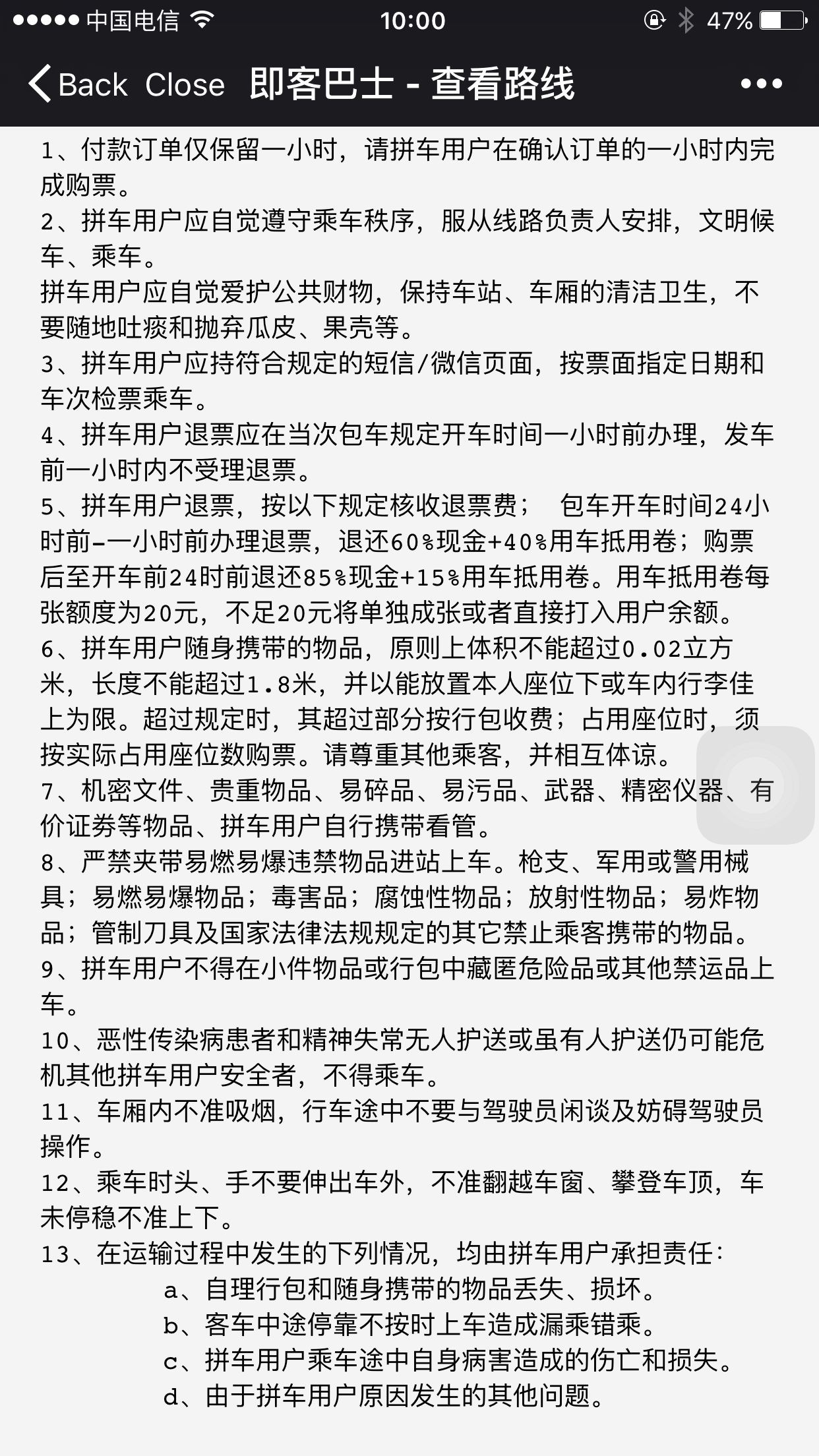
最后，选择班次，如下：



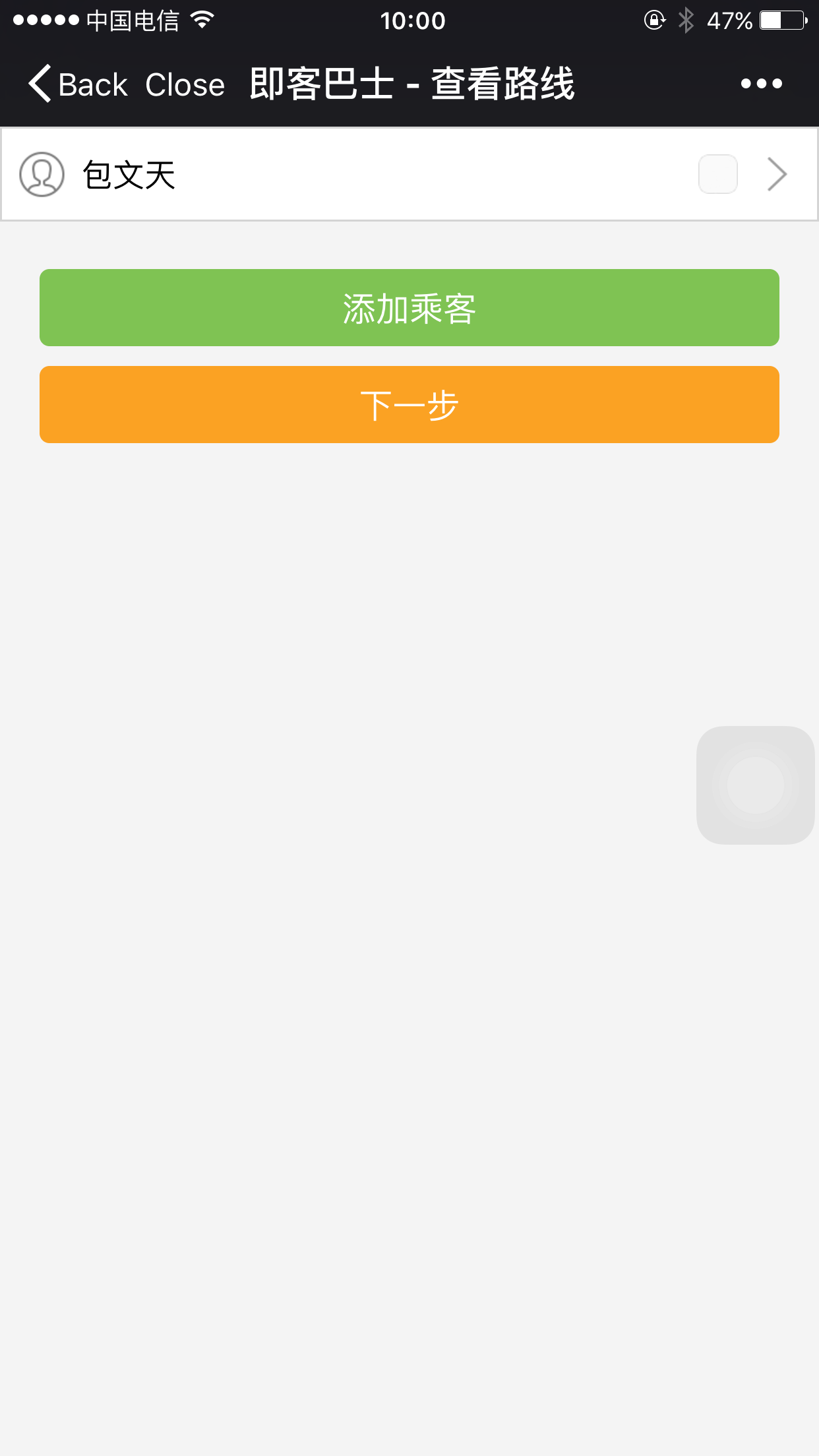
经过以上选择，我们有了一个初步的计划，信息如下：



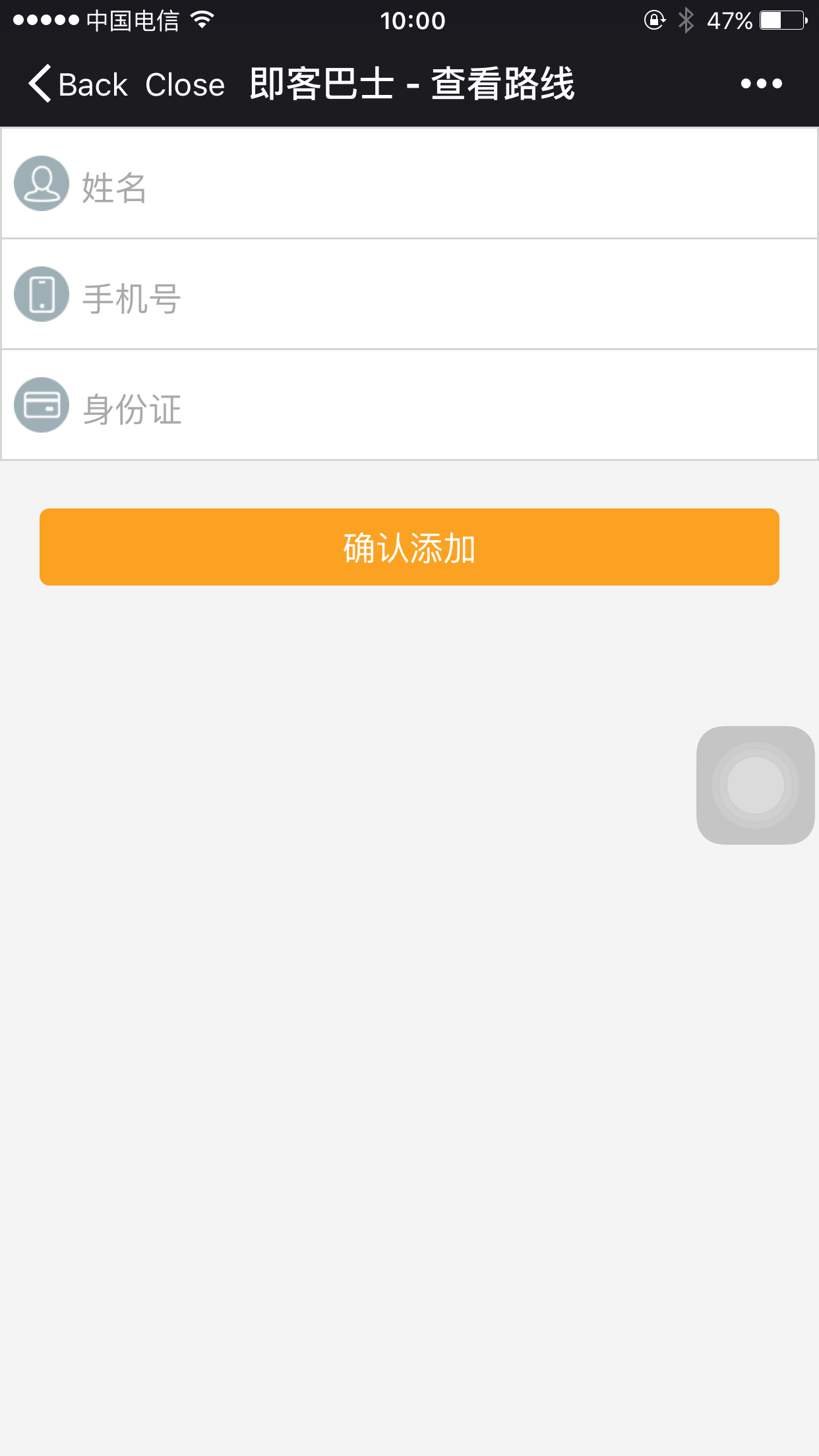
线路详情会提供有关这趟线路的详细信息（处于开发阶段），乘客须知里放有免责协议以及安全须知，如下：



确认无误后，点击加入路线，这是会自动跳转到乘客选择界面：



出现的是账户已有的乘客信息，还可以添加乘客，点击添加乘客按钮：



在这个界面中，需要我们输入姓名，手机号，身份证号等信息，用户巴士公司与乘客进行后续联系。填写无误后，点击确认添加，会返回刚才选择乘客的界面，只是这时多出了刚才新添加的乘客：



