

Research and Implementation of Android Intelligent Terminal Thinking Map

Fanghe Liao¹, Zhansheng Chen^{1,2*}, Cheng Jiang¹

¹School of Applied and Technology, Beijing Union University, Beijing

²School of Computer and Information Technology, Beijing Jiaotong University, Beijing

Email: *ldtchenzs@bnu.edu.cn

Received: Jan. 6th, 2016; accepted: Jan. 24th, 2016; published: Jan. 30th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

To deal with the rise of instant buy, and the demand for tourism, based on Android wide application prospects, using the relevant interface that Map Baidu provides, a simple and practical mobile map service was designed and developed. The personal center was added based on the surrounding search and route search, which was convenient for users to share scenic spots and collect routes. Making full use of the portability of intelligent terminal, it has brought great convenience for people's travel and shopping, having a certain application value.

Keywords

Android, Mobile Map, Baidu Map API, Scenic Spots Sharing

Android智能终端Thinking Map的研究与实现

廖方鹤¹, 陈战胜^{1,2*}, 江城¹

¹北京联合大学应用科技学院, 北京

²北京交通大学计算机与信息技术学院, 北京

Email: *ldtchenzs@bnu.edu.cn

收稿日期: 2016年1月6日; 录用日期: 2016年1月24日; 发布日期: 2016年1月30日

*通讯作者。

摘要

应对即时性团购的兴起以及爱好旅游用户的需求, 基于Android广泛应用的前景, 利用Baidu Map提供的相关接口, 设计并开发出一款简洁实用的手机地图服务, 在保留周边搜索、路线查询的基础上增加了个人中心, 便于用户进行景点分享、路线收藏等, 充分利用移动智能终端的便携性, 为人们的出行和购物带来了极大的便利, 具有一定的应用价值。

关键词

Android, 手机地图, Baidu Map API, 景点分享

1. 引言

随着现代社会网络化、数字化、信息化技术高速发展, 以及 3G 业务的全面展开, 使得智能移动终端实现了从通讯网络终端到互联网终端的转变。对于广大爱好旅游的用户群体而言, 手机地图系统借助移动终端和无线通信网络为其提供了无缝沟通、高效信息沟通、便捷出行的诸多优势。因此, 如何借助已有的地图技术开发智能终端的地图服务系统, 具有重要的意义。其中, 包力泰[1]利用 Google Map 开发工具, 借助智能手机的 3G、GPS 功能, 实现了我的位置、天气预报、手机归属地、本地公交和地图模式四大功能, 给出了设计思路和核心代码, 有重要的借鉴价值。其中, 在用户定位过程中, 综合借鉴了 Network 方式速度快和 GPS 定位精确的优点, 具体思路是调用 LocationManager()服务先获取当前位置和卫星信息, 然后接受子线程发送的经纬度, 从而确定用户位置。在公交线路实现方面将文本格式的公交线路解析成固定模式, 建立数据表, 并考虑了文本中的半角、全角符号等问题, 从而实现了公交线路以及任意两站之间所有路线的查询。天气功能的实现则是调用 GPS 确定用户所处经纬度, 反查城市并利用 Google 网站天气预报的接口, 从而获得天气预报。柳婷[2]则对 Google Maps API、Microsoft Virtual Earth API、yahoo Maps API 和 Baidu Maps API 进行对比分析后, 采用百度地图接口进行手机地图开发, 主要包含地图浏览、地点搜索和自我定位功能, 该文功能点介绍详细, 测试结果表明响应速度较快, 加载时间短。翟玉山[3]基于 Google Maps 实现手机地图应用, 通过目标位置的经纬度实现查询目标坐标, 集合 GPS 行程轨迹计算行程距离并显示在手机屏幕, 通过 GPS 定位技术实现自我定位, 也可以借用手机地图规划出发路径, 满足了用户的基本需求。赵情融[4]利用 Google Map 和相应 API 库设计了一个可以记录用户行走轨迹的个人地图应用程序, 主要跟踪使用者的移动轨迹并显示于地图上。徐光侠[5]等人利用 XML 技术, 整合 Google Maps 与传感器开发原理, 设计开发了周边商店查询、天气预报等功能的生活服务性软件, 能够使用特定字符短信快速获取当前地理信息。吴元君[6]就地图服务系统的实现关键技术、系统部分功能的实现策略进行了介绍, 但细节介绍不够详实。徐丽仙[7]研究了在 Android 平台上利用 Baidu Map 实现地图加载和地址编码的转换, 但功能实现中关键技术介绍不够详细深入。张艳芳[8]基于 Google Maps 实现了城市地图的浏览, 自我定位、位置查询和路径跟踪等功能, 满足了手机用户对地图的需求。

