北京工业大学

2016-2017学年 第1学期

软件学院

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称： | Android移动应用程序开发 | | |
| 报告性质： | □作业报告 □实验报告 | | |
| 学号： | 14081106 | 姓名： | 张昊赟睿 |
| 任课教师： | 王志强 | 课程性质： |  |
| 学分： |  | 学时： |  |
| 班级： | 140811 | 成绩： |  |
| 小组成员： |  | | |
| 教师评语： |  | | |

2017年 01 月 01 日

# 一、实验目的

运用这学期所学的安卓知识编写一个小程序。从需求出发，到功能设计，UI设计，编写程序，测试调整，完成一款移动应用。完整的走一遍编写程序的流程，巩固所学知识，提高自己编写程序的能力。

# 二、实验环境

Windows8

Android studio

Android API 23

# 三、需求分析

随着移动设备技术的蓬勃发展，手机已经是人们生活不可或缺的一部分，随时随地的都要携带手机。所以手机相对于纸笔来说更方便，随时有想记录的想法时就可以记录下来。而记录自己的想法又是非常重要的一件事情，所以想写一个这样的app。

# 四、功能介绍

每一条笔记有标题和内容，自动记录时间日期，可以增加、删除、修改笔记。

**主页面**

列表显示每条笔记的标题和日期，列表顺序按照创建时间排列，点击进入编辑页面可以修改，长按会弹出是否删除的提示框。

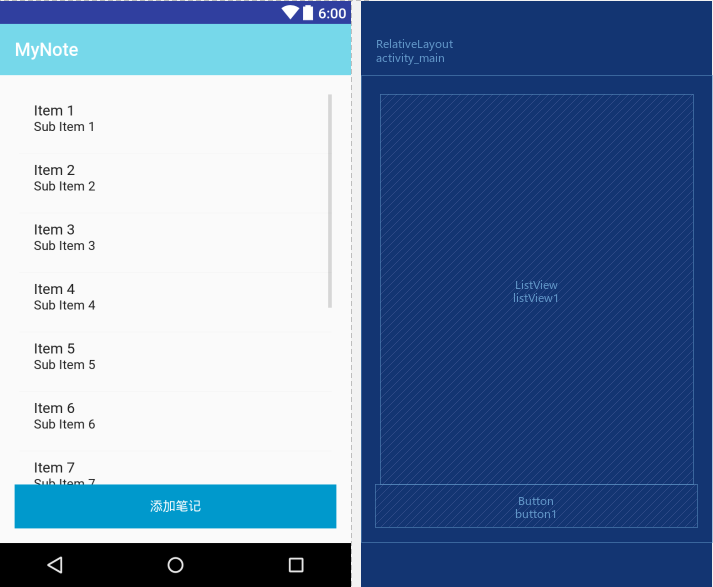
点击下方添加笔记按钮添加笔记，跳转到编辑页面。

**编辑页面**

通过界面的跳转可以跳转到编辑页面，在这个页面可以编辑笔记，点击保存按钮保存。点击手机右下角的返回键，如果是空则不保存，否则自动保存（防止误点返回键导致所写内容遗失）。