若要选中乘客，在后面的方框中打钩。确认无误后，点击下一步，可以选择优惠券：



优惠券的后端逻辑比较复杂，现阶段的策略上支持管理员发放优惠券，用户自助选择，优惠策略比较多样，例子可以参考上图。选择优惠券后，点击下一步，弹出订单确认：



点击OK，订单就生成啦~

3.查看订单信息及付款

订单生成之后，可以在我的即客->我的订单中查看：

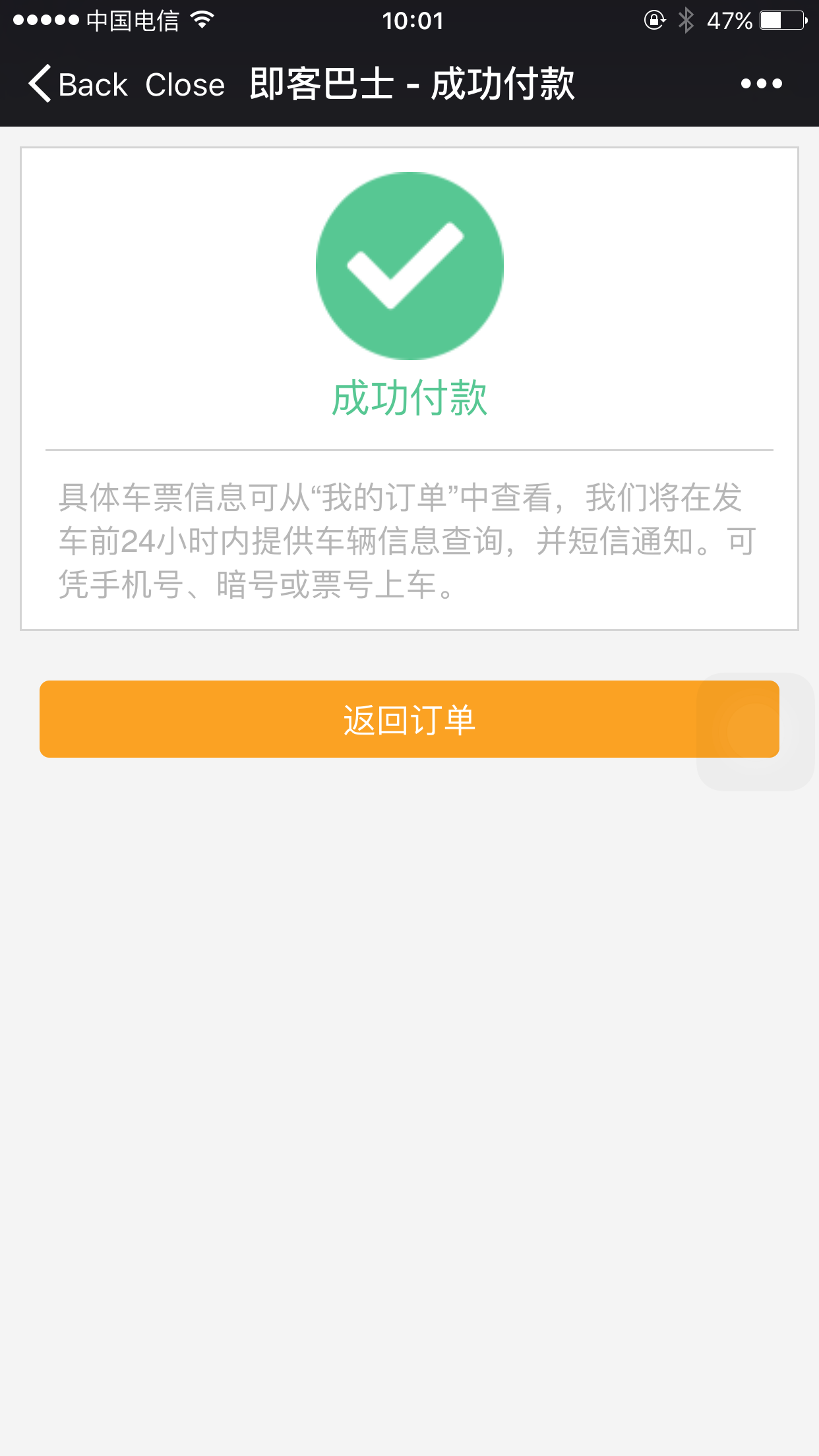




其中，未付款的订单有去付款的按钮，下单之后要注意及时付款才能不耽误出行哦~

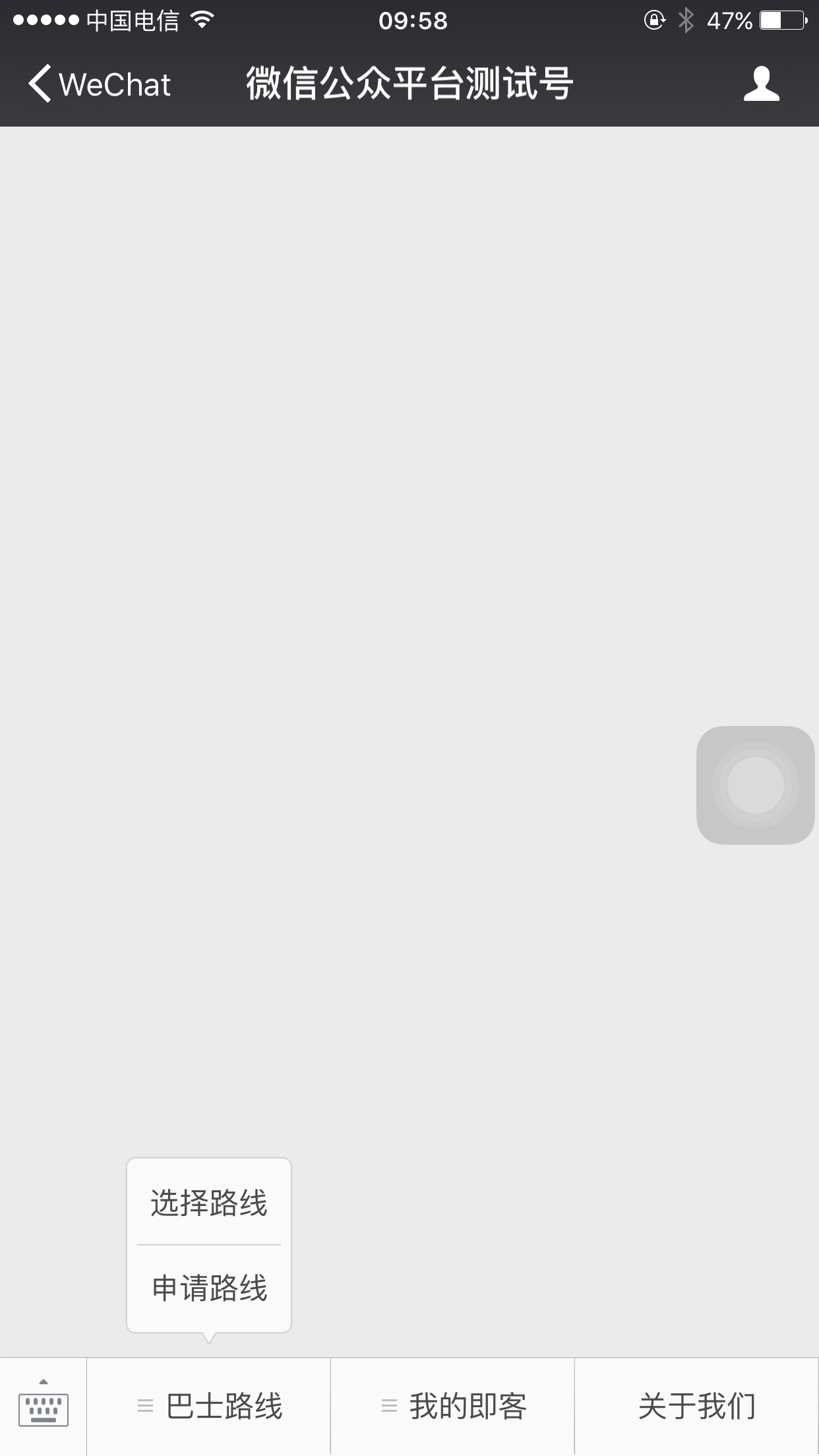


选择同意协议并付款，会弹出模拟付款的确认信息，如果同意，点击确认，模拟付款即完成。

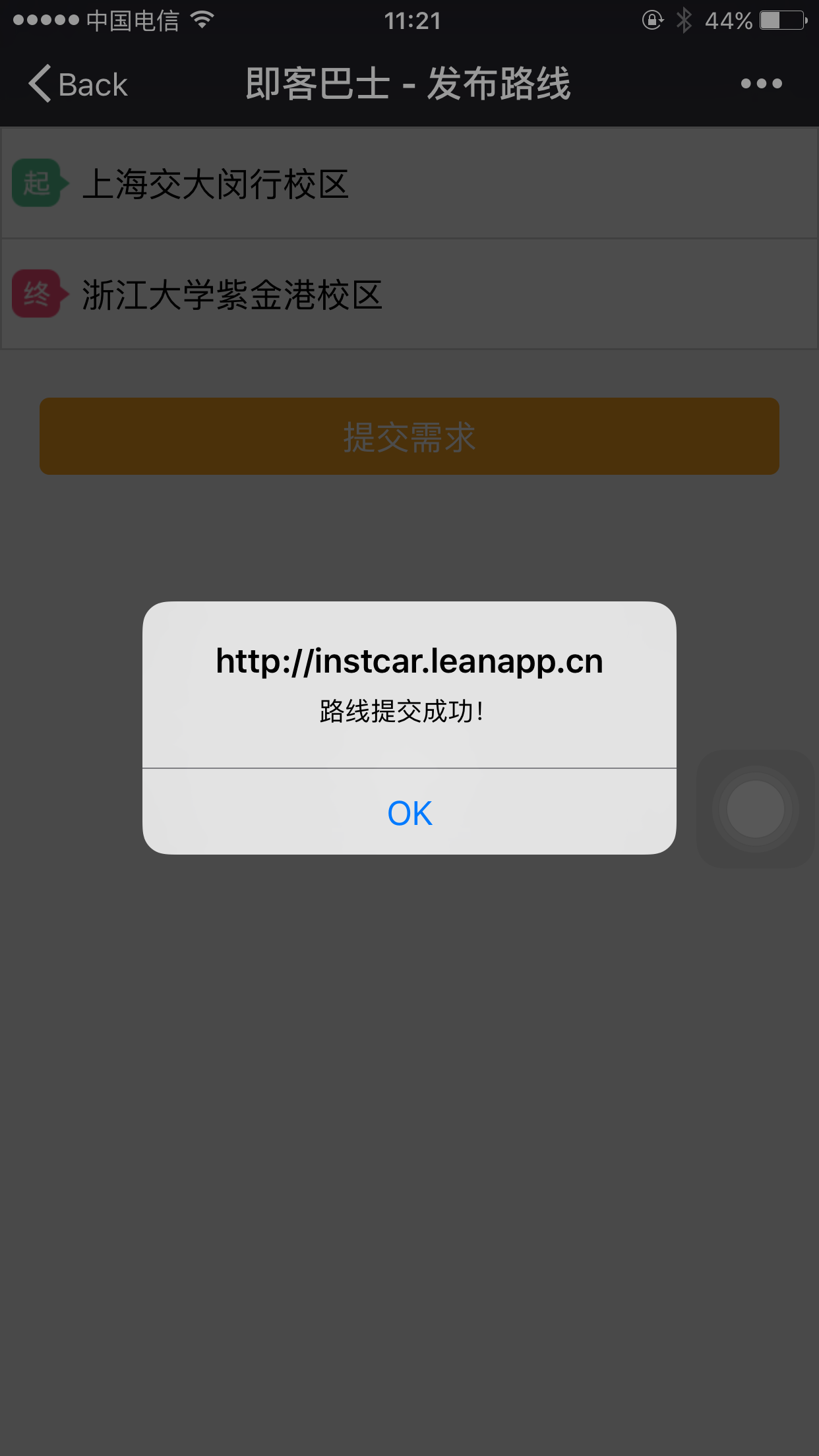


4.路线申请

InstCar中现有的路线都是管理员根据巴士公司的服务情况添加的，但一个很自然的想法是，如果有些线路巴士公司没有开设，但是用户需求量很大怎么办？我们开设了线路申请功能，用户可以提交线路，如果系统后台有足够多的类似请求，管理员会与巴士公司联系来增设相应线路。



通过巴士路线->申请路线进入相应菜单，填写申请路线的始发地和目的地，确认：



**4.1.3性能**

高效：数据处理速度、数据库读取写入速度都很快，可以在很短时间做出反应

稳定：进行了各种错误判断，使得程序几乎不会出错

### 4.1.4输入项

用户的按键操作

### 4.1.5输出项

用户订单信息，账户信息，公众号推送信息。

### 4.1.6局部变量

无

### 3.1.7窗口对象说明

本模块涉及16个主要窗口，下面以各窗口从主界面进入路径命名，说明各窗口对象：

1.主界面

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Button** | 巴士路线 | Button | 主要功能 |
| **Button** | 我的即客 | Button | 显示账户相关信息 |
| **Button** | 关于我们 | Button | 显示InstCar信息 |

2.主界面->选择路线

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Button** | 起点 | Button | 选择起点 |
| **Button** | 终点 | Button | 选择终点 |
| **Button** | 路线 | Button | 选择线路 |
| **Button** | 查看详情 | Button | 进入下一级菜单 |

3.主界面->选择路线->起点

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Button** | 各个城市 | Button | 选择始发城市 |

4. 主界面->选择路线->起点->各个城市

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Button** | 各个具体始发地 | Button | 选择具体始发地 |

5. 主界面->选择路线->终点

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Button** | 各个城市 | Button | 选择到达城市 |

6.主界面->选择路线->终点->各个城市

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Button** | 各个具体目的地 | Button | 选择具体目的地 |

7. 主界面->选择路线->选择班次

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Button** | 选择具体班次 | Button | 无 |

8. 主界面->选择路线->查看详情

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **List** | 发车时间 | list | 无 |
