

论文题目 基于 Android 平台的图书管理系统手机客户端开发

专业学位类别	工 程 硕 士
学 号	200992231003
作 者 姓 名	赵 鲲
指 导 教 师	秦志光 教 授

分类号_____密级_____

UDC ^{注1}_____

学 位 论 文

基于 Android 平台的图书管理系统手机客户端开发

(题名和副题名)

赵 鲲

(作者姓名)

指导教师	秦志光	教 授
	电子科技大学	成 都
	林宏权	高 工
	宁波市科技信息研究院	宁 波

(姓名、职称、单位名称)

申请学位级别 **硕士** 专业学位类别 **工程硕士**

工程领域名称 **软 件 工 程**

提交论文日期 **2013.3.25** 论文答辩日期 **2013.5.4**

学位授予单位和日期 **电子科技大学** **2013 年 6 月 23 日**

答辩委员会主席_____

评阅人_____

注 1：注明《国际十进分类法 UDC》的类号。

A MOBILE APPLICAION OF LIBRARY MANAGEMENT SYSTEM BASED ON ANDROID

A Master Thesis Submitted to
University of Electronic Science and Technology of China

Major: Software Engineering

Author: Zhao Kun

Advisor: Qing Zhiguang

School : School of Information and Software Engineering

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期：_____ 年 _____ 月 _____ 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日期：_____ 年 _____ 月 _____ 日

摘 要

随着信息技术的发展，使得人与人之间的沟通更加方便，引领着人们走向更舒适的生活方式。移动互联网的到来，弥补了传统 PC 台式机的不足。3G 时代的来临，给信息交流带来更大的便捷。Android 操作系统作为目前最广泛的移动终端操作系统之一，近几年来得到了广泛的应用。本文基于 Android 操作系统设计和开发了一套图书管理系统，以满足现代校园图书馆管理的需求。

本文以图书馆管理实际需求出发，研究了系统具体模块的代码编码和优化方法，力求提供便捷的查询服务和友好的功能界面，实现图书借阅管理的人性化、智能化。主要的工作包括分析了图书管理系统的具体需求，设计了系统各个模块的整体框架结构和具体功能的处理流程。并在此基础上以 Eclipse 为开发工具，使用客户/服务器架构，以 Java 语言为开发工具对系统进行了实现。图书管理手机客户端系统实现了用户管理功能，能对用户进行分类管理，提供图书查询功能，可以按照用户提供的查询条件对图书进行查询。同时提供了后台的管理功能。

通过对系统的测试和分析，系统达到了设计目标，能够满足图书管理的日常需求。目前系统已经投入实际的运行，应用效果良好

关键词：ANDROID 系统，图书馆管理系统，软件工程

ABSTRACT

With the development of information technology, the communication between people is more convenient, leads people towards a more comfortable way of life. The arrival of mobile Internet, make up for the shortcomings of traditional PC desktop. The coming of the 3G era, more convenient for the exchange of information. Android operating system as one of the mobile terminal operating system is currently the most widely, in recent years has been widely used. In this thesis, a library management system is designed and developed based on Android operating system, to meet the requirements of modern campus library management.

The thesis analyses the the actual requirements of library management and researches the methods of coding and optimization for the modules of the system. It strives to provide convenient inquiry service and friendly interface in a humanization, intelligent library management system. The main work includes the analysis of the specific requirements of the library management system, design the process framework of each module in the system and the concrete function. And on this basis, using Eclipse as a development tool to realize the system, using Java language as programming language.

Through the test and analysis of the system, it has achieved the goals of design, can meet the daily requirements of the library management. At present, the system has been put into practical application and shows the good effects.

Keywords : android, library management system, software engineering

目 录

第一章 前 言	1
1.1 课题研究的目的及意义.....	1
1.2 课题的研究现状.....	1
1.3 课题的研究内容.....	2
1.4 论文的章节安排.....	2
第二章 相关技术	4
2.1 ANDROID 技术.....	4
2.1.1 Android 概述	4
2.1.2 Android 开发环境搭建	10
2.1.3 Android 的应用程序接口	11
2.2 开发工具介绍和关键技术.....	11
2.2.1 Eclipse.....	11
2.2.2 MyEclipse.....	12
2.2.3 关键技术介绍.....	13
2.3 开发平台搭建.....	14
2.3.1 Eclipse 配置	14
2.3.2 配置 android sdk	16
2.3.3 创建基于 Android 的项目.....	19
2.3.4 启动开发环境.....	22
2.4 本章小结.....	23
第三章 系统需求分析	24
3.1 系统功能用例.....	24
3.2 系统功能分析.....	24
3.2.1 用户注册功能分析.....	25
3.2.2 用户登录功能分析.....	25
3.2.3 图书查询功能分析.....	25

3.2.4 用户管理功能分析.....	26
3.2.5 图书管理功能分析.....	26
3.3 系统其它需求.....	26
3.4 本章小结.....	27
第四章 系统设计.....	28
4.1 系统功能设计.....	28
4.1.1 用户注册功能设计.....	29
4.1.2 用户登录功能设计.....	30
4.1.3 图书查询功能设计.....	31
4.1.4 用户管理功能设计.....	32
4.1.5 图书管理功能设计.....	33
4.2 数据库结构设计.....	34
4.2.1 数据库概念设计.....	34
4.2.2 数据库表结构设计.....	35
4.3 本章小结.....	37
第五章 系统实现.....	38
5.1 系统整体部署.....	38
5.2 通用功能模块的实现.....	39
5.2.1 数据库连接模块实现.....	39
5.2.2 服务访问模块实现.....	39
5.3 具体功能模块实现.....	41
5.3.1 注册模块功能实现.....	41
5.3.2 登录模块实现.....	44
5.3.3 图书查询模块实现.....	50
5.3.4 用户管理模块实现.....	53
5.3.5 图书管理模块.....	55
5.4 本章小结.....	58
第六章 系统测试与分析.....	59
6.1 测试环境描述.....	59
6.2 功能测试.....	59

目 录

6.2.1 用户注册功能测试.....	59
6.2.2 用户登录功能测试.....	59
6.2.3 图书查询功能测试.....	60
6.2.4 其他功能测试.....	60
6.3 性能测试.....	60
6.4 本章小结.....	61
第七章 总结与展望.....	62
7.1 总结.....	62
7.2 展望.....	62
致 谢.....	63
参考文献.....	64

第一章 前言

1.1 课题研究的目的地及意义

移动手机的普及促使移动互联网时代的开始了，同时引领着另一个崭新的时代的到来。常说的移动互联网是指将移动通信网络和计算机互联网合并在一起成为一个通用的互连资源，让各种智能终端能便捷地进行数据服务^[1-2]。

智能手机是已经发展成为移动互联网时代的常用数据交互载体，它拥有普通手机的所有功能，它具有独立的操作系统，可以看作是一个特殊应用的“小型电脑”，能够方便安装、卸载程序，同时具备快速的数据处理和控制功能，可随时随地利用 2G 或 3G 网络进行无线通信。

移动互联网在资本的资助下如弦在箭，近十几年来飞速发展，市场潜力巨大，造就了许多经济神话。Windows 操作系统让微软统造就了 PC 机的繁荣，成就了 PC 操作系统的霸主地位。移动操作系统的诞生让 PC 操作系统的市场地位逐渐转移，Android、Windows Mobile phone 和 iPhone 为首的智能手机操作系统在群雄角逐中脱颖而出，占据了大部分市场份额。这对于广大的手机爱好者无益是一种福音，它不仅提供了一个与外界交互的新平台，又给互联网注入了新的活力。在 Google 推出的开放平台 Android 系统拥有得天独厚的优势，其底层是功能强大的 Linux 开源操作系统，这让许多移动运营商和手机开发商看到了 Android 的光明前景，开始涉足移动业务的公司纷至沓来^[3-10]。

在这样的在时代背景下，Android 操作系统已经在近几年来流行于师生校园，并且成为手机游戏的先锋，也是时尚人士的新宠。Android 手机这样的趋势一定会让消费者感受到无穷的魅力。因此，在 Android 手机开发一套功能全面的图书管理系统是会给学生给校园带来更多的便捷。

1.2 课题的研究现状

当前基于 PC 机的信息网络构建十分健全，可以在电子阅览室进行图书阅览，但是这都在要在固定的地方查询，都是基于 WEB 的网络的服务。为了满足随时随地进行互联网络资源共享，随时了解图书管图书动态，基于移动手机的互联网智

能交互平台是必不可少的。

目前，在移动手机上随时访问图书资源的通讯终端软件很不健全，为了弥补这种不足，基于 Android 的开放平台的图书系统开发是相当实用的。以往都是基于 WEB 的传统应用，在 Android 等智能手机日益风靡全球的今天，基于移动平台的互联网络体验将是一个崭新的话题。

1.3 课题的研究内容

3G 时代的来临，基于移动手机的应用不断涌现，本课程研究的内容也是对 3G 网络应用的挖掘，开发一套适应于校园师生的图书管理系统。这样的系统具备随时接入网络、保护隐私和安全、及时更新互联网的资源，这是面临的新挑战，具有现实重要意义。因此本系统在手机客户端提供查询所使用的软件，设计用户界面。从管理模式上讲，分为服务器和客户端。服务器端响应客户端注册、查询、管理的请求，并把处理完成的数据返回给客户端。在注册时，移动终端获取用户注册的姓名、入学年份、所在学院和性别等信息。当用户选择提交信息时，手机终端通信移动互联网将这些信息发送到服务器，并请求服务器处理，完成注册请求。当服务器收到请求命令，并将新用户插入到数据库，后提示用户注册成功。在登录界面，手机应能接收用户的登录信息，并将请求发送到服务器。在查询时，本系统提供用户书名输入编辑框，提供检索服务。本系统还提供管理身份的用户，能管理注册过的所有用户，并有删除用户和管理图书的权限。

本系统适用于各类图书管的管理，操作简便，满足日常图书维护的需求。本平台采用客户端/服务器架构模式，用 Tomcat 作为服务器处理日常操作，用 Android 系统作为终端开发应用软件。本软件采用 JAVA 语言编写，JDBC 连接 Oracle 数据库实现数据交互。

1.4 论文的章节安排

本文的章节安排如下：

第一章 绪论，介绍论文研究课题的研究目的和意义，课题的研究现状以及主要的研究内容；

第二章 相关技术，介绍课题开发的系统环境，相关开发工具以及使用的关键技术等内容；

第三章 系统需求分析，介绍论文中系统功能性和非功能性需求；

第四章 系统设计，介绍系统的总体架构，系统的功能设计和数据库设计等内容；

第五章 系统实现，介绍系统具体的编码实现过程，并对实现的系统进行展示；

第六章 系统测试与分析，介绍系统的测试过程，并对测试结果进行分析；

第七章 总结与展望，对本文主要工作进行总结，并对未来的工作方向和工作目标进行展望。

第二章 相关技术

在进行 Android 应用软件开发之前，对 Android 系统基础知识和整体设计理念的了解是必不可少的。Android 系统的平台优势是其它系统不可比拟的，它因其操作系统开源，人人都可以为它添砖加瓦。它支持 EDGE、2G、3G 至 4G 网络，不受无线信号网络环境因素的影响，手机网络随意接入已不是难题。下对 Android 做进一步认识。

2.1 Android 技术

2.1.1 Android 概述

在英语中，Android 一词的本身意义指“机器人”，也是在 2007 年 11 月 5 日 Google 同时公布的智能手机操作系统。无线宽带接入技术日趋成熟，移动互联网用户数量急剧增长，Android 移动智能操作系统在这样的时代背景下应运而生。它是由 Google 为统领，由 30 多家公司共同研发的。Android 是移动时代的战略产物，以它为平台的智能手机在全球占领首要地位，是 Google 推进移动互联技术的阶段性成果。Android 系统有以下 5 大特征：应用程序无限共享、系统源代码开放性、应用程序能够并行运行、应用程序可以嵌入到无线网、应用程序平等的创建。Ancroid 研发团队阵容不断壮大，T-Mobile、三星、LG、HTC 也随之加入到平台创新和手机业务的拓展。Android 操作系统是运行在 Dalvik 虚拟机上，非常适合在终端上使用。它能够对 WIFE，蓝牙，相机等外围设备进行扩展，并根据需要开发出自己的驱动程序。由于它底层是开源的 Linux 系统，许多智能设备开发公司纷纷加入 Android 平台扩展阵营，并开发出拥有自身特色的 Android 系统^[11-15]。

Android 操作系统底层是以 Linux 为内核，底层框架是用 C/C++代码为基础的编写的，但是应用程序是基于 Java 语言，同时支持各种数据库的访问，支持 SQL 结构化查询语言。正是由于这种特点，Android 目前可以下面两种方式进行程序开发^[16-17]：

- 1) 基于底层库的 C 语言编程；
- 2) 基于 JAVA 的应用层编程；

从系统本身来看，每一个可独立运行的应用程序都是一个存在于 Android 移动平台的。

在通常的情况下，系统中的应用程序都会运行于 Linux 系统核心进程当中。当 Android 平台应用程序中的任何程序需要执行时，Android 就会运行这个程序，并开启进程，然而这时候其它的应用程序需求系统的资源时并且这个程序又不需要运行时，就让这个进程结束。

在 Android 中任意不同的进程都有独有的一个 VM，所以每个程序的代码与其它程序的代码是互不相干、互不影响执行的。通常，每个 LINUX 用户的 ID 运行一个程序会有不同方式。因此，每个用户的程序运行会不同。当然，也可将不同用户不同系统的程序运行配置权限都设置为一置，可以达到程序执行的同样效果。

应用程序组件：Android 应用程序运行和实例化必须使用组件，如编辑框、列表框、状态栏、进度条、按钮等。如果系统中已经提供了相关组件或自己开发了一个组件时，系统就可以调用，并不是每次使用时都要重复编写。Android 应用程序对程序员来说没有入口地址，它是在系统生存周期里执行。Android 系统运行时必备的四组件如下：

Activity：是关于用户交互界面，捕获用户的日常操作。例如：菜单项列表 activity，它是一个可以供用户选择展示的菜单项列表，并且可以制作要发送消息到的联系人界面，还能作为短消息应用程序，可以设计帮助栏，提示如何给选定的手机用户编写短信息。以及如何浏览以前的短消息和改变设置等。另外，Activity 实例还可以显示一些包含说明文字的照片。

Android 应用程序可以有一个或多个 activity 对象，每个 activity 可以有相应的对话框或设置。每个 activity 的所要完成的功能，以及 activity 的数量，这要关乎我们所开发软件的性质。在默认时，总会有一个 activity 被设置为软件在启动的第一个界面。在程序运行时，用 Activity 可实现从一个界面跳转到另外一个。一个 activity 对象实例运行还能用另外的窗口。比如在 activity 界面正在运行的时候可以弹出的供用户响应的对话框，或者是当使用者选中了某个选项后显示的必要的帮助信息。