基于市场上 Android 手机市场占有率高达 70%的现状, 以及基于 Baidu Map 地图技术进行智能终端地图服务系统开发介绍不够详实的情况, 开发一款基于 Android 平台的 Thinking Map 系统。该系统针对旅游爱好群体, 在大多数 APP 提供位置服务的基本功能上, 增加了旅游路线、景点相册的分享, 景点路线的收藏、预定酒店等功能, 具有较好的应用价值。

2. Android 体系结构

Android 作为一个移动设备的开发平台,其架构分层清晰、分工明确。软件层次结构主要包括三部分,即操作系统、中间件和应用程序。从整体角度来看,平台由四个部分组成,由顶向下分别为:应用程序、应用框架、运行库和内核。

2.1. 应用程序层

应用程序层位于架构的最上面一层,通常应用在用户界面与交互及前后台的接口中。Java 语言是目前最广泛使用的应用程序开发语言,能够实现个性化的功能设计,保证功能完善、性能优良。Android 本身自带短信通讯程序、邮件、浏览器、计算器、联系人等常用应用软件,能够实现用户常规需求。

2.2. 应用框架层

应用程序框架是 Android 开发的基础,提供应用程序 API,主要包含 Activity Manager、Window Manager、Location Manager 以及 Notification Manager 等九大部分。应用框架层对硬件访问提供一个抽象,开发者在实现自己的应用程序时,可以直接调用这些底层框架,无需重新构建自己的开发平台,降低了系统复杂度和设计难度,减少了设计的实现成本,简化了应用程序的开发流程,提高了组建的重用性和适用性,并能通过彼此提供的接口进行数据的交互,任何其他程序只要正确安装接口,就可以直接调用这些功能。

2.3. 系统运行库层

Android 系统提供的核心库有多媒体库 Media Framework、负责数据存储、交互和显示管理的 Surface Manager、3D 绘图函数库 OpenGL、Web 视图 LibWebCore、小型嵌入式关系型数据库 SQLite 等。Java 语言核心库内的 Dalvik 虚拟机和大部分功能都是由 Android 运行库中的核心库提供的。

2.4. Linux 内核层

Android 系统的底层管理服务主要由 Linux 内核层来提供,主要包含内存、进程管理、安全机制、驱动模块和网络堆栈,是介于底层硬件和上层应用之间的虚拟抽象层,负责上下层之间的交互。各层各尽其职,提供固定的 SAP (Service Access Point),适应软件工程高内聚、低耦合的开发理念。

3. 百度地图简介

百度地图是百度提供的一项网络地图搜索服务,覆盖了国内近 400 个城市、数千个区县。在百度地图里,用户可以查询街道、商场、楼盘的地理位置,也可以找到离您最近的所有餐馆、学校、银行、公园等等。2014 年百度与诺基亚达成协议,未来将提供中国内地以外的地图数据服务。

在电子商务时代,团购成为大家购物新的代名词,通常分为计划性团购和即时性团购。其中,计划性团购是用户实现进行家庭外出、与朋友约会出游等,利用 PC 端使用百度地图进行消费场所的位置、消费水平、口碑效果等进行综合审订后进行团购的一种形式。而即时性团购则是发生在团队加班临时聚餐、偶遇同学聚会唱歌娱乐的突发情况下,临时通过智能终端使用百度地图定位周边团购信息,就位置、评价、价格和环境等因素进行的综合评定后,决定前往消费的一种团购形式。2013 年手机地图市场监测报告显示,手机地图的使用过程中,附近、热点查询功能渗透率不断攀升达到 42.58%。其中餐饮、银行、娱乐、商城超市、酒店信息查询在生活服务信息查询行为的渗透率最高。在电子商务盛行的读图时代,一张高清精美的图片胜过千言万语,但消费者同样关心着寸土寸金手机屏幕上进行在线团购操作带来的流量和资费问题。为此,百度地图推出了离线地图包,从而解决了用户耗费流量、享受畅快速度

