

diary 说明文档

完成人：詹萍

学号：20152100027

完成时间：2017 年 12 月 10 日

一、软件名称

diary

二、软件内容简介

这是一款记录日常事件的日记本软件。

本软件既可以当成日记本使用，也可以当作记事本来使用，核心是实现记录的功能！

为了方便用户记录事件、作业等信息的功能，本软件主要实现了浏览日记列表、添加日记、查看已有日记内容、修改日记、删除日记、查询日记等功能。

三、界面设计

1、主页

本次的主页 MainActivity 使用的是系统自带的框架 Basic Activity。

这个框架已经实现了页面中浮动的按钮（在本软件中作为添加日记使用），Toolbar 的工具栏、具有不同 item 的 Menu。使用时只需要在这些控件对应的方法内实现对应的功能即可。

引起我注意的主要是将 content_main 的内容显示在 activity_main 中。

实现方法：

在 activity_main.xml 中添加如下代码：

```
<include layout="@layout/content_main" />
```

在 content_main.xml 中添加如下代码：

```
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="com.example.win.diary.MainActivity"
tools:showIn="@layout/activity_main"
```

主页的元素主要有：粗体展示标题“我的日记”，ListView 展示日记列表、添加日记的圆形加号按钮、Toolbar 上面的 Menu 的实现、隐藏的搜索栏。如图 1 所示。

