

北京交通大学

移动应用开发期末项目报告

“简闻” 新闻应用

组号：10

小组成员：

组 长： 杨富全

学 号： 18221213

邮 箱： 18221213@bjtu.edu.cn

北京交通大学

2021 年 6 月

目录

1 概述.....	3
2 背景.....	3
3 项目说明.....	3
3.1 功能模块.....	3
3.2 逻辑流程.....	4
3.3 技术说明.....	5
3.4 代码架构.....	11
4 应用效果.....	13
5 开发过程中遇到的问题.....	14
6 思考及感想.....	16
7 小组分工.....	16
8 参考资料.....	17

1 概述

“简闻”新闻软件是基于安卓平台，采用 JAVA 语言，从网络开放新闻借口获取数据，为了让公众能够随时随地方便且快捷的获取最新的新闻咨询信息，且支持查看浏览记录和收藏列表的安卓手机应用。

2 背景

随着智能手机的普及，越来越多的人习惯在手机上获取新闻消息。而同类新闻软件，往往存在信息繁杂，广告众多，人们的使用体验很差。所以开发出一块简洁美观，功能实用的新闻应用成为我的目标。开发安卓移动应用的技术也比较成熟，可以利用 Retrofit2 和 GSON 控件发送 HTTP 请求从网上相应的 API 借口获得新闻消息，通过安卓本身丰富的控件(如:RecyclerView,CardView,WebView 等等)对这些消息进行解析并展示。考虑到许多用户有喜欢收藏新闻的习惯，所以增加可以查看浏览记录和收藏新闻的功能，此功能可以使用 SQLite 轻量数据库实现。下面从几个方面介绍这款新闻应用。

- 实用性

一款新闻应用的基本功能就是能够随时从网络中获取新闻展示给用户，我开发的这块应用能够获取新闻，并且将新闻的简要以卡片的形式展示给用户，点击新闻卡片，之后能够查看新闻的详细内容。