View 类均继承自 View 基类，Android 窗口显示的内容都是由一些层次结构的 view 类对象构成的。每个 View 类对象可以控制界面的各自的窗口区域进行显示，并且响应用户所能操作的可控区域。因此，我们可以用 view 类设计出各种便于用户进行交互的界面。软件开发人员可以用 view 类设计游戏窗界面，并在根据玩家的操作完成相应的动作。Activity 是具备一定生命周期的组件，用户的程序可以在

这些周期里激发回调方法。在界面创建时，可以使用 `onCreate` 方法初始化界面，在 `onDestroy` 中销毁控件或释放使用到的内存空间。

Service: 是一个在一定的时间段内处于后台运行，而且没有可视化的用户界面。比如，`service` 可以从远端服务器上请求数据和资源以供 `activity` 对象使用，`service` 还提供后台数据运算和实时处理其它服务的功能。在后台，同时可以启动它在后台提供相应服务功能。应用程序的每一个 `service` 都继承自 `Service` 基类。

例如说，音乐播放器的音乐播放列表就是一个例子。音乐播放器可以有歌曲点选，声场切换，模式切换，背景音效等功能，并提供用户选择歌曲的相关信息。但是，音乐播放器的 `activity` 可以让其它的服务来处理这个隐藏的进程，这样，即使关闭界面后也可以让音乐程序在继续在后台执行，从而达到多任务的目的。`Android` 系统了为了达到后台运行的目的，`android` 音乐播放器 `activity` 可以让一个 `service` 在后台启动。如此程序将不再显示与屏幕上，所以系统的其它程序也一样可以将这个 `activity` 用 `service` 在屏幕后台执行，这样可以无妨碍的进行其它操作。

如果绑定到一个没有运行的 `service`，则将它启动，程序员可以用接口函数实现与 `service` 通信，并执行相关命令。对多媒体的 `service` 来说，该接口能使用户完成一系列的功能，如恢复停止、播放、暂停、上下一曲等等常规的功能。

应用程序进程的主线程内有 `service` 运行的，`activity` 和系统的其它组件类似。它不会对其他组件和交互界面有任何影响，这些线程一般又会派生出新线程来完成音乐回放功能。

Broadcast receiver: 是接收广播通知信息的一个处理组件。许多都是系统代码产生的广播通知信息，比如：通知电池电量变低、时间时区的改变、开关机时的广播，相机打开与关闭，来电提示等，利用这些消息可以通知其它需要使用到的应用程序下载到自己的设备上运行。

一个应用程序可以对所有它认为重要的通知信息予以响应，并使用程序下载到设备上工作。`Broadcast Receiver` 基类是所有应用程序 `receiver` 的父类。

`Broadcast receiver` 同样没有可视化用户界面。然而，我们可以同过两种方式来响应它们收到的信息，第一种方式可以用 `activity` 实例来处理获取的消息，第二种方式使用 `NotificationManager` 实例来响应和处理消息。通知系统响应时，用于引起用户注意力的方式也有很多种：如震动、铃声。

Content Provider: `Android` 系统提供了跨应用程序的数据共享机制，即 `Content Provider`，它支持多应用程序读取或存储数据。当应用程序获得了读取的权限时，它就可以用 `Content Provider` 接口获取音频、图片、数据变量等信息。`Content`

Provider 常用的方法有查询 `query(Uri,String[],String,String[],String)`, 插入 `insert(Uri,ContentValues)`, 更新 `update(Uri,ContentValues,String,String[])`, 删除 `delete(Uri,String,String[])`, `getType(Uri)`等。除了操作自带的 `ContentProvider` 对象, 如通讯录和媒体存取, 还可自己定义 `ContentProvider`, 自己定义的必须要在 `Manifest` 文件中配置和声明。任何 `content provider` 都可以被 `ContentResolver` 进行呼叫连接; 他们互相协作管理相关的进程间的通信。

Android 系统如果接收到某个程序的消息, 并且收到的这个消息需要某一个程序组件处理时, Android 系统就会启动相关模块去执行它, 完成它受需求的任务, 并保证它能顺利执行, 如果这个程序没有开启, Android 会自动调用并开启。

1、激活 intent 组件:

`ContentProvider` 提供了数据调用方法, 但是进行实际操作时我们使用 `ContentResover` 中对应方法。当 `ContentResolver` 的请求 `Content Provider` 时, 收到请求的消息时就会激活。`Intent` 是一种异步消息, 它可以用来激活 `broadcast receive`、`activity` 和 `service` 这几种组件。我们可以在 `intent` 实例中也中存放你所需要发送的消息。它可以作为一切信息的载体, 比如各种实例之间传送的一段文字, 触摸点坐标信息, 文本中正在处理的文字, 或从像机中获取的一幅位图等。

Android 系统每种组件有不同的激活方式:

(1)、Android 系统可以通过 `Activity.startActivityForResult`、`IntentContext.setActivity()`函数来启动, 给一个 `activity` 委任一个新的工作。此 `Activity` 可以使用本身的函数 `getIntent()`查看哪些 `int` 是激活的。Android 通过传递给它随后的任何 `intent`, 出现需要使用 `activity` 来启动另一个 `activity` 的情况, 可以调用 `onNewIntent()` 函数来实现。只有调用 `startActivityForResult()`才能启动它期望的那个 `activity` 返回一个结果, 而 `startActivity()`却不能。例如, 用户挑选一张照片, 如果它所启动了另外一个 `activity`, `activity` 也许想知道哪张照片被选中了。其将会调用 `activity` 的 `onActivityResult()`方法, 将对象封装在一个 `Intent` 中, 并传递给发出。

(2)、`service` 可以向在使用中的 `Context.startService()`发送新的消息, 传递一个 `Intent` 对象至以启动一个 `service`。Android 会调用在此使用 `onStart()`方法将 `Intent` 对象传递给 `activity`。

(3)、应用程序要发起一个广播可以通过传递一个包含消息的 `Intent` 对象到 `Context.sendOrderedBroadcast()`, `Context.sendBroadcast()`, 或 `Context.sendStickyBroadcast()`以及其它类似的方法。

2、关闭组件

`ContentResolver` 只有在 `content provider` 发出请求时处于激活状态，并且仅在响应一条广播信息的时候处于活动状态。它提供了用户界面的 `activity` 则不同，它会一直处于活动状态。同样的，`service` 也可以长时间处于运行状态。因此 Android 系统有很多方法可以用来结束 `service` 和 `activity`。

`Activity` 调用自身的 `finish()` 方法来关闭，而 `Service` 通过调用自身的 `stopSelf()` 方法。启动和关闭一个 `activity` 实例分别调用 `startActivityForResult()` 和 `finishActivity()` 方法时。服务进程 `Service` 通过调用 `Context.stopService()` 方法来停止。

当运行中的组件不会再被使用或不再为系统活动调用其它组件操作时，Android 系统也会自动回收占用的相关内存。

3、manifest 文件

在程序启动之前，系统要保证每个组件是完整的，因此所需要的组件信息会被在根目录 `manifest` 文件中申明，它命名为应用程序 `JAVA` 包。它描述了代码提供商，服务，应用程序活动，以及该应用程序的权限（如相机开启、网络访问）等功能。同时它还申明需要的 `Android SDK` 版本。

`Manifest` 文件是一个结构化的 `XML` 文件，每一个 `AndroidManifest.xml` 文件是 `Android` 应用程序必不可少的，它是 `Manifest` 文件。它会做很多的程序描述，比如声明应用程序要用到的各个组件，指明应用程序需连接到的库名称（某些系统已固定的名称除外），以及应用程序应获得权限的名称，比如拍照权限，连接网络，获取 `SD` 资源权限等。

图 2-1 是这是一个 `activity` 的 `manifest`，它用 `manifest.xml` 文件声明，第一排为 `xml` 的版本号及编码方法，第 3 排为包名称，第 6 排描述了最低 `Android` 的版本号。

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.briup"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="9" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:icon="@drawable/book"
        android:label="@string/app_name" >
        <activity
            android:label="@string/app_name"
            android:name=".LibraryActivity" >
            <intent-filter >
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

图 2-1 manifest.xml 文件

name 名称包含在<activity>之中，它声明了此 activity 的 Activity 子类。label 和 icon 是 activity 的图标和标签的资源文件。

4、Intent 过滤器

Intent 消息可以激活 Android 的三个核心组件—Broadcast Receiver，Activity，Service。它抽象地描述执行某种操作。比如，intent 动作常量 ACTION_CALL 表示发起电话呼叫，ACTION_EDIT 为用户显示编辑的数据，Broadcast Receiver 中的 ACTION_SCREEN_ON 表示显示屏打开等。Intent 消息可以传递数据，图片等信息给另外一个 Activity。还可以更新更新某些操作或传递广播方法。要 Android 找到合适的组件来响应 intent，就必须为 intent 指定一个非显式的目标。该过程需要 Intent 对象和所有目标的 int 相比较才能过滤器完成。组件若有通知 Android 程序处理的 intent 就需要用 intent 过滤器的配合。如图 2-2 所示，其中将 intent 过滤器加入了两个 activity。