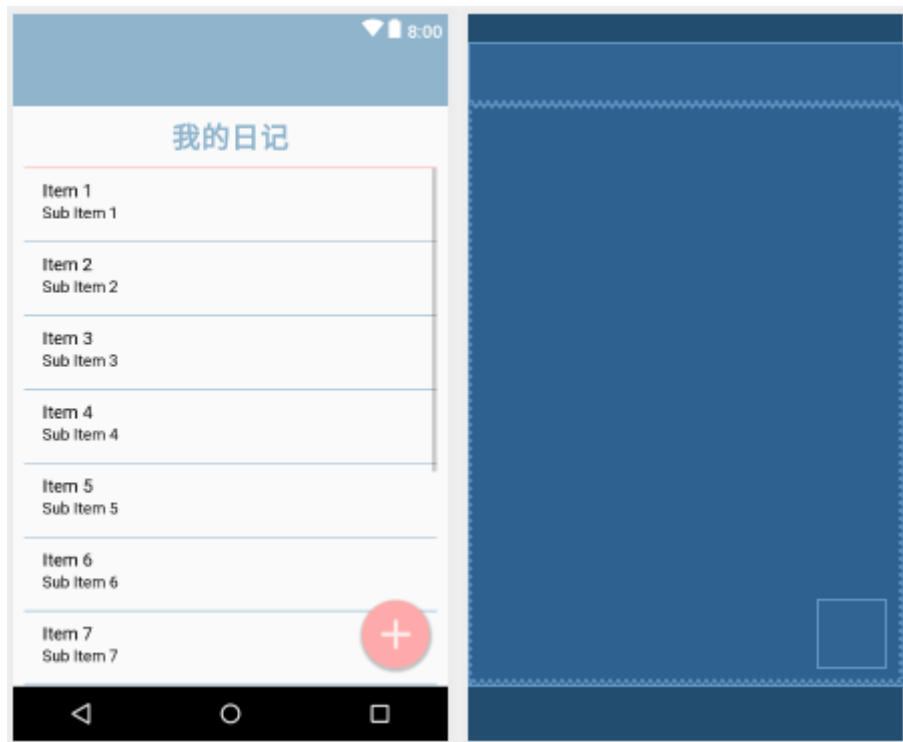


图 1

2、编辑日记信息的页面。

在点击添加日记的时候是添加日记的页面，主要元素有：可编辑的标题栏、可编辑的内容栏、按钮“**确定**”和按钮“取消”。

在点击求改日记的时候是修改日记的页面，主要元素有：显示原有标题的可编辑的标题栏、显示原有内容可编辑的内容栏、按钮“**修改**”和按钮“取消”。

将两个布局相识的功能使用同一个页面来实现。

如图 2 所示。



图 2

3、显示在 ListView 上的日记 item

主要元素有：根据日期不同二显示不同的图标，可缩略显示的日记标题。

如图 3 所示。

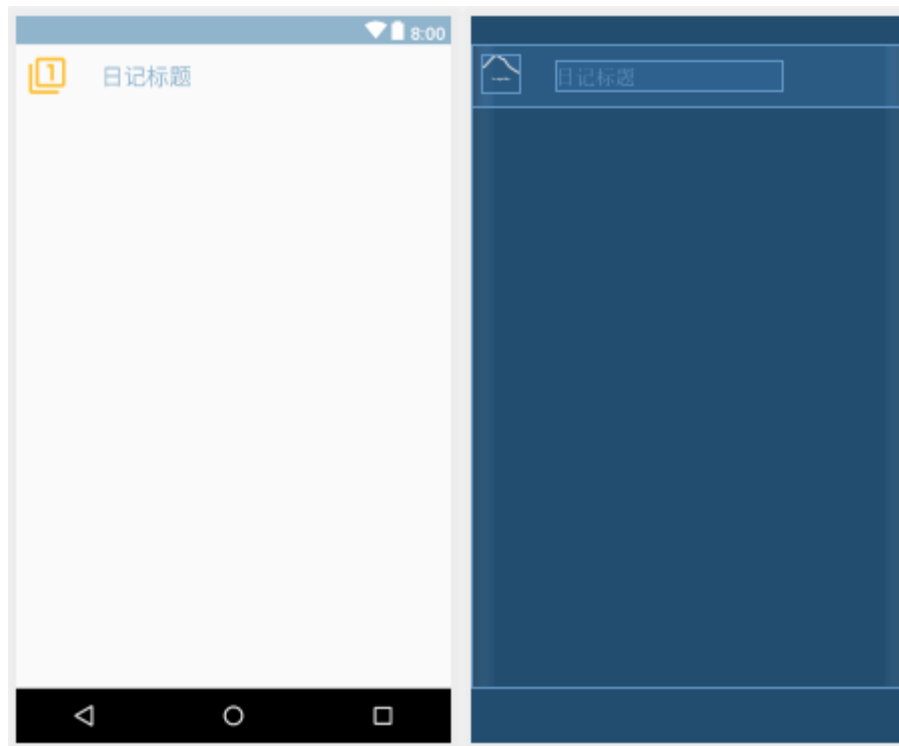


图 3

四、代码设计

1、代码目录展示。如图 4 所示。

①将 MyHelper 类（继承自 SQLiteOpenHelper 类）独立写成一个类文件，保证了该类良好的可重用性。在类中建立本软件的数据库 diary.db 以及记录日记信息的表格 diary。

②将 MyAdapter 类（继承自 BaseAdapter 类）独立写成一个类文件，保证了该类良好的可重用性。在类中实现对 ListView 的适配，每一个 item 的数据通过构造函数的参数传进本类中。

