
《软件开发实践》课程报告

课题名称： E-Book-一款在线追书软件

课题负责人名（学号）： 2015141463064

同组成员名单（角色）： 黄智忠

指导教师： 洪玫

评阅成绩：

评阅意见：

提交报告时间：2017 年 6 月 15 日

E-Book-一款在线追书软件的设计与实现

软件工程 专业

学生 黄 智 忠 指导老师 洪 玫

【摘要】

由于现代人生活时间的碎片化以及移动互联网的兴起，人们移动阅读的需求越来越迫切。排队、购物、候车，人们随时可能拿出手机翻看未读完的小说、杂志等等。对于数量众多的网络文学爱好者来说，没有一款方便的在线追书软件。本项目基于Android平台，结合爬虫技术从小说网站爬取内容，以人性化的书架管理和阅读方式展现给用户。本项目在测试以及功能验证阶段，基本完成了既定的项目目标，使网络文学爱好者随时随地地追看自己喜欢的小说。

关键词：网络文学， 安卓， 爬虫

1. 引言（项目背景）

由于现代人生活时间的碎片化以及移动互联网的兴起，人们移动阅读的需求 越来越迫切。排队、购物、候车，人们随时可能拿出手机翻看未读完的小说、杂志等等。对于喜爱传统文学的读者来讲，一部 kindle 足以满足所有需求。但是除了传统文学以外，我国还有市场庞大、内容丰富的网络文学，以及数量众多的网络文学爱好者。

网络文学指作者在网上更新，每日更新时间、内容、字数不定，所以需要读者自己去查看更新阅读。

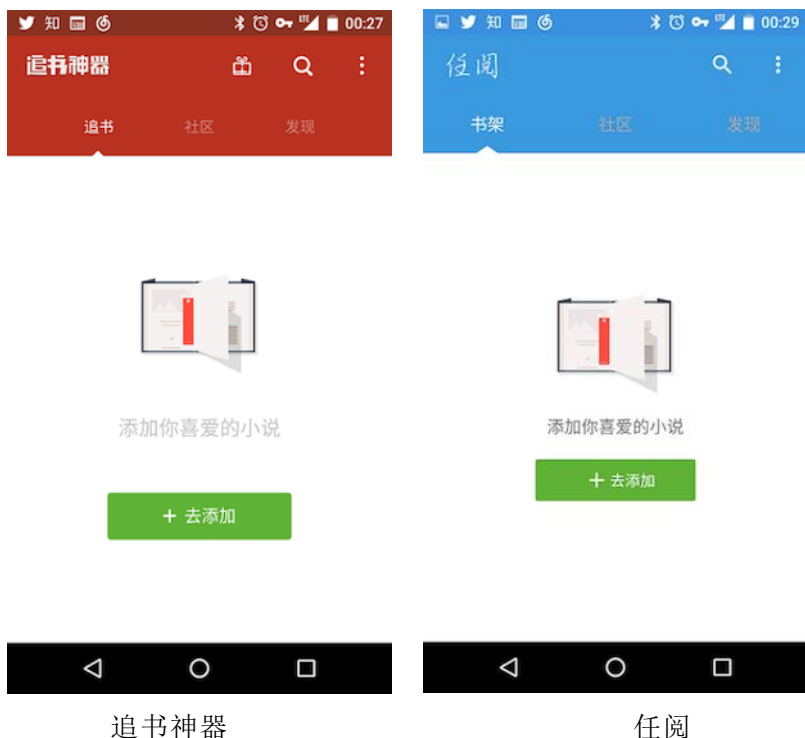
网络文学市场现在有许多平台，除了几大正版阅读网站之外，还有许许多多质量层次不齐的盗版阅读网站。对于大部分由于金钱紧张、或者不愿意为正版付账的读者来讲，盗版网站成为他们追读更新小说的首选。

中国网络文学的现状是市场很大，但是盗版也很严重。目前盗版网站数量多，受众也多。但是由于盗版网站在整个网络文学市场中的定位，就不可避免出现很多问题，例如盗版网站与正版网站更新的不同步，有的网站更新快，有的网站更新慢；盗版网站都存在很多广告弹窗问题，在阅读时不断弹出广告，而且不能关闭；这类盗版网站大部分都没有移动端应用，不能在线观看。

以上这些弊端对于读者来讲很不方便。读者需要忍受广告、只能在电脑上或者手机浏览器中阅读更新，甚至在多个网站之间切换等一系列问题，对于喜爱网络文学的读者来讲是一种折磨。

本项目针对以上问题，旨在开发一款方便所有网络文学爱好者的安卓APP。通过从各大盗版网站爬取书籍，读者在手机上阅读更新，更可以将喜爱的书籍加入书架，查看小说排行榜。在快节奏中，这款应用将节约用户很多时间。

2. 相关工作（对现有的系统进行分析）



图一 市场上现有的两款产品

上图为现有的市场上存在两款追书应用。两款应用外观相仿，功能也类似，并且较好地实现追书的功能，除此之外还具有排行榜和评论社区等功能。但是这两款应用中都具有一些不足的地方，第一款具有比较多的广告信息，另外是在后期会对用户进行收费阅读；第二款应用无广告，然而更新不及时或者容易交互不方便。

Material Design 中文名材料设计语言，是由 Google 推出的全新的设计语言，这种设计语言旨在为手机、平板电脑、台式机和“其他平台”提供更一致、更广泛的“外观和感觉”。

现有的基于 Material Design 风格的应用和原生控件已经比较多了，例如下拉刷新控件，Google 已经官方提供，而不需要开发者自行开发，并且较为美观。因此，基于 Material Design 风格可以使应用交互更加便于用户，美观以及加快开发流程。

网络爬虫（又被称为网页蜘蛛，网络机器人，在 FOAF 社区中间，更经常的称为网页追逐者），是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。另外一些不常使用的名字还有蚂蚁、自动索引、模拟程序或者蠕虫。现有阶段，爬虫技术已经较为成熟，市场上有大量的可供选择第三方开源工具可供选择，例如 jsoup 是一款解析 HTML 文档的工具，可以直接从网页中提取出需要的小说信息。

3. 问题描述（需求分析）

3.1 运行环境

编码操作系统：MacOS

服务器操作系统 centOS

服务器要求：内存 1G 以上，带宽
1M/bps 以上

应用部署平台：Android

客户端要求：Android 5.0 以上，内存 2G 以上流畅使用

编程工具：Android Studio, Pycharm

编程语言：Java, Python

硬件设备：云服务器，Google Nexus（原生安卓操作系统）

数据库：SQLite, MySQL

第三方开源库：Jsoup（Java），okHTTP（Java），LitePal（Android），JJson（Python）

3.2 系统功能描述

1. 登录

登录功能为用户登录应用，需要输入正确的账户名和密码才能成功登录。当 用户未完全输入账户名或密码，账户名或密码错误时向用户提示登录失败和 错误信息。

2. 注册

当用户没有账号时，在登录界面选择注册，然后填写用户名和密码进行注册。 当用户名和密码不符合规范时，注册失败。和远程服务器数据库对比如果用 户名已存在，也注册失败，提示用户失败信息。注册成功后，将用户名和密码回写到服务器数据库中。