| **List** | 路线 | List | 无 |
| **Button** | 路线详情 | Button | 无 |
| **Button** | 乘客须知 | Button | 无 |
| **Button** | 加入路线 | Button | 进入下一级菜单 |

9. 主界面->选择路线->查看详情->乘客须知

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Text** | 安全守则 | Text | 无 |

10. 主界面->选择路线->查看详情->加入路线

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Button** | 乘客姓名 | Button | 点击确认乘客信息 |
| **Button** | 添加乘客 | Button | 无 |
| **Button** | 下一步 | Button | 点击进入下级菜单 |

11. 主界面->选择路线->查看详情->加入路线->添加乘客

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Input** | 姓名 | Text | 乘客姓名 |
| **Input** | 手机 | Text | 乘客手机 |
| **Input** | 身份证号码 | Text | 乘客身份证号码 |
| **Button** | 确认添加 | Button | 返回加入路线 |

12. 主界面->选择路线->查看详情->加入路线->下一步

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Text** | 优惠券信息 | Text | 无 |
| **Button** | 下一步 | Button | 进入下一级菜单 |

13. 主界面->申请路线

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Input** | 始发地 | Text | 无 |
| **Input** | 目的地 | Text | 无 |
| **Button** | 提交需求 | Button | 返回主菜单 |

14.主界面->我的即客->我的订单

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **List** | 订单信息 | List | 无 |
| **Button** | 取消订单 | Button | 无 |
| **Button** | 去付款 | Button | 无 |

15. 主界面->我的即客->我的订单->去付款

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **List** | 发车时间 | list | 无 |
| **List** | 路线 | List | 无 |
| **Button** | 线路详情 | Button | 无 |
| **Button** | 乘客须知 | Button | 无 |
| **Button** | 用车协议 | Button | 无 |
| **Button** | 乘客和车票 | Button | 无 |
| **Button** | 同意用车协议并付款 | Button | 进入下一级菜单 |
| **Button** | 取消订单 | Button | 无 |

14.主界面->我的即客->我的订单->去付款->同意用车协议并付款

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Text** | 成功付款 | Text | 无 |
| **Button** | 返回订单 | Button | 返回查看订单 |

15.主界面->我的即客->检票上车

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **List** | 车票信息 | List | 无 |
| **Button** | 检票上车 | Button | 需待后台管理员分配座位后才能使用 |

