

**Tribhuvan University**

**Faculty of Humanities and Social Science**

**SECURE CHAT APPLICATION**

**USING RSA ALGORITHM AND LSB STEGANOGRAPHY**

**A PROJECT REPORT**

**Submitted to**

**Department of Computer Application**

**Padmakanya Multiple Campus**

**Bagbazar, Kathmandu**

***In partial fulfillment of the requirements for the Bachelors in Computer Application***

Submitted by

Saraswoti Shrestha (380143)

August, 2022

Under the Supervision of

**Mr. Kumar Prasun**



**Tribhuvan University**

**Faculty of Humanities and Social Science**

**Padmakanya Multiple Campus**

# **SUPERVISOR’S RECOMMENDATION**

I hereby recommend that this project prepared under our supervision by Mr. Kumar Prasun entitled “**SECURE CHAT APPLICATION USING RSA ALGORITHM AND LSB STEGANOGRAPHY**”in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Kumar Prasun

Project Supervisor, Program Director

BCA Department

Padmakanya Multiple Campus

Bagbazar, Kathmandu



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Padmakanya Multiple Campus**

# **LETTER OF APPROVAL**

This is to certify that this report is prepared by Saraswoti Shrestha entitled **“SECURE CHAT APPLICATION USING RSA ALGORITHM AND LSB STEGANOGRAPHY**” in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| **SIGNATURE of Supervisor**  Mr. Kumar Prasun  Supervisor  Padmakanya Multiple Campus, BCA | **SIGNATURE of HOD/Coordinator**  Mr. Kumar Prasun  Director  Padmakanya Multiple campus, BCA |
| Mr. Yogesh Deo  Internal Examiner | Prof. Subarna Shakya  External Examiner |

# **ABSTRACT**

Over the years, many techniques were developed to ensure the confidentiality of communicated information, with cryptography techniques being the most popular. Communication that is encrypted, however, can occasionally give an attacker enough information to provoke a full-scale attack and steganography is often needed to hide the existence of communicated information.

Steganography is not only the art of hiding data but also hiding the fact of transmission of secret data. Steganography hides the secret data in another file in such a way that only the recipient knows the existence of message.

Image Steganography is the technique of hiding the data within the image in such a way that prevents the unintended user from the detection of the hidden messages or data. The most common method used for image steganography is Least Significant Bit (LSB) substitution method. Every image has three components (RGB). This pixel information is stored in encoded format in bytes. In a grayscale image, the intensity of each pixel is stored in 8 bits (1byte). Similarly, for a color (RGB-red, green, blue) image, each pixel requires 24 bits (8bits for each layer). The least significant bits containing this information for every pixel can be modified to store the hidden text. For this, the preliminary condition is that the text to be stored has to be smaller or of equal size to the image used to hide the text. The human visual system cannot detect changes in the color or intensity of a pixel.

The paper titled “Secure Chat Application using RSA algorithm and LSB steganography” is basically concerned with secure communication among the users. The sender sends the plain text to the receiver. Before the message reaches to the sender the system encrypts the message using RSA algorithm and embed inside cover image and sends to the receiver. The receiver receives the image with encrypted text. The image is called as stego image. When the clicks on the image the encrypted text get decrypted with the RSA decryption algorithm and finally the secrete message is displayed on the screen. Thus, the application allows secure communication among the sender and the receiver.

# **ACKNOWLEDGMENT**

I would like to express my special thanks of gratitude to my supervisor **Mr. Kumar Prasun** who gave me the golden opportunity, guidance, inspiration and constructive suggestion to do this wonderful project on the topic “Secure Chat Application using RSA and LSB Steganography”, which also helped me in doing a lot of research and I came to know about so many tools and technologies. I would also like to express a deep sense of gratitude to him, for his exemplary guidance, monitoring and constant encouragement throughout which helped us in completing this task through various stages.

I appreciate all the personal and professional advice given by my teachers **Mr. Basanta Chapagain** and **Er. Shree Krishna Maharjan** during my working tenure.

Finally, I would also like to thank my family and friends who encouraged and helped me a lot in finalizing this project within the limited time frame.

**TABLE OF CONTENTS**

[**SUPERVISOR’S RECOMMENDATION** i](#_Toc112307054)

[**LETTER OF APPROVAL** ii](#_Toc112307055)

[**ABSTRACT** iii](#_Toc112307056)

[**ACKNOWLEDGMENT** iv](#_Toc112307057)

[**LIST OF FIGURES** vii](#_Toc112307058)

[**LIST OF TABLES** viii](#_Toc112307059)

[**ABBREVIATIONS** ix](#_Toc112307060)

[**CHAPTER 1** 1](#_Toc112307061)

[**INTRODUCTION** 1](#_Toc112307062)

[**1.1 Introduction** 1](#_Toc112307063)

[**1.2** **Problem Statement** 2](#_Toc112307064)

[**1.3** **Objectives** 2](#_Toc112307065)

[**1.4 Scope and Limitations** 3](#_Toc112307066)

[**1.5 Report Organization** 3](#_Toc112307067)

[**CHAPTER 2** 4](#_Toc112307068)

[**BACKGROUND STUDY AND LITERATURE REVIEW** 4](#_Toc112307069)

[**2.1 Background Study** 4](#_Toc112307070)

[**2.2 Literature Review** 4](#_Toc112307071)

[**CHAPTER 3** 6](#_Toc112307072)

[**SYSTEM ANALYSIS AND DESIGN** 6](#_Toc112307073)

[**3.1 System Analysis** 6](#_Toc112307074)

[**3.1.1 Requirement Analysis** 7](#_Toc112307075)

[**3.1.2 Feasibility Analysis** 9](#_Toc112307076)

[**3.1.3 Process Modelling: DFD** 9](#_Toc112307077)

[**3.2 System Design** 11](#_Toc112307078)

[**3.2.1 Architectural Design** 11](#_Toc112307079)

[**3.2.2 Interface Design** 13](#_Toc112307080)

[**3.2.3 Physical DFD** 15](#_Toc112307081)

[**3.3 Algorithm Details** 16](#_Toc112307082)

[**CHAPTER 4** 18](#_Toc112307083)

[**IMPLEMENTATION AND TESTING** 18](#_Toc112307084)

[**4.1 Implementation** 18](#_Toc112307085)

[**4.1.1 Tools used** 18](#_Toc112307086)

[**4.1.2 Implementation details of modules** 19](#_Toc112307087)

[**4.2 Testing** 19](#_Toc112307088)

[**4.2.1 Test Cases for Unit Testing** 19](#_Toc112307089)

[**4.2.2 Test Cases for System Testing** 21](#_Toc112307090)

[**CHAPTER 5** 24](#_Toc112307091)

[**CONCLUSION AND FUTURE RECOMMENDATIONS** 24](#_Toc112307092)

[**5.1 Conclusion** 24](#_Toc112307093)

[**5.2 Lesson Learnt/Outcomes** 24](#_Toc112307094)

[**5.3 Future Recommendations** 24](#_Toc112307095)

[**REFERENCES** 26](#_Toc112307096)

[**APPENDICES: SYSTEM SCREENSHOTS**](#_Toc112307097)

# **LIST OF FIGURES**

[Figure 3.1. 1 Iterative Waterfall Model 6](#_Toc112299346)

[Figure 3.1.1. 1 Use Case Diagram for Secure Chat Application 8](#_Toc112299358)

[Figure 3.1.3. 1 Context Diagram for Secure Chat Application 10](#_Toc112299388)

[Figure 3.1.3. 2 Level 1 DFD for Secure Chat Application 11](#_Toc112299389)

[Figure 3.2.1. 1 System Flow Secure Chat Application 12](#_Toc112299413)

[Figure 3.2.2. 1 Registration page of Secure Chat Application 13](#_Toc112299442)

[Figure 3.2.2. 2 Login screen of Secure Chat Application 14](#_Toc112299443)

[Figure 3.2.2. 3 Dashboard of Secure Chat Application 14](#_Toc112299444)

[Figure 3.2.3. 1 Physical DFD for Secure Chat Application 15](#_Toc112299460)

# **LIST OF TABLES**

[Table 4.2.1. 1 Test case for User Registration of Secure Chat Application 20](#_Toc112299514)

[Table 4.2.1. 2 Test case for User Login of Secure Chat Application 21](#_Toc112299515)

[Table 4.2.2. 1 Test Cases for System Testing of Secure Chat Application 22](#_Toc112299528)

# **ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| DFD  GUI  LSB  MySQL  PIL  RSA | :  :  :  :  :  : | Data Flow Diagram  Graphical User Interface  Least Significant Bit  My Structured Query Language  Python Imaging Library  Rivest, Shamir, Adleman |
| XAMPP  UI | :  : | **X (cross platform), Apache, MySQL, PHP, Perl**  User Interface |

# 

# **CHAPTER 1**

# **INTRODUCTION**

## **1.1 Introduction**

Communication through messaging over the internet has become the basic necessity of every growing area. However, security threats make human to think once before sharing message and are not able to share secret data to the respective receiver. Therefore, secret communication is very essential factor in information technology that continues to create challenges with increasing levels of sophistication as almost all communications occur through digital media. [1]

The secure chat application will aid in secure communication between users. The sender can send normal text and the receiver can get the encrypted message. Any other third person or intruder trying to steal message will not be able to easily read the message. This chatting application uses cryptography and steganography technique to encrypt the message sent by user to receiver.

