

实验二

我的日记本app设计与实现说明文档

作者：梁子宏 学号：20162180176

完成时间：2018.12.24

一、软件名称

我的日记本。我的日记本是安卓课程上老师布置的第二个实验，要求现场完成并展示。

基本功能是利用 `sqlite` 和 `listview` 去完成日记本的基本功能，增加日记，添加背景音乐等。

二、软件内容简介

知识点使用：**ListView** 的优化（滑动图片错乱的原因以及解决方法），**MVP** 设计模式，上传图片功能，`Sqlite` 的基本操作，**Activity** 的规则化启动方法，**ListView** 的难用之处。

老师要求实现的是一个纯文字的“我的日记本”app，可以添加一些附加功能比如播放音乐，`sqlite` 数据库的相关操作。

以上的基本功能由于老师课堂上讲得非常详细，在这里基本的操作就不展现出来。

以往一样，**MVP** 对于大部分的 `android` 应用是一个非常合适的设计模式。其普通思路即为根据功能对 `activity` 和 `presenter` 设计两个接口，并且实现，在启动过程中，`activity` 和 `presenter` 互相持有，于是

在以往需要在 `activity` 中实现的功能代码搬到 `presenter` 上，做到功能和界面代码的解耦，便于实现和对功能进行小组分工。

当然，不足之处就是需要编写的代码就比原来的要更多，但我认为在老师布置的作业和实验 `app` 上多加使用，面对中大型程序时才不会不知所措一脸茫然。

在特色功能方面，加上配图应该算是一个。选择图片配上文字内容发布，应该才是日记本的精髓。而图片的上传，则涉及到 `Uri` 和 `File` 的使用。

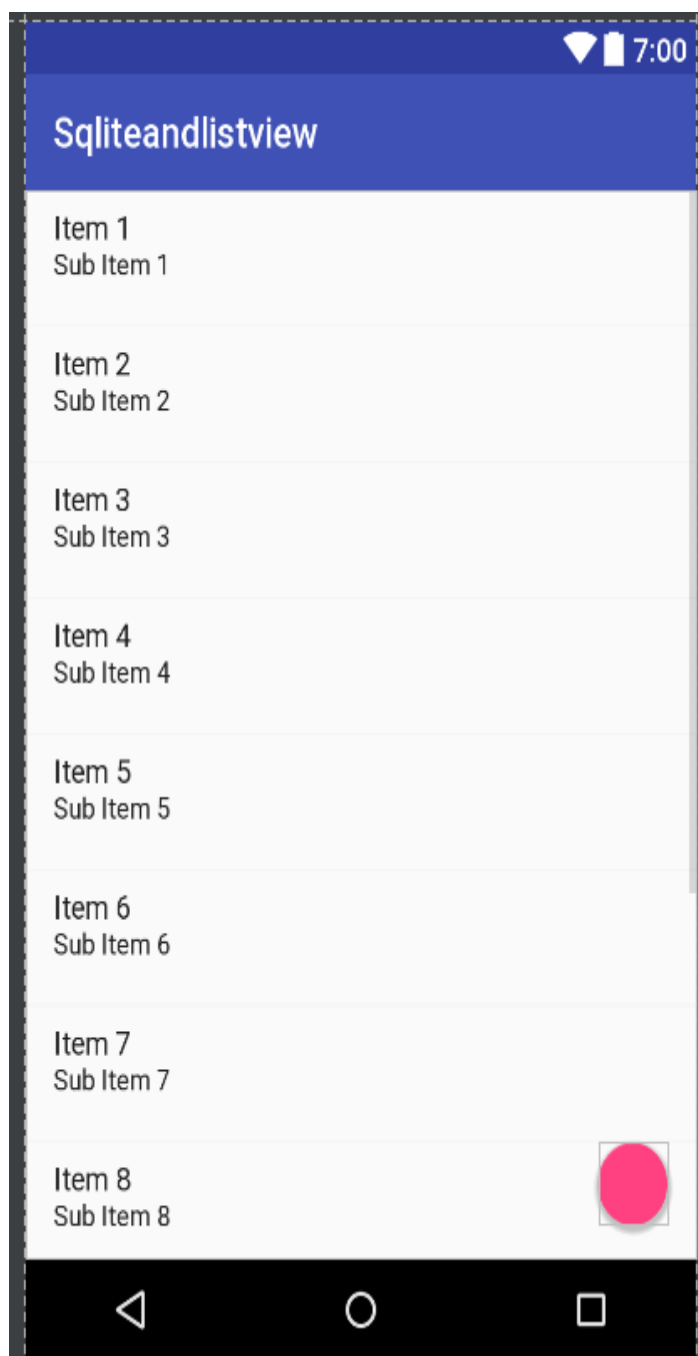
而在代码的具体编写上面，我对 `Activity` 的跳转方式做了进一步的统一化，将每个 `Activity` 的跳转写到自己的静态函数中，并且在参数中暴露想要传递的数据，如此一来，在面对多个 `activity` 时，不会因为忘记参数而浪费时间查找，也不会导致开发的其他组员面对这个 `activity`，不知道要传什么参数进去。

最后，想谈一谈 `ListView`。`ListView` 是我用过最难用的控件，原因在于我们初学者如果不能深入地去探究一下 `ListView` 的源码，我们是不能完全驾驭 `ListView` 的使用，会出现各种各样的问题。我在后面将会说一下我遇到的一些坑。

