计算机科学与工程学院

课程设计报告

**题目全称：航空订票系统Android客户端**

**题目难度等级：4**

**指导老师： 程红蓉 职称：教授**

**学生姓名：李文靖 学号：2014060102012 班号：2014060102**

|  |  |  |  |
| --- | --- | --- | --- |
| **设计过程（30分）** | **课程设计报告（30分）** | **课程设计程序（40分）** | **总分** |
|  |  |  |  |

**备注：如参加答辩，请指导教师只给出设计过程30分，课程设计报告、课程设计程序由答辩老师给出。**

**如参加答辩，请答辩老师签字：**

**主要任务：**

**管理员可以录入或者导入/导出各个航线的航班信息，可以修改航班信息，可以增加/删除/修改操作工作员信息。操作工作员可以实现订票订票、退订，查看航线.**

**详细功能描述：**

1. **录入航线信息:每条航线信息包括航班号、飞机号、目的地、订票数、余票数共5项。**
2. **订票业务:客户信息包括姓名, 航班号, 座位号。有新客户订票时, 先输入客户的姓名，身份证号和他提出的航班号, 查询该航线的订票情况, 若有余票, 则为客户办理订票手续, 分配/选择户一个座位号。若无余票, 则输出客满信息。如果该航班已经无票，可以查询输出可选择的其他航班信息。**
3. **退票业务:根据客户提出的航班号和姓名, 办理退票, 从数据库中删除该客户的信息, 并修改数据库中相应航线的订票数和余票数。**
4. **修改航班信息：当航班信息改变可以修改航班数据文件。**
5. **输出全部航线信息和全部客户信息。**
6. **退出系统。要求用数据库保存信息。**

**预期成果或目标：、**

**软件代码，课程设计报告**

**指导老师评语：**

**指导教师签字：**

**摘要**

飞机航空系统作为一种交通运输方式，在现代社会越来越显得重要。然而客户从购买机票到前往机场乃至登机都会浪费很多的时间。此时，寻找一种更加安全、便捷的购票方式就成为了一个亟待解决的问题。

智能手机在人们生活中正扮演者越来越重要的角色，尤其是大量的Android手机，因其相对廉价而性能又好，占据了大量的市场份额。

在本次课程设计中，将Android操作系统和航空订票系统结合起来，实现一个基于Android系统的航空订票系统Android客户端，改系统可供客户查询航班、购票，还支持航空工作人员即使对数据库进行操作。

**关键词：航空系统，Android，购票，客户端**

## 目录

## 第一章 引言

### 1.1 研究意义----------------------------------------------------------------------- 1

### 1.2 研究现状----------------------------------------------------------------------- 1

### 1.3 工作内容----------------------------------------------------------------------- 2

## 第二章 相关技术基础

### 2.1 AndroidStudio的基本使用------------------------------------------------- 3

### 2.2 编译环境搭建及使用-------------------------------------------------------- 3

### 2.3 Java编程------------------------------------------------------------------------- 3

### 2.4 xml界面编程---------------------------------------------------------------------3

### 2.6 本章小结--------------------------------------------------------------------------3

## 第三章 系统需求分析

### 3.1 功能需求------------------------------------------------------------------------ 4

### 3.2 性能需求------------------------------------------------------------------------ 4

### 3.3 本章小结------------------------------------------------------------------------ 4

## 第四章 系统设计和实现

### 4.1 总体设计----------------------------------------------------------------------- 5

### 4.2 数据库设计-------------------------------------------------------------------- 6

### 4.3 详细设计和实现------------------------------------------------------------- 6

**4.3 本章小结----------------------------------------------------------------------- 15**

## 第五章 系统测试

### 5.1 测试环境------------------------------------------------------------------------ 16

### 5.2 测试用例和结果--------------------------------------------------------------- 16

### 5.3 本章小结------------------------------------------------------------------------ 36

## 第六章 总结

### 6.1 课程设计总结------------------------------------------------------------------- 37

### 6.2 课程设计展望------------------------------------------------------------------- 37

1. **引言**

**1.1 研究意义**

通过在Android系统，进行航空订票系统的设计与实现，充分运用所学的数据库、Java语言、数据结构、操作系统、设计模式等知识。了解一个较完整的Android 项目的构建过程，同时，通过较大量的编码提高需求分析、数据库设计、框架搭建、代码设计、编码及制图的能力。

**1.2 研究现状**

随着时代的进步社会分工的不断细化，各个行业联系的不断密切，人们出行越来越多，出行的方式也是百花齐放，作为空中重要的交通工具，飞机航空系统也在不断地得到优化，其中重要的一项就是飞机航空订票系统，因此好的订票系统关系着人们的日常出行方便与否，设计并编制出符合人们需要的航空订票系统是一项繁重而艰巨的任务。订票系统在现实生活中应用很广泛，通过这个系统可以使订票、退票等变得简便。与传统的人工用纸、笔来记录办理订票、退票等相比，使用这个系统也会增加工作效率随着科学技术的不断发展，计算机科学日渐成熟，其强大功能已为人们所深刻认识，它己进入人类社会的各个领域并发挥着越来越重要的作用。采用计算机进行信息化管理已成为衡量各个国家科学化和现代化的重要标志，而交通方式的全面自动化、信息化则是其中重要的组成部分，在很大程度上影响着企业的经济效益和社会效益。因此，本文所研究的航空订票系统具有一定的使用价值和现实意义。随着科技与经济的发展，越来越多的人选择乘飞机，这跟我国的经济增长有很大关系，人们在追求快节奏的生活方式，所以做飞机无疑成了首选。而且随着网络的盛行，航空订票系统就显得尤为重要，这个系统主要是为了方便大家，让大家能够快速、清晰、准确地了解航班信息，而不至于像以前那样排队等候，从而避免耽搁乘客大量的等待时间。完成此系统，需要综合运用数据结构课程中学到的几种典型数据结构，以及程序设计语言（Java语言），自行实现一个较为完整的应用系统的设计与开发。通过课程设计，自己通过系统分析、系统设计、编程调试，写实验报告等环节，进一步掌握应用系统设计的方法和步骤，灵活运用并深刻理解典型数据结构在软件开发中的应用。学会将知识应用于实际的方法，提高分析和解决问题的能力，增加综合能力。

Android是一种基于[Linux](http://baike.baidu.com/view/1634.htm" \t "/home/levent_j/文档\\x/_blank)的自由及开放源代码的[操作系统](http://baike.baidu.com/view/880.htm" \t "/home/levent_j/文档\\x/_blank)，主要使用于[移动设备](http://baike.baidu.com/view/8323830.htm" \t "/home/levent_j/文档\\x/_blank)，如[智能手机](http://baike.baidu.com/view/535.htm" \t "/home/levent_j/文档\\x/_blank)和[平板电脑](http://baike.baidu.com/view/74538.htm" \t "/home/levent_j/文档\\x/_blank)，由[Google](http://baike.baidu.com/view/105.htm" \t "/home/levent_j/文档\\x/_blank)公司和[开放手机联盟](http://baike.baidu.com/view/1245202.htm" \t "/home/levent_j/文档\\x/_blank)领导及开发。

