Project Synopsis:

Title: Decentralized Communication: Building a Peer-to-Peer Chat Application

Team Name: NetTalk

Team Members:

- Brianna Gurrola
- Marissa Rascon-Arriaga
- Sebastian Ochoa

Introduction:

There's a significant hazard in data breaches, privacy invasions and server outages when there's a dependency on centralized servers. Our team project has a focus on a development of a peer to peer chat application, which will create a central server to have users communicate with one another directly. Our decentralized strategy will aim to strengthen communication systems against various threats.

Project Objectives:

- Decentralization: We aim to establish a serverless communication system where every device will operate as both client and server.
- Security: For a private and secure environment we plan to include secure hash algorithms (SHA) and advanced encryption standards(AES).
- File Sharing: Our app will feature drag-and-drop functionality for file sharing utilizing both TCP for reliability and UDP for faster transmissions, adaptable to both file sizes and network conditions.

Detailed Tasks:

1. Architecture:
    - We will use a Distributed Hash Table to create a more scalable P2P network that is effective for message routing.
    - Set up protocols for establishing and verifying connections using public-key infrastructure.
2. Communication Protocols:
    - Use Python socket programming to manage data transfers using TCP and UDP.
    - Develop a dynamic switching mechanism between TCP and UDP according to network performance and data type.
3. Security:
    - Implement end-to-end encryption using AES-256.
    - Use SHA-256 for hashing passwords and other authentication data.

    - Establish a digital signature system to confirm the integrity and authenticity of the messages and files that are sent.

4. User Authentication and Management:
    - Design a registration and login system using secure credentials stored on the user’s device.
    - Implement a friend request and approval framework to securely handle contacts within the network.
5. File Transfer Protocol:
    - Develop a protocol for both direct and relayed file transfers, ensuring dependability with failed direct connections.
    - For maximum bandwidth we will incorporate file compressions and decompression techniques.
6. Interface Development:
    - Create a user-friendly interface using frameworks like PyQt or Electron, that supports real-time messaging and notifications.
    - Include features such as message timestamps, read receipts, and typing indicators.
7. Testing and Quality Assurance:
    - Conduct unit and integration testing for every module.
    - Perform stress tests to evaluate system performances.
    - Engage in peer testing to refine the application based on user input.
8. Deployment and Maintenance:
    - Prepare the application for deployment across various operating systems.
    - Create a continuous feedback loop for improvements and updates.

Timeline:

- Weeks 1-2: Research and finalize the network architecture and communication protocols.
- Weeks 3-5: Develop and test security features.
- Weeks 6-8: Implement and test messaging and file sharing functionalities.
- Weeks 9-10: Develop the user interface.
- Weeks 11-12: Final testing, adjustments, and preparation.

Conclusion: Our team’s goal is to transform the way we think about digital communication by ensuring that conversations are not just secure and private, but also robust against central failures. This project is a step forward to providing users dependable communication skills as well as giving us hands on practice.