所以这个 `APP` 我的界面并没有太多设计，以简约为主。

三、界面设计

1、主界面



2、添加日记功能



Sqliteandlistview

标题

标题

日记内容

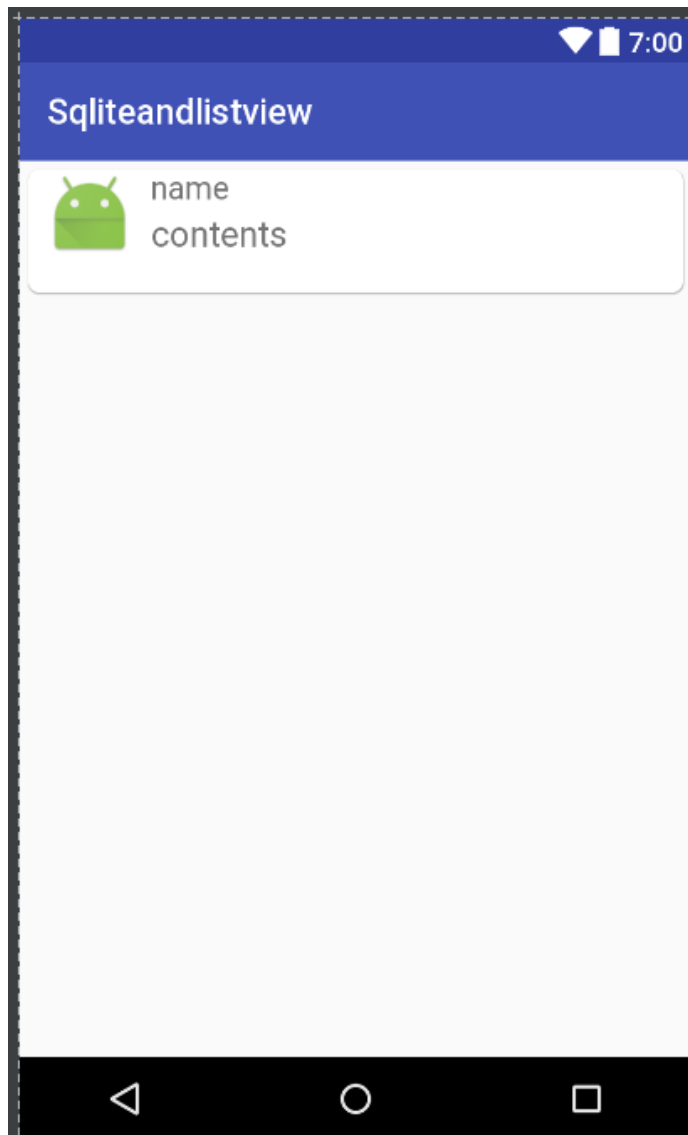
写下对此题的感想

图片

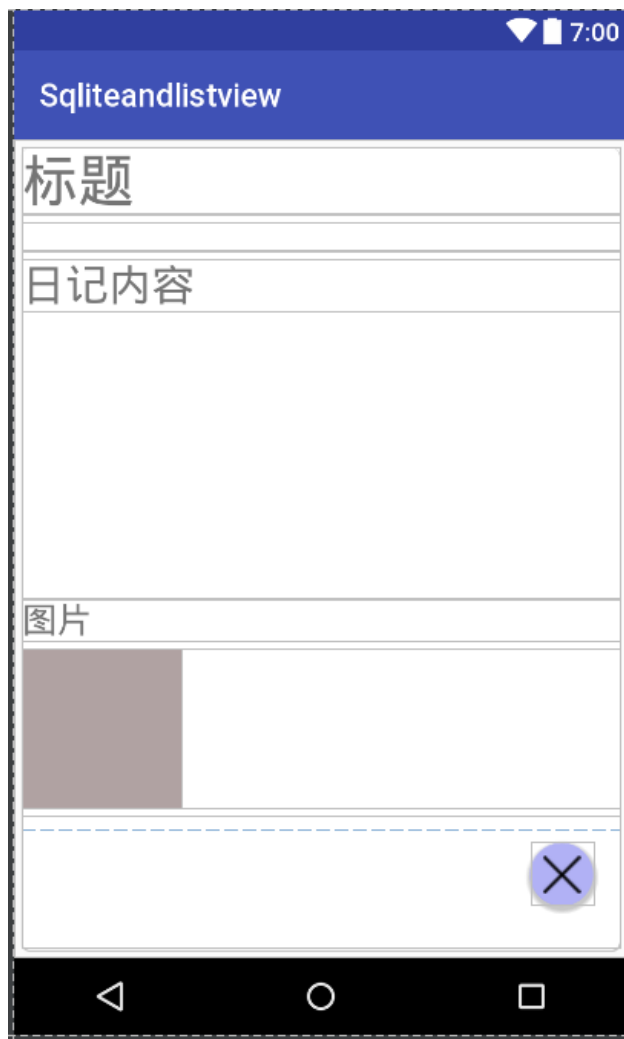
+

发送

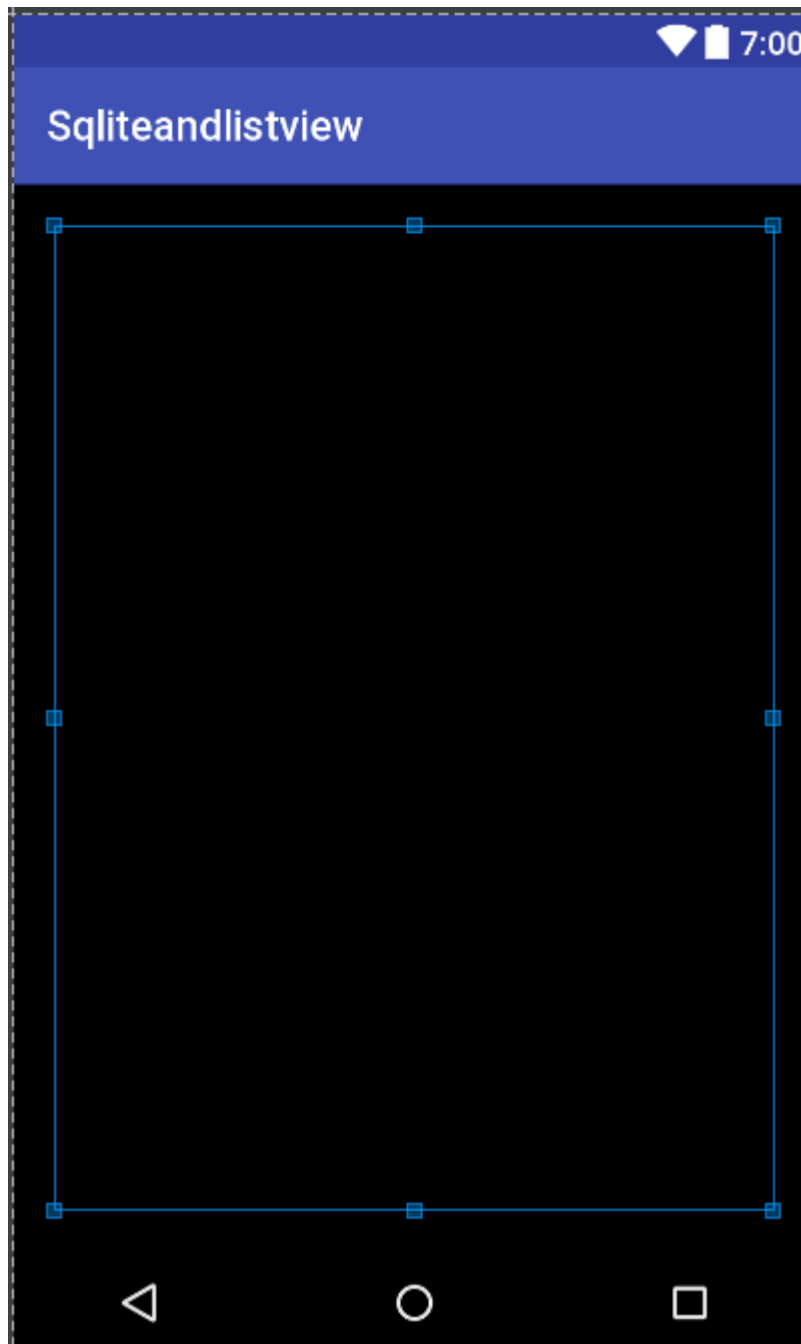
3 提供 ListView 使用的 Item 界面



4、观看日记详情和删除功能

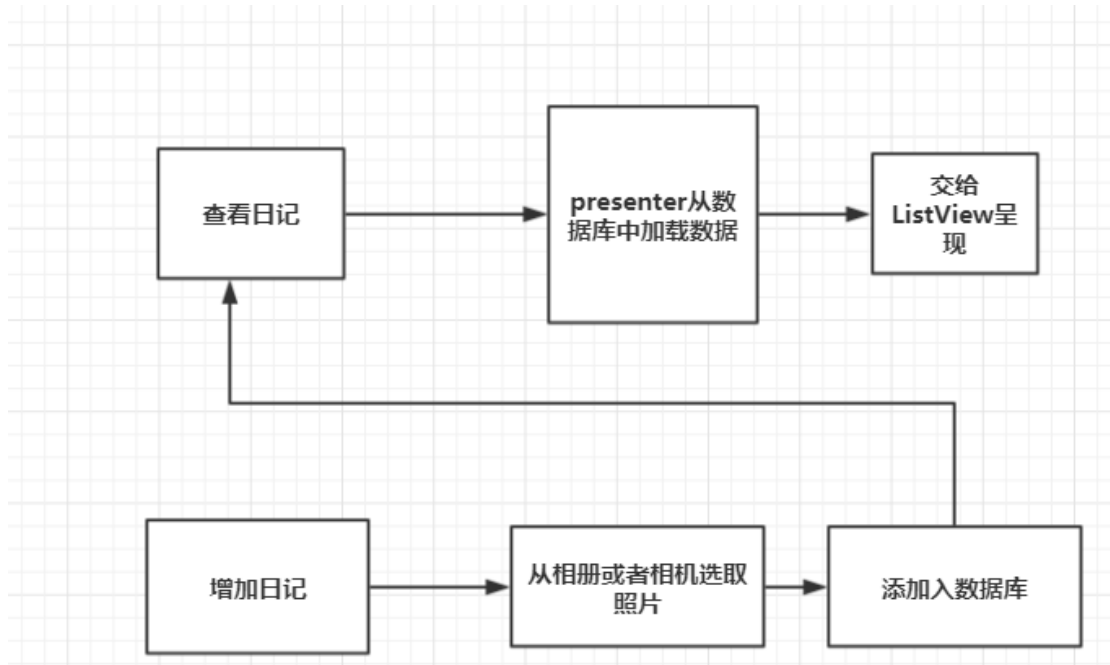


5、点击图片放大观看功能



四、代码设计（代码在最后）

五、我的日记本流程图



六、程序的不足之处

1、因为这是实验课，所以没有太多时间设计 UI，所以界面比较简单，没有太多的设计。

2、ListView 在这里我没有在代码中介绍 ListView 与 ScrollView 的冲突，因为这两者的配合在 评论 展示中是经常使用的。

3、没有进行数据库的查询功能。

七、难点和解决办法

1、图片的选取和展示。

图片选取在安卓上虽然只是很小的一段功能，但是要编写的代码并不少。

首先或获取 SD 卡的读写权限和相机的使用权限。

接着使用 `Intent` 根据用户选择是相册还是相机而跳转到相应的界面中。并且要在主界面中使用 `onActivityResult` 来接受返回的信号，根据信号的不同，使用不同的函数将返回的 `Intent` 数据解析出来并展示照片。

在相机和相册的选择函数中，通常会涉及到 `Uri` 的获取和解析，`Uri` 可以理解为 `android` 系统对某个文件的路径，所以根本目的我们是拿到 `Uri` 就能把照片显示出来。

而在一些网络服务器上，如果要上传图片，不能仅仅上传 `Uri`，而是创建文件。此时就要根据 `Uri` 去获取文件名或者是创建新的文件。

获取相册和相机代码整理如下：

```
// -----  
// -----  
  
// 选择照片的一堆函数  
  
// 打开那个选择框  
private void showNormalDialog(){  
    final AlertDialog.Builder normalDialog =
```

```

        new AlertDialog.Builder(this);
normalDialog.setTitle("选择照片的途径");
normalDialog.setMessage("打开");
normalDialog.setPositiveButton("相册",
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            applyPermission();
        }
    });
normalDialog.setNegativeButton("拍照",
    new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            goTakingPhoto();
        }
    });
// 显示
normalDialog.show();
}

/**
 * 申请权限
 */
private void applyPermission(){
    //如果没有得到 读写 sd 卡的权限
    if(ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED)
        requestPermissions(new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},100);
    else {
        openAlbum();
    }
}

/**
 * 这里反馈是否得到权限，如果得到权限就继续 openAlbum，得不到就会断掉
 */
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
    switch (requestCode){