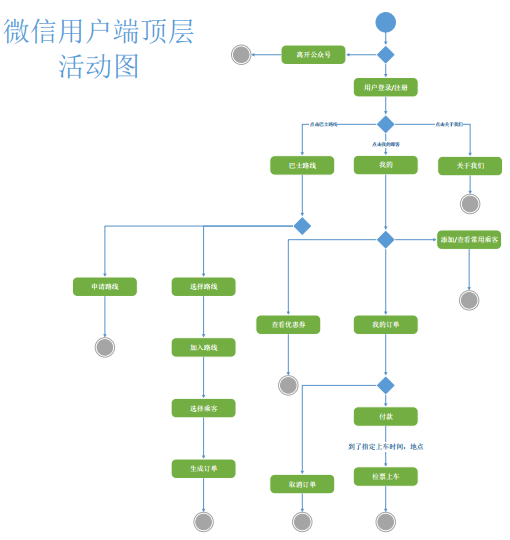
16. 主界面->我的即客->我的优惠券

|  |  |  |  |
| --- | --- | --- | --- |
| **所属类** | **内容** | **类型** | **注释** |
| **Text** | 优惠券信息 | Text | 无 |

### 4.1.8设计方法

考虑到移动互联网的趋势，设计用户界面时我们抛弃了传统的网页端，改用微信公众号的形式作为客户端。见面设计上，采用开发者模式自定义功能选项，但是也保持了公众号一贯的风格，使得用户使用起来简单易学。用户端逻辑尽量符合平时购票流程，降低学习使用的时间成本，方便推广。

### 4.1.9 UML类图



### 4.1.10接口

系统使用了REST API，同时调用了AngularJS，JQuery，Express，NodeJS自带的原生API。

### 4.1.11存储分配

系统所有数据全部托管于云端LeanCloud。

### 4.1.12注释设计

注释的原则：

1.注释形式统一

整个应用程序中，其他程序部分都应使用具有一致的标点和结构的样式构造注释。

2.注释的简洁

注释内容要简单、明了、含义准确，防止注释的多义性。

3.注释的一致性

在代码编写之前或编写的过程中完成注释的构造，避免代码完成后才加入注释。另外，如代码进行修改，相应的注释也要进行修改。

4.注释的位置

保证注释与其描述的代码相邻，在相应行代码的上方或者该行的后方添加注释。

5.注释的数量

注释不应过多，着重描述程序功能和各组成部分之间的相互关系。逐行解释程序如何工作的低级注释无需添加。

6.必加的注释

程序中典型的算法、变量的声明和程序的接口必须有注释，循环和逻辑分支组成的代码中也必须有注释。

注释的格式和规范：

本系统中采取三种注释方法，以便提高注释的可读性和程序的易维护性。

1.单行注释

这是程序中最为常见的注释部分，在变量声明和函数内部采用该注释方法。以就进原则为标准，在需要注释的代码行附近进行单行注释。

注释格式：//注释内容

2.函数注释

在程序中出现的函数需要采用该注释方法，注释位置为函数声明的下一行，通常一行即可完成注释。

注释格式：/\*\*注释内容\*/

3.文档注释

在每个类的前部需要采用该注释方法，用以说明项目名称、项目开发者和版本等信息，采取多行注释。

注释格式：/\*\*

\*注

\*释

\*内

\*容

\*/

代码中的注释实例截图如下：



### 4.1.13限制条件

无

### 4.1.14测试计划

参见《组装测试计划》与《测试计划》。

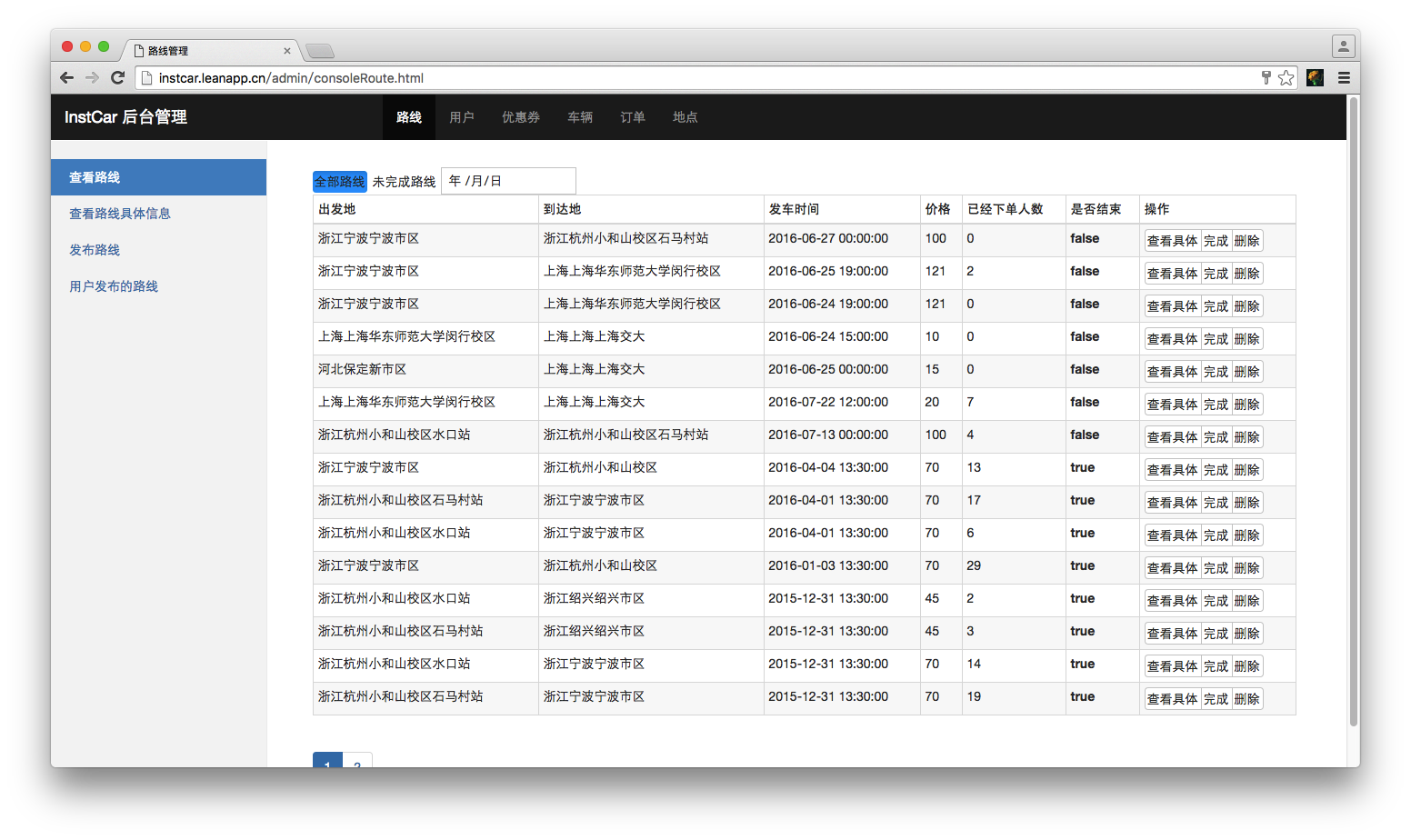
### 4.1.15尚未解决的问题

无

## 4.2管理员端设计说明

### 4.2.1模块描述

本模块为管理员控制后端数据的图形界面。



### 4.2.2功能

管理员端主要功能：

* + - 1. 路线管理
      2. 用户管理
      3. 优惠券管理
      4. 车辆管理
      5. 订单管理
      6. 地点管理

