

# 本科毕业设计(论文)

题 目 掌上个性化图书馆的设计与实现

系 别 计算机信息技术系

专业名称 软件工程

班级名称 112302

学生姓名 车蓂羽

学 号 131144

指导教师 陈世亮

毕业时间 2017年6月 日

毕业 任务书

设计

论文

一、题目

掌上个性化图书馆的设计与实现

二、指导思想和目的要求

掌上个性化图书馆的设计与实现课题来源于生产实习，主要实现根据用户初始化设置和用户行为路径分析的个性化图书推荐。主要研究方法为建立用户日志数据，找到用户触发事件的关键词，从而查询出用户感兴趣的内容，实现个性化图书推荐。

三、主要技术指标(要修改)

在具有搜索和导航功能的页面上添加用户点击事件的布点，并且创建相关事件数据库，记录用户行为路径的日志数据，主要数据字段包括：用户ID，用户访问步骤seq，用户访问时间visit-time，用户点击事件event。

四、进度和要求

|  |  |
| --- | --- |
| 第一周 | 确定具体功能需求，完成开题报告 |
| 第二周 | 完成UI高保真图绘制 |
| 第三至四周 | 在Android Studio上完成界面布局 |
| 第五至六周 | 完成后台逻辑跳转代码 |
| 第七周 | 创建数据库 |
| 第八至九周 | 完成数据库与布局适配 |
| 第十周 | 中期检查汇报 |
| 第十一周 | 完成个性化查询 |
| 第十二周 | 调试单元测试，模块耦合测试，整体功能测试； |
| 第十三周 | 总结验收时、撰写论文； |
| 第十四周 | 演示文稿制作、论文定稿、答辩准备； |
| 第十五周 | 毕业答辩 |

五、主要参考书及参考资料

[1] Pelgrims K. Gradle for Android[M]. Packt Publishing Ltd, 2015.

[2] Zapata B C. Android studio application development[M]. Packt Publishing Ltd, 2013.

[3] Craig C, Gerber A. Learn Android Studio: Build Android Apps Quickly and Effectively[M]. Apress, 2015.

[4] Developers A. API Guides[J]. 2014.

[5] Eckel B. Thinking in JAVA[M]. Prentice Hall Professional, 2003.

[6] Bloch J. Effective java[M]. Pearson Education India, 2008.

[7] <https://developer.android.com>

[8] https://github.com

[9] https://www.oschina.net

[10] http://www.csdn.net

[11] https://yq.aliyun.com

[12] https://segmentfault.com

[13] 王晓禹,石丽.基于JSON实现Android智能终端与Web服务器 “面向对象” 的信息交换[J].数字技术与应用,2012(4):224-225.

学生 \_\_\_\_\_\_\_\_\_\_ 指导教师 \_\_\_\_\_\_\_\_\_\_ 系主任 \_\_\_\_\_\_\_\_\_\_\_

**(须本人手写签名）**

## 摘要

**摘 要：**软件开发逐步偏向于移动端，Android应用更是蓬勃发展，基于谷歌公司制作的集成式开发工具Android Studio，能开发出兼容多个Android版本的应用。与Eclipse开发工具相比，Android Studio能开发的范围更广，集成度更高，有逐步取代Eclipse的趋势。本课题所设计的掌上个性化图书馆就是运用AndroidStudio工具和Java语言还有Android设备自带的SQLite数据库实现的。其中本地数据库部分同时运用了传统的SQlite数据库创建方式和现在流行的litepal数据库框架方式；整个App的布局采用符合谷歌Android 5.0设计规范的Material Design ；个性化图书推荐功能采用多维度的图书推荐，从几个简单维度进行分析抽取，实现基本个性化推荐，同时，数据库表的创建方式满足低耦合性，方便后期对表进行升级。

**关键词：** Android Studio Java litepal

**Abstract:** Software development is gradually biased towards mobile, Android applications are booming, based on Google's integrated development tool Android Studio, can develop a number of applications compatible with Android version. Compared with Eclipse development tools, Android Studio can develop a wider, more integrated, and gradually replace the Eclipse trend. This topic is designed by the Pocket Personalization Library is the use of AndroidStudio tools and Java language Android device comes with SQLite database implementation. Which the local database part of the use of the traditional SQlite database creation and the current litepal database framework; the entire App layout with Google Android 5.0 design specifications of the Material Design; personalized book recommendation features multi-dimensional book recommendation, from Several simple dimensions of analysis and extraction, to achieve the basic personalized recommendation, at the same time, the database table to create a low coupling to facilitate the latter part of the table to upgrade.

