SECURITY PACKAGE

Adarsh G - 20Z204
Jeevan Krishna K V - 20Z220
Viraj Agarwal - 20Z259
Nirmal M - 20Z267
Ajay Deepak P M - 21Z431
SundarSree B G - 21Z435

19Z512 - SOFTWARE PACKAGE DEVELOPMENT

BACHELOR OF ENGINEERING

BRANCH : COMPUTER SCIENCE AND ENGINEERING

NOVEMBER 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641004

19Z512 - SOFTWARE PACKAGE DEVELOPMENT

Bona fide record of work done by

Adarsh G - 20Z204
Jeevan Krishna K V - 20Z220
Viraj Agarwal - 20Z259
Nirmal M - 20Z267
Ajay Deepak P M - 21Z431
SundarSree B G - 21Z435

BACHELOR OF ENGINEERING

-

BRANCH: COMPUTER SCIENCE AND ENGINEERING

…………………………

[Dr. Kavitha]

Faculty Incharge

Certified that the candidates were examined in the viva-voce examination held on ……………………….

………………………… …………………………

(Internal Examiner) (External Examiner)

CONTENTS

CHAPTER **Page No.**

List of Figures……………………………………………………………………………….4
List of Tables……………………………………………………………………………….6
Acknowledgment……………………………………………………………………………. 8

1. Introduction………………………………………………………………………………… 9
1.1 Purpose……………………………………………………………………………………..9
1.2 Document Conventions……………………………………………………………………9
1.3 Intended Audience and Reading Suggestions……………………………………………..9
1.4 Product Scope…………………………………………………………………………………9

2. System Features………………………………………………………………………….10
2.1 Product Perspective…………………………………………………………………………10
2.2 Product Functions…………………………………………………………………………...10
2.3 User Classes and Characteristics……………………………………………………………11
2.4 Operating Environment……………………………………………………………………...12
2.5 Design and Implementation Constraints……………………………………………………12
2.6 User Documentation…………………………………………………………………………12
2.7 Assumptions and Dependencies……………………………………………………………12

3. System Architecture……………………………………………………………………13
3.1 Architectural Representation………………………………………………………………...13
3.2 Architectural Goals and Constraints……………………………………………………...14
3.3 UML Diagrams……………………………………………………………………………...14
3.3.1 Use Case Diagram……………………………………………………………………15
3.3.2 Use Case Description Table…………………………………………………………16
3.3.3 Activity Diagram……………………………………………………………………..18
3.3.4 Data Flow Diagram…………………………………………………………………..21
3.3.5 State Chart Diagram…………………………………………………………………22
3.3.6 Class Diagram………………………………………………………………………..23
3.3.7 Sequence Diagram.…………………………………………………………………24
3.3.8 Architecture Diagram.………………………………………………………………25

4. Implementation…………………………………………………………………………....26
4.1 Overview of User Interface…………………………………………………………………26
4.2 Interface Images……………………………………………………………………………...26

5. External Interface Requirements………………………………………………………33
5.1 User Interfaces.……………………………………………………………………………33
5.2 Hardware Interfaces………………………………………………………………………..33
5.2 Software Interfaces…………………………………………………………………………...33

6. Testing……………………………………………………………………………………34
6.1 Unit Testing…..……………………………………………………………………………...34
6.2 Integration Testing…………………………………………………………………………34
6.3 Incremental-Integration Testing……………………………………………………………....34
6.4 Functional Testing………………………………………………………………………...34
6.5 Acceptance Testing…………………………………………………………………….....34
6.6 Usability Testing…………………………………………………………………………….34
6.7 Test Cases……………………………………………………………………………………...35

7. Bibliography..……………………………………………………………………………..39

LIST OF FIGURES

S.NO	NAME OF THE FIGURE	FIGURE NUMBER	PAGE NUMBER
1	Use Case Diagram	3.1	15
2	Activity Diagram for Encryption and Decryption	3.2	18
3	Activity Diagram for OTP Generation and Authentication	3.3	19
4	Activity Diagram to perform Steganography	3.4	20
5	Level 0 Data Flow Diagram	3.5	21
6	Level 1 Data Flow Diagram	3.6	21
7	Level 2 Data Flow Diagram	3.7	21
8	State Chart Diagram	3.8	22
9	Class Diagram	3.9	23
10	Sequence Diagram	3.10	24
11	Architecture Diagram	3.11	25
12	OTP Generation and Authentication	4.1	26
13	Steganography (encryption)	4.2	27
14	Display Encrypted Image	4.3	27

15	Steganography (decryption)	4.4	28
16	Blowfish (encryption)	4.5	28
17	Blowfish (encrypted file)	4.6	29
18	Blowfish (decryption)	4.7	29
19	Blowfish (decrypted file)	4.8	30
20	RGB Technique (original file)	4.9	30
21	RGB Technique (encryption)	4.10	31
22	RGB Technique (encrypted image file)	4.11	31
23	RGB Technique (decryption)	4.12	31
24	Steganography (key error)	6.1	38
25	Steganography (file format error)	6.2	38

LIST OF TABLES

S.NO	NAME OF THE TABLE	TABLE NUMBER	PAGE NUMBER
1	Use Case Description Table for Encryption	3.1	16
2	Use Case Description Table for Decryption	3.2	16
3	Use Case Description Table for Steganography	3.3	17
4	Caesar Cipher Encryption Test Case	6.1	35
5	Caesar Cipher Decryption Test Case	6.2	36
6	Steganography Encryption Test Case	6.3	37

REVISION HISTORY

Name	Date	Reason For Changes	Version

ACKNOWLEDGEMENT

We express our deep sense of gratitude to Dr. K PRAKASAN, Principal, PSG College of Technology, for having provided the necessary environments to carry out our project successfully.

We profusely thank Dr. G SUDHA SADHASIVAM, Professor & Head, Department of Computer Science and Engineering who has given her support by giving permission to implement our idea.

We express our gratitude to our guide Dr. C KAVITHA, Assistant Professor (Selection Grade), Department of Computer Science and Engineering, for guiding us effectively and being a linchpin throughout the project cycle.

We would like to thank our tutor, Mrs. T ANUSHA, Assistant Professor (Senior Grade), Department of Computer Science and Engineering, for her support and guidance throughout the course.

We extend our thanks to all our department staff and our friends for their timely help and support.

1 INTRODUCTION

1.1 Purpose

The purpose of this project is to develop a new approach in hiding the secret information in an image, by taking advantage of the benefits of combining cryptography and steganography.

1.2 Document Conventions

In this SRS, font type 'Times' bold with font size '20' indicates the main headings, font type 'Times' bold with font size '14' indicates the subheadings and the entire description of each is given by the font type 'Times' with the font size of '12'.

1.3 Intended Audience and Reading Suggestions

The different types of readers that the document is intended for are the developers, integrators of all modules, testers, users (the tutor and others), etc… This Report contains the overall description, System features, System Architecture, Tools used. It also has a table of contents with respective page numbers helping the readers to move to the respective pages.

1.4 Product Scope

The Scope of this project is to provide security to the data of the users. For some users, the data might be changed by an unauthorized person in the network. Only Authorized persons i.e, the ones using our application can change the information. The scope of the project is to hide the data in an image using steganography, provide ciphers and ensure that the quality of concealing data must not be lost.

2 SYSTEM FEATURES

The package contains all basic ciphers along with some new ones, modules for steganography, and image hiding. This software contains the interface to access all the functionalities and a messenger interface for exchanging encrypted messages between two parties.

2.1 Product Perspective

The system allows a user to log in to their account and upload files (messages), encrypt the messages, and send them to a receiver. At the receiver's end, the acquired message is decrypted and the file can be interpreted. The user can also encrypt a message using an image(steganography) and can be decrypted at the receiver end using the same image and a key.

2.2 Product Functions

The product emphasizes the functionalities of the package. The secure transmission of a message through a network requires it to be encoded efficiently. Users can log in to their accounts and send encoded messages using different ciphers available whose key is only transparent to the user himself/herself and to the receiver. Some of the functions are stated below:

2.2.1 User Authentication and Authorization

Once connected to the internet, users will be able to enter into the system. In order to see the interface related to his/her role type, the user's account should be authorized, and also his/her username and password should be authenticated. These tasks are basically held by the functions implemented under the header of the Authentication and Authorization major function.

2.2.2 Ciphers

Ciphers that are implemented in this package are:

1. Caesar
2. Transpositional
3. Rail Fence
4. Monoalphabetic
5. Blowfish
6. Hill
7. Playfair

2.2.3 Encryption

The package provides the necessary modules to encrypt a message and the users can select the type of cipher and keys of their accord to encode the message. Encryption can be done in multiple folds using either the same cipher or different ones. The order of encryption using different ciphers of the message can also significantly increase the difficulty to crack the message.

2.2.4 Decryption

The receiver uses the modules to decrypt the message sent. The key to decrypting the message must be given to the receiver or else it is impossible to decrypt the message. The order and number of encryption ciphers must also be made transparent with the receiver. After decryption, the receiver can interpret the message.

2.2.5 Steganography

Steganography makes it possible to encode a message inside an image. Modules that can be used to implement the same require a key and an image to encrypt the message. Users must upload an image and the message along with a key. Then, encryption of the message into the image takes place. The image is then sent to the receiver's end who can decrypt the message out of the image using the key.

2.3 User Classes and Characteristics

The system provides various services depending on the type of users

1. Sender
2. Receiver

2.3.1 Sender

The senders or encryptors use modules that are specific to encryption.The senders can make use of different ciphers in the package.

2.3.2 Receiver

The receivers or decryptors use modules specific to decryption. The keys for decryption must be made transparent only to the indented receivers.

2.4 Operating Environment

This application is compatible with many browsers. For example, Google Chrome, Mozilla Firefox, Internet Explorer, etc. Since all these browsers render Streamlit (Python), it becomes a cross-platform package.

2.5 Design and Implementation Constraints

The package is developed using Python. The modules implemented are

- Ciphers
- Steganography
- OTP generation and authentication
- Image Encryption and Decryption
- Audio Encryption and Decryption

User Interfaces are implemented using Streamlit(Python)

2.6 User Documentation

This system includes features that are compatible with all the browsers that support Streamlit (Python). The package provides a description of all the features, functionality, classes, and functions used in it.

2.7 Assumptions and Dependencies

The Users (Sender and Receiver) of this package must have knowledge about how the encryption works and the format of keys provided for each cipher. There is no limitation in the operating system in which the package works. However, the system will need a server that is always online. Users can access the system with any internet browser.

3 SYSTEM ARCHITECTURE

3.1 Architectural Representation

This document describes the architecture using the views, using the RUP naming convention. The views used to document the security package are

Logical view

Audience: Designers.

Area: Functional Requirements: Describes the design's object model. Also describes the most important use-case realizations.

Related Artifacts: Design model.

Process view

Audience: Integrators.

Area: Non-functional requirements: Describes the design's concurrency and synchronization aspects.

Related Artifacts: No specific Artifact.

Implementation view

Audience: Programmers.

Area: Software components: Describes the layers and subsystems of the application.

Related Artifacts: Implementation model, Components.

Deployment view

Audience: Deployment managers.

Area: Topology: Describes the mapping of the software onto the hardware and shows the system's distributed aspects.

Related Artifacts: Deployment model.

Use Case view

Audience: All the stakeholders of the system including the end-users.

Area: Describes the set of scenarios and/or use cases that represent some significant and central functionality of the system.

Related Artifacts: Use Case Model, Use Case documents.

Data view

Audience: Data specialists, Database administrators

Area: Persistence: Describes the architecturally significant persistent elements in the data model**.**

Related Artifacts: Data model.

3.2 Architectural Goals and Constraints

This section describes the software requirements and objectives that have some significant impact on the architecture

Technical Platform

The package will be uploaded to PyPi and the users can install it using pip.

Security

The package allows users to encrypt and decrypt files before sending them over a public/unknown network. This means that attackers must not be able to track the algorithms or keys that are used to decrypt the encrypted data.

Reliability/Availability

The package is available 24/7 as there are no databases involved in the implementation of the system.

Internationalization

The final package will contain descriptions of all the functions, information of parameters used by each function and the return type. The package will undergo several changes before being uploaded to PyPi (Internationalization) and then will be available for installation.

3.3 UML Diagrams

A UML diagram is a diagram based on the Unified Modeling Language with the purpose of visually representing a system along with its actors, roles, actions, artifacts or classes, in order to understand, maintain, or document information about the system.