3. 书架

a) 同步书架

当用户在应用主页下拉刷新时，自动对比服务器数据库中的用户书架信息，如果不同则下载那些不同的书籍信息。

b) 搜索小说

在应用主页顶部的功能栏中的搜索按钮点击后，出现在搜索界面中。用户在输入框输入小说名，然后应用提交到小说网站，将搜索结果爬取下来之后在应用中显示给用户。用户点击小说可查看具体信息。

c) 添加小说

在查看小说具体信息的界面点击“添加”按钮加入书架，或者在用户阅读小说返回但没有将小说加入书架时，提醒用户是否将小说加入书架。是则加入，否则取消操作。添加成功将结果返回远程服务器，远程服务器更新数据库。

d) 删除小说

在应用主页书架管理界面，用户长按小说名，出现对话框，然后点击删除按钮，则删除成功。删除成功将结果返回远程服务器，远程服务器更新数据库。

4. 书评

a) 发表书评

用户在评论区点击“添加”按钮，并输入了书评标题和书评具体内容和相关的小说名，然后发表书评。发送到服务器存储起来，当用户刷新时，则从服务器数据库下载数据。

b) 删除书评

用户删除自己发布的书评，传输到服务器对比是否是用户发布的书评，如果是，则删除成功，然后将结果显示给用户。

c) 回复书评

用户直接在书评详情界面填写输入框点击发送，回复其他用户的评论则

先点击该评论然后填写并发送。发送到服务器存储起来，当用户刷新时，则从服务器数据库下载数据。发送到服务器存储起来，当用户刷新时，则从服务器数据库下载数据。

d) 删除回复

用户删除自己发布的回复，传输到服务器对比是否是用户发布的回复，如果是，则删除成功，然后将结果显示给用户。

e) 查看书评

用户刷新，然后从服务器下载书评，显示在应用上。

5. 提醒

用户提醒可以提醒许多信息，例如小说已更新、有用户回复书评、读书状况、推荐小说等。用户可以点击查看提醒的具体信息。

6. 设置用户信息

用户可以修改个人信息，例如出生日期，性别等，然后用户确认后，将结果传输到远程服务器数据库进行更新。

7. 阅读

a) 调整字体大小 用户在阅读小说时，以点击屏幕中央出现下拉列表的方式修改阅读小说 界面的字体大小，例如加号+键增大字体，减号-减小字体。

b) 查看章节目录 用户在阅读小说时，以点击屏幕中央出现下拉列表的方式查看章节目录， 以及点击选择特定的章节之后并查看章节内容。

c) 查看小说介绍

用户在阅读小说时，以点击屏幕中央出现下拉列表的方式查看小说介绍，进入小说介绍界面后，介绍内容包括小说、小说简介、小说封面和小说字数。

d) 下载整本小说用户在阅读小说时，以点击屏幕中央出现下拉列表的方式出现下载按钮， 用户点击下载按钮应用通过网络将所有章节内容缓存到用户本地数据库，使用户可以在没有网络的情况下阅读小说。没有网络或下载失败时 提示用户连接网络或重新尝试下载。

e) 小说阅读翻页 用户在阅读小说时，用户点击屏幕两侧以达到切换阅读小说内容的目的，不管是章节内或章节之间的切换。

8. 其他

a) 查看小说排行

用户滑动小说主界面点击排行榜查看不同种类小说排行。用户点击特定的小说排行信息，并查看小说排行，显示小说排行信息。

b) 查看小说分类信息和相应数量

用户进入分类界面，并且可以看到分类信息和相应数目，点击后即可看到对应分类的显示结果。用户进入分类界面，显示小说所有的分类和数目，并且显示对应的分类排行。

c) 设置阅读切换页面效果

用户进入设置界面后设置阅读翻页效果，然后选中选择两种翻页效果，切换。用户进入设置界面后设置阅读翻页效果，然后选中选择两种翻页效果，切换成功相应的翻页特效。

d) 清除缓存

用户进入设置界面后，点击清除缓存。用户进入设置界面后，点击清除缓存，清除缓存成功，并提示用户。

e) 下拉刷新

用户在书架界面管理书架时，下拉书架列表对书架进行更新。用户下拉界面，下拉成功，并刷新书架列表，如果小说有更新，则提示用户。

3.3 系统处理的数据描述

3.3.1 静态数据

本项目的静态数据有：

1. 服务器 IP 地址和数据库密码
2. 盗版小说网站主页链接和搜索界面连接

3.3.2 动态数据

输入数据	输入描述	输出数据
小说名	用户搜索小说	应用提交小说名 后从盗版小说网站爬 取的小说结果
书评的标题和正文内容	用户在书评区发布书评	存储到服务器数据库中，当用户刷新时 显示书评信息
回复的书评和回复的正文内容	用户回复书评	存储到服务器数据库中，当用户刷新时 显示回复信息
用户信息	用户在设置界面 修改信息	应用将用户信息 存储到服务器数据库中，然后同步在设置界

		面显示
要添加或删除到书架的小说信息	用户添加或删除小说到书架	应用将更新本地和服务器数据库中，然后同步在书架界面显示
用户账户和密码	用户登录应用	应用将账户和密码传输到服务器然后验证是否正确，返回一个布尔值

3.3.4 数据词典

对系统中出现的各个实体的属性进行整理，形成数据字典，此可以作为后继开发过程中数据结构设计、数据库设计、数据库表结构设计的主要来源。

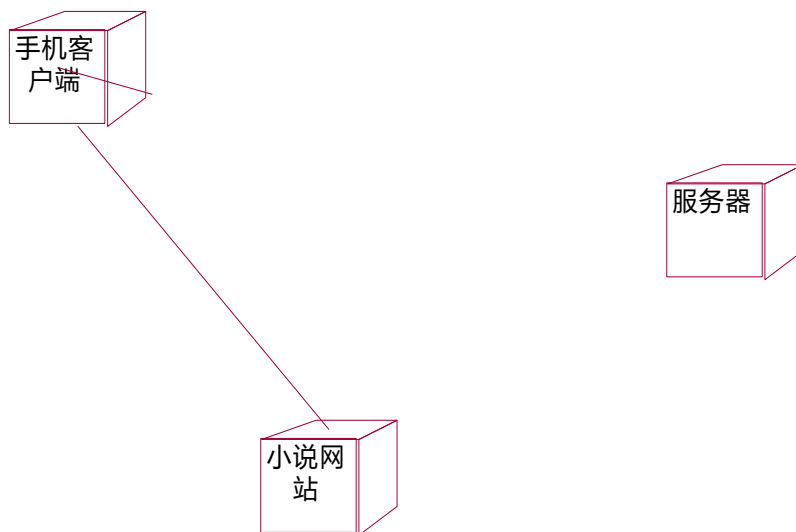
数据名	属性	数据描述
Bookinfo	BookName: 书名 BookURL: 书籍资源网址 BookIntroduction: 小说介绍 BookPic: 小说封面	主要存储小说的信息，一本书拥有一个 BookChapter, Comment, Reply, Notification
BookChapter	Map<ChapterName, ChapterURL>	存储小说的章节名和章节资源链接，BookChapter 包含

