

Java 大作业实验报告

陈旭 2017011341
陈浩展 2017011114

2019 年 9 月

1 项目介绍

1.1 项目目标

- 用 android 编程实现一个有丰富功能的新闻 app;
- 通过项目提升学生的 android 编程能力;
- 通过实现一个项目训练同学开发项目的能力和合作能力;
- 让学生学会如何让自己的程序更有层次感;
- 通过一个有趣的项目提升学生的 Java 的了解和兴趣。

1.2 项目成果

我们开发的软件叫做 Echos, Echos 是一个新闻类 App, 中文翻译为回声新闻。Echos 致力于为用户提供最好的新闻阅览体验。Echos 包括丰富的功能, 不仅可以浏览最新的新闻, 还专门为所有的用户开发了社区, 用户可以在社区中发帖表达自己的看法, 也可以在其中进行评论和讨论; 除了社区功能, Echos 还给用户提供了个性化的订阅内容, 用户可以自由添加自己想看的内容。

2 功能实现

	功能	子功能		子性能	
		介绍	得分	介绍	得分
基础功能	系统支持	要保证程序在安卓机上正常运行，测试过程中程序不崩溃	8	运行流畅不卡顿	2
	页面布局	布局合理，点击处理正确	8	美观，图片布局合理（新闻字太小，扣 1 分）	1
	分类列表	添加和删除操作	4	修改时有动态特效	1
	新闻列表	正确显示新闻列表的消息，布局和展示，点击进入新闻详情页面正确。	8	正确展示图片和视频	2
		实现新闻的本地存储，看过的新闻列表在离线的环境下也可以浏览（看过的页面未标灰，扣 4 分）。	4	本地可存储新闻数量大响应快	2
		上拉获取更多新闻，下拉刷新最新新闻。	4	上拉下拉时添加特效	1
		显示新闻的来源和时间	4	布局合理美观	1
		新闻关键词搜索，历史记录	4	搜索页面合理美观	1
	分享收藏	使用微信、微博等 SDK 分享，新闻详情页面点击分享可以分享到常用的 app，分享内容带有新闻摘要和图片	4	分享页面合理美观	1
		新闻详情页面点击收藏的添加和删除，实现收藏新闻的本地存储。收藏页的正确展示，点击可以进入新闻详情等	8	收藏页面合理美观	2
	新闻推荐	根据用户看过的新闻推荐相关的新闻，参考今日头条等	8	推荐新闻准确	2

表 1: 基础功能介绍

	功能	介绍	申请分数
附加功能	用户登录	用户登录后收藏内容等可以云同步, 更换设备不丢失	8
	论坛功能	登录的用户可以在社区中发帖、对他人的帖子进行评论、对他人的评论进行回复	8
	新闻列表	根据新闻类型不同在新闻列表中显示不同的信息, 丰富多样	2
	夜间模式	可以在夜间模式与日间模式之间进行切换	4
	多语言	软件中部分文字可随系统语言自动变化	1
	软件设置	包括清空缓存、查看软件信息等功能	1
	RSS 订阅	集成 RSS 订阅功能, 可以添加自己喜欢的 RSS 源	4

表 2: 附加功能介绍

3 具体实现的内容

分类	类名	介绍
Activity	MainActivity	程序主界面
	SettingsActivity	设置界面
	UserInfoActivity	用户信息显示界面
	PostActivity	用户发帖记录显示界面
	CommentsActivity	用户评论记录显示界面
	FavoritesActivity	用户收藏显示界面
	HistoryActivity	浏览记录显示界面
	SearchActivity	搜索框与推荐关键词显示界面
	SearchResultActivity	搜索结果显示界面
	LoginActivity	登录界面
	SignupActivity	注册界面
	ViewActivity	新闻内容显示界面
	NewPostActivity	发帖界面
	FeedbackActivity	反馈界面
	ChannelActivity	频道管理界面
	RSSChannelActivity	RSS 频道管理界面
	CommentAreaActivity	帖子评论区界面
Fragment	PageFragment	新闻、社区、订阅分类下的子界面，包含于主界面
	ListFragment	频道下的子界面，包含于 PageFragment
	ImageFragment	显示图片的子界面，用于新闻内容显示界面
	VideoFragment	显示视频的子界面，用于新闻内容显示界面
View	MyListView	自定义列表框
Service	RetrofitService、EchosService	用于发送各种网络请求
Manager	User、HistoryManager 等	管理用户信息、历史记录等
Data	News、Post 等	存储各种信息
Helper	ThemeColors、TabViewHelper 等	帮助管理系统资源
ListItem	ListItemInfo 等	管理列表项的样式
Adapter	ListItemAdapter 等	列表内容适配器
Configure	Configure	软件配置信息类

