

实验 2

——我的日记本 App

软件名称: _____ MyDiary _____

完成人: _____ 唐福梅 _____

学号: _____ 20152100025 _____

班级: _____ 15 级软件 5 班 _____

完成时间: _____ 2017-12-08 _____

一、软件内容简介

这是一款能够撰写日记及的软件，具有添加、编辑、删除（包括删除指定日记和清空日记列表）日记的功能，通过 SQLite 存储数据，然后通过 ListView 显示数据。

软件亮点：

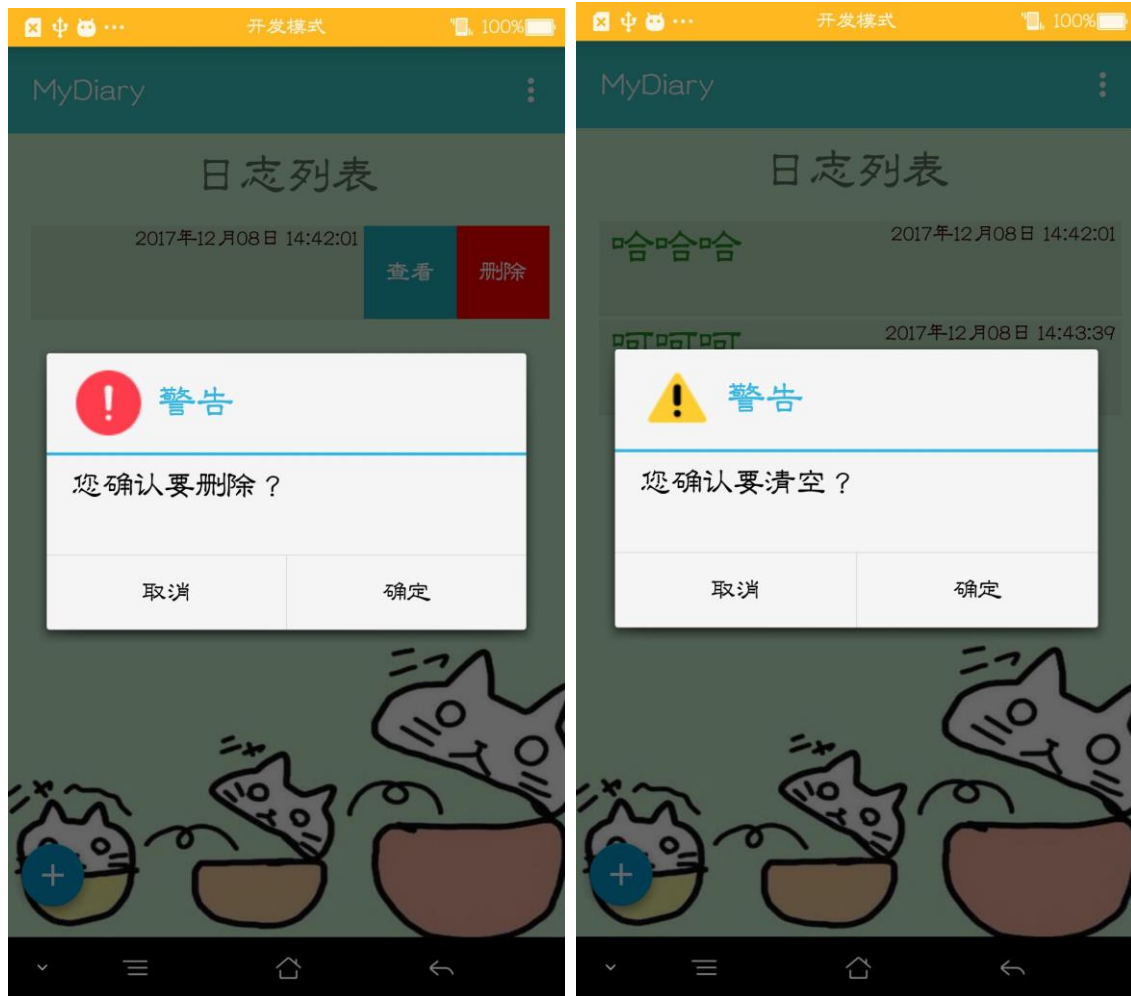
1) 使用了侧滑实现查看和删除日记。



2) 导入了 GitHub 上的一个悬浮按钮开源库。这个悬浮按钮是能够弹出二级悬浮按钮，二级悬浮按钮包括添加日记功能和回到日记顶部功能（浏览到后面的日记时可以一键回到日记列表头部）



