

# Software Requirement Specification Document for a End to End Secured chat system: "ChatSafe"

Akram Amr, Marwan Hawash, Omar Hisham Assal, Omar Hisham Farouk  
Supervised by: Dr Nermine Naguib , Eng. Nada Ayman

January 14, 2024

Table 1: Document version history

Version	Date	Reason for Change
1.0	6-Dec-2023	SRS First version's specifications are defined.
1.1	31-Dec-2023	System Scope, Problem statement updated
2.0	3-Jan-2024	Project Overview updated
2.1	3-Jan-2024	Database Diagram updated

**GitHub:** <https://github.com/akramhammam5/ChatSafe>

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose of this document . . . . .	4
1.2	Scope of this document . . . . .	4
1.3	Business Context . . . . .	4
<b>2</b>	<b>Similar Systems</b>	<b>5</b>
2.1	Academic . . . . .	5
2.2	Business Applications . . . . .	8
<b>3</b>	<b>System Description</b>	<b>9</b>
3.1	Problem Statement . . . . .	9
3.2	System Overview . . . . .	9
3.3	Phase I . . . . .	9
3.4	PhaseII . . . . .	9
3.5	PhaseIII . . . . .	9
3.6	System Scope . . . . .	10
3.7	System Context . . . . .	11
3.8	Objectives . . . . .	11
3.9	User Characteristics . . . . .	12
3.9.1	Registered Users . . . . .	12
3.9.2	Unregistered Users (Without Account) . . . . .	12
<b>4</b>	<b>Functional Requirements</b>	<b>13</b>
4.1	System Functions . . . . .	13
4.2	Detailed Function Specification . . . . .	15
<b>5</b>	<b>Design Constraints</b>	<b>16</b>
5.1	Standards Compliance . . . . .	16
5.2	Software Limitations . . . . .	17
<b>6</b>	<b>Non-Functional Requirements</b>	<b>17</b>
<b>7</b>	<b>Data Design</b>	<b>18</b>
7.1	Database . . . . .	18
<b>8</b>	<b>Preliminary Object-Oriented Domain Analysis</b>	<b>20</b>
<b>9</b>	<b>Operational Scenarios</b>	<b>20</b>
9.1	Operational Scenario 1: User Registration . . . . .	20
9.1.1	Description . . . . .	20
9.1.2	Steps . . . . .	20
9.2	Operational Scenario 2: Initiating a Chat . . . . .	20
9.2.1	Description . . . . .	20
9.2.2	Steps . . . . .	21

9.3	Operational Scenario 3: Receiving and Decrypting a Message	21
9.3.1	Description	21
9.3.2	Steps	21
9.4	Operational Scenario 4: Upgrading to Premium	21
9.4.1	Description	21
9.4.2	Steps	21
<b>10</b>	<b>Project Plan</b>	<b>22</b>
10.1	Project Schedule	22
10.1.1	Phase 1: Requirements Gathering and Analysis (Dec 2023 - Jan 2024)	22
10.1.2	Phase 2: System Design and Architecture (Jan 2024)	22
10.1.3	Phase 3: Implementation (Feb 2024 - Mar 2024)	22
10.1.4	Phase 4: Testing and Quality Assurance (Apr 2024)	22
10.1.5	Phase 5: Deployment and User Training (May 2024)	22
10.2	Supportive Documents	23
10.2.1	Users-Survey	23

## **Abstract**

The field of CyberSecurity specially audio steganography has achieved significant advancements in recent years, offering exciting possibilities for data security, and digital forensics. However, alongside these advancements, several challenges existed, requiring ongoing research and development efforts. the lack of knowledge of the typical user, coupled with the seeming difficulty of straightforward procedures, encourages them to search for a easy to use apps to help them get what they actually need. Steganography is a very effective method that is used to hide data by governments, large organizations and individuals that prefer to hide their data without the need to know anything extra or behind the details. Steganography techniques can be used for data security and hiding without being noticed by an adversary. Our proposed system is a web application that should help organizations and individuals be able to hide data securely inside mp3 files.

# **1 Introduction**

## **1.1 Purpose of this document**

The Software Requirements Specification (SRS) document is intended for the application's developers and stakeholders and aims to illustrate the proposed system's features, explain its objectives, and set guidelines for the developers while working on it. All the data that have been stolen cost time and a lot of money to get secured again.[1]

## **1.2 Scope of this document**

This document provides a detailed description of the application, functional and non-functional requirements, basic interface and data designs, and an analysis of the classes required and their relationships. These Diagrams should hep developers to be able to implement their future tasks.

## **1.3 Business Context**

According to IBM [2] In 2023, the global average cost of a data breach was USD 4.450 million, a 15 percent increase in three years, also As a result of a breach, 51 percent of organizations intend to increase security investments, including incident response (IR) planning and testing, employee training, and threat detection and response tools. The three essential elements of steganography—security, capacity, and robustness—make covert information transfer via text files and the development of covert communication channels worthwhile mentioned by Simplilearn article [3]. As mentioned that data security and breaches becoming a challenge every year. Steganography and Cryptography importance in general is surely guaranteed as organizations and governments are investing more as time passes to overcome data theft problems that makes our application helpful in a lot of cases when it comes to business.

## 2 Similar Systems

### 2.1 Academic

In paper [4] the authors used Images to hide data using multi-layered security model but this time they added Morse code into the layer. Their system is from two stages of cryptography and one stage of steganography. In the first stage of their proposal, the important text will divide into two parts then the first part will encrypt by Caesar Cipher while the second part will be encrypted by Vigenere Cipher. In the second stage, the ciphertext will be modified into Morse code. In the third stage, the ciphertext will then be hidden in an image by using the LSB technique . Lenna, Peppers, Baboon, and Boat are images that are used as cover data of size  $512 \times 512$  and grey colour. The secret message has 104 characters that divide into two parts. The first 52 characters are encrypted by Caesar Cipher while the other 52 characters are encrypted by Vigenere Cipher. The ciphertext of Caesar Cipher and the ciphertext of Vigenere Cipher collect in one stream and coding by morse codes. The results provided a great robustness and secure method. The paper as usual used Images as the medium to hide data to it. As Audio is also popular, it didn't grab a lot of attention like images through the years. Maybe images are more effective but that shouldn't let researches focus more on image steganography.

