

附件 1:

学 号	201725040402
成 绩	

移动终端开发技术 课程设计报告

题 目	书架空间
班 级	软件技术四班
学 号	201725040402
姓 名	董永顺
小组成员	张庆旭 赵鹏宇 杨吉钰 董永顺
指导教师	陈媛媛

2018 年 6 月 29 日

目 录

1 引言.....	1
1.1 研究背景意义.....	1
1.2 开发运行环境.....	1
1.3 相关技术简介.....	2
1.3.1 MY SQL.....	2
1.3.2 Listview.....	3
1.3.3 Sharedpreferences	3
1.3.4 Button.....	4
1.3.5 自定义 view.....	4
1.3.6 MD5 加密算法	4
1.3.7 Intent 显示意图	5
1.3.8 Fromage	5
2 系统分析.....	7
2.1 系统的整体架构.....	7
2.2 系统的功能划分.....	8
2.3 传统结构与 SOA 架构的分析.....	8
3 部分源码.....	9
3.1 MD5 密码加密算法	9
3.2 碎片 INDEX 框架	10
3.3 闪动 log	13
4 结果展示.....	14
5 总结.....	16

1 引言

1.1 研究背景意义

现在的学习软件和平台是越来越多，但是总体上学习软件和系统都有各自的优缺点，还是有的资料搜集的不够，学生们应用起来不够自由，很少有学习软件会提供讨论的功能让学生们对学习进行讨论对我们的学习资源进行评价，这对我们提供相关的学习资料方面有了一个方先，使得我们更能发现学生所需要的资源。

我们的这个软件对学生的帮助有很大的帮助，不仅提供视频教学，还有观看历史的保存，观看历史保存这个功能会使我们在学习中知道我们学习了那些内容，对我们的复习有很大的帮助，不会盲目的从头开始以至于一直在复习前几章的知识点。最重要的是我们提供资源讨论区这个功能，资源讨论区这个功能会使我们不断的完善资源的需求，让同学们在这个讨论区进行自由的讨论学习。对学生的教育有重大的意义。

1.2 开发运行环境

Android Studio 是谷歌推出一个 Android 集成开发工具，基于 IntelliJ IDEA. 类似 Eclipse ADT, Android Studio 提供了集成的 Android 开发工具用于开发和调试。Android Studio 是谷歌推出了新的 Android 开发环境，开发者可以在编写程序的同时看到自己的应用在不同尺寸屏幕中的样子。

在 IDEA 的基础上，Android Studio 提供：

- 基于 Gradle 的构建支持
- Android 专属的重构和快速修复
- 提示工具以捕获性能、可用性、版本兼容性问题
- 支持 ProGuard 和应用签名
- 基于模板的向导来生成常用的 Android 应用设计和组件
- 功能强大的布局编辑器，可以让你拖拉 UI 控件并进行效果预览

1.3 相关技术简介

1.3.1 MY SQL

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System, 关系数据库管理系统) 应用软件。

MySQL 是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

1.3.2 Listview

ListView 控件可使用四种不同视图显示项目。通过此控件，可将项目组成带有或不带有列标头的列，并显示伴随的图标和文本。可使用 ListView 控件将称作 ListItem 对象的列表条目组织成下列四种不同的视图之一：1. 大(标准)图标 2. 小图标 3. 列表 4. 报表 View 属性决定在列表中控件使用何种视图显示项目。还可用 LabelWrap 属性控制列表中与项目关联的标签是否可换行显示。另外，还可管理列表中项目的排序方法和选定项目的外观。

1.3.3 Sharedpreferences

SharedPreferences 是 Android 平台上一个轻量级的存储类，用来保存应用的一些常用配置，比如 Activity 状态，Activity 暂停时，将此 activity 的状态保存到 SharedPereferences 中；当 Activity 重载，系统回调方法 onSaveInstanceState 时，再从 SharedPreferences 中将值取出。

SharedPreferences 提供了 java 常规的 Long、Int、String 等类型数据的保存接口。SharedPreferences 类似过去 Windows 系统上的 ini 配置文件，但是它分为多种权限，可以全局共享访问。

提示最终是以 xml 方式来保存，整体效率来看不是特别的高，对于常规的轻量级而言比 SQLite 要好不少，如果真的存储量不大可以考虑自己定义文件格式。xml 处理时 Dalvik 会通过自带底层的本地 XML Parser 解析，比如 XMLpull 方式，这样对于内存资源占用比较好。