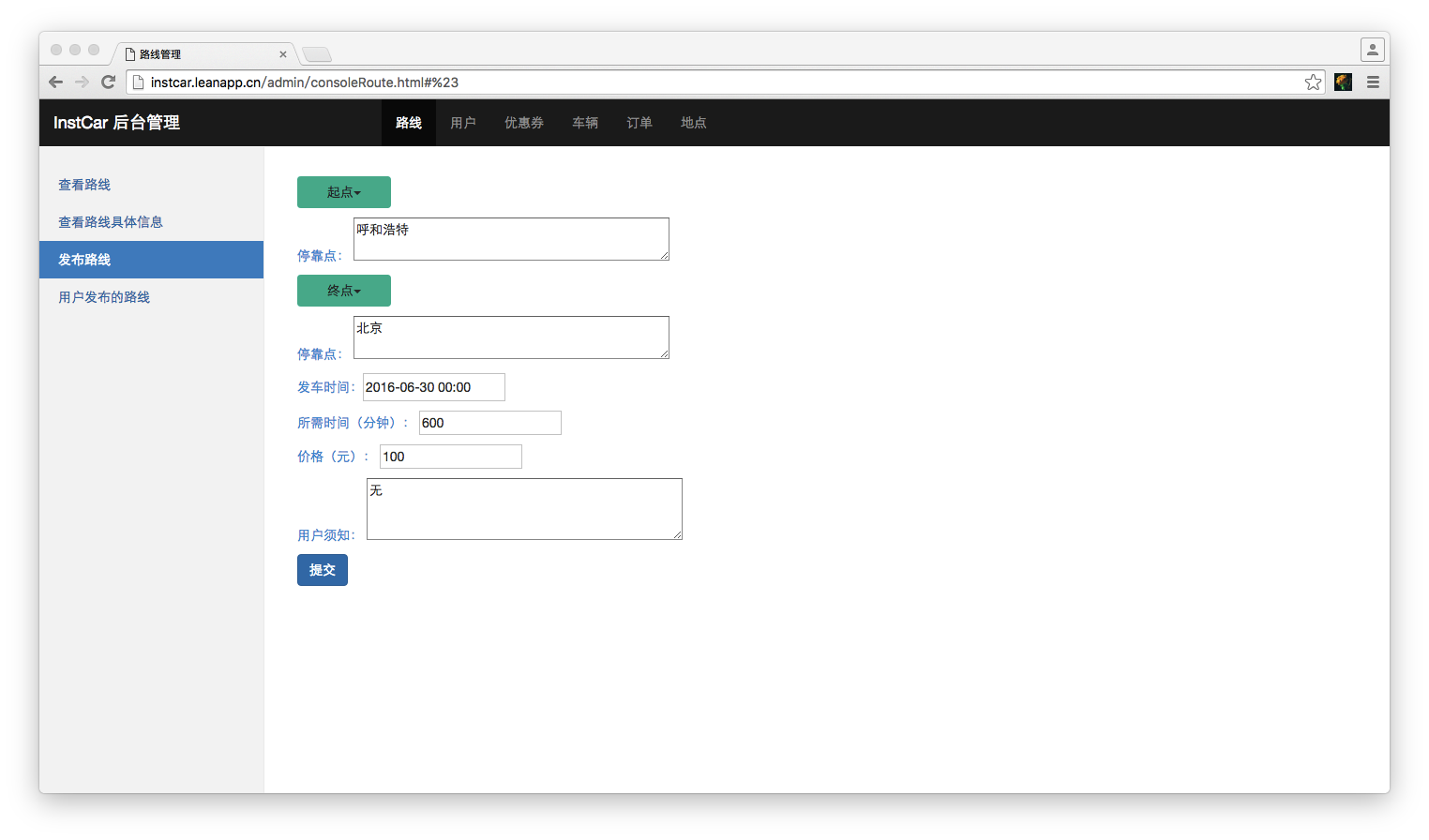
1.路线管理

从管理员端，可以查看到每条线路的详细信息：



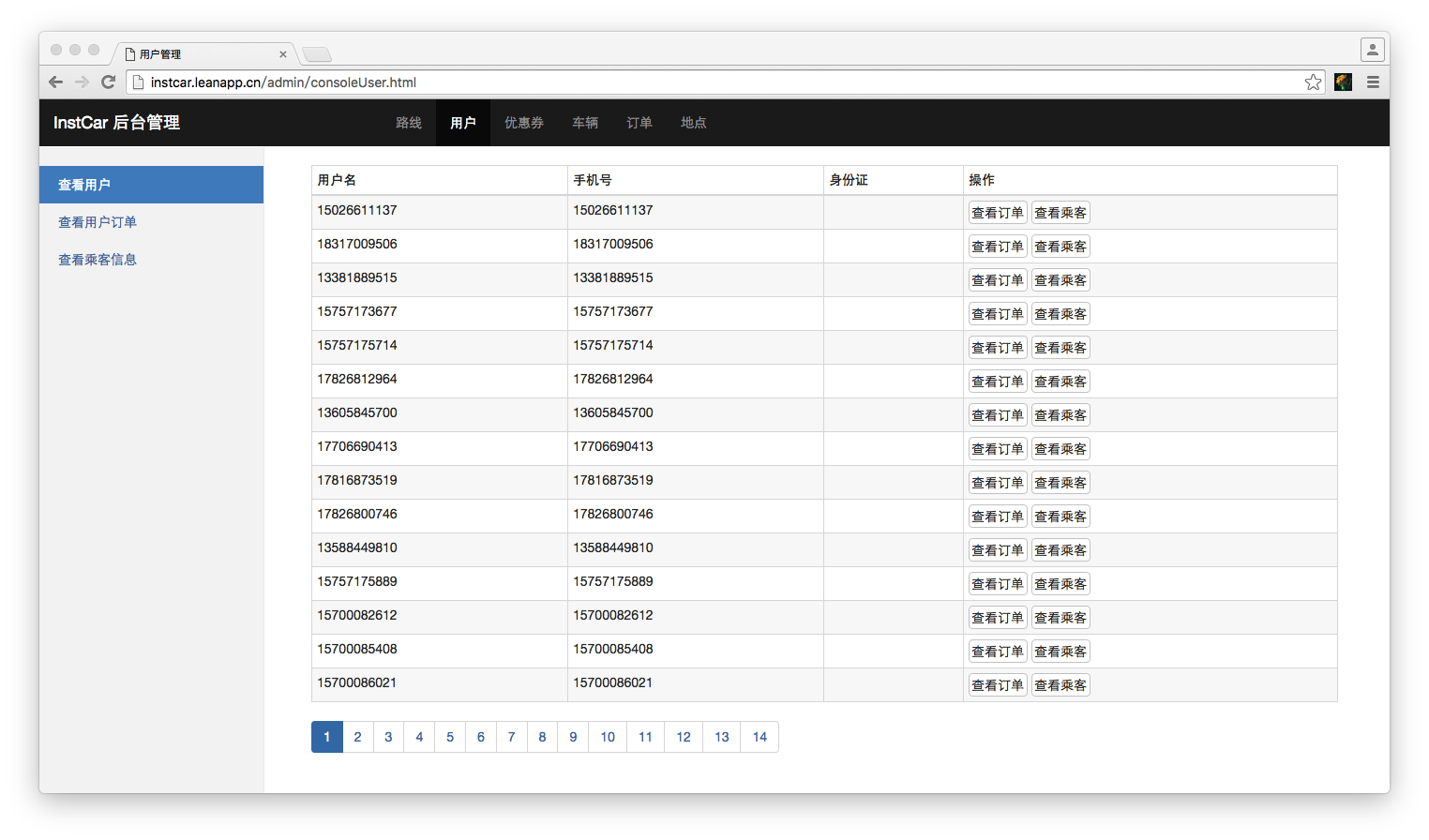
由上图可知，这些信息主要包括：路线信息（始发地，目的地，停靠地，发车日期等），票务信息（价格等），乘客信息（乘客人数等），以及班次信息（车辆信息，司机信息等）。可见，管理员端对于路线有更加明确的掌握。

