Names: Ali Jalil, Joshua Egwaikhide, Nick Chalardsoontornvatee

Class Descriptions:

1. **User**:
    a. Represents a participant in the messaging system. Attributes include id, name, and isAuthenticated (indicating the user's authentication status). Methods allow the user to compose, encrypt, decrypt, and verify messages.
2. **Message**:
    a. Represents a message created by the user with content, timestamp, senderId, and receiverId attributes. This class acts as a data container for messages in plain text that remains on the user client.
3. **EncryptedMessage**:
    a. Stores encrypted content, timestamps, and sender/receiver IDs for messages in their encrypted state. Similar to Message, it primarily serves as a data container. Unlike Message, it is transferred to the server where it can be directed to the user specified in the receiverId attribute.
4. **Server**:
    a. Manages user authentication and message transportation. Attributes include activeUsers, representing currently logged-in users. It has methods for authenticateUser, transportMessage, and routeMessage, as well as a private method synchronizeMessages to keep messages consistent across users and ensure every sent message is received by the intended user on next login.

Relationships Summary:

1. **User and Message/EncryptedMessage**:
    a. Users create and interact with Message and EncryptedMessage objects, representing the composition relationship. Users need these classes to compose messages and address them to the appropriate user. The message then needs to be encrypted to ensure only the authorized sender and recipient know of the contents of the message.
2. **Server and User/EncryptedMessage**:
    a. The server aggregates users and their messages. It manages users without owning them exclusively, supporting their communication needs within the app. When EncryptedMessage is sent from a user to another user, the server either delivers it to the appropriate user, or retains the EncryptedMessage until the intended user logs in again.