3) 为了防止用户误删日记，使用了 AlertDialog 给用户进行提醒。



二、界面设计

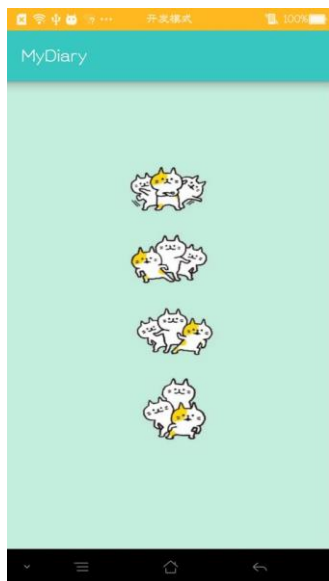
1) 简介

软件的主题色是绿色。绿色是比较积极向上的颜色，代表着活力、希望。从使用者的角度来说，看着绿色的背景，给人一种凉爽平静的感觉能让人的心情变得平静，眼睛没有那么容易疲劳。

2) 界面组成

软件一共有 startActivity、MainActivity、DiaryAdd、DiaryCheck、DiaryUpdate 五个 activity 组成。其中 startActivity 显示启动动画；MainActivity 显示日记列表，包含添加日记、删除（清空）日记、查看日记操作；DiaryAdd 界面是日记的添加界面；DiaryCheck 显示要查看的日记的具体信息，包括编辑日记的操作；DiaryUpdate 对日记进行修改。

3) 各个界面的背景



startActivity



MainActivity



DiaryAdd



DiaryCheck



DiaryUpdate

三、代码设计

(一) 开发过程

- (1) 按照实验二给出的示例日记本，构造了一个可以实现基本功能的日记本
- (2) 实现侧滑弹出删除日记功能
- (3) 重写 SimpleAdapter (使用 SimpleAdapter 对 ListView 进行数据适配)，设置侧滑菜单 (删除、查看) 的点击响应事件
- (4) 添加了悬浮按钮 (开源库) 并实现按钮的点击响应事件

(5) 为实现删除功能的控件（Toolbar 里面的清空和侧滑的删除）设置了 `AlertDialog`，防止用户误删。

(6) 进行软件测试，检查各个模块是否合理，进行代码的优化

1. 修改了查看日记的页面。

本来查看页面的布局和添加/修改页面是一样的，标题和日记内容都使用了 `EditText` 来显示日记的内容，后面改为了 `TextView`（只让用户查看不让其进行编辑）。重新设计了布局。在左上角显示了写日记的时间，日记的标题水平居中，颜色为绿色且字号偏大，日记的内容字号较小，因背景是绿色的所以将字的颜色设置为白色。



2. 更改了 `Listview` 的 `item` 的背景色。

`item` 一开始的背景色是白色（当时没有加背景图）。但是加了绿色的背景图之后，白色的背景色让 `item` 看起来很怪异。将 `item` 的背景色设为透明（十六进制为：`#11000000`）的之

后，看起来就舒服很多。



3. 测试时发现，修改界面只有保存和删除按钮，没有返回按钮。
将删除按钮改为了返回按钮。

(7) 设置软件图标和启动动画

(二) 具体实现

【1】对日记进行操作的函数（封装在 DiaryOperation 类中）

(1) 添加

```

public void add(Diary diary) {
    String sql = "insert into myDiary(title,content,pubdate) values(?,?,?)";
    //得到SQLite数据库
    db = dbHelper.getWritableDatabase();

    //执行sql语句
    db.execSQL(sql, new String[]{diary.getTitle(), diary.getContent(), diary.getPubdate()});

    db.close();
}

