

ICP8

Login Application

- ICP Group 4
- Name: Sailaja Narra
- Email: sntnn@umsystem.edu

My Partner

- Partner: Venkata Mahesh Mokkapati
- Email: vmzwn@umsystem.edu
- Repo: <https://github.com/UMKC-APL-WebMobileProgramming/ICP8-Mahesh68>

My report

ICP8 Video:

<https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=fb069a8e-1ce1-477e-9624-adcb016bcea9>

Source code: <https://github.com/UMKC-APL-WebMobileProgramming/ICP8-SailajaNarra>

In this task we have learnt about the android studio and application creation using it.

Installed the JDK and Android Studio initially after downloading them from their official sites.

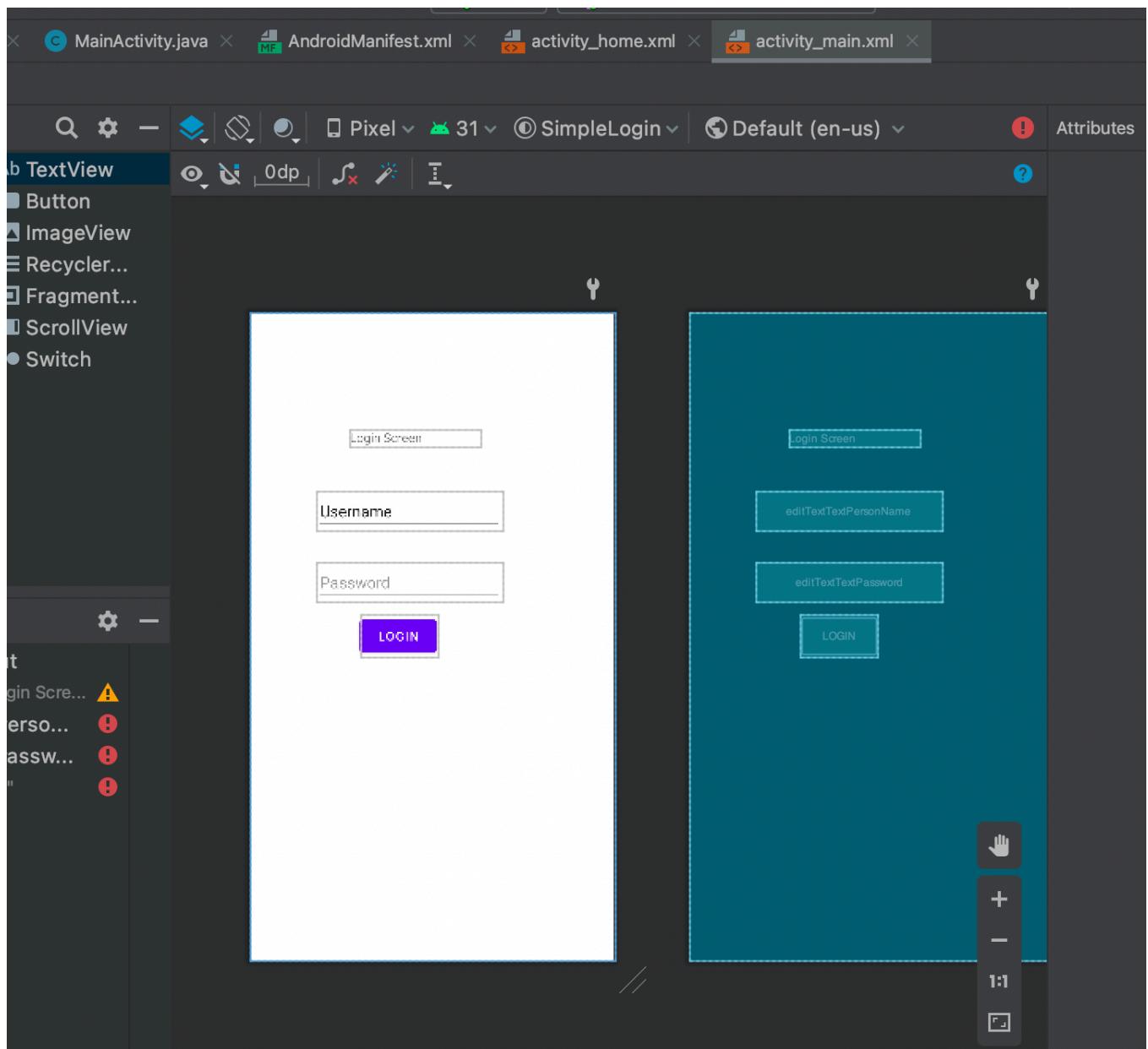
Initially created an new application in the android studio projects after selected blank activity.

