



Y3408767

分类号: TN919

密 级:

单位代码: 10422

学 号: 201511817



山东大学
SHANDONG UNIVERSITY

硕士学位论文

Thesis for Master Degree

论文题目:

基于 Android 的书籍阅览与管理系统的设计与实现

Design and Implementation of Books Reading and

Management System Based on Android

作 者 姓 名

李淑敏

培 养 单 位

信息科学与工程学院

专 业 名 称

信息与通信工程

指 导 教 师

袁东风 教授

合 作 导 师

2018 年 5 月 15 日



分类号: TN919

单位代码: 10422

密 级:

学 号: 201511817



山东大学

SHANDONG UNIVERSITY

硕士学位论文

Thesis for Master Degree

论文题目: 基于Android的书籍阅览与管理系统的设计与实现

Design and Implementation of Books Reading and
Management System Based on Android

作者姓名 李淑敏

培养单位 信息科学与工程学院

专业名称 信息与通信工程

指导教师 袁东风 教授

合作导师 _____

2018 年 5 月 15 日

原创性声明

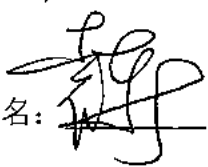
本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

论文作者签名：李淑敏 日期：2018.5.15

关于学位论文使用授权的声明

本人完全了解山东大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权山东大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

(保密论文在解密后应遵守此规定)

论文作者签名：李淑敏 导师签名： 日期：2018.5.15

目录

摘要	I
ABSTRACT	III
符号说明	V
第 1 章 绪论	1
1.1 选题背景	1
1.2 国内外研究现状	2
1.3 研究内容及意义	3
1.3.1 研究内容	3
1.3.2 研究意义	4
1.4 论文组织	4
1.5 本章小结	5
第 2 章 相关技术研究	6
2.1 Android 自定义 View 研究	6
2.1.1 View 及事件分发机制	6
2.1.2 自定义 View	8
2.2 Glide 控件研究	10
2.3 Android 系统动画技术研究	14
2.4 本章小结	17
第 3 章 系统需求分析与系统设计	18
3.1 系统需求分析	18
3.1.1 功能需求分析	18
3.1.2 非功能需求分析	21
3.2 系统设计	21
3.2.1 系统架构设计	21
3.2.2 系统框架模式设计	22
3.2.3 系统主要模块设计	23
3.3 本章小结	26

第 4 章 书籍阅览与管理系统的实现	27
4.1 系统界面框架的实现	27
4.2 系统功能模块的实现	29
4.2.1 读书模块的实现	29
4.2.2 读书笔记模块的实现	35
4.2.3 读书计划模块的实现	39
4.2.4 云备份模块的实现	42
4.2.5 好书推荐模块的实现	45
4.2.6 登录模块的实现	50
4.3 本章小结	55
第 5 章 书籍阅览与管理系统的测试	56
5.1 测试环境	56
5.2 测试项目及结果	56
5.2.1 功能测试	56
5.2.2 流畅度测试	58
5.2.3 稳定性测试	59
5.3 本章小结	61
第 6 章 总结与展望	62
6.1 总结	62
6.2 展望	63
参考文献	64
致谢	68
硕士学位期间的科研成果和参加的项目	69

CONTENTS

Chinese Abstract	I
English Abstract.....	III
Symbols.....	V
Chapter 1 Introduction	1
1.1 Thesis Background	1
1.2 Research Status at Home and Abroad.....	2
1.3 Main Research Contents and Significance	3
1.3.1 Main Research Contents.....	3
1.3.2 Research Significance	4
1.4 Thesis Structure	4
1.5 Conclusion	5
Chapter 2 Related Technologies Research	6
2.1 Research on Android Custom View.....	6
2.1.1 View and Event Propagation of Android.....	6
2.1.2 Custom View	8
2.2 Research on Glide.....	10
2.3 Research on Android Animation Technology.....	14
2.4 Conclusion	17
Chapter 3 Requirements Analysis and System Design.....	18
3.1 System Requirements Analysis	18
3.1.1 Functional Requirements Analysis	18
3.1.2 Non-functional Requirements Analysis.....	21
3.2 System Design	21
3.2.1 System Architecture Design	21
3.2.2 System Framework Design.....	22
3.2.3 System Main Module Design	23
3.3 Conclusion	26

Chapter 4 Implementation of Book Reading and Management System	27
4.1 The Implementation of System Interface Framework	27
4.2 The Implementation of System Functional Module	29
4.2.1 The Implementation of Book Reading Module	29
4.2.2 The Implementation of Reading Notes Module	35
4.2.3 The Implementation of Reading Plans Module	39
4.2.4 The Implementation of Cloud Backup Module	42
4.2.5 The Implementation of Recommended Books Module	45
4.2.6 The Implementation of Login Module	50
4.3 Conclusion	55
Chapter 5 Test of Book Reading and Management System	56
5.1 Test Environment	56
5.2 Test Items and Results	56
5.2.1 Functional Test	56
5.2.2 Fluency Test	58
5.2.3 Stability Test	59
5.3 Conclusion	61
Chapter 6 Summary and Future Work	62
6.1 Summary	62
6.2 Future Work	63
References	64
Acknowledgements	68
Scientific Achievements and Projects	69

摘要

随着移动互联网的飞速发展,移动智能终端凭借其丰富的功能、便捷的使用对人们的生活、工作以及学习都产生了深远影响。在“全民阅读”的时代背景下,阅读愈发成为人们生活中不可或缺的一部分,人们的阅读模式也深受移动互联网技术的影响,正在悄然改变。阅读资源不再止于纸质书籍,电子书也成为人们阅读的重要资源之一。根据第十四次全国国民阅读调查报告数据显示,2016 年我国成年国民的数字化阅读方式的接触率达到 68.2%,移动阅读终端成为数字阅读主要入口。据 StatCounter 统计,目前 Android 系统已经成为全球操作系统市场中的第一大操作系统。因此,开发一款可用于 Android 设备的书籍阅览与管理系统具有非常广泛的应用价值和实际意义。

本文首先在深入研究 View 的层次结构、View 的工作原理和 Android 事件分发机制的基础上实现自定义 View。对比并分析了 Android 主流图片加载框架并从源码角度研究了 Glide 框架,学习其设计架构、设计模式及编码风格。并且,还研究了 Android 系统动画技术。

其次,对 Android 应用市场中现有阅读类应用进行分析与对比,总结其功能及特点。在调研的基础上,发现了用户在当今快节奏的生活中读书缺乏计划性、备份较繁琐等问题。针对以上问题,对本系统进行了全面的系统需求分析,包括功能需求和非功能需求。最终设计了基于 C/S 架构、采用 MVC 框架的系统总体设计方案,并对系统具体功能模块进行了规划。

再次,提出并实现了一种基于网络状况与预约模式的备份策略。首先检测当前设备的联网类型及网络质量提醒用户,并提供了预约备份功能。该策略避免了用户在不知情的情况使用移动数据流量,而且预约模式的设计可以减少用户操作,优化用户体验。

在以上研究的基础上,论文设计并实现了基于 Android 系统的书籍阅览与管理系统。本系统为用户提供游客与账户两种登录模式,并且实现了书籍的阅览与管理、读书笔记、读书计划、书籍云备份、好书推荐功能。最后从系统功能、系统流畅度、系统稳定性三个方面对系统进行测试工作。对系统每个功能设计并实施

了功能测试用例，利用 dumpsys、Monkey 等工具分别测试系统流畅度及稳定性，最终系统各项测试均表现良好。

本系统在满足用户高效读书的基本需求上，制定读书计划功能增强用户读书计划性，支持添加读书笔记，而且提供了基于网络状况与预约模式的书籍云备份功能解决用户存储空间不足及不同设备间的数据迁移问题，还通过网络爬虫工具 Jsoup 为用户推荐评分较高的优秀书籍，为用户打造了一个多功能的高效读书平台。

关键词： Android 操作系统；电子书；备份；阅读计划

ABSTRACT

With the rapid development of the mobile Internet, mobile smart terminals have a profound influence on people's lives, work and learning through their rich functions and convenient use. In the era of "all people reading", reading has become an indispensable part of people's lives. People's reading patterns are also deeply influenced by mobile Internet technologies and are quietly changing. Reading resources are no longer limited to paper books, and e-books have become one of the most important resources for people to read. According to the data from the 14th National Reading Survey Report, the contact rate of adult citizens' digital reading of in China was 68.2% in 2016. Moreover, mobile reading terminal applications have become the main entrance for digital reading. According to StatCounter statistics, at present Android system has become the largest operating system in the global operating system market. Therefore, developing a book reading and management system that can be used for Android devices has a very wide range of application value and practical significance

First of all, this thesis focuses on the custom View for this system based on the study of the View's hierarchical structure, View's working principle and Android event distribution mechanism. Besides, this thesis compare and analyze the mainstream image loading framework of Android and study the Glide framework from the perspective of source code to learn its design architecture, design patterns and coding styles. In addition, this thesis discusses the Android system animation technology.

Secondly, after analyzing and comparing the existing reading applications in the Android application market, this thesis summarizes their functions, advantages and disadvantages. Based on the research, we found that users have problems in planning, backup and cumbersome reading in today's fast-paced life. In response to the above problems, a comprehensive system requirements analysis was conducted on the system, which is from the perspective of functional requirements and non-functional requirements. The requirements analysis is combined with the actual needs of users such

as efficient reading, formulating reading plans and quick reading notes. Finally, the overall design of the system based on the C/S framework and the MVC framework is designed, and the specific functional modules of the system are planned as well.

Again, a backup strategy based on network conditions and reservation modes is proposed and implemented. Firstly, the system detects the current device's network type and network quality to remind the user, and provides an reservation mode for backup. This strategy avoids users from using mobile data traffic without their knowledge. Besides, the design of a reservation mode can reduce user operations and improve the user experience.

Based on the above research, a book reading and management system based on the Android system was designed and implemented. This system provides users with two login modes, including visitors and accounts. And it also realizes the reading and management of books, reading notes, formulating reading plans, books cloud backup, and good book recommendation functions. Finally, the system is tested from three aspects: system function, system fluency, and system stability. The functional test cases were designed and implemented for each function of the system. The fluency and stability of the system were tested by using tools such as dumpsys and monkey. Finally, all tests of the system performed well.

This system satisfies the basic needs of users to read books. In addition, it lets users formulate reading plans to urge themselves to read books and add reading notes. It also provides a book cloud backup function based on network status and reservation mode, which can solve the problems of insufficient storage space and data migration between different devices. Besides, the system can recommend high-ranking excellent books for users through web crawler tools. The whole system creates a multi-functional and efficient reading platform for users.

Key Words: Android operating system; e-book; backup; reading plan

符号说明

缩写	英文全称	汉语翻译
SDK	Software Development Kit	软件开发工具包
C/S	Client/Server	客户端/服务器
MVC	Model View Controller	模型、视图、控制器
OOM	Out Of Memory	内存不足
API	Application Programming Interface	应用程序接口
ADB	Android Debug Bridge	安卓调试桥接工具
UI	User Interface	用户界面
HTTP	HyperText Transfer Protocol	超文本传输协议
JSON	JavaScript Object Notation	JavaScript 对象符号
ANR	Application Not Responding	应用程序无响应
GUI	Graphics Users Interface	图形用户界面
ORM	Object Relational Mapping	对象关系映射

第 1 章 绪论

1.1 选题背景

随着现代科技的飞速发展，3G/4G 技术不断成熟，移动互联网技术逐渐大放异彩。根据相关调查报告显示，早于 2012 年，移动互联网用户量就超越了 PC 互联网用户^[1]。据工信部发布的 2017 年 11 月份通信行业经济运行情况显示，我国移动互联网用户总数达到惊人的 12.5 亿户，同比增长 16.6%^[2]。作为移动互联网技术中必不可少的移动智能终端也如雨后春笋般快速发展，移动智能终端逐渐摆脱最初功能单一的传统定位，转变为新型的综合信息处理平台^[3]。

操作系统是智能移动终端的灵魂^[4]，早期的 Symbian 系统已经近乎绝迹，现在主要有 Android、iOS、Windows Mobile 等操作系统^[5]。Android 系统自 2008 年面世以来，作为一个开放式的移动互联网操作系统取得了飞速的发展，据 StatCounter 统计，Android 系统现在已经成为全球操作系统市场中的第一大操作系统^[6]。而且，Android 系统的应用不再局限于手机、平板等传统移动设备，更着眼于可穿戴设备、智能电视、车载、机器人等更多领域，Android 系统未来的发展前景不可估量。

腹有诗书气自华，读书是提升自我、滋润心灵的重要方式。我国也在大力推动大众阅读，“全民阅读”已经连续多年被写入政府工作报告，阅读成为人们生活中不可或缺的一部分，它对于一个人、一个民族、一个国家都至关重要。Android 移动智能终端的发展带动了其应用市场的飞速发展，目前 Android 应用种类繁多、数量巨大，Android 终端凭借其丰富的功能、便捷的使用对人们的学习、工作以及生活都产生深远影响^[7]。民众的阅读模式正在悄然改变，已然由传统的纸质阅读逐渐转变为数字化阅读方式。根据第十四次全国国民阅读调查报告数据显示，2016 年我国成年国民的数字化阅读方式的接触率达到 68.2%，而移动阅读终端成为数字阅读主要入口^[8]。

目前，作为数字阅读重要资源的电子书不仅有环保、低成本、方便存储携带

等特点^[9]，而且可以满足用户对于碎片化时间的合理利用，深受人们的喜爱。面对设备中大量的电子书时，人们在管理书籍时会比较麻烦，而且在当今快节奏的生活中用户读书缺乏计划性，也难以进行高效地读书、管理笔记等。大量电子书将占用较多存储空间，这对设备有限的存储空间也是一个问题。因此，开发一款基于 Android 系统的书籍阅览与管理系统具有较高的应用价值与实际意义。

本论文设计的书籍阅览与管理系统依托于山东大学中国虹计划协同创新中心袁东风教授带领其团队研制的具有低成本、低功耗、易操作等特点的中国虹计算机^{[10][11]}，其搭载 Android 4.4 操作系统。本文设计并实现的系统运行在第四代中国虹计算机上，解决用户阅读电子书的需求，为其提供高效多样的功能。

1.2 国内外研究现状

针对用户阅读书籍这一巨大需求，各种电子书阅读器、阅读类应用等应运而生。本文在华为应用商城、安卓市场、豌豆荚等多个 Android 应用商店中进行搜索，可以发现现在已有很多阅读类软件，本文选取了一些具有代表性的阅读应用进行分析与研究。

掌阅 iReader 是独立垂直阅读领域的佼佼者^[12]，在下载量及用户评分方面均表现突出，深受用户好评。它不仅支持多格式电子书籍，还支持杂志、漫画的阅览，在线书城提供大量书籍资源，在社交化方面提供评论发帖等功能，但是会给用户推送很多广告、消息。QQ 阅读凭借其天然的品牌优势以及庞大的用户群体，成为用户阅读的佳选之一^[12]。QQ 阅读支持虚拟书架、朗读模式、阅读本地及在线书籍，在书库、精选等部分提供海量小说资源，但部分为付费小说，设置了书评广场等，而且可以根据用户阅读基因智能推荐书籍。

在国外电子书阅读市场中，不得不提起大名鼎鼎的亚马逊旗下的 Kindle 阅读器。2007 年 11 月，Kindle 阅读器一经推出便受到人们的热捧。Kindle 采用了电子墨水屏技术，在阅读时接近纸质阅读体验，在亚马逊的庞大的在线书城支持下，用户可以下载海量电子书^[13]。目前，亚马逊发布了支持 Android、iOS 的 Kindle 应用，可以实现与 Kindle 阅读器无缝阅读。ReadEra 作为一款优秀的阅读器，提供了

良好的阅读体验，支持按一定条件对文档进行分组，但是操作上不支持中文，而且对于中文电子书可能会显示乱码。Scribd 则以其优质的阅读资源获得人们的青睐，它可以为用户提供包括书籍、有声读物、杂志、新闻及文档库等多种资源。

论文^[14]设计并实现的安卓文本阅读器以保护视力为出发点，在实现文本阅读的基础上，通过安卓传感器，根据当前阅读环境实时亮度调整屏幕亮度，并加入了提醒休息功能。

论文^[15]设计并实现了基于语音提示和文字识别的无障碍移动电子书阅读器，针对视障人群，在控制操作方式和语音朗读方式方面创新，主要是简化操作流程，利用语音提示辅助操作，提供全文朗读、逐段朗读、逐句朗读、逐字朗读、基于文字图片识别的朗读等，为视障人群提供一个高效的阅读器。

1.3 研究内容及意义

1.3.1 研究内容

本论文旨在设计并实现一款书籍阅览与管理系统，该系统采用 C/S 架构，包括客户端和服务端两部分。客户端运行在 Android 设备中，用户可以直接使用，而服务器端则负责数据存储与处理服务。

本论文在调研 Android 应用市场中现有阅读类应用的基础上，结合用户阅读行为的实际需求，完成系统需求分析和系统设计。本系统提供了游客模式和账户登录两种模式，满足不同用户的多样需要。在此基础上，为用户提供书籍的阅读与管理、读书笔记、读书计划、书籍云备份、好书推荐等功能。

为了丰富系统 UI 展示形式、为用户提供更好的视觉效果及提高系统的稳定性、易用性，本论文深入研究了 Android 自定义 View 技术、目前 Android 主流图片加载框架以及动画技术等相关技术。在深入研究 View 的层次结构、View 的工作原理和 Android 事件分发机制的基础上实现自定义 View。接着对图片加载框架进行了对比，从源码角度研究了 Glide 并将其集成到本系统，提高了系统图片加载性能。通过对 Android 系统动画技术的研究，在本文设计的书籍阅览与管理系统中实现界面切换等动画效果。为了实现好书推荐功能，还调研了 Android 网络爬虫技术现状，

并通过 Jsoup 进行数据爬取工作。

在本系统的开发周期中，多次进行测试工作，设计并实施功能测试用例，利用 dumpsys、Monkey 等工具分别进行系统流畅度、稳定性的测试。

1.3.2 研究意义

本论文设计并实现的书籍阅览与管理系统，为用户提供了良好的阅读体验，对比现在已有应用，本系统的优势如下：

1. 本文提出并实现了一种基于网络状况与预约模式的备份策略。检测当前设备的联网类型及网络质量提醒用户，并为用户提供预约备份模式。该策略可以有效地避免用户在不知情的情况使用移动数据流量，而且预约模式的设计可以简化用户操作，提高用户体验。

2. 基于上述策略实现了本系统的书籍云备份功能，备份书籍操作快捷、方便，不仅为用户提供强大的云端存储服务，而且用户可以凭借账户实现在不同 Android 设备间的数据迁移。