从管理员端还可以发布路线：

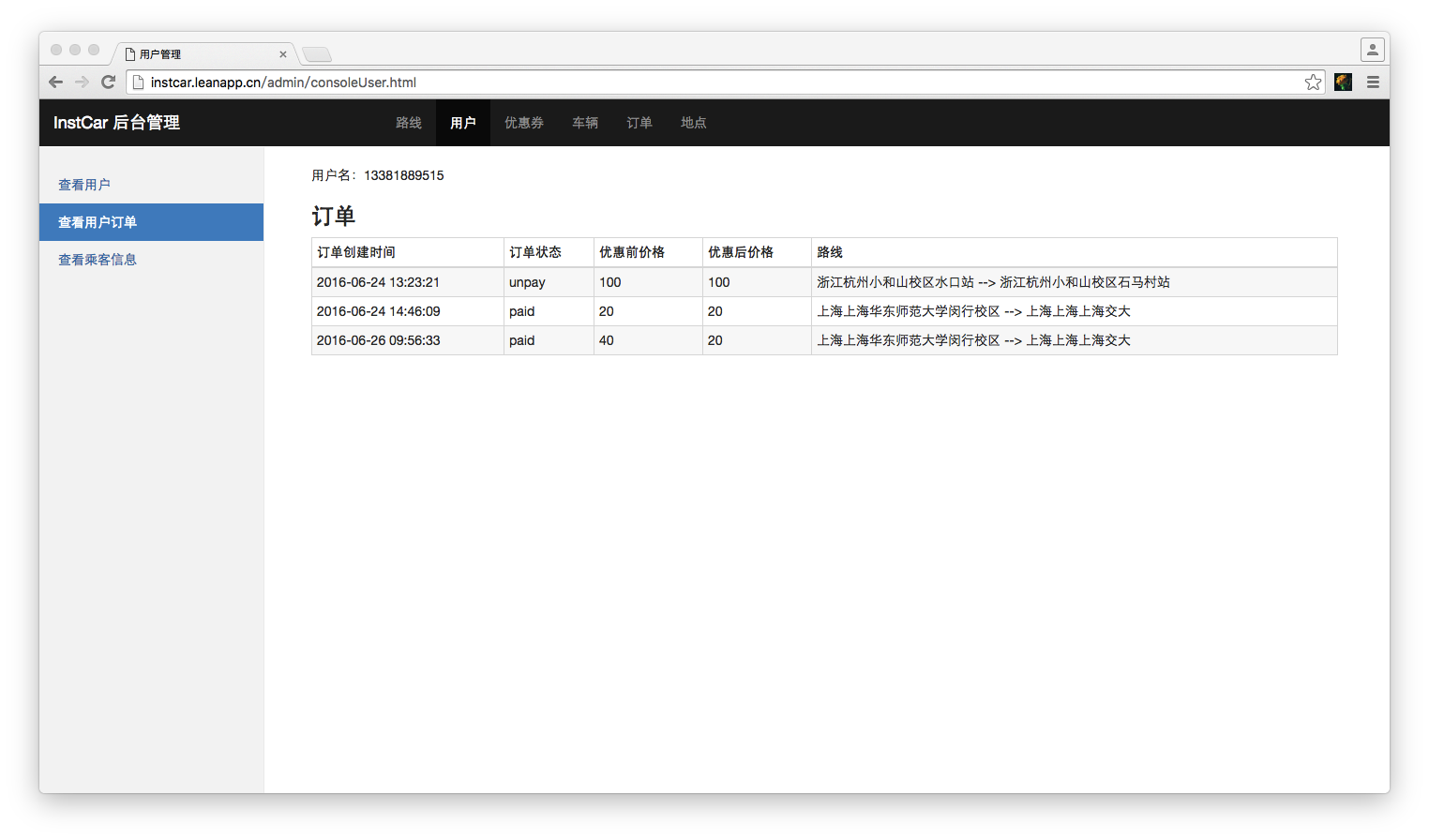


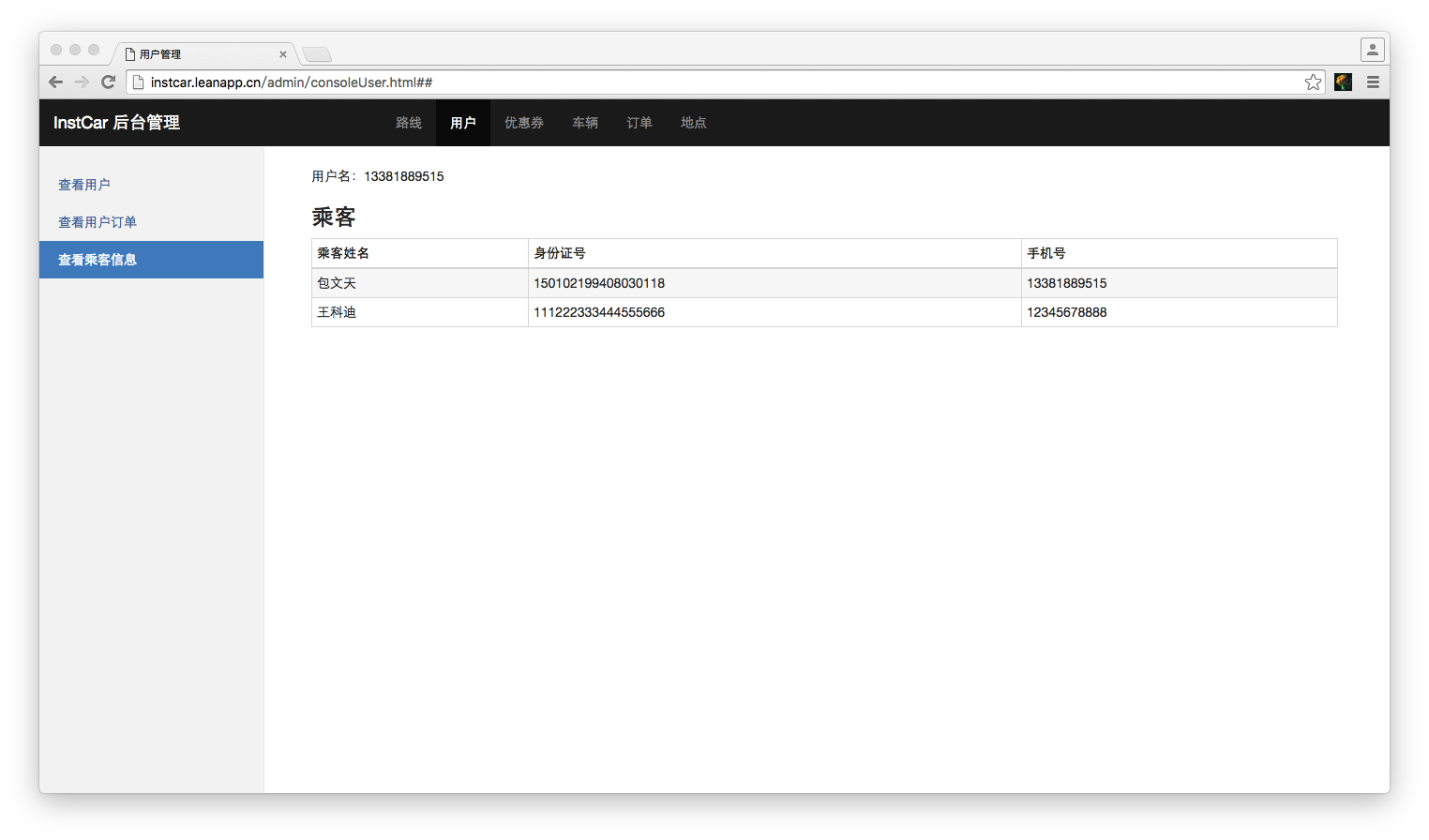
2.用户管理

从管理员端能看到每位乘客的信息：



具体包括乘客的订单情况，以及乘客的个人信息：



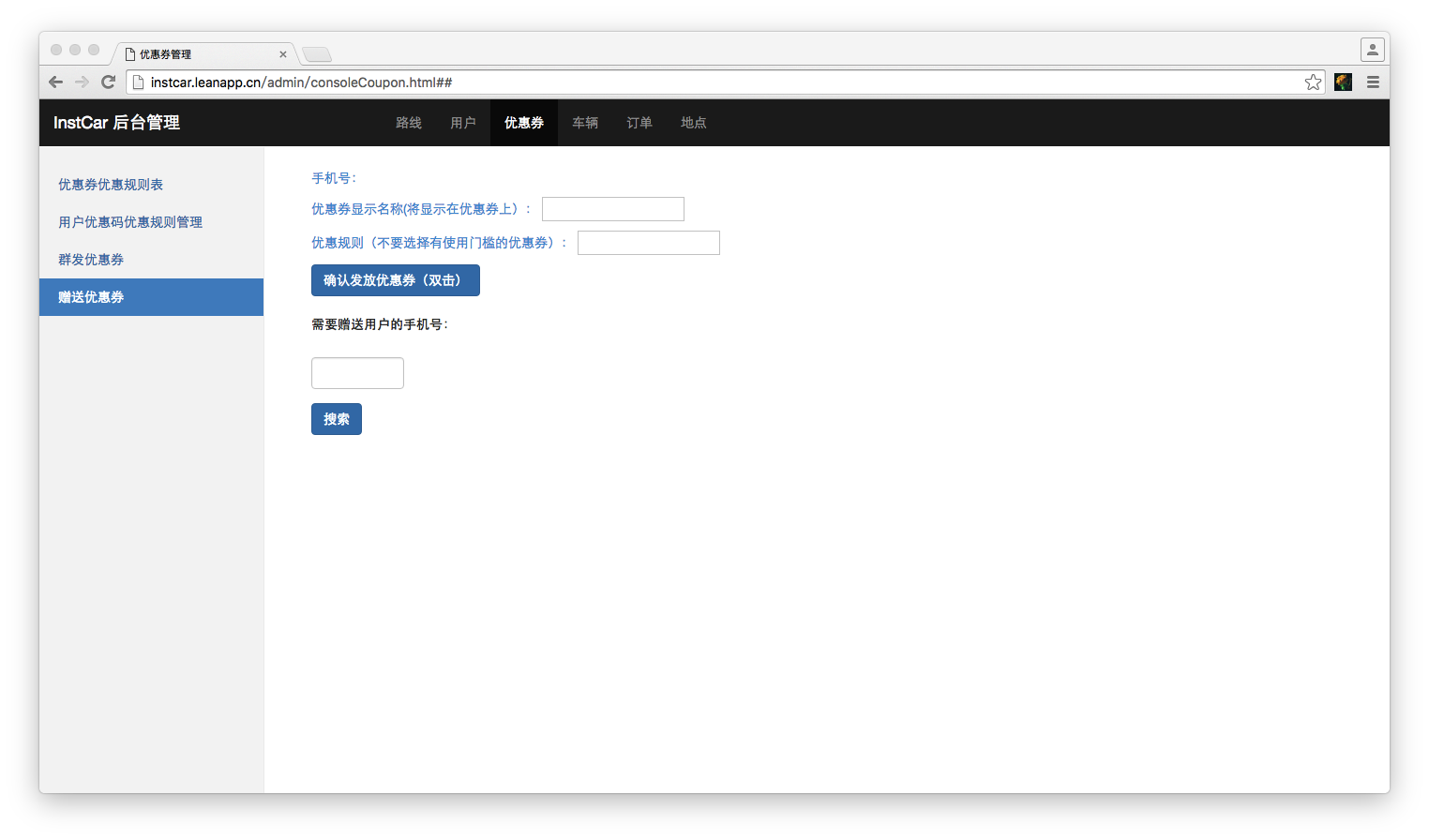


3.优惠券管理

优惠券管理主要涉及优惠规则的设定，以及优惠券的分发：



选择赠送优惠券，可以根据用户的手机号等信息向指定用户派发优惠券：



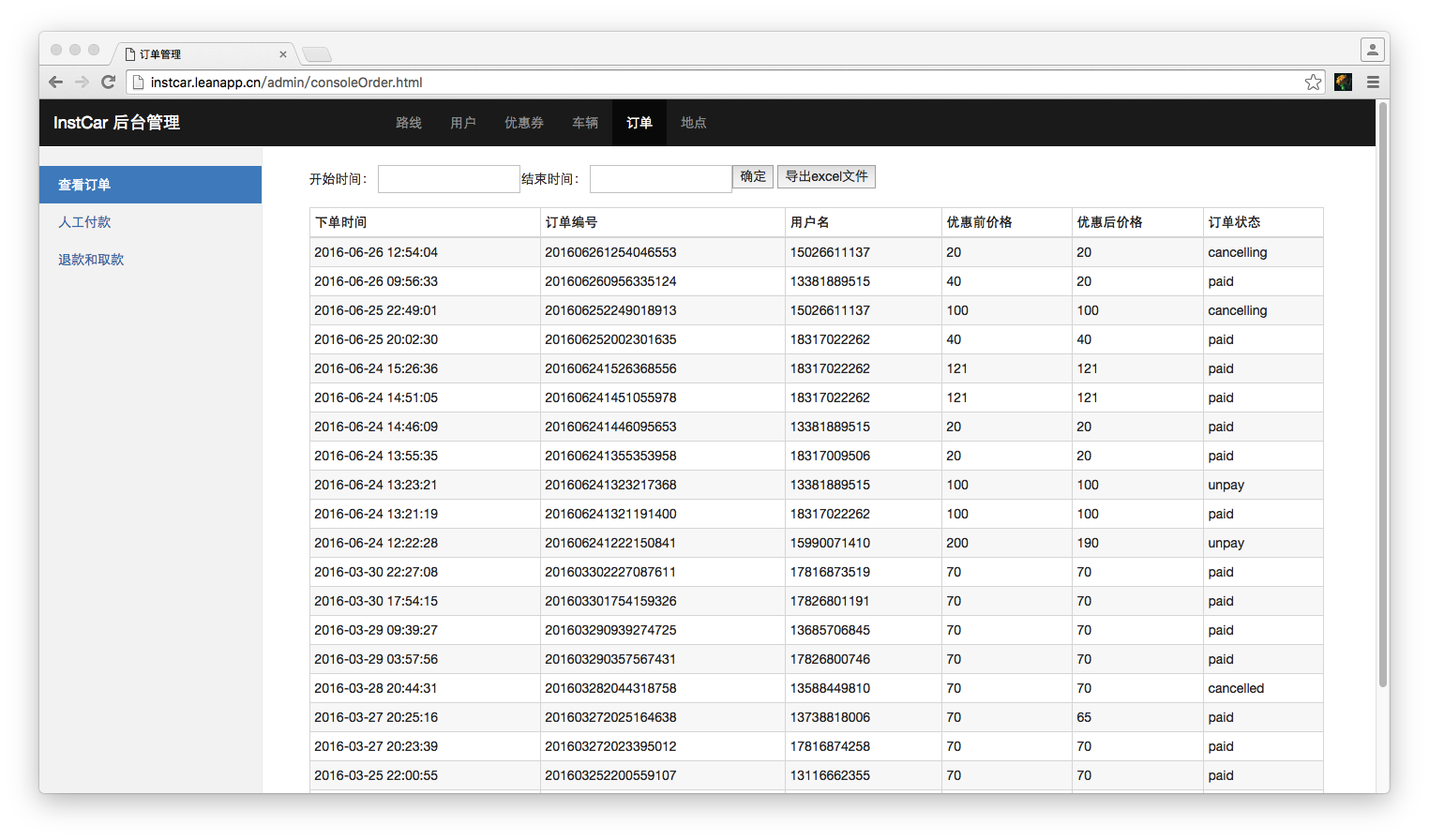
4.车辆管理

车辆管理可以查看现有车辆的信息，以及添加新的车辆：



5.订单管理

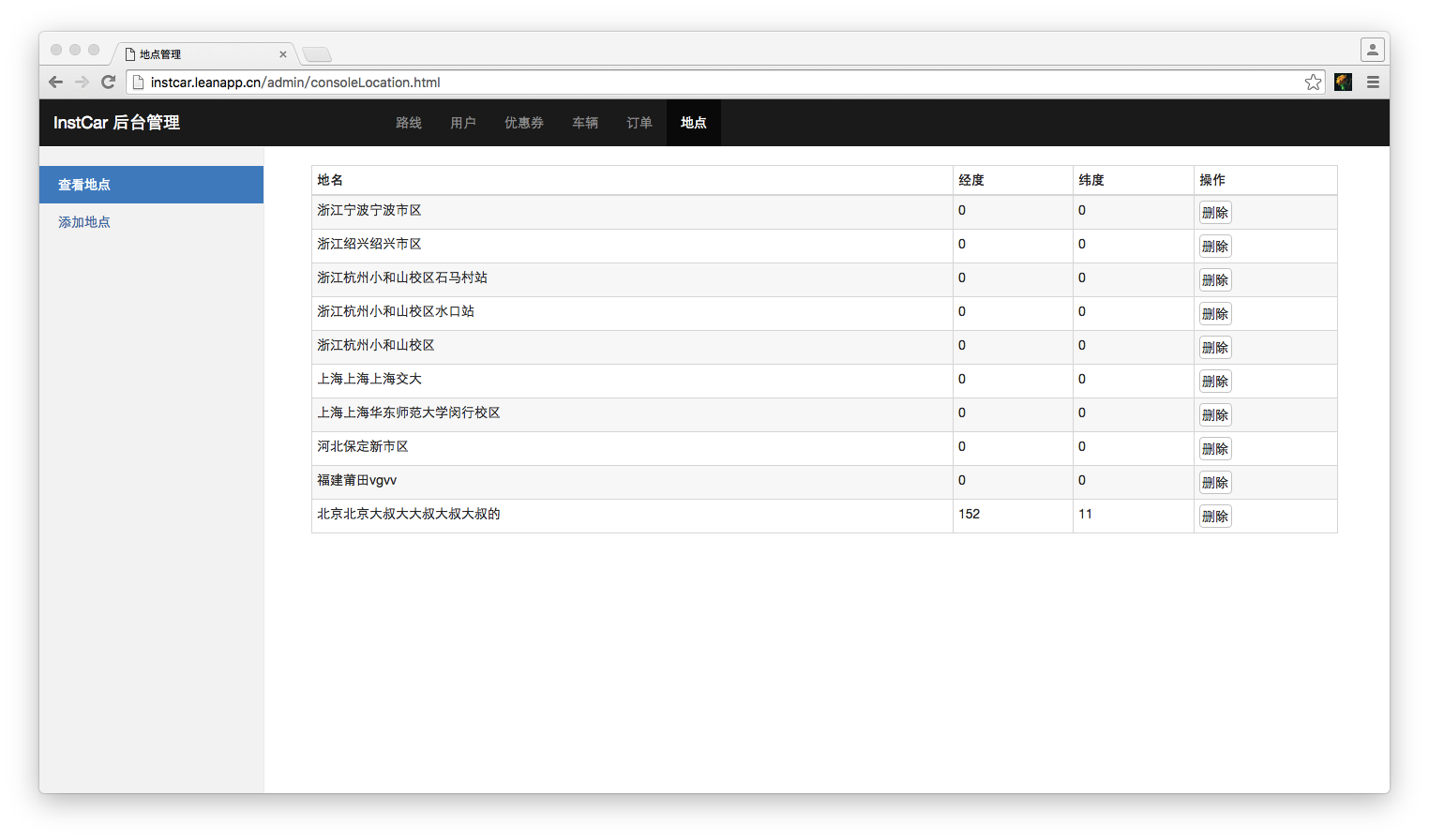
订单管理方便管理员在后台监控订单生成以及处理情况，具体可以查看订单的支付状态，优惠状态等，以及处理退订申请。





6.地点管理

地点管理使管理员能够根据用户提交路线的申请情况增设线路。



### 4.2.3性能

方便：图形界面，简洁直观，方便管理员处理后端数据。

高效：与数据库链接良好，相应快速。

### 4.2.4输入项

管理员导入的数据库。

### 4.2.5输出项

无

### 4.2.6局部变量

无

### 4.2.7设计方法

管理员端采用项目组自己设计的网页，php语言方便与数据库交换数据，图形化界面方便处理用户请求。

### 4.2.8 UML类图



### 4.2.9接口

图形界面

### 4.2.10存储分配

数据托管于云端Leancloud

### 4.2.11限制条件

无

### 4.2.12测试计划

参见《组装测试计划》与《测试计划》。

### 4.2.13尚未解决的问题

至详细设计止，本模块未发现尚未解决的问题。

# 编程协定

## 操作系统

开发阶段：

开发平台因为各个组员的开发环境不同，涉及Windows7/8和macOS

前端：AngularJS 1.5.7

后端：NodeJS 4.4.6

数 据 库：Leancloud

文档工具：Microsoft Office 2010

作图工具：Rational Rose 2003

使用阶段：

浏览应用：Chrome, Wechat

## 编程工具

Sublime Text 3.1.1.4

Atom

Git

## 编辑、调试、联接程序

Sublime Text 3.1.1.4

Atom

Git

## 编译工具

Sublime Text 3.1.1.4

Atom

Git

## 模拟、仿真数据

数据使用根据实际情况预测的仿真数据。

## 诊断、测试程序

Sublime Text 3.1.1.4

Atom

Git

## 检测程序

Sublime Text 3.1.1.4

Atom

Git

# 数据结构

## 物理结构



## 库结构

InstCar项目致力于开发真实可用的巴士平台，数据库共包括用户，地点，优惠券，优惠方式，订单，路线，计划，票务，交易记录凡九项。

## 链路结构

无

## 表结构

\_User Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| ObjectId | String | Yes | Yes |