**Keywords:** Android Studio Java litepal

**目 录**

[本科毕业设计(论文) 2](#_Toc24158)

[毕业 任务书 3](#_Toc18526)

[摘要 5](#_Toc785)

[第一章 绪 论 6](#_Toc29905)

[1.1 项目背景 6](#_Toc22070)

[1.2 课题主要内容 6](#_Toc31115)

[1.3 论文组织结构 6](#_Toc14000)

[第二章 Android平台与技术 7](#_Toc20673)

[2.1 Android平台介绍 7](#_Toc31036)

[2.2 Android技术介绍 7](#_Toc6709)

[2.3 Android开发环境介绍 10](#_Toc11454)

[2.4 本章小结 10](#_Toc1720)

[第三章 个性化图书推荐的概要设计 10](#_Toc11835)

[3.1 需求分析 10](#_Toc6348)

[3.2 功能模块 11](#_Toc29047)

[3.3 界面设计 11](#_Toc11083)

[3.4 数据库设计 11](#_Toc19330)

[3.5本章小结 13](#_Toc30261)

[第四章 个性化图书馆的详细设计 13](#_Toc14017)

[4.1登录功能详细设计 13](#_Toc19230)

[4.2我的订阅和我的收藏功能详细设计 14](#_Toc1080)

[4.3我的爱好功能详细设计 15](#_Toc19915)

[4.4关键字搜索详细设计 16](#_Toc19483)

[4.5个性化图书推荐详细设计 17](#_Toc21252)

[4.6本章小结 18](#_Toc16368)

[第五章 个性化图书馆的功能实现 18](#_Toc7844)

[5.1实现主要功能的Activity 18](#_Toc21200)

[5.2用户基本数据查询的关键代码 20](#_Toc17255)

[5.3实现个性化推荐功能的关键代码 21](#_Toc4221)

[5.4实现效果截图 21](#_Toc14869)

[5.5本章小结 21](#_Toc624)

[结论 21](#_Toc13035)

[参考文献 21](#_Toc14856)

[致谢 21](#_Toc31608)

[毕业设计小结 21](#_Toc4480)

## 第一章 绪 论

### 1.1 项目背景

现在智能手机的使用相当普遍，其中占市场份额最大的又属Android操作系统。由于Android操作系统的开源性使得基于Android平台的应用软件迅猛发展，使用人数也是越来越多。

目前，专门服务于学校的应用软件的种类还不多，大致一般分为两种类型：校园通一类；图书馆App一类。前者出现次数最多并且使用广泛，后者因为服务的局限性和可拓展性目前没有被广泛关注并且深入的研究开发。

市场上关于图书馆的应用软件主要有以下几种模式：（1）B/S模式：浏览器/服务器对等传输形式，这种模式运用广泛，几乎每个学校的图书馆官网都是运用的这种形式；（2）C/S模式：客户端/服务器对等传输的形式，目前这种模式运用较少，这类应用的主要代表是超星图书馆App，这类App主要涵盖的功能为学校官网图书馆B/S模式的功能，即图书馆管理系统的借书还书，检索馆藏功能。

### 1.2 课题主要内容

基于Android客户端应用使用的广泛性和服务于学校图书馆的客户端应用软件的功能单一性，本课题设计的主要方向为改进C/S模式下的图书馆应用软件的个性化推荐功能。

本文将从三个维度实现个性化图书推荐功能：用户收藏图书的类别、用户设定的兴趣类别、用户检索历史关键词。用户可以方便快捷地登录该掌上图书馆，查询馆藏和个人信息，快速获取自己感兴趣的图书种类。这有利于最大化图书馆的资源价值，提高学生的阅读量。