- 用户体验

我设计的这款应用简洁大方，没有广告烦恼，使用一个按钮便可以实现对应用的全部操作，操作逻辑简单，非常容易上手，用户体验较于同类产品更佳。

3 项目说明

3.1 功能模块

功能主要分为四个模块，如下图所示：

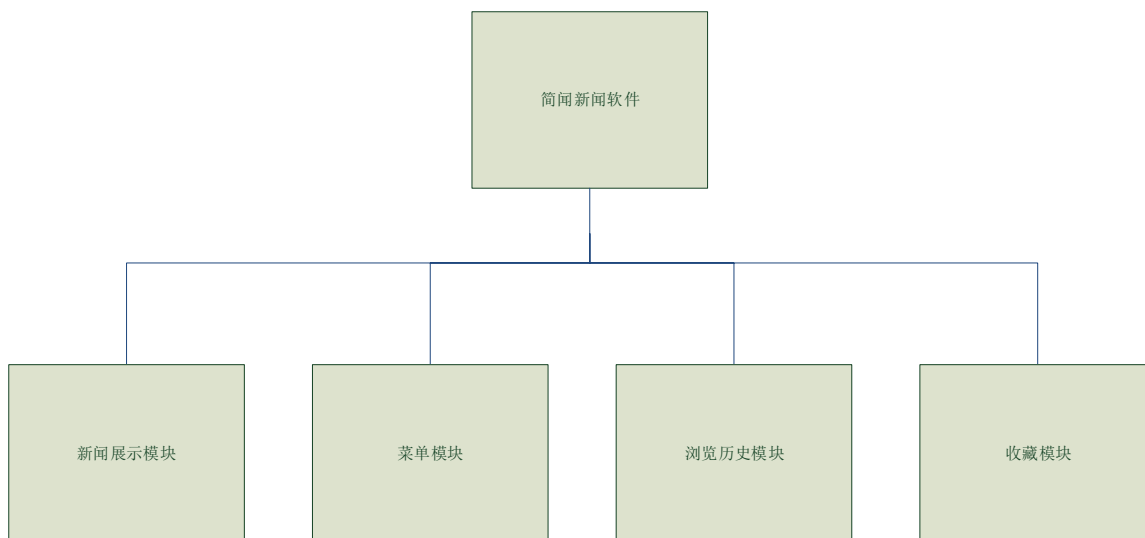


图 3-1 功能模块

新闻展示模块主要由 MainActivity 和 News 两个活动构成，主要功能是展示新闻的简要和详细信息。

菜单模块主要由 BoomButton 组件，出现在每一个 Activity 中，主要功能是点击跳转到不同类型新闻页面或者跳转到浏览历史和收藏页面。

浏览历史模块主要由 ClickedHistory 活动构成，主要功能是展示浏览过新闻的概要信息，可以选择收藏，删除历史信息或者浏览相关历史。

收藏模块主要由 Person 活动构成，主要功能是展示收藏页面，可以选择删除或者浏览相应新闻。

3.2 逻辑流程

如下图所示，为“简闻”应用的使用逻辑。

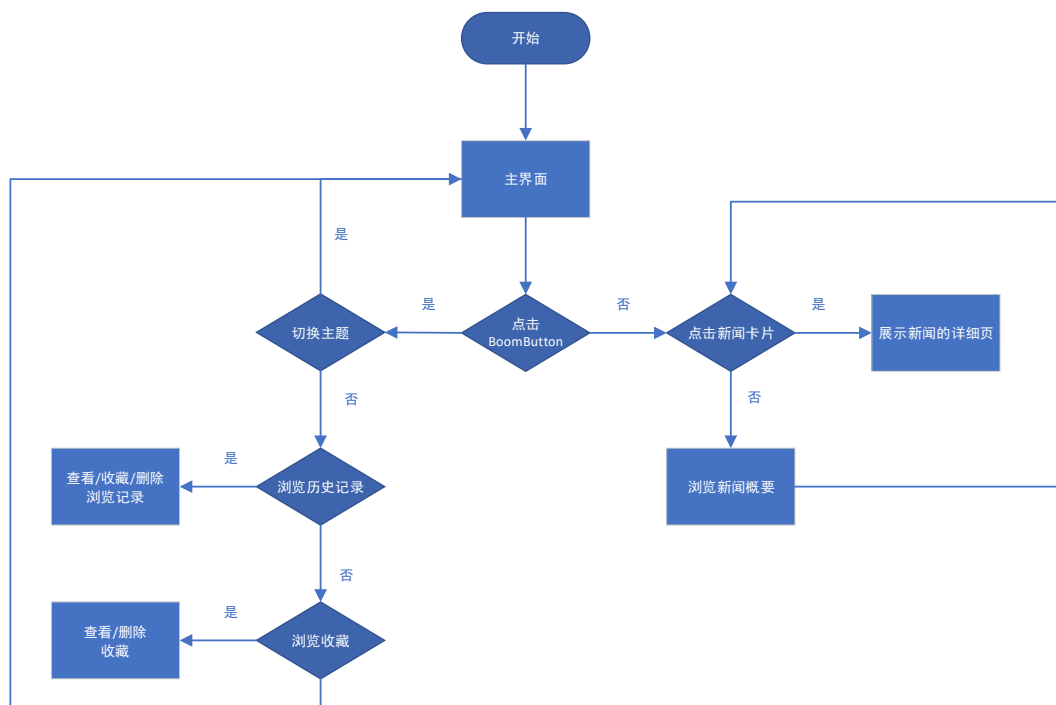


图 3-2 运行逻辑图

在主界面，如果不点击 BoomButton 就一直浏览新闻的概要信息，点击卡片就会跳到新闻的详细页。

如果点击关于新闻种类的按钮，可以更换主页展示新闻的类别。如果点击浏览记录按钮，跳转到浏览记录页面，记录页面对记录提供查看，收藏，删除三种操作。点击收藏按钮，跳转到收藏页面，收藏页面对收藏列表项提供查看，删除操作。