3. Android 客户端提供了丰富、实用的功能。本系统在满足用户读书的基本需求上，通过读书计划功能解决了用户读书缺乏计划性的问题，支持添加读书笔记，通过高效的云备份功能解决用户存储空间不足的问题，好书推荐为用户推荐评分较高优秀书籍，为用户打造了一个方便、易用的读书平台。

1.4 论文组织

本论文在调研了国内外阅读类应用的现状之后，总结了该技术领域中现有应用的主要功能和不足之处。随后根据用户的实际需求，在全面、完整的系统需求分析之后，设计并实现了基于 Android 的书籍阅览与管理系统。

本文的章节安排如下：

第一章为绪论。本章首先介绍了论文的选题背景、国内外研究现状，接着阐述本文研究的内容和意义，最后梳理了论文的章节安排。

第二章为相关技术研究。为了更好地完成开发工作，保证系统性能良好、运

行稳定，本章对涉及到的部分 Android 相关技术进行了详细介绍，包括 Android 自定义 View、主流图片加载框架以及 Android 系统动画技术。

第三章为系统需求分析与系统设计。首先进行了功能需求分析和非功能需求分析，在系统需求分析的基础上设计了本系统的功能模块，同时对系统整体架构和框架模式分别进行了设计。

第四章为书籍阅览与管理系统的实现。详细阐述了系统的整体框架以及各个功能模块的实现。

第五章为书籍阅览与管理系统的测试。本章对系统进行了测试，首先介绍了测试环境，然后从功能测试、流畅度测试和稳定性测试等角度对本系统进行了测试工作。

第六章为总结与展望。本章对论文的工作进行了总结，并分析了本书籍阅览与管理系统中存在的不足，最后对下一步的改进和完善工作进行了展望。

1.5 本章小结

本章首先介绍了本论文的选题背景，阐述了国内外研究现状，接着梳理了本论文主要研究内容及意义，最后对论文的组织结构做了安排。

第 2 章 相关技术研究

2.1 Android 自定义 View 研究

View 是 Android 中的视图的呈现方式，Android 提供了一套图形用户界面 (GUI)，里面包含很多控件，但是很多时候系统提供的控件无法满足实际开发的需求，如果全部选用系统原生控件，则应用界面的同化度会比较严重。那么怎么样才能做出与众不同的效果呢？答案就是自定义 View，也可以叫自定义控件。自定义 View 技术需要多种 Android 技术的支持，主要涉及 View 的层次结构、事件分发机制和 View 的工作原理等^[16]。

2.1.1 View 及事件分发机制

在 Android 的设计中，控件分为 View 和 ViewGroup 两种。无论是 Android 自带控件还是开发者自定义的控件，View 都将是它们的基类。Android 采用了组合器设计模式设计了 ViewGroup 与 View。Viewgroup 继承自 View，而且 View 本身可以是单一控件，也可以是一组控件。View 的层次结构为树状结构，如图 2-1 所示。

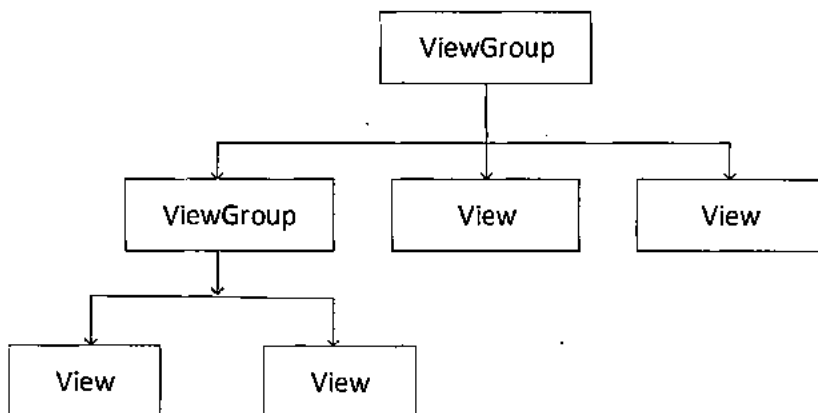


图 2-1 图形用户界面的组件层次

移动设备一大特色就是用户可以通过屏幕进行操作，当用户触摸设备屏幕时，就会产生一系列事件，产生点击事件 MotionEvent 对象。安卓中的 View 是以树形

结构存在的，可能存在多个 View，当用户触摸屏幕时可能会有多个 View 可以响应事件，为了解决应由哪个 View 消费点击事件这一个问题，Android 设计了事件分发机制。

每个 Activity 都会通过一个 Window 管理所要展示的视图。Window 是一个抽象类，是所有视图的最顶层容器，而 PhoneWindow 则是 Window 的唯一实现类，它管理了一棵视图树^[17]。DecorView 是 PhoneWindow 的一个内部类，PhoneWindow 的指示通过 DecorView 传递给下面的 View，Decor 对象被实例化后将作为应用的 ViewRoot（根控件）。在视图树渲染时，绘制将从 ViewRoot 的 performTraversals 方法开始。在 Android 中，每个 Activity 中用户界面组成如图 2-2 所示。

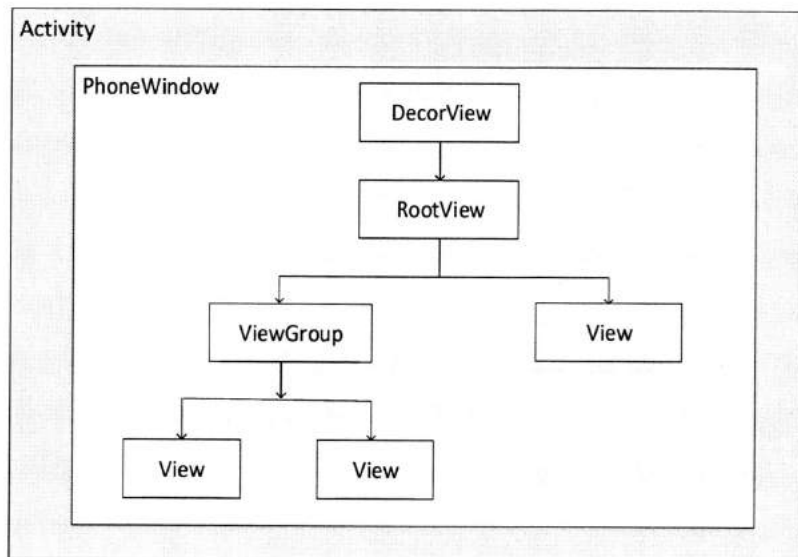


图 2-2 Android 用户界面组成

Android 的事件分发机制采用了责任链模式。当点击事件（MotionEvent）产生后，它的传递过程顺序为 Activity->Window->DecorView->ViewGroup->View，点击事件最先传递给 Activity，按顺序依次传递，如果该点击事件没有被处理，那么这个事件会反方向传递，如果传递至 Activity 也没有处理，则该事件将不会再被消费。在传递事件的时候，有三个方法起到了至关重要的作用。这三个方法分别是

dispatchTouchEvent 方法负责事件分发, onInterceptTouchEvent 方法负责事件拦截, 而 onTouchEvent 方法负责事件消费。

在 Android 事件分发过程中, 无论是 Activity、ViewGroup 还是 View, 都将调用 dispatchTouchEvent 和 onTouchEvent, 但是并非都会调用 onInterceptTouchEvent 方法。因为 Activity 和 View 所处位置, 不用进行事件拦截。因此, 在 View 和 Activity 中都未实现 onInterceptTouchEvent 方法。

2.1.2 自定义 View

实现自定义 View 需要了解其绘制流程^[18], 如图 2-3 所示, 接下来进行详细介绍。

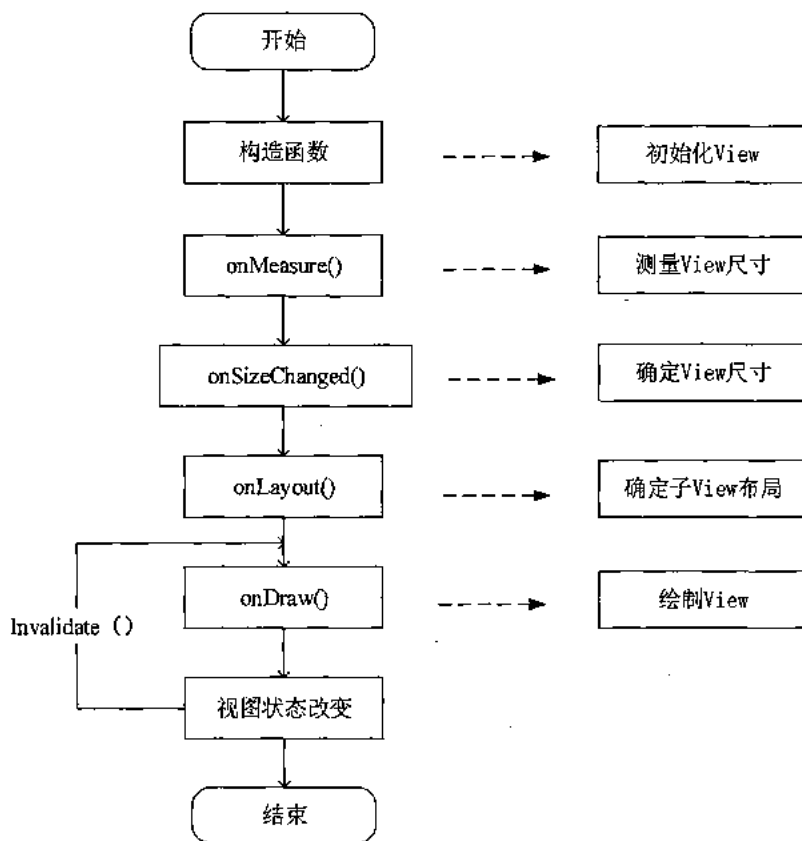


图 2-3 自定义 View 绘制流程函数调用链

1. 构造函数是 View 的入口, 在构造函数中可以实现初始化并获取自定义 View

属性。其中，开发者通常会自定义 View 的属性，不过也可以通过系统提供的属性进行定义。

2. 测量 View 尺寸

onMeasure 方法作为核心步骤，其作用是实现 View 尺寸测量。在视图树渲染时，绘制是从 ViewRoot 的 performTraversals 方法开始。首先调用 onMeasure 方法，该方法接收两个参数，分别是 widthMeasureSpec 和 heightMeasureSpec，这两个参数和各参数的测量模式（specMode）结合即可获的视图尺寸。测量模式有 3 种类型^[17]：

1) UNSPECIFIED 类型，表示开发者可以将视图设置为任意大小，一般不会使用该类型。

2) EXACTLY 类型，表示父视图希望子视图的大小由 specSize 决定。match_parent、具体数值对应的都是该类型，较常见。

3) AT_MOST 类型，表示子视图最多达到 specSize 中指定的大小，一般来说 wrap_content 对应这种类型，较常见。

3. 确定 View 尺寸

控件大小发生改变时调用 onSizeChanged 方法，其作用是确定控件的宽度和高度。所以在初始化时被调用一次，onSizeChanged 方法的主要调用是在构建视图之后，但在绘制之前完成，在这个时候，系统将计算该视图的大小，并通过回调函数 onSizeChanged 方法进行通知。

4. onLayout 方法用于确定子 View 的位置，在进行自定义 ViewGroup 时将会调用该方法。

5. 绘制内容

onDraw 方法将实现绘制内容，也就是我们真正关心的部分，将通过 Canvas 和 Paint 进行绘制。整个 View 是一张画布，即 Canvas，而 Paint 代表画笔，可以在画布上绘制各种图形和元素。Canvas 通过 Canvas.drawXXX 方法绘制图形等，其关键参数为 Paint，调用 Paint 的 set 方法可以对绘制内容的颜色和风格进行简单的设置^[19]。

在实现自定义 View 之后，一般将暴露接口，对外提供操作方法和监听回调函数，其作用是控制 View 的状态，或者监听 View 的变化等。当视图状态变化时，调用 Invalidate 方法，其作用是使 View 重新调用一次 draw 过程。

在本书籍阅览与管理系统中，为了给用户提供更好的视觉体验、满足用户操作习惯，自定义了很多控件，将在后续章节中介绍。

2.2 Glide 控件研究

良好的用户界面是用户考察一个应用是否值得使用的重要因素，为了提供用户更友好的用户界面，在应用中使用图片就显得尤为必要。如果只是实现获取图片并且展示在界面上这一功能，操作会比较简单，但是若考虑到系统的整体性能，那就需要考虑如何更加高效地加载图片。在 Android 中，图片在内存中是以 Bitmap 的形式存储，占用内存资源较大^[20]，图片的大小、数量等不确定因素都会对系统的性能产生影响。当需要在界面上加载很多图片时，情况就变得复杂起来，将涉及到内存资源占用过大、分发、图片显示错乱、图片缓存机制请求优先级处理等问题。

为了更好地加载图片，许多优秀的 Android 图片加载框架应运而生，随着这些框架的不断发展与完善，越来越多的开发者倾向于借助开源框架来实现图片处理相关功能^[21]。在这些开源框架中，佼佼者便有 UIL(Universal-Image-Loader)、Picasso、Fresco 以及 Glide。这四种框架均可实现多种资源的加载与显示，可自由配置解码器、显示器、进行多级缓存等，根据系统性能调整初始化配置，具有较高的自适应度和良好的兼容性，但是各自又有其特色，在使用这些框架时应结合系统需求进行选择。

UIL 可以算作最早的被广泛使用的图片加载框架，但是现在作者已停止对其进行维护，所以不再推荐使用。Fresco 是 FaceBook 于 2015 年 3 月开源的图片加载框架，Instagram App 中就使用该框架。Picasso 是 Square 公司发布，由许多开源项目的贡献者完成。它针对下载图像和缓存进行了优化，并且通过使用方法链，有助于开发者直观、方便的进行编程。Picasso 开源库可以帮助我们加载一个 Android

资源图像、文件图像和网络图像。

Glide 是 Google 发布的一个高效的图片开源框架，其作者为 Bumptech，被应用于众多 Google 官方应用中^[22]。Glide 与 Picasso 很相似，API 几乎完全一样，它完全遵循 BSD、MIT 以及 Apache 2.0 协议发布。Glide 提供了高性能、可扩展的图片解码管道（decode pipeline）以及自动的资源池技术，支持获取、解码和显示图像和动画 GIF。Glide 的主要目的是尽可能实现平滑和快速地滚动任何类型的图像列表^[23]。

关于以上四个经典的图片加载框架特点分析如下表 2-1 所示。

表 2-1 典型图片加载框架特点分析

图片框架	特点
Universal-Image-Loader (UIL)	支持下载进度监听，可以在 View 滚动中暂停图片加载，实现多种内存缓存算法；配置较繁琐；不支持 GIF 图片加载；设计了一套独立的缓存机制，该缓存机制没有良好适配 HTTP 的缓存机制。
Glide	Glide 不仅是一个图片缓存，它支持 Gif、WebP、缩略图，甚至是 Video；支持优先级处理；其生命周期与 Activity/Fragment 生命周期一致，便于管理；Glide 默认通过 HttpURLConnection 获取数据，但是也可以通过 okhttp、Volley 进行网络通信；内存友好。
Picasso	支持图片缓存中使用的监控，包括缓存命中率、已使用内存大小、节省的流量等；支持优先级处理；手机切换到飞行模式或网络类型变换时会自动调整线程池的最大并发数。
Fresco	两个内存缓存加上 Native 缓存构成了三级缓存；支持流式；对多帧动画图片支持更好；底层设计 C++。

基于以上对图片加载四大框架的分析与对比，在充分考虑本论文所设计的书籍阅览与管理系统的功能需求与非功能需求，本论文最终选取了 Glide 开源框架实现系统中加载图片、显示图片的功能。在深入研究 Glide 框架的源码后，理解了 Glide 框架的设计原理，学习其设计模式、编码风格与使用方法，这有助于本系统的开发工作。

Glide 开源框架总体设计结构如图 2-4 所示，框架整体分为四层结构。

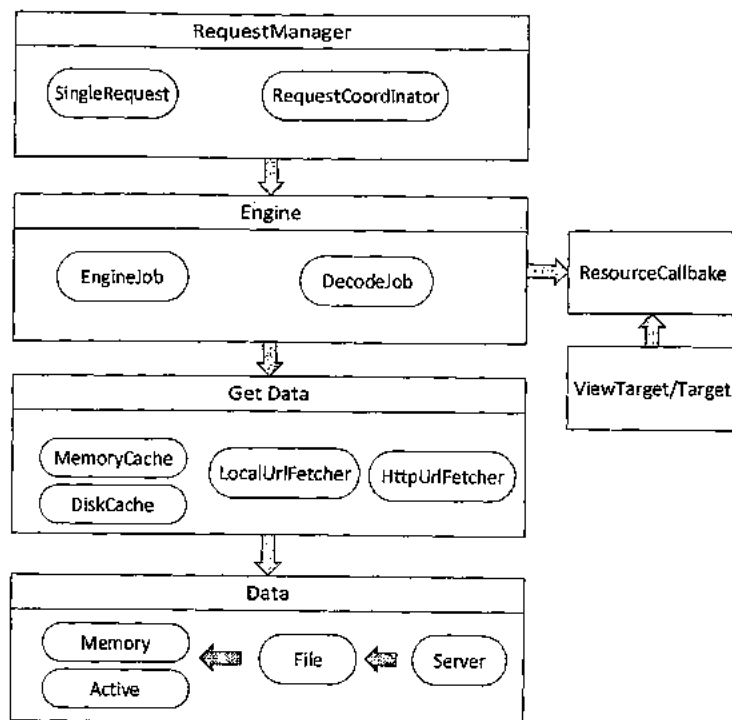


图 2-4 Glide 开源框架总体设计结构

作为第一层的 RequestManager 层，其主要作用为生成和管理请求。当 Glide 收到加载及显示资源的任务后，首先创建了 buildRequest 以创建请求，如果定义的是 thumbnail 请求，那么将创建 ThumbnailRequestCoordinator 请求，该请求包含 FullRequest 和 ThumbnailRequest，否则就创建 Request。

在执行 RequestManager 层后接着执行 Engine 层。Engine 主要作用是任务创建、发起、回调，管理存活和缓存的资源，其中 EngineJob 将通过线程池调度 DecodeJob，执行添加、移除资源回调等，并在 Engine 任务完成时回调主线程。作为调度任务的核心类的 DecodeJob 则实现了 Runnable 接口，解析来自缓存或者原始的资源，应用转换动画和 Transcode。并且，还将调用 Generator 执行加载数据资源的任务，数据资源加载完毕，将接着调用 DecodeJob 的 onDataFetcherReady 方法对资源进行处理。