In paper [5] The authors explained a system that is suggested to improve the security of private text data on personal computers by combining steganography and cryptography. To attain the highest level of security, the system uses audio-based steganography and RSA cryptography as sequential layers. The authors discussed another researches that used a great methods for data security. These methods encrypt the data using cryptographic algorithms, and then use modified LSB algorithms to conceal the encrypted text inside audio files. These methods have the advantage of increasing text hiding capacity and offering security. The proposed system combines separate layers of cryptography and steganography to ensure high security in computer applications. The public cryptography system known as the RSA algorithm, which requires two keys for both encryption and decryption, is used by the cryptography layer. The steganography layer uses least significant bits (LSBs) to increase the number of hidden bits in audio files, hiding the encrypted data within. This method preserves security while increasing system capacity. They designed the system using Matlab, First in the encryption process they converted the secret message to binary representation to be encrypted using RSA. On the other side they converted the audio file into binary samples to hide data into LSBs. Their Audio cover was 16-bits per sample. The paper's results were done by adding a secret message into 15 different cover audios. As shown in the following table, They used different capacity and more capacity will lead to less security and vice versa.

Audio test-file number	High security and low capacity (1 LSB)		Medium security and capacity (2 LSB)		Low security and high capacity (3 LSB)	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
1	352.9	136.2	705.9	130.8	1058.8	119.0
2	306.1	129.2	612.2	125.3	918.4	111.7
3	73.1	123.5	146.2	119.2	219.3	106.6
4	756.0	95.4	1512.0	88.8	2268.0	75.4
5	123.5	125.4	247.0	121.6	370.5	107.4
6	135.8	122.9	271.7	117.4	407.6	104.2
7	352.9	135.5	705.9	129.2	1058.8	116.5
8	289.1	126.2	578.2	119.3	867.4	105.7
9	756.0	96.5	1512.0	90.5	2268.0	79.3
10	350.8	134.1	700.0	128.8	950.7	122.5
11	756.0	93.4	1512.0	87.99	2268.0	81.3
12	756.0	93.1	1512.0	86.08	2268.0	74.5
13	756.0	94.3	1512.0	88.37	2268.0	76.8
14	120.2	123.5	240.4	119.2	360.5	100.6
15	450.3	110.3	900.6	105.5	1250.0	98.2

Figure 1: Testing results of stego-embedding the encrypted fixed sensitive data “text” in 15 different audios

In paper [6] The significance of covert communication in the context of cybersecurity is covered in the introductory section, along with the use of steganography and cryptography as data hiding techniques. It emphasizes how difficult the field of audio steganography is because of how sensitive the human auditory system is. The proposed method uses two-level encryption to boost capacity and robustness and combines XORing and LSB (Least Significant Bit) to improve security. The paper started comparing different steganography techniques and their difference and how they work. Their proposed method uses The LSB coding with XORing method as technique for embedding and retrieving data in audio signals. To improve security, it combines the simple LSB method with an XOR operation on the LSBs. The LSB of the sample is modified or remains unchanged depending on the outcome of the XOR operation and the message bit. This method improves security while maintaining a high bit rate. Data embedding entails converting the host audio message and secret message into binary bit sequences. Following that, the XORing procedure is used, in which the LSB (Least Significant Bit) is flipped based on the message bit to be embedded. The stego audio signal is made up of modified cover audio samples. The LSB and the bit next to the LSB are XORed to retrieve the message bit, which is then converted to decimals during the data retrieval process. Finally, the secret message is rebuilt. As shown in Figure 2 the MSE and PSNR is measured with different data capacity.

Data size	MSE	PSNR
10B	0.00021	36.70
50B	0.00042	33.76
100B	0.00069	31.26
300B	0.00112	29.47
500B	0.00147	28.32

Figure 2: MSE and PSNR analysis

## 2.2 Business Applications

### 1-Stegonaut.com

Stegonaut is a web application that hides a secret message in music by allowing the user to choose the music and type the message. A music file is then created with the message hidden within the chosen music. This will be done by taking chunks of text and hiding them in unused bits in the header of the music MP3 file. These bits are used because they are not heavily utilized by encoders and decoders. By using this technique, this application aims to add information and data to an MP3 file without affecting its quality.



Figure 3: Business Application 1

### 2-Telegram

Telegram is far more than just a chat platform. It's become a global social media platform, attracting large user communities and allowing accounts to quickly reach millions of followers through broadcasts. Secure messaging is just one application for it, and it is not automatic for Telegram to use end-to-end encryption. You'll have to go to "Secret Chat Mode" if you wish to use it.

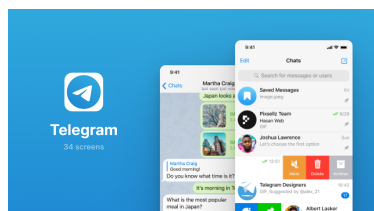


Figure 4: Business Application 2



## **3 System Description**

### **3.1 Problem Statement**

The cybersecurity field is the future and is going to be the most in demand the next few years[7]. Cryptography and Steganography are used in a lot of applications when it comes to data safety, but their main problem was the lack of new ideas and techniques. As attacks increases day by day, we need an effective way to help reducing them. While Steganography is considered old fashion and cryptography is threatened by quantum attacks, security researchers started looking for a new way to utilize them. Combining Cryptography with Steganography turned out to be effective. The problem is most of researches focused about using mult-layer models to hide data inside images. Audio Steganography is another great way for hiding data and still under research trying to come out with new ideas that would better utilize its performance. While Audio Steganography is effective it comes with some challenges such as, the capacity of data to hide in an audio file, Mp3 compression and how to make sure data is not lost after the hiding process. In addition it is better to use our solution in an applications that would let users more safe and private with a good and easy UI and with the care of UX.

### **3.2 System Overview**

#### **3.3 Phase I**

The Scenario works as follows, The User registers in ChatSafe web app be able to access his/her ChatSafe account. After logging in the user choose if the he/she wants to start chatting with his/her friend for example or just secure a cover audio. Securing Cover Audio without chatting is not going to require an account.