Steganography is a technique that hides the secret data in another file in such a way that only the recipient knows the existence of message. In order to safely transmission of confidential data, the multimedia objects like audio, video, images are used as a cover source to hide the data. So, there is no knowledge of the existence of the secret message in the cover source. Image Steganography is the technique of hiding the data within the image in such a way that prevents the unintended user from the detection of the hidden messages or data. If a third person views the cover image in which the information is hidden inside, he or she will have no clue that there is any covering data, in this way the individual won’t endeavor to decode the data..

This chat application will first encrypt the sender’s message using RSA and then embed it inside image using LSB steganography. The receiver must decrypt the encrypted message before viewing the actual message. It becomes difficult for a naked eye to find the encrypted message in an image. Thus, this application provides secure transmission of messages among the users.

## **Problem Statement**

In today’s world, the communication is the basic necessity of every growing area. Everyone wants the secrecy and safety of their communicating data. In our daily life, we use many pathways like internet or telephone for transferring and sharing information, but it’s not safe at a certain level. The problem of unsecure communication is compounded by the fact that much of this information is sent over the public Internet and may be processed by third parties, as in e-mail or instant messaging (IM).

The goal of steganography is to avoid the suspicion to the existence of a hidden message. This approach of information hiding technique has recently become important in a number of application areas. Digital audio, video, and pictures are increasingly dramatically with distinguishing but imperceptible marks, which may contain a hidden notice or serial number or even help to prevent unauthorized copying directly. In this era, there are many hacking cases. Many industries such as bank, company worry that all their personal and industry data will be explore by unauthorized people. There are some of the techniques that used in steganography such as domain tools or simple system such as least significant bit (LSB) insertion and noise manipulation, transform domain that involve manipulation algorithms and image transformation such as discrete cosine transformation and wavelet transformation. However, there are some of the techniques that share the characteristic of both of the image and domain tools such as patchwork, pattern block encoding, spread spectrum methods and masking.

## **Objectives**

The proposed software system is expected to meet the following objectives:

* To securely communicate data in such a way that it remains confidential.
* To encrypt the message to be transmitted.
* To hide the encrypted message within a cover image in such a way that others cannot recognize the presence of the hidden message.

## **1.4 Scope and Limitations**

**Scope**

The scope of this project is very wide. There are many additional features, which are planned to be incorporated during the future enhancement of this project. Communication has become the basic necessity of every growing area over the internet.

The proposed system is developed for secure communication among the users. The users first need to register into the system and then login to the application using user name and password. The system encrypt the message sent by sender and embed inside an image and the image can be called as stego-image. The system then send the stego-image to the receiver. Receiver then receives the stego- image and have to click on it to receive the real message.

**Limitations**

The system is finally completed and well-developed as planned. However, there are several limitations in our system.

Some of the limitations of the project are listed below:

* Sender can send only text messages.
* Messages are not stored in database.
* Users must run simultaneously.

## **1.5 Report Organization**

Report organizing for each chapter that has been documentation is refer to specific format and it is easy to understand by the readers for the whole of the report. This report is begun with chapter 1 that consists of the introduction, problem statement, objectives, scopes and limitations of the system. For the next chapter, it explains about the Background and literature review which is explain about the related research paper for the similar system. For the chapter 3, it discussed about the System analysis, Requirement analysis (Functional and Non- Functional), Feasibility analysis (Economical, Technical, Operational) and, Process Modelling, System Design (Architectural Design, Interface Design, Physical DFD) that has been used during the development of the system and system design. Then, for chapter 4 deals with the implementation process and testing process. Lastly, chapter 5 consists of conclusion on the project development and Future Recommendations.

# **CHAPTER 2**

# **BACKGROUND STUDY AND LITERATURE REVIEW**

## **2.1 Background Study**

The proposed application is develop for secure communication. Over the years, many techniques were developed to ensure the confidentiality of communicated information, with cryptography techniques being the most popular. Communication that is encrypted, however, can occasionally give an attacker enough information to provoke a full-scale attack and steganography is often needed to hide the existence of communicated information.

The proposed system uses both cryptography and steganography technique to achieve secure communication. The application allows users to login into the system only after registration. The sender then enter message and send to the receiver. The system encrypts the plain message into cypher text using RSA encryption algorithm and then selects the random cover image to further embed the cypher text inside the image using LSB image steganography technique. The image generated is the secure stego image. The stego image is sent to the receiver. On the other side i.e. receiver, receives the stego image. Finally, when the image is clicked by the receiver, the system displays the secret text hidden in the stego image.

## **2.2 Literature Review**

There has been some work already done and find some gap to improve in existing technique.

According to Enhancing the Security and Quality of LSB Based Image Steganography, the proposed technique in a variation of plain LSB (Least Significant Bit) algorithm. The stego-image quality has been improving by using bit-inversion technique. Some researcher mainly focuses on image for data security. [2]

According to Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption, this paper discusses the design of a Robust image steganography technique based on LSB (Least Significant Bit) insertion and RSA encryption technique. Steganography is the term used to describe the hiding of data in images to avoid detection by attackers. Steganalysis is the method used by attackers to determine if images have hidden data and to recover that data. The application discussed in this paper ranks images in a user’s library based on their suitability as cover objects for some data. By matching data to an image, there is less chance of an attacker being able to use steganalysis to recover the data. Before hiding the data in an image, the application first encrypts it. The steganography method proposed in this paper and illustrated by the application is superior to that used by current steganography tools. [3]

Viber:

Viber is a calling and messaging app in which communication between the Viber Back End and clients is encrypted using HTTPS and is based on a proprietary TCP protocol. Servers are hosted on secure virtual private cloud (VPC) on AWS (Amazon Web Services). It uses end-to-end communication as chat data are not stored in any server after delivery of message. Viber has not published which algorithm is used for message encryption. [4]

# **CHAPTER 3**

# **SYSTEM ANALYSIS AND DESIGN**

## **3.1 System Analysis**

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives.

This system is designed with the series of processes starting with requirement gathering and analysis, design, coding and testing and maintenance. During requirement analysis, all the functional and nonfunctional requirement are analyzed and system is developed according to the requirement then designing of the system is carried out. After the design process, coding and development part is started then after unit testing followed by integrating the system and testing of the system. If the testing is positive then system is implemented otherwise some maintenance is done and again testing is done until system come in operation.

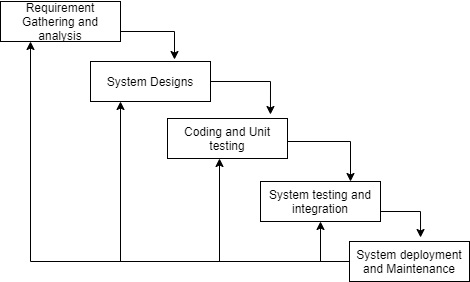


Figure 3.1. Iterative Waterfall Model

### **3.1.1 Requirement Analysis**

There are two types of requirements:

#### **i.** **Functional Requirements**

The user using the proposed Secure Chat Application can perform the following tasks:

**User registration**

Users must register to the application to log in to the system.

**User login**

Users must insert their credentials to use this application. Both sender and receiver have to be logged in to share their message.

**Message type**

The sender must type text message to send the message to the intendent receiver.

**Encryption and Decryption**

The text message transferred by this application will be in encrypted form. In order to view the message decryption process need to be carried out.

**Steganography**

On sender side, the sender must receive stego image which includes encrypted message hidden in the image. This mechanism adds an extra layer of security.