```

```

        case 100:
            //如果得到权限
            if(grantResults.length>0 && grantResults[0]==
PackageManager.PERMISSION_GRANTED)
                openAlbum();
            else
                Toast.makeText(this, "你拒绝了权限", Toast.LENGTH_SHORT).show();
        }
    }

/**
 * 拍照 启动函数
 */
private void goTakingPhoto(){
    String name="xxx"+".jpg";

    //创建 图片将要存放的位置
    file=new File("/sdcard/Pictures/",name);
    try{

        file.createNewFile();
        filePath = file.getPath();
    }catch (Exception e){
        e.printStackTrace();
    }

    if(Build.VERSION.SDK_INT>=24)
        imageUri= FileProvider.getUriForFile(this,"sqlite.fileprovider",file);
    else
        imageUri= Uri.fromFile(file);

    //启动相机
    Intent intent=new Intent("android.media.action.IMAGE_CAPTURE");
    intent.putExtra(MediaStore.EXTRA_OUTPUT,imageUri);
    startActivityForResult(intent,TAKE_PHOTO);
    return;
}

/**

```

```

    * 打开相册
    */
private void openAlbum(){
    Intent AlbumIntent=new Intent("android.intent.action.GET_CONTENT");
    AlbumIntent.setType("image/*");
    startActivityForResult(AlbumIntent, CHOOSE_FROM_ALBUM);
}

/**
 * 接收 照完相 或者选择完照片 的 回归信号
 * @param requestCode
 * @param resultCode
 * @param data
 */
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {

    switch(requestCode){
        case TAKE_PHOTO:
            if(resultCode==RESULT_OK){
                //将拍摄的照片显示出来
                try {
                    Bitmap bitmap =
BitmapFactory.decodeStream(this.getContentResolver()
                            .openInputStream(imageUri));
                    picture.setImageBitmap(bitmap);
                }
                catch (Exception e){e.printStackTrace();}

            }
            break;

        case CHOOSE_FROM_ALBUM :
            if(resultCode==RESULT_OK)
                //判断手机的 android 版本，高低版本用不同的函数方法去解决
                if(Build.VERSION.SDK_INT>=19)
                    handleImageOnKitKat(data);
                else
                    handleImageBeforeKitKat(data);
            break;
    }
}

```

```

        default:
            break;
    }

}

@TargetApi(19)
public void handleImageOnKitKat(Intent data){
    String imagePath=null;
    Uri uri=data.getData();
    imageUri = uri;
    if(DocumentsContract.isDocumentUri(this,uri)) {
        //如果是 document 类型的 uri, 则通过 document id 处理
        String docId = DocumentsContract.getDocumentId(uri);
        if ("com.android.providers.media.documents".equals(uri.getAuthority()))
        {
            String id = docId.split(":")[1]; //解析出数字格式的 ID
            String selection = MediaStore.Images.Media._ID + "=" + id;
            imagePath = getImagePath
                (MediaStore.Images.Media.EXTERNAL_CONTENT_URI, selection);

            // Toast.makeText(myActivity, "第一个", Toast.LENGTH_SHORT).show();
        } else if
("com.android.providers.downloads.documents".equals(uri.getAuthority())) {
            Uri contentUri = ContentUris.withAppendedId(Uri.parse("content://downloads/public_downloads"),
                Long.valueOf(docId));
            imagePath = getImagePath(contentUri, null);
            // Toast.makeText(myActivity, "第二个", Toast.LENGTH_SHORT).show();
        }
    }
    else if("content".equalsIgnoreCase(uri.getScheme())) {
        imagePath = getImagePath(uri, null);
        // Toast.makeText(myActivity, "第三个", Toast.LENGTH_SHORT).show();
    }
    else if("file".equalsIgnoreCase(uri.getScheme())) {
        Toast.makeText(this, "第四个", Toast.LENGTH_SHORT).show();
        imagePath = uri.getPath();
    }

    displayImage(imagePath);
}

```

```

private void handleImageBeforeKitKat(Intent data){
    Uri uri=data.getData();
    String imagePath=getImagePath(uri,null);
    displayImage(imagePath);

    Toast.makeText(this, "第0个", Toast.LENGTH_SHORT).show();
}

private String getImagePath(Uri uri, String selection){
    String path=null;
    //通过 Uri 和 selection 来获取真实的图片路径
    Cursor
cursor=this.getContentResolver().query(uri,null,selection,null,null);

    if(cursor!=null){
        if(cursor.moveToFirst())

path=cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));
        cursor.close();

    }
    return path;
}

/**
 * 通过 图片的位置, 可以通过 BitmapFactory 解析得到 bitmap, 然后 imageView 就可以直接
setImageBitmap
 * @param imagePath
 */
private void displayImage(String imagePath){
    if(imagePath!=null) {
        Bitmap bitmap= BitmapFactory.decodeFile(imagePath);
        picture.setImageBitmap(bitmap);
    }
    else
        Toast.makeText(this, "failed to get image", Toast.LENGTH_SHORT).show();
}

```

2、Activity 的统一跳转方式

当项目进行团队开发的时候，跳转 activity 是一个非常多见的功能，但不可能每个开发人员都要熟记哪个 activity 的需要传递的 intent 是什么，要传递什么参数，这样一来不利于效率的提高。

于是可以在每个 Activity 都创建可以外部调用的 static 函数，然后在这个函数中暴露 activity 想要获取的参数，在创建。

我的 WriteDiaryActivity 的静态创建函数如下：

把 Intent 的代码放入自己 activity 中

```
public static void startAction(Context context){
    Intent intent=new Intent(context,WriteDiaryActivity.class);
    context.startActivity(intent);
}
```

如果没有其他参数，这样就行了

而要跳转：

```
add_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        WriteDiaryActivity.startAction(MyApplication.getContext());
    }
});
```