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.briup"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".StartActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".InfoActivity">
            <intent-filter>
                <action android:name="com.briup.NEW_ACTION"></action>
                <category android:name="android.intent.category.DEFAULT"></category>
            </intent-filter>
        </activity>
    </application>
</manifest>

```

图 2-2 manifest 文件中声明 intent 过滤器

在上面申明的代码中：`android.intent.action.MAIN`与 `android.intent.category.LAUNCHER` 的结合是第一个过滤器，这种组合在 Android 中是较为常见。它标明了这个 activity 应该在应用程序中启动显示，就是在屏幕显示应用程序的表列供用户启动。

Intent 在 manifest.xml 文件中可以使用任意次，有个需要用到相关功能的地方都要申明。如果 broadcast receiver 在程序中创建注册过，可以不用申明，但是要显式的说明。其它的 intent 过滤器都要在 manifest.xml 文件中进行配置和说明。

2.1.2 Android 开发环境搭建

Android 编程环境主要包括：JDK、Ide 和 Android SDK。最为常见的配置模式是：JDK+ECLIPSE+ADT+ADK(WINDOWS)^[19-22]。

APK 即 Android 安装开发包(anapk), 在英语中是 Android Package 的意思。APK 文件是与 Symbian Sis 系统的 Sisx 文件一样，都是基于手机的安装程序文件。我们只要将这种类型的文件包拷贝到 Android 模拟器或基于 Android 系统的手机中执行就可进行安装。

通过 Android sdk 软件开发工具开发的程序用工具编译成格式为 apk 的安装程序文件，以 apk 为后缀的文件和以 sis 为后缀的文件一样最终放到手机操作系统上进行安装。APK 格式文件类可以用 ZIP 工具解压，它和 ZIP 文件一样，只是扩展名被修改为 apk，通过解压工具解压缩后，可以看到它形成 Dex 文件，Dex 其实全

称是 Dalvik VM executes, 即为 Android Dalvik 的可执行文件, 并不是 Java ME 的字节码, 是 Dalvik 字节码。

Software Development Kit (软件开发工具包) 简称 SDK, 就是指辅助开发人员进行软件开发的相关范例、文档和工具的集合。SDK 工具箱是软件工程师作为各种硬件平台、软件框架、软件包和操作系统等编写应用软件的开发工具, 是使用系统资源必不可少的。我们知道, 一般来说 SDK 即是开发微软公司的 Windows 平台的开发软件所使用的工具。运用 API 接口函数能够快捷的为某个程序实现功能, 和对系统资源的访问, 有时候也可以指包含能与某些硬件平台进行通信的接口函数。不管什么平台的软件开发工具包, 一般会包含支持性的技术注释、程序范例代码或者其它作为参考资料, 比如微软的 MFC, JAVA 的 API 文档手册, 当然 Android 平台也有自己的说明文档, 可以从官方网站上获取。功能强大的软件开发工具应包括实际的代码编写工具, 断点调试运行工具等。

2.1.3 Android 的应用程序接口

Application Programming Interface 的英文缩写是 API, 在 Android 系统中, 它其实就是 Android 操作系统留给代码开发人员的调用接口函数。在 WINDOWS, LINUX 等其它平台也有类似的 API 接口, 应用程序调用操作系统的 API 函数可以实现各种功能, 可以访问系统资源, 同是让操作系统执行相关的命令^[23-27]。

在以命令操作的为主的 DOS 时期, API 已经问世了, 最初的调用 API 的形式是以中断 (INT 21h) 给用户的使用的, 在 DOS 系统运行的可执行代码都直接或间接的使用中断调用方式来完成操作系统某些功能。例如, 我们先将 AH 的值赋为 30h, 用系统中断 INT 21h 就可获得当前 DOS 操作系统的版本信息。但是在现在 Windows 系统中的 API 非常全面, 它提供给用程序员各种各样的函数以供使用, 这些接口都是以函数方式实现的。在 Windows 中同样可以在程序中调用 GetVersionEx() 函数, 即可以取得当前操作系统的版本信息。

2.2 开发工具介绍和关键技术

2.2.1 Eclipse

Eclipse 是集成 JAVA/C/C++ 语言的开发平台, 它是通过插件扩展开发环境, 可以集成 Android SDK, CDT, JDT 来支持 Android 等应用软件的编写。无论在 Windows

还是 Linux 平台它都有自己的版本,可以做到与平台无关的应用程序开发。Eclipse 核心代码很小,所有插件都依靠插件才能运转。Eclipse 由于其代码开源,包括 Oracle, Borland, Rational Software 在内的公司也加入到 Eclipse 扩展阵营^[28-31]。

Eclipse 起始于 1999 年,由多家公司联盟开发了一个功能强大的软件开发平台。它的历史及版本信息如表 2-1 所示。

表 2-1 Eclipse 的版本号

版本代号	平台版本	主要版本发行日期	SR1 发行日期	SR2 发行日期
Callisto	3.2	2006 年 6 月 26 日	N/A	N/A
Europa	3.3	2007 年 6 月 27 日	2007 年 9 月 28 日	2008 年 2 月 29 日
Ganymede	3.4	2008 年 6 月 25 日	2006 年 9 月 24 日	2009 年 2 月 25 日
Galileo	3.5	2009 年 6 月 24 日	2006 年 9 月 25 日	2010 年 2 月 26 日
Helios	3.6	2006 年 6 月 23 日	2006 年 9 月 24 日	2011 年 2 月 25 日
Indigo	3.7	2006 年 6 月 22 日	2006 年 9 月 23 日	2012 年 2 月 24 日
Juno(预定)	3.8 及 4.2	2006 年 6 月 27 日	2006 年 9 月 28 日	2013 年 2 月 22 日
Kepler	4.3	2006 年 6 月 26 日	2006 年 9 月 27 日	2014 年 2 月 28 日

Eclipse 集成环境 (IDE) 对 JAVA 的支持其它软件是无可比拟的,所有插件,都是开源的,遵守共同的约定章程。Eclipse 的大小可根据自己的情况灵活裁剪,它支持各种语言,目前对 C/C++、COBOL、PHP 等编程语言的支持也日趋完善。Eclipse 安装方法分三种:直接复制法、连接法或使用 Eclipse 自带的图形界面安装。

Eclipse 还自带有一个版本的 JAVA JDK,因此在本计算机没有安装 JDK 的情况下,可以完成一般程序开发。Eclipse 平台在开发大范围工作站如 JAVA WEB 也较为广泛。它支持的平台包括 Linux、Solaris、QNX、Mac OS 以及其它平台。

Android 工程就是使用 Eclipse 集成开发环境进行开发的,要运用 Eclipse 进行程序开发,首先要安装 JDK 并配置环境变量,再下载 Android SDK,并安装。如果没有真实手机,可以安装 ADT 插件,安装虚拟机后就可以进行系统仿真。具体细节将在下面章节介绍。

2.2.2 MyEclipse

MyEclipse 是一个功能强大的开发工具,它是对 Eclipse 插件的扩展,融合了各种开源插件,对 JAVA 语言各种开发框架进行整合。它支持 JAVA 前台、后台开发、支持 Hibernate, Ajax, JSP、JDBC 等数据连接工具。它自身带有数据库操作访

问功能，能简易的用 SQL 语言对数据进行创建，删除，插入。它自身带有服务器，能对编写的网页实时浏览^[32-35]。

MyEclipse 几乎包括了所有开源产品的插件，并可以随时对单独模块升级。MyEclipse 可运行在 Windows 操作系统、Mac 操作系统和 Linux 操作系统平台，支持 Java EE 部署和测试。

MyEclipse 是当今 Java 开发使用最为普遍的工具之一，它是对 Eclipse 开发框架的扩展，集成 WEB 开发工具，支持 Spring、Hibernate、Spring Security、JavaBean、HTML5 和 JavaEE 6。MyEclipse 的发布大大加强了开发进度，缩短了时间，节约了成本，它有完整的代码编写、断点调试、资源发布、后期评测的工具。它对 Java Servlet、AJAX、JSP、JDBC 数据库访问性能卓越，因此选用它作为服务端的开发。

下面是 MyEclipse 所具有的功能和支持的服务：

1. 对 EJB 的支持；
2. 对 JAVA WEB 动态网页的开发；
3. 对 MySql、Oracle、SQLite 数据库的支持；
4. MyEclipse 整合帮助；
5. JavaEE 项目的开发；
6. JavaEE 开发；
7. 连接各种服务的支持。

本系统采用 Browser/Server 架构模式，用 MyEclipse 自带的服务器或 Tomacat 服务器提供数据服务。需要解决的是创建不同的 servlet 来提供不同的服务。

2.2.3 关键技术介绍

Android 的核心技术包括资源的访问、用户 UI 界面设计、基本单元 Activity 的使用、组件之间通信的 Intent 信使、Android Service 组件、Android 广播事件处理 Broadcast Receiver、Android 中的数据存取、数据共享共享工具 Content Provider、Android 系统的多媒体应用。

在 Android 中，外观布局一般使用 XML 文件，它在工程目录的 res\xml 下，如果用户使用了 XML 文件，可以使用 Resources.getXML()函数访问它。Android 对 asserts 文件的资源的访问是不能直接进行的，但它可以用 AssertManager 以二进制来读取。所有的音频，图像，视频，动画等资源都在 res 文件夹下，动画或图片放在其子目录 drawable 下，程序中可以使用 getResouse()方法加以调用。Activity

是用于展示在手机界面上的布局容器，上面可以有进度条、日历表、按钮、TextView、ListView 常见组件。它有一定的生命周期，它是系统通过回调我们自己编写的类来实现一定功能。

2.3 开发平台搭建

2.3.1 Eclipse 配置

