FUSION

An Android Based Secure Chat App

By:

SOBIA MANZOOR

Supervised By:

Ms. IFRAH FARUKH

Department Of Computer Science

Quaid-e-Azam University

Islamabad

2014-2019

ACKNOWLEDGEMENT

All the praises, thanks and acknowledgements are for the creator ALLAH Almighty, the most Beneficent, the most Merciful, who gave me strength and enabled me to undertake and execute this task. Countless salutations upon the Holy Prophet Hazrat Muhammad (S.A.W), source of knowledge for enlightening with the essence of faith in ALLAH and guiding the mankind, the true path of life.

First and foremost, I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my Project Supervisor, Ms. Ifrah Farukh Khan for her invaluable advice, guidance and her enormous patience throughout the development of the project. I am sincerely grateful to her for sharing her truthful and illuminating views on a number of issues related to the project.

In addition, I would also like to express my gratitude to my family and friends who had helped and given me encouragement in my study.

ABSTRACT

Fusion is an Android application which allows the user to encrypt and decrypt data. Data can be images, text messages and files. For encryption and decryption, both Asymmetric and Symmetric Cryptography algorithms are used (hybrid approach). The project also implements Steganography algorithm and allows the user to send encrypted data to other users of this application. Increased security of confidential data during transmission and decreased chances of data leakage are achieved by using the combination of Hybrid Cryptography and Steganography.

Contents

List of Tables

List of Figures

CHAPTER 1

COMPARATIVE ANALYSIS OF THE TECHNIQUES

PROJECT OVERVIEW

Fusion is an android application. The project will allow the user to encrypt and decrypt images, text messages and files by using Hybrid Cryptography (Asymmetric Cryptography algorithm "Elliptic Curve Cryptography (ECC)" and Symmetric Cryptography algorithm "Advance Encryption Standard (AES)") and will also allow the user to conceal the encrypted images, text messages and files inside some other cover image by using Steganography algorithm "Least Significant Bit (LSB)". The combination of hybrid cryptography and steganography will increase the security of confidential data (images, text messages and files) during the transmission and will decrease the chances of data leakage. And user can also send the encrypted data (image/text messages/ files) through this application to other users of the same application.

This Chapter gives the comparative analysis of different algorithms of the techniques that are used in Fusion.

1. CRYPTOGRAPHY

1.1 Introduction

"Cryptography is a branch of cryptology dealing with the design of algorithms for encryption and decryption, intended to ensure the secrecy and/or authentication of messages"-William Stallings.

1.2 General Cipher Model

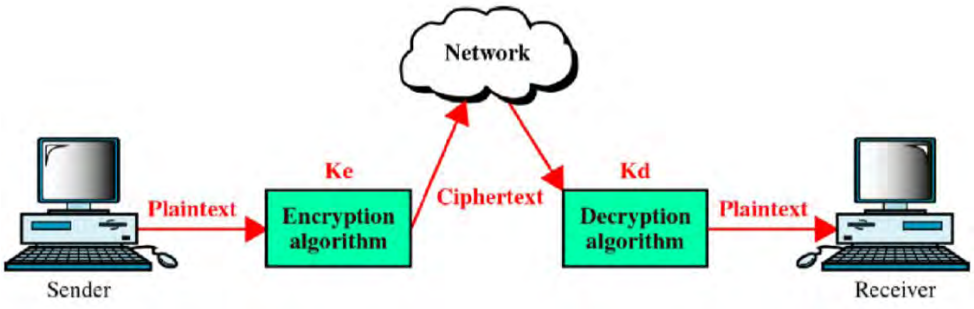

Figure 1 General Cypher Model

1.2 Technical Terms

Plaintext:

Plaintext is the original message. It is an input to encryption algorithm during the encryption process to generate ciphertext. Plaintext is in readable format.

Ciphertext:

Ciphertext is the encrypted/coded message. It is an output of encryption algorithm. It is also an input to the decryption algorithm during decryption process and results in plaintext. Ciphertext is in unreadable format that is scrambled format.

Cipher:

Cipher is an algorithm that converts plaintext to ciphertext and vice versa.

Key:

The information used in cryptographic process to encrypt and decrypt messages. Cryptography keys are known only to sender and receiver. It is the key on which the strength of cryptographic process depends on.

Encipher (Encrypt):

Encipher is a process that converts plaintext to ciphertext. Inputs to encipher are plaintext and cryptographic key, and the output of encipher is the generated ciphertext.

Decipher (Decrypt):

Decipher is a process that converts ciphertext to plaintext. The inputs to decipher are ciphertext and cryptographic key, and the output of decipher is generated plaintext. Decipher is a reverse process of encipher.

Cryptanalysis (Code Breaking):

The study of methods to decipher a ciphertext without knowing cryptographic key is called cryptanalysis.

Cryptology:

The combination of cryptography and cryptanalysis fields is known as cryptology.

1.3 Types of Cryptography

Cryptography has two types:

- Symmetric Key Cryptography or Conventional Encryption
- Asymmetric or Public Key Cryptography

Asymmetric Cryptography

Asymmetric Cryptography uses a key pair, one of them is public key and the other is private key. Both are used to encrypt and decrypt messages. Public key is named as public because it is known to everyone while the private key is a secret key that is why it is named as private. As asymmetric cryptography uses a key pair, it is also known as Public Key or Two-Key Cryptography. In the process of exchanging messages, the keys are owned by each participant (sender and receiver). It is named as Asymmetric because separate keys are used for encryption and decryption processes such that those who encrypt a message cannot decrypt it using the same key.

The message is encrypted with public key of sender and is decrypted with the private key of receiver so that encryption is achieved.

Major advantage:

The major advantage of Asymmetric Cryptography is Confidentiality. It uses a key pair so single key is not shared between sender and receiver. Confidential communication is possible using the public key as private key is kept secret.

Why need Asymmetric Cryptography

There were two major issues that lead to the need of Asymmetric Cryptography.

- Secure distribution of key
- Verification that message comes from claimed sender.

Requirements of Asymmetric Cryptography

Asymmetric cryptography requires three things to give confidentiality:

- Encryption Algorithm
- A private key known only to receiver for the decryption of encrypted messages.
- A public key known to everyone used to encrypt messages.

Asymmetric Cryptography Model

The general model of asymmetric cryptography is shown below:

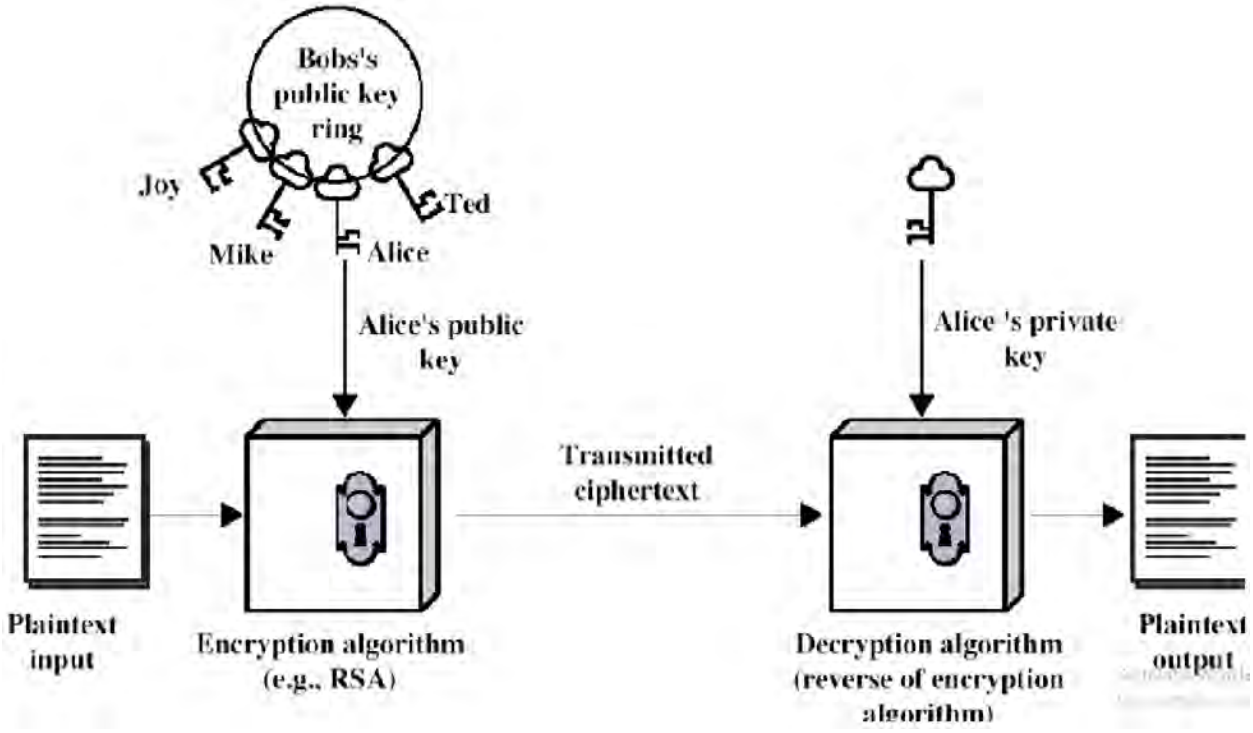

Figure 2 Asymmetric Cryptography Model

Let us suppose that sender A sends a message M to a receiver B. The public key of sender A is "e" and the private key of receiver B is "d". The equations for encryption and decryption processes are:

$C = E_e (M)$ (Encryption)

$M = D_d (C)$ (Decryption)

Applications of Asymmetric Cryptography

The applications of Asymmetric cryptography are:

- Confidentiality
- Digital Signatures
- Key Exchange

Strength Factor of Asymmetric Cryptography

The strength of asymmetric cryptography relies on two factors:

- Finding private key is computationally infeasible even if someone knows the algorithm and public key.
- Factorization of large integers.

Example Algorithms of Asymmetric Cryptography

Example algorithms of Asymmetric cryptography are RSA (Rivest, Shamir, Adleman), ECC (Elliptic Curve Cryptography), Diffie-Hellman, SSL, SSH and PGP (Pretty Good Privacy).

Criteria for Algorithms Comparison

The criteria used for comparing Asymmetric cryptography algorithms are as follows:

a) **Make Use of:**
   Defines the mathematical technique, the algorithm is using.

b) Strength Factor:

Strength factors describe the factor on which the strength of algorithm depends.

c) Security:

The resistance of an algorithm against attacks defines security. Usually key size measures the security.

- **Key Size:**

Large key sizes slow down the cryptography process but it also increases security by making the exhaustive searching difficult.

d) Limitations (known Attacks):

Limitations of an algorithm are defined in terms of two factors:

1. How often an algorithm is vulnerable to different types of attacks?
2. How fine the algorithm works when all the resources are given?

Comparative Analysis of Algorithms

Table 1 Comparative Analyses of Asymmetric Algorithms

S. No	Parameters	RSA	ECC	Diffie-Hellman
1.	Make Use of	Large integers (prime numbers) e. g 1024 bits	Elliptic curve theory	Large integers
2.	Strength Factor	Cost of factorizing large prime numbers	Properties of elliptic curves, Computing discrete logarithms	Computing discrete logarithms
3.	Security	Strong	Stronger than others	Strong
4.	Key Generation	Yes, a Complex process	Yes	Yes
5.	Key Generation Time	More than others	Least	Less than others
6.	Key Size	1024-4096 bits	112-512 bits	1024-4096 bits
7.	Invulnerable to	Brute Force Attack is Infeasible- given size of numbers; also Mathematical attacks are infeasible because of factorizing large prime numbers.	N/A	Mathematical attacks are infeasible because of factorizing the large integers.
8.	Memory Usage	Requires large memory because of large key sizes	Requires Less memory because of small key sizes	Requires large memory because of large key sizes
9.	Processing keys and messages	Slow due to computing large integers	Faster than others because of small key sizes	Slow due to computing large integers
10.	Power Usage	More than others	Less than RSA	N/A
11.	Computational	Exponential	Less than others	More than ECC

	Time	because of large key sizes		
12.	Computational Complexity	Higher	Lower	Lower
13.	Vulnerable to	Man in the middle attack, chosen cipher attack because of multiplicative property "product of 2 ciphers is equal to encryption of product of respective plaintexts" [13]	N/A	DOS, Man in the middle attack

Comparative Analysis of Algorithms Based on Functionality

Some algorithms provide all of functionalities while some provide specific functionality.

Table 2 Comparative Analyses of Asymmetric Algorithms Based on Functionality

S. No	Algorithm	Confidentiality	Digital Signature	Key Exchange
1.	RSA	Yes	Yes	Yes
2.	ECC	Yes	Yes	Yes
3.	Diffie-Hellman	No	No	Yes
4.	DSS	No	Yes	No

Comparative Analysis of Algorithms Based on Key Sizes

ECC provides the same amount of security as RSA with smaller key sizes.

Table taken from Book [2]

Table 3 Comparative Analyses of Asymmetric Algorithms Based on Key Sizes

S.No	RSA key size	ECC key size	Diffie-Hellman
1.	512 bits	112 bits	512 bits
2.	1024 bits	160 bits	1024 bits
3.	2048 bits	224 bits	2048 bits
4.	3072 bits	256 bits	3072 bits
5.	7680 bits	384 bits	7680 bits
6.	15360 bits	512 bits	15360 bits

Performance Evaluation

Performance evaluation of algorithms is also known as Scalability evaluation. In the analysis of algorithms scalability is the major factor.

The performance evaluation of the RSA and ECC was conducted with different sizes of images.

The evaluation is conducted on a laptop which has the following properties:

Windows Edition: Windows 10 Home

System:

- Processor: Intel (R) Core (TM) i5-7200
- CPU: 2.50 GHz – 2.71 GHz
- RAM: 8.00 GB
- System Type: 64-bit Operating System , x-64 based Processor
- Pen n Touch: No Pen and Touch is available

Emulator:

The emulator used is Pixel 3 XL. The specifications of emulator are:

- Device Name: Android SDK built for x86
- API Level: 29
- Android Version 10
- Internal Storage: 4 GB
- SD Card: 535 MB

Evaluation Parameters:

Resource Usage:

The resource usage is done using Android Profiler. Android profiler is a tool for the identification of performance holes and to check that the application is using appropriate amount of memory, power and networking data bandwidth. The android profiler monitors the CPU, memory usage and networking while the application is running on a physical device or on an android emulator.

- **CPU Profiler:** Defines the CPU usage in real time for the running application.
  Memory Profiler: Defines how much memory an algorithm uses for encryption/decryption or for other functions performed by the algorithm. The efficiency of an algorithm is great if the memory usage is small that is they are inverse proportional to each other.
- **Network Profiler:** Network profiler shows the speed of data transfer in Mb/sec.
- **Energy Profiler:** Defines the energy used by the running app in terms of CPU, networking and location tracking activity.

The graphs of above profilers are shown below in which the image of size 5.73 MB is encrypted using both hybrid cryptography and steganography.

ECC:

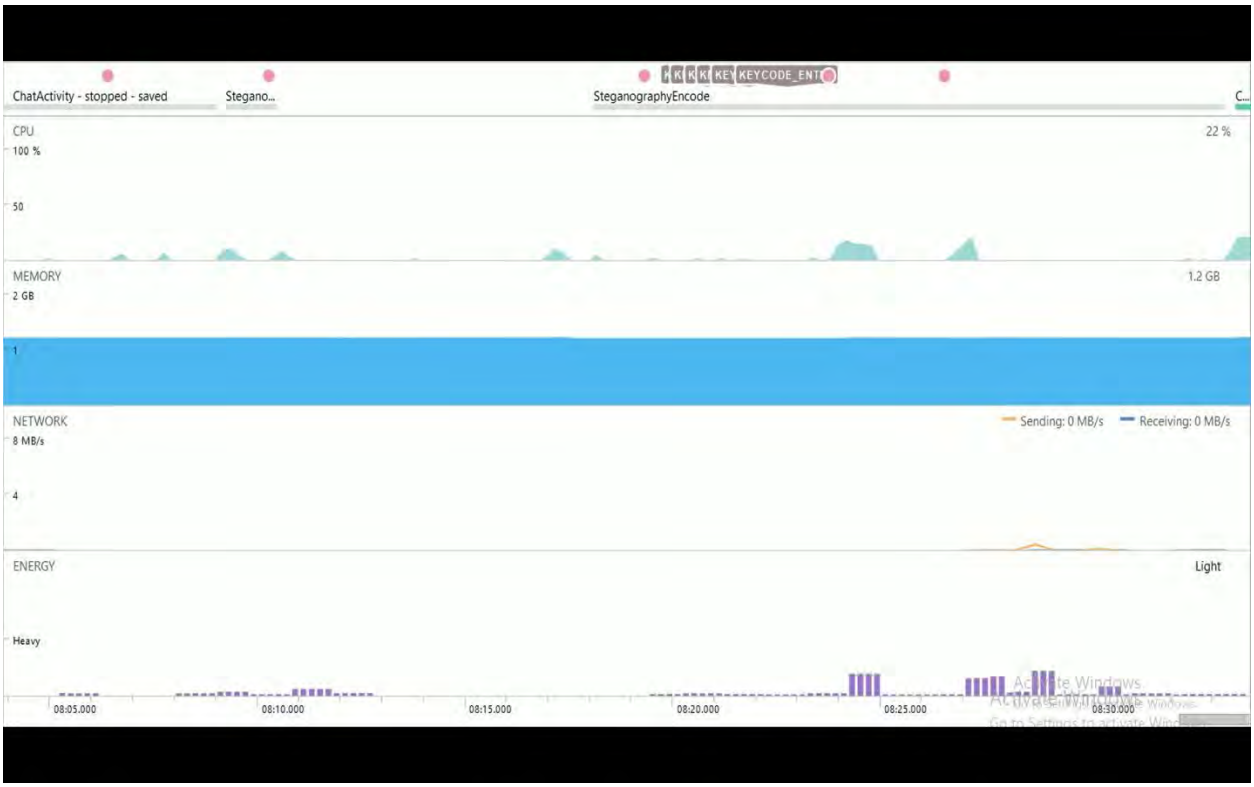

Figure 3 Graph of ECC

RSA:

Figure 4 Graph of RSA

And when the image size is 3.66 MB then the graphs are:

ECC:

Figure 5 Graph of ECC 2

RSA:

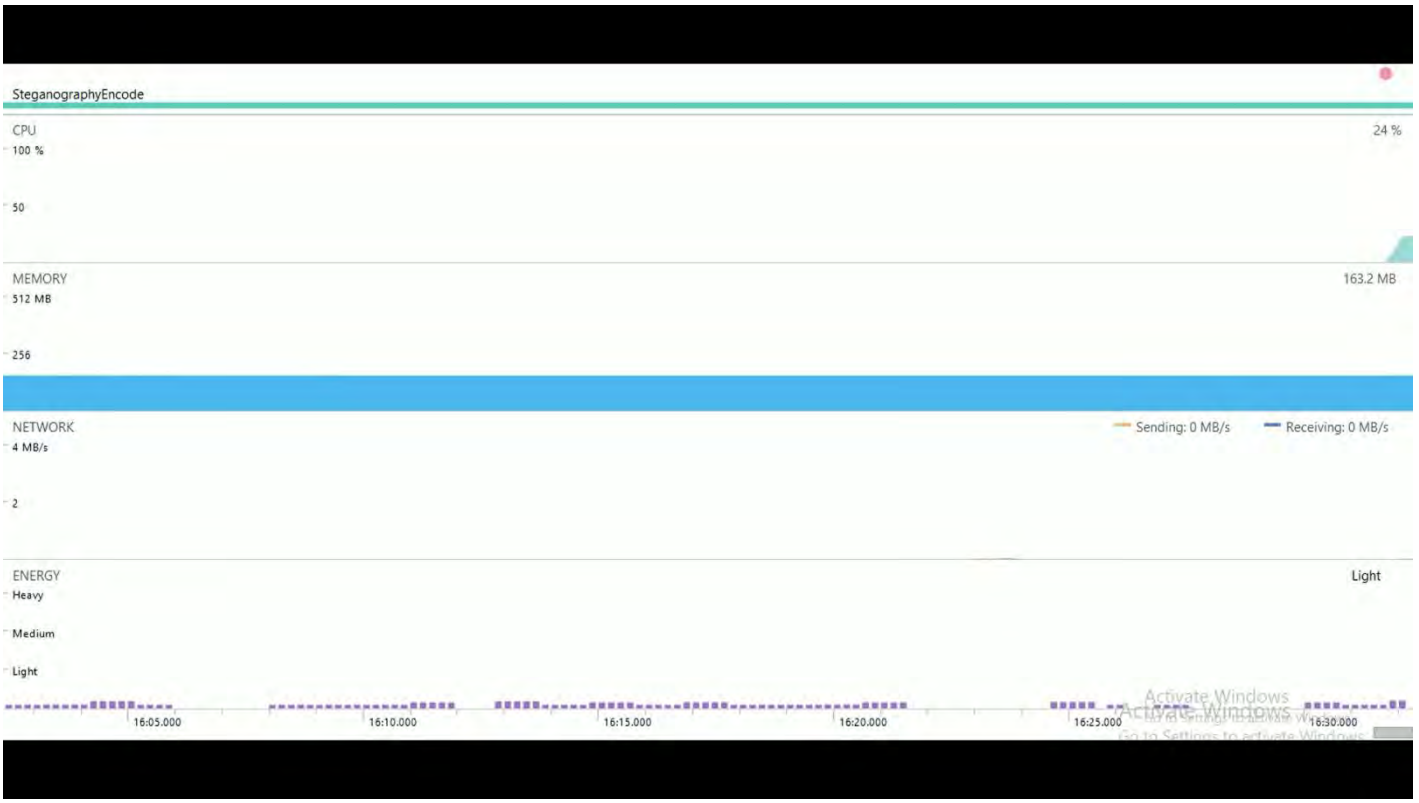

Figure 6 Graph of RSA 2

Chosen Algorithm

The algorithm that has been chosen is ECC after evaluating asymmetric algorithms. ECC provides an equal amount of security as RSA but with smaller key sizes; [14] also states that. ECC provides useful power/battery consumption as compared to RSA. [24] Also states that the security of ECC is highest among all the public key cryptographic algorithms. Because of these features ECC is being used wildly for cellular programs.

