

ITIS/ITCS 5180 Mobile Application Development In Class Assignment 09

Basic Instructions:

1. Please download the support files provided with the assignment and use them when implementing your project.
2. 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.
3. Export your Android project and create a zip file which includes all the project folder and any required libraries. The file name is very important and should follow the following format: **Group#_IC09.zip**. Submit the exported file using the provided canvas submission link.
4. **The required Android Virtual Device (AVD) should have minimum SDK version set to 25 and target SDK at 28+.**
5. **Failure to follow the above instructions will result in point deductions.**

In Class 09 (100 points)

In this assignment you will be using HTTP connections, authentication, and implement an app to send e-mails. The API details are provided in the Postman file that is provided with this assignment. For authentication you need to pass the token returned from login api as part of the header as described in the Postman file. Use OkHttp and Shared Preferences to implement the application.

<div style="background-color: #333; color: white; padding: 5px; text-align: center;">Mailer</div> <div style="padding: 20px;"> <div style="margin-bottom: 10px;"> <input style="width: 90%;" type="text" value="Email"/> </div> <div style="margin-bottom: 10px;"> <input style="width: 90%;" type="password" value="Password"/> </div> <div style="text-align: center; margin-top: 10px;"> <div style="background-color: #ccc; padding: 5px 20px; margin: 2px;">Login</div> <div style="background-color: #ccc; padding: 5px 20px; margin: 2px;">Sign Up</div> </div> </div>	<div style="background-color: #333; color: white; padding: 5px; text-align: center;">Sign Up</div> <div style="padding: 20px;"> <div style="margin-bottom: 10px;"> <input style="width: 90%;" type="text" value="First Name"/> </div> <div style="margin-bottom: 10px;"> <input style="width: 90%;" type="text" value="Last Name"/> </div> <div style="margin-bottom: 10px;"> <input style="width: 90%;" type="text" value="Email"/> </div> <div style="margin-bottom: 10px;"> <input style="width: 90%;" type="password" value="Choose Password"/> </div> <div style="margin-bottom: 10px;"> <input style="width: 90%;" type="password" value="Repeat Password"/> </div> <div style="text-align: center; margin-top: 10px;"> <div style="background-color: #ccc; padding: 5px 20px; margin: 2px;">Sign Up</div> <div style="background-color: #ccc; padding: 5px 20px; margin: 2px;">Cancel</div> </div> </div>	<div style="background-color: #333; color: white; padding: 5px; text-align: center;">Inbox</div> <div style="padding: 20px;"> <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 10px;"> <div style="display: flex; align-items: center;"> <div style="text-align: center;"> Bob Smith </div> </div> <div style="font-size: 20px;">➔</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 1st email... 🗑 </div> <div>Oct 12, 2018</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 2nd email... 🗑 </div> <div>Oct 12, 2018</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 3rd email... 🗑 </div> <div>Oct 11, 2018</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 4th email... 🗑 </div> <div>Oct 10, 2018</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 5th email... 🗑 </div> <div>Oct 10, 2018</div> </div> </div>
(a) Login Screen	(b) Sign Up Screen	(c) Inbox/ Home Screen
<div style="background-color: #333; color: white; padding: 5px; text-align: center;">Inbox</div> <div style="padding: 20px;"> <div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 10px;"> <div style="display: flex; align-items: center;"> <div style="text-align: center;"> Bob Smith </div> </div> <div style="font-size: 20px;">➔</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 1st email... 🗑 </div> <div>Oct 12, 2018</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 2nd email... 🗑 </div> <div>Oct 12, 2018</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 3rd email... 🗑 </div> <div>Oct 11, 2018</div> </div> <div style="margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> Subject of the 4th email... 🗑 </div> <div>Oct 10, 2018</div> </div> <div style="height: 40px; border: 1px solid #ccc; margin-top: 10px;"></div> </div>	<div style="background-color: #333; color: white; padding: 5px; text-align: center;">Create New Email</div> <div style="padding: 20px;"> <div style="margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> Send to: <div style="border: 1px solid #ccc; padding: 2px 10px; flex-grow: 1;">User</div> <div style="font-size: 12px;">▼</div> </div> </div> <div style="margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> Subject <div style="border: 1px solid #ccc; padding: 2px 10px; flex-grow: 1;"></div> </div> </div> <div style="margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> Message <div style="border: 1px solid #ccc; padding: 10px; flex-grow: 1; min-height: 100px;"></div> </div> </div> <div style="text-align: center; margin-top: 10px;"> <div style="background-color: #ccc; padding: 5px 20px; margin: 2px;">Send</div> <div style="background-color: #ccc; padding: 5px 20px; margin: 2px;">Cancel</div> </div> </div>	<div style="background-color: #333; color: white; padding: 5px; text-align: center;">Create New Email</div> <div style="padding: 20px;"> <div style="margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> Send to: <div style="border: 1px solid #ccc; padding: 2px 10px; flex-grow: 1;">User</div> <div style="font-size: 12px;">▼</div> </div> </div> <div style="margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> Subject <div style="border: 1px solid #ccc; padding: 2px 10px; flex-grow: 1;"> <div style="font-size: 10px; border-bottom: 1px solid #ccc; padding: 2px 5px;">Firstname Last</div> <div style="font-size: 10px; border-bottom: 1px solid #ccc; padding: 2px 5px;">Ujwala Sunny</div> <div style="font-size: 10px; border-bottom: 1px solid #ccc; padding: 2px 5px;">Tom Hunter</div> <div style="font-size: 10px; border-bottom: 1px solid #ccc; padding: 2px 5px;">My Name</div> <div style="font-size: 10px; border-bottom: 1px solid #ccc; padding: 2px 5px;">Their Name</div> <div style="font-size: 10px; border-bottom: 1px solid #ccc; padding: 2px 5px;">No Name</div> </div> </div> </div> <div style="margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between;"> Message <div style="border: 1px solid #ccc; padding: 10px; flex-grow: 1; min-height: 100px;"></div> </div> </div> <div style="text-align: center; margin-top: 10px;"> <div style="background-color: #ccc; padding: 5px 20px; margin: 2px;">Send</div> <div style="background-color: #ccc; padding: 5px 20px; margin: 2px;">Cancel</div> </div> </div>
(d) Deleted One Email	(e) Create New Email	(f) Users Drop Down Spinner

