Software Requirements Specification

For

Secure Chat Application

06.03.2025

Prepared by

Specialization	SAP ID	Name
B.Tech CSE-CSF	500106290	Aadeesh Jain
B.Tech CSE-CSF	500105717	Abhinav Saini
B.Tech CSE-CSF	500106838	Aman Anand
B.Tech CSE-CSF	500107436	Deepanshu Chowdhury

School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

Table of Contents

Revision History

Date	Change	Reason for Changes	Mentor Signature

1. INTRODUCTION

1.1 Purpose of the Project

This project aims to develop a secure chat application utilizing AES encryption to ensure reliable and encrypted communication between users. The primary problem addressed is the need for secure real-time communication that safeguards the confidentiality and integrity of messages while ensuring reliable delivery.

1.2 Target Beneficiary

The project is aimed at organizations, individuals, and groups seeking secure communication channels to protect sensitive conversations from unauthorized access, whether for personal use, or business applications.

1.3 Project Scope

The encrypted chat application will enable users to send messages securely by encrypting the content using AES. It will include user-friendly features like chat history, user authentication, and secure storage of messages and encryption keys. The project deliverables include the encrypted chat system, system documentation, user manuals, and testing reports.

1.4 References :

Developing a Real-Time Secure Chat Application:

- This article discusses the development of a secure chat application similar to WhatsApp and Signal. It covers end-to-end encryption, security protocols, and more:
- Developing a Real-Time Secure Chat Application like WhatsApp & Signal

2. PROJECT DESCRIPTION

2.1 Reference Algorithm

- AES Encryption Algorithm: This is used for encrypting and decrypting the messages exchanged between users.

2.2 Data/Data Structures

The application will use a combination of data structures:

- HashMap: For storing user credentials and session data securely.
- Encrypted Message Structure: Data structure for holding encrypted messages and corresponding metadata (sender, receiver, timestamp).

2.3 SWOT Analysis

Strength:

-Secure messaging platforms prioritize data protection and privacy.

- End-to-End Encryption (E2EE): Ensures that only authorized parties can read messages.

- Standardization: Adherence to industry standards enhances interoperability.

Weaknesses:

-Complexity: Implementing robust security features can lead to complexityin code and system architecture.

-Key Management: Proper key management is essential for E2EE but can be challenging.

-Performance Overhead: Strong encryption may impact performance, especially on resource-constrained devices.

Opportunities:

-User Trust: Platforms that prioritize security gain user trust and loyalty.

-Market Demand: Growing awareness of privacy concerns drives demand for secure messaging.

-Innovation: Opportunities exist for novel features, integrations, and user experiences.

Threats:

-Emerging Algorithms: As encryption algorithms evolve, platforms must stay updated to maintain security.

-Implementation Flaws: Vulnerabilities in code or configuration can compromise security.

-Regulatory Challenges: Compliance with data protection laws and regulations poses challenges.

2.4 Project Features

- End-to-end encryption using AES.
- Multi-device access with synchronized encrypted chat data.

2.5 User Classes and Characteristics

- General Users: Individuals using the app for personal communication.
- Enterprise Users: Employees or businesses requiring secure channels for business related communication.

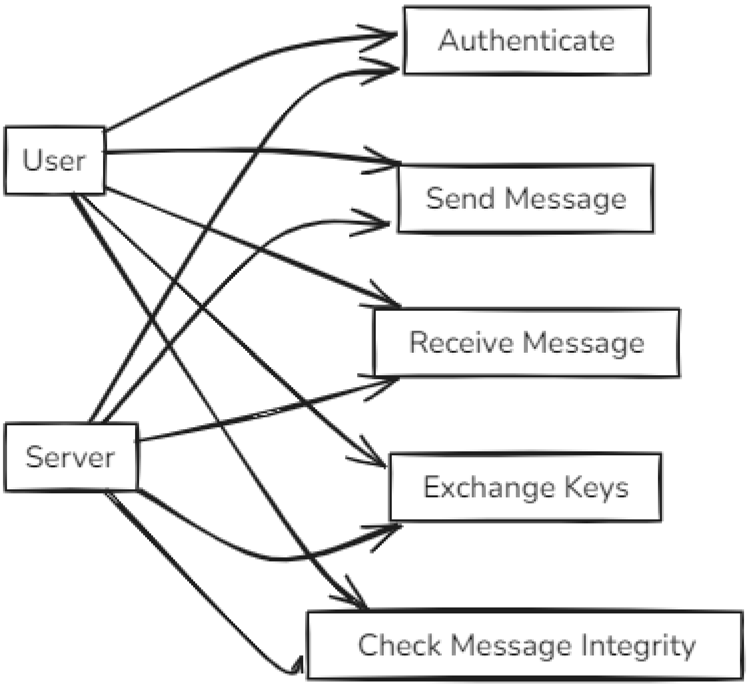
2.6 Design and Implementation Constraints