The overall process can be shown using use case diagram. The system involves an actor, the user. Here, two users are assumed as an agents between whom the communication is done. Both the user needs to create an account before having the conversation. After which they must login for further chatting process. The message send by the sender is encrypted in the form of cipher text where encryption is done based on RSA algorithm. Here, RSA algorithm is used to provide public key for both the sender and receiver in order to encrypt message in sender’s side and to decrypt message from receiver’s side. This helps to make the conversation secure from unknown sector. Also, steganography algorithm is implemented in order to provide more security.

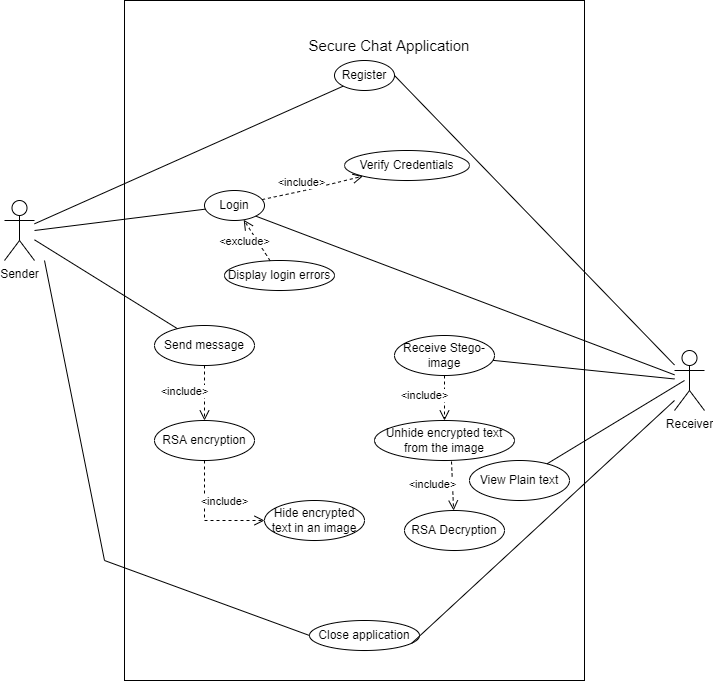


Figure 3.1.1. Use Case Diagram for Secure Chat Application

#### **ii.** **Non-functional Requirements**

The system should provide following non-functional requirements:

**i. Security and Reliability**

This application should win trust of the user. It must ensure that the messages being transferred will not be accessed by any unwanted hackers.

**ii. User Friendly**

Users won’t need any sort of high level knowledge to use the functionality of this application. The users with basic knowledge of login process and messaging process can send or receive information through message.

### **3.1.2 Feasibility Analysis**

A feasibility study considers many factors, including economic, technical, operational, and scheduling to determine whether a project can succeed.

It analyses the proposed system from different aspects so that it makes us clear that how the system will be practical and functioning properly.

1. **Technical**

Users can encrypt and decrypt message in one click so there will be no complexity. Users and developers are familiar with chatting application and how it works. I have used python language to build the application with the features like chatting with encryption so the project is technically simple.

#### **Operational**

The project uses RSA encryption algorithm in order to provide secure channel for messaging because of which users are free to share their secret message. Also for providing more security LSB steganography is used so that the privacy is maintained among the users.

Without any technical background, user can use this application. So, this project is operationally feasible.

#### **Economical**

This application is economically feasible because with limited budget and limited time the system can be easily develop using own laptop, hence it is economically feasible.

### **3.1.3 Process Modelling: DFD**

**Context level diagram:**

The diagram shown below is the context level diagram for Secure Chat Application.

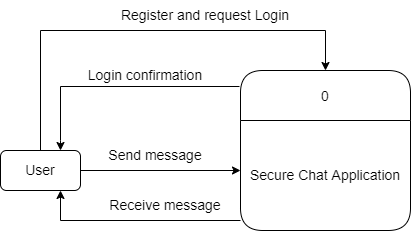
****

Figure 3.1.3. Context Diagram for Secure Chat Application

The context diagram shows that the users must register an account to log in to the system and then they can communicate securely.

**Level 1 DFD:**

In level 1 DFD, there are eight processes, in which the first two process shows registration and login process. After login users can communicate securely with each other. Sender sends plain message which is encrypted using RSA algorithm and further the encrypted text is embed inside image and this stego image is send to the receiver. The receiver receives the stego image. The encrypted text is extracted from image and further decrypted to get the plain message.

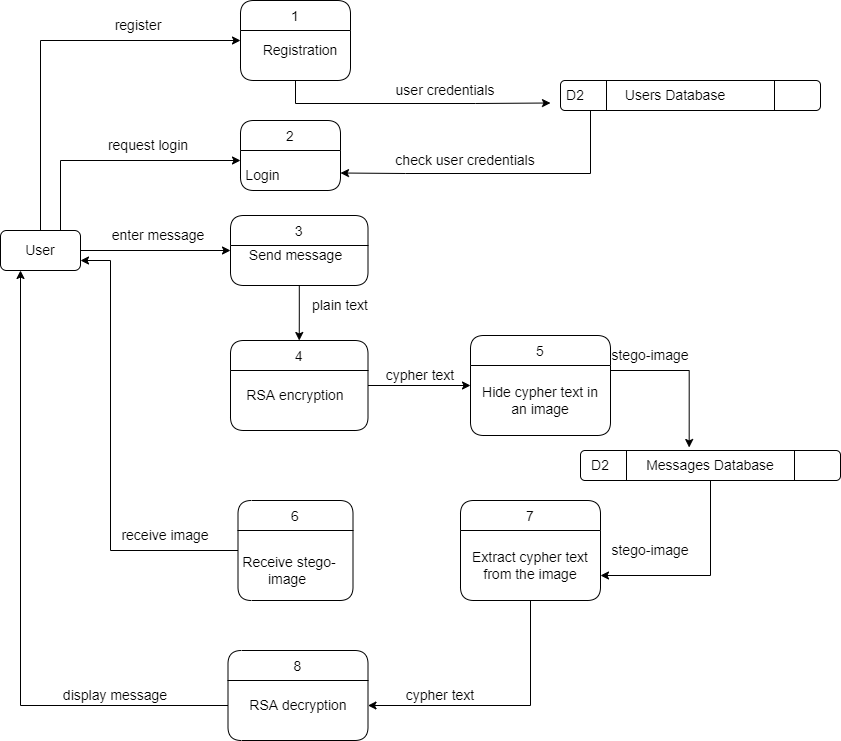
****

Figure 3.1.3. Level 1 DFD for Secure Chat Application

## **3.2 System Design**

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements

### **3.2.1 Architectural Design**

Architectural design represents the structure of data and program components that are required to build a computer-based system. The system architecture is the conceptual model that defines the structure, behavior and more views of the system.

**System Flowchart**

System flowcharts are a way of displaying how data flows in a system and how decisions are made to control events.

The overall process of Secure Chat Application can be explained with the help of flowchart as shown below:

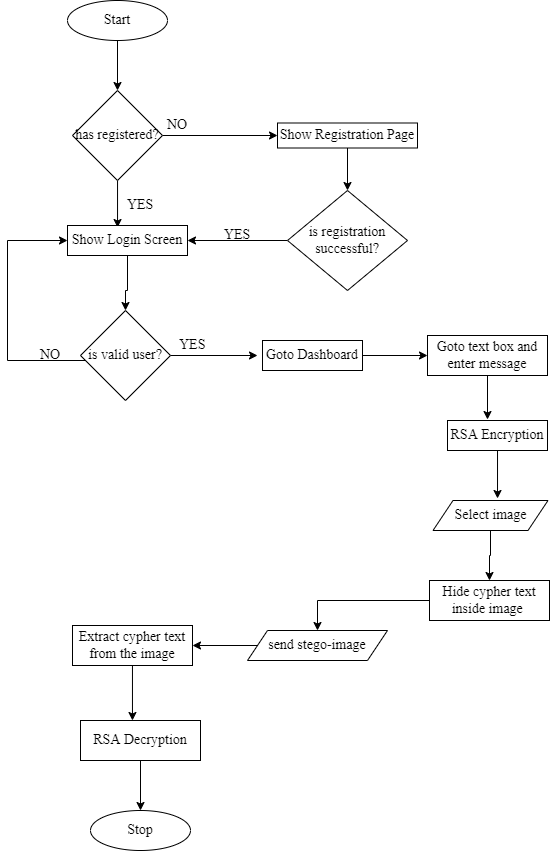


Figure 3.2.1. System Flow Secure Chat Application

In order to use the application, the user must register in to the system. After signing up, the user must login. The users can successfully log in to the system if their respective credentials match during login process. Then user will be directed to the dashboard after successful login, else the login page will be displayed. From the dashboard, the user will be able to send and receive message. The sender can write his/her desired message, which will be encrypted using RSA algorithm. Then encrypted message will be then embed inside an image and send to the receiver. The receiver will then receive an image. The encrypted text will first be separated from the image, then it will be decrypted to get the actual message.

