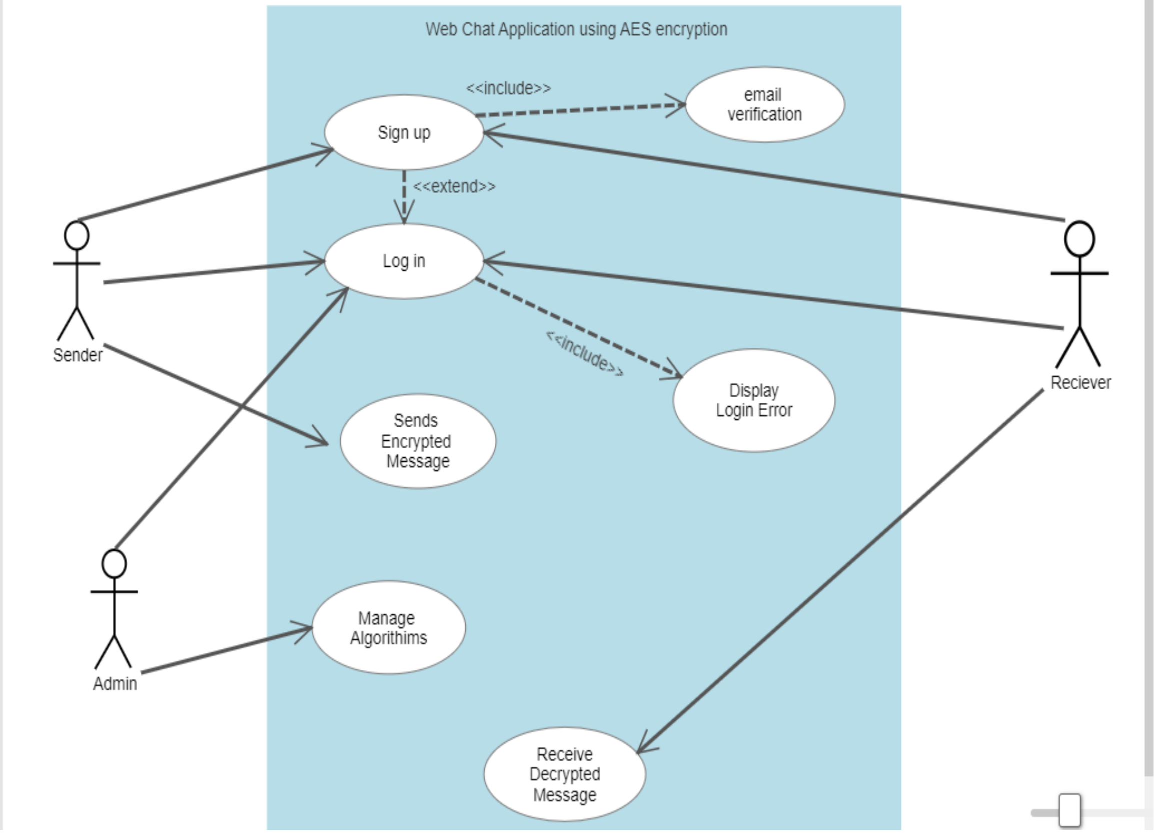
Chapter 4

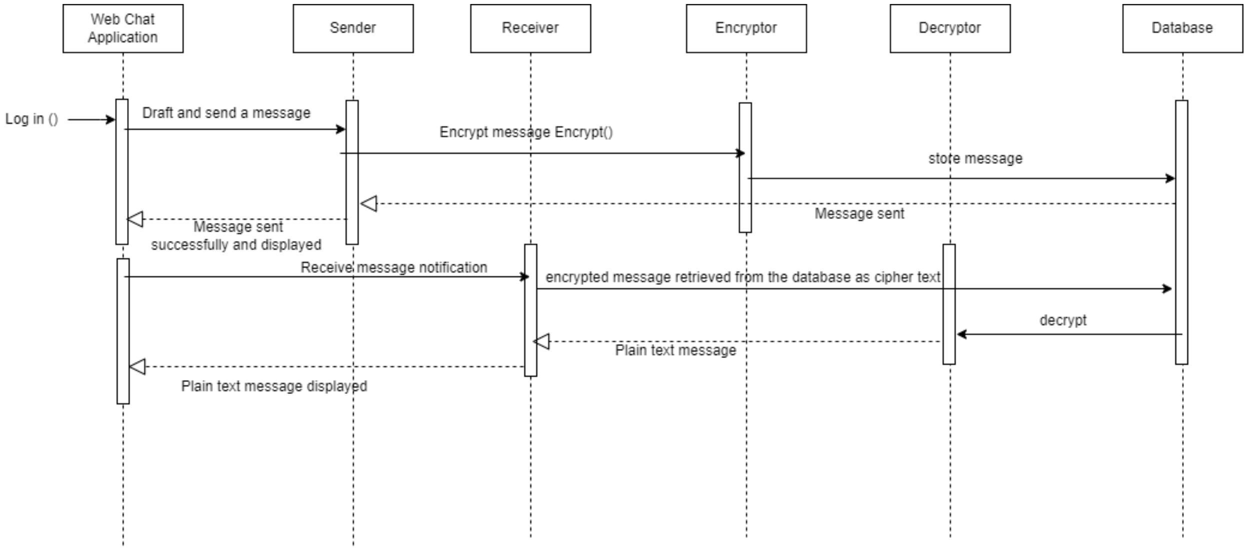
4.1 Use Case diagram.

