



華南師範大學

本科学生实验（实践）报告

院 系：计 算 机 学 院

实验课程：移动应用开发

实验项目：登录界面的实现

软件名称：LoginApp

指导老师：曹阳

开课时间：2018 ~ 2019 年度第 1 学期

专 业：计算机科学与技术

班 级：2016 级 4 班

学 生：余梓权

学 号：20162180149

完成时间：2018 年 11 月 13 日

华南师范大学教务处

权利声明：该项目的所有内容，包括文档、源代码、资源文件，曹阳老师和本人都拥有对其进行修改、复制、引用、阅读的权利。

版本说明：截止该文档完成时间，该项目 App 为 V 1.0.0。

目录

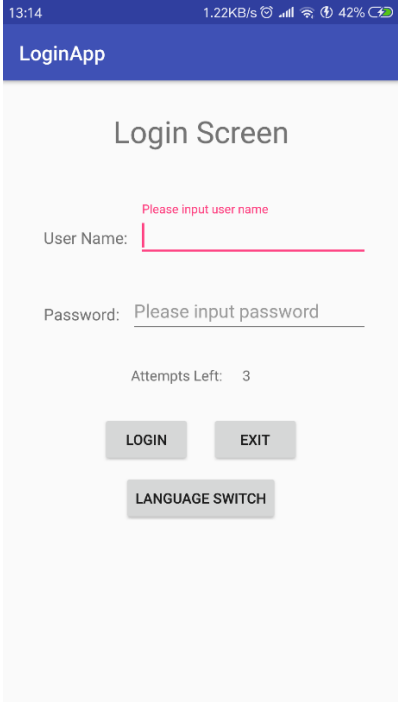
一、软件内容简介	4
二、界面设计.....	4
(一) 未填写个人信息时	4
(二) 填写个人信息完毕时	5
(三) 登录失败时	5
(四) 登录成功时	6
(五) 语言切换时	6
三、代码设计.....	7
(一) 实现双语的两份 strings.xml 文件	7
(二) 语言状态的加载.....	7
(三) 语言的切换.....	8
(四) 登录按钮的响应.....	8
(五) 语言状态的存储.....	9
(六) xml 布局文件（部分）	10
(七) UI 测试结果	11
四、软件操作流程	11
(一) 如何填写个人信息	11
(二) 如何登录	11
(三) 如何退出界面	11
(四) 如何切换语言	11
五、难点和解决方案.....	12
(一) 语言实时切换的实现	12
(二) 语言状态的保存和加载	12
(三) UI 测试的相关包的引入	12
六、不足之处和今后的设想	13
(一) 不足之处	13
(二) 今后的设想.....	13
七、参考文献.....	13

一、软件内容简介

实现一个支持双语（中文和英文）的登录界面，管理安全问题和错误尝试，并实现 UI 测试。其中，该 App 不仅可以自动适应手机的系统语言，还可以做到双语的手工实时切换，并将该 App 的语言状态保存起来，下次打开 App 会以之前设置的语言状态加载 app 的所有界面。

二、界面设计

（一）未填写个人信息时



（二）填写个人信息完毕时



13:13 0.08KB/s 41%

LoginApp

登录窗口

请输入用户名

用户名: admin

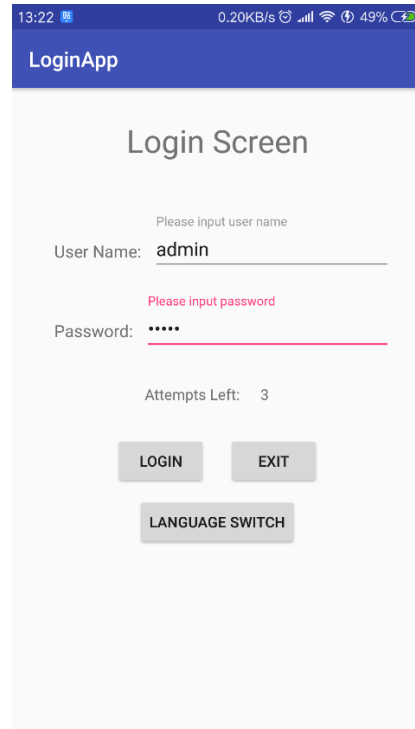
请输入密码

密码:

可尝试的剩余次数: 3

登录 退出

语言切换



13:22 0.20KB/s 49%

LoginApp

Login Screen

Please input user name

User Name: admin

Please input password

Password:

Attempts Left: 3

LOGIN EXIT

LANGUAGE SWITCH

（三）登录失败时



13:15 0.03KB/s 42%

LoginApp

登录窗口

请输入用户名

用户名: admin0

请输入密码

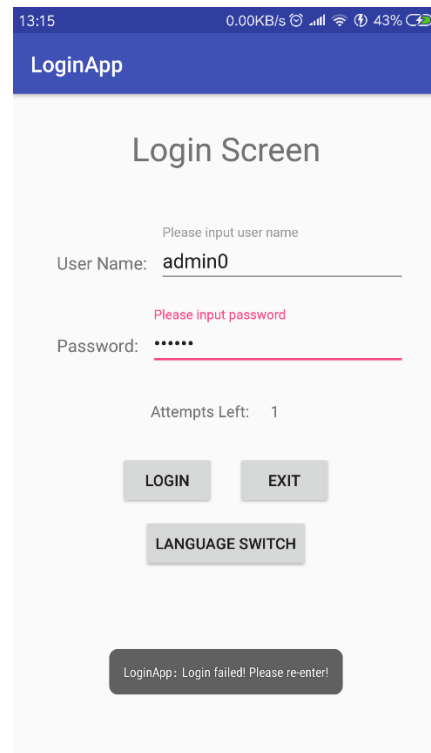
密码:|

可尝试的剩余次数: 2

登录 退出

语言切换

登录界面: 登录失败! 请重新输入!



13:15 0.00KB/s 43%

LoginApp

Login Screen

Please input user name

User Name: admin0

Please input password

Password:

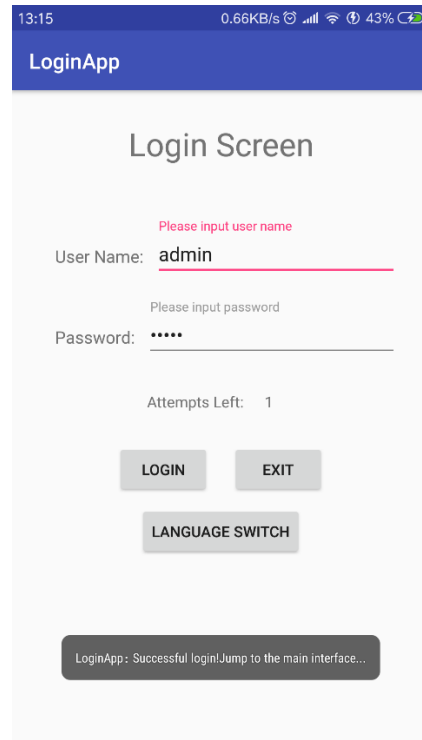
Attempts Left: 1

LOGIN EXIT

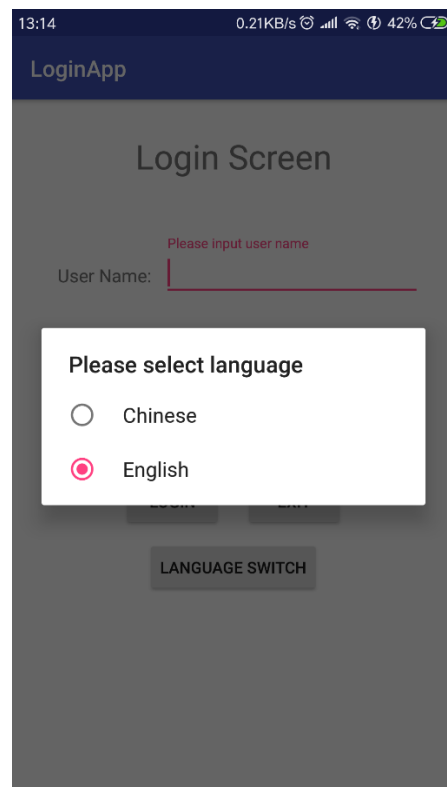
LANGUAGE SWITCH

LoginApp: Login failed! Please re-enter!

（四）登录成功时



（五）语言切换时



三、代码设计

(一) 实现双语的两份 strings.xml 文件

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">登录界面</string>
    <string name="login_interface_title">登录窗口</string>
    <string name="user_name_title">用户名: </string>
    <string name="user_name_hint">请输入用户名</string>
    <string name="password_title">密码: </string>
    <string name="password_hint">请输入密码</string>
    <string name="attempts_hint">可尝试的剩余次数: </string>
    <string name="btn_login_text">登录</string>
    <string name="login_success_hint">登录成功! 跳转至主界面.....</string>
    <string name="login_fail_hint">登录失败! 请重新输入!</string>
    <string name="login_disabled">您的尝试次数已达上限!</string>
    <string name="btn_exit_text">退出</string>
    <string name="btn_switch_language">语言切换</string>
    <string name="select_language_title">请选择语言</string>
    <string name="zh">中文</string>
    <string name="en">英文</string>
</resources>
```

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">LoginApp</string>
    <string name="login_interface_title">Login Screen</string>
    <string name="user_name_title">User Name:</string>
    <string name="user_name_hint">Please input user name</string>
    <string name="password_title">Password:</string>
    <string name="password_hint">Please input password</string>
    <string name="attempts_hint">Attempts Left:</string>
    <string name="btn_login_text">LOGIN</string>
    <string name="login_success_hint">Successful login!Jump to the main interface...</string>
    <string name="login_fail_hint">Login failed! Please re-enter!</string>
    <string name="login_disabled">Your number of attempts has reached the limit!</string>
    <string name="btn_exit_text">EXIT</string>
    <string name="btn_switch_language">Language Switch</string>
    <string name="select_language_title">Please select language</string>
    <string name="zh">Chinese</string>
    <string name="en">English</string>
</resources>
```

