

2023年夏季Java小学期大作业

陈奕韬 2022010784

代码结构

代码树如下：

```
main
├── AndroidManifest.xml
├── ic_launcher-playstore.png
├── java
│   └── com.java.chenyitao
│       ├── DetailActivity.java
│       ├── HistoryFragment.java
│       ├── HomeFragment.java
│       ├── MainActivity.java
│       ├── SearchFragment.java
│       ├── StaredActivity.java
│       ├── ViewedActivity.java
│       └── model
│           ├── MyCustomEntity.java
│           ├── NewsSqliteOpenHelper.java
│           ├── MsgType.java
│           └── News.java
└── res
    ├── drawable
    ├── drawable-hdpi
    ├── drawable-mdpi
    ├── drawable-xhdpi
    ├── drawable-xxhdpi
    ├── drawable-xxxhdpi
    ├── layout
    │   ├── activity_detail.xml
    │   ├── activity_main.xml
    │   ├── activity_stared.xml
    │   ├── activity_viewed.xml
    │   ├── fragment_history.xml
    │   ├── fragment_home.xml
    │   ├── fragment_search.xml
    │   └── news_item.xml
    ├── menu
    │   ├── menu_detail.xml
    │   ├── menu_home.xml
    │   ├── menu_main.xml
    │   └── menu_setting.xml
    ├── mipmap-anydpi-v26
    ├── mipmap-hdpi
    ├── mipmap-mdpi
    └── mipmap-xhdpi
```

```
|— mipmap-xxhdpi
|— mipmap-xxxhdpi
|— values
|— values-night
|— xml
```

- `com.java.chenyitao`下为具体的代码实现，其中：
 - `model`下为三个封装类和一个枚举类，
 - `MainActivity.java`为主界面，其中：
 - `HomeFragment.java`为主界面的主页，
 - `SearchFragment.java`为主界面的搜索页，
 - `HistoryFragment.java`为主界面的历史记录页，
 - `DetailActivity.java`为新闻详情页，
 - `StaredActivity.java`为收藏页，
 - `ViewedActivity.java`为浏览历史页。
- `res`下为资源文件，其中：
 - `layout`下为各个界面的布局文件，对应着各个Activity和Fragment，
 - `menu`下为各个界面的菜单文件，对应着ActionBar或者BottomNavigationBar的选项，

具体实现

首先较为粗略地对应大作业要求

功能	实现
程序在测试机上正常运行，测试过程中程序不崩溃	✅ 在虚拟机上正常运行
正确展示新闻列表，点击可进入新闻详情页面	✅ 通过okhttp3完成对新闻内容的爬取 使用RecyclerView以及重写的NewsAdapter在HomeFragment进行展示 点击跳转至DetialActivity
上拉获取更多新闻，下拉刷新最新新闻	✅ 使用SmartRefreshLayout实现上拉刷新、下拉加载 & 上拉下拉增加特效🎁
按关键词、分类和时间搜索新闻，展示搜索结果	✅ 在SearchFragment实现搜索功能, 展示与HomeFragment类似
删除和添加分类(分类需包含接口categories参数支持的各种取值，以及展示所有类别新闻的“全部”分类)	✅ 使用FlycoTabLayout实现顶部导航栏 通过BottomMenu支持添加或删除分类
详情页中显示新闻的来源和时间	✅ 在DetailActivity中展示新闻来源和时间
详情页中正确播放新闻对应的视频	✅ 使用JzvdStd实现视频播放
本地存储看过的新闻，保证离线状况下仍然可以访问	✅ 使用SQLite实现本地存储浏览历史

功能	实现
本地存储历史记录及收藏记录，并可展示历史记录、收藏记录列表	✔ 使用SQLite实现本地存储浏览历史和收藏记录 在ViewedActivity以及StaredActivity展示记录