#### **3.4 PhaseII**

The user should upload a cover Audio while creating his/her account to use it as the cover audio for encryption and steganography. The user enters a secret text message or a Voice note as an input. First the Text is encrypted using AES encryption Algorithm. After that a cipher text is Produced and embedded to the echoes of the cover audio. Finally the secret message is decrypted after Extracting it from the Steg Audio to be recieved by the user. In the other side the Cover audio will introduce echoes to hide data inside them and generate the audio containing the secret message. After extracting the secret Audio we produce the Cipher Text and decrypt it.

#### **3.5 PhaseIII**

After decrypting the cipher text the secret message should be able to appear to be sent to the reciever side in our application. If The user is sending a Voice Note then the voice note will be directly embedded to the introduced echoes of the cover audio and extracted at the receiving side.

## ChatSafe

A Secured Chat System for private and Secure Communication

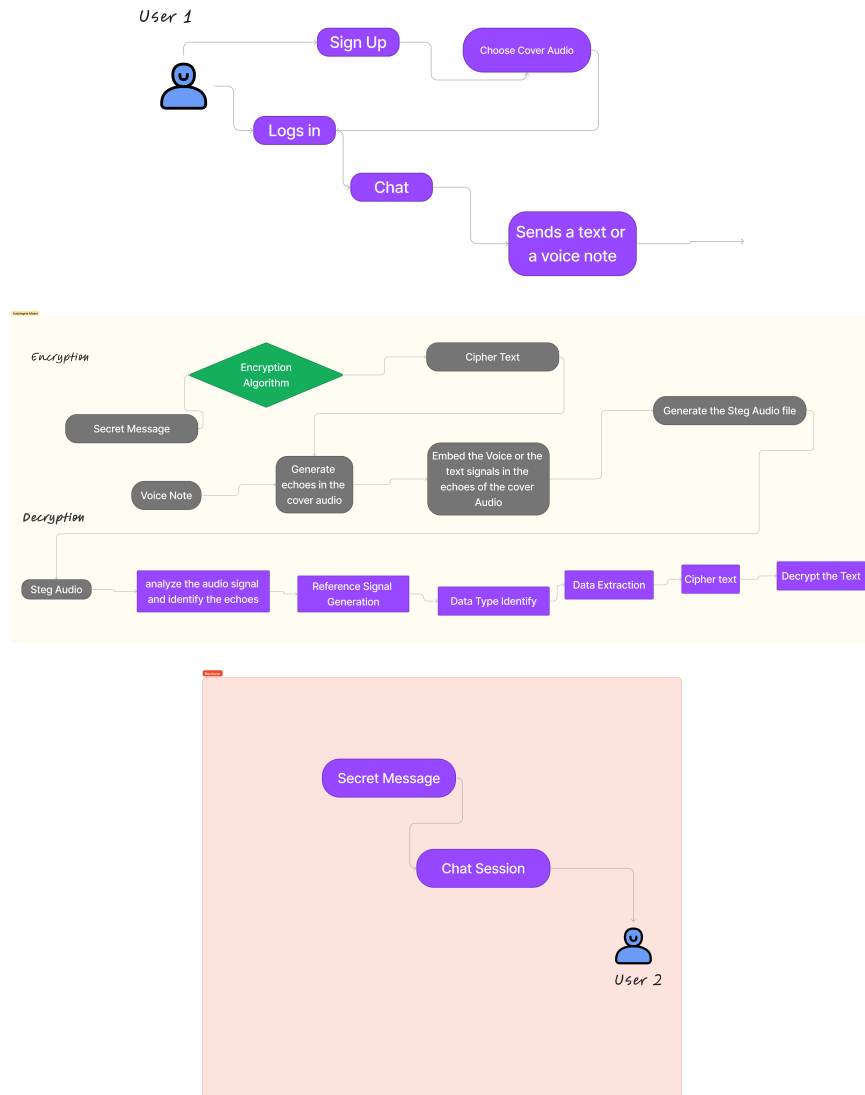


Figure 5: System Overview Diagram

### 3.6 System Scope

Our system is a secure chat application with audio steganography capabilities, offering users a unique communication experience. Users can register by uploading a cover audio file, serving as the carrier for encrypted text messages. Upon initiation of a chat, text messages undergo AES encryption, and the ciphertext is embedded into the cover audio file using steganography techniques.

The system enables secure message exchanges, with receiver decrypting and retrieving the original text via our system. Non-registered users can also utilize the system by uploading a cover audio file, embedding a text message, and subsequently decrypting the hidden text with our system. The system includes user registration, secure messaging, AES encryption, steganographic embedding, and cover audio file management. Additionally, a free trial option allows users to use our system several times, after which they are prompted to upgrade to the premium version for chatting without limitations.

### 3.7 System Context

The main vision of the App is to create a secure and usable chat system that can secure the user's messages if the message is text message or a voice record, in addition to providing a smooth, easy and comfortable user experience for the User. The Web application also should provide a security feature to secure music files without using the chat system. It's also important to consider a good looking and easy to use UI with comfortable User experience.

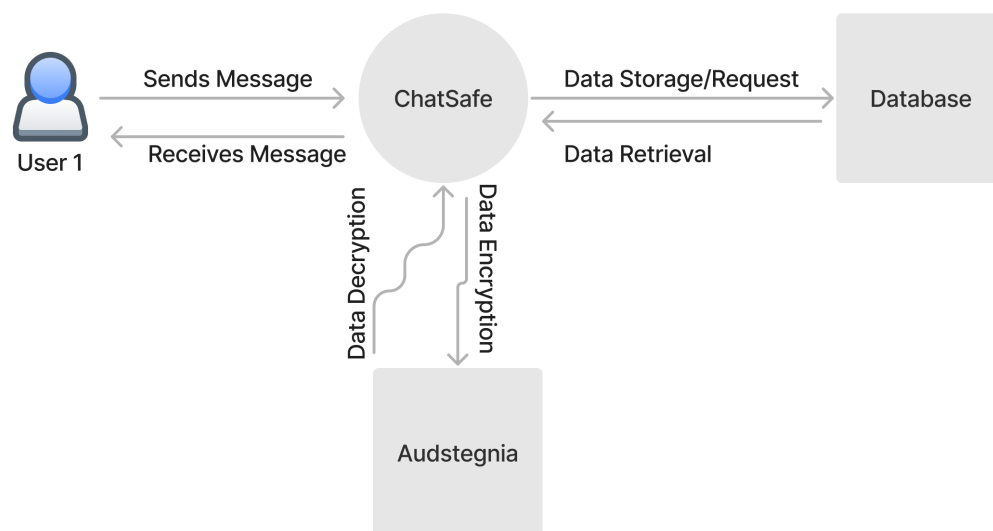


Figure 6: Context Diagram overview

### 3.8 Objectives

System objectives are:

- Our system focus on providing secure chat system for users to chat and interact with their friends.