Symmetric Cryptography

The other names of Symmetric Cryptography are Conventional Cryptography, Private Key Cryptography, Secret Key Cryptography and Single Key Cryptography. The key is only between sender and receiver. Single cryptographic key is used for both encryption and decryption of messages. Symmetric cryptography was commonly used before the invention of asymmetric key cryptography in 1970's.

Requirements for Symmetric Cryptography

Symmetric cryptography requires two things to provide confidentiality:

- Encryption Algorithm
- A shared secret key known only to sender and receiver.

Conventional Encryption Model

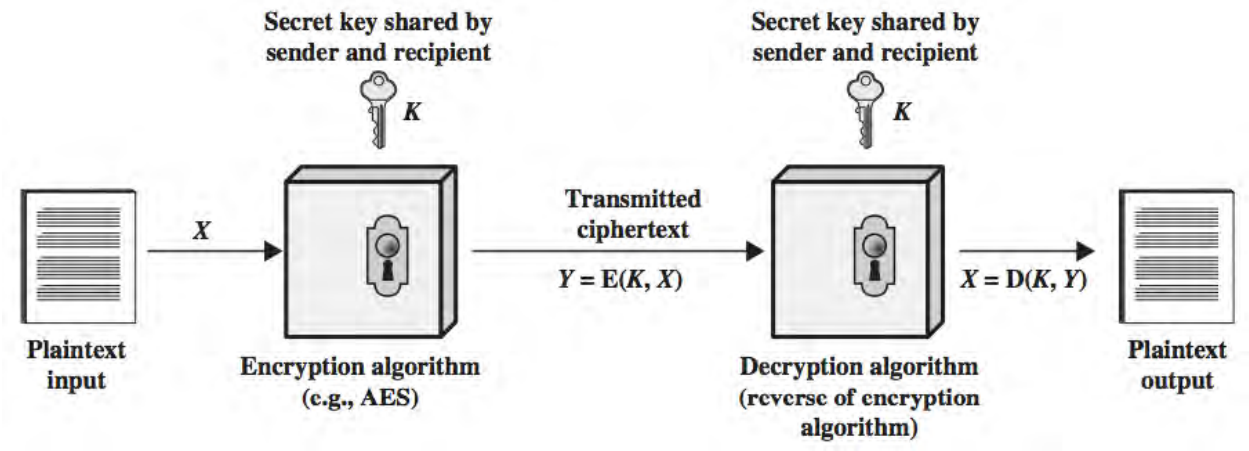

Figure 7 Conventional Encryption Model

The equations for encryption and decryption processes are:

$C = E_k(P)$ (Encryption)

$P = D_k(P)$ (Decryption)

Where,

P is plaintext

C is ciphertext

K is secret key

E is encryption algorithm

D is decryption algorithm

Techniques of Symmetric Cryptography

Substitution:

In the substitution technique of symmetric cryptography encryption is achieved by replacing letters of message (plaintext) with other letters, numbers and symbols. For example Caesar Cipher, Mono-Alphabetic Cipher, Poly-Alphabetic Cipher (Vigenere Cipher)

Transposition:

In the transposition technique of symmetric cryptography encryption is achieved by changing the positions of letters and/or symbols of message (plaintext). This technique is also called permutation.

Example Algorithms

Example algorithms of symmetric cryptography are AES (Advanced Encryption Standard), DES (Data Encryption Standard), 3-DES (Triple Data Encryption Standard), Blowfish, CAST5, IDEA, TEA, Two fish, RC6, RC5, Serpent and MARS.

Criteria for Comparison of Algorithms

The algorithms chosen for comparison are AES, DES, 3-DES, IDEA, Blowfish, Twofish and CAST-128. The parameters of comparative analysis are:

Architecture:

The architecture criteria define the structure that the algorithm follows, type (block/stream), block size it operates on, size of key, Number of rounds and whether the algorithm generates sub-key or not.

The precise description is as follows:

a) **Type:**

- **Block Cipher:**

  The block ciphers divide the messages in blocks of different sizes during cryptographic process (encryption/ decryption).

- **Stream Cipher:**

  The stream ciphers encrypt/decrypt the messages as a bit or byte per unit time during the cryptographic process (encryption/decryption).

b) **Block Size:**

If the block size on which the algorithm operates on is large than the security of that algorithm is also great; they have a direct proportional relationship. But due to the large size of block the cryptographic process (encryption/ decryption) slows down; they have inverse proportional relationship.

c) **Number of Rounds:**

By increasing the number of rounds an algorithm has, the security of that algorithm gets improved; but the efficiency of the algorithm decreases. The number of rounds most of the algorithms have is 16 rounds.

d) **Sub- Key Generation:**

If an algorithm generates a sub key during encryption/decryption then the complexity of that algorithm also increases. The advantage of increased complexity is that it makes the cryptanalysis hard but there is a tradeoff that it slows down the encryption process.

Security:

The resistance of an algorithm against attacks defines security. Usually key size measures the security.

- **Key Size:**

Large key sizes slow down the cryptography process but it also increases security by making the exhaustive searching difficult.

Flexibility:

Defines that whether an algorithm can be changed according to requirements.

Scalability:

Scalability is a major element in analyzing an encryption algorithm. The parameters on which scalability depends are memory usage, power usage, encryption time, decryption time, computational efficiency, throughput and performance.

- **Encryption Time:**
  The time taken by an algorithm to encrypt a message is called encryption time.
- **Decryption Time:**
  The time taken by an algorithm to decrypt a message is called decryption time.
- **Power Usage:**
  Defines that how much battery/power the algorithm consumed during encryption/decryption of a message.
- **Memory Usage:**
  Memory usage defines how much memory an algorithm requires for encryption/decryption or number of functions performed by the algorithm. If an algorithm uses a small memory then that algorithm is more efficient; memory usage and efficiency have inverse relation.
- **Encryption Rate:**
  Encryption rate defines the processing time an algorithm takes for a particular data size. The parameters on which it depends are processor speed, complexity of algorithm etc. An algorithm is considered better if it has smaller encryption rate.
- **Throughput:**
  The throughput of an algorithm is calculated by dividing the total size of plaintext by total encryption or decryption time. If the throughput of an algorithm is large than the power consumed by algorithm is small; they also have inverse proportional relationship.
- **Computational Efficiency:**
  The complex computation an algorithm is doing during encryption process of a message defines the computational efficiency.

Limitations (known Attacks):

Limitations of an algorithm are defined in terms of two factors:

1. How often an algorithm is vulnerable to different types of attacks?
2. How fine the algorithm works when all the resources are given?

Comparative Analysis of Algorithms in terms of Architecture

Table 4 Comparative Analyses of Symmetric Algorithms Based on Architecture

S.No	Parameters	DES	3 DES	AES	IDEA	Blowfish	Two Fish	CAST-128
1.	Type	Block Cipher	Block Cipher	Block Cipher	Block Cipher	Block Cipher	Block Cipher	Block Cipher
2.	Plaintext/Ciphertext Block Size (bits)	64 bits	64 bits	128 bits	64 bits	64 bits	128 bits	64 bits
3.	No of S-Boxes	8	8	1	N/A	4	4	4
4.	Number of Rounds	16	48	10/12/14, depending on the key size	8	16	16	If the size of key is less than 80 bits then 12 rounds otherwise 16 rounds.
5.	Structure	Fiestel structure	Fiestel structure, but iterates 3 times	Fiestel structure	Substitution-permutation structure	Fiestel structure	Fiestel structure	Fiestel structure
6.	Key Size (bits)	56 bits (8 parity bits)	112/ 168 bits	128/192/ 256 bits, default key size is 256 bits	128 bits	32-448 bits	128/192/ 256 bits	40- 128 bits; in 8 bits increment fashion.
7.	Sub-Key Generation	Yes, 16 sub-keys	Yes, 16 sub-keys	Yes, 10 sub-keys	Yes, 52 sub keys	Yes, 18 sub-keys	Yes, 10 sub-keys	N/A
8.	Sub Key Size (bits)	48 bits	48 bits	128/192/ 256 bits	128 bits	32 bits	128/192/ 256 bits	N/A
9.	Expanded Key Size (bits)	N/A	N/A	176/208/ 240 bits	N/A	448 bits-4168 bytes	448 bits-4168 bytes	N/A

Comparative Analysis of Algorithms Based on Security

Table 5 Comparative Analyses of Symmetric Algorithms Based on Security

S.No	Parameters	DES	3 DES	AES	IDEA	Blowfish	Two Fish	CAST-128
1.	Key Size (bits)	56 bits	168 bits	128/192/ 256 bits	128 bits	32-448 bits	128/192/ 256 bits	40-128 bits
2.	Strength Factor	2^{56} = 7.2 x 10^{16} keys	It perform DES 3 times with 2-3 different keys	Variable key sizes	Make use of multiple group operations	Variable key sizes	Variable key sizes, no weak keys.	Make use of variable key size operations
3.	Security Level	Insecure	Adequate	Excellent	Strong	Excellent	Good	Strong
4.	Invulnerable for	Linear and differential attacks	Man in the middle attack	Collision attacks, square attacks, impossible differential attacks, reversed key schedule attack	Strong resistance against differential cryptanalysis	Differential related key attacks	Highly resistive to slide attack, related key differential attack and other related key attacks.	Resistant against both linear and differential attacks

Comparative Analysis of Algorithms Based on Flexibility

Table taken from [8].

Table 6 Comparative Analyses of Symmetric Algorithms Based on Flexibility

S.No	Algorithm	Flexible	Modification	Comments
1.	DES	No	None	The structure of DES does not support any modification.
2.	3 DES	Yes	168 bits	The structure of 3 DES is same as DES, it does not support any changes but as it iterates DES 3 times, the key size is extended to 168 bits.
3.	AES	Yes	128,192,256 bits	The structure of AES was extendable to the multiple of 64 bits, have same sub key size as the size of the key.
4.	IDEA	No	None	The structure of IDEA does not support any modifications.
5.	Blowfish	Yes	32-448 bits	Key length must be multiples of 32 bits.
6.	Twofish	Yes	256 bits	Two fish keys, other than the default sizes, are always padded with "0" bits up to the next default.
7.	CAST-128	Yes	64,128,256 bits	64 bit CAST was too expose to different type of linear and differential

				attacks, due to its flexible structure it was modified to 128 and 256 bits, increasing its strength and security.

Comparative Analysis of Algorithms Based on Limitations

Table 7 Comparative Analyses of Symmetric Algorithms Based on Limitations

S.No	Algorithm	Attacks
1.	DES	Linear cryptanalysis attacks, brute force attack, have weak keys.
2.	3 DES	Differential cryptanalysis, related key attacks, Susceptible of certain variations of man in the middle attack.
3.	AES	No serious weakness, inverse cipher implementation is inappropriate.
4.	IDEA	Susceptible to minimum rounds version and weak keys, related key differential timing attack, collision attack and key schedule attacks.
5.	Blowfish	Issues of weak keys and differential attacks.
6.	Twofish	Susceptible to chosen key attack
7.	CAST-128	Susceptible to differential related key attack

Comparative Analysis of Algorithms Based on Scalability

Table 8 Comparative Analyses of Symmetric Algorithms Based on Scalability

S.No	Parameter	DES	3 DES	AES	IDEA	Blowfish	Twofish	CAST-128
1.	Encryption Speed	Slow	Very Slow (Iterates DES 3 times which is already slow)	Very Fast	-	Fast	Fast	-
2.	Memory Usage	-	Moderate	Low	-	High	Low	-

Chosen Algorithm

The algorithm that has been chosen is AES after comparing selected symmetric cryptography algorithms. AES stands for Advanced Encryption Standard. The security level of AES is excellent, it does not have any serious weaknesses, and also AES is found to be faster and better than other symmetric cryptography algorithms [9]. For small devices like cell phones and tablets AES is found to be best [12]. Other selected algorithms have some weaknesses and some of them are slow like DES and 3-DES. DES and IDEA both are found to be slow and have issues of weak keys through which if encryption is applied twice, it is possible to recover plaintext. 3-DES is very slow as it iterates DES 3 times and DES itself is slow. Blowfish is vulnerable to particular types of attacks as mentioned above in Limitations section also it has issues of weak keys. Because of the tradeoffs between evaluation parameters of other algorithms AES is found to best in terms of security, flexibility, performance, power usage, memory usage etc. [11] [6]

Comparative Analysis of Private and Public Key Cryptography

Table 9 Comparative Analysis of Private and Public Key Cryptography

S.No	Parameter	Private Key Cryptography	Public Key Cryptography
1.	Memory Requirement	Requires less memory for encryption/decryption	Requires more memory for encryption/decryption
2.	Computation Speed	Much faster than Asymmetric Cryptography	Slower than Symmetric Cryptography
3.	Key Distribution	Complex, requires a medium through which key is to be distributed like a third party.	Simple, separate keys for encryption and decryption
4.	Number of keys	1 key	2 keys
5.	Key Generation	Easy	Difficult
6.	Computations	Computations are easy	Computations are complex
7.	Technique	Encrypt plaintext as symbols and characters	Encrypt plaintext as numbers
8.	Authentication	Does not provide authentication	Provides authentication (digital signatures)
9.	Key Size	Small key sizes	Large key sizes
10.	Security	Less secure than asymmetric cryptography because of a single shared key.	More secure because it uses two separate keys, one for encryption and other for decryption.
11.	Computational Time (encryption/decryption/key exchange/key generation time)	Less	More

Asymmetric key cryptography compliments Symmetric key cryptography. Due to large key size and greater computational time of asymmetric cryptography it is used for key exchange and data encryption/decryption is done by symmetric cryptography. Combination of both provides better solution.

STEGANOGRAPHY

The steganography technique consists of a cover media into which the secret message is embedded to hide the information from the observer. The process of embedding produces the stego medium whose data is replaced by the secret hidden message.

Why need Steganography

Cryptography has helped a lot in date security. But in cryptography there is a chance of a malicious activity which decrypts the encrypted data; there is also a chance that the encrypted data becomes suppressed or become susceptible to different attacks. Due to these issues the need of steganography arose to disguise human eye. The main purpose of steganography is to insert the secret message in the non-essential pixels of digital image to increase the communication security. Combination of cryptography and steganography provides a better data security.

Image Steganography

Image steganography uses a cover image to conceal/embed/hide the secret data or message in it. The secret data or message is embedded in to the cover image in non-essential pixels. The generated cover image with secret data embedded in it is called stego image. It is the most widely used steganography technique.

General Image Steganography Model

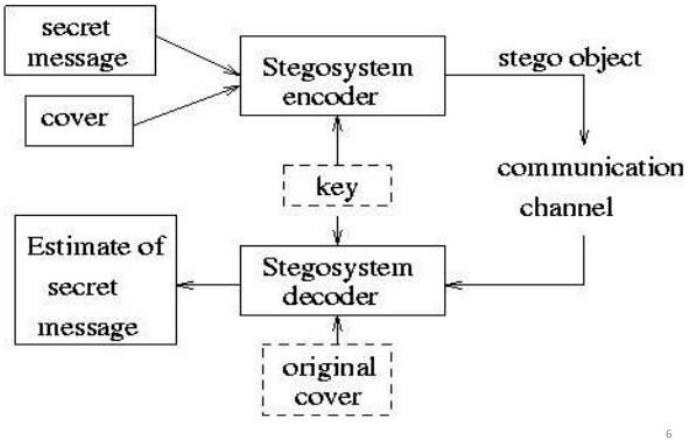

Figure 8 General Model of Image Steganography

Technical Terms

- **Secret Message:**

The data or message that is to be embedded in cover image is called secret message.

- **Cover Image:**

Image in which the secret data embeds is the cover image.

- **Stego Image:**

Stego image is a modified cover image having the secret data or message hidden in it.

- **Stego Key:**

Key used for the process of embedding and extracting data in and from cover image respectively.

- **Embedding Process:**

Embedding Process is a process of hiding secret message in cover image.

- **Extraction Process:**

The process of recovering secret message from the stego image is called extraction process.

- **Steganalysis:**

The art of extracting data from cover image and identifying if the image has data hidden in it or not.

Categorization of Image Steganography

Image steganography is categorized in different types as:

1. **Transform Domain:** It contains JPEG images.
2. **Image Domain:** It includes LSB (Least Significant Bit) and MSB (Most Significant Bit) in BMP and JPG images.
3. **Spread Spectrum:** It contains patch work.

Requirements of Image Steganography

Image steganography requires four things to provide data security:

1. Cover image
2. Steganography algorithm
3. Secret message
4. Stego key

Applications of Image Steganography

1. Spread Spectrum
2. Meteor Scatter Radio
3. Medical Safety
4. Indexing of Voice Mails and so on.

Example Algorithms

The example algorithms of Image steganography are LSB (Least Significant Bit) insertion (Spatial Domain Embedding technique), transform domain methods such as DCT (Discrete Cosine Transform), DWT (Discrete Wavelet Transform) (last two are Frequency Domain Embedding techniques).

Evaluation Criteria

- **Mean Square Error (MSE):**

The average of the squares of "errors" is called mean square error. It is a difference between estimator and what is estimated. [19]

Let us suppose the size of cover image C is M x M, and the size of Stego image S is N x N, then both C and S has pixel value (x,y) from 0 - M-1 and 0 – N-1 respectively. The MSE is calculated as:

$$MSE = 1 / MN \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (C(x,y) - S(x,y))^2$$

- **Peak Signal to Noise Ratio (PNSR):**

  The division of maximum possible value of a signal by the power of distorting noise that affects the quality of its representation is called peak signal to noise ratio [19]. It measures stego image's quality. High value of PSNR means high quality of stego image.

  Let us suppose the size of cover image C is M x M, and the size of Stego image S is N x N, then both C and S has pixel value (x,y) from 0 - M-1 and 0 – N-1 respectively. The PSNR is calculated as:

  $PSNR = 10.\log_{10}(MAX^2 / MSE)$

  Where,

  MAX = Maximum possible pixel value of image.

  MSE = Mean Square Error

- **Varying Degree of Payload:**

  The maximum capacity of cover image to embed data inside is called varying degree of payload.

- **Temper Resistance:**

  The resistance of embedded data in cover image against alteration and change is called temper resistance.