Figure 1: App Wireframes

Part A: Login (15 points)

API	Description	Method	Requires Authentication
/api/login	Login	POST	No
/api/signup	Signup	POST	No
/api/inbox	Get all the emails	GET	Yes
/api/users	Get registered users	POST	Yes
/api/inbox/add	Send an email	POST	Yes
/api/inbox/delete/{id}	Delete email from inbox	GET	Yes

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

1. The user should provide their email and password. The provided credentials should be used to authenticate the user using the provided login 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, finish the Login Screen, and save the token in Shared Preferences.
 - b) If the user is not successfully logged in, then show a toast message indicating that the login was not successful.
2. Clicking the “Sign Up” button should start the Signup Screen Figure 1(b), and finish the login Screen.

Part B: SignUp (15 points)

Create the Signup screen to match Figure 1(b), with the following requirements:

1. Clicking the “Cancel” button should finish the Signup Screen and start the Login Screen.
2. The user should provide their first name, last name, email, password and password confirmation. Perform the required validation(the given password and the repeated password must match). Clicking the “Sign Up” button should submit the user’s information to signup API.
 - 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 Chat Screen and finish the Signup Screen.

Part C: Inbox (40 points)

This is the home screen which displays the list of emails as shown in Figures 1 (c, d, and f) Please follow the instructions below:

1. There should be two ImageViews/Buttons: (i) Create New Email, and (ii) Logout.
2. Clicking on Create New Email button (top-left at Figure 1 (c)) should take you to Create New Email screen.
3. Clicking on the Logout button (top-right) should log you out and take you to the Login Screen.
4. In between two ImageButtons, there should be a TextView to display the username.
5. The email list should be implemented using RecyclerView. Each item in the list should have three things:
 1. TextView with Subject. It should display only the first line. If the sentence exceeds the limit of the line, end it with three dots. See figure 1(c). HINT: use `android:maxLines="1"`
 2. TextView with DateTime. Display the date when the email was created.
 3. An ImageView/Button with Delete icon.
6. Clicking on any of the item in the RecyclerView should open the Display Mail Screen.
7. Clicking on the Delete ImageView should call the Delete API and delete the particular message from Inbox.

Part D: Create New Email (25 Points)

Clicking on Create New Email ImageView should take you to Create New Email Screen. Please follow the instructions:

1. You should get the users using the Users API, and add them in a Spinner (Drop Down). Every item in the Spinner should display "FirstName LastName."
2. You have to select the user you want to send the email to.
3. Next, you need to display an EditText to get the subject.
4. And, display a multiline EditText for the message.
5. Finally, a Send button to send.
6. Clicking on the Send button should call Add API and send the message. Display a Toast saying that the message is sent, and finish the activity.
7. Clicking on Cancel button should finish the activity.

Part E: Display Mail (5 Points)

Design the UI yourself that contains:

1. Sender Name.
2. Subject.
3. Message.
4. Created at.
5. A Close button to finish the Activity.

Check Assignment Description for Rubrics!

Rubrics

Criteria		Points
Part A:	Verify User Credentials using API and Toast for Unsuccessful attempt.	7
Part A:	Save the token in Shared Preferences	5
Part A:	Logic for SignUp Activity	3
Part B:	Validation	7.5
Part B:	Call API to validate and Toast for error message.	5
Part B:	Save token and Toast about the user created.	2.5
Part C:	Get Username and display it in TextView	2.5
Part C:	Logout ImageButton to logout, and finish the activity.	2.5
Part C:	Logics for Create New Email ImageButton	2.5
Part C:	Setting up RecyclerView	15
Part C:	Single line Subject in RecyclerView item	2.5
Part C:	Display Date in proper format	2.5
Part C:	Delete button Logics (API calls and handling the RecyclerView)	7.5
Part C:	RecyclerView logics for clicking on a item to open Display Mail.	5
Part D:	Call users API and generate drop down Spinner.	10
Part D:	Multiline EditText	5
Part D:	Implement Send button and call inbox/add API to send the email	7.5
Part D:	Implement Cancel button functionalities.	2.5
Part E:	Display Mail Activity	5
	Total	100