- Our system will be deployed on a reliable domain server to ensure availability.
- Authentication functionality to ensure the confidentiality between the sender and receiver.
- Our web application will provide security in which non logged in users to upload music with a text message, where the output will be the uploaded music with the hidden message. While logged in users able to chat with their friends and their messages will be encrypted by using AES encryption algorithm to ensure the message integrity

## 3.9 User Characteristics

The users of the ChatSafe system can be broadly categorized into two groups based on their interactions with the application:

### 3.9.1 Registered Users

- **Account Creation:** Users interested in utilizing the full functionality of ChatSafe need to register on the ChatSafe web app. Account creation involves providing necessary details and uploading a cover audio during the registration process.
- **Authentication:** Registered users must log in to their ChatSafe accounts to access the system. This ensures the security and privacy of their stored cover audio and encrypted messages.
- **Communication:** Users can choose to initiate secure chats with their friends. They have the option to send encrypted text messages or voice notes through the ChatSafe platform.
- **Cover Audio Selection:** During account creation, users upload a cover audio file. This audio file is used for encryption and steganography purposes. The echoes in the cover audio serve as a medium to hide encrypted data.
- **Message Encryption:** Users can input secret text messages or voice notes, which are encrypted using the Advanced Encryption Standard (AES) algorithm. The resulting ciphertext is then embedded into the echoes of the chosen cover audio.
- **Message Decryption:** At the receiving end, users can extract the hidden message from the steganographic audio. The extracted ciphertext is decrypted using the appropriate key, revealing the original secret message.

### 3.9.2 Unregistered Users (Without Account)

- **Cover Audio Securement:** Users who do not wish to register for an account can still use ChatSafe for securing cover audio without engaging in the chat functionality. This process does not require account creation.
- **Limited Features:** Unregistered users have limited access to features, primarily focused on securing cover audio. They do not have access to the chat functionalities available to registered users.

## 4 Functional Requirements

### 4.1 System Functions

**ID: 001** Users should be able to create an account

**ID: 002** User should be able to upload a cover audio

**ID: 003** Registered users should be able to initiate a chat with another registered user

**ID: 004** Users should be able to send encrypted text messages within the cover audio file to another user

**ID: 005** The system should employ AES encryption to secure text messages exchanged between users

**ID: 006** Text messages, once encrypted, should be hidden within the cover audio file using steganography techniques

**ID: 007** Recipients should be able to receive the cover audio file and decrypt the hidden text using the application

**ID: 008** Non-registered users should have the option to upload a cover audio file and embed a text message for secure communication

**ID: 009** The system should allow a free trial allowing users a limited number of application uses

**ID: 010** Users should be prompted to upgrade to the premium version for continued access

**ID: 011** Premium users should have unlimited access to the application's features beyond the free trial limit

**ID: 012** Users should be able to upload a new cover audio file or change their existing one

**ID: 013** The application should have a user-friendly interface for easy navigation and usage

**ID: 014** Users should receive feedback and notifications about the status of their messages, trial limits, and premium upgrade options

**ID: 015** The system should support common audio formats for cover files

**ID: 016** The system should provide clear and informative error messages for users in case of unsuccessful operations



Figure 7: Use case diagram

## 4.2 Detailed Function Specification

<b>Name</b>	Initiation of chat
<b>Code</b>	ID: 003
<b>Priority</b>	High
<b>Critical</b>	No
<b>Description</b>	Allow registered users to initiate a chat with another registered user.
<b>Input</b>	Sender's credentials, recipient's credentials, message
<b>Output</b>	Chat initiation, session setup
<b>Pre-condition</b>	Both users are registered.
<b>Post-condition</b>	A chat session is established, allowing secure communication.
<b>Dependancy</b>	ID:001 Create Account

<b>Name</b>	AES Encryption
<b>Code</b>	ID: 005
<b>Priority</b>	High
<b>Critical</b>	Yes
<b>Description</b>	Implement AES encryption on texts to ensure confidentiality.
<b>Input</b>	Plain text message
<b>Output</b>	Encrypted Cipher text
<b>Pre-condition</b>	-
<b>Post-condition</b>	Encrypted text is securely stored or transmitted.
<b>Dependancy</b>	ID:004 Send message

<b>Name</b>	Steganography
<b>Code</b>	ID: 006
<b>Priority</b>	High
<b>Critical</b>	Yes
<b>Description</b>	Apply steganography to hide encrypted text within cover audio
<b>Input</b>	Encrypted Cipher text, cover audio file
<b>Output</b>	Audio file with hidden text
<b>Pre-condition</b>	AES encryption is applied on the text
<b>Post-condition</b>	Concealed audio file is ready for transmission.
<b>Dependancy</b>	AES Encryption (ID: 005)

<b>Name</b>	Decryption Process
<b>Code</b>	ID: 007
<b>Priority</b>	High
<b>Critical</b>	Yes
<b>Description</b>	Enable recipients to decrypt hidden text from the cover audio file.
<b>Input</b>	Concealed audio file, application
<b>Output</b>	Original text
<b>Pre-condition</b>	Cover audio file contains hidden text.
<b>Post-condition</b>	Original text is <u>displayed</u> to the recipient.
<b>Dependancy</b>	Steganography (ID: 006)

## 5 Design Constraints

### 5.1 Standards Compliance

Users have the capability to upload their music in either WAV or MP3 file formats to our platform. To ensure optimal performance and efficient processing, there is a set maximum size limit for the uploaded audio files, preventing them from exceeding a specified size. Additionally, users can input text messages associated with their uploaded file, and these messages have a specific size limit in bits. This dual restriction on both audio file size and text message size ensures effective experience when using our web application.