| WxopenId | String | Yes | No |
| ACL | ACL | Yes | No |
| Password | String | Yes | No |
| Username | String | Yes | No |
| Emailverified | Boolean | Yes | No |
| Balance | Number | Yes | No |
| Mobilephonenumber | String | Yes | No |
| Idnumber | Number | Yes | No |
| Importfromparse | Boolean | Yes | No |
| Mobilephoneverified | Boolean | Yes | No |
| Createdat | Data | Yes | No |
| Updatedat | Data | Yes | No |
| Salt | String | No | No |
| Email | String | No | No |
| Sessiontoken | String | No | No |
| Isdeleted | Boolean | No | No |
| Completed | Boolean | No | No |
| Mydiscountcode | String | No | No |
| Ismember | Boolean | No | No |
| Idnumber | String | No | No |
| Agentinfo | String | No | No |
| Myagent | String | No | No |
| Authdata | Object | No | No |

Area Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| Destroutenumber | Number | Yes | No |
| ACL | ACL | Yes | No |
| Name | String | Yes | No |
| Type | String | Yes | No |
| Pid | String | Yes | No |
| Sourceroutenumber | Number | Yes | No |
| Ctreatedat | Date | Yes | No |
| Updatedat | Date | Yes | No |
| Isdeleted | Boolean | No | No |

Cartype Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| ACL | ACL | Yes | No |
| Type | String | Yes | No |
| Company | String | Yes | No |
| Isdeleted | Boolean | Yes | No |
| Totalseat | Number | Yes | No |
| createdat | Date | Yes | No |
| updatedat | Date | Yes | No |
| Carsppullier | Pointer | No | No |

