Names: Ali Jalil, Joshua Egwaikhide, Nick Chalardsoontornvatee

Class Descriptions:

1. **User:**
    a. Represents a participant in the messaging system. It has attributes like id, name, and isAuthenticated, which tracks the authentication status. Methods allow the user to compose, encrypt, decrypt, and authenticate messages.
2. **Message**:
    a. Represents a message created by the user, with attributes such as content, timestamp, senderId, and receiverId. This class acts as a container for messages in plain text, stored on the user's client.
3. **EncryptedMessage:**
    a. Holds encrypted content, along with timestamps and sender/receiver IDs. Like Message, it serves as a data container. Unlike Message, this class represents messages in their encrypted form and is transferred to the server for routing to the intended recipient.
4. **Server:**
    a. Manages user authentication and message transport. It has a collection of active users and methods like authenticateUser, transportMessage, and routeMessage. The server also includes a private method, synchronizeMessages, which ensures that messages are consistent across users and that all sent messages are received by the correct user upon their next login.

Relationships Summary:

a) **User and Message/EncryptedMessage**:
    i) Users create and interact with Message and EncryptedMessage objects. They use these classes to compose messages, address them to specific recipients, and then encrypt the message to ensure that only authorized users can access its content.

b) Server and User/EncryptedMessage:
    i) The server handles user authentication and message routing, but it does not own the users. It facilitates communication between users by transporting EncryptedMessage objects. The server ensures the delivery of messages to the correct user or retains them until the intended user logs in again.