### 1.3 论文组织结构

本论文共分为五章，每章的组织结构安排和内容如下：

1. 是引言。本章主要介绍选题的背景和内容。
2. 是实现环境介绍。本章主要介绍开发平台的优势和组成，从Android的系统和Activity生存周期来介绍主要用到的技术。
3. 是掌上图书馆个性化推荐的概要设计。本章从项目的需求开始分解，得出几个主要功能模块，并设计了整个数据结构表。
4. 是对主要功能的详细设计。本章主要对每个模块功能做详细的流程图分析，预先设计代码的编写逻辑。
5. 是实现主要功能的过程。本章分步介绍了实现主要功能所需要的类和方法，并展示出部分功能的关键代码和最终实现截图。

## 第二章 Android平台与技术

### 2.1 Android平台介绍

##### 2.1.1 Android平台的优势

Android 是一个基于Linux内核的开放源代码移动操作系统，由Google成立的Open Handset Alliance 持续领导与开发，只要设计用于触屏移动设备如智能手机和平板电脑与其他便携式设备。[来源于维基百科]

Android操作系统能成为现今最大的操作系统，少不了它强大的优势支撑。Android具有强大的开放性，允许任何终端厂商加入到Android联盟中来，使其拥有更多的开发者，从而促进其拥有丰富的软件资源，消费者也可以用更低的价位获得心仪的手机。正是由于Android系统的开放性，它提供给第三方开发者一个十分宽泛、自由的环境，不会受到各种限制。同时，最大的优势之一还有无缝结合Google优秀的服务。

##### 2.1.2 Android平台的组成

Android平台大致包含四个组成部分：（1）Linux内核层：该层包含各种驱动，并提供操作系统的基本功能。（2）中间层：该层包括Libraries和Android运行时的环境。（3）应用程序框架：该层是编写核心应用所使用的API框架，开发者可以使用这些框架来开发自己的应用程序，但必须遵守该框架的开发原则。（4）应用层：该层是各种应用软件，包括通话、短信、日历、地图、浏览器等核心应用程序，这些应用程序都是使用Java写的。

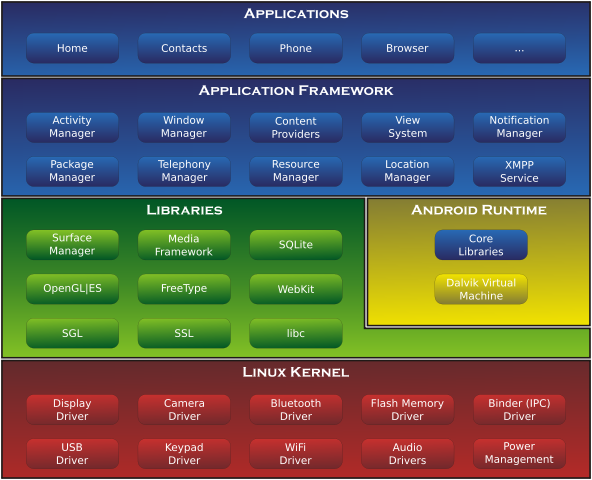
### 2.2 Android技术介绍

##### 2.2.1 Android系统介绍

Android操作系统的核心属于Linux内核的一个分支，具有典型的Linux调度和功能，除此之外，Google为了能让Linux在移动设备上有良好的运行，对其进行了修改和扩充。Android把Linux中的本地 X Window System去除了，并且不支持标准的GUN库。2008年，Patrick Brady提出Android HAL架构图。该架构隔开了Android framework和Linux kernel，使Android在移动设备的运行效率更高。

目前Android操作系统具备的功能有：显示布局、数据存储、网络、信息、语言、浏览器、支持Java、媒体支持、流媒体支持、硬件支持、多点触控、蓝牙、多任务处理、语音功能、无限共享功能、截图功能。

Android的系统架构图示



##### 2.2.2 Android 页面Activity生存周期

Activity是一个应用组件，用户可与其提供的屏幕进行交互，以执行拨打电话、拍摄照片、发送电子邮件或查看地图等操作。[来源于developer.android.com/]当一个新的Activity被执行时，它被放置于Activity stack 的最顶端，并且变成running Activity。一个Activity的生命周期有四个状态：Active（活动），Paused（暂停），Stopped（停止），Dead/Inactive（已回收或未启动）。以下是Activity的生命周期图，矩形表示回调方法。[来源于developer.android.com]