The UML diagrams used for this package:

- Use Case Diagram
- Activity Diagram
- Data Flow Diagram
- State Chart Diagram
- Class Diagram
- Sequence Diagram
- Architecture Diagram

3.3.1 Use Case Diagram

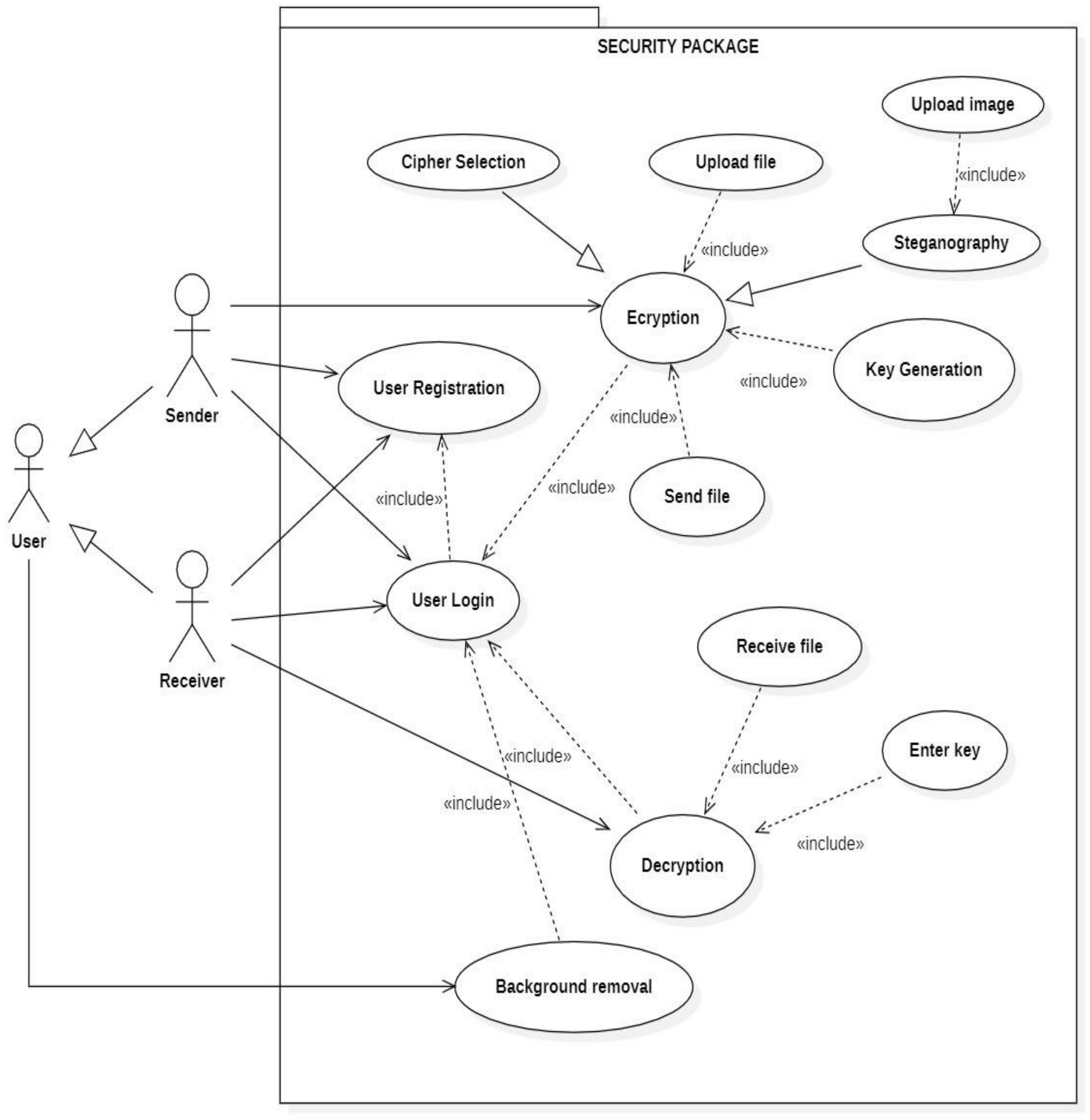


Figure 3.1 - Use Case Diagram

Description: This Use Case Diagram shows the user's possible interaction with the system and the dynamic behavior of the system. The User can use this package to perform the necessary encryption and decryption and their interaction with these modules is shown.

3.3.2 Use Case Description Table

Description: The Use Case Description table shows the description for the Use Cases. The description for the Use Cases are mentioned for Encryption, Decryption and Steganography.

Table 3.1 - Use Case Description Table for Encryption

Use Case ID:	001		
Use Case Name:	Encryption		
Created By:	Team 7	Last Updated By:	Team 7
Date Created:	09-08-2022	Date Last Updated:	15-08-2022

Actor:	Sender
Description:	The sender encrypts the file and sends
Preconditions:	The sender must register and login
Postconditions:	The receiver receives an encrypted file
Priority:	High
Frequency of Use:	When the user needs to send an encrypted file to a receiver
Normal Course of Events:	1. Sender logins 2. Uploads the file to be sent 3. Select the type of cryptography 4. Generates key 5. Encrypts the file 6. Sends the file to the receiver
Alternative Courses:	Sender cannot upload the file
Exceptions:	File size cannot be too large.
Includes:	Upload file, Key Generation, Send File
Special Requirements:	The system must have sufficient internet connectivity to use the modules of the security package. The system should have basic hardware configurations. RAM: 4GB Operating system: Windows, Linux
Assumptions:	Sender sends the key to the receiver
Notes and Issues:	The sender can choose the type of cryptography from different ciphers. The sender generates a key every time a message is encrypted.

Table 3.2 - Use Case Description Table for Decryption

Use Case ID:	002		
Use Case Name:	Decryption		
Created By:	Team 7	Last Updated By:	Team 7
Date Created:	09-08-2022	Date Last Updated:	15-08-2022

Actor:	Receiver
Description:	The receiver receives the file and decrypts it
Preconditions:	The Receiver must register and login
Postconditions:	The receiver reads the decrypted message
Priority:	High
Frequency of Use:	When the user needs to read a decrypted message

Normal Course of Events:	1. Receiver logins 2. Receives the encrypted file 3. Enters the key 4. Decrypts the file 5. Read the message
Alternative Courses:	Receiver didn't receive the file and key
Exceptions:	Incorrect Key: Key doesn't decrypt properly
Includes:	Receive file,Enter Key
Special Requirements:	The system must have sufficient internet connectivity to use the modules of the security package. The system should have basic hardware configurations. RAM: 4GB Operating system: Windows, Linux
Assumptions:	Sender sends the key to the receiver
Notes and Issues:	The receiver gets the key from the sender and the receiver decrypts the file based on the encryption algorithm used.

Table 3.3 - Use Case Description Table for Steganography

Use Case ID:	003		
Use Case Name:	Steganography		
Created By:	Team 7	Last Updated By:	Team 7
Date Created:	09-08-2022	Date Last Updated:	15-08-2022

Actor:	Sender
Description:	Sender uses steganography to encrypt the file.
Preconditions:	The sender must register and login
Postconditions:	File encrypted using steganography
Priority:	High
Frequency of Use:	When steganographic encryption is necessary
Normal Course of Events:	1. Sender logins 2. Uploads the file to be sent 3. Uploads the image for steganography 4. Encrypts the file 5. Sends the file to the receiver
Alternative Courses:	Sender cannot upload the file or the image Sender cannot encrypt the file
Exceptions:	File size cannot be too large.
Includes:	Upload image
Special Requirements:	The system must have sufficient internet connectivity to use the modules of the security package. The system should have basic hardware configurations. RAM: 4GB Operating system: Windows, Linux
Assumptions:	Sender sends the key to the receiver
Notes and Issues:	Based on the key, the image can be decrypted on the receiver side.

3.3.3 Activity Diagram

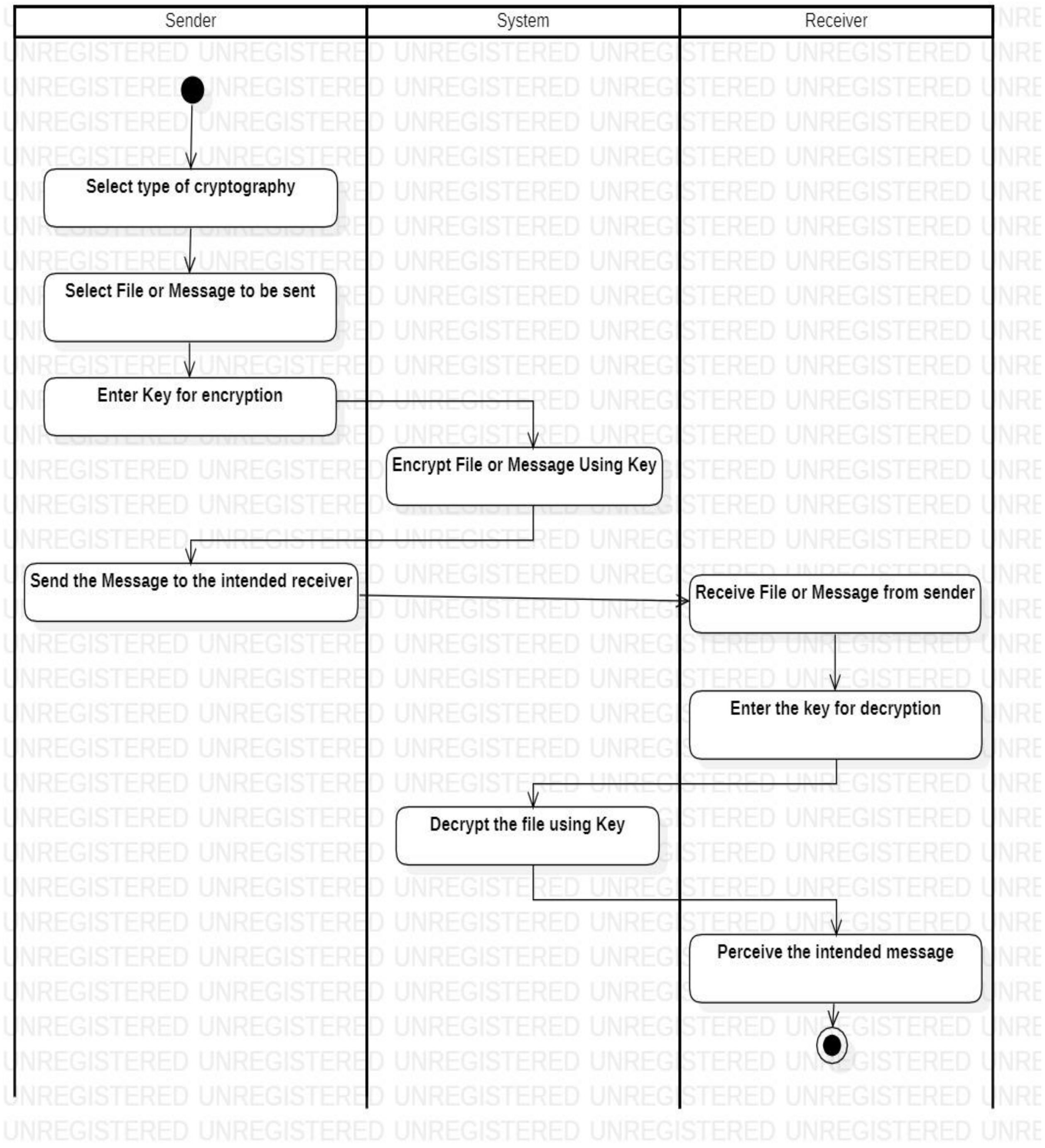

Sender
System
Receiver
Select type of cryptography
Select File or Message to be sent
Enter Key for encryption
Encrypt File or Message Using Key
Send the Message to the intended receiver
Receive File or Message from sender
Enter the key for decryption
Decrypt the file using Key
Perceive the intended message
UNREGISTERED

Figure 3.2 - Activity Diagram for Encryption and Decryption

Description: This Activity Diagram shows the workflow of Encryption and Decryption. In this diagram it shows the stepwise activities and actions for the Encryption and Decryption to be performed.

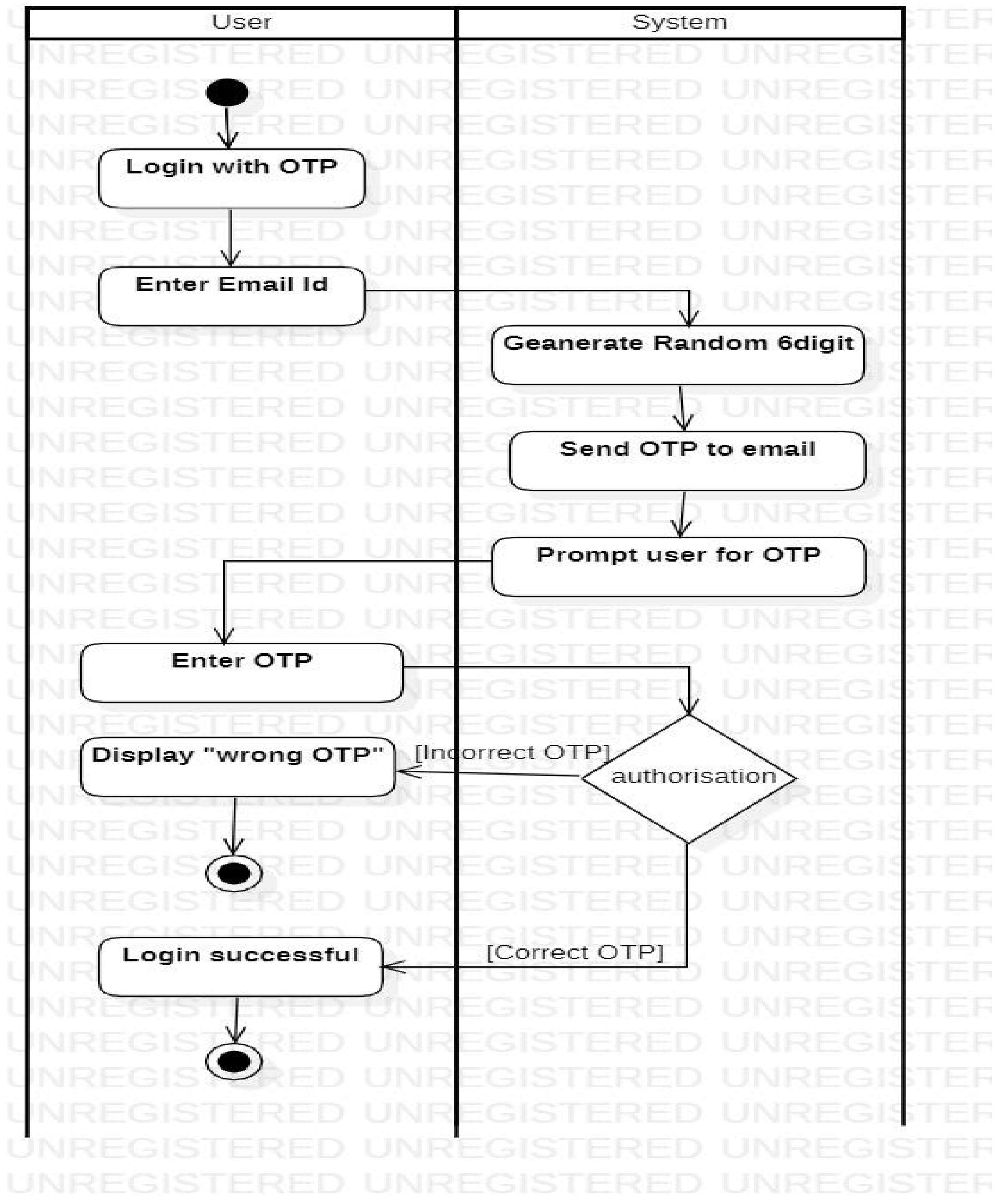


Figure 3.3 - Activity diagram for OTP Generation and Authentication

Description: This Activity Diagram shows the workflow of OTP generation and authentication. In this diagram it shows the stepwise activities and actions for the OTP generation and authentication to be performed.

