The University of Jordan
Faculty of Engineering and Technology
Department of Computer Engineering

Programming of Networks Protocol
Course Final Project
Eng. Asma Abdel Karim

In this course project, you are required to design and implement a messaging system using a client/server model and the Java API according to the following details:

1. The client side:

Your application is intended to allow clients to send messages to each other such that each client can send and receive messages. The client must have access to two types of messages groups accordingly:

- Inbox: that consists of received messages.
- Sent: that consists of sent messages.

Each message must have the name of the sender, the receiver, a title field, a content, the date/time it was sent, and whether it is read or not.

In order to be able to use the application, the client must have an account by signing up the first time he uses the application with a username and password. The user uses his username and password then to login in each time he uses the application.

When the client application starts, the application must ask the user whether he wants to sign-up or log-in.

- If the user selects to sign-up, it must read his username and password and send them to the server to register him. The server may respond with successful sign-up, or may respond with failure if the username is already taken. Your application must inform the user that the username is taken and ask him to re-enter the username and password in this case. Once the user is signed-in successfully, he must be informed of that, and he is returned to the first two options (sign-up or log-in).
- If the user selects to log-in, it must ask him to enter the username and password and send them to the server. If the server responds that they are accepted then the user is logged-in. If the server responds with a log-in failure, this could be either because the username is not found, or the username is incorrect. The user must be informed of the failure reason and he must be returned to the first two options (sign-up or log-in).

When the client is logged-in successfully, the following main menu of options must be displayed to him:

- Send a message.

  If the user selects this option, a list of the registered users must be displayed to him. He must select the recipient name and enter his message. The message must be placed in the recipient inbox. The client may get an error message from the server if the recipient username is not found, in this case the client must re-enter a valid recipient name.

- View messages.

  If the user selects this option, he must be prompted whether he wants to view inbox messages or sent messages.

  - If he selects inbox messages, a list of the messages in the inbox including the name of the sender, the title, the date/time, and whether it is read or not must be displayed. The user can then select a message to display its contents. A message is considered read if the user displayed its contents at least once.
  - If he selects sent messages, a list of the messages in the sent group including the name of the recipient, the title, the date/time, and whether it is read or not (by the recipient) must be displayed. The sender can then select a message of the sent messages to display its contents.
- Log-off: if the user selects this option, the connection to the server must be closed and the application must terminate.

  For all the previous options, the user must be able to return to the main menu as long as he is logged in.

2. The server side:

The server application is where the clients connect to send and view their messages. The server must be able to handle several clients at the same time. It must also save info of all registered clients and their messages (even when the server application terminates).

When a client is first connected to the server, the server must receive which operation the client wants to perform, sign-up or log-in:

- If the operation received is to sign-up, it must receive the username and password, check if there is another user with the same name. If there is another user with the same name it must respond with a failure and get back to receive a requested operation. If there is no other user with the same name, it must register the user and respond with successful sign-up.
- If the operation received is to log-in, it must receive the username and password and search the registered users. If the user is found, it must respond with successful log-in. IF the username is not found or the password is incorrect, it must respond with the error to the client and get back to receive a requested operation.

When the client is logged-in successfully the server must keep on receiving the requested operation of the clients' main menu and respond accordingly:

- If the received operation is to send a message. It must read all the message info from the client (title, content, recipient name, and date) and place it in the requested recipient inbox. If the requested recipient username is not found, the server must return an error message to the client and receive the message info again.
- If the received operation is to view messages:
  - If the selected messages to be viewed are the inbox messages, the server must return a list of the inbox messages to the client. If the client selects a certain message to display its contents, the server must respond with the message and mark it as read if it was unread.

- If the selected messages to be viewed are the sent messages, the server must return a list of the sent messages to the client. If the client selects a certain message to display its contents, the server must respond with the message.

Your final submission must include the following:

1. Your code fully documented.
2. A report that includes the UML class diagram showing the design of your application data and methods.