## 5.2 Software Limitations

To ensure optimal performance, the server needs sufficient and scalable storage capacity to handle user uploaded files and embedded text data. Additionally, it should be equipped with sufficient RAM to facilitate the concurrent processing of multiple audio files. Also, adequate network bandwidth is required for efficient file uploads and downloads by users.

# 6 Non-Functional Requirements

## Security

- The system must adhere to industry-standard encryption practices, using the AES algorithm with a key length of at least 128 bits.
- The encryption process should be resistant to known cryptographic attacks, ensuring the confidentiality of the hidden information.
- The system must implement robust access controls to prevent unauthorized access to sensitive data.

## Performance

- The encryption and encoding processes should not exceed a specified time limit for a given input size, ensuring real-time or near-real-time processing.
- The system must be scalable to handle a large volume of text inputs, ensuring efficient performance even during peak usage.

## Reliability

- The system should demonstrate a high level of reliability, with minimal downtime or disruptions during normal operation.
- Error handling mechanisms must be in place to gracefully manage unexpected issues, providing informative error messages and logs for troubleshooting.

## Usability

- The user interface should be intuitive and user-friendly, allowing users to easily input text and retrieve the encoded message without the need for extensive training.
- The system should provide clear feedback on the status of the encryption process, including success or failure notifications.

## **Portability**

- The ChatSafe system should be platform-independent, capable of running on various operating systems such as Windows, macOS, and Linux.
- The system must be compatible with commonly used browsers and devices to ensure accessibility for a wide range of users.

## **Scalability**

- The system should be designed to scale horizontally to accommodate increasing user loads and growing database.
- Scalability measures should be in place to handle larger MP3 files or voice notes and longer text inputs without a significant impact on performance.

## **Maintainability**

- The codebase must be well-documented, with clear comments and documentation for ease of understanding and future maintenance.
- Updates and patches should be deployable with minimal disruption to the existing system, ensuring maintainability over time.

## **Interoperability**

- The Audstegnia system should be compatible with common MP3 players and audio software to ensure seamless playback of the encoded messages.
- The system should adhere to relevant audio file standards and conventions to enhance interoperability with external applications.

## **Compliance**

- The system must comply with relevant data protection and privacy regulations, ensuring the secure handling of user data.
- Compliance with legal standards and intellectual property rights must be maintained throughout the development and deployment of the system.

# **7 Data Design**

## **7.1 Database**

The database is utilized to store all users' data safely. User info including: Profile pics, Encrypted Mp3 files if any, choosed cover audio and password, any payment that have been made, information

about messages sent, and friendships shall be stored in the database and used for retrieval or session management. Moreover, Secured audio files are stored in the database with their chosen format. The Entity Relationship Diagram (ER Diagram) in the following figure demonstrates the structure of the stored data.