表 3: 类信息概览

3.1 Articles 文字类的封装

在向服务器申请内容的时候，我们需要特定格式类接受信息，所以我们应对不同的申请内容以及格式定义了不同的类。

3.1.1 News

最基础的新闻类，因为访问新闻 api 中返回的内容很多，所以我们借助了第三方工具 GsonFormat 生成了代码，之后的内容同样都是用了这个工具。

在 News 类中主要的内容以及信息都在 DataBean data 的成员变量中，我们对 api 返回的 json 文件中的所有信息都进行了接收，以便在未来调用的时候更加方便。

这里面在获得 image 和 video 的时候，我们发现不能直接用列表进行获得，虽然有的一些新闻返回的是 image="[]"，而有的新闻直接返回 image=""，在后续的处理中会出现问题，所以只能先存储成 String 之后在调用的时候再进行处理。这个问题我们在处理的时候出了一些问题，浏览一些特定的新闻时候会闪退，debug 了很久才解决掉。

3.1.2 Post

这个类对应的是帖子的内容，返回的所有帖子都在一个定义的子类 DataBean 的列表里面，调用的时候可以通过秩访问里面的内容。

3.1.3 Favourite

这个类对应的是向服务器申请得到的 favourite 内容。与 Post 内容类似，在这里不再赘述。

3.1.4 PostComments

这个类对应的是评论，与上类似。

3.1.5 RSSData

在获得 rss 信息的时候，我们遇到了一些困难，在申请信息的时候我们得到并不是 json 格式的信息而是 xml 格式的信息，所以需要在服务器上进行解析变成 json 格式的文件，我们再用 android 编程中进行获取。所以造成了在获取个别 rss 信息的时候很慢，例如：<https://www.zhihu.com/rss>。在显示 rss 内容的时候有一些内容是 html 格式的信息，又因为本次大作业不能用任何形式的 html 解析器，所以我们只能显示 rss 的连接，然后知道用户点击链接获取详情。

3.1.6 ArticlePack

这个类对应的是一种特殊情况，因为我们的 app 有不同的内容，包括新闻和帖子和 rss 信息，所以我们在加载收藏和历史纪录的时候需要同时显示这三种内容，利用一个统一的 Pack 去封装这些信息，让功能的实现更加方便。

3.2 ListItemInfo 的衍生类介绍

ListItemInfo 的目的是为了显示文章表格而定义的类，但是又因为不同的内容需要不同的显示方式，所以在写了最基本的一些内容之后，我们将 ListItemInfo 衍生成了 4 个类。

3.2.1 PlainListItemInfo

这个容器处理的是只有文字没有内容的文章或者帖子，构造的时候需要标题和副标题以及 data 结构体，显示的时候只有两行文字，一行为标题，一行为副标题。

3.2.2 DefaultListItemInfo

这个容器处理的是有 1 张或者两张图片的新闻页，构造的时候需要标题、副标题以及图片的 url，显示的时候除了两行标题之外，右侧还会有一张缩略图。

3.2.3 ImageListItemInfo

这个容器处理的是有多张图的内容，构造的时候需要一个标题以及三个图片的 url，显示的时候顶上是主标题，下面是三张缩略图。

3.2.4 CommentListItemInfo

这个容器处理的是评论区的显示，构造的时候只需要 PostComments 里面的 DataBean 成员，显示的时候会显示评论内容、发布者以及发布时间。

3.3 Activity 的具体实现

3.3.1 MainActivity

这一部分是对主页面进行管理，包括了对其中的子界面（PageFragment）的管理和左侧侧滑窗口（Drawer）的管理。

3.3.2 ChannelActivity

这一部分是对频道进行管理，频道也就是新闻的种类，通过这个窗口，用户可以自由的选择和去除频道，并且可以进行长按拖拽。

3.3.3 FavouriteActivity

这一部分显示的是我的收藏，收藏的部分在登陆之后才可以加载出来，可以通过在服务器利用 MainActivity 中的 echosServices.getFavourite 放申请“follow.php”，从而得到一个 json 文件，解析出来就得到了收藏的内容，通过定义 favourite 类对于解析下来的内容进行封装。收藏的内容都用 PlainListItemInfo 进行封装，从而在界面中显示出来。通过实现 MyListView.MyListViewPullListener 接口实现下拉加载和上拉刷新。在下拉加载中，通过访问 Favourite 中的 id 进行排序，echosServices.getFourite 的参数为 begin，可以的到序号小于 begin 的所有内容，通过记录这个最后的 id 进行记录与更新。

