

## 实验 1：开发 Windows 窗体应用程序

学号	1183710106	实验时间	2019 年 7 月 10 日																																										
姓名	王翰坤																																												
程序名称	基于 .Net Standard 2.0 和 Xamarin 的移动平台图书管理系统																																												
平时(10 分)		程序(60 分)		报告(30 分)	总分																																								
1. 程序功能描述 (200 字以上) (10 分)					得分																																								
<p>注：由于我的电脑的系统为 macOS，所以只能使用 VS for mac 开发。而 VS for mac 不支持 .Net Framework 的 Windows 开发，所以最终选择了利用 Xamarin 框架搭建 Android 应用，与实验要求略有出入，望老师谅解！</p> <p>程序利用 Xamarin 框架和 SQLite 数据库，实现了一个 Android 平台的较完整的图书管理系统。其主要内容包括：</p> <p>1) 用户的注册和登录。打开 APP，首先进入用户注册/登录界面。用户注册时输入用户名和密码，系统自动分配用户 ID。登录系统时，输入用户 ID 和相应的密码即可。登陆成功时，系统会友善地提示“您太强了！”。</p> <p>2) 图书信息的浏览和搜索。进入系统主页后，用户可浏览系统中所有的图书信息（按添加时间先后排列）。主页上方有搜索栏，用户输入关键字，系统实时给出搜索结果（搜索域为书名和作者）。</p> <p>3) 图书信息的添加、修改和删除。用户可向系统中添加新图书（需输入书名、作者、描述等信息，系统自动分配图书 ID、设定“在馆”状态）。对已添加的图书，用户可以修改“描述”信息。若某图书未被借出，用户还可以删除该图书的信息。</p> <p>4) 图书的借阅和归还。用户可借阅系统中状态为“在馆”的图书（借阅后，状态会自动更改为“借出”）。对于已借阅的图书，用户可以归还。</p> <p>5) 借阅记录的分类浏览。用户可浏览自己所有的借阅记录，并可选择只看“未还”的记录。</p>																																													
2. 数据库设计说明 (5 分)					得分																																								
<p>1) UserInfo (用户信息表)</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>字段名</th> <th>数据类型</th> <th>精度</th> <th>字段含义</th> <th>备注</th> </tr> </thead> <tbody> <tr> <td>ID</td> <td>int</td> <td>/</td> <td>用户 ID</td> <td>第一关键字，自动增加</td> </tr> <tr> <td>Name</td> <td>string</td> <td>/</td> <td>用户名</td> <td></td> </tr> <tr> <td>Password</td> <td>string</td> <td>/</td> <td>密码</td> <td></td> </tr> <tr> <td>strID</td> <td>string</td> <td>/</td> <td>ID 转字符串</td> <td>方便显示</td> </tr> </tbody> </table> <p>2) BookInfo (图书信息表)</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>字段名</th> <th>数据类型</th> <th>精度</th> <th>字段含义</th> <th>备注</th> </tr> </thead> <tbody> <tr> <td>ID</td> <td>int</td> <td>/</td> <td>图书 ID</td> <td>第一关键字，自动增加</td> </tr> <tr> <td>Name</td> <td>string</td> <td>/</td> <td>书名</td> <td></td> </tr> </tbody> </table>						字段名	数据类型	精度	字段含义	备注	ID	int	/	用户 ID	第一关键字，自动增加	Name	string	/	用户名		Password	string	/	密码		strID	string	/	ID 转字符串	方便显示	字段名	数据类型	精度	字段含义	备注	ID	int	/	图书 ID	第一关键字，自动增加	Name	string	/	书名	
字段名	数据类型	精度	字段含义	备注																																									
ID	int	/	用户 ID	第一关键字，自动增加																																									
Name	string	/	用户名																																										
Password	string	/	密码																																										
strID	string	/	ID 转字符串	方便显示																																									
字段名	数据类型	精度	字段含义	备注																																									
ID	int	/	图书 ID	第一关键字，自动增加																																									
Name	string	/	书名																																										