这样一句简单的代码

我的 WatchPictureActivity 的静态创建函数如下：

```
public static void startPictureActivity(Context context, String
StringOfUrlOfPic){
    Intent intent=new Intent(context,WatchPictureActivity.class);
    intent.putExtra("StringOfUrl",StringOfUrlOfPic);
    intent.putExtra("way","uri");
}
```

```

        context.startActivity(intent);
    }

    // 数据库中拿网址
    public static void startPictureActivityByInternet(Context context, String url) {
        Intent intent=new Intent(context,WatchPictureActivity.class);
        intent.putExtra("StringOfUrl",url);
        intent.putExtra("way","url");
        context.startActivity(intent);
    }

    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.watchpicturereview_layout);
        image=(ImageView)findViewById(R.id.PictureWatcher);
        image.setOnClickListener(this);

        Intent intent=getIntent();
        String way = intent.getStringExtra("way");
        switch (way){
            case "uri":
                String StringOfUrlOfPic=intent.getStringExtra("StringOfUrl");
                Uri uri = Uri.parse(StringOfUrlOfPic);
                image.setImageURI(uri);
                break;
            case "url":
                String url = intent.getStringExtra("StringOfUrl");
                Glide.with(this).load(url).into(image);
                break;
        }
    }
}

```

activity 跳转的代码如下：

```

viewHolder.picture.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (diary.getPicUrl()!=null)
            WatchPictureActivity.startPictureActivity(MyApplication.getContext(),
            diary.getPicUrl());
    }
}

```



```
}  
});
```

3、创建自己的 Application

创建自己的 Application 能够在任何地方都能获取到 程序的 Context，能够方便地在 adapter，model 或者其他地方使用 Toast 或者其他东西。

还有一个小功能就是能够存放各种全局变量，通过这个自己创建的 MyApplication，能够在任何地方存储变量。

4、ListView 的使用

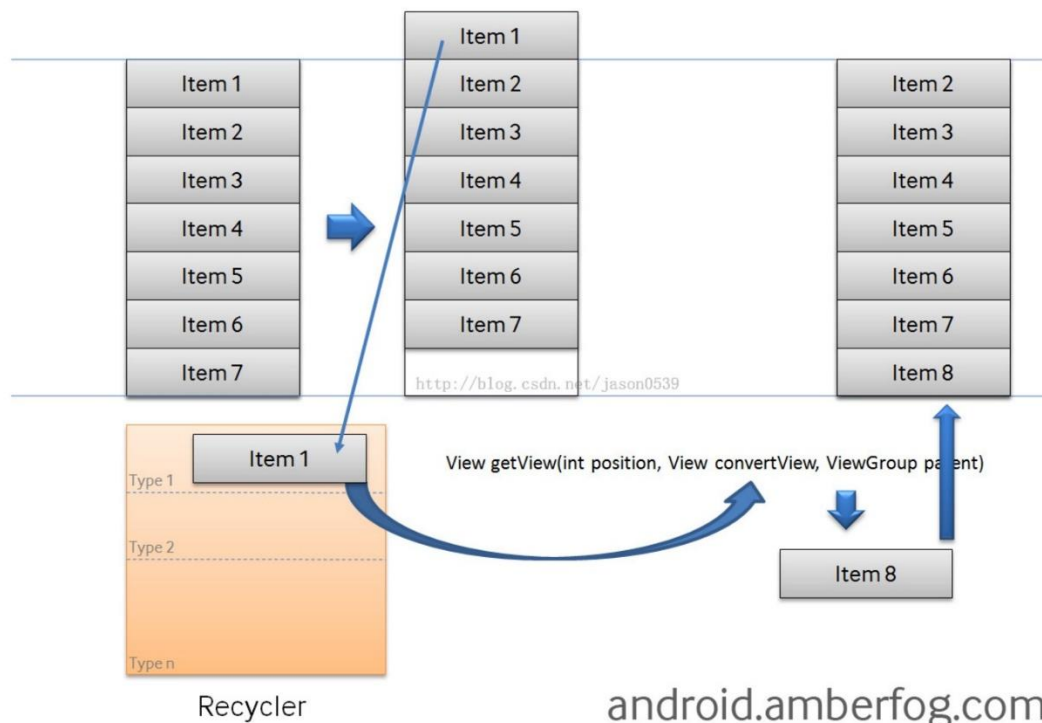
ListView 这个控件常规的方法使用并不难，但是其实在使用过程中，会遇到很多 bug。

例如，我们都知道在 adapter 中的 getView（）函数中对控件初始化，顺便还能加载图片之类的。

A. 通常，getView（）函数为了加快 ListView 的顺畅程度，在加载一个 listItem 项时，会加载若干次 getView（），而如果你的图片在这里加载并放到 ImageView 中，则会进行若干次网络请求，则会太浪费资源了。

B. 为了缓解这一情况，我们通常会使用 viewHolder 和 convertView 的重用机制，此时，当我们使用加载图片时，在快速滑动时，就会造成图片错位的 bug。

图片错位的原理：



如果我们只是简单显示 list 中数据，而没用 convertView 的复用机制和异步操作，就不会产生图片错位；重用 convertView 但没用异步，也不会有错位现象。但我们的项目中 list 一般都会用，不然会很卡。在上图中，我们能看到 listview 中整屏刚好显示 7 个 item，当向下滑动时，显示出 item8，而 item8 是重用的 item1，如果此时异步网络请求 item8 的图片，比 item1 的图片慢，那么 item8 就会显示 item1 的 image。当 item8 下载完成，此时用户向上滑显示 item1 时，又复用了 item8 的 image，这样就导致了图片错位现象(item1 和 item8 是用的同一块内存哦)。

而为了解决这个 bug，必须加上以下的代码。



```

viewHolder.picture.setTag(diary.getPicUrl());
viewHolder.picture.setImageResource(R.mipmap.ic_launcher);

if (viewHolder.picture.getTag() != null &&
viewHolder.picture.getTag().equals(diary.getPicUrl())) {
    viewHolder.picture.setImageURI(Uri.parse(diary.getPicUrl()));
}

