

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)

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

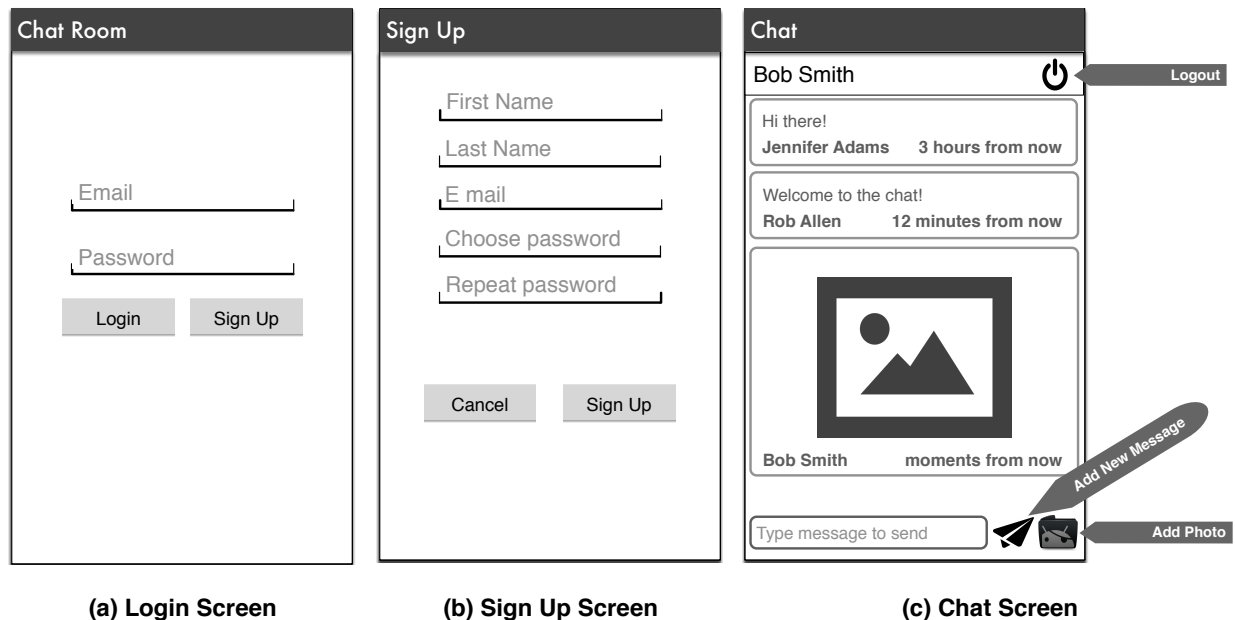


Figure 1: App Wireframes

The API details are provided in the Postman file that is provided with this assignment. The main APIs are :

- Login API: POST /api/login
- Signup API: POST /api/signup
- Add File API: POST /api/file/add **(Requires Access Token)**
- Get File API: GET /api/file/{id} **(Requires Access Token)**
- Get Messages List API: GET /api/messages **(Requires Access Token)**
- Add New Message API: POST /api/message/add **(Requires Access Token)**

Part A: Login (20 points)

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

1. The launcher Screen should be set to the Login Screen/ 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.
- 3. Clicking the “Sign Up” button should start the Signup Screen and finish the login Screen.

Part B: SignUp (20 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. 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 Chat Screen and finish the Signup Screen.

Part C: Chat (60 points)

The screen should show the list of messages retrieved from the server. Also this screen allows the user to add new text and image messages. The requirements are as follows:

- 1. Display the list of messages received as show in Figure 1(c). Note that list if messages shows both text and image messages.
 - 1. In order to display the image files, you should call the get file api.
 - 2. Display the time the message was created using the **prettytime Library**.
 - 3. Display the first and last name of the user that created the message.
- 2. When the user types a text message and clicks the “Add New Message” icon, it should call the Add message api to store the new message. Then should reload the list to display the latest messages.
- 3. When the user clicks on the “Add Photo” button, it should open the image gallery for the user to select the image to be sent to the server by calling the following sequence:
 - 1. First the image file should be stored on the server using the add file API, which should return the file id. In addition to the token you need to send the MIME type of the file to the api. (Check Postman)
 - 2. Second the returned file id should be sent to the add message api to create a new message that contains the new file id.
- 4. Clicking the top logout button should logout the user by performing the following:
 - 1. Delete the token from the shared preferences.
 - 2. Show the login screen
 - 3. Finish the chat screen.