可以使用 Eclipse 自带的工具对插件进行安装，第一步要打开 Eclipse，在菜单栏上依次选择 help->Install New SoftWare 当我们见到图 2-3 界面时：

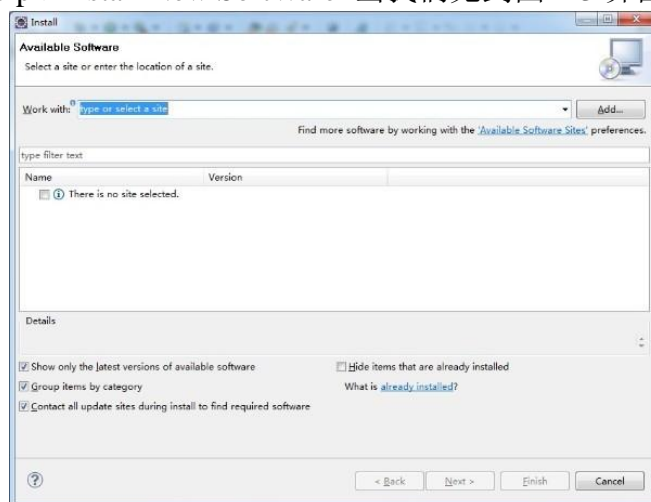


图 2-3 其它软件安装界面

就点击 右边的 Add 按钮，图 2-4 所示的界面就展现出来

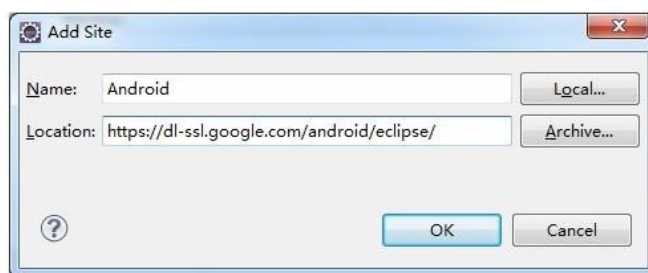


图 2-4 路径选择

在下排位置指定栏输入: `https://dl-ssl.google.com/ android/eclipse/`，如果在连接时出现错误，只要将 `https` 改换成 `http` 就可以了，name 栏中的名称：可以自定义合规的字符，如：Android，然后点击 OK，将出现图 2-5 所示的下界面：

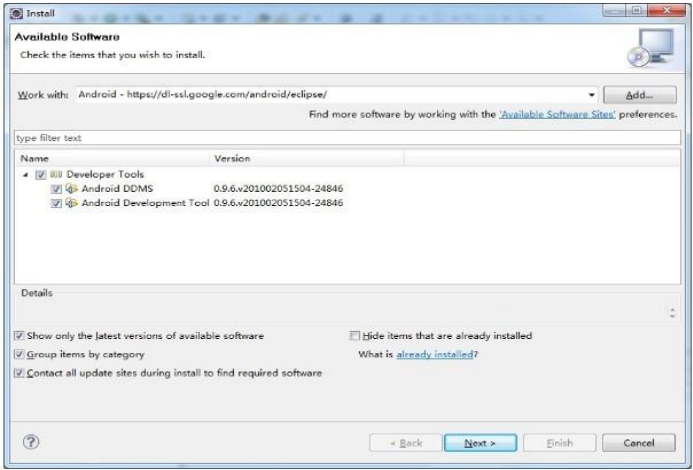


图 2-5 选择要安装的软件

点击 Next 按钮,出现如图 2-6 所示的界面:

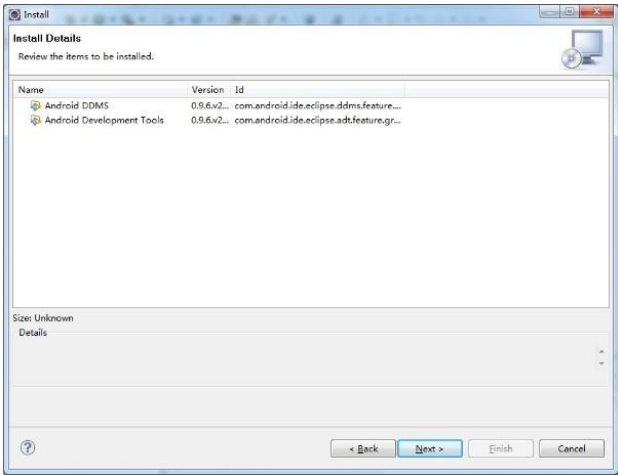


图 2-6 选择安装

再点击 NEXT,出现如图 2-7 所示的界面:

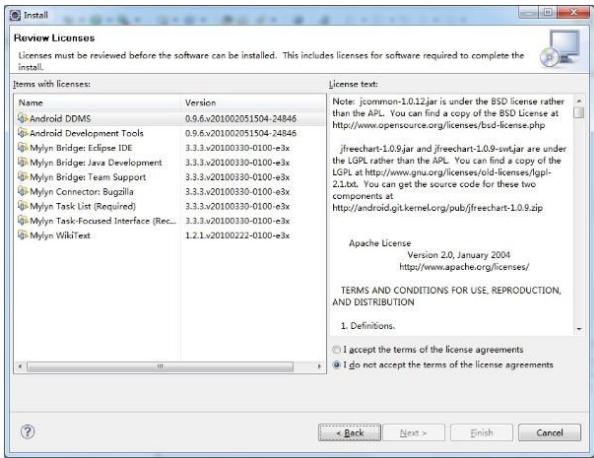


图 2-7 详细选择

选择 I accept the terms of the license agreements 项，再点 Next，进入插件安装界面：

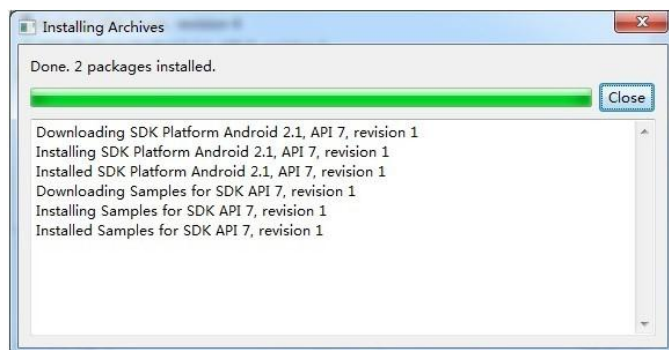


图 2-8 插件安装界面

当安装完成出现如下窗口：



图 2-9 完成安装

点击左边的 Yes 按钮，再重启 Eclipse 软件。

2.3.2 配置 android sdk

(1)在 eclipse 菜单栏点击菜单 window->preferences，出现如图 2-10 界面：

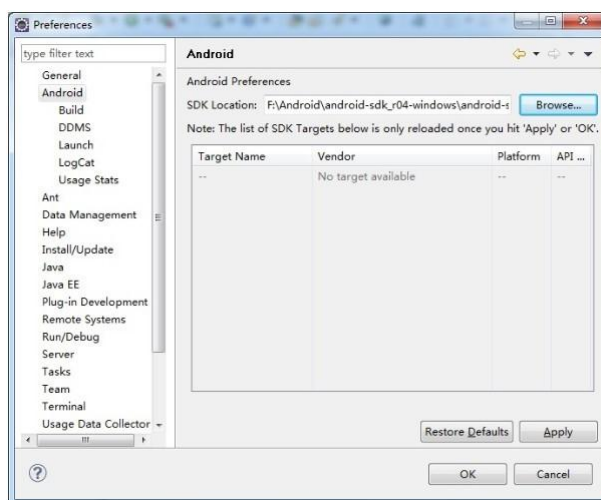


图 2-10 首选项配置

然后，选择你的下载的 Android SDK 解压路径，如果选错了会提示错误，这是升级的工具，目前里面还没有一个 Android 操作系统版本的 SDK

(2) 在 eclipse 菜单栏依次选择 window->Android sdk and avd manager 就可以对 SDK 版本升级，单击升级后会搜索更新的版本，得到如图 2-11 所示的界面：

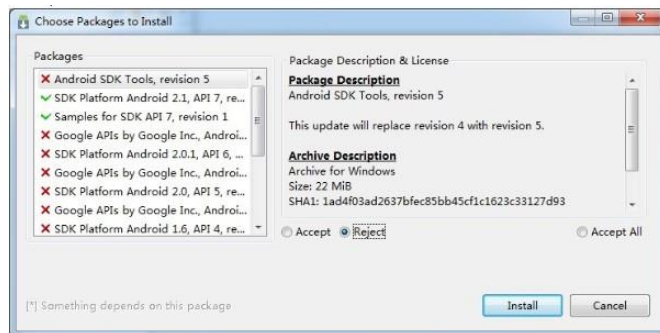


图 2-11 选择要安装的部件

先选中左边的某一些版本，点击 accept 开始 SDK 安装，如果不安装点击 reject 即可。本系统选择了 Android 2.1 及 Android2.2 版本，并选择右边的 Accept 单选按钮，并点击 install，就可以进入如图 2-12 所示的安装界面：

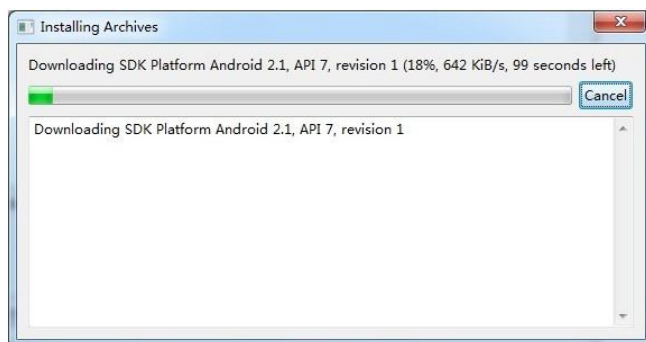


图 2-12 安装 SDK

安装完成后会出现如下栏：

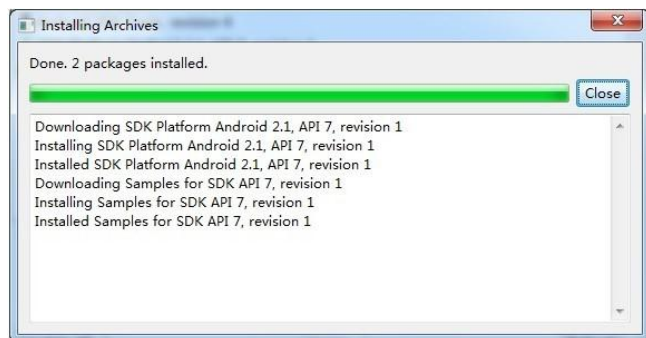


图 2-13 安装完成图示

(3)新建虚拟机 AVD 和前面提到的方法相同，进入 Android 虚拟机管理工具，选中其中的 Vitural Devices ，然后在点选 New 按钮，出现界面如图 2-14 所示：

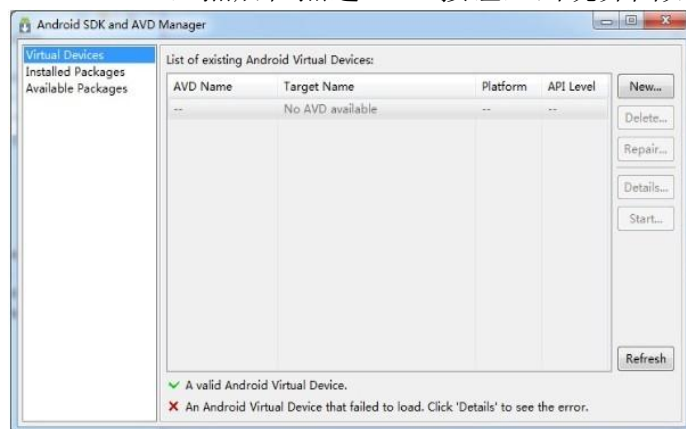


图 2-14 新建虚机

点击右边的 New 按钮后，就可进入如下界面：

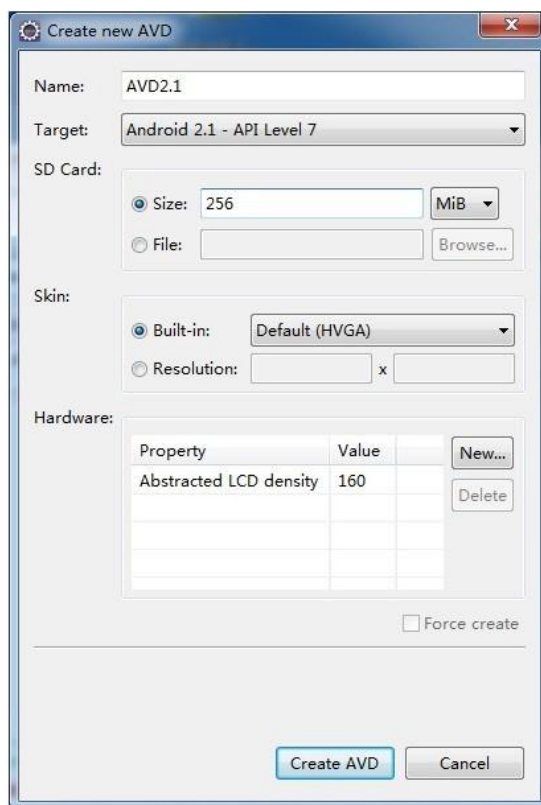


图 2-15 虚拟机参数设置

填写一个名字，在 arget 里选择你所需要的 SDK 版本名称。自定义 SD 卡的大小，然后点击 Create AVD，出现如图 2-16 的提示窗口：



图 2-16 创建完成

这样 AVD 就创建好了

2.3.3 创建基于 Android 的项目

1) 依次选择 file->new->other 进入图 2-17 所示界面:

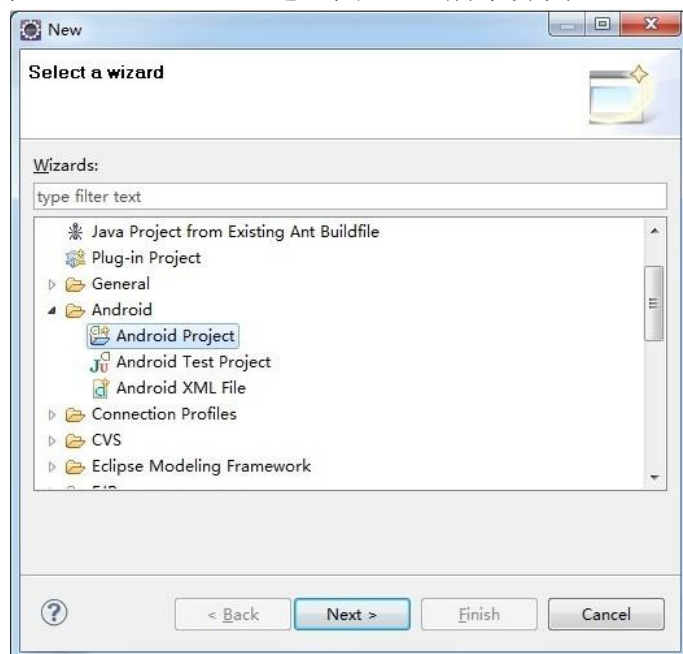


图 2-17 新建项目

选择新建 Android 栏下面的 Android Project 项目, 点击 Next 按钮后, 如图 2-18 所示:

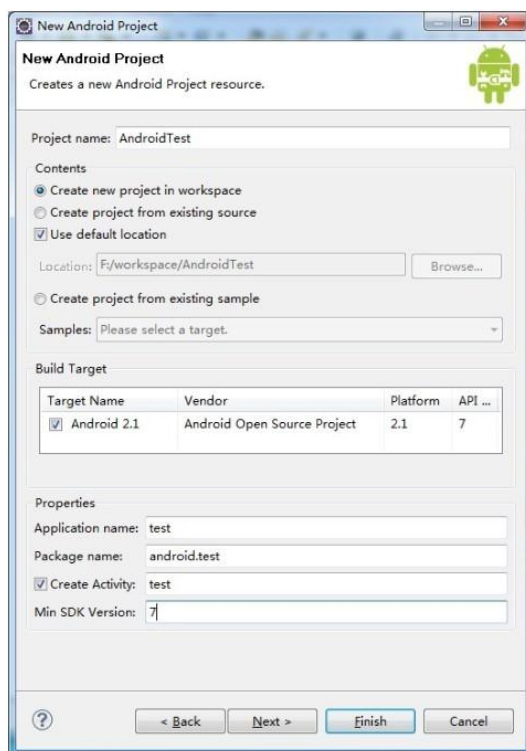


图 2-18 项目设置

这里在 **Project name** 栏中填写项目名称，**Activity** 名称，并且指定运行此程序的最小版本数。然后，点击上面界面的 **Next** 后，出现如图 2-19 界面。

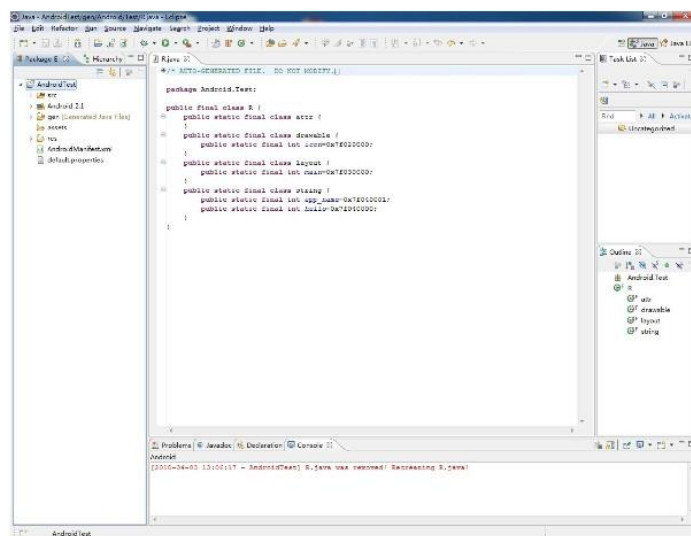


图 2-19 建好项目后画面

在某些情况下，如 `xml` 文件错误，会造成 `R` 文件丢失，程序不能运行。如果 `xml` 文件有问题，`R.java` 文件是不会自动生成的，在这种情况下，只要将项目清理一下，就能修正的错误。此外 `R` 文件也可以通过 `Eclipse` 工具手动生成。

2) 配置运行项

在文件管理界面，鼠标右键点击项目名称->Run as -> Run Configuration 进入如图 2-20 所示配置界面

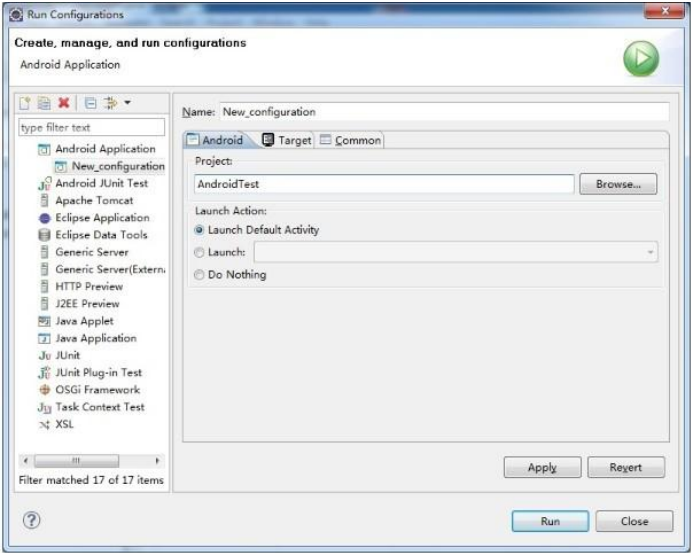


图 2-20 项目配置

在此界面上，点击 Browse 按钮，选择左边你要运行的项目名称选择其中的 Target，立即切换到图 2-21 界面：

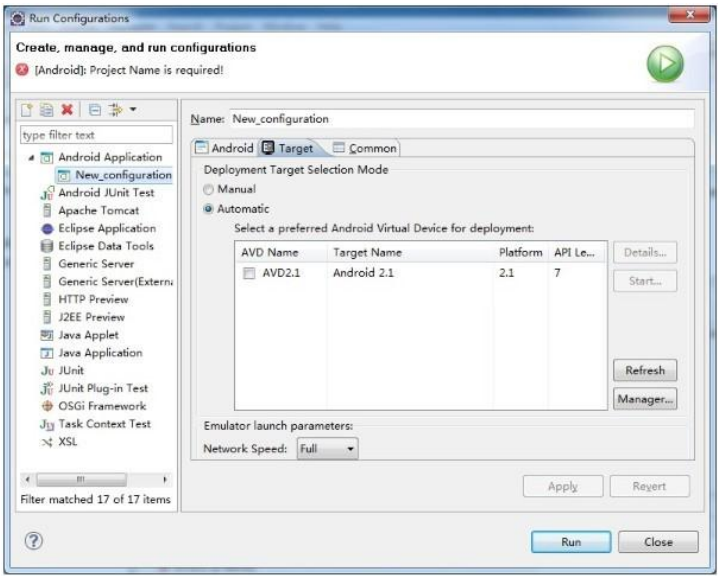


图 2-21 选择虚拟机版本

将此界面的 AVD 选择状态设置为选择运行的。

3) 用虚拟机测试项目

在左侧文件管理器中，依次鼠标右键项目名称->run as ->Android Application，

这时系统会打开 ADT 模拟器，如图 2-22 所示：



图 2-22 虚拟机开始启动

画面将会显示正在进入标志，1-2 分钟后，就可启动虚拟机，并运行应用程序



2-23 程序运行示例

运行的结果将显示如下虚拟机上。

2.3.4 启动开发环境

到此为止，Android 开发环境就搭建好了，因为没有下载具体某个版本的 SDK

工具，只下载了 android sdk r4 的升级工具，如果要下载某个具体版本，只要在 Android SDK 管理工具中选择，并下载就可以了。也可以 Google 官方网站下载相应的版本，并存放在 SDK 的配置目录下，只是使用 SDK 升级工具更加方便快捷，易于管理。

进入虚拟机管理界面后，点击开始按钮便可以运行虚拟机，虚拟机启动界面后如图 2-24 所示。



图 2-24 android 虚拟机视图

就像真实的手机界面一面，虚拟机模拟程序的整个运行，除了不支持照相机，红外线等模拟外，其它程序都可以虚拟的方式展现给用户。虚拟机有各种不同的版本，最好选择以后真实手机的版本，这样解决了没有真实手机就可以进行程序开发的目的。

2.4 本章小结

本章详细介绍了 Androd 手机开发环境的搭建流程，包括 Eclipse 软件、JDK、Android SDK 的下载、以及虚拟机的建立。另外，本章还演示了 Android 工程的建立方法和程序的配置运行。

第三章 系统需求分析

本系统主要完成图书管理的基本业务流程，系统中的用户包括普通用户和管理员用户两种类型，系统整体功能需求可划分为前台功能和后台管理功能两大部分。

3.1 系统功能用例

系统中普通用户使用的功能包括用户注册、用户登录、图书查询具体的系统功能，管理员用户使用的功能包括用户管理和图书管理功能，如图 3-1 所示。

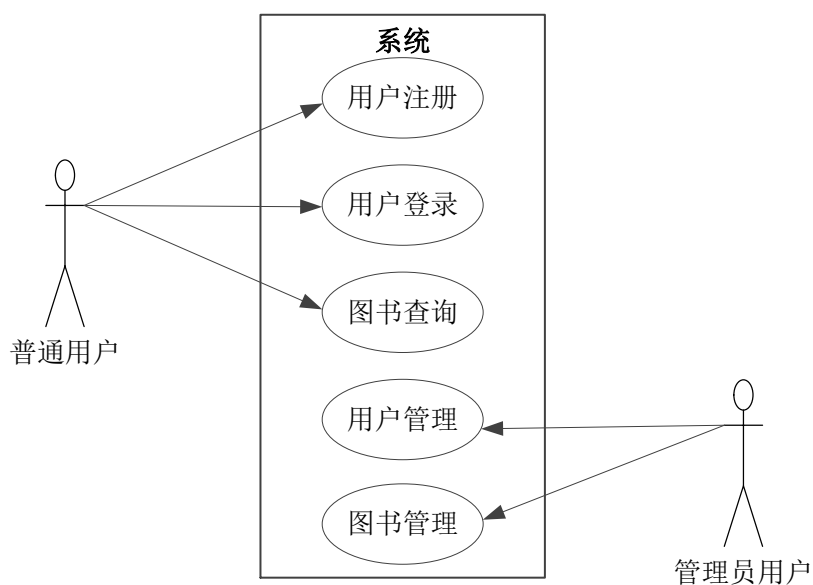


图 3-1 系统功能用例图

3.2 系统功能分析

本移动图书馆系统采用客户/服务器模式进行架构，因此每一功能分别对应前台的功能需求和后台的管理功能需求。

3.2.1 用户注册功能分析

用户注册是图书管理系统中的基本需求，通过注册功能可以实现用户在系统中的信息注册。普通用户可以使用学号，姓名，密码，入学年份，性别等信息向系统提出注册申请，通过系统的注册服务对信息的审核后，服务器的会在数据库中保存用户的注册信息，并将用户接收为系统的合法用户，用户则可使用注册信息登录系统使用相关的功能。

注册功能的前台主要实现用户信息的接收和发送，用户在客户端的注册功能界面中提交注册信息后，前台会将信息发送给服务器处理，当服务器接收到用户的注册请求后，并启动后台的注册服务程序，完成用户注册数据的审核和更新操作。如果注册信息符合系统要求，如不存在同名用户，则注册成功。否则提示用户注册信息冲突，用户注册失败需重新输入符合要求的注册信息。

3.2.2 用户登录功能分析

用户登录实现对用户访问的基本控制，判断用户属于普通登录还是管理员登录，并且根据用户登录的不同身份显示不同的用户界面和访问权限。

用户登录时通过用户登录界面输入用户登录的用户名和用户口令，并由前台程序发送给服务器，后台用户登录服务程序会核查普通用户表中的用户信息是否正确，如果用户名和用户口令一致，则允许用户登录，并返回登录成功信息，前台收到登录成功信息后进入指定用户功能界面。

3.2.3 图书查询功能分析

图书查询实现普通用户对图书的检索功能，合法用户可进入查询界面。图书查询支持精确查询和模糊查询两种查询方式。

用户设置查询条件后，由前台程序将服务查询请求提交给后台图书查询服务程序，服务程序响应前台的查询请求，对数据库中的图书信息进行查询，并返回符合条件查询结果，前台程序收到查询结果后，在前台用户界面上显示查询结果。包括图书的相关信息，如图书名、出版社、出版时间等。

3.2.4 用户管理功能分析

用户管理实现管理员用户对普通用户的维护和管理，主要的管理需求包括用户查询、用户锁定、用户删除等。当普通用户存在违规行为时，如图书借阅超限未还，可对用户账户进行锁定，限制用户对图书查询等基本功能的使用，当某些用户账户长期不使用或是存在严重违规的情况是还可将用户从系统中删除，实现对用户的注销。

前台程序主要提供管理员的操作界面，可接收管理员的用户管理信息，生成用户管理请求，并发送给后台的用户管理服务程序，后台服务程序接收到管理员用的用户管理请求后，对数据库中的用户信息进行操作，如查询、锁定和删除，并将结果返回给前台程序，前台程序将在用户界面上显示用户管理的结果。

3.2.5 图书管理功能分析

图书管理实现对图书信息的维护、主要包括图书信息的增加、修改、删除等操作。管理员用户可对图书管理，前台程序提供图书管理的用户界面，根据管理员的操作，并将相应的管理请求发送给后台图书管理服务程序，后台服务程序接收到管理员的图书管理需求，对数据库中的图书信息进行操作，如增加、修改、删除，并将结果返回给前台程序，前台程序将在用户界面上显示图书管理的结果。

3.3 系统其它需求

系统的非功能性需求主要包括性能需求、易用性需求和安全性需求。

1、性能需求

要求系统的性能满足应用需要，系统最大响应时间低于 15 秒，平均响应时间低于 7 秒，最大并发数不低于 50。

2、易用性需求

要求系统提供友好的用户图形界面，用户无需专门培训即可对系统进行正常的的使用。

3、安全性需求