```

(2) 修改/更新日记

```

//更新
public void update(Diary diary) {
    //String sql = "update myDiary set title=?,content=?, pubdate=?, where id=?";

    //得到SQLite数据库
    db = dbHelper.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("title",diary.getTitle());
    values.put("content",diary.getContent());
    values.put("pubdate",diary.getPubdate());
    db.update("myDiary",values,"id=?", new String[]{Integer.toString(diary.getId())});

    db.close();
}

```

(3) 删除指定 id 的日记（侧滑删除的时候调用）

```

//删除指定id号的日志
public void delete(int id) {
    String sql = "delete from myDiary where id=?";
    //得到SQLite数据库
    db = dbHelper.getWritableDatabase();

    //执行sql语句
    db.execSQL(sql, new String[]{Integer.toString(id)});

    db.close();
}

```

(4) 根据 id 选择 item（侧滑查看时候调用）


```

//根据id选择item
public Diary select(int id) {
    Diary diary = null; //用于接受选择的日记
    String sql = "select * from myDiary where id=?"; //选择指定id号的日记

    //得到SQLite数据库
    db = dbHelper.getReadableDatabase();

    //执行sql语句
    Cursor cursor=db.rawQuery(sql, new String[]{Integer.toString(id)});
    if(cursor.moveToFirst()){
        String title=cursor.getString(cursor.getColumnIndex("title"));
        String content=cursor.getString(cursor.getColumnIndex("content"));
        String pubdate=cursor.getString(cursor.getColumnIndex("pubdate"));
        diary=new Diary(id,title,content,pubdate);
    }

    cursor.close();
    db.close();
    return diary;
}

```

(5) 删除全部日记（清空的时候调用）

```

//清空
public void deleteAll(){
    String sql = "delete from myDiary";
    //得到SQLite数据库
    db = dbHelper.getWritableDatabase();

    //执行sql语句
    db.execSQL(sql);

    db.close();
}

```

(6) 获取全部记录（初始化 ListView 控件时调用）

```

//获取所有日志
public List<Diary> getAllDiary(){
    List<Diary> list=new ArrayList<>();
    String sql="select * from myDiary";

    //得到SQLite数据库
    db=dbHelper.getReadableDatabase();

    //执行sql语句
    Cursor cursor=db.rawQuery(sql,null);
    while (cursor.moveToNext()){
        int id=cursor.getInt(cursor.getColumnIndex("id"));
        String title=cursor.getString(cursor.getColumnIndex("title"));
        String content=cursor.getString(cursor.getColumnIndex("content"));
        String pubdate=cursor.getString(cursor.getColumnIndex("pubdate"));

        Diary diary=new Diary(id,title,content,pubdate);
        list.add(diary);
    }

    cursor.close();
    db.close();
    return list;
}

```

【2】重写 SimpleAdapter 的 getView 方法，以实现侧滑查看以及删除功能

（1）实现删除键的点击事件监听

```

listItemView.my_delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        int id=Integer.parseInt(listItem.get(mPosition).get("id").toString());
        new DeleteDialog(id,new DiaryOperation(activity),activity).delete();
    }
});

```

（2）实现查看键的点击事件

```

listItemView.my_check.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent=new Intent(context,DiaryCheck.class);

        intent.putExtra("id",Integer.parseInt(listItem.get(mPosition).get("id").toString()));

        activity.startActivity(intent);
        activity.finish();
    }
});

