

基于 Android 的手机通讯录 管理系统的研究与实现

笄林梅

(江苏财经职业技术学院 计算机系, 江苏 淮安 223003)

摘要:针对传统手机通讯录管理的数据少、浏览和查询不便等缺陷,研究开发了基于 Android 的手机通讯录管理系统. 该系统使用 Java 语言,在 Eclipse 集成开发平台上实现,数据库设计采用 SQLite. 本系统能够快速、便捷地实现通讯录的显示、添加、删除、修改和查询等操作,具有良好的应用前景.

关键词:Android;手机通讯录;Java;Eclipse

中图分类号:TP315.69 **文献标志码:**A **DOI:**10.3969/j.issn.2095-476X.2013.03.014

Research and implementation of mobile phone address book management system based on Android

DA Lin-mei

(Department of Computer, Jiangsu Vocational and Technical College of Finance & Economics, Huaian 223003, China)

Abstract: Aiming at the limited data and inconvenience of browse and query of the traditional mobile phone address book management, the mobile phone address book management system based on Android was developed. This system used Java language, and was realized in the Eclipse integrated development platform. The database design of the system used SQLite. The system can quickly and conveniently realize address book display, add, delete, modify and query operations. It has a good application prospect.

Key words: Android; mobile phone address book; Java; Eclipse

0 引言

随着智能手机的逐渐普及,人们对手机应用的需求更加广泛. Android 的兴起,给智能手机业务开发带来了新鲜的血液. Android 的优势在于:源代码开放,便于开发人员更清楚地把握业务的实现细节;基于 Java 语言,使得 Android 应用开发更加容易;底层基于 Linux,容易移植到各种嵌入式平台中^[1].

通讯录管理是手机应用的重要部分,通过它可

以得到准确快捷的通讯录信息. 传统手机中通讯录管理存在着数据少、浏览和查询不便等缺点. 目前,基于 Android 操作系统的手机得到广泛应用,虽然 Android 自带了功能强大的、具有多种管理功能的通讯录管理系统,但是,手机用户对通讯录管理功能的需求越来越高,尽可能快速地添加、显示、修改、删除、查询通讯录数据,使通讯录管理更加方便、灵活是手机用户的迫切要求. 据调查,目前有较多的开发者已开发出基于 Android 的通讯录管理系统. 本文的研究目标在于:开发功能全面、用户体验丰

富、使用方便的通讯录管理系统,进一步满足部分 Android 手机用户的需求。

1 系统设计

1.1 Android 的系统架构

Android 系统采用分层架构,从上到下依次为应用层、应用框架层、系统运行库层和 Linux 内核层。其中,应用层是 Android 自带的核心应用程序,如联系人管理、电子邮件、短信、日历、浏览器、地图等,这些程序都是基于 Java 语言的。应用框架层提供了包括视图系统、内容提供器,以及窗口、活动、通知、位置、资源、电话和包管理器等部分组件,是 Android 应用程序开发的基础^[2]。

1.2 系统技术

本系统是基于 Android 技术,在 Windows 7 环境下,使用 Java 语言,在 Eclipse 集成开发平台上实现的。数据库设计采用 Android 系统集成的嵌入式关系型数据库 SQLite。

Android 的开发工具包(Android SDK)作为 Android 的软件包以及软件框架、硬件平台、操作系统等建立应用软件的开发工具的集合,能够完成手机通讯录系统人机交互的设计和良好界面的开发。

Eclipse 是一个开放源代码的 Java 可扩展开发平台。在本系统开发过程中,通过插件组件构建开发环境。ADT 是 Android 在 Eclipse 上的开发工具,安装 ADT 可以为 Android 系统开发提供开发工具的升级或变更。

SQLite 的主要设计目标是嵌入式软件,能够支持多种主流操作系统,处理速度快。在本系统开发中,使用 SQLite 来存储结构化数据。

为了简化界面设计,本系统采用目前比较流行的界面设计方法,即将界面描述部分的程序代码抽取到程序外部的 XML 描述文件中。在系统设计中,需要定义包含 Activity 在内的组件及相应的函数并使用 Intent 实现屏幕间的移动。Activity 是 Android 的呈现层,显示可视化的用户界面,接收与用户交互所产生的界面事件。系统包括多个 Activity,Intent 可以实现不同 Activity 之间的切换和数据传递。Intent 即意图,它是动作的完整描述,包含了 Activity 与其他组件之间交互的数据信息^[3]。

