项目简介

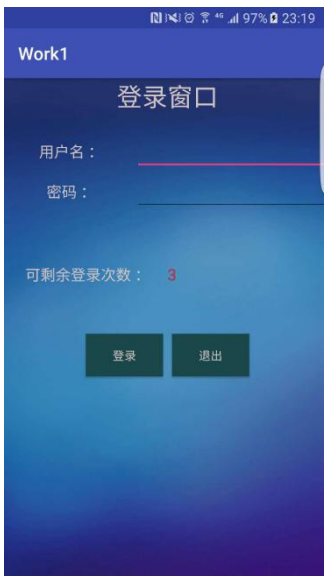
软件名称：登陆界面
开发平台：android studio
完成人：莫振尧
学号：20152100175
完成时间：2017/11/3

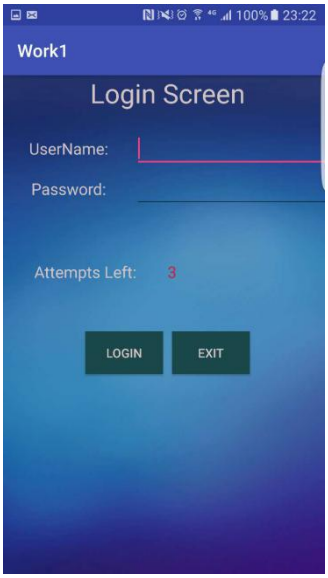
软件内容简介

一个登陆界面，有基本的用户账号密码检测和错误登陆登陆超次数的处理。

界面设计

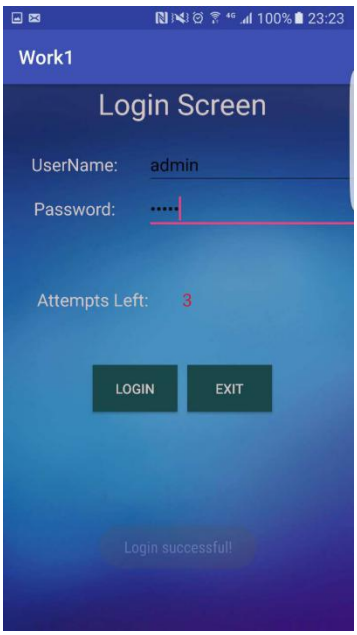
主界面





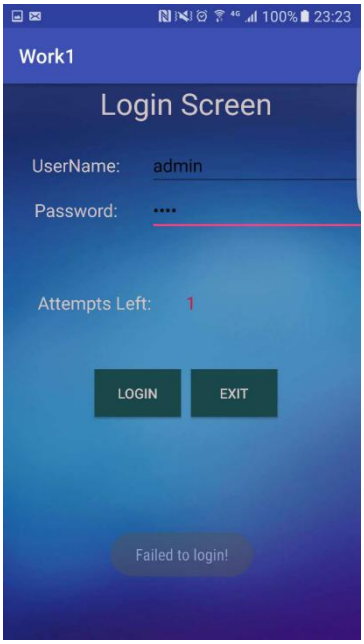
登陆成功


100% 23:20
Work1
登录窗口
用户名：
admin
密码：
可剩余登录次数：
3
登录
退出
登录成功！

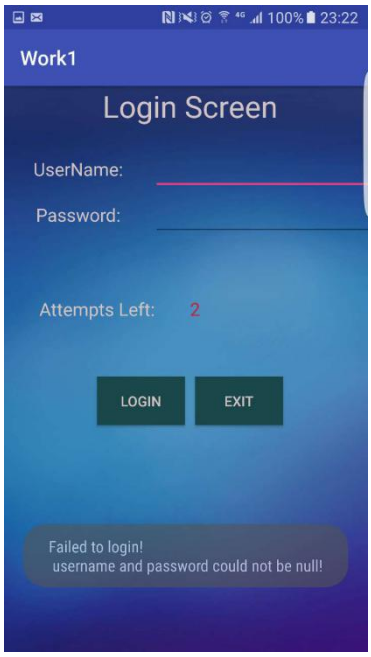

100% 23:23
Work1
Login Screen
UserName:
admin
Password:
Attempts Left:
3
LOGIN
EXIT
Login successful!