```

【3】将弹出提示对话框封装成 AlertDialog 类

```

public void delete() {
    AlertDialog.Builder alertdialogbuilder=new AlertDialog.Builder(context);
    alertdialogbuilder.setIcon(R.drawable.delete_alert);
    alertdialogbuilder.setTitle("警告");
    alertdialogbuilder.setMessage("您确认要删除?");
    alertdialogbuilder.setPositiveButton("确定", click1);
    alertdialogbuilder.setNegativeButton("取消", click2);
    AlertDialog alertDialog=alertdialogbuilder.create();
    alertDialog.show();
}

private DialogInterface.OnClickListener click1=(arg0, arg1) -> {;
    DO.delete(id);

    //提示
    Toast.makeText(context,"删除成功!",Toast.LENGTH_SHORT).show();

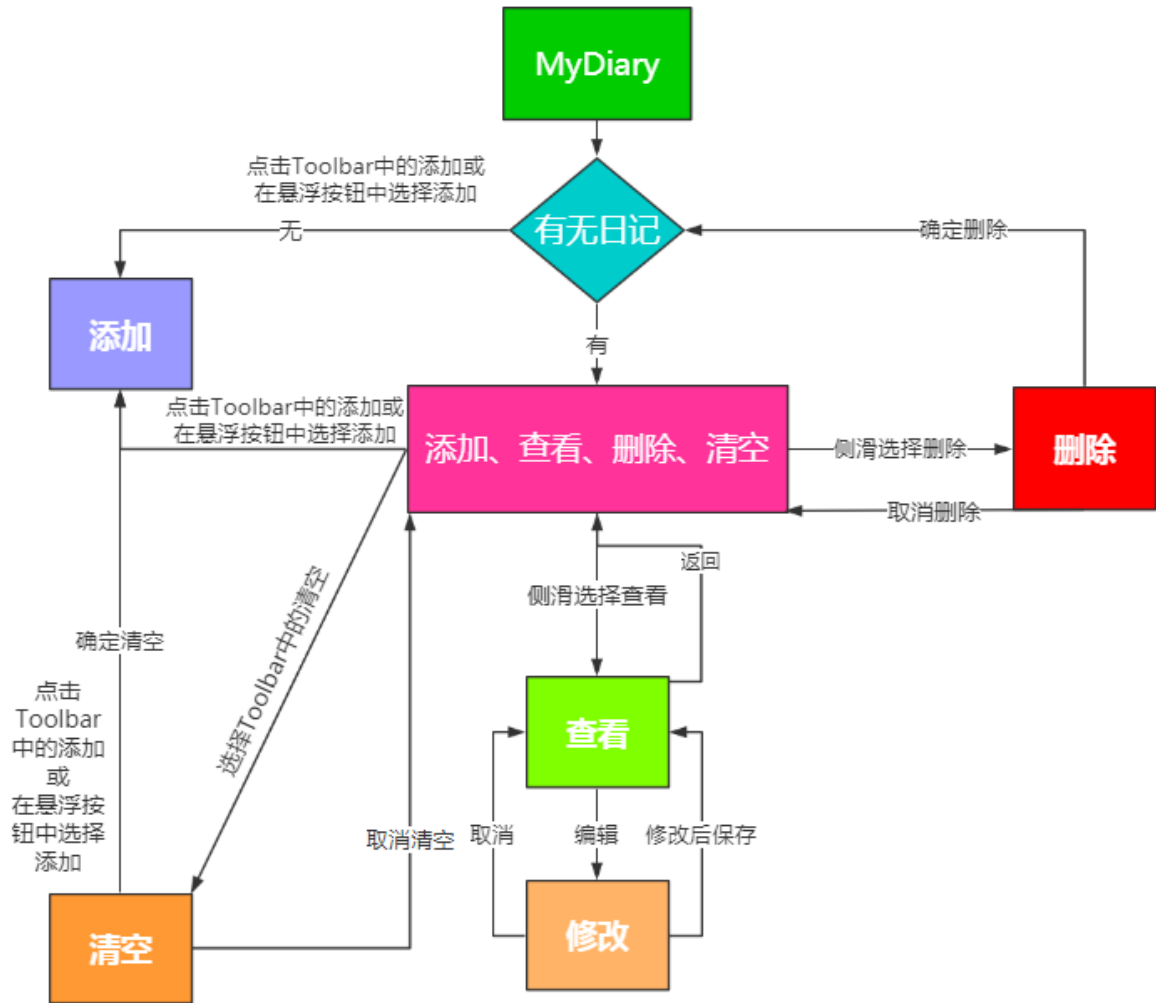
    //直接跳转到日志列表视图
    Intent in=new Intent(context,MainActivity.class);
    context.startActivity(in);
    Activity activity = (Activity) context;
    activity.finish();
};

private DialogInterface.OnClickListener click2=new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface arg0,int arg1) {
        arg0.cancel();
    }
};