Author	string	/	作者	
Description	string	/	描述	可修改
Lent	bool	/	是否被借出	
strLent	string	/	借出状态（文字）	“在馆”、“借出”

### 3) RecordInfo（记录信息表）

字段名	数据类型	精度	字段含义	备注
ID	int	/	记录 ID	第一关键字，自动增加
BookID	int	/	被借图书 ID	
UserID	int	/	借阅用户 ID	
Returned	int	/	是否已归还	
strReturned	string	/	归还状态（文字）	“已还”、“未还”
BookName	string	/	书名	
Date	DateTime	/	借阅时间	自动记录

### 3. 程序设计说明（要有必要的图和文字说明）（10 分）

得分

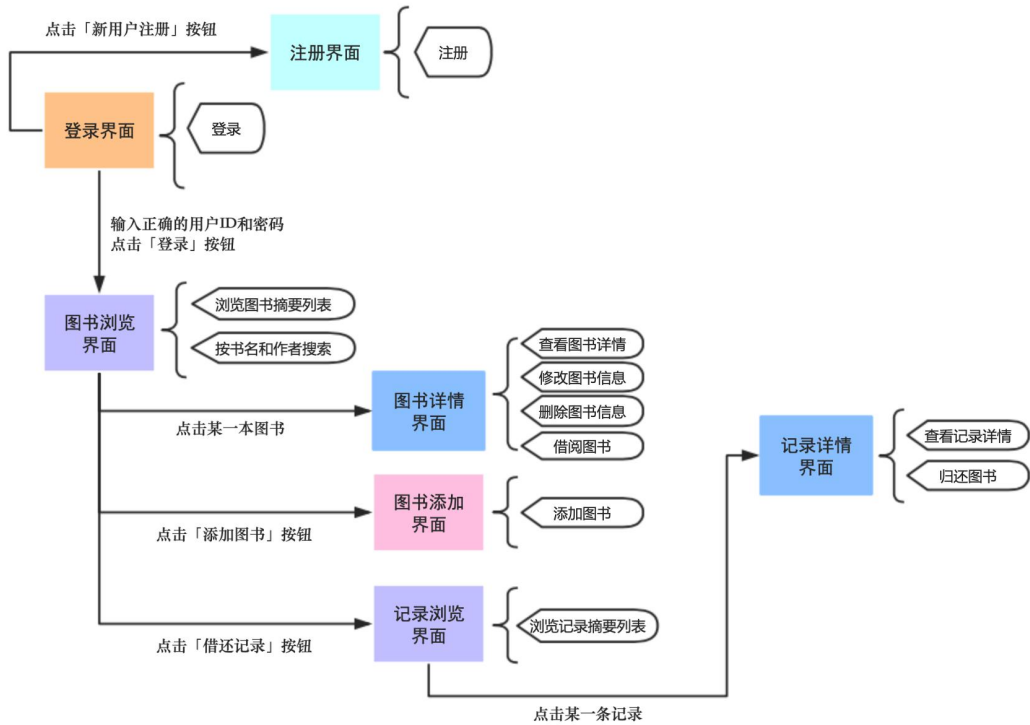
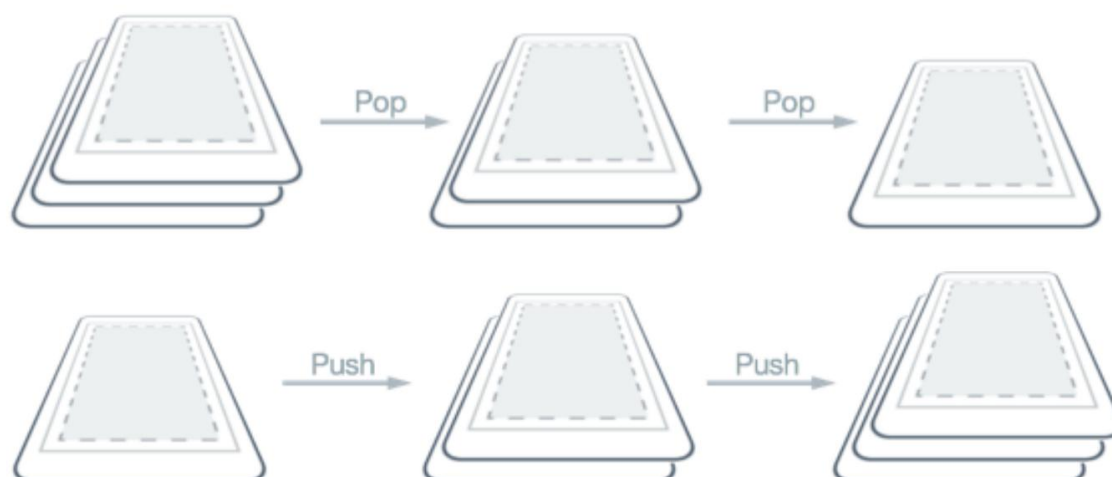