的问题。

3.1. 百度地图 API 简介

百度地图 API 是基于 JavaScript 语言编写的应用程序接口,便于开发设计人员访问百度服务和数据,创建功能全面、交互性强的地图应用程序。百度地图 API 类分为核心、服务两大部分,数量高达 586 个,接口多达 1000 多个,提供的数据服务有定位、地点检索、公交方案等。

3.2. 百度地图 API 使用原则

使用百度地图 API 开发 Android 系统应用之前需要做好两个工作,具体如下:

1) 注册百度用户,申请百度地图的开发密钥(Key)

百度地图的类库不属于 Android 平台的基本类库,要使用百度的 API key,需要通过百度用户获取 API 密钥,通过初始化 MapActivity init()方法将密钥写入到用户引用的 API 程序中,才能显示百度地图和位置相关的数据。其中,MapActivity 类即地图类,与底层网络相连接,用于显示百度地图的抽象 Activity 类,在 init()方法中加入申请的 API Key 即可完成开发密钥的初始化,代码实现如 5.3.2 所示。

2) 下载百度地图 API 开发包

要在 Android 系统应用中使用百度地图 API,需要在所建工程中引用 baidumapapi.jar 和 libBMapApiEngine.so 开发包。具体操作是将 baidumapapi.jar 置于工程里 libs 目录下,libBMapApiEngine.so 置于 libs|armeabi 子目录下,然后设置 Add External JARs 即可。

4. 系统总体分析

4.1. 系统需求分析

针对智能终端的性能、网络通信能力与 PC 机、以及 PC 网络通信水平存在一定的差距,本文设计研发的 APP 采用 C/S 模式且智能终端运行 APP 为瘦客户端。

针对旅游爱好群体,本文着眼于“界面简洁、功能满足、特色服务”的路线,在大多数 APP 提供位置服务的基本功能上,增加了旅游路线、景点相册的分享,景点路线的收藏、预定酒店等功能,设计思路如下。

- 1) 用户通过登录,可以查看旅游途中收藏的路线和景点相册。
- 2) 用户打开 GPS,可以实现自我定位,从而搜索周边团购信息。
- 3) 用户在出游之前,通过路线选择进行路程规划。
- 4) 为节省流量和享受畅速服务,用户可在 WIFI 覆盖下实现离线地图下载。
- 5) 为了确保出行顺利,用户可以查看天气预报。
- 6) 为了分享旅途心得,用户可进行景点分享和路线收藏。

4.2. 系统功能分析

系统功能需求从需求分析出发,定义了开发人员必须实现的系统功能,以满足用户的业务需求。根据本文研发系统面向的消费群体,Thinking Map 系统功能如图 1 所示。

如图 1 所示,Thinking Map 系统功能共分为七大模块,分别是用户登录注册,搜索定位,路线选择,周边查看,离线下载、天气预测和个人中心共八大模块。其中,用户登录注册模块中包含用户个人注册、用户登录和用户退出功能;搜索定位模块是在用户打开 GPS 的情况下,能够快速匹配用户输入地点及其周边信息并标识显示在地图上;路线选择是根据用户输入的起点和终点生成多条路线供用户选择;周边查看则根据在用户实现自我定位的前提下,显示附近餐饮、娱乐、加油站等信息以及用户评价信息等;

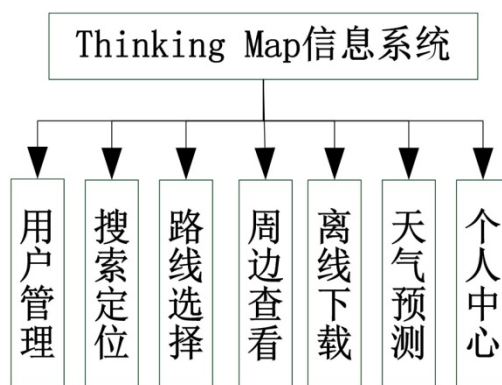


Figure 1. The function block diagram of Thinking Map system

图 1. Thinking Map 系统功能模块图

离线下载提供离线地图，允许用户在 WIFI 覆盖环境下下载，以享受高质量的信息服务，减少流量产生的额外费用；天气预测便于用户安排出行；个人中心为旅游爱好者提供路线收藏，景点共享的服务。