- Memory and processing power requirements for AES encryption
- Integration with multiple platforms.
- Data transfer rate and communication latency constraints.

2.7 Assumption and Dependencies

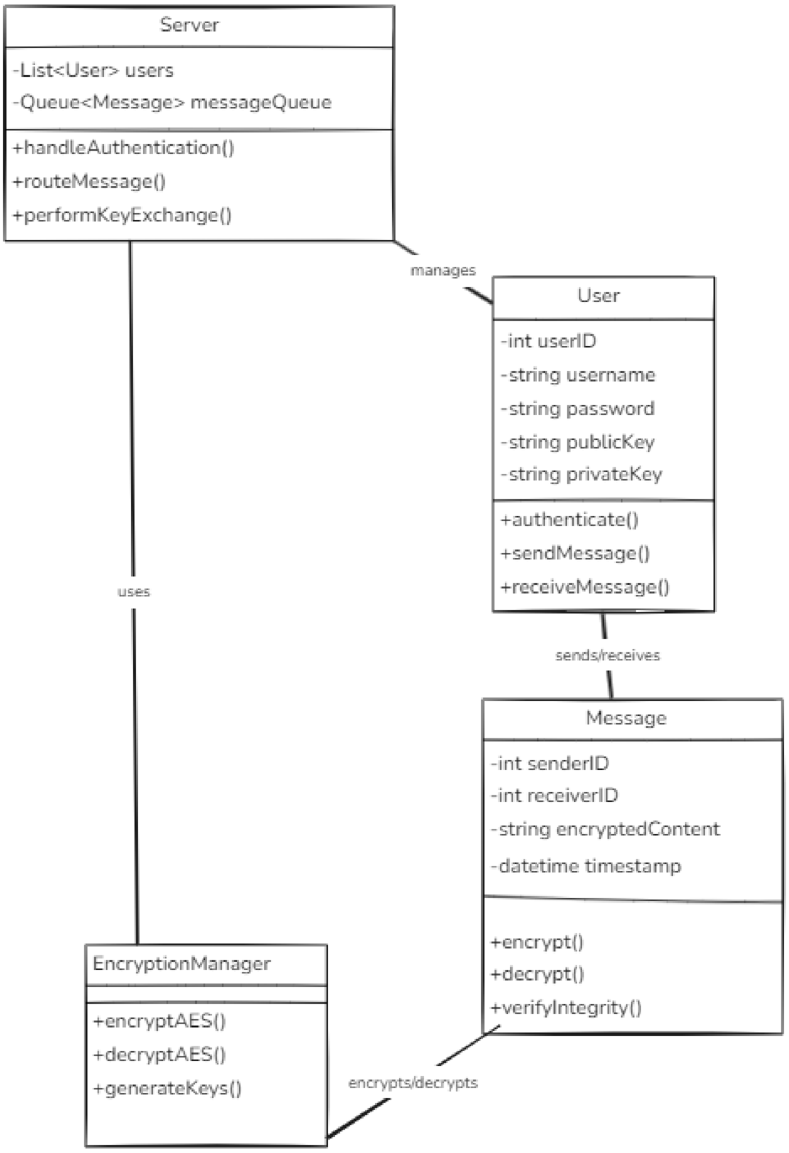
- Assumes proper AES key management and secure storage of user credentials.