The use case diagram represents the system's functionality where two clients communicate, one as a sender and the other as a receiver. Once the Signup and Login is complete, the sender sends a plaintext message through the user interface. The server then encrypts the plaintext message before forwarding it to the receiver. The system decrypts the message before it gets to the receiver, ensuring security. If a reply is made by the receiver, then we go back to encryption. This cycle of encryption, decryption, and communication continues between the clients. The administration can login and change the encryption modules to increase them for added security or reduce them for faster communication.



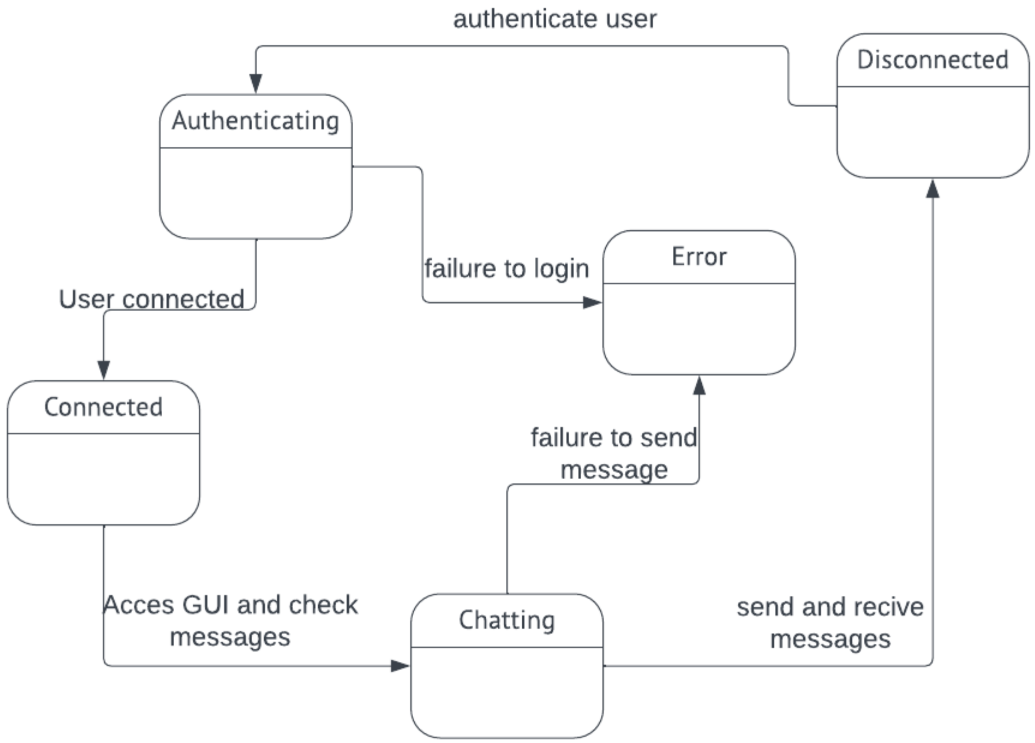
4.2 Sequence Diagrams

The sequence diagram visually represents the sequential flow of events and messages between the client and encryption processes. It helps illustrate the timing and ordering of interactions, making it easier to understand the system's behavior and the steps involved in secure communication within the chat web application. From the user on the left acting as the sender and the one on the right acting as the receiver.