Android 是运行于Linux kernel之上，但并不是[GNU](http://baike.baidu.com/view/36272.htm" \t "/home/levent_j/文档\\x/_blank)/Linux。因为在一般GNU/Linux 里支持的功能，Android 大都没有支持，包括[Cairo](http://baike.baidu.com/view/5207172.htm" \t "/home/levent_j/文档\\x/_blank)、[X11](http://baike.baidu.com/view/575115.htm" \t "/home/levent_j/文档\\x/_blank)、[Alsa](http://baike.baidu.com/view/1558228.htm" \t "/home/levent_j/文档\\x/_blank)、[FFmpeg](http://baike.baidu.com/view/856526.htm" \t "/home/levent_j/文档\\x/_blank)、[GTK](http://baike.baidu.com/view/173704.htm" \t "/home/levent_j/文档\\x/_blank)、[Pango](http://baike.baidu.com/view/2941612.htm" \t "/home/levent_j/文档\\x/_blank)及[Glibc](http://baike.baidu.com/view/1323132.htm" \t "/home/levent_j/文档\\x/_blank)等都被移除掉了。Android又以Bionic 取代Glibc、以Skia 取代Cairo、再以[opencore](http://baike.baidu.com/view/3905960.htm" \t "/home/levent_j/文档\\x/_blank)取代[FFmpeg](http://baike.baidu.com/view/856526.htm" \t "/home/levent_j/文档\\x/_blank)等等。Android 为了达到商业应用，必须移除被GNU GPL授权证所约束的部份，例如Android将驱动程序移到 Userspace，使得Linux driver 与 Linux kernel彻底分开。Bionic/Libc/Kernel/ 并非标准的Kernel header files。Android 的 Kernel header 是利用工具由 Linux Kernel header 所产生的，这样做是为了保留常数、数据结构与宏。

**1.3 工作内容**

课程设计中，将会接触到以下几种工作：

1. 需求分析。针对课程项目命题和要求分析其主要和次要功能点。
2. 学习调研。由于之前并没用经历过一个完整的Android项目的构建，因此学习和调研相关技术是十分必要的。
3. 代码设计。杂乱无章的代码会导致程序效率低下、错误百出；而组织严密、结构清晰的代码及设计模式有助于我们理清思路，减少错误。
4. 代码编码。这一项直接与编程语言和系统调用相关，想要在这方面有所建树，则必须通过大量的练习来实现。
5. **相关技术基础**

**2.1 AndroidStudio的基本使用**

这一部分主要包括：AndroidStudio的基本功能，Gradle构建项目方法，将项目代码托管到git上

**2.2 编译环境搭建及使用**

这一部分主要包括：jdk的安装及配置环境变量，sdk的下载与安装，代码版本控制工具git命令的配置和使用

**2.3 Java编程**

这一部分主要包括：项目框架的基本搭建思路，设计模式的合理运用，变量及方法的设计，各类常见的语法错误，异常处理的合理使用

**2.4 xml界面编程**

这一部分主要包括：app所有界面的注册及设计，图片（icon、logo）、颜色、字符串、数组、等资源的部署，drawable\value\layout等文件目录的构建

**2.6 本章小结**

要实现本次课程设计，首先必要的知识储备是必不可少的。在拥有足够的知识储备后，对编程环境的熟悉，编程工具的熟练使用都有助于我们写出高质量的程序。我在本次课程设计中用到的工具一共有如下几种：

1. jvm java虚拟机
2. sdk Android开发工具
3. Android Studio Android开发目前来讲最好用的IDE
4. gradle Android项目的构建工具，可以方便的添加第三方依赖
5. git 非常优秀的开源代码版本控制工具，并在云端提供远程代码库
6. **系统需求分析**

**3.1 功能需求**

1.实现Android客户端，有普通客户和管理员两种身份，普通客户可根据出发地、目的地以及出行日期查询航班相关信息，通过填写姓名、身份证号、航班号以及出行日期来订票，可通过客户姓名、身份证号、出行日期来查询已订机票信息，并选择是否退票。管理员可以检视所有航班及客户信息，并对数据进行增删改查。

2在浏览器中，可通过登陆Bmob云后台来对数据库中数据直接进行操作。

**3.2 性能需求**

尽可能的将代码模块化，封装好完备的功能方法并将它们按功能的不同分别加入不同的类中，以此减少冗余代码命题高性能；通过Bmob后端云服务构建数据库，调用先关api对数据库进行操作。

**3.3 本章小结**

课程设计中，产出结果应为一个apk文件作为客户端安装包，后端数据库可直接通过浏览器进入Bmob后端云来操作。

1. **系统设计和实现**

**4.1 总体设计**

所有代码均托管在github上：<https://github.com/Levent-J/AirlineBookSystem>

分包:

Activity

Adapter

Base

Bean

Fragment

Net

Utils

Widget

App

Activity：

MainActivity

AboutActivity

AdminActivity

BoardingActivity

EditClientActivity

EditFlightActivity

GuideActivity

LoginActivity

PlaceListActivity

PriceActivity

Adapter:

AdminClientAdapter

AdminFlightAdapter

DynamicAdapter

Fragment:

AdminClientFragment

AdminFlightFragment

BookFragment

DynamicFragment

MainFragment

QueryFragment

Base:

BaseActivity

BaseFragment

BaseFragment

Bean:

Client

Flight

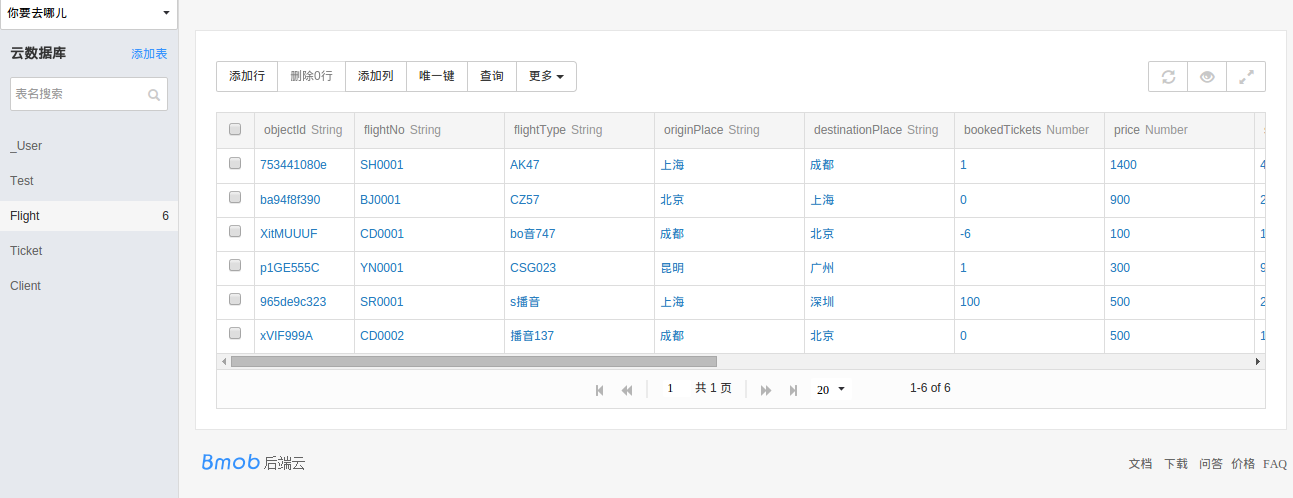
Utils:

DividerItemDecoration

Widget:

InputDialog

**4.2 数据库设计**



**4.3 详细设计和实现**

**4.3.1 增加客户信息**

private void addClient(String name, String idCard, String flightNo,String data) {

Client client = new Client();

client.setName(name);

client.setIdCard(idCard);

client.setFlightNo(flightNo);

client.setData(data);

client.save(new SaveListener<String>() {

@Override

public void done(String s, BmobException e) {

if (e==null){

isBooked = true;

}else {

Toa("网络繁忙，请稍后再试");

Log.e("Bmob","save error:"+e.getMessage());

}

}

});

}

**4.3.2 更新航班票量**

BmobQuery<Flight> query = new BmobQuery();

query.addWhereEqualTo("flightNo", flightNo);

query.findObjects(new FindListener<Flight>() {

@Override

public void done(List<Flight> list, BmobException e) {

if (e == null) {

String id = list.get(0).getObjectId();

int booked = list.get(0).getBookedTickets();

int surplus = list.get(0).getSurplusTickets();

Flight flight = list.get(0);

flight.setBookedTickets( booked + 1);

flight.setSurplusTickets( surplus - 1);

flight.update(id, new UpdateListener() {

@Override

public void done(BmobException e) {

if (e == null) {

isUpdated = true;

startActivity(new Intent(getActivity(), MainActivity.class));

getActivity().finish();

} else {

Toa("网络繁忙，请稍后再试");

Log.e("Bmob", "update error:" + e.getMessage());

}

}

});

} else {

Toa("网络繁忙，请稍后再试");

Log.e("Bmob", "change error:" + e.getMessage());

}

}

});

**4.3.3 查询航班信息**

BmobQuery<Flight> query = new BmobQuery<>();

query.addWhereEqualTo("originPlace", origin);

query.addWhereEqualTo("destinationPlace", destination);

query.addWhereEqualTo("data", day);

query.findObjects(new FindListener<Flight>() {

@Override

public void done(List<Flight> list, BmobException e) {

loading.setVisibility(View.GONE);

if (e == null) {

if (list.size() > 0) {

adapter.replaceData(list);

}else {

Toa("暂无航班");

}

} else {

Toa("网络繁忙，请稍后再试");

Log.e("Bmob", "error:" + e.getMessage());

}

}

});

**4.3.4 删除客户信息**

Client c = new Client();

c.setObjectId(client.getObjectId());

c.delete(new UpdateListener() {

@Override

public void done(BmobException e) {

if (e == null) {

Toast.makeText(mContext,"删除成功！",Toast.LENGTH\_SHORT).show();

} else {

Log.e("Bmob","delete error:"+e.getMessage());

}

}

});

**4.3.5 删除航班信息**

Flight f = new Flight();

f.setObjectId(flight.getObjectId());

f.delete(new UpdateListener() {

@Override

public void done(BmobException e) {

if (e==null){

Toast.makeText(mContext,"删除成功！",Toast.LENGTH\_SHORT).show();

}else {

Toast.makeText(mContext,"网络繁忙，请稍后再试",Toast.LENGTH\_SHORT).show();

Log.e("Bmob","delete error:"+e.getMessage());

}

}

});

**4.3.6 新增及修改航班信息**

private void commitFlight(String mOrigin, String mDestination, String mDate, String mStart, String mEnd,

String mFlight, String mType, int mPrice, int mBooked, int mSurplus) {

commit.setText("请等待");

commit.setEnabled(false);

Flight flight = new Flight();

flight.setOriginPlace(mOrigin);

flight.setDestinationPlace(mDestination);

flight.setData(mDate);

flight.setStartTime(mStart);

flight.setEndTime(mEnd);

flight.setFlightNo(mFlight);

flight.setFlightType(mType);

flight.setPrice(mPrice);

flight.setBookedTickets(mBooked);

flight.setSurplusTickets(mSurplus);

if (isNew){

createFlight(flight);

}else {

updateFlight(flight);

}

}

private void updateFlight(Flight flight) {

flight.update(objectId, new UpdateListener() {

@Override

public void done(BmobException e) {

if (e==null){

finish();

}else {

Toa("网络繁忙，请稍后再试");

commit.setText("提交");

commit.setEnabled(true);

Log.e("Bmob","save error:"+e.getMessage());

}

}

});

}

private void createFlight(Flight flight) {

flight.save(new SaveListener<String>() {

@Override

public void done(String s, BmobException e) {

if (e==null){

finish();

}else {

Toa("网络繁忙，请稍后再试");

commit.setText("提交");

commit.setEnabled(true);

Log.e("Bmob","save error:"+e.getMessage());

}

}

});

}

**4.3.7 日期选择器**

pickerPopWin = new DatePickerPopWin.Builder(getContext(), new DatePickerPopWin.OnDatePickedListener() {

@Override

public void onDatePickCompleted(int year, int month, int day, String dateDesc) {

data.setText(year+"-"+month+"-"+day);

}

})

.textConfirm("选择")

.textCancel("取消")

.btnTextSize(16)

.viewTextSize(25)

.colorCancel(Color.parseColor("#999999"))

.colorConfirm(Color.parseColor("#7C4DEF"))

.minYear(1970)

.maxYear(2020)

.dateChose("2016-07-02")

.build();

**4.3.8 城市选择列表**

private void startCityList(int code) {

Intent intent = new Intent(getActivity(),PlaceListActivity.class);

intent.putExtra("type",code);

startActivityForResult(intent,code);

}

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode==REQUEST\_ORIGIN){

originPlace.setText(data.getStringExtra("place"));

}else {

destinationPlace.setText(data.getStringExtra("place"));

}

}

**4.3.9 gradle app**

apply plugin: 'com.android.application'

android {

compileSdkVersion 23

buildToolsVersion "23.0.3"

useLibrary 'org.apache.http.legacy'

defaultConfig {

applicationId "com.levent\_j.airlinebooksystem"

minSdkVersion 14

targetSdkVersion 23

versionCode 1

versionName "1.0"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

compile fileTree(dir: 'libs', include: ['\*.jar'])

testCompile 'junit:junit:4.12'

compile 'com.android.support:appcompat-v7:23.3.0'

compile 'com.jakewharton:butterknife:7.0.1'

compile 'com.android.support:design:23.3.0'

compile 'com.android.support:cardview-v7:23.2.0'

compile 'com.android.support:recyclerview-v7:23.2.0'

compile 'cn.bmob.android:bmob-sdk:3.5.0'

compile 'cn.bmob.android:bmob-push:0.8'

compile 'com.brucetoo.pickview:library:1.1.1'

compile 'com.wang.avi:library:1.0.5'

compile 'com.nineoldandroids:library:2.4.0'

compile 'com.squareup.picasso:picasso:2.5.2'