Activity生命周期图



##### 2.2.3 Android SQLite数据库框架介绍

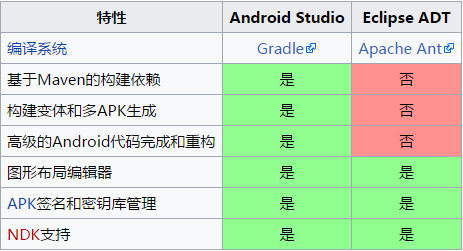
SQLite是遵守ACID的关系数据库管理系统，它被继承在用户程序中，不是一个客户端/服务器结构的数据库引擎。SQlite库链接到程序中，并是它的一个组成部分。要用传统的SQLite方法创建数据库，需要继承SQLiOpenHelper帮助类，并且重写onCreate()和onUpgrade()方法，然后分别在这两个方法中去实现创建、升级数据库的逻辑，最后获取SQLiteDatabase实例，数据库就创建成功了。

项目中的数据库往往不止一个表，当有很多表组成一个数据库时，传统的SQLite创建表的方式就显得繁琐了。所以采用目前在GitHub上开源的一款Android数据库框架LitePal。它采用对象关系映射（ORM）模式，封装平时开发最常用的一些数据库功能，使开发者不用编写一行SQL语句就可以完成各种建表、增删改查操作。

### 2.3 Android开发环境介绍

本项目采用的开发环境为Android Studio 2.2.2，它是一个基于JetBrains IntelliJ IDEA专门为Android平台开发程序特殊定制的集成开发环境。Android Studio有很多突出的特点：可视化布局、开发者控制台、Beta版本测试、基于Gradle的构建支持、Android特定代码重构和快速修复、Lint提示工具、支持ProGuard和应用签名功能、基于模板的向导来生成常用的Android应用设计和组件、自带布局编辑器、支持构建Android Wear 应用、内置Google Cloud Platform 支持。Android Studio和传统的 Eclipse ADT相比，具备多个优点。

Android Studio和Eclipse ADT比较图



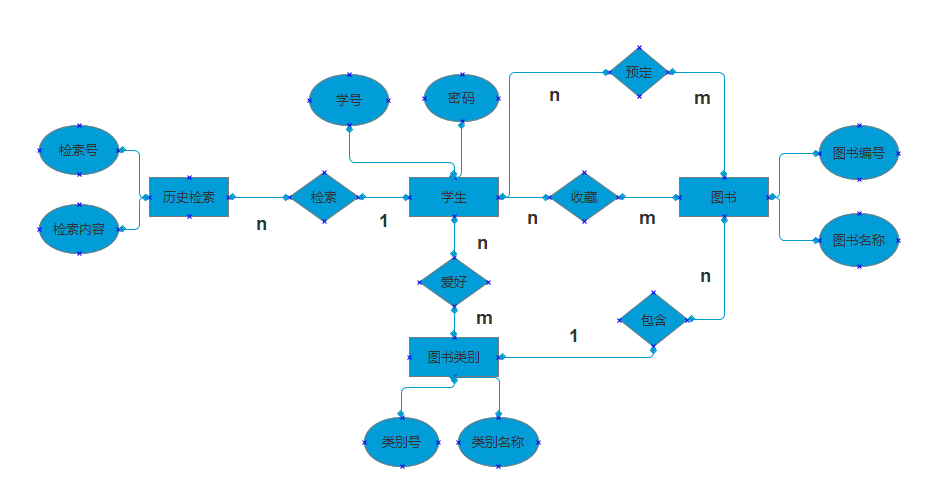
### 2.4 本章小结

通过介绍Android平台、Android系统技术和开发Android所需环境，能初步对Android开发有个简单的了解。同时和传统Android开发工具Eclipse ADT做了对比，能够发现，开发Android应用的环境越来越集成化和宽泛化，这是时代的趋势，也是必然结果。