Addded internet permission in the AndroidManifest.xml file to enable internet access in the device/emulator.

SimpleLogin – AndroidManifest.xml [SimpleLogin.app]

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.simplelogin">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="SimpleLogin"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SimpleLogin">
        <activity android:name=".HomeActivity"></activity>
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Now a home login screen is created in the resources layout file



Here two input fields and logout button is created using layout design.

And the login button logic is handled in the main activity.java file

The screenshot shows the Android Studio interface with the project navigation bar at the top. The current file is `MainActivity.java`, which contains Java code for a login application. The code includes imports, class definition, onCreate method, and a myLogin() method. The code is annotated with comments and several TODO markers (green exclamation points).

```
simplelogin > MainActivity > myLogin
HomeActivity.java < MainActivity.java < AndroidManifest.xml < activity_home.xml < activity_main.xml

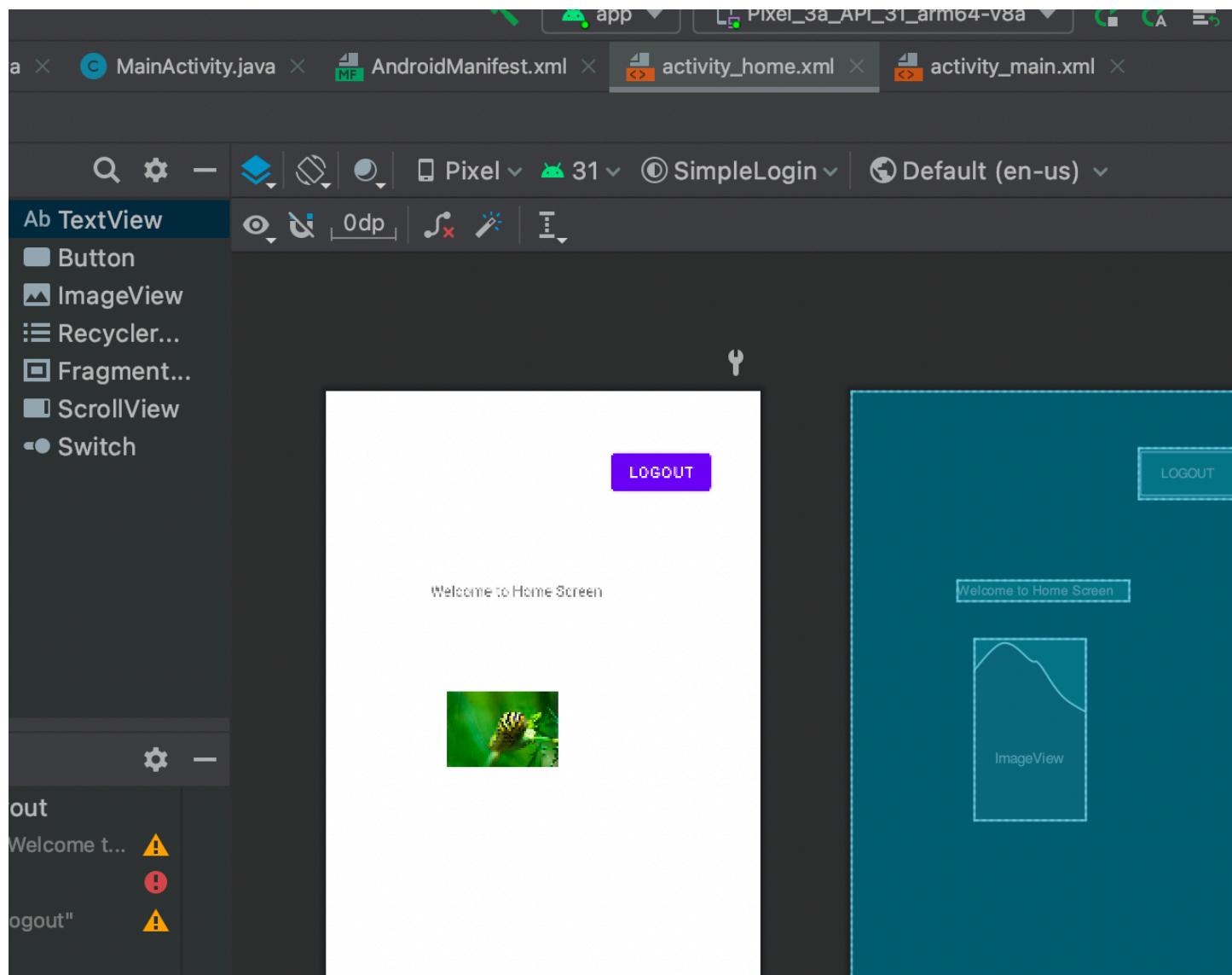
1 package com.example.simplelogin;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     private TextView disp;
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13
14        // login button is created and event is added on click
15        Button btn = (Button) findViewById(R.id.button);
16        btn.setOnClickListener(new View.OnClickListener(){
17            @Override
18            public void onClick(View v) { myLogin(); }
19        });
20
21    }
22
23    // on click of login this method will be called
24    public void myLogin() {
25
26        // input fields reference is taken and stored into local variables as strings
27        EditText usernameCtrl = (EditText) findViewById(R.id.editTextTextPersonName);
28        EditText passwordCtrl = (EditText) findViewById(R.id.editTextTextPassword);
29
30        String username = usernameCtrl.getText().toString();
31        String password = passwordCtrl.getText().toString();
32    }
33
34}
```