1.3.4 Button

对话框中的一种图形元素。当激活它时，能执行指定的功能。用户可以采用以下方法来激活按钮：用鼠标在其上单击，或者当按钮获得聚焦时按下“Return”或“Enter”键。edition

1.3.5 自定义 view

自定义 View 是安卓内置的自定义控件，可以按照自己的需求，应用所需要的格式等进行自定义功能

1.3.6 MD5 加密算法

MD5 的全称是 Message-Digest Algorithm 5（信息-摘要算法），在 90 年代初由 MIT Laboratory for Computer Science 和 RSA Data Security Inc 的 Ronald L. Rivest 开发出来，经 MD2、MD3 和 MD4 发展而来。作用是让大容量信息在用数字签名软件签署私人密匙前被“压缩”成一种保密的格式（就是把一个任意长度的字节串变换成一定长的大整数）。不管是 MD2、MD4 还是 MD5，它们都需要获得一个随机长度的信息并产生一个 128 位的信息摘要。虽然这些算法的结构或多或少有些相似，但 MD2 的设计与 MD4 和 MD5 完全不同，那是因为 MD2 是为 8 位机器做过设计优化的，而 MD4 和 MD5 却是面向 32 位的电脑。这三个算法的描述和 C 语言源代码在 Internet RFCs 1321 中有详细的描述，这是一份最权威的文档，由 Ronald L. Rivest 在 1992 年 8 月向 IETF 提交。

.1.3.7 Intent 显示意图

Intent（意图）主要是解决 Android 应用的各项组件之间的通讯。

Intent 负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述，Android 则根据此 Intent 的描述，负责找到对应的组件，将 Intent 传递给被调用的组件，并完成组件的调用。

因此，Intent 在这里起着一个媒体中介的作用，专门提供组件互相调用的相关信息，实现调用者与被调用者之间的解耦。

例如，在一个联系人维护的应用中，当我们在一个联系人列表屏幕(假设对应的 Activity 为 listActivity)上，点击某个联系人后，希望能够跳出此联系人的详细信息屏幕(假设对应的 Activity 为 detailActivity)

为了实现这个目的，listActivity 需要构造一个 Intent，这个 Intent 用于告诉系统，我们要做“查看”动作，此动作对应的查看对象是“某联系人”，然后调用 startActivity (Intent intent)，将构造的 Intent 传入，系统会根据此 Intent 中的描述，到 Manifest 中找到满足此 Intent 要求的 Activity，系统会调用找到的 Activity，即为 detailActivity，最终传入 Intent，detailActivity 则会根据此 Intent 中的描述，执行相应的操作。

