

# 深 圳 大 学

## 实 验 报 告

课程名称： 数据库系统

实验序号： 实验四

实验名称： 团体项目结项报告

项目名称： 旧书交易系统（OBT）

组 别： OBT 一人组

成员名单：

学 号： 2019151067 姓 名： 金惠童

实验日期： 2021 年 6 月 16 日-2021 年 6 月 27 日

## 目录

一、	实验目的:	3
二、	实验要求:	3
三、	实验设备及环境配置:	3
四、	实验内容	3
1.	项目简介	3
2.	项目架构	5
3.	项目关键设计: (关键的数据表, 以及其完成某关键功能的代码片段)	12
4.	项目效果图	23
5.	任务分工及所完成的状况。评分标准	27
五、	问题分析	27
1.	安卓直连 mysql	27
2.	关于 mysql 存储图片的问题	29
六、	实验心得	30
七、	参考文献	31
八、	诚信承诺	33

## 一、 实验目的：

1. 了解 DBMS 系统的功能、Web based 数据库的软件组成及工具；
2. 掌握数据库软件的使用方法；
3. 掌握 php + mysql 或其他界面结合数据库操作方法；
4. 掌握数据库系统软件的制作过程。

## 二、 实验要求：

1. 构思一个以数据库为后台的，具备某种功能的软件系统；
2. 从模块化描述该系统的功能，对各个模块进行描述；
3. 按照每个模块的功能，设计相应的数据库以及程序；
4. 完成模块间的接口，整合成为一个完整系统并测试所有的功能；
5. 按要求完成开发报告并按时提交

## 三、 实验设备及环境配置：

计算机、数据库管理系统如 DB2, mysql 、Android Studio 等软件。

## 四、 实验内容

### 1. 项目简介

#### ➤ 概述

OBT (old book trade) 深大旧书交易平台，是专门为深大学生量身定制的二手教科书、二手课外书等书籍的交换平台。同学们用过的教科书、课外书都可以用在本平台上换成需要的教科书或者课外书。旧书交易系统是为师生服务的公益平台项目，旨在实现书籍回收利用、促进学习交流、节省学生开销等宗旨。目前 OBT 已经完成安卓手机端的开发，未来将对第一版本进行完善，并择机开发 ios 版本和鸿蒙版本。

#### ➤ 意义

OBT 旧书交易系统是一个专门为深大学生量身定做的二手书交易平台，其意义总结有以下几点：

- 1) 实现校内图书的回收利用，为绿色环保、开源节流贡献自己的一己之力；
- 2) 促进同学之间的交流与联系，切实解决卖书难、买书难的问题。

- 3) 不经过第三方商家, 实现没有中间商赚差价, 真正实现**压低图书价格**, 平常在其他旧书店 20 块钱才买得到的书, 在 OBT 上只需 10 元就能买到;
- 4) 探索**使用 Q 点结算**, 不涉及金钱交易, 真正实现图书互换。

经过我们前期的市场调研, 目前深大大概有 66.3% 以上的学生有意愿购入和使用二手书, 而他们迫于当前二手书交易的局限, 只能从校外的旧书店用较高的价格购买。而 OBT 的推出, 将有望于改变这一现状, **实现每年 5000 册图书的交换, 惠及深大 3 万多人**。同时也将助力绿色环境建设, 帮助贫困学生以较低价格获得心仪图书。

#### ➤ 软件卖点

##### A . 避免资源浪费

很多同学的教材使用完之后, 多数选择扔掉或者放着不用, 这会造成极大的资源浪费。如果提供 OBT 这样一个平台, 就能让资源循环利用。既能环保, 又可以节省同学们的日常开支。

##### B . 使用数据库驱动

鉴于教材的多种多样, 我们根据深大的课程体系, 按照年级、专业进行分类, 并为每一本书贴上标签, 分好编号, 使得 OBT 平台更好地为深大学子服务。

##### C . 不止于购书

OBT 平台将提供论坛专区, 用于同学们在上面分享读书笔记, 看书心得, 或者是分享考试复习资料, 探讨学习问题等, 将会是一个校内的专属学习论坛

#### ➤ 软件创新点

##### A . 交易方式不同

与传统交易平台不同, OBT 将采用类似于“物物交换”的交易模式, 所有交易不涉及金钱, “卖”出去的书籍将根据定价和新旧程度获得一定数量的 Q 点, 后续可以使用 Q 点“购入”自己需要的书籍。

##### B . 为深大学子量身定做

将二手书在校内流传, 一是避免了中间商赚差价, 二是由于课程体系和教师差异不大, 教材上的笔记等内容都能很好地被再利用。所购入的二手书也更加适合自己。

##### C . 提供互评机制

与现行单方面的评价不同, OBT 平台买卖双方都可对对方做出评价, 评价包括打分和文字评论 (可配上图片), 系统将根据评价生成个人信用分, 此外, 当进行交易的时候, 同学们可以根据评价选择适宜的买方或者卖方。

#### ➤ 主要功能

- A . 非常细节地复现目前主流 App 的登录注册功能, 带有正则表达式判断、随机图片

验证码、以及手机验证码等功能。

B. 模拟淘宝、京东首页设计，具备“轮播广告位”，可以引入广告获取收益，从而维系平台的维护，并且利用优质广告丰富同学们的生活。

C. 和现有平台不同，OBT 实现买卖一体化，只需要下载一个 App 就可以完成买、卖、售后相关操作，操作方便，节约手机内存。

D. 和现有平台不同，OBT 不涉及金钱交易，单纯地使用 Q 点作为物与物交换的媒介，所有交易使用 Q 点进行结算，能够模拟实现书籍交换的机制。

E. 图书管理方面：实现基本的增删查改、以及根据特定条件、复合条件排序。有购物车功能。

F. 用户管理方面：用户可以上传头像、修改更新个人资料，多用户时还可以实现自由的添加、删除、切换等。

G. 订单管理方面：包含未支付、已支付、待收货、确认收货、评价等阶段属性。并根据图书卖家和买家对这些订单进行标识，从数据库读取时采用嵌套查询的方式。

## 2. 项目架构

本项目共包含五大模块：登录注册模块、首页设计模块、图书管理模块、用户管理模块、订单管理模块，每个模块之间的简单关系如图 4-2-1 所示，整个项目的流程请参见中期报告。接下来，我们将逐一对每个模块的内容和功能进行详细的介绍。

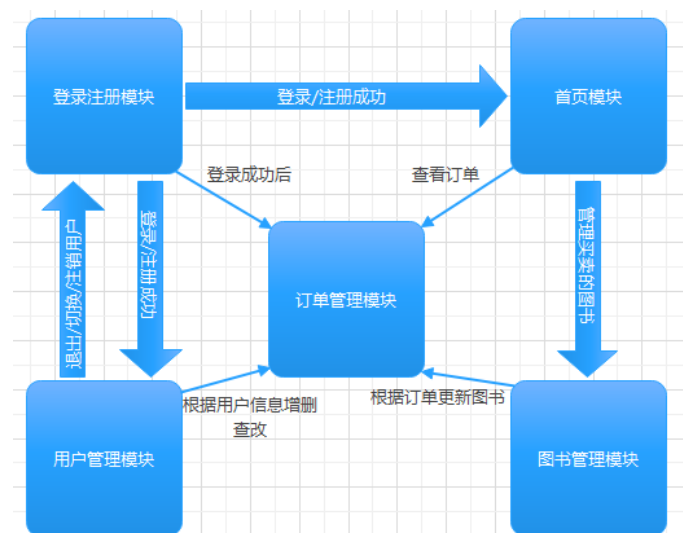


图 4-2-1 各个模块的简单关系

### (1) 登录注册模块

#### ➤ 包含的活动和功能

登录注册模块主要包含三个活动的内容：LoginActivity（登录）、RegisterActivity（注册）、

ForgotPasswordActivity（忘记密码）。其中每个活动包含的功能如表 1 所示。

表 1 登录注册模块包含的活动和功能

活动	功能
LoginActivity	1. 根据用户输入的账号实时匹配头像； 2. 提供记住密码功能； 3. 提供随机验证码，防止机器登录，安全性好； 4. 提供用户存在性检测、密码匹配功能。
RegisterActivity	1. 对手机号码、密码、学号分别进行不同要求的正则化检查 2. 提供手机验证码验证，防止用户滥用手机号 3. 用户在注册的时候可以选择上传自己的头像
ForgotPasswordActivity	1. 提供手机验证码功能，验证通过才可修改密码 2. 提供用户存在性检差，如果不存在当然无法修改 3. 对新设置的密码提供正则表达式检查

➤ 流程图

如图 4-2-2 所示，从左到右依次是登录、注册、找回密码的逻辑流程图。我们可以看到每一步的操作都非常严谨地保证了系统的安全性，对账号的保护包括但不限于人机，另外对用户的开放度也做到适度并友好，用户可以轻松修改并找回自己的密码，同时还可以更换自己喜欢的头像。