要求用户登录系统，必须提供合法的用户信息，合法用户才能进行下一步操

作。系统需要区分不同用户，并实现不同的用户访问控制管理。

3.4 本章小结

本章对移动图书管理系统的需求进行了分析，主要包括系统的功能的用例分析，具体功能的处理流程以及系统的其他非功能性需求。

第四章 系统设计

4.1 系统功能设计

本系统里采用客户端/服务器架构模式，服务器完成数据的管理和对远程的服务请求作出响应，客户端就是请求获得数据的终端，本系统中主要指智能手机，因此系统平台选用 Android 操作系统。服务端与客户端的分开管理，让完成开发任务的程序员分工更明细，更便于整个系统的开发。

从管理员对图书的管理操作和用户对图书信息的需求两个角度来看，用 ANDROID 操作系统开发的数据库查询界面应该简捷适用，同时也应有自己的特色。此系统将用户分为两类，图书管理员和普通用户。在客户器端主要完成的功能有登录、注册、对普通用户权限设置、密码找回、图书查询、图书入库添加书目共等几大模块；对图书管理系统中普通普通用户来说应该有登录、图书查询、注册、密码找回等功能；本系统将管理员给予最高权限，它有对图书信息的修改，查询，增加新书的功能，还能对普通用户进行删除，权限分配。对于普通用户，只能在前台客户端进行书目检索。然而，在后台只有管理才能对数据库进行的操作。前台和后台相辅相成，成为一个完整的整体。系统的整体功能结构如图 4-1 所示。

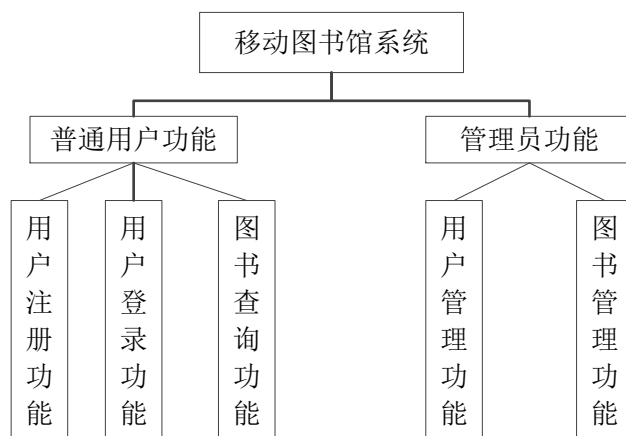


图 4-1 系统整体功能结构图

4.1.1 用户注册功能设计

用户注册功能设计包括界面设计和处理流程设计，分别介绍如下。

1、用户注册功能界面设计

用户注册界面中包含用户名、密码、证件号等基础用户信息，如图 4-2 所示。

Mobile Library

用户注册

用户名

密 码

证件号

确认

重置

取消

V1.0版

4-2 用户注册功能界面

2、用户注册功能处理流程设计

注册功能的处理顺序为，用户输入注册信息，前台程序生成注册请求发送给服务器，服务器向数据库发出注册操作，数据库进行相应操作并返回操作结果，服务器根据结果生成操作响应，并返回个前台程序并在用户界面上显示，如图 4-3 所示。

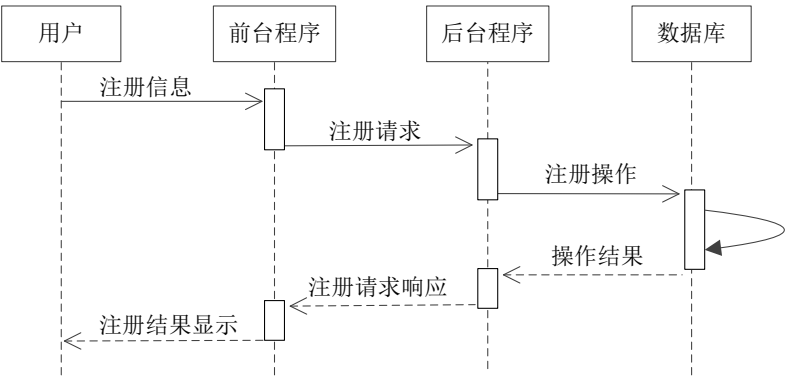


图 4-3 注册功能顺序图

4.1.2 用户登录功能设计

1、用户登录功能界面设计

用户登录功能界面元素包括用户名和用户密码的输入，以及用户类型选择等相关信息，如图 4-4 所示。

图 4-4 登录功能界面设计

2、用户登录功能流程设计

用户登录功能的处理顺序为，用户输入登录信息，前台程序生成登录请求发送给服务器，服务器向数据库发出用户登录信息验证操作，进行相应操作并返回操作结果，服务器根据结果生成操作响应，并返回前台程序，验证成功进入用户界面，否则显示验证失败提示，如图 4-5 所示。

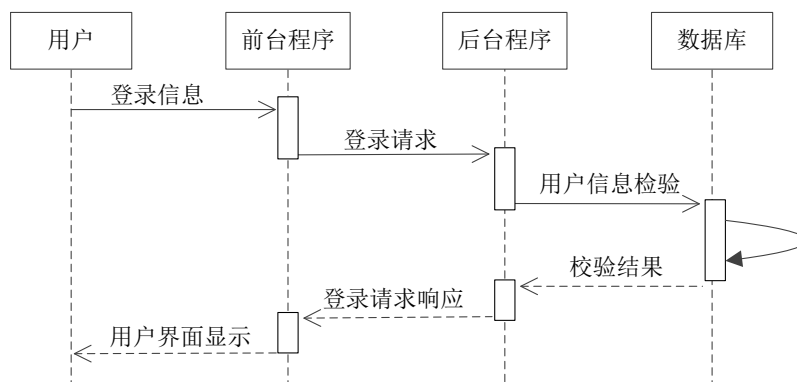


图 4-5 登录功能处理顺序图

4.1.3 图书查询功能设计

1、图书查询功能界面设计

图书查询功能界面包含查询条件输入的主要选项，图书名、著作者、出版社，以及出版时间。如图 4-6 所示。

Mobile Library

图书查询

图书名

著者名

出版社

出版时间

2013年5月

周	周	周	周	周	周
	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31				

确认

重置

取消

V1.0版

图 4-6 图书查询界面设计

2、图书查询功能流程设计

图书查询功能根据用户输入的查询条件在后台数据库中检索符合条件的图书，并返回用户界面显示。图书查询功能处理顺序如图 4-6 所示。

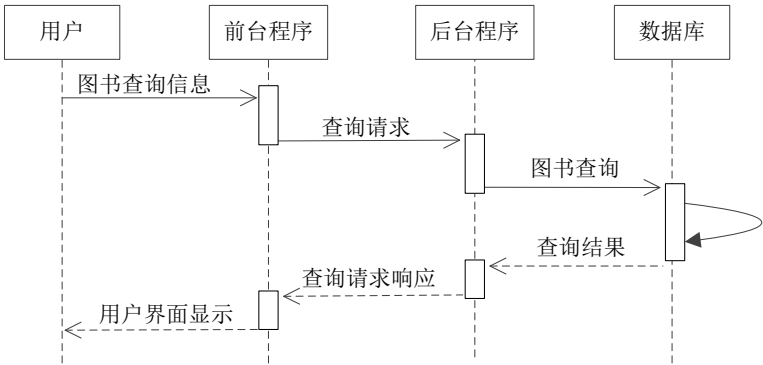


图 4-6 图书查询功能处理顺序图

4.1.4 用户管理功能设计

1、用户管理功能界面设计

用户管理界面包括用户名，证件号，注册时间等输入，可进行查询、修改、删除等操作，用户管理界面设计如图 4-7 所示。

图 4-7 用户管理界面设计

2、用户管理功能流程设计

管理员用户可根据用户名、证件号等用户信息，对系统注册用户进行查找和列举并执行相应的管理操作，用户管理功能处理顺序如如图 4-8 所示。

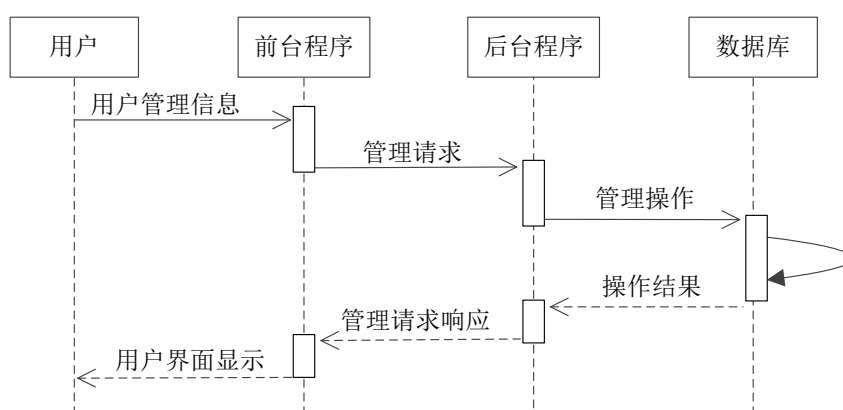


图 4-8 用户管理功能处理顺序图

4.1.5 图书管理功能设计

1、图书管理功能界面设计

图书管理功能界面包括图书名、著作者、出版社等图书信息的输入，可进行图书的增加、修改、删除图书信息等功能，图书管理功能界面设计如图 4-9 所示。

Mobile Library

图书管理

图书名

著者名

出版社

出版时间

2013年5月

周	周	周	周	周	周
	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31				

查询

修改

新增

删除

重置

退出

V1.0版

图 4-9 图书功能界面设计

2、图书管理功能处理流程设计

管理员可通过图书管理功能对图书信息进行维护，图书管理功能处理顺序如图 4-10 所示。

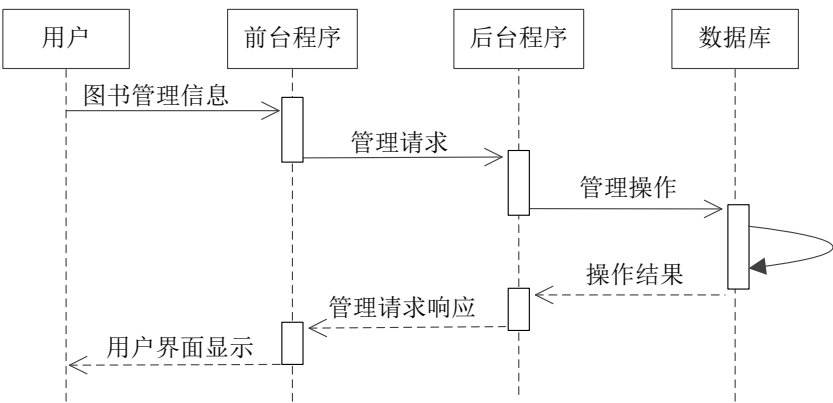


图 4-10 图书管理功能处理顺序图

4.2 数据库结构设计

本系统中数据库选用 **MYSQL** 数据库，以下简要介绍数据库的设计过程，主要分为概念设计和表结构逻辑设计两部分。

4.2.1 数据库概念设计

本节中对系统中数据对象的属性和对象间的关系进行分析，并由此得出数据对象的具体的实体关系描述。

系统中的管理员实体主要包括两个属性用户名和密码，同时为了方便使用将对管理员分配一个唯一的身份标识 (**ID**)，管理员实体关系如图 4-11 所示。

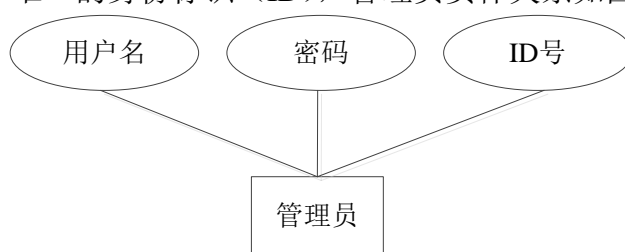


图 4-11 管理员 E-R

系统中的普通用户实体主要包括用户名、密码、证件号、注册时间等属性，为了便于对用户进行统一管理，对用户分配唯一的 **ID** 作为标识符。普通用户实体关系如图 4-12 所示。

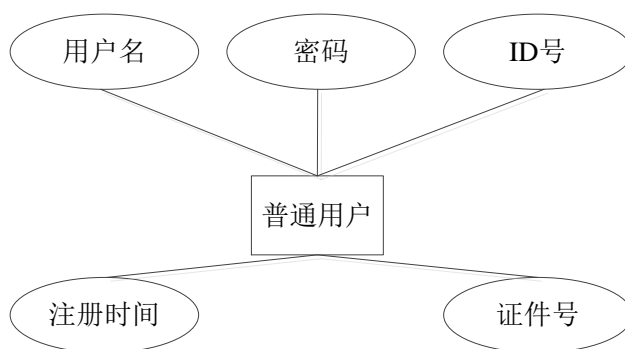


图 4-12 普通用户 E-R

系统中的图书实体包含各种相关属性，书名、出版社名称、作者名、图书状态、借阅时间信息等，当图书增加成功后会为其分配唯一的标识符 **ID**，结合书名绑定为主键，图书实体关系如图 4-13 所示。

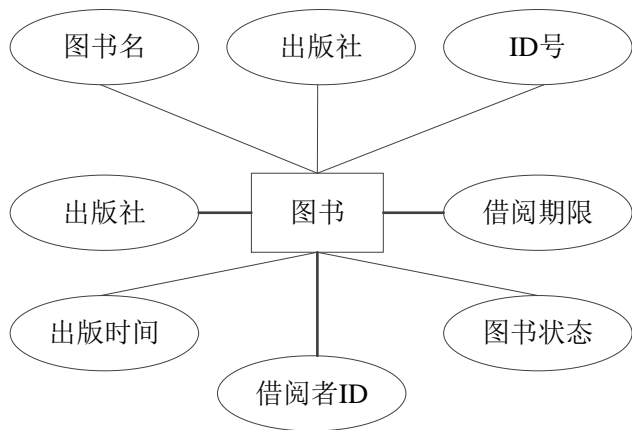


图 4-33 图书 E-R

系统中的借出图书实体包括图书名、图书编号、借阅者 ID，图书作者、外借时间、最大外借期限等，外借图书实体关系如图 4-34 所示。

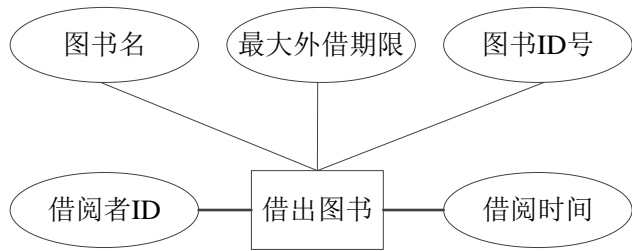


图 4-34 外借图书 E-R

各个系统实体之间关系如图 4-15 所示。

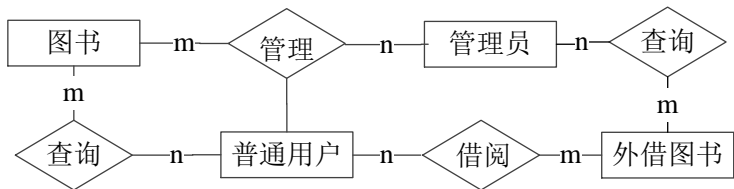


图 4-35 总体关系图

4.2.2 数据库表结构设计

结合对系统中数据实体的 E-R 图可对数据库中的表结构进行设计，具体结果如下。

1、管理员数据表

管理员数据表中存储管理员的用户信息，如表 4-1 所示。

表 4-2 管理员表

字段名	数据类型	长度	主键	备注
UsrId	自动编号	10	是	AUTO_INCREMENT
Usrname	Varchar2	10	否	管理员的登录账号
Passwd	Varchar2	20	否	管理员登录的密码

2、普通用户数据表中保存普通用户的基本信息，如表 4-3 所示

表 4-3 普通用户表

字段名	数据类型	长度	主键	备注
ID	Int	4	是	AUTO_INCREMENT
Usrname	Varchar2	10	否	唯一标识用户，不可缺
Passwd	Varchar2	20	否	普通用户用户登录密码
CertId	Varchar2	20	否	用户注册的身份证号
Time	Date	4	否	用户注册时间

表 4-4 图书表

字段名	数据类型	长度	主键	备注
ID	Int	10	联合主键	AUTO_INCREMENT
Bookname	Varchar2	50	联合主键	图书名称
author	Varchar2	10	否	图书作者名
publisher	Varchar2	50	否	出版社名称
publishtime	Date	4	否	出版时间
status	Boolean	1	否	图书状态
timelimit	Int	4	否	最大借阅时间
BorrowerID	Int	4	否	借阅者 ID

表 4-5 外借图书信息表

字段名	数据类型	长度	主键	备注
ID	Int	4	是	AUTO_INCREMENT
Borrowdate	Date	4	否	外借时间
timelimit	Int	4	否	最大的外借时间
Book_id	Varchar2	10	否	图书编号

Bookname	Varchar2	50	否	图书名字
----------	----------	----	---	------

续表 4-5 外借图书信息表

字段名	数据类型	长度	主键	备注
Author	Varchar2	10	否	著作者名
User_id	Varchar2	10	否	借阅者 id

4.3 本章小结

本章中介绍了系统的设计过程，包括系统的整体功能结构，系统功能的界面和处理过程设计，以及系统的数据库设计过程。

第五章 系统实现

5.1 系统整体部署

本系统采用客户服务器模型，整体架构使用 MVC 三次架构，分为客户端、业务处理服务器、数据库服务器。服务器选择 Tomcat，在客户端单独开发。

在 Tomcat 中使用 servlet 发布服务功能。客户端和业务处理服务器之间的互连接采用 http 协议连接请求。因为客户端不使用通用的浏览器，因此在 Tomcat 和客户端的通信中需要通过对象序列化作为信息传输的内容。

移动图书管理系统主要分为了三个部分组成。分别为客户端、图书管理服务器和数据库服务器。客户端提供用户操作界面，图书管理服务器作为业务处理，数据库服务器作为数据处理。整体处理过程如下：

- 1) 客户端收到用户操作请求后，对请求操作的实体信息进行对象序列化，通过 http 协议发送请求给图书管理服务器；
- 2) 图书管理服务器接受到请求后，接受信息，执行反对象序列化，获得其操作的实体类对象。随后进行处理，并把响应信息序列化返回给客户端。
- 3) 客户端接受到响应信息后，后进行反序列化，并且显示相应的结果。

系统的整体功能部署如图 5-1 所示。

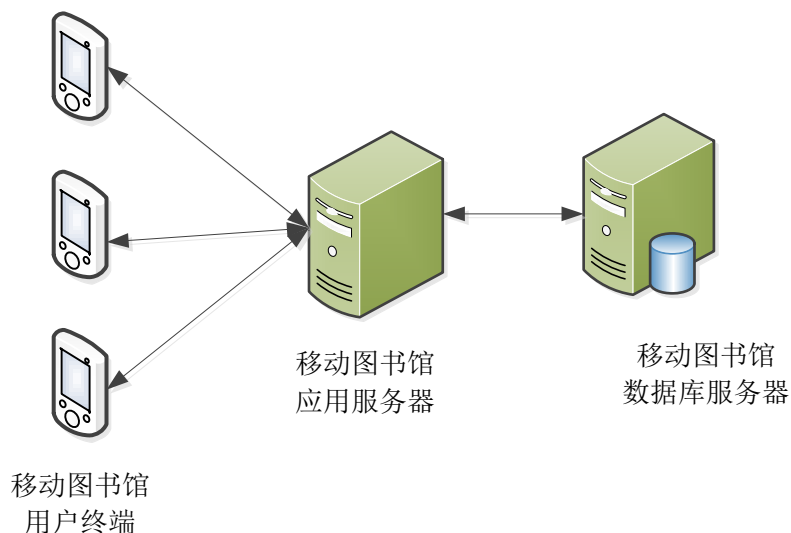


图 5-1 移动图书馆系统整体功能部署

5.2 通用功能模块的实现

5.2.1 数据库连接模块实现

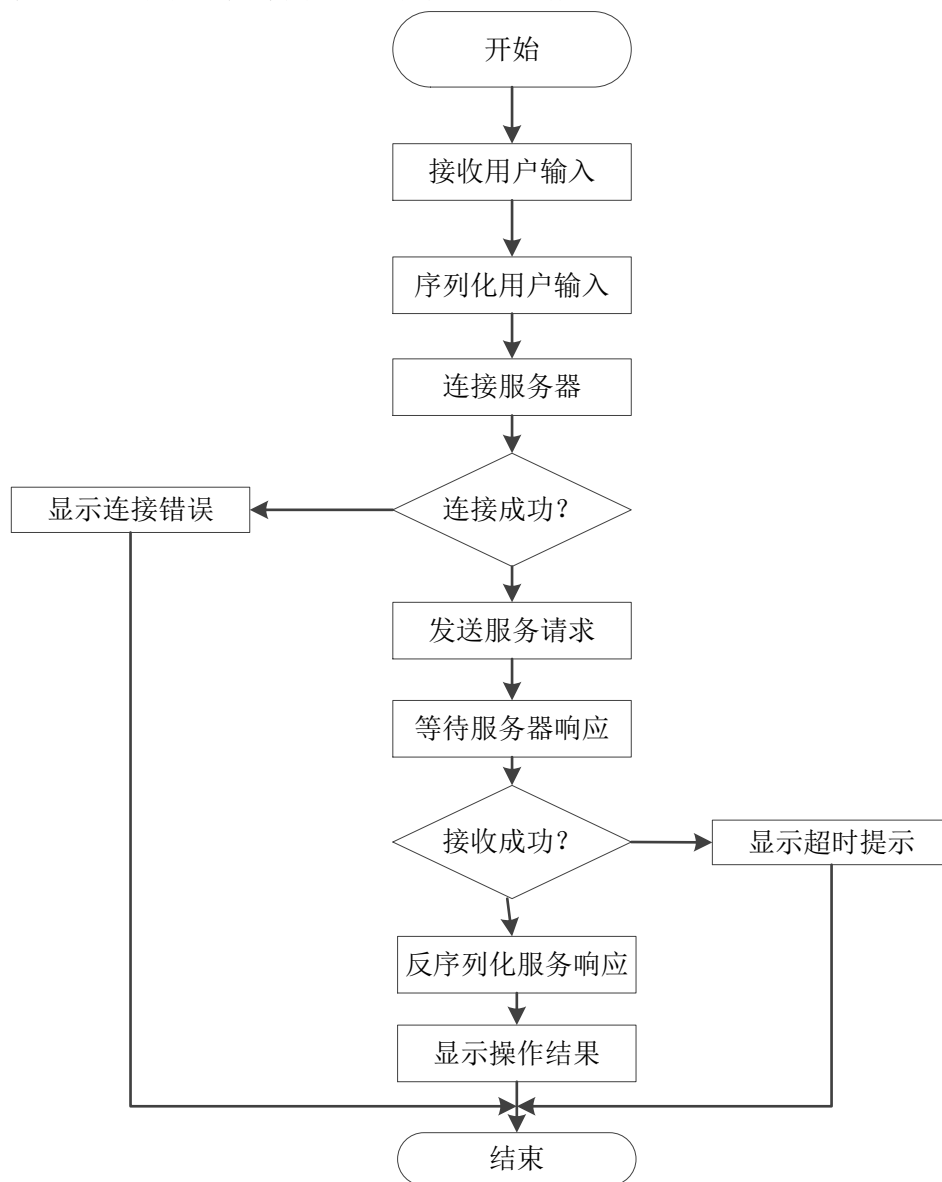
数据库连接是本系统中的通用功能，spring 中提供了 com.briup.jdbc 包作为 JAVA 调用 SQL 语句的标准 API，JDBCConnectionFactory.java 可为 JAVA 语言访问数据库提供了高级的操作方式。使用 com.briup.jdbc 需要首先定义连接地址、用户名和访问密码等三个参数。