(二) 语言状态的加载

```
private void initLocaleLanguage()
{
    Resources resources = getResources();
    Configuration configuration = resources.getConfiguration();
    DisplayMetrics displayMetrics = resources.getDisplayMetrics();

    configuration.locale = new Locale(mSharedPreferences.getString("language",
        Locale.getDefault().getLanguage()), country: "");
    resources.updateConfiguration(configuration, displayMetrics);
}
```

(三) 语言的切换

```
/**
 * 语言切换
 *
 * @param language 切换的目标语言
 */
private void switchLanguage(String language)
{
    Resources resources = getResources();
    Configuration configuration = resources.getConfiguration();
    DisplayMetrics displayMetrics = resources.getDisplayMetrics();

    if ("zh".equals(language))
    {
        configuration.locale = Locale.SIMPLIFIED_CHINESE;
    }
    else if ("en".equals(language))
    {
        configuration.locale = Locale.ENGLISH;
    }
    else
    {
        configuration.locale = Locale.getDefault();
    }
    resources.updateConfiguration(configuration, displayMetrics);

    finish();

    Intent intent = new Intent( packageContext MainActivity.this, MainActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(intent);
}
```

(四) 登录按钮的响应

```
@OnClick({R.id.btn_login, R.id.btn_exit, R.id.btn_switch_language})
public void handleAllClick(View v)
{
    switch (v.getId())
    {
        case R.id.btn_login:
            if (sAttemptTimes > 0)
            {
                if ("admin".equals(mEtUserName.getText().toString()) && "12345".equals(mEtPassword.getText().toString()))
                {
                    Toast.makeText( context MainActivity.this, getResources().getString(R.string.Login_success_hint), Toa
                }
                else
                {
                    sAttemptTimes--;
                    mTvAttemptsLeft.setText(sAttemptTimes + "");

                    if (sAttemptTimes == 0)
                    {
                        mBtnLogin.setEnabled(false);
                        Toast.makeText( context MainActivity.this, getResources().getString(R.string.Login_disabled), Toa
                    }
                }
            }
            else
            {
                Toast.makeText( context MainActivity.this, getResources().getString(R.string.Login_fail_hint), Toa
            }
        }
    }
}
```


（五）语言状态的存储

```
@BindView(R.id.btn_login)
Button mBtnLogin;

@BindView(R.id.btn_exit)
Button mBtnExit;

@BindView(R.id.btn_switch_language)
Button mBtnSwitchLanguage;

private static int sAttemptTimes = 3;

private String[] mItems;

private SharedPreferences mSharedPreferences;

private SharedPreferences.Editor mEditor;
```

```
AlertDialog alertDialog = new AlertDialog.Builder(context: this).setTitle(getResources().getString(R.string
{
    @Override
    public void onClick(DialogInterface dialog, int which)
    {
        if (which == selectedIndex)
        {
            return;
        }

        mEditor = mSharedPreferences.edit();

        if (which == 0)
        {
            mEditor.putString(s: "language", s1: "zh");
            switchLanguage("zh");
        }
        else
        {
            mEditor.putString(s: "language", s1: "en");
            switchLanguage("en");
        }

        mEditor.apply();
    }
}).create();
```