Engine 可以通过多种数据源尝试进行资源获取，数据源包括内存缓存、磁盘

缓存、下载器从网络中下载，获取资源到后，通过 Transformation 进行图片处理后返回给 Target/ViewTarget。Target 是资源显示器，用于显示或操作资源，如 ImageView 等。Glide 加载并显示资源的工作流程如图 2-5 所示。

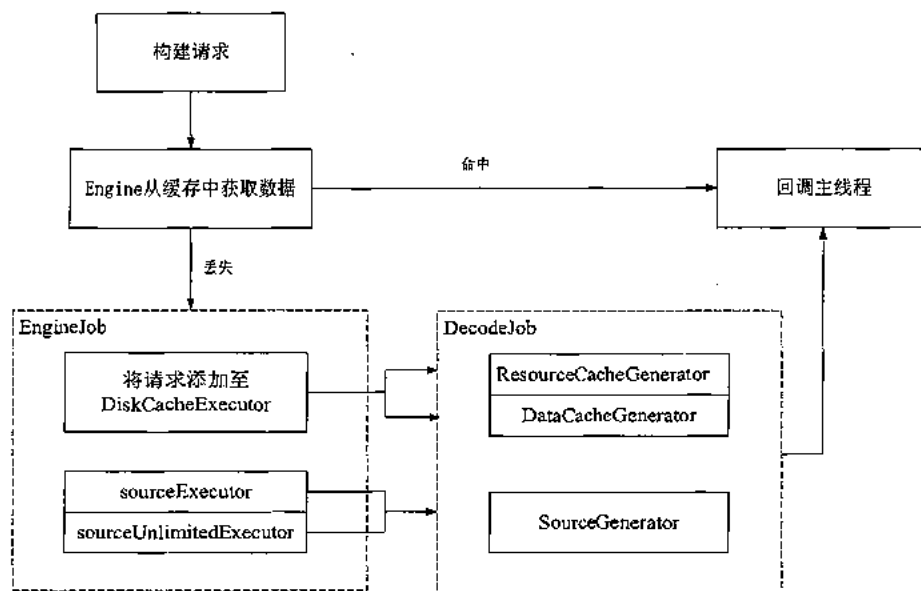


图 2-5 Glide 显示资源工作流程图

Glide 默认使用 RGB_565 进行解码^[24]，在 Android 中对图片进行解码主要有两个方法：ARGB_8888（每像素 4 字节存储）和 RGB_565（每像素 2 字节存储）。其中 ARGB8888 解码的图片质量较高，但 Glide 的解码方式使得其内存占用相对较小，如果开发者希望图片质量更高，也可以更改解码方式，通过调用 setDecodeFormat 方法设置。

Glide 通过使用默认的内存和磁盘缓存以达到尽量少的进行网络请求的目的。因此，Glide 中设计了多种缓存策略并且提供了方法设置内存和磁盘缓存行为。skipMemoryCache 方法决定是否使用内存缓存，其参数为布尔值，若设置为 true 将跳过内存缓存。diskCacheStrategy 方法决定 Glide 磁盘缓存的行为。该方法的参数是枚举值，各个值的意义如下。

- ❖ DiskCacheStrategy.NONE：禁用磁盘缓存；
- ❖ DiskCacheStrategy.SOURCE：仅对原有全分辨率的图像进行缓存；

❖ DiskCacheStrategy.RESULT: 仅对降低分辨率或者是完成转换之后的最终图像进行缓存;

❖ DiskCacheStrategy.ALL: 缓存所有版本的图像(默认行为);

在本系统中集成 Glide 框架,需要在 AndroidStudio 上 build.gradle 文件中添加依赖,并且在 AndroidManifest.xml 中添加网络权限。

```
dependencies {
    compile 'com.github.bumptech.glide:glide:3.5.2'
    compile 'com.android.support:support-v4:22.0.0'
}
```

```
<uses-permission android:name="android.permission.INTERNET"/>
```

在使用 Glide 时,最好在主线程中使用,因为它需要 Context,否则可能会导致内存泄露或者严重的 Crash 现象,使用 Glide 加载图片,其生命周期会和 Activity、Fragment 的生命周期一致,在 onStop 方法中自动暂停请求行为,在 onStart 方法的时候重新启动请求,动画也会在执行 onStop 方法的时候停止,这样可以降低系统电量消耗。此外,当设备的网络状态发生改变的时候,所有失败的请求会自动重启,保证数据的正确性。

2.3 Android 系统动画技术研究

动画几乎已成为一个优秀的应用中不可缺少的一部分,它展现出一种更友好的交互方式,在 Android 等众多操作系统中动画的使用越来越广泛。Android 系统作为一个性能优越的开源平台,提供了多种资源进行界面设计,其中动画(Animation)丰富了 Android 应用的表现形式,为用户呈现更好的视觉效果。在应用开发中,经常会使用动画技术。最初,Android 提供了逐帧动画和补间动画这两种动画,Android 3.0 (API Level 11)之后,又提供属性动画。

1. 逐帧动画(Frame Animation)

逐帧动画的原理是首先定义动画图片资源,然后逐张依次地显示静态图片资

源，以此达到动画效果^[25]。这是根据“视觉暂留”的机理进行设计的，给予用户动画错觉。

在定义动画图片资源时，可通过 XML 资源文件进行定义，其元素标签为 `<animation-list.../>...</animation-list>`，子元素标签为 `<item... />`，与此同时在 Java 代码中通过 `AnimationDrawable` 即可实现逐帧动画。在该动画中，最开始定义的静态图片将决定动画展示内容^[26]。

在使用该动画时，需要注意的是图片不易过多、过大，否则容易导致 OOM 问题。

2. 补间动画(Tween Animation)

Android 目前支持的补间动画有多种，如下表 2-2 所示。在补间动画中，有“三要素”：开始帧、结束帧、动画持续时间。在开发者定义三要素之后，Android 计算出需要在开始帧、结束帧之间插入的帧数，并计算所有补入帧的图像^[27]。当所有帧都被计算出来后，依次展示每一帧，原理类似逐帧动画。另外，Android 提供了 `Interpolator` 接口，根据特定算法以达到调节动画的变化速度的效果。`Interpolator` 的主要的实现类有 `LinearInterpolator`、`AccelerateInterpolator`、`DecelerateInterpolator` 等，通过选取合适的类即可实现动画匀速、加速、减速、按正弦曲线等多种速率进行变换^[28]。

表 2-2 Tween Animation 4 种基本动画

类	说明
<code>AlphaAnimation</code>	设置动画透明度。透明度从 0 到 1 变化
<code>ScaleAnimation</code>	设置动画的大小缩放。以 X、Y 轴的缩放参数设置缩放比，取值 <code>[0.0f-1.0f]</code>
<code>TranslateAnimation</code>	设置动画的位移变化。以 X、Y 轴坐标表示
<code>RotateAnimation</code>	设置动画的旋转角度。需 <code>pivotX</code> 、 <code>pivotY</code> 指定旋转中心的坐标

在使用补间动画时，仅涉及两个关键帧，由此看出其优点是可以节省空间，效率较高且使用方便。

3. 属性动画(Property Animation)

Android 3.0(API Level 11)之后,在已有两种动画的基础上 Google 又提供了一种崭新的动画类型,即属性动画。在深入研究逐帧动画、补间动画后,总结了不足之处。首先,补间动画的作用对象是 UI 组件,不能对非 View 对象执行动画效果,显然这限制了动画在某些情形下的使用。其次,由于并非真正的修改 View 的属性,所以不具备交互性,当某个元素发生动画后,其响应事件的位置依然在动画执行前的坐标位置。最后,呈现的动画效果有限,无法实现复杂的动画效果。基于以上原因,Android 提出了属性动画,进一步丰富 Android 动画资源,让应用呈现更吸引人的视觉效果。

属性动画不再局限于 View 对象,它可以对任何对象的属性执行动画,这将大大加强了属性动画的性能,通过属性动画的使用,可以使得开发者实现更加绚丽的效果。

属性动画的工作机制是在一定时间间隔内,通过不断对对象的属性值进行改变,并不断的重绘对象而形成动画,从而实现该对象在该属性上的动画效果。属性动画需要定义几个要素,Duration、Interpolator、repeatCount、repeatMode、ordering 和帧刷新频率,通过对要素的设置可以实现动画效果。Interpolator 为插值器,其工作原理^[17]如下图 2-6 所示。

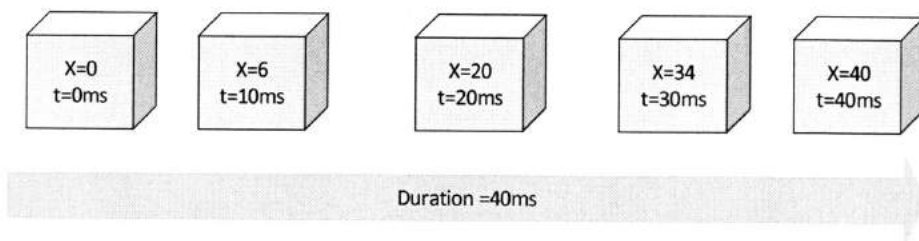


图 2-6 Interpolator 工作原理示意图

在 Android 开发中经常使用属性动画,下面将介绍属性动画的使用步骤,具体如下:

- (1) 使用 ValueAnimator 或者 ObjectAnimator 创建动画对象;
- (2) 设置 Animator 对象的属性,必须有对应的 get 方法和 set 方法,内部会通过 Java 反射机制来调用 set 方法修改该对象的属性值;

- (3) 为 Animator 对象设置事件监听器, 可以监听动画开始与结束、动画执行重复动作等;
- (4) 若存在多个动画展示, 可以使用 AnimatorSet 进行动画组合;
- (5) 调用 Animator 对象的 start 方法启动动画;

2.4 本章小结

本章对书籍阅览与管理系统开发中所需的部分相关技术进行了介绍。首先主要从 View 的层次结构、View 的工作原理和 Android 事件分发机制等方面介绍了 Android 自定义 View 相关技术。接着对 Android 中主流图片加载框架进行了分析与对比, 从源码角度深入分析 Glide 框架并总结 Glide 特点。最后阐述了 Android 系统动画技术, 对 Android 中三种动画进行了详细介绍。

第3章 系统需求分析与系统设计

3.1 系统需求分析

在整个软件生命周期中，需求分析扮演着重要角色。软件需求分析的目的是明确定义从系统外部能发现系统所具有的满足于用户的特点、功能及属性等，全面、完整、精确地描述系统是什么样的，这在整个软件开发过程中都将起到十分重要的作用^[29]。对于开发者来讲，必须要做好软件需求分析，软件需求分析的质量高低会对软件开发成功与否产生很大的影响^[30]。所以，在进行系统设计之前，本论文将先进行系统需求分析。

3.1.1 功能需求分析

经过对国内外阅读软件的对比与分析，调研用户的实际要求后，总结了用户的功能需求。移动阅读应用作为满足用户碎片化阅读的重要工具，首先需要满足用户阅览与管理书籍、读书笔记的基本需求，并且需要一些个性化设置。其次，用户在读书时表现出缺乏计划性，在了解目前优质书籍资讯时也比较困难。最后，当存储大量电子书时，设备本地存储空间不足，而且希望在不同设备间可以快速实现数据迁移。

本论文所提出的书籍阅览与管理系统，针对用户需求进行了功能需求分析，其主要功能有书籍阅览、书籍管理、制定读书计划、书籍云备份、读书笔记、好书推荐等六大功能，为用户提供了高效的阅读服务及良好的阅读体验。而且，除上述功能还需要登录模块，用户可以选择账户模式或者游客模式登录系统。另外，还要设置菜单栏对用户信息、返回首页、使用教程、注销账户等功能进行集中管理。

本书籍阅览与管理系统主要分为基本功能、系统管理功能两部分，接下来将对其分别进行功能需求分析。首先是客户端基本功能需求分析，该部分包括满足用户读书行为需求等，基本功能用例图如图 3-1 所示。

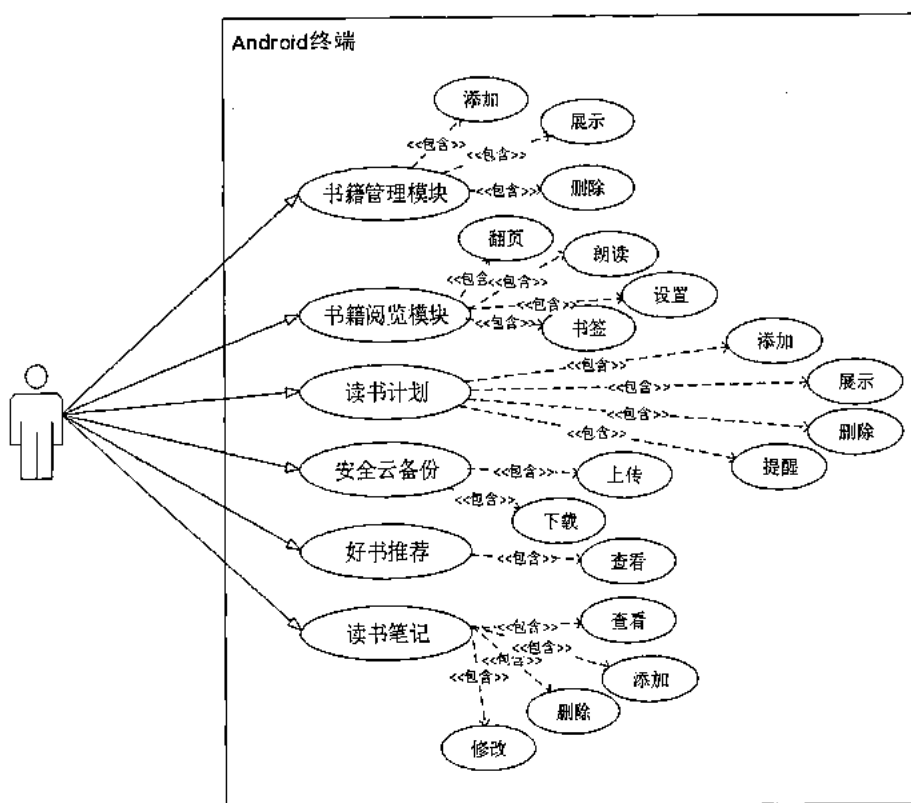


图 3-1 客户端基本功能用例图

1. 书籍管理功能：为了方便用户查看电子书籍，提供书籍展示栏。用户可以扫描本地设备，选择书籍导入到书籍展示栏。在书籍展示栏界面，用户可以删除书籍，点击某本书籍进入阅读界面。

2. 书籍阅读功能：将实现解析、显示书籍，支持前后翻页阅读、添加查看书签、有声朗读，并且可以进行阅读基本设置以获取良好的阅读体验。在设置菜单中，提供多种操作来满足用户读书需求。设置主要包括调节字体、调节亮度、跳转、夜间模式。

3. 读书计划功能：在读书计划界面显示当前用户所有的阅读计划，在添加界面用户可以设置书名与时间完成添加计划。在读书计划界面用户可以删除读书计划，当到达用户设置的时间时，将在系统通知栏提醒用户。

4. 书籍安全云备份功能：用户可以扫描本地设备，选取书籍后上传至服务器

端，也可以查看服务器端所有书籍，支持下载书籍至本地设备。

5. 读书笔记功能：用户可以随意添加读书笔记，在笔记展示界面可以查看读书笔记的标题、内容、添加时间，同时支持用户删除笔记。

6. 好书推荐功能：主要提供优秀书籍资讯信息，用户可以了解到当前比较热门的书籍信息，并且可以搜索自己感兴趣的书籍。

下面对系统管理功能需求进行详细分析，管理功能用例图如图 3-2 所示。

1. 登录功能：用户在使用本系统时，可以选择游客模式，无需注册账户。若用户选择账户登录模式，则需注册账户，在进行账户注册时需要依次填写用户名、邮箱、密码、再次输入密码、选择头像，在缺少任意信息给予提醒，完成注册后即可通过账户名、密码登录；同时支持找回密码，用户可自行通过发送邮箱重置密码。
2. 菜单栏功能：此功能包括显示用户信息（包括头像、用户名、邮箱）、首页、注销用户等，其中首页功能为导航功能，其作用是让用户可随时返回系统主界面。

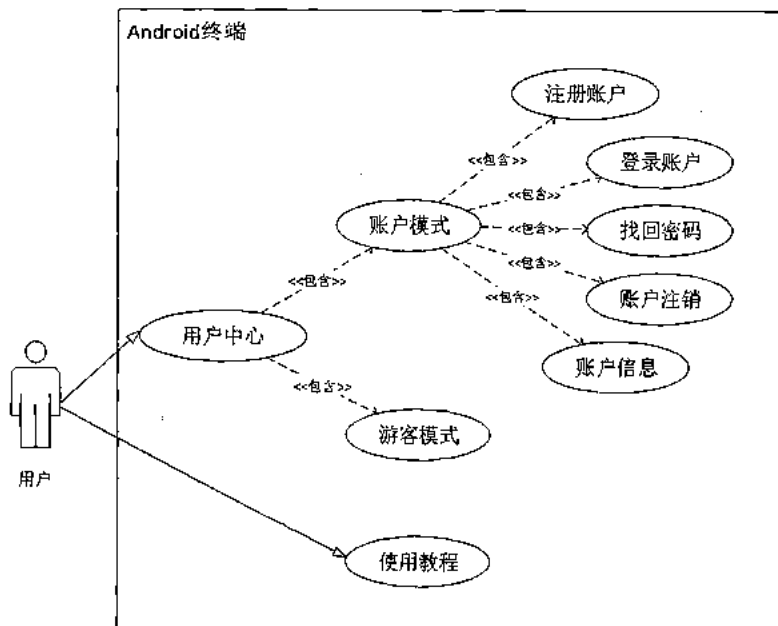


图 3-2 系统管理用例图

3.1.2 非功能需求分析

非功能需求（NFR）可以被定义为描述系统的属性、特性及约束集合，它是决定软件系统优劣的重要因素。一款优秀的软件在满足功能需求的基础上，也必须实现非功能需求^[31]。本文主要从以下几个方面探讨书籍阅览与管理系统的非功能需求。

1. 易用性分析

系统的易用性与可用性有所不同，良好的易用性能够增加用户好感度。在设计该书阅览与管理系统时，应该站在用户的角度，充分考虑用户的认知和操作习惯，明确功能设置，做到界面友好易用，使用方便。

2. 稳定性分析

在一定的条件下运行系统，系统必须能够在长时间内实现系统所具备的所有功能。确保系统拥有较强的稳定性是一个软件的必备的基本特性，若系统缺乏稳定性，它将很快被用户淘汰。在本系统开发过程中，基于完整全面的需求定义分析后设计、实现了系统功能，这有利于系统稳定性的提高。另外，对于用户错误输入进行处理，提高系统鲁棒性。在系统完成后，设计测试用例测试系统功能，并对流畅度、稳定性等进行测试；