5. 系统详细设计

5.1. 搭建系统开发环境

- 1) 下载安装 JDK。Android 应用开发基于 Java 语言，用户需要下载 Java 开发工具包，即所谓的 JDK，然后安装并设置好 JDK 环境变量。
- 2) 下载安装 Eclipse。Eclipse 是基于 Java 语言的集成开发环境，可扩展性好，应用广泛，深受 Android 开发者的好评，作为 Android 应用的首选开发平台并不为过。
- 3) 更新 Eclipse 下载 ADT。ADT 作为 Android 开发应用插件，可集成到 Eclipse 中。
- 4) 创建 AVD。AVD 全称 Android Virtual Device，是基于 Android OS 的虚拟设备，能够虚拟 Android 平台，便于用户调试运行 Android 应用程序。

在成功搭建了 Android 应用程序的开发环境后，下载百度地图 API 开发包和百度地图的开发密钥，就可以进行 Thinking Map 的开发。此外，在 AVD 上继续调试运行后，本文将会移植 APP 到多款不同的 Android 真机设备进行运行，检测效果。

5.2. 系统架构设计

本系统采用 MVC 开发模式，主要涉及视图 UI 层、控制层、业务逻辑层和数据访问层，总体架构如图 2 所示。

如图 2 所示，UI 层就是视图层(V)，是与用户互动的接口，用于接收用户的输入和输出系统对用户的反馈信息。控制层(C)作为 UI 层与业务逻辑层的接口，使得程序结构清晰，代码简装，事件处理更加高效。模型(M)表示企业数据和业务规则，是整个系统的核心，包含事件处理、查找业务、搜索等，在数据交互中起到承上启下的作用。数据访问层主要是将系统数据存储于 SQLite 中，并调用 MAP API，从而实现数据的存储于读取。

5.3. 关键操作代码实现

本小节主要介绍在开发过程的地图加载、GPS 信号获取、地理位置显示等关键点的设计思路以及地图初始化、路线选择的核心代码。

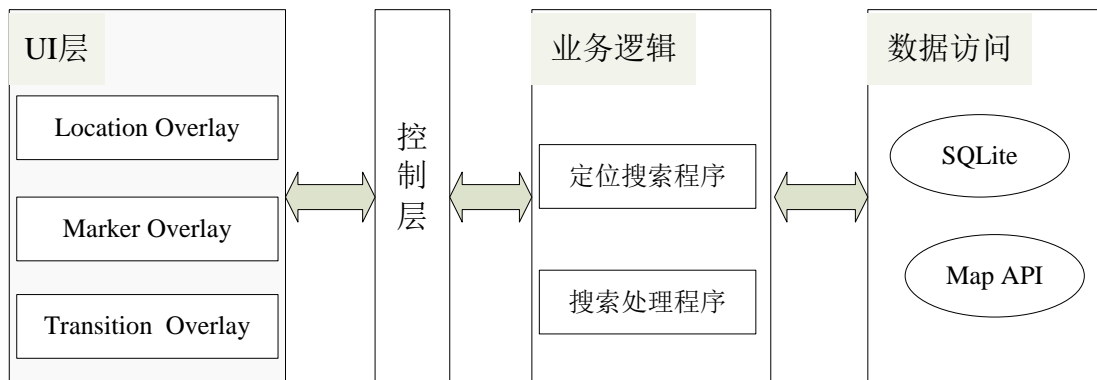


Figure 2. System development framework
图 2. 系统开发架构

5.3.1. 关键点设计思路

1) 地图加载模块。地图加载包含加载、加载成功、退出三个阶段，当用户输入期望查询的地点后点击提交，系统会自动加载地图并以输入地点为中心显示附近的景点、地标等。需要注意的是，除了地图首次加载之外，系统默认加载的地点是用户最后一次查询的地点。

2) GPS 信号获取模块。GPS 卫星导航电文是实现 GPS 定位的必要前提，用户可以通过终端设备接收 GPS 卫星信号，获得用户的位置信息，从而利用 GPS 进行导航和定位。