### **3.2.2 Interface Design**

User interface is the front-end application view to which user interacts in order to use the software. It is used to build interfaces in system focusing on looks and styles.

Here we tried to design user friendly interfaces so that users find easy to use and pleasurable. The following are the UI design of Login page, Registration page, System dashboard.

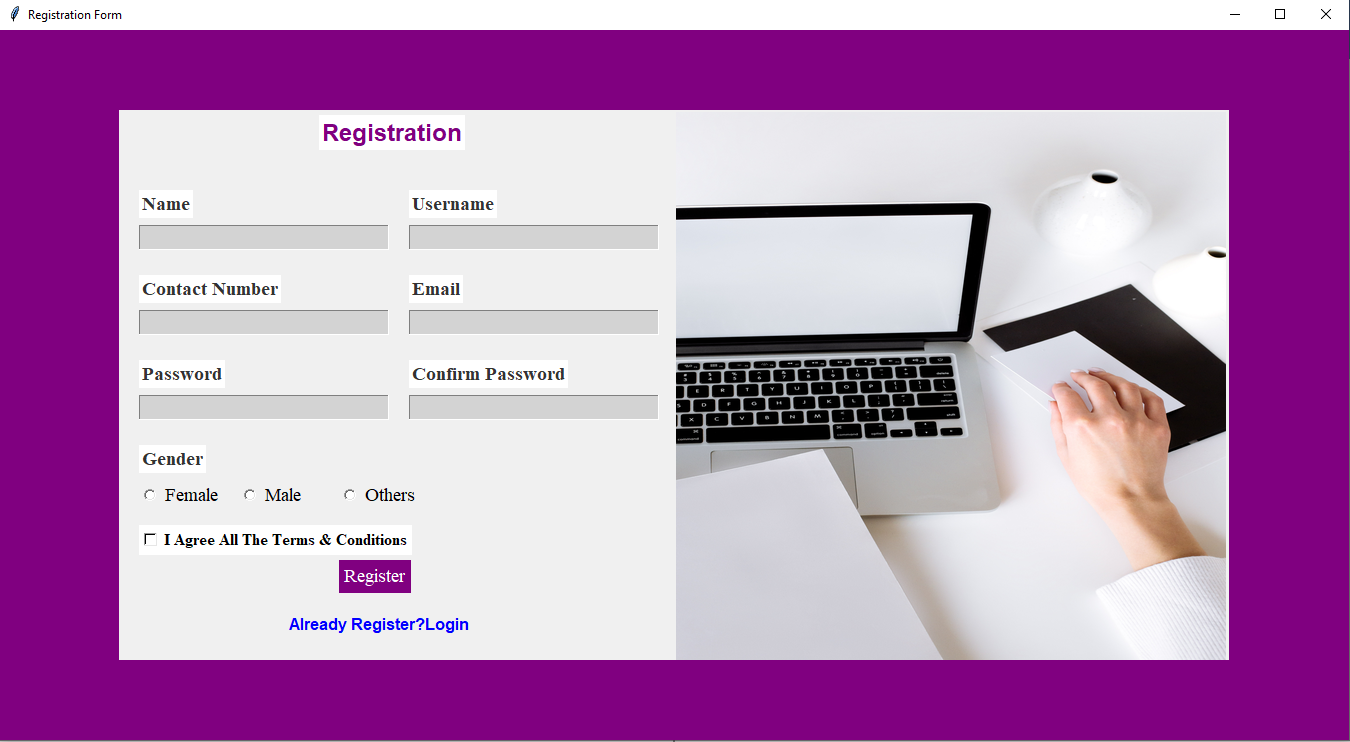


Figure 3.2.2. Registration page of Secure Chat Application

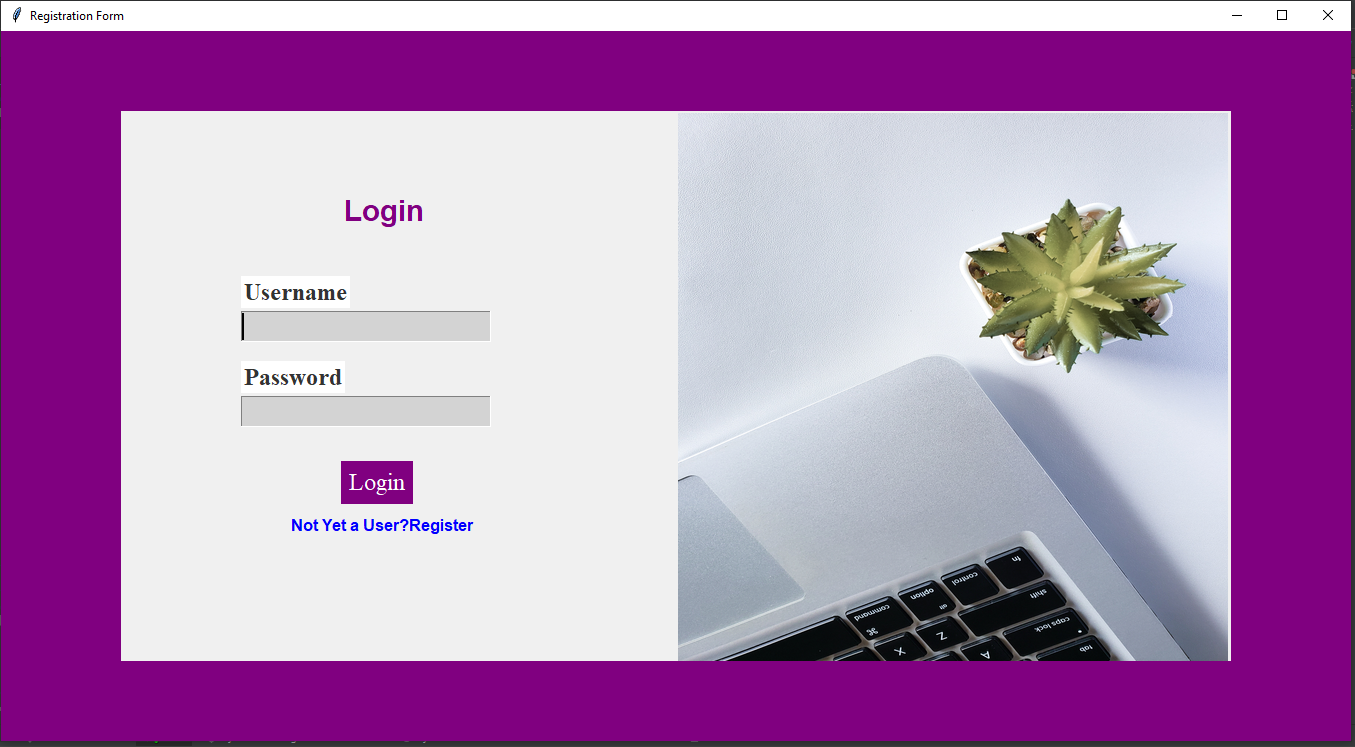


Figure 3.2.2. Login screen of Secure Chat Application

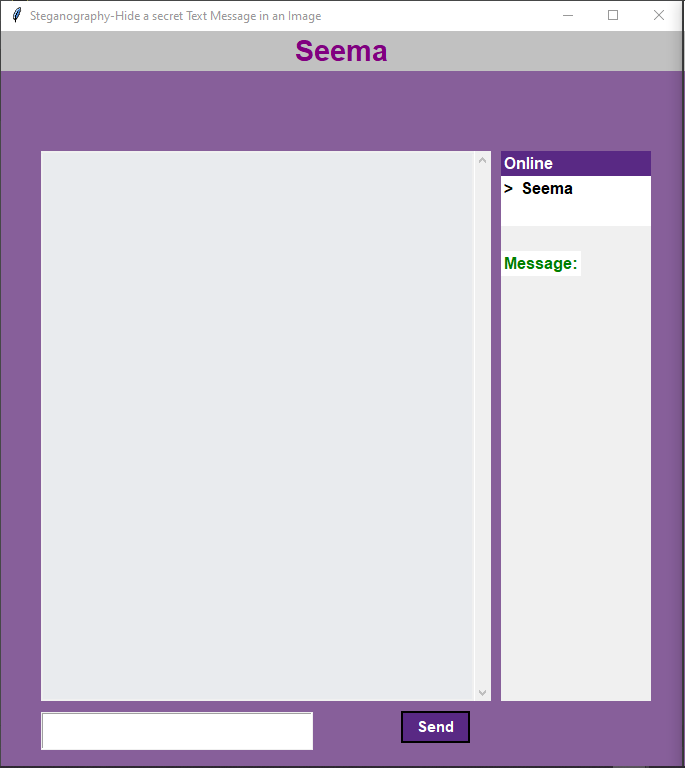


Figure 3.2.2. Dashboard of Secure Chat Application

### **3.2.3 Physical DFD**

Physical DFD depicts how the system will be implemented or how the current system operates.