3.3 技术说明

首先为了便于说明技术，所以我将技术模块按照使用目的分为四类：新闻获取，菜单按钮，浏览记录和收藏，新闻展示，每个模块使用的技术如下图所示：

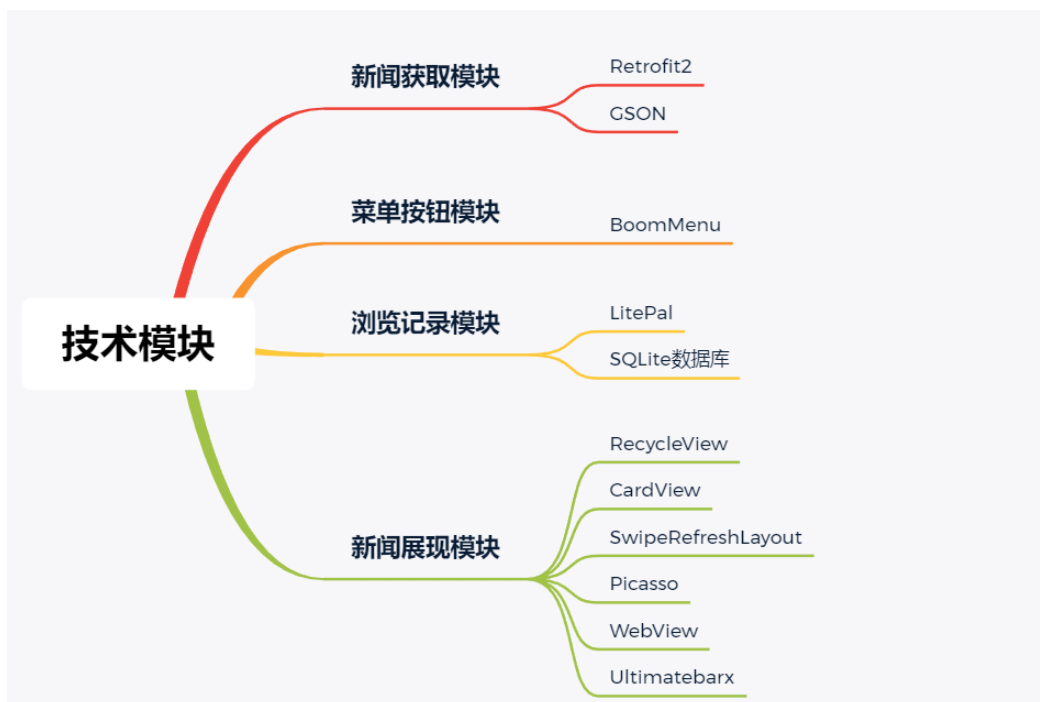


图 3-3 技术模块图

下面我根据模块详细的介绍运用的技术

3.3.1 新闻获取

应用的技术：**Retrofit2,GSON 组件**

新闻应用首先应当获取新闻数据，本应用是通过聚合数据网站中的新闻头条 API 获取新闻数据，主要使用了 Retrofit2 和 GSON 组件实现这一功能。

Retrofit2 是一个网络访问框架，和 OkHttp 同样出自 Square 公司，Retrofit 内部依赖于 OkHttp，但是功能上做了更多的扩展,比如返回结果的转换功能，可以直接对返回数据进行处理。Gson（又称 Google Gson）是 Google 公司发布的一个开放源代码的 Java 库，主要用途为序列化 Java 对象为 JSON 字符串，或反序列化 JSON 字符串成 Java 对象。通过这两个组件可以使得安卓应用发送 HTTP 请求，从 API 获取 JSON 数据后，将其转化为对象，便于后续的操作。