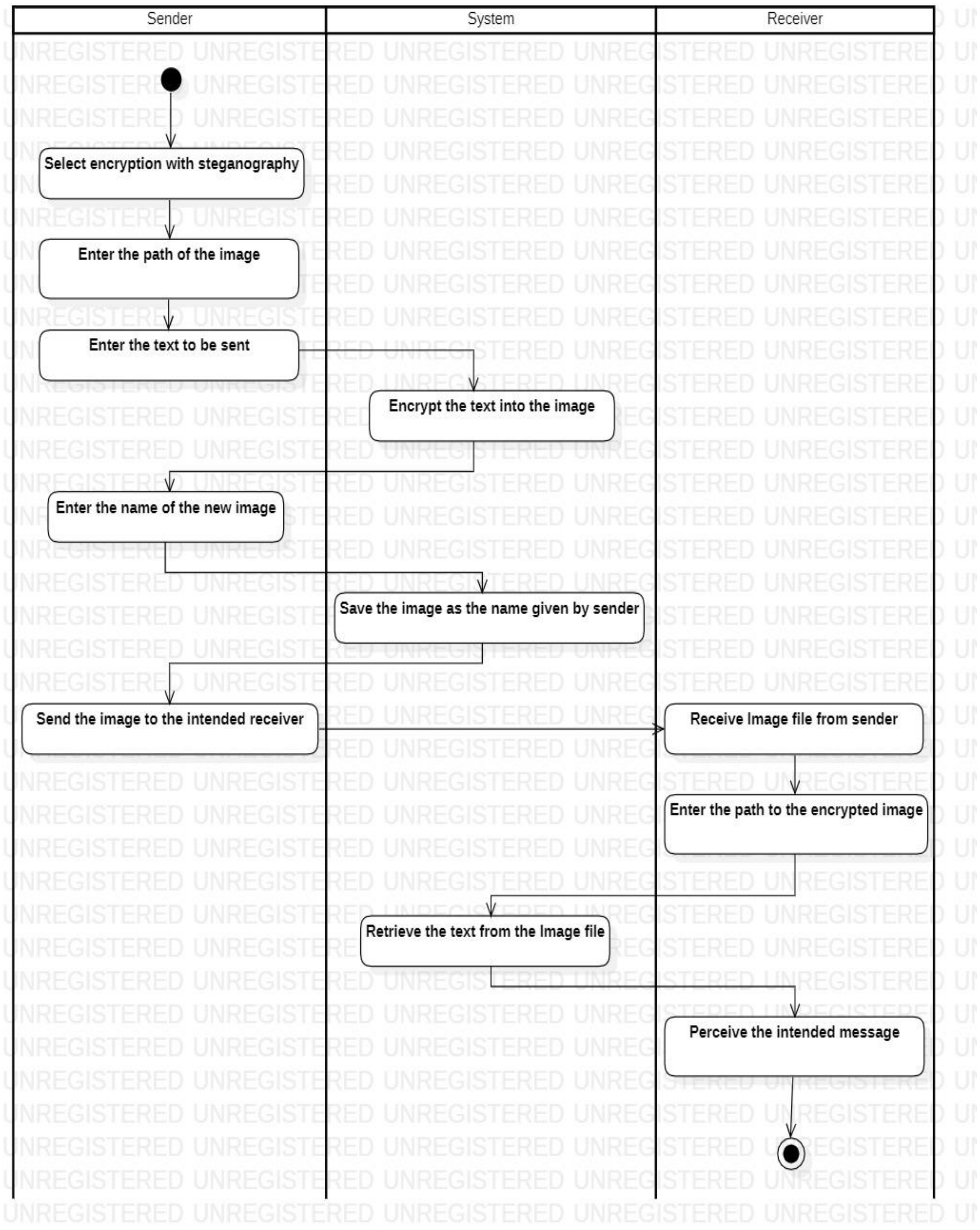

Sender
System
Receiver
Select encryption with steganography
Enter the path of the image
Enter the text to be sent
Encrypt the text into the image
Enter the name of the new image
Save the image as the name given by sender
Send the image to the intended receiver
Receive Image file from sender
Enter the path to the encrypted image
Retrieve the text from the Image file
Perceive the intended message

Figure 3.4 - Activity diagram to perform Steganography

Description: This Activity Diagram shows the workflow of Steganography. In this diagram it shows the stepwise activities and actions for the Steganography to be performed.

3.3.4 Data Flow Diagram

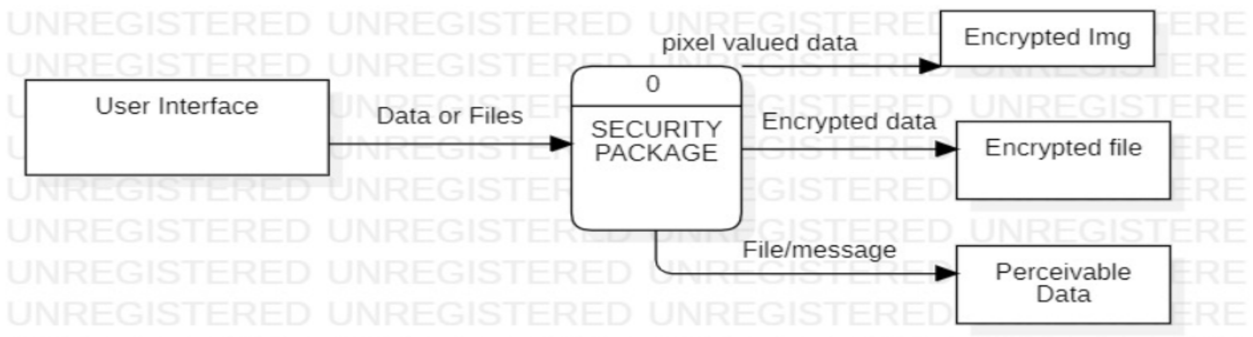


Figure 3.5 - Level 0 Data Flow Diagram

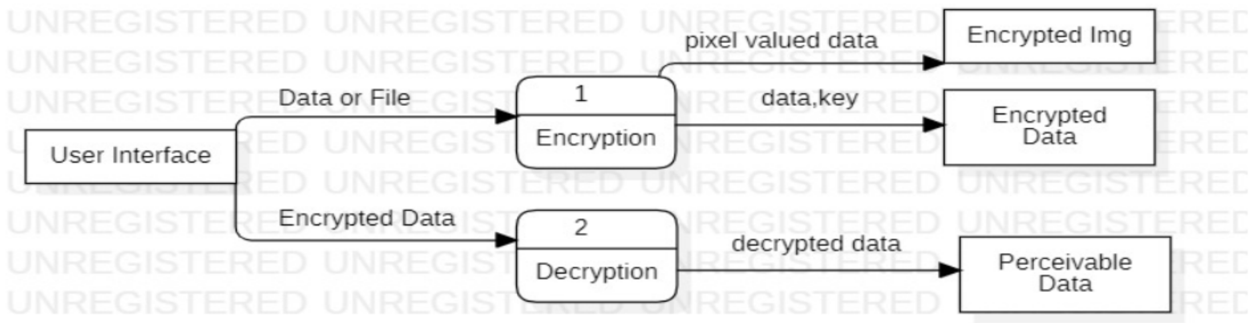


Figure 3.6 - Level 1 Data Flow Diagram

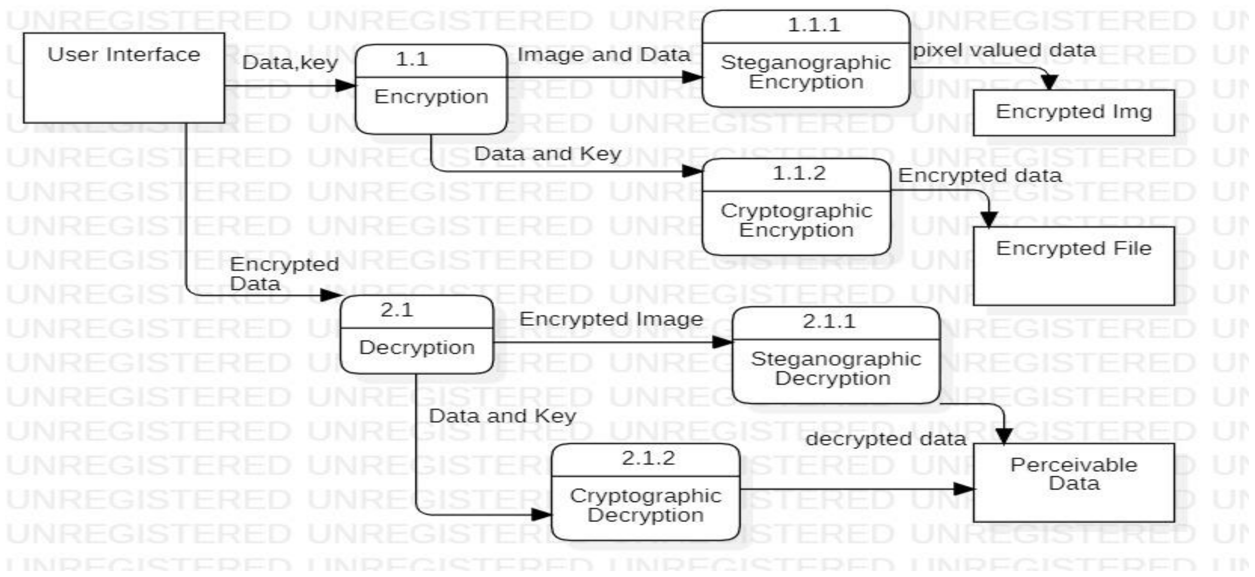


Figure 3.7 - Level 2 Data Flow Diagram

Description: The Data Flow Diagram represents the flow of data through a process or a system. The Data Flow Diagram for this package is shown in three levels with respective inputs and outputs. This Data Flow Diagram focuses on the process of Encryption and Decryption.

3.3.5 State Chart Diagram

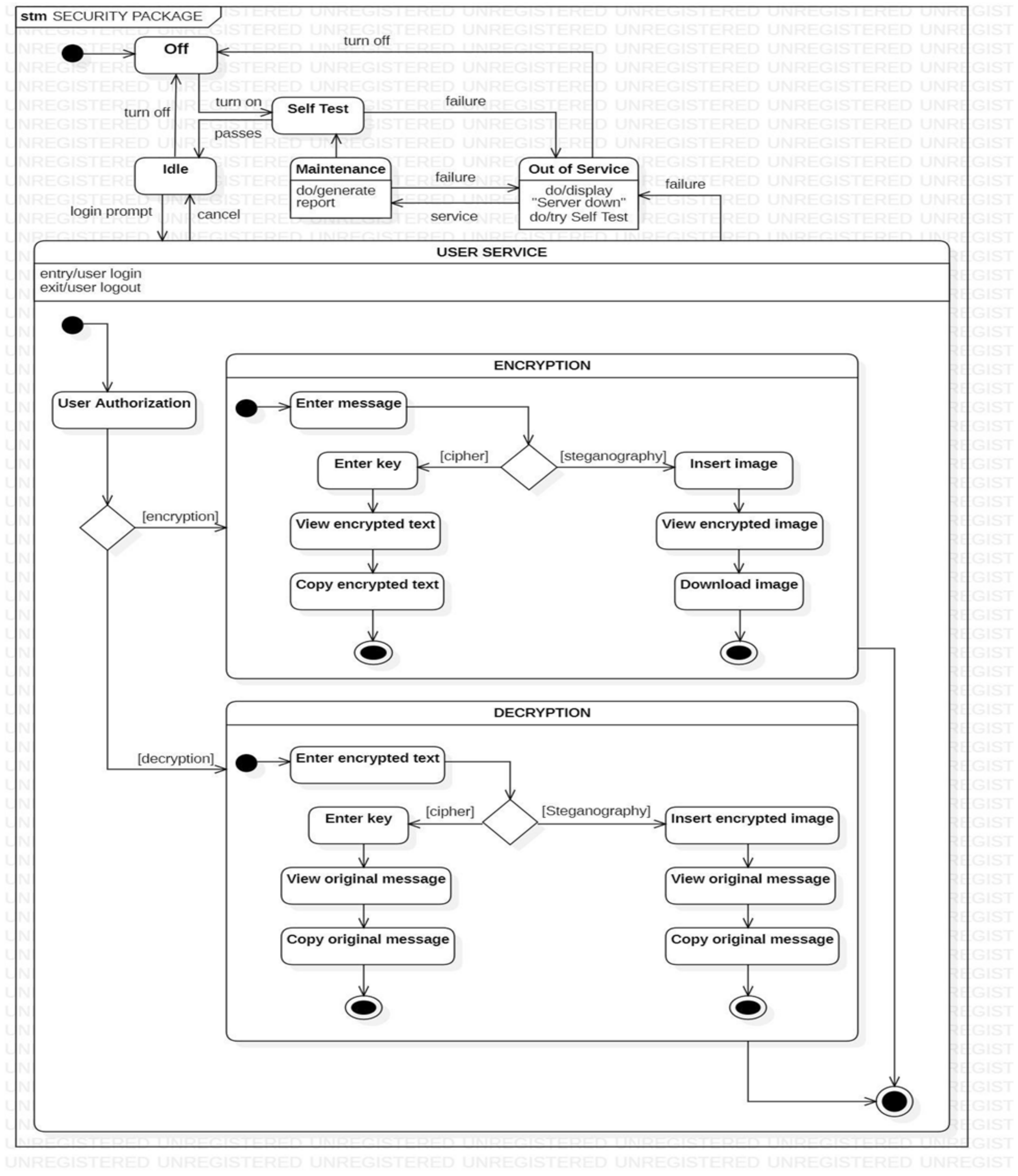


Figure 3.8 - State Chart Diagram

Description: The State Chart Diagram shows the states of different objects and their interactions in the system. In this diagram, different states while performing Encryption and Decryption and while using this package.