Web Chat Application
Sender
Receiver
Encryptor
Decryptor
Database
Log in ()
Draft and send a message
Encrypt message Encrypt()
store message
Message sent
Message sent successfully and displayed
Receive message notification
encrypted message retrieved from the database as cipher text
decrypt
Plain text message
Plain text message displayed

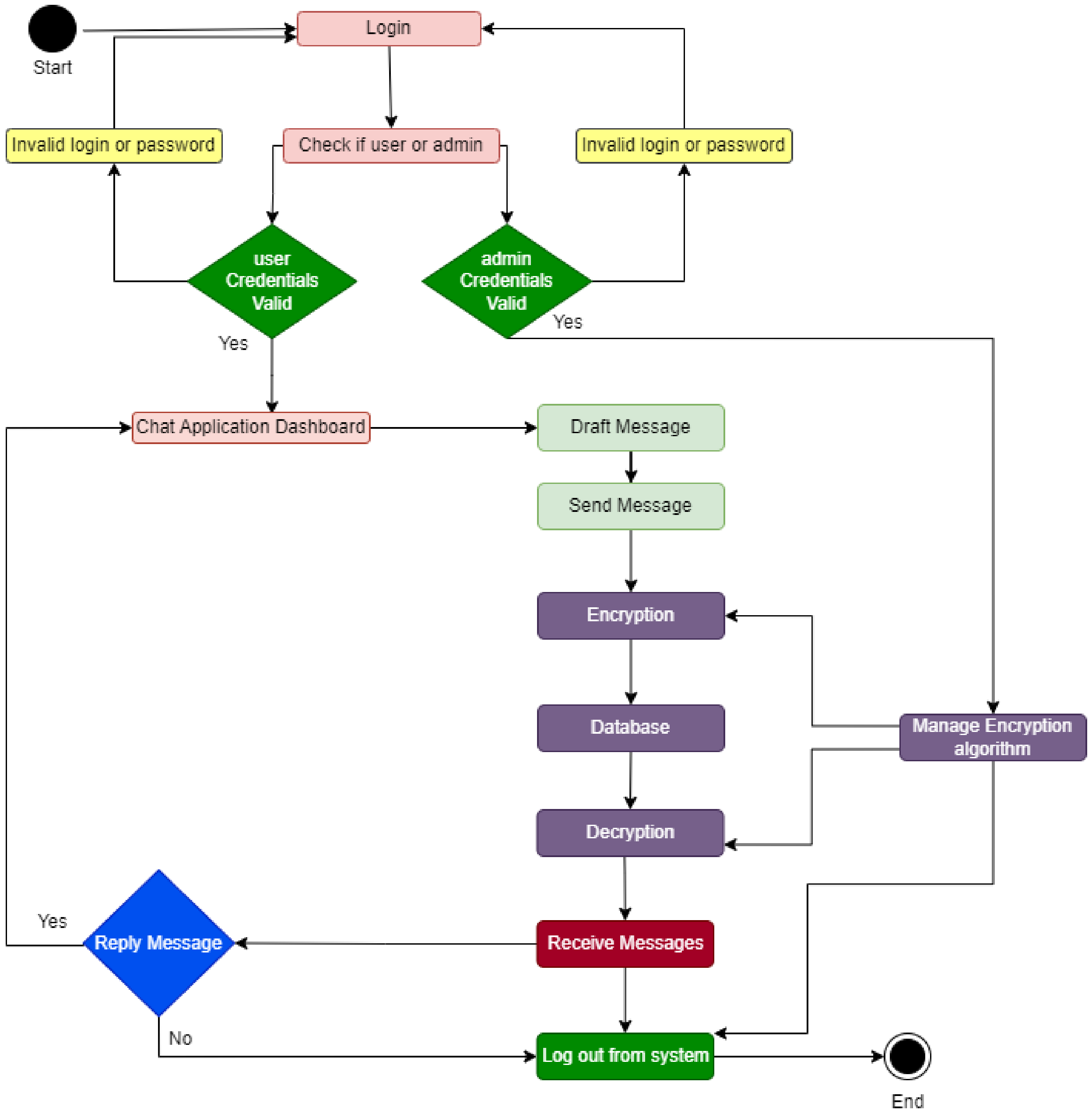
4.3 State Transition Diagrams

The state transition diagram visually represents the different states and transitions within the chat web application. It helps in understanding the flow of actions and states as the user interacts with the application, including the encryption and decryption processes involved in secure message transmission.