图 1 APP 界面结构与功能图

程序设计流程图如图 1 所示。几个重要的代码编写要点如下：

1) 程序采用了 Xamarin.Forms 中的 NavigationPage 导航页设计。图中“登录界面 (MainPage)”为程序根页面，其他页面以栈结构依次堆叠在根页面上方，返

回时后叠先出。压入界面时，传入 `BindingContext` 作为联系的对象



2) 采用了异步编程的方法 (`await/async`)。异步编程方法在移动应用程序开发中十分流行。对于长时间运行的任务 (如下载数据等)，使用异步方法有助于保持用户界面响应的持续性，避免 UI 总是等待当前任务完成而停止相应用户输入。

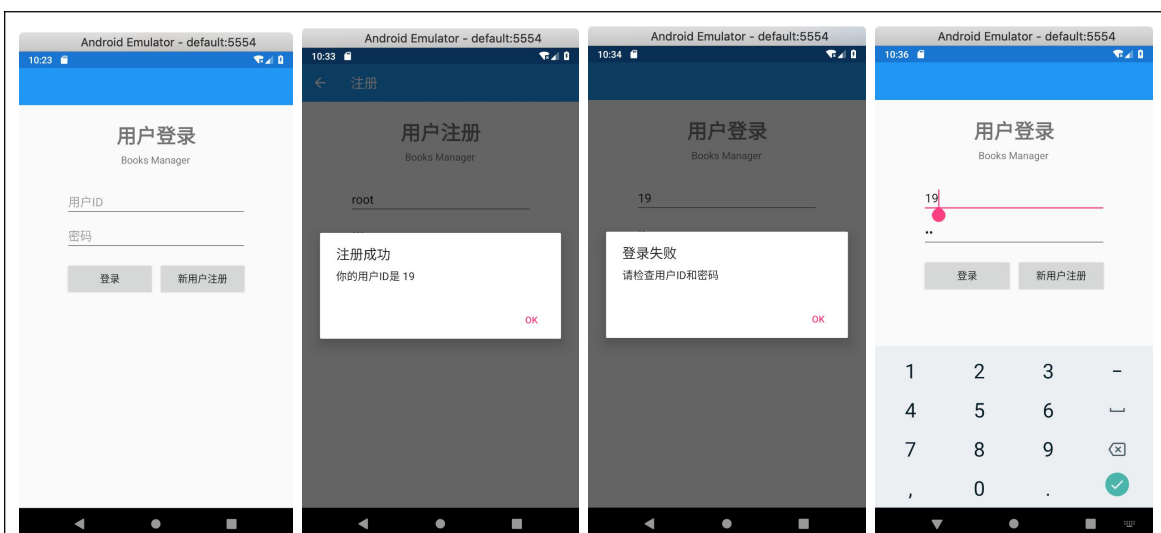
3) 采用了 SQLite 作为数据库管理系统。由于 Xamarin 对 MySQL 等主流数据库系统不友好，故使用 SQLite。它是一个轻型数据库系统，实现了自给自足的、无服务器的、零配置的、事务性的 SQL 数据库引擎。SQLite 引擎不是一个独立的进程，它直接访问其存储文件。程序灵活运用了许多 SQLite 的 C# 异步接口，如 `InsertAsync` (插入)、`UpdateAsync` (更新)、`DeleteAsync` (删除)、`.Where` 成员 (`Where` 子句)、`ExecuteScalarAsync` (从 `count`、`max` 等函数中获取常量) 和 `QueryAsync` (用 SQL 语句查找) 等，效率高、可读性强。