- **Computational Complexity:**

  Computational complexity refers to the level of difficulty to embed and extract data to and from cover image.

- **Robustness:**

  In case of image manipulations like image cropping, image scaling, image filtering and addition of noise, and statistical attacks the embedded data in cover image should stay fixed.

- **Perceptual Transparency:**

  The quality of cover image should not degrade after the embedding process.

Comparative Analysis of Steganography Algorithms

Table 10 Comparative Analyses of Steganography Algorithms

S. No	Parameter	LSB	DCT	DWT
1.	Technique	Embeds message in the Least Significant Bits of cover image.	Transforms the image to frequency domain which gives coefficients. The coefficients are then used to hide the message in cover image.	Decompose the image in sub bands which are based on frequency components.
2.	Change in Stego Image	Results in insignificant changes in cover image's color.	No Significant changes in stego image as compared to cover image.	Results in no changes.
3.	Capacity of Embedding	High	Less	Less
	Distortion	Yes, in each pixel if the number of embedded data bits exceeds 3 then cause distortion.	Minimum distortion	No distortion
5.	Vulnerability	Yes, Vulnerable to statistical detection methods	Immune to basic attacks	Not much affected by statistical attacks
6.	Computational Efficiency	Efficient	Less efficient	Less efficient
7.	Embedding Time	Less	More	More
8.	PNSR	High	Less	Higher
9.	Robustness	Less robust	Robust	Highly Robust
10.	Image Type	BMP (lossless compression), other formats can also be used.	JPG	-
11.	MSE	Less	Higher than LSB	Lesser than LSB

Chosen Technique

The technique that has been chosen is LSB [19]. As the embedding capacity of LSB is high so large payloads of data can be embedded in BMP images; also from the human eye the stego image is less evident [15]. LSB takes less time for embedding and extracting processes because it is efficient. Because of hiding with coefficients in Frequency domain techniques the payload

capacity is less although it prevents statistical detections. Other algorithms have tradeoffs between robustness and payloads.

CHAPTER 2

SOFTWARE PROJECT MANAGEMENT PLAN

2.1 INTRODUCTION

This chapter briefly describes the project, its deliverables, the project milestones, tools and techniques used for the development of project and roles and responsibilities.

2.2 PROJECT OVERVIEW

This project is an android application. The project will allow the user to encrypt and decrypt images, text messages and files by using Hybrid Cryptography (Asymmetric Cryptography algorithm "Elliptic Curve Cryptography (ECC)" and Symmetric Cryptography algorithm "Advance Encryption Standard (AES)") and will also allow the user to conceal the encrypted images, text messages and files inside some other cover image by using Steganography algorithm "Least Significant Bit (LSB)". The combination of hybrid cryptography and steganography will increase the security of confidential data (images, text messages and files) during the transmission and will decrease the chances of data leakage. And user can also send the encrypted data (image/text messages/ files) through this application to other users of the same application.

2.3 PROJECT DELIVERABLES

Project deliverables for this application are:

- Software Project Management Plan (SPMP)
- Software Requirements Specification (SRS)
- Software Design Description (SDD)
- Software Test Documentation (STD)
- Application which enables the encryption/decryption of images, text messages and files.

2.4 PROJECT ORGANIZATION

Software Process Model

I have chosen the Waterfall model because the requirements are clear. The model works in a linear sequential approach. The project's progress flows in one direction that is downward which means no overlapping of phases and the next phase will begin after the completion of first phase for example encryption needs to be done before decryption. That is why waterfall process model is preferred over other models.

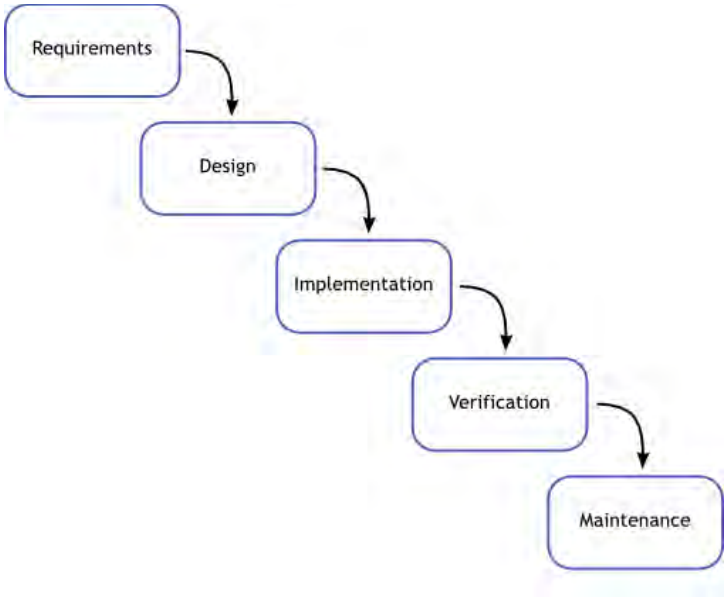

Figure 9 Waterfall Process Model

Roles and Responsibilities

I am responsible for developing this application and there is no further division of roles and responsibility.

Tools and Techniques

The tools and techniques required for the development of the application are shown in table:

Table 11 Tools and Techniques

Sr. No.	NAME	VERSION	DESCRIPTION
1.	MS Word	2010	MS Word is used for project documentation.
2.	Project Libre	1.7	Project Libre is used to make project plan.
3.	Argo UML	v.33.4	Argo UML is used to make use case diagram.
4.	Draw.io	-	An online tool to make system diagrams.
5.	Android Studio	3.0.1	Android Studio is used for the development of application.
6.	Firebase Database	-	Firebase is used to store user data.
7.	Elliptic Curve Cryptography (ECC)	-	Asymmetric Cryptography Algorithm for Encryption and Decryption of AES keys.
8.	Advance Encryption Standard (AES)	-	Symmetric Cryptography Algorithm for the encryption and decryption of images, files and text messages.
9.	Least Significant Bit (LSB)	-	Steganography Algorithm for concealing encrypted images, text messages and files into cover images.

10.	Java SDK	-	Required for android studio.

2.5 PROJECT MANAGEMENT PLAN

This section contains the description of project management plan that is how resources and time will be managed in the development of project.

Tasks

Requirements Analysis Phase:

- Identify Requirements
- Define Use Cases
- Develop Analysis Model
- Develop SRS

Design Phase:

- Develop Design
- Evaluate Design
- Develop Software Test Documentation

Implementation Phase

Integration Phase

Testing Phase

- ## Requirements Analysis Phase

  The following table contains the tasks and sub tasks of Requirements Analysis Phase and their description.

Table 12 Requirements Analysis Phase

Sr. No.	Task		Description
1.	**Identify Requirements**	Description	The initial step in the development of project is the identification of requirements. The requirements include both functional and non-functional requirements.
		Deliverables and Milestones	Requirements are collected and reviewed.
		Resources Needed	• People: Sobia Manzoor Supervisor • Software: MS Word • Hardware: Laptop
		Dependencies and Constraints	None
		Risks and Contingencies	None
2.	**Define Use cases**	Description	This task includes defining and writing use cases and making use case diagram.
		Deliverables and Milestones	Use cases are written down and reviewed
		Resources Needed	• **People:** Sobia Manzoor Supervisor • **Software:** MS Word Argo UML • **Hardware**: Laptop
		Dependencies and Constraints	None
		Risks and Contingencies	None
3.	**Develop Analysis Model**	Description	This task includes making domain model for the system.
		Deliverables and Milestones	Domain model is reviewed.
		Resources Needed	• **People:** Sobia Manzoor Supervisor

			• **Software:** Draw.io • **Hardware**: Laptop
		Dependencies and Constraints	None
		Risks and Contingencies	None
4.	**Develop SRS**	Description	This task includes making Software Requirements Specification document which contains the description of functional and non-functional requirements.
		Deliverables and Milestones	• SRS document is reviewed. • Requirement Analysis phase is complete.
		Resources Needed	• **People:** Sobia Manzoor Supervisor • **Software:** MS Word • **Hardware**: Laptop
		Dependencies and Constraints	None
		Risks and Contingencies	None

- **Design Phase**

The following table contains the tasks and sub tasks of Design Phase and their description.

Table 13 Design Phase

Sr. No.	Task		Description
1.	**Develop Design**	Description	This task includes the development of architectural design and detailed design of the system.
		Deliverables and Milestones	Architectural and detailed design is reviewed.
		Resources Needed	• **People:** Sobia Manzoor Supervisor • **Software:** MS Word Draw.io • **Hardware**: Laptop
		Dependencies and Constraints	The development of designs requires the completion of requirements analysis phase.
		Risks and Contingencies	None
2.	**Evaluate Design**	Description	This task includes evaluation and verification of the design.
		Deliverables and Milestones	Design phase is completed.
		Resources Needed	• **People:** Sobia Manzoor Supervisor • **Software:** MS Word • **Hardware**: Laptop
		Dependencies and Constraints	Architectural Design and Detailed Design of the system should be complete.
		Risks and Contingencies	None
3.	**Develop Software Test**	Description	This task includes defining test cases for the system.

	Documentation	Deliverables and Milestones	Test cases are reviewed.
		Resources Needed	• **People:** Sobia Manzoor Supervisor • **Software:** MS Word • **Hardware**: Laptop
		Dependencies and Constraints	To define the test cases, design phase should be complete.
		Risks and Contingencies	None

- ## Implementation Phase

The following table contains the tasks and sub tasks of Implementation Phase and their description.

Table 14 Implementation Phase

Sr. No.	**Task**		**Description**
1.	**System Implementation**	Description	This phase includes the development of application.
		Deliverables and Milestones	Implementation of the application is complete.
		Resources Needed	• **People:** Sobia Manzoor Supervisor • **Software:** Android Studio Firebase Database • **Hardware**: Laptop Android Cell Phone
		Dependencies and Constraints	The development of application requires the completion of design phase.
		Risks and Contingencies	None

- **Integration Phase**

The following table contains the tasks and sub tasks of Integration Phase and their description.

Table 15 Integration Phase

Sr. No.	Task		Description
1.	**System Integration**	Description	This phase includes the integration of all the software modules.
		Deliverables and Milestones	Application is developed completely.
		Resources Needed	• **People:** Sobia Manzoor Supervisor • **Software:** Android Studio Firebase Database • **Hardware**: Laptop Android Cell Phone
		Dependencies and Constraints	Implementation phase should be completed.
		Risks and Contingencies	None

- **Testing Phase**

The following table contains the tasks and sub tasks of Testing Phase and their description.

Table 16 Testing Phase

Sr. No.	Task		Description
1.	**System Testing**	Description	This phase includes the testing of application based on different testing parameters.
		Deliverables and Milestones	Application is tested.
		Resources Needed	• **People:** Sobia Manzoor Supervisor • **Software:** Android Studio Firebase Database • **Hardware**: Laptop Android Device (Cell phone or tablet).
		Dependencies and Constraints	For the testing of application, integration phase should be complete.
		Risks and Contingencies	None

Timetable

The time table of the project is as below:

	Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Intrusion detection an...**	**80 days?**	**9/25/17 8:00 AM**	**1/12/18 5:00 PM**		**Software;Hardware;sobia**
2	Problem Understanding	1 day?	9/25/17 8:00 AM	9/25/17 5:00 PM		
3	**Software Project Man...**	**4 days?**	**9/26/17 8:00 AM**	**9/29/17 5:00 PM**	**2**	**Software;Hardware;sobia**
4	Introduction	1 day?	9/26/17 8:00 AM	9/26/17 5:00 PM		
5	Project Organization	2 days?	9/27/17 8:00 AM	9/28/17 5:00 PM		
6	Project Management Plan	2 days?	9/28/17 8:00 AM	9/29/17 5:00 PM		
7	**Analysis and Require...**	**24 days?**	**10/2/17 8:00 AM**	**11/2/17 5:00 PM**	**3**	**Software;Hardware;sobia**
8	**Requirements Anal...**	**9 days?**	**10/2/17 8:00 AM**	**10/12/17 5:00 PM**		
9	Define Requirements	6 days?	10/2/17 8:00 AM	10/9/17 5:00 PM		
10	Review Case Study	4 days?	10/9/17 8:00 AM	10/12/17 5:00 PM		
11	**Develope SRS**	**15 days?**	**10/13/17 8:00 AM**	**11/2/17 5:00 PM**	**8**	**Sobia;PC;Software**
12	**Identify Requirem...**	**12 days?**	**10/13/17 8:00 AM**	**10/30/17 5:00 PM**		
13	External Interface ...	3 days?	10/13/17 8:00 AM	10/17/17 5:00 PM		
14	Software Product ...	5 days?	10/17/17 8:00 AM	10/23/17 5:00 PM		
15	Software System A...	2 days?	10/23/17 8:00 AM	10/24/17 5:00 PM		
16	**Database Requir...**	**4 days?**	**10/25/17 8:00 AM**	**10/30/17 5:00 PM**		
17	Identify Entities	1 day?	10/25/17 8:00 AM	10/25/17 5:00 PM		
18	Identify Relationsip	1 day?	10/26/17 8:00 AM	10/26/17 5:00 PM		
19	Develope Domain...	2 days?	10/27/17 8:00 AM	10/30/17 5:00 PM		
20	Review Requirements	1 day?	10/30/17 8:00 AM	10/30/17 5:00 PM		
21	Finalize SRS	3 days?	10/31/17 8:00 AM	11/2/17 5:00 PM		
22	1st Deliveable	1 day?	11/3/17 8:00 AM	11/3/17 5:00 PM		
23	**Develop System Desi...**	**16 days?**	**11/6/17 8:00 AM**	**11/27/17 5:00 PM**	**7**	**Software;Hardware;sobia**
24	**SYSTEM ARCHITECT...**	**6 days?**	**11/6/17 8:00 AM**	**11/13/17 5:00 PM**		
25	Develop Architectura...	5 days?	11/6/17 8:00 AM	11/10/17 5:00 PM		
26	Review Archutectura...	1 day?	11/11/17 8:00 AM	11/13/17 5:00 PM		
27	**Data Design**	**3 days?**	**11/13/17 8:00 AM**	**11/15/17 5:00 PM**		
28	Define Database Ar...	3 days?	11/13/17 8:00 AM	11/15/17 5:00 PM		
29	Normalize ERD	1 day?	11/15/17 8:00 AM	11/15/17 5:00 PM		
30	**Detail Design**	**5 days?**	**11/16/17 8:00 AM**	**11/22/17 5:00 PM**	**14**	
31	Create Sequence Dia...	3 days?	11/16/17 8:00 AM	11/20/17 5:00 PM		
32	Create Class Diagram	2 days?	11/21/17 8:00 AM	11/22/17 5:00 PM	31	
33	**Interface Design**	**3 days?**	**11/23/17 8:00 AM**	**11/27/17 5:00 PM**	**24;27;30**	**People;Software;Hardwa...**
34	Develope Interface ...	2 days?	11/23/17 8:00 AM	11/24/17 5:00 PM		
35	Review Interface De...	1 day?	11/24/17 8:00 AM	11/24/17 5:00 PM		

Figure 10 Project Plan

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 INTRODUCTION

This chapter covers the software requirements specification of Fusion application. The Software Requirements Specification document contains the description of a software system to be developed. It lays out functional and non-functional requirements and includes a set of use cases that describe user interactions with the system.

3.2 PURPOSE

The purpose of Software Requirements Specification is to provide a detailed overview of the software product to be built. It describes the purpose and complete declaration for the development of system. It contains the functional and non-functional requirements of the software product. The final product will be having features mentioned in this document.

3.3 PROBLEM DEFINITION

The mentioned problems have corresponding solutions in the Proposed Solution Section respectively.

- Asymmetric Cryptography encrypts/decrypts limited size of blocks.
- Symmetric cryptography only has a single shared key between sender and receiver so there are issues of key distribution and key breaking.
- Cryptography has helped a lot in data security. But there is a possibility that the malicious user decrypts the encrypted data, or the encrypted data becomes suppressed, or become susceptible.

3.4 PROPOSED SOLUTION

- Symmetric Cryptography is used to encrypt/decrypt data (images/text messages/files). It has no limitation of block size.
- The key of symmetric algorithm is encrypted with asymmetric algorithm before sharing between sender and receiver. And in asymmetric algorithm single key is not shared

between sender and receiver. As the private key is kept secret, using the public key confidential communication is possible.

- Use of steganography to prevent data from human eye. The image steganography inserts the secret message in the non-essential pixels of digital image to increase the communication security.

Asymmetric key cryptography compliments Symmetric key cryptography. Due to large key size and greater computational time of asymmetric cryptography it is used for secure key exchange and data encryption/decryption is done by symmetric cryptography because of its capacity to encrypt/decrypt large amount of data. Combination of hybrid cryptography and image steganography provides a better data security.

ECC has the highest security quality in per bit among all the current public key cryptographic algorithms. ECC provides the same security with a 164 bit key for that purpose different structures require a 1024 bit key. Thus ECC provides equal amount of security with less computing power and useful battery resource usage that is why it is best suitable for cellular programs.

AES is found to be the most secure with no serious weaknesses, faster and better. AES is the best in terms of security, flexibility, performance, power usage, memory usage and so on that is why it is best suitable for small devices. Other algorithms have some tradeoffs between these parameters.

In LSB the large payloads of data is embedded in BMP images and the stego image is less evident from human eyes. LSB is efficient and it takes less time for embedding data in images.

Project Flow

Encryption:

- The image/text file or text message is encrypted with AES algorithm.
- AES algorithm generates the encrypted string (text message/image/file) and AES secret key.
- The AES secret key is then encrypted with ECC algorithm.

- ECC generate keys (public and private) and encrypts AES secret key with public key of receiver. (Single Level Encryption)
- The encrypted string (text message/file/image) is then embedded into cover image using stego key (Double Level Encryption).