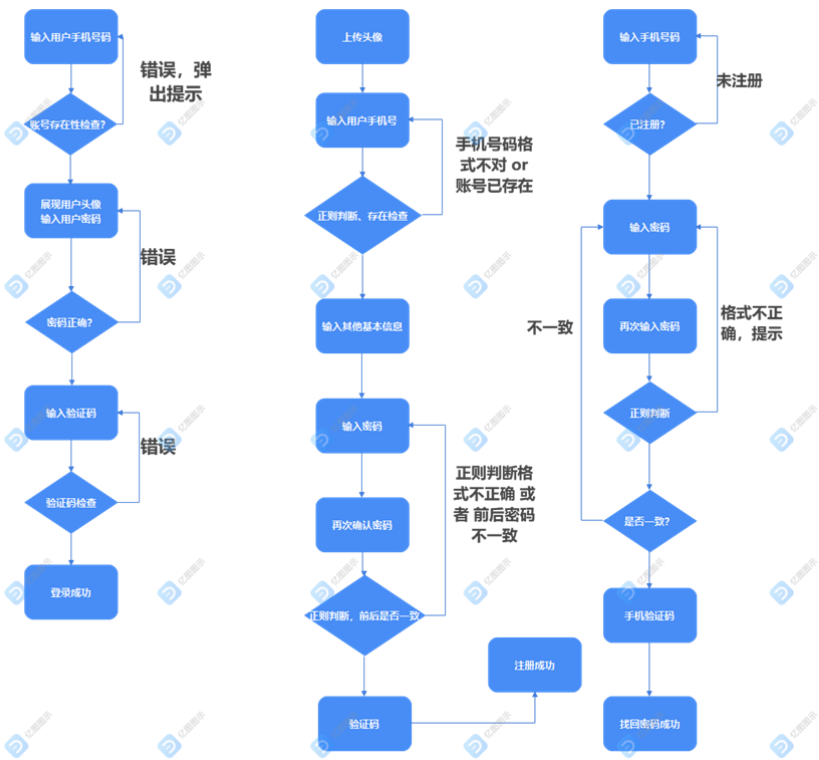


图 4-2-2 登录注册模块流程图

➤ 前后端相关代码

如表 2 所示，罗列出该模块设计的相关代码文件。

表 2 登录注册模块相关文件

Java 代码	布局文件 XML 代码	支撑类
ForgetPasswordActivity, LoginActivity, RegisterActivity,	activity_login.xml, activity_register.xml, activity_forget_password.xml	MyDatabaseHelper, DBConnectHelper, IdentifyCode

(2) 首页设计模块

➤ 包含的活动和功能

首页设计模块主要包括两个活动：HomeFm（首页模块）和 SearchActivity（搜索活动）。其中每个活动包含的功能如表 3 所示。

表 3 首页设计模块的活动和功能

活动	功能
HomeFm	1. 具备轮播广告位，可以投放优质广告 2. 广告位下面，是优质图书推荐，可以根据用户习惯推荐相应图书 3. 用户可以对推荐的图书选择加入购物车或者购买
SearchActivity	1. 使用哈希表对数据库的一些关键信息进行索引 2. 支持用户模糊搜索，不同关键词搜索 3. 支持按照特定条件对搜索结果进行排序 4. 支持多个条件复合对搜索结果进行排序

➤ 流程图

如图 4-2-3 所示，从左到右依次是轮播广告位、智能推荐、搜索活动的流程图。可以看到，我们对轮播广告位的嵌入自然而合理，对于搜索功能的实现符合用户平常习惯，并且我们也不断更新用户的喜好，并根据喜好推荐相应内容。

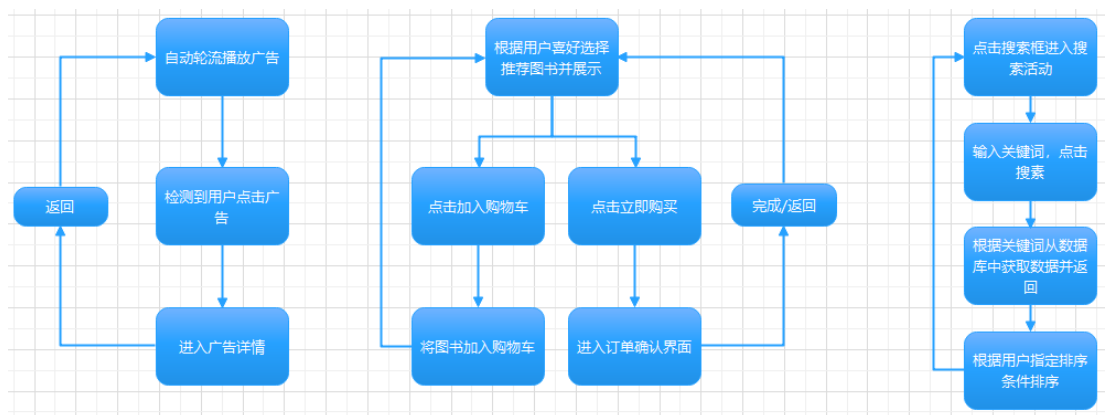


图 4-2-3 首页模块流程图

### ➤ 前后端相关代码

如表 4 所示，罗列出该模块设计的相关代码文件。

**表 4 首页模块相关前后端代码文件**

Java 代码	布局文件 XML 代码	支撑类
HomeFm	fm_home.xml	MyDatabaseHelper,
SearchActivity	home_book_item.xml	DBConnectHelper,
BookHomeAdapter	activity_search.xml	BookShop

### (3) 图书管理模块

#### ➤ 包含的活动和功能

图书管理模块主要涉及：HomeFm（首页），CartFm（购物车），ShopFm（我的商店），AddNew（添加新书）四个活动。其中每个活动及其对应的功能如表 5 所示。

**表 5 图书管理模块涉及的活动和功能**

活动	功能
HomeFm	1. 图书推荐功能 2. 对喜欢的图书加购和购买
CartFm	1. 加入购物车的书籍都在这里 2. 根据购买记录更新书籍 3. 可以结算购物车的书籍或者删除购物车的书籍
ShopFm	1. 查看我正在卖的图书 2. 管理现在在卖的图书，包括增删查改
AddNem	1. 添加新书，编辑对应信息