}

**4.3.10 gradle project**

buildscript {

repositories {

jcenter()

}

dependencies {

classpath 'com.android.tools.build:gradle:1.3.0'

// NOTE: Do not place your application dependencies here; they belong

// in the individual module build.gradle files

}

}

allprojects {

repositories {

jcenter()

maven { url "https://raw.github.com/bmob/bmob-android-sdk/master" }

}

}

task clean(type: Delete) {

delete rootProject.buildDir

}

**4.4 本章小结**

在所有代码中，最重要的莫过于Activity和Fragment中的代码。主要包含了界面管理与控制，云数据获取，网络交互等功能。整个项目由gradle构建，结构清晰明了。

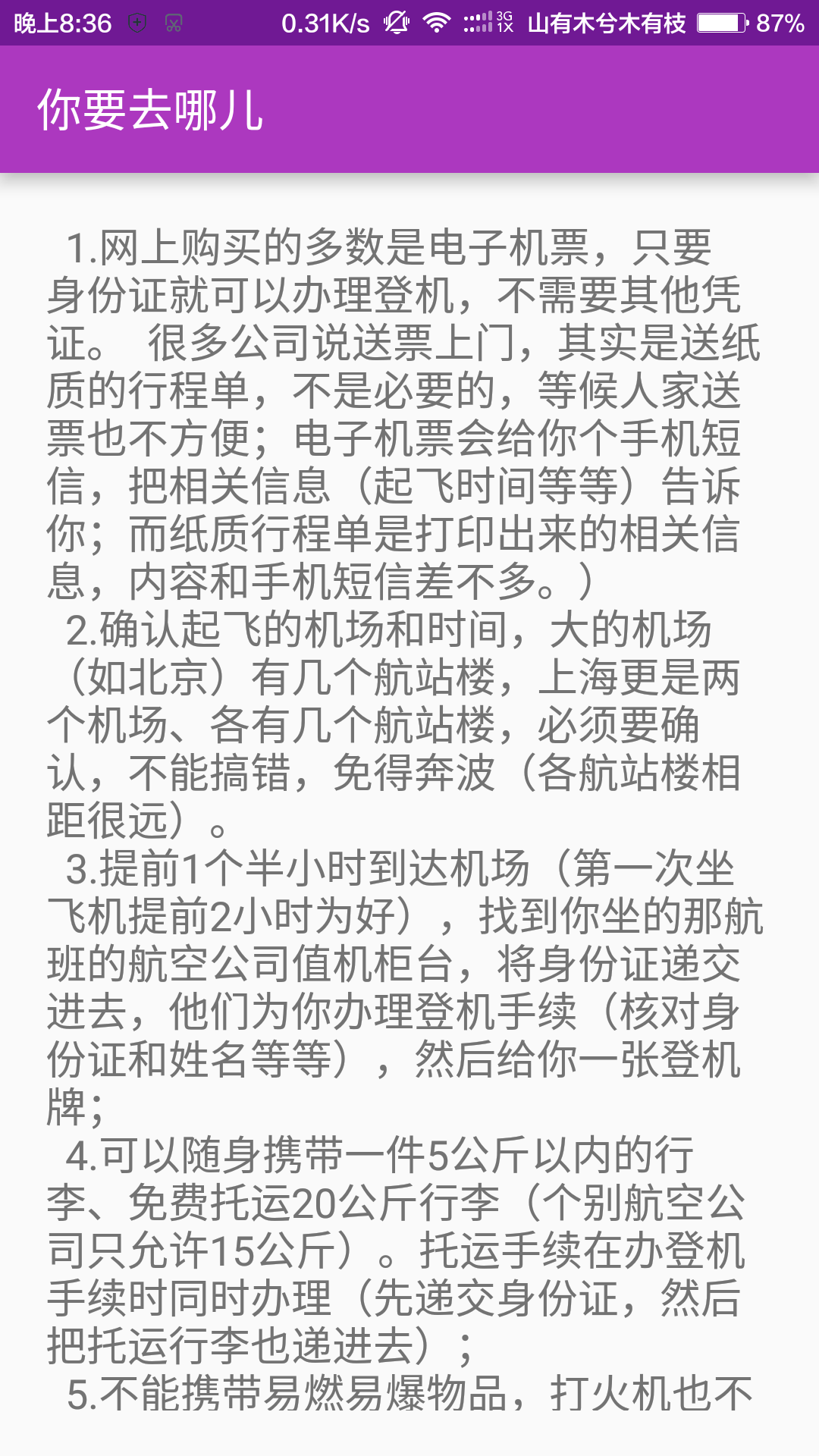
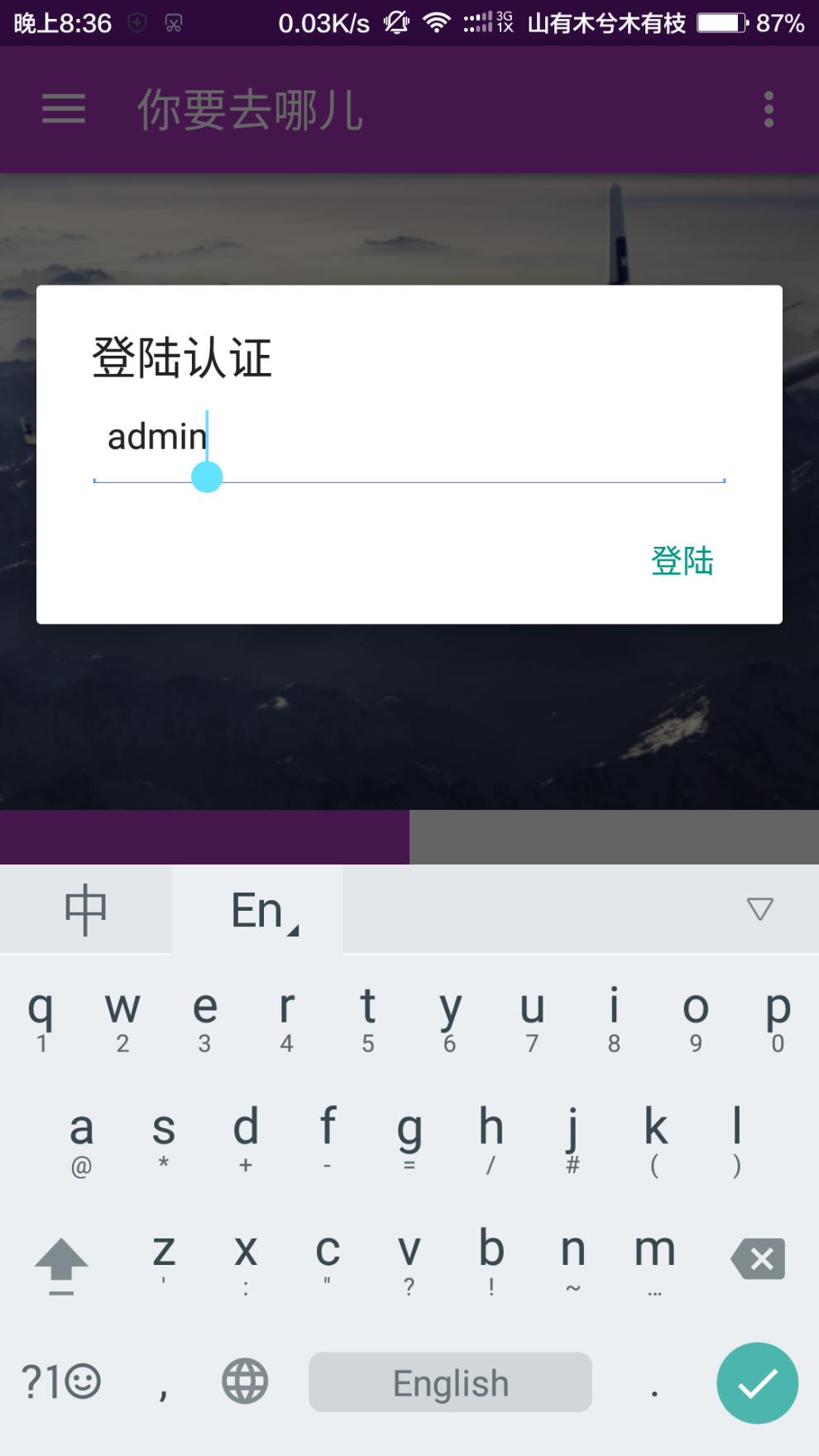
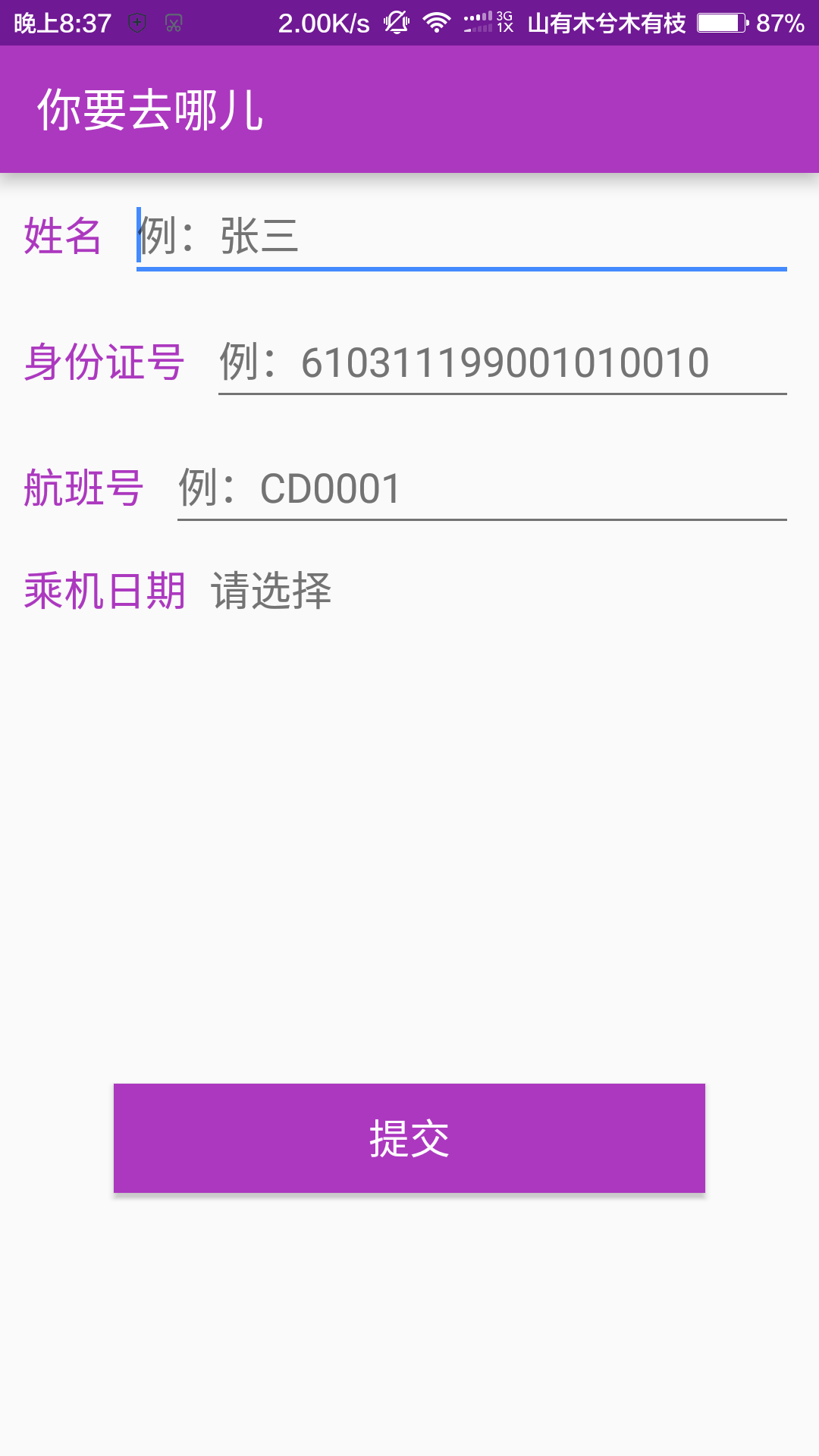