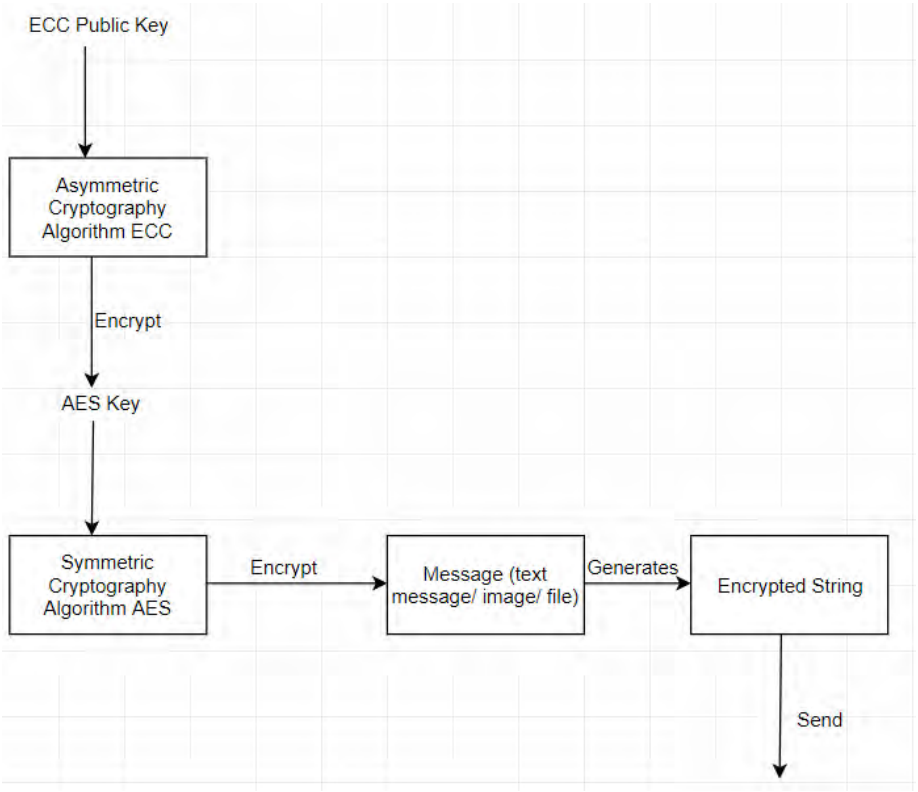

Figure 11 Project Flow Single Level Encryption

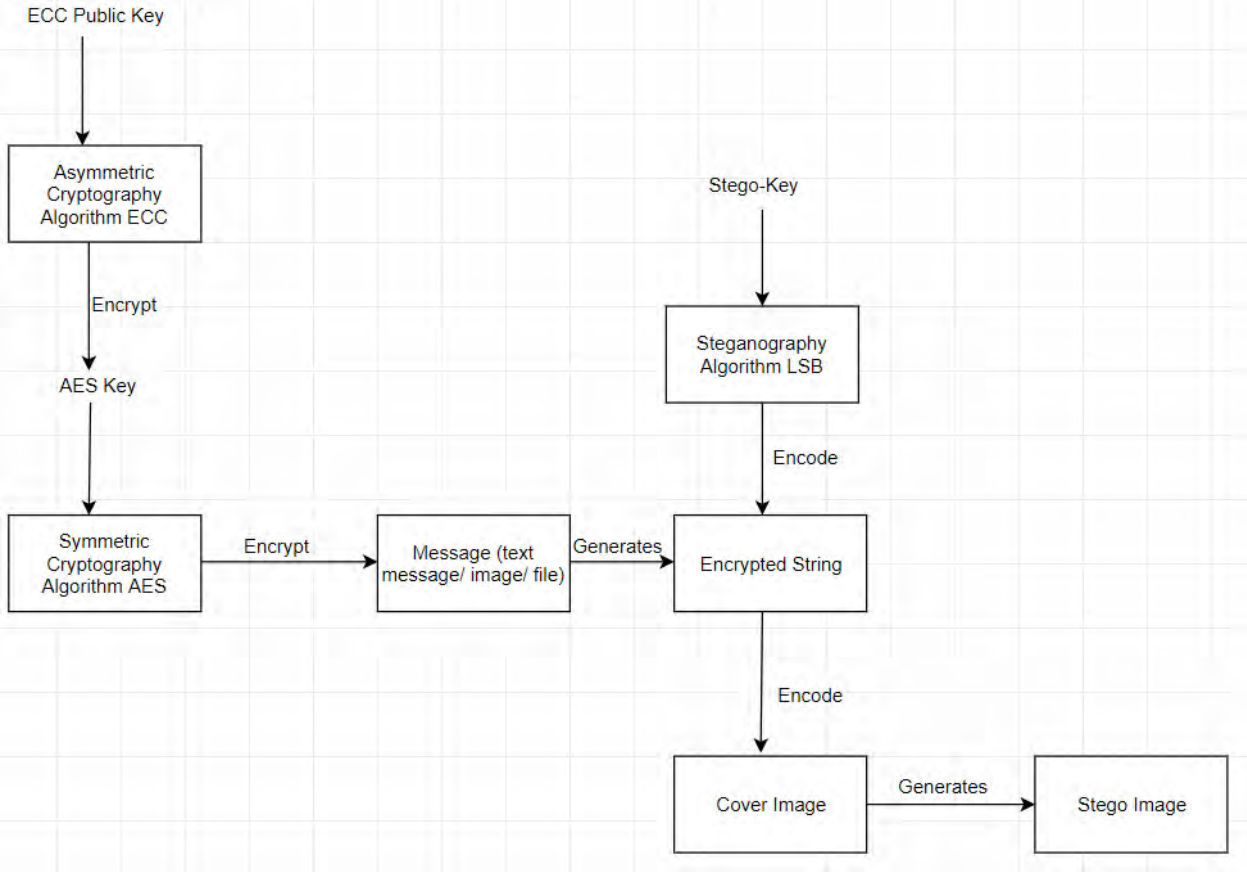

Figure 12 Project Flow Double Level Encryption

Decryption:

- The encrypted string (text message/file/image) is extracted from stego image using stego key (Double Level).
- Receiver uses its private key to decrypt AES encrypted key using ECC.
- AES key is used to decrypt encrypted string (image/file/text message) (Single Level).
- The original content is retrieved.

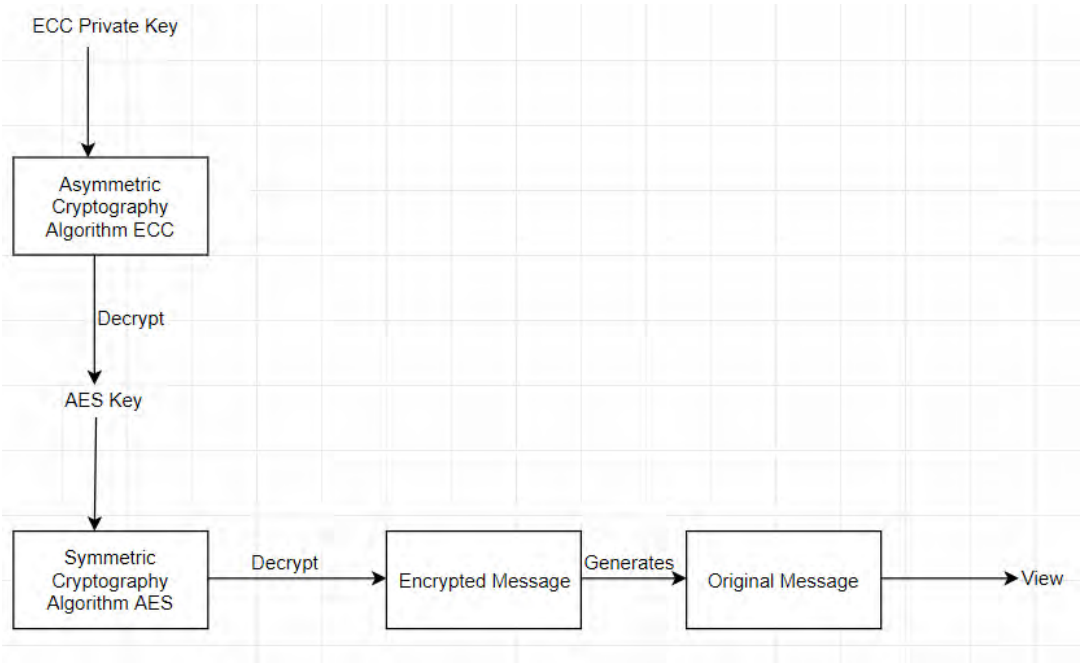

Figure 13 Project Flow Decryption (if message is encrypted with Single Level Encryption)

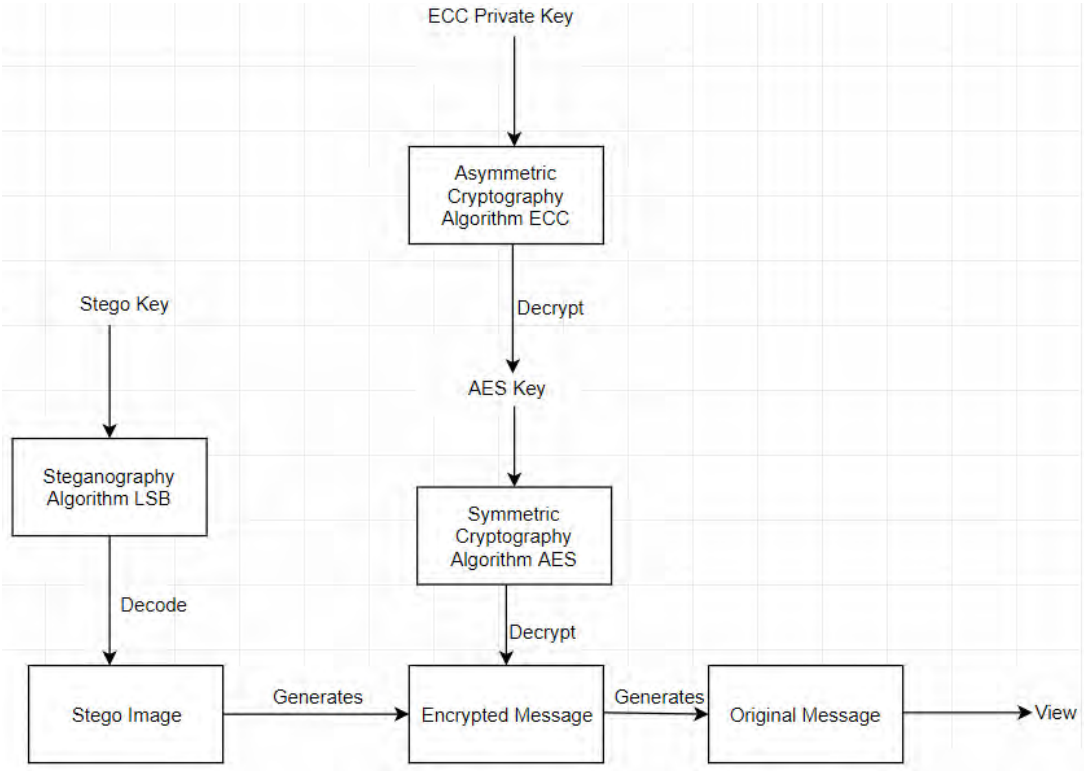

Figure 14 Project Flow Decryption (if message is encrypted with Double Level Encryption)

3.5 SCOPE

The project is an android application which is intended to provide users secure transmission and storage of images, text messages and files using the combination of Hybrid Cryptography (asymmetric cryptography and symmetric cryptography) and steganography algorithms, Elliptic Curve Cryptography, Advance Encryption Standard and Least Significant Bit (LSB) respectively. The application will provide the following functionality:

- Non-Registered users can create their account to get registered.
- One time login to application using user's phone number and also login through email.
- Registered User can set username and set/update user status and profile picture.
- Encrypting images, text messages or files through hybrid cryptography.
- Hiding the encrypted images, text messages or files in another cover image through image steganography.
- Registered User can send/receive the encrypted images, text messages or files.
- Decoding: retrieving encrypted image, text message or file from cover image and then decrypting it.
- Auto saving of encrypted/decrypted data (images/text files).
- Registered users can send or accept requests to or from other registered users.
- Registered users can search people (other users of the application) to send a request, their contacts to start a chat with and their chats to send a message.
- Registered users can delete a request, contact, chat or a particular message.

3.6 OVERALL DESCRIPTION

This section of the SRS contains general information about the software and its requirements. It contains the description of how the system will interact with other systems, constraints and assumptions about the system and the details of system interfaces.

Product Perspective

The benefit of this application is that it allows the users to send and store text messages, files, and images not only in encrypted form but also concealed in another cover image to disguise others about the original contents. The encrypted images, files or text messages will be decrypted by this application. The system is intended to be used on Android Platform. The system is stand alone and not depending on any other system or sub-system.

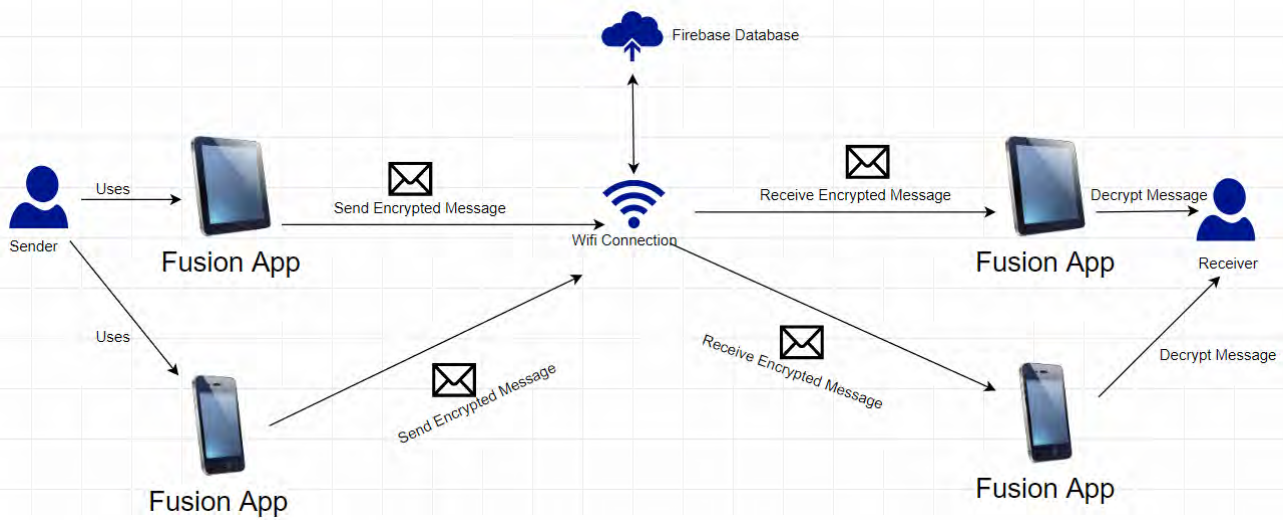

Figure 15 System Block Diagram

System Interfaces

This section describes every system interface and identifies the functionality of the software to accomplish the system requirement and the interface description to match the system.

Software Interfaces

PLATFORM: Android SDK framework

ANDROID EMULATOR: Pixel 2 XL API 29

TECHNOLOGIES USED: Java, XML, ECC, AES, LSB, RSA

DATABASE: Firebase

Hardware Interfaces

The application will work on android smart phones or tablets. The system will interact with camera of device to capture images and android device must have internet connection to run this application.

User Interfaces

The color scheme will be selected in such a way to enhance visibility and contrast; also it does not affect readability. All the interactions between application and user will be done using touch screen interface of the android device. Menus and buttons will be provided to perform different actions like type text messages, select images, encryption/decryption etc. Error messages will also be displayed on the screen. The screen orientation will be portrait.

Communication Interfaces

All the communication between sender and receiver will be performed over the internet.

Product Functions

The system will provide following functions:

Table 17 Product Functions

FUNCTION	DESCRIPTION
Login	Allows registered user one time login to the application using phone number and registered user can also login through email.
Update Profile Picture	Allows registered user to set/update a profile picture for the account.
Set Username	Allows registered user to set a username.
Update User Status	Allows registered user to set/update a status for example "Available".
Load an Image	Allow registered user to load image from gallery or by capturing it using camera for encryption process.
Load File	Allow registered user to load text file from phone for encryption process.
Send Message	Allow registered user to type text message using keyboard for encryption process.
Encryption	Allow registered user to encrypt loaded image, file or typed text message and hide the encrypted string in another cover image.
Save Data (images/ files)	Automatically saves encrypted and decrypted files or images in the phone.
Send Data (images/files/text messages)	Allow registered user to send encrypted images, files or text messages to other users.
Receive Data (text messages/ files/ images)	Allow registered user to automatically receive encrypted images, files or text messages from other users when online.
Delete Chat	Allow registered user to delete entire chat with a particular user including text messages, files and images.
Delete Message	Allow registered user to delete a single message with a particular user including text message, file and image.
Decryption	Decode the stego-image to retrieve encrypted image, file or text message and then decrypt it.
User Online Status	Displays the online/offline status of the registered user.
Display Contacts	Displays a list of existing contacts of a particular user.
Display Chats	Displays a list of existing chats of a user.
Display Single Chat	Displays a chat to the user with other particular user.
Register User	Allow non-registered users to create their accounts using phone numbers or emails to get registered.
Search	Allow registered user to search other users to send a request, contacts to start a chat with particular user, chats to send message to a particular user.
Request	Allow registered users to send/ accept requests to or from other users.

Delete Request	Allow a registered user to delete a request.
Delete Contact	Allow a registered user to delete a contact.

User Characteristics

- The users must have technical expertise to use smart phone and android applications.
- The users should have some knowledge of privacy and security.

General Constraints

- The application will be developed only for android operating system.
- Every user must have internet connection to use this application.
- Every user must be registered first before using this application.
- The android device should have enough memory to install the application, store encrypted and decrypted images and files and the minimum API level should be 28.
- The application cannot encrypt/decrypt more than 1 image/ text message or file at a time.

Assumptions and Dependencies

- Application is dependent on access to Internet.
- The user should know English language as the interface is provided in English only.

3.7 SPECIFIC REQUIREMENTS

Use Case Model

The system has the following use cases:

Registered Users:

- Login
- Update Profile Picture
- Set Username
- Update Status
- Load an Image

- Load File
- Send Message
- Encrypt Message
- Encrypt Message Twice
- Send Message (encrypted text message/image/file)
- Decrypt Message (received text message/image/file)
- Delete Chat
- Delete Message
- Change Settings
- Delete Contact
- Search
- Send Request
- Accept Request
- Delete Request

Non-Registered User:

- Register User by Email
- Register User by Phone Number

Table 18 UC-1 Register User by Phone Number

UC-1	**Register User by Phone Number**
Primary Actor	Non-Registered user
Stakeholder and Interest	Non-Registered user wants to get register.
Pre-Condition	Application must be installed successfully.
Post-Condition	User is registered successfully.
Main Success Scenario	1. User opens the application. 2. System asks user to enter phone number with correct country code. 3. User enters the phone number. 4. System sends user a 6 digit verification code. 5. User receives the verification code. 6. System asks user to enter verification code. 7. User submits the verification code. 8. System validates the phone number. 9. User accesses the application.

Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 3a. User enters invalid phone number. User should enter valid phone number. 3b. User enters the incorrect country code. User should enter the correct country code.
Special Requirements	None
Frequency of Occurrence	Only one time when the application is installed.

Table 19 UC-2 Register User by Email

UC-2	**Register User by Email**
Primary Actor	Non-Registered user
Stakeholder and Interest	Non-Registered user wants to get register.
Pre-Condition	Application must be installed successfully.
Post-Condition	User is registered successfully.
Main Success Scenario	1. User opens the application. 2. System asks user to enter email address and password. 3. User enters the email address and password. 4. User accesses the application.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 2a. User enters invalid email address. User should enter valid email address again. 2b. User enters the invalid password. User should enter a valid password.
Special Requirements	None
Frequency of Occurrence	Only one time when the application is installed.

Table 20 UC-3 Login

UC-3	**Login**
Primary Actor	Registered user
Stakeholder and Interest	Registered user wants to use the application.
Pre-Condition	User must be registered.
Post-Condition	User is logged in successfully.
Main Success Scenario	1. User opens the application. 2. System asks user to enter email address and password. 3. User enters the email address and password. 4. User accesses the application.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 2a. User enters invalid email address. User should enter valid email address again. 2b. User enters the invalid password. User should enter a valid password.
Special Requirements	None
Frequency of Occurrence	Every time user wants to log in.

Table 21 UC-4 Update Profile Picture

UC-4	**Update Profile Picture**
Primary Actor	Registered User
Stakeholder and Interest	User wants to update the profile picture.
Pre-Condition	User must be registered.
Post-Condition	Profile picture is updated successfully.
Main Success Scenario	1. User goes to settings option. 2. System displays settings screen. 3. User selects the profile picture. 4. System asks user to select source to choose picture from for example gallery, camera, drive etc. 5. User selects the source. 6. System opens the selected source. 7. User selects the profile picture. 8. System displays the selected picture and asks to edit. 9. User edits the profile picture that is crop, rotate, flip etc. 10. System displays the edited picture. 11. User updates the profile picture. 12. System saves and displays the selected profile picture.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 11a. User cancels the profile picture.

	System returns to settings screen. 12a. System fails to update picture. User should try to update the picture again.
Special Requirements	Selected profile picture must be in good quality and is not blur/invisible/distorted.
Frequency of Occurrence	Every time user wants to update profile picture.

Table 22 UC-5 Set Username

UC-5	**Set Username**
Primary Actor	Registered User
Stakeholder and Interest	User wants to set a username for his/her account.
Pre-Condition	User must be registered.
Post Condition	Username is set successfully.
Main Success Scenario	1. User goes to settings option. 2. System displays the settings screen. 3. User selects a username option. 4. System displays username field to edit. 5. User sets the username. 6. System displays the typed username. 7. User updates the username. 8. System saves and displays Username.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 7a. User cancels the username. System returns to settings screen. 8a. System fails to update username. User should try to update the username again.
Special Requirements	Username should be meaningful
Frequency of Occurrence	The first time user gets registered.