③将日记的信息封装成一个类 MyDiary。方便信息的读取和存储。

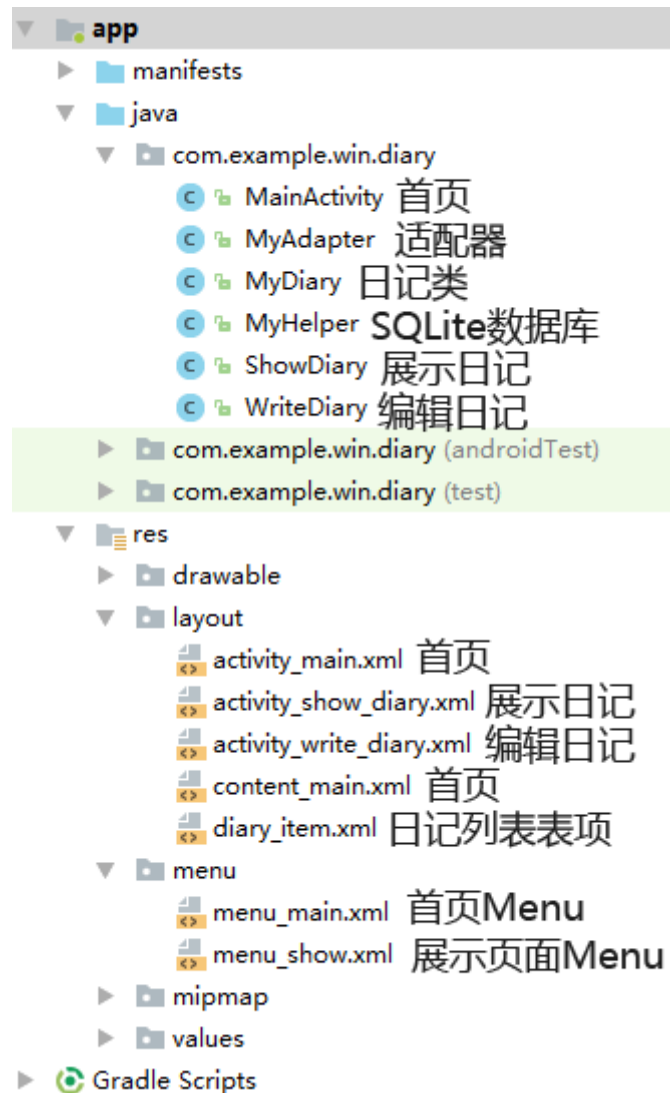


图 4

2、ListView 以及 Item 的亮点

①将新添加的 Item 显示在 ListView 的最上面。

即按照添加的时间顺序，将最新的内容显示在最上方。

实现方法，将数据从数据库读出时从最后面开始往回读，这样就可以将新的数据适配在 ListView 的上方。

②不同的日期显示不同的图标。

如图 5 所示。在图 5 中为了展示明显，我用日记标题来说明了周一到周日各自对应的图标。UI 界面通过不同图标的点缀变得比较吸引用户。

实现方法主要是根据添加日记时获取的系统时间，通过编写函数来将日期转化为对应的周一到周日，并设置对应的图标。设定好 Calendar 类对象年月日之后，获取属性 Calendar.DAY_OF_WEEK 之后即可得到对应的星期几。



图 5

③对于标题较长的 item 采用缩略显示。

如图 6 所示。在标题末尾显示“...”

实现方法：

在 item 中标题的 TextView 设置如下属性即可：

```
android:singleLine="true"
android:ellipsize="end"
```



图 7

2、搜索功能的亮点。

①SearchView 的显示与隐藏

在不需要使用搜索功能时隐藏 SearchView。通过设置 SearchView 的属性：`android:visibility="gone"` 即可隐藏 SearchView。

在需要搜索时，点击 Toolbar 的 Menu 的 item “搜索”，之后将 SearchView 的属性设置为可见，并且将 item 的 Title 文本设置为“结束搜索”。再点击“结束搜索”则将属性设置为不可见，Title 设置为“搜索”。

实现代码如下：

```
//结束搜索
```

```
if(searchDiary.getVisibility() == View.VISIBLE) {
```

```

        searchDiary.setVisibility((View.GONE));
        item.setTitle("搜索");
    }
    //搜索
    else if(searchDiary.setVisibility() == View.GONE) {
        //设置查找的控件可见
        searchDiary.setVisibility(View.VISIBLE);
        item.setTitle("结束搜索");
    }
}

```

②搜索时的动态显示。

在点击搜索之后显示出对应的搜索结果，其余情况下只要没有点击搜索均展示原先的列表的所有内容。

实现在取消搜索之后展示原先的列表的所有内容、实现在删除搜索文本的时候显示原先的列表的所有内容。

主要是给 SearchView 设置事件：setOnQueryTextListener()。在其中：

```

public boolean onQueryTextChange(String s) {

    //正常情况下将列表设置为全部数据
    mlistview.setAdapter(new MyAdapter(MainActivity.this,DiaryList));
    return true;
}