		多个键值对，每个键值对的值为章节内容 BookContent
BookContent	Content:String	存储小说正文内容
Comment	PublishTime:Date Title:评论标题 Content:评论内容 CommentUser:发布的用户	存储书评内容，包括发布时间、标题和正文内容
Reply	PublishTime:Date ReplyToUser:被回复的用户 ReplyFromUser:回复的用户 Comment:回复的书评信息 Title: 回复标题 Content:回复内容	存储书评的回复内容，包括发布时间、标题和正文内容，被回复的用户和回复的用户，回复的书评信息
Notification	NotifyTime:Date Title:提醒标题 Content:提醒内容	存储用户提醒的内容，包括发布时间、标题和正文内容

4. 解决方案（概要设计）

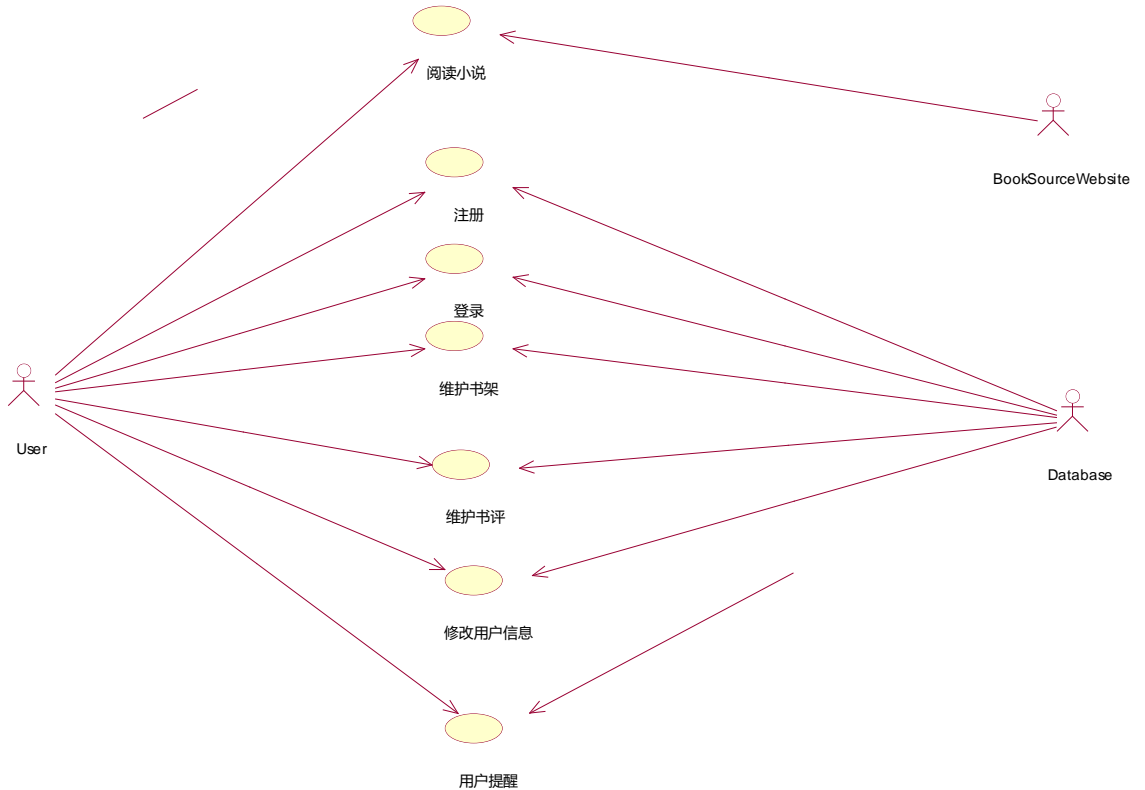
本系统是典型的 C/S 架构，基于功能和现实条件的约束选择的系统构建组成结构。由于需要同步和验证用户数据，所以数据只能存储在云服务器中。另外由于盗版小说网站访问限制，如果服务器多次访问则会被拒绝，因此小说爬虫只能归入本地客户端。另外的一些构建主要根据功能划分，小说阅读、显示书架书品等，都能从下至上的比较容易地实现整个系统。

系统部署图：



图二 系统部署图

系统用例图：



图三 系统用例图

4.1 架构描述

本项目前端主要实现一个安卓平台的应用与用户交互，编程环境为 Java 和 Android studio。这个应用在结构上可以分为 UI 设计、小说爬虫、阅读器三个部分，每个部分之间各成单元，但是也互相影响。

● 界面设计

在 UI 设计方面会根据 Google 推出的 material design 标准统一设计风格，Material Design 结合卡片式设计，又结合现实世界里纸张的隐喻，统一了 Google 在设计上的表达，从而展示出

一种强烈的风格。以此整个应用在界面上更有立体感和适应整个安卓操作系统。

出于对用户使用习惯上的考虑，在 UI 设计上需要直观的展示我们的主要功能。例如用户登录之后就能够直接看到个人书架，查看评论区则只需要左右滑动等等。

在项目正式编码之前，为了更形象的体验整个应用的 UI，还会基于“原型开发”，利用墨刀开发出一款粗略的应用原型，以预览我们的项目应用。完成原型后，给予用户体验，然后根据反馈做一定修改，直至最后满意为止。

◆ 小说爬虫

小说爬虫是整个应用的底层功能之一，爬虫爬取网站的 HTML 网页，然后解析整个字符串，获取的内容包括小说的介绍、章节目录和内容以及搜索结果。用户在线阅读其实就是一个爬虫爬取的过程，除了在线阅读之外，用户还可以下载书籍，意味着将爬虫爬取的内容保存到手机本地数据库中，除非用户清除应用缓存。

因为大部分网站采取的是百度的站内搜索，而百度在一个 IP 短时间多次访问是会被拒绝的，所以本项目不会采取爬虫在服务器上运行的方案（服务器外网 IP 不会更改），而是在安卓手机上运行爬虫。