```
private static String db_url;
private static String username;
private static String usrpasswd;
在 JDBCLinkFactory.java 类中实现数据库的创建和连接的核心代码如下所示。
//定义 Properties 属性类
Properties retproperty = new Properties();
//从输入流中加载属性列表
retproperty.load(new FileInputStream("E:\Myeclipse workspace\Library
Service\src\com\briup\jdbc\jdbcinfo"));
//用指定的键 url 获取它的键值
Connect_url=pro.getProperty("db_url");
//获得用户密码
myIdpasswd = retproperty.getProperty("usrpasswd");
//加载 Mysql 数据库驱动
new Mysql.jdbc.driver.MysqlDriver();
//读取用户名
username = retproperty.getProperty("username");
//用获得的参数连接数据库
Connection newconn=DriverManager.getConnection(db_url, username, usrpasswd);
```

5.2.2 服务访问模块实现

Spring-dao.jar 在包 com.briup.dao 封装了很多数据库操作的方法，因此可以调用 DAO 的接口函数实现各种操作，如增加记录，删除记录，查询记录等。DAO 模型是关系性数据库的基本设计理念，在设计上可将它抽象成一个对象进行操作。

用户通过使用注册的 Servlet 类来访问后台的服务，所使用到的 servlet 需要在 com.briup.servlet 包中进行定义，通过重载 doPost()和 doGet()方法来获取用户的服务请求信息。其客户端服务访问流程如图 5-2 所示。



5-2 客户端服务访问流程

服务器端接收用户服务并进行服务响应的程序流程如图 5-3 所示。

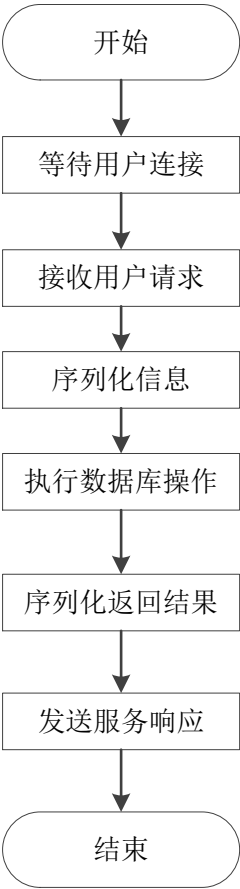


图 5-3 服务器处理流程图

所有服务功能的处理流程均采用这样的方式进行，因此在具体功能模块的实现中对于处理流程不再赘述。

5.3 具体功能模块实现

本节介绍系统主要功能模块的实现过程。

5.3.1 注册模块功能实现

5.3.1.1 后台功能实现

注册模块的后台功能由 `registerseverlet` 完成，通过 `doPost()`和 `doGet()`方法来获取新用户的注册信息，如表单中的用户名，密码，证件号，注册日期等。注册用的参数有 `username`，`passwd`，`certnubmer`，`registerdate` 实现代码如下。