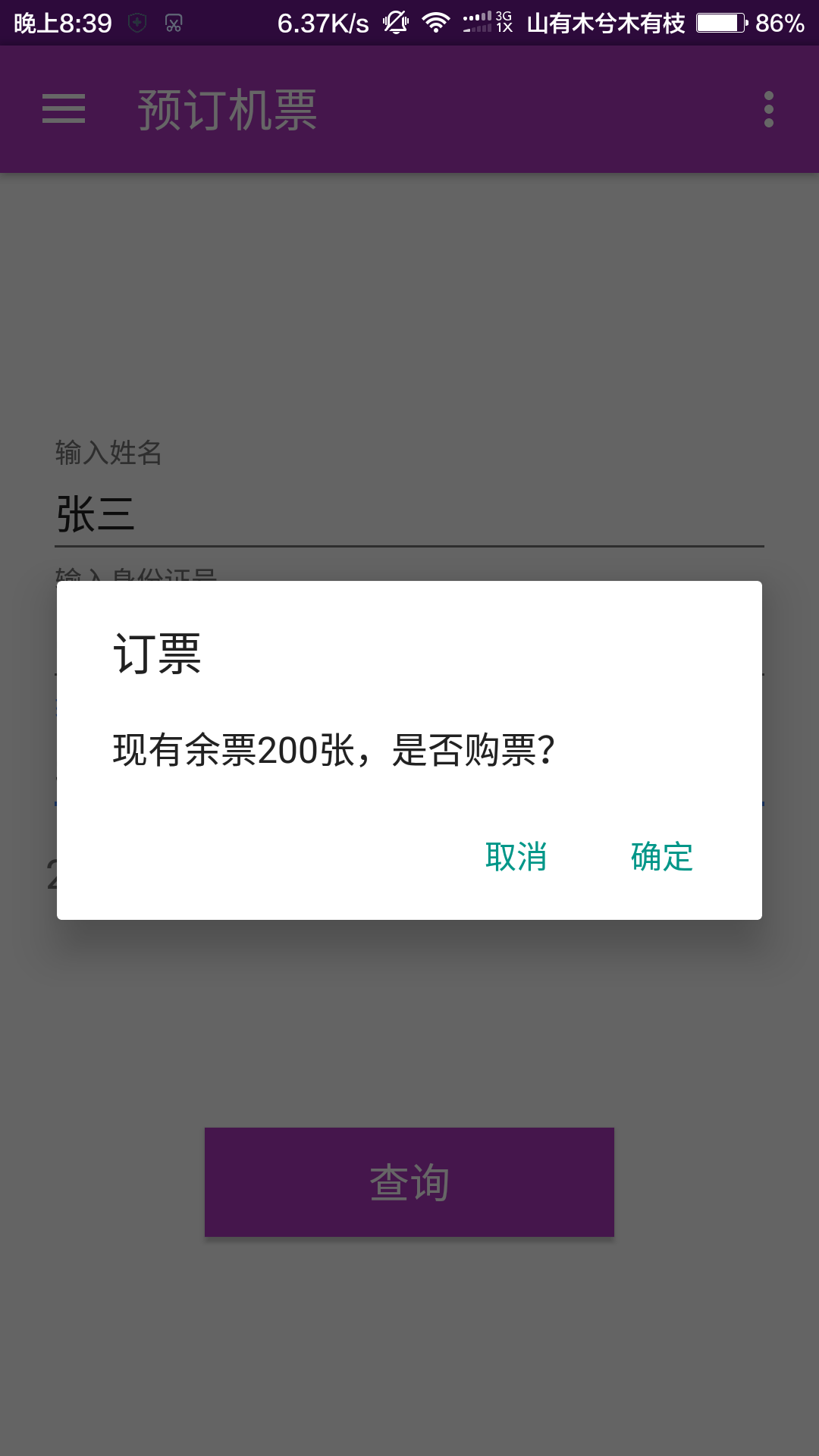
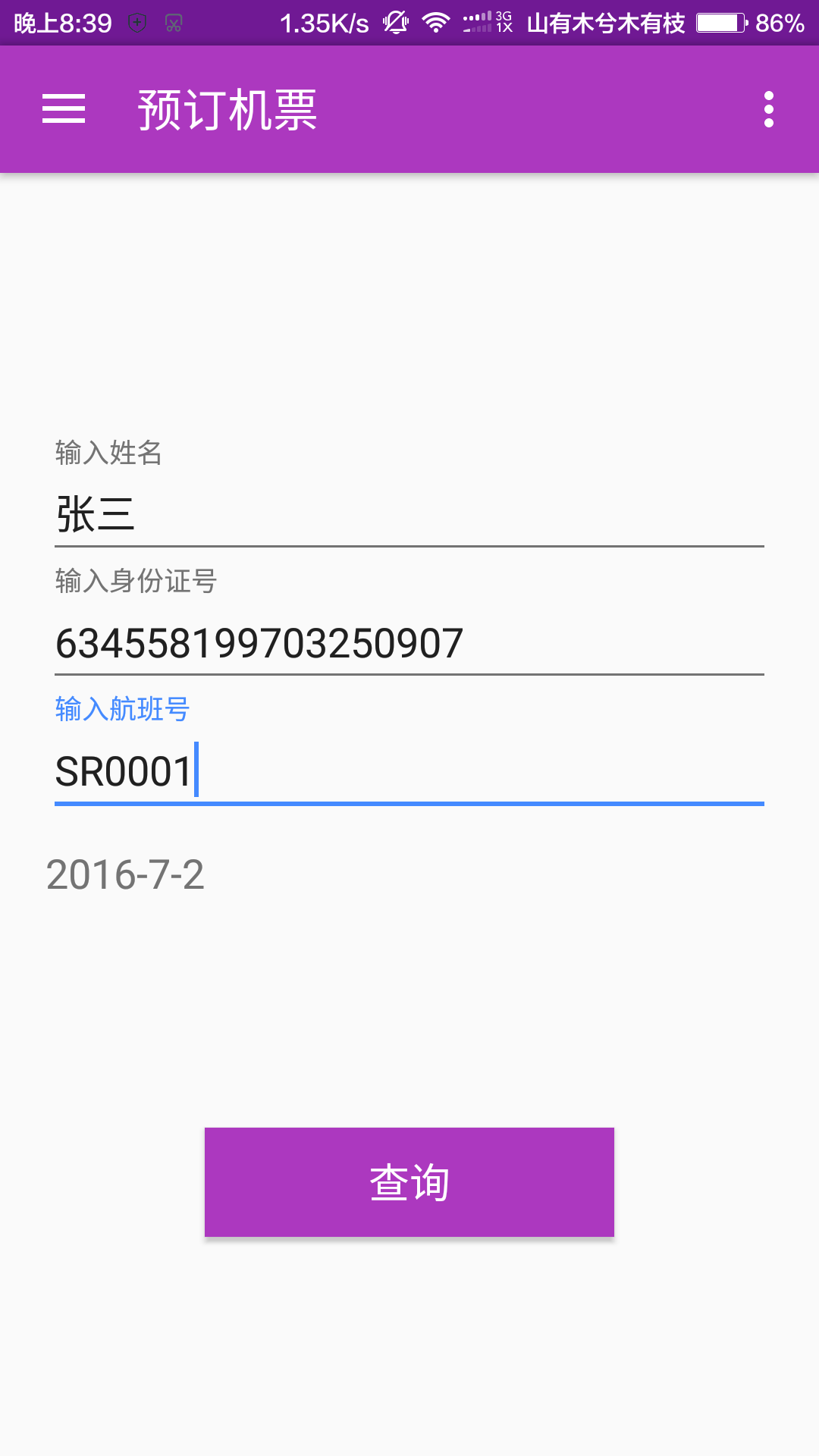
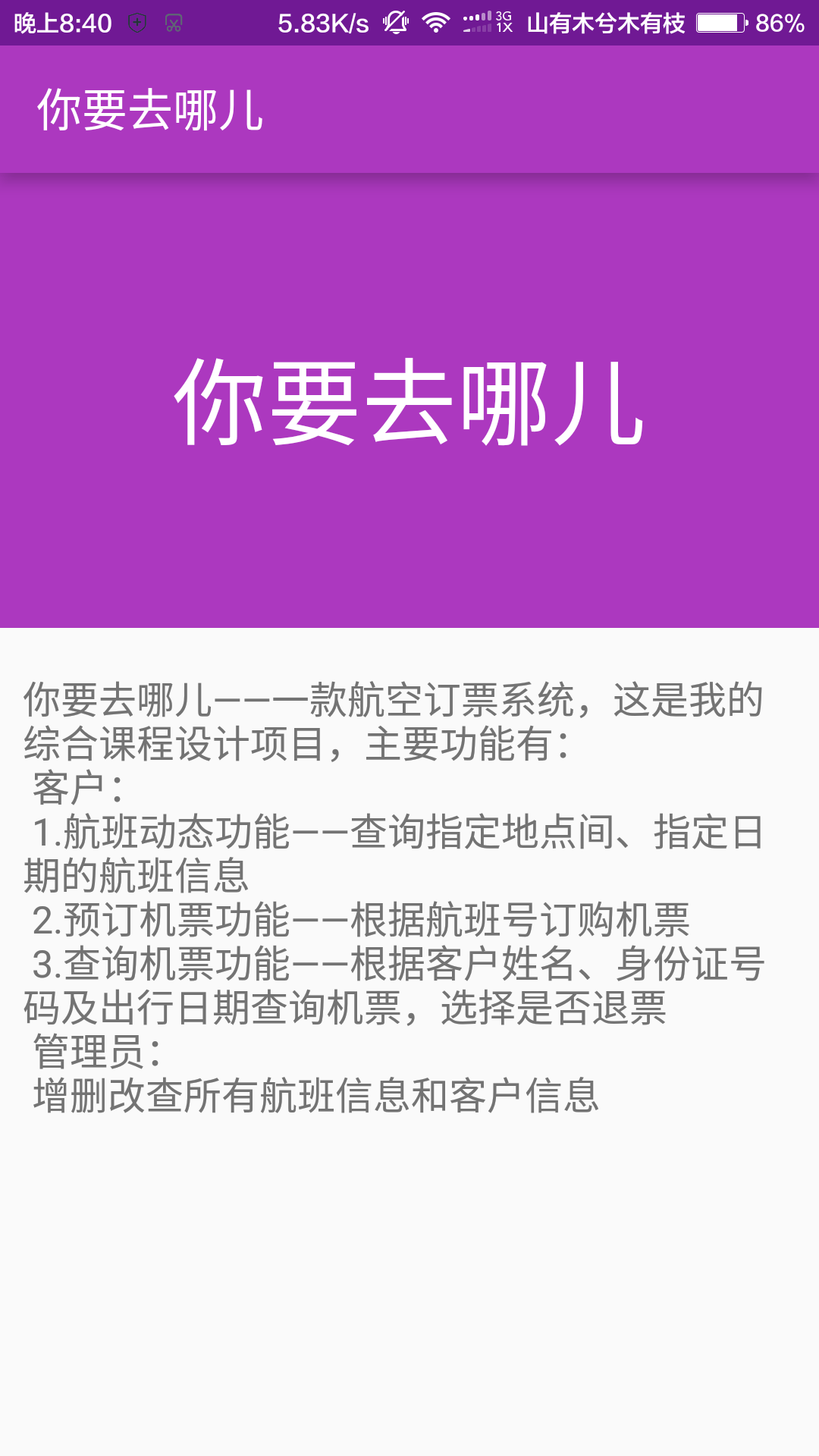
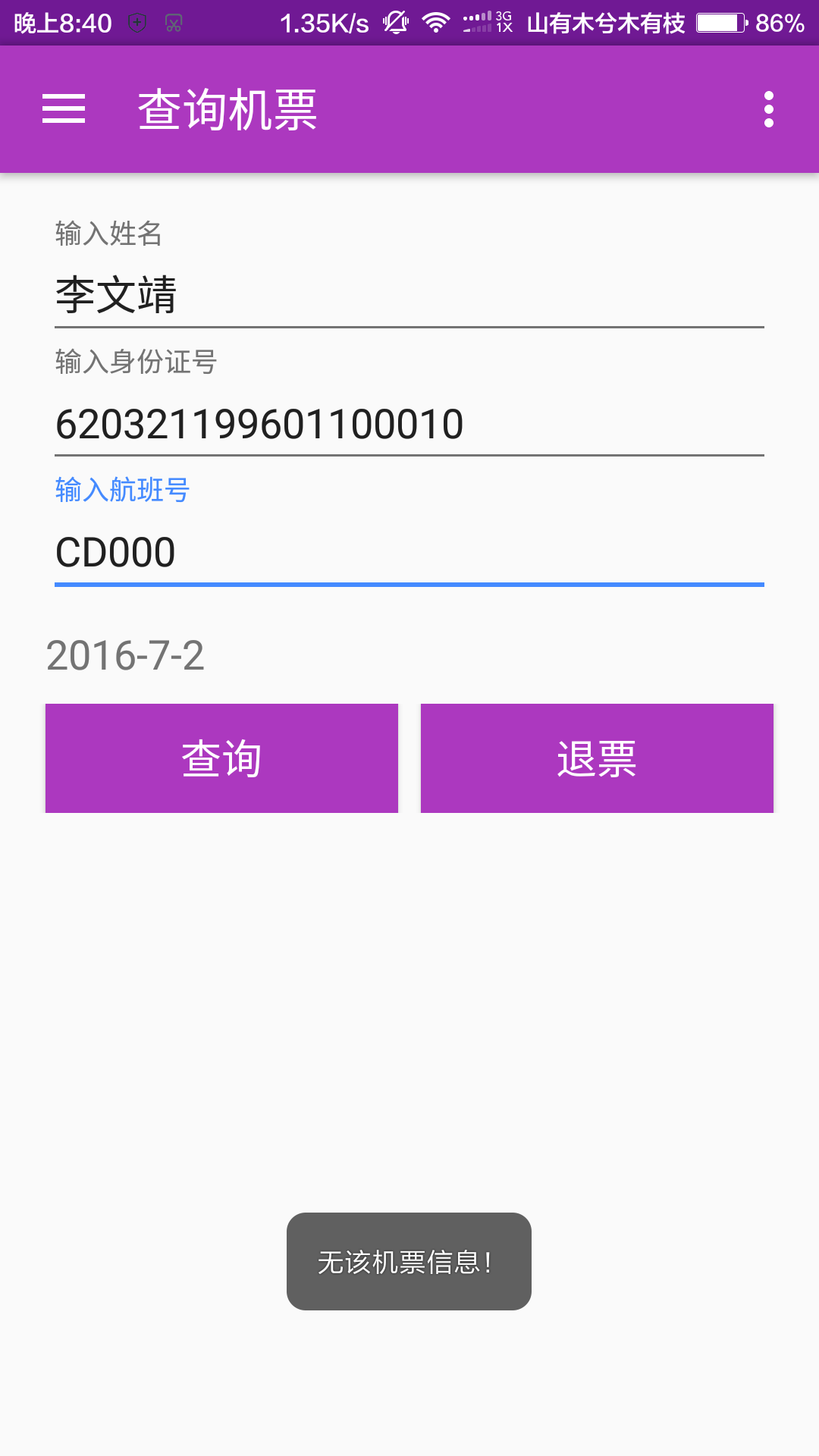
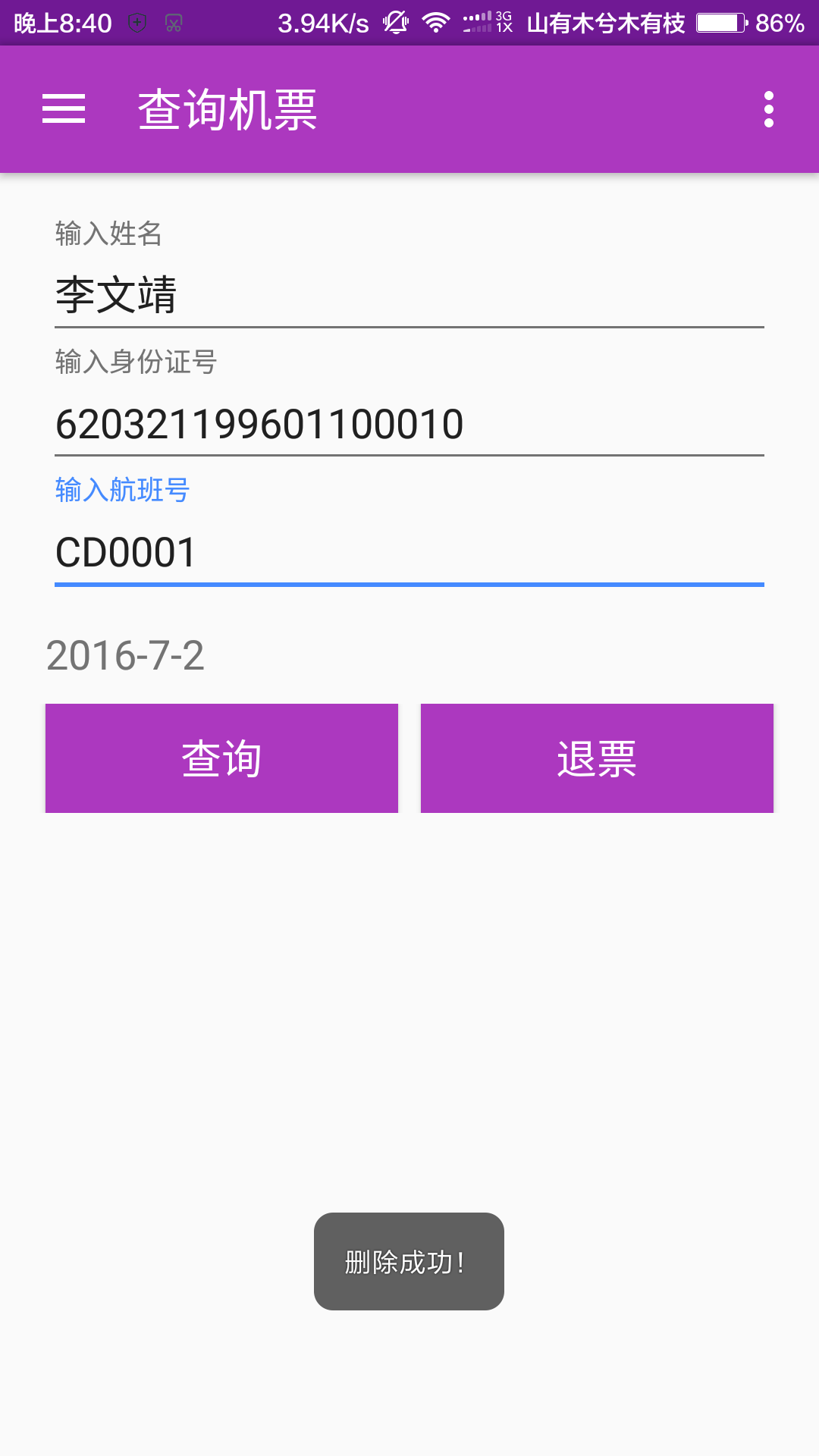
1. **系统测试**