3. 可扩展性分析

若一个系统具有良好的可扩展性，就意味着在添加或者修改系统功能时，能够更容易地在原有系统基础上进行开发。在本论文的书籍阅览与管理系统的设计与实现过程中，整体编码充分考虑里氏替换等原则^[32]，同时恰当的使用单例模式、建造者模式等设计模式进一步提高系统的可拓展性。

3.2 系统设计

3.2.1 系统架构设计

本论文设计的书籍阅览与管理系统采用了 C/S 架构。C/S 架构全称是 Client/Server 架构，是一种被人广泛使用的软件系统体系架构^[33]。在本系统中客户

端是运行在 Android 终端设备上的应用程序，用于与用户交互，服务器端主要是提供数据存储与处理功能，将系统任务适当分配至客户端和服务端，降低了系统开销^[34]。系统中的数据小部分存储于 Android 本地设备，由于 Android 终端设备硬件资源的限制，如计算能力有限、存储空间较小等限制，所以大部分数据存于服务器数据库。客户端与服务器之间通过网络进行交互，进行数据传输。

系统的总体架构图如图 3-3 所示：

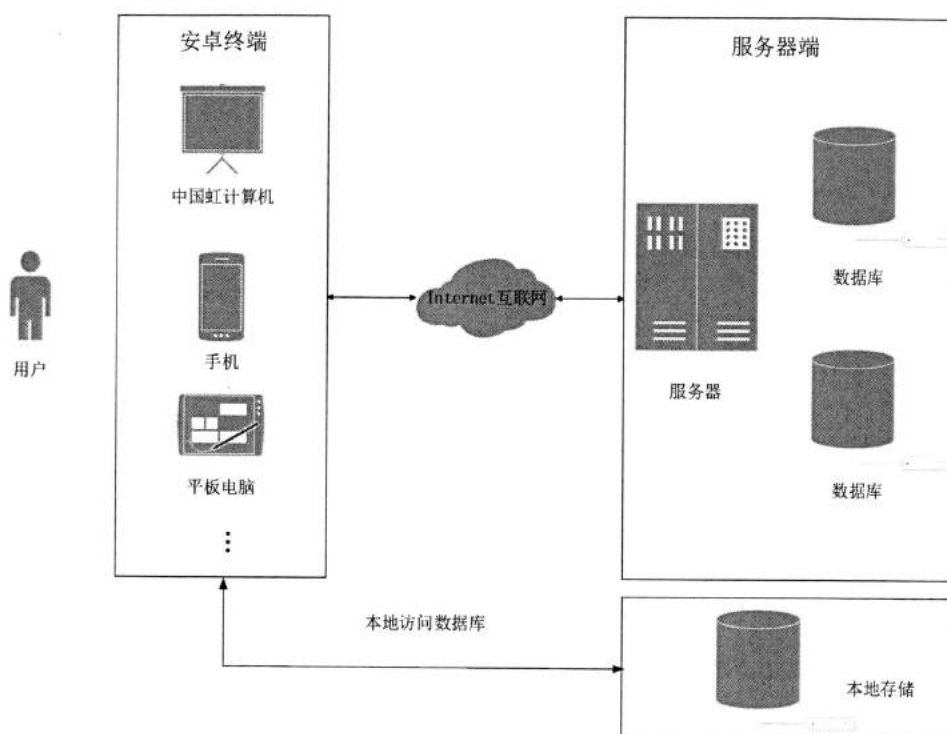


图 3-3 系统的总体架构图

3.2.2 系统框架模式设计

该书籍阅览与管理系统在开发时框架模式采用了 MVC，MVC 结构如图 3-4 所示。早在 1970 年 MVC 的概念就被提出了，MVC 即 Model-View-Controller，主要是通过将数据模型和视图分离，控制器连接两者以解耦这种方式，使得系统拥有某些良好的特性^{[35][36]}。使用 MVC 框架开发的软件，可以有效地降低开发和维护成本，方便系统后期维护。为了将 MVC 应用到系统中，本系统将采用 XML 文

件进行系统视图的描述与定义，将业务逻辑处理封装在模型层，而 Activity 将承担起控制器的作用。

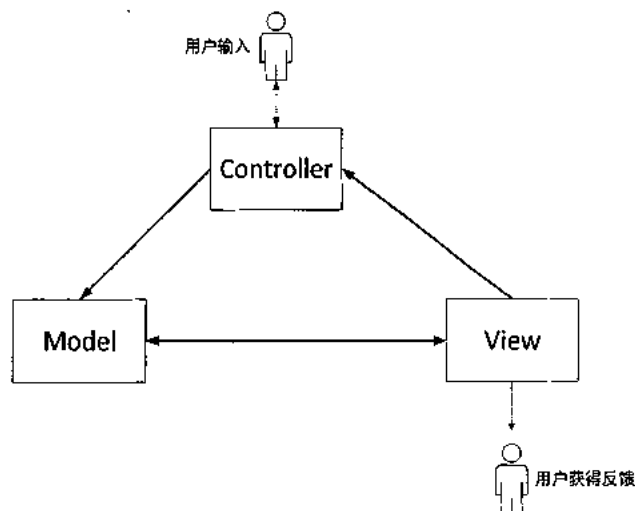


图 3-4 MVC 结构图

3.2.3 系统主要模块设计

根据前面的系统需求分析，充分考虑用户使用习惯，对该书籍阅览与管理系统进行功能模块设计，包括登录模块、主界面功能模块和用户中心模块。在本文的设计中，主界面的功能模块包括读书模块、读书计划、读书笔记、书籍云备份、好书推荐，系统主界面功能模块设计如图 3-5 所示。另外，部分模块还包含数个功能子模块。

登录模块：用户点击进入该系统，首先进入登录模块。在该界面中，用户可以通过账户、密码登录系统。该模块还设计了验证码功能，为登录接口增加第三个参数，目的是防止恶意暴力破解登录，也避免不间断的登录尝试以减轻服务器端的压力。本模块支持通过注册用户功能注册账户，注册时的邮箱可以作为找回密码的媒介。在找回密码界面输入邮箱就可以找回密码，以防止用户密码遗失。在该模块中，当用户在没有连接网络时希望使用该系统，可以选择游客模式，无需账户直接进入主界面，但是在该模式下无法使用云备份等功能。通过登录模块之后即可进入系统主界面。

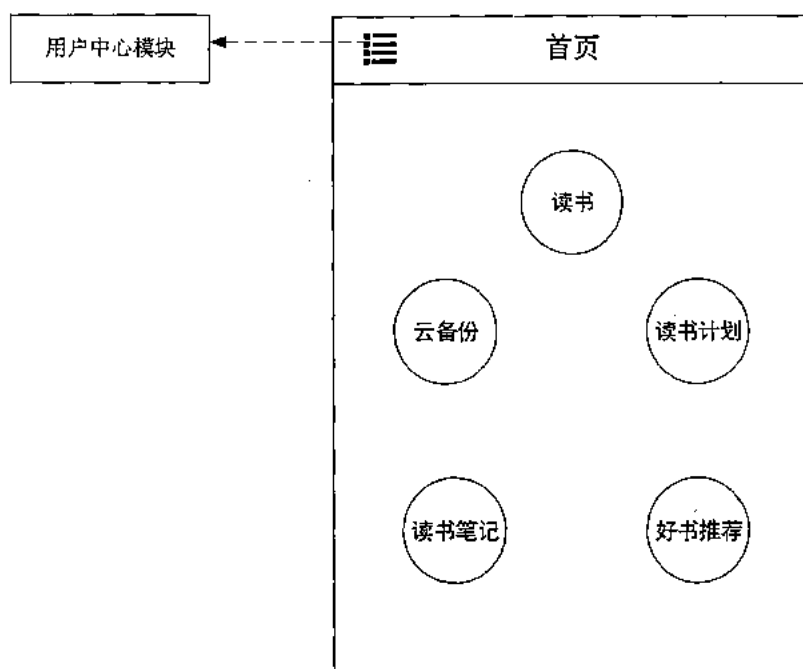


图 3-5 系统主界面功能模块设计图

读书模块：该模块包括书籍管理界面与书籍阅览界面。在主界面中点击选读书即进入书籍管理界面，在书籍管理界面中，包括书籍展示栏、添加书籍、返回主界面按钮。用户点击添加书籍可以扫描设备 SD 卡，为了方便用户查找书籍文件，实现筛选功能，在 SD 卡目录界面只显示文件夹和 TXT 文件，用户自主选择一个或多个文件后加入点击添加按钮将书籍加入展示栏。长按某书籍后，用户可以选择删除书籍。

在书籍展示栏中，单击某本书进入书籍阅览界面。在该界面，可以阅读书籍、返回书籍管理界面、进行阅读设置等。设置菜单项分别为字体、亮度、跳转、书签、夜间模式、有声朗读等。单击书签选项，可以添加书签和查看我的书签。

读书计划：在主界面中点击选择该模块即进入读书计划界面，进入该模块第一个子界面展示已有读书计划。单击添加按钮进入添加界面，编辑书名并设置时间之后单击保存按钮完成添加。在展示计划界面滑动计划，单击删除图标完成删除。

读书笔记：在主界面中点击选择该模块即进入读书笔记界面，首先是子界面展示已有读书笔记，滑动某读书笔记可以选择删除。单击添加按钮进入添加界面，编辑标题并设置时间内容之后单击保存按钮即可。

好书推荐：在主界面中点击选择该模块即进入好书推荐界面，以列表形式展示热门书籍，包含书名、作者、评分等信息，单击喜欢的书籍进入书籍详情界面，在该界面将展示书籍内容及书评。在界面中有搜索框，可以搜索书籍。

书籍云备份：在主界面中点击选择该模块即进入书籍云备份界面。该界面展示当前用户备份的所有书籍名称，用户点击书籍将弹出下载对话框。点击上传按钮将浏览 SD 卡文件，选择希望上传的文件进入上传模块。在备份书籍时，采用了基于当前网络状况与预约模式的备份策略。

用户中心：该模块以侧滑工具栏的形式实现，从上至下展示用户信息、首页导航、使用教程、注销用户，用户信息包括用户头像、用户名、用户注册邮箱。当用户以游客模式登录时，显示默认头像、用户名及邮箱。本功能界面设计效果如图 3-6 所示。

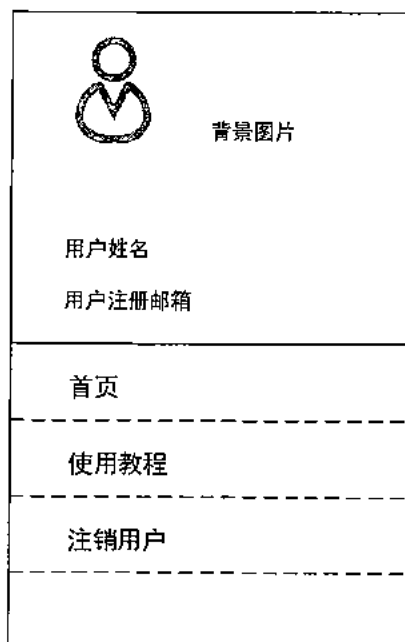


图 3-6 用户中心模块设计图

3.3 本章小结

在本章中,首先从功能需求和非功能需求的角度对书籍阅览与管理系统进行了系统需求分析,随后对系统的整体架构、框架模式进行了设计,最后根据需求分析对系统进行功能模块划分,并对每个模块进行了详细的规划与设计。

第 4 章 书籍阅览与管理系统的实现

本论文设计的书籍阅览与管理系统是基于 Android 操作系统进行开发的，由 Java 语言编写，开发平台则选择了 Google 为 Android 提供的官方 IDE 工具 Android Studio 2.2.3，JDK 版本为 JDK 1.8。本章将详细地介绍本系统各个模块的具体实现过程。

4.1 系统界面框架的实现

该系统界面框架包括主界面和侧滑菜单栏两部分，实现效果分别如图 4-1 和图 4-2 所示。在系统主界面实现了功能选择逻辑，功能包括读书、云备份、读书计划、读书笔记、好书推荐，当用户点击图标即可进入对应模块，而在菜单栏，则显示了基本用户信息、返回首页、注销等功能，基本用户信息包括用户头像、用户名、注册邮箱。

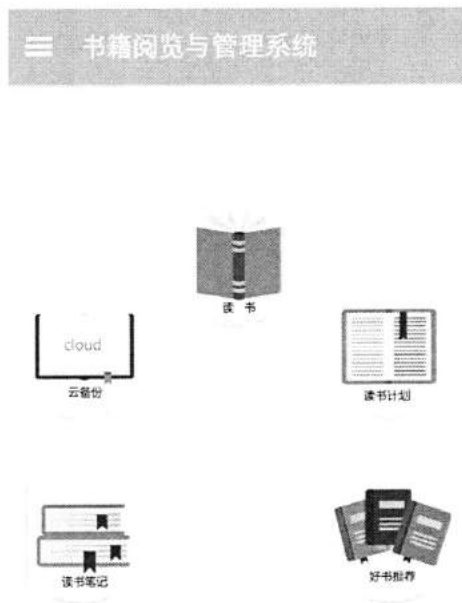


图 4-1 系统主界面效果图



图 4-2 侧滑栏菜单效果图

在系统主界面，由 ToolBar 实现标题栏，而主界面的功能选择部分，本文结合了 Fragment 实现。Google 在 Android 3.0 开始引入 Fragment，其特点为模块化 UI、组装更灵活，内存开销较低^[37]。本文中通过工厂模式来生成 Fragment，首先创建 MasterFragment 继承 Fragment，完成布局并且实现每个 Fragment 对应的处理。在 MainActivity 实现 OnFragmentInteractionListener 接口，通过该接口处理 Fragment 与 Activity 的互动事件，用户点击不同功能就将跳转到相应的功能模块。

在侧滑菜单栏中，菜单项功能包括用户信息（包括头像、用户名、邮箱）、返回首页、使用教程、注销用户。自从谷歌推出 Android Material Design Library 后，其受到众多开发者青睐追捧。在本系统中，为了实现良好的侧滑交互体验，使用了该库中的 DrawerLayout 控件与 NavigationView 控件共同实现侧滑菜单栏功能。在使用时，需要在 build.gradle 文件中添加依赖：

```
compile 'com.android.support:design:23.0.1'
```

首先，在 XML 布局文件中使用 DrawerLayout 与 NavigationView 进行界面设

计。该模块根布局为 `RelativeLayout`，其又包含两个子布局：`AppBarLayout` 和 `DrawerLayout`，`DrawerLayout` 中包含 `FrameLayout`、`NavigationView` 抽屉布局，其中内容布局代码要写于抽屉布局之前。在进行 `NavigationView` 属性定义时，需特别注意定义以下属性：

```
android:layout_gravity="start"
app:headerLayout="@layout/drawer_header"
app:menu="@menu/drawer"
```

`android:layout_gravity="start"`，表示该 `View` 将从左边滑出。

`app:headerLayout` 引用一个 `layout` 文件，作为导航菜单顶部的 `HeaderView`，用于显示用户信息。在自定义 `drawer_header` 文件中，使用 `ImageView` 控件显示头像，`TextView` 显示用户名、邮箱。

`app:menu` 引用一个 `menu` 文件，设置导航菜单内容的 XML 布局。在 `menu` 文件夹下定义 `drawer` 文件，设置 `MenuItem` 选项的菜单，菜单项分别为返回首页、使用教程、注销。

在 `MainActivity` 类中，通过 `setNavigationItemSelectedListener` 方法为菜单项添加监听器，重写 `onNavigationItemSelectedListener` 方法，根据选中的 `MenuItem`，可以分别进行相应操作。例如点击注销，则调用 `logOut` 方法退出，同时清除本地缓存的 `CurrentUser` 对象。菜单顶部的信息展示，则是调用 `UserData.getObjectByKey` 方法，以 `username`、`email`、`photoId` 作为查询字段从服务器端获取数据显示在相应控件。对于头像图片，使用 `Glide` 加载显示。

4.2 系统功能模块的实现

4.2.1 读书模块的实现

读书模块是该系统的核心模块之一，用于满足用户读书、管理书籍的需求，主要分为书籍展示与删除、导入书籍、阅览书籍这三个功能。

1. 本模块数据存储的设计

在 Android 应用中, 不可避免的会遇到数据存储问题。Android 系统提供了多种数据存储方案, 将数据保存在各种存储介质中: 数据存储方式主要有文件存储、SQLite 数据库、网络存储等, 其特点及适用范围各有不同^[38]。

当数据量较小且格式简单时, 考虑使用 SharedPreferences。SharedPreferences 将数据保存在应用软件的私有存储区, 只能被对应的应用读取^[39]。本模块中加载书籍浏览界面时的当前阅读进度、亮度、字体大小、夜间模式等配置信息就是使用的 SharedPreferences 进行存储。

对于书籍列表、书签等数据量较大的数据, 本模块采用 SQLite 数据库进行存储, SQLite 是 C 和 C++ 实现的一个轻量级关系型数据库, 但是 Android 在 Framework 层封装了 Java 接口, 使得它被链接到程序中, 成为应用的一部分^{[17][40]}。在本系统中, 使用了 LitePal 框架对数据库进行操作, LitePal 采取的是对象关系映射(ORM)模式, 每一张表都应该对应一个模型(Model)^[41]。所以分别创建了继承自 DataSupport 的模型类 BookList、BookMark, 模型类中的每一个成员变量对应表的每一列, 并在 assets 目录下的 litepal.xml 文件中, 在<list>标签下加入各个模型类的声明。在系统中, 可以直接调用 save 方法存储数据, find、findAll 等方法查询数据, update、updateAll 等方法修改数据, delete、deleteAll 等方法删除数据。在使用时, 需要先初始化数据库, 比如在获取书籍列表模型类的数据时, 主要代码如下:

```
SQLiteDatabase db = Connector.getDatabase();
bookLists = new ArrayList<>();
bookLists = DataSupport.findAll(BookList.class);
```

为了实现书籍展示栏功能, 创建了书籍列表信息模型类 BookList, 其成员变量有 id (书籍 ID)、bookname (书籍名称)、bookpath (书籍路径)。

为了实现添加和查看书签功能, 创建了书签模型类 BookMark, 其成员变量有 id (书籍 ID)、markbegin (书签开始位置)、marktext (书签内容)、marktime(书签时间)、bookpath (书籍路径)。

2. 本模块功能的实现

(1) 书籍展示与删除功能

在书籍展示栏，考虑到移动终端尺寸问题，书籍以网格的形式展示，每行显示两列，实现效果如图 4-3 所示。ListView 与 GridView 都可以实现列表展示，其主要区别是 ListView 一般只显示一列，但是本模块中书籍是以多行、多列展示的，因此采用 GridView 控件实现书籍展示。