Table 23 UC-6 Update User Status

UC-6	**Update User Status**
Primary Actor	Registered User
Stakeholder and Interest	User wants to update a status.
Pre-Condition	User must be registered.
Post Condition	User status is updated successfully.
Main Success Scenario	1. User goes to settings option. 2. System displays the settings screen. 3. User selects the user status option. 4. System displays user status field to edit. 5. User sets a status for example “Software Developer”. 6. System displays the typed user status. 7. User updates the user status. 8. System saves and displays the updated user status.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 7a. User cancels the username. System returns to settings screen. 8a. System fails to update username. User should update the username again.
Special Requirements	User status should be meaningful.
Frequency of Occurrence	At any time user wants to update user status.

Table 24 UC-7 Load an Image

UC-7	**Load an Image**
Primary Actor	Registered User
Stakeholder and Interest	User wants to load an image.
Pre-Condition	Registered user is logged in.
Post-Condition	Image is successfully loaded.
Main Success Scenario	1. User goes to the contacts option. 2. System displays the list of existing contacts. 3. User selects a particular contact to send image to. 4. System displays the chat with that user. 5. User inserts image either from gallery or capture via camera. 6. System displays the selected image.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 5a. Image is not loaded properly. Reload the image. 5b. User cancels the selected image. System returns to the user chat screen.

Special Requirements	None.
Frequency of Occurrence	Every time user wants to load an image.

Table 25 UC-8 Load File

UC-8	**Load File**
Primary Actor	Registered User
Stakeholder and Interest	Registered user wants to load file.
Pre-Condition	Registered user is logged in.
Post-Condition	File is successfully loaded.
Main Success Scenario	1. User goes to the contacts option. 2. System displays the list of existing contacts. 3. User selects a particular contact to send file to. 4. System displays the chat with that user. 5. User inserts file from phone storage. 6. System displays the selected file.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 5a. File is not loaded properly. Reload the file. 5b. User cancels the selected file. System returns to the previous screen.
Special Requirements	None.
Frequency of Occurrence	Every time user wants to load file.

Table 26 UC-9 Encrypt Message

UC-9	**Encrypt Message**
Primary Actor	Registered User
Stakeholder and Interest	Registered user wants to encrypt (image/file/text message) to prevent it from unauthorized access.
Pre-Condition	Image/file is loaded and text message is typed successfully.
Post-Condition	Encryption occurs successfully.
Main Success Scenario	1. User prepares the message (image/ file/ text message). 2. User selects the send option. 3. System asks user to whether encrypt AES key using ECC or RSA. 4. User selects the option. 5. System asks the user to select encryption level (single/double). 6. User selects the option for encryption level.

	7. System sends the encrypted data.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 5a.System gives options for encryption level • Single level encryption. • Double level encryption. Extension Point: *Double Encrypt*
Special Requirements	None
Frequency of Occurrence	Every time user wants to encrypt message (file/ text message/image) by single level encryption.

Table 27 UC-10 Encrypt Message Twice

UC-10	**Encrypt Message Twice**
Primary Actor	Registered User
Stakeholder and Interest	Registered user wants to encrypt (image/file/text message) to prevent it from unauthorized access.
Description	Use case "Encrypt Message Twice" is performed by registered user after which encrypted message (from single level encryption) will be embedded in cover image. This use case extends "Encrypt Message" use case and is inserted at extension point Double Encrypt.
Pre-Condition	Message is encrypted (by single level encryption) successfully.
Post-Condition	Double level encryption occurs successfully.
Main Success Scenario	1. User selects the double level encryption. 2. System asks user to select cover image. 3. User selects the cover image. 4. System embeds the encrypted message in cover image and displays.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 4a. System fails to embed encrypted message. User should request for embedding again.
Special Requirements	The cover image should be in good quality, not distorted.
Frequency of Occurrence	Every time user wants to embed encrypted message in cover image.

Table 28 UC-11 Send Message

UC-11	**Send Message**
Primary Actor	Registered user
Stakeholder and Interest	User wants to send the encrypted message.
Pre-Condition	Message is encrypted successfully.
Post-Condition	Message is sent successfully.
Main Success Scenario	1. User sends the encrypted message. 2. System displays the sent message.
Alternative Flow	1a. Message contains: Encrypted Image/File/Text Message 1b. At any time, the application fails to respond. User should restart the application. 2a. System does not send message. User should request for sending again.
Special Requirements	None
Frequency of Occurrence	Every time user wants to send a message.

Table 29 UC-12 Delete Chat

UC-12	**Delete Chat**
Primary Actor	Registered user
Stakeholder and Interest	User wants to delete a particular chat.
Pre-Condition	User is logged in.
Post-Condition	Chat is deleted successfully.
Main Success Scenario	1. User goes to chats option. 2. System displays list of existing chats. 3. User deletes the chat. 4. System removes that chat from chats section.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application 4a. System fails to delete the chat. User should try deleting again.
Special Requirements	None
Frequency of Occurrence	Every time user wants to delete a chat.

Table 30 UC- Delete Message

UC-13	**Delete Message**
Primary Actor	Registered user
Stakeholder and Interest	User wants to delete a particular message.
Pre-Condition	User is logged in.
Post-Condition	Message is deleted successfully.
Main Success Scenario	1. User goes to chats options. 2. System displays the list of existing chats. 3. User opens a particular chat. 4. System displays the chat from which the message is to be deleted. 5. User deletes the message. 6. System removes that message from the chat.
Alternative Flow	1a. Message can be: Image/File/Text Message 1b. At any time, the application fails to respond. User should restart the application 6a. System fails to delete. User should request for deletion again.
Special Requirements	None
Frequency of Occurrence	Every time user wants to delete a message.

Table 31 UC-14 Decrypt Message

UC-14	**Decrypt Message**
Primary Actor	Registered user
Stakeholder and Interest	User wants to recover the encrypted message.
Pre-Condition	Encrypted stego-image/message is received successfully.
Post-Condition	Stego-image/encrypted message is decrypted successfully.
Main Success Scenario	1. User requests for decryption of stego-image/encrypted message. 2. System decrypts the stego-image/encrypted message and displays the original message. 3. User views the original message.
Alternative Flow	1a. Message can be: Text Message/File/Image 1b. At any time, the application fails to respond. User should restart the application. 2a. System does not decrypt the stego-image/encrypted message. User should request for decryption again.
Special Requirements	The decrypted message should be in good quality.

Frequency of Occurrence	Every time encrypted message/stego-image is received.

Table 32 C-15 Change Settings

UC-15	**Change Settings**
Primary Actor	Registered user
Stakeholder and Interest	User wants to make changes in settings.
Pre-Condition	User is logged in.
Post-Condition	Settings are updated successfully.
Main Success Scenario	1. User goes to settings options. 2. System displays the settings screen. 3. User updates the settings. 4. System saves and displays the updated settings.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 4a. System does not update the settings. User should request for updating again.
Special Requirements	None
Frequency of Occurrence	Every time user wants to update settings.

Table 33 UC-16 Delete Contact

UC-16	**Delete Contact**
Primary Actor	Registered user
Stakeholder and Interest	User wants to delete a particular contact.
Pre-Condition	User is logged in.
Post-Condition	Contact is deleted successfully.
Main Success Scenario	1. User goes to contacts options. 2. System displays the list of existing contacts. 3. User deletes the contact. 4. System removes the contact from contacts list.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application 4a. System fails to delete. User should request for deletion again.
Special Requirements	None
Frequency of Occurrence	Every time user wants to delete a contact.

Table 34 UC-17 Delete Request

UC-17	**Delete Request**
Primary Actor	Registered user
Stakeholder and Interest	User wants to delete a particular request.
Pre-Condition	User is logged in.
Post-Condition	Request is deleted successfully.
Main Success Scenario	1. User goes to requests options. 2. System displays the list of received requests. 3. User deletes the request. 4. System removes the request from received requests list.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application 4a. System fails to delete the request. User should request for deletion again.
Special Requirements	None
Frequency of Occurrence	Every time user wants to delete a request.

Table 35 UC-18 Accept Request

UC-18	**Accept Request**
Primary Actor	Registered user
Stakeholder and Interest	User wants to accept the request from sender.
Pre-Condition	User is logged in.
Post-Condition	Request is accepted successfully.
Main Success Scenario	1. User goes to requests options. 2. System displays the list of received requests. 3. User accepts the request. 4. System adds the request sender to the user's contacts lists. 5. System updates the contacts and displays.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application
Special Requirements	None
Frequency of Occurrence	Every time user wants to accept a request.

Table 36 UC-19 Send Request

UC-19	**Send Request**
Primary Actor	Registered user
Stakeholder and Interest	User wants to send request to a user.
Pre-Condition	User is logged in.
Post-Condition	Request is sent successfully.
Main Success Scenario	1. User goes to people options. 2. System displays the list of users. 3. User selects a particular user. 4. System displays that user's profile. 5. User sends the request. 6. System displays that the request is sent.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application. 6a. System fails to send a request. User should send the request again.
Special Requirements	None
Frequency of Occurrence	Every time user wants to send a request.

Table 37 UC-20 Search

UC-20	**Search**
Primary Actor	Registered user
Stakeholder and Interest	User wants to search people, contacts or chats.
Pre-Condition	User is logged in.
Post-Condition	Search is completed successfully.
Main Success Scenario	1. User goes to search option. 2. System displays the search menu. 3. User selects an option from the menu. 4. User searches the required information. 5. System displays the search results.
Alternative Flow	1a. At any time, the application fails to respond. User should restart the application 2a. Options available in menu are: • Search People • Search Contacts • Search Chats 5a. System fails to search. User should search again.
Special Requirements	None

Frequency of Occurrence	Every time user wants to make a search.

Use Case Diagram

Argo UML is used to make use case diagram of the system and is as follows:

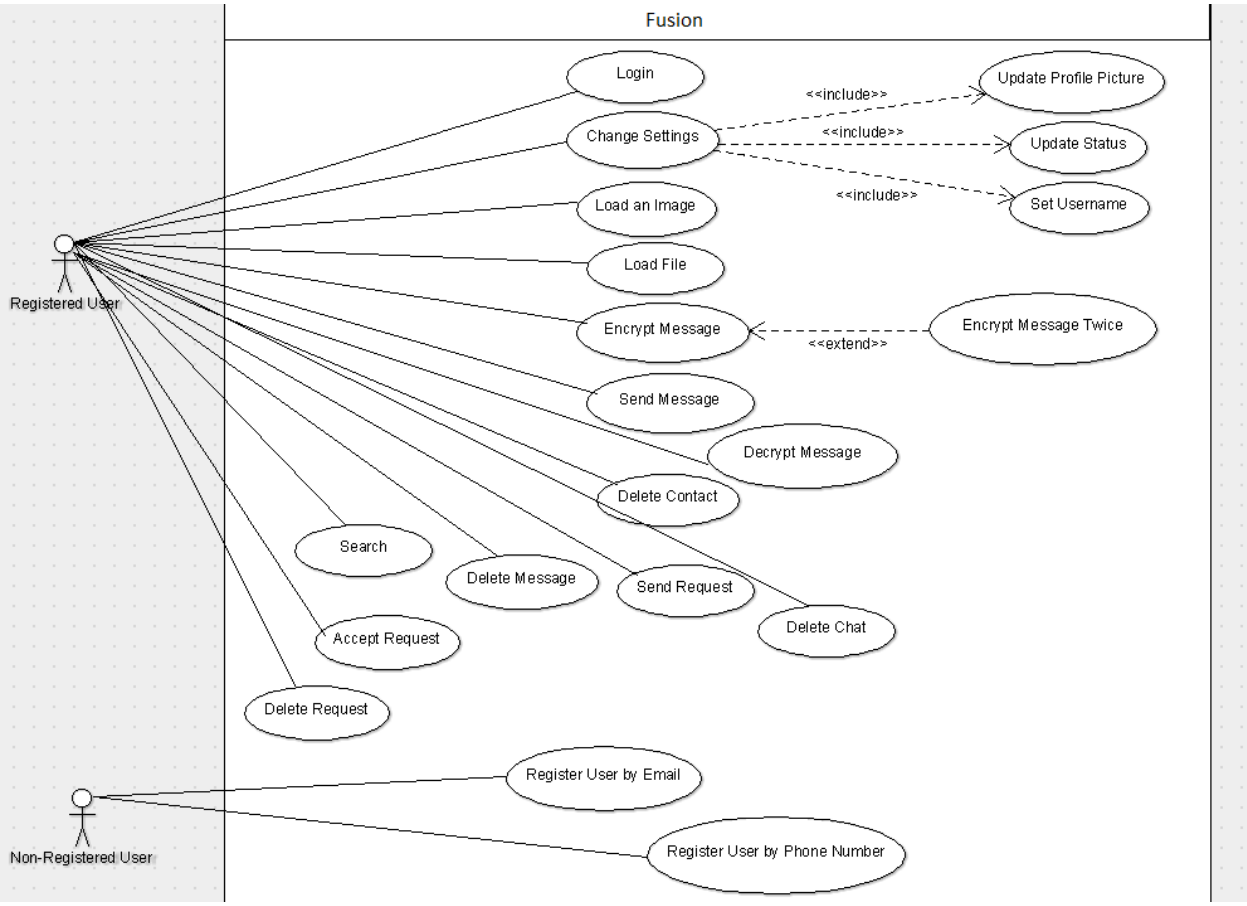

Figure 16 Use Case Diagram

Domain Model

Draw.io is used to make domain model of the system which shows the relationship between real world entities.

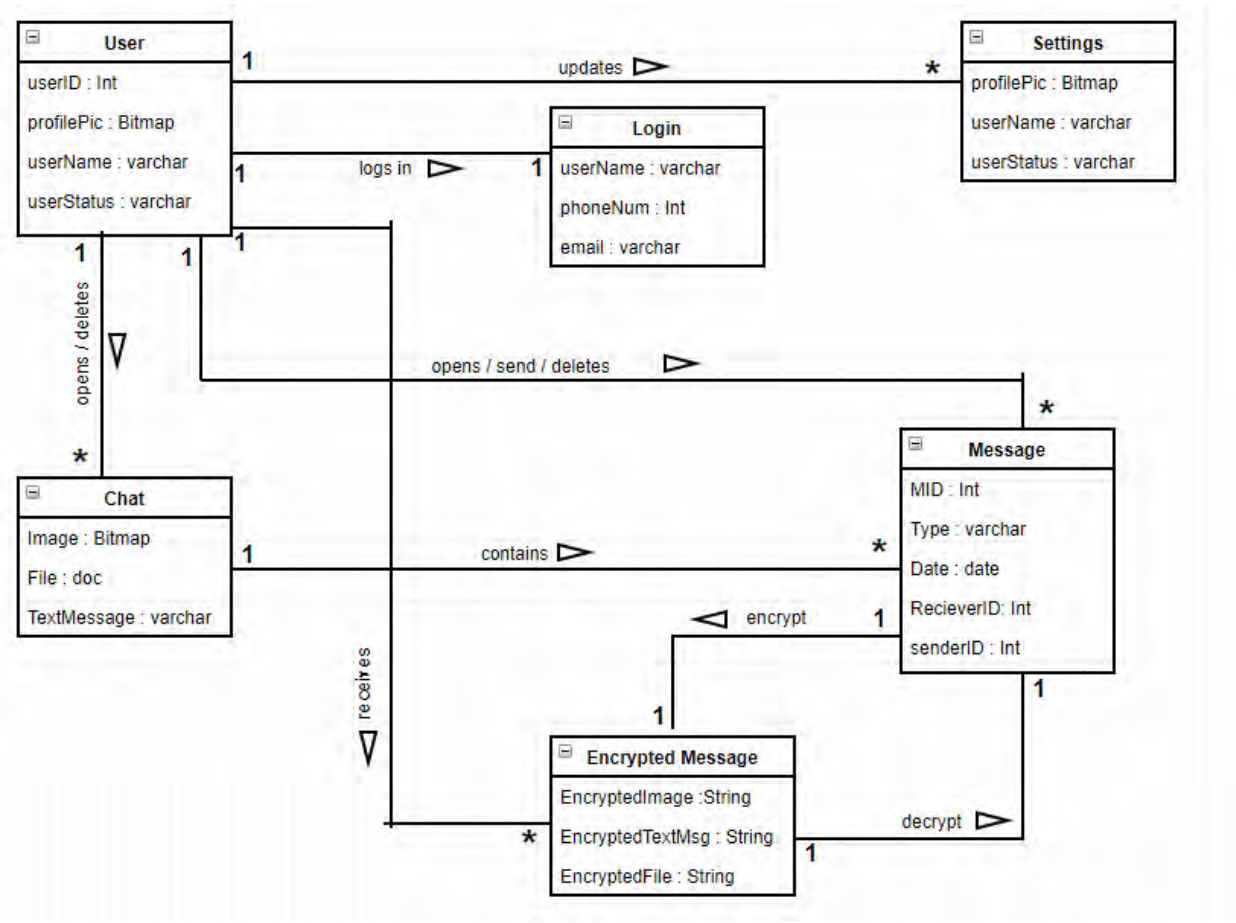

Figure 17 Domain Model

3.8 PERFORMANCE REQUIREMENTS

Number of Terminals Required

Two terminals are required for sender and receiver.

Number of Simultaneous Users

One user performs encryption and the other user (receiver) performs decryption.

Amount and Type of Information to be handled

Text message, files and images will be handled. And one text message, file or image can be encrypted or decrypted at a time.

3.9 SOFTWARE QUALITY ATTRIBUTES

Software quality attributes define overall factors that affect run-time behavior, application design, and user experience. To develop high quality application, software system attributes are the benchmarks that describe system's intended behavior within the environment for which it was built. The software quality attributes for the system are:

Reliability

The application shall never crash, other than error cause by failure of the operating system. The application must produce correct and consistent result that is it should be consistent with the quality of images before and after encryption and decryption. If there is an error, then application will display an appropriate error message such that user will not feel any ambiguity while using the application.

Availability

The system is an android based application so it will be available all the time after its installation, but it also requires an internet connection to avail the services of the application.

Security

The security section describes the need to control access to the data. This includes controlling who may view and alter application data.

- Only registered users can use the application.
- Password sent to the database for login purpose will be in encrypted form.
- User data like images, text messages, files will be stored in database in encrypted form.

Maintainability

Modular approach will be used to maintain the software and to ensure any future modification. Updates will be done in some separate files so that actual software will not be disturbed.

Portability

Fusion is an android based application so it ensures the portability. It will require an android phone to run application. It requires a smart phone with minimum version up to 5.1 (Lollipop) and the API level is 28.

Performance

Fusion is dependent on the Internet connection so its performance could be affected by the internet speed and also by the specification of the application device.

3.10 DATABASE REQUIREMENTS

No SQL database will be used for this system to store all information. No SQL database provides a mechanism for storage and retrieval of data that is modeled in, means other than tabular relations (ERD) used in relational databases.

I have used Firebase database for my project which is No SQL, real-time and no relational. Data is stored in tree structure and in JSON format.

CHAPTER 4

SOFTWARE DESIGN DESCRIPTION

4.1 INTRODUCTION

The Software Design Document is a document which provides a written description of the software design of the product. It provides details of how the system should be built. The Software Design Document contains the description and graphical documentation of the software design for the project including Architecture Design which contains description of the architecture of the software and Interface Design which describes internal and external program interfaces and design of user interfaces. It includes the description of how the software will meet the requirements.

4.2 PURPOSE

The purpose of the Software Design Document is to describe the architecture and system design of Fusion application. It contains information necessary to provide description of the details for the software to be built. It allows the software development to proceed with an understanding of what is to be built and how it is expected to be build.

4.3 REQIREMENTS TRACEABILITY MATRIX

The Requirements Traceability Matrix or RTM captures all requirements proposed by the client or development team and their traceability in a single document delivered at the conclusion of the life cycle.

In other words, it is a document that maps and traces user requirements with test cases, interfaces and sequence diagrams.

