ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 3

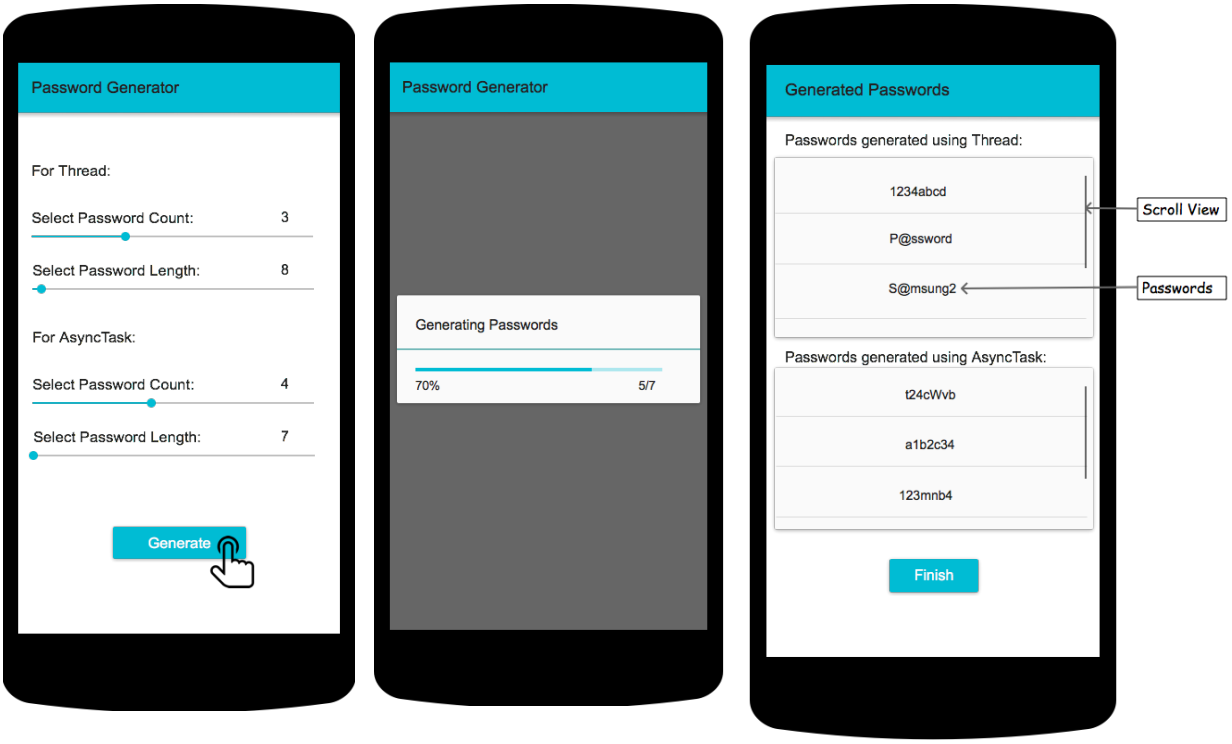
Basic Instructions

---

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Export your Android project and create a zip file which includes all the project folder and any required libraries.
5. Submission details:
   a. Only a single group member is required to submit on canvas for each group.
   b. The file name is very important and should follow the following format: **Group#_InClass03.zip**
   c. You should submit the assignment through Canvas: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 3 (100 points)

In this assignment you will get familiar with Android concurrency models. The app is a password generator app to help the user to choose strong passwords. The User selects how many passwords they want the app to generate, the length of each password and then chooses one of them. This application is composed of two activities, namely the Main Activity and GeneatedPasswords Activity.



(a) Main screen (b) Progress dialog (c) GeneratedPasswords Activity

Figure 1: Password Generator App

Part 1: Main Activity (75 points):

The interface should be created to match the user interface (UI) presented in Figure 1. You will be using layout files, and strings.xml to create the user interface. Perform the following tasks:

1. The main activity contains 4 SeekBars, 2 for Threads and 2 for AsyncTasks.
2. You are provided with a Util class that contains a static method getPassword(int length). The length is the length of the password to be generated. This method is expected to take long time to execute and returns a random String password. Import the provided Java file by simply dragging the file into the src folder under your project package in Android Studio.
3. For Threading:
   a. You should create a Thread Pool size of 2, and use the pool to execute your created threads.
   b. First 2 SeekBars are for Threads.

   i. Select Password Count SeekBar: this is used to set the number of passwords to be generated. The SeekBar minimum should be 1 and maximum 10. The TextView beside the seeker should show the selected SeekBar progress whenever the user changes the SeekBar progress (moves the SeekBar).
   ii. Select Password Length SeekBar: this Seek bar is used to set the password length. The SeekBar minimum should be 7 and maximum 23. The TextView beside the seeker should show the selected SeekBar progress whenever the user changes the SeekBar progress (moves the SeekBar).
4. For AsyncTask you will use AsyncTask instead of Threads.
5. Clicking on Generate button will start the execution for both the Threads and AsyncTasks in the backgrounds, and display a progress dialog which displays the progress of the Threads.
6. For Threads:
   a. When you press the Generate button it should start the execution of a background thread and return the list of all passwords (based on the selected count and length) by using the getPassword() method. For example, if the count was set to 5, the getPassword() method will run 5 times in the background thread, and return 5 passwords. The list of these 5 passwords should be returned to the main thread. While the passwords are being generated, display a ProgressBar indicating the progress, see Figure1(b).
   b. To be able to exchange messages between the child thread and the main thread use the Handler class. Either use messaging or setup a runnable message.
   c. The ProgressBar should not be cancelable. The ProgressBar should be dismissed after all the getPassword() calls are completed.
7. For AsyncTasks:
   a. It is similar to the tasks in Threads. You just need to do all the things using AsyncTasks here.
8. After all the passwords received in the main Thread, create a new intent to start GeneratedPasswords activity.

Part 2: GeneratedPasswords Activity (25 points):

In this activity, you should display the generated passwords. Perform the following tasks:

1. Create two ScrollViews to display passwords generated using Threads and AsyncTasks.
   a. ScrollView for passwords generated using Threads: list all the generated passwords using threads in a ScrollView and Dynamic Layout (Figure 1(c)). HINT: use LayoutInflater.
   b. Display another ScrollView for passwords generated using AsyncTasks (Figure 1(c)).

You need to implement Threads and AsyncTasks simultaneously in this assignment. Clicking on Generate button will start everything altogether.