4) 程序采用 `CodeBehind` 类创建窗体前端 (即 XAML)。XAML 的编写综合运用了 `StackLayout`、`Grid`、`ListView` 等布局方式和 `Label`、`Entry`、`Button` 等组件，使得 UI 简洁高效。

5) 程序文件安排。根目录下有 `Data` 和 `Models` 两个文件夹以及若干 XAML 组 (每个组包含一个 `.xaml` 文件和一个同名 `.cs` 文件)。每个名字以 `Page` 结尾的组对应一个界面；`App.xaml` 组为默认启动文件。`Data` 文件夹中包含 `BooksDatabase.cs` 文件，它是数据库和数据库操作函数的类。`Models` 文件夹中则包含 `UserInfo.cs`、`BookInfo.cs` 和 `RecordInfo.cs` 三个文件，分别对应于 `UserInfo` 类、`BookInfo` 类和 `RecordInfo` 类。

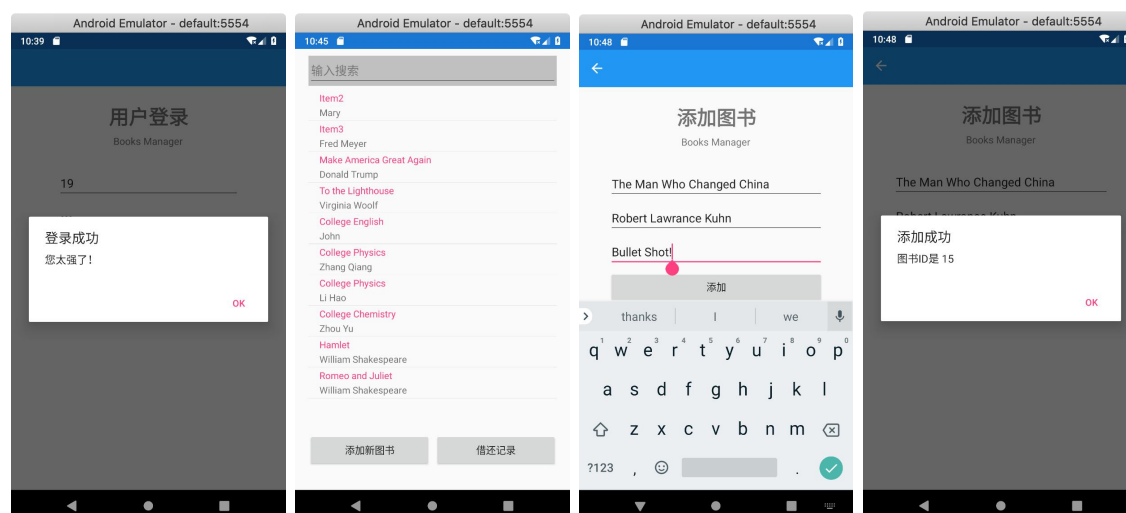
4. 程序运行流程及截图 (要有必要的文字说明) (5 分)