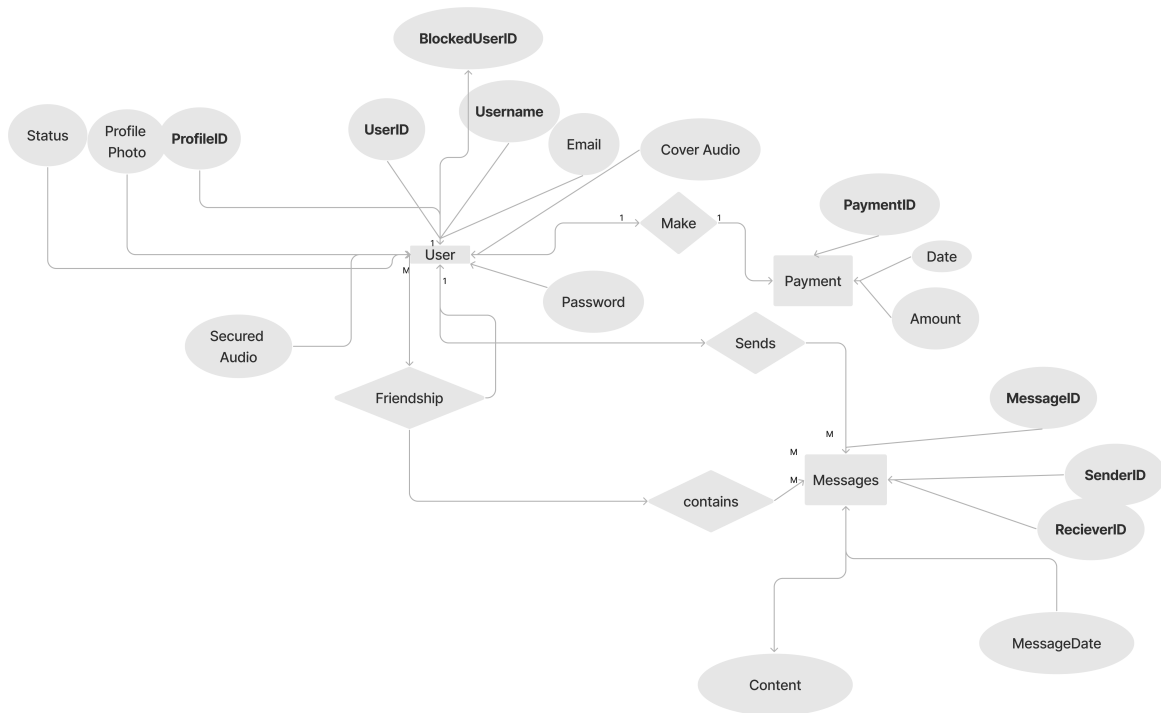


Figure 8: Database ER Diagram Overview

## 8 Preliminary Object-Oriented Domain Analysis

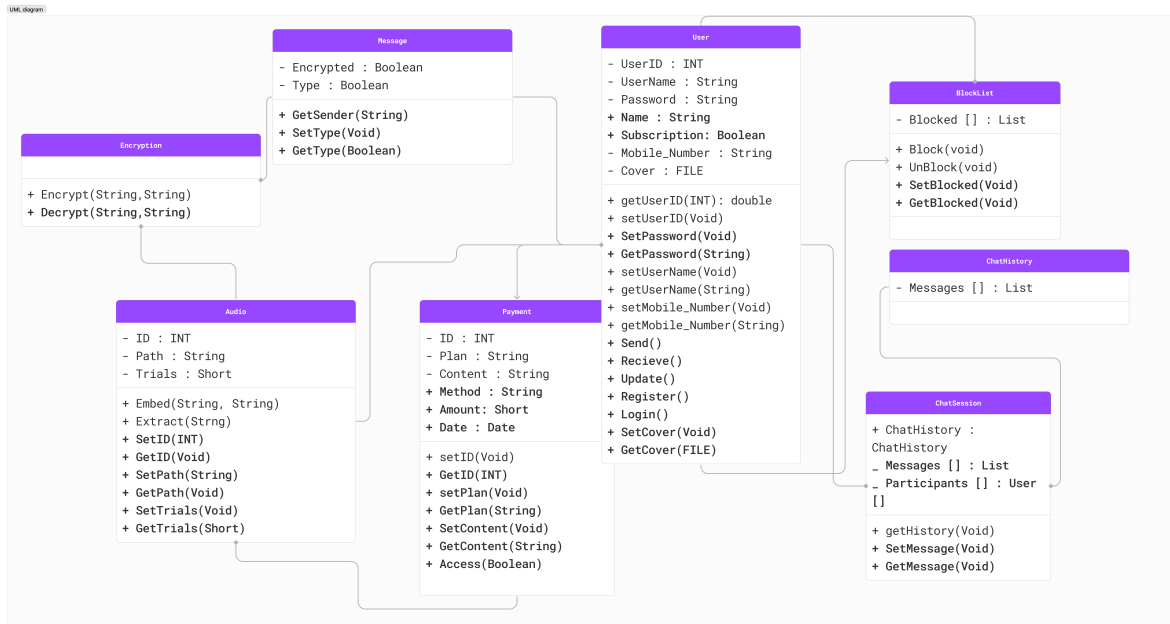


Figure 9: UML Diagram Overview

## 9 Operational Scenarios

### 9.1 Operational Scenario 1: User Registration

#### 9.1.1 Description

A new user, Alice, decides to join the secure chat application.

#### 9.1.2 Steps

- Alice opens the application and clicks on the "Sign Up" button.
- She enters her details, including username and password.
- The system prompts her to upload a cover audio file.
- After successful registration, Alice's account is created, and her cover audio is associated with her profile.

### 9.2 Operational Scenario 2: Initiating a Chat

#### 9.2.1 Description

Alice wants to initiate a secure chat with her friend Bob.

### **9.2.2 Steps**

- Alice logs into the application.
- She navigates to the chat initiation section and selects Bob as the recipient.
- The system prompts Alice to enter the text message she wants to send securely.
- Alice writes her message, and the system encrypts it using AES.
- The encrypted message is then hidden within her cover audio using steganography.
- Alice sends the cover audio to Bob.

## **9.3 Operational Scenario 3: Receiving and Decrypting a Message**

### **9.3.1 Description**

Bob receives a cover audio file from Alice.

### **9.3.2 Steps**

- Bob logs into the application.
- Bob navigates to his chat with Alice.
- The system detects the hidden encrypted message, decrypts it using AES.
- The decrypted message is displayed to Bob, revealing the original text.

## **9.4 Operational Scenario 4: Upgrading to Premium**

### **9.4.1 Description**

After using the free trial, Alice decides to upgrade to the premium version.

### **9.4.2 Steps**

- Alice receives a notification about her trial limit being reached.
- She clicks on the upgrade option in the notification.
- The system prompts her to choose a premium plan.
- After payment, Alice gains unlimited access to the application's features.

## **10 Project Plan**

### **10.1 Project Schedule**

The development of the ChatSafe system will be carried out in multiple phases, each focusing on specific aspects of the project. The project schedule is outlined below:

#### **10.1.1 Phase 1: Requirements Gathering and Analysis (Dec 2023 - Jan 2024)**

- Analyze and document user characteristics, system scope, and context.
- Conduct a comprehensive review of similar systems in academic and business applications.

#### **10.1.2 Phase 2: System Design and Architecture (Jan 2024)**

- Develop detailed system functions and specifications.
- Design constraints, including standards compliance and hardware limitations, are identified.
- Create an initial data design ER diagram.
- Conduct preliminary object-oriented domain analysis.

#### **10.1.3 Phase 3: Implementation (Feb 2024 - Mar 2024)**

- Begin the development of the ChatSafe web application.
- Implement the identified system functions and features.
- Ensure compliance with security standards, especially in the encryption and steganography processes.
- Focus on creating a user-friendly and intuitive interface.

#### **10.1.4 Phase 4: Testing and Quality Assurance (Apr 2024)**

- Conduct comprehensive testing of the implemented features.
- Address and fix any identified bugs or issues.
- Perform security testing to ensure the robustness of encryption and steganography processes.
- Ensure compliance with performance, reliability, and usability requirements.

#### **10.1.5 Phase 5: Deployment and User Training (May 2024)**

- Deploy the ChatSafe system for public access.
- Ensure a smooth transition for users from the testing environment to the live system.

## 10.2 Supportive Documents

### 10.2.1 Users-Survey

As shown in the figures below, here some results that we have from the survey we made. It proves that most people don't know what steganography is, that most people listen to music every day, so they will love to hide their message in music, and that most of them like to do this operation on a web application.