LOGIN method is called on click of login button.

The screenshot shows the Android Studio interface with the project 'simplelogin' open. The main window displays the Java code for the `MainActivity` class, specifically the `myLogin()` method. The code performs user input validation and checks if the provided username ('Admin') and password ('Admin') match. If valid, it starts an activity named `HomeActivity`. If invalid, it displays a toast message indicating that both the username and password are required.

```
32     public void myLogin() {
33
34         //      input fields reference is taken and stored into local variables as strings
35         EditText usernameCtrl = (EditText) findViewById(R.id.editTextTextPersonName);
36         EditText passwordCtrl = (EditText) findViewById(R.id.editTextTextPassword);
37
38         String username = usernameCtrl.getText().toString();
39         String password = passwordCtrl.getText().toString();
40
41         boolean validationFlag = false;
42
43         //      validation for inputs fields
44         if(!username.isEmpty() && !password.isEmpty()){
45             if(username.equals("Admin") && password.equals("Admin")) {
46                 validationFlag = true;
47             }
48         }
49
50         if(validationFlag){
51             Intent redirect = new Intent( packageContext: MainActivity.this, HomeActivity.class);
52             startActivity(redirect);
53             Toast login_success = Toast.makeText(getApplicationContext(), text: "Login Successful", Toast.LENGTH_SHORT);
54             login_success.show();
55             disp.setText("Login Successful");
56
57         }
58         else{
59             Toast empty_password = Toast.makeText(getApplicationContext(), text: "Please check username and password", Toast.LENGTH_SHORT);
60             empty_password.show();
61             disp.setText("Please check username and password");
62         }
63     }
64 }
```

Now home page is created using layout design



After logging into the home screen, a display text and a button is created.