以 Java 实现，本地运行可以减轻服务器的压力。

在整个客户端的爬虫设计上，项目会尽量分离爬虫部分，因此其他开发者可以很方便的扩展应用支持的源网站数量。

◆ 阅读器

阅读器意味着用户阅读小说的部分，是本项目的主要功能之一。有一个良好的阅读体验，整个项目也成功了一半。以腾讯的掌阅阅读器为例，在阅读时实现了类似于纸质书籍的界面，还包括翻页的特效。本项目除了实现这些以外，还包括许多其他的特色功能。

对于网络文学，比较经典的特点就是章节比较多，所以本项目能够在阅读时查看章节并且在章节之间切换。除此之外，因为还可以查看小说介绍和进入小说评论区。

设置字体大小和样式可以适应不同的用户的需求。

● 服务器

服务器部分有 Python 实现，充分利用了 Python 作为脚本语言的特性，由于服务器与客户端的分离，两者之间也不会有任何影响。服务器上用 MySQL 数据库存储用户数据、利用 HTTP 协议与客户端之间的进行数据传输。

服务器在用户登录时验证用户密码，以便保护用户的个人隐私。对于更换手机的用户来讲，服务器上存储的用户书架也可以传输给客户端同步。以及其他的用户评论等。

4.2 数据库结构设计

4.2.1 服务器数据库

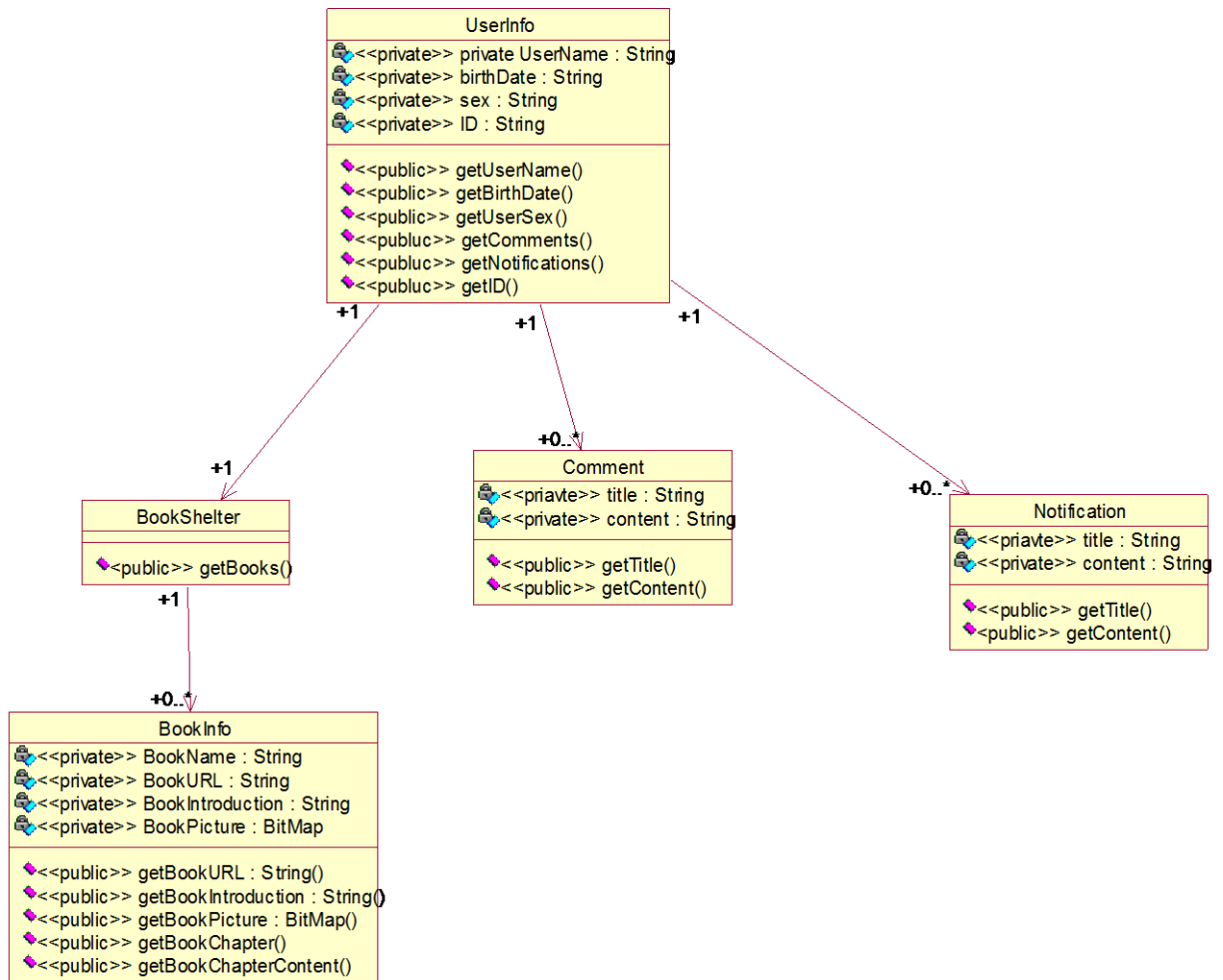
服务器数据库使用 MySQL。MySQL 是一个关系型数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

4.2.2 客户端数据库

客户端数据库使用 SQLite。SQLite，是一款轻型的数据库，是遵守 ACID 的关系型数据库管理系统，它包含在一个相对小的 C 库中。比起 Mysql、PostgreSQL 这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。