The users must first log in to the system to use the application. All the users can successfully login to the system if the given credentials match with the information stored in database. After login users can communicate securely with each other. Sender sends plain message which is encrypted using RSA algorithm and further the encrypted text is embed inside image and this stego image is send to the receiver. The receiver receives the stego image. The encrypted text is extracted from image and further decrypted to get the plain message.

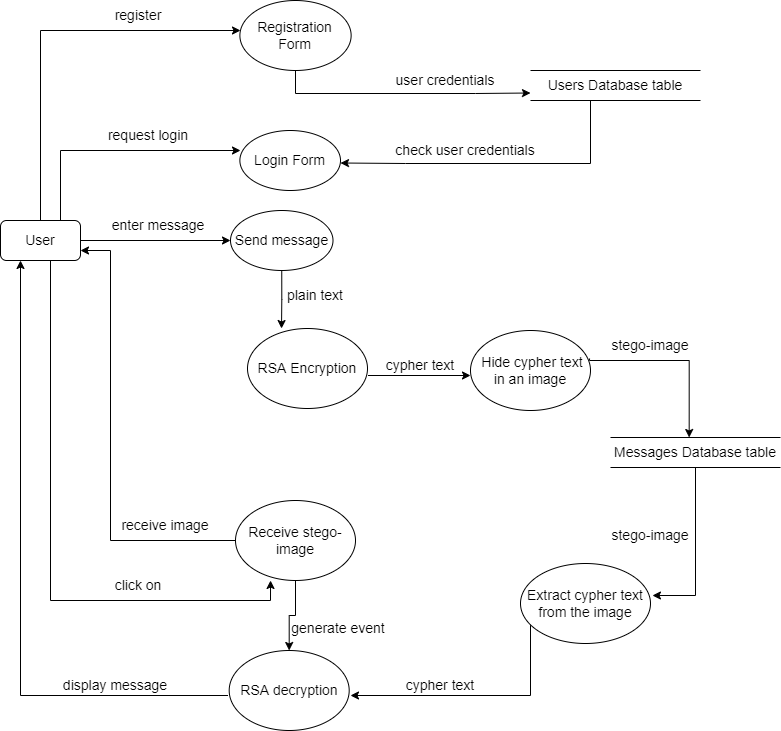


Figure 3.2.3. Physical DFD for Secure Chat Application

## **3.3 Algorithm Details**

**RSA algorithm for Cryptography:**

RSA algorithm is an asymmetric encryption also called as public key encryption technique which uses two distinct, yet related keys for encryption and decryption. The public key, is used for encryption and the other, private key is for decryption. RSA algorithm was invented by Rivest, Shamir and Adlemen in year 1978 and hence named as RSA algorithm.

Procedures to generate public and private keys:

* Select two large prime numbers, p and q.
* Multiply these numbers to find n=pxq, where n is called the modulus for encryption and decryption.
* Choose a number e less than n, such that n is relatively prime to (p - 1) x (q -1). It means that e and (p - 1) x (q - 1) have no common factor except 1. Choose "e" such that 1<e< φ(n), e is prime to φ(n), gcd(e, d(n))=1
* If n = p x q, then the public key is <e, n>. A plaintext message m is encrypted using public key <e, n>. To find cipher text from the plain text following formula is used to get cipher text C.

C= me mod n

Here, m must be less than n. A larger message (>n) is treated as a concatenation of messages, each of which is encrypted separately.

* To determine the private key, we use the following formula to calculate the d such that:

De mod {(p-1)x(q-1)}=1

Or,

De mod φ (n) = 1

* The private key is <d, n>. A cipher text message c is decrypted using private key <d, n>. To calculate plain text m from the cipher text c following formula is used to get plain text m.

m = cd mod n

**LSB Steganography:**

The LSB based technique is mainly uncomplicated and simple approach through which message bits are embedded within the least significant bits of cover image. In the LSB steganography method and for the purpose of covering the secret messages, the least significant bits of the cover-image are exploited. Thus, this method is considered one of the most common techniques that include the standard LSB replacement.

Consider the following cover-image and secret message in bits. The LSB replacement alternates the last bits of the cover image with each bit belong to the messages that are required to be hidden. The next example is to show the method of standard LSB replacement. The stego-image is the result after embedding the secret message.

Cover Image Pixels:

00110011 11101001 01101010 10101001

11011000 10001101 10001100 01101101

Secret Message:

1 0 0 0 1 1 1 0

Result (Stego-image):

00110011 11101000 01101010 10101000

11011001 10001101 10001101 01101100

For RGB, a technique alternates the least significant bits of each channel of Red, Green or Blue with the secret message bits. The result of the LSBs alternation causes minor changes in the RGB colors and therefore, it is difficult to be noticed by the human eye.

# **CHAPTER 4**

# **IMPLEMENTATION AND TESTING**

## **4.1 Implementation**

### **4.1.1 Tools used**

1. **Front-End Tool**

* **Tkinter module**

Tkinter is **Python's default GUI module** and also the most common way that is used for **GUI programming** in Python.

1. **Backend Tool**

The different backend tools used in this project are:

* **Python**

Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. It is used for backend purpose to create connectivity with database and also used in form validation, storing data in database tables, retrieving data form database tables, and other manipulation works.

* **Server**

XAMPP is a free and open-source cross-platform which provides the Apache web server, MySQL database (actually MariaDB), Php and Perl (as command-line executables and Apache modules) all in one package. Hence, in Secure Chat Application, MySQL database is used to store user details.

1. **Documentation Tool**

* **MS word**

Microsoft word is a word processing and documentation creation platform in which this document is created.

1. **Diagram Tool**

* **Draw.io**

This is used to generate diagrams such as use case, context, DFD for system analysis and design of the Secure Chat Application. Diagrams were created using this tool in order to save time since all components are available with drag and drop functions.

### **4.1.2 Implementation details of modules**

Various methods used are listed below:

* rsa.encrypt()

This method is used to encrypt the message of the sender using RSA algorithm.

* rsa.decrypt()

This method is used to decrypt the cypher text using RSA algorithm.

* encrypt\_text\_in\_image()

This method is used to hide encrypted message i.e. cypher text inside an image of format jpg or png. The generate image is stego image.

* decrypt\_text\_in\_image()

This method is used to extract encrypted message i.e. cypher text from the stego image.

## **4.2 Testing**

Testing is a very important phase for any types of software. A software should go through different testing process to ensure that the website is working in the manner in which it was intended to. During the phase of the development of the system, the system is tested time and again. The series of testings conducted are as follow:

### **4.2.1 Test Cases for Unit Testing**

Generally, a software constitutes of several different modules and so does this project. A modules or unit can refer to a function, individual program or even a procedure. Here is the list of test cases.

