软件设计模式

E 4　Multi-thread Programming

组号：CH

组员：黄志远，陈相

组长：陈相

报告撰写人：黄志远

报告撰写时间：2016/06/22

组员分工

学号	姓名	角色/分工	完成度
SA15226501	黄志远	组员/	100%

1.主题概述

1、修改 E 3 中的 LoginOrRegister 代码， 当导航到该页面时， 使用 SharedPreferences 保存或读取用户登录状态。

2、修改 LoginOrRegister 代码，当用户点击登录时，按 E 3 中登录规则判断是否通过，如果登录成功则将当前登录名输入框的值写入到 SharedPreferences 中的 userName 记录中，另外再写入 loginState 值为 1。其他保持不变。

3、修改 LoginOrRegister 代码，当用户点击注册时，将当前登录名输入框的值写入到 SharedPreferences 中的 userName 记录中，另外再写入 loginState 值为 1。其他保持不变。

4、修改 LoginOrRegister 代码，当用户点击返回时，判断 SharedPreferences 中是否有用户名记录， 如无， 则保持原功能不变； 否则在 SharedPreferences 写入 loginState 值为 0。

5、修改 MainScreen 和 SCOSEntry 代码，参照以上任务，将用户是否登录的判断逻辑改成使用 SharedPreferences 的 loginState 值进行功能实现。

6、在 SCOS 的 es.source.code.activity 包中，新建 SCOSHelper.java 类为 Activity 子类。该 Activity 中设置 RelativeLayout 布局为：使用 GridView 以 Dashboard 风格显示该页面帮助选项：用户使用协议，关于系统，电话人工帮助，短信帮助，邮件帮助。并实现功能。

7、修改 E 3 中 FoodOrderView 代码，使用 AsyncTask 模拟结账功能。当用户点击结账按钮时，启动 AsyncTask，并使用 ProgressBar 实时显示结账进度，要求 AsyncTask 运行 6 秒后模拟完成结账功能。当结账完成后，修改结账按钮为不可点击状态，并使用 Toast 提示用户本次结账金额及积分增加情况。

2、假设

开发工具：Android studio
Android SDK：19

3、实现或证明

1、使用 SharedPreference 保存或读取用户登录状态，在 LoginOrRegister 界面判断是否存在登录状态：若有则隐藏注册按钮，否则隐藏登录按钮并读取用户名设置为默认的登录用户。

```
    void haveRecord(){
        SharedPreferences sp = getSharedPreferences("user_reocrd",
Activity.MODE_PRIVATE);
        String name = sp.getString("name", "");
//          String pass = sp.getString("password","");
        if(checkText(name)){
            tvSignIn.setVisibility(View.INVISIBLE);
            editName.setText(name);
        }
        else {
            btnLogin.setVisibility(View.INVISIBLE);
        }
    }
```

效果如下：

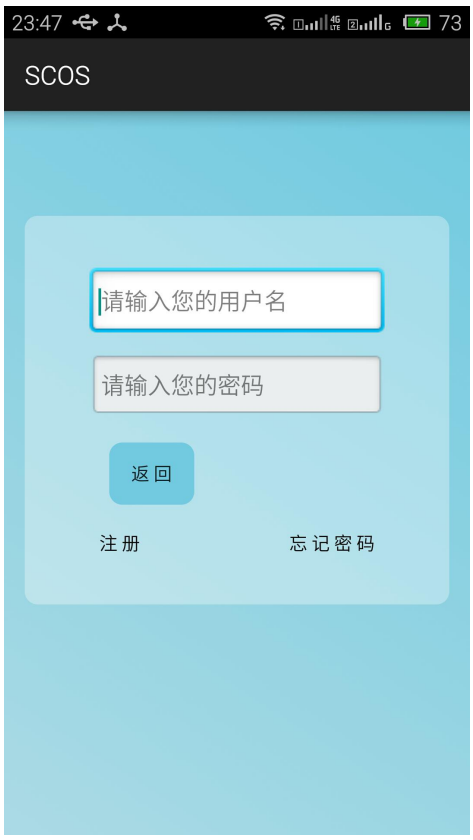

图 1 首次注册登录按钮隐藏

图 2 注册按钮隐藏

2、修改 LoginOrRegister 代码，完成点击登录后的输入验证，判断是否通过，通过后将该用户信息保存到 SharedPreference 中，并将 loginState 值置为 1。

```
    //save the login record
    void saveReocde(String name){
        SharedPreferences sp = getSharedPreferences("user_reocrd",
Activity.MODE_PRIVATE);
        SharedPreferences.Editor editor = sp.edit();
        editor.putString("name",name);
//          editor.putString("password",pass);
        editor.putInt("loginState",1);
        editor.apply();
        Toast.makeText(this,"write success",Toast.LENGTH_SHORT).show();
    }
```

图 3 输入验证失败

图 4 注册并保存成功

3、修改 LoginOrRegister 代码，当点击注册时，将当前登录名输入框的值写入到 SharedPreference 中的 username 中，另外再写入 loginState 值为 1。当用户点击返回时，判断 SharedPreference 是否有用户名记录，如无，保持原功能不变，否则在 SharedPreference 中写入 loginState 值为 0。修改 MainScreen 和 SCOSEntry 代码，将用户是否登录的判断逻辑改成使用 SharedPreference 和 loginState 值进行判断。

```
Intent intent = this.getIntent();
String strReceive = "";
User receiveUser = (User)intent.getSerializableExtra("user");
strReceive = intent.getStringExtra("data");

SharedPreferences sp = getSharedPreferences("user_reocrd",
Activity.MODE_PRIVATE);
int loginstate = sp.getInt("loginState",1);
if(loginstate==1){
    user.setUserName(receiveUser.getUserName());
    user.setPasswrod(receiveUser.getPasswrod());
    user.setOldUser(receiveUser.getOldUser());
}
else{
    user.setUserName("root");
    user.setPasswrod("root");
    user.setOldUser(false);
}
```

