

移动互联网技术及应用

大作业报告

题目： 视频播放器(抖音视频解析)

姓名	班级	学号
韩世民	2019211308	2019211496

2022.6

目录

1.	相关技术.....	3
2.	系统功能需求.....	3
3.	系统设计与实现.....	3
4.	系统可能的扩展.....	7
5.	总结体会.....	7

1. 相关技术

- (1) 安卓 UI 编写，布局嵌套、组件属性设置等。
- (2) 使用 Retrofit 进行网络请求，实现了带参数的 GET 请求。
- (3) 使用 GSON 解析 Response 中 JSON 格式的响应
- (4) 使用 JCVideoplayer 实现视频播放器，支持进度条拖动改变进度、支持全屏播放、支持视频缓存、视频封面展示等。
- (5) 使用 getExternalStorageDirectory 查找本地路径，在本地进行视频存储。
- (6) 使用单例模式创建了 DownloadUtil 类，类中使用 OkHttpClient 请求网络资源，使用 FileOutputStream 写入本地文件，使用回调函数实现实时更新下载进度。

2. 系统功能需求

需求 1.许多小白手机用户拿到视频链接不知道如何才能下载到本地进行存储，使用本软件可以直接通过视频链接一键下载到本地，方便快捷。

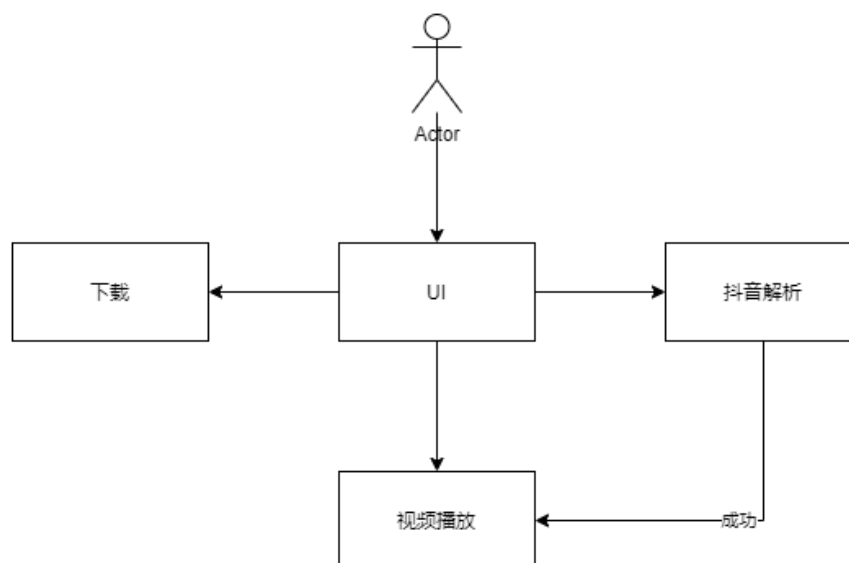
需求 2.看到抖音中好看的视频想要下载到本地，但是无法下载到本地，或者下载到本地会出现水印，影响视频观看，本软件可以通过抖音视频分享链接获取无水印的视频进行下载。

需求 3.部分网络视频以及抖音视频，无法在线全屏观看，通过本软件可以实现全屏观看，提高观影体验。

3. 系统设计与实现

(1) 总体设计

系统主要有四部分组成，分别是 UI、抖音视频解析、视频播放器、视频下载



其中 UI 并没有做过多的渲染,只是使用基础组件,应用课程中学习到的 Activity、布局等方面的知识,构建了如下图所示的界面:



其次是抖音视频解析的实现,抖音视频解析依赖于我自己实现的 api 接口 (<http://api.xcrobot.top/dy/jx/api.php?url=>), 部署在自己的服务器上, 软件中只需要带上用户输入的参数进行网络请求, 网络请求利用课程中学习到的 Retrofit 进行, 收到响应以后利用课程中学到的 GSON 解析 JSON 格式的响应

视频播放器使用开源库实现, 支持视频缓存、全屏播放、进度条拖动等功能, 视频播放器播放的内容可以是用户指定解析的抖音视频, 也可以是用户获得的视频链接。

下载功能被单独封装了一个类, 是通过 OkHttpClient 获取网络数据, 使用 FileOutputStream 写入本地文件, 其中文件路径的获取, 利用了课程中讲解过的本地存储相关的内容。