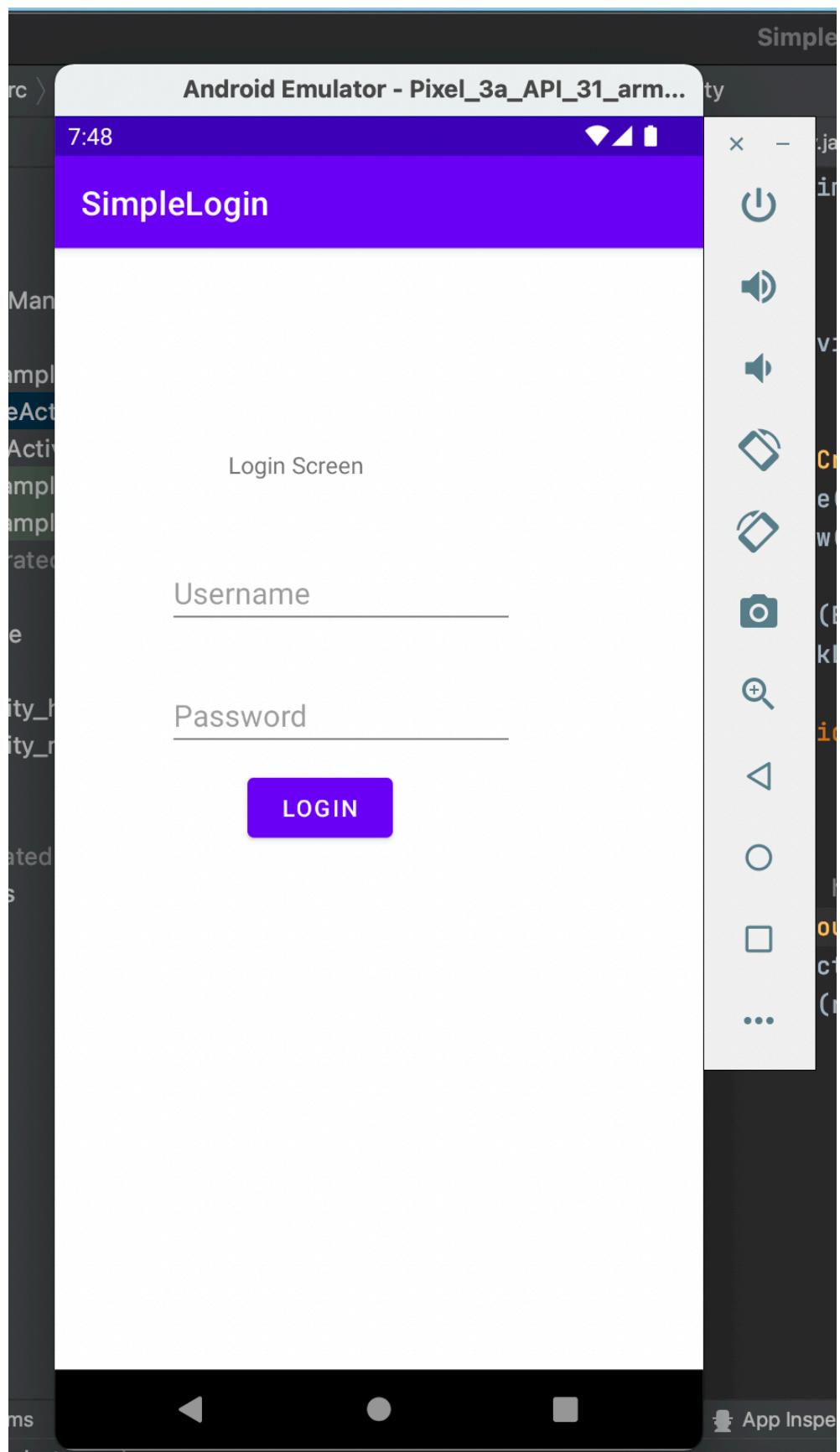
A navigation logic for this button is handled in the home activity.java file

```
1 package com.example.simplelogin;
2
3 import ...
4
5 public class HomeActivity extends AppCompatActivity{
6     @Override
7
8     protected void onCreate(Bundle savedInstanceState){
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_home);
11
12        Button btn = (Button) findViewById(R.id.button2);
13        btn.setOnClickListener(new View.OnClickListener(){
14            @Override
15            public void onClick(View v) { myLogout(); }
16        });
17    }
18
19    // navigation from home screen to login screen
20    public void myLogout() {
21        Intent redirect = new Intent(HomeActivity.this, MainActivity.class);
22        startActivity(redirect); //redirect to login page
23    }
24
25}
```