得分

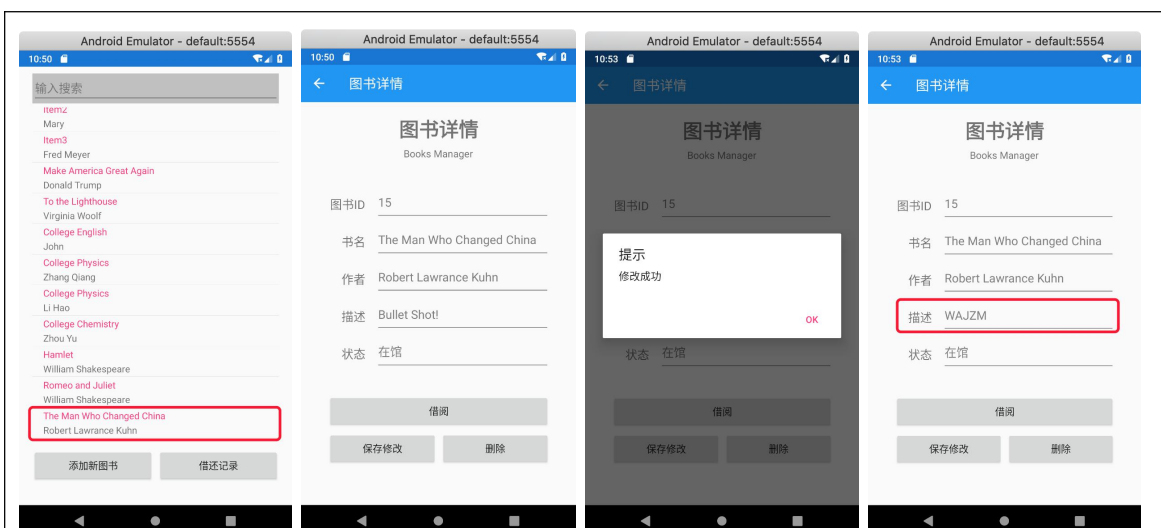


本节中的截图序号以“截图 X-Y”形式表达，其中 X 为行序号，Y 为列序号。

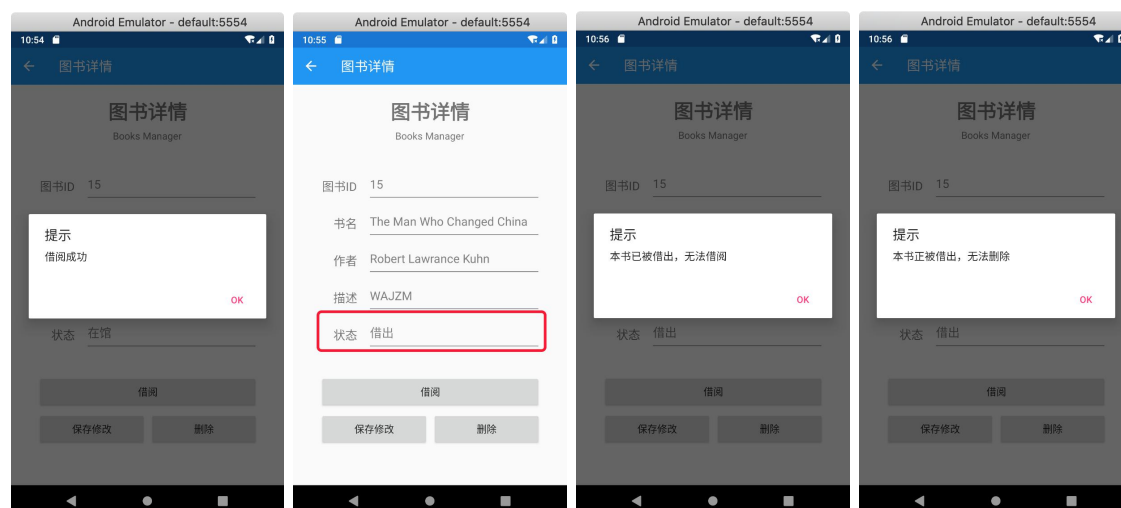
打开 BooksManager APP，进入登录主界面（截图 1-1）。点击“新用户注册”，输入用户名和密码，系统提示注册成功并自动分配了用户 ID（截图 1-2）。系统回到用户登录界面，输入用户名和相应的密码。输入错误时，会提示登录失败（截图 1-3）。另外，为避免出现错误，输入用户 ID 时仅弹出虚拟数字键盘；为保证账户安全，密码不明文显示，仅以小黑点展示（截图 1-4）。