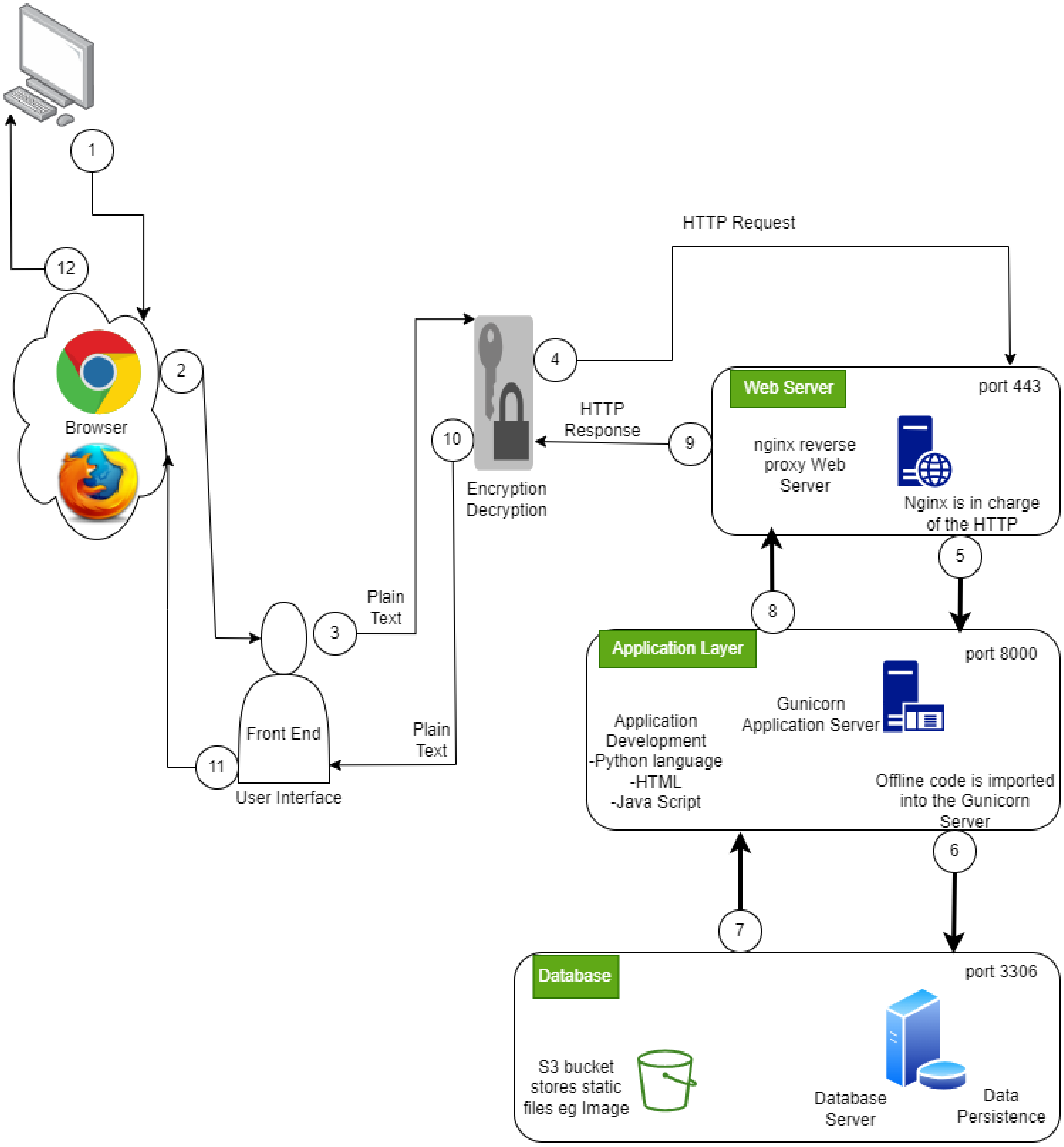
4.4 Activity Diagram

The activity diagram visually represents the sequence of activities and decisions involved in the chat web application, including login, establishing connections, sending/receiving messages, and logout. It provides a high-level overview of the steps involved in the chat application's operation while emphasizing on AES encryption module and decryption that ensure secure communication.