登陆失败


94% 23:35
Work1
登录窗口
用户名：
admin
密码：
可剩余登录次数：
1
登录
退出
登录失败！


100% 23:23
Work1
Login Screen
UserName:
admin
Password:
Attempts Left:
1
LOGIN
EXIT
Failed to login!

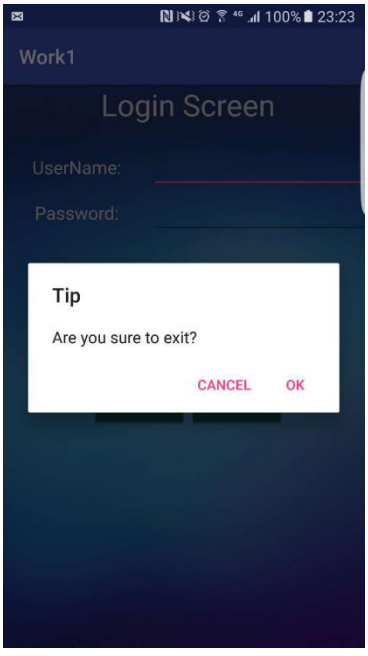
空检测


100% 23:22
Work1
Login Screen
UserName:
Password:
Attempts Left: 2
LOGIN
EXIT
Failed to login!
username and password could not be null!


98% 23:19
Work1
登录窗口
用户名：
密码：
可剩余登录次数： 2
登录
退出
登录失败！
用户名和密码不可为空！

离开程序


100% 23:21
Work1
登录窗口
用户名： admin
密码： ••••
提示
你确定要离开吗？
取消
确定


100% 23:23
Work1
Login Screen
UserName:
Password:
Tip
Are you sure to exit?
CANCEL
OK

代码设计

```
public class MainActivity extends AppCompatActivity {
    private String username;
    private String password;

    private EditText usernameText;
    private EditText passwordText;

    private int wrongLoginLeftNum;

    private TextView loginLeftNum;

    private Button loginBtn;
    private Button exitBtn;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initial();
        clickLogin();
        clickExit();
    }

    private void initial(){
        this.username="admin";
        this.password="admin";
        this.wrongLoginLeftNum=3;

        this.usernameText=(EditText) findViewById(R.id.user_name);
        this.passwordText=(EditText)findViewById(R.id.passwd);

        this.loginLeftNum=(TextView)findViewById(R.id.login_left_num);
        this.setLoginLeftNumText();

        this.loginBtn=(Button)findViewById(R.id.login);
        this.exitBtn=(Button)findViewById(R.id.exit);
    }

    private void setLoginLeftNumText(){
        this.loginLeftNum.setText(String.valueOf(this.wrongLoginLeftNum));
```

```
        this.loginLeftNum.setTextColor(this.getResources().getColor(R.color.colorNumText));
    }

    private void clickLogin(){
        this.loginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(MainActivity.this.wrongLoginLeftNum==0){
                    MainActivity.this.tooMuchWrongLogin();
                }

                String inputUsrename= MainActivity.this.usernameText.getText().toString();
                String inputPasswd= MainActivity.this.passwordText.getText().toString();

                if(inputPasswd.equals("")||inputUsrename.equals("")){
                    String loginNull=MainActivity.this.getString(R.string.login_null);
                    Toast.makeText(MainActivity.this,loginNull,Toast.LENGTH_LONG).show();
                    MainActivity.this.wrongLoginLeftNum--;
                }else
if(inputUsrename.equals(MainActivity.this.username)&&inputPasswd.equals(MainActivity.this.pa
ssword)){
                    MainActivity.this.loginSuccessful();
                }else{
                    MainActivity.this.loginFail();
                }

                MainActivity.this.setLoginLeftNumText();
            }
        });
    }

    private void tooMuchWrongLogin(){
        String canNotLogin=MainActivity.this.getString(R.string.can_not_login);
        new AlertDialog.Builder(MainActivity.this)
                .setTitle(MainActivity.this.getString(R.string.dialog_tip))
                .setMessage(canNotLogin)
                .setPositiveButton(MainActivity.this.getString(R.string.dialog_sure) ,
                        new DialogInterface.OnClickListener() {
                            public void onClick(
                                    DialogInterface dialoginterface, int i){
                                System.exit(0);
                            }
                        }).show();
    }
```

```
    private void loginSuccessful(){
        String loginSucc=MainActivity.this.getString(R.string.login_success);
        Toast.makeText(MainActivity.this,loginSucc,Toast.LENGTH_LONG).show();
        MainActivity.this.wrongLoginLeftNum=3;
    }

    private void loginFail(){
        String loginFail=MainActivity.this.getString(R.string.login_fail);
        Toast.makeText(MainActivity.this,loginFail,Toast.LENGTH_LONG).show();
        MainActivity.this.wrongLoginLeftNum--;
    }



    private void clickExit(){
        this.exitBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                MainActivity.this.ifExit();
            }
        });
    }

    private void ifExit(){
        String exitQuest=MainActivity.this.getString(R.string.dialog_exit);
        new AlertDialog.Builder(MainActivity.this)
                .setTitle(MainActivity.this.getString(R.string.dialog_tip))
                .setMessage(exitQuest)
                .setPositiveButton(MainActivity.this.getString(R.string.dialog_sure) ,
                        new DialogInterface.OnClickListener() {
                            public void onClick(
                                    DialogInterface dialoginterface, int i) {
                                System.exit(0);
                            }
                        }
                )
                .setNegativeButton(MainActivity.this.getString(R.string.dialog_cancel) ,null
)
                .show();
    }
}
```