Table 38 Requirements Traceability Matrix

Requirement ID	Requirement Name	Sequence Diagram	Interface	Test Case
UC-1	Register User by Phone Number	Yes	Yes	Yes
UC-2	Register User by Email	Yes	Yes	Yes
UC-3	Login	Yes	Yes	Yes
UC-4	Update Profile Picture	Yes	Yes	Yes
UC-5	Set Username	Yes	Yes	Yes
UC-6	Update User Status	Yes	Yes	Yes
UC-7	Load an Image	Yes	Yes	Yes
UC-8	Load File	Yes	Yes	Yes
UC-9	Encrypt Message	Yes	Yes	Yes
UC-10	Encrypt Message Twice	Yes	Yes	Yes
UC-11	Send Message	Yes	Yes	Yes
UC-12	Delete Chat	Yes	No	Yes
UC-13	Delete Message	Yes	No	Yes
UC-14	Decrypt Message	Yes	Yes	Yes
UC-15	Change Settings	Yes	Yes	Yes
UC-16	Delete Contact	Yes	No	Yes
UC-17	Delete Request	Yes	Yes	Yes
UC-18	Accept Request	Yes	Yes	Yes
UC-19	Send Request	Yes	Yes	Yes
UC-20	Search	Yes	Yes	Yes

4.4 SYSTEM ARCHITECTURAL DESIGN

Architectural design is concerned with understanding of how a system should be organized and designing the overall structure of that system. In the model of the software development process, architectural design is the first stage in the software design process. It is a critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them. The output of an architectural design process is an architectural model that describes how the system is organized as a set of communicating components.

Chosen System Architecture

The chosen system architecture is 3-tier. According to Techopedia "3-tier architecture is client/server architecture in which the functional process logic, data access, computer data storage, and user interface are developed and maintained as independent modules on separate platforms". 3-tier architecture is a software design pattern and well-established software architecture.

System Architecture design of Fusion is shown in the diagram below. Draw.io is used to make this diagram.

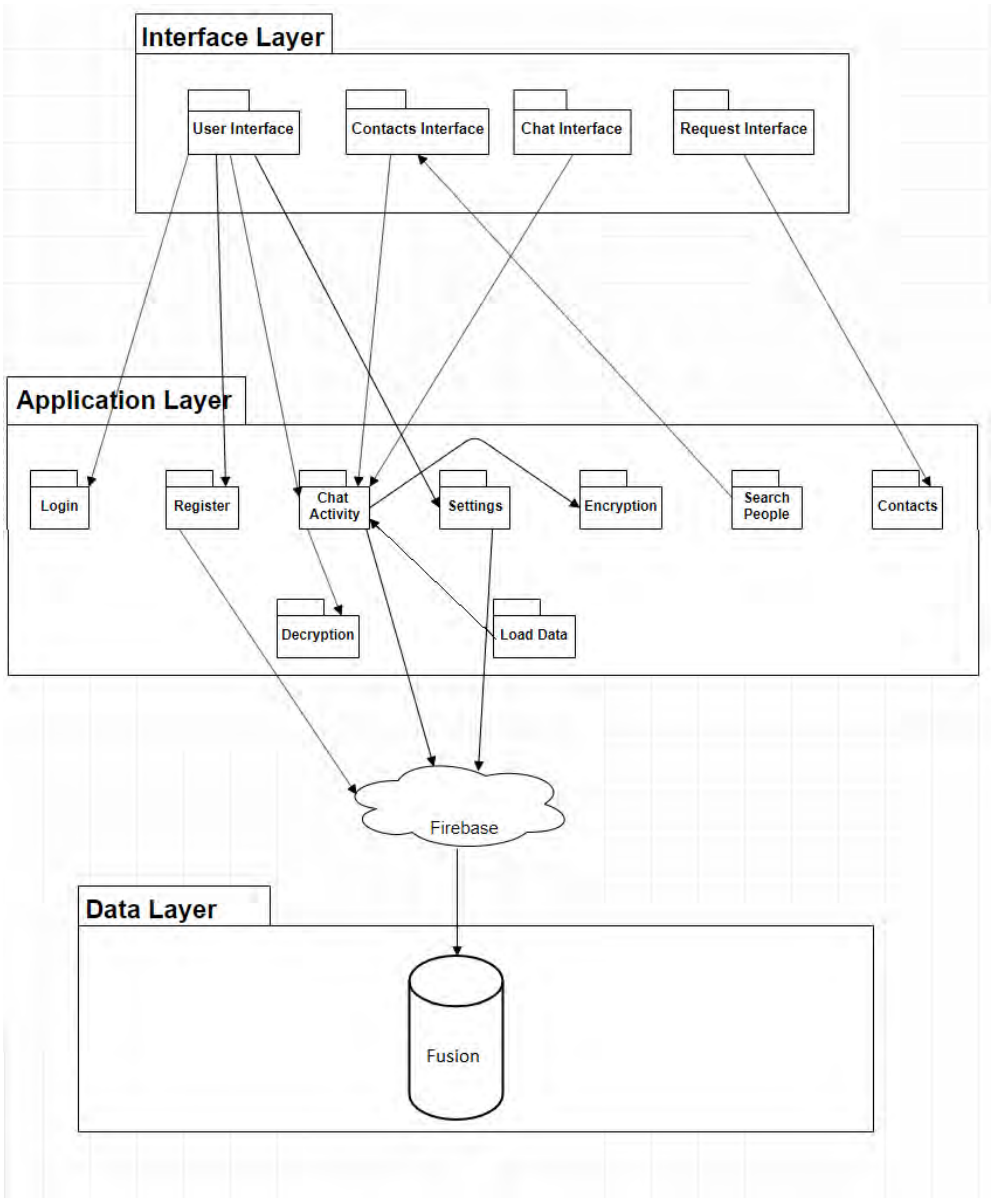

Figure 18 System Architecture Design

4.5 DETAILED DESCRIPTION OF COMPONENTS

A component diagram, also known as UML component diagram, describes the organization and wiring of the physical components in the system.

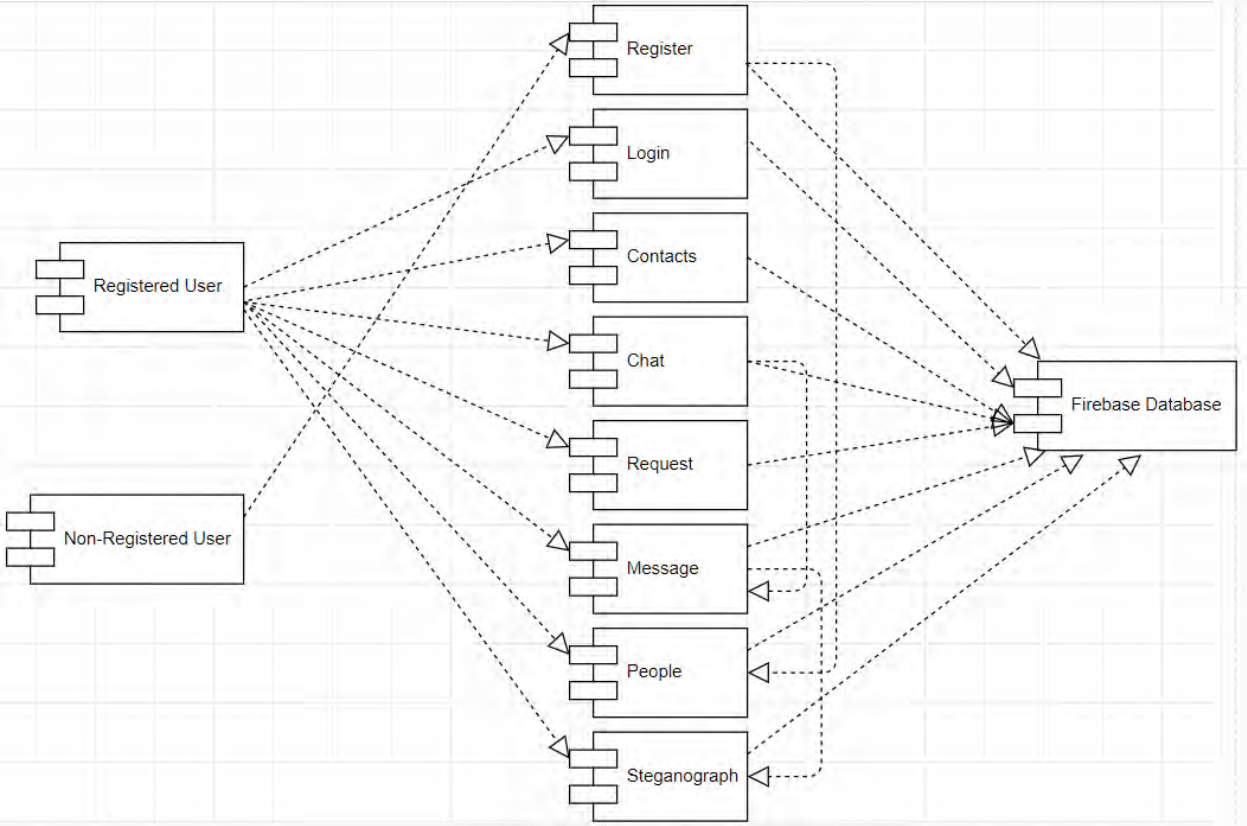

Figure 19 Component Diagram

4.6 SYSTEM INTERFACE DESCRIPTION

System Interface Description is used to depict systems and sub-systems and identify the resources flows between them. It is a logical characteristic of each interface between the software product and the hardware components of the system. The system interface of Fusion System is shown below.

Draw.io is used to make the system interface diagram.

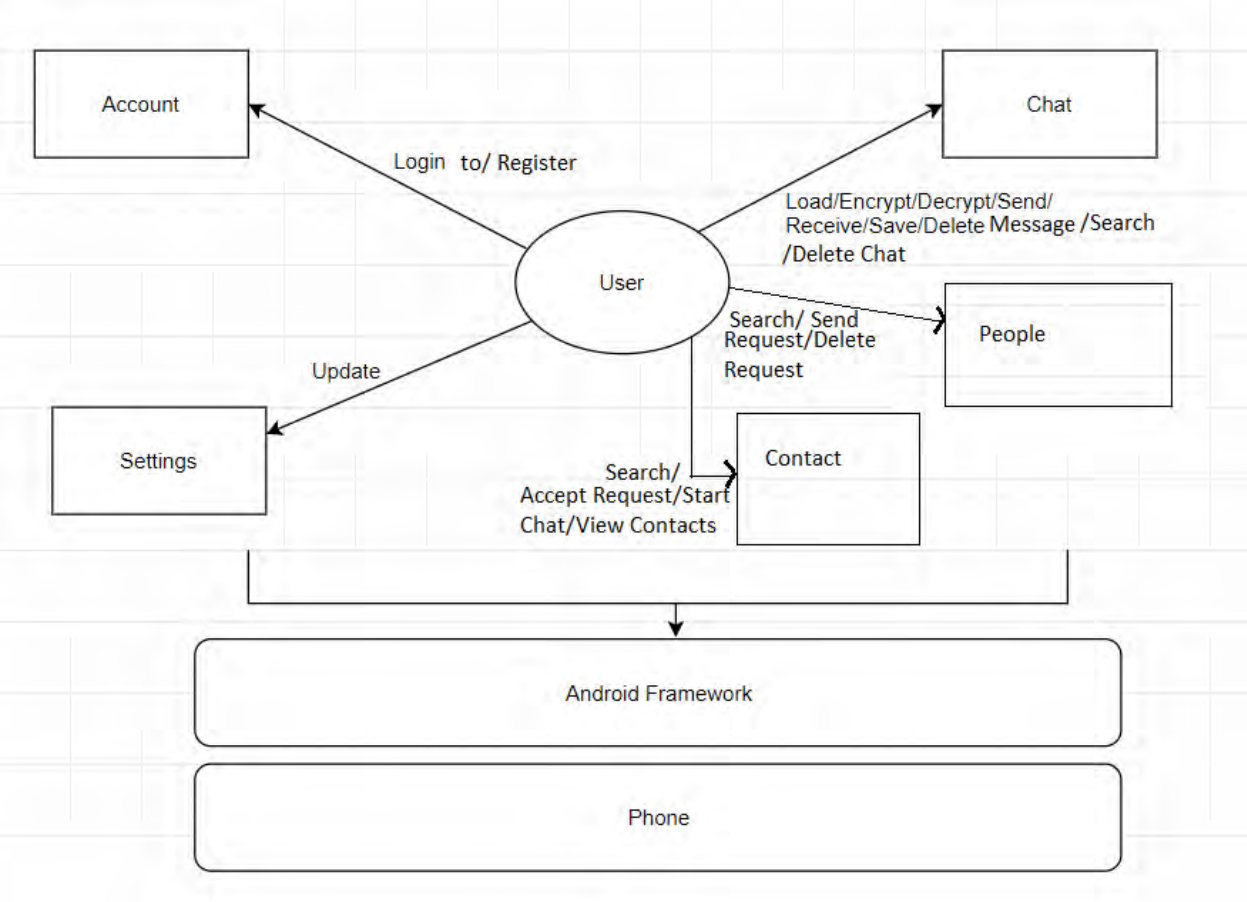

Figure 20 System Interface

4.7 USER INTERFACE DESIGN

A User Interface or UI is a junction between the software product and the user. In this section the user interface of Fusion System is discussed.

Description of the User Interface

In Fusion System user can interact with the system by using touch screen interface of Android device (cell phone or tablet). The color scheme will be selected in such a way to enhance visibility and contrast also it does not affect readability. Menus and buttons will be provided to perform different actions like type text messages, select images, encryption/decryption etc. Error messages will also be displayed on the screen. The screen orientation will be portrait.

When user clicks on the application icon a splash screen will appear for few seconds and then the Login screen appears. Home Screen contains the following sections Chat, Contacts, Requests, People and Settings.

- When the user clicks on the Chat option, the list of existing chats will be appeared.
- On clicking a particular chat in a list, the chat opens.
- In that chat user can upload file, image or write text message. The uploaded item then encrypted and embedded in another cover image, and can be send or save. Similarly the received stego images are decrypted and restored to original content that is text message, file or image.
- When user clicks on the contacts option, the list of contacts will be appeared. On clicking a particular contact user can start chatting with him/her.
- When user goes to settings option, the settings screen will appear where user can update profile picture, user status or username.

Prototypes

Some of the prototypes of screens are:

Figure 21 Prototype for Login A

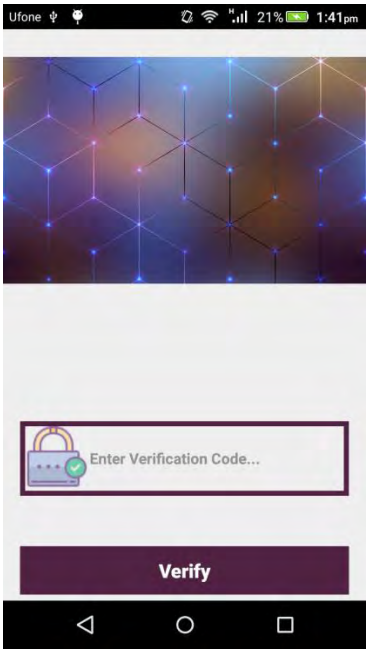

Figure 22 Prototype for Login B

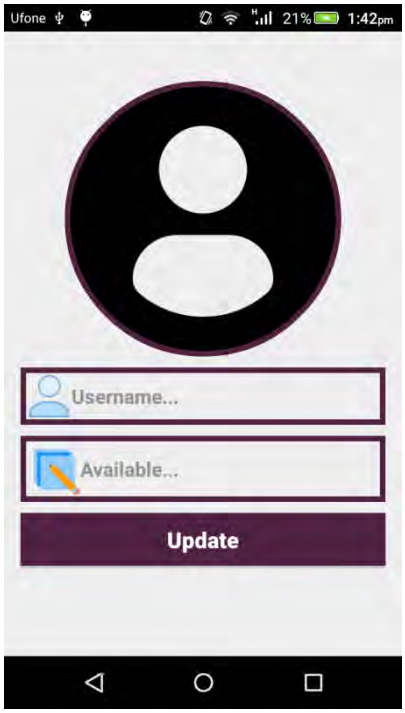

Figure 23 Prototype for Settings

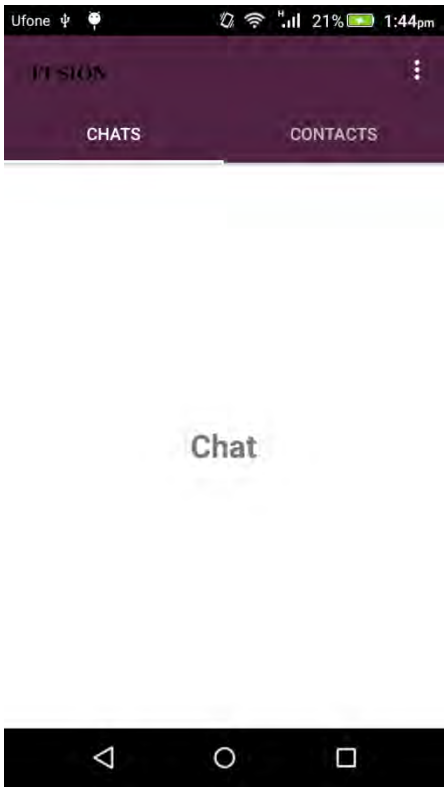

Figure 24 Prototype for Home Screen

4.8 SYSTEM SEQUENCE DIAGRAM

A System Sequence Diagram is an interaction diagram that shows how objects interact with one another and in what order. The System Sequence Diagrams of the Fusion made in Draw.io are given below.