具体的实现方法有如下几步：

1.添加依赖

利用 Gradle 引入依赖

```
//发送网页请求
implementation 'com.squareup.retrofit2:retrofit:2.5.0'
//解析 JSON 数据
implementation 'com.squareup.retrofit2:converter-gson:2.2.0'
```

2.根据 Json 对象设计创建相应的类

由聚合数据返回的 JSON 数据样例可知，应当创建三个类来实现 JSON 的三层嵌套。

3.编写 Api 请求类，确定请求参数，编写请求函数。

4.在主函数中调用请求函数，发送网络请求，转化为相应的对象后，以便进行下一步的操作

3.3.2 菜单按钮

应用的技术：**BoomMenu** 控件

菜单按钮是新闻类别和功能的切换器，BoomMenu 是来自于 Github 上的一个开源控件，其形状类似 FlatButton，但与 FlatButton 不同的是 BoomMenu 点击后会弹出多个选项按钮，弹出按钮的形状，个数和弹出动画都可自由设置，功能十分强大

具体的实现方法有以下几步：

1.添加依赖

```
//爆炸按钮
implementation 'com.nightonke:boommenu:2.1.1'
```

2.在布局文件中添加按钮，并设置其相关属性

```
<com.nightonke.boommenu.BoomMenuButton
    android:id="@+id/bmb"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="50dp"
    app:bmb_normalColor="@color/white"
    app:bmb_autoHide="true"
    app:bmb_dimColor="@color/white"
    app:bmb_highlightedColor="@color/white"/>
```

3.创建 Menu 类定义 BoomMenu 中的菜单项并设置相关方法

```
private static int[] imageResources = new int[]{
    R.drawable.homepage,
    R.drawable.createtask_fill,
    R.drawable.star,
    R.drawable.header,
    R.drawable.nation,
    R.drawable.eco,
    R.drawable.sport,
    R.drawable.war,
    R.drawable.yule
};
```

4.在活动中对填充的按钮个数以及点击响应事件进行设置

```
bmb = (BoomMenuButton) findViewById(R.id.bmb);
assert bmb != null;
bmb.setButtonEnum(ButtonEnum.SimpleCircle);
bmb.setPiecePlaceEnum(PiecePlaceEnum.DOT_9_1);
bmb.setButtonPlaceEnum(ButtonPlaceEnum.SC_9_1);
for (int i = 0; i < bmb.getPiecePlaceEnum().pieceNumber(); i++) {
    addBuilder();
}
```

设置相应的相应事件：

```
bmb.addBuilder(new SimpleCircleButton.Builder()
    .normalImageRes(BuilderManager.getImageResource())
    .listener(new OnBMClickListener() {
        @Override
        public void onBoomButtonClick(int index) {
            switch (index){
                case 0:
                    Log.d(TAG, "onBoomButtonClick: 0");
                    Intent intent = getIntent();
                    intent.putExtra("type", "top");
                    finish();
                    startActivity(intent);
                    break;
            }
        }
    })
```

3.3.3 浏览记录和收藏

应用技术：LitePal,SQLite 数据库

LitePal 是一款开源的 Android 数据库框架，采用了对象关系映射(ORM)的模式，将平时开发时最常用的一些数据库功能进行了封装，使得开发者不用编写一行 SQL 语句就可以完成各种建表、增删改查的操作。使用 LitePal 让记录存储在数据库变得更加容易。

具体的实现方法如下：

1.引入依赖

```
//LitePal  
implementation 'org.litepal.guolindev:core:3.2.3'
```

2.创建映射对象

根据想要存储的新闻信息，创建与之对应的类。

```
public class History extends LitePalSupport {  
    String uniquekey;  
    String title;  
    String url;  
    String author;  
    String date;  
    String category;  
    String thumbnail_pic_s;  
    .....  
}
```