(2) 系统组成

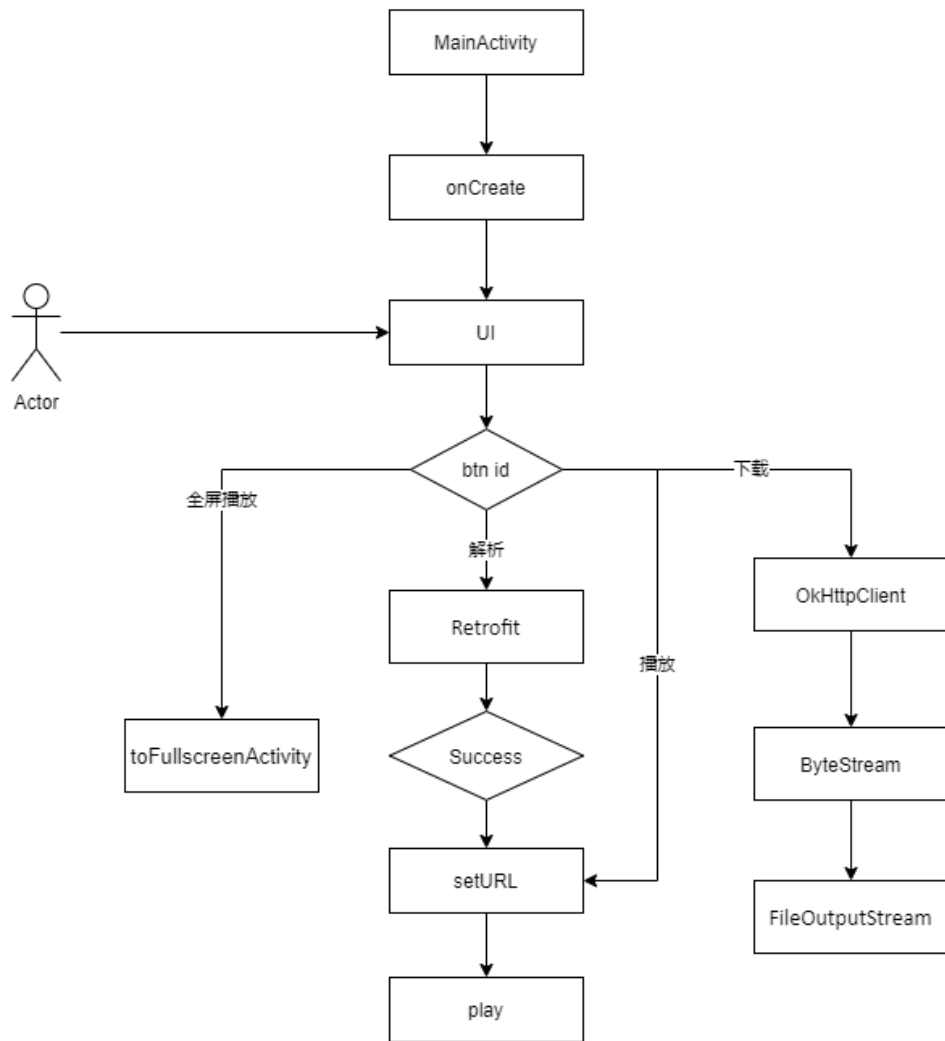
系统主要由 UI 接收用户请求并传递给后端, 根据用户不同的请求作出不同的响应。

如果用户请求解析视频, 则使用 Retrofit 模块, 通过 UI 获得到的用户输入的参数向 API 接口进行 GET 请求, 如果解析成功, 则使用 GSON 模块解析响应, 取出其中的视频链接、标题、封面等参数, 并将这些参数保存到内部的类变量中, 播放器模块通过读取类变量修改相应的参数, 实现抖音视频的去水印解析和播放。

如果用户请求下载视频, 则首先通过 UI 获取下载视频链接编辑框中的内容, 利用 OkHttpClient 请求该链接中的资源, 并将获取到的 response 以 ByteStream 的形式保存, 传递给 FileOutputStream 存储到本地文件当中, 其中文件的路径通过 getExternalStorageDirectory 获得, 文件的标题则使用类变量中的标题参数。

如果用户请求全屏播放, 则调用播放器模块的全屏播放函数, 其中视频的链接、标题、封面等都是通过类变量传递。

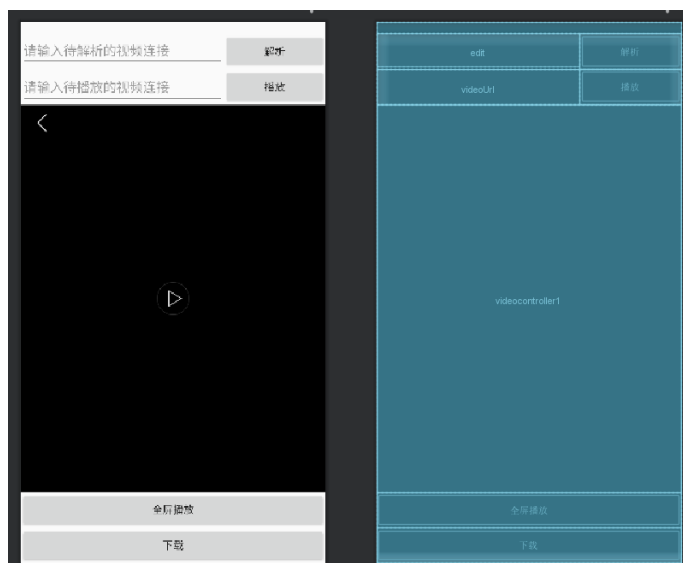
如果用户请求播放, 则首先通过 UI 获取播放视频链接编辑框中的内容, 之后调用播放器模块的播放函数, 其中视频的标题、封面等则是通过类变量传递。