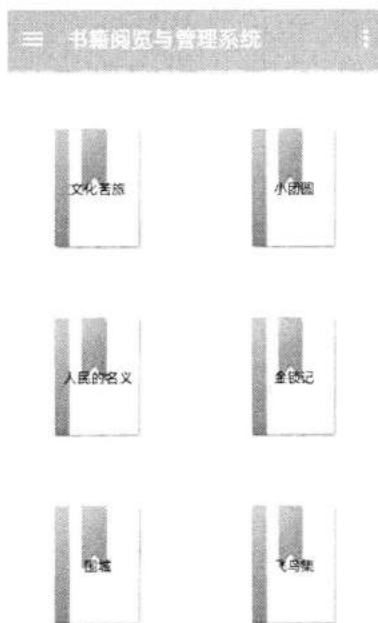


图 4-3 书籍展示栏效果图

书籍展示栏界面主要由 Toolbar 和 GridView 控件实现，首先为 GridView 设置 BookListAdapter，在 getView 方法中为每个 Item 设置布局，使用 ImageView 和 TextView 分别显示背景图片及书名信息。在 OnCreatView 方法中调用 findAll 方法从本地数据库中获取书籍列表作为适配器的数据源。当要删除书籍时，用户长按某本书籍，触发监听器弹出 AlertDialog，点击确认删除该书籍，将调用 delete 方法删除数据库中书籍列表中对应的数据，同时回调 notifyDataSetChanged 刷新列表。

(2) 导入书籍功能

进入读书模块后，点击添加书籍后跳转至导入书籍模块，该模块实现效果如

图 4-4 所示。为了方便用户查找书籍，用户在浏览当前设备 SD 卡上的文件时，将只显示当前目录下的文件夹、TXT 格式的文件，非 TXT 文件不再显示。用户通过选择框选择文件后点击添加即可将文件加入书籍展示栏，FileAcitvity 是该模块的实现类。



图 4-4 书籍导入界面效果图

该模块通过 ListView 控件展示文件目录，需要创建适配器 FileAdapter。调用 Environment.getExternalStorageDirectory 方法获得 SD 卡的第一层的对象，通过 File 的 listFiles 方法获取该层目录信息 File[] 数组。接着，遍历该数组将文件夹以及 TXT 格式的文件分别存储在 ArrayList 对象中，并使用 ListView 将它们显示出来。每个 Item 由 ImageView、TextView、CheckBox 控件实现，若该 Item 显示文件夹，则设置 CheckBox 不可见，若为文件，则可见，用户可以点击 CheckBox 选中该文件。当点击添加书籍按钮时，开启新线程将用户选中的所有书籍名称及其存储路径添加到数据库中。当用户单击某 Item 时，将打开对应文件，判断该文件是否在 BookList 表中，若不存在则将该文件信息保存。

用户点击返回按钮可以返回上一级目录，在存储文件路径时定义了 Stack 类型

的变量 `pathStack`，当前目录的路径为 `pathStack` 栈的顶部路径。用户每次进入文件夹，就会调用 `push` 方法将上一次的路径存入 `pathStack` 中，当用户返回上一级时，`pathStack` 会 `pop` 方法移除堆栈最上层路径。

导入书籍模块目录展示逻辑流程图如图 4-5 所示。

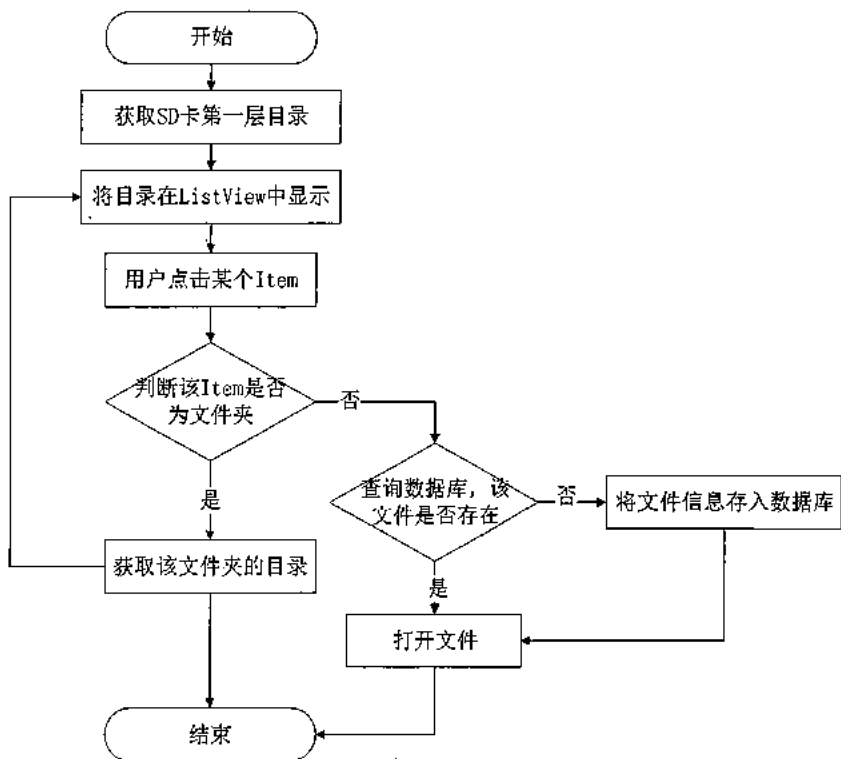


图 4-5 目录展示流程图

(3) 阅览书籍功能

该模块功能是读取书籍文件并在设备屏幕中显示书籍内容，可以通过滑动进行翻页，用户也可以添加、查看书签，支持有声朗读，并为用户提供一些个性化设置选择。

为了让用户可视范围最大化地用在书籍内容上，阅览书籍时的界面采用沉浸式模式进行设计，实现效果如图 4-6 所示。为了实现该设计，首先通过调用 `getWindow().getDecorView` 方法获取当前界面的 `DecorView`，接着通过 `setSystemUiVisibility` 方法可以设置系统 UI 元素的可见性。通过这些操作，就可以

使得界面默认情况下是全屏的，而且状态栏、导航栏等都处于隐藏，实现沉浸式模式的设计。

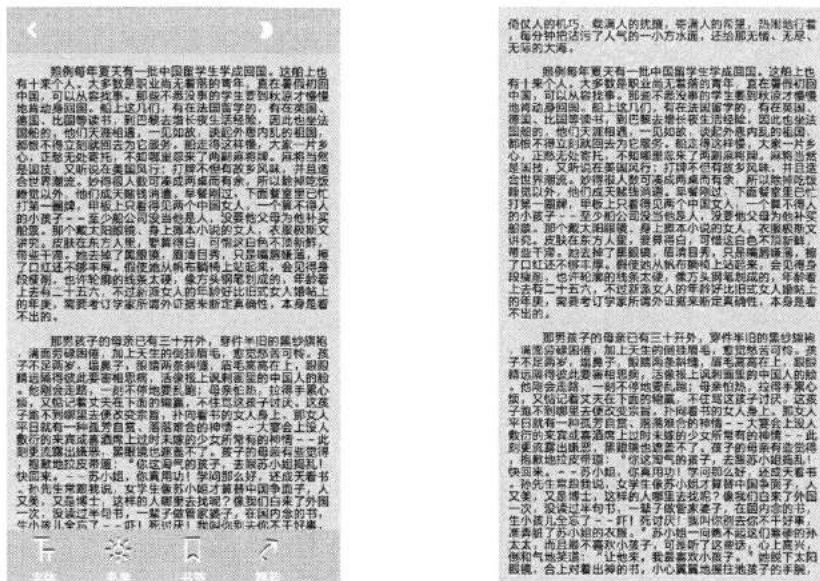


图 4-6 阅读界面效果图

使用 SharedPreference 存储当前阅读书籍的路径、阅读位置、文字大小、亮度值、夜间模式标识符、朗读记录等信息，在初始化界面时读取 SharedPreference 中存储数据进行界面的配置。

在阅读界面，将根据当前阅读进度显示书籍内容。受限于屏幕的大小，书籍不能在一页中全部展示，必须考虑对书籍分页。本模块的分页策略如下：获取文本长度及每页显示的字符数，计算出总页数。将文本数据分成数个 PageData，内部维护了一个书页的队列，该队列缓存了由三个 PageData 数据转化而来的书页列表。这样就能实现翻页的无缝切换，提供较好的整个阅读体验。读取书籍信息是通过 MappedByteBuffer 从指定位置开始读取文件，MappedByteBuffer 是 Java NIO 引入的文件内存映射方案，在处理大文件时读写性能极高。

检测到用户 ACTION_DOWN 的位置在屏幕中间时，弹出工具栏，界面顶部和底部都是工具栏菜单，实现效果如图 4-6 中顶部及底部所示，中间为空白的 TextView，当用户点击该 TextView 则隐藏工具栏。点击左右两侧或者左右滑动可

以翻页。

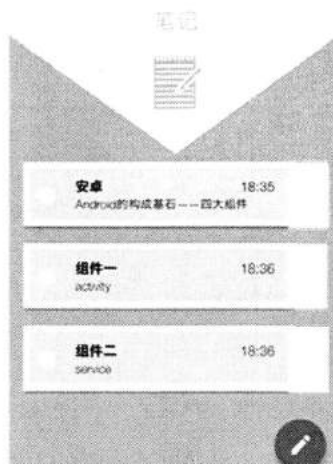
用户点击底部菜单栏的书签图标，可以添加书签或者查看我的书签。当用户添加书签时，将读取当前书籍 ID、当前页的第一个文字在整个文本的位置、当前页的前两行、时间、书籍路径并进行封装，首先判断该书签在数据库中是否存在，若不存在则调用 save 方法将书签记录存储到数据库中，否则添加失败。在我的书签列表界面，用户可以查看当前书籍的全部书签，点击某个书签可以跳转到书签所在页。

为用户提供的个性化设置由 TextView 控件实现，主要包括以下几个功能。跳转功能是根据用户设置的百分比计算出应显示页面的位置，然后在界面中加载显示完成跳转。该调节亮度功能，只针对本系统的亮度而不影响系统亮度，实现该功能需要 WindowManager 的帮助，获取当前 Window，为其添加将用户设置的亮度属性。当用户点击夜间模式按钮，将 Boolean 类型变量 isNight 值设为 true，将背景设为黑色，文字为白色颜色。当用户打开书籍时，从 SharedPreferences 读取夜间模式标识符 isNight，若为 true，则夜间模式，否则白天模式。

本模块实现了自动朗读功能。Android 自 1.6 版开始支持 TTS (TextToSpeech) 服务，即语音合成，但是该 TTS 不支持中文，所以本系统采用了科大讯飞的语音合成服务。在使用前需要在项目中集成语音服务 SDK，完成初始化并通过 SpeechSynthesizer 方法合成对象，当用户点击语音朗读按钮时，获取当前页的内容并传递给 startSpeaking 方法开始朗读，并在回调的监听方法中检测进度，本页完成后自动度下一页内容。在朗读时，点击界面弹出退出语音菜单，点击退出则调用 stopSpeaking 方法结束自动朗读。

4.2.2 读书笔记模块的实现

本模块将以列表的形式展示账户下所有的读书笔记，如果用户以游客模式登录，将无法查看任何笔记。用户点击悬浮按钮进入添加界面，编辑标题、内容后保存。对某一笔记，用户点击可以查看详细内容，滑动即可删除笔记。该模块实现效果分别如图 4-7(a)(b)所示。



(a) 笔记展示效果图



(b) 笔记查看效果图

图 4-7 读书笔记效果图

为了实现读书笔记功能，在服务器端设计了一张数据库表 NotesInfo（读书笔记表），其字段有 ObjectId（读书笔记 ID）、username（用户名）、noteTitle（笔记标题）、noteContent（笔记内容）、noteTime（创建时间）。

为了操作服务器端数据库，SDK 需要创建数据表对应的模型类，因此创建了 NotesInfo 类，在成员变量声明时成员变量的名称和数据表的字段名一致，并且为其分别实现 set 和 get 方法。

NotesFragment 类主要实现笔记的显示与删除功能。为了实现读书笔记以列表

的形式进行展示，采用了 RecyclerView 控件。RecyclerView 是一个强大的数据展示控件，它不仅通过桥接的方式将布局抽离增加了灵活性和易用性，而且其采用观察者模式设计对的适配器也增强了其可扩展性^[42]。首先，为 RecyclerView 创建适配器 NotesAdapter，然后在适配器中创建 ArrayList 类型的变量 mDataSet 存储笔记信息，在 onCreateViewHolder 中返回每个 Item 的 View，而 onBindViewHolder 方法则是渲染数据至 View，最后设置 OnItemClickListener 监听器以处理 Item 点击事件。

在 NotesFragment 中重写 onCreateView 方法，使用 LayoutInflater 加载布局文件生成 View 对象，然后向服务器端请求查询数据。若当前用户为游客模式，则 mDataSet 为空。若以账户登录，则调用 findObjects 方法查询服务器端的数据表，并将结果封装于 mDataSet，并传递给适配器，最终通过调用 notifyDataSetChanged 方法刷新数据。该模块显示流程如图 4-8 所示。在服务器端查询数据的主要代码如下所示：

```
Query.findObjects(new FindListener<NoteInfo>() {
    @Override
    public void done(List<NoteInfo> list, Exception e) {
        if(e==null)
        {
            for (NoteInfo notes : list)
            {
                notes.getNoteTitle();
                notes.getUserName();
                notes.getNoteContent ();
                notes.getNoteTime();
                mDataSet.add(notes);
            }
        }else{
            Toast.makeText(getApplicationContext(),
                "获取数据失败", Toast.LENGTH_SHORT).show();
            mDataSet = new ArrayList<>();
        }
    }
}
```

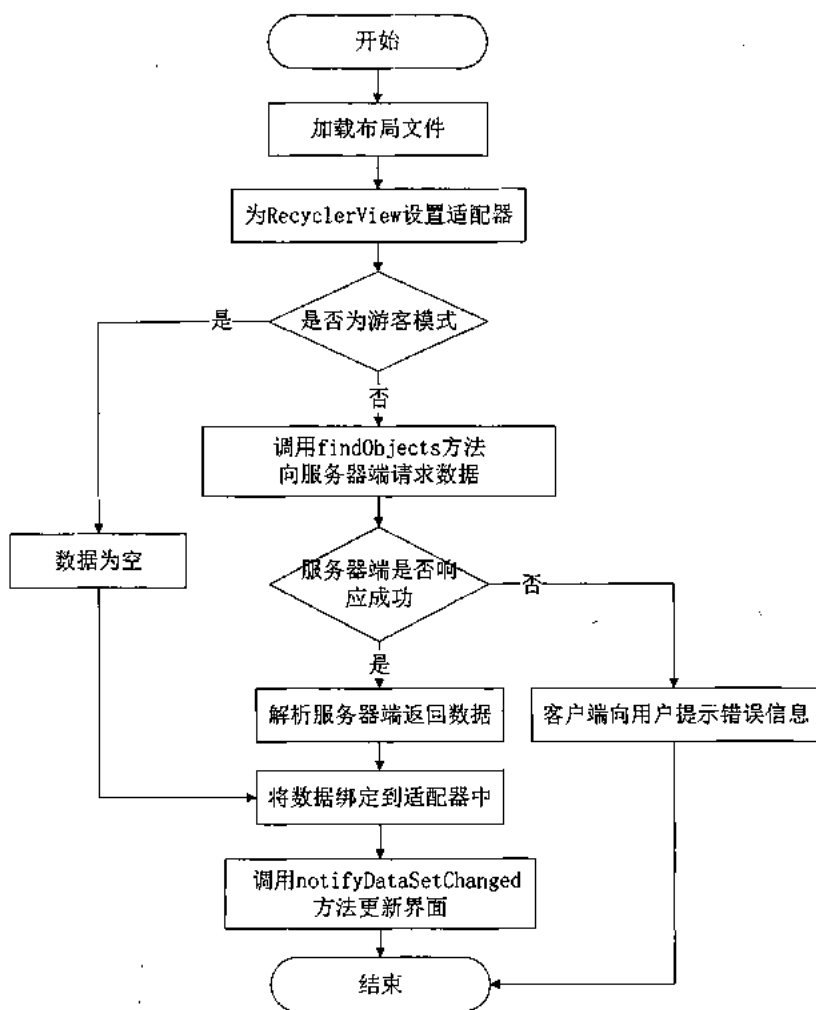


图 4-8 显示笔记模块流程图

当用户向左滑动某个 Item，出现删除图标，单击该图标完成删除操作。在 RecyclerView 的 Item 布局文件中，以 FrameLayout 为根布局，SwipeLayout 为子布局方式。在适配器 NotesAdapter 中，将滑动模式设置为 LayDown 模式，并且通过 addSwipeListener 方法为其绑定监听器。当用户点击删除图标时，执行 deleteItem 方法，向服务器端发送删除数据请求。具体做法是调用 getObjectld 方法获取被删除笔记的 ID，然后以此作为参数调用 delete 方法删除数据。同时，调用 remove 方法删除 mDataSet 中笔记信息，最后更新列表中的数据。

当用户单击某个 Item 时, 将触发监听器, 跳转至笔记详情界面。在 `onItemClick` 方法中, 创建 `Intent` 对象, 通过 `putExtra` 方法将笔记标题、内容、创建时间等设置为 `Intent` 的属性信息。调用 `startActivityForResult` 方法启动 `NoteDetailsActivity` 类。`startActivityForResult` 方法的参数分别是 `intent` 和 `requestCode`, 将 `requestCode` 设置为 `RENEW_NOTE`。使用 `FloatingActionButton` 控件作为添加按钮, 并为其设置监听事件, 点击即跳转至添加笔记界面。将 `startActivityForResult` 方法中的 `requestCode` 参数设置为 `NEW_NOTE`。

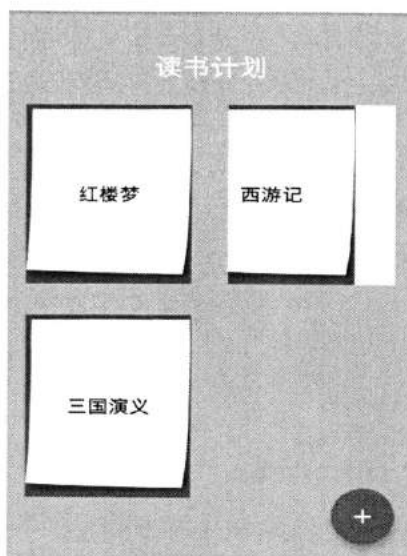
重写 `NotesFragment` 的 `onActivityResult` 方法, 用来接收传回来的结果, 根据 `requestCode` 类型以确定执行修改笔记或者添加笔记操作。若修改笔记, 则调用 `NoteInfo` 的 `update` 方法修改数据库, 若添加笔记, 则调用 `save` 方法在数据库中新增一条数据。