**5.1 测试环境**

小米3C

系统：Android 6.0

**5.2 测试用例和结果**



**5.3 本章小结**

以上就是测试数据及其截图

1. **总结**

**6.1 课程设计总结**

在本次课程设计中，我充分利用了androidstudio+linux终端+gradle+git这一组工具，极大地提升了编程效率。AndroidStudio下进行编程，代码提示、目录结构等非常束缚，让我的精力可以集中于解决眼前的问题。在界面设计中，采用了google官方推荐的Material Design设计风格以及Material Design颜色，控件方面，用了RecyclerView作为列表显示，FloatingActionButton作为图特殊按钮，NavigationDrawerLayout作为策划菜单，同时单个Activity配合多个Fragment一起使用，极大的减少了冗余工作量的同时，使代码结构清晰而完整。后台数据库采用了Bmob云端数据库，免去了自己搭建后台服务器写api的繁琐以及购买服务器的花费，能更高效而便捷的操作数据，同时Bmob后端云在硬件和软件层面都为大家建立了安全机制。增加了数据安全隔离，分布式存储架构，多租户虚拟化技术，密钥匹配，访问权限控制等功能，进一步保障了数据的安全，降低风险。最后关于gradle,Gradle是一个基于Apache Ant和Apache Maven概念的项目自动化建构工具。它使用一种基于Groovy的特定领域语言(DSL)来声明项目设置，抛弃了基于XML的各种繁琐配置。gradle对多工程的构建支持很出色，工程依赖是gradle的第一公民。 gradle支持局部构建。支持多方式依赖管理：包括从[maven](http://baike.baidu.com/view/336103.htm" \t "/home/levent_j/文档\\x/_blank)远程仓库、[nexus](http://baike.baidu.com/view/840006.htm" \t "/home/levent_j/文档\\x/_blank)私服、ivy仓库以及本地文件系统的jars或者dirs 轻松迁移：gradle适用于任何结构的工程（Gradle can adapt to any structure you have.）gradle的整体设计是以作为一种语言为导向的，而非成为一个严格死板的框架。免费开源。

**6.2 课程设计展望**

想要在编程方面有所建树，就必须多加练习，这没有捷径可走。但在不断练习的过程中，也不应当减少学习新知，毕竟学以致用，学习和练习交替上升才是迅速提升自己的最佳手段。通过此次课程设计，使我更加扎实的掌握了有关Android开发方面的知识，在设计过程中虽然遇到了一些问题，但经过一次又一次的思考，一遍又一遍的检查终于找出了原因所在，也暴露出了前期我在这方面的知识欠缺和经验不足。过而能改，善莫大焉。在课程设计过程中，我们不断发现错误，不断改正，不断领悟，不断获取。最终的检测调试环节，本身就是在践行“过而能改，善莫大焉”的知行观。这次课程设计终于顺利完成了，在设计中遇到了很多问题，终于游逆而解。在今后社会的发展和学习实践过程中，一定要不懈努力，不能遇到问题就想到要退缩，一定要不厌其烦的发现问题所在，然后一一进行解决，只有这样，才能成功的做成想做的事，才能在今后的道路上劈荆斩棘，而不是知难而退，那样永远不可能收获成功，收获喜悦，也永远不可能得到社会及他人对你的认可。在这为期两周的编码中，不仅培养了独立思考、动手操作的能力，在各种其它能力上也都有了提高。更重要的是，我学会了很多学习的方法。而这是日后最实用的，真的是受益匪浅。要面对社会的挑战，只有不断的学习、实践，再学习、再实践。这对于我们的将来也有很大的帮助。以后，不管有多苦，我想我们都能变苦为乐，找寻有趣的事情，发现其中珍贵的事情。就像中国提倡的艰苦奋斗一样，我们都可以在实验结束之后变的更加成熟，会面对需要面对的事情。

**参考文献**

[1]