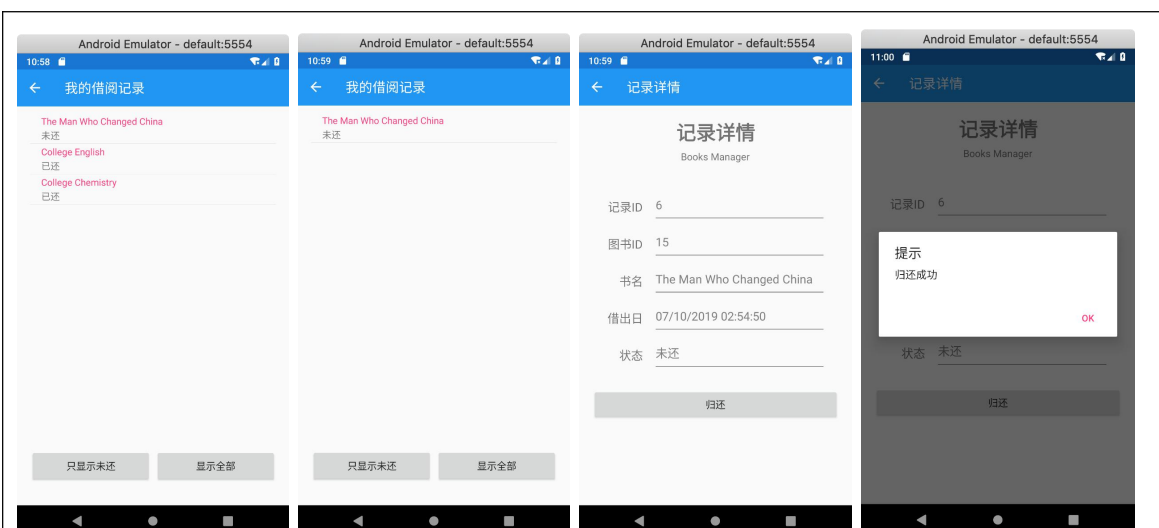
二者输入正确时，会提示登录成功（截图 2-1）并自动进入图书浏览界面（截图 2-2）。点击“添加新图书”按钮，在新页面中输入图书信息（截图 2-3）并点击“添加”，系统提示添加成功并分配了图书 ID（截图 2-4）。



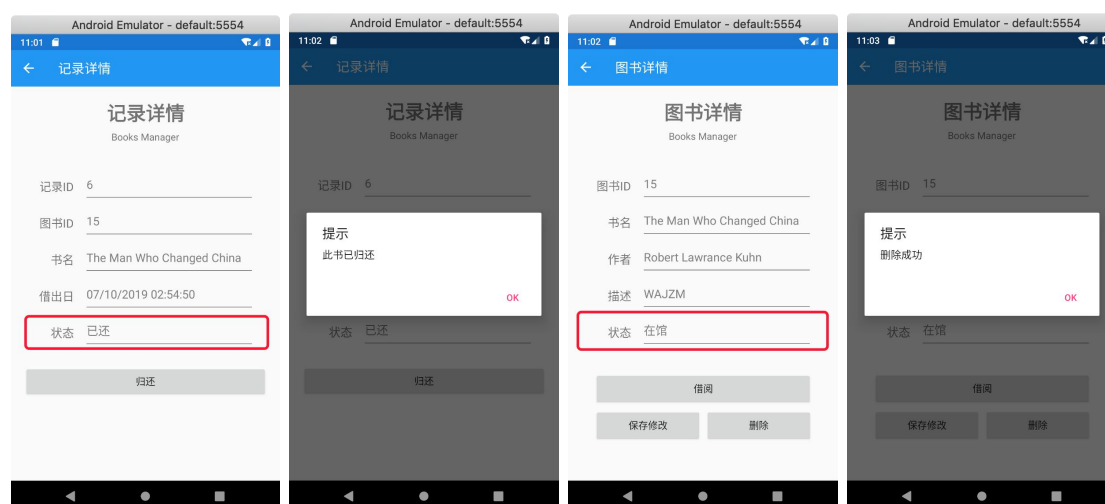
系统自动返回图书浏览界面，可以看到刚刚添加的图书出现在了列表中（截图 3-1）。点击这本书，可以查看图书详情——系统给出的信息与刚刚输入的信息完全一致（截图 3-2）。尝试修改图书描述为“WAJZM”，点击“保存修改”，可以发现系统提示修改成功（截图 3-4、截图 3-4）。