## 第三章 个性化图书推荐的概要设计

### 3.1 需求分析

本课题主要实现掌上图书馆个性化推荐，所以功能需求围绕个性化推荐展开。为了把问题的数据结构清晰、准确的描述出来，以下建立实体-联系图（ER图）来表示数据模型。



### 3.2 功能模块

该掌上个性化推荐图书馆根据需求分析，主要实现五个功能模块，分别是用户登录、我的收藏/订阅模块、我的爱好模块、馆藏查询模块、个性化推荐模块。

1）用户登录模块：提供数据库中存在的用户直接登录应用的功能。由于应用面向的对象是学校的学生，所以不提供注册接口。

2）我的收藏/订阅模块：展示用户收藏/订阅的图书名称和图书封面。由于目前主要设计个性化图书推荐功能，所以增加了手动新增图书信息的功能。新增图书按照图书号、图书名称和图书所属类别号三个必填属性增加。

3）我的爱好模块：提供所有图书类别的标签集，用户可自定义选择喜欢的图书类别标签。已往的传统B/S设计的图书馆系统，没有学生自主选择爱好类型的功能，学生要查询自己感兴趣的图书信息十分不便。更不能实现简单的个性化图书推荐。

4）馆藏查询模块：主要提供对馆藏图书的查询，可以从多个维度进行查询。目前实现模糊查询，并且记录保存最近五条的检索历史。

5）个性化推荐模块：展示系统对用户推荐的图书，推荐的来源为上述三个功能模块内容的整合。根据我的收藏/订阅图书的图书类别，我的爱好设置的图书类别，检索历史关键字，查询数据库中所有满足查询条件的图书，并展示在该模块中。

### 3.3 界面设计

随着Android应用设计的不断规范化和扁平化，本课题应用的界面设计风格依照Android 5.0 及以上的新特性控件库设计，使用Material Design 设计规范。

### 3.4 数据库设计

通过对客户端功能和界面的设计，要想在本地实现数据之间的信息传递，需要创建数据库。本项目使用Android设备中自带的SQLite数据库，通过在项目中创建一个model文件夹，存放建立的数据表，并在项目逻辑代码中直接编写启动代码来调用数据库。下面列出了项目运行时需要涉及的数据信息表，共7张表，分别是学生信息表、图书类别表、图书预定表、图书收藏表、检索历史表、图书推荐表、所有图书信息表。

1. 学生信息表，用于存放学校在籍学生的信息，如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 说明 |
| user\_id | int | Primary key | 学号，唯一标识 |
| password | int | Not null | 登录密码 |
| type\_id | int | Foreign key | 爱好图书类别 |
| collect\_id | int |  | 收藏图书号 |
| book\_id | int |  | 预定图书号 |
| search\_id | int |  | 检索历史号 |

1. 图书类别表，用于存放所有图书的类别信息，如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 说明 |
| Type\_id | int | Primary key | 图书类别号 |
| Type\_name | String | Not null | 图书名称 |

1. 图书预定表，用于存放学生预定的图书信息，如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 说明 |
| Libs\_id | int | Primary key | 预定书号，联合主键 |
| Libs\_name | String | Not null | 预定书名 |
| Type\_id | int | Foreign key | 预定图书类型 |
| User\_id | String | Primary key | 学号，联合主键 |

1. 图书收藏表，用于存放学生收藏的图书信息，如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 说明 |
| Collect\_id | int | Primary key | 收藏书号，联合主键 |
| Lib\_name | String | Not null | 收藏书名 |
| Type\_id | int | Foreign key | 收藏图书类型 |
| User\_id | int | Primary key | 学号，联合主键 |

1. 检索历史表，用于存放学生检索的前五条检索信息，如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 说明 |
| Search\_id | int | Primary key | 检索号，联合主键 |
| User\_id | int | Primary key | 学号，联合主键 |
| Search\_name | String | Not null | 检索内容 |

1. 图书推荐表，用于存放指定学生的推荐图书信息，如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 说明 |
| Offer\_id | int | Primary key | 推荐号，联合主键 |
| Book\_name | String | Not null | 图书名称 |
| User\_id | int | Primary key | 学号，联合主键 |
| Type\_id | int | Foreign key | 图书类别号 |
| Book\_id | int | Foreign key | 图书号 |