```

在 imageView 中加上一个 Tag，并且预设一张图片，然后在进行一个判断，当 tag 和相应的 url 相同时，才会加载图片。

C. ListView 和 ScrollView 的冲突。不知道为什么，当 ListView 和 ScrollView 共同使用时，ListView 只会显示一个单项，这是非常不合理的。（这是我以前遇到过的，突然想其写下的，）

解决方法：

1、定义自己的 MyListView 类，加上如下代码

```

    public MyListView(Context context) {
        super(context);
    }

    public MyListView(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public MyListView(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
    }

    public MyListView(Context context, AttributeSet attrs, int defStyleAttr, int defStyleRes) {
        super(context, attrs, defStyleAttr, defStyleRes);
    }

    @Override
    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {

```

```

int expandSpec = MeasureSpec.makeMeasureSpec(Integer.MAX_VALUE >> 2,
    MeasureSpec.AT_MOST);
super.onMeasure(widthMeasureSpec, expandSpec);
}

```

2、设计一个自己的 UIHelper 类

```

public class UIHelper {

    public static void setListViewHeightBasedOnChildren(ListView listView) {
        ListAdapter listAdapter = listView.getAdapter();
        if (listAdapter == null) {
            // pre-condition
            return;
        }

        int totalHeight = 0;
        for (int i = 0; i < listAdapter.getCount(); i++) {
            View listItem = listAdapter.getView(i, null, listView);
            listItem.measure(0, 0);
            totalHeight += listItem.getMeasuredHeight();
        }

        ViewGroup.LayoutParams params = listView.getLayoutParams();
        params.height = totalHeight + (listView.getDividerHeight() *
(listAdapter.getCount() - 1));
        listView.setLayoutParams(params);
    }
}

```

八、今后对安卓开发的设想

我们编写代码，不仅是为了眼前的功能或者眼前的代码，我们需要为将来要添加的功能去做铺设，也为了使以后的自己能看懂代码，所以用到一定的设计模式十分有必要，不仅要把功能做出来，更要把代码写得好看，扩展性强，易于添加代码，易于维护，才是写代码更高的目标。同时，我也意识到，要真正地弄懂 `android`，要出更好的 `app`，必须要读源码，做更深入的探究，才能更好地做出一个自己的提升。

最后，再次证实了 `cs` 的真理：纸上得来终觉浅，绝知此事要躬行。

各个界面代码就不展示出来，因为都比较简单的设计。👉

Presenter 的具体函数 `LoadDiaryPresenter.java`

```
public class LoadDiaryPresenter {  
  
    private Context mContext;  
    private SQLiteDatabase database;  
    private List<Diary> arr = new ArrayList<>();  
    private DiaryAdapter adapter;
```

```

public LoadDiaryPresenter(Context context)
{
    mContext = context;
    database = MyApplication.getDatabase();
}

public void loadDiary(ListView listView)
{
    Cursor
cursor=database.query(MyApplication.TABLENAME,null,null,null,null,null,null);
    if(cursor!=null){
        while (cursor.moveToNext()) {
            String title = cursor.getString(cursor.getColumnIndex("title"));
            String content = cursor.getString(cursor.getColumnIndex("content"));
            String date = cursor.getString(cursor.getColumnIndex("date"));
            String picUrl = cursor.getString(cursor.getColumnIndex("picUrl"));

            Diary diary = new Diary();
            diary.setContent(content);
            diary.setTitle(title);
            diary.setDate(date);
            diary.setPicUrl(picUrl);
            arr.add(0, diary);
        }
    }
    adapter = new DiaryAdapter(mContext, R.layout.comment_item, arr);
    listView.setAdapter(adapter);
}

public List<Diary> getList(){ return arr;}
}

```

主页 activity 的设计 MainActivity.java

```

package com.example.liangzihong.sqliteandlistview;

import android.database.sqlite.SQLiteDatabase;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;

```

```

import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

import Adapters.Diary;
import Application.MyApplication;
import Presenters.LoadDiaryPresenter;

public class MainActivity extends AppCompatActivity {

    private LoadDiaryPresenter loadDiaryPresenter;
    private ListView myListView;
    private FloatingActionButton add_button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main_layout);
    }

    @Override
    protected void onResume() {
        super.onResume();
        init();
    }

    private void init() {

        loadDiaryPresenter = new LoadDiaryPresenter( this);
        myListView = (ListView)findViewById(R.id.listview);
        add_button = (FloatingActionButton)findViewById(R.id.add_button);
        add_button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                WriteDiaryActivity.startAction(MyApplication.getContext());
            }
        });

        myListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override

```

```

        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            //
        }
    });

    loadDiaryPresenter.loadDiary(myListView);

}
}

```

model 设计 Diary.java

```

package Adapters;

import java.util.Date;

/**
 * Created by Liang Zihong on 2018/12/24.
 */

public class Diary {
    private String date;
    private String title;
    private String content;
    private String picUrl;

    public Diary(){}

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}

```



```

    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content = content;
    }

    public String getPicUrl() {
        return picUrl;
    }

    public void setPicUrl(String picUrl) {
        this.picUrl = picUrl;
    }
}

```

DiaryAdapter 适配器的代码

DiaryAdapter.java

```

package Adapters;

import android.content.Context;
import android.net.Uri;
import android.support.constraint.solver.Cache;
import android.util.Log;
import android.view.LayoutInflater;

```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;

import com.bumptech.glide.Glide;
import com.example.liangzihong.sqliteandlistview.R;
import com.example.liangzihong.sqliteandlistview.WatchPictureActivity;

import java.util.List;

import Application.MyApplication;

/**
 * Created by Liang Zihong on 2018/12/24.
 */

public class DiaryAdapter extends ArrayAdapter<Diary> {
    private int resourceId;
    private Context smallContext;
    private ListView mListView;

    public DiaryAdapter(Context context, int textViewResourceId, List<Diary> list)
    {
        super(context, textViewResourceId, list);
        resourceId=textViewResourceId;
        smallContext = context;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        if (mListView== null)
        {
            mListView = (ListView)parent;
        }

        final Diary diary=getItem(position);
        DiaryAdapter.ViewHolder viewHolder;
        View view;
        if(convertView==null){

```