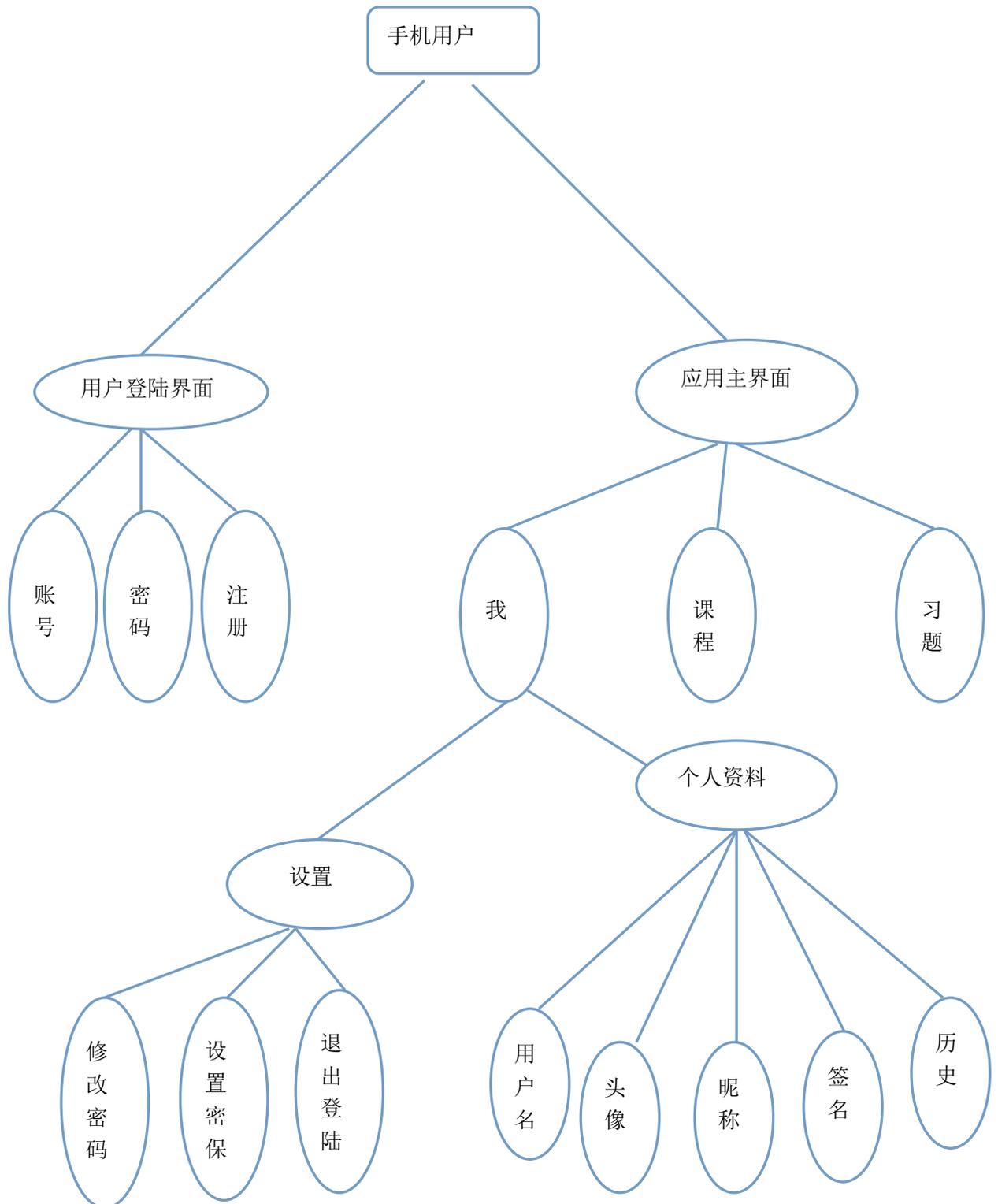
1.3.8 Fromage

因为 Activity 有它的局限性，同样的界面在手机上显示可能很好看，在平板上就未必了，因为平板的屏幕非常大，手机的界面放在

平板上可能会有过分被拉长、控件间距过大等情况。这个时候更好的体验效果是在 Activity 中嵌入“小 Activity”，然后每个“小 Activity”又可以拥有自己的布局这就是 Fragment 为了让界面可以在平板上更好地展示，Android 在 3.0 版本引入了 Fragment(碎片)功能，它非常类似于 Activity，可以像 Activity 一样包含布局。Fragment 通常是嵌套在 Activity 中使用的，现在想象这种场景：有两个 Fragment，Fragment 1 包含了一个 ListView，每行显示一本书的标题。Fragment 2 包含了 TextView 和 ImageView，来显示书的详细内容和图片。如果现在程序运行竖屏模式的平板或手机上，Fragment 1 可能嵌入在一个 Activity 中，而 Fragment 2 可能嵌入在另一个 Activity 中，而如果现在程序运行在横屏模式的平板上，两个 Fragment 就可以嵌入在同一个 Activity 中了由此可以看出，使用 Fragment 可以让我们更加充分地利用平板的屏幕空间

2 系统分析

2.1 系统的整体架构



2.2 系统的功能划分

具体分为三大部分——个人管理、资源管理和云端同步

个人管理：包括个人资料，观看历史个性设置，密码修改等模块

资源管理：分为多个部分：视频、课件、资料、软件等

云端同步：保存个人数据的云端，开辟个人云盘等

2.3 传统结构与 SOA 架构的分析

传统架构：

存在问题：1. 模块之间耦合度太高，其中一个升级其他都得升级

2. 开发困难，各个团队开发最后都要整合一起

3. 系统的扩展性差

4. 不能灵活的进行分布式部署

SOA: 分布式系统架构——SOA 服务体系架构

分布式架构：（SOA: 面向服务的体系结构，是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来）把系统按照模块拆分成多个子系统

优点：1. 把模块拆分，使用接口通信，降低模块之间的耦合度

2. 把项目拆分成若干个子项目，不同的团队负责不同的子项目

3. 增加功能时只需要在增加一个子项目，调用其它系统的接口就可以

4. 可以灵活的进行分布式部署

缺点： 系统之间交互需要使用远程通信，接口开发增加工作量

3 部分源码

3.1 MD5 密码加密算法

```
public class MD5Utils {

    /**
     * MD5 加密算法
     */

    public static String md5(String text) {
        MessageDigest digest = null;
        try {
            digest = MessageDigest.getInstance("md5");
            byte[] result = digest.digest(text.getBytes());
            StringBuilder sb = new StringBuilder();
            for (byte b : result) {
                int number = b & 0xff;
                String hex = Integer.toHexString(number);
                if (hex.length() == 1) {
                    sb.append("0" + hex);
                } else {
                    sb.append(hex);
                }
            }
            return sb.toString();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return "";
        }
    }
}
```

```
}
```