1. 所有图书信息表，用于存放所有图书信息，如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 约束 | 说明 |
| Book\_id | int | Primary key | 图书号 |
| Book\_name | String | Not null | 图书名称 |
| Type\_id | int | Foreign key | 图书类别号 |

### 3.5本章小结

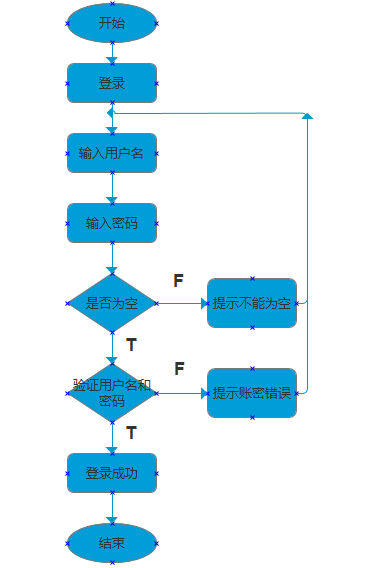
本章通过对本项目的需求、功能、数据库进行设计分析，简要了解了登录功能、我的爱好功能、我的收藏/订阅功能和为我推荐功能的整体脉络。

## 第四章 个性化图书馆的详细设计

### 4.1登录功能详细设计

根据本项目的需求分析，用户打开软件后，用户将进入登录页面，只要输入数据库中已经存在的学号和匹配密码，就可以成功登录。用户登录模块设计如下：

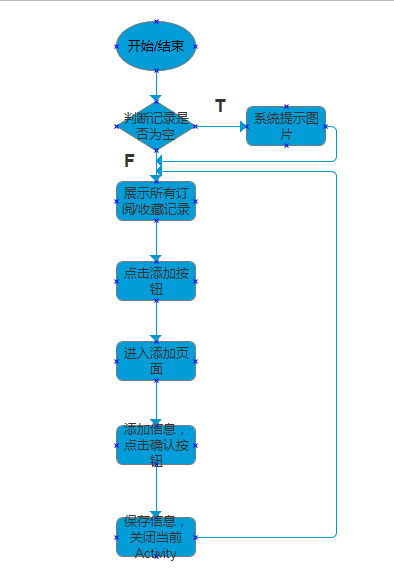
1. 用户名或者密码为空：系统提示用户名或密码不能为空。
2. 密码错误：当用户输入错误的登录密码时，系统提示密码或用户名错误。
3. 用户不存在错误：当用户输入数据库中不存的用户名时，系统提示不存在该用户。



### 4.2我的订阅和我的收藏功能详细设计

用户点击个人主页可以进入我的订阅/我的收藏查看或添加订阅/收藏，订阅/收藏模块设计：

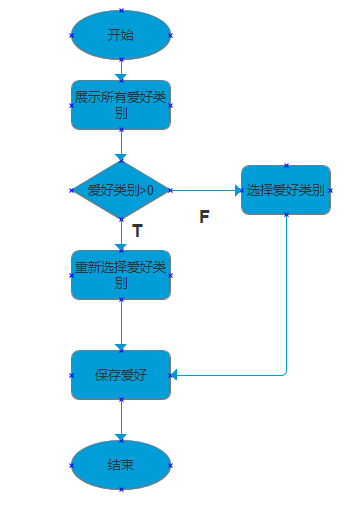
1. 添加订阅/收藏：用户点击底部的添加按钮可以在添加页面输入图书号、图书名称、图书类别，确定后把添加的图书展示到我的订阅/收藏页面。
2. 订阅/收藏记录判空：当用户第一次点开订阅/收藏页面时，系统要展示订阅/收藏历史前，需要判断收藏记录是否为空，若没有记录，则出现图片提示没有任何收藏。