Coupon Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| Discountrule | Pointer | Yes | No |
| Isdeleted | Boolean | Yes | No |
| ACL | ACL | Yes | No |
| Validprice | Number | Yes | No |
| Isused | Boolean | Yes | No |
| Type | String | Yes | No |
| Isvalid | Boolean | Yes | No |
| Validendtime | Date | Yes | No |
| Order | Pointer | Yes | No |
| Validstarttime | Date | Yes | No |
| Discountprice | Number | Yes | No |
| Codenumber | Number | Yes | No |
| User | Pointer | Yes | No |
| Show | String | Yes | No |
| Ctreatedat | Date | Yes | No |
| Ipdatedat | Date | Yes | No |
| Name | String | No | No |
| Validsource | Pointer | No | No |
| Validdest | Pointer | No | no |

DiscountRule Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| ACL | ACL | Yes | No |
| Description | String | Yes | No |
| Isdeleted | Boolean | Yes | No |
| Type | String | Yes | No |
| Discountfunction | String | Yes | No |
| Createdat | Date | Yes | No |
| Updatedat | Date | Yes | no |

Order Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| ACL | ACL | Yes | No |
| Coupons | Relation | Yes | No |
| Passengers | Relation | Yes | No |
| Isdeleted | Boolean | Yes | No |
| Ordernumber | String | Yes | No |
| Pricetopay | Number | Yes | No |
| State | String | Yes | No |
| User | Pointer | Yes | No |
| Tickets | Relation | Yes | No |
| Agentcommision | Number | Yes | No |
| Createdat | Date | Yes | No |
| Updatedat | Date | Yes | No |
| Route | Pointer | Yes | No |
| Price | Number | Yes | No |

Passenger Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| ACL | ACL | Yes | No |
| Isdeleted | Boolean | Yes | No |
| Phone | String | Yes | No |
| Name | String | Yes | No |
| Idnumber | String | Yes | No |
| User | Pointer | Yes | No |
| Createdat | Date | Yes | No |
| Updatedat | Date | Yes | no |

Route Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| ACL | ACL | Yes | No |
| Destlng | Number | Yes | No |
| Isdeleted | Boolean | Yes | No |
| Starttime | Date | Yes | No |
| Sourcelng | Number | Yes | No |
| Sourcefullname | String | Yes | No |
| Duration | Number | Yes | No |
| Source | Pointer | Yes | No |
| Sourcebusstop | String | Yes | No |
| Destfullname | String | Yes | No |
| Isfinished | Boolean | Yes | No |
| Sourcelat | Number | Yes | No |
| Notice | String | Yes | No |
| Destbustop | String | Yes | No |
| Dest | Pointer | Yes | No |
| Price | Number | Yes | No |
| Destlat | Number | Yes | No |
| Createdat | Date | Yes | No |
| Updatedat | Date | Yes | No |
| Passengernumber | Number | Yes | No |

Schedule Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| ACL | ACL | Yes | No |
| Takenseat | Number | Yes | No |
| Isdeleted | Boolean | Yes | No |
| Route | Pointer | Yes | No |
| Isfinished | Boolean | Yes | No |
| Totalseat | Number | Yes | No |
| Cycher | String | Yes | No |
| Code | String | Yes | No |
| Createdat | Date | Yes | No |
| Updatedat | Date | Yes | No |
| Drivername | String | Yes | No |
| Password | String | no | No |
| Patatenumber | String | no | No |
| Tellername | String | no | No |
| Driverphone | String | no | No |
| company | String | no | no |

Token Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| ACL | ACL | Yes | No |
| Schedule | Pointer | Yes | No |
| Isdeleted | Boolean | Yes | No |
| Seatnember | Number | Yes | No |
| State | String | Yes | No |
| Route | Pointer | Yes | No |
| Passenger | Pointer | Yes | No |
| Code | String | Yes | No |
| Order | String | Yes | No |
| Createdat | Date | Yes | No |
| Updatedat | Date | Yes | No |

Transaction Table

|  |  |  |  |
| --- | --- | --- | --- |
| 物理名称 | 数据类型 | 必须的 | Prime Key |
| Objectid | String | Yes | Yes |
| ACL | ACL | Yes | No |
| Order | Pointer | Yes | No |
| User | Pointer | Yes | No |
| Createdat | Date | Yes | No |
| Updatedat | Date | Yes | No |
| Amount | Number | Yes | No |
| Isdeleted | Boolean | Yes | No |
| Chargeid | String | Yes | No |
| Extra | String | Yes | No |
| Type | String | Yes | No |
| Currency | String | Yes | No |
| Isfinished | Boolean | Yes | No |
| Paid | String | Yes | No |
| Channel | String | Yes | No |
| Time | Date | No | No |
| Otherid | String | No | No |
| Transaction | String | No | No |
| Refundid | String | No | No |

## 内部存贮区分配

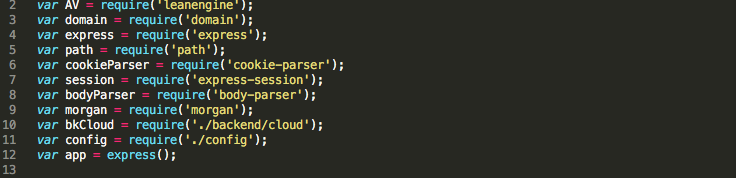
本软件数据托管于LeanCloud云端数据库，存储分配方案由云端负责。

# 逻辑结构

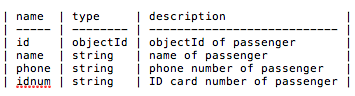
## 常量定义

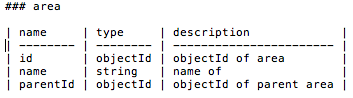
【此处加入常量定义】

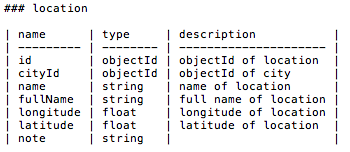
## 变量定义

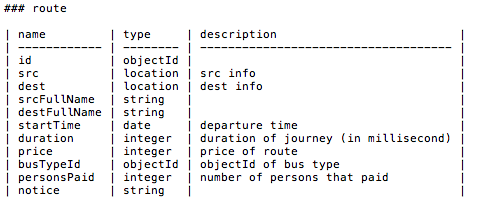


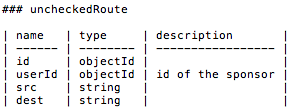
## 结构定义

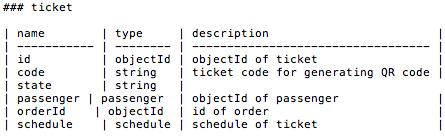


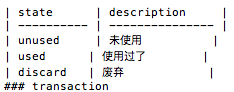


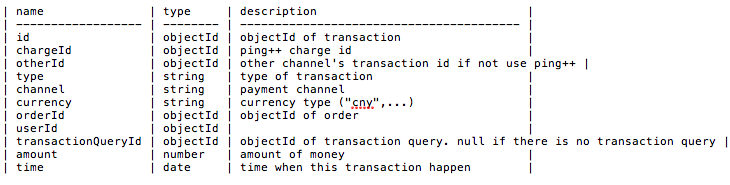


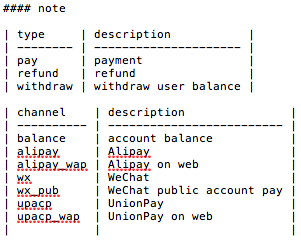


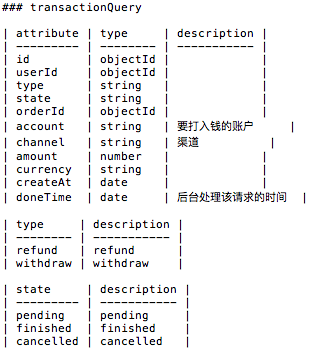


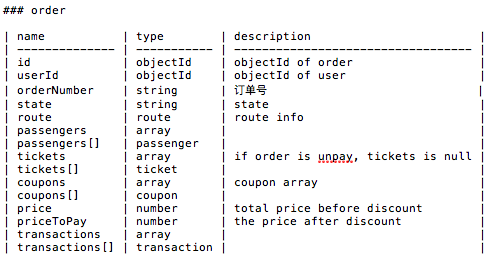


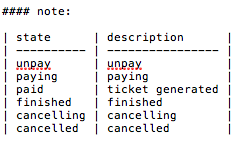




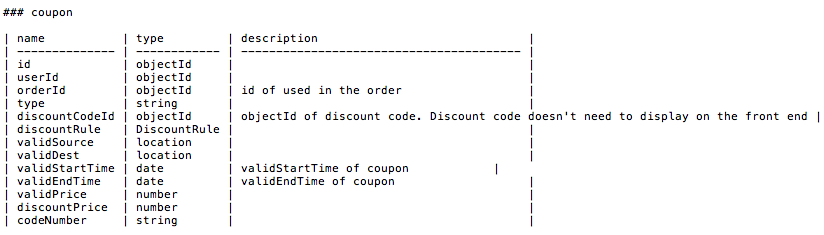


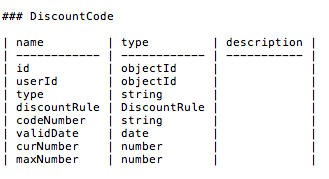


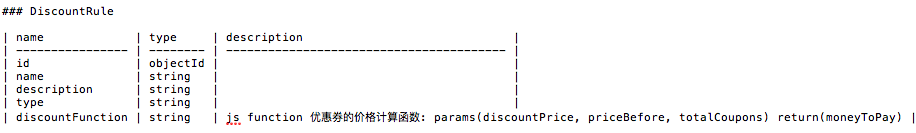


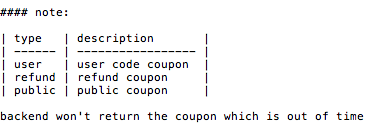












详见/doc/api.md

## 全局定义

无全局定义

# 源文件列表

