**学号:2013012450**



**2017 届本科生毕业论文（设计）**

**Android电子书阅读器设计与实现**

**学院(系)：信息工程学院**

**专业年级：软件133**

**学生姓名：张晨**

**指导教师：王湘桃**

**完成日期：2017年5月18日**

**Android电子书阅读器设计与实现**

摘 要

随着Android/iOS智能手机的流行，电子书的市场逐渐越来越大。电子书已经成为了大家休闲，充电的重要方式。因此，提供一个好的阅读APP，保障大家的用户体验，是我这款APP的制作初衷。

这款APP使用MVP（model+view+presenter）开发模式，使用presenter层将model和view层的耦合大大降低。此APP既可以联网下载也可以直接导入本地书籍，为用户提供了多种阅读模式，大大的提高了用户体验。

本文将主要对Android电子书阅读器的需求分析、开发过程进行深入探讨。

关键词：电子书、MVP、用户体验。

**Design and Implementation of Android e - book Reader**

**Abstract**

With the popularity of Android / iOS smart phones, e-book market is getting bigger and bigger. E-book has become an important way for everyone to relax and recharge. Therefore, to provide a good reading APP, to protect everyone's user experience, I am the original intention of this APP production.

This APP uses the MVP (model + view + presenter) development model, using the presenter layer to model and view layer coupling greatly reduced. This APP can be downloaded from the Internet can also be directly into the local books, to provide users with a variety of reading mode, greatly improving the user experience.

This article will mainly on the Android e-book reader needs analysis, development process in-depth discussion.

**Key words：**Keywords: eBook, MVP, user experience.

目录

[第1章 引言 - 6 -](#_Toc482884911)

[1.1 项目背景及意义 - 6 -](#_Toc482884912)

[1.2 主要研究问题 - 7 -](#_Toc482884913)

[1.3 系统实现目标 - 7 -](#_Toc482884914)

[第2章 需求分析 8](#_Toc482884915)

[2.1 功能性需求 8](#_Toc482884916)

[2.1.1系统功能结构 8](#_Toc482884917)

[2.1.2 系统用况说明 9](#_Toc482884918)

[2.2 非功能性需求 10](#_Toc482884919)

[界面需求 10](#_Toc482884920)

[性能需求 10](#_Toc482884921)

[运行环境约束 10](#_Toc482884922)

[第3章 整体架构设计 11](#_Toc482884923)

[3.1 APP接口获取 11](#_Toc482884924)

[3.2 APP图片资源文件获取 12](#_Toc482884925)

[3.3 APP系统架构设计 12](#_Toc482884926)

[3.1.1架构设计——MVP架构 12](#_Toc482884927)

[3.1.2 系统框架选择 14](#_Toc482884928)

[第4章 具体模块实现 14](#_Toc482884929)

[4.1 基础模块实现 14](#_Toc482884930)

[4.1.1 BaseActivity.java实现 15](#_Toc482884931)

[4.1.2 BaseContract.java实现 15](#_Toc482884932)

[4.1.3 BaseReadView.java实现 16](#_Toc482884933)

[4.2 网络加载模块实现 18](#_Toc482884934)

[4.2.1 相关JavaBean定义 18](#_Toc482884935)

[4.2.2 下载信息bean定义 19](#_Toc482884936)

[4.2.3 自定义网络拦截器 19](#_Toc482884937)

[4.2.4 网络接口数据获取 20](#_Toc482884938)

[4.3 阅读模块实现 27](#_Toc482884939)

[4.4 登录模块实现 29](#_Toc482884940)

[4.5 自定义view布局实现 31](#_Toc482884941)

[第5章 APP测试和评价 32](#_Toc482884942)

[5.1 测试环境 32](#_Toc482884943)

[5.2 功能测试 32](#_Toc482884944)

[5.3 UI测试 32](#_Toc482884945)

[5.4 单元测试 32](#_Toc482884946)

[第6章 总结 32](#_Toc482884947)

[珍重申明 33](#_Toc482884948)

# 引言

## 项目背景及意义

近几年，我国的手机普及率逐年增高，利用手机上网的人也越来越多。数据显示，截止2010 年12 月底，我国网民规模达到4.57亿；手机网民规模达到3.03 亿，手机网民在总体网民中的比例从2009 年的60.8%提高到了66.2%。正是由于这一庞大数量手机网民的存在，使手机电子书作为一种新的出版产品，在如火如荼地发展。

对于手机电子书，目前还没有严谨的概念。一般认为，手机电子书就是通过手机阅读的电子书。   
    本文所谈的手机电子书主要是指第三种。   
    （一）手机本身的特点限制   
    由于手机的低分辨率，上面呈现的字体较小，在狭窄的手机屏幕上阅读不是很方便，特别是对一些中老年人群来说。   
    同时，手机电子书每页的内容有限，读者需要不停地翻页才能获得更多内容。以上这些缺点是由手机固有的特点决定的。   
    但是随着人们阅读习惯的改变和手机技术的发展，以上缺点也将得到改善。   
    （二）手机电子书内容粗糙   
    目前我国的手机电子书的发展正处于初级阶段，在内容上还存在一些问题，这主要表现在两方面。一是手机电子书在内容上多以玄幻、言情、武侠、都市、青春类居多，这些内容更像一种快餐，更有不少的内容以欲望的描写来博得读者眼球，这些浅阅读的内容加速文化垃圾的累加，不利于青少年的健康成长，不利于整个社会文化的进步；二是审核不严格，有的甚至没有经过审核就直接面对读者，以至于手机电子书中错字别字比比皆是，内容粗糙。   
    （三）版权问题   
    和传统出版一样，手机电子书的发展也面临着版权问题。   
    在搜索引擎里输入“手机电子书下载”，我们可以轻而易举地找到很多家相关网站，有些手机电子书可以完全免费下载，有的是消费积分才能下载。但无论哪种方式，手机电子书作者并不能从中受益，作者的合法权益得不到保障。日益严重的盗版，严重影响到创作者和出版商的积极性，从而威胁到手机阅读的质量。   
    （四）盈利模式   
    目前的手机电子书的盈利模式基本上是下载付费模式，这一盈利模式几乎完全由电信运营商控制，出版单位处于弱势地位。尽管出版单位在内容制作上付出大量的精力和成本，但获利微小。据了解，现在下载一本手机电子书通常为两元，个别的为三元，出版社与电信运营商从中分成，而出版社的利润最多只能拿到四成，有的甚至更少，这严重挫伤了出版人的积极性，也阻碍整个手机电子书产业的发展。因此，手机电子书的出版需要一个能够实现利益平衡和持续稳定的盈利模式。