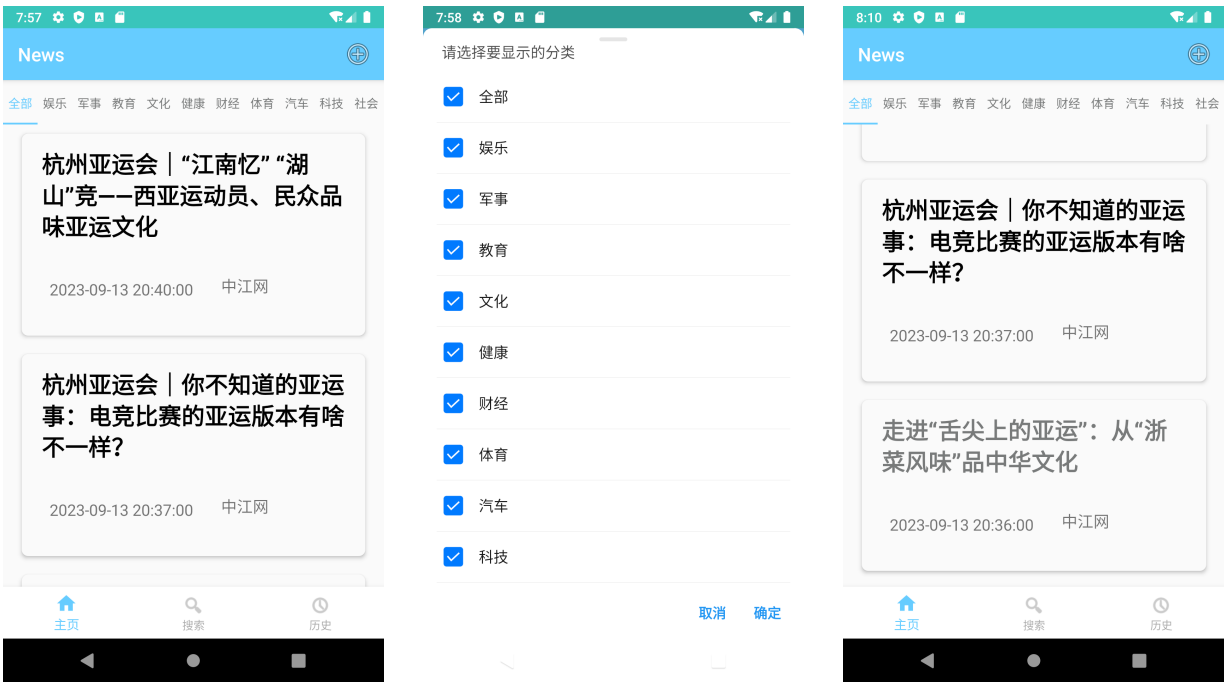
1. 主界面

进入app后，首先进入主界面，主界面对应着MainActivity，主界面包含三个Fragment，分别为主页、搜索页和历史记录页，三个Fragment采用FrameLayout作为布局容器，通过BottomNavigationBar进行导航，并使用FragmentManager进行切换，其中主页为默认选中项。