3) 地理坐标显示模块。该模块涉及搜索定位、周边查看和路线选择，实现思路是在系统显示默认地图后，添加系统权限来支持对 LBS 硬件的访问，从而获得系统服务 Location Manager 对象，利用返回的位置提供者获得位置及其周边信息，从而为旅游爱好者的出行计划提供有价值的参考。其中，需要关注的是在百度地图上进行添加图标、提示框等作业问题，这些均属于地图图层问题。其实，地图 APP 就是有若干图层拼合而成的，包括街道、高楼、街心公园、旅馆、加油站等及其内容简介，以及定位图标、矢量图等覆盖在地图上的元素，统称为地图覆盖物。此外，需要注意的是地图覆盖物同样具有自己的位置坐标，并且不能忘记图层涉及多个缩放等级，这些图层所涉及的地图覆盖物会随着用户的缩放、拖动等操作而进行相应的缩放和移动。

5.3.2. 地图初始化核心代码

```

bMapManager = new BMapManager(MyMapActivity.this);
bMapManager.init(Constant.MAPKEY, new MKGeneralListener() {
public void onGetPermissionState(int arg0) {
    if (arg0 == 300) {
        Toast.makeText(MyMapActivity.this, "验证失败",
            Toast.LENGTH_LONG).show();
    }
}
public void onGetNetworkState(int arg0) {
    Toast.makeText(MyMapActivity.this, "网络加载失败", Toast.LENGTH_LONG).show();
}
});
this.initMapActivity(bMapManager);
bMapManager.start();
.....
  
```

5.3.3. 路线选择核心代码

路线查询模块在位置查询的基础上需要获取不同类型路线的选择与显示，是一个较为综合的功能模块。本位以路线选择模块的实现为例介绍如下。

1) 路线查询模块的功能

本模块的功能是出发点与终点之间所有路线按时间、经济、换乘的路线分类显示在手机终端上，提供给用户多种选择，便于用户参考。

2) 路线查询的业务流程

根据用户对出现路线的业务分析可知，路线查询业务流程图如图 3 所示。

3) 路线查询的用例图

结合路线查询的业务流程，绘制路线查询的用例图如图 4 所示。

4) 路线查询的功能展示

路线查询模块的运行结果如图 5 所示。其中，图 5(a)是系统加载后首先显示用户的当前位置，通过

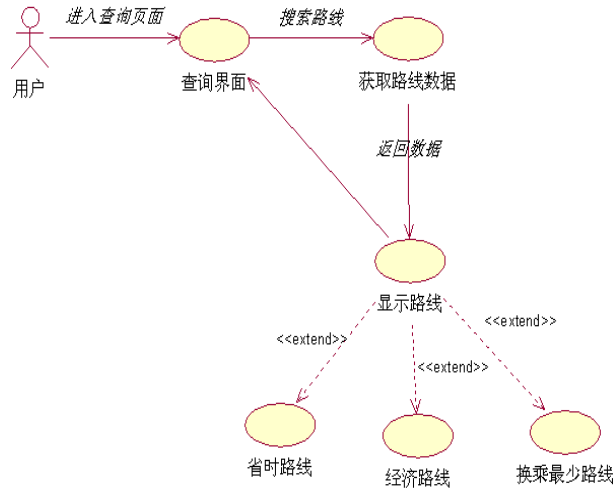


Figure 3. Route query business process diagram
图 3. 路线查询业务流程图

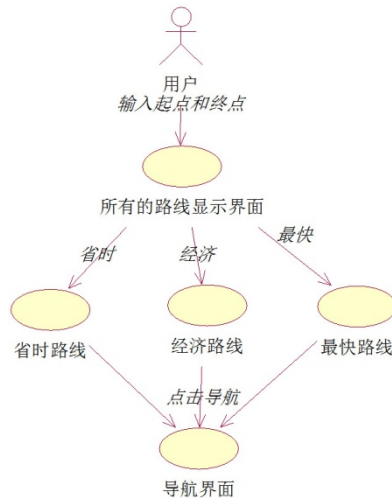


Figure 4. Route query use case diagram
图 4. 路线查询用例图



Figure 5. Route query function display
图 5. 路线查询功能展示

通过 GPS 模块, 获取当前位置经纬度, 将当前位置设置为中心。图 5(b)所示, 为 Thinking Map 的路线选择模块, 对话框中显示了起点与终点的距离和导航路线之间的具体信息, 具体包含上车车站、公交编号、下车车站以及步行的信息。图 5(c)则是地图上路线图的标识曲线高亮显示。