4.5 Systems Architecture

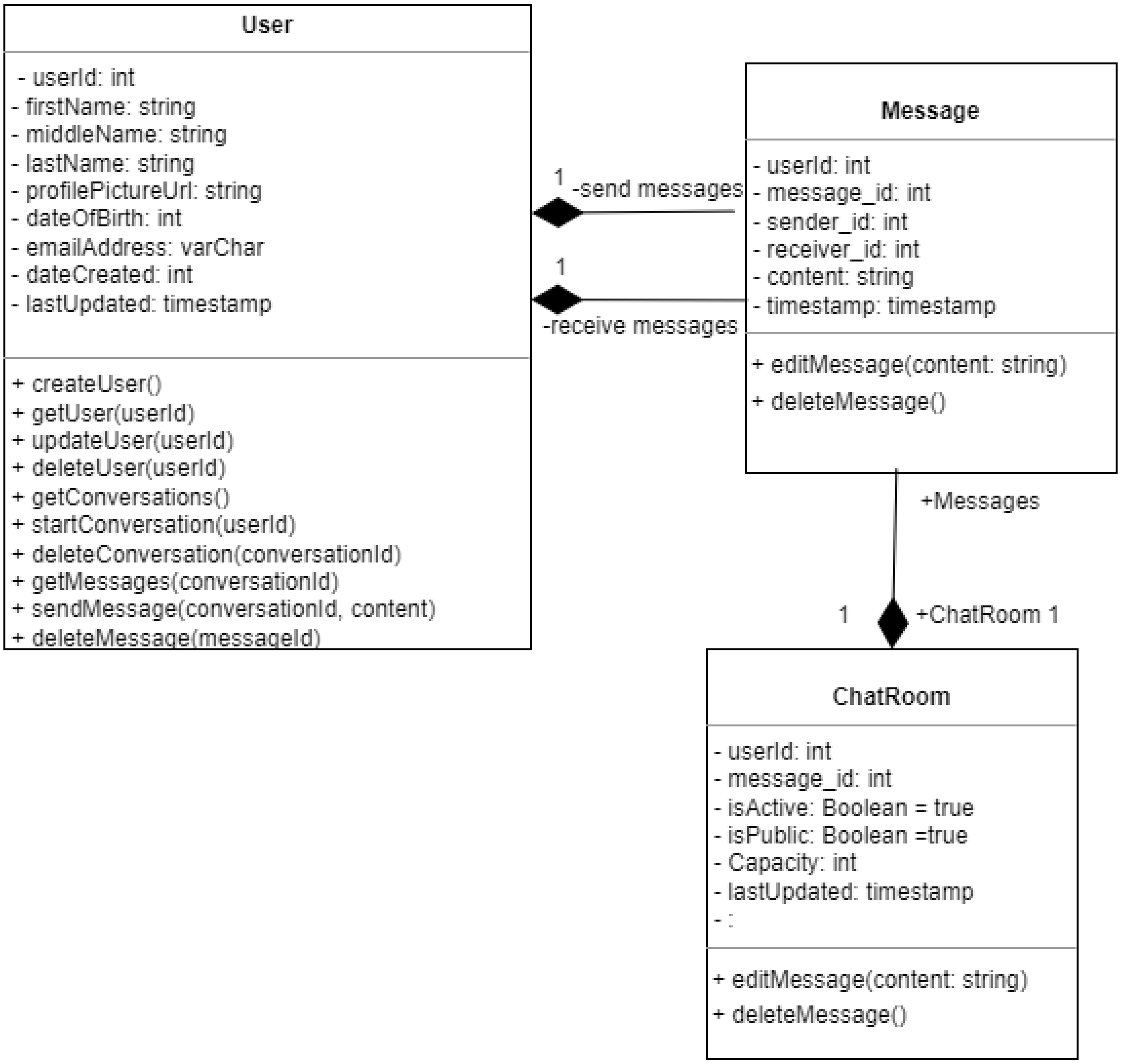
The system architecture diagram provides a visual representation of the components, modules, and interactions within the chat web application, emphasizing the encryption-related components and their integration with other elements of the system. It helps in understanding how the various parts of the system work together to provide secure chat functionality. It also helps to show what part of the system is back end what part of the system stores data and the different layers and what they hold.