1.1 主页

主页对应HomeFragment

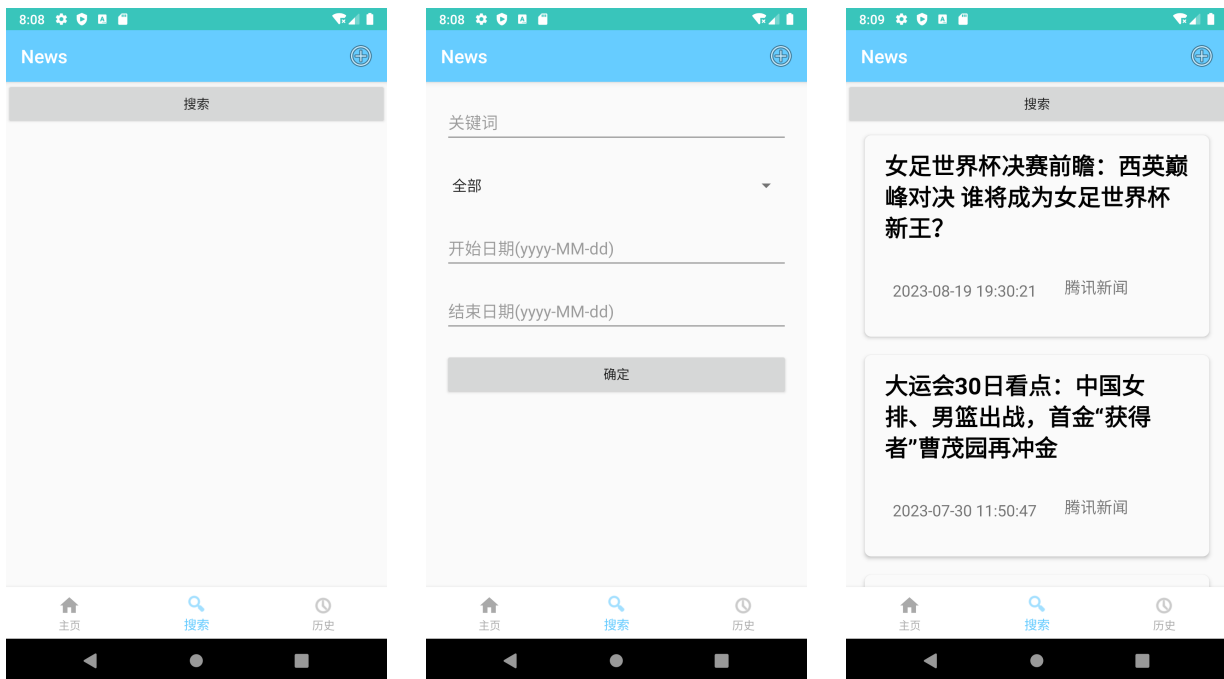
- 网络请求方面，使用okhttp3爬取信息，然后使用fastjson2对相应内容进行解析，获取新闻信息，此外每次网络请求都新建了线程，并重写了handleMessage方法，能够根据网络请求的结果进行相应的处理
- 新闻展示方面，使用CardView来承载每一条新闻，具体资源layout对应news_item.xml，通过Glide来加载图片内容，使用RecyclerView来展示新闻列表，通过NewsAdapter继承RecyclerView.Adapter来实现新闻列表的适配，MyViewHolder继承了RecyclerView.ViewHolder来实现新闻列表的复用
- 上下拉方面，使用SmartRefreshLayout添加OnRefreshListener和OnLoadMoreListener来实现上拉刷新和下拉加载，同时在下拉刷新时会清空原有的新闻列表并爬取最新的新闻，上拉加载时会在原有的新闻列表上添加新的新闻，并使用了SmartRefreshLayout提供的动画效果
- 类别选择方面，使用FlycoTabLayout来实现顶部导航栏，点击相应类别后将会再次以该类别作为参数发起网络请求，通过BottomMenu来实现添加或删除类别，在BottomMenu中添加了多选菜单，可以同时添加或删除多个类别，同时在添加或删除类别后会立即更新顶部导航栏并再次发起网路请求
- 点击进入详情页方面，使用Intent传递新闻的信息，跳转至DetailActivity，并在MainActivity重写了onActivityResult方法，当从DetailActivity返回时，会根据返回的newsID将该新闻的标题变为灰色，不过需要该新闻对应的CardView划出RecyclerView的缓存



1.2 搜索页

搜索页对应SearchFragment

- 网络请求方面，与主页相同，都使用了okhttp3以及fastjson2
- 搜索方面，搜索页初始只有一个“搜索”的按钮，点击后，“搜索”按钮将会消失，然后出现三个EditText，一个Spinner以及一个“确定”的按钮，完成各种搜索条件的输入，点击“确定”后，“搜索”的按钮将会再次出现，其他的组件消失，这是通过设置各组件的Visibility实现的，并且“搜索”按钮会一直悬浮在新闻列表上，此外，搜索结束后将会根据输入的搜索条件发起网络请求，获取搜索结果，然后使用RecyclerView展示搜索结果，搜索结果的展示与主页相同，点击进入详情页方面也与主页相同，不过搜索界面的新闻标题并没有要求看过新闻标题变灰，因此设定其固定为黑色，并且只需要普通的Intent即可