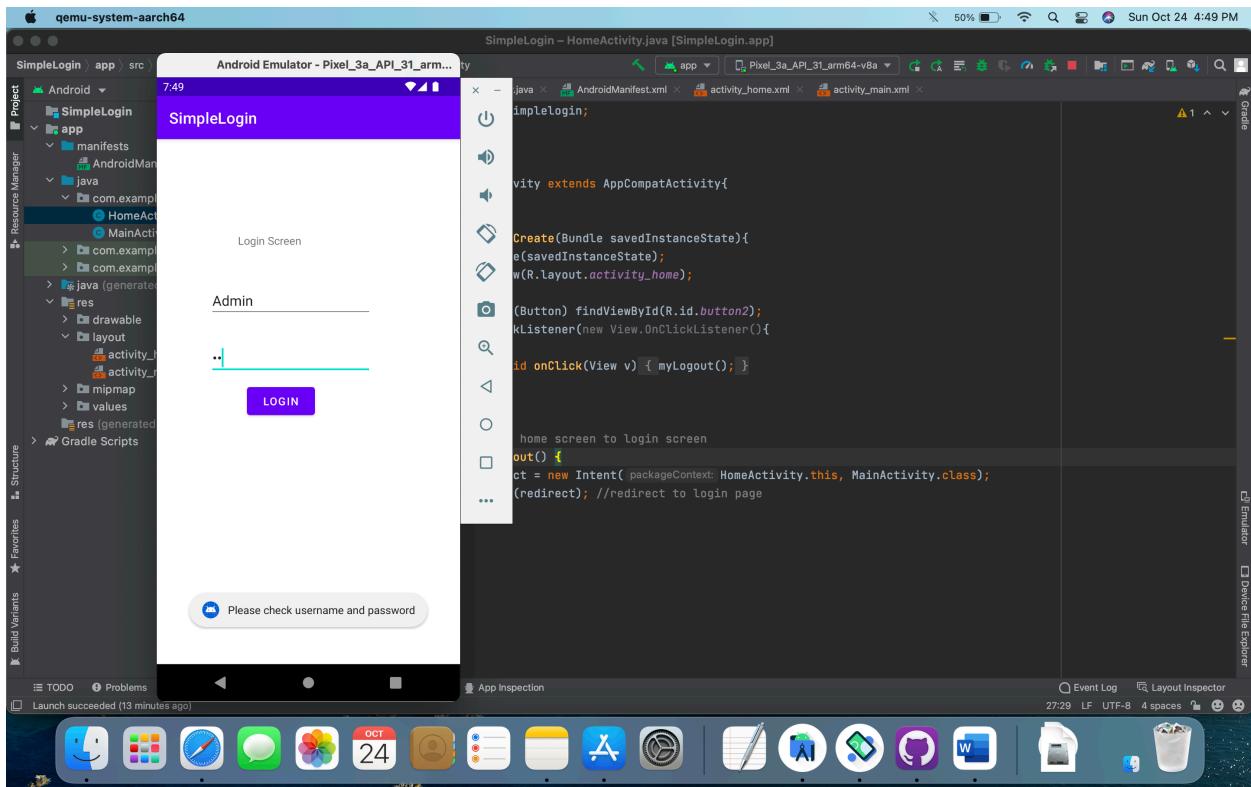
All the documentation is written in the code also with proper comments.