4.6 Class Diagram

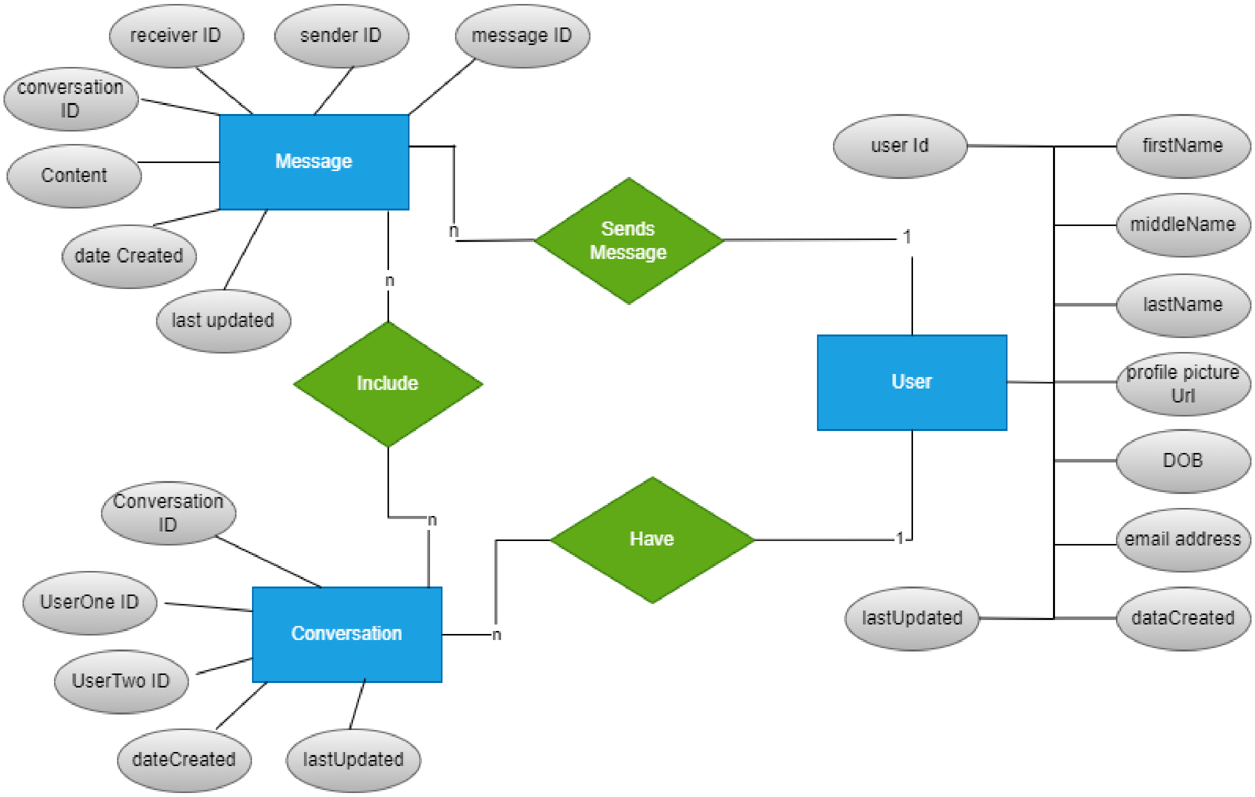
A Represents a blueprint for creating objects and defines the attributes and behaviors of those objects.

The diagram below defined different attributes and their various actions.