## 主要研究问题

此APP旨在为Android智能手机用户提供一个优秀的电子书阅读器。提高阅读的质量和效率。

此APP主要研究的问题是，本地图书的导入加载和阅读，网络数据的稳定传输，并将网络数据保存本地。以及最关键的，正确的加载epub格式的书籍，并提供一个交互良好的界面。

## 系统实现目标

Android电子书阅读器可以提供一个优秀的手机端电子书阅读软件，大大的提高了阅读的效率和质量。以下我从用户的角度阐述该APP实现的功能。

1. 作为普通用户，可以在线搜索书籍，并且根据自己喜好，选择是否留存。如果留存，即将网络数据保存在本地，可以在首页得到书籍的最新动态。
2. 作为书籍的已阅读者，可以对某本书籍发起评论。并且可以与其他评论发起沟通交流。
3. 如果数据库没有想要的书籍，可以在论坛中发起讨论，通过设置链接，可以得到有效的帮助。大大扩展了APP的社交性。

# 需求分析

需求分析就是开发人员要准确地理解用户的要求，对用户的需求进行细致的调查分析，然后将其转化为完整的需求定义，再由需求定义转换到相应的需求规格说明的过程。

在软件项目的开发过程中，需求分析是关乎软件项目开发成功与失败的重要因素。有资料显示，在软件项目中，返工所造成的开支占了总开支的很大一部分比重，然而，导致返工最重要的因素就是需求分析不明确。所以，软件项目的需求分析是软件设计与实现的基础，对整个软件项目来说也是至关重要的。如果能采用一些技术来科学地进行项目需求分析，则能避免可能导致需求分析失败的情况，圆满的完成软件项目的需求分析任务，并且为后续软件开发打下坚实的基础。(孟亚辉 2009)

需求分析的重要性对于软件项目开发的重要性不言而喻，所以做好需求分析是软件开发的第一步，本章主要从功能需求、非功能需求两个方面来论述系统的需求。

## 功能性需求

系统功能性需求是指系统必须提供的服务，应列举出所开发的软件在职能上应做什么，即所有的功能。本节主要介绍系统要实现的功能，使用系统功能结构、用况图和活动图来论述系统的功能性需求。

### 2.1.1系统功能结构

简化的系统功能结构如图2-1所示，系统的功能模块分成以下几个模块：登陆模块，推荐模块，阅读模块，评论模块，发现搜索模块。

登陆模块主要通过QQ，微信，微博第三方的认证登陆系统APP。并可以修改个人信息。

推荐模块主要通过接口获取网络数据，按照用户性别得到推荐的书籍，并将得到的书籍信息添加到推荐主页中。通过长按等操作，可以完成删除，缓存，置顶，收藏等功能。

阅读模块主要是阅读书籍，并可以设置字体大小，屏幕亮度等。并可以通过手指阅读上一页（下一页）

评论模块主要是通过接口获取网络数据，并加载到评论列表中。

发现搜索模块主要是通过关键字搜索，搜索模板书籍，并将搜索记录保留。

### 2.1.2 系统用况说明

图2-1 系统功能结构图

（1）普通游客用况说明如图2-2用况图所示：

如下为对用况“选择性别”的描述。

“晨阅电子书阅读器”APP启动：

弹出对话框，选择游客性别。

若通过，游客选择性别：

通过“追书神器”接口，获得推荐书籍，并将得到书籍信息加载到推荐页面。

选择完成后，进入推荐主页面。

若用户长按弹出相关操作。

若游客选择置顶操作；

系统将选择中的书籍在列表中置顶。

若游客选择“书籍详情”；

系统跳入书籍详情页面。

若游客选择“缓存全本”；

系统使用retrofit框架联网下载缓存本书，将本书缓存至手机本地文件夹中。

若游客选择“删除”；

系统将选中书籍从列表中删除。

若游客选择“批量操作”；

显示批量操作对话框，游客通过勾选相关书籍完成多选，并完成操作。

游客手指滑动进入“社区”页面；

点击“综合讨论区”；

APP进入“综合讨论区”页面。

点击“书评区”；

APP进入“书评区”页面。

点击“书荒互助区”；

APP进入“书荒互助区”页面。

点击“女生区”；

APP进入“女生区”页面。

点击“原创区”；

APP进入“原创区”页面。

游客滑动进入“发现”版块；

点击“排行榜”；

APP进入“排行榜”页面。

点击“主题书单”

APP进入“主题书单”页面。

点击“分类”

APP进入“分类”页面。

## 非功能性需求

由于考虑到APP的运行效率，用户体验等要求，这里对系统的一些非功能性需求提出一些明确要求。

### 界面需求

手机APP交给用户使用，首当其冲的就是界面，要求界面美观大方，界面配色明亮合适。界面设计主题要满足Google提出的material design 要求。因为Android系统版本众多，碎片化严重，必须在界面设计上，考虑到版本适配问题。