(六) xml 布局文件（部分）

```
        android:layout_height="match_parent"
        android:layout_margin="10dp"
        android:gravity="bottom"
        android:textSize="16sp"
        android:text="@string/user_name_title"/>

        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:id="@+id/et_user_name"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="@string/user_name_hint"/>

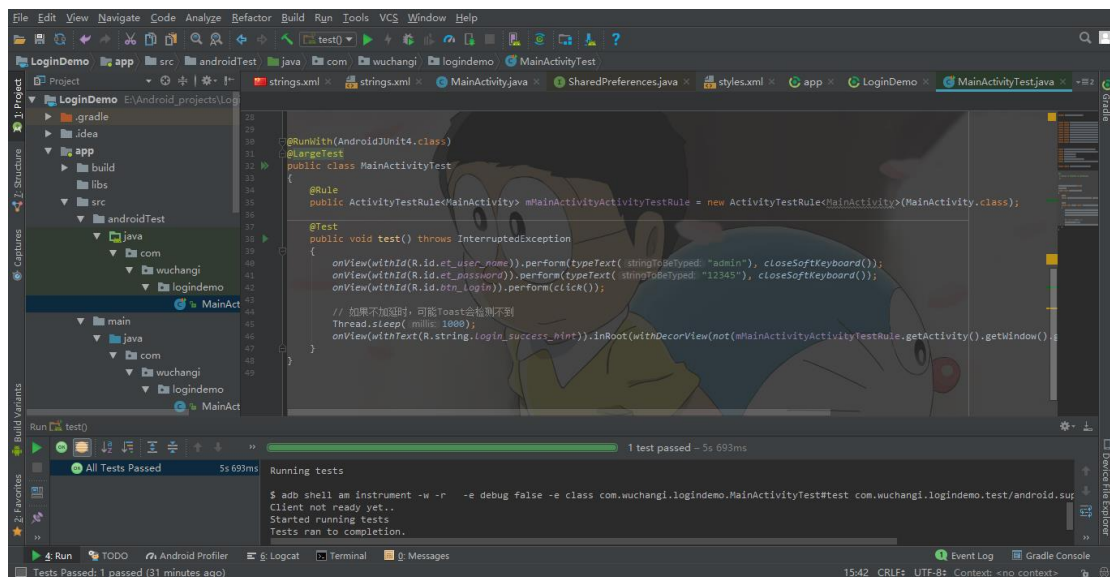
        </android.support.design.widget.TextInputLayout>

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:gravity="bottom"
            android:textSize="16sp"
            android:text="@string/password_title"/>
```

（七）UI 测试结果



四、软件操作流程

（一）如何填写个人信息

点击界面的两个输入框，当光标停留在其中即可按照提示输入相关个人信息。

（二）如何登录

点击界面左下方的登录按钮即可。

（三）如何退出界面

点击界面右下方的退出按钮即可

（四）如何切换语言

点击界面最下方的切换语言按钮，会弹出一个列表框，从列表框中选中目标语言即可，整个 APP 会自行重启并重新设置语言。

五、难点和解决方案

（一）语言实时切换的实现

采用了重启 Activity 并且清空所有任务栈和进程的方式加以实现。

```
Intent intent = new Intent(packageContext, MainActivity.class);
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
startActivity(intent);
```

（二）语言状态的保存和加载

保存语言状态使用 SharedPreferences，App 再次打开后，要实现加载之前保存的语言状态，通过以下方式加以实现：

```
resources.updateConfiguration(configuration, displayMetrics)
```

这里有一个坑，语言状态的加载（initLocaleLanguage），必须置于 setContentView 方法之前，否则无效。

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);

    mSharedPreferences = PreferenceManager.getDefaultSharedPreferences(context, this);

    initLocaleLanguage();

    setContentView(R.layout.activity_main);

    ButterKnife.bind(target, this);

    mItems = new String[]{getResources().getString(R.string.zh), getResources().getString(R.string.en)};
}
```

（三）UI 测试的相关包的引入

使用谷歌推荐的 espresso 进行 UI 测试之前，必须以静态方式（import static）导入相关包，而且必须是自己手工导入！也就是自己写代码！就算你调用了相关方法（比如 onView 方法），然后按 ALT+Enter 键，AS 也没有帮你自动导入，虽然你已经添加了 espresso 的依赖，这确实是件很头疼的事情。虽然之前写了挺久的 Android，但是第一次进行 UI 测试，被坑了很久。以后记得进行 UI 测试之前，先静态导入（复制粘贴）以下代码段：

```
import static android.support.test.espresso.Espresso.onView;  
import static android.support.test.espresso.action.ViewActions.click;  
import static android.support.test.espresso.action.ViewActions.closeSoftKeyboard;  
import static android.support.test.espresso.action.ViewActions.typeText;  
import static android.support.test.espresso.assertion.ViewAssertions.matches;  
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;  
import static android.support.test.espresso.matcher.ViewMatchers.withId;  
import static android.support.test.espresso.matcher.ViewMatchers.withText;  
import static android.support.test.espresso.matcher.RootMatchers.withDecorView;
```

六、不足之处和今后的设想

（一）不足之处

因为手头还有其他作业和项目需要本人去完成，所以这次实验只是做了基本的功能和界面。该 App 功能还不够完善，比如对用户的个人密码没有进行加密保存。界面也不够美观，没有进行详细的界面设计，MD 风格没有完全彰显出来。

（二）今后的设想

如果以后有时间的话，打算用 **MVP+MD 风格+login/register 模块+json+Retrofit+SSM/Spring Boot+MD5 加密** 的方式实现这个小样例，顺便复习一下之前学过的相关 Android 端和后端知识。

七、参考文献

- 1、《阿里巴巴 Android 开发手册》