UI 测试

过程：

不得不说 UI 测试就是个坑，即使看了教程，还是费了很多功夫才终于成功运行了测试。下面说一下步骤：

首先我们要修改 buil.gradle 文件,在 denpencise 里面添加如下内容：

```
//ADD THESE LINES:
androidTestCompile 'com.android.support.test:runner:0.5'
androidTestCompile 'com.android.support.test:rules:0.5'
androidTestCompile 'com.android.support.test.espresso:espresso-core:2.2.2'
```

这是第一个坑，因为教程里添加的内容为：

```
//ADD THESE LINES:
androidTestCompile 'com.android.support.test:runner:0.2'
androidTestCompile 'com.android.support.test:rules:0.2'
androidTestCompile 'com.android.support.test.espresso:espresso-core:2.1'
```

这是因为作者的 android 版本与我现在的 android 版本，幸好 androidstudio 对此有智能提示，你将鼠标移动到 0.2 那里它便提示你要修改版本为 0.5。这个倒是没什么问题。

这不是最坑的，更坑的是，我在 ExampleInstrumentedTest 里面写好了测试代码，结果它报错，显示：

Error:Conflict with dependency 'com.android.support:support-annotations' in project ':app'. Resolved versions for app (24.2.1) and test app (23.1.1) differ. See http://g.co/androidstudio/app-test-app-conflict for details.

我百度找了一下，发现要在 buil.gradle 的 configurations.all 里面添加这么一句：

resolutionStrategy.force 'com.android.support:support-annotations:23.1.1'

但我在 buil.gradle 里面翻来覆去着看了好久还是没找到哪里有什么 configurations.all，然后我继续百度，终于发现我应该直接在 denpencise 里面添加：

```
configurations.all {
    resolutionStrategy.force 'com.android.support:support-annotations:23.1.1'
}
```

这个坑浪费了我好长时间。填完这个坑，我运行程序又一个坑出现了：

java.lang.RuntimeException: No activities found. Did you forget to launch the activity by calling getActivity() or startActivitySync or similar?