SSD for Phone Number Login

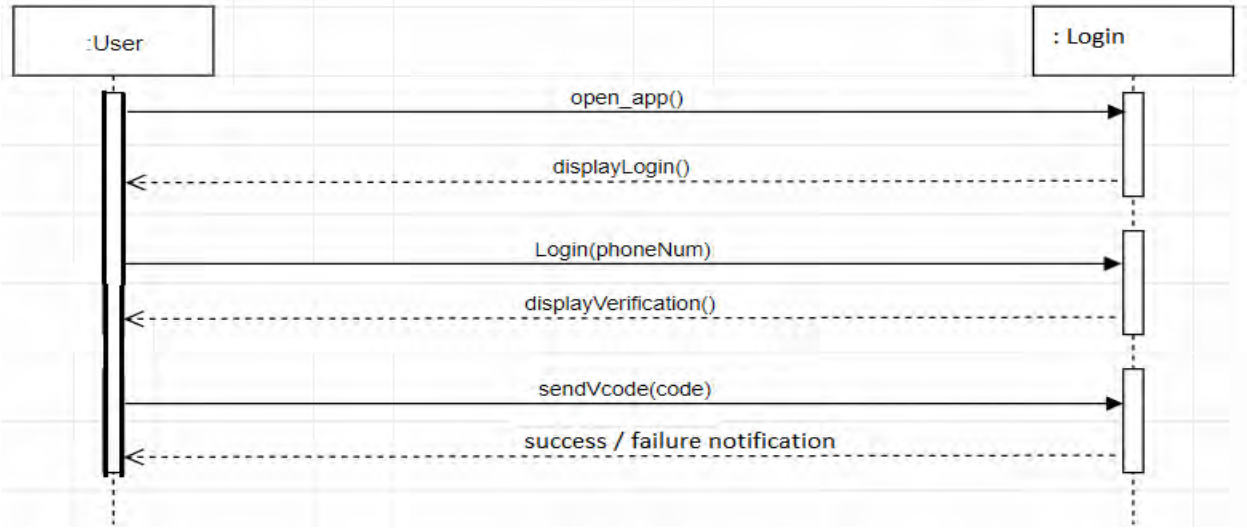

Figure 25 SSD for Phone Number Login

SSD for Update Profile Picture

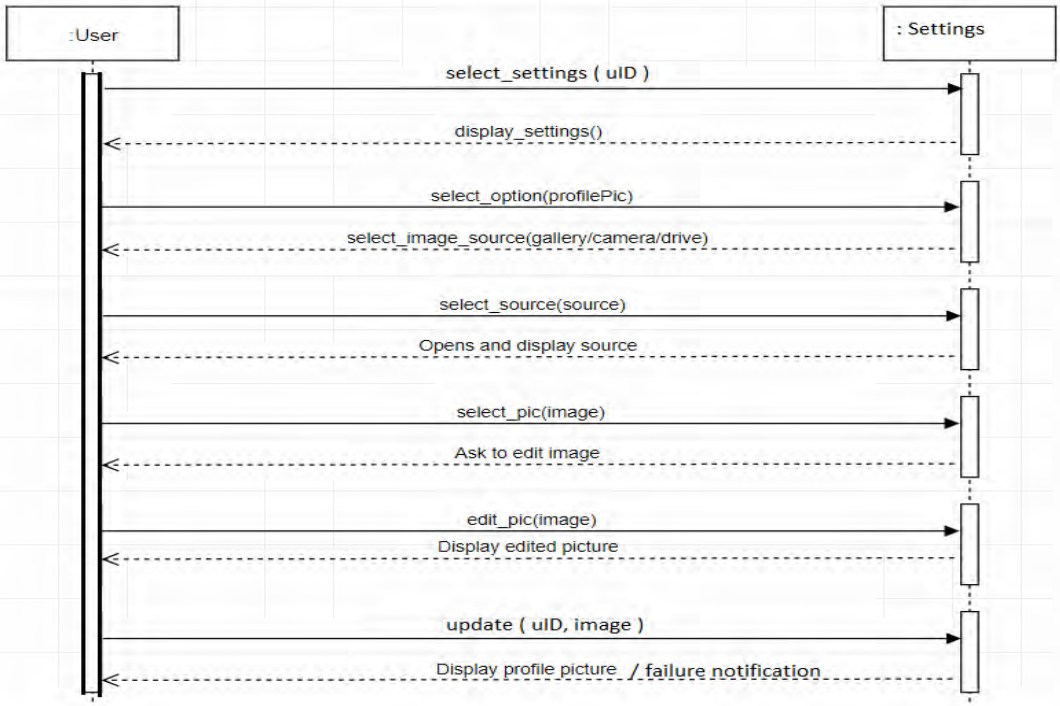

Figure 26 SSD for Update Profile Picture

SSD for Set Username

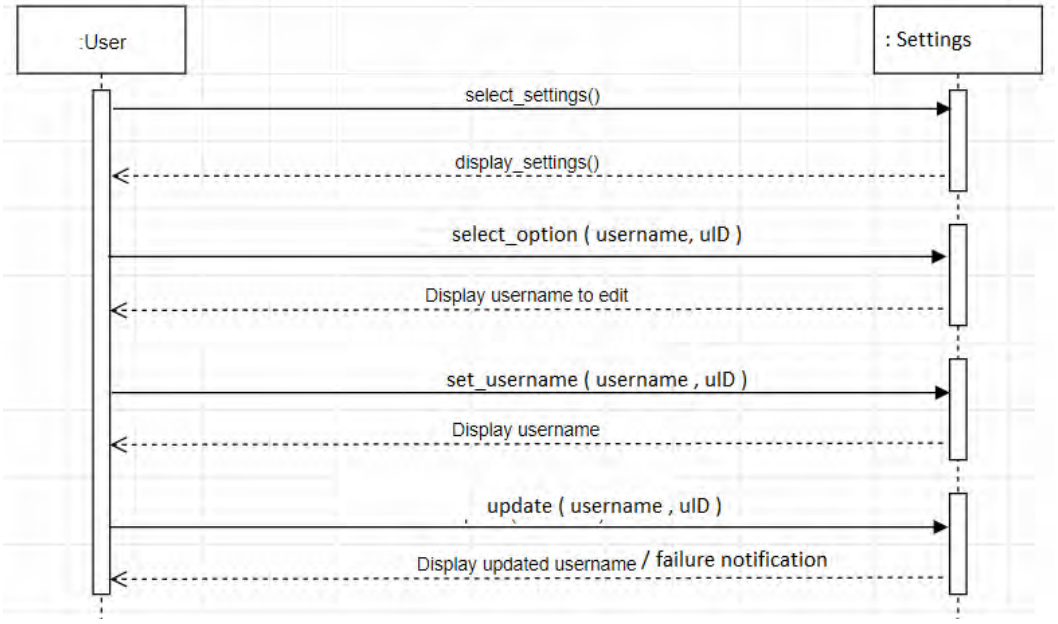

Figure 27 SSD of Set Username

SSD for Update User Status

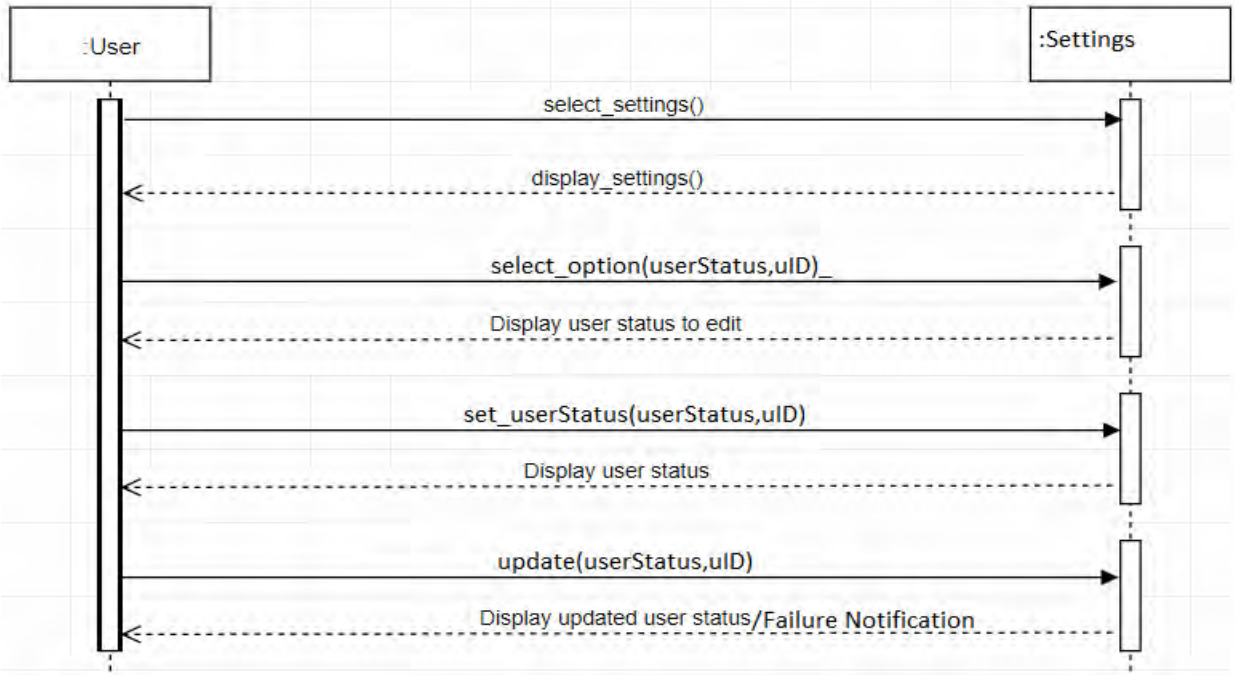

Figure 28 SSD for Update User Status

SSD for Load an Image

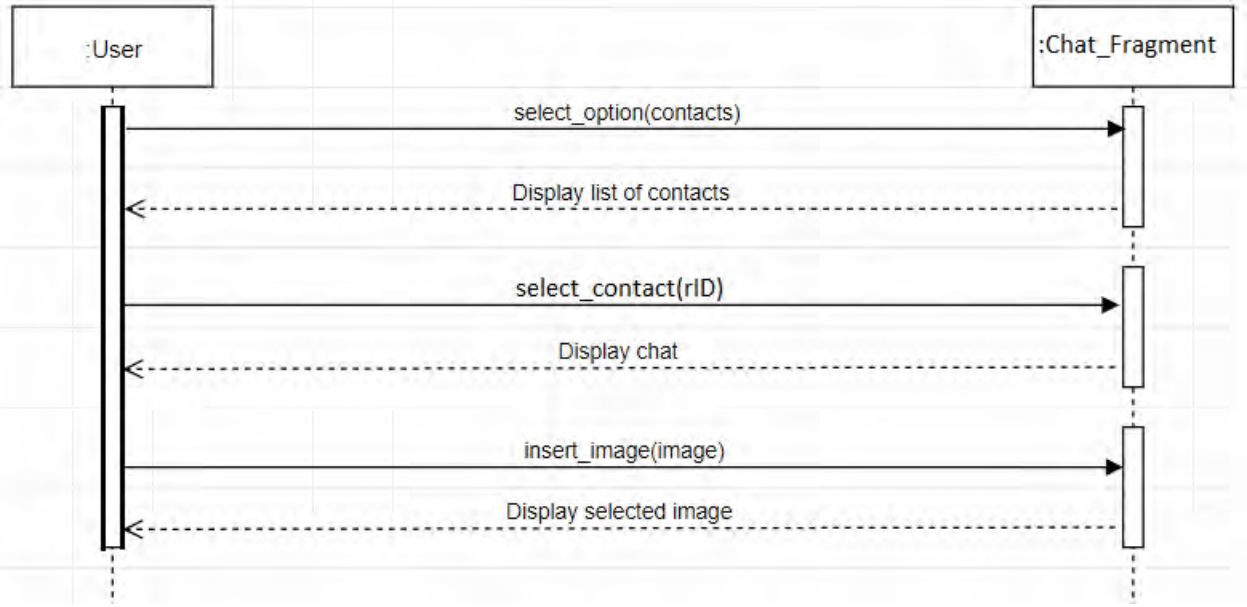

Figure 29 SSD for Load an Image

SSD for Load File

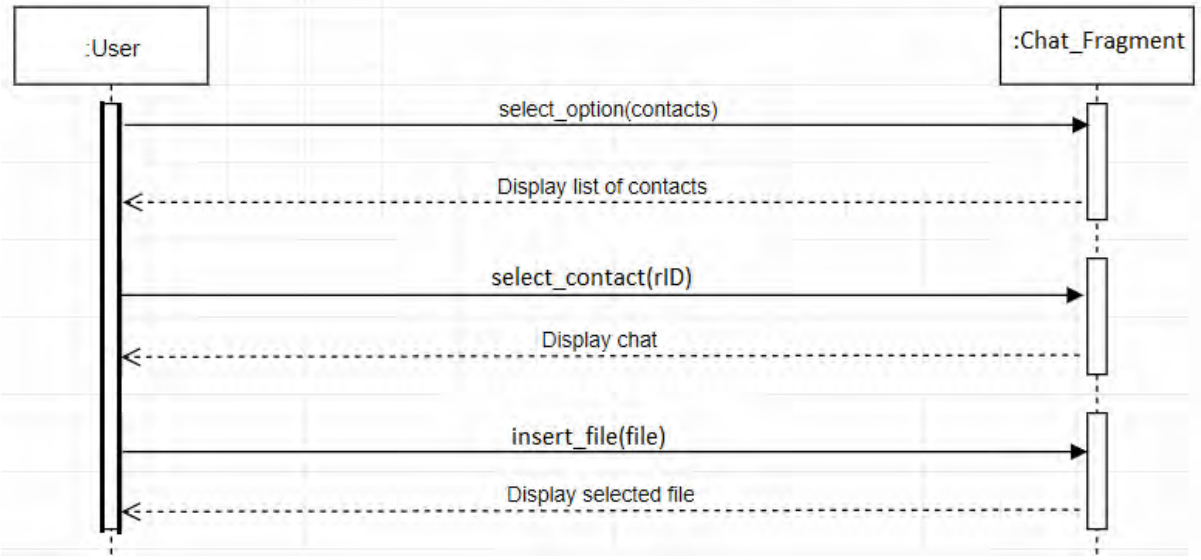

Figure 30 SSD for Load File

SSD for Send Message

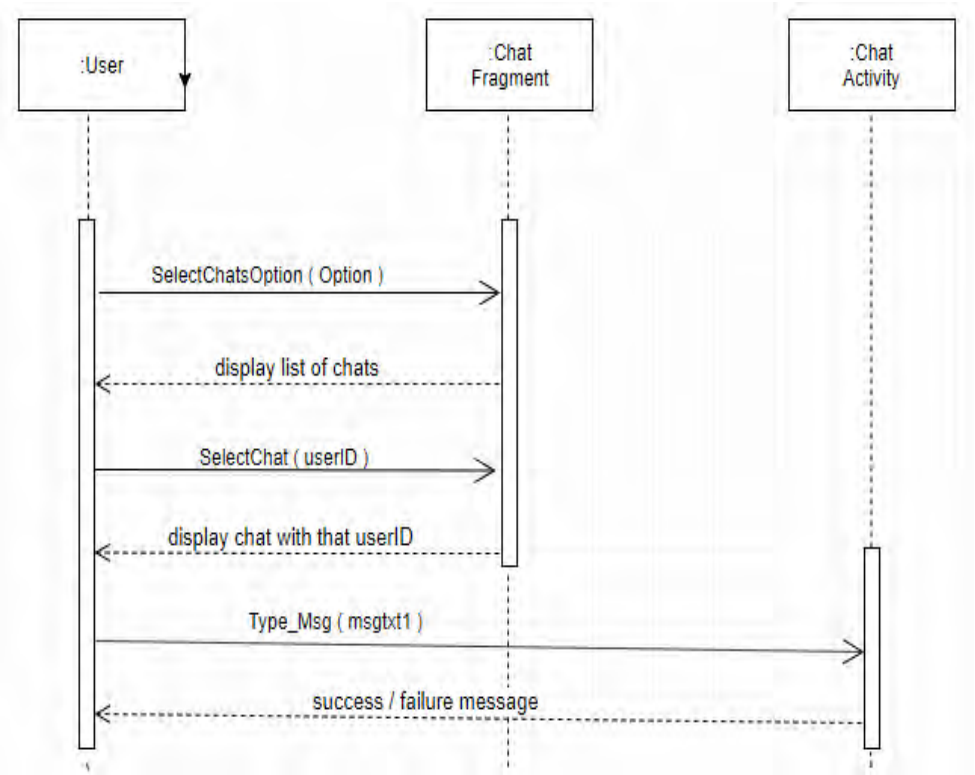

Figure 31 SSD for Send Message

SSD for Encryption

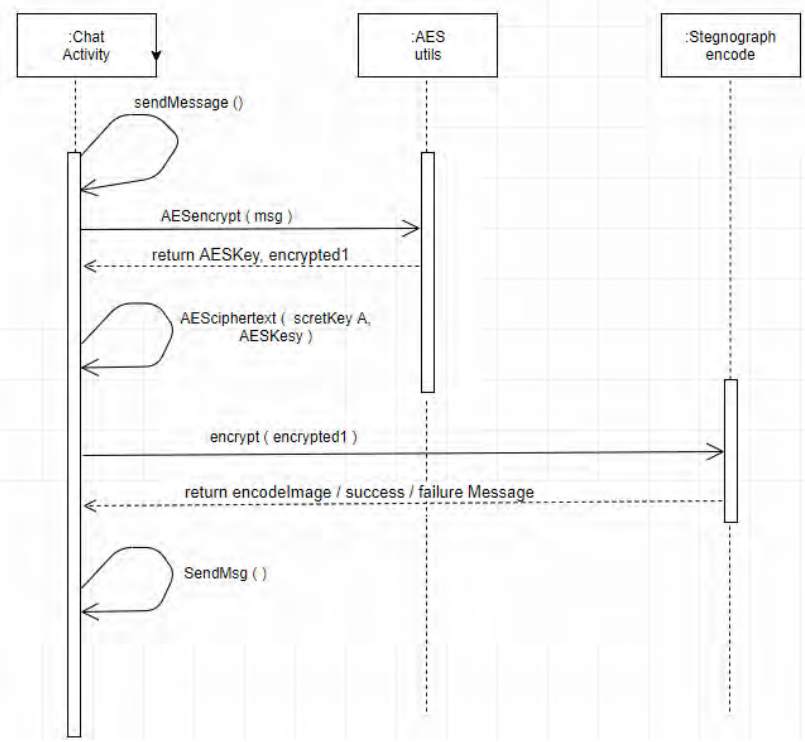

Figure 32 SSD for Encryption

SSD for Decryption

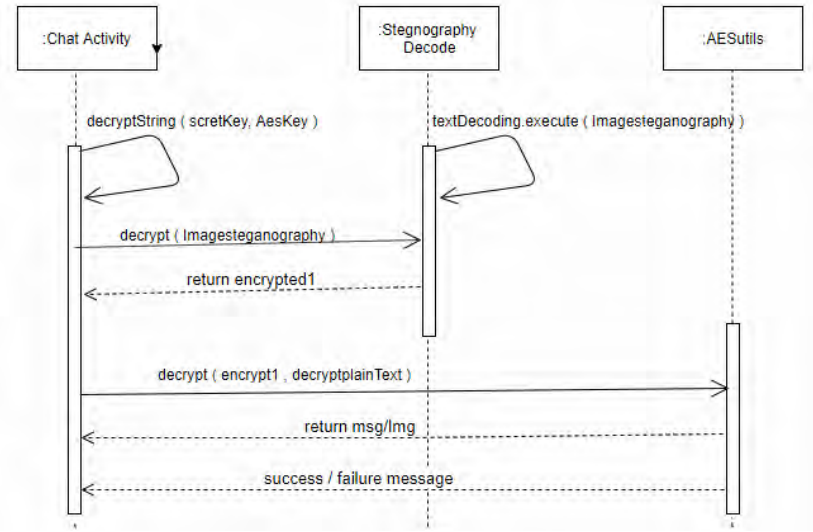

Figure 33 SSD for Decryption

SSD for Delete Chat

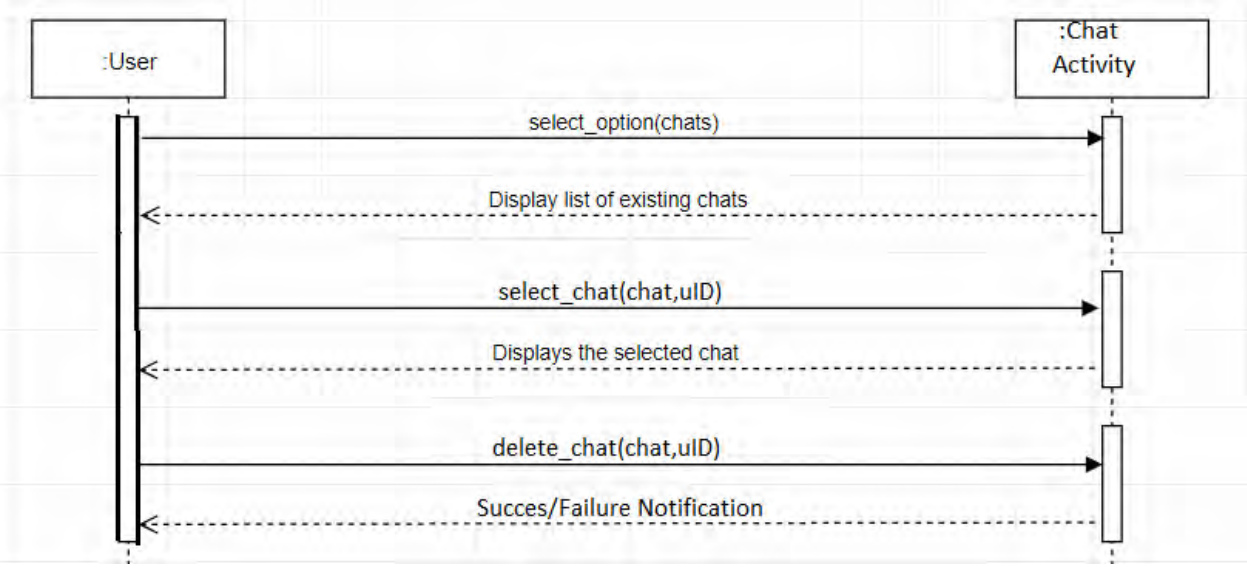

Figure 34 SSD for Delete Chat

SSD for Delete Message

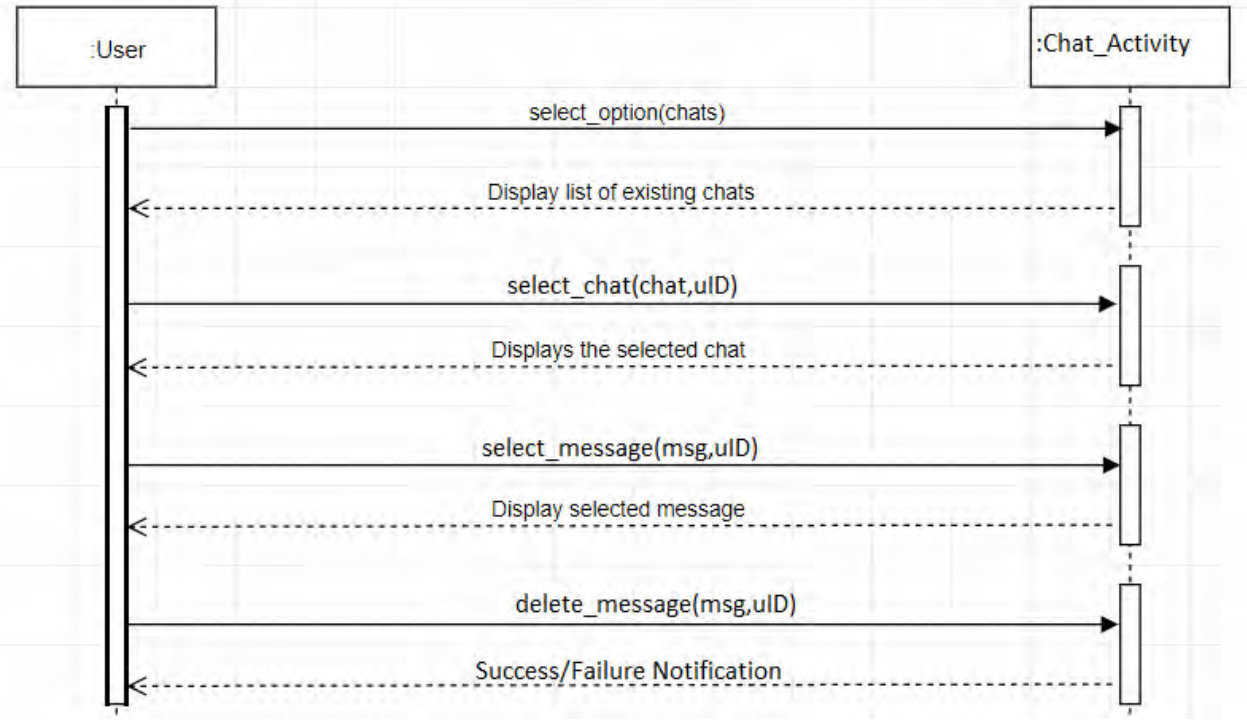

Figure 35 SSD of Delete Message

SSD for Delete Contact

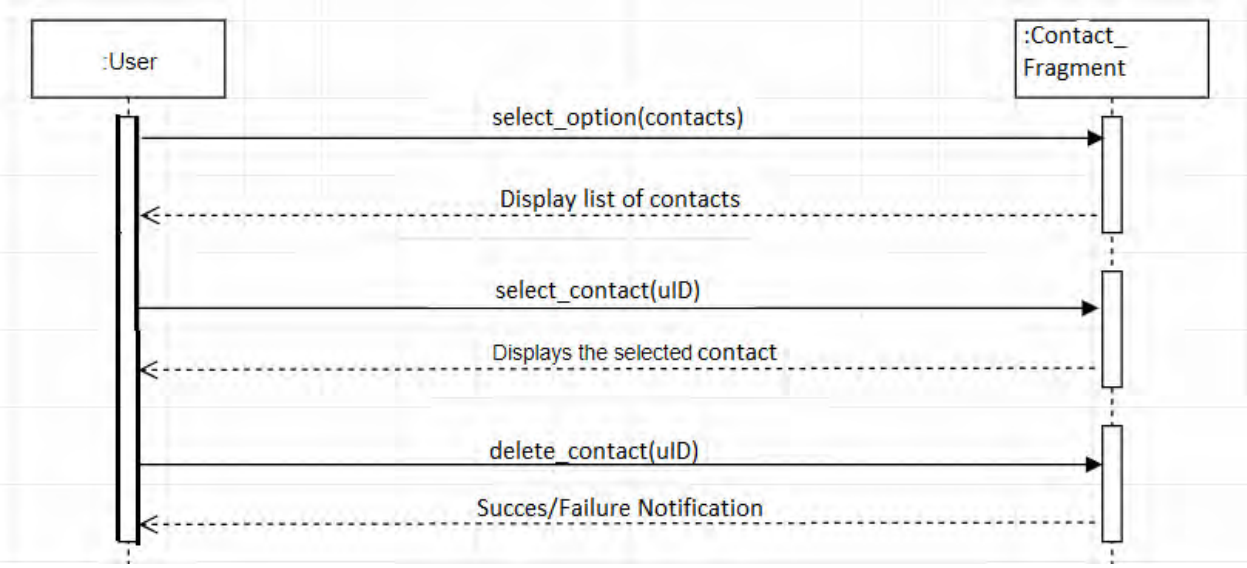