**User Registration**

Table 4.2.1. Test case for User Registration of Secure Chat Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Scenario** | **Input** | **Expected Result** | **Actual Result** | **Test Result** |
| 1 | Open Project in PyCharm | Python register.py | Open registration page | Open registration page | Pass |
| 2 | Enter invalid password | Name: Seema Lama  Username: Seema  Contact no.:9898989898  Email: [Seema123@gmail.com](mailto:Seema123@gmail.com)  Password: 123  Confirm Password: 123 | \*at least one uppercase, \*at least one lowercase, \*at least one digits \*may include special character \_ # @ % \* $ - \*minimum 6 character \*maximum 16 characters | Registration Failed | Pass |
| 3 | Enter valid password | Name: Seema Lama  Username: Seema  Contact no.:9898989898  Email: [Seema123@gmail.com](mailto:Seema123@gmail.com)  Password:Seema123  Confirm Password: Seema123 | Direct to Login page | Direct to Login page | Pass |

**User Login**

Table 4.2.1. Test case for User Login of Secure Chat Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Scenario** | **Input** | **Expected Result** | **Actual Result** | **Test Result** |
| 1 | Open Project in PyCharm | Python Client\_Chat.py | Open login page | Open login page | Pass |
| 2 | Enter invalid password | Username: Seema  Password: 123 | Invalid user or password | Login Failed | Pass |
| 3 | Enter valid password | Username: Seema  Password: Seema123 | Direct to the dashboard | Direct to the dashboard | Pass |

### **4.2.2 Test Cases for System Testing**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. It refers to system-integration testing which evaluates how the various components of an application interact together in the full, integrated system. System testing verifies that an application performs tasks as designed.

Table 4.2.2. Test Cases for System Testing of Secure Chat Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Scenario** | **Input** | **Expected Result** | **Actual Result** | **Test Result** |
| 1 | Register with empty fields | Name:  Username:  Contact no.:  Email:  Password:  Confirm Password: | All fields are Required. | All fields are Required. | Pass |
| 2 | Register with no username | Name: Seema Lama  Username:  Contact no.:9898989898  Email: [Seema123@gmail.com](mailto:Seema123@gmail.com)  Password:Seema123  Confirm Password: Seema123 | All fields are Required. | All fields are Required. | Pass |
| 3. | Register with wrong email format | Name: Seema Lama  Username: Seema  Contact no.:9898989898  Email: 123  Password:Seema123  Confirm Password: Seema123 | Please enter a valid email address. | Please enter a valid email address. | Pass |
| 4 | Register with different character in password and confirm password field | Name: Seema Lama  Username:  Contact no.:9898989898  Email: [Seema123@gmail.com](mailto:Seema123@gmail.com)  Password:Seema123  Confirm Password: Seema | Password didn’t match | Password didn’t match | Pass |
| 5. | Register with proper datad | Name: Seema Lama  Username: Seema  Contact no.:9898989898  Email: [Seema123@gmail.com](mailto:Seema123@gmail.com)  Password:Seema123  Confirm Password: Seema123 | Redirect to login page | Redirect to login page | Pass |
| 6. | Login with wrong credentials | Username:  Password: | All Fields are Required | All Fields are Required | Pass |
|  |  |  |  |  |  |
| 7. | Login with right characters | Username: Seema  Password: Seema123 | Redirect to dashboard | Redirect to dashboard | Pass |
| 8 | Sending message | Text | Message sent | Message sent | Pass |
| 9. | Receiving message | Click on received image | Display secret message | Display secret message | Pass |

# **CHAPTER 5**

# **CONCLUSION AND FUTURE RECOMMENDATIONS**

## **5.1 Conclusion**

The secure chat application has been developed successfully. The system is designed for secure communication. As we can see that many internet user uses unsecure ways to communicate among each other, the information can easily be leaked from one person to another. But with the help of this application sender and receiver can have secret conversation. Doing so the privacy can be maintained between the communicating users.

To achieve this facility, user must first register an account. Then after the registration process, they must login with verified username and password. Once the user is logged in they are able to communicate. The sender can simply send plain text to communicate with intended receiver. The text is encrypted using RSA algorithm. Further, the encrypted text is hidden inside an image file. The image is further sent to the destined user. The destined user, receiver, can then decrypt the received message using reverse algorithm of LSB technique and encryption.

To conclude, the project was successfully developed for having simple and secret conversation among users that helps in maintaining privacy.

## **5.2 Lesson Learnt/Outcomes**

The development of Secure Chat Application allow me to learn and gain the knowledge in different aspects. It enhanced my knowledge on cryptography and image steganography. Project development taught me how to turn ideas into reality. I learned lots of problem-solving skills, proper planning, communication, team work, proper use of guidelines, time management, writing skills and management of team in order to achieve goals.

## **5.3 Future Recommendations**

Main objective of the project is to provide secure chatting application using encryption and steganography method. This helps in maintaining the security. But, even though the application helps to perform secure conversation the application needs more of the features that needs to be added to improve its performance. Some of them can be given as:

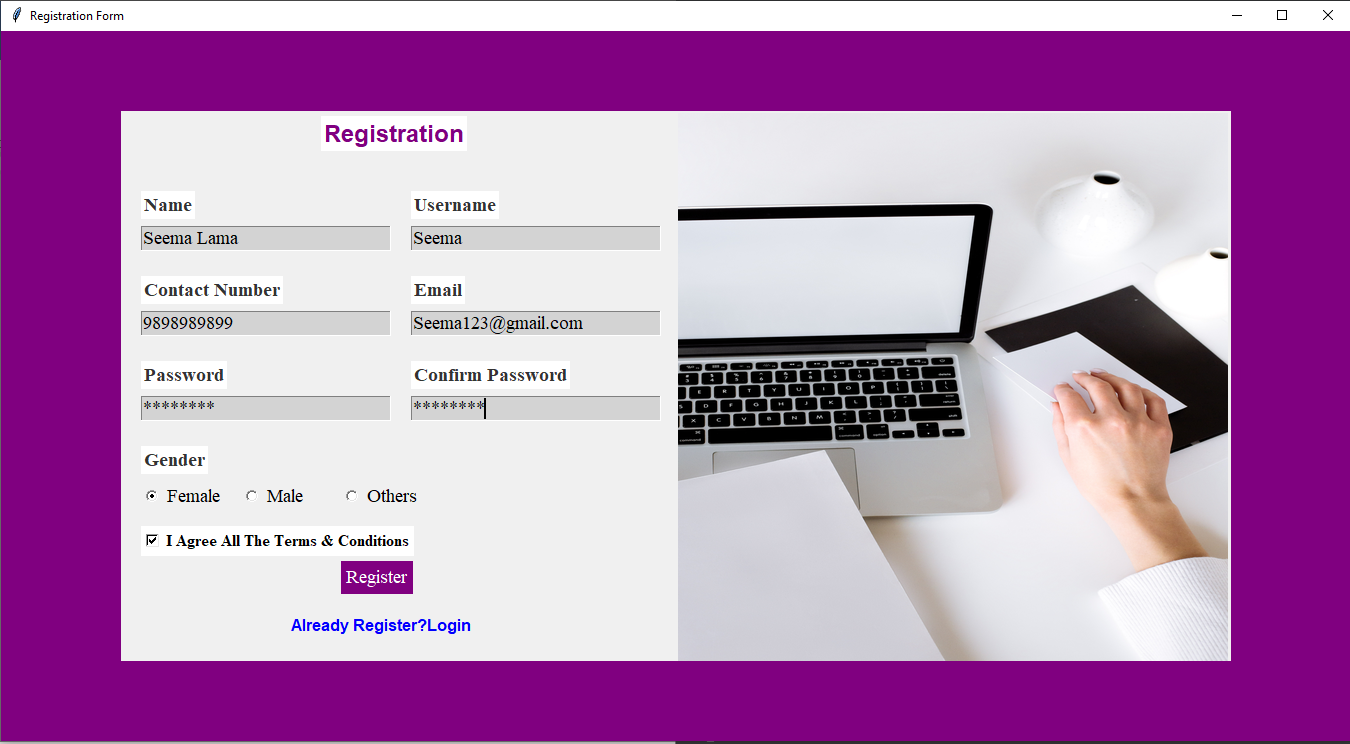
* The system only includes the encryption of text. Other data such as audio, video cannot be encrypted yet.
* Communication can only be done between two users. Group communication is not available.
* Communicated message is not stored into database.

