# 摘 要

**摘要**：“我的书架”的主要功能是为用户来对纸质书籍进行管理的一款安卓应用。

“我的书架”使用android studio开发，并使用了安卓内置的SQLite数据库，SQLite是一款轻型的数据库，是遵守ACID的关联式数据库管理系统，它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。比起 Mysql、PostgreSQL这两款开源世界著名的数据库管理系统来讲，它的处理速度比他们都快。由于SQLite的核心引擎本身不依赖第三方的软件，使用它也不需要安装。并且SQLite中所有的信息都包含在一个文件内。这个文件可以自由复制到其它目录或其它设备上，方便同步。

**关键词：安卓应用； android studio； 书籍管理；**

目 录

[摘 要 I](#_Toc25855)

[第1章 绪 论 1](#_Toc3399)

[1.1 本论文的背景和意义 1](#_Toc21683)

[1.2 本论文的主要方法和研究进展 1](#_Toc7829)

[1.3 本论文的主要内容 1](#_Toc32174)

[1.4 本论文的结构安排 1](#_Toc4757)

[第2章 结构与界面设计 2](#_Toc17404)

[2.1 数据结构 2](#_Toc2920)

[2.1.1 user数据表 2](#_Toc24175)

[2.1.2 site数据表 2](#_Toc29925)

[2.1.3 book数据表 2](#_Toc8336)

[2.2 软件结构 3](#_Toc14168)

[2.3 界面设计 3](#_Toc12500)

[2.3.1 应用图标 4](#_Toc10578)

[2.3.2 登录/注册界面 4](#_Toc30132)

[2.3.3 主界面 4](#_Toc235)

[2.3.4 添加书籍与书籍详情界面 5](#_Toc6240)

[2.3.5 检索界面 6](#_Toc18468)

[第3章 产品的技术与实现 7](#_Toc12830)

[3.1 数据库 7](#_Toc13646)

[3.2 预加载 7](#_Toc8601)

[3.3 自定义条目与多选 7](#_Toc3991)

[3.4 自定义对话框 8](#_Toc12202)

[结 论 9](#_Toc18747)

[致 谢 10](#_Toc10078)

[参考文献 11](#_Toc991)

第1章 绪 论

## 1.1 本论文的背景和意义

手机已经是人们日常生活中不可缺少的工具，但如今市面上并没有一个面向爱好传统纸质书的管理工具，所以“我的书架”在这种情况下诞生。

## 1.2 本论文的主要方法和研究进展

本论文的主要为通过老师上课教授的Android studio的使用方法、HTML的相关知识和在遇到没有接触过的功能需求时通过参考上网搜索相关的实现方法学习如何实现其功能来完成的，在这个过程中提升了我学习的能力，拓展了我的视野，增强了我的编程能力。

## 1.3 本论文的主要内容

本文将详细讲解“我的书架”的数据结构，界面设计以及实现的技术与方法。

## 1.4 本论文的结构安排

本文共分为以下3个部分

(1)绪论：主要介绍了该应用的背景与主要作用。

(2)结构与界面设计：主要介绍了该应用的数据结构和应用界面进行设计。

(3)产品的技术与实现：具体的开发实现和细节，使用的具体的库和方法。

# 第2章 结构与界面设计

## 2.1 数据结构

“我的书架”使用了安卓内置的SQLite数据库，SQLite是一款轻型的数据库，是遵守ACID的关联式数据库管理系统，它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。比起 Mysql、PostgreSQL这两款开源世界著名的数据库管理系统来讲，它的处理速度比他们都快。由于SQLite的核心引擎本身不依赖第三方的软件，使用它也不需要安装。并且SQLite中所有的信息都包含在一个文件内。这个文件可以自由复制到其它目录或其它设备上。方便同步数据。该应用采取了[树形结构](https://baike.baidu.com/item/%E6%A0%91%E5%BD%A2%E7%BB%93%E6%9E%84?fromModule=lemma_inlink" \t "https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E7%BB%93%E6%9E%84/_blank)的数据结构，共用了user，site与book三个数据表。

### 2.1.1 user数据表

user数据表作为存放用户基础数据的表，结构如图所视。

表2-1 user表结构

|  |  |  |
| --- | --- | --- |
| **参数名** | **说明** | **类型** |
| id | 用户id(主键) | int |
| user\_name | 用户名 | string |
| password | 密码 | string |

### 2.1.2 site数据表

site数据表作为存放用户暑假基础数据的表，其中包含了来自user表的外键，其结构如图所视。

表2-2 site表结构

|  |  |  |
| --- | --- | --- |
| **参数名** | **说明** | **类型** |
| id | 地址id(主键) | int |
| user\_id | 用户id(外键来自use的id) | int |
| site | 地址 | string |