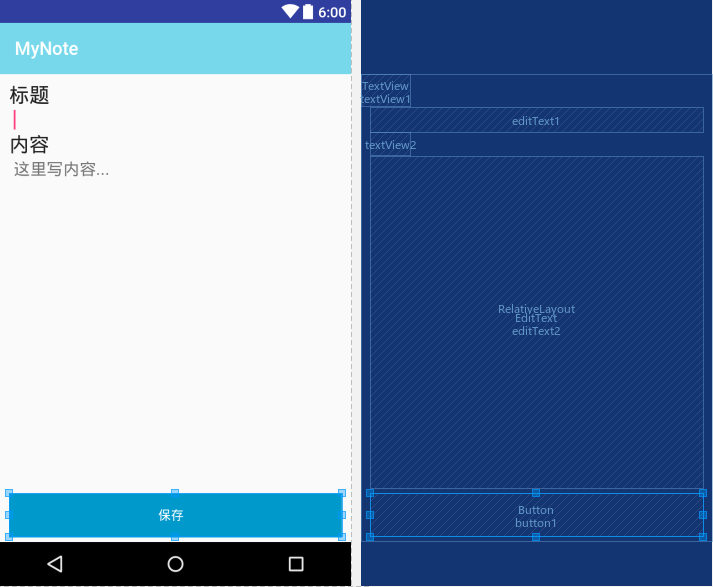
# 五、界面布局

**主界面**



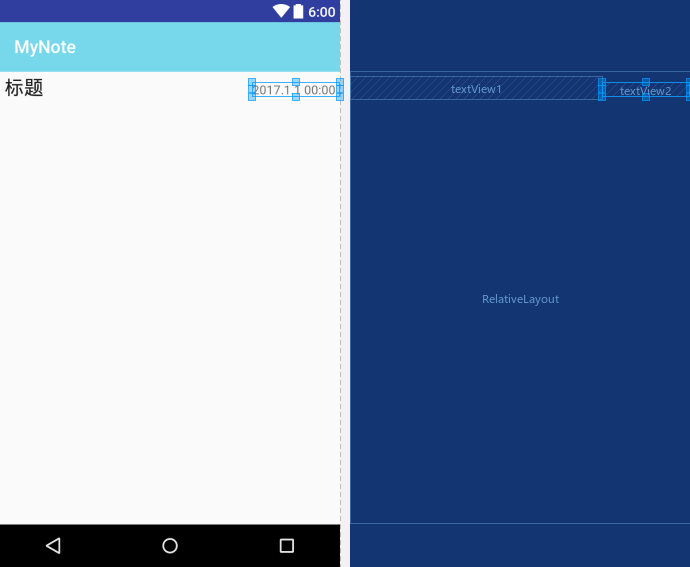
包含一个button点击添加笔记和一个listview显示已有的笔记列表

**编辑界面**



包含两个textview用来提示下面输入标题或内容；两个edittext，一个用来写入标题，另一个用来编辑内容；还有一个button，点击保存。

**ListView中转站**



包含两个textView，显示标题和时间日期，用于主页面ListView的。

# 六、类说明

**MainActivity.java**主界面

这个类主要包括3个点击事件，分别为

1，ListView的长按点击事件，用来AlertDialog来判断是否删除数据。

2，ListView的点击事件，跳转到第二个界面，用来修改数据

3，新建便签按钮的点击事件，跳转到第二界面，用来新建便签

**SecondAtivity.java**编辑界面

用来编辑日记

主要包括一个方法，isSave()用来保存数据;

包含一个按钮的点击事件：保存按钮的点击事件，调用isSave()方法实现保存数据。

**SaveData.java**用来暂时存储数据

**private** String **title**;*//标题***private** String **content**;*//内容***private** String **times**;*//时间***private int ids**;*//编号*

包含五个构造函数，用来满足不同地方调用时的需求。因为在每个地方所需数据不一样，比如主页面显示时不需要content，所以构造时不传content。

每个属性有一个get函数，获取属性。

**MyAdapter.java**继承BaseAdapter用来适配主界面里的ListView

重载了getCount() getItem() getItemId() getView()

**MyDatabaseHelper.java**继承了SQLiteOpenHelper用来建立数据库

数据库存储：建了一个BOOK表，表中有ids(主键)，title(标题)，content(内容)，times(时间日期)。

**MyDataBase.java**用来进行对数据库的处理操作

把所有数据库操作都放到了这里，实现数据的增、删、改、查。

**public** ArrayList<SaveData> getArray() {}//得到列表界面的listview的array数据

**public** SaveData getTiandCon(**int** id)//返回可能要修改的数据

**public void** toUpdate(SaveData cun)//修改数据

**public void** toInsert(SaveData cun) //添加数据

**public void** toDelete(**int** ids)//删除数据

# 七、核心代码

## MyDataBase

//别的类实例化这个类的同时，创建数据库

**public** MyDataBase(Context con) {  
 **this**.**context** = con;  
 **myDatabaseHelper** = **new** MyDatabaseHelper(**context**);  
}

//得到填充ListView用的array数据，从数据库里查找后解析。第一个界面调用

**public** ArrayList<SaveData> getArray() {  
 ArrayList<SaveData> array = **new** ArrayList<SaveData>();  
 ArrayList<SaveData> array1 = **new** ArrayList<SaveData>();  
 **myDatabase** = **myDatabaseHelper**.getWritableDatabase();  
 Cursor cursor = **myDatabase**.rawQuery(**"select ids,title,times from mybook"**, **null**);  
 cursor.moveToFirst();  
 **while** (!cursor.isAfterLast()) {  
 **int** id = cursor.getInt(cursor.getColumnIndex(**"ids"**));  
 String title = cursor.getString(cursor.getColumnIndex(**"title"**));  
 String times = cursor.getString(cursor.getColumnIndex(**"times"**));  
 SaveData cun = **new** SaveData(id, title, times);  
 array.add(cun);  
 cursor.moveToNext();  
 }  
 **myDatabase**.close();  
 **for** (**int** i = array.size(); i > 0; i--) {  
 array1.add(array.get(i - 1));  
 }  
 **return** array1;  
}