# **REFERENCES**

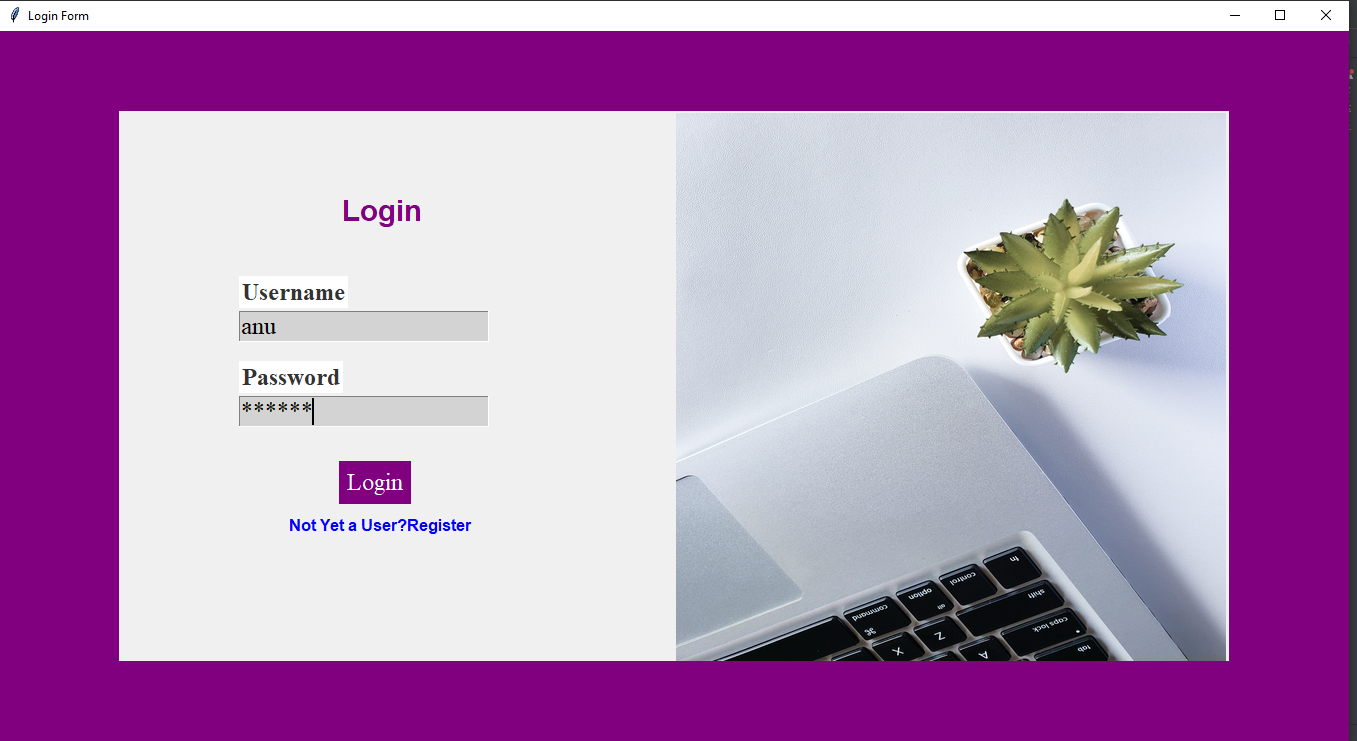
|  |  |
| --- | --- |
| [1] | T. Morkel, "IMAGE STEGANOGRAPHY APPLICATIONS," p. 141, 2012. |
| [2] | Nadeem Akhtar, "Enhancing the Security and Quality ofLSB based Image Steganography," p. 6, 2013. |
| [3] | M. Juneja, "Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption," pp. 302-305, 2009. |
| [4] | T. Sutikno, "WhatsApp, Viber and Telegram which is Best for Instant Messaging?," *Inernational Journal of Electrical and COmputer ENgineering(IJECE),* pp. 909-914, 2016. |