```

3、删除功能的亮点。

①长按删除。

长按 ListView 的某个 item 弹出对话框询问：是否删除日记，对话框如图 8 所示。

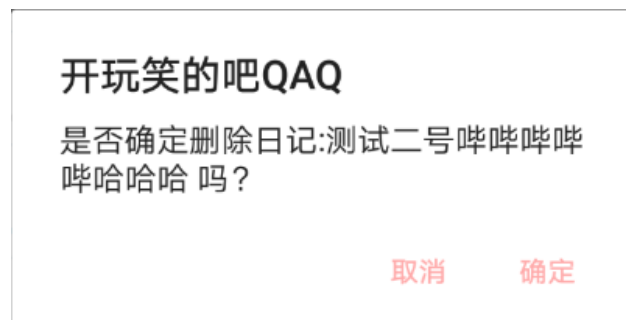


图 8

实现方法：主要是在适配器的 getView 方法中实现长按的监听事件，并对对话框的按钮设置点击事件，代码如下所示：

```

//为每一个 diary item 设置长按删除的监听事件
diary.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View view) {
        AlertDialog hintDialog;
        hintDialog = new AlertDialog.Builder(context)
            .setTitle("开玩笑的吧 QAQ")
    }
}

```

```

        .setMessage("是否确定删除日记:" +
tv_showTitle.getText() + " 吗? ")
        .setPositiveButton("确定", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialogInterface, int i) {
                //从数据库删除
                MyHelper myHelper = new MyHelper(context);
                //获得可以读取数据的 SQLiteDatabase 对象
                SQLiteDatabase db =
myHelper.getWritableDatabase();
                db.delete("diary", "_id=?",
                    new
String[] {diaryList.get(p).getId()});
                //从日记列表删除
                diaryList.remove(p);
                notifyDataSetChanged();
            }
        })
        .setNegativeButton("取消", null)
        .show();
    return true;
}
});

```

②在展示日记的页面的 Menu 中设置 item: 删除日记。

实现方法: 设置 MenuItem 的选中事件, 代码如下所示:

//删除日记

```

if (id == R.id.action_delete) {
    //从数据库删除
    MyHelper myHelper = new MyHelper(this);
    //获得可以读取数据的 SQLiteDatabase 对象
    SQLiteDatabase db = myHelper.getWritableDatabase();
    db.delete("diary", "_id=?", new String[] {showDiary.getId()});
    //跳转回到首页
    Intent intent = new Intent>ShowDiary.this, MainActivity.class;
    startActivity(intent);
    return true;
}

```

4、添加与修改页面共用。

经过观察我发现添加日记与修改日记的页面布局相似, 都是展示日记标题以及日记内容。因此我决定复用一個页面来实现这两个功能。

那么如何分辨跳转到这个页面的目的是添加还是修改呢?

很简单，只需要在跳转前设置跳转目的即可：

添加日记：

//设置跳转的目的

```
intent.putExtra("purpose", "add");
```

修改日记：

//设置跳转的目的

```
intent.putExtra("purpose", "update");
```

之后在根据不同的跳转来展示不同的内容，并且对按钮点击事件实现不同的功能即可：

添加时将标题和内容设置为空。点击按钮“确定”将数据写入数据库，并返回首页，点击按钮“取消”则直接返回首页。

修改时保留原有的标题和内容，让用户在这个基础上进行修改。点击按钮“修改”更新数据库中的内容并返回显示修改后的内容。点击按钮“取消”则直接返回显示页面。

```
Intent intent = getIntent();
purpose = intent.getStringExtra("purpose");
if(purpose.equals("add")) {
    title.setText("");
    content.setText("");
    enter.setText("确定");
}
else if(purpose.equals("update")) {
    updateDiary = (MyDiary) intent.getSerializableExtra("diary");
    title.setText(updateDiary.getTitle());
    content.setText(updateDiary.getContent());
    enter.setText("修改");
}
```