*//返回可能要修改的数据***public** SaveData getTiandCon(**int** id) {  
 **myDatabase** = **myDatabaseHelper**.getWritableDatabase();  
 Cursor cursor = **myDatabase**.rawQuery(**"select title,content from mybook where ids='"** + id + **"'"**, **null**);  
 cursor.moveToFirst();  
 String title = cursor.getString(cursor.getColumnIndex(**"title"**));  
 String content = cursor.getString(cursor.getColumnIndex(**"content"**));  
 SaveData cun = **new** SaveData(title, content);  
 **myDatabase**.close();  
 **return** cun;  
}

//第二个界面调用

*//修改*

**public void** toUpdate(SaveData cun) {  
 **myDatabase** = **myDatabaseHelper**.getWritableDatabase();  
 **myDatabase**.execSQL(**"update mybook set title='"** + cun.getTitle() + **"',times='"** + cun.getTimes() + **"',content='"** + cun.getContent() + **"' where ids='"** + cun.getIds() + **"'"**);  
 **myDatabase**.close();  
}

*//添加***public void** toInsert(SaveData cun) {  
 **myDatabase** = **myDatabaseHelper**.getWritableDatabase();  
 **myDatabase**.execSQL(**"insert into mybook(title,content,times)values('"** + cun.getTitle() + **"','"** + cun.getContent() + **"','"** + cun.getTimes() + **"')"**);  
 **myDatabase**.close();  
}  
  
*//删除***public void** toDelete(**int** ids){  
 **myDatabase**=**myDatabaseHelper**.getWritableDatabase();  
 **myDatabase**.execSQL(**"delete from mybook where ids="**+ids+**""**);  
 **myDatabase**.close();  
}

## MyAdapter

//用来填充ListView,把这个类从主页面中分离，只要传过来两个值，inflater ,和数据ArrayList<>;

**public** MyAdapter(LayoutInflater inf, ArrayList<SaveData> array) {  
 **this**.**inflater** = inf;  
 **this**.**arrayList** = array;  
}

**public** View getView(**int** position, View convertView, ViewGroup parent) {  
 ViewHolder vh;  
 **if** (convertView == **null**) {  
 vh = **new** ViewHolder();  
 convertView = **inflater**.inflate(R.layout.***adapter\_listview***, **null**);  
 vh.**tv1** = (TextView) convertView.findViewById(R.id.***textView1***);  
 vh.**tv2** = (TextView) convertView.findViewById(R.id.***textView2***);  
 convertView.setTag(vh);  
 }  
 vh = (ViewHolder) convertView.getTag();  
 vh.**tv1**.setText(**arrayList**.get(position).getTitle());  
 vh.**tv2**.setText(**arrayList**.get(position).getTimes());  
 **return** convertView;  
}  
  
**class** ViewHolder {  
 TextView **tv1**, **tv2**;  
}

## MainActivity

*// 点击listView里面的item,进入到第二个页面用来修改笔记***listView**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Intent intent=**new** Intent(getApplicationContext(),SecondAtivity.**class**);  
 intent.putExtra(**"ids"**,**arrayList**.get(position).getIds() );  
 startActivity(intent);  
 MainActivity.**this**.finish();  
 }  
});

*//长按后来判断是否删除数据***listView**.setOnItemLongClickListener(**new** AdapterView.OnItemLongClickListener() {  
  
 @Override  
 **public boolean** onItemLongClick(AdapterView<?> parent, View view,  
 **final int** position, **long** id) {  
 *//AlertDialog,来判断是否删除笔记。* **new** AlertDialog.Builder(MainActivity.**this**)  
 .setTitle(**"删除"**)  
 .setMessage(**"是否删除笔记"**)  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
  
 }  
 }).setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 **myDataBase**.toDelete(**arrayList**.get(position).getIds());  
 **arrayList**=**myDataBase**.getArray();  
 MyAdapter adapter=**new** MyAdapter(**inflater**,**arrayList**);  
 **listView**.setAdapter(adapter);  
 }  
 }).create().show();  
 **return true**;  
 }  
});

*//按钮点击事件，用来新建笔记***button**.setOnClickListener(**new** View.OnClickListener() {  
  
 @Override  
 **public void** onClick(View v) {  
 Intent intent=**new** Intent(getApplicationContext(),SecondAtivity.**class**);  
 startActivity(intent);  
 MainActivity.**this**.finish();  
 }  
});

## SecondAtivity

Intent intent = **this**.getIntent();  
 **ids** = intent.getIntExtra(**"ids"**, 0);  
 *//默认为0，不为0,则为修改数据时跳转过来的* **if** (**ids** != 0) {  
 **saveData** = **myDatabase**.getTiandCon(**ids**);  
 **ed1**.setText(**saveData**.getTitle());  
 **ed2**.setText(**saveData**.getContent());  
 }