```

四、软件操作流程

（一）流程说明

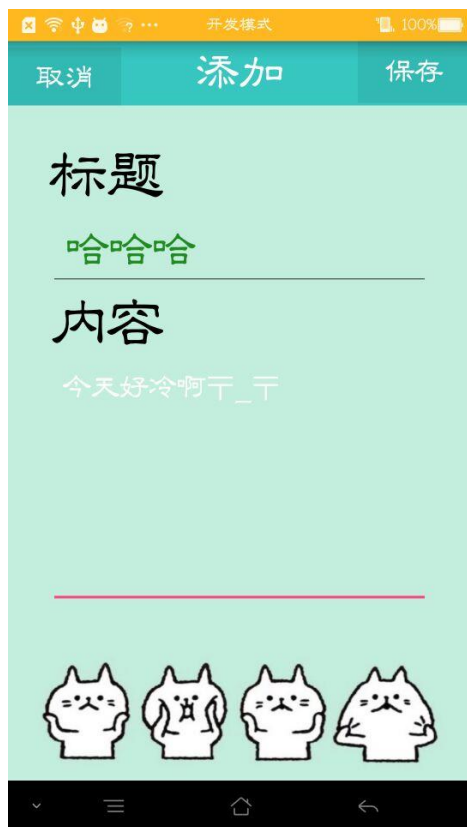


(二) 测试截图

(1) 主界面



(2) 添加



(3) 查看



(4) 修改

取消

修改

保存

标题

干_干

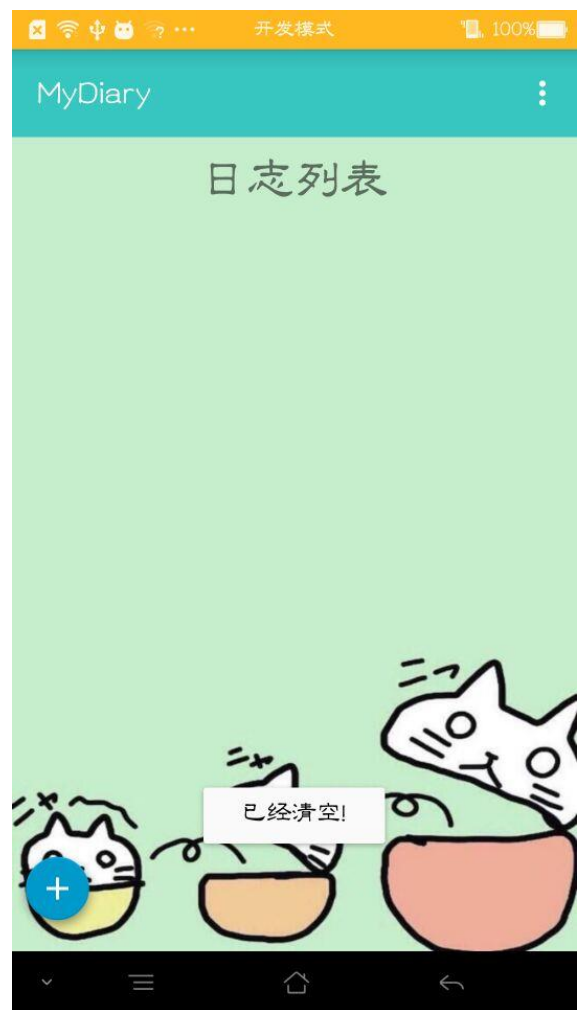
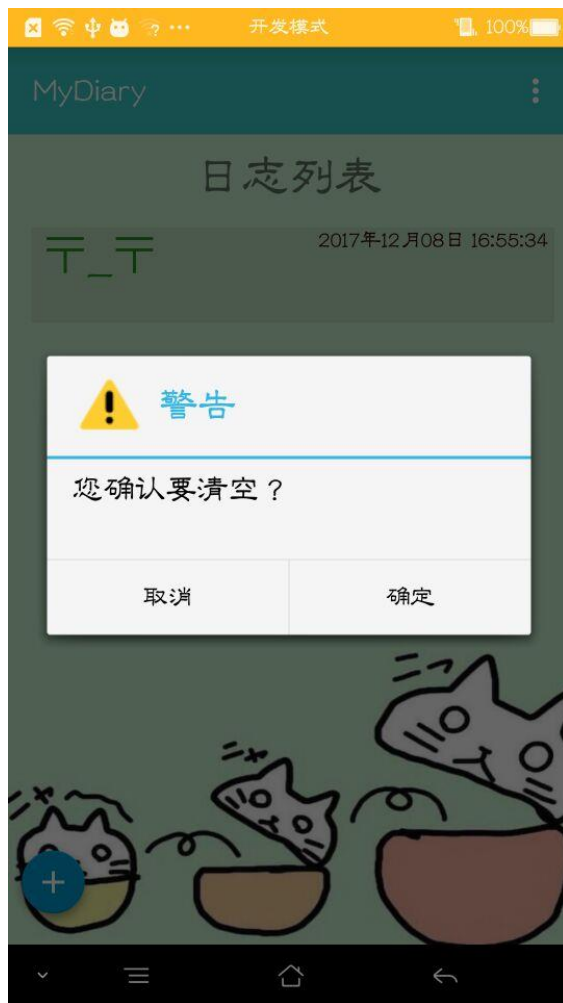
内容