### 2.1.3 book数据表

book数据表作为存放用户暑假基础数据的表，其中包含了来自user表与site表的外键，其结构如图所视。

表2-3 book表结构

|  |  |  |
| --- | --- | --- |
| **参数名** | **说明** | **类型** |
| id | 书id(主键) | int |
| user\_id | 用户id(外键来自use的id) | int |
| book\_name | 书名 | string |
| author | 作者 | string |
| nationality | 国籍 | string |
| king | 类型 | int |
| comment | 备注 | string |
| site | 地址(外键来自site的id) | int |
| date | 最后更改日期 | string |
| like | 收藏 | int |

## 2.2 软件结构

在用户登录完成后进入主页，可通过滑动界面或点击底部导航栏对主页显示内容进行切换，可显示藏书展示或个人界面。可在藏书展示界面批量删除或添加书籍。个人界面提供了跳转到个人详情，书架详情，收藏界面和设置界面的按钮。检索界面可通过藏书显示的检索按钮，书籍详情，收藏界面等跳转。具体结构见下图。

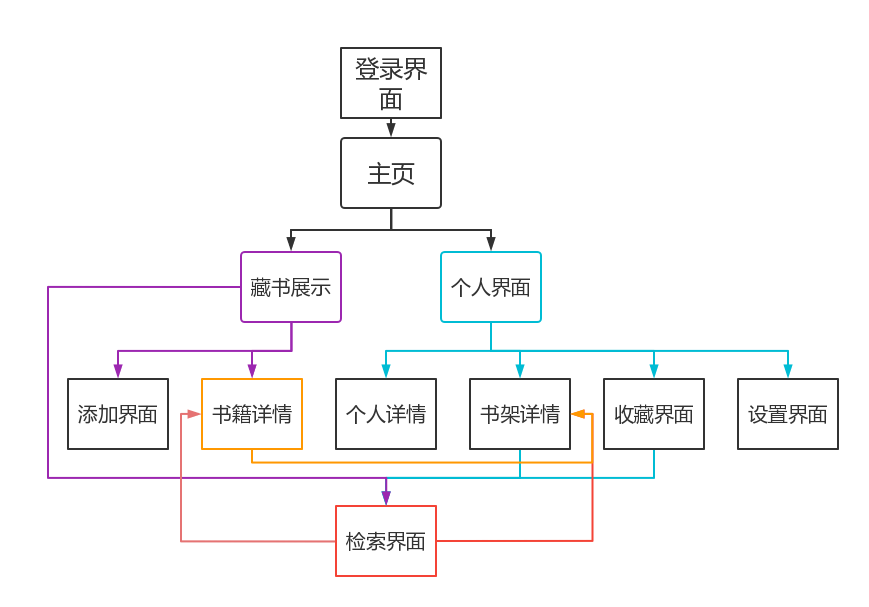


图2-1 软件结构

## 2.3 界面设计

“我的书架”是一款面对爱好传统纸质书籍的用户，此类用户安静，内敛，因此界面与图标风格应偏向简约可以清晰的看到书籍的信息。

### 2.3.1 应用图标

应用图标是以蓝白为主色调。通过简单的色块组成书架的轮廓。



图2-2 软件结构

### 2.3.2 登录/注册界面

该界面是以蓝白为主色调。通过点击底部文字进行转换。

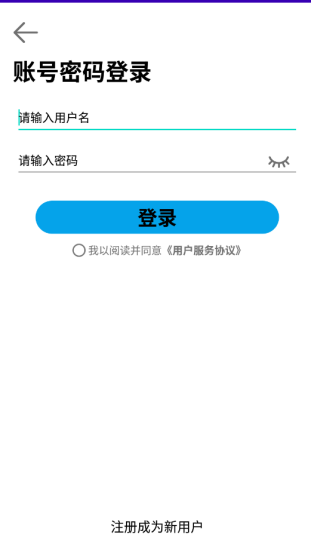
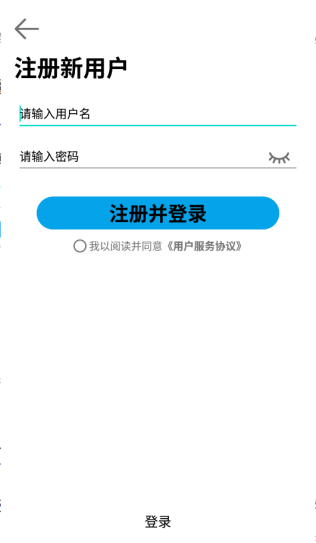
 