5) 路线查询关键代码

下面分别介绍 Thinking Map 路线查询中自驾车路线查询(Code A)、周边信息获取(Code B)、的实现核心代码。

Code A (驾车线路查询):

```
public void onGetDrivingRouteResult(MKDrivingRouteResult result, int error) {
    if (result == null) { return; }
    drrList = new ArrayList<MKRoute>(); // 构建保存路线的 List
    for (int i = 0; i < result.getNumPlan(); i++) {
        drrList.add(result.getPlan(0).getRoute(i)); // 将信息导入 List
    }
    handler.sendMessage(0x011);
}
```

Code B (周边信息查询):

```
public void onGetPoiResult(MKPoiResult result, int type, int error) {
    if (result == null) {return;}
    // 建立中心点周围遮挡物
    PoiOverlay poiOverlay = new PoiOverlay(MyMapActivity.this, mapView);
    // 将返回的数据装在给中心点遮挡物中
```



```
poiOverlay.setData(result.getAllPoi());
// 地图视图加载遮挡物
mapView.getOverlays().add(poiOverlay);
mapView.invalidate();// 刷新视图
}

public void onGetTransitRouteResult(MKTransitRouteResult result,int error) {
    if(result == null) {return;}
    trpList = new ArrayList<MKTransitRoutePlan>();
    for (int i = 0; i < result.getNumPlan(); i++) {
        trpList.add(result.getPlan(i));
        int n = result.getPlan(i).getNumLines();
        for (int j = 0; j < n; j++) {
            System.out.println(result.getPlan(i).getRoute(j).getDistance()
                + "-" + result.getPlan(i).getLine(j).getTitle()+"\n");
        }
    }
    handler.sendMessage(0x100);
}

public void onGetWalkingRouteResult(MKWalkingRouteResult result,int error) {
    .....
    drrList = new ArrayList<MKRoute>();
    for (int i = 0; i < result.getNumPlan(); i++) {
        drrList.add(result.getPlan(0).getRoute(i))
    }
    .....
}

private Handler handler = new Handler() {
    public void handleMessage(android.os.Message msg) {
        if(msg.what == 0x001) {
            proDialog.dismiss();
            dialog = suggestionInfoDialog("您的起点位置");
            .....
            dialog.show();
        } else if (msg.what == 0x010) {
            proDialog.dismiss();
            dialog = suggestionInfoDialog("您的终点位置");
            .....
        } else if (msg.what == 0x011) {
```

```
.....
    } else if (msg.what == 0x101) {
        onceStartMyLocation();
        mkSearch.init(bMapManager, new MySearchListener());
    }
};};

private Dialog suggestionInfoDialog(String style) {
    LayoutInflater inflater = getLayoutInflater();
    View entryView = inflater.inflate(R.layout.list_info, null);
    AlertDialog.Builder builder = new AlertDialog.Builder(MyMapActivity.this);
    builder.setIcon(getResources().getDrawable(R.drawable.place_icon));
    builder.setTitle(style);
    builder.setView(entryView);
    listInfo = (ListView) entryView.findViewById(R.id.list_info);
    .....
    return builder.create();
}

class ListInfoListener implements OnItemClickListener {
    .....
    mapView.invalidate();
    if (btn_style != 0) {
        .....
    } else if (btn_style == 4) {
        if (line_style.equals("transit")) {
            // 向全局变量类中添加公交线路计划
            Goble.trpplan = trpList.get(position);
            transitOverlay = new TransitOverlay(MyMapActivity.this, mapView);
            .....
            mapController.setZoom(15); // 设置缩放级别为 16
            mapController.setCenter(geoPoint); // 将本地设置为地图中心点
            mapView.invalidate();
        }
    }
}
handler.sendMessage(0x002);
}

public View getView(int position, View convertView, ViewGroup parent) {
    .....
}
```

```

// 获取内容资源布局
LinearLayout linearLayout = (LinearLayout) mInflater.inflate(R.layout.content, null);
contentName.setTextSize(20);
linearLayout.addView(contentName);
linearLayout.addView(contentContent);
.....
String lineContent = "起点坐标:经度<" + route.getStart().getLatitudeE6() * 1.0 / 1E6
+ "> 纬度<" + route.getStart().getLongitudeE6() * 1.0 / 1E6 + ">\n";
lineContent = lineContent + "终点坐标:经度<" + route.getEnd().getLatitudeE6() * 1.0 / 1E6
+ "> 纬度<" + route.getEnd().getLongitudeE6() * 1.0 / 1E6 + ">\n";
contentName.setText("路线总距离: " + distance + "米\n");
contentContent.setText(lineContent);
} } else if (btn_style == 4) {
.....
return linearLayout;
}}