3.创建 xml 参数文件，设置数据库表的映射类名路径

```
<list>  
    <mapping class="com.example.news.litepal.History" ></mapping>  
    <mapping class="com.example.news.litepal.Likes"></mapping>  
</list>
```

这样在活动中就可以利用简单的几行代码实现数据库的增删改查

由于使用逻辑为当用户点击新闻卡片，详细浏览某一新闻，就需要记录在数据库中，所以在主活动的 **CardView** 的点击响应事件中，增加如下几行代码，将点击的新闻相关信息存储在数据库中。

```
History history =new History();  
history.setUrl(item.getUrl());  
history.setAuthor(item.getAuthor_name());  
history.setTitle(item.getTitle());  
history.setUniquekey(item.getUniquekey());  
history.setThumbnail_pic_s(item.getThumbnail_pic_s());  
history.setCategory(item.getCategory());  
history.save();
```

同样的在浏览记录页面和收藏页面，设置相关的按钮的响应事件，实现相关记录的删除和收藏操作

3.3.4 新闻展示

应用的技术:

RecyclerView,CardView,SwipeRefreshLayout,Picasso,WebView,Ultimatebarx 以及基本的展示控件

新闻展示部分主要是 UI 的展示部分，即利用 Android 提供的展示控件实现新闻内容的展示。本应用主框架是 RecyclerView 和 CardView 的嵌套实现可滑动浏览的信息流展示，SwipeRefreshLayout 实现下滑刷新的操作，Picasso 实现对图片 URL 的解析，WebView 实现对网页的 URL 的解析，Ultimatebarx 实现导航栏的透明。

RecyclerView 的实现步骤:

1. 添加依赖

```
//解析图片 URL  
implementation 'com.squareup.picasso:picasso:2.5.0'  
//Material  
implementation 'com.google.android.material:material:1.3.0'  
//Recycleview  
implementation 'androidx.recyclerview:recyclerview:1.2.1'  
//透明导航栏  
implementation 'com.zackratos.ultimatebarx:ultimatebarx:0.6.0'
```

2.设计 RecyclerView 中的单项的布局

利用 CardView 作为框架，将新闻的简要，类别和事件展现在新闻图片之上。

3.编写 RecyclerViewAdapter 类

在类中对单项的布局中的各个控件内容通过 Data 类保存的新闻信息进行设置，如保存的图片 URL 使用 Picasso 进行解析

```
String imageUrl = item.getThumbnail_pic_s();  
Picasso.with(context).load(imageUrl).into(holder.imageView);
```

4.在活动中对 RecyclerView 进行设置，使内容显示在活动中

```
adapter=new Adapter(MainActivity.this, data);  
recyclerView.setAdapter(adapter);
```

下滑刷新的实现主要是使用 SwipeRefreshLayout 实现，实现步骤如下:

1.在布局文件中创建此布局

```
<androidx.swiperefreshlayout.widget.SwipeRefreshLayout>
.....
</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>
```

2.在主函数中设置下拉事件的响应事件

由于下拉需要刷新新闻，所以下拉的响应事件便设置为再次发送网络请求，获取请求，重新设置 RecyclerView 的内容。

```
swipeRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
    @Override
    public void onRefresh() {
        retrieveJson(type,API_KEY);
    }
});
```

WebView 实现对新闻详细页 URL 的解析:

```
WebView webView = findViewById(R.id.webview);
Intent intent =getIntent();
String url = intent.getStringExtra("url");
```

Ultimatebarx 控件实现导航栏的自定义显示:

```
UltimateBarX.with(this)
    .fitWindow(true)
    .light(true)
    .color(Color.TRANSPARENT)
    .drawableRes(R.drawable.bg3)
    .applyStatusBar();
```

Ultimatebarx 控件实现导航栏的自定义显示:

```
UltimateBarX.with(this)
    .fitWindow(true)
    .light(true)
    .color(Color.TRANSPARENT)
    .drawableRes(R.drawable.bg3)
    .applyStatusBar();
```