1.3 记录页

记录页对应HistoryFragment

- 记录页有两个大按钮（实际上是CardView），点击后，分别通过Intent跳转至ViewedActivity和StaredActivity



2. 详情页

详情页对应DetailActivity

- 展示内容方面，图片展示同样使用Glide，视频播放使用的是JzvdStd，而其他内容的填充则是通过Intent传递的新闻信息来实现的，总体的滚动效果使用了ScrollView
- 顶部栏方面，使用了ActionBar，其中包含了返回按钮，收藏按钮，点击返回按钮后，会返回上一个界面，点击收藏按钮后，会将该新闻的信息存入数据库，未收藏的新闻收藏按钮显示为只有边框的黄星，而收藏后的新闻收藏按钮显示为实心的黄星，同时收藏按钮的点击事件也会根据新闻是否已经收藏来进行相应的处理



3. 浏览历史页

浏览历史页对应ViewedActivity

- 展示新闻列表方面，与主页相同，使用RecyclerView展示新闻列表，不过此时的新闻列表是从数据库中获取的，因此需要使用SQLite读取本地存储，同时需要重写NewsAdapter来实现新闻列表的适配，MyViewHolder继承了RecyclerView.ViewHolder来实现新闻列表的复用，此外历史记录是按照浏览时间由近及远排序的，并且在历史记录中浏览的新闻同样也会更新浏览时间
- 点击新闻进入详情页的方法与其他一致，且仍可收藏该新闻



4. 收藏页

收藏页对应StaredActivity

- 展示新闻列表方面，与浏览历史页相同，不同的是收藏记录是按照收藏时间由近及远排序的
- 点击新闻进入详情页的方法与其他一致，且可取消收藏该新闻



5.数据库

数据库的实现对应`NewsSqliteOpenHelper`，该类继承自`SQLiteOpenHelper`，并创建了名为`news`的表，使用`newsID`作为主键，并定义了`title`, `date`, `publisher`, `content`, `isStared`, `createTime`, `starTime`这么几列，每列的含义与其名字相同，此外还涉及以下操作：

- 存储历史记录，当用户点进某条新闻的详情页时，会将该新闻的信息存入数据库，其中`createTime`为当前时间，若该新闻已经存在于数据库中，则更新`createTime`，这是在`DetailActivity`中调用`NewsSqliteOpenHelper`实现的
- 存储收藏记录，当用户在详情页点击收藏按钮时，会根据该新闻的`isStar`状态进行相应，若该新闻未收藏，则将`starTime`更新为当前时间，这同样是在`DetailActivity`中调用`NewsSqliteOpenHelper`实现的
- 新闻标题变灰，在展示新闻列表时，每次复用`MyViewHolder`时，都会根据该新闻的`newsID`查询数据库，若该新闻存在于数据库中，则将该新闻的标题变为灰色，这是在`HomeFragment`中调用`NewsSqliteOpenHelper`实现的

6.其他

- 新闻的信息被`News`类所表示，`News`类包含了`newsID`等各种必需的新闻信息
- `MyCustomEntity`类继承了`CustomTabEntity`，是`FlycoTabLayout`展示顶部导航栏所要求的类型，其中包含了`title`、`selectedIcon`以及`unSelectedIcon`三个属性，由于需求较为简单，故只使用了`title`属性
- `MsgType`是一个枚举类，包含了`REFRESH_SUCCESS`, `LOAD_SUCCESS`, `LOAD_FAILURE`, `REFRESH_FAILURE`, `INIT_FAILURE`这么几个值，分别对应着刷新成功、加载成功、加载失败、刷新失败以及初始化失败，这些值在`HomeFragment`中被用作`Message`所携带的信息，用来相应网络请求线程的结果