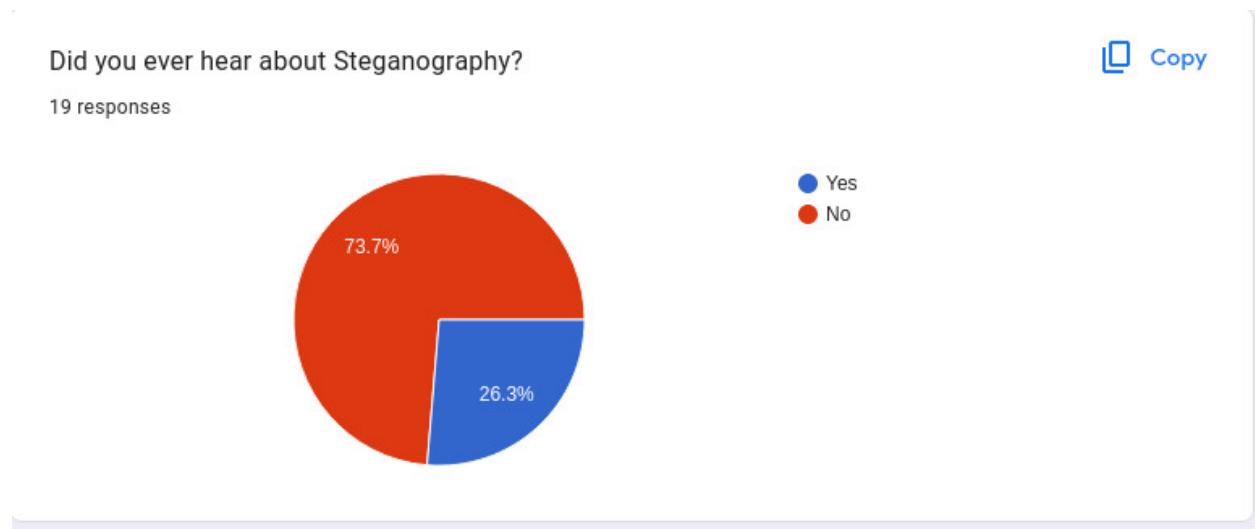


Figure 10: Chart 1

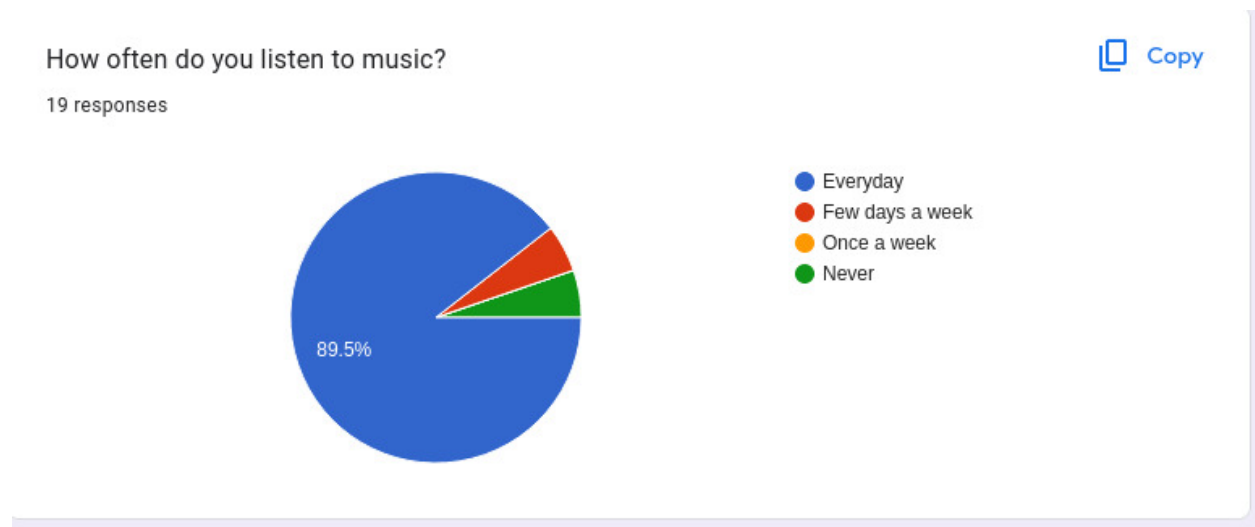



Figure 11: Chart 2

Have you ever used a method to protect your data other than passwords ?

 Copy

19 responses

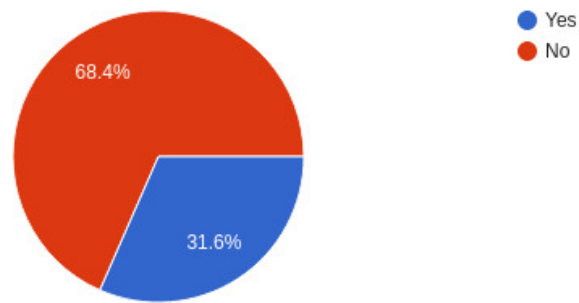


Figure 12: Chart 3

If someone introduced a new effective method to secure your data effectively would you try or use it?

 Copy

19 responses

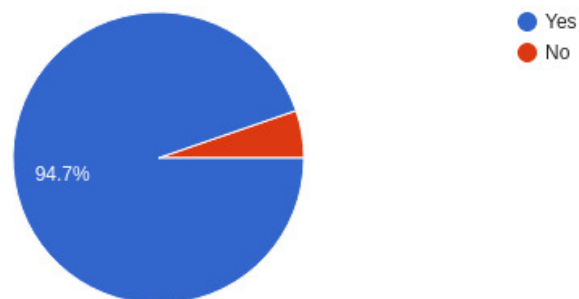



Figure 13: Chart 4



What type of medium would you prefer to hide your data?

 Copy

19 responses

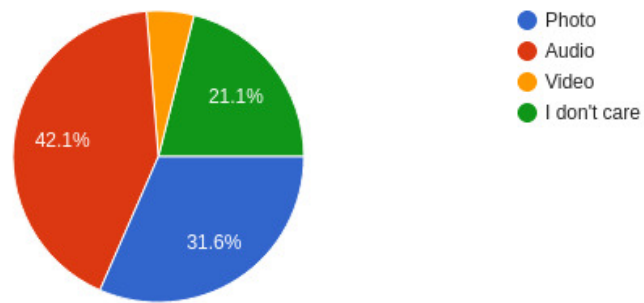


Figure 14: Chart 5

What type of application would you use to try the proposed method?