### 性能需求

APP安装包大小不得超过20M，考虑到手机可能处于移动弱网情况下，必须可以实时监测网络环境，并在弱网条件下可以稳定下载缓存书籍。

### 运行环境约束

开发环境：Windows7/Windows10

开发工具：Android Studio 2.3

模拟器：Genymotion

运行操作系统：Android 4.0+

开发语言：Java

# 整体架构设计

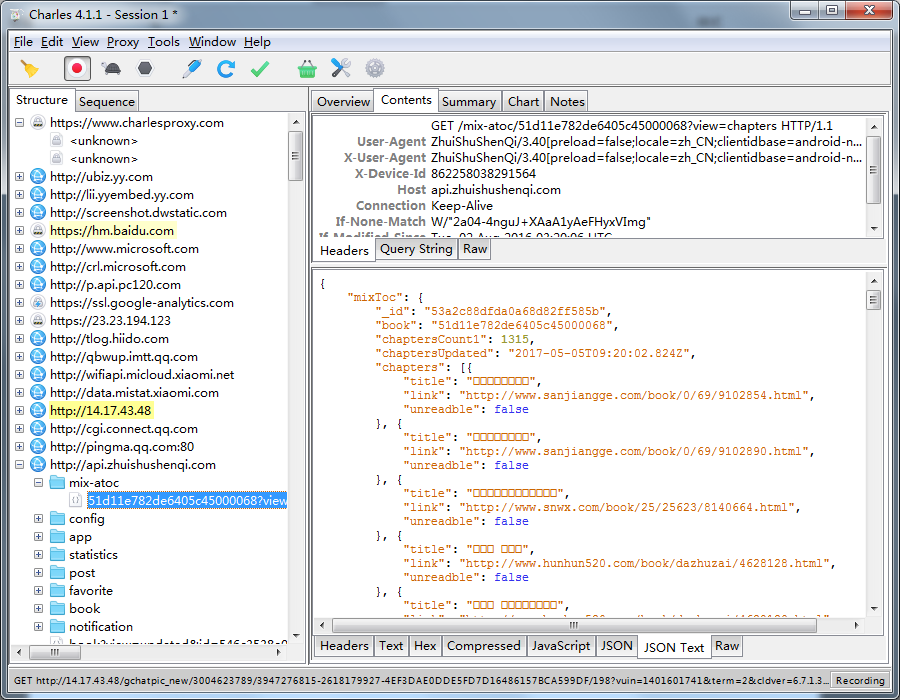
本章节主要介绍“晨阅电子书阅读器”整体架构设计内容。其中包括：APP接口获取，界面资源准备，APP系统架构设计，自定义布局设计五个部分。以下将会逐一介绍相关内容。

## APP接口获取

市场上主流电子书阅读器都提供了网上书城的功能。因时间技术有限，只有获取第三方的API调用。

通过技术调研，选择使用Charles软件进行抓包破解。

市场主流电子书APP为“掌阅”，通过抓包分析，其API使用了MAC加密，无法顺利获取api。通过技术调研，目标APP定为“追书神器”。

通过Windows平台下使用Charles，数据源使用小米5测试手机，抓包获取“追书神器”API。如图3-1，图3-2为软件运行截图。

通过抓包分析，得到以下API接口：

## APP图片资源文件获取

使用Android Killer对“追书神器”APP进行反编译，得到图片资源。如图3-3所示。

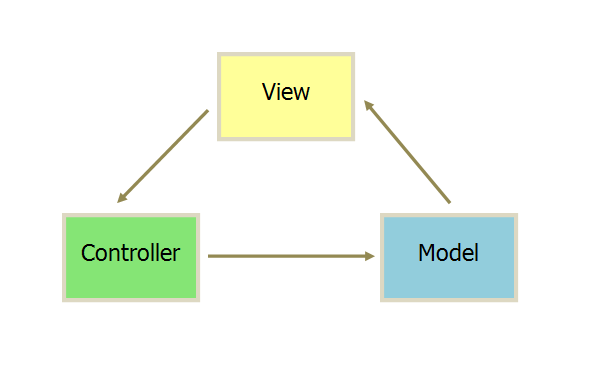
## APP系统架构设计

此APP采用模块化开发流程，整个APP坚持“高内聚，低耦合”的开发理念。各个模块之间相互独立，降低耦合，大大提高后期的扩展性和可维护性。为了APP的后期的维护和使用，在开发上面，我们选择使用了一些成熟的框架用于简便开发。一下，我将从架构设计，框架包装使用等各个方面介绍。

### 3.1.1架构设计——MVP架构

目前安卓市场主流的开发架构分成MVC、MVP、MVVM等多种架构。

MVC：常用于java后台开发（ssh）开发。MVC为model（模型）-view（界面）-controller（控制器）简称。Android发展之初，大量java从业人员转行，使得MVC模式沿用至今。

在Android开发中，MVC结构清晰明朗，但是缺点明显——耦合度过高。使用MVC会使view层和controller层高度耦合，一般耦合于Activity或Fragment界面中。使得界面代码量大，结构冗余，不便于日后的扩展和管理。

MVP：为MVC的后继发展模式。MVP为model（模型）-view（界面）-presenter（主持关系）简称。

MVP基本理念为“面向接口编程”，抛弃MVC之前的“面向对象编程”理念，使用接口（Interface）连接各层，大大的降低了耦合度，佐以泛型，可以实现很高的扩展性。MVP架构中，Presenter层中，保留view层和model层的接口引用。使得view层和model层直接脱离耦合，大大降低的了耦合度。但是MVP缺点明显——使得代码量增多。

MVVM架构，由Google于2012年提出。基于Data-Binding技术。因本项目不涉及，不过多介绍。

### 3.1.2 系统框架选择

（1）网络框架——Retrofit 2.0+

此项目选择Retrofit作为网络框架。Retrofit为Square公司开发的开源高效率的网络框架。内部封装Okhttp网络请求库，将实现的细节和协议全部封装。不需要人为的去写put、get、post等请求，封装后，使用注解就可以完成get，put请求。

Retrofit默认支持RxJava，通过使用RxJava，完成链式请求，代码“一气呵成”，大大减少了代码量，使用RxJava的特定的运算符，可以轻松实现多线程等功能。

（2）图片加载框架——Glide

Glide由Google在谷歌开发者大会上推出的一种开源图片加载框架。Glide和Picasso非常类似，直接使用静态方法加载图片，图片使用URL唯一标识，可以是本地图片地址，亦可以为网络图片地址。