3.3.6 Class Diagram

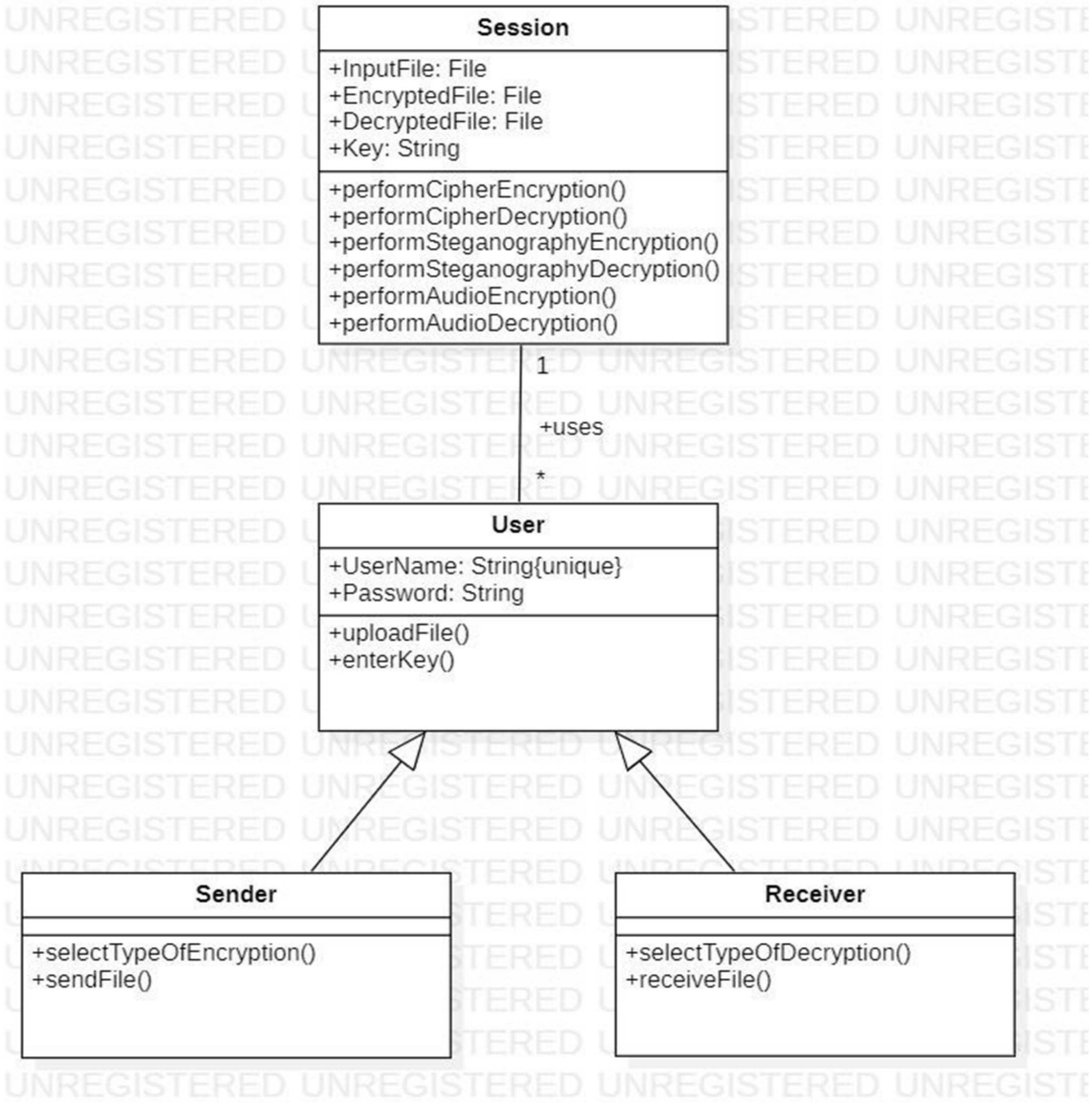

Session
+InputFile: File
+EncryptedFile: File
+DecryptedFile: File
+Key: String
+performCipherEncryption()
+performCipherDecryption()
+performSteganographyEncryption()
+performSteganographyDecryption()
+performAudioEncryption()
+performAudioDecryption()
1
+uses
*
User
+UserName: String{unique}
+Password: String
+uploadFile()
+enterKey()
Sender
+selectTypeOfEncryption()
+sendFile()
Receiver
+selectTypeOfDecryption()
+receiveFile()
UNREGISTERED

Figure 3.9 - Class Diagram

Description: The Class Diagram describes the structure of the package, classes, operations and their relationship. In this diagram we have four classes: Session, User, Sender and Receiver. The Sender and Receiver classes are generalized to the User class. The Class diagram shows the relationship and the functions User performs in a particular Session.

3.3.7 Sequence Diagram

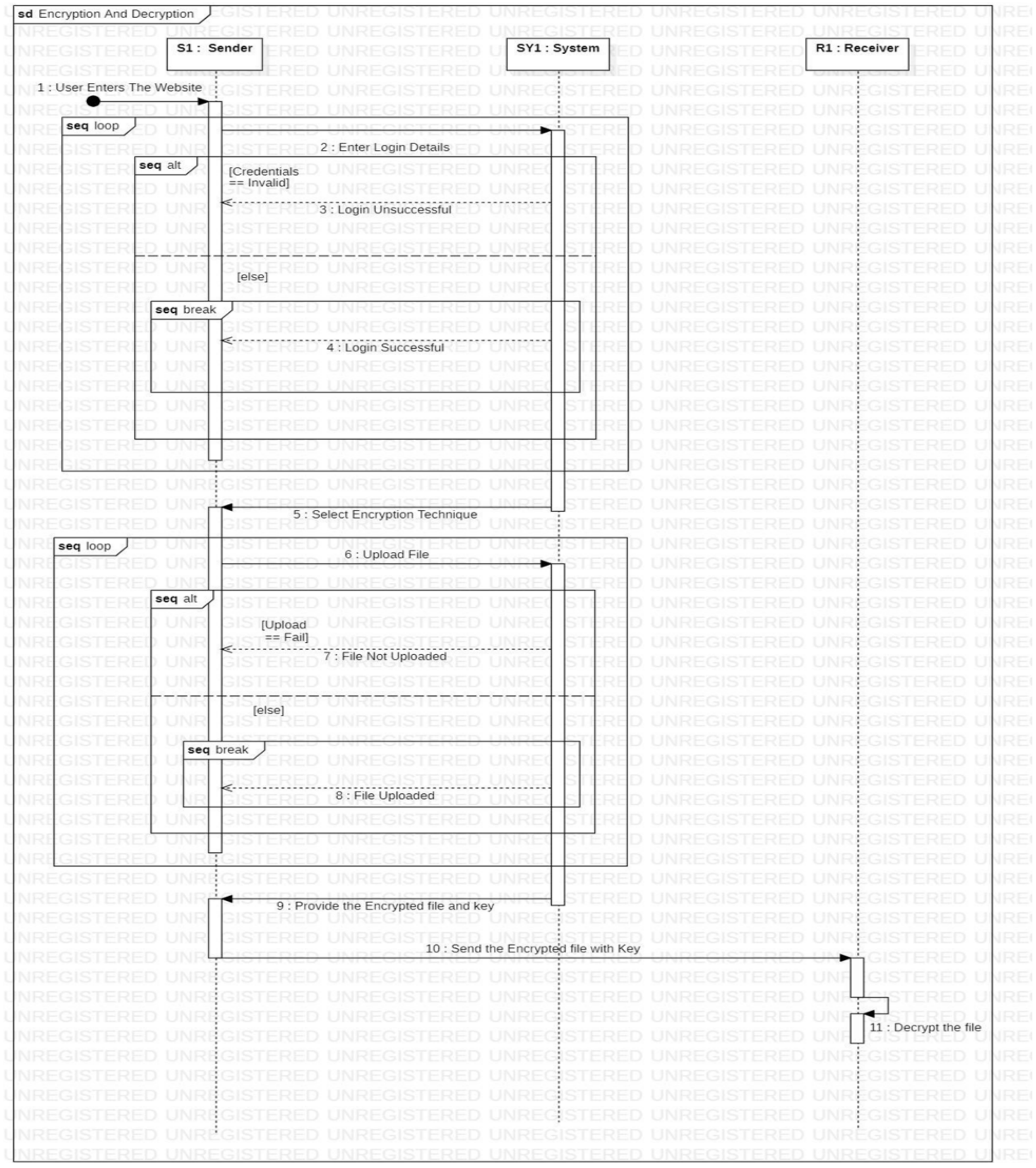


Figure 3.10 - Sequence Diagram

Description: The Sequence Diagram shows the process interactions arranged in a time sequence. In this diagram it shows the different actions to perform the Encryption and Decryption in a time sequence.

3.3.8 Architecture Diagram

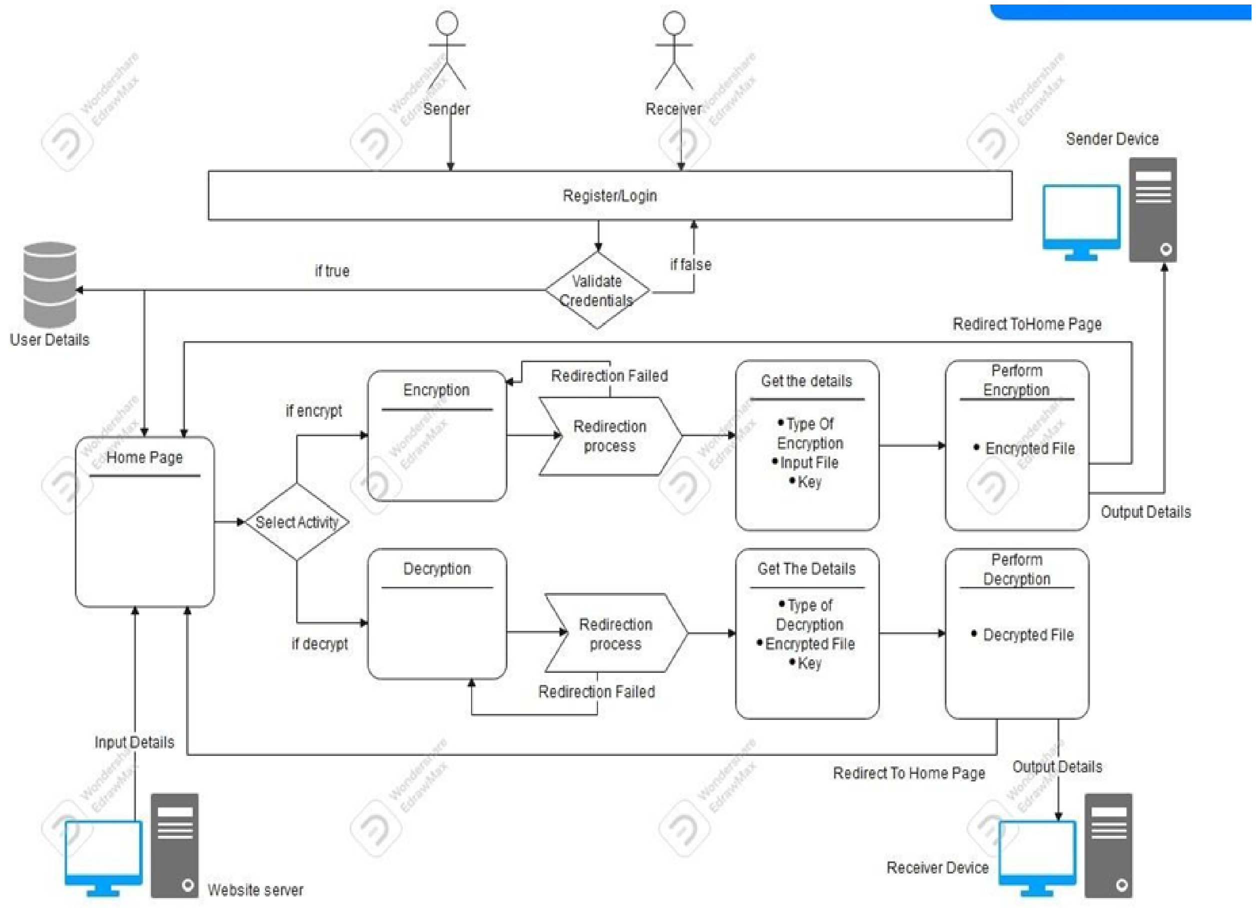


Figure 3.11 - Architecture Diagram

Description: The Architecture Diagram shows the visual representation that maps out the physical implementation for the components in this package. In this diagram it shows the physical implementation of the package with servers and databases. The package is physically represented with operations of Encryption and Decryption it performs.