1.3 系统功能

据手机功能的调查数据显示,九成以上消费者

都使用手机通讯录工具。随着手机通讯录功能的不断加强与完善,手机通讯录对于人们的意义,已不仅仅是单纯地显示电话号码,而是向着个性化、人性化的方向发展。今后的发展趋势就是从通讯录发展为名片夹,即管理包含联系人的手机号码、QQ、地址、电子邮件、备注等多项内容。

Android 手机通讯管理系统的主要功能包括:添加、显示、修改、删除、查询、呼叫、发送短信、发邮件、备份数据、还原数据等。

1.4 系统业务流程

Android 手机通讯管理系统是一个集 Web、数据库、移动终端为一体的应用程序。它的业务流程是:用户进入手机通讯管理系统后,会看到联系人列表主菜单,其中包括添加联系人、搜索联系人、退出菜单项。

该系统的主要用户是 Android 手机用户。对于用户而言,通讯录上按钮是用户向系统输入操作请求的输入设备,GUI 是用于向用户输出信息的输出设备。系统的业务流程如图 1 所示。

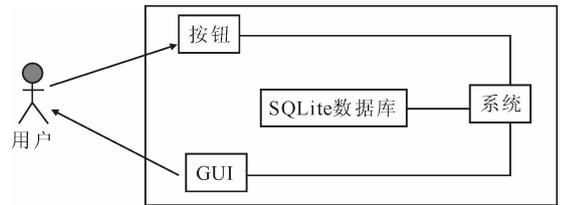


图 1 系统业务流程图

2 Android 通讯录管理的实现

本文以联系人浏览功能、添加联系人和查找联系人为例介绍系统功能的实现。

2.1 联系人浏览功能

在通讯录系统登录界面输入正确的用户名和密码后,进入系统并显示联系人列表界面,该列表由 ListView 控件生成。打开数据库(如果数据库不存在则创建数据库,并创建数据表),查找数据库中的所有联系人,把联系人姓名和移动电话号码这 2 项填充到 ListView 的 adapter 中。每 1 行显示 1 个联系人的姓名和手机号码,联系人的显示顺序是根据插入数据库的顺序显示的。点击某个联系人会进入查看联系人界面,可以查看联系人的详细信息,对联系人进行编辑、删除、拨打电话和发送短信等操作。

点击 MENU 按钮时,就会显示主菜单项,其中

包括添加、查找、删除、菜单和退出。

点击添加菜单,会进入添加联系人界面,可以输入联系人相关信息,完成联系人添加功能;点击查找菜单,会进入联系人查找界面,可以进行联系人查找,搜索想要找的联系;点击菜单,会出现显示所有、删除所有、备份数据、删除数据、更新、返回等详细信息;点击退出菜单,则会退出该系统。系统主菜单如图 2 所示。



图 2 系统主菜单

关键代码如下:

```
public ArrayList getAllUser(boolean privacy) {
    ArrayList list = new ArrayList();
    Cursor cursor = null;
    if(privacy) {
        cursor = dbInstance.query(DB_TABLENAME,
            new
            String[] { "_id", "name", "mobilephone", "office-
            phone", "familyphone", "address", "othercontact", "
            email", "position", "company", "zipcode", "remark", "
            imageid" },
            "privacy = 1", null, null, null, null);
    } else {
        cursor = dbInstance.query(DB_TABLENAME,
            new
            String[] { "_id", "name", "mobilephone", "office-
            phone", "familyphone", "address", "othercontact", "
            email", "position", "company", "zipcode", "remark", "
            imageid" },
            "privacy = 0", null, null, null, null);
    }
}
```

2.2 添加联系人模块

添加联系人界面使用 EditText 控件来添加联

系人所有信息,处于可编辑状态,手机号和座机号的 EditText 设定为只能输入数字。添加完信息后点击确定按钮,触发确定按钮,点击监听事件,从而对数据库中该联系人的信息进行添加并添加到数据库中,然后自动返回联系人浏览界面。点击取消按钮会返回联系人浏览界面。添加联系人界面如图 3 所示。