`NoteDetailsActivity` 是添加笔记和笔记详情的实现类。该界面包含笔记标题、内容以及保存按钮三部分。标题和内容通过 `EditText` 控件实现, 通过 `getExtra` 方法获取 `intent` 中的属性信息后绑定到 `EditText` 控件上, 因 `EditText` 特性, 所以在该界面用户也可以修改笔记。当用户点击保存按钮时, 将 `intent` 属性信息设置为笔记标题、内容、时间, 调用 `setResult` 方法, 该方法参数分别为 `requestCode` 和 `intent`, 并调用 `finish` 方法销毁 `Activity`。当 `NoteDetailsActivity` 被销毁时回调 `NotesFragment` 中的 `onActivityResult` 方法。

4.2.3 读书计划模块的实现

读书计划模块功能主要是添加、删除以及展示所有的读书计划, 在添加计划时设置计划完成时间, 当到达指定时间后会在系统通知栏提醒用户。该模块部分效果如图 4-9(a)(b)所示。

在 Java 代码中, 创建了 `BookPlanInfo` 类, 其将作为读书计划数据对象。相对应地, 在服务器中建一张数据库表 `BookPlanInfo` (读书计划表), 其字段有 `ObjectId` (读书计划 ID)、`username` (用户名)、`bookName` (书籍名称)、`alarmTime` (提醒时间)。



(a) 读书计划删除效果图



(b) 读书计划查看效果图

图 4-9 读书计划效果图

PlanFragment 是显示、添加、删除读书计划的逻辑实现类，该部分与读书笔记的展示与删除所采取的方案类似，采用 RecyclerView 与 SwipeLayout 结合实现列表展示计划且滑动删除功能，其中重要的部分就是为 RecyclerView 设置 Adapter。为 RecyclerView 创建适配器 PlanAdapter 时，在 onCreateViewHolder 方法中设置 XML 布局文件，为 Item 设置仿便签效果的布局。与服务器端的数据交互中，主要调用了 bookPlanInfo 的 findObjects、delete 等方法进行查询与删除。

当用户点击添加按钮进入读书计划添加界面，在 EditText 控件中可以输入书籍

名称。在制定读书计划时间时，用户点击显示时间的 `TextView`，就会弹出日期选择界面，该界面实现类是 `CalendarDialog`，重写该类的 `onCreateDialog` 方法，首先通过 `LayoutInflater` 生成 `View` 对象，实例化 `Date` 和 `MyCalendar` 对象，创建 `Dialog` 显示日历，用户选中某时间后点击 `Dialog` 的确认按钮时，回调刷新 `TextView` 中的时间。

当到达用户设置的书籍计划完成时间时将在系统通知栏进行提醒，为了实现该功能，主要使用了以下几个组件。`AlarmManager`，其本质是全局定时器，功能是在指定时间启动其他组件。在系统的通知栏中显示消息需要使用 `Notification`。`Service` 代表可执行程序，其在后台运行，没有用户界面。`BroadcastReceiver` 本质上是一个系统级的全局监听器，负责监听各个程序发出的广播^[43]。

在 `Android` 机制中，若 `BroadcastReceiver` 的 `onReceive` 方法不能在 10 秒内执行完成，则 `Android` 会认为该程序无响应，弹出 `ANR`，影响用户使用体验。为了避免这种情况，不能在 `onReceive` 方法中执行耗时操作。耗时操作可以通过使用新线程去完成，但是考虑到 `BroadcastReceiver` 生命周期较短，这里最终采取的方案是通过 `Intent` 启动 `Service` 解决该问题。`Service`、`BroadcastReceiver` 都属于 `Android` 的四大组件，在使用时必须要在 `AndroidManifest.xml` 中配置 `Service` 和 `Receiver`。

接下来将介绍提醒用户的具体实现过程，用户设置时间后调用 `sendBroadcast` 方法发送广播，启动指定的广播接收器 `BroadcastReceiver`。在 `BroadcastReceiver` 的 `onReceive` 方法中，通过调用 `startService` 方法启动指定服务时，将会回调 `AlarmService` 类（继承自 `Service`）的 `onStartCommand` 方法，在该方法中获取书籍名称以及提醒时间，并分别创建了 `Intent`、`PendingIntent` 对象，通过调用 `getSystemService(Context.ALARM_SERVICE)` 方法获得 `AlarmManager` 实例，通过 `set` 方法为 `AlarmManager` 设置参数，将 `PendingIntent` 对象绑定到 `AlarmManager`，这样可以到达指定时间时启动 `pendingIntent` 对象，调用 `pendingIntent` 对象绑定的 `Intent` 对象，从而启动指定的 `Service`，在该 `Service` 类中实现在通知栏提醒用户的功能。实现该功能首先需调用 `Context.getSystemService(NOTIFICATION_SERVICE)` 方法获得 `NotificationManager` 实例，然后实例化 `Notification` 对象并设置相关属性，

到达预设时间后调用 NotificationManager 的 notify 方法在通知栏显示提醒消息。

4.2.4 云备份模块的实现

本模块提出并实现了一种基于网络状况与预约模式的备份策略，提高系统性能，优化用户体验。为了解决用户设备存储空间不足及不同设备间数据迁移的问题，基于该策略实现了本系统的书籍云备份功能，使用本功能必须拥有账户。用户可以选择某本书籍并上传存储至服务器，也可以查看并下载本账户备份的所有书籍。实现的部分效果如图 4-10 所示。

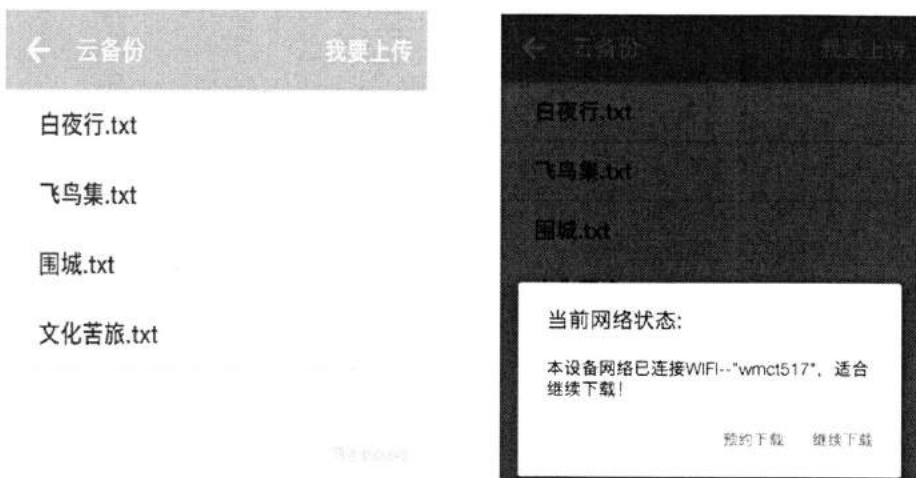


图 4-10 云备份模块效果图

1. 书籍备份策略的设计与实现

现在移动设备联网方式一般为 WIFI 和移动数据流量，使用 WIFI 联网通常是免费的，而用户使用移动数据流量进行文件上传与下载等操作会使用较多的数据流量。众所周知，移动数据流量需要收取费用，上传下载操作会给用户带来不必要的费用支出。如果用户因使用流量原因而取消上传下载，当用户再次希望上传或下载时需要重新操作，导致用户操作繁琐。

为了解决以上这些问题，本文提出并实现了一种基于网络状况与预约模式的备份策略，首先检测当前设备的联网类型及网络质量，提醒用户，并设计了预约备份模式。

在用户使用手机设备备份书籍时，根据用户当前联网状态，为用户提供多种

备份方式，给用户多样、合理、安全、便捷的选择，让用户更加智能方便的备份书籍，避免了用户在不知情的情况使用移动数据流量，避免不必要的费用花销。而且，在用户暂时不想继续备份时，引入了预约备份模式的概念，简化操作步骤，提高用户体验。该书籍备份策略如图 4-11 所示，接下来将介绍其实现过程。

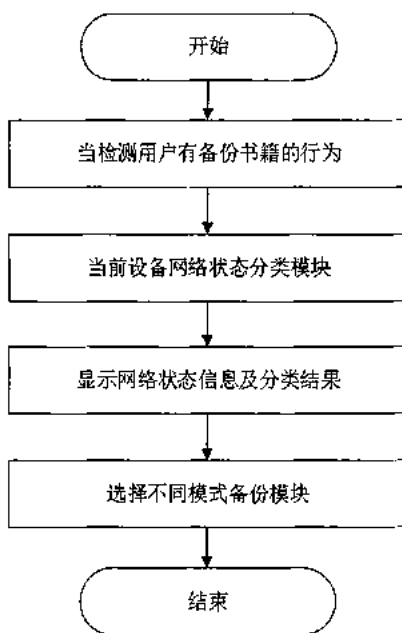


图 4-11 基于网络状况与预约模式的备份策略

(1) 检测设备当前网络联网类型

随着 Android 系统版本的不断更新，系统提供的检测网络状态的方法有所变化，为了提高本系统的兼容性，本文中的网络状态检测将根据系统不同版本提供相应的检测方案。首先检测当前设备的 Android 系统 API Level，如果是 API Level 23 之前的版本，则需要调用 `getSystemService(Context.CONNECTIVITY_SERVICE)` 方法得到 `ConnectivityManager` 对象 `connMgr`，通过 `connMgr` 就可以获取对应的 `NetworkInfo` 对象，该对象封装了网络连接的所有信息，根据参数 `TYPE_WIFI` 和 `TYPE_MOBILE`，判断当前设备所联接的网络是 WIFI 还是移动数据网络；如果是 API Level 23 及以后的版本，在获得 `ConnectivityManager` 对象 `connMgr` 后，将调

用 `getAllNetworks` 方法获取 `connMgr` 中当前设备网络联接的所有信息, `for` 循环遍历所有网络信息即可获得 `NetworkInfo` 对象, 通过该对象就可以判断当前设备联网类型。

(2) 检测设备当前网络质量

当用户进入云备份界面时开始计算当前网速, 并将网络速度及时反馈给用户。在计算网速时, 利用 `Timer` 类开启后台线程定时计算网速, 调用 `TrafficStats` 类中的 `getUidRxBytes` 方法获取当前流量值 `r1`, 间隔 `N` 秒后读取一次流量值 `rN`, 则根据流量差及时间计算当前网速。

在悬浮窗口中实时显示网速。自定义一个 `Service` 类, 用于实现创建悬浮窗口。要实现悬浮窗口效果, 获取窗口管理 `WindowManager`, 然后获取并设置窗口的 `Param`, 调用 `addView` 将其添加到 `windowManager`。需要注意的是必须设置悬浮窗口为不抢占焦点状态, 不影响界面其它操作, 否则会导致其他控件失去焦点。在计算网速后将其绑定到该悬浮窗口。

(3) 选择备份模式

用户根据反馈的当前设备网络状况选择不同的备份模式。当用户下载书籍时, 点击某书籍后弹出 `Dialog`, 若放弃下载, 则关闭 `Dialog`。若希望立即下载, 则点击继续下载。若希望预约下载, 则点击预约。

预约模式有两种方式, 一是当设备连接到 `WIFI` 时进行下载, 实现过程为首先自定义了一个广播接收器, 在该接收器中将检测当前网络类型, 若为 `WIFI`, 则启动自定义 `Service` 类执行下载任务。在注册广播时, 为 `IntentFilter` 添加属性值 `ConnectivityManager.CONNECTIVITY_ACTION`, 当用户设备网络状态变化后发送广播。广播接收器接收广播后启动服务执行下载。二是当设备连接到指定名称的 `WIFI` 时下载, 实现与模式一的类似, 在该接收器中检测当前网络类型时, 还需要获取当前设备联接的 `WiFi` 名称。如果当前设备联接网络类型为 `WIFI`, 则创建 `WifiManager` 对象, 并通过 `WifiManager.getConnectionInfo` 方法获得 `WifiInfo` 对象, 最后调用 `getSSID` 方法获取当前连接的 `WIFI` 名称。对获取的 `WIFI` 名称与用户指定名称进行对比, 若名称一致则开启自定义 `Service` 类下载。上传与下载类似, 不

再赘述。

2. 云备份功能的实现

在数据库中创建了表 FileInfo（备份书籍表），其主要字段有 ObjectId（备份书籍 ID）、username（用户名）、fileName（书籍名称）、bookFilePath（文件存储 url）。对应地，在 Java 中创建了模型类 FileInfo。

进入该模块，首先用户可以查看服务器中当前已备份的书籍，书籍通过 RecyclerView 控件以列表的形式进行展示，调用 findObjects 方法向服务器端发送请求获取书籍名称作为 RecyclerView 的数据源。为每个 Item 绑定监听器，当用户点击某本书籍时开始执行下载模块。点击 Toolbar 的菜单选项上传，可以浏览 SD 卡上的文件，选中文件后可进入上传模块。

在上传书籍时，将书籍在本地存储路径传递给 uploadblock 方法，采用分片上传模式，避免单个 HTTP 数据量太大造成的连接超时问题，特别是网络状况较差时，分片可以在一定程度提高上传成功率，避免无休止的失败重试。当文件上传后，在 done 方法中，若成功，则调用 getFilename 方法和 getFileUrl 方法分别获取文件名称、文件存储位置 url 并将其保存在数据库中。

在下载书籍时，当用户点击某一个 Item，获取该 Item 中的文件名称，先调用 findObjects 方法查询获取该文件对象，从而调用 getBookFilePath 方法得到文件 url，接着调用 download 方法将书籍下载到默认路径。

4.2.5 好书推荐模块的实现

本模块将为用户展示评分较高的优秀书籍，让用户开拓视野，发现好书。本模块包括好书浏览、搜索书籍及查看详情功能。书籍以列表的形式供用户浏览，用户可以浏览书名、书籍封面、作者、评分、评论数等信息，如果对某本书籍感兴趣，点击书籍就可以进入书籍详情界面，在该界面中可以查看本书主要内容及部分书评。在标题栏提供了搜索功能，用户可以随时搜索书籍查看信息。下面介绍各功能具体的实现。

在实现本模块推荐书籍列表、书籍详情时，本论文面临缺乏数据来源的问题。如果创建数据库资源提供该服务，则存在数据量过大，后期的维护和更新过于繁

琐，而且无法保证数据的实时性等问题。为解决该问题，在开发中通常有两种解决方案，一是利用第三方接口提供的数据，二是通过网络爬虫工具获取数据。在实现搜索、查看书籍详情时采用了方案一，在获取推荐给用户书籍列表时，为了加强推荐书籍的时效性，采用了方案二。

1. 通过网络爬虫获取推荐书籍数据

在调研 Android 中网络爬虫现状后，选择了 Jsoup 工具爬取数据。Jsoup 是一款 Java 的 HTML 解析器，可直接解析某个 URL 地址、HTML 文本内容。使用 Jsoup 从网页中获取数据，不需要数据库的支持，而且 Jsoup 强大的选择器功能支持开发者方便地解析 Html 元素^{[44][45]}。

为了使用 Jsoup，需要在 build.gradle 文件中添加以下依赖：

```
compile 'org.jsoup:jsoup:1.11.2'
```

在通过 Jsoup 爬取数据时，首先以 URL 为参数，发送 GET 请求获取 Document 对象，然后通过调用 select(String cssQuery)方法获取符合属性选择器要求的标签集合 Elements，通过 for 循环遍历解析 HTML 元素以获取所需数据^[46]。在获取数据时，可以将数据封装成对象，并调用 add 方法保存在 ArrayList 对象中。Jsoup 爬取数据的主要流程如图 4-12 所示。其中，通过 Jsoup 获取推荐书籍数据的部分代码如下：

```
Document document = Jsoup.parse(response);
Elements elements = document.select("#content");
Elements books = elements.select("li.media");
for (Element book : books)
{
    RecommendedBookInfo book = new RecommendedBookInfo();
    String id = bookElement.select("div.media__img a").attr("href");
    id = id.substring(id.indexOf("subject") + 8, id.length() - 1);
    book.setId(id);
    ... //省略部分解析代码
    mDataSet.add(book);
}
```

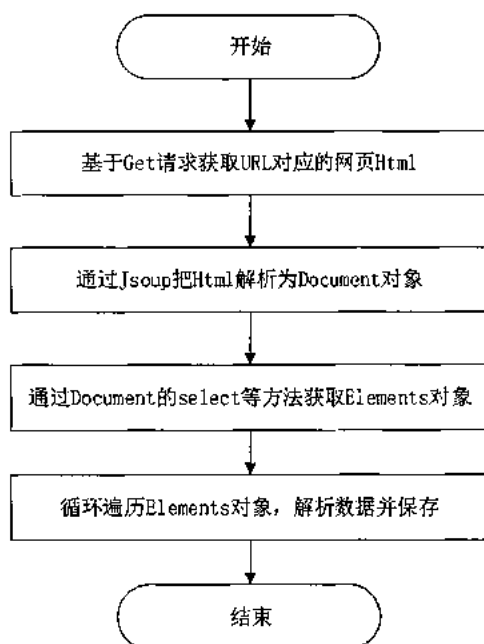


图 4-12 Jsoup 获取数据流程图

2. 好书推荐功能的实现

为了更好地处理书籍数据，在 Java 代码中，创建 RecommendedBookInfo 实体类，并实现 Parcelable 接口将数据序列化^[47]，其将作为推荐书籍数据对象。该类成员变量有 id（推荐书籍 ID）、title（书籍名称）、author（作者）、publisher（出版社）、publishDate（出版日期）、rate（评分）、bitmap（封面）、reviewCount（评论数量）、content（内容）、summary（评论）等，并实现分别为其实现 set、get 等方法。

为了防止线程阻塞、提高系统网络性能，创建了 MyAsyncHttp 类用于进行网络请求，MyAsyncHttp 继承自 AsyncHttpClient，它是基于 Apache HttpClient，将所有的网络请求独立于 UI 主线程之外，并且通过回调方法处理网络请求的结果，实现了异步请求。在该类中，getAsyncHttpResponseData 方法对 GET 请求进行了封装以提高代码复用性。

好书浏览、搜索书籍业务逻辑主要在 RecommendedBookActivity 类中实现。在用户进入好书浏览界面时，首先调用 inflate 方法加载布局文件，然后为书籍展