 Copy

19 responses

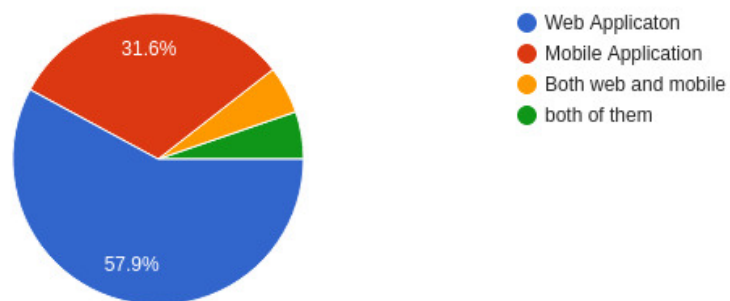


Figure 15: Chart 6

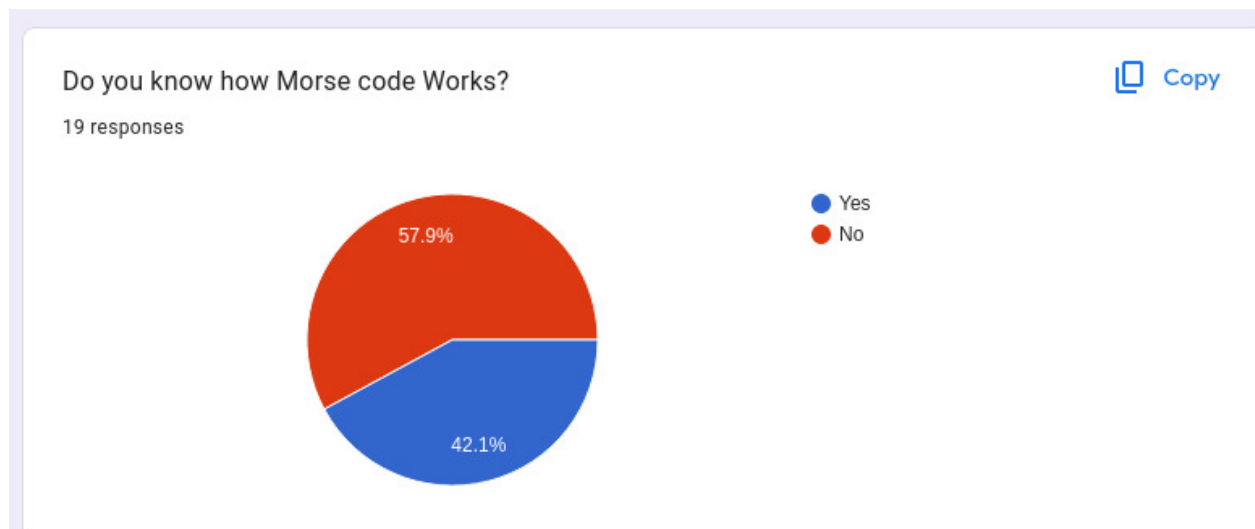


Figure 16: Chart 7

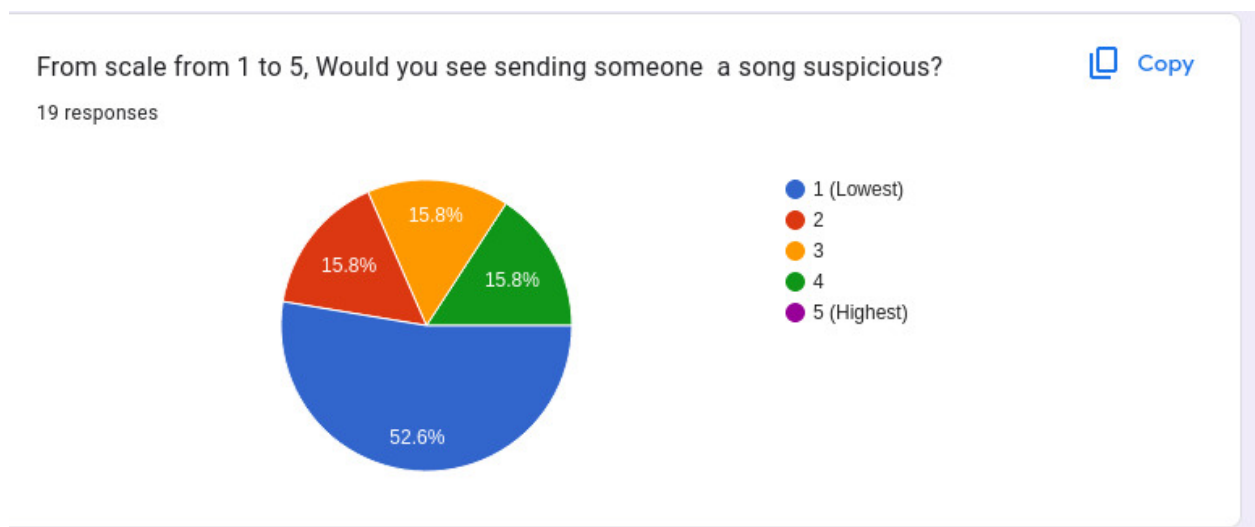


Figure 17: Chart 8

## References

- [1] Simplilearn. *Investing now can save millions*. <https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification>. 2023.
- [2] IBM. *Investing now can save millions*. <https://www.ibm.com/reports/data-breach>. 2023.
- [3] Simplilearn. *Investing now can save millions*. <https://www.simplilearn.com/what-is-steganography-article>. 2023.
- [4] Mohammed Majid Msallam and Fayez Aldoghan. “Multistage Encryption for Text Using Steganography and Cryptography”. In: *Journal of Techniques* 5.1 (2023), pp. 38–43.
- [5] Nouf Al-Juaid and Adnan Gutub. “Combining RSA and audio steganography on personal computers for enhancing security”. In: *SN Applied Sciences* 1 (2019), pp. 1–11.
- [6] P.G. Mamatha, T. Ravi Kumar Naidu, and T.V.S. Gowtham Prasad. “A Multi-Level Approach of Audio-Steganography and Cryptography”. In: *International Journal of Innovative Research in Computer and Communication Engineering* 2.4 (2014), pp. 56–61.
- [7] Simplilearn. *Investing now can save millions*. <https://www.statista.com/outlook/tmo/cybersecurity/worldwide>. 2023.