图 3 添加联系人界面

关键代码如下:

```
public long insert(User user) {
    ContentValues values = new ContentValues();
    values.put("name", user.username);
    values.put("mobilephone", user.mobilePhone);
    values.put("officephone", user.officePhone);
    values.put("familyphone", user.familyPhone);
    values.put("address", user.address);
    values.put("othercontact", user.otherContact);
    values.put("email", user.email);
    values.put("position", user.position);
    values.put("company", user.company);
    values.put("zipcode", user.zipCode);
    values.put("remark", user.remark);
    values.put("imageid", user.imageId);
    values.put("privacy", user.privacy);
    return dbInstance.insert(DB_TABLENAME, null,
    values);
}
```

2.3 查找联系人模块

采用 SQL 模糊查询,可以只输入联系人姓名或号码中的一部分,在 SQL 语句中联系人姓名的前后添加“%”,即可查找到所有包含该部分的联系人,并在 ListView 中显示所有联系人的姓名和手机号

码. 按姓名查找联系人界面如图 4 所示.



图 4 按姓名查找联系人界面

关键代码如下:

```
public ArrayList getUsers( String condition, boolean privacy) {
    ArrayList list = new ArrayList();
    String strSelection = "";
    if(privacy) {
        strSelection = " and privacy = 1";
    } else {
        strSelection = " and privacy = 0";
    }
    String sql = "select * from " + DB_TABLENAME
    + " where 1 = 1 and ( name like '%" + condition +
    "%' +
        " or mobilephone like '%" + condition + "%'
    or familyphone like '%" + condition + "%' +
        " or officephone like '%" + condition +
    "%' )" + strSelection;
    Cursor cursor = dbInstance.rawQuery(sql, null);
    while( cursor.moveToNext() ) {
        HashMap item = new HashMap();
        item.put( "_id", cursor.getInt( cursor.getColumnIndex( "_id" ) ) );
        item.put( " name", cursor.getString( cursor.getColumnIndex( " name" ) ) );
        item.put( " mobilephone", cursor.getString( cursor.getColumnIndex( " mobilephone" ) ) );
        item.put( " officephone", cursor.getString( cursor.getColumnIndex( " officephone" ) ) );
        item.put( " familyphone", cursor.getString( cursor.getColumnIndex( " familyphone" ) ) );
        item.put( " address", cursor.getString( cursor.getColumnIndex( " address" ) ) );
```

```
item.put( " othercontact", cursor.getString( cursor.getColumnIndex( " othercontact" ) ) );
        item.put( " email", cursor.getString( cursor.getColumnIndex( " email" ) ) );
        item.put( " position", cursor.getString( cursor.getColumnIndex( " position" ) ) );
        item.put( " company", cursor.getString( cursor.getColumnIndex( " company" ) ) );
        item.put( " zipcode", cursor.getString( cursor.getColumnIndex( " zipcode" ) ) );
        item.put( " remark", cursor.getString( cursor.getColumnIndex( " remark" ) ) );
        item.put( " imageid", cursor.getInt( cursor.getColumnIndex( " imageid" ) ) );
        list.add( item);
    }
    return list;
}
```

3 结语

Android 平台因其优异的性能和开放性,在智能手机上日益得到广泛应用. 本文针对现有手机通讯录系统数据少、浏览和查询不便等缺陷,设计了基于 Android 的通讯录管理系统. 系统开发采用 Java 语言,在 Eclipse 平台上实现,数据库设计采用 SQLite. 该系统交互性好、操作简便、易于扩展,具有良好的手机应用前景.

参考文献:

- [1] 韩超,梁泉. Android 系统原理及开发要点详解[M]. 北京:电子工业出版社,2010.
- [2] 方传蔚,谢维波. 基于 Android 的短信加密研究与实现[J]. 通信技术,2012,45(6):14.
- [3] 弋改珍,解争龙,张琨. 基于 Android 的火车时刻表查询系统设计与实现[J]. 现代电子技术,2012, 35(4):41.
- [4] 舒后,王国伟. 基于 Android 系统的手机交通查询软件的设计与实现[J]. 北京印刷学院学报,2011, 19(4):68.
- [5] 夏帮贵. 基于 Android 平台的移动学习系统[J]. 西华大学学报:自然科学版,2011,30(5):81.
- [6] 刘卫国,姚显禹. Android 与 J2EE 平台间即时通信的研究与实现[J]. 计算机系统应用,2008(12):118.