```

        view=
        LayoutInflater.from(getContext()).inflate(resourceId, parent, false);
        viewHolder=new DiaryAdapter.ViewHolder();
        viewHolder.picture = (ImageView) view.findViewById(R.id.item_picture);
        viewHolder.date = (TextView) view.findViewById(R.id.item_date);
        viewHolder.title = (TextView) view.findViewById(R.id.item_title);
        view.setTag(viewHolder);
    }
    else{
        view=convertView;
        viewHolder=(DiaryAdapter.ViewHolder) view.getTag();
    }

    viewHolder.date.setText(diary.getDate());
    viewHolder.title.setText(diary.getTitle());

    viewHolder.picture.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (diary.getPicUrl()!=null)

WatchPictureActivity.startPictureActivity(MyApplication.getContext(),
diary.getPicUrl());
        }
    });

//    if (diary.getPicUrl()!=null)
viewHolder.picture.setTag(diary.getPicUrl());
viewHolder.picture.setImageResource(R.mipmap.ic_launcher);

    if (viewHolder.picture.getTag() != null &&
viewHolder.picture.getTag().equals(diary.getPicUrl())) {
        viewHolder.picture.setImageURI(Uri.parse(diary.getPicUrl()));
    }

    return view;
}

class ViewHolder
{
    ImageView picture;

```

```
        TextView title;  
        TextView date;  
    }  
}
```

自定义继承 Application 的 MyApplication 类的代码

MyApplication.java

```
package Application;  
  
import android.app.Application;  
import android.content.Context;  
import android.database.sqlite.SQLiteDatabase;  
  
/**  
 * Created by Liang Zihong on 2018/12/24.  
 */  
  
public class MyApplication extends Application {  
    public static Context context;  
    public static SQLiteDatabase database;  
    public static String TABLENAME = "Diarys";  
  
    @Override  
    public void onCreate() {  
  
        context = getApplicationContext();  
        database=openOrCreateDatabase("test1.db",MODE_PRIVATE,null);  
        database.execSQL("create table if not exists " + TABLENAME + " (_id  
integer primary key autoincrement, " +  
        "title text not null, content text not null, date text not null,  
picUrl text )");  
  
        super.onCreate();  
    }  
  
    public static Context getContext() {  
        return context;  
    }  
}
```

```
public static SQLiteDatabase getDatabase() {return database;}  
}
```

写日记的代码 WriteDiaryActivity.java

```
package com.example.liangzihong.sqliteandlistview;  
  
import android.Manifest;  
import android.annotation.TargetApi;  
import android.content.ContentUris;  
import android.content.ContentValues;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.net.Uri;  
import android.os.Build;  
import android.os.Bundle;  
import android.provider.DocumentsContract;  
import android.provider.MediaStore;  
import android.support.annotation.NonNull;  
import android.support.design.widget.TextInputEditText;  
import android.support.v4.content.ContextCompat;  
import android.support.v4.content.FileProvider;  
import android.support.v7.app.AlertDialog;  
import android.support.v7.app.AppCompatActivity;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.Toast;  
  
import java.io.File;  
import java.text.SimpleDateFormat;  
import java.util.Date;
```

```

import Adapters.Diary;
import Application.MyApplication;

/**
 * Created by Liang Zihong on 2018/12/24.
 */

public class WriteDiaryActivity extends AppCompatActivity {

    private TextInputEditText write_title;
    private TextInputEditText write_content;
    private ImageView picture;
    private Button send_button;

    private File file = null;
    private String filePath = null;
    public static final int CHOOSE_FROM_ALBUM=1;
    public static final int TAKE_PHOTO=0;
    private Uri imageUri=null;

    private SQLiteDatabase database;

    public static void startAction(Context context){
        Intent intent=new Intent(context,WriteDiaryActivity.class);
        context.startActivity(intent);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.write_diary_layout);
        database = MyApplication.getDatabase();
        init();
    }

    private void init()
    {
        write_title = (TextInputEditText)findViewById(R.id.write_title);
        write_content = (TextInputEditText)findViewById(R.id.write_content);
        picture = (ImageView) findViewById(R.id.write_picture);
        send_button = (Button)findViewById(R.id.write_send);
    }

```

```

send_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String title = write_title.getText().toString();
        String content = write_content.getText().toString();
        String picUrl = null;
        if (imageUri!=null)
            picUrl = imageUri.toString();

        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy 年 MM
月 dd 日 HH:mm:ss");
        Date dateObject = new Date(System.currentTimeMillis());
        String date = simpleDateFormat.format(dateObject);

        //database.insert()插入功能
        ContentValues values=new ContentValues();
        values.put("title",title);
        values.put("content",content);
        values.put("date",date);
        values.put("picUrl",picUrl);
        database.insert(MyApplication.TABLENAME,null,values);
        values.clear(); //清除
        Toast.makeText(WriteDiaryActivity.this, "上传成功",
Toast.LENGTH_SHORT).show();

        finish();
    }
});

picture.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showNormalDialog();
    }
});
}

```

```

// -----
// -----

// 选择头像的一堆函数

// 打开那个选择框
private void showNormalDialog(){
    final AlertDialog.Builder normalDialog =
        new AlertDialog.Builder(this);
    normalDialog.setTitle("选择头像照片的途径");
    normalDialog.setMessage("打开");
    normalDialog.setPositiveButton("相册",
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                applyPermission();
            }
        });
    normalDialog.setNegativeButton("拍照",
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                goTakingPhoto();
            }
        });
    // 显示
    normalDialog.show();
}

/**
 * 申请权限
 */
private void applyPermission(){
    //如果没有得到 读写 sd 卡的权限
    if(ContextCompat.checkSelfPermission(this,
        Manifest.permission.WRITE_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED)
        requestPermissions(new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},100);
}

```



```

        else {
            openAlbum();
        }
    }

    /**
     * 这里反馈是否得到权限，如果得到权限就继续 openAlbum，得不到就会断掉
     */
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        switch (requestCode){
            case 100:
                //如果得到权限
                if(grantResults.length>0 && grantResults[0]==
PackageManager.PERMISSION_GRANTED)
                    openAlbum();
                else
                    Toast.makeText(this, "你拒绝了权限", Toast.LENGTH_SHORT).show();
            }
        }

    /**
     * 拍照 启动函数
     */
    private void goTakingPhoto(){
        String name="xxx"+" .jpg";

