

ITIS/ITCS 5180 Mobile Application Development  
In Class Assignment 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. **The minimum SDK version should be set to 20, and the target SDK version should be set to 25.**
7. **Failure to follow the above instructions will result in point deductions.**

### In Class Assignment 09 (100 points)

**Chat Room**

Email

Password

Login Sign Up

(a) Login Screen

**Sign Up**

First Name

Last Name

E mail


Choose password

Repeat password

Cancel Sign Up

(b) Sign Up Screen


**Message Threads**

**Bob Marley** 

Current Threads


In Class Discussions

Homework Discussions

Add New Thread 

(c) Message Threads Screen


**Message Threads**

**Bob Marley** 

Current Threads


In Class Discussions

Homework Discussions

Final Exam 

(d) Adding a new message thread


**Message Threads**


**Bob Marley** 

Current Threads

In Class Discussions


Homework Discussions

Final Exam 

Add New Thread 


(e) Added a new thread, can delete your own thread

**Chatroom**

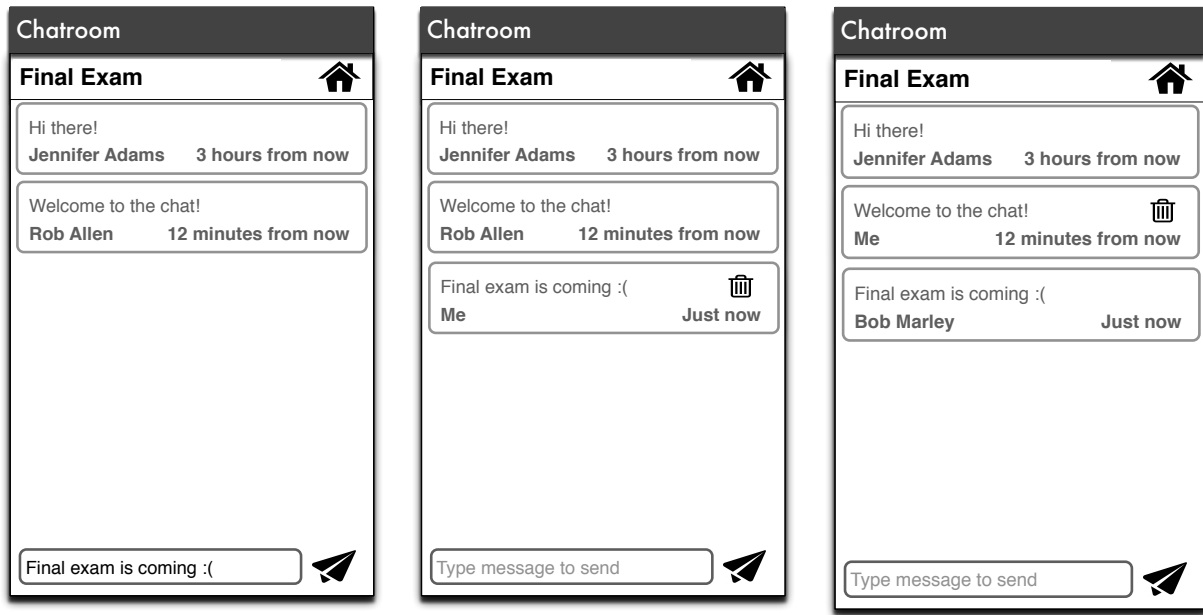
**Final Exam** 

Hi there!  
Jennifer Adams 3 hours from now

Welcome to the chat!  
Rob Allen 12 minutes from now

Type message to send 

(f) Final Exam thread



(g) Bob is posting a message

(h) Message posted, can delete your own message

(i) View of Rob's Appscreen

Figure 1: App Wireframes

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 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.

API	Description	Method	Requires Authentication
/api/login	Login	POST	No
/api/signup	Signup	POST	No
/api/thread	Get all the threads	GET	Yes
/api/thread/add	Add a new thread	POST	Yes
/api/thread/delete/{thread_id}	Delete a thread	GET	Yes
/api/messages/{thread_id}	Get all message in a thread	GET	Yes
/api/message/add	Add a new message to a thread	POST	Yes
/api/message/delete/{message_id}	Delete a message	GET	Yes

### **Part A: Login (10 points)**

This is the launcher screen of your 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, 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.
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 the 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: Message Threads (35 points)**

This is the home screen which displays the list of threads as shown in Figures 1(c-e). Please follow the instructions below:

1. The top of the screen should display the name of the currently logged in user. The logout button is located beside the display name.
  - Clicking on the 'logout' icon should log the user out (delete the token), finish the screen, and open the login screen.
2. The list should display the list of message threads. Which should be retrieved using the thread api.
  - Clicking on a thread item should open the Chatroom screen of that thread.
3. The EditText at the bottom of the screen should enable the user to create a new thread.
  - After entering the new thread message, the user clicks the add button, which should send the new thread information to the add new thread api.
  - After a successful addition of a new thread, the list should be updated to show the updated thread list.
4. The user should be allowed to delete threads that they have created.

- Display a remove icon beside the title of the threads that were created by the current user. Note: the thread information includes the user\_id of the user that created the thread.
- Clicking the remove icon should delete the thread by communicating with the delete thread api. Then the list should be updated to display the updated list of threads.

#### **Part D: Chatroom (40 points)**

This activity displays the list of messages in the particular thread. Also this screen allows the user to add new text messages. The requirements are as follows:

1. The top of the screen should display the name of the thread as shown in Figure 1(f). Beside the name of the thread there is a home icon button, when clicked should close this activity and display the Message Threads screen.
2. For each message display the full name of the user that created the message, the message text, and the time the message was created using the Prettytime Library.
  - Display the delete icon only for messages that were posted by the currently logged in user.
3. The EditText at the bottom of the screen should enable the user to create a new message.
  - After entering the new message, the user clicks the add button, which should send the new thread information to the add new message api.
  - After a successful addition of a new message, the list should be updated to show the updated message list.
4. The user should be allowed to delete messages that they have created.
  - Clicking the remove icon should delete the message by communicating with the delete message api. Then the list should be updated to display the updated list of message.

**Good Luck!**