Figure 36 SSD for Delete Contact

SSD for Settings

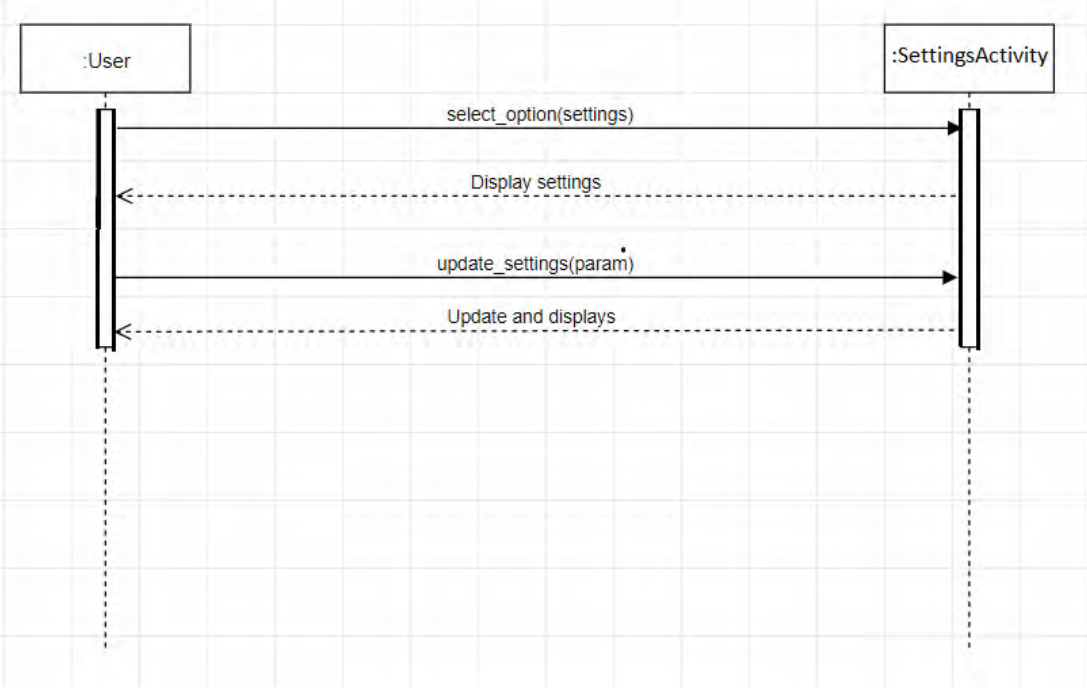

Figure 37 SSD for Settings

SSD for Search

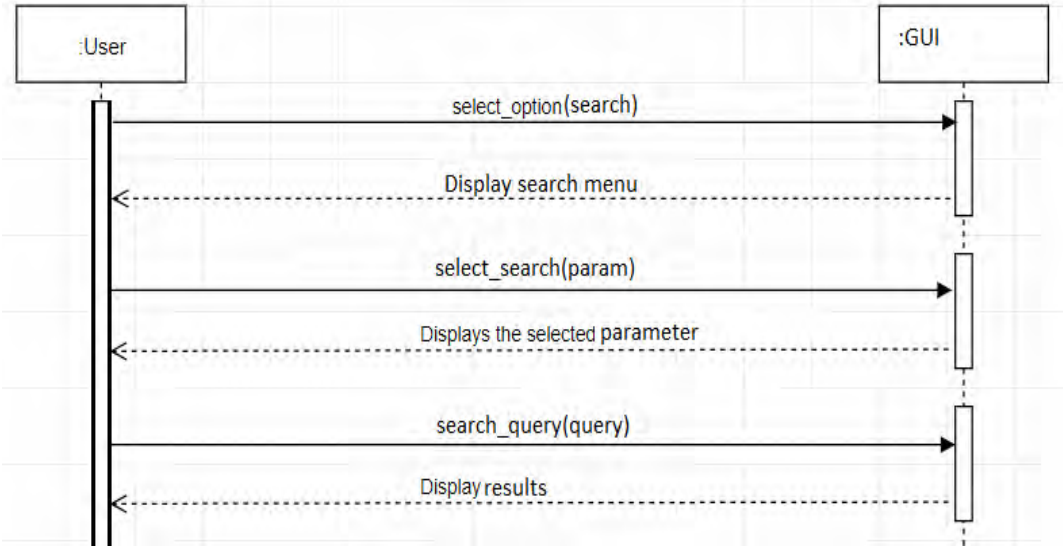

Figure 38 SSD for Search

SSD for Send Request

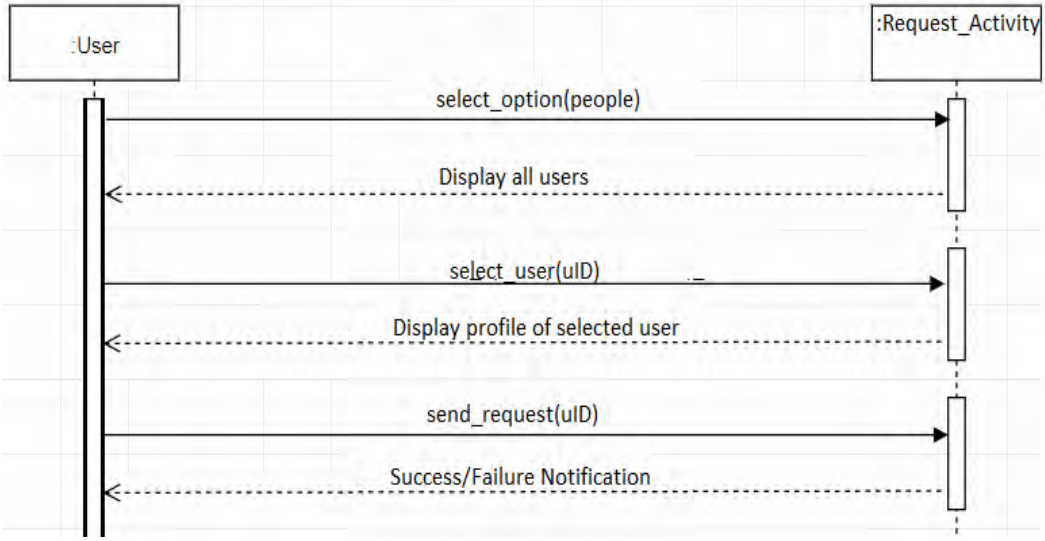

Figure 39 SSD for Send Request

SSD for Accept Request

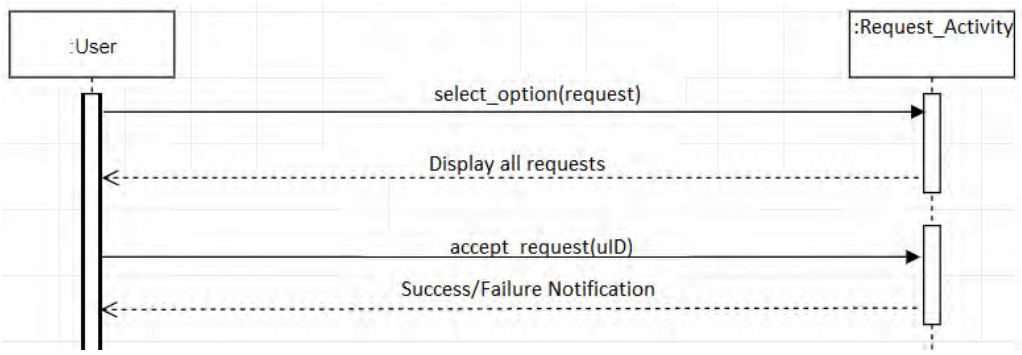

Figure 40 SSD for Accept Request

SSD for Delete Request

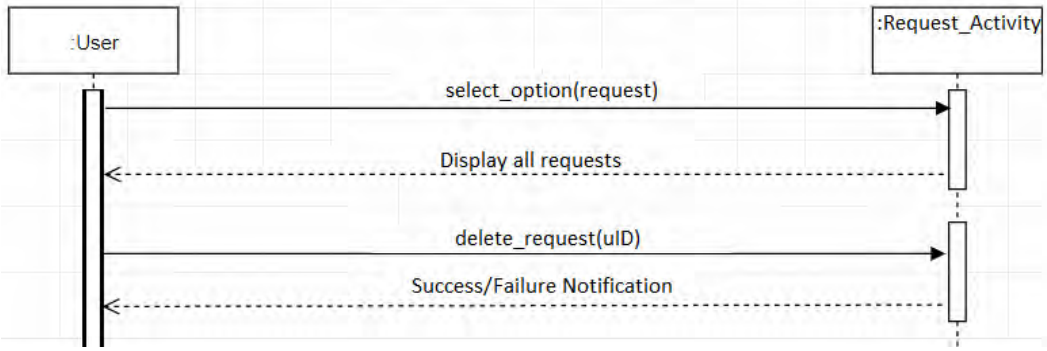

Figure 41 SSD for Delete Request

SSD for Register by Email

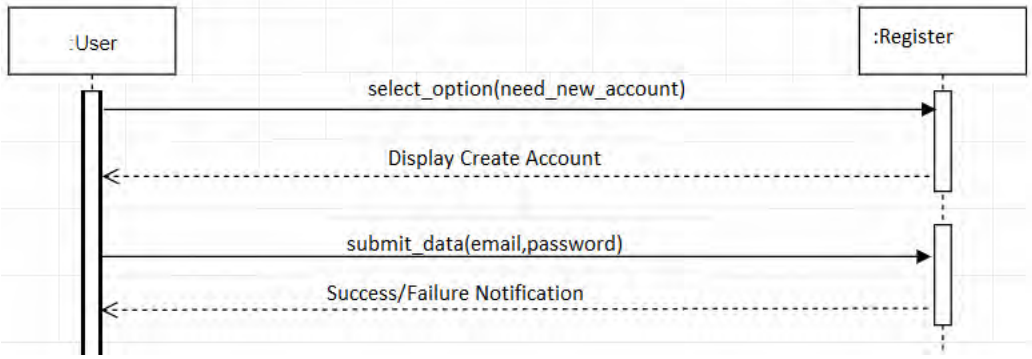

Figure 42 SSD for Register by Email

SSD for Register by Phone Number

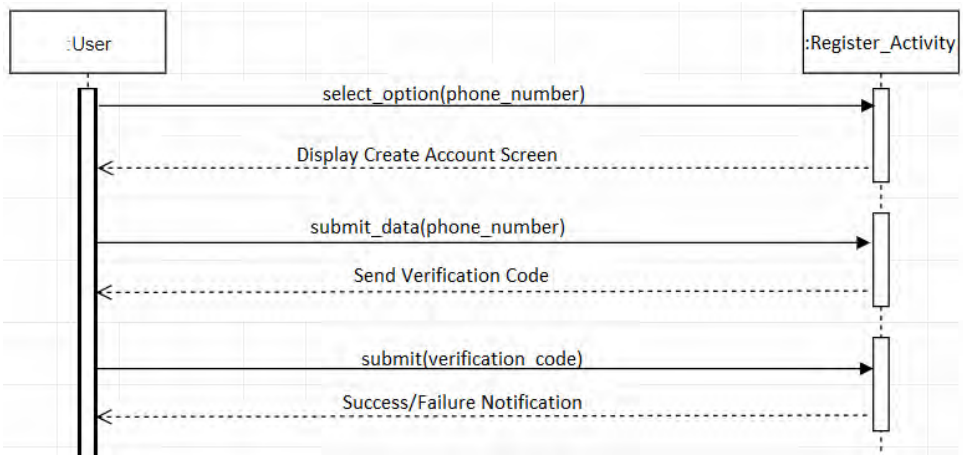

Figure 43 SSD for Register by Phone Number

4.9 CLASS DIAGRAM

A class diagram is an illustration of relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

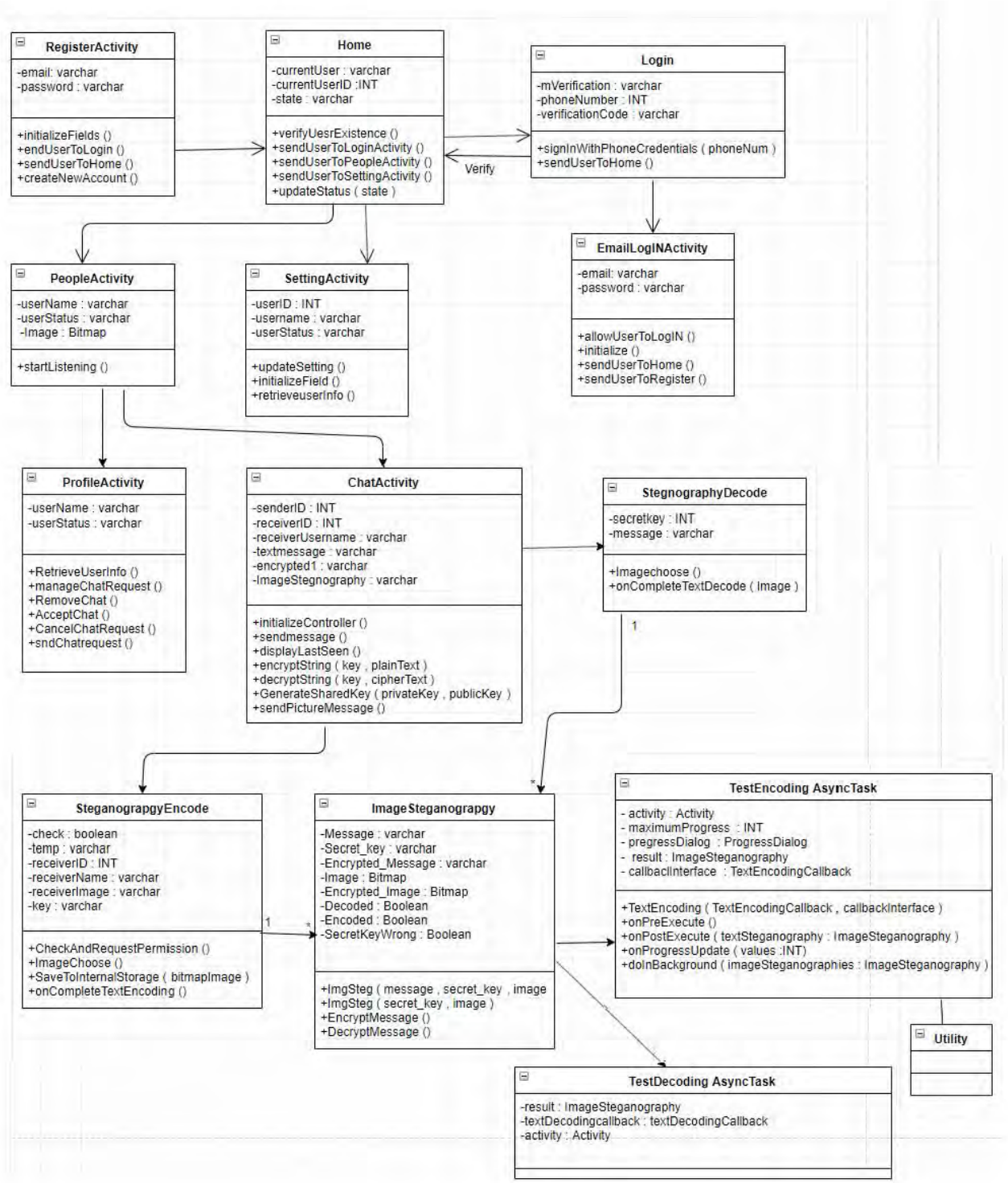

Figure 44 Class Diagram

4.10 ACTIVITY DIAGRAMS

Activity diagram is a UML behavior diagram which shows flow of control or object flow with emphasis on the sequence and conditions of the flow.

Activity Diagram for Non-Registered User

Activity Diagram for Register by Email