3.4 代码架构

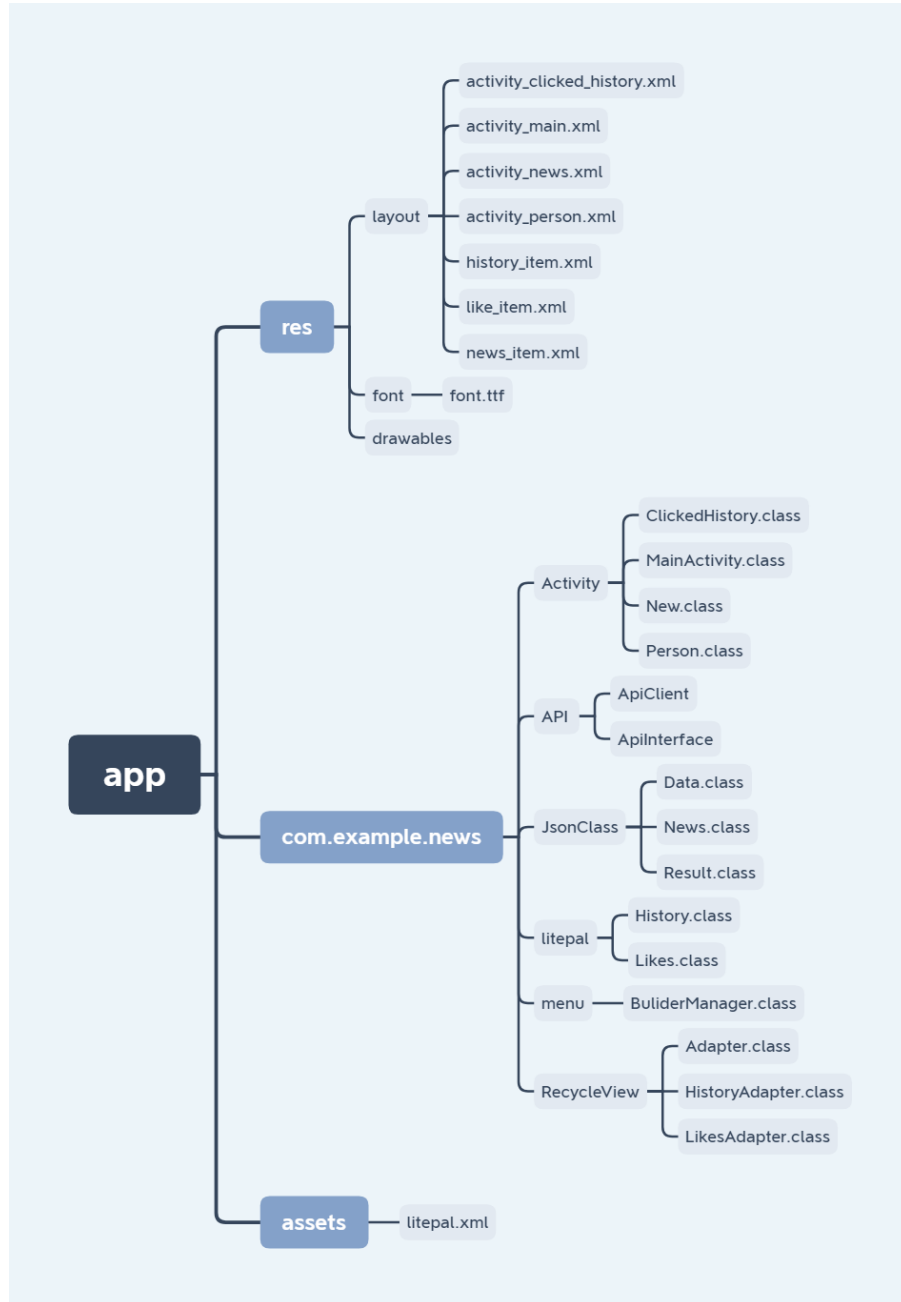


图 3-4 代码架构图

整个代码的架构可以分成三部分，**res** 部分，**news** 部分以及 **assers** 部分。其中 **res** 中为应用中使用的资源包括字体，图片，布局文件，**news** 部分为应用的具体逻辑实现部分，**assets** 部分保存数据库的配置文件。下来着重介绍 **news** 部分中类的作用。