今天好冷啊干_干

(5) 删除



(6) 清空



五、难点（或遇到的问题）和解决方案

【问题 1】编写好代码、编译没有出错后，连接手机进行调试。一点击软件就出现闪退的情况。

【解决 1】是建表的 SQL 语句缺少了一个右括号导致的。加上即可。（在同一个地方跌倒两次 o(∩_∩)o）

```
db.execSQL("CREATE TABLE myDiary(id INTEGER PRIMARY KEY
AUTOINCREMENT," +
        "title VARCHAR(20),content VARCHAR(1000),pubdate
VARCHAR(50))");
```

【问题 2】点击了清空键，ListView 没有及时刷新

【解决 2】

做完清空操作后加上以下语句 `listView.setAdapter(null);` //将适配器设置为 null 就可以立即显示清空的结果（在 MainActivity 中的 onCreate 方法中，listview 会

重新进行数据适配器的绑定)

【问题 3】使用了 `SwipeLayout` 实现侧滑弹出删除、查看按钮，但是不知道要怎么实现事件的监听

【解决 3】继承 `ListView` 的数据适配器，重写 `getView` 方法，在里面进行事件的监听

六、不足之处和今后的设想

不足之处：

- 1) 由于使用了侧滑布局实现侧滑删除功能导致 `ListView` 原来的点击事件响应有问题，所以把直接点击查看改成了侧滑查看。即没有实现直接点击查看日记功能
- 2)

今后的设想：

- 1) 添加上直接点击查看日记的功能
- 2) 给软件加上一些动效，以实现更好的交互，让用户体验更好。

七、参考文献

[1] 克拉幸福 ran. android 数据存储：使用 `SQLite` 开发简易心情日记[DB/OL].

<http://www.jianshu.com/p/b4fde6b835a3>, 2012,2012-11-12/2017-12-06

[2] developerHaoz. Android 一款十分简洁、优雅的日记 APP [DB/OL].

<https://zhidao.baidu.com/question/556306924.html>, 2017,2017-01-27/2017-12-06

[3] 水田如雅. Android——继承 `SimpleAdapter` 重写自己的 `getView` [DB/OL].

<http://blog.csdn.net/lhc1105/article/details/49330227>, 2015,2015-10-22/2017-12-06

[4] 俞其荣. 让你的 `ListView` 更炫酷，实现侧滑删除效果 [DB/OL].

<http://www.jianshu.com/p/d738bfcac7c7>,2016, 2016-08-17/2017-12-07

[5] 学习园地. Android Studio 导入第三方类库的方法 [DB/OL].

<http://www.cnblogs.com/neozhu/p/3458759.html>,2013, 2013-12-05/2017-12-07

[6] @鉴客. 7 种形式的 Android Dialog 使用举例 [DB/OL].

https://oschina.net/question/54100_32486,2011, 2011-05-17/2017-12-08

八、引用的 GitHub 开源代码

(1) trity1993 的 FloatingActionButton

链接地址为: <https://github.com/trity1993/FloatingActionButton>

(2) lzyzd 的 SwipeLayout

链接地址为: <https://github.com/lzyzd/SwipeLayout>