```

5.3.4. SQLite 数据库的应用

本程序中 SQLite 数据库主要用于保存用户的注册信息，下面分别介绍用户注册(Code C)、用户登录(Code D)的实现核心代码。

Code C (用户注册):

```

public boolean register(User user){
    SQLiteDatabase sdb=dbHelper1.getReadableDatabase();
    String sql="insert into user(username,password,age,sex) values(?,?,?,?)";
    Object obj[]={user.getUserName(),user.getPassword(),user.getAge(),user.getSex()};
    sdb.execSQL(sql, obj);
    return true;
}

```

Code D (用户登录):

```

public boolean login(String username,String password){
    SQLiteDatabase sdb=dbHelper1.getReadableDatabase();
    String sql="select * from user where username=? and password=?";
    Cursor cursor=sdb.rawQuery(sql, new String[]{username,password});
    if(cursor.moveToFirst()==true){
        cursor.close();
        return true;
    }
    return false;
}

```

5.3.5. 百度地图 API 使用指南

开发者可前往 <http://developer.baidu.com/map/index.php?title=androidsdk> 参考百度地图官方 API 文档，在此简要介绍使用步骤。

Step 1: 创建并配置工程。

Step 2: 在 AndroidManifest 中添加开发密钥、所需权限等信息。

Step 3: 在布局 xml 文件中添加地图控件。

Step 4: 在应用程序创建时初始化 SDK 引用的 Context 全局变量。在 SDK 各功能组件使用之前都需要调用 SDKInitializer.initialize(getApplicationContext())方法，本文建议该方法放在 Application 的初始化方法中。

Step 5: 创建地图 Activity，管理地图生命周期。

6. 系统测试与评价

本文采用 AVD 在电脑编程实现后，移植到真实手机(MI4)和华为荣耀 6 plus 分别进行测试，界面简洁，使用便捷且基本可以满足热爱旅游彭侯的需求。图 6(a)~图 6(f)展示了真机运行的效果，图 7 显示了移植真机后该软件的基本信息，占用系统资源总计 4.56 M。

如图 6 所示，用户可以方便的输入目的地，从而获得多条出行路径。当用户注册为合法用户后，登陆进入个人中心，即可享受该软件提供的餐饮、购物等便捷服务。

7. 结束语

本文基于 Android 智能终端，将定位技术和 Baidu Map API 运用于手机地图 Thinking Map 系统，旨在服务旅游爱好群体，功能相对完备，能够给与外出旅游者提供极大的便利。

系统测试结果表明，在 Wifi 信号覆盖且较佳的情况，地图在线反映较为灵敏，反应时间在 5 s 以内，



(a)

(b)

(c)

