

ITIS/ITCS 5180 Mobile Application Development  
In Class 10

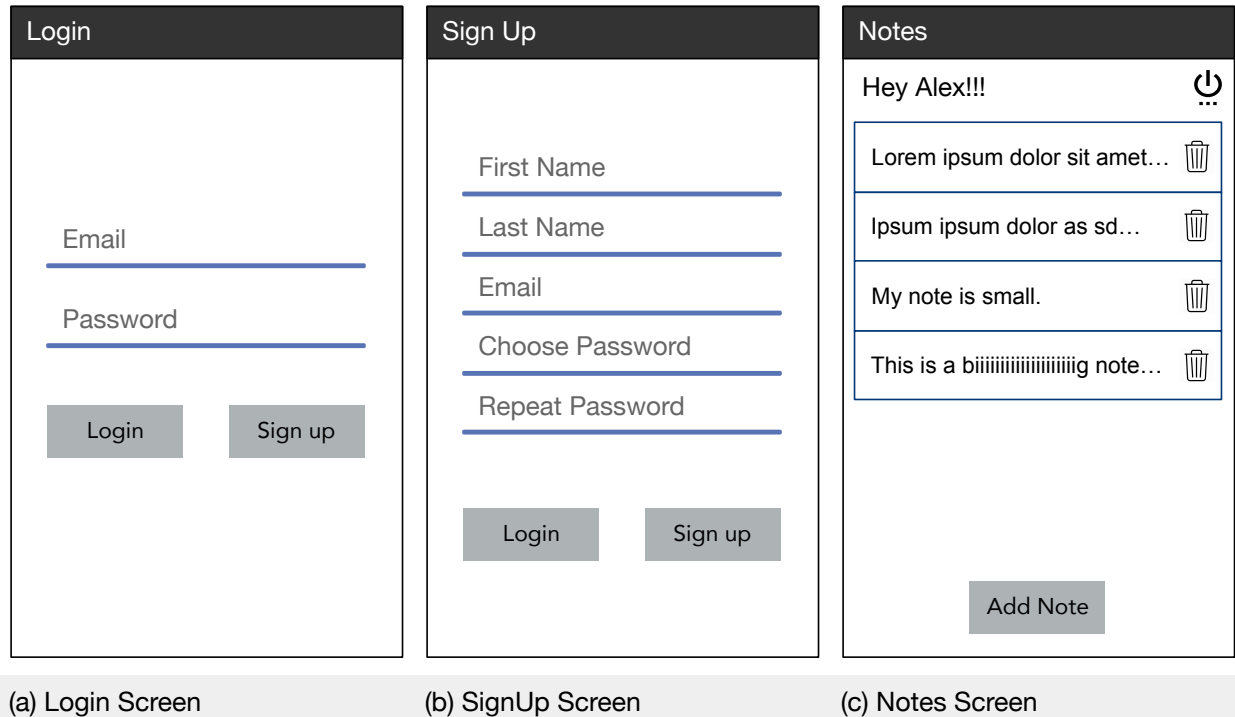
---

**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 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. Compress the contents of your project folder. The file name is very important and should follow the following format: **Group#\_IC10.zip**
  - b. Only one group member is required to submit on behalf of the whole group.
  - 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 10

In this assignment we will be using OkHTTP library to implement a simple note taker application, which uses an API hosted remotely. We will use the API to sign up, login, save new notes, get notes, delete the notes, and so on. We will use the Shared Preferences to save the web token locally, and manage the session.



**Figure 1: App Wireframes**

**API details:** Please extract the Postman collection file from the support files and import it to understand the structure of GET and POST messages to the API server. See the following table to understand the structure in details.

API endpoint	<a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/</a>
Authentication endpoint	<a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/">auth/</a>
Note endpoint	<a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/">note/</a>
Sign Up	POST, <a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/register">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/register</a>
Login	POST, <a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/login">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/login</a>
Post a new note	POST, <a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/post">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/post</a>
Get a note	GET, <a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/get">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/get</a>
Get all notes	GET, <a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/getall">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/getall</a>
Delete a note	GET, <a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/delete">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/note/delete</a>
Logout	GET, <a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/logout">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/logout</a>
Get user details	GET, <a href="http://ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/me">ec2-3-91-77-16.compute-1.amazonaws.com:3000/api/auth/me</a>

**Login and SignUp (30 points):**

If you are not logged in, or does not have a valid session token, the Login screen should be the startup screen of your app. Clicking on Sign Up button on Login screen should take you to the Sign Up screen. Sign Up screen should accommodate first name, last name, email, password, and repeat password, see Figures 1 (a and b). You should check if the passwords matches before you call the Sign Up API.

**After you sign up:**

1. You should receive an authentication token from the API response. Save the token to Shared Preferences. It will be valid for 24 hours.
2. It should start the notes screen.

**After you Login:**

1. Similar to the Sign Up API, Login API will return you an authentication token. Save it to Shared Preferences.
2. It should take you to the notes screen.

**Please Note: when the app starts, check if your session is still working. Get the previously saved token from Shared Preferences, use the token to get you to the notes screen. If there is no token stored or the token is no longer valid, load the Login screen.**

**Notes screen (50 points):**

The requirements are as follows:

1. Fetch the name of the user and display it on top, see figure 1(c).
2. Put a logout button beside the greetings.
3. If you click on the logout button, you should
  1. Call the logout API.
  2. Delete the stored token from Shared Preferences.
  3. Take the user to the Login screen.
4. Clicking on Add Note button should take you to the Add Note screen. You need to design the screen yourself.
5. Get all the current notes using the API, and display them. Make sure you do not display more than 100 characters of the note, see figure 1 (c).
6. Clicking on the delete button on a note, should use the API to delete the note from the server, and refresh the list.

**Add Note screen (15 points):**

This screen should include:

1. An EditText, able to accept up to 1000 characters.
2. A post button.
3. Clicking on Post button should use the API to add the note in the server.

**Display Note screen (5 points):**

Clicking on a note item on the list should open this screen. You need to design it yourself. It should include:

1. A TextView to display the whole note.
2. And a Close button to get back to the Notes screen.

**Rubrics:**

<b>Rubric</b>	<b>Points</b>
Login: OkHTTP request building	7.5
Login: fetch data and save token to Shared Preferences	7.5
SignUp: OkHTTP request building	8
SignUp: fetch data and save token to Shared Preferences	5
Notes Screen: Check SharedPreferences at startup, and manage login tokens	10
Notes Screen: OkHTTP Request building	5
Notes Screen: OkHTTP response management	10
Notes Screen: Managing ListView/RecyclerView/DynamicLayout	10
NotesScreen: Deleting a note (API + List)	15
Add a Note (Character limit, and others)	15
Display a Note	5
<b>Total</b>	<b>100</b>