4、在 SCOS 的 es.source.code.activity 包中，新建 SCOSHelper.java 类，并在布局文件中国设置 RelativeLayout，使用 GridView 以 DashBoard 风格显示该页面帮助选项：用户使用协议，关于系统，电话人工服务，短信帮助，邮件帮助。

图 5　SCOSHelper 布局界面

```
 else if(position == 2){
//
Toast.makeText(getApplicationContext(),"2",Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent();
                    intent.setAction("android.intent.action.CALL");
                    intent.setData(Uri.parse("tel:" + "5554"));
                    startActivity(intent);
                }
```

图 6　电话人工服务自动拨号界面

```
else if(position == 3){
    String phone = "5554";
    String context = "test scos helper";
    SmsManager manager = SmsManager.getDefault();
    ArrayList<String> list = manager.divideMessage(context);
    for(String text:list)
        manager.sendTextMessage(phone,null,text,null,null);
    Toast.makeText(getApplicationContext(),"求助短信发送成功
",Toast.LENGTH_SHORT).show();
}
```

图 7　短信帮助自动发送界面

```
                else if(position == 4){
                    Thread MailSender = new Thread(new Runnable() {
                        @Override
                        public void run() {
                            Intent data=new Intent(Intent.ACTION_SENDTO);

data.setData(Uri.parse("mailto:huangzy099@163.com"));
                            data.putExtra(Intent.EXTRA_SUBJECT, "subject");
                            data.putExtra(Intent.EXTRA_TEXT, "text");
                            startActivity(data);
                            Message message = new Message();
                            message.what = 1;
                            mhandler.sendMessage(message);
                        }
                    });
                    MailSender.start();
```

```
//
Toast.makeText(getApplicationContext(),"4",Toast.LENGTH_SHORT).show();

                }
 mhandler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            switch (msg.what){
                case 1:
                    Toast.makeText(getApplicationContext(),"求助邮件发送成
功",Toast.LENGTH_SHORT).show();
                    break;
                default:
                    break;
            }
            super.handleMessage(msg);
        }
    };
}
```

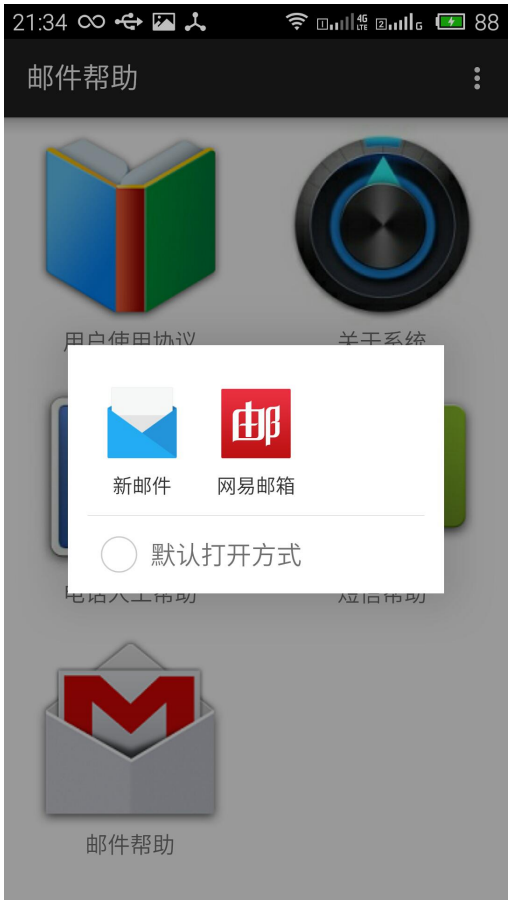

图 8 选择邮件客户端

图 9 编译发送邮件

5、修改 FoodOrderView 代码，使用 AsyncTask，并使用 ProgressBar 实时显示结账进度，运行 6 秒模拟完成结账功能，切且结账完成修改结账按钮为不可点击状态，并使用 Toast 提示用户本次结账金额及积分增加情况。

图 10 模拟结账进度界面

4、结论

1，Android 应用开发中，如何提高程序性能和避免 ANR 异常的出现？

什么是 ANR 异常？

application not response 应用程序无响应。在用户进行一个操作，如果用户在 6 秒内没有操作的话，就会出现异常。

为什么会出现 ANR 异常？

android 系统中处理用户操作的工作时在主线程中执行的，如果我们的程序在主线程中进行一些耗时的操作，导致用户的操作 6 秒不能够处理，就会出现异常。主线程休眠，那么再点击程序，必须等主线程睡醒后才会反应。所以在主线程中不要做太耗时的工作，因为主界面会阻塞。

解决办法：

（1）避免在主线程上进行复杂耗时的操作，比如说发送接收网络数据/进行大量计算/操作数据库/读写文件等。这个可以通过使用 AsyncTask 或者使用多线程来实现。

（2）broadCastReceiver 要进行复杂操作的的时候，可以在 onReceive()方法中启动一个 Service 来处理

（3）在设计及代码编写阶段避免出现出现同步/死锁或者错误处理不恰当等情况。

2，多线程编程中，如何保证线程安全？

（1）不要跨线程访问共享变量

（2）使共享变量是 final 类型的

（3）将共享变量的操作加上同步

5.参考文献

暂无

6.会议备注

会议时间：2016、06、022
地点：　　思贤楼 209
参加人员：　黄志远、陈相
主要内容：　实验四的分工