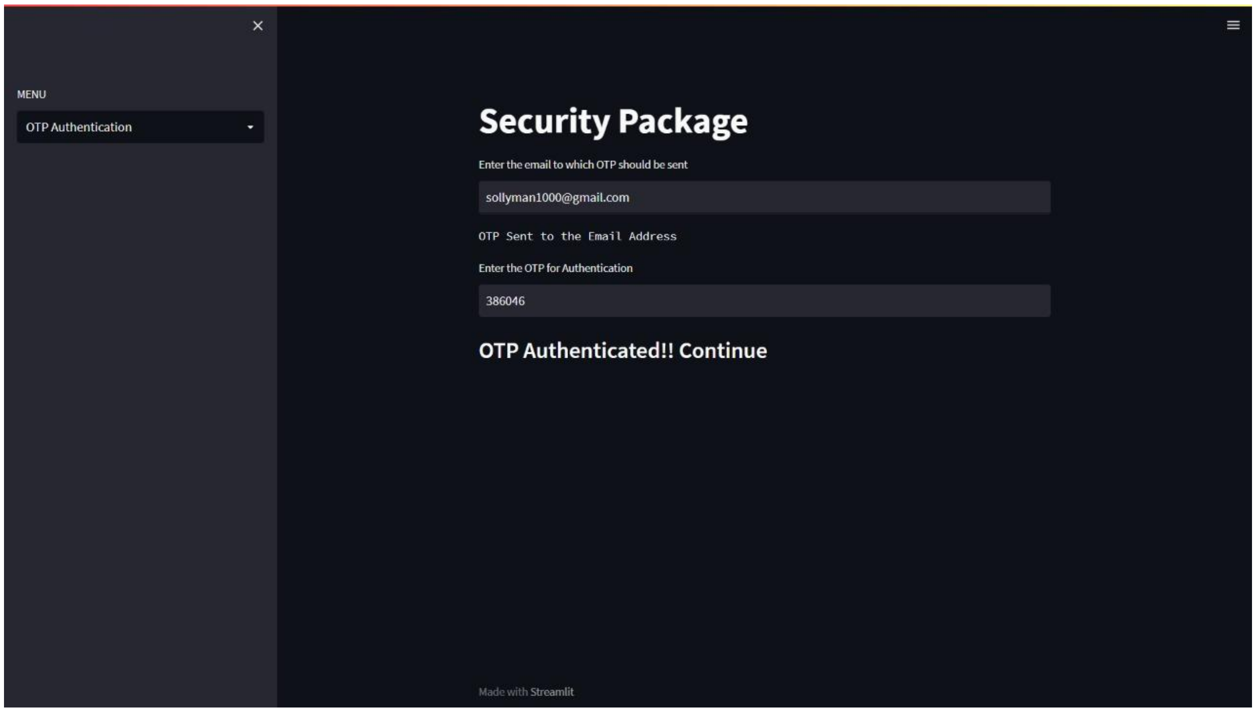
4 IMPLEMENTATION

4.1 Overview of User Interface

The interface for accessing the package is implemented using Streamlit. It provides a user-friendly and easy-to-understand interface. The web application provides the interface for

1. Uploading files and images to be encrypted.
2. Selection of cipher.
3. Steganography.
4. Encoding a file.
5. Decoding a file.
6. Specifying keys for encryption/decryption
7. Background hiding.
8. User Authentication.

4.2 Interface Images



Security Package

Enter the email to which OTP should be sent

sollyman1000@gmail.com

```
OTP Sent to the Email Address
```

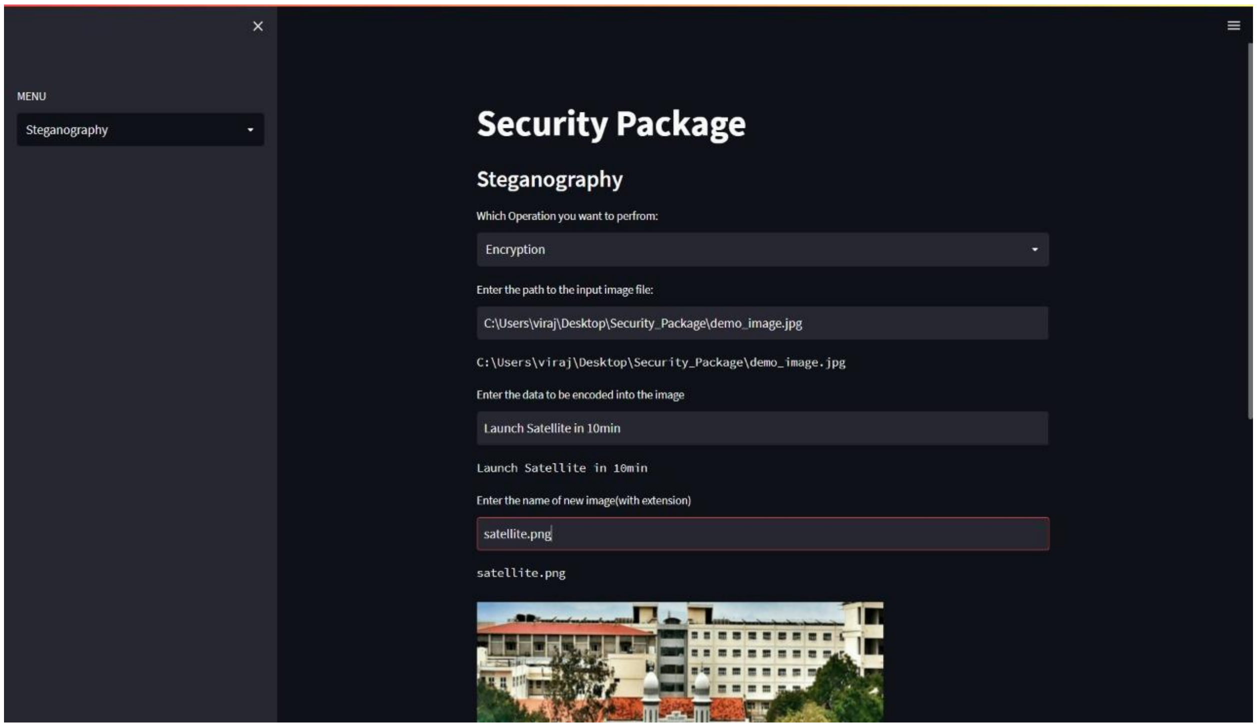
Enter the OTP for Authentication

386046

OTP Authenticated!! Continue

Figure 4.1 - OTP generation and authentication

Description: The Figure 4.1 shows the implementation of OTP generation and verification.The user enters a valid email-id and a 6-digit OTP is sent to the entered mail.If correct OTP is entered then the user can login.



Which Operation you want to perfrom:

Encryption

Enter the path to the input image file:

C:\Users\viraj\Desktop\Security_Package\demo_image.jpg

```
C:\Users\viraj\Desktop\Security_Package\demo_image.jpg
```

Enter the data to be encoded into the image

Launch Satellite in 10min

```
Launch Satellite in 10min
```

Enter the name of new image(with extension)

satellite.png

```
satellite.png
```

Figure 4.2 - Steganography (encryption)

Description: The path of image file and text to be encrypted is entered.

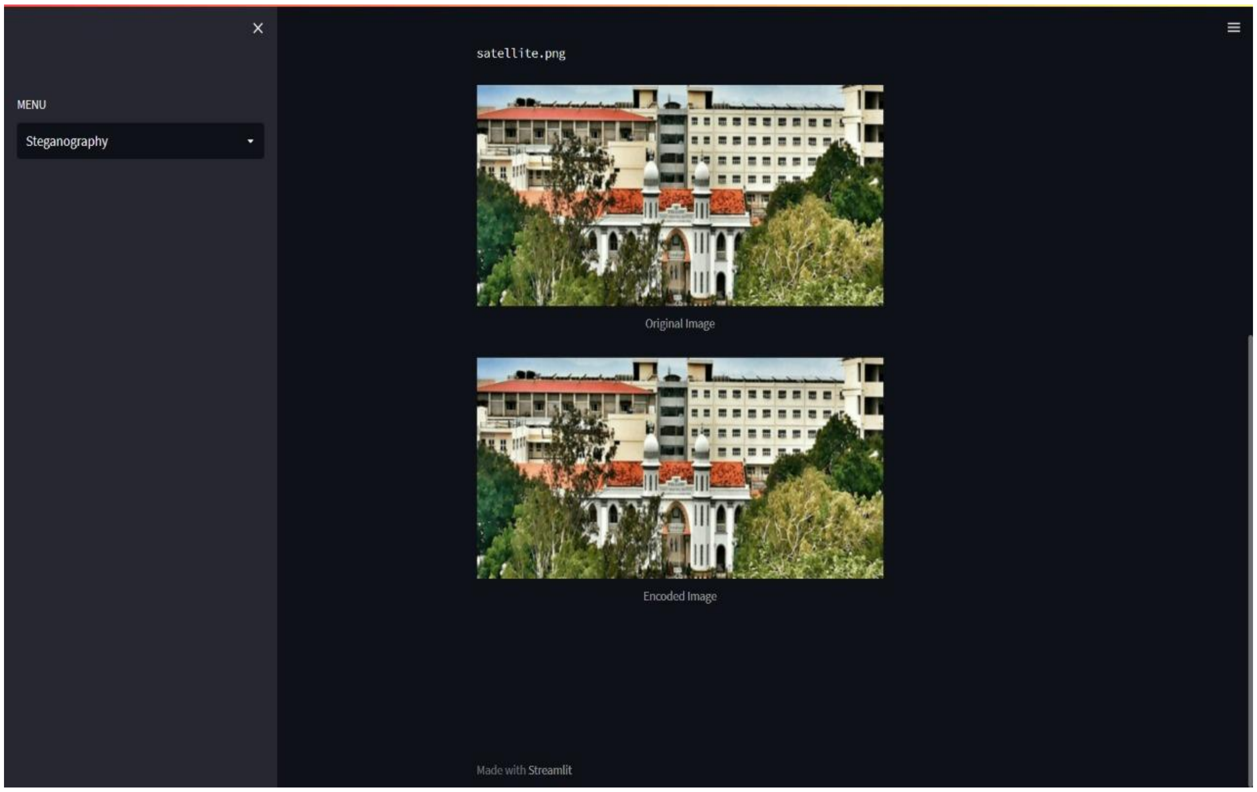
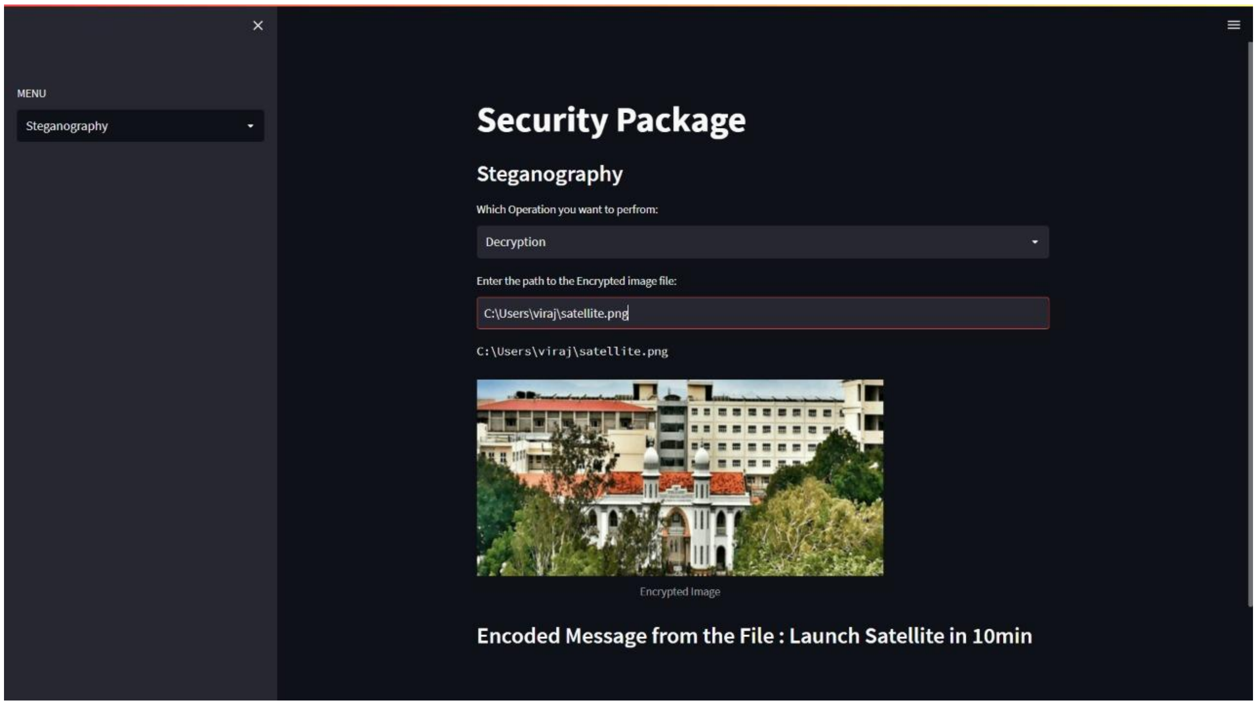


Figure 4.3 - Display Encrypted Image

Description: This image shows the original and encrypted image.