3.2 碎片 INDEX 框架

```
public class MainActivity extends AppCompatActivity {  
    private RadioButton course, exercises, myInfo;  
    private RelativeLayout title_abr, history;  
    private MyInfoView myInfoView;  
    @Override  
    protected void onCreate(Bundle savedInstanceState){  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        mBodyLayout = (FrameLayout) findViewById(R.id.fram);  
        initBottomBar();//获取底部按钮  
        setListener();//按钮监听  
        setInitStatus();  
    }  
  
    /*  
    获取底部按钮  
    */  
    private void initBottomBar() {  
        course = (RadioButton) findViewById(R.id.rb1);  
        exercises = (RadioButton) findViewById(R.id.rb2);  
        myInfo = (RadioButton) findViewById(R.id.rb3);  
    }  
  
    /*  
    按钮监听  
    */  
    private void setListener() {  
        initBottomBar();  
        course.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                replace(new index_1());  
            }  
        });  
    }  
}
```

```

    }
});
exercises.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        replace(new index_2());
    }
});
myInfo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        replace(new MyInfoView());
    }
});
}

private void setInitStatus() {
}

/*
    用户点击两次退出应用
*/

protected long exitTime; //记录第一次的点击时间
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getAction() ==
    KeyEvent.ACTION_DOWN) {
        if ((System.currentTimeMillis() - exitTime) > 2000) {
            Toast.makeText(MainActivity.this, "再按一次退出应用",
            Toast.LENGTH_SHORT).show();
            exitTime = System.currentTimeMillis();

```

```

        } else {
            MainActivity.this.finish();
        }
    }
    return true;
}

/*
    按钮添加碎片
*/
public void replace(Fragment fragment) {
    getSupportFragmentManager().beginTransaction().replace(R.id.fram,
fragment).commit();
}

/*
    获取 shared preferences 中的登陆状态
*/
private boolean readLoginStatus() {
    SharedPreferences sp = getSharedPreferences("loginInfo",
Context.MODE_PRIVATE);
    Boolean isLogin = sp.getBoolean("isLogin", false);
    return isLogin;
}

/*
    清除 shared preferences 中的登陆状态
*/
private void clearLoginStatus() {
    SharedPreferences sp = getSharedPreferences("loginInfo",
Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sp.edit();//获取编辑器
    editor.putBoolean("isLogin", false);
    editor.putString("loginUserName", "");
}

```

```
        editor.commit();
    }

}
```

3.3 闪动 log

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_flash_);
    flash = findViewById(R.id.flash);

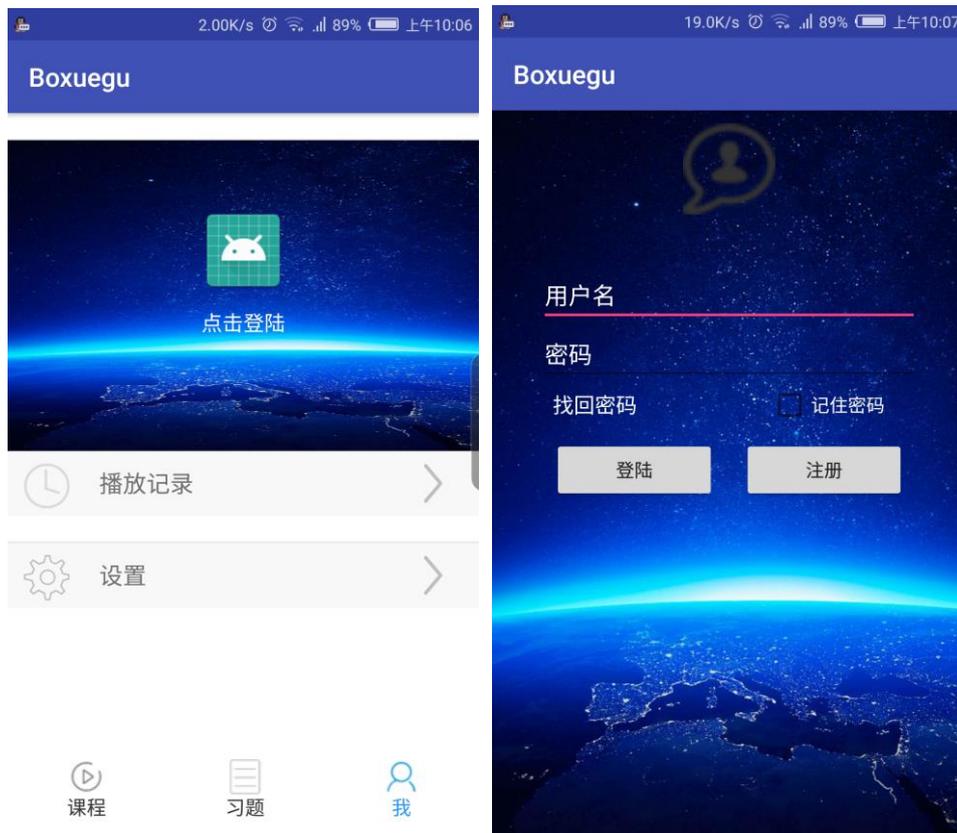
    new Timer().schedule(new TimerTask() {
        @Override
        public void run() {
            Intent intent=new Intent(flash_Activity.this,MainActivity.class);
            startActivity(intent);
        }
    },2000);