```
PrintWriter output = response.getWriter();
```

```
String username = request.getParameter("username");//获得提交表单用户名
String certnubmer = request.getParameter("certnubmer");//获得提交表单标号
LibraryDao mydao =new LibraryDaoImplementations();//创建控制类函数实例
String registerdate= request.getParameter("registerdate");//获得提交表单的日期
String passwd = request.getParameter("passwd");//获得提交表单密码
//查找是否存在有同名用户
try {User findregistereduser=(User)mydao.findByUsername(username);
    if(findregistereduser!=null)
    {retxml=MakeXML.makeXml("用户名已被注册，请重新输入", false);
    }else{
    User name=new User();//创建用户对象
    name.setUsername(username);//设置用户名
    name.setUserID(id);//设置用户编号
    name.setPassword(passwd);//设置密码
    name.setDate(register 达特);//设置注册时间
    name.save(name);//保存用户信息
    retxml=MakeXML.makeXml("注册成功", true);
    }
}
```

5.3.1.2 前台功能实现

前台注册模块 Register.Act.java 实例提供用户注册的界面，核心代码如下。

```
mSpinner.setOnItemSelectedListener(new OnItemSelectedListener() { //获取按钮
    的消息事件
    public static String makeString(String input1, String input2, String input3, String
    input4, String servlet){ //
    String retstr=null; //定义 注册信息 registinfo 容器
    List<BasicNameValuePair>registinfo=new ArrayList<BasicNameValuePair>();
    list.add(new BasicNameValuePair("username", input1.trim())); //向集合类添加用户
    //名
    registinfo.add(new BasicNameValuePair("passwd", input2.trim())); //向集合类添
```


加密码

registinfo.add(new BasicNameValuePair("registerdate", input3.trim()));//向集合类
添加注册时间

registinfo.add(new BasicNameValuePair("certnubmer", input4.trim()));//向集合类
添加证件编号

```

ConnectionMethod connection1=new ConnectionMethod(); //创建新连接
HttpResponse rsp = connection1.method(registinfo, serveraddr);
String serveraddr=connurl+servlet.trim();//服务器地址
try { //如果连接创建成功，并且在 200 毫秒内响应
    if(rsp!=null&& rsp.getStatusLine().
        getStatusCode()==200)
    {
        //创建 DocumentBuilderFactory 类的对象
        DocumentBuilderFactory docbuilder_o=DocumentBuilderFactory.newInstance();
        // 取得 DocumentBuilder 类的对象;
        DocumentBuilder docbuilder_o= docbuilder_o.newDocumentBuilder();
        // 取得 doc 文件
        Document mydoc= docbuilder_o.parse(rsp.getEntity().getContent());
        //得到所有结点信息
        NodeList returnerr=document.getElementsByTagName("false");
        //获得错误的信息，并存在 returnerr 中
        NodeList retmsg= mydoc.getElementsByTagName("message");
        //获得返回的信息，并存在 returemsg 中
        if(returnerr!=null&& returnerr.getLength()>0)
        { //如果返回错误消息不为空，提取信息
            returninfor= returnerr.item(0).getFirstChild().getNodeValue();
        } //如果返回消息不为空，取得信息
        if(retmsg!=null&& retmsg.getLength()>0)
        { //获得返回的消息
            retmsg =messages.item(0).getFirstChild().getNodeValue();
        }
    }
}

```

```

    } catch (Exception e) {        //抛出所有异常
        // TODO: handle exception //打印所有异常
        e.printStackTrace();
    }
    return retmsg;}

```

系统实现的功能界面如图 5-4 所示。



图 5-4 用户注册界面

5.3.2 登录模块实现

5.3.2.1 后台功能实现

用户登录模块和用户注册的处理过程类似，通过重写 `doPost()` 方法来获取用户登录信息，由服务器上的 `loginseverlet` 完成相应的处理。

```
Print.Writer.output = response.getWriter();//获得输出流
```

```
System.output.println("欢迎登录移动图书馆");//打印出提示信息
```

```
String retstring =null; //定义字符串变量，初始化为空
```

```
String usrxname =request.getParameter("usrxname").trim();//获取用户名String
```

```

passwd=(String)request.getParameter("passwd").trim();//获取提交表单中的用户密码
LibraryDao operation=new LibraryDaoImplementations( );//创建图书管理对象实例
    if(sign.equals("normal")){//如果成功获得数据
        if(usrname!=null)
        {
            try {
com.briup.pojo.User retusr= operation.findByUsrname(usrname);
//在用户表中查询用户名
                if(retusr!=null)//用户存在
                {
if(retusr.getPasswd().equalsIgnoreCase(passwd))//密码校对
                    {
                        retstring =MakeXML.makeXml("登录成功", true);//输出登录成功信息
                    }else{
                        retstring =MakeXML.makeXml("密码错误", false);
                    }
                }else{
                    retstring =MakeXML.makeXml("用户名错误", false);
                }
            } catch (Exception e) {//捕获异常
                System.output.println("连接失败");
                output.println("网络出错</font>");
            }
        }
    }else if(sign.equals("admin")){//管理员登录
        try {//查询管理员表
            Admin adminusr=sign.findByAdminname(usrname);
            if(adminusr!=null)//管理员账户存在
            {
                if(adminusr.getPasswd().equals(passwd))//密码校对
                {
                    retstring =MakeXML.makeXml("登录成功", true);
                }else{

```

```

        retstring =MakeXML.makeXml("密码错误", false);
    }
}
else{
    retstring =MakeXML.makeXml("用户名不存在", false);
}
} catch (Exception e) { //捕获所有异常消息
    e.printStackTrace();//显示所有异常消息
}
}

```

5.3.2.2 前台功能实现

登录功能使用 LoginActivity.java 类作为登录界面 Activity。在登录界面显示当前时间，需要输入用户名，密码等信息，并选择登录的用户类型，要么是一般的用户，要么是图书管理员用户，登录后就可以进行图书查询或管理操作。步骤主要是校验用户和密码是否正确。

登录界面需要两个编辑框，两个静态显示框，以及 4 个按钮和两个单选按钮，具体代码如下：

```

if(sign.equals("admin"))//封装用户信息
{
    List<BasicNameValuePair>
    userinfo=new ArrayList<BasicNameValuePair>();//创建容器
    CommenMethod connect=new CommenMethod();//创建连接方法实例
    String denludizhi=url+"loginservlet";          //跳转的登录地址
    userinfo.add(new BasicNameValuePair("username",  username));//添加用户名
    userinfo.add(new BasicNameValuePair("passwd",  passwd));//增加密码
    userinfo.add(new BasicNameValuePair("sign",  sign));//增加标志位
    HttpResponse  httpLink=cm.method(userinfo,  connurl); //创建 response
    响应
    try {//如果创建成功，在 100 毫秒内连接成功
        if(httpLink!=null&& httpLink.getStatusLine().getStatusCode()==200)
        { //创建解析器工厂实例

```

```

        DocumentBuilderFactory
newresolver=DocumentBuilderFactory.newInstance();
        //创建加载 xml 文档(document)
        DocumentBuilder build= newresolver.newDocumentBuilder();
        Document owndoc=build.parse(httplink.getEntity().getContent());
        //然后获取文档根节点
        NodeList infomation = owndoc.getElementsByTagName("infomation");
        //获取根节点中所有子节点列表
        NodeList errinfo= owndoc.getElementsByTagName("errinfo");//获得错误
if(errinfo!=null&& errinfo.getLength()>0)//如果存在错误消息
    {
errstr= errinfo.item(0).getFirstChild().getNodeValue();//获取节点的当前值
        Toast.makeText(this, errstr, Toast.LENGTH_LONG).show();//显示错误//
提示信息
        login_username.setText("");//清空登录编辑框
        login_passwd.setText("");//清空登录编辑框
    }
if(infomation!=null&&messages.getLength()>0) //如果有消息
    {
        xiaoxistr = infomation.item(0).getFirstChild().getNodeValue();//得到消息内
容
        Toast.makeText(this, xiaoxistr, Toast.LENGTH_LONG).show();//将消息
提示给用户
        Intent newjaimian =new Intent(this, AdminActivity.class);//创建登录成功
的新界面
        startActivity(newjaimian);    //启动登录成功的新界面}}
    } catch (Exception erro) {        //捕获系统异常
        erro.printStackTrace();    //打印异常信息
    }
} else if(biaonzhi.equals("normal"))    // 如果标志位为正常 1
{
    String connurl=url+"loginservlet";

```

```

        //保存个人信息，创建 myinformation 容器
        List<BasicNameValuePair>
        myinfomation=new ArrayList<BasicNameValuePair>();
        myinfomation.add(new BasicNameValuePair("passwd", passwd)); //存入密
        码信息 myinfomation.add(new BasicNameValuePair("username", username)); //存
        入用户名信息

        myinfomation.add(new BasicNameValuePair("sign", sign)); // 存入标志位
        CommenMethod myconnection=new CommenMethod(); //获得连接方法
        HttpResponse linkrsp111= myconnection.method(myconnection ,
        denludzhi);
        try {
        if(linkrsp111!=null&& linkrsp111.getStatusLine().getStatusCode()==200)
        {
            //创建解析器工厂实例 DocumentBuilderFactory
            newresolver =DocumentBuilderFactory.newInstance();
            //使用工厂获得解析器实例
            DocumentBuilder documentbder= newresolver.newDocumentBuilder();
            Document owndoc = documentbder.parse(response.getEntity().getContent());
            //得到节点
            NodeList returnmsg= owndoc.getElementsByTagName("msg");
            NodeList errlist= owndoc.getElementsByTagName("false");
            if(errlist!=null&& errlist.getLength()>0)
            {
                //获取出错提示
                falsestring= errlist.item(0).getFirstChild().getNodeValue();
                Toast.makeText(this, falsestring, Toast.LENGTH_LONG).show();//提示错
                误
            }
            login_username.setText(""); //清空用户编辑框
            login_passwd.setText(""); //清空密码编辑框
        }
        if(returnmsg!=null&& returnmsg.getLength()>0)//如果返回消息不为空
        {

```

```

msg= returnmsg.item(0).getFirstChild().getNodeValue();//获得节点信息
Toast.makeText(this, strmessage, Toast.LENGTH_LONG).show();//显示提
//示信息

Intent adminUI=new Intent(this, ManageActivity.class);//创建跳转到另外一
//个 Activity 的 Intent 消息
adminUI.putExtra("usrname", username);
startActivity(adminUI); //启动登录成功后的管理员界面}
}
} catch (Exception e) { //捕获异常
e.printStackTrace(); //打印异常
} //下面在屏幕上显示提示信息
Toast.makeText(this, msg, Toast.LENGTH_LONG).show();}
}else{//如果输入不完整, 向用户输出提示信息
Toast.makeText(this, "信息不完整", Toast.LENGTH_LONG).show();
}
}

```

具体实现的登录界面如图 5-5 所示。



图 5-5 用户登录功能界面

5.3.3 图书查询模块实现

5.3.3.1 后台功能实现

通过在服务器端编写 SearchBookServlet, 重写 doPost()和 doGet()方法获取提交的 TextView 中的图书名字或作者名字, 来完成图书的查询功能, 主要参数为 bookname 和 author。实现的核心代码如下。

```
String bookname=request.getParameter("bookname").trim();//获得书名
String author=request.getParameter("author").trim();//获得//作者姓名
String status=request.getParameter("status").trim();//是否有书存在
//status 为 yes 表示图书处于借出状态, 为 no 表示图书未借出
System.out.println("图书查询中");
String retstring =null; //定义字符串变量, 并赋空值
List bookstring =null;//定义容器, 并初始化为空
LibraryDao chaozhuo=new LibraryDaoImplementations();//得到图书
的操作方法

if(status.equals(""))//如果状态为空
{
try { //查询数据库的书
bookstring = chaozhuo.findBookByName(bookname, author);
if(retstring!=null)
{
retstring =MakeXML.makeXml(retstring, bookstring.size()+"");
}
} else{//输出信息
retstring =MakeXML.makeXml(retstring, "此书已借出");
}
} catch (Exception e) { //捕获所有异常消息
e.printStackTrace();//打印所有异常消息
}
} else{//如果状态不为空, 查询所有的书目, 包括已借完
try {
```



```

list=mydao.findBookByName(bookname, author, status);
    if(list!=null)
    {
        retstring=MakeXML.makeXml(list, list.size()+"");
    }
    else{
        retstring =MakeXML.makeXml(list, "此书不存在");
    }
} catch (Exception e) { //捕获所有异常消息
    e.printStackTrace(); //打印所有异常消息
}

```

5.3.3.2 前台功能实现

图书查询的前台功能在 `booksearchActivity.java` 类中实现，作为图书查询界面所用的 Activity 对象。通过输入图书名等实现检索操作，核心代码如下。

```

String bookname=request.getParameter("bookname").trim();
String author=request.getParameter("author").trim();
String status=request.getParameter("status").trim();
    // yes 表示借出，no 表示可借
    System.out.println("图书可借");
    String retxml=null;
    List retlist=null;
    LibraryDao mydao=new LibraryDaoImplementations();
    if(status.equals(""))
    {
        try {
            retlist=mydao.findBookByName(bookname, authorname);
            if(retlist!=null)
            {
                retxml=MakeXML.makeXml(retlist, retlist.size()+"");
            }
        }
        else{

```

```
        xml=MakeXML.makeXml(retlist,"没有该图书");
    }
} catch (Exception e) {
    e.printStackTrace();
}
}else{
    try {
list=mydao.findBookByName(bookname, author,status);
        if(list!=null)
        {
            xml=MakeXML.makeXml(list,list.size()+"");
        }
        else{
            xml=MakeXML.makeXml(list,"没有该图书");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

图书查询功能界面如图 5-6 所示。

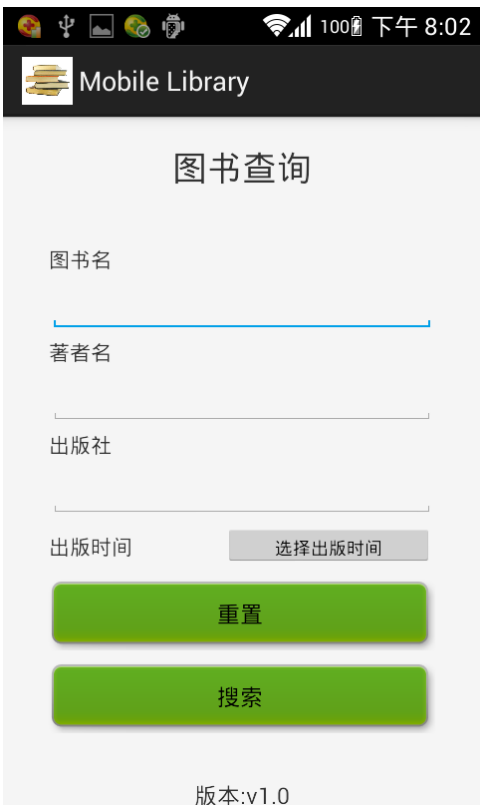


图 5-6 图书查询功能界面

5.3.4 用户管理模块实现

5.3.4.1 后台功能实现

用户管理功能 `usermanagerseverlet` 完成，通过 `doPost()`和 `doGet()`方法来实现特定用户管理功能，该 `servlet` 用来获取 `TextView` 中的用户名，通过用户名查找数据库是否有存在的用户，如果存在就返回给管理员，并可对查找到的用户执行相应的管理操作。这里获取用户的参数用 `usname` 表示，用户查找功能实现的核心代码如下。

```
PrintWriter output = response.getWriter();    //获得文本输出流打印对象
String usname =
    request.getParameter("usname").trim();//从表单中获取用户名
String userstring=null;    //定义userstring并赋空值
LibraryDao operatype=new LibraryDaoImplementations();/取得ADO操
作的方法
try {    //查找函数查询数据库中符合要求的用户
```

```

com.briup.pojo.User usr= operatype.findByUsrname(username);
if(user!=null)//如果数据库的用户名存，就返回给管理员
{
    try { //输出所有用户信息
        userstring =MakeXML.makeXml(yonghu, "");
    } catch (Exception e) { //捕获所有异常
        e.printStackTrace();//打印所有异常
    }
}
}else{ //如果输入有误
    try {
        xml=MakeXML.makeXml("密码输入不正确，请重新输入
", false);

    } catch (Exception e) { //捕获所有系统异常
        e.printStackTrace();//打印所有系统异常
    }
}
} catch (DataAccessException falsemsg) { //捕获所有异常
    falsemsg.printStackTrace(); //打印所有异常
}
}