Security Package

Steganography

Which Operation you want to perfrom: Decryption

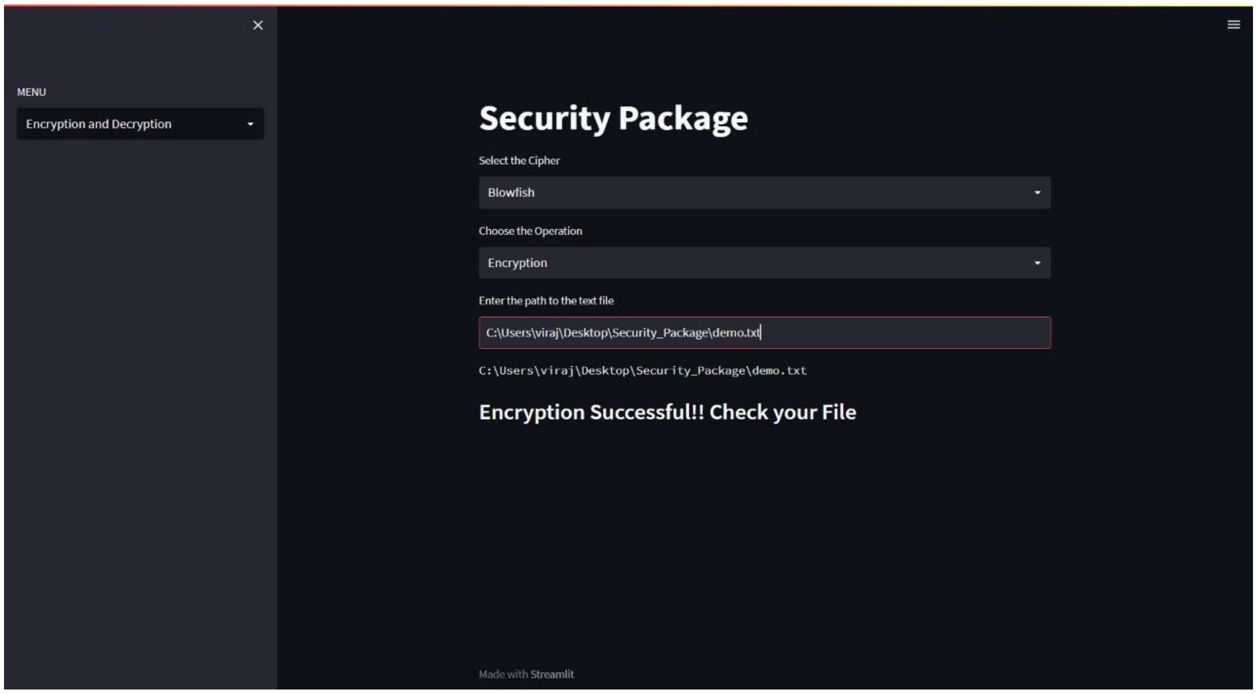
Enter the path to the Encrypted image file: C:\Users\viraj\satellite.png

```
C:\Users\viraj\satellite.png
```

Encoded Message from the File : Launch Satellite in 10min

Figure 4.4 - Steganography (decryption)

Description: The Figure 4.2, 4.3 and 4.4 shows the implementation of steganography. In Figure 4.4, the path to the encrypted image is entered so the text can be retrieved.



Security Package

Select the Cipher: Blowfish

Choose the Operation: Encryption

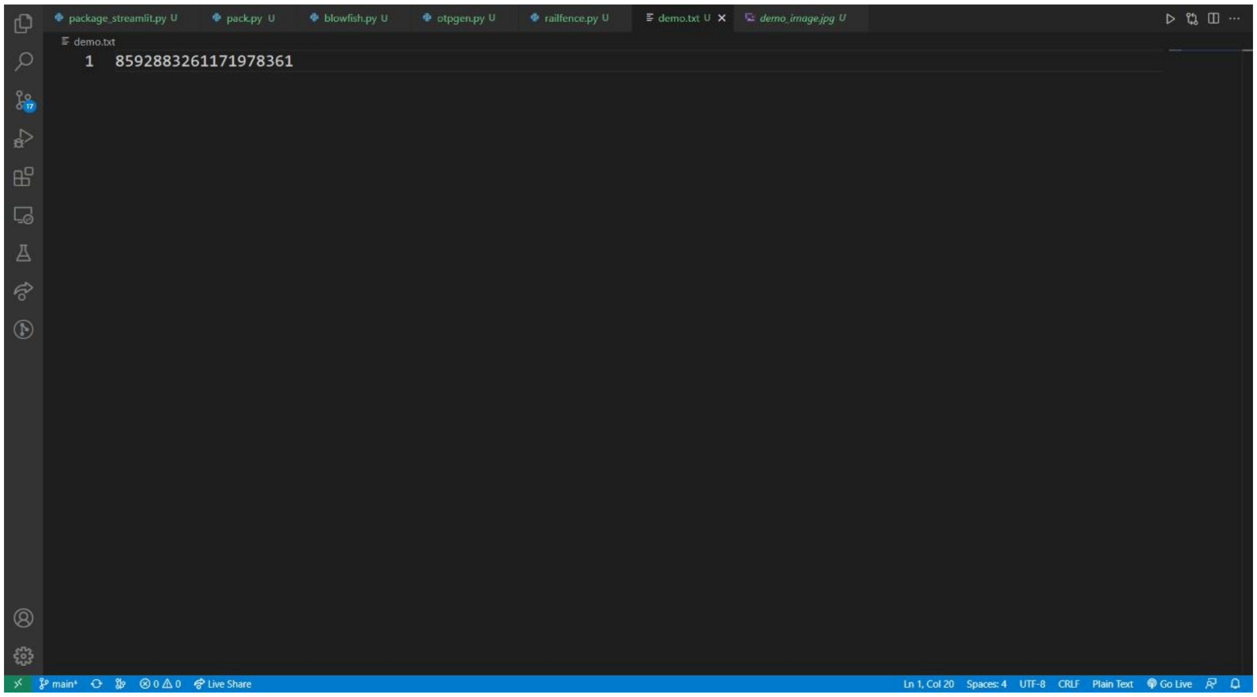
Enter the path to the text file: C:\Users\viraj\Desktop\Security_Package\demo.txt

```
C:\Users\viraj\Desktop\Security_Package\demo.txt
```

Encryption Successful!! Check your File

Figure 4.5 - Blowfish (encryption)

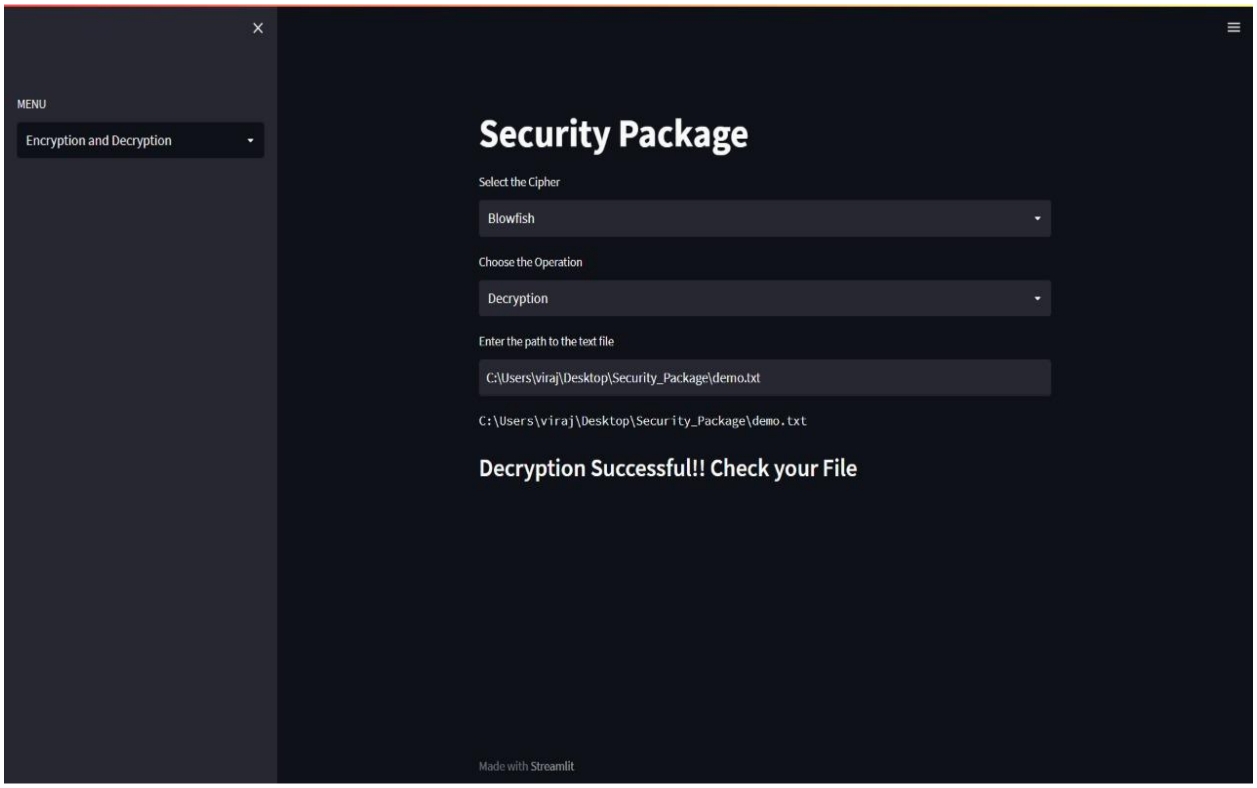
Description: In Figure 4.5, path to a file containing numbers is specified for BlowFish encryption. The file “demo.txt” contains the integer “55”.



```
8592883261171978361
```

Figure 4.6 - Blowfish (encrypted file)

Description: Figure 4.6 is the file “demo.txt” after it has been encrypted.



Select the Cipher: Blowfish

Choose the Operation: Decryption

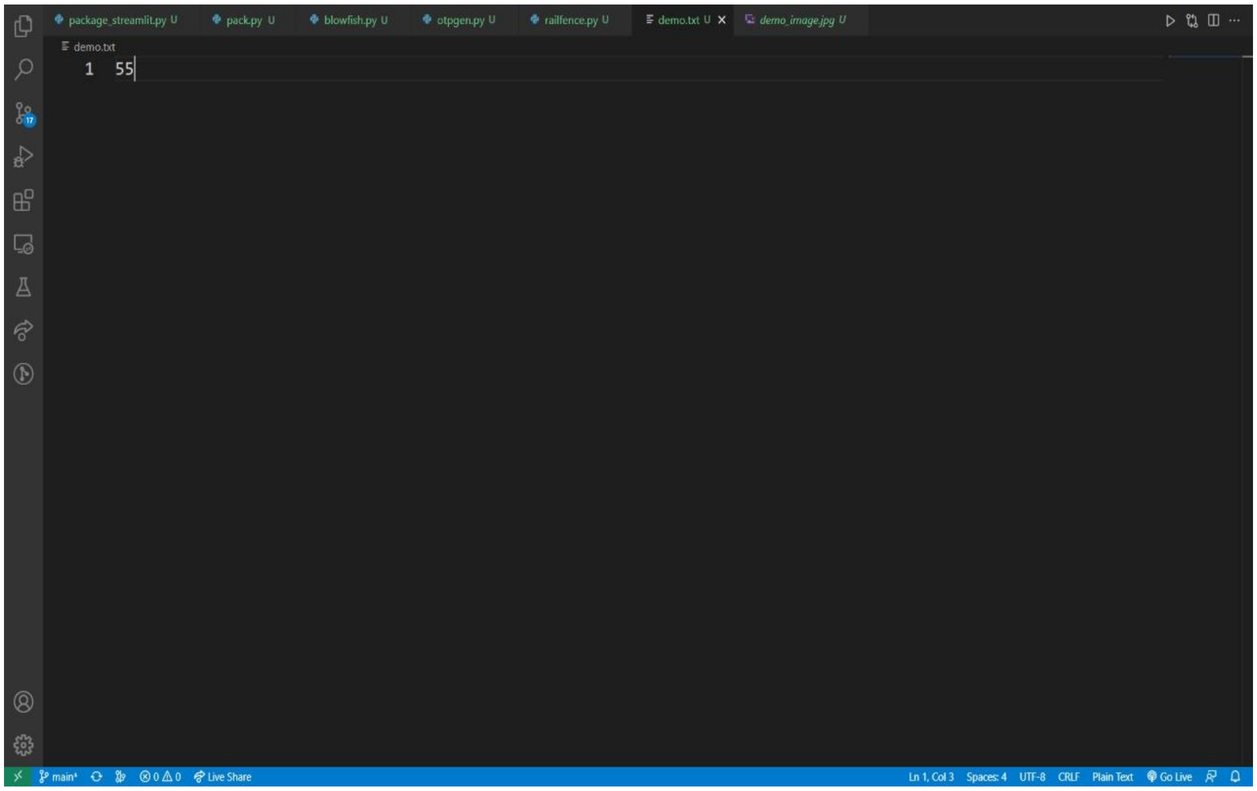
Enter the path to the text file: C:\Users\viraj\Desktop\Security_Package\demo.txt

```
C:\Users\viraj\Desktop\Security_Package\demo.txt
```