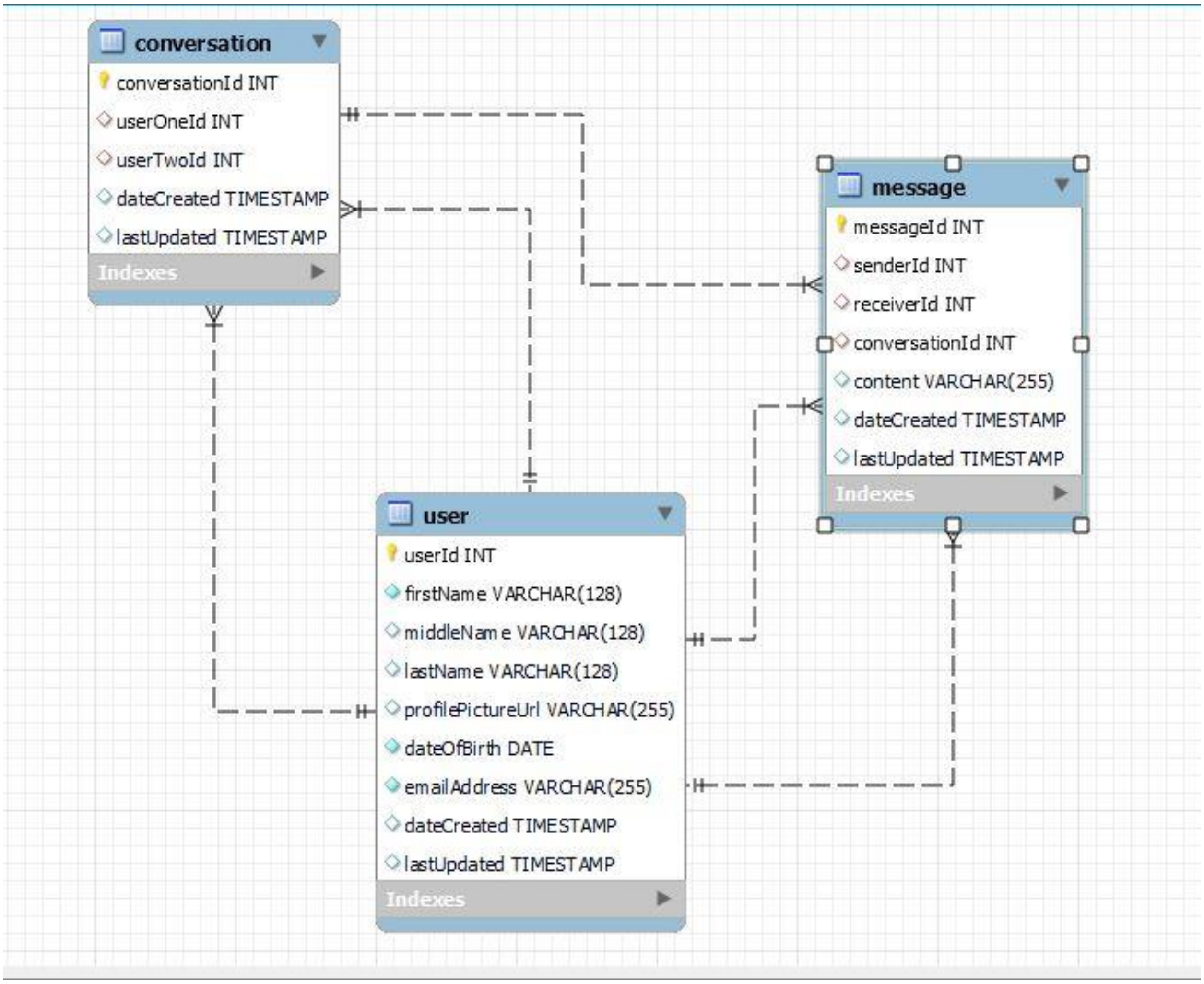
4.7 ERD

ERD diagrams are used to visualize the entities, attributes, and relationships in a database system. They aid in communication and understanding of the database structure among developers. They ensure data integrity and consistency by enforcing constraints and referential integrity. ERD diagrams assist in database maintenance, modification, and optimization for improved performance. Below is a Chinese notation ERD diagram that uses circles rectangles and diamonds to illustrate relationships.



4.8 Database Schema

They represent entities or objects within the database. Each table consists of columns and rows to store data. The columns Define the attributes or properties of the entities in a table. They represent the specific data elements that can be stored. Things like Primary Keys: Identify unique records within a table. They ensure data integrity and serve as a reference for establishing relationships with other tables. Foreign Keys: Establish relationships between tables by referencing the primary key of another table. They maintain data integrity and enable data retrieval across related tables. Relationships: Show the associations and dependencies between tables. Common relationship types include one-to-one, one-to-many, and many-to-many relationships.


conversation
conversationId INT
userOneId INT
userTwoId INT
dateCreated TIMESTAMP
lastUpdated TIMESTAMP
Indexes
message
messageId INT
senderId INT
receiverId INT
conversationId INT
content VARCHAR(255)
dateCreated TIMESTAMP
lastUpdated TIMESTAMP
Indexes
user
userId INT
firstName VARCHAR(128)
middleName VARCHAR(128)
lastName VARCHAR(128)
profilePictureUrl VARCHAR(255)
dateOfBirth DATE
emailAddress VARCHAR(255)
dateCreated TIMESTAMP
lastUpdated TIMESTAMP
Indexes