### 4.3我的爱好功能详细设计

用户在个人中心界面点击进入我的爱好界面后，可进行图书爱好类型设置，爱好设置模块设计：

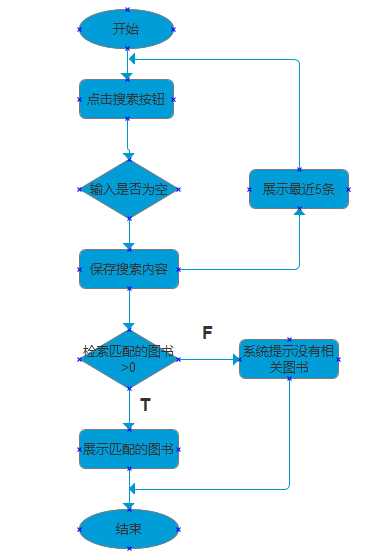
1. 查询展示所有图书类别：系统从数据库中查询所有图书类别，并用流式布局显示在当前页面。
2. 查询展示用户爱好类型：系统查询用户设置的爱好类型，并用被点击模式展示在当前页面。
3. 修改爱好设置：用户在当前页面，点击爱好标签后，点击保存按钮，系统保存用户当前选择的爱好。



### 4.4关键字搜索详细设计

用户在搜索框中输入搜索关键词后，点击搜索按钮，可模糊查询包含关键词的图书名称，关键词检索模块设计：

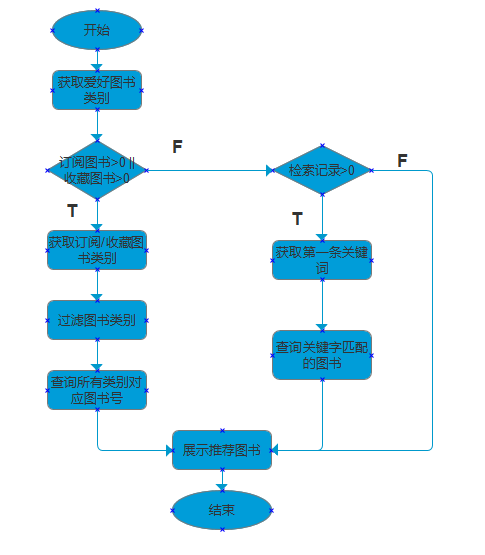
1. 匹配内容为空：当用户输入的关键词在数据库中匹配条数为零时，系统提示没有相关内容。
2. 展示所有匹配内容：系统把用户输入的关键词在表中进行模糊匹配，展示所有匹配成功的图书名称。
3. 存储检索内容：系统把用户每次输入的检索内容存储起来，并且在检索历史框中展示最近5条的检索历史。



### 4.5个性化图书推荐详细设计

当用户设置了图书爱好类别后，系统可根据用户的订阅/收藏图书类别、检索关键词对应的图书类型来给用户推荐图书，个性化推荐模块设计：

1. 获取爱好图书类型：从用户设置的爱好类型中获取爱好图书类型存入推荐表的爱好数组中。
2. 过滤订阅/收藏图书类型：以数组的形式分别存储订阅图书的类别和收藏图书类别，再以爱好类别数组为基数，依次把订阅类别数组和收藏类别数组的每个元素同爱好类别数组比较是否相同，相同继续比较下一个，不同元素存储在爱好数组的+1位置，最后得到新图书类别数组。
3. 根据检索记录获取图书设置：当检索记录不为空时，只获取最近检索的关键字，模糊查询获取图书名字和图书号。
4. 展示图书优先级：优先展示检索关键词查询到的图书，再展示推荐表中图书类别数组对应的全部图书。



### 4.6本章小结

本章节主要以分析业务流程的方式来对各功能进行详细设计，每个功能的流程图基本表示为具体功能代码的逻辑。首先设计好流程图后，对接下来的编写代码的整体思路有很好的引导作用。

## 第五章 个性化图书馆的功能实现

### 5.1实现主要功能的Activity

Android的界面是由布局和组件协同完成的。要有好的人机交互体验就要有良好的界面布局。Android目前有五种布局方式，分别是LinearLayout（线性布局）、RelativeLayout（相对布局）、FrameLayout（帧布局）、TableLayout（表格布局）、AbsoluteLayout（绝对布局），其中最常用的为前四种布局方式。

本论文设计的掌上图书馆个性化推荐应用主要采用LinearLayout布局、RelativeLayout布局、FrameLayout布局和TableLayout布局。