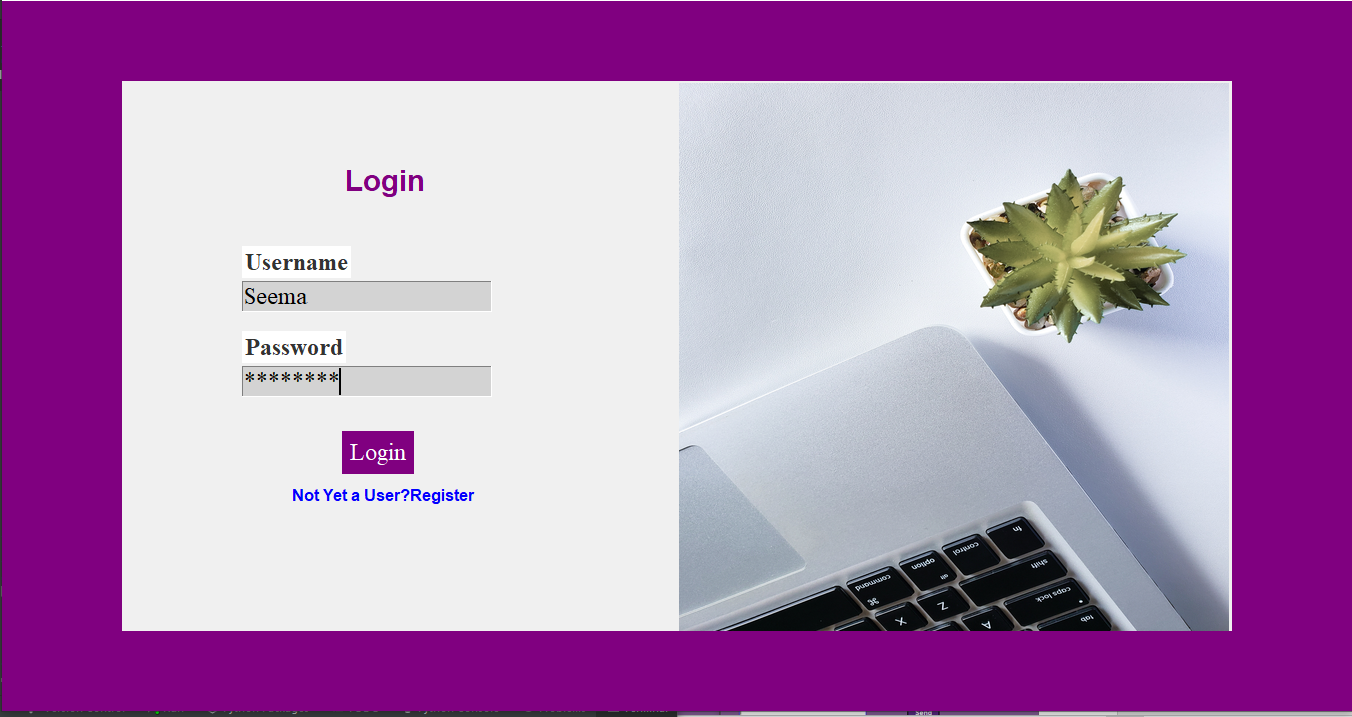
# **APPENDICES: SYSTEM SCREENSHOTS**

* **User Registering an account**

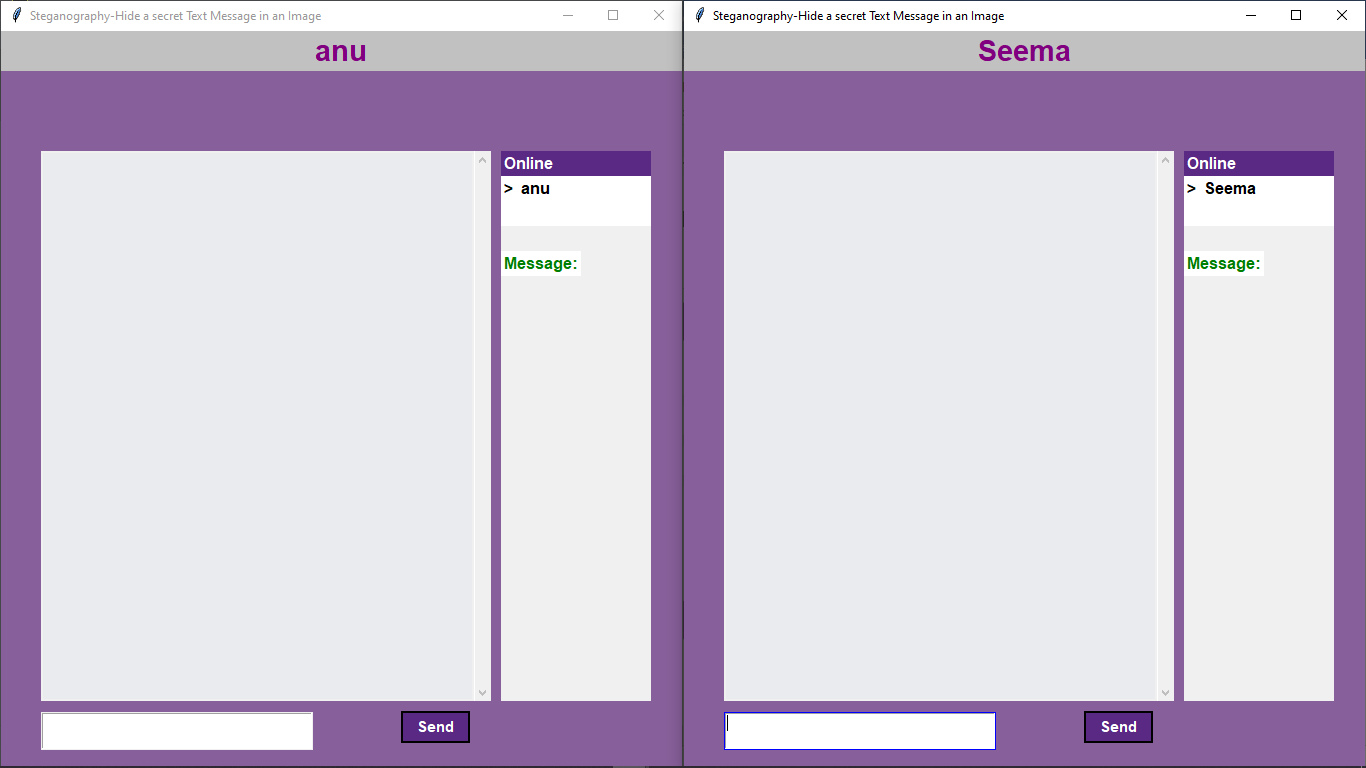


* **Users Logging to the system**

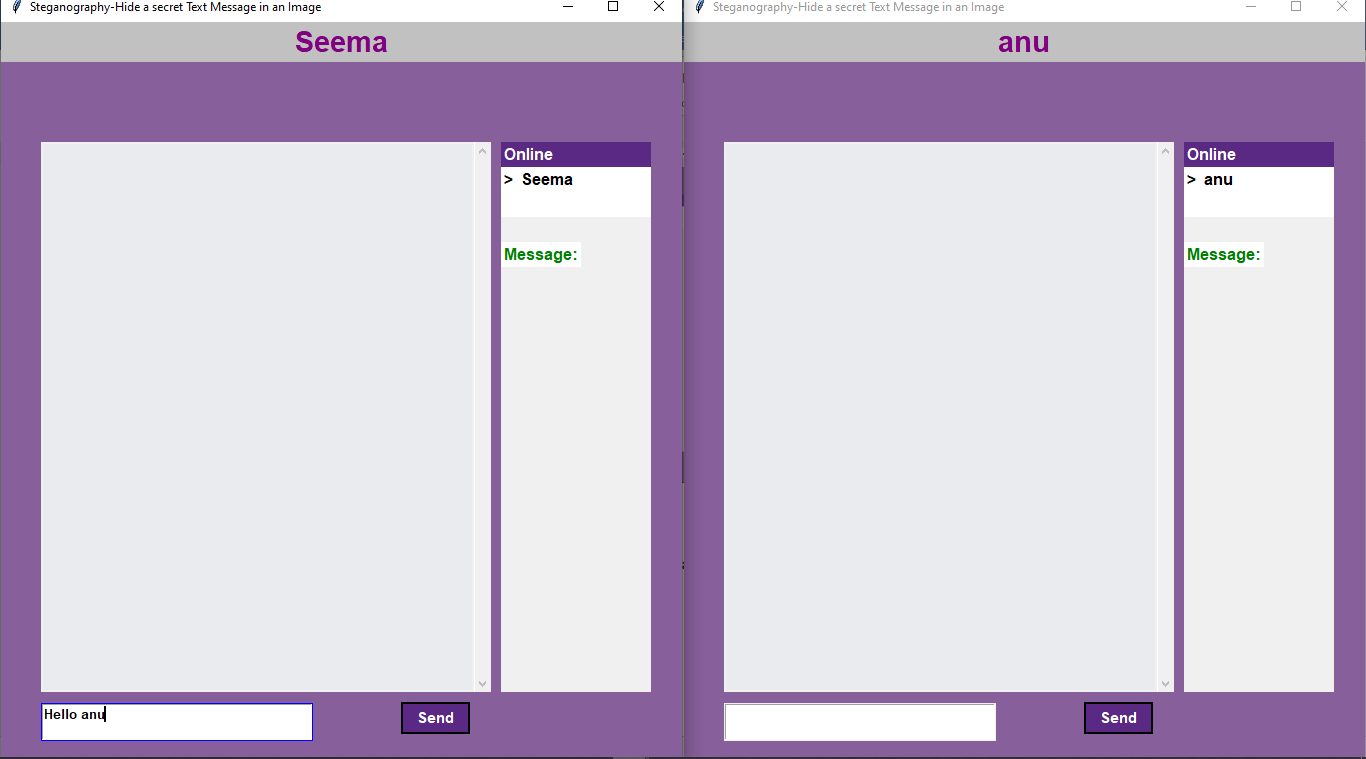




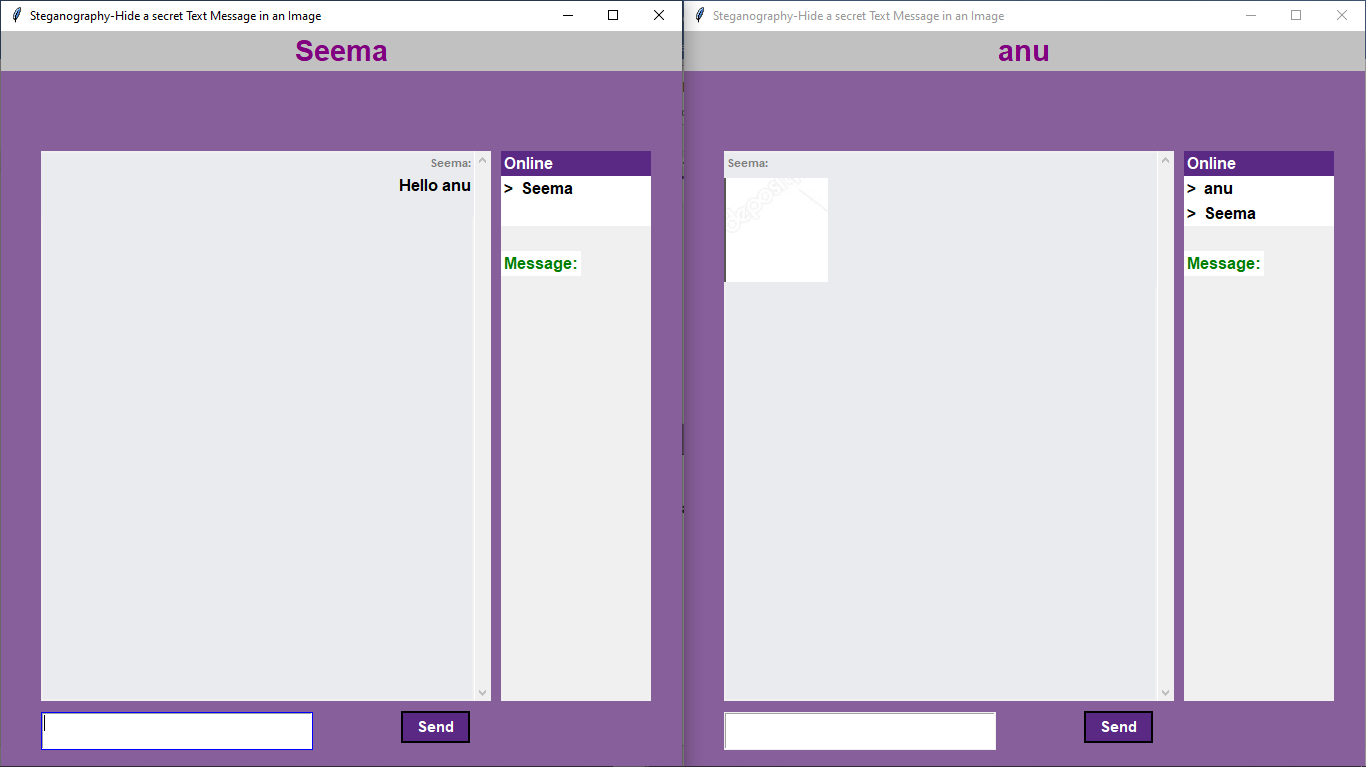
* **System Dashboard Page**



* **Sender Sending Message**



* **Receiver Receiving Stego-image**



* **Receiver Receiving the Secrete Message when Clicked on the Stego Image**