（3）事件传递库——EventBus 3.0

Android中需要处理很多事件，点击事件，长按事件，返回事件等等。这些系统支持的事件，Android系统已经写成了接口用于事件的传递。但是实际开发中，还需要自定义各种事件。官方推荐使用接口传递事件事件开发，但是实际开发中，使用接口会使Activity或Fragment实现很多接口，重现很多方法，使得代码极其冗余不堪。实际开发中，常用EventBus作为事件传递。

（4）观察事件库——RxJava

RxJava专注于异步编程可观察数据。内部使用观察者模式，迭代器模式处理实时数据。使用 Observable作为被观察者，使用Observer作为观察者，利用 Operators 来熟练操作它们。

# 具体模块实现

## 基础模块实现

Android系统的APP几乎全部以Activity或者Fragment的形式展示。每个Activity或者Fragment都会处理相同的事情。如果每一个界面分开来写，将会增大代码量，关键会使代码冗余，可读性下降。我们可以选择让所有Activity、Fragment继承于同一个BaseActivity（BaseFragment）。

除了Activity和Fragment，因为此APP采用MVP架构，每一个Activity（Fragment）将作为view层，像数据库操作，网络操作等将会作为model层，中间需要Presenter层连接，为了减少相同的代码量，我们也需要BasePresenter，BaseContract等。

### 4.1.1 BaseActivity.java实现

此APP界面设计严格符合Google的material design设计理念，所以基类选择android.support.v7.app下的AppCompatActivity。

Android界面初始化由onCreate()方法开始，完成初始化等功能，所以重写onCreate()方法。

@Override

protected void onCreate(Bundle savedInstanceState) {

……

}

在onCreate()方法中，我们需要完成初始化等功能。因为该类为基类，定义为abstract类。所以初始化控件，初始化数据等具体的实现，留给具体的Activity实现。

public abstract int getLayoutId();

protected abstract void setupActivityComponent(AppComponent appComponent);

public abstract void initToolBar();

public abstract void initDatas();

除了上述的初始化内容外，我们还需要做界面准备内容。众所周知，Android版本很多，碎片化严重，所以适配各个版本就显得十分必要。

protected void androidVersion19and20() {

……

}

}

此方法用于专门适配sdk版本为19和20的手机，如果版本为19或者20，人为的透明状态栏，实现各个版本界面之间的协调一致。

### 4.1.2 BaseContract.java实现

Contract类主要就是用来“绑定”view层和presenter层的内容。所以为了便于扩展，我们将Contract写成了接口（interface），内部再定义BasePresenter和BaseView两个内部接口。两个接口中，主要的方法就是绑定view（Activity/Fragment）和解绑定的方法。

public interface BaseContract {

interface BasePresenter<T> {

void attachView(T view);

void detachView();

}

interface BaseView {

void showError();

void complete();

}

}

### 4.1.3 BaseReadView.java实现

作为一个电子书阅读器APP，最重要的功能自然是阅读。那么阅读界面的美观与否将会直接关系到整个APP的用户体验。所以，我们通过自定义view的形式来加载书籍，阅读。

Android自定义view是个非常复杂的功能，可以分成如下的大致几个步骤：

（1）编写style.xml属性文件。

（2）自定义一个java类，继承于View类。

（3）在初始化中，通过特定的方法得到xml文件中定义的属性，完成初始化。

（4）重写onDraw()，onLayout()，onMeasure()方法，完成测量，布局，绘制过程。

因为此BaseView.java为基础模块，所以出于扩展性考虑，我们定义为继承于View的抽象类。由具体实现类去继承实现。

在构造方法中，我们需要完成一些基本的初始化内容。

public BaseReadView(Context context, String bookId, List<chapters> chaptersList,

OnStateChangeListener listener) {

……

init(mTheme);

}

在构造方法中，最重要的就是init(int theme)方法。此方法用于完成初始化内容。

public synchronized void init(int theme) {

……// 自动跳转到上次阅读位置

……// 输出日志，便于以后调试

……// 如果跳转位置为0，捕获异常。重置标记位。

}

如上面所写（具体实现代码，因篇幅问题，会包含附录中），初始化操作中，先自动跳转到上次阅读位置，如果得到上次的阅读位置为0，捕捉出现的异常，输出日志，重新定义标记位。

除了初始化以外，还需要处理手势事件。重写onTouchEvent(MotionEvent e)，在方法中判断手势，并根据手势进行特定操作。

@Override

public boolean onTouchEvent(MotionEvent e) {

switch (e.getAction()) {

case MotionEvent.ACTION\_DOWN:

……//手势为按下操作时，触发事件。

case MotionEvent.ACTION\_MOVE:

……

postInvalidate();//手势为拖动操作时，触发事件。

case MotionEvent.ACTION\_UP:

……//手势为抬起操作时，触发事件。

case MotionEvent.ACTION\_CANCEL:

……//手势为取消操作时，触发事件。

}

return true;

}

如上所示，Android手势分成，按下，移动，抬起等几种，通过switch判断来判断事件类型，触发事件，完成操作。最后return true表示将此事件处理完毕，不会继续往下传递。

在上述的移动事件中，调用了一个postInvalidate()方法。系统执行时，会自动回调onDraw(Canvas canvas)方法进行view的重新绘制，所以我们需要重写onDraw(Canvas canvas)方法。