##### 5.1.1用户登录模块

登录功能主要在SignInActivity类中实现，该类继承自Activity方法，因为是基于本地数据库实现的应用，所以在Oncreat()方法中创建数据库，当用户点击登录按钮时，设置按钮的监听setOnclickListener()方法来验证输入用户名和密码，这里自定义一个checkIdAndPwd()方法来进行验证。

为了以后的优化和交互体验，本界面增加了忘记密码链接，使用getClickable()方法来继承SpannableString 类实现文本变链接的效果。

##### 5.1.2订阅/收藏模块

订阅功能在MyBookActivity类中实现，继承自AppCompatActivity类和OnclickListener接口。界面布局采用RelativeLayout的布局方式，使用Android SDK提供的android.support.v7.widget.ToolBar作为导航栏，使用android.support.v7.widget.RecyclerView作为容器显示list数据，并且自定义一个list\_view\_book\_item.xml来布局每个Item。为了添加数据方便，不扩大化项目工程，采用了手动添加预定图书的方式。

在Oncreat()方法中，为了获取全部预定的图书，自定义了一个getAllBook()方法，在这个方法中判断预订的图书是否为空，不为空时，自定义了populatedDataFromDB()方法来加载数据库中的预定信息，为空时，自定义了一个继承Toast类的showToast()方法来用图片提示用户没有订阅数据。并且为了提高交互体验，自定义了一个onActivityresult()方法来专门提示用户的操作情况，并且刷新数据。

在RecycleView中加载数据时，需要开发者定义一个适配器类来适配要输出的数据形式，本项目定义了一个MyAdapter的适配器来适配数据。由于项目事先运用了LitePal数据库框架，所以只需要重写Adapter中的几个方法，并且由于适配的布局是比较流行的RecycleView布局，所以需要继承RecycleView.Adapter类，需要重写MyAdapter()方法、onCreatViewHolder()方法、onBindViewHolder()方法和getItemCount()方法。

##### 5.1.3爱好类型设置模块

爱好设置在MyLikeActivity类中实现，要实现流式布局标签需要自定义一个继承ViewGroup的FlowTagLayout类，并且在类中重写onMeasure()方法和onLayout()方法。还需要新建一个适配器来填充数据和更新数据，这个适配器类定义为TagAdapter，写法和前面的RecycleView的适配器一样。定义好标签布局item\_tag\_view.xml和点击标签状态变化selector，就能够通过设置标签的点击监听事件setOnTagClickListener()和选择监听事件setOnTagSelectorListener()来实现点击和选择效果，并给事件添加回调。

##### 5.1.4个性化推荐模块

### 5.2用户基本数据查询的关键代码

以加载预定图书信息为例，加载时判空用showToat()方法，加载时用populateDataFromDB()方法，并且传递给适配器MyAdapter的对象Adapter。

*/\*\*  
 \* 获取全部预定图书方法  
 \*/* private void getAllBook() {  
 List<LibsInfo> all = DataSupport.*findAll*(LibsInfo.class);  
 if (all != null && all.size() > 0){  
 LibsInFoList=all;  
 populateDataFromDB();  
 Adapter.notifyDataSetChanged();  
 } else {  
 showToast();  
 }  
}  
 private void showToast(){  
 Toast toast =new Toast(this);  
 ImageView img=new ImageView(this);  
 img.setImageResource(R.drawable.*img\_book\_hint*);  
 toast.setView(img);  
 toast.show();  
 }  
 private void populateDataFromDB(){  
 mList.clear();  
 List<String> columnList= new ArrayList<String>();  
 columnList.add("name");  
 mList.add(columnList);  
 if (LibsInFoList!=null&&LibsInFoList.size()>0){  
 for (int i=0;i<LibsInFoList.size();i++){  
 LibsInfo user=LibsInFoList.get(i);  
 List<String> stringList=new ArrayList<String>();  
 stringList.add(String.*valueOf*(user.getId()));  
 stringList.add(user.getLib\_name());  
 mList.add(stringList);  
 }  
 Adapter = new MyAdapter(MyBookActivity.this, LibsInFoList);  
 mRecyclerView.setAdapter(Adapter);  
 }  
 }

### 5.3实现个性化推荐功能的关键代码

### 5.4实现效果截图

### 5.5本章小结

## 结论

## 参考文献

## 致谢

## 毕业设计小结