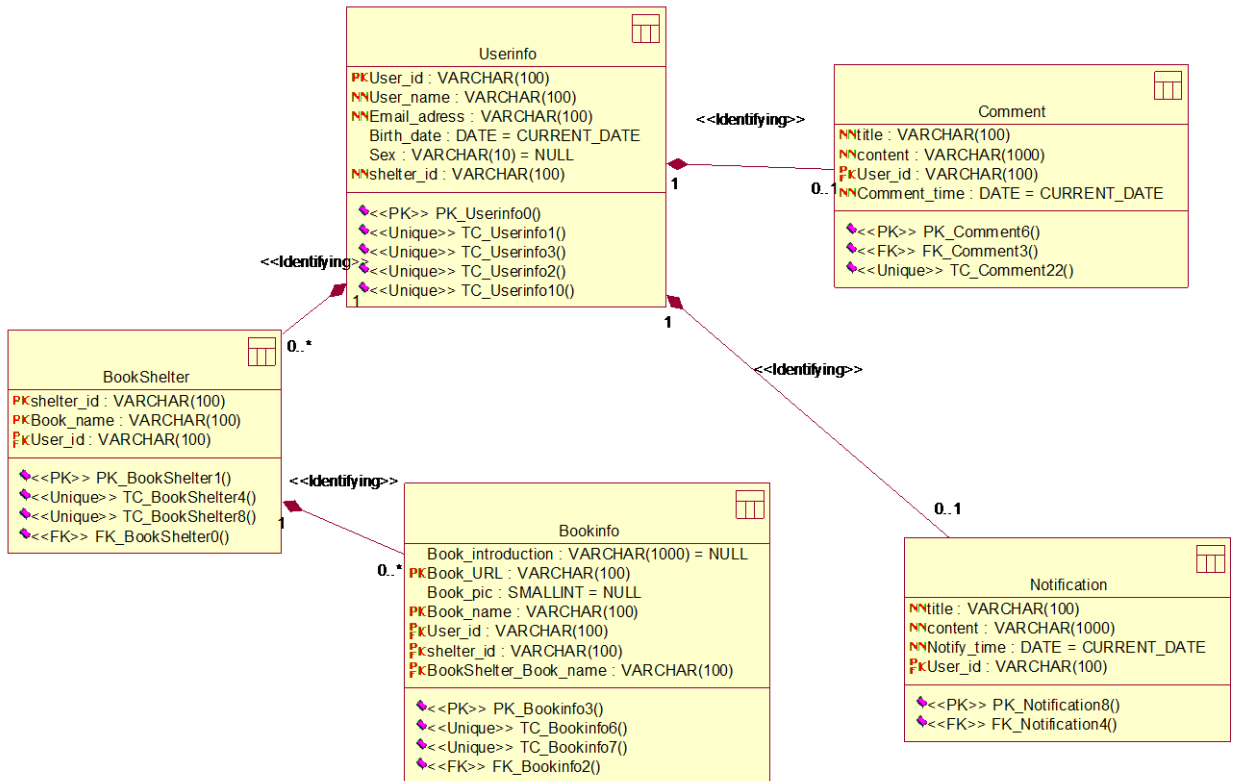
将会使用 LitePal 对客户端数据库进行管理。LitePal 是一款开源的 Android 数据库框架，采用了对象关系映射 (ORM) 的模式，将平时开发时最常用的一些数据库功能进行了封装，使得开发者不用编写一行 SQL 语句就可以完成各种建表、增删改查的操作。

类图：



图四 应用类图

数据库设计：



图五 数据库设计图

4.3 界面交互设计

4.3.1 界面对象和行为

4.3.1.1 主页

主页为进入应用首先看到的界面，小说书架会以列表的形式展现，用户通过长按小说名对书架进行管理，轻点小说则开始阅读小说。主页顶部有一个搜索按钮，将会进入一个拥有输入框的界面，用户输入小说名后搜索结果也将列表展现，然后点击则开始阅读。

用户退出阅读时如果未加入书架则弹出对话框提示用户是否加入书架。另外用户在主页下拉刷新小说列表时，如果检测到小说更新了，则弹出对话框提示用户更新了多少章节。

4.3.1.2 阅读小说界面

在阅读小说界面，用户可以阅读小说。点击屏幕中央后出现下拉框，显示各种选项，用户可以在这个界面内切换小说阅读的字体大小和特效以及阅读界面背景等设置。在上面的弹出框中有小说简介按钮，点击后进入小说介绍界面。在选项中点击目录按钮后进入小说目录界面，进入后点击相应的章节进入小说阅读界面刷新出现新的章节。

4.3.1.3 小说排行和分类

在小说排行界面，可以看到各类小说的排行榜，点击进去之后即可看到对应的周和热度等排行和小说具体信息。在分类界面用户可以看到各类小说的分类信息和相应分类的数量，点击进去就可以看到相应分类的排行。再点击即可进入小说介绍界面。

4.3.1.4 用户设置

在这个界面用户可以看到两个列表，分别是切换阅读效果和清除缓存。用户点击切换阅读效果时可以切换滑动翻页和模拟仿真翻页，而点击清除缓存时以一个列表显示删除缓存和清空书架。