@Override

protected void onDraw (Canvas canvas) {

calcPoints ();//计算按下滑动的出发点。

drawCurrentPage (canvas);//绘制当前页面。

drawNextPageWithShadow (canvas);//绘制下一页，并且绘制边角阴影。

drawCurrentPageShadow (canvas);//绘制当前页面的边角阴影。

drawCurrentBackArea (canvas);//绘制边角黑色背景。

}

以上的五个方法全部定义为抽象方法，具体实现在具体控件类中实现。

除了初始化，事件处理外，还有一些“基础操作”定义，比如：前一页，后一页，跳转至制定页面等。因为篇幅问题就不一一叙述了，具体参考附件代码。

## 网络加载模块实现

作为一个电子书阅读器，联网功能显得非常重要，所以网络模块显得非常重要。一下将会详细介绍网络模块的实现。

### 4.2.1 相关JavaBean定义

联网搜索下载书籍，数据流全部以json流的方式解析，所以要自定义很多特定的JavaBean对象。

书籍列表（书籍ID，书籍title，书籍作者，书籍简介，书籍推荐性别，书籍封面地址，书籍章节数）

书单列表详情（列表ID，更新日期，列表title，书籍作者，推荐性别，书籍创作时间，书籍收藏数，书籍分享链接，书籍评论列表）

书籍详情（书籍ID，书籍作者，书籍封面，书籍创作日期，书籍收藏数，书籍章节数，社区帖子数，书籍推荐性别，书籍类型，最新更新日期，书籍字数）

章节详情（章节ID，章节title，章节链接，章节可读性标志位，章节创作时间，章节更新时间，章节简介）

分类列表详情（书籍list，列表ID，列表title，列表书籍数，列表分享链接，列表收藏数）

评论列表（列表ID，列表内容，评论列表，评论书籍作者，回复列表）

讨论列表（列表ID，列表title，讨论列表，讨论类型，更新日期，讨论数，列表点赞数）

讨论详情（讨论ID，讨论title，讨论内容，讨论作者，讨论类型，讨论优先级，目前状态，分享链接，讨论回复数）

书荒区帖子详情（帖子ID，帖子类型，帖子发起作者，帖子内容，帖子状态，帖子发起日期，帖子评论数，帖子分享链接）

书荒区帖子列表（列表ID，列表title，列表作者，当前状态，更新日期，帖子评论数，帖子推荐性别）

### 4.2.2 下载信息bean定义

除了上述的自定义JavaBean外，因为需要联网下载，所以我们还需要自定义下载相关的Bean对象。

下载标志位（目前下载章节，章节title，开始下载的章节ID，结束下载的章节ID）

下载提示消息（下载bookID，下载书籍信息，下载完成标志位）

### 4.2.3 自定义网络拦截器

此APP选择的网络框架为Retrofit，我们需要配置一个拦截器用以网络请求拦截。Okhttp提供了一个Interceptor接口，我们需要自定义一个拦截器实现该接口，重写其中的方法。

public interface Interceptor {

Response intercept(Chain chain) throws IOException;

interface Chain {

Request request();

Response proceed(Request request) throws IOException;

Connection connection();

}

}

如上代码所示，我们主要要重写intercept(Chain chain)方法。

@Override

public Response intercept(Chain chain) throws IOException {

Level = this.level;

Request = chain.request();//得到请求链中的请求

Connection = chain.connection();//得到connection对象

Protocol = connection != null ? connection.protocol() : Protocol.HTTP\_1\_1;//得到协议

boolean logBody = level == Level.BODY;//得到body标志位

boolean logHeaders = logBody || level == Level.HEADERS;//得到请求头的标志位

……//对请求进行判断，并进行拦截操作。  
 logger.log(message);//将拦截信息通过日志进行打印，便于日后调试

return response;

}

如上代码所示，自定义拦截器，主要重写intercept方法，得到connection，request，protocol等对象。通过得到的对象，继而得到请求的详情，判断其请求的合理性，得到错误日志，打印日志，最后将response对象返回。

### 4.2.4 网络接口数据获取

（1）获取推荐书籍列表：

请求地址：<http://api.zhuishushenqi.com/book/recommend>

请求类型：GET

请求参数：用户性别（String gender）

返回值：

"books" : [ {

"\_id" : "5533e64fbbbdb15d2606e9fa",

"title" : "爆宠狂妻：神医五小姐",

"author" : "扇骨木",

"shortIntro" : "她是将军府的五小姐，却是东辰国第一废物，花痴成性，因为追求男子，被跟班失手打死；她是天之娇女，却被害身亡，惨遭家族灭门，从此背负血海深仇。 当天才穿越到废物的身...",

"cover" : "/agent/http://img1.chuangshi.qq.com/upload/cover/20150327/cb\_5514c56b06c3c.jpg",

"hasCp" : true,

"latelyFollower" : 27706,

"retentionRatio" : 50.97,

"updated" : "2017-05-17T04:04:29.276Z",

"chaptersCount" : 2061,

"lastChapter" : "第2070章"

}, {

"\_id" : "565d37853c75b7744a83d1d7",

"title" : "纸婚厚爱：总裁轻装上阵",

"author" : "苏安好",

"shortIntro" : "席公子有三好:爱慕暖央，宠慕暖央，调戏慕暖央。 世人却只知其一，不知其二。 可只有慕暖央知道他骨子里透着怎样的坏，说好的只是造个小的，他把大的也收买了是几个意思...",

"cover" : "/agent/http://img1.chuangshi.qq.com/upload/cover/20151103/cb\_5638cc5e8d3f3.jpg",

"hasCp" : true,

"latelyFollower" : 23998,

"retentionRatio" : 32.42,

"updated" : "2016-11-02T12:54:53.788Z",

"chaptersCount" : 1207,

"lastChapter" : "第1207章 ：大结局"

},{

"\_id" : "56e4ca70bbd8ab8f07f6bcb9",

"title" : "萌宝快递：拐个妈咪送爹地",

"author" : "密云不雨",

"shortIntro" : "她是走投无路的落魄千金，他是高高在上的商业大亨。 黑暗之中，他无情的说，你不过是我生孩子的工具。 结果，她的身影却夜夜入梦。 五年后，天降萌娃。 “女人我要包养...",

"cover" : "/cover/147869274176430",

"hasCp" : true,

"latelyFollower" : 22167,

"retentionRatio" : 49.17,

"updated" : "2017-05-16T20:04:28.610Z",

"chaptersCount" : 2578,

"lastChapter" : "第2578章 原来他的初恋女友是这样的"

}

}

（2）得到书籍的章节信息

请求地址：[http://api.zhuishushenqi.com/mix-toc/{bookId}](http://api.zhuishushenqi.com/mix-toc/%7bbookId%7d)