示控件 RecyclerView 绑定适配器 RecommendedBookAdapter。在 onCreate 方法中开启新线程调用 getAsyncHttpResponseBody 方法发送请求从豆瓣排行榜网页中爬取推荐书籍的数据, 在 onSuccess 方法中, 解析 response, 获取 Document 对象后再获取标签集合对象, 并通过一个 for 循环将书籍 ID、书名、作者等信息 set 到 RecommendedBookInfo 实例中, 调用 add 方法添加到 ArrayList 类型的 mDataSet 中, 最后通过 notifyDataSetChanged 方法, 更新列表, 最终好书列表界面效果如图 4-13 所示。



图 4-13 好书列表界面效果图

通过 onCreateOptionsMenu 方法添加搜索菜单项, 然后添加监听器, 当用户在搜索栏中输入书名点击搜索后, 通过豆瓣公开的搜索图书 API, 将用户输入的书名作为关键字, 发送 GET 请求, 获取查询结果 response, 其数据类型为 JSONObject, 调用 optJSONArray 方法转换为 JSONArray, 通过 for 循环遍历 jsonArray 将书籍 ID、书名、作者、评分、评论人数封装至 RecommendedBookInfo 对象, 调用 add 方法保存在 mDataSet 中, mDataSet 为 ArrayList<RecommendedBookInfo> 类型。最后将 mDataSet 作为 RecyclerView 的数据源, 调用 notifyDataSetChanged 方法刷新显示界面。

在搜索结果界面, 用户点击某书籍将进入书籍详情界面。搜索界面最终效果

如图 4-14 所示。



图 4-14 书籍搜索界面效果图

对于 RecyclerView 的每个 Item，通过 `setOnClickListener` 设置监听器，当被点击时，将获取当前书籍的 ID，通过豆瓣公开的获取图书信息 API，并以书籍 ID 为参数发送 GET 请求获取该书籍的详细信息，解析结果信息调用 `set` 方法保存在 `RecommendedBookInfo` 对象中。调用 `startActivity` 跳转到查看详情界面。通过 `Bundle` 将获取的书籍信息传递到详情界面，取出数据后将数据渲染至各控件。由于书籍详情信息太多而不能在一屏中全部显示，所以在实现该界面时，采用在 `ScrollView` 内部设置 `LinearLayout` 布局，在 `LinearLayout` 中添加 `TextView` 控件实现本界面布局，最终实现竖直滚动查看书籍详情的效果，书籍详情界面如图 4-15 所示。

其中，在详情界面可以通过以下代码获取数据：

```
Intent intent = new Intent(context, BookDetailsActivity.class);
Bundle bundle = new Bundle();
bundle.putParcelable(RecommendedBookInfo.TAG, RecommendedBookInfo);
intent.putExtras(bundle);
```



图 4-15 书籍详情界面效果图

4.2.6 登录模块的实现

本系统登录模块主要包括三个界面，分别是登录界面、注册界面、找回密码界面。当用户点击系统图标后，如果用户之前已登录且未注销账户，则直接进入系统主界面。如果用户首次登录或者之前已注销账户，则进入登录界面，在该界面用户可以选择游客模式或者使用账户登录系统，点击游客模式则进入主界面，而输入用户名、密码、验证码后则以账户模式进入主界面。在注册界面用户可以注册账户，注册时所用的邮箱可以作为找回密码的媒介。该登录模块主要流程如图 4-16 所示。

本模块中界面均通过 XML 布局文件进行定义，在 `onCreate` 方法中通过 `setContentView` 调用布局文件，这样可以将视图控制逻辑与 Java 代码剥离，符合 MVC 原则。而登录、注册、找回密码三个功能均通过 `Activity` 类实现，并且三者通过 `Intent` 机制进行跳转，比如用户在登录界面，点击找回密码后进入找回密码界面的关键代码如下：

```
Intent intent = new Intent(LoginActivity.this, ForgetPasswordActivity.class);
startActivity(intent);
```

这种设计可以有效降低系统耦合性，对系统的可扩展性和可维护性的提高也有所帮助。

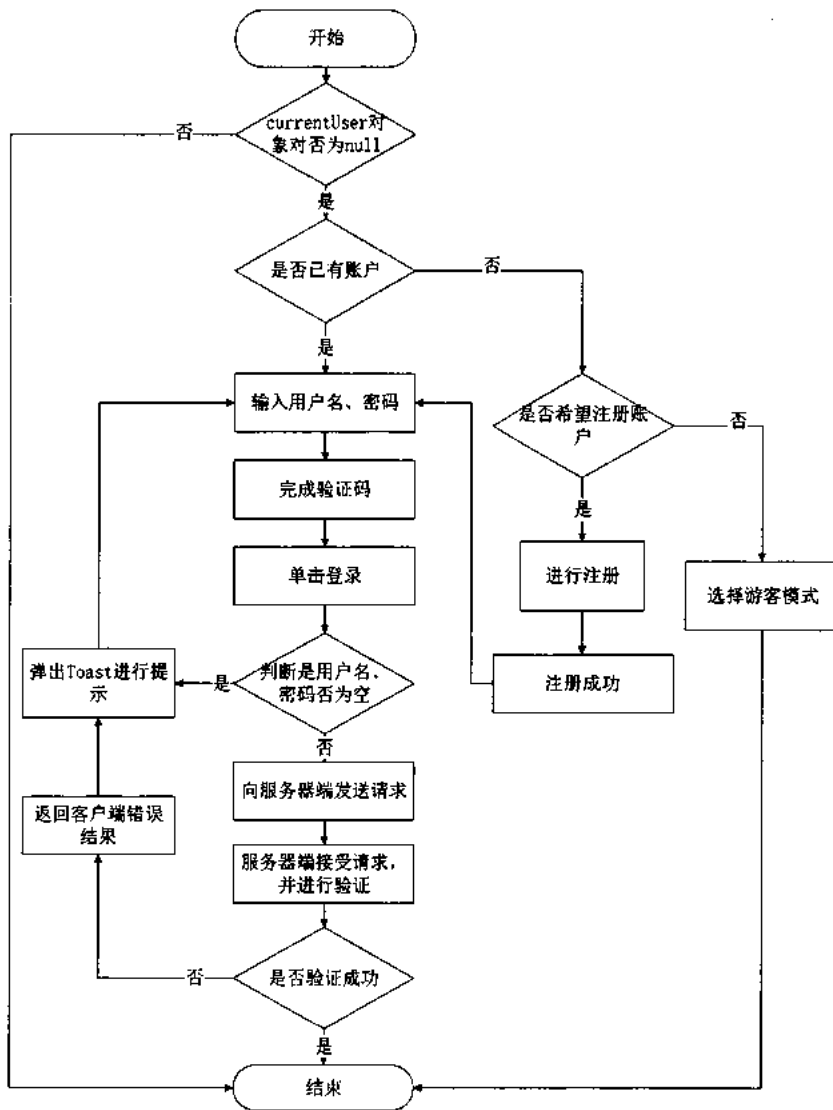


图 4-16 登录模块流程图

本模块涉及到的类主要有以下三个。

1. LoginActivity 类

该类是登录模块的主要逻辑实现类。在 XML 布局文件中一级布局管理器采用了 RelativeLayout，通过 EditText、TextView、ImageView 控件实现布局，并且为各

个控件创建监听器事件。最终实现登录界面的效果如图 4-17 所示。

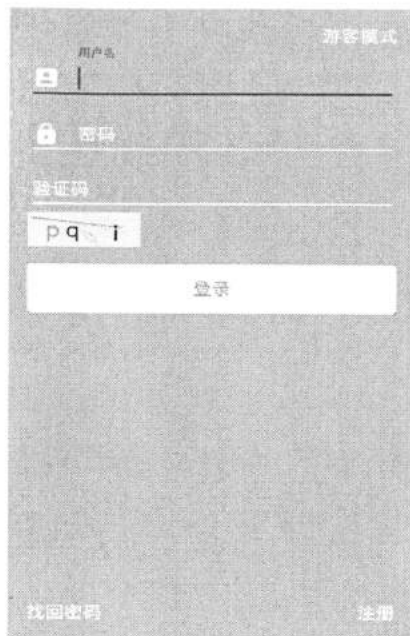


图 4-17 登录界面效果图

在数据库中创建了表 User（用户信息表），其主要字段有用户 ID(objectId)、用户名(username)、密码（password）、头像 ID（photoId）、注册时所用邮箱(email)。对应地，在 Java 中创建了模型类 UserData。

用户在登录时，输入用户名、密码，并完成图形验证码的验证操作。输入完毕后点击登录按钮，登录按钮的监听器中将调用 invalidate 方法实现校验功能。将校验信息封装至 invalidate 方法中，主要通过 isEmpty 方法检测用户所输入的信息是否为空，通过获取 EditText 的字符串后判断是否与 getCode 方法的值相等来校验验证码。若通过校验，则调用 login 方法向服务器端发送用户信息数据，根据服务器端返回结果进行处理。返回结果正确则直接进入系统主界面，否则通过 Toast 通知用户错误原因。

其中，为了实现图形验证码功能，创建 CheckCodeUtil 类。在设计该类时，采用单例设计模式，避免消耗过多资源。首先在该类中首先创建 Bitmap 对象并通过 Canvas 获得画布，绘制验证码字符串，该字符串颜色、内容均随机产生，最后再

使用 Path 绘制干扰线。另外，还需要暴露一个公有静态方法 `getInstance`，在第一次调用时完成初始化。当用户无法辨别验证码时，可以点击图片更换验证码。

为了避免用户每次启动系统都要进行登录操作，在用户注册成功或是第一次登录成功时，在本地磁盘缓存一个 `CurrentUser` 对象。如果用户进行了注销，会清除本地缓存的 `CurrentUser` 对象。当启动系统时，先获取这个缓存的用户对象，若非空，则进入系统主界面。主要代码如下：

```
userData = UserData.getCurrentUser(UserData.class);

    if (userData != null) {

        Intent intent = new Intent(this, MainActivity.class);
        startActivity(intent);
        finish();
    } else {

        userData = new UserData();
        setIsTrial(true);
    }
}
```

用户点击游客模式，可以直接进入主界面，将 `setIsTrial` 参数设为 `true`，以便为后续功能提供标识，以此区分账户用户与非账户用户权限。

2. ForgetPasswordActivity 类

该类是本模块的找回密码的实现类，该界面效果如图 4-18 所示。当用户忘记密码时，可以通过向注册时所使用的邮箱发送请求重置密码。该界面顶部由 `Toolbar` 实现，通过 `setNavigationOnClickListener` 添加监听器完成返回上一界面。用户在编辑框内输入邮箱后单击发送按钮，调用 `resetPasswordByEmail` 方法查询服务器端数据库中的用户表，若该邮箱地址存在，则向该邮箱发送一封可以进行密码重置的邮件，用户根据向导点击重置连接，打开该链接网页，根据提示他们可以输入一个新的密码。若该邮箱地址不存在，则 `Toast` 反馈用户邮箱无效。



图 4-18 找回密码界面效果图

3. SignUpGridActivity 类

该类是本模块的注册账号的实现类，实现效果为图 4-19 所示。该模块 XML 布局文件采用 CoordinatorLayout 作为布局的根节点，由 Toolbar 和数个 TextView 控件组成。Toolbar 实现导航展示功能，用户依次填写用户名、邮箱、密码、确认密码并选择头像，最后单击申请，单击行为将触发监听器进行校验两次密码是否一致、其他项是否为空。通过校验则调用 signUp 方法向服务器端发送请求，将用户名、密码、邮箱、头像 ID 都上传至服务器，查询数据库用户表中用户名和邮箱是否唯一。若是唯一的则注册成功，否则调用 Toast 显示错误对象。

用户头像部分被设计为按行、列分布的方式显示，因此采用网格视图 GridView 实现。创建了 MyPhotosAdapter，其是 BaseAdapter 的子类，为 GridView 提供显示的图片。



图 4-19 注册账号界面效果图

4.3 本章小结

本章主要阐述了书籍阅览与管理系统的实现过程。首先介绍了客户端应用界面框架的实现，主要包括系统主界面及侧滑菜单栏。然后介绍了主要功能模块的实现，主要包括读书模块、读书笔记模块、读书计划模块、云备份模块、好书推荐模块、登录模块。在实现书籍备份功能时，提出并实现了一种基于网络状况与预约模式的备份策略，避免用户在不知情的情况下使用流量，并简化备份操作流程，提高了用户体验。

第5章 书籍阅览与管理系统的测试

随着用户对软件质量越来越重视，软件测试作为保证软件质量的重要手段，其重要性不言而喻。通过有效的测试工作，可以提高系统的应用性，保证系统稳定地运行^{[48][49]}。本章中通过多种测试技术对本书籍阅览与管理系统进行了测试。

5.1 测试环境

书籍阅览与管理系统的测试环境包括一台高性能电脑和多部 Android 系统设备。高性能电脑已配置 Android 应用开发环境，将作为本书籍阅览与管理系统的开发平台和 Android 模拟器的运行平台。选取三部 Android 设备安装本文开发的系统进行测试，每个设备的硬件配置如表 5-1 所示。

表 5-1 测试环境配置表

设 备	角色	硬件配置
高性能电脑	开发平台	内存：8GB 操作系统：Windows 7
中国虹计算机（第四代）	Android 客户端	运行内存：1GB 操作系统：Android 4.4
华为平板	Android 客户端	运行内存：4GB 操作系统：Android 6.0
坚果 Pro	Android 客户端	运行内存：4GB 操作系统：Android 7.1.1

5.2 测试项目及结果

5.2.1 功能测试

功能测试是应用测试过程中最基本的测试环节。在 Android 设备上安装、卸载

本系统，多次进行开启、关闭操作以检验其开启、关闭速度，在不同设备上使用本系统检测系统的可见外观设计是否合理、易用以及兼容性^[50]。为了测试本系统功能是否全部实现、输入输出能否正确，本文还设计了每个功能模块的测试用例^{[51][52]}，例如，表 5-2 是读书笔记功能的测试用例，其余模块不再赘述。根据测试用例依次对各模块测试，结果表明本系统通过了功能测试。

表 5-2 读书笔记功能测试用例表

测试名称	读书笔记
用例描述	在联网正常或失败的情况下，以不同模式登录，分别测试读书笔记的增、删、改、查功能。
预置条件	以不同模式成功进入系统主界面；
预期结果	在游客模式下，有网、无网状态均无法进行笔记的增、删、改、查； 在个人账户模式下，联网正常时可以进行笔记的增、删、改、查。若联网失败，界面不显示笔记，并提醒用户无法获取数据；
测试步骤	以游客模式登录： 1.在联网正常、失败状态下分别执行下述操作； 2.在系统主界面点击“读书笔记”，进入笔记界面； 3.在笔记界面不显示笔记记录； 4. 在笔记界面点击右下角的添加按钮，弹出提示框提醒用户无权添加笔记； 以个人账户登录： 1.在联网正常、失败状态下分别执行下述操作； 2.在系统主界面点击“读书笔记”，进入笔记界面； 3.在笔记界面查看是否正确显示该账户所有笔记记录； 4.在笔记界面点击某条笔记，进入笔记详情界面，查看笔记具体内容，并且分别修改笔记内容、标题，点击保存按钮，返回笔记界面查看是否修改成功； 5.在笔记界面点击右下角的添加按钮，只输入标题或不进行任何输入，点击保存按钮时，提示内容无效；只输入内容或者正确输入标题、内容，点击保存按钮，添加成功； 6.在笔记界面，向左滑动某条笔记，点击删除；
测试结果	同预期，测试通过

5.2.2 流畅度测试

为了保证用户良好的使用体验，确保用户与应用之间的交互平滑流畅，需要进行系统流畅度测试。在 Android 中，流畅度是指响应和渲染应用 GUI 的性能，当前衡量一个系统是否流畅的主要指标是 FPS（每秒帧刷新频率）^[53]，本论文测试书籍阅览与管理系统的 FPS，作为系统流畅度的衡量指标。

Android 系统的用户界面显示工作是由 CPU 和 GPU 配合完成的。为了给予用户更好的流畅度体验，Google 在 Jelly Bean(Android 4.1 版本)引入了新特性 Project Butter（黄油计划），在系统的渲染和动画方面都将采用垂直同步技术（V-Sync）机制，旨在限制帧速率低于刷新速率^[54]。Android 将支持帧率为 60fps 的动画效果，即绘制一帧的时间仅为 16.6ms，系统能够以每秒 60 帧的帧速运行，而不会出现帧丢失或帧延迟等卡顿现象。如果帧率小于 60fps，则用户就可能会有应用不流畅的感觉。

本文通过 Android 系统内置的 dumpsys 检测应用动画渲染耗时情况，帮助定位性能不足，评价系统流畅度。dumpsys 是一个 Android 设备里的可执行文件，用于提供系统服务状态信息。对 dumpsys 传递 gfxinfo 参数可以输出当前设备的指定应用的动画帧的状态^[55]。下面介绍测试步骤：

1. 将运行本系统的设备设置中的“开发者选项”选项下的 GPU 呈现模式分析设为“在 adb shell dumpsys gfxinfo 中”；
2. 在命令窗口中，进入 Android SDK 的 platform-tools 所在路径，然后输入命令启动 dumpsys，将所测结果信息保存到本地文件中，测试代码如下：

```
adb shell dumpsys gfxinfo 应用包名 > 信息存储路径
```

其中，本系统客户端包名是 com.wmct.readingbook，测试结果信息将存储在电脑本地文件中，路径为 F:\mylogs.txt。

3. 打开 mylogs.txt 文件，对测试结果数据进行处理并绘制本系统运行时每帧渲染耗时折线图，如图 5-1 所示，该图表示了用户运行该系统好书推荐等模块时的帧渲染耗时情况。

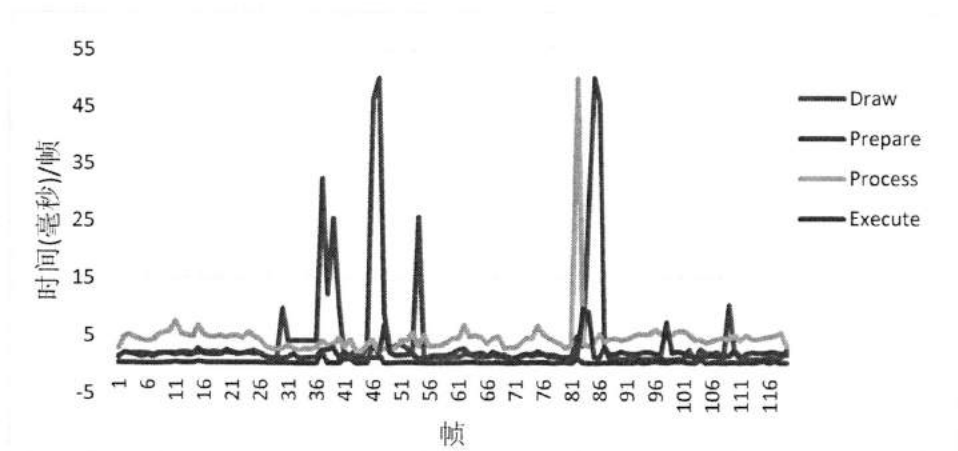


图 5-1 本系统运行某模块时每帧渲染耗时

其中，横坐标代表帧数，纵坐标代表每一帧的渲染耗时，单位为毫秒，各图例含义如下：**Draw** 代表创建和更新视图显示列表的时间，若该部分时间很高，可能是需要绘制较多自定义视图或 **onDraw** 执行任务较多；**Process** 表示图像引擎渲染视图显示列表的耗时，UI 界面中 **View** 数量会对该时间产生影响；**Execute** 表示 CPU 等待 GPU 处理的耗时；**Prepare** 代表准备时间^{[56][57]}。