        //创建 图片将要存放的位置
        file=new File("/sdcard/Pictures/",name);
        try{

            file.createNewFile();
            filePath = file.getPath();
        }catch (Exception e){
            e.printStackTrace();
        }

        if(Build.VERSION.SDK_INT>=24)
            imageUri= FileProvider.getUriForFile(this,"sqlite.fileprovider",file);
    }

```

```

else
    imageUri= Uri.fromFile(file);

    //启动相机
    Intent intent=new Intent("android.media.action.IMAGE_CAPTURE");
    intent.putExtra(MediaStore.EXTRA_OUTPUT,imageUri);
    startActivityForResult(intent,TAKE_PHOTO);
    return;
}

/**
 * 打开相册
 */
private void openAlbum(){
    Intent AlbumIntent=new Intent("android.intent.action.GET_CONTENT");
    AlbumIntent.setType("image/*");
    startActivityForResult(AlbumIntent,CHOOSE_FROM_ALBUM);
}

/**
 * 接收 照完相 或者选择完照片 的 回归信号
 * @param requestCode
 * @param resultCode
 * @param data
 */
@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {

    switch(requestCode){
        case TAKE_PHOTO:
            if(resultCode==RESULT_OK){
                //将拍摄的照片显示出来
                try {
                    Bitmap bitmap =
BitmapFactory.decodeStream(this.getContentResolver()
                            .openInputStream(imageUri));
                    picture.setImageBitmap(bitmap);
                }
                catch (Exception e){e.printStackTrace();}
            }
    }
}

```

```

    }
    break;

    case CHOOSE_FROM_ALBUM :
        if(resultCode==RESULT_OK)
            //判断手机的 android 版本，高低版本用不同的函数方法去解决
            if(Build.VERSION.SDK_INT>=19)
                handleImageOnKitKat(data);
            else
                handleImageBeforeKitKat(data);
            break;

        default:
            break;
    }

}

}

@TargetApi(19)
public void handleImageOnKitKat(Intent data){
    String imagePath=null;
    Uri uri=data.getData();
    imageUri = uri;
    if(DocumentsContract.isDocumentUri(this,uri)) {
        //如果是 document 类型的 uri，则通过 document id 处理
        String docId = DocumentsContract.getDocumentId(uri);
        if ("com.android.providers.media.documents".equals(uri.getAuthority()))
        {
            String id = docId.split(":")[1]; //解析出数字格式的 ID
            String selection = MediaStore.Images.Media._ID + "=" + id;
            imagePath = getImagePath
                (MediaStore.Images.Media.EXTERNAL_CONTENT_URI, selection);

            // Toast.makeText(myActivity, "第一个", Toast.LENGTH_SHORT).show();
        } else if
        ("com.android.providers.downloads.documents".equals(uri.getAuthority())) {
            Uri contentUri = ContentUris.withAppendedId(Uri.parse("") +
                "content://downloads/public_downloads"),
            Long.valueOf(docId));
            imagePath = getImagePath(contentUri, null);
            // Toast.makeText(myActivity, "第二个", Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

    }
    else if("content".equalsIgnoreCase(uri.getScheme())) {
        imagePath = getImagePath(uri, null);
//        Toast.makeText(myActivity, "第三个", Toast.LENGTH_SHORT).show();
    }
    else if("file".equalsIgnoreCase(uri.getScheme())) {
        Toast.makeText(this, "第四个", Toast.LENGTH_SHORT).show();
        imagePath = uri.getPath();
    }

    displayImage(imagePath);
}

private void handleImageBeforeKitKat(Intent data){
    Uri uri=data.getData();
    String imagePath=getImagePath(uri,null);
    displayImage(imagePath);

    Toast.makeText(this, "第0个", Toast.LENGTH_SHORT).show();
}

private String getImagePath(Uri uri, String selection){
    String path=null;
    //通过 Uri 和 selection 来获取真实的图片路径
    Cursor
cursor=this.getContentResolver().query(uri,null,selection,null,null);

    if(cursor!=null){
        if(cursor.moveToFirst())

path=cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));
        cursor.close();

    }
    return path;
}

/**
 * 通过 图片的位置, 可以通过 BitmapFactory 解析得到 bitmap, 然后 imageView 就可以直接
setImageBitmap
 * @param imagePath
 */
private void displayImage(String imagePath){

```

```

        if(imagePath!=null) {
            Bitmap bitmap= BitmapFactory.decodeFile(imagePath);
            picture.setImageBitmap(bitmap);
        }
        else
            Toast.makeText(this, "failed to get image", Toast.LENGTH_SHORT).show();
    }
}

```

放大观察图片的 activity 代码 WatchPictureActivity.java

```

package com.example.liangzihong.sqliteandlistview;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.ImageView;

import com.bumptech.glide.Glide;

/**
 * Created by Liang Zihong on 2018/12/24.
 */

public class WatchPictureActivity extends AppCompatActivity implements
View.OnClickListener{
    private ImageView image;
    // 本地相册或者拍照
    public static void startPictureActivity(Context context, String
StringOfUrlOfPic){
        Intent intent=new Intent(context,WatchPictureActivity.class);
        intent.putExtra("StringOfUrl",StringOfUrlOfPic);
        intent.putExtra("way","uri");
        context.startActivity(intent);
    }
}

```

```

}

// 数据库中拿网址
public static void startPictureActivityByInternet(Context context, String url)
{
    Intent intent=new Intent(context,WatchPictureActivity.class);
    intent.putExtra("StringOfUrl",url);
    intent.putExtra("way","url");
    context.startActivity(intent);
}

@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.watchpictureview_layout);
    image=(ImageView)findViewById(R.id.PictureWatcher);
    image.setOnClickListener(this);

    Intent intent=getIntent();
    String way = intent.getStringExtra("way");
    switch (way){
        case "uri":
            String StringOfUrlOfPic=intent.getStringExtra("StringOfUrl");
            Uri uri = Uri.parse(StringOfUrlOfPic);
            image.setImageURI(uri);
            break;
        case "url":
            String url = intent.getStringExtra("StringOfUrl");
            Glide.with(this).load(url).into(image);
            break;
    }

}

@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.PictureWatcher:
            finish();
    }
}

```

```
}  
}
```