➤ 流程图

如图 4-2-4 是我的商店（即卖家）图书管理，如图 4-2-5 是购物车的图书管理，图 4-2-6 是首页的图书管理。

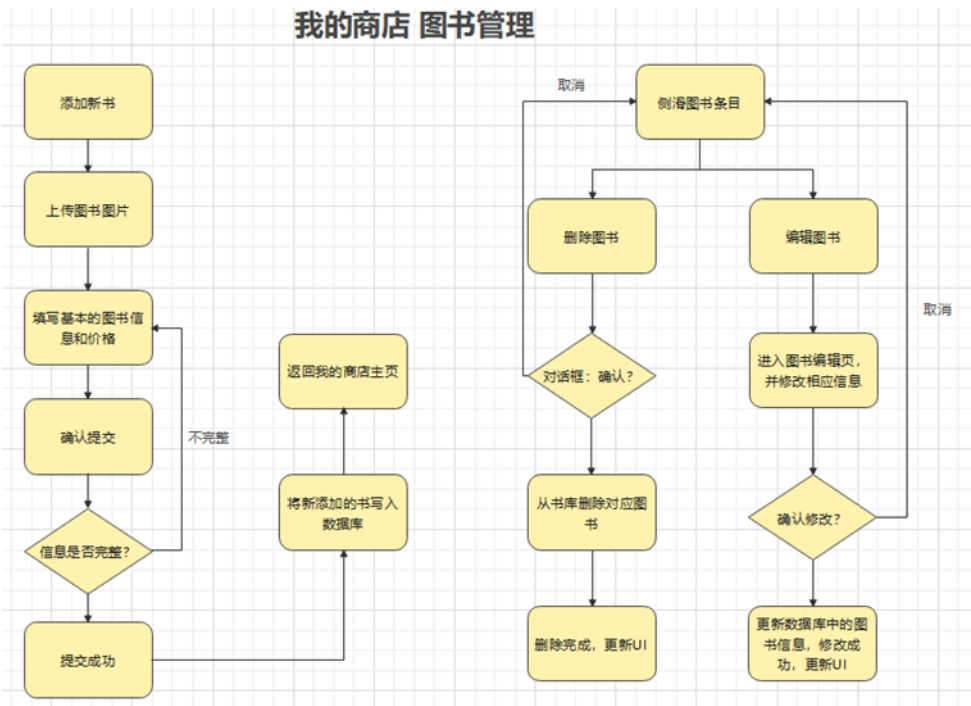


图 4-2-4 我的商店图书管理流程图

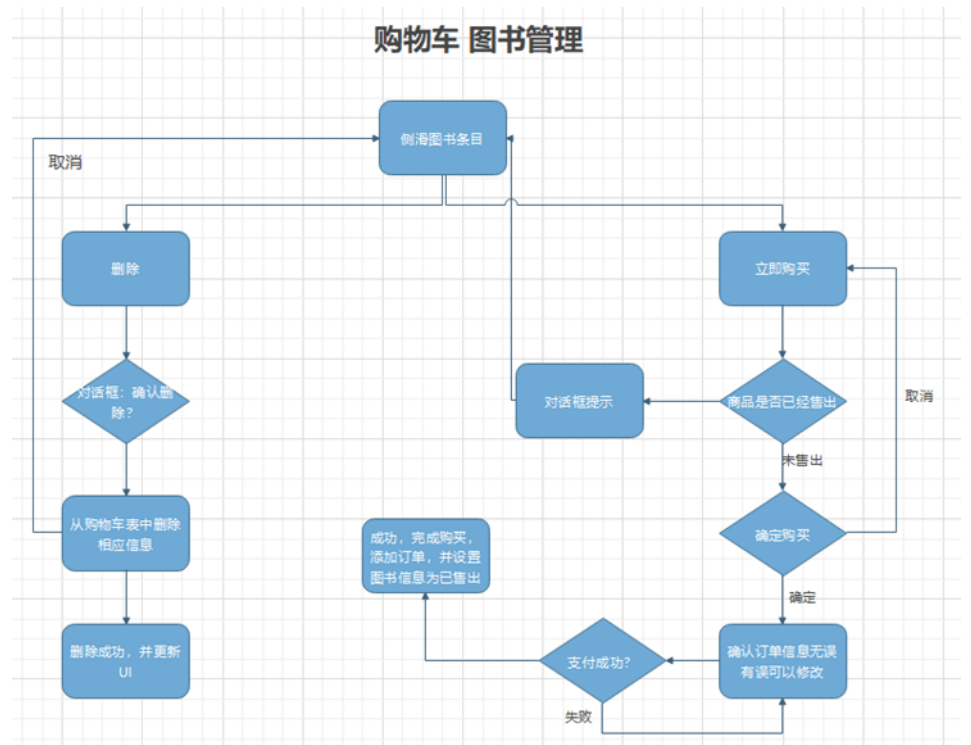


图 4-2-5 购物车图书管理流程图

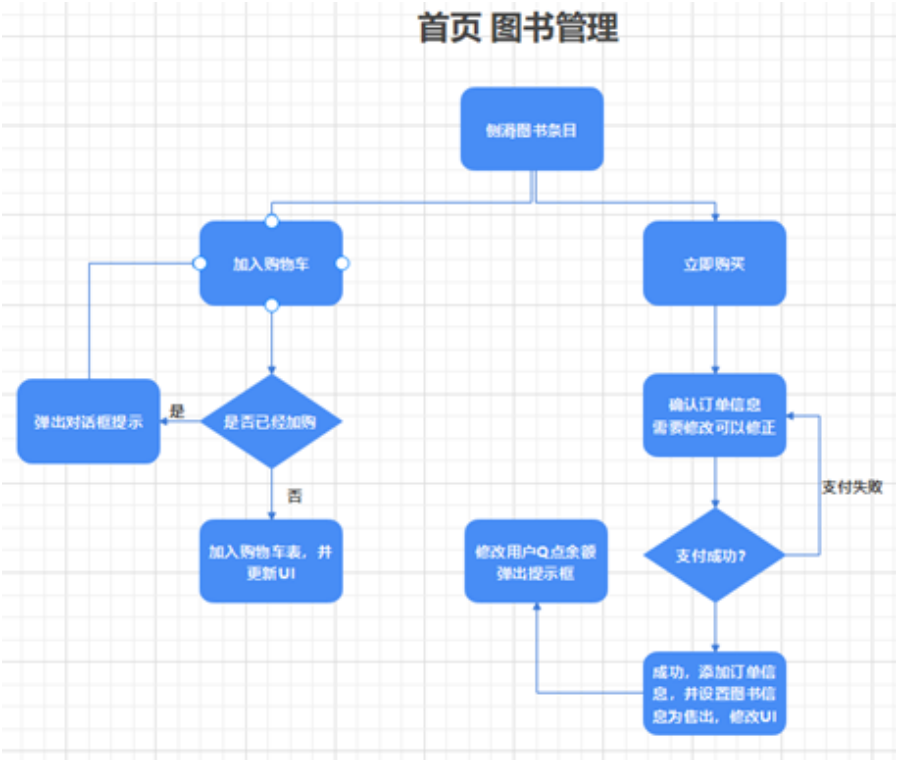


图 4-2-6 首页图书管理流程图

➤ 前后端代码

如表 6 所示，罗列出该模块设计的相关代码文件。

表 6 图书管理模块相关代码文件

Java 代码	布局文件 XML 代码	支撑类
MainActivity	fm_home.xml	DBConnectHelper,
AddNew	home_book_item.xml	BookShop
CartFm	fm_cart.xml	BookCartAdapter
HomeFm	fm_shop.xml	BookHomeAdapter
ShopFm	add_nem.xml	BookShopAdapter

(4) 用户管理模块

➤ 包含的活动和功能

用户管理模块主要涉及 MineFm、LoginActivity、RegisterActivity 和 ForgetPassWordActivity 四个活动。其中每个活动对应的功能如表 7 所示。

表 7 用户管理模块相关活动和功能

活动	功能
MineFm	1. 查看个人信息，个性化设置等（类似微信） 2. 可以进入个人主页，对个人信息进行修改和更新 3. 拥有多用户管理功能，用户可以选择切换到不同的账号
LoginActivity	1. 根据用户输入的账号实时匹配头像； 2. 提供记住密码功能； 3. 提供随机验证码，防止机器登录，安全性好； 4. 提供用户存在性检测、密码匹配功能。
RegisterActivity	1. 对手机号码、密码、学号分别进行不同要求的正则化检查 2. 提供手机验证码验证，防止用户滥用手机号 3. 用户在注册的时候可以选择上传自己的头像
ForgetPassWordActivity	1. 提供手机验证码功能，验证通过才可修改密码 2. 提供用户存在性检差，如果不存在当然无法修改 3. 对新设置的密码提供正则表达式检查

➤ 流程图

由于登录注册的流程图前面已经叙述过了，这里只叙述 MineFm 也即个人中心的流程图。如图 4-2-7 所示。

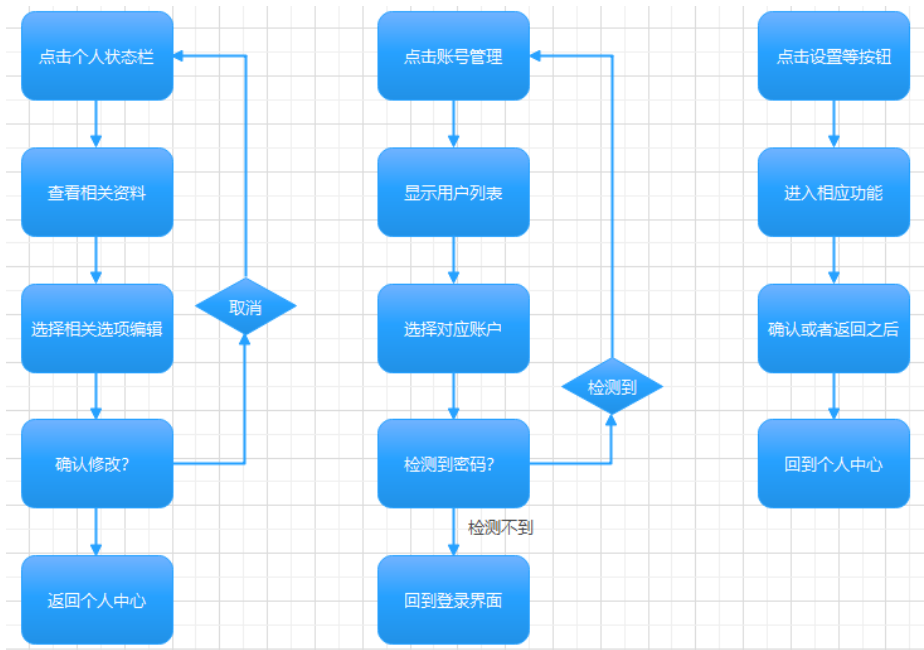


图 4-2-7 Mine（个人中心）流程图

➤ 前后端代码

如表 8 所示，罗列出该模块设计的相关代码文件。

表 8 用户管理模块相关代码文件

Java 代码	布局文件 XML 代码	支撑类
MainActivity	fm_mine	DBConnectHelper
MineFm	activity_login.xml	User
LoginActivity	activity_register.xml	UserInfo
RegisterAcitivity	activity_main.xml	DBHelper

#### (5) 订单管理模块

这一模块由于项目时间有限, 还没有实现用户界面, 不过其他模块已经预留了相应的 API 接口, 数据库也设计好了, 后面会继续将项目做完整。

### 3. 项目关键设计: (关键的数据表, 以及其完成某关键功能的代码片段)

#### (1) 数据库设计

##### ➤ 登录注册模块

这一块数据库主要分为两张表, 一张是存放账号和密码的 user 表, 一张是存放账号和详细信息的 userInfo 表。如图 3-1-1 所示。

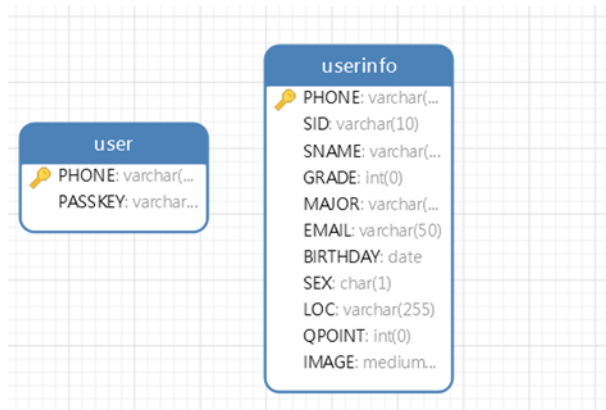


图 3-1-1 登录注册模块数据表

这里主要要对几点进行说明:

- A. 由于手机号、密码、学号这些不需要参与任何的运算, 只是一串数字, 因此采用 varchar 存储, 这样可以有多少位就存多少位, 节省空间。
- B. 对于用户头像 IMAGE, 在 java 中, 我们需要将一张图片先转为 bitmap 编码, 然后再转为 byte 数组, 然后转成 String 类型, 最后使用 blob 存储。这里不用 text, 是因为其空间大小不足。
- C. 将账号密码和账号信息分开存放, 是为了安全起见, 因为后面会有很多环节需要访问信

息表，如果密码也放在信息里面，就很容易造成信息泄露。

D . 记住密码功能，是将密码保存到手机数据库中（使用 SQLite），然后进入登录界面的时候自动填充。

➤ 首页设计模块

这个模块主要涉及 book 表，从 book 里面选出相应的推荐书籍展示出来，然后点击购买或者加购操作会跳到其他对应的模块进行完成。如图 3-1-2 所示。

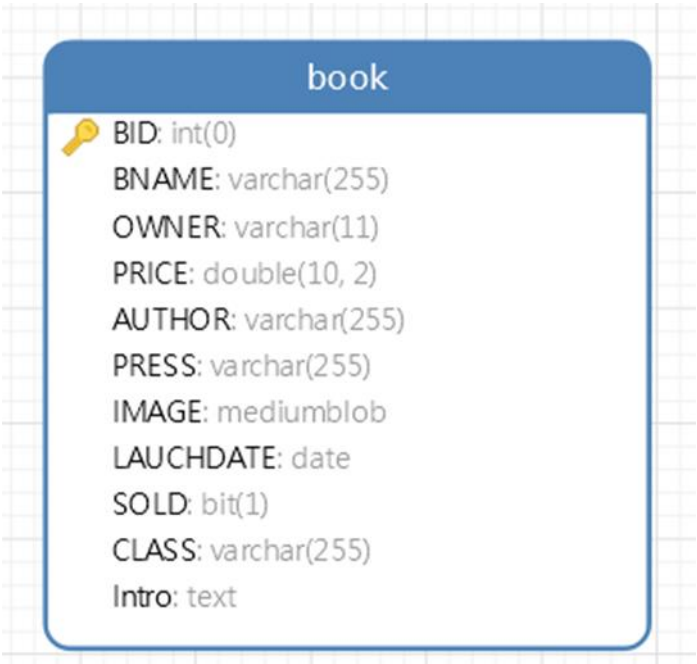


图 3-1-2 首页设计模块数据表

这里需要重点说明几点：

- A . 每一本图书都有唯一的序列号 BID，索引的时候都是根据 BID 索引的。
- B . 每一本图书都适用 SOLD 标记是否已经售出，如果售出，其他用户便不能再加购或者购买，也不能展示在商品页。
- C . 图书图片还是跟用户头像一样，适用 Blob 存储，解码和译码过程也大概一致。
- D . CLASS 主要用于对图书分类，方便后续的智能推荐和排序。

➤ 图书管理模块

图书管理模块会涉及到 book 表，购物车表，订单表等，分别对应展示、加购等操作，他们之间也存在一定的关系，比如一本书可以被多个购物车（以用户手机号作为索引）收藏，一个购物车也可以收藏多本数，因此他们是多对多的关系；而对于订单而言，因为一个订单只能有一本书，而一本书也只能对应一个订单，因此订单和书籍是一对一的关系。其数据表如图 3-1-3 所示，ER 图（省去属性，只展示关系）如图 3-1-4 所示。

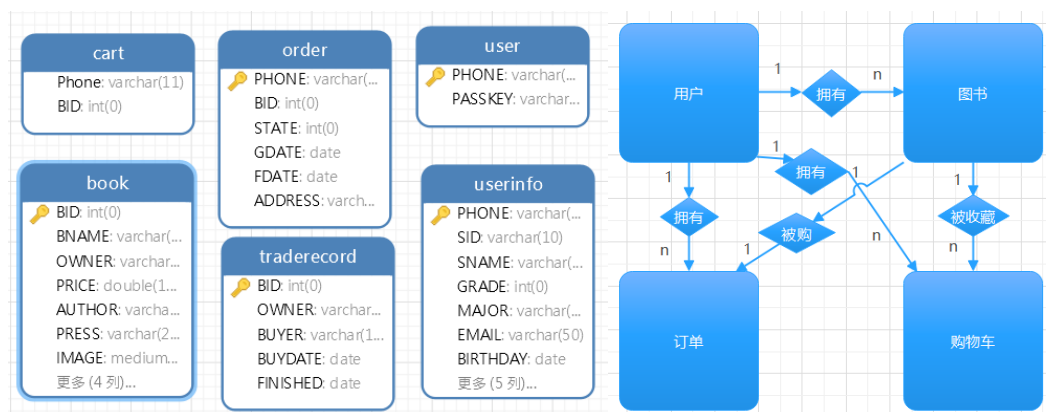


图 3-1-3 图书管理模块数据表

图 3-1-4 图书管理模块 ER 图 (省去属性)

这里需要重点说明几点：

A . 关于订单状态的说明：为了更节省空间地表示其状态，这里使用 int 型，我们规定 -1 表示未支付状态、1 表示已支付未发货状态，2 表示已发货未收货状态，3 表示已收货状态，4 表示订单完成（即卖方收到 Q 点）。

B . 其余的表在之前也相应介绍过，这里不再重复。注意，这里为了方便，使用账户登录密码作为支付密码。

C . 在用户和订单因为是 1 对多的关系，所以我们需要将用户的主码（手机号码）作为订单的外码。其余情况也类似。

### ➤ 用户管理模块

这个模块主要涉及用户的一些信息，以及手机自身的 SQLite 数据库相关表（不同于 mysql，不能多台手机共享），其中主要包括 user 表、userInfo 表、userlist 表（SQLite）。他们之间存在一些关系，比如 user 和 userInfo 是一一对应的，把密码分开其实是为了账号的安全，因为我们在使用 app 的过程中会经常性的访问数据表，如果密码放在一起，就很有可能会泄露。然后 userlist 和 user 是一对多的关系，userlist 存放着添加在手机中的用户列表（包括密码），当用户切换时，可以自动取出账号密码去 mysql 数据库匹配。其数据表和 ER 图（只画出表之间的关系）如图 3-1-6 所示。

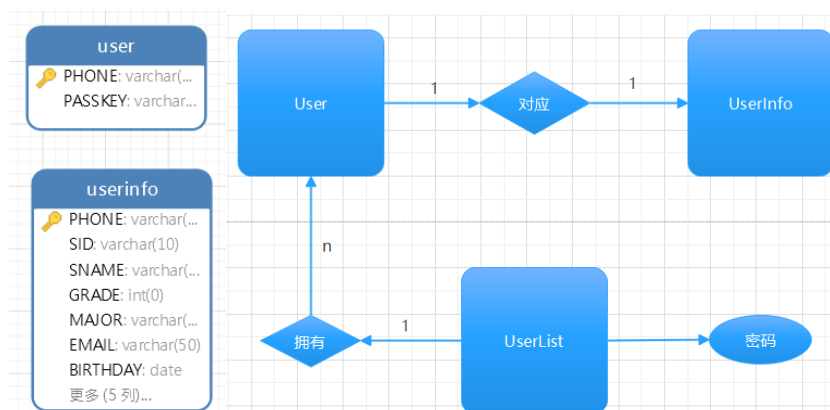


图 3-1-6 用户管理模块数据表 (左) 和 ER 图 (右)

这里需要重点说明几点：

- A. 和主流的 App 一样，我们为了不增担手机的负担，只将用户的账号名和密码存放在 userList 中，并且为了保证密码的安全，我们在后端代码中实现了对密码的加密。
- B. 对于 User 表，因为账号是惟一的，因此在匹配账号密码的过程中，我们加入条件 limit 1，这样当找到一个匹配关系后就结束，不会浪费额外的不必要查询。
- C. 对于 phone 和 sid（学号），因为我们并不需要对其进行数据上的运算，而是将其作为一个索引值，所以使用 varchar 数据类型。考虑到一个学生只有一个学号，但是一个学生可以使用不同手机号码注册不同的账号，因此我们使用手机号码作为账号的主键，但是使用学号作为学生的实名认证。

### ➤ 订单管理模块

这一模块主要涉及的到数据表有订单信息表和交易记录表，其中订单信息表主要包含一些重要的买卖双方的信息，比如交易双方账号、交易状态等，而交易记录表主要是记录相关的交易快照，存储着订单号和交易状态。交易记录和订单信息是一一对应的，交易记录的设置一是方便查询，二是作为订单重要的存根。如图 3-1-7 所示，是订单管理模块对应的数据表和 ER 图（省略属性）。