USE-Case:

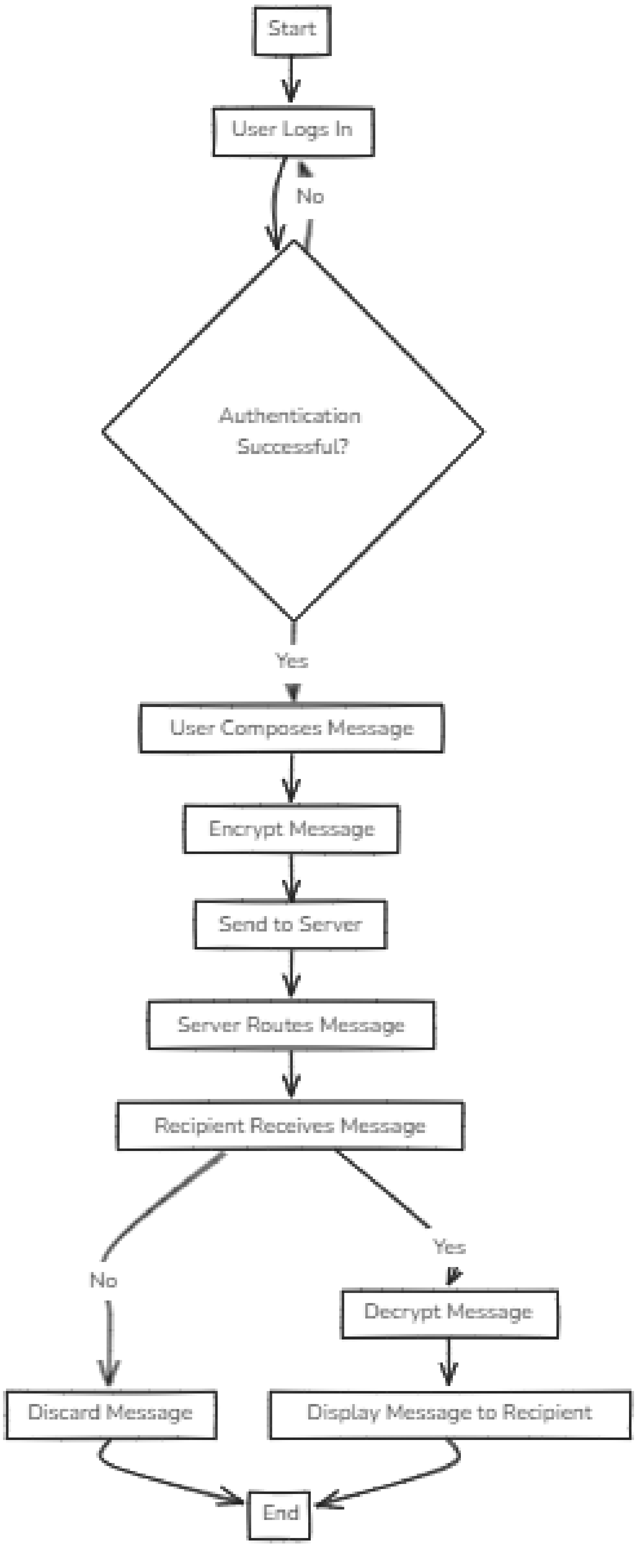


User
Server
Authenticate
Send Message
Receive Message
Exchange Keys
Check Message Integrity