4.3.2 部分 UI 截图



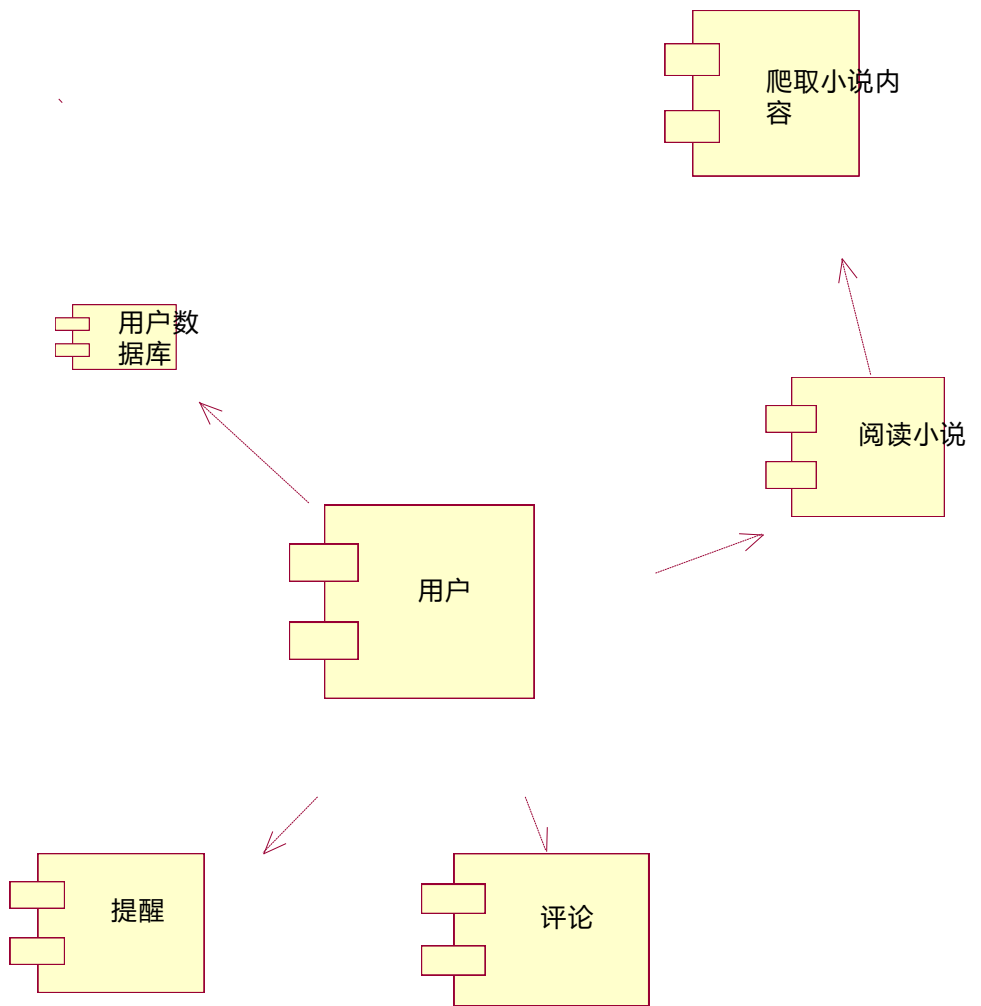
图六 搜索界面截图

5 实现细节（详细设计、实现）

本系统根据应用的不同功能划分模块。用户模块中包含对用户信息的各种操作，书架的管理和用户个人信息的修改等。

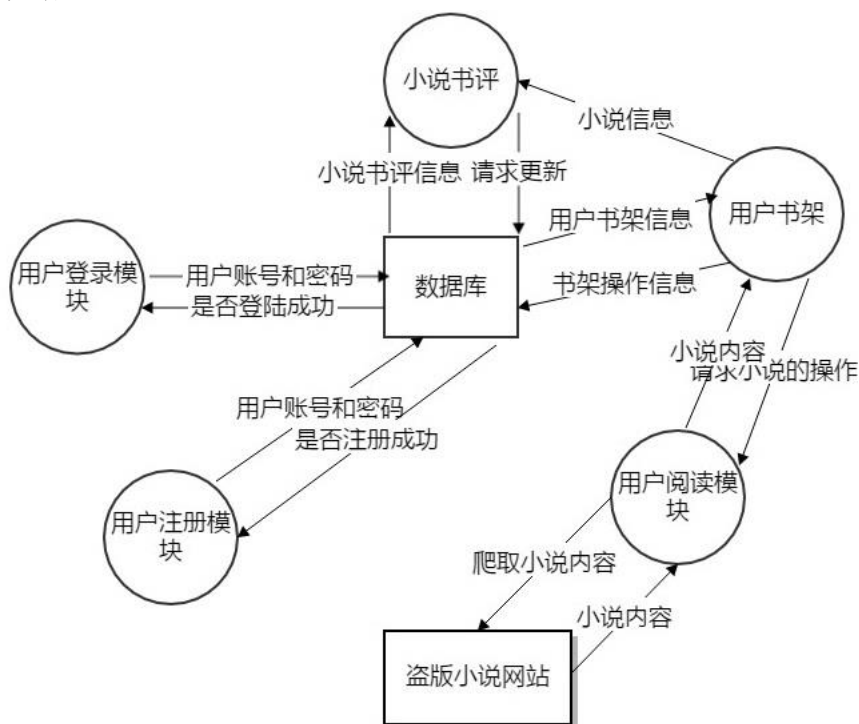
爬取小说模块则主要功能为小说内容的爬取，包括小说章节、介绍、正文等。其他提醒、评论等模块则实现提醒、评论信息的存储和发表、删除操作功能。

系统构件图：



图七 系统构件图

数据流图：



图八 系统数据流图

本项目前端主要实现一个安卓平台的应用与用户交互，编程环境为 Java 和 Android studio。这个应用在结构上可以分为 UI 设计、小说爬虫、阅读器三个部分，每个部分之间各成单元，但是也互相影响。

■ 界面设计

在 UI 设计方面会根据 Google 推出的 material design 标准统一设计风格，Material Design 结合卡片式设计，又结合现实世界里纸张的隐喻，统一了 Google 在设计上的表达，从而展示出一种强烈的风格。以此整个应用在界面上更有立体感和适应整个安卓操作系统。

出于对用户使用习惯上的考虑，在 UI 设计上需要直观的展示我们的主要功能。例如用户登录之后就能够在个人书架，查看评论区则只需要左右滑动等等。

在项目正式编码之前，为了更形象的体验整个应用的 UI，还会基于“原型开发”，利用墨刀开发出一款粗略的应用原型，以预览我们的项目应用。完成原型后，给予用户体验，然后根据反馈做一定修改，直至最后满意为止。

■ 小说爬虫

小说爬虫是整个应用的底层功能之一，爬虫爬取网站的 HTML 网页，然后解析整个字符串，获取的内容包括小说的介绍、章节目录和内容以及搜索结果。用户在线阅读其实就是一个爬虫爬取的过程，除了在线阅读之外，用户还可以下载书籍，意味着将爬虫爬取的内容保存到手机本地数据库中，除非用户清除应用缓存。

因为大部分网站采取的是百度的站内搜索，而百度在一个 IP 短时间多次访问是会被拒绝的，所以本项目不会采取爬虫在服务器上运行的方案（服务器外网 IP 不会更改），而是在安卓手机上运行爬虫。

以 Java 实现，本地运行可以减轻服务器的压力。

在整个客户端的爬虫设计上，项目会尽量分离爬虫部分，因此其他开发者可以很方便的扩展应用支持的源网站数量（小说网站很多，

每个网站解析字符串的逻辑不同，因此一个团队的精力是有限的，不可能支持所有网站）。

■ 阅读器

阅读器意味着用户阅读小说的部分，是本项目的主要功能之一。有一个良好的阅读体验，整个项目也成功了一半。以腾讯的掌阅阅读器为例，在阅读时实现了类似于纸质书籍的界面，还包括翻页的特效。本项目除了实现这些以外，还包括许多其他的特色功能。

对于网络文学，比较经典的特点就是章节比较多，所以本项目能够在阅读时查看章节并且在章节之间切换。除此之外，因为还可以查看小说介绍和进入小说评论区。

设置字体大小和样式可以适应不同的用户的需求。

■ 服务器

服务器部分有 Python 实现，充分利用了 Python 作为脚本语言的特性，由于服务器与客户端的分离，两者之间也不会有任何影响。服务器上用 MySQL 数据库存储用户数据、利用 HTTP 协议与客户端之间的进行数据传输。

服务器在用户登录时验证用户密码，以便保护用户的个人隐私。对于更换手机的用户来讲，服务器上存储的用户书架也可以传输给客户端同步。以及其他的用户评论等。

6 实验验证（测试、分析、评价）

6.1 采用的测试技术和测试方法、过程

测试是整个项目的重要部分，对于保障项目的执行和最后结果的交付有重要作用。单元测试保证了每个功能单元的正常执行，例如安卓客户端的阅读器、爬虫等单元。

在测试阶段我们采用了单元测试和功能测试相结合的方法，单元测试保证了每个功能单元的正常执行，功能测试保证了应用实现了具体功能并且达到了预期的功能效果即期望的输出，再结合单元测试的测试结果，确定功能的正确性和如果出现错误，定位应用出现错误的位置并修正错误。