请求方式：GET

请求参数：书籍ID（String bookId）

返回值：

{

"mixToc" : {

"\_id" : "583011c58094b1e228165dd0",

"book" : "583011ba771130ed756b9d07",

"chaptersCount1" : 127,

"chaptersUpdated" : "2017-05-16T17:04:53.881Z",

"chapters" : [ {

"title" : "001：河间匪徒",

"link" : "http://www.luoqiu.com/read/196037/28080150.html",

"unreadble" : false

}, {

"title" : "002：落难贵女",

"link" : "http://www.luoqiu.com/read/196037/28080153.html",

"unreadble" : false

},{

"title" : "003：宫斗直播系统",

"link" : "http://www.luoqiu.com/read/196037/28080155.html",

"unreadble" : false

}, {

"title" : "004：身陷匪窝",

"link" : "http://www.luoqiu.com/read/196037/28166303.html",

"unreadble" : false

},{

"title" : "005：铁口直断（一）",

"link" : "http://www.luoqiu.com/read/196037/28344002.html",

"unreadble" : false

}, {

"title" : "006：铁口直断（二）",

"link" : "http://www.luoqiu.com/read/196037/28344004.html",

"unreadble" : false

},{

"title" : "007：铁口直断（三）",

"link" : "http://www.luoqiu.com/read/196037/28386288.html",

"unreadble" : false

}, {

"title" : "008：铁口直断（四）",

"link" : "http://www.luoqiu.com/read/196037/28485660.html",

"unreadble" : false

},{

"title" : "009：铁口直断（五）",

"link" : "http://www.luoqiu.com/read/196037/28558285.html",

"unreadble" : false

}, {

"title" : "010：铁口直断（六）",

"link" : "http://www.luoqiu.com/read/196037/28568926.html",

"unreadble" : false

}

……

}

（3）根据作者姓名得到所出版书籍

请求地址：http://api.zhuishushenqi.com/book/accurate-search

请求方式：GET

请求参数：作者姓名（String author）

返回值：

{

"books" : [ {

"\_id" : "56928442c49f3bce42b7f521",

"title" : "龙王传说",

"author" : "唐家三少",

"shortIntro" : "伴随着魂导科技的进步，斗罗大陆上的人类征服了海洋，又发现了两片大陆。魂兽也随着人类魂师的猎杀无度走向灭亡，沉睡无数年的魂兽之王在星斗大森林最后的净土苏醒，它要带...",

"cover" : "/agent/http://image.cmfu.com/books/3681560/3681560.jpg",

"cat" : "玄幻",

"site" : "qidian",

"majorCate" : "玄幻",

"minorCate" : "异界大陆",

"banned" : 0,

"latelyFollower" : 160299,

"followerCount" : 0,

"retentionRatio" : 67.88,

"lastChapter" : "第1053章 千伬"

}, {

"\_id" : "523a89f3d87e739d3a005f30",

"author" : "唐家三少",

"cover" : "/agent/http://image.cmfu.com/books/2517792/2517792.jpg",

"shortIntro" : "大陆传奇，一战成名；凤凰圣女，风火流星神界刀法；双升融合，金阳蓝月，雷霆之濒，这里没有魔法，没有斗气，没有武术，却有武魂㿂唐门创立万年之后的斗罗大陆上，唐门式微...",

"site" : "zhuishuvip",

"title" : "斗罗大陆II绝世唐门",

"cat" : "玄幻",

"majorCate" : "玄幻",

"minorCate" : "异界大陆",

"banned" : 0,

"latelyFollower" : 17173,

"followerCount" : 5805,

"retentionRatio" : 51.4,

"lastChapter" : "第六百二十二竿 大结屿"

}, {

"\_id" : "50befdea2097935131000001",

"author" : "唐家三少",

"cover" : "/agent/http://image.cmfu.com/books/1436015/1436015.jpg",

"shortIntro" : "国际巅峰调酒大师、品酒大师，有酒神之称的李解冻，为了品尝出土的汉代美酒迌醉死于而立之年 当他丿觉醒来时，却发现自己来到了一个名叫五行大陆的世界，变成了丿个叫偿...",

"title" : "酒神",

"site" : "zhuishuvip",

"cat" : "奇幻",

"majorCate" : "奇幻",

"minorCate" : "魔法校园",

"banned" : 0,

"latelyFollower" : 3714,

"followerCount" : 5973,

"retentionRatio" : 49.55,

"lastChapter" : "第六百八十八竿 大结屿！我就是你的烈焰啊！"

}

}

**4.2.5 自定义缓存机制实现**

作为一款有效的电子书阅读器，除了可以联网下载书籍外，还需要提供可靠的缓存。否则用户每次使用都需要联网，这对用户而言将带来很大的经济负担，大大的影响用户体验。所以我们必须自定义缓存机制

因为目前的手机的存储空间动辄上百G，所以我们选择将下载的书籍以文件的形式缓存至本地存储中，以便用户断网情况下可以随时阅读书籍。

Android系统提供了两种缓存机制——LruCache/DiskLruCache，分别为内存缓存机制和磁盘缓存机制。出于实际需求开发，决定采用文件缓存机制进行缓存。

实际开发中，自定义缓存机制类似于SharedPreferences，但是比SharedPreferences功能强大很多。不仅基础数据类型，像JsonObject，JsonArray，Bitmap等对象，通过流化操作再通过io数据流写入文件中。

**4.2.6 多线程控制机制实现**

Android将线程分成主线程（UI线程）和工作线程。UI线程完成界面绘制更新等与界面相关的操作；而工作线程主要处理耗时操作，比如数据库访问，网络请求等耗时操作，操作完成后将结果传递给UI线程，通知UI线程更新界面。

由上所述，网络的多线程操作就显得极为重要。一般Android开发中，可以使用Java基本线程操作，线程池AsyncTask，Handler等多种方式完成多线程操作。但是基于上述的几种方案会使代码堆积，不利于调试和扩展。

在第三章已经叙述，此APP开发中使用了事件观察库——RxJava。在RxJava中通过操作符可以实现各种功能，其中就包括了多线程。RxJava中提供了subscribeOn和observeOn两个操作符。

.subscribeOn(Schedulers.*io*())  
.observeOn(AndroidSchedulers.*mainThread*())