news 按照实现功能不同如图分为六类：

Activity：实现三个主界面界面的主要活动

API：有关网络请求的类和接口

JsonClass：有关 JSON 数据对应的实体类

litepal：有关数据库的映射类

menu：关于 BoomMenu 的填充设置类

RecyclerView：三个 Activity 中 Recycleview 的适配器类

4 应用效果

下面分别为新闻浏览页，新闻详细页，菜单页，浏览记录页，收藏页



图 4-1



图 4-2

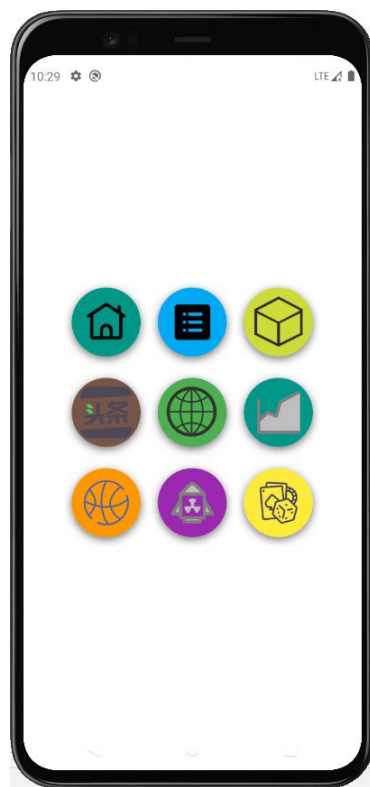


图 4-3



图 4-4



图 4-5

5 开发过程中遇到的问题

问题一：无法切换新闻类别

解决方法：

由于获取新闻类别，需要发送不同的网络请求，即发送网络请求的参数不同。最简单的方法便是设置 7 个 Activity，分别发送 7 种类别的不同请求。但这样一来显得太过复杂，所以想到使用 Intent 传递消息可以改变新闻请求的参数。在点击不同类别的按钮，传递不一样参数，改变主活动中参数的值，就可以达到获得不同类别的新闻。

```
//主活动中当点击到转换新闻类别按钮
Intent intent = getIntent();
intent.putExtra("type", "top");
finish();
startActivity(intent);
```

问题二：UI 不美观

解决方法：

由于 UI 始终达不到理想的状态，所以我选择请教了设计有关专业的朋友，从他那里得到了很多建议，将 UI 设计的还算是简洁大方。



图 5-1 朋友关于 UI 的建议



图 5-2 修改前



图 5-3 修改后

6 思考及感想

通过这次的大作业，我掌握了从网络 API 中获取信息并解析其返回的 JSON 类型的数据的方法，并掌握了 SQLite 的基本操作，掌握一些页面展示的控件的使用方法，以及一定的 UI 设计知识。并将这些知识综合运用起来开发出一款支持选择新闻类别，查看浏览记录，收藏新闻的简易新闻应用。在实践中将学习的知识运用起来，收获非常大。

7 小组分工

组员	学号	负责内容	贡献百分比
杨富全	18221213	全部	100%

8 参考资料

- BoomMenu:<https://github.com/Nightonke/BoomMenu>
- LitePal:<https://github.com/guolindev/LitePal>
- Retrofit2:<https://github.com/zhpanvip/Retrofit2>
- Picasso:<https://github.com/square/picasso>
- Ultimatebarx:<https://github.com/Zackratos/UltimateBarX>