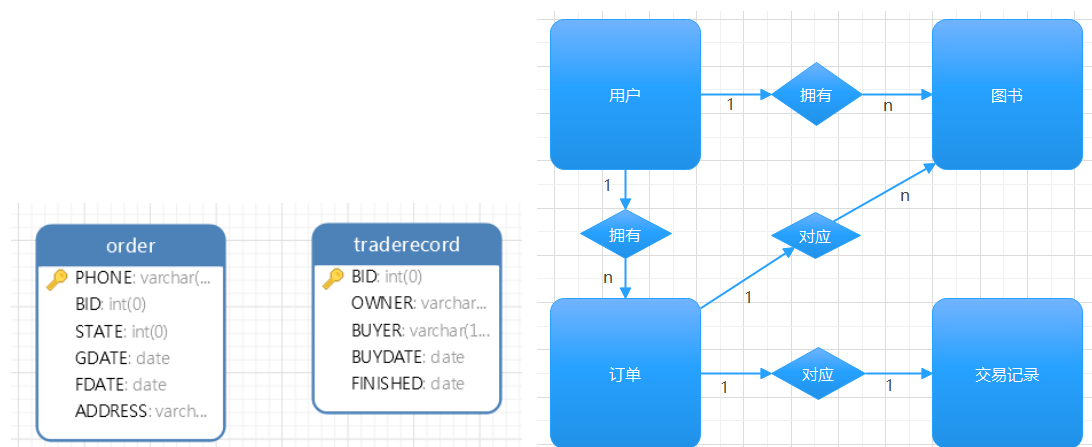


图 3-1-7 订单管理模块数据表（左）和 ER 图（右）

这里需要重点说明几点：

- A. 为了节省空间，我们使用一个 int 型数据来记录不同的订单状态，比如使用 0 表示订单处于刚创建为付款状态，使用 1 表示订单已付款为发货状态，使用 2 表示订单已发货未收货状态，以此类推。
- B. 另外，在每次订单创建的同时，我们要更新 book，将图书设置为已售出，避免重复出售的情况。

### (2) 关键功能及代码

#### ➤ 连接远程 mysql

因为我们每次使用 jdbc 连接，都需要创建一个连接器，所以我们这里干脆将其写成一个 Helper 类，并且在里面放着我们的 ip 地址和密码，以及数据库的相关信息，如图 3-2-1 所示。

```
private static String driver = "com.mysql.jdbc.Driver";
// 加入utf-8编码，避免中文乱码
private static String url = "jdbc:mysql://192.168.1.106/mysql2021?characterEncoding=utf-8" + "&serverTimezone=PDT";
//private static String url = "jdbc:mysql://192.168.1.106:3306/mysql2021?characterEncoding=utf-8" + "&serverTimezone=PDT";
private static String user = "root"; // 用户名
private static String password = "123456"; // 密码

/*
 * 连接数据库
 */
public static Connection getConn() {
    Connection conn = null;
    try {
        Class.forName(driver);
        conn = (Connection) DriverManager.getConnection(url, user, password);
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    if (conn == null)
        System.out.println("conn is null!!!!!!");
    return conn;
}
```

author: 金惠童

图 3-2-1 连接远程数据库

#### ➤ 正则表达式检查手机号、学号和密码的合法性

以检查密码合法性为例，我们设定密码长度 6-16 位，并且必须至少两种字符以上组合。首先我们要创建正则表达式，然后使用这个正则规则检查密码，根据检查结果我们给出用户不同的提示信息。如图 3-2-2 所示。

```
// 密码长度要求6到16位，可以包含数字、字母和字符，不能包含中文
String Regex_password = "^(?![a-zA-Z]*$)(?!\\D+$).{6,16}$";
Pattern pattern = Pattern.compile(Regex_password); // 将正则表达式编译成模板
// 进行匹配
Matcher matcher = pattern.matcher(password_1);
if (!matcher.find()) {
    Toast.makeText(context, RegisterActivity.this, text, "密码格式错误", Toast.LENGTH_SHORT).show();
} else {
    id = studentId.getText().toString();
}
```

author: 金惠童

图 3-2-2 正则表达式检查密码合法性

#### ➤ 向 mysql 数据库写入注册的账号密码和账号信息

这里我们获取用户的相关信息之后，建立数据库连接器，然后开启一个子线程写入，写入成功就是注册成功，如果写入不成功给用户一定的回馈信息。如图 3-2-3 所示。



```

private void insertUser() {
    phone = userPhone.getText().toString();
    password_1 = userPassword1.getText().toString();
    name = userName.getText().toString();
    id = studentId.getText().toString();
    grade = Integer.parseInt(user_grade.getText().toString());
    major = user_major.getText().toString();
    new Thread((Runnable) () -> {
        Connection connection = DBConnectHelper.getConnection();
        String sql_1 = "INSERT INTO USERINFO(PHONE, SID, SNAME, GRADE, MAJOR, QPOINT, IMAGE) " +
            "VALUES('"+phone+"','"+id+"','"+name+"','"+grade+"','"+major+"', 0, '"+imageString+"')";
        String sql_2 = "INSERT INTO USER(PHONE, PASSKEY) " +
            "VALUES('"+phone+"','"+password_1+"')";
        Statement statement;
        try {
            statement = connection.createStatement();
            statement.executeUpdate(sql_1);
            statement.executeUpdate(sql_2);
            Message msg = new Message();
            msg.what = INSERT_FINISH;
            handler.sendMessage(msg);
            statement.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }).start();
}

```

author: 金惠童

图 3-2-3 向 mysql 数据库写入注册时的账号密码和账号信息

#### ➤ 注册和找回密码时，发送手机短信验证码，并检查验证码正确性

这一部分我们使用了第三方的 SDK，有一部分免费的资源开放，具体的配置这里不展开细说，主要展示一下使用时的代码，我们通过网络向第三方服务器发送请求之后，该服务器会发送一个验证码到手机端，然后用户填写验证码之后，我们重新向第三方服务器发送请求验证验证码的正确性，通过返回的结果我们执行对应的操作。如图 3-2-4 所示。

```

// 通过sdk发送短信验证码(请求获取短信验证码，在监听(eh)中返回)
SMSSDK.getVerificationCode(country, phone);
// 做倒计时操作
Toast.makeText(context, RegisterActivity.this, text: "已发送", Toast.LENGTH_SHORT).show();
sendMsg.setClickable(false);
// register.setEnabled(false);
sendMsg.setText("重新发送("+TIME+")");
// 开启一个子线程进行一分钟倒计时
new Thread((Runnable) () -> {
    for (int i = TIME; i > 0; i--) {
        TIME--;
    }
}).start();

```

author: 金惠童

图 3-2-4 发送手机短信验证码

#### ➤ 登录时，查看账号和密码是否匹配

这里依然适用 mysql 语句进行查询，为了查询快速，限制条件 limit 1，即找到一个匹配结果后就返回。如图 3-2-5 所示。



图 3-2-5 匹配账号密码

### ➤ 登录时，随机图片验证码的生成与匹配

这里的图片验证码是后台自己生成的，其逻辑也比较简单，就是随机生成四个字符，然后使用安卓自带的绘图工具将这四个字符画出来，加上一些干扰就完成了，为了方便使用，我们写了一个图片验证码的支撑类，其部分代码如图 3-2-6 所示，使用方法如图 3-2-7 所示。



图 3-2-6 identifyCode 部分代码

```

// 判断验证码是否正确
private boolean judgeIdentifyCode() {
    String phoneCode = et_phoneCode.getText().toString().toLowerCase();
    if (".".equals(phoneCode)) {
        Toast.makeText(context: LoginActivity.this, text: "请输入验证码",
            Toast.LENGTH_SHORT).show();
        return false;
    } else if (phoneCode.equals(realCode)) {
        return true;
    } else {
        Toast.makeText(context: LoginActivity.this, text: "验证码错误",
            Toast.LENGTH_SHORT).show();
        iv_shoeCode.setImageBitmap(IdentifyCode.getInstance().createBitmap());
        realCode = IdentifyCode.getInstance().getCode().toLowerCase();
        return false;
    }
}

```

author: 金惠童

图 3-2-7 随机图片验证码的生成与匹配

### ➤ 记住密码功能

主要使用 checkbox 供用户选择是否记住密码，如果记住密码，那么便将用户的密码存储到 SQLite 中，在下次进入的时候自动填充即可，如图 3-2-8 所示。

```

SQLiteDatabase db = SQHelper.getWritableDatabase();
Cursor cursor = db.query(table: "rememberAccount", columns: null, selection: null, selectionArgs: null,
    groupBy: null, having: null, orderBy: null, limit: null);
if (cursor.moveToFirst()) {
    telPhone.setText(cursor.getString(cursor.getColumnIndex(columnName: "account")));
    password.setText(cursor.getString(cursor.getColumnIndex(columnName: "password")));
    remember_pass.setChecked(true);
    String account = cursor.getString(cursor.getColumnIndex(columnName: "account"));
    String password = cursor.getString(cursor.getColumnIndex(columnName: "password"));
    db.delete(table: "rememberAccount", whereClause: "account=?", new String[] {account});
    db.delete(table: "rememberAccount", whereClause: "password=?", new String[] {password});
}

