


National University of Computer and Emerging Sciences, Lahore Campus

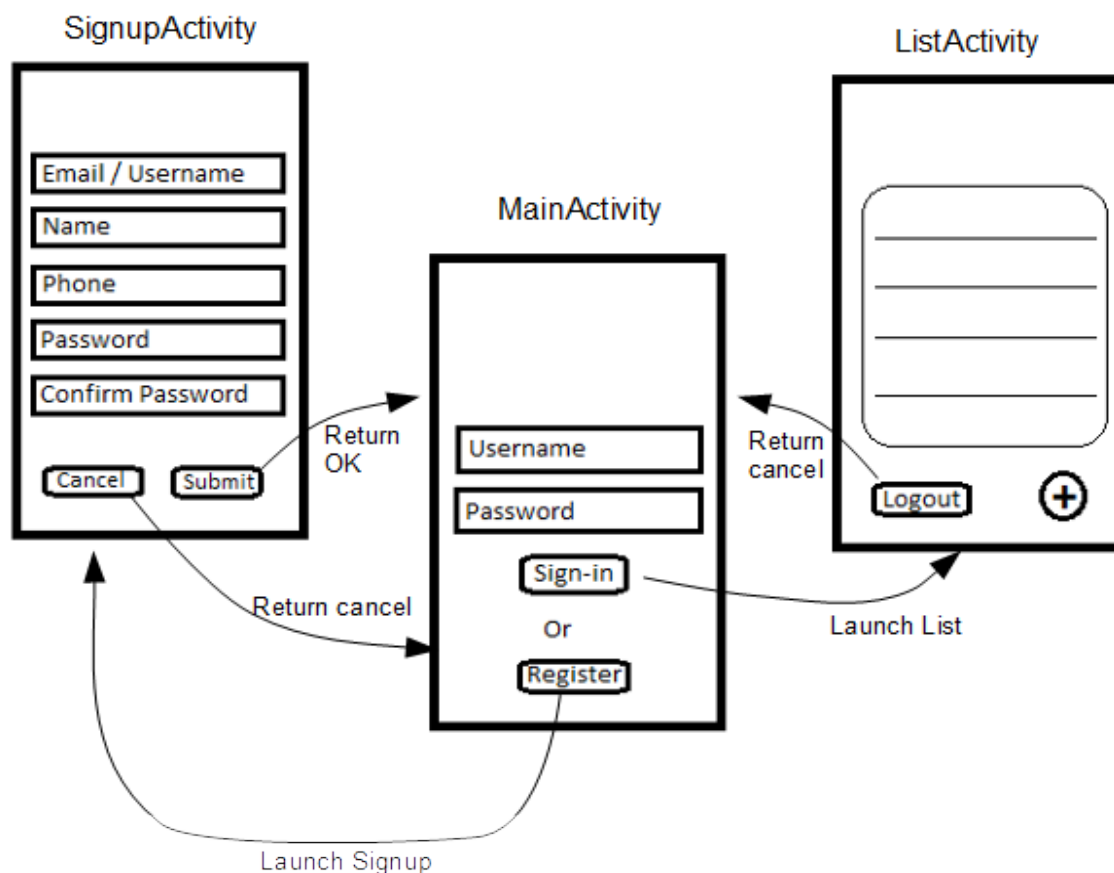
	Course Name:	Software for Mobile Devices	Course Code:	CS-4039
	Degree Program:	BS(CS)	Semester:	Spring 2022
	Exam Duration:	60 Minutes	Total Marks:	25
	Paper Date:	21 - Mar - 2022	Weight:	12.5
	Section:	ALL	Page(s):	5
	Exam Type:	Midterm-I		

Student Name:_____ **Roll No.**_____ **Section:**_____

Instruction/Notes: This exam is open book and open notes. While writing code, make best effort to write correct and relevant code only. Minor syntactic errors are acceptable and will be ignored during marking but overall concept and approach must be correct. **Also, note that Q2 is different for different sections.**

Question 1

20 points



Consider an application comprising of three activities (Main, Signup and List) with the navigation workflow as illustrated above, and explained as follows:

1. Main Activity is displayed first asking for user to enter credentials. Assuming that sign-in is successful, it launches List Activity, which performs other app-specific actions but returns back when Logout is pressed.
2. If user selects "Register" option on Main Activity, it launches Signup Activity where user enters her profile information. Either of the following possibilities may happen:

(a) Clicking on **submit** returns the result back successfully to Main Activity that considers it a proper **sign-in** and **automatically takes the user again to List Activity**

(b) Clicking on **cancel** returns the result to Main Activity which chooses to do nothing

Given the partial code skeletons for each of the three activities below, implement the workflow by adding appropriate code in the relevant functions. Note that you are only required to write code for the workflow only and ignore other aspects such as layouts and other functionalities.

```
public class MainActivity extends Activity {
    private Button signin;
    private Button register;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        signin = (Button) findViewById(R.id.signin);
        register = (Button) findViewById(R.id.register);
        // other relevant code goes here ...

    }

    public void onClick(View v){

    }

}
```

```
public class SignupActivity extends Activity {
    private Button submit;
    private Button cancel;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        submit = (Button) findViewById(R.id.submit);
        cancel = (Button) findViewById(R.id.cancel);
        // other relevant code goes here ...

    }

    public void onClick(View v){

    }

}
```

```
public class ListActivity extends Activity {
    private Button Logout;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list);
        logout = (Button) findViewById(R.id.logout);
        // other relevant code goes here ...

    }

    public void onClick(View v){

    }

}
```

Question 2 (For Sections BCS-6A and BCS-6B)**2 + 3 = 5 points**

(a) What shall be the output of the following Kotlin code:

```
var courseTitle: String? = null
print(courseTitle ?: "SMD")
```

(b) Consider you are developing an application in which you need to add “isPalindrome()” functionality to built in String class e.g.

1. “Anna”.isPalindrome() should return true
2. “SMD”.isPalindrome() should return false
3. “Madam”.isPalindrome() should return true

What is the most appropriate way to do this and **how** do we implement such an approach?

Question 2 (For Section BCS-8A)**5 points**

List three design patterns that are used in the Android Framework to facilitate UI Programming. Briefly explain the problem solved by each pattern in the context of Android Framework.