```

5.3.4.2 前台功能实现

用户管理前台功能在 `UsrmanageActivity.java` 类中实现，作为用户管理界面所用的 `Activity` 对象。通过输入用户名等实现检索操作，核心代码如下。

```

String username=et_username.getText().toString().trim();
Toast.makeText(this, "请等待·····", Toast.LENGTH_LONG).show();

Intent intent=new Intent(this,usrmanageInfoActivity.class);
intent.putExtra("username", username);
startActivity(intent);

}else{

    Toast.makeText(this, "请输入正确的用户名",
    Toast.LENGTH_LONG).show();
}

```

}

用户管理界面的实现如图 5-7 所示。



图 5-7 用户管理功能界面

5.3.5 图书管理模块

5.3.5.1 后台功能实现

图书管理功能由后台的 `bookmanageseverlet` 完成，通过重写 `doPost()`和 `doGet()`方法获得表单中参数，它包括图书名，出版社，图书数量，出版社信息等。管理员可对查出的图书进行相应的操作，图书查询实现的关键代码如下

```
PrintWriter output = response.getWriter();           //获得文本输出流打印对象
String bookname=request.getParameter("bookname").trim(); //获取图书名
String author=request.getParameter("author").trim();//获得著作者名
String bookclass=request.getParameter("bookclass").trim();
//String status=request.getParameter("status").trim();//图书是否可阅
```

```

String stortime=request.getParameter("stortime").trim();//获得时间
String bookamount=request.getParameter("bookamount").trim();//获得数量
String xml=null;                //定义 xml 字符串，并同赋空值
int bookoaccount=Integer.parseInt(bookamount);//将得到的字符串转变为整形
    if(bookamount>0) //如果本数大于 1
    {
        LibraryDao mydao =new LibraryDaoImplementations(); //创建图书操作
函数方法
        for(int i=0;i<bookamount;i++) //
        {
            Book bookobj=new Book( );//创建图书对象
            bookobj.setBookname(bookname); //
            bookobj.setAuthor(author);
            bookobj.setBookclass(bookclass);
            bookobj.setStatus("no");
            bookobj.setStortime(stortime);
            try {
                mydao.save(bookobj); //向服务器保存信息
                System.output.println("saved success! ");//打印提示信息
            } catch (Exception e) {
                try {
                    xml=MakeXML.makeXml("处理过程出错", false);
                } catch (Exception falsemsg) { //捕获系统所有异常
                    falsemsg.printStackTrace();// 打印出错提示
                }
                e.printStackTrace();
            }
        }
    }
    try {
        xml=MakeXML.makeXml("图书信息保存成功", true);//提示用户
    } catch (Exception e) { //捕获所有异常

```

```
e.printStackTrace(); // 打印出错提示  
}
```

5.3.5.2 前台功能实现

图书管理前台模块 `managerBookActivity.java` 实现。可根据需要完成图书增加删除等管理和维护功能，图书增加的核心代码如下。

```
String bookamount=1+"";  
String bookname=et_bookname.getText().toString().trim();  
String author=et_author.getText().toString().trim();  
String publisher =et_publisher.getText().toString().trim();  
String bookclass=et_bookclass.getText().toString().trim();  
String stortime=et_stortime.getText().toString().trim();//  
bookamount=et_abbookamount.getText().toString().trim();//得到图书数量  
String bookinfo=myMyClientServers.makeBookString(bookname, author,  
publisher, bookclass, stortime, bookamount, "savebookservlet");//数据库中增加图书信息  
Toast.makeText(this, bookinfo, Toast.LENGTH_LONG).show();//显示提示信息。
```

图书管理具体实现的功能界面如图 5-8 所示。



图 5-8 图书管理功能界面

5.4 本章小结

本章对系统的实现过程进行了描述，给出了整体功能部署，通用功能模块的实现，和主要的系统功能模块的编码实现过程，并对系统实现的效果进行了展示。

第六章 系统测试与分析

6.1 测试环境描述

在系统测试中,服务器使用 ubuntu 12.10 quantal 操作系统,应用服务器使用 Tomcat7.0,数据库选用 Mysql 5,移动终端选取 HTC one S,终端操作系统为 android 2.4。

6.2 功能测试

功能测试主要对系统的用户登录、用户注册、图书查询、用户管理和图书管理等主要功能模块进行了黑盒测试。

6.2.1 用户注册功能测试

用户注册功能测试包含对正常用户注册和非正常用户注册两种情况的测试,测试过程和结果如表 6-1 所示。

6-1 用户注册功能测试用例

测试功能点	操作过程	预期结果	测试结果
正常用户注册	1、输入用户名,密码等信息; 2、选择确定。	返回注册成功结果	通过
非正常用户注册	1、输入已有用户名,或不完整的注册信息; 2、选择确定	返回注册失败提示	通过

6.2.2 用户登录功能测试

用户登录功能测试包含对正常用户登录和非正常用户登录两种情况的测试,测试过程和结果如表 6-2 所示。

6-2 用户注册功能测试用例

测试功能点	操作过程	预期结果	测试结果
正常用户登录	1、输入用户名，密码等信息； 2、选择确定。	返回登录成功结果	通过
非正常用户登录	1、输入不存在用户账户信息，或不完整的用户账户信息； 2、选择确定	返回登录失败提示	通过

6.2.3 图书查询功能测试

图书查询功能测试主要测试了正常查询和非正常查询两种情况，测试过程如表 6-3 所示。

6-3 图书查询功能测试用例

测试功能点	操作过程	预期结果	测试结果
正常图书查询	1、输入图书名，出版社等信息； 2、选择确定。	返回查询结果	通过
非正常图书查询	1、输入不存在的图书信息； 2、选择确定	返回查询失败提示	通过

6.2.4 其他功能测试

测试过程中对用户管理和图书管理等功能也进行了相应的测试，测试点覆盖了主要的功能，测试结果均达到了预期效果。

6.3 性能测试

对于系统性能主要进行了响应时间的测试，因为使用大量移动终端进行实际测试受到资源获取的限制，因此本次测试是通过在虚拟机上模拟移动终端对服务器进行请求来完成的，分别采用 10，20，50 三种不同的并发数进行了测试。测试结果如表 6-4 所示。

表 6-4 性能测试结果

并发数	连接成功率	最小响应时间 (s)	最大响应时间 (s)	平均响应时间 (s)
10	100%	0.91	2.12	1.73
20	100%	1.22	3.43	2.63
50	100%	3.41	6.75	5.41

性能测试结果达到了预期的要求，因为该测试在局域网环境下进行，考虑实际的应用场景，真实系统的性能指标应略低于本测试结果。

6.4 本章小结

本章对移动图书馆系统作了相关的功能测试，主要的功能工作正常，系统的响应时间和并发数等性能指标也达到了设计目标。

第七章 总结与展望

7.1 总结

本课题研究完成了基于 Android 平台的移动图书管理系统的设计与实现。其该系统采用了 C/S 模式可以完成用户注册、用户管理、图书查询、图书管理等基本功能。

本文的主要工作内容包括对相关技术的研讨，分析系统的具体需求，完成系统的总体设计、功能设计和数据设计，并对系统前台和后台的功能实现过程进行了详细的介绍，最后通过测试验证了设计和实现的有效性。

在整个工作过程中，本人对 android 应用开发技术有了更加深刻的认识，并积累了实际的开发经验，为以后从事类似的软件工程应用研究和开发工作奠定了良好的基础。

7.2 展望

目前移动图书管理系统的功能仅实现了基本的图书查询、用户管理和图书管理功能，下一阶段可以进一步丰富系统的功能，并可以考虑与现有的图书管理系统集成，实现图书借阅，在线阅读等功能，使之成为一个更加完善的应用系统。

致 谢

我要感谢在这几年求学中给予我极大关怀的指导老师，他是我学习和理想的导师，在学习中遇到难题时给了启迪，在生活上给我无微不至的照顾，在颓废丧气时给了信心和勉励，让我有柳岸花明，重现阳光的温暖。老师宽阔的视野，独到的见解，都让我们常常的折服。转眼间，我们求学生涯也即将划上句号，老师的谆谆教诲，还言犹在耳，他的治学态度和对人生的深刻认识，是我以后工作的榜样。在读专业硕士阶段，老师和同学对工作的认真态度和科研精益求精，不求甚解的精神，让我深受感染。我对学术、对项目研发的态度就是在这热情无限的氛围中得到潜移默化。另外，我愿意向我伸出热心的援助之手和提出宝贵建议的人表示诚挚的谢意。

虽然，在整个软件设计过程中的路途是充满曲折和挫折的，但它我以后的继续学习和开拓思维是必不可少的一步，为以后的工作奠定了扎实的基础。做毕业设计时付出的辛勤劳动和洒下的汗水会得到实实在在的回馈。完成毕业设计时，那种精神喜悦和成就感难以言表。

其次，要感谢与我一起求学的同学，它们的建议和共同形成的学习氛围是很重要的，他们身上我感受到很多自己的思维所不能及的地方，也学到了终身受益的人生道理，他们我面对挑战和困难时最有鼓励作用的战友。

我还要感谢我的爸妈，给了我物质上和精神上的支持，我才能如愿的完成学业。有他们的体贴和谅解，我才能腾出更多的时间专心学习，做出更多的成绩，每当遇到困难一蹶不振时，是他们的激励才让我扫平了奔向前方的道路。

同时感谢评阅本论文的学者和专家，你们指导和提议我会虚心接受，再创佳绩。美好的学习时光即将结束，心中无限感慨，思绪万千，这是我人生重要的里程碑，我一定要用学到的知识，回报社会，回报祖国。

参考文献

- [1] 杨丰盛.《Android 应用开发揭秘》[M].北京：机械工业出版社，2010.1
- [2] 吴亚峰，吴亚光.《Android 应用案例开发大全》[M].北京：人民邮电出版社，2012
- [3] 韩超.《Android 经典应用程序开发》[M].北京：电子工业出版社，2012
- [4] 吴亚峰，于复兴.《Android 应用开发完全自学手册 -- 核心技术、传感器、2D/3D、多媒体与典型案例》[M].北京：人民邮电出版社，2012
- [5] 罗杰（美国），李耀亮.《Android 应用开发》[M].北京：人民邮电出版社，2010
- [6] 王家林.《大话企业级 Android 应用开发实践》[M].北京：电子工业出版社，2011.8
- [7] 李刚.《疯狂 Android 讲义》[M].北京：电子工业出版社，2011.7
- [8] 余志龙.《Android SDK 开发范例大全》[M].北京：人民邮电出版社，2010.6
- [9] 郭宏志.《Android 应用开发详解》[M].北京：电子工业出版社，2010.10
- [10] 吴亚峰.《Android 核心技术与实例详解》[M].北京：电子工业出版社.2010.10
- [11] 李芝兴.《Java 程序设计之网络编程》[M].北京：清华大学出版社，2009.8
- [12] Raj Kamal.《EMBEDDED SYSTEMS ARCHITECTURE PROGRAMMING AND DESIGN》[M].北京:清华大学出版社，2005
- [13] 殷兆麟主编.《Java 语言程序设计》[M]. 北京:高等教育出版社，2002.12
- [14] (美)Ivor Horton 著.《Java2 编程指南(JDK 1.3 版)》[M]. 北京：电子工业出版社，2001.11
- [15] 张建沛.《数据库原理及应用系统开发》[M].北京：中国水利水电出版社，1999.4
- [16] 金尊和等编著.《Java 高级编程》[M].北京:电子工业出版社，1997.8
- [17] 肖刚编著.《Java Web 服务器应用程序设计》[M].北京:电子工业出版，2001.9
- [18] 王迪化等编著.《JSP/Servlet—基于 Java 最新网站建设工具》[M].北京:清华大学出版社，2001.6
- [19] 林海编著.《浏览器/服务器应用开发》[M]. 北京：科学出版社. 2000.4
- [20] 庞慧娟等.《利用 JSP 实现基 JDBC-ODBC 桥的 Web 数据库的访问》[J].计算机应用，2003 年. 07 期：112-114
- [21] 严潮红.《基于仿真的 CAD/CAE 几何数据传递研究》[M]. 南京:南京航空航天大学硕士学位论文，2004
- [22] Android 官方网站 <http://developer.android.com/reference/java/net/Socket.html>

- [23]Tomas Katysovas, A first look at Google Android, Free University of Bolzan[J],Internet Technologies 2, 2007-2008, 12-26
- [24]Nicolas Gramlich, Android Programming [M], <http://andbook.anddev.org>, 2008.02.21
- [25]Ravi S. Sandhu, Edward J. Coynek, Charles E. Youmank, Role-Based Access Control Models[J], IEEE Computer Volumes 29 Number 2 February 1996, 38-47
- [26]Jerome (J.F.) dImARZIO,Android A Programmer's Guide[M], McGraw-Hill/Osborne Media, 2008.07.30
- [27]Reto Meier, Professional Android Applicationp [M], Wiley John & Sons, Incorporated 2008.12
- [28]Mark Pilgrim, Anatomy & Physiology of an Android [M]
- [29]Matt Henricksen, PWilliam Caelli, Securing grid data using mandatory access controls[J],Fifth Australasian Symposium on Grid Computing and e-Research(AusGrid 2007),25-32
- [30]Maoto.X, Linux Intrusion Detection System [M]. World-wide web page available at <http://www.lids.org>
- [31]T.Linden. Operating System Structures to Support Security and Reliable Software[J].ACM Computing Surveys, 8(4),December 1976.
- [32]Peter Loscocco, Stephen Smalley. Integrating Flexible Support for Security Policies into the Linux Operating System [J]. In Proceedings of the FREENIX, Jane 2001.
- [33]Improving heuristic mini-max search by supervised learning [J] Michael Buro Jan. 2002 Artificial Intelligence page 85-90
- [34]Parallel randomized best-first minimax search [J], Yaron Shoham, May. 2002 Artificial Intelligence page 165-196
- [35]Statistical feature combination for the evaluation of game positions [J], Michael Buro Journal of Artificial Intelligence Research Jun. 1995 page 272-282