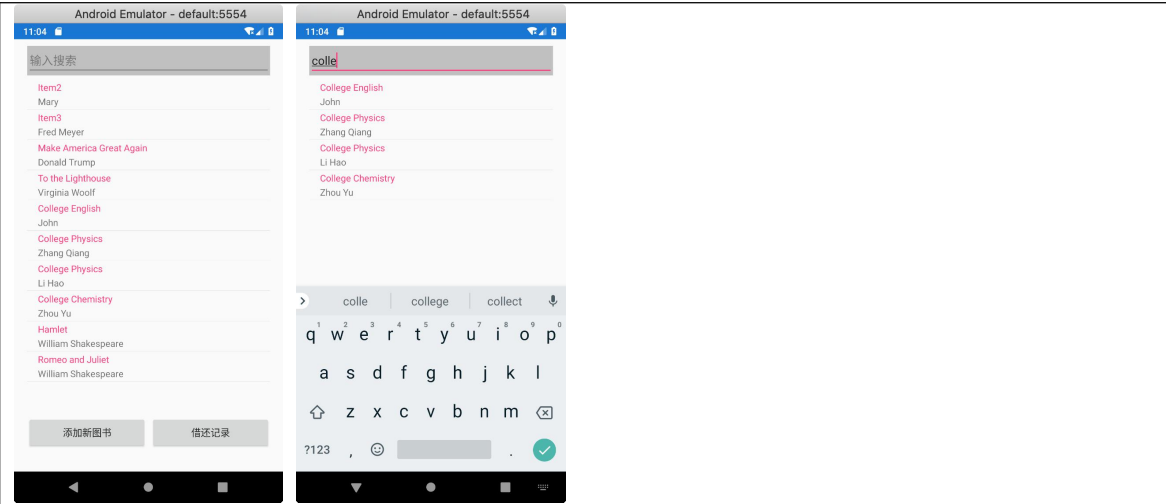
点击“借阅”，系统提示借阅成功（截图 4-1），图书状态变为“借出”（截图 4-2）。若再次点击“借阅”，系统会提示无法借阅（截图 4-3）。若点击“删除”，系统提示图书正被借出，无法删除（截图 4-4）。



回到图书浏览界面，点击“借还记录”，可以查看当前用户的借阅记录（截图 5-1）。为了方便演示，这里另添加了两条借阅记录，状态为“已还”。点击“只显示未还”，列表中只剩下了刚刚借出的那一条记录（截图 5-2）。点击这一条记录，可以查看记录详情，包括图书 ID、书名、借出时间等信息（截图 5-3）。点击“归还”按钮，即可归还对应的图书（截图 5-4）。



归还后，记录状态显示为“已还”（截图 6-1）。若再点击“已还”，系统会提示此书已经归还过了（截图 6-2）。回到图书详情界面，再查看这本书的详细信息，发现状态已经变回了“在馆”。在馆状态下点击“删除”按钮，系统提示删除成功。



回到图书浏览界面，原先的那条图书信息已经消失（截图 7-1）。系统还支持图书搜索。在搜索栏中输入关键字，系统会即时给出对书名和作者子段匹配的结果（截图 7-2）。

教师评语