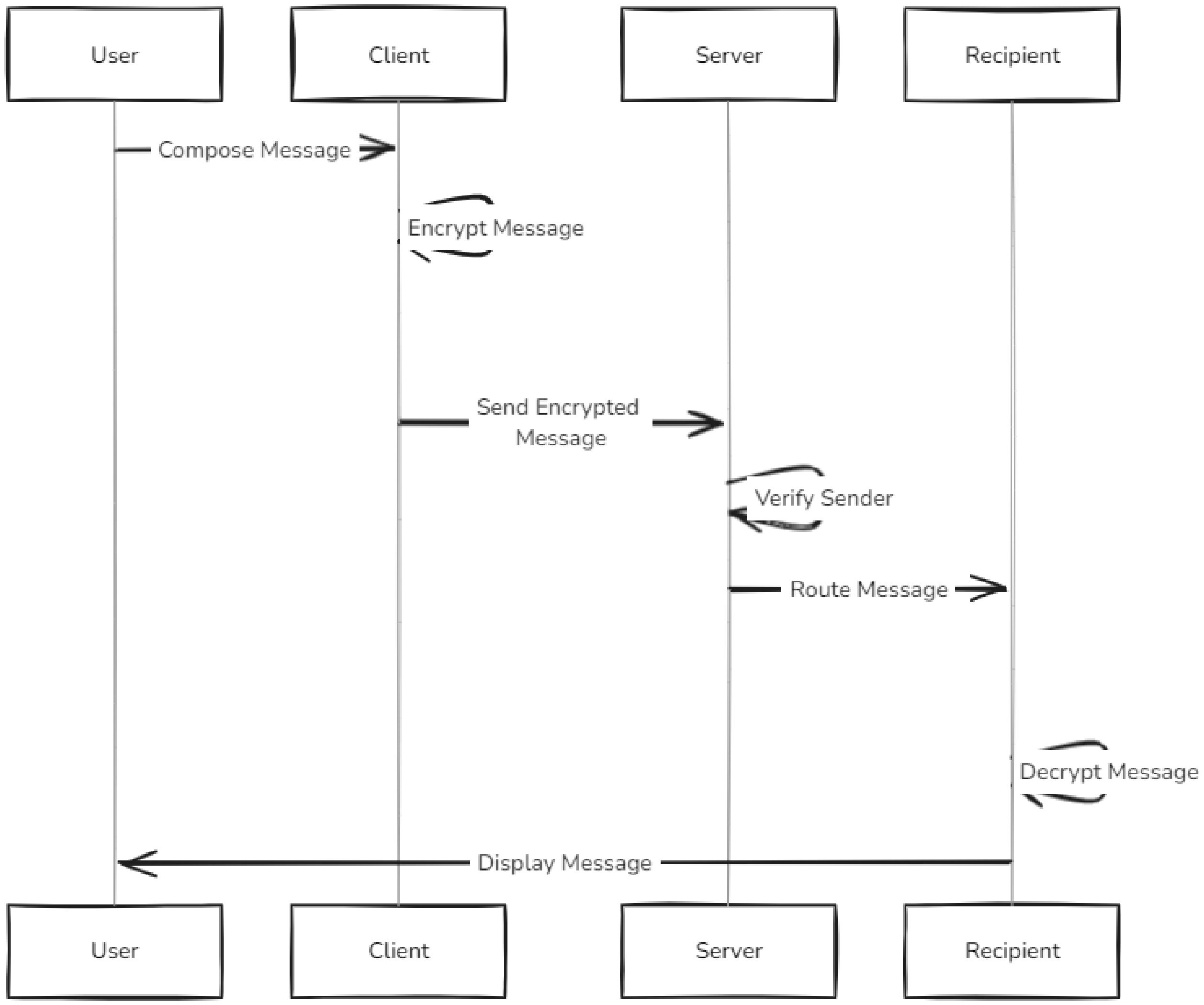
Class Diagram:



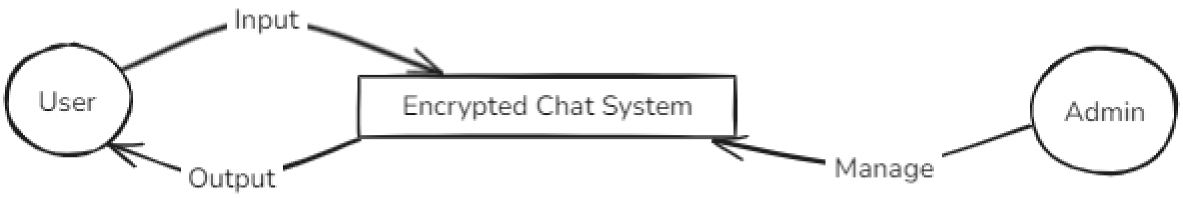
Activity:

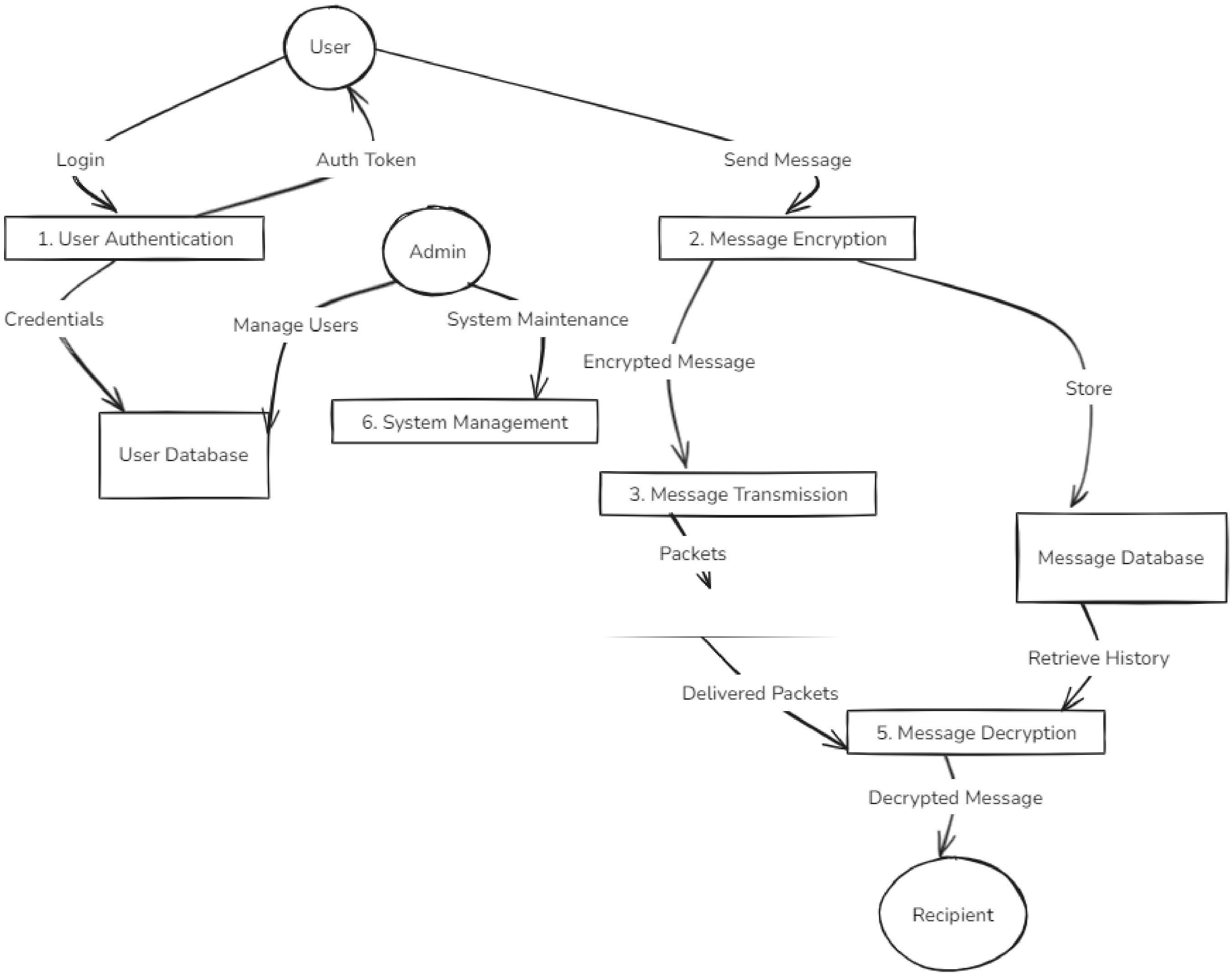

Start
User Logs In
No
Authentication Successful?
Yes
User Composes Message
Encrypt Message
Send to Server
Server Routes Message
Recipient Receives Message
No
Yes
Decrypt Message
Discard Message
Display Message to Recipient
End

Sequence:

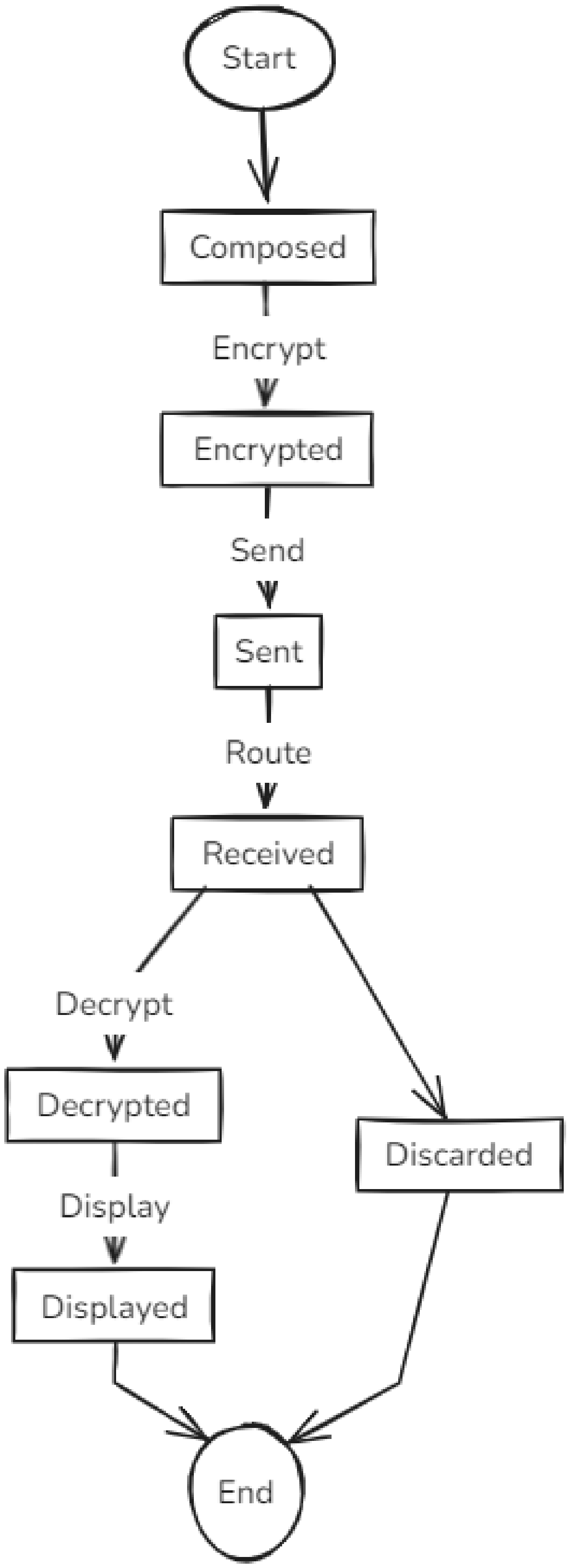

User
Client
Server
Recipient
Compose Message
Encrypt Message
Send Encrypted Message
Verify Sender
Route Message
Decrypt Message
Display Message
User
Client
Server
Recipient

Data Flow diagram





State Diagram.



3. SYSTEM REQUIREMENTS

3.1 User Interface

- Chat Window: A window where users can send and receive messages in real-time.
- User Options:
    - Login functionality for secure access.
    - Chat history management, allowing users to view past conversations (encrypted).

3.2 Software Interface

- Encryption Module: Provides AES encryption and decryption functionality for ensuring secure message transmission.

3.3 Database Interface

- Database:
    - Stores user credentials (hashed for security), encryption keys, and chat history (encrypted to maintain confidentiality).

3.4 Protocols

- AES (Advanced Encryption Standard): For encryption and decryption of messages, ensuring confidentiality and data integrity.

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Performance Requirements

- The system must handle multiple concurrent users without degrading performance.
- Messages should be delivered within 2 seconds to ensure real-time communication.

4.2 Security Requirements

- AES Encryption: All messages must be securely encrypted before transmission and decrypted upon receipt.
- Authentication: User authentication mechanisms (such as username and password, or two-factor authentication) must be implemented to ensure that only authorized users can access the system.
- Data Integrity:
  - Implement mechanisms like message digests or checksums to detect tampering or corruption of messages during transit.

4.3 Software Quality Attributes

- Reliability:
  - The system must ensure reliable message delivery, with no loss of data, even during network interruptions.
- Maintainability:
  - The system should be designed in a modular way to allow easy updates (such as adopting new encryption algorithms or protocols).
- Usability:
  - The user interface should be designed to be simple and intuitive, even for non-technical users. Features like tooltips, error messages, and easy navigation should be included.

5.OTHER REQUIREMENTS

-External Integrations:

- Support for integrating cloud services to enable seamless file storage and sharing directly from cloud storage providers like Google Drive or Dropbox.
- Future plans include integration with authentication services (e.g., OAuth) for user identity management.

Appendix A: Glossary

- AES: Advanced Encryption Standard, a symmetric encryption algorithm widely used for securing data.
- Encryption: The process of converting plain text into a secure format to prevent unauthorized access.

Appendix B: Analysis Model

- This section can include diagrams such as Data Flow Diagrams (DFD), Activity Diagrams, or Sequence Diagrams that will help validate the design and functionality of the system.

Appendix C: Issues List

- Keep track of open issues or unresolved requirements, such as:
    - Potential performance bottlenecks when scaling the application.
    - Future updates to encryption algorithms or protocols.
    - Possible user interface enhancements (e.g., dark mode, accessibility features).