(3) 各模块设计

系统总共有五个主要文件，

其中 UI 主要通过布局文件实现，布局中使用三个线性布局相互嵌套，实现了总体布局为纵向线性，局部布局水平线性分布，布局效果如下图所示：



ApiService.java 是一个接口类，用于通过 Retrofit 模块发送 GET 请求

DownloadUtil.java 是下载类，除了 DownloadUtil 类外还包含一个 OnDownloadListener 接口，用于实时获得下载进度。DownloadUtil 类使用了 OkHttpClient 进行网络请求，使用 FileOutputStream 进行文件写入。

MainActivity.java 主要是界面类，onCreate 函数中加载了界面布局，并设置了按钮的监听函数，也是主要的软件逻辑实现区域，抖音视频的解析、全屏播放、下载、播放视频等功能的实现都是在监听函数中。

urlResponse 函数用于解析接口返回的 JSON 结构数据，其中定义了 urlResponse 类，以便通过 GSON 可以直接解析返回的 JSON 结构数据。

(4) 关键代码

抖音视频解析：

ApiService 的 getVideo 用于发起请求，其中 editText 中的内容是用户输入的抖音视频链接。

在 onResponse 的回调中，使用 GSON 解析 response，拿到了 url、cover 和 title，同时将解析后得到的 url 保存在播放编辑框中，以便后续下载使用。

```
case R.id.search:
    Log.d("tag", "msg: " + editText.getText());
    ApiService.getVideo(editText.getText().toString()).enqueue(new Callback<urlResponse>() {
        @Override
        public void onResponse(Call<urlResponse> call, Response<urlResponse> response) {
            if (response.body() != null) {
                if (response.body().code == 200) {
                    url = response.body().url;
                    videoText.setText(url);
                    cover = response.body().cover;
                    title = response.body().title;
                    videoController1.setUp(url, cover, title);

                    Toast.makeText(context, MainActivity.this, "解析成功", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(context, MainActivity.this, "Error code: " + response.body().code + response.body().msg,
                        Toast.LENGTH_SHORT).show();
                }
            }
        }
        @Override
        public void onFailure(Call<urlResponse> call, Throwable t) {
            Log.d("tag", "retrofit", t.getMessage());
            Toast.makeText(context, MainActivity.this, t.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
    break;
```

下载逻辑代码：

使用 OkHttpClient 请求网络数据，之后进行文件目录是否存在的检验，最后通过 FileOutputStream 将 OkHttpClient 获取到的 byteStream 写入文件

```
public void onResponse(Call call, Response response) throws IOException {
    InputStream is = null;
    byte[] buf = new byte[2048];
    int len = 0;
    FileOutputStream fos = null;
    // 储存下载文件的目录
    File dir = new File(destFileDir);
    if (!dir.exists()) {
        dir.mkdirs();
    }
    File file = new File(dir, destFileName);
    try {
        is = response.body().byteStream();
        long total = response.body().contentLength();
        fos = new FileOutputStream(file);
        long sum = 0;
        while ((len = is.read(buf)) != -1) {
            fos.write(buf, 0, len);
            sum += len;
            int progress = (int) (sum * 1.0f / total * 100);
            // 下载中更新进度条
            listener.onDownloading(progress);
        }
        fos.flush();
        // 下载完成
        listener.onDownloadSuccess(file);
    } catch (Exception e) {
        listener.onDownloadFailed(e);
    } finally {
        if (is != null) is.close();
        if (fos != null) fos.close();
    }
}
```

4. 系统可能的扩展

- (1) 倍速播放
播放器可以实现对视频进行 1.5、2 倍速等模式播放
- (2) 单独下载音频
可以尝试将音频与视频分离，满足用户只下载音乐的需求，比如用户想保存视频中的背景音乐时可能需要这个功能。
- (3) 拖动屏幕调节音量
可以尝试捕获拖动动作并做出响应，通过上下滑动调节媒体音量，这也是大部分播放器都支持的功能。
- (4) 拖动屏幕调节进度
可以尝试捕获拖动动作并做出响应，通过左右滑动调节视频进度，目前只支持拖动进度条改变视频进度，不够方便
- (5) 拖动屏幕调节亮度
可以尝试捕获拖动动作并做出响应，通过上下滑动调节屏幕亮度，这也是大部分播放器都支持的功能。

5. 总结体会

移动互联网技术的发展其实不仅仅体现在移动性方面，诸如手机 app 的很多功能，或许与移动性关系不大，但是却是 PC 端发展很久都很难发展这么快的，这其实是因为手机普及的速度要高于 PC，同时，这也极大地推动了手机 app 的发展。其次 APP 的包管理工具非常方便快捷，使得 APP 的开发难度极大的下降，就比如这次实现的这个软件，就使用了很多非常高效的包，使得 APP 的开发成本极低，例如 GET 请求的发送、JSON 的解析等都非常方便，但是我们作为北邮学子，不应该仅仅停留在会导包会用包，应该花时间去拜读这些包的源码，学习其代码。