AMERICAN UNIVERSITY OF BEIRUT
MAROUN SEMAAN FACULTY OF ENGINEERING & ARCHITECTURE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
EECE 350 - Computer Networks

Network Programming Project
YallaChat

Project Guidelines

- Students are required to work in groups of four.
- Each group is required to work on the project on their own. Any copying will be counted as cheating and treated accordingly.
- Group members are expected to put balanced effort on the development of the different requirements of the project.
- Designate one member of the group to submit the project deliverables (code and report) to Moodle.

Deliverables

- Source code of the project with documentation.
- Up to 6-page report and two appendices, as per the following:
  - Cover page listing the group members (name and ID number) and the workload distribution (in percentage of individual effort to total effort).
  - The report is divided into two main parts, each detailing one of the project phases.
  - Both phases will be submitted at the same time, but it is expected that you finalize the first phase in two weeks, leaving ample time to implement the second phase.
  - Description of the system architecture and the protocol used between the communicating entities, whether in the client-server mode or in the hybrid mode.
  - A tabular presentation of all the project features indicating the ones that are successfully implemented.
  - Description of the implementation of the different functionalities.
  - Description of an interesting future networking project.
  - One appendix that includes snapshots of the application depicting the main features.
  - One appendix that includes a table showing a breakdown of the project tasks. Indicate next to each task the name of the group member who was mainly responsible for its implementation.
- You are expected to perform a demo to present the application and all supported features.
- You should be ready to answer any question related to the implementation of the project.

General Description

In this project, you are required to design and implement a serious competitor of **WhatsApp**. Your **YallaChat** platform uses Python as the programming language and is based on a simple yet effective design. The platform starts with a very simple social media design that provides basic features of adding friends and exchanging text messages with them. You then work on improving this platform gradually starting from communicating with a group of friends and exchanging multimedia messages, to upgrading to a more scalable and efficient architecture, and finally enabling high interactions among your **YallaChat** users.
For the purpose of your application, you need to decide whether TCP or UDP is to be used as the underlying transport layer protocol. Please note that you may select any protocol as long as you have enough reasons to justify your choice. You start developing your platform as a basic social media application based on a client server architecture. The clients, serving as users of the platform, create user accounts and log in to the centralized server. They can view the list of users on the server, select and add a friend, and send messages to the selected friend. The server receives the messages and sends them immediately to the selected friend in case the friend is online. Otherwise, the server saves the messages in its repository and delivers them once the friend gets online. For the communication between the client and the server, you need to clearly define your application protocol to be correctly implemented. You then modify your code to include multithreaded server to serve multiple users simultaneously. After successfully implementing the aforementioned, you can focus on improving your **YallaChat** platform and supporting more advanced features that are considered the second phase of the project. In this second phase, the application has to be extended to utilize peer-to-peer architecture and allow friends to create and join groups and react to exchanged messages. These messages can be text, voice, pictures, and videos. A searching functionality can also be added to look for messages related to a given theme.

Implementation

I. Basic YallaChat (80 pts)

Client - Design and implement the client side of your **YallaChat** platform that has the following functionalities:

- The client has a graphical user interface (GUI).
- The user opens the client application and connects to the server application using the server domain name and port number.
- The user is given the option to register or to login.

- If the user is not registered, they can sign up and provide the server with name, email address, username, and password.
- In the case of a registered user, the user logs in with their existing account where the server performs authentication by verifying the username and password.
- When the user is authenticated, they get a list of chats they have made with friends or group of friends. The list should be displayed in the client GUI. There is no need for the client to get a list of all its messages, it only needs to get just enough messages to be displayed on the screen.
- The user can add friends and view whether they are online or offline
- The user can initiate communication with a selected friend and exchange text messages through the server.

Server - Design and implement the server side of the **YallaChat** platform that has the following functionalities:

- The server takes as a command line argument the port number on which it would be listening.
- The server allows the users to register and login.
- Every time a new user signs up for a new account, the server adds a new user.
- Every time a user logs in, the server authenticates them by verifying their username and password.
- The server application manages a database that stores user accounts.
- If a user sends messages to an offline friend, the server keeps these messages in its repository and delivers them immediately once the friend goes online
- If the friend is online, the server immediately delivers messages sent to it and it does not need to save the messages in its repository.
- The server serves multiple clients simultaneously.

II. Extending YallaChat to support advanced features (20 pts)

In this phase, you are asked to modify your system to implement some advanced and creative features. Below is a list of possible features, some of which are considered as required while the rest are optional. You are, however, strongly encouraged to come up with a creative feature and implement it.

- *(required)* Users can create group of friends and communicate with them
- *(required)* Users can exchange different types of multimedia messages including text, voice, pictures, and video
- *(required)* Upgrade your system to support a hybrid client/server and peer-to-peer architecture. Once a user selects an online friend to communicate with, the server shares with the user the friend's address so they can communicate with each other in a peer-to-peer manner
- *(required)*In the case of group communication, consider two options and select one to be implemented in your project by justifying your choice.
  (a) The first option is to let the server receive a message from one user and push it to all friends in the group once the messages arrive to the server.
  (b) The second option is to have the user who sends a new message sends it to all its friends in the group by establishing a peer-to-peer connection with them.
- *(optional)*Users can request to visit selected chats of their friends. If approved by the friend, a user can go over the chats then leave. The list of visitors to a given chat and the time of their visits can be viewed by all concerned users.
- *(optional)* The platform supports efficient searching based on a specified topic.
- *(required)* An additional creative feature of your choice.