```

author: 金惠童

图 3-2-8 记住密码功能的实现

### ➤ 在 mysql 中存储图片

将图片转成 bitmap 格式，再转成 byte 数组，最后转成 String，如图 3-2-9 所示。

```

// 将图片转换成Base64编码
try {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    Bitmap bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), uri);
    bitmap.compress(Bitmap.CompressFormat.JPEG, quality: 100, baos);
    byte[] imageBytes = baos.toByteArray();
    imageString = Base64.encodeToString(imageBytes, Base64.DEFAULT);
} catch (Exception e) {
    e.printStackTrace();
}

```

author: 金惠童

图 3-2-9 在 mysql 中存储图片

### ➤ 轮播图的实现

在安卓开发当中，轮播图的概念在现行的主流 App 中时常见到，其主要的实现原理其实也并不复杂。主要的难点集中在几个方面：

A. 将图片设置成可以滑动模式：使用 ViewPager 可以将图片放入，并实现左右的滑动。如图 3-3-10 所示。

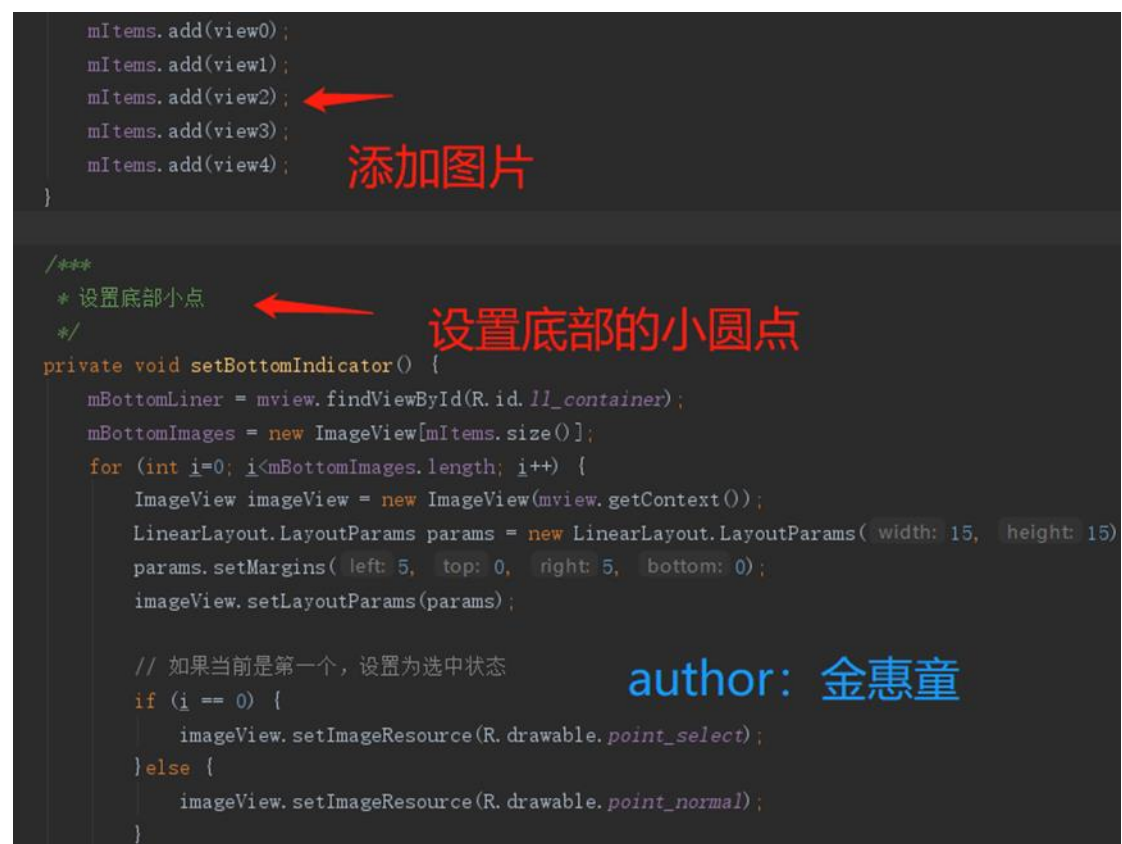


图 3-3-10 添加图片 设置小圆点

B. 如何实现自动轮播：使用 Java 的线程机制，用 Thread#sleep 设计定时器，定时向主线程发送信息，主线程接到信息后就可以更新 UI 了。如图 3-3-11 所示。



图 3-3-11 定时器实现

C. 如何在用户长按图片时停止自动播放，松开后又继续自动播放：这个主要是依赖安卓系统自带的监听器，监听用户是否长按，如果长按我们先暂停定时器即可。如图 3-3-12 所示。

```

@Override
public boolean onTouch(View view, MotionEvent event) {
    int action = event.getAction();
    switch (action) {
        case MotionEvent.ACTION_DOWN:
            isAutoPlay = false;
            break;
        case MotionEvent.ACTION_UP:
            isAutoPlay = true;
            break;
    }
    view.performClick();
    return false;
}

```

监听器实现

author: 金惠童

图 3-3-12 监听器实现

### ➤ 获取系统当前时间

这个主要是用来记录交易时间的，还有书籍的上架时间，通过这些时间我们可以计算出相关的信息，比如上架多久了，订单花了多久。如图 3-3-13 所示。

```

SimpleDateFormat simpleDateFormat = new SimpleDateFormat( pattern: "yyyy-MM-dd");
Date date = new Date(System.currentTimeMillis());
final String now = simpleDateFormat.format(date); // 获取当前日期

```

图 3-3-13 获取系统当前时间

### ➤ 添加新的书籍信息

我们首先需要获取前端用户填写的信息，然后构造 mysql 连接器，连接远程数据库，使用 mysql 将数据写入 book 表，如图 3-3-14 所示。

```

new Thread(new Runnable() {
    @Override
    public void run() {
        Connection connection = DBConnectHelper.getConnection();
        String sql = "INSERT INTO BOOK(BNAME, OWNER, PRICE, AUTHOR, PRESS, IMAGE, LAUNCHDATE, SOLD, CLASS, INTRO)"
            + "VALUES('"+book_name+"', '"+phone+"', '"+book_price+"', '"+book_author+"', '"+book_punish+"', '"+
            +imageString+"', '"+now+"', 0, '"+book_class+"', '"+book_intro+"')";
        Statement statement;
        try {
            statement = connection.createStatement();
            statement.executeUpdate(sql);
            Message msg = new Message();
            msg.what = INSERT_FINISH;
            handler.sendMessage(msg);
            statement.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}).start();

```

mysql语句

author: 金惠童

图 3-3-14 添加新的书籍信息

### ➤ 从数据库中读取书籍信息

同样我们使用 mysql 语句从远程数据库中筛选出需要的信息,然后将这些信息展示出来,这里以购物车为例,如图 3-3-15 所示。

```
String sql = "SELECT shop.BID, shop.IMAGE, shop.BNAME, shop.PRICE, shop.AUTHOR, shop.PRESS FROM BOOK as shop, CART as cart WHERE cart.PHONE='"+phone
Statement st;
List<BookShop> bookShops = new ArrayList<>();
try{
    st = connection.createStatement();
    ResultSet rs = st.executeQuery(sql);
    while (rs.next()) {
        // 读取数据
        int BID = rs.getInt( columnLabel: "BID");
        byte[] imageBytes = Base64.decode(rs.getString( columnLabel: "IMAGE"), Base64.DEFAULT);
        Bitmap book_image = BitmapFactory.decodeByteArray(imageBytes, offset: 0, imageBytes.length);
        String book_name = rs.getString( columnLabel: "BNAME");
        String book_author = rs.getString( columnLabel: "AUTHOR");
        String book_press = rs.getString( columnLabel: "PRESS");
        Double book_price = rs.getDouble( columnLabel: "PRICE");

        BookShop book = new BookShop(BID, book_name, book_image, book_author, book_press, book_price);
        bookShops.add(book);
    }
}
```

mysql语句

遍历结果

author: 金惠童

添加数据

图 3-3-15 加载书籍信息

➤ 修改数据库数据

修改数据用到 update 句式,其原理也跟查询和插入一样的,我们这里以修改密码为例,获取新的密码之后,将对应账号地方的密码修改,如图 3-3-16 所示。

```
private void updateUser() {
    new Thread((Runnable) () -> {
        Connection connection = DBConnectHelper.getConnection();
        String sql = "UPDATE USER SET PASSKEY = '"+password_1+"' WHERE PHONE = '"+phone+"'";
        Statement statement;
        try {
            statement = connection.createStatement();
            statement.executeUpdate(sql);
            Message msg = new Message();
            msg.what = UPDATE_FINISH;
            handler.sendMessage(msg);
            statement.close();
        } catch (SQLException e) {
        }
    })
}
```

author: 金惠童

图 3-3-16 修改数据库数据

➤ 前端：设置沉静式标题栏

如图 3-3-17 所示。

```
/**
 * 设置沉浸式标题
 */
QMUStatusBarHelper.translucent(getActivity());
topbar = (QMUITopBarLayout)mview.findViewById(R.id.topbar);
topbar.setTitle("0BT旧书交易平台").setTextColor(ContextCompat.getColor(getActivity(), R.color.qmui_config_color_white));
topbar.setBackgroundColor(ContextCompat.getColor(getActivity(), R.color.qmui_btn_blue_bg));
```

