Computer Networks and Internet 1 – Final Project

Author: Tal Glikin 326304292

Technical Part

I'm using Windows 10 operating system, and Python 3.11.

I have created several files working on the app:
A repository called "Project_dbs", inside it there are 3 files:
1. a CSV file called "users_db" that saves info regarding all the users – three columns in the CSV are "username", "password" (which is being hashed before being inserted in the database), and "role" that specifies whether they are admins or regular users.

2. a CSV file called "groups_db" that saves info regarding all the groups – three columns in the CSV are "GroupName", "GroupID" and "Participants" (which is a list that saves all the users connected to the group, every element in the list is a dictionary with username, his IP and his port).

3. another file that is called "messages" and saves all the messages in each group – every CSV file in "messages" is a CSV associated with the group and called "GroupName_GroupID" – inside it in one column every message that was sent in the group by every user is being saved there.

Theoretical Questions

5.1 Security

1. How can you enhance the security of your application? Discuss potential vulnerabilities and propose solutions.

Answer: There are many ways to enhance the security of the application like Implementing strict input validation to prevent attacks like command injection or limiting the number of login attempts to prevent brute-force type attacks.

2. What are the risks associated with transmitting messages in plain text? How might you implement encryption to address this concern?

Answer: The risk in transmitting messages in plain text is that they can be intercepted during transmission by attackers who can either steal the sensitive data or change and alter it before it reaches the other side.

5.2 Scalability

1. How would you design your application to handle a large number of concurrent users? Discuss potential bottlenecks and scalability challenges.

Answer: An app that has a lot of users using it requires a database that can handle a large volume of data so Database scalability, the app's resource management, and the network latency in case of a lot of users can all be a bottleneck.

2. What strategies could be employed to distribute the load and balance the server's resources effectively?

Answer: Monitoring the server's performance and resource management in the background and limiting the traffic (messages being sent, users logging in, etc) accordingly can balance the server's resources effectively.

5.3 Reliability and Fault Tolerance

1. Describe how you would ensure the reliability of your application. What measures can be taken to handle server failures or unexpected crashes?

Answer: To prevent server failures and unexpected crashes I would include a lot of error handling in the code wherever it's needed, make sure the application can run properly even when a certain service fails (meaning the application isn't dependent on a single feature) and maybe even set an automatic restart to the server if it fails for some reason.

2. How might you implement message persistence to ensure that messages are not lost even if the server restarts?

Answer: I can implement message persistence by creating a database for messages sent in each group – that way even when the server restarts, the messages will still be stored on that database.

5.4 User Authentication

1. Explain the importance of user authentication in your application. What methods could be used to authenticate users securely?

Answer: User authentication ensures that only authorized users can access the system and helps preserve privacy for each user. Using a three-way authentication (2FA) with an email or authenticator app or even biometric factors like fingerprint are some ways to authenticate users securely.

2. How would you handle user registration and password management to enhance the overall security of the system?

Answer: As I mentioned in the previous question, F2A, and saving passwords in a database only after using strong cryptographic hashing algorithms are two ways – more ways include requiring strong passwords that can't be easily guessed, limiting the number of login attempts, and more.

5.6 Protocol Design

1. Explain the choice of communication protocol for your application (e.g., TCP, UDP). What factors influenced your decision?

Answer: I chose TCP for my application because it is far more reliable than UDP and ensures that messages will arrive, and in the right order – which is crucial for a group chat application.

2. Explain about the socket handshake – which protocol you used, what are the commands, are they blocking commands. How would you design the message format and protocol for communication between the client and server?

Answer: in my code I used TCP – a protocol that does a three-way handshake, in this protocol the client initiates the connection by sending an SYN (Synchronize) packet to the server, and the server sends back an SYN-ACK (Synchronize-Acknowledge) packet to the client and finally, the client sends an ACK (Acknowledge) packet to the server, confirming the connection establishment.
the typical (blocking)TCP commands that I used are connect, send, and recv.

5.7 Message Ordering and Delivery

1. How would you address the issue of message ordering and delivery in your application? Discuss potential scenarios and solutions.

Answer: Sometimes messages may not arrive in the right order they were sent in – a solution to this is using time stamps or sequence numbers to know the order of messages at any time. Also, messages may arrive wrong or not arrive at all – and for that, we use the TCP protocol.

2. What mechanisms could be employed to ensure that messages are delivered in a timely and reliable manner?
Answer: TCP provides features such as acknowledgment, retransmission, and sequencing, ensuring that messages reach the destination reliably.

5.8 Persistent Storage

1. Explain the importance of persistent storage in your server. How did you design and implement the storage on the server?

Answer: Having persistent storage that preserves user data through server restarts is important because this information is vital for the working of the app and the users themselves, and persistent storage prevents this information from being erased.

2. Discuss the trade-offs between different storage solutions in the context of your application.

Answer: There are many storage solutions and each of them has its advantages and disadvantages so there are a lot of trade-offs between Simplicity, Flexibility, complexity, built-in querying capabilities, and many more. In my project, I chose to use CSV files because my app doesn’t require a big database or too complex querying.