后面很多的实现部分与这里很相似，都是通过 echosService 向服务器申请信息然后通过一个特定的类进行解析得到信息，这里面写的较为详细，后面类似的 Activity 就不再赘述。同样的关于下拉刷新和上拉加载，实现过程基本相同，后面就不再一一进行具体描述

3.3.4 CommentsActivity

这一部分显示的是我的回帖其中的内容, 同样通过 MainActivity.echosService 访问服务器, 并且解析成 PostComments 类。这里面通过新的一种文字载体 CommentListItemInfo 将显示内容封装起来进行显示。

3.3.5 FeedbackActivity

这一部分显示的是意见反馈, 用户可以通过输入内容发表自己的意见, 开发人员可以接收到用户的意见。在这个界面中, 输入的内容在一个 EditText 里面, 通过点击提交按钮, 将会调用 MainActivity.echosService.addFeedback 进行上传内容。

3.3.6 HistoryActivity

这一部分现实的是我的浏览记录, 通过调用 HistoryManger 的一些静态方法进行获得缓存中存储的新闻。这里面因为浏览的可能是帖子或者新闻, 所以为了方便管理, 定义一个 ArticlePack 的类进行分类, ArticlePack 中有两个成员变量分别是 News.DateBean 和 Post.DataBean 存储数据, 可以根据不同的需求进行不同类别的访问。上拉加载和下拉加载接口实现与 FavouriteActivity 类似, 在此不再赘述。

3.3.7 LoginActivity

通过在 EditText 里面输入帖子的内容和标题, 然后再点击提交之后将发帖的内容构造为 Post.DataBean 内容, 上传。

3.3.8 SearchActivity

这一部分管理的是搜索窗口, 在搜索的时候我们会通过对 keyword 进行计数并且排序, 从而得到最受欢迎的内容提前加载好放在搜索的内容之中, 用户若要搜索这些内容就能够更加快捷的进行搜索。同样的这些内容还会在搜索框中显示, 用户可以在浏览新闻列表的界面看到, 这样可以方便用户了解到当前的热点。

3.3.9 SearchResultActivity

这一部分显示的是搜索内容, 可以根据关键词显示信息。

3.3.10 SettingsActivity

这一部分实现的是设置内容, 因为没有想到很多关于设置的内容, 所以就只实现了一个清楚缓存和一个关于我们的窗口。点击清楚缓存可以清除应用缓存的新闻或者帖子, 释放空间; 点击关于我们会弹出关于项目组成员的介绍。

3.3.11 SignupActivity

这一部分管理的是登陆界面

3.3.12 UserInfoActivity

这一部分显示的是我的信息界面，这一部分相对简单，显示的就是用户的昵称以及是否退出登陆。在点击昵称之后会显示一个 AlerDtDialog 询问是否更改昵称，可以选择在输入内容之后实现更改用户名。

3.3.13 ViewActivity

这一部分管理的是新闻以及帖子的显示内容。

3.4 关于信息存储与传递的实现

对于需要存储与传递的信息，先将其转化为 Json 的格式，传递之后再转换回原本的格式进行处理。

3.5 关于用户功能的实现

这一部分的后端是在我们的个人服务器上使用 php 搭建的，数据库使用的是 MySQL，在应用中与服务器后端进行交互以实现用户登录、论坛讨论等功能。

4 总结和心得

通过本次的项目开发，我们收获到了很多：

- 当开发一个很大的项目的时候，前期的构思与框架是非常重要的，如果框架设计的好，在后面进行实现的过程中就能省很多力，而且写出来的程序会很健壮，不会出现很多错误；
- 在两人协作开发时，对版本控制系统（Git）的使用就显得尤为重要，在使用的过程中虽然遇到了一些困难（例如合并冲突等），但最终还是得以解决。在这个过程中，我们也慢慢熟悉了使用 Git 来进行协作开发，这对我们未来的软件开发有很大的帮助；
- 实现附加功能部分，后端服务的内容也非常重要，在我们当中有一位同学（陈旭）此前做过相关的项目，在这方面比较熟悉，因此在我们的分工中，他就比较偏向这一部分，这样让我们两人在总体效率上都有所提升；
- 项目中用到了大量的类和 XML 文件，一开始比较混乱，到后面我们进行了整理，使得我们在后期的开发更加井井有条，提高了效率；
- 熟悉了 Android Studio 的各种使用方法，提升了我对 Java 和 Android 编程的兴趣，在未来困难会在日常中写一些小的 Android 程序解决遇到的问题。