*//保存按钮的点击事件，和返回按钮是一样的功能，所以都调用isSave()方法；* **bt1**.setOnClickListener(**new** View.OnClickListener() {  
  
 @Override  
 **public void** onClick(View v) {  
 isSave();  
 }  
 });  
}

*// 返回按钮调用的方法。*@Override  
**public void** onBackPressed() {  
 *//super.onBackPressed();* SimpleDateFormat formatter = **new** SimpleDateFormat(**"yyyy.MM.dd HH:mm:ss"**);  
 Date curDate = **new** Date(System.*currentTimeMillis*());*//获取当前时间* String times = formatter.format(curDate);  
 String title = **ed1**.getText().toString();  
 String content = **ed2**.getText().toString();  
 *//是要修改数据* **if** (**ids** != 0) {  
 **saveData** = **new** SaveData(title, **ids**, content, times);  
 *//saveData = new SaveData(title, ids, content);* **myDatabase**.toUpdate(**saveData**);  
 Intent intent = **new** Intent(SecondAtivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 SecondAtivity.**this**.finish();  
 }  
 *//新建笔记* **else** {  
 **if** (title.equals(**""**) && content.equals(**""**)) {  
 Intent intent = **new** Intent(SecondAtivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 SecondAtivity.**this**.finish();  
 } **else** {  
 **saveData** = **new** SaveData(title, content, times);  
 **myDatabase**.toInsert(**saveData**);  
 Intent intent = **new** Intent(SecondAtivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 SecondAtivity.**this**.finish();  
 }

**private void** isSave() {  
 SimpleDateFormat formatter = **new** SimpleDateFormat(**"yyyy.MM.dd HH:mm:ss"**);  
 Date curDate = **new** Date(System.*currentTimeMillis*());*//获取当前时间* String times = formatter.format(curDate);  
 String title = **ed1**.getText().toString();  
 String content = **ed2**.getText().toString();  
 *//是要修改数据* **if** (**ids** != 0) {  
 **saveData** = **new** SaveData(title, **ids**, content,times);  
 **myDatabase**.toUpdate(**saveData**);  
 Intent intent = **new** Intent(SecondAtivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 SecondAtivity.**this**.finish();  
 }  
 *//新建笔记* **else** {  
 **saveData** = **new** SaveData(title, content, times);  
 **myDatabase**.toInsert(**saveData**);  
 Intent intent = **new** Intent(SecondAtivity.**this**, MainActivity.**class**);  
 startActivity(intent);  
 SecondAtivity.**this**.finish();  
 }  
}

# 八、效果展示

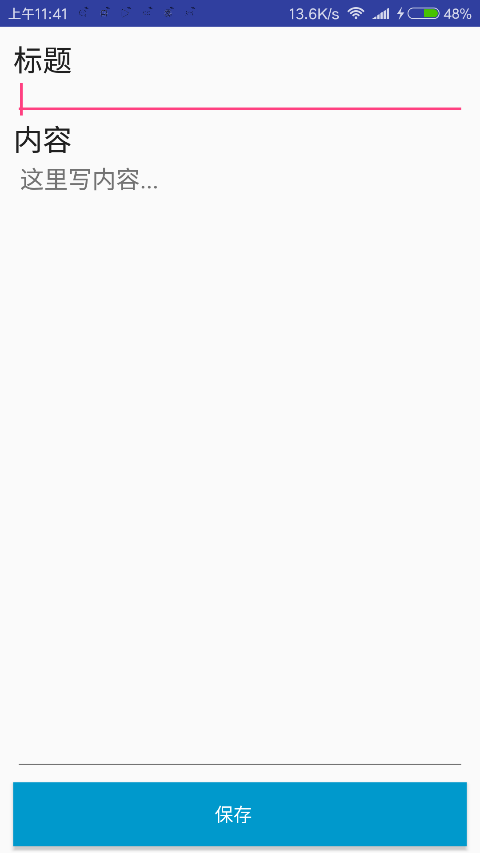
**主界面**



**长按弹出选择删除框**



**编辑界面**



# 九、结语展望

感觉自己在编写程序时最大的问题还是前期的规划，虽然开始写之前有一个大致的目标，也思考了都有几个类，每个类的属性方法之类的，但只想了个大概，真的动手写还是有点想到哪里写哪里的感觉，就会混乱，缺乏整体感，然后再去整理规划有点浪费时间精力了。以后再做程序的时候，开始写之前应该明确想清楚设计思路，并且落实写下来，有一个明确的整体规划。

因为时间有限，所以现在就只做到这个程度，之后还想加入标签和搜索功能。对于每一条笔记有个标签，方便查找和分类。新增一个activity是关于label标签的一个listview，数据存储上在原有基础上加一个label属性。在列表显示页显示部分内容等。