

ITIS/ITCS 4180/5180 Mobile Application Development
In Class 09

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 lose 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#_InClass09.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 09 (100 points)

<div style="background-color: #00a0e3; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> Chit Chat 🔗 </div> <div style="padding: 10px;"> <p>Email <input style="width: 80%;" type="text"/></p> <p>Password <input style="width: 80%;" type="password"/></p> <p style="text-align: center;">Login</p> <p style="text-align: center; margin-top: 10px;">Or, Sign Up</p> <p>First Name: <input style="width: 80%;" type="text"/></p> <p>Last Name: <input style="width: 80%;" type="text"/></p> <p>Email <input style="width: 80%;" type="text"/></p> <p>Password <input style="width: 80%;" type="password"/></p> <p style="text-align: center;">Sign Up</p> </div>	<div style="background-color: #00a0e3; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> Chit Chat 🔗 </div> <div style="padding: 10px;"> <p>Your Subscribed Channels:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> App Development View </div> <div style="border: 1px solid #ccc; padding: 5px;"> Politics View </div> <div style="text-align: center; margin-top: 20px;"> 👍 Add more </div> </div>	<div style="background-color: #00a0e3; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> Chit Chat 🔗 </div> <div style="padding: 10px;"> <p>Your Subscribed Channels:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> App Development View </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Politics View </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Fun 👍 Join </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Channel x Join </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Channel y Join </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Channel z Join </div> <div style="border: 1px solid #ccc; padding: 5px;"> Channel a Join </div> <p style="text-align: center; margin-top: 10px;">Done</p> </div>
(a) Login/ Sign Up Screen	(b) Main Screen	(c) Add More Channels
<div style="background-color: #00a0e3; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> Chit Chat 🔗 </div> <div style="padding: 10px;"> <p>Your Subscribed Channels:</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> App Development View </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Politics View </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Fun 👍 View </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Channel x Join </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Channel y Join </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Channel z Join </div> <div style="border: 1px solid #ccc; padding: 5px;"> Channel a Join </div> <p style="text-align: center; margin-top: 10px;">Done</p> </div>	<div style="background-color: #00a0e3; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> Chit Chat 🔄 🔗 </div> <div style="padding: 10px;"> <p>Fun</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p><i>Ashley Jones</i></p> <p>Hello Everyone! Let's have some Fun!!</p> <p><i>Posted 30 min ago</i></p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p><i>John Williams</i></p> <p>Have you watched the Movie Logan?</p> <p><i>Posted 5 min ago</i></p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p><i>Alice Bob</i></p> <p>Hi Ashley! Nice to see you here!</p> <p><i>Posted now</i></p> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <input style="width: 70%;" type="text"/> Send </div> </div>	<div style="background-color: #00a0e3; color: white; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> Chit Chat 🔄 🔗 </div> <div style="padding: 10px;"> <p>Fun</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p><i>Ashley Jones</i></p> <p>Hello Everyone! Let's have some Fun!!</p> <p><i>Posted 35 min ago</i></p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p><i>John Williams</i></p> <p>Have you watched the Movie Logan?</p> <p><i>Posted 5 min ago</i></p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p><i>Alice Bob</i></p> <p>Hi Ashley! Nice to see you here!</p> <p><i>Posted now</i></p> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <input style="width: 70%;" type="text"/> Send </div> </div>
(d) View Channel	(e) Channel Screen	(f) Refreshed Screen

Figure 1: App Screens

In this assignment you will get familiar with using with HTTP connections, authentication, and implement an app to share messages.

The API details are provided in the Postman file that is provided with the assignment.

The main APIs are:

- The base URL: <http://52.90.79.130:8080/Groups/>
- LoginAPI: POST /api/login
- SignUp API: POST /api/signUp
- Subscriptions API: GET /api/get/subscriptions
- Channels API: GET /api/get/channels
- Get Messages API: GET /api/get/messages?channel_id=<CHANNEL_ID>
- Post Messages API: POST /api/post/message
- Subscribe Channel API: POST /api/subscribe/channel

Part A: Login

This is the launcher screen of you app. The wireframe for this screen is shown in Figure 1(a and b). The requirements are as follows:

1. The launcher Screen should be set to the Login Screen/ Main Screen. When the app first starts, the Login screen should check if there is a current user session, by checking if there is a valid token (stored in the shared preferences):
 - a) If there is a valid token, then start the Chats Screen and close the login screen.
 - b) If there is no current session, then the Login screen should be used to provide user login.
2. The user should provide their email and password. The provided credentials should be used to authenticate the user using the provided auth api. Clicking the “Login” button should submit the login information to the api to verify the user’s credentials.
 - a) If the user is successfully logged in then start the Chat Screen, and finish the Login Screen.
 - b) If the user is not successfully logged in, then show a toast message indicating that the login was not successful.

Part B: Sign Up

The Second part of the Login screen is the Sign Up, see Figure 1(a). Follow the instructions:

1. The user should provide their first name, last name, email, password and password confirmation. Clicking the “Sign Up” button should submit the user’s information to signup API to verify the user’s credentials.
 - a) If the signup API is not successful display an error message indicating the error message received from the api.
 - b) If the signup API is successful, then store the returned token in the shared preferences, and display a Toast indicating that the user has been created. Then start the Main Screen and finish the Login Screen.

Part C: Channel Screen

After Login/SignUp the user should be taken to the Main Screen. In the provided API there are multiple Channels the user can subscribe. It will display the channels the user has already subscribed to.

1. In the Main Screen, display the list of channels the user has already subscribed to.
2. On the Action Bar add a logout icon button. Clicking on the logout icon should finish the session, delete the token from the shared preferences, create a Toast, and take the user to the Login/SignUp Screen.
3. In the list of the subscribed channel, for each channel, there should be a View button. Clicking on the View button will take the user to that particular channel.
4. There should be a button named Add more. Clicking on Add more button will take the user to a new Screen to display all the available channels, see Figures 1(b and c).
5. In each of the item in the list, there should be the name of the channels and a button. You should display View button, if the channel is already subscribed. Or, display Join button for any other channels. You do not need to sort the list.
6. Clicking on the Join button should allow the user to subscribe the channel. Subsequently, make a Toast confirming the subscription, and change the button 's label from Join to View.
7. Clicking on Done button at the bottom of the screen should take the user to the main screen.

Part D: ChitChat Screen

Clicking on the View button from any screen should take the user to the Channel Screen, see Figures 1(e and f). Follow the instructions:

1. On the Action bar there should be two buttons, Refresh and Logout. Clicking on the Refresh button should refresh the screen to load latest posts.
2. Clicking on the Logout button should log the user out to the Login/SignUp Screen.
3. On the top of the screen, display the name of the channel. Then display a list of posts.
4. For each post, display the Full name of the User, Message, and time of posting. Use Prettytime library.
5. Below the list of posts, there should be a InputText for message the user wants to send to the channel.
6. Clicking on Send button should post the message to the channel, and display the current posts, see Figure 1(f).
7. Keep the posts by other users Left aligned, and posts by the user Right aligned.