图 3-3-17 设置沉浸式标题栏

➤ 前端：加载书籍 recyclerView 列表

如图 3-3-18 所示。





图 3-3-18 加载 recyclerView 列表

➤ 前端：设置底部导航栏

如图 3-3-19 所示。



图 3-3-19 设置底部导航栏

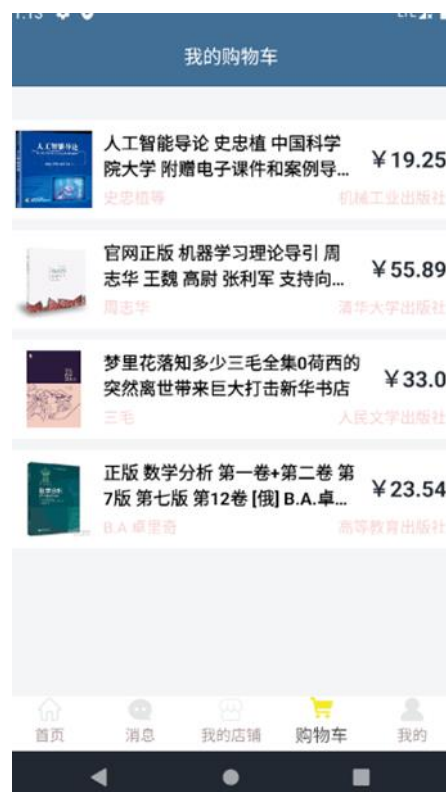
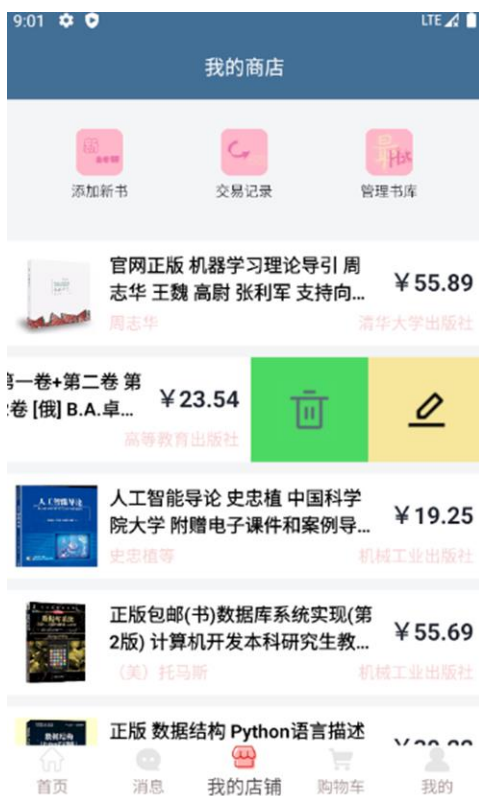
## 4. 项目效果图

➤ 登录注册



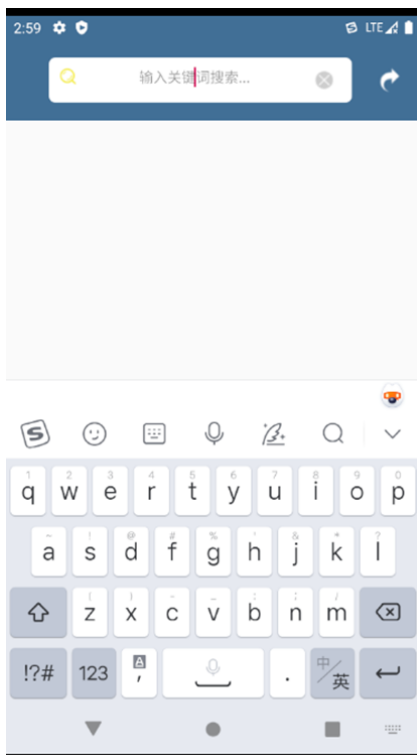
➤ MainActivity







## ➤ 其他界面



5. 任务分工及所完成的状况。评分标准

本项目全程由个人独立完成（评分分配 100%），其中包括前期市场调研，后期的前后端代码实现，以及 presentation，实验报告的完成等，所写代码目前已达 2 万行左右，未来还将继续优化，将项目开源到 GitHub 上，项目计划书如图 4-5-1 所示，里面包含一些工作计划和完成情况。

任务进展	编	任务（阶段）	具体工作（子任务）	计划开始/完成时间		负责人	进度	备注
	1	UI设计 + 与数据库关联	学习安卓开发四大组件的使用	2021/3/25	2021/3/28	金惠童	100%	
			学习使用Google比较高级的UI控件，使得App界面更为美观	2021/4/3	2021/4/7	金惠童	100%	参考GitHub和CSDN
			学习如何实现安卓App和数据库的关联，并实现demo	2021/4/8	2021/4/12	金惠童	100%	实现小demo
			学习socket通信与服务器的基本应用	2021/4/13	2021/4/15	金惠童	100%	
	2	完成数据库设计 + 实现登录注册功能	设计账号密码登陆库，并设计登录注册逻辑	2021/4/16	2021/4/20	金惠童	100%	
			设计书籍管理库、0点管理库、交易管理库	2021/4/21	2021/4/25	金惠童	100%	
			设计评价管理库、个人信息管理库	2021/4/25	2021/4/30	金惠童	100%	
	3	完成App内部功能	实现上架、下架、交易功能	2021/5/1	2021/5/15	金惠童	1000%	
			实现搜索功能	2021/5/16	2021/5/25	金惠童	100%	
			实现评论+社区论坛功能	2021/5/26	2021/5/31	金惠童	0%	

图 4-5-1 项目计划书

五、 问题分析

1. 安卓直连 mysql

通过网上查询资料，最终的敲定方案是使用 jdbc 连接 mysql，连接代码如图 5-1-1 所示。

```

public class DBConnectHelper {
    private static String driver = "com.mysql.jdbc.Driver";
    // 加入utf-8编码, 避免中文乱码
    private static String url = "jdbc:mysql://10.0.2.2:3306/mysql2021?characterEncoding=utf-8" + "&serverTimezone=PDT";
    //private static String url = "jdbc:mysql://172.20.10.2:3306/mysql2021?characterEncoding=utf-8" + "&serverTimezone=PDT";
    private static String user = "root"; // 用户名
    private static String password = "030889pioneer"; // 密码

}

* 连接数据库
**/
public static Connection getConn() {
    Connection conn = null;
    try {
        Class.forName(driver);
        conn = (Connection) DriverManager.getConnection(url, user, password);
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    if (conn == null)
        System.out.println("conn is null!!!!!!");
    return conn;
}

```

author: 金惠童

图 5-1-1 安卓使用 jdbc 直连 mysql

但是在实现过程中遇到一些问题，这里呈现问题和解决方案。

#### A . 重写了需要重写的四个方法之后，报错 method call expected.

这个问题找了很久答案，一开始以为是代码写错了，后来发现是版本的问题，参考链接 <https://stackoverflow.com/questions/16418193/java-method-call-expected>，由于 jdbc 高版本不太稳定，容易出错，因此将其换成低版本的就可以了。

#### B . 配置好 jdbc 后安卓虚拟机连不上 mysql

网上有很多资料显示，本地连接 mysql 的时候，只需要输入默认的地址 127.0.0.1 或者 localhost 就可以了（真的是误人子弟），按照教程做了之后一直报错，检查了网络也没有问题，mysql 权限也没有问题。

最后上了外网找，也找了很久，最终在一个论坛上看到，评论区有个人说虚拟机的地址比较特殊，默认 10.0.2.2，将信将疑试了一下，可以了。最后的代码如图 5-1-2 所示。

```

private static String driver = "com.mysql.jdbc.Driver";
// 加入utf-8编码, 避免中文乱码
private static String url = "jdbc:mysql://10.0.2.2:3306/mysql2021?characterEncoding=utf-8" + "&serverTimezone=PDT";

```

图 5-1-2 虚拟机连接 mysql 地址

#### C . 关于真实测试机无法远程连接 mysql 的问题

虽然虚拟机连上了，但在测试过程中实际的手机仍然无法连接 mysql，这又是什么原因呢？检查网络、权限、主机地址都没有问题，然后又是一波疯狂找资料，终于有位博主这样写道，如图 5-1-3 所示。