通过subscribeOn和observeOn两个操作符，RxJava将会在io线程（工作线程）中处理耗时操作，在mainThread（UI线程）中“观察”结果，最后通过subscribe操作符进行处理。

**4.2.7 网络错误处理**

因为此APP为移动端使用。移动端设备体积小，便携性高，所以广受市场喜爱。但是移动端设备地域广，网络情况复杂。所以如何在弱网情况下提供服务就十分有必要。除了网络环境外，还有可能出现服务器问题，如何可以第一时间在设备上检测出来，并提供切实有效的解决方案显得尤为重要。所以对于网络错误的处理尤为关键。

Android实际开发中，使用的主流网络框架比如volley，Okhttp，Retrofit等都提供了网络错误处理的功能。

本APP采用的为Retrofit框架，并在实现中融入了RxJava。RxJava链式反应的最后一步将使用subscribe处理结果。使用subscribe时，需要传入一个Observer对象。

Observer对象实际为一个自定义的接口（interface），内部定义了三个方法。

public interface Observer<T> {void onCompleted();  
void onError(Throwable e);void onNext(T t);  
}

从方法名中可以看出，需要在onError方法中处理异常信息。在实际开发当中，onError方法中，可以通过日志系统输出异常的信息，也可以通过第三方的统计sdk（如友盟sdk）进行统计。

## 阅读模块实现

作为一个电子书阅读器APP，最重要的功能自然是阅读，所以阅读模块显得极其重要，这节将会着重介绍一下阅读模块的设计和实现。

**4.3.1 自定义阅读view的实现**

在第三章基础架构中，提到了BaseReadView的实现，并阐述了自定义view的基本流程。但是BaseReadView仅仅为一个抽象类，具体方法要在具体的实现类中完成。

项目的难点在于书籍的绘制，要可以准确的绘制到界面上。这里，采用的重写onDraw(Canvas canvas)的方式实现。

public synchronized void onDraw(Canvas canvas) {

if (mLines.size() == 0) {//得到需要绘制书籍的字数值

//如果为空表示书籍看完，重新更新目前标志位。

}

if (mLines.size() > 0) {//表示需要绘制图书

if (mBookPage != null) {

……//表示背景图不为空，绘制背景。

} else {

……//如果为空，直接绘制纯色

}

……// 绘制标题

// 绘制阅读页面文字

for (String line : mLines) {//开启for循环，逐行绘制书籍内容

if (line.endsWith("@")) {

//如果以“@”号结尾，表示一句话结束。

} else {

//绘制需要的内容

}

y += mFontSize;

//更新高度

}

// 绘制提示内容

if (batteryBitmap != null) {

//提示背景不为空，绘制提示内容

}

//绘制内容

}

}

**4.3.2 滑动阴影以及手势的绘制**

为了模拟翻书的效果，市场主流的阅读APP都会有随着手势模拟翻书的效果，这个实现的难点在于随之手指的触摸，获取阴影曲线，并绘制出阴影后面的渐变阴影。

Android系统中提供了可以绘制渐变的类GradientDrawable /ShapeDrawable，通过此可以完成渐变阴影的绘制。使用GradientDrawable时，需要提供两个变量，一个是渐变的方向，另一个是颜色数组，表示渐变初始颜色和结束颜色。

上述第二个问题得到了解决，问题集中到如何根据手指得到曲线，用来模拟翻书的效果。

\mathbf{B}(t) = (1 - t)^{2}\mathbf{P}_0 + 2t(1 - t)\mathbf{P}_1 + t^{2}\mathbf{P}_2 \mbox{ , } t \in [0,1]这里，我采用了贝塞尔曲线进行绘制。贝塞尔曲线是计算机图形学中非常重要的参数曲线。由1962年法国数学家提出。我们在这里采用二阶贝塞尔曲线，二阶贝塞尔曲线由给定的定点P0、P1、P2的函数B（t）追踪。

因为手指翻书，上下各需要一段曲线，我们以手机按下的点作为顶点，算出两个之前的斜率，通过斜率算出两个直线的交点。根据斜率生成带曲率的三角型路径。然后生成底部半圆的path，通过贝塞尔公式算出右侧月半圆的path，下一步就是解析path之间的折叠区域，算出折叠区域和替补区域，最后合并月半圆区域。最后根据剪裁区填充画布。

**4.3.3 字体大小调节**

一款实用的阅读APP，必须可以手动调节字体大小，此APP我使用SharePerference进行存储字体大小，点击调节字体大小的按钮时，从配置中读取已存储的字体大小。通过calFont(int process)方法计算字体大小。

private void calcFontSize(int progress) {

// progress range 1 - 10

if (progress >= 0 && progress <= 10) {

ScreenUtil.dpToPxInt(12 + 1.7f \* progress);

}

}

其中涉及到了一个方法dpToPxInt()，此方法为工具方法，是将dp转化为px的一个工具方法，实现也很简单，就是dp \* density，density为屏幕密度，可以通过系统方法获取。

**4.3.4屏幕亮度以及自动亮度开启**

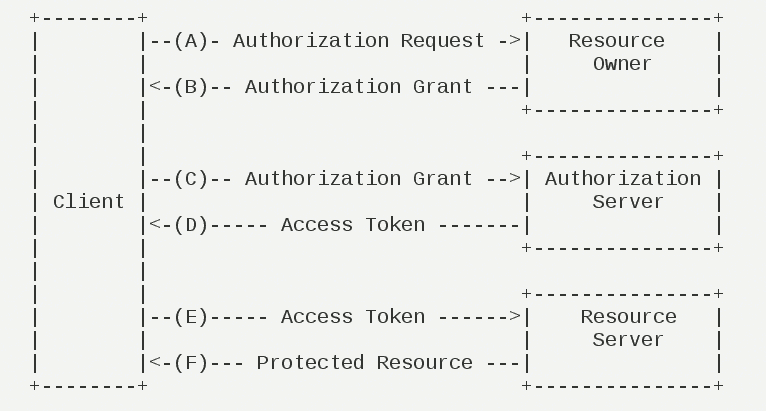
与上面调节字体类似，使用SharePerference进行全局控制。这里需要自定义一个工具类，专门用来调节屏幕相关设置的工具类。系统提供了系统变量可以获取自动亮度System.SCREEN\_BRIGHTNESS\_MODE，判断是否为自动亮度后，使用System类的putInt方法设置屏幕亮度。