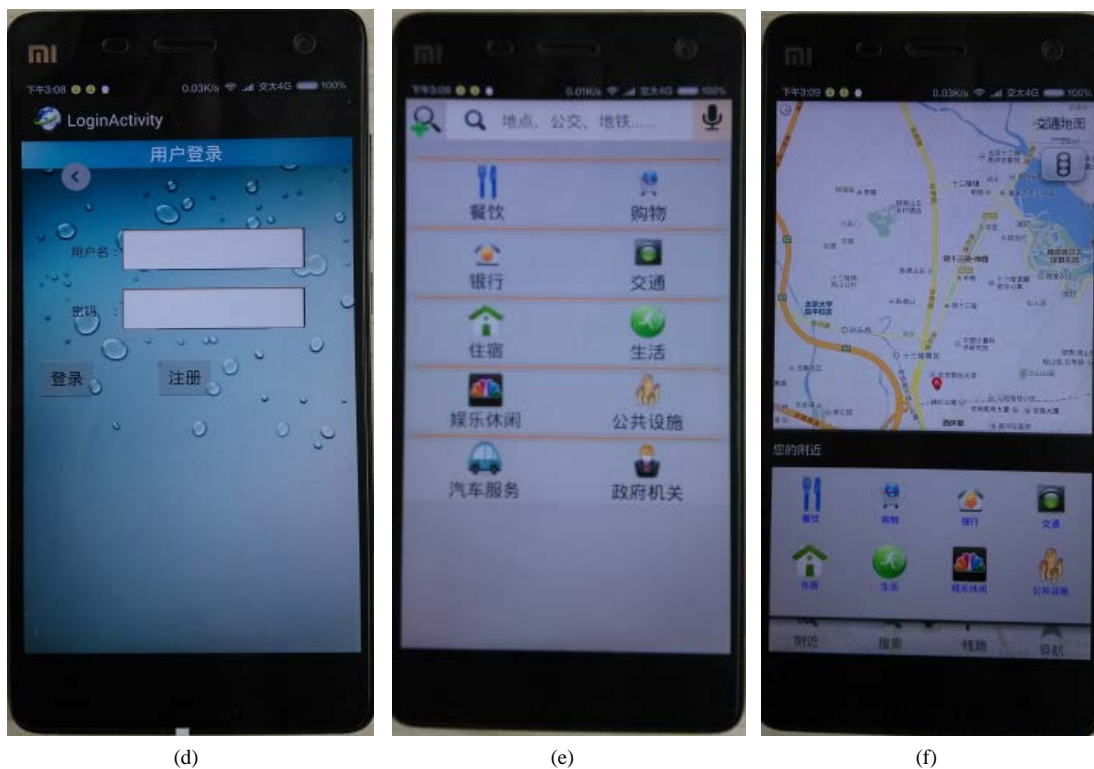


Figure 6. Thinking Map portable test effect
图 6. Thinking Map 移植测试效果



Figure 7. Software basic information
图 7. 软件基本信息

符合用户心理承受能力。此外，若采用离线地图方式，速度更佳。综上所述，本文开发的 Thinking Map 系统具有如下特点：

- 1) 全文就 Baidu Map API 进行手机地图开发，从设计到实现介绍较为详实。
- 2) 该系统分为模拟器和不同型号的真机模拟进行测试，运行速度符合用户心理预期。
- 3) 在参考文献中公有的路线查询、周边搜索等公共模块上，添加了个人中心模块，便于爱好出游的用户发表旅游日志、景点共享、路线收藏等，具有小众的应用前景。

本应用目前存在不足主要是软件针对旅游搜集的地区素材不够充分，界面设计不够大方美观，将在后续工作中继续完善并加以推广使用。

基金项目

北京联合大学“启明星”大学生科技创新项目(201511417SJ029 和 201511417SJ045)，北京联合大学新起点计划项目资助(zk10201303)，北京市职业院校教师素质提高工程资助项目(京教财(2012)21 号)。

参考文献 (References)

- [1] 包力泰. 基于 Android 系统利用 Google Map 开发信息查询软件[D]: [硕士学位论文]. 呼和浩特: 内蒙古大学, 2012.
- [2] 柳婷. 基于 Android 手机地图服务系统的设计与实现[D]: [硕士学位论文]. 北京: 北京邮电大学, 2012.
- [3] 翟玉山. 基于 Android 平台手机地图的设计与实现[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2011.
- [4] 赵情融. 基于 Android 平台的个人移动地图应用[D]: [硕士学位论文]. 武汉: 华中科技大学, 2011.
- [5] 徐光侠, 封雷, 涂演, 等. 基于 Android 和 Google Maps 的生活辅助系统的设计与实现[J]. 重庆邮电大学学报(自然科学版), 2012, 24(2): 242-247.
- [6] 吴元君. 基于 Android 的 Google Map 服务系统开发[J]. 合肥学院学报(自然科学版), 2014, 24(2): 37-40(49).
- [7] 徐丽仙. Android 系统中基于 Baidu Map 的应用技术研究[J]. 无线互联科技, 2014(9): 86-87(131).
- [8] 张艳芳. Android 手机 Web 地图服务的应用设计[D]: [硕士学位论文]. 南昌: 南昌大学, 2010.