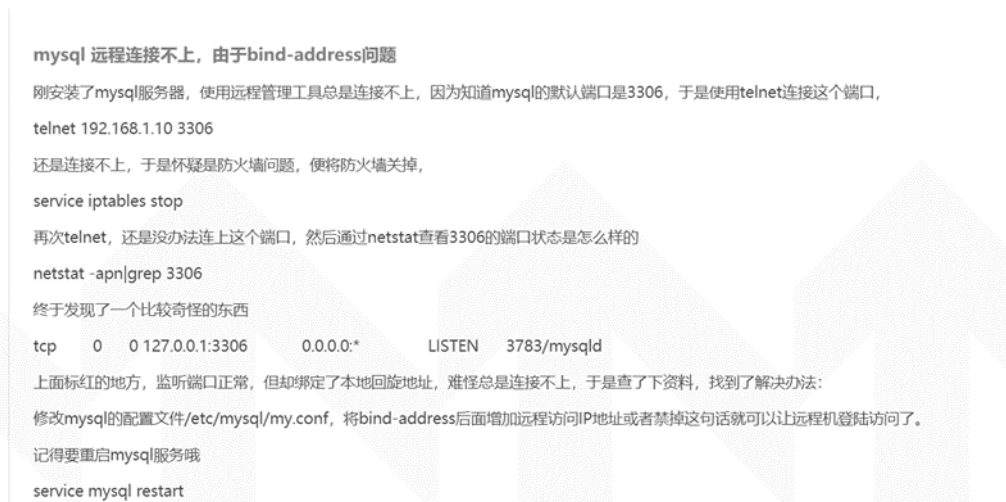


图 5-1-3 博文截图

看着挺有道理的, 试着做了一下, 这里要注意博主的文章是 Linux 系统的, 在 Windows 系统下, 该配置文件的位置在如图 5-1-4 所示路径下。

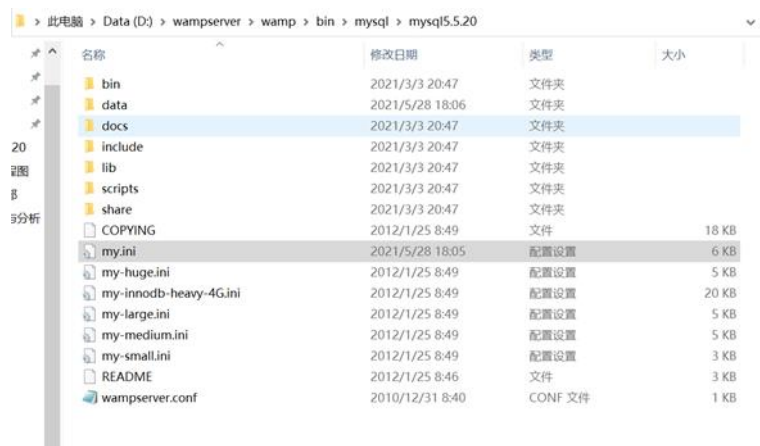


图 5-1-4 配置文件路径

按照教程修改之后, mysql 终于连接成功了!!! 如图 5-1-5 所示。

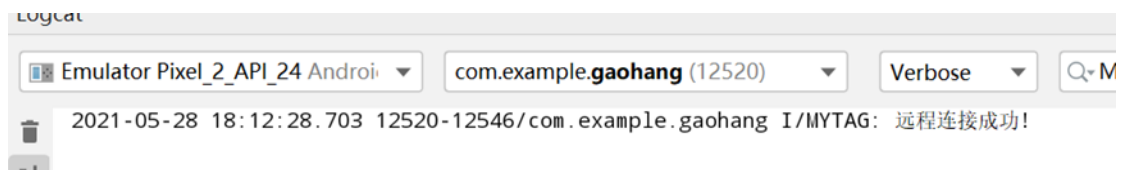


图 5-1-5 连接成功

## 2. 关于 mysql 存储图片的问题

在一般的 App 中, 难免遇到需要存储照片的问题, 而在 mysql 中又没有一种称为“照片”的数据结构, 那怎么办呢? 原来在安卓当中 ImageView 是可以转化为 Bitmap 编码的, 而 Bitmap 编码又可以转化成 byte 数组, 最后我们如果将 byte 数组转化成字符串, 就可以放进去 mysql 数据库了。参考博文:

[https://blog.csdn.net/weixin\\_44853386/article/details/110845106](https://blog.csdn.net/weixin_44853386/article/details/110845106)。

并根据我实际的项目内容，做了具体的代码实现，如图 5-2-1 所示是将图片转为 String 类型，然后放进 mysql 数据库。如图 5-2-2 所示是将图片从 mysql 中取出，并且解码成 Bitmap 编码格式。

```
// 将图片转换成Base64编码
try {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    Bitmap bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), uri);
    bitmap.compress(Bitmap.CompressFormat.JPEG, quality: 100, baos);
    byte[] imageBytes = baos.toByteArray();
    imageString = Base64.encodeToString(imageBytes, Base64.DEFAULT);
} catch (Exception e) {
    e.printStackTrace();
}
```

author: 金惠童

图 5-2-1 将图片转为 String 类型

```
// 如果存在获取头像
try {
    st = connection.createStatement();
    ResultSet rs = st.executeQuery(sql);
    if (rs.first()) {
        ResultSet imgRs = st.executeQuery(sql_2);
        while (imgRs.next()) {
            byte[] imageBytes = Base64.decode(imgRs.getString( columnName: "IMAGE"), Base64.DEFAULT);
            final Bitmap decodedImage = BitmapFactory.decodeByteArray(imageBytes, offset: 0, imageBytes.length);
            // 在主线程修改头像
            LoginActivity.this.runOnUiThread() → {
                account_icon.setImageBitmap(decodedImage);
            };
        }
    }
}
```

author: 金惠童

图 5-2-2 将 String 取出转成 bitmap 编码

这里还要注意，由于这里的编码太长，需要使用 blob 才能放的下。

## 六、 实验心得

### 1. 实验总结

本次实验采用项目形式的开发，从零到有，从数据库设计，到万丈高楼平地起，期间遇到了许多的困难，但也自学到了不少的东西，特别是对课程内容的理解和应用加深了不少。

本项目实现了基于深大市场的旧书交易系统，从根本上解决了深大学生买旧书难、卖旧书难等问题。本项目采取 jdbc 直连 mysql 的方式，可以很好地将 mysql



语句融合到安卓开发中，使得这个项目跟数据库的关联度进一步增强。

其次，为了更好的用户体验，在项目构建过程中，对 App 界面的设计参考了很多现有的 App，尽可能做出比较美观的界面。为了后续的开发，我们也预留了很多的 API 接口，后续可以作为毕业设计。

总的来说，这次大作业付出很多，但也收获很多，彭老师的教学方案灵活多变，自由选择的空间很大，这一点非常值得提倡。

## 2. 自我评价

对比上一次的项目计划表，发现有些功能不太一样，但是大体上实现了预期的计划。在时间上，本来计划 5 月 31 号结项的，现在看来得再多一个星期的时间。主要的时间耗费在安卓连接 mysql 问题上（受了网上很多误导）。其次比较耗时的就是 UI 设计了，为了设计好看一些的 UI，需要好好地学习安卓，好好找素材。

为了实现较为细致的功能，尽量与现行主流 App 的设计相仿，考虑和调试也画了较多的时间。整个项目（前端+后端）代码量达到一万五千行左右，所有的工作都是一个人完成（论队友的重要性），因此工作量可以说是普通同学的几倍吧，但是学到的也挺多的。

如果对这次进行自我评价的话，根据上面的叙述，我会对自己的工作评价为优秀。我对我的项目充满信心，对自己付出的巨大工作量表示肯定。（嘻嘻~）

## 七、 参考文献

[1]《第一行代码》 郭霖

[2] Socket 通信 <https://blog.csdn.net/shankezh/article/details/70763579>.

[3] Android Socket 通信, <https://www.jianshu.com/p/c27c4e9b2211>.

[4] Android 远程连接 mysql

<https://blog.csdn.net/csdnzouqi/article/details/87382143>.

[5] Android 连接 mysql, 实现增删查改:

[https://blog.csdn.net/Tobey\\_r1/article/details/95162790?utm\\_medium=distribute.pc\\_relevant\\_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.control&dist\\_request\\_id=1328740.12973.16168357635733493&depth\\_1-utm\\_source=distribute.pc\\_relevant\\_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.control](https://blog.csdn.net/Tobey_r1/article/details/95162790?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.control&dist_request_id=1328740.12973.16168357635733493&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.control).

[6] Apache 服务器下载 :

[https://blog.csdn.net/vagabond\\_/article/details/90436858](https://blog.csdn.net/vagabond_/article/details/90436858).

[7] 基于 Android 校园跳蚤市场交易系统的设计与实现:

[https://blog.csdn.net/m0\\_37655297/article/details/83184323](https://blog.csdn.net/m0_37655297/article/details/83184323).

[8] Android 实现社交应用中的评论和回复功能:

<https://www.jianshu.com/p/eda8d09c9d7a>.

[9] Android App 第三方开源组件汇总:

<https://zhuanlan.zhihu.com/p/47122534>.

[10] Android 开发后台数据接口交互实现简单的登录案例:

[https://blog.csdn.net/qq\\_34317125/article/details/80533685](https://blog.csdn.net/qq_34317125/article/details/80533685).



## 八、 诚信承诺

个人总分权重分配表

(1 人总分为 100%， 2 人为 190%， 3 人 270%， 4 人 350%)

排序	姓名	学号	项目个人权重
1	金惠童	2019151067	100 %
2			%
3			%
4			%

我组成员总共\_1\_名， 权重总和为： 100 %

本组成员郑重承诺在 Database Software Design and Implementation 项目实施的过程中不发生任何不诚信现象，一切不诚信所导致的后果均由本组成员承担。同时我组成员同意此项目个人项目得分按照教师评分乘以“个人总分权重分配表”的个人权重获得。

签名（手签，并拍照或者扫描附于此）：