## 登录模块实现

几乎所有APP都会含有登录注册模块，登录的用户有别于普通游客，可以获取更多的服务，比如云存储，这样可以随时随地更新自己的书架。这个功能将会极大的提高用户体验，也可以分享链接，将自己认为好的书籍或者评论通过第三方分享出去。此APP采用的登录方式为第三方登录，即使用qq，微信，微博账户登录。使用第三方登录时，需要接入新浪，腾讯的第三方的sdk。以下将在授权认证方式上介绍。

**4.4.1 OAuth2.0认证介绍**

OAuth是一个关于授权（authorization）的开放网络标准，在全世界得到广泛应用，目前的版本是2.0版。Oauth认证是在客户端和服务提供商之间，添加一个“授权层”。这样下来，用户不能直接登录服务端，必须通过“授权层”进行授权。“客户端”登录时拿到授权层给予的令牌（token），这个令牌含有所属的权限范围和有效期。所以，一段时间过后，微博，微信会提醒用户进行重新登录。

Oauth认证流程可以如图所示：

（1）用户打开客户端后，客户端要求用户授予权限。

（2）用户点击确认，同意授权。

（3）客户端得到用户授权后，开始申请权限令牌。

（4）服务器对客户端的信息确认无误后，同意对其发放令牌。

（5）客户端拿到授权令牌后，转而向资源服务器申请资源。

（6）得到申请后的资源后，开放资源，继续服务。

**4.4.2 OAuth2.0认证方式**

Oauth认证有如下几种方式：

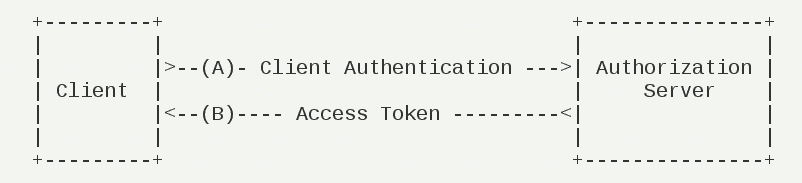
（1）授权码认证

（2）客户端认证

（3）密码模式

（4）简化模式

实际开发中，简化模式适用的场景很少，最多的还是客户端认证，“晨阅”APP使用的就是客户端认证方式。

客户端认证流程如下图，点击微博，微信，qq登录按钮后，第三方sdk会自动唤起对应的官方APP，开启官方APP的认证界面，让用户授权。

**4.4.3 OAuth2.0 认证过程（以新浪认证为例）**

（1）申请appkey，登录新浪开发者官网，注册自己的APP，获取appkey。

（2）填写APP相关信息，以及填写开发者相关信息（需要实名制认证）。

（3）提交审核后，等待新浪审核结果。

（4）如果审核通过，将会在控制台中看到统计数据。

（5）集成新浪sdk，参照官方文档，让用户提交授权。

## 自定义view布局实现

上面介绍了ReadView的自定义实现。整个APP，如果要实现真正的扁平化设计，需要要大量自定义控件和布局。一下，我将详细介绍自定义view的相关内容。

**4.5.1 viewpager的指示器的自定义实现**

Viewpager是Android中十分重要的一种控件，它实现了让界面随着手势的滑动而切换。大大的提高了用户的体验感。但是原生的viewpager并没有提供指示器，只是单纯的手势滑动而已。为了提高更好的用户交互，需要自定义指示器。

根据APP效果图可以看出，指示器其实是个线性布局，所以自定义一个继承LinearLayout的布局。

指示器含有很多自定义属性，选中字体的颜色，默认字体的颜色等，还有提示图片的方格，默认为三角提示，tab上的内容，tab的数量，文字大小，指示器的高宽等。

在构造方法中，需要通过context. obtainStyledAttributes()方法得到xml文件中自定义的默认属性。

指示器初始化后，还需要可以跟着手参与滑动，需要重写onScroll()方法。

if (offset>0&&mChild>0) {

//表示需要滑动偏移

if (mChildCountt != 1) {//表示viewpager的子模块不止1个

//算出对应的滑动偏移量

} else {

// 为count为1时 的特殊处理

//算出count只有一个的时候的偏移量。

}

}

如上的一段代码是用来计算指示器偏移量，得到偏移量后，调用scrollTo()方法开始滑动，最后调用invalidate()方法，调用onDraw方法刷新重新绘制。

# APP测试和评价

## 测试环境

此APP测试环境分成模拟器测试和真机测试两种。模拟器测试采用的genymotion模拟器，测试机型为Google Custom Phone 系统版本为Android 4.3，真机测试机为小米5，系统版本为Android 6.0.1。

## 功能测试

采用白盒测试，以Activity和Fragment为出发点，逐个界面进行测试。通过日志系统将系统日志输出，通过日志跟踪错误信息，进而达到调试功能的作用。

## UI测试

UI设计参考Android平台的追书神器和iOS平台的“快读”，配色按照自己意愿配色。

UI设计中，主要考虑的问题是Android碎片化问题，要竟可能的兼容各个版本。

## 单元测试

Android studio集成了Junit4 作为单元测试。在特定的单元测试包中，编写Junit类，完成特定功能的单元测试。

# 总结

此APP为毕业设计自选题目，因为本人并不擅长后台系统开发，所以采用“盗版”方式得到网络接口。通过这个APP的制作呢，让我对Android开发更加熟悉。制作此APP，让我对很多框架Retrofit，MVP，glide有了更深的理解。也对dragger这些框架有了全新的了解。开发中，代码量很大，出现了很多的问题和困惑，在自己不断的努力下，也一一克服，整个过程对自己确实是个很大的考验。

# 珍重申明

此APP接口通过抓包使用“追书神器”API，未经过官方授权认可。此APP仅作为内部学习交流使用，任何个人及组织不得使用其参与任何商业活动。