6.2 系统测试用例和自测记录

情况分类	单元描述	输入数据	期望输出	实际输出	测试状态 (P/F)
注册成功	用户在初次使用应用没有账号时，在应用登录界面选择注册账号，输入用户名为账号并设置密码及确认	用户名 (username) 为 6-16 位英文字符，密码 (password) 为 9-20 位字符，确认密码 (verify_password) 为 9-20 位字	更新数据库中用户信息，提示用户注册成功	后台查看没有更新数据库，但是提示用户注册成功。	F

	密码后选择注册，注册成功	符，且 password 与 verify_password 相同，密码可包含任意键盘字符			
用户已存在	用户输入用户名和密码后，传输到服务器，服务器发现已存在同用户名的账户，则服务器返回注册失败结果并以对话框提示用户用户名已	用户名（username）为 6-16 位英文字符，密码（password）为 9-20 位字符，确认密码（verify_password）为 9-20 位字符，且 password 与 verify_password	客户端弹出提示用户因用户名已存在导致注册失败	后台没有提示用户用户名已存在，注册失败	F

	存在	相同，密码可包含任意键盘字符，且用户名在服务器的数据库中已存在			
用户名或密码不符合要求（过长、过短或包含非法字符）	用户选择注册，但是用户名包含非英文字符或密码过长或过短，提示用户修改用户名或密码	用户名不符合成功注册中输入要求，例如用户名包含非英文字符或密码过长或过短	提示用户因用户名或密码不符合规范导致注册失败	后台提示用户名不规范	P

导致注册失败					
密码和确认密码不同注册失败	用户选择注册后客户端发现密码和确认密码不同提示用户注册失败	用户名和密码符合规范但密码和确认密码不同	提示用户因密码和确认密码不同导致注册失败，并重新注册	后台提示用户名和密码不规范	P
用户未完全输入而导致注册失败	用户在注册时没有输入用户名、密码或者确认密码而导致客户端提醒用户注册失	用户用户名、密码或者确认密码三个输入框至少一个输入框没有填写任何内容	客户端弹出提示用户因未完整填写信息导致注册失败，并提示用户填写完整	后台提示用户名和密码不规范	P

	败，请完全填写				
网络未连接	因用户设备网络差或者未连接网络，用户无法完成注册	用户名 (username) 为 6-16 位英文字符，密码 (password) 为 9-20 位字符，确认密码 (verify_password) 为 9-20 位字符，且 password 与 verify_password 相同，密码可包含 任意键盘字符	提示用户打开网络连接，请稍后重试	后台没有提示，应用崩溃后重新启动	F

成功登录	用户在登录界面输入正确的用户名和密码后点击登录按钮后成功登录	用户输入正确的用户名（username），密码（password），然后点击登录按钮	提示用户已成功登录，进入应用主界面	后台提示用户成功	P
用户名和密码不匹配（包括用户名不存在，用户名与密码不匹配）	用户输入不匹配的用户名和密码或不存在的用户名后登录失败	不匹配的用户名和密码或不存在的用户名	弹出对话框提示用户因用户名或密码错误登录失败，并重新输入	后台没有提示	F

网络连接失败	因用户设备网络差或者未连接网络，而导致服务器无法正常提供服务，用户无法完成登录	任意字符	提示用户打开网络连接，请稍后重试	后台没有提示	F
显示搜索结果	用户在搜索界面中输入将要搜索的小说名，然后客户端显示所有搜索结果，点进去就能查看小说介绍和阅读小说	小说名 (bookname)必须是包含汉字或英文的字符串，且不能为空	搜索结果中的小说介绍和地址被爬取下来，搜索结果以列表显示在搜索界面，用户点击后进入显示小说的介绍的界面	可以显示小说搜索结果，和预期结果一致。	P

显示 章节目录	用户在阅读小说时，以点击屏幕中央出现下拉列表的方式查看章节目录，以及点击选择特定的章节之后并查看章节内容	小说章节目录 (bookChapterList) 为一个 list 列表，可以不包含任何内容。	用户在阅读小说的界面时，可以点击屏幕中央出现下拉列表，显示出章节目录，并点击特定章节查看章节内容。	用户在阅读小说的界面时，可以点击屏幕中央出现下拉列表，显示出章节目录，并点击特定章节查看章节内容。	P
------------	--	--	---	---	---

显示章节内容	用户在阅读小说时，全屏显示小说内容，并且通过点击屏幕两侧切换阅读小说前面或者后面的内容	小说章节内容 (bookChapterContent) 为一个字符串，可以包含任何内容，且可为空。	用户选择阅读小说章节时，应用从源网站爬取到了章节真正内容并显示到屏幕上	用户选择阅读小说章节时，应用从源网站爬取到了章节真正内容并显示到屏幕上	P
显示小说介绍	用户在阅读小说时，以点击屏幕中央出现下拉列表的方式查看小说介绍，进入小说介绍界面	小说信息 (bookinfo) 为一个包含小说信息的类，包括小说介绍 (bookIntro) 为一个内容根据源网站的字符串，书名 (bookName)	用户选择查看小说介绍后进入小说界面，然后应用从源网站爬取到了小说封面、书名和简介等信息并显示到屏幕上	用户选择查看小说介绍后进入小说界面，然后应用从源	P

	后，介绍内容包括小说、小说简介、小说封面和小说字数	是一个不为空的英文或中文字符串，小说封面（bookPic）是一张图片，在安卓上为 Bitmap 类型		网站爬取到了小说封面、书名和简介等信息并显示到屏幕上	
调整字体大小	用户在阅读小说时，以点击屏幕中央出现下拉列表的方式修改阅读小说界面的字体大小，例如加号+键增大字体，减号-减小	用户按下字体控制键则改变字体大小相关变量（fontsize），fontsize 为一个 6-10 的整型数据，用户按下一次加号键则增加 1 或按下一次减号键则减小 1，超过 6-10 的范围时则	小说阅读界面的字体大小成功地根据用户意愿修改。如果用户按下的范围超过了应用指定的范围即 6-10，则弹窗显示字体大小已超过范围	小说阅读界面的字体大小成功地根据用户意愿修改。如果用户按下的范围超	P

	字体	不能被更改		过了应用指定的范围即6-10，则弹窗显示字体大小已超过范围	
下载整本小说	用户在阅读小说时，以点击屏幕中央出现下拉列表的方式出现下载按钮，用户点击下载按钮应用通过网络将所有章	用户按下下载按钮的按钮事件（OnButtonClick Listener）	小说章节目录和小说章节内容被成功的缓存到用户本地数据库中	小说章节目录和小说章节内容被成功的缓存到用户本地数据库中	P