Output:

Emulator

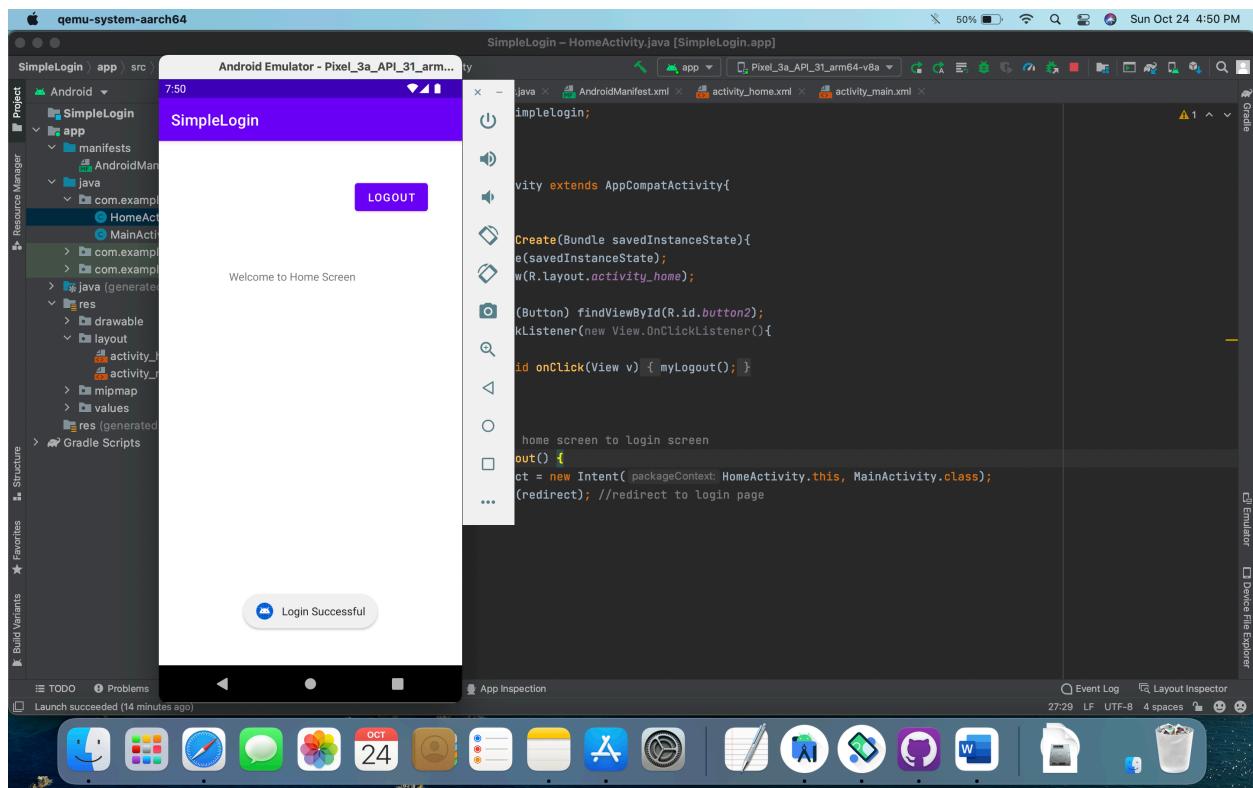


When user enters wrong credentials



After successful login

Below is the home screen



On click of logout button page will be redirected back to login screen.