Decryption Successful!! Check your File

Figure 4.7 - Blowfish (decryption)

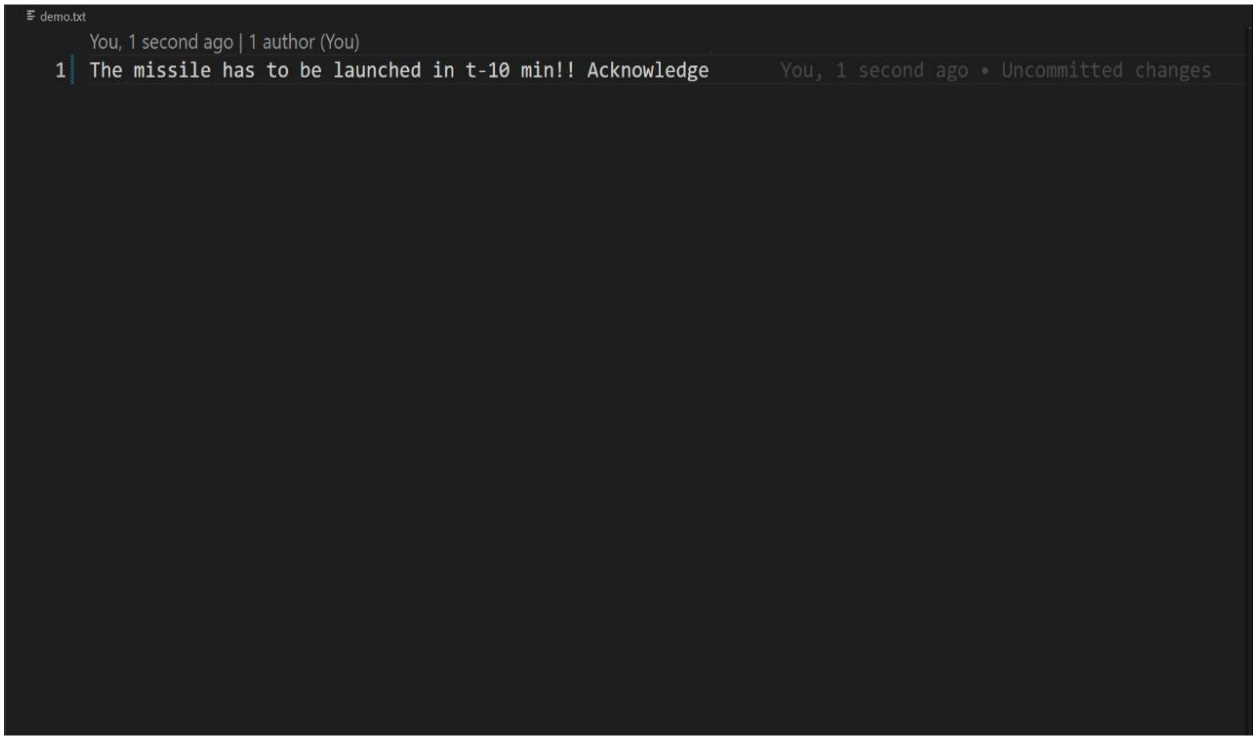
Description: In Figure 4.7, the path to the encrypted file is entered.



```
55
```

Figure 4.8 - Blowfish (decrypted file)

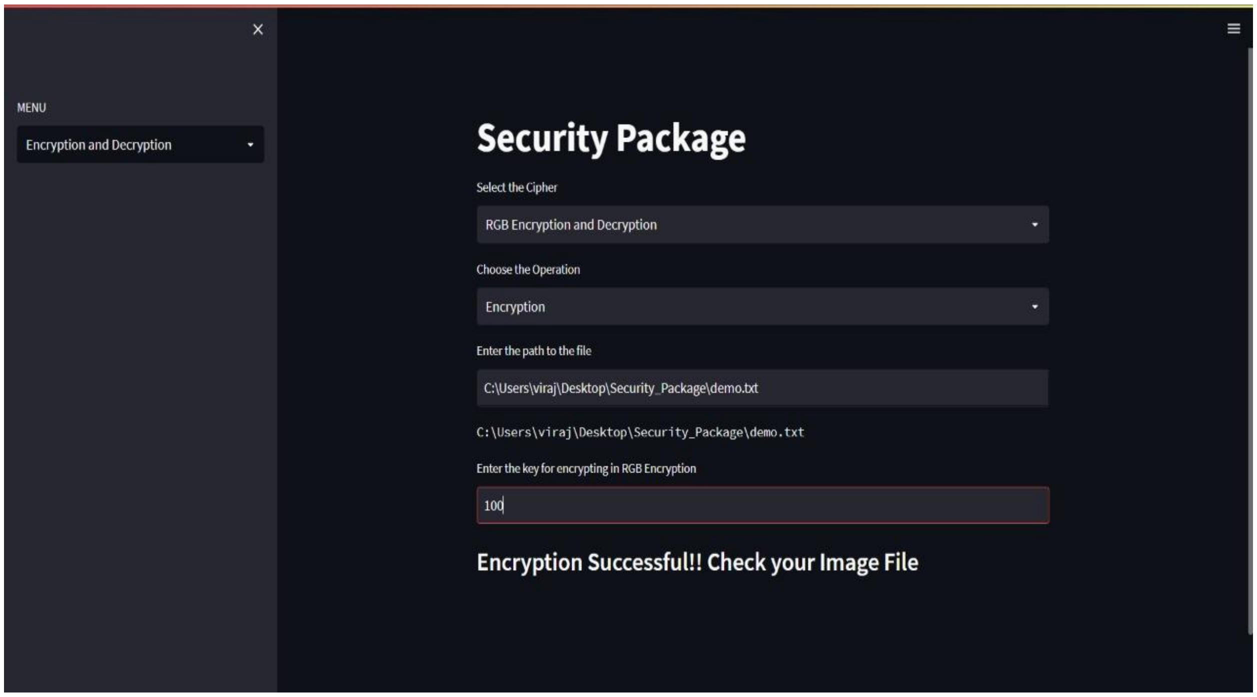
Description: Figure 4.8 shows the decrypted file “demo.txt”.


demo.txt
You, 1 second ago | 1 author (You)
You, 1 second ago • Uncommitted changes

```
The missile has to be launched in t-10 min!! Acknowledge
```

Figure 4.9 - RGB Technique (original file)

Description: Figure 4.9 shows the file to be encrypted.



Security Package

Select the Cipher: RGB Encryption and Decryption

Choose the Operation: Encryption

Enter the path to the file: C:\Users\viraj\Desktop\Security_Package\demo.txt

```
C:\Users\viraj\Desktop\Security_Package\demo.txt
```

Enter the key for encrypting in RGB Encryption: 100

Encryption Successful!! Check your Image File

Figure 4.10 - RGB Technique (encryption)

Description: In Figure 4.10, the path to the original text file i.e.“demo.txt” and key is entered for encryption.

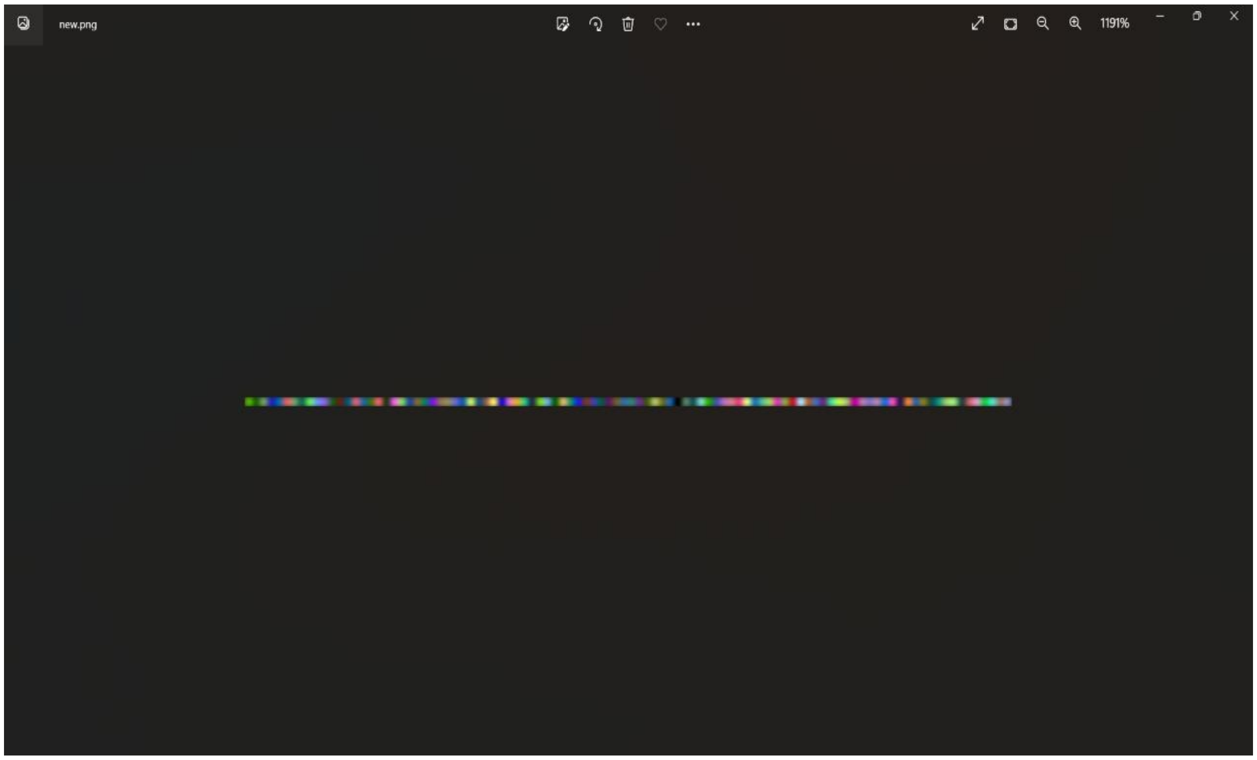
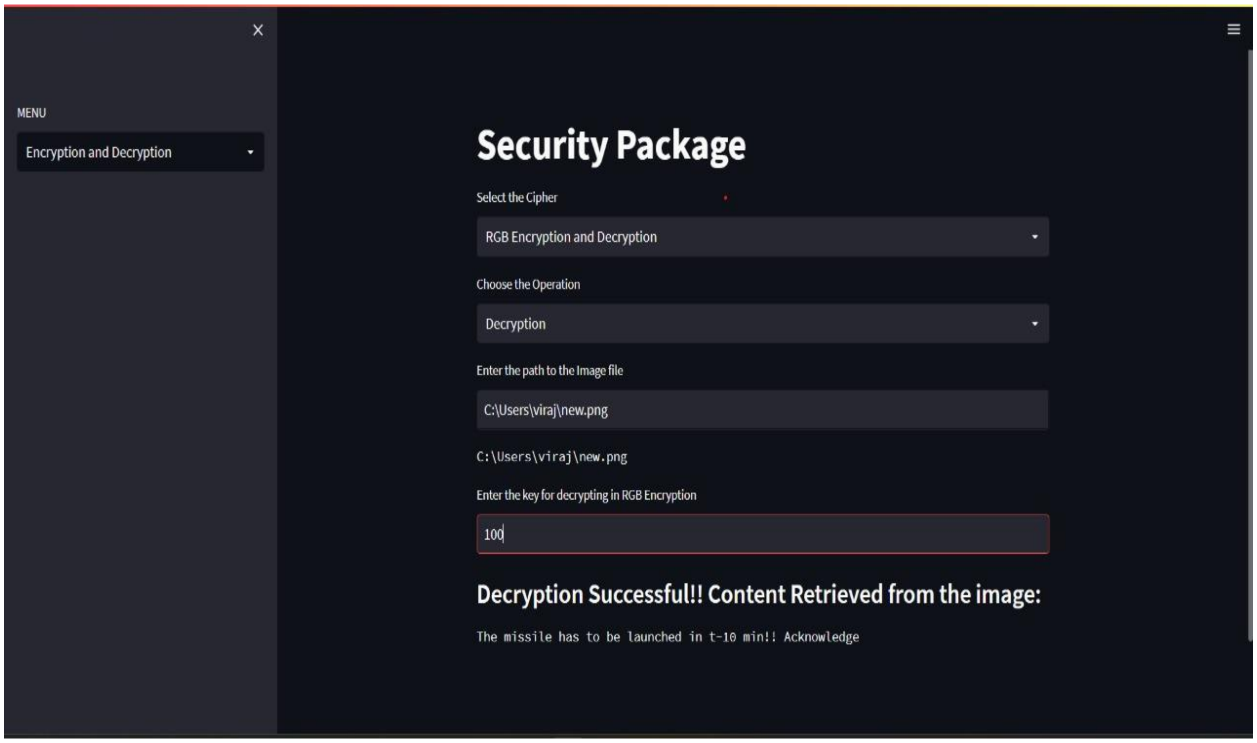


Figure 4.11 - RGB Technique (encrypted image file)

Description: Figure 4.11 shows the encrypted file as a strip of pixels (.png).