（点击事件代码实现比较简单此处省略。）

5、在 drawable 资源文件中实现圆角按钮。

实现比较简单，不赘述。

只需要在需要使用的地方引用即可。

五、软件操作流程

在真机上操作：

1、浏览日记列表。

上下滑动 ListView 的内容即可。

列表展示日记标题，并通过图标来展示日记记录的日期是属于星期几，像这样展示星期几对于用户来说对短期的日期有比较直观的感受。

为了展示下拉效果，我事先在数据库中添加了一些日记的数据，下拉列表。如图 9 所示。



图 9

2、添加日记。

- ① 点击圆形按钮“加号”从主页面跳转到添加日记页面。如图 10 所示。
- ② 编辑日记信息，点击按钮“确定”返回首页。或者直接点击按钮“取消”，返回首页。添加之后可以看到列表中新添加的内容。如图 11 所示。



图 10



图 11

3、搜索日记。

- ① 点击菜单项，显示“搜索”。
- ② 点击“搜索”即出现搜索栏，如图 12 所示。根据搜索栏的提示输入日记标题。



图 12

- ③ 输入后点击搜索 icon 搜索日记。删去搜索文本之后显示原列表。如图 13 所示。



图 13

- ④ 点击 Menu 的结束搜索。再次隐藏搜索栏。如图 14 所示。



图 14

4、点击查看日记详情。

点击 ListView 的某个 item 查看日记详情。如图 15 所示。内容比较长可以下拉显示，如图 16 所示。



图 15



图 16

5、修改日记。

点击 Menu 的“修改日记”，跳转到修改日记页面对日记进行修改。如图 17 所示。点击按钮“修改”或者按钮“取消”，返回展示日记的页面。如图 18 所示。



图 17



图 18

6、删除日记。

①在展示日记的页面的 Menu 选择“删除日记”，之后跳转回到首页，可以看到日记被删除。如图 19 所示。



图 19

②长按 ListView 的某个日记 item，弹出对话框，选择删除日记，可以看到日记被删除，如图 20 所示。



图 20

六、难点和解决方案

1、搜索与 ListView 的显示

① 难点：

点击搜索之后，更新 ListView 的显示，让其只显示搜索返回的结果。当没有点击搜索时，或者删除搜索栏的文本时，显示原先 ListView 中的所展示的所有 item，即搜索后返回的问题。

② 解决方案：

通过新建一个只用于保存搜索结果的 SearchList 来将每次搜索的结果保存起来，并用这个 List 的数据去作为适配显示，更新 ListView 的显示。

通过设置 SearchView 的文本变化监听事件，当文本变动（没有点击搜索时），显示所有的 item。

通过切换 Menu 上的“搜索”和“结束搜索”以及是否隐藏 SearchView 来控制搜索功能。

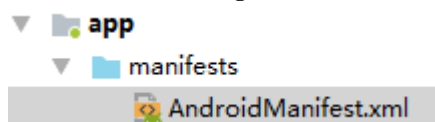
2、Activity 重复生成的问题

① 难点：

从主页面跳转到功能页面，再从功能页面跳转回到主页面；重复上述步骤，会发现任务栈中保留了很多主页面和功能页面的 Activity。这样在不断点击返回时系统会从任务栈中不断调出之前打开的压在栈底的 Activity，但我们希望每个页面只在栈中保留一份。

② 解决方案：

在如下目录的 AndroidManifest.xml 文件中对每个 Activity 的启动模式进行设置，设置为 singleTask 即可。



七、不足之处

1、暂且没有实现“假”删除的功能。当用户点击删除时真的删除了日记的数据，可能会出现用户误删的情况，但在这点上通过设置对话框提示可以减少这种误删的发生。

2、尚未实现模糊搜索，关于这点，我正在查看相关资料，貌似需要适配器的类继承过滤器的接口，尚待完善。

3、日记尚未做到图文并茂，只能添加文字。

八、今后的设想

- 1、实现假删除的功能，在数据库表项设置一个删除标志，当用户点击删除之后将标志的值设置为删除状态，实际不删除用户的日记的信息。
- 2、实现日记图文并茂，可以从手机相册选择或者打开相机来拍照获取图片。
- 3、实现模糊搜索，ListView 根据搜索的文本自动匹配、动态返回符合查询结果的表项。