四个参数之和即为一帧的渲染耗时，如果持续较长时间超过 16.6ms，用户会感觉到卡顿。但是由于可能存在其他进程的影响，并不能完全体现流畅度，不过一定程度上可以体现系统流畅度，帮助分析系统性能。由图中可以看出，在运行好书推荐等功能时整体流畅度良好，但是在某些情况下显示书籍列表时会比较缓慢。

本论文对书籍阅览与管理系统全部功能进行了多次流畅度测试，结果显示系统整体流畅度良好。

5.2.3 稳定性测试

一款应用在设计系统功能性需求的基础上，还需要维持长期稳定的运行，所以必须保证系统具有良好的稳定性以及健壮性。本文主要利用 **Monkey** 工具进行压力测试以验证系统的稳定性。

Monkey 是一个命令行工具，在模拟器或者实体设备中均可使用。当 **Monkey**

运行时，它将生成并发送伪随机的用户事件流到系统，根据输入的事件执行次数对待测应用进行压力测试^{[58][59]}。通过 Monkey 测试可以发现系统崩溃（Crash）、系统无响应（ANR）等严重影响系统质量的问题。Crash 会导致系统无法正常使用，而 ANR 现象会给用户带来糟糕的使用体验。当应用程序不响应用户输入时，在特定时间内（Activity 中 5 秒或者在 BroadcastReceiver 中为 10 秒），Android 将提供给用户 ANR 对话框^[60]。在命令窗中，输入命令启动 Monkey，具体测试代码如下：

```
adb shell
cd data/data
monkey -p 应用包名 -v 执行次数
```

其中，参数-p 是用来指定一个或者多个包，指定包名后，Monkey 工具将只启动指定包名对应的应用，若不指定，则启动设备中所有应用。参数-v 代表日志信息详细级别。本系统客户端包名为 com.wmct.readingbook，设置执行次数依次设为 5000、10000、20000。

对本系统进行多次 Monkey 测试，部分结果如下图 5-2、5-3、5-4 所示。

```
Events injected: 5000
:Sending rotation degree=0, persist=false
:Dropped: keys=3 pointers=0 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=20811ms (0ms nobile, 0ms wifi, 20811ms not connected)
// Monkey finished
```

图 5-2 Monkey 测试结果图

```
Events injected: 10000
:Sending rotation degree=0, persist=false
:Dropped: keys=5 pointers=1 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=47526ms (0ms nobile, 0ms wifi, 47526ms not connected)
// Monkey finished
```

图 5-3 Monkey 测试结果图

```
Events injected: 20000
:Sending rotation degree=0, persist=false
:Dropped: keys=17 pointers=2 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=41940ms (0ms nobile, 0ms wifi, 41940ms not connected)
// Monkey finished
```

图 5-4 Monkey 测试结果图

本论文对书籍阅览与管理系统进行了多次 Monkey 测试，据测试结果数据显示，运行期间本系统未发生 Crash 或 ANR 现象，系统具有良好的稳定性。

5.3 本章小结

本章主要是对本文设计并实现的书籍阅览与管理系统进行了测试，首先对测试环境进行介绍，选取了多款 Android 设备进行测试工作，然后从系统功能、系统流畅度、系统稳定性三个方面依次进行测试。在功能测试中针对该系统各个模块设计了测试用例并实施，结果显示功能已满足本文设计要求；通过 dumpsys 工具测试了流畅度，系统整体流畅度表现良好；通过 Monkey 工具进行压力测试，系统显示出较强的稳定性。

第 6 章 总结与展望

6.1 总结

随着移动互联网的飞速发展，移动智能终端技术日新月异，Android 终端凭借其丰富的功能、便捷的使用对人们的生活、工作以及学习都产生了深远影响。阅读，作为生活中不可或缺的一部分，也深受其影响。人们的阅读模式正在悄然改变，不仅阅览纸质书籍，电子书也成为人们阅读的重要资源。但是人们在管理大量电子书籍时会比较麻烦，而且在阅读时缺乏计划性，难以进行高效地读书、管理笔记等。大量电子书将占用较多存储空间，这对设备有限的存储空间也是一个问题。因此，本论文设计并实现了一款书籍阅览与管理系统。

本论文主要围绕以下几个方面展开研究和开发：

1. 对 Android 应用市场中现有书籍阅读类应用进行分析与对比，总结其功能及优缺点。在调研的基础上，结合用户阅读行为的实际需求，设计了本书籍阅览与管理系统具有的功能。

2. 为了更好地完成开发工作，本文深入研究了一些 Android 应用开发关键技术，包括自定义 View 技术、目前 Android 主流图片加载框架及 Android 系统动画技术。在深入研究 View 的层次结构、View 的工作原理和 Android 事件分发机制的基础上实现自定义 View，一定程度上减轻系统界面的同化度。对图片加载框架进行了对比，重点研究了 Glide 并将其集成到本系统，提高系统图片加载性能。深入研究 Glide 源码，学习其设计模式及编码风格等，有助于对本系统的设计与实现。最后通过对 Android 系统动画技术的研究，在本文设计的书籍阅览与管理系统中实现界面切换动画效果等，这丰富了系统 UI 展示形式，为用户提供绚丽的视觉效果。

3. 本文提出并实现了一种基于网络状况与预约模式的备份策略。检测当前设备的联网类型及网络质量并提醒用户，设计预约备份模式。该策略避免了用户在不知情的情况使用移动数据流量，并且预约模式的设计可以简化用户操作，提高用户体验。

4. 为了获取好书推荐功能中的数据源, 研究了 Android 网络爬虫技术现状, 并通过 Jsoup 进行数据爬取工作。

5. 基于系统需求分析, 设计了书籍阅览与管理系统的架构、框架模式及系统功能模块。利用 Android Studio 开发平台, 完成了书籍阅览与管理系统的开发工作。本系统为用户提供游客与账户两种登录模式, 提供书籍的阅读与管理、读书笔记、读书计划、书籍云备份、好书推荐功能。在满足用户读书的基本需求上, 制定读书计划督促用户读书, 支持读书笔记, 而且提供书籍云备份功能解决用户存储空间不足的问题, 还可以通过好书推荐查看评分较高优秀书籍, 通过账户可以实现数据快速迁移, 为用户打造了一个多功能的读书平台。

6. 主要从系统功能、系统流畅度、系统稳定性三个方面对书籍阅览与管理系统进行测试工作。设计并实施了功能测试用例, 利用 dumpsys、Monkey 等测试工具发现问题并解决, 最终系统各项测试表现良好。

6.2 展望

本论文设计并实现的基于 Android 系统的书籍阅览与管理系统, 虽然已基本实现了预期功能, 满足用户需求, 但由于研究时间较短及个人能力有限等因素, 本系统还有需要改进和完善的方面:

1. 目前本系统中书籍格式只支持 TXT 格式, 未来可以研究 Android 如何解析 EPUB 等格式的书籍, 进一步拓展系统支持的书籍类型。

2. 本系统中好书推荐功能有待完善。目前推荐的书籍数据是从网络中爬取的, 依靠豆瓣数据, 未来可以根据用户在服务器端的读书记录、笔记内容等数据, 进行大数据分析, 针对用户阅读行为提供个性化推荐。

参考文献

- [1] 梁辰. 移动互联网迎来重大转折, 手机上网用户数首超 PC[J]. 通信世界, 2012(28):20-20.
- [2] 2017 年 11 月份通信业经济运行情况[OL].
<http://www.miit.gov.cn/n1146312/n1146904/n1648372/c5967653/content.html>.
- [3] 张萌.“互联网+”时代移动智能终端发展建议[J]. 通信企业管理, 2016(3):59-61.
- [4] 王跃.我国移动智能终端操作系统平台发展研究[J].信息通信技术, 2012, 6(4):30-34.
- [5] 落红卫. 移动操作系统现状分析与发展建议[J]. 保密科学技术, 2015(7):25-29.
- [6] Statcounter 实时数据[OL].
<http://gs.statcounter.com/os-market-share#monthly-201702-201802>.
- [7] 唐博海. 移动互联网现状与发展趋势探究[J]. 通讯世界, 2017(22):34-35.
- [8] 谢强, 刘术华. 我国数字阅读最新发展及对图书馆电子书服务的启示[J]. 图书馆, 2017(1):24-27.
- [9] 苏秦, 谢金海. 纸质书与电子书比较分析[J]. 唯实(现代管理), 2015(2):42-43.
- [10] 高凯. 基于中国虹计算机的文件管理系统设计与研究[D]. 山东大学, 2015.
- [11] 张浪. 基于中国虹计算机的城市服务系统的设计与实现[D]. 山东大学, 2017.
- [12] 韩静华, 武丽莎. 以用户体验为中心的阅读类 APP 设计研究[J]. 包装工程, 2017(24):124-129.
- [13] 蔡坤. 国内电商移动阅读客户端对比研究[D]. 安徽大学, 2015.
- [14] 刘洪平. 光线自适应的安卓文本阅读器应用[D]. 吉林大学, 2015.
- [15] 廖新辉. 基于语音提示和文字识别的无障碍移动电子书朗读器的研究与实现 [D]. 浙江大学, 2016.
- [16] 任玉刚. Android 开发艺术探索[M]. 电子工业出版社, 2015.
- [17] 何红辉. Android 开发进阶[M]. 人民邮电出版社, 2016.
- [18] 王建功. 基于 Android 的自定义控件设计[J]. 德州学院学报, 2015(2):62-65.

- [19]郑秋梅, 刘真, 苏政,等. 基于 Android 的自定义通用可视化控件[J]. 计算机系统应用, 2017, 26(1):222-226.
- [20]陈之曦, 王晶, 张磊. Android 开源框架中 GIF 格式图片加载模块的应用与移植[J]. 电信技术, 2016(12):31-34.
- [21]张延年, 米洪. Android 应用开发中图片高效处理的研究[J]. 智能计算机与应用, 2015, 5(6):75-78.
- [22]Glide 官方介绍[OL].<https://github.com/bumptech/glide>.
- [23]Y.J. Song, S.Y. Ihm, E.Y. Park, Yunsik Son, and Y.H. Park, A Study on Comparison Analysis of Performance and Usage between Picasso and Glide, In Proceeding of The 4th International Conference on Information Science and Industrial Applications (ISI 2015), Daejeon, South Korea, September 20-22, 2015. work, 1979.
- [24]Song Y J, Soo-Bin O U, Lee J W. An Analysis of Existing Android Image Loading Libraries: Picasso, Glide, Fresco, AUIL and Volley[J]. 2016(imeia).
- [25]Jackson W. Android Image Animation: Frame-Based Animation Using XML Constructs[J]. 2013:217-238.
- [26]李钦华, 刘芳华. Android 图形图像处理技术的研究综述[J]. 无线互联科技, 2015(9):110-111.
- [27]宋国柱. Android 图形图像处理技术研究[J]. 电脑知识与技术, 2014(8):1800-1801.
- [28]李刚. 疯狂 Android 讲义.第 3 版[M]. 电子工业出版社, 2015.
- [29]冯阿芳, 石研. 软件需求分析的思考[J]. 中国新技术新产品, 2010(16):45-45.
- [30]杨毅, 杨杰. 一种提高软件需求分析质量的方法[J]. 计算机系统应用, 2014, 23(5):16-20.
- [31]罗新星, 李柱辉, 赵玉洁. 软件非功能需求国内外研究综述[J]. 计算机应用研究, 2015, 32(4):972-977.
- [32]何红辉, 关爱民. Android 源码设计模式解析与实战[M]. 人民邮电出版社,

- 2015.
- [33]丁鹏, 刘莉. C/S,B/S 模式中的分布式计算与 MTS 解决方案[J]. 计算机应用研究, 2000, 17(12):60-62.
- [34]Zhang J, Zhu X, Liang X. C/S and B/S Mixed Style and the Application[C]// International Workshop on Education Technology and Computer Science. IEEE, 2009:682-686.
- [35]Sokolova K, Lemercier M. Android Passive MVC: Novel Architecture Model for the Android Application Development[C]// Iaria, Patterns. 2013:7-12.
- [36]王倩, 季红. 基于 MVC 的 web 与 Android 双模式在线教学平台的研究与设计[J]. 软件, 2013(10):66-69.
- [37]陆志平, 胡晨骏. 利用接口回调实现 Android 中 Fragment 之间的通信[J]. 现代计算机, 2017(31).
- [38]Yang L, Yuan S, Jiang P, et al. Application Study on Data Storage Mechanism Based on Android[J]. Computer Knowledge & Technology, 2014.
- [39]许姗姗.Android 中 SharedPreferences 知识的深入研究[J]. 安徽电子信息职业技术学院学报, 2017, 16(5):8-12.
- [40]马获蕾, 汤海凤. Android 系统中 SQLite 数据库的研究[J]. 电脑知识与技术, 2013(28):6243-6245.
- [41]申圣兵. Android 开源数据库框架 LitePal 的研究与应用[J]. 中国新通信, 2016, 18(10):49-50.
- [42]欧阳宏基, 杨卫忠, 赵蕾. 观察者模式在 Java 事件处理中的应用研究[J]. 微处理机, 2013, 34(4):77-79.
- [43]卿斯汉.Android 广播机制安全性研究[J]. 电信科学, 2016, 32(10):27-35.
- [44]盛雪丰.Android 开发一大神器--Jsoup[J]. 电脑知识与技术, 2015(8):63-65.
- [45]Wang J, Yang S, Wang Y, et al. The crawling and analysis of agricultural products big data based on Jsoup[C]// International Conference on Fuzzy Systems and Knowledge Discovery. IEEE, 2016:1197-1202.

- [46]Jsoup 官方介绍[OL].<https://jsoup.org/>.
- [47]郭霖. 第一行代码[M]. 人民邮电出版社, 2016.
- [48]黄涛. Android 应用软件测试研究[J]. 计算机教育, 2017(5):103-105.
- [49]李明艳. 基于移动互联网的软件测试技术研究及应用[J]. 电子技术与软件工程, 2016(4):42-42.
- [50]冯亚娜, 刘星妍. 基于移动平台的 APP 软件测试[J]. 信息技术, 2015(4):206-207.
- [51]马瑞芳, 王会燃. 计算机软件测试方法的研究[J]. 小型微型计算机系统, 2003, 24(12):2210-2213.
- [52]刘璐. Android 智能终端功能测试方法设计与系统实现 [J]. 软件, 2014(12):79-82.
- [53]苏敏. Android 系统 UI 性能测试方法的研究[D]. 武汉理工大学, 2015.
- [54]D. Kim, N. Jung, and H. Cha. Content-centric display energy manager for mobile devices. In Proceedings of Design Automation Conference (DAC), ACM, 2014.
- [55]徐宜生. Android 群英传[M]. 电子工业出版社, 2016.
- [56]Huang T, Zhang Z, Zhu X Y. Analyzing GUI running fluency for Android apps[C]// ACM Workshop on Mobile Sensing, Computing and Communication. ACM, 2016:16-22.
- [57]Profiling GPU Rendering Walkthrough[OL].
<http://developer.android.com/tools/performance/profile-gpu-rendering/index.html>.
- [58]高蕾. Android 软件测试技术分析[J]. 电子技术与软件工程, 2017(5):59-59.
- [59]蔡增柱. 基于 Android 移动平台测试相关技术研究[D]. 华南理工大学, 2012.
- [60]Keeping your app responsive[OL].
<http://developer.android.com/training/articles/perf-anr.html>.

致谢

研究生生活转眼就要接近尾声了，回首这三年的时光，我遇到很多人，也经历很多事情，我在不断收获，也在不断成长。在此我要感谢这三年来各位老师、同学、朋友对我的帮助和关爱。

我要特别感谢我的导师袁东风教授的热情关怀和悉心指导。在研究生的三年中，袁东风教授用渊博的知识指导我的科研工作，促使我不断进步。袁老师为我提供了良好的科研条件，让我有机会参与到实际的项目中，在实践中锻炼自己，提升能力。而且，还提供很多与专家、业界优秀人才交流探讨的机会，让我获益匪浅。这三年来，我一点一滴的进步与成长都离不开袁老师的培养和教育。这三年的收获是我一生的财富，使我终生受益，在此向袁老师表示最真诚地感谢。

同时由衷感谢 WMCT 实验室的每一位兄弟姐妹们，自从加入 WMCT，成为其中的一员，我就深切感受到每位成员对我的帮助，很幸运能够加入这样一个团结和谐欢乐的团队。在他们身上我学习到了很多，他们的科研精神和乐观的生活态度都深刻地影响着我。无论在科研工作或者是生活当中，都感谢他们对我的帮助与照顾。三年的时光，我们结下了深厚的友谊，在以后的人生道路上，期待还有与大家携手并进的机会。

衷心感谢我的父母与朋友，他们永远是最坚强的后盾。他们总是默默付出，一直陪伴着我，感谢他们多年以来对我的支持和包容。

最后，非常感谢在百忙之中抽出宝贵时间评阅本论文的各位专家与老师，谢谢！

李淑敏

硕士学位期间的科研成果和参加的项目

申请的国家发明专利

- [1] 袁东风, 李淑敏. 一种安卓手机中下载应用程序的方法, 发明专利,
201710724842.5

参加的项目

- [1] 山东省自主创新及成果转化重大专项: 基于虹计划的教育云系统

学位论文评阅及答辩情况表

论文评阅人	姓 名		专业技术 职 务	是否博导 (硕导)	所在单位	总体评价※	
	匿名					合格	
	白智全		副教授	是	山东大学	合格	
答辩委员会成员	姓 名		专业技术 职 务	是否博导 (硕导)	所 在 单 位		
	主席	刘弘	教授	是	山东师范大学		
	委 员	张海霞	教授	是	山东大学		
		马丕明	副教授	是	山东大学		
		张文胜	副教授	是	山东大学		
		周晓天	讲师	是	山东大学		
	答辩委员会对论文的 总体评价※		良好	答辩秘书	张明强	答辩 日期	2018年5月19日
备注							

※优秀为“A”；良好为“B”；合格为“C”；不合格为“D”。