Security Package

Select the Cipher: RGB Encryption and Decryption

Choose the Operation: Decryption

Enter the path to the Image file: C:\Users\viraj\new.png

```
C:\Users\viraj\new.png
```

Enter the key for decrypting in RGB Encryption: 100

Decryption Successful!! Content Retrieved from the image:

```
The missile has to be launched in t-10 min!! Acknowledge
```

Figure 4.12 - RGB Technique (decryption)

Description: In Figure 4.12, the path of the encrypted image file and key is entered for decryption. The system displays the text encrypted within the strip of pixels.

5 EXTERNAL INTERFACE REQUIREMENTS

5.1 User Interfaces

Must provide a user interface that is accessible through any internet browser, the major ones being Google Chrome and Internet Explorer 12. The software provides a better graphical interface for the user, any admin can operate on the system performing required tasks such as viewing and updating the information.

The system shall support user interface and layouts for

- Common PC
- PDA
- Mobile Phone

Allows users to submit the updated information and get approval from their supervisor.

- Front-end: Streamlit(Python)
- Back-end: Python and its Libraries

All pages of the system are following a consistent theme and structure. The occurrence of the error is minimized through the use of Drop Box, Scroll down thereby reducing the amount of text input from users. The error message should be located beside the error input which clearly highlights and tells the user how to solve it.

5.2 Hardware Interfaces

All Components used are capable of being executed on Personal computers with Windows OS Platforms and other Platforms like Linux, and Unix.

Operating system: Windows	**Hard disk**: 40 GB or more
RAM: 4 GB	**Graphics Card**: Intel or Nvidia

Processor: Quad-Core CPU Internet connection with good speed.

5.3 Software Interfaces

Operating system: We have chosen Windows OS for its best support and user-friendliness.

Python: It is easy to implement cryptographic functions and it reduces the overhead of the developers.

Streamlit: It is an open-source app framework in Python language. Streamlit lets you turn data scripts into shareable web apps. It is Python, open-source, and free.

6 TESTING

Testing can be classified into various types like Unit Testing, Integration Testing, Incremental- Integration Testing, etc. They are explained as follows.

6.1 Unit Testing

During the developmental phase, each functionality developed was tested against all the boundary cases. On all the pages, the data entered by the users are properly validated for its type and the type of value it can have.

6.2 Integration Testing

All the developed modules are integrated to realize a functionality and it is tested for its appropriateness. This testing helped to make the modules to be more cohesive with the supporting modules to implement the required functionality.

6.3 Incremental-Integration Testing

Incremental-Integration testing was done repeatedly whenever a new feature is added. Every time a new functionality was added, all the other functionality of that module was tested to ensure the new addition had not affected the consistency of the previous working module because of the dependency. Whenever it got affected all the bugs were fixed and then the next task was implemented.

6.4 Functional Testing

This test was done several times to check the functionality irrespective of the architecture of the software inside. This testing helped to implement the functionality faster. Once the required task was implemented, then we organized the code such that the software gets to better architecture.

6.5 Acceptance Testing

Acceptance testing was done to verify whether the software met all the requirements.

6.6 Usability Testing

The system usability was checked by allowing the naive user to use the system at different levels of development and thereby get feedback from them to improve the usability. Compiling all such feedback the system was implemented with more usability and with user-helper documentation.

6.7 Test Cases

The Test Case is about the Encryption of Caesar Cipher.

Test Case	Test case Description	Step	Test Data	Steps to Perform	Expected Result	Actual Result
1	Caesar Cipher - Encryption	1	“Caesar Cipher”	Select the Cipher	Cipher selected	User selects a cipher of his choice
		2	“Encryption”	Choose operation	Operation selected	User choose the action of operation
		3	path-of-file	Enter the path of tcxt file	Path of the text file is uploaded	The text to be encrypted is mentioned
		4	Key	Enter a key for encryption	Key for encryption is uploaded	The system obtains the key for encrypting with.
		5	Enter prompt	Click the Enter Button	Encryption of the text file is done	An Encrypted file is saved in the disk

Table 6.1 - Caesar Cipher Encryption Test case

The Test Case is about the Decryption of Caesar Cipher.

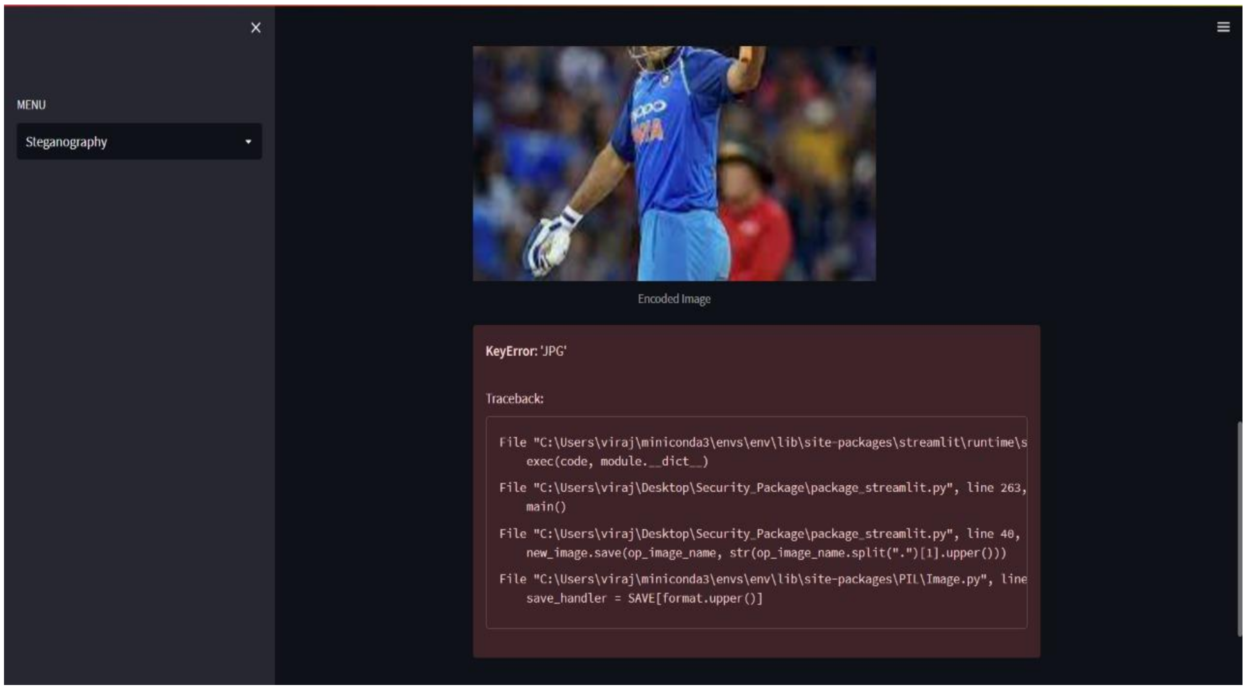
Test Case	Test case Description	Step	Test Data	Steps to Perform	Expected Result	Actual Result
2	Caesar Cipher - Decryption	1	“Caesar Cipher”	Select the Cipher	Cipher selected	User selects a cipher of his choice
		2	“Decryption”	Choose operation	Operation selected	User choose the action of operation
		3	path-of-file	Enter the path of tcxt file	Path of the text file is uploaded	The text to be encrypted is mentioned
		4	Key	Enter a key for decryption	Key for decryption is uploaded	The system obtains the key for decrypting with.
		5	Enter prompt	Click the Enter Button	decryption of the text file is done	A decrypted file is saved in the disk

Table 6.2 - Caesar Cipher Decryption Test case

The Test Case is about Encryption with Steganography.

Test Case	Test case Description	Step	Test Data	Steps to Perform	Expected Result	Actual Result
3	Steganography	1	"Stegano-graphy"	Menu option	Steganography selected	User selects steganography
		2	"Encryption"	Choose operation	Operation selected	User choose the action of operation
		3	path-of-file	Enter the path of .png file	Path of the image file is uploaded	The image to be encrypted is mentioned and displayed
		4	text	Enter a text for encryption	Text to be encryption is uploaded	The system obtains the text for encrypting.
		5	"new.png"	Enter the name of the new image file (encrypted)	Encrypted Image name is uploaded	The encrypted image name will be "new.png"
		6	Enter prompt	Click the Enter key	The text gets encrypted within the image	The encrypted image is saved to the local disk and displayed

Table 6.3 - Steganography Encryption Test case



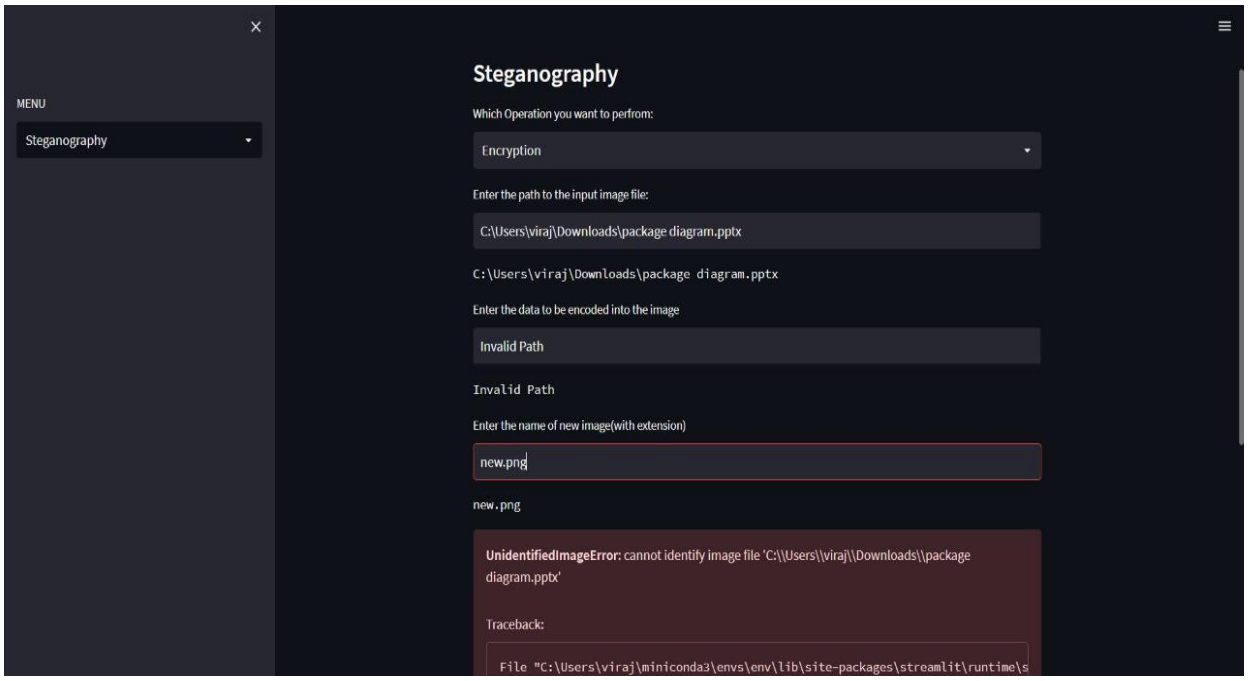
KeyError: 'JPG'

Traceback:

```
File "C:\Users\viraj\miniconda3\envs\env\lib\site-packages\streamlit\runtime\s
    exec(code, module.__dict__)
File "C:\Users\viraj\Desktop\Security_Package\package_streamlit.py", line 263,
    main()
File "C:\Users\viraj\Desktop\Security_Package\package_streamlit.py", line 40,
    new_image.save(op_image_name, str(op_image_name.split(".")[1].upper()))
File "C:\Users\viraj\miniconda3\envs\env\lib\site-packages\PIL\Image.py", line
    save_handler = SAVE[format.upper()]
```

Figure 6.1 - Steganography (key error)

Description: The second argument of save is not the extension, it is the format argument as specified in image file formats and the format specifier for JPEG files is JPEG, not JPG.



Steganography

Which Operation you want to perfrom: Encryption

Enter the path to the input image file: C:\Users\viraj\Downloads\package diagram.pptx

```
C:\Users\viraj\Downloads\package diagram.pptx
```

Enter the data to be encoded into the image: Invalid Path

```
Invalid Path
```

Enter the name of new image(with extension): new.png

```
new.png
```

UnidentifiedImageError: cannot identify image file 'C:\\Users\\viraj\\Downloads\\package diagram.pptx'

Traceback:

```
File "C:\Users\viraj\miniconda3\envs\env\lib\site-packages\streamlit\runtime\s
```

Figure 6.5 - Steganography (file format error)

Description: The module does not accept non-image files for encrypting text into them.Only .png files are used for Steganography

7 BIBLIOGRAPHY

- Michael Jackson, “Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices’’, ACM Press, January 1995.
- Karl E. Wieger, “Software Requirements”, 2nd Edition, Microsoft Press, February 2003.
- Roger S. Pressman, “Software Engineering: A Practitioner’s Approach”, 6th Edition, McGraw Hill International, June 2004.
- Hans, Knebl, Helmut, Delfs , "Introduction To Cryptography Principles And Applications", 3rd Edition, Springer- Verlag, Berlin Heidelberg, October 2015.
- William Stallings, "Cryptography and Network Security: Principles and Practice", 7th Edition, Prentice Hall of India, Pearson Education, New Delhi, June 2017.