图2-3 登录界面 图2-4 注册界面

### 2.3.3 主界面

该界面是以蓝灰白为主色调。可通过滑动界面或点击底部导航栏对主页显示内容进行切换，可显示藏书展示或个人界面。在藏书展示中长按书籍条目可进入多选模式。

图2-5 藏书展示 图2-6 个人界面

图2-7 藏书展示填充数据 图2-8 藏书展示多选 图2-9 个人界面

### 2.3.4 添加书籍与书籍详情界面

添加书籍与书籍详情界面共同使用一个界面模板，通过在不同界面点击生成不同界面。该模板中包含了书籍了名称，作者，国籍，存放书架，备注等信息。

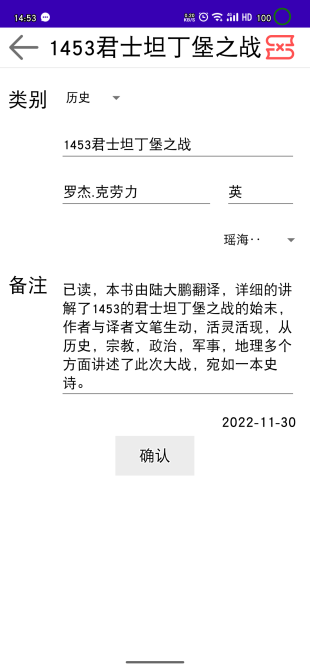
 

图2-10 添加书籍模式 图2-11 书籍详情模式

### 2.3.5 检索界面

该界面是对书籍进行检索的界面，会根据输入框中的内容进行模糊检索，在筛选中还可以调整检索种类，检索应用位置，存放地等。同时可以调整检索条目的排列顺序。该界面也可通过长按条目进入多选模式，对多条条目进行操作。

图2-12 检索界面 图2-13 检索界面多选 图2-14 筛选

# 第3章 产品的技术与实现

## 3.1 数据库

该软件使用了安卓内置的SQLite数据库。通过新建数据库类继承SQLiteOpenHelper类使用其getWritableDatabase()方法获取SQLiteDatabase对象，使用SQLiteDatabase对象的execSQL()，insert()，query()，delete()，update()方法实现对创建数据库，和数据库的增删改查操作。其中的execSQL()方法在SQLiteOpenHelper的onCreate接口实现仅一次创建数据库。

为了实现检索界面的模糊检索，筛选与排序在使用SQLiteDatabase对象的query()方法时，其中的selection，selectionArgsnew，orderBy变量是通过字符串拼接的方式完成。

## 3.2 预加载

预加载是通过FragmentActivity与ViewPager的方式实现，让主界面继承FragmentActivity类通过生成FragmentPagerAdapter并将其与需要预加载的FragmentActivity进行绑定再通过ViewPager对象的setAdapter()方法将FragmentActivity与ViewPager进行绑定。通过selectTab的接口完成底部导航的按钮与FragmentActivity预加载绑定。

这样就完成了预加载页面的设置，既可以通过滑动页面的方式切换预加载页面，也可以通过点击底部导航栏的按钮进行预加载页面的切换，同时底部按钮也会随着预加载页切换而进行变换。为了切换页面时数据可以刷新在onResume接口中添加了刷新数据的方法。

## 3.3 自定义条目与多选

定义条目与多选是通过 ListView与BaseAdapter完成的。通过继承BaseAdapter将其中的view通过LayoutInflat对象的from(context).inflate()方法将条目的布局文件与其绑定，并使用短时存储器提升其效率。

在显示代码中实例化刚刚自定义的BaseAdapter类，并将布局中的ListView控件通过ListView中的setAdapter()方法与BaseAdapter进行绑定。这样自定义条目就完成了。

而多选则是在实例化BaseAdapter类时多传入一组用于表示是否选中的参数，在BaseAdapter中设置区别，这样在显示代码中也可以通过这组参数区别出具体选中的数据，在触发添加或减少多选对象时通过生成新的BaseAdapter对象和setAdapter()与notifyDataSetChanged()方法刷新条目的显示。

## 3.4 自定义对话框

自定义对话框是通过AlertDialog.Builder与View实现。

通过构造方法生成AlertDialog.Builder对象，使用setView()将View对象与其绑定，最后通过Builder的show()方法生成对应的对话框。但对话框的实质其实还是View。

# 结 论

该应用使用SQLite搭建了一个小型的数据库，用于存储数据。用简约的界面把数据库中的数据以图形化的方式呈现出来。用户可以自由的添加和删除书籍。并拥有可以对书名.作者.备注等进行模糊检索的检索系统。通过FragmentActivity与ViewPager的方式实现了预加载，通过 ListView与BaseAdapter完成定义条目与多选的功能。AlertDialog.Builder与View实现自定义对话框。

以上就是“我的书架”应用。作为一个管理书籍的应用，可以满足绝大部分的需求，但还是有许多的不足。在之后的更新里会逐步完成其余类似联网.等功能等