	节内容缓存到用户本地数据库，使用户可以在没有网络的情况下阅读小说。没有网络或下载失败时提示用户连接网络或重新尝试下载				
小说阅读翻页	用户在阅读小说时，用户点击屏幕两侧以达到切换阅读小说内容的目的，不管是章节内或章	用户点击屏幕的位置事件 (clickPosition)，分为屏幕左侧和屏幕右侧	用户点击屏幕左右两侧以达到向前或向后切换阅读小说内容的目的	用户点击屏幕左右两侧以达到向前或向后切换阅读小说	P

	节之间的切换			内容的目的	
网络连接失败	用户手机没有连接网络或网络不稳定的情况下，阅读、刷新、下载小说内容	检测用户手机网络连接情况（ifNetworkConnected），网络连接则为真，网络未连接或连接不稳定则为假	用户手机没有连接网络或网络不稳定时则弹窗给用户显示网络未连接	用户手机没有连接网络或网络不稳定时则弹窗给用户显示网络未连接	P

添加小说	用户在小说介绍界面点击加入书架按钮、或者在退出阅读小说界面但为将小说加入书架时弹出窗口提示用户是否将小说加入书架。成功加入书架后将小说信息存储到用户本地数据库并和服务器数据库保持同步	用户点击加入书架的按钮事件（OnClickListener）	添加书架成功后，用户刷新小说书架时成功出现在书架中，并成功存储到本地数据库和服务器数据库	添加书架成功后，用户刷新小说书架时成功出现在书架中，并成功存储到本地数据库和服务器数据库	P
------	---	--------------------------------	--	--	---

删除小说	用户在书架界面管理书架时，选中小说后点击删除，从书架中删除小说。成功删除小说后将小说信息从用户本地数据库并和服务端数据库保持同步，并且同步本地小说缓存数据	用户在书架界面管理书架时，选中小说后点击删除按钮的按钮事件 (OnButtonClickListener)	小说成功地从用户小说书架界面、用户本地数据库和服务端数据库中删除	小说成功地从用户小说书架界面、用户本地数据库和服务端数据库中删除	P
------	---	---	----------------------------------	----------------------------------	---

同步 书架	用户在书架界面下拉刷新书架列表，然后同步服务器中的小说书架列表并刷新本地数据库中保存更改，应用从本地数据库重新读入书架信息	用户下拉书架列表的事件 (SwipeRefreshLayout), 本地数据库和服务器数据库中书架书架列表(bookShelfList, List<Bookinfo>, 可为空)	应用书架界面显示同步和刷新后的小说书架	应用书架界面显示同步和刷新后的小说书架	P
查看 书架	用户登录后会出现在书架界面，书架界面以列表显示用户书架的所有小说，包括	用户成功登录，本地数据库书架列表 (bookShelfList, List<Bookinfo>, 可为空)	应用书架界面显示小说书架	应用书架界面显示小说书架	P

	小说名和最近更新的章节。				
--	--------------	--	--	--	--

6.3 测试结果分析和评价

在测试过程中，发现由于本项目基于 Google 的 material design 标准，所以只支持 Android 5.0 操作系统及以上版本。另外在用户手机未联网阅读时，只有在加载页面才会出现网络状况的提示。并且由于用户阅读的小说资源皆来自盗版小说网站，所以小说章节内容、小说更新取决于源网站，本项目不能独立于小说源网站。当小说源网站不能够正常浏览和搜索时，可能会出现页面空白或乱码等未知 bug。

最终的测试结果分析可知，应用实现了大部分预期的功能，但是稳定性和应用运行效率还有待提高。还有一点是本项目的数据精度在一定程度上准确、可靠和真实，对数据库进行操作时和与小说网站的内容搜索和爬取也不会影响到正常使用，当时当用户网速受限时则有一定影响。最重要的，本应用对数据查询的准确率很准确。

7 结束语（结论、进一步的工作）

本系统从三月初开始开发，直至六月初正式结束，中间经过了启动（开发计划）、需求分析、设计、实现、测试五个阶段，用时三个月左右。中间经历过许多困难。在三月初时，我开始开发这套系统，期望快速开发迭代出一个能够展示的 demo，实际也行的通了，花费两三个星期，应用就已经能看。再到后面经过需求分析、建模、设计等阶段，又逐渐有了明确的需求和安全等方面的意识。实际上当前的系统和以前的系统没有什么关系了，相当于推倒一套已经完成的系统，重新编码，设计等等，所以可以说中间过程坎坷，花费的精力比较多。当然，结果也是挺好的。

我这组只有我一个人，工作也不可能分工，所以可以说我完全全的经历了一整套软件工程开发流程。相对的其他文档或者编码工作分开来的小组，我的收获也比较大，从一系列的文档编写能力，到建模能力，再到前后端代码编写能力，都得到了一定的锻炼。在 GitHub 上一共有将近 60 次 commit 的纪录，每一次 commit 内容都有记录着我这一次 commit 的内容。

跨时三个月，直至完成了大部分的需求。如今要我来评论整个开发系统，我可以分为优缺点来完成。

优点在于我在许多阅读器的局限的阅读功能上，扩展出许多人性化方面的功能。用户可以设置翻页的效果，是模拟翻页还是滑动翻页；可以设置夜间和日间的阅读模式，对保护用户的眼睛有很大的作用；可以调节阅读时的字体大小和背景，来满足不同用户的需求；再比如搜索时，有智能提示功能；可以查看分类信息的小说，看到排行榜，找到自己喜欢的小说。

另一点是这个应用相对来说是比较稳定的，长期使用也不会遇到莫名崩溃的情况。

然而缺点也有，相对于最初设立的项目需求，我没有实现登录和云同步书架的功能。登录功能可以说很简单，我可以调用微信或者 QQ

的第三方登录接口实现登录，即使自己写一个登录其实也很容易，但是云同步书架的功能由于时间的限制，实现起来比较复杂，所以并没有实现这个功能。因此在我想来，既然服务器不能同步了，那登录其实也没有什么意思，我就做一个盗版小说网站和用户之间的桥梁的应用就已经足够，还很方便不用登录。还有一点是在小说缓存过程中，不能很好的实现缓存功能，只能自动缓存后续三个章节，这是一个小缺陷。

因此，总结起来，后续的对应用的进一步完善主要是针对缺点的完善。加入同步功能后，再加入登录功能，使用户在换手机时书架和阅读记录等数据都不会丢失。可以说是一个非常值得完善的地方。而第二个缺点则需要后续过程中对应用进行修复，完善。最后才能尽善尽美，做出一个好的系统。

参考文献

- [1] 巴斯，克莱门茨，凯兹曼. 软件构架实践. 清华大学出版社, 2002
- [2] 普雷斯曼. 软件工程实践者的研究方法. 软件构架实践. 机械工业出版社, 2005

附录：

1. 项目开发计划；
2. 软件需求规格说明书；
3. 软件设计文档；
4. 软件源代码；
5. 软件测试文档；
6. 用户手册；