没有活动在进行，我百度了好久还是没找到什么有用的建议，然后我对比一下我的代码和教程的代码，发现少了这么几行：

```
@Rule
public ActivityTestRule<MainActivity> mActivityRule = new ActivityTestRule<>(
        MainActivity.class);
```

然后我贴上这代码，然后可以了，谢天谢地，坑终于填完了，泪流满面啊。

配置与具体测试代码：

buil.gradle

```
apply plugin: 'com.android.application'


android {
    compileSdkVersion 24
    buildToolsVersion "25.0.0"
    defaultConfig {
        applicationId "com.example.dell.work1"
        minSdkVersion 15
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
rules.pro'
        }
    }
    //ADD THESE LINES:
    packagingOptions {
        exclude 'LICENSE.txt'
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:24.2.1'
```

```
    compile 'com.android.support.constraint:constraint-layout:1.0.0-alpha9'
    testCompile 'junit:junit:4.12'

    //ADD THESE LINES:
    androidTestCompile 'com.android.support.test:runner:0.5'
    androidTestCompile 'com.android.support.test:rules:0.5'
    androidTestCompile 'com.android.support.test.espresso:espresso-core:2.2.2'

    configurations.all {
        resolutionStrategy.force 'com.android.support:support-annotations:23.1.1'
    }

}
```

ExampleInstrumentedTest

```
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Rule
    public ActivityTestRule<MainActivity> mActivityRule = new ActivityTestRule<>(
            MainActivity.class);

    @Test
    public void useAppContext() throws Exception {
        // Context of the app under test.
        Context appContext = InstrumentationRegistry.getTargetContext();

        assertEquals("com.example.dell.work1", appContext.getPackageName());
    }

    @Test
    public void loginSuccessful(){
        onView(withId(R.id.user_name)).perform(typeText("admin"), closeSoftKeyboard());
//line 1
        onView(withId(R.id.passwd)).perform(typeText("admin"), closeSoftKeyboard()); //line
1

        onView(withId(R.id.login)).perform(click()); //line 2

        String expectedText = "3";
        onView(withId(R.id.login_left_num)).check(matches(withText(expectedText))); //line 3
```

```
    }

    public void fialTologin(){
        onView(withId(R.id.user_name)).perform(typeText("admin"), closeSoftKeyboard());
//line 1
        onView(withId(R.id.passwd)).perform(typeText("ad"), closeSoftKeyboard()); //line 1

        onView(withId(R.id.login)).perform(click()); //line 2

        String expectedText = "2";
        onView(withId(R.id.login_left_num)).check(matches(withText(expectedText))); //line 3

    }

}
```

难点与解决方案

难点

1. 布局为各个控件分配一定的空间，保持控件之间有足够的间隔，令它们不太过紧凑。
2. 添加背景，给文字设置大小和颜色等。

解决方法

1、在此次实验中，我还是采用了线性布局，先将整体从垂直方向分为五个模块，分别是登录窗口、用户名、密码、剩余登陆次数、登陆与退出，各个模块又有水平布局，其中为了美观，我在密码、剩余登陆次数、登陆与退出三者之间加了两个 space 以加大它们之间的间隔。

2、为了美观界面，我还给界面设置了背景，添加背景先要在 drawable 添加图片，一开始找了好久不知道怎么添加（因为没办法右键打开文件浏览器添加）。后来百度了好久终于知道，直接用快捷键 Ctrl+c 对准目标图片，然后选中在 AndroidStudio 选中 drawable 文件夹按下 Ctrl+v 就好。之后就是在 xml 中添加相应的代码即可。

3、至于对文字颜色的修改，因为多个文本都是同一种颜色(灰色)，所以我在 color.xml 中添加了一种颜色，然后然让那些文本空间直接引用颜色。

总结与未来的设想

登录界面的设置并不涉及复杂的算法，主要是要设置出简明漂亮、让人耳目一新的界面，

这方面，我想着可以模仿 qq 做一个动态的背景图，这样的界面感觉要比普通界面要漂亮的多。

关于 UI 测试这是个好东西，以后要测试的时候，靠这个写一下代码，就可以省下我们手工操作测试的好多工作。