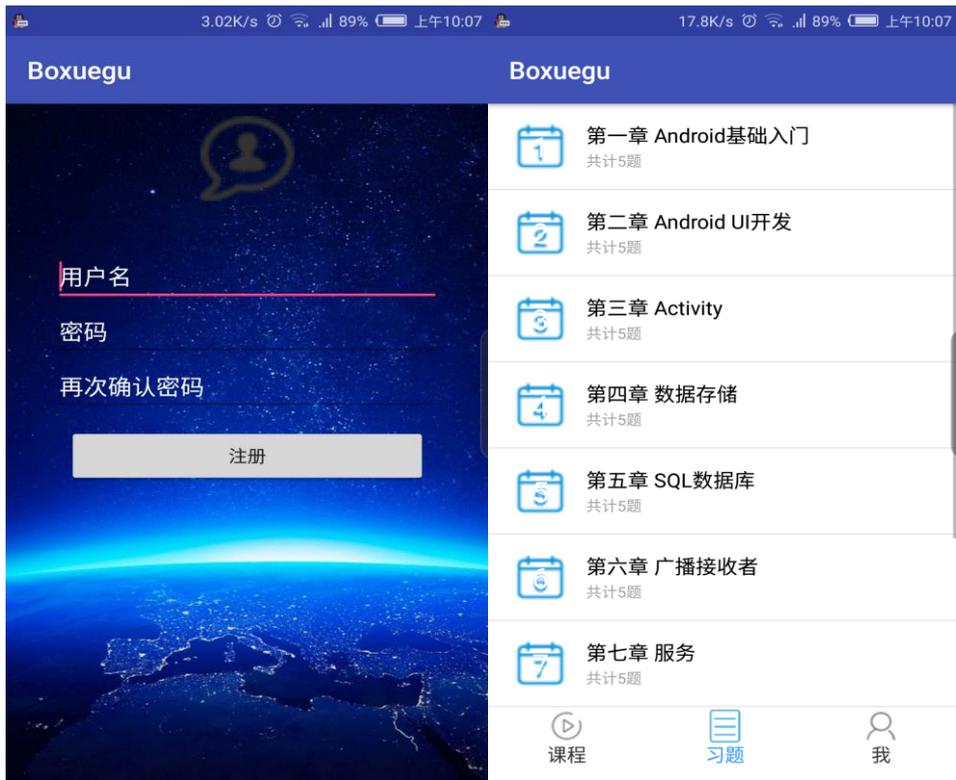
    new Handler().postDelayed(new Runnable() {
        @Override
        public void run()
        {

        }

    }, 2000);
}
```

4 结果展示





5 总结

经过艰苦的奋斗，我们的 android 平台的书架空间软件的设计与实现终于完成了，系统成功通过测试，基本上实现模拟器的实现和手机操作的实现。一切准备就绪后开始进行项目设计的启动，一开始还是遇到了各种各样的问题，通过自己的不断尝试，不断的学习，终于解决了项目中的一些难点，在这过程中我感觉到经历很多，收益很多。其中我了解了很多以前在书本中无法学习到的知识，我发现只有自己实践才能更好地提升自己的能力，我发现光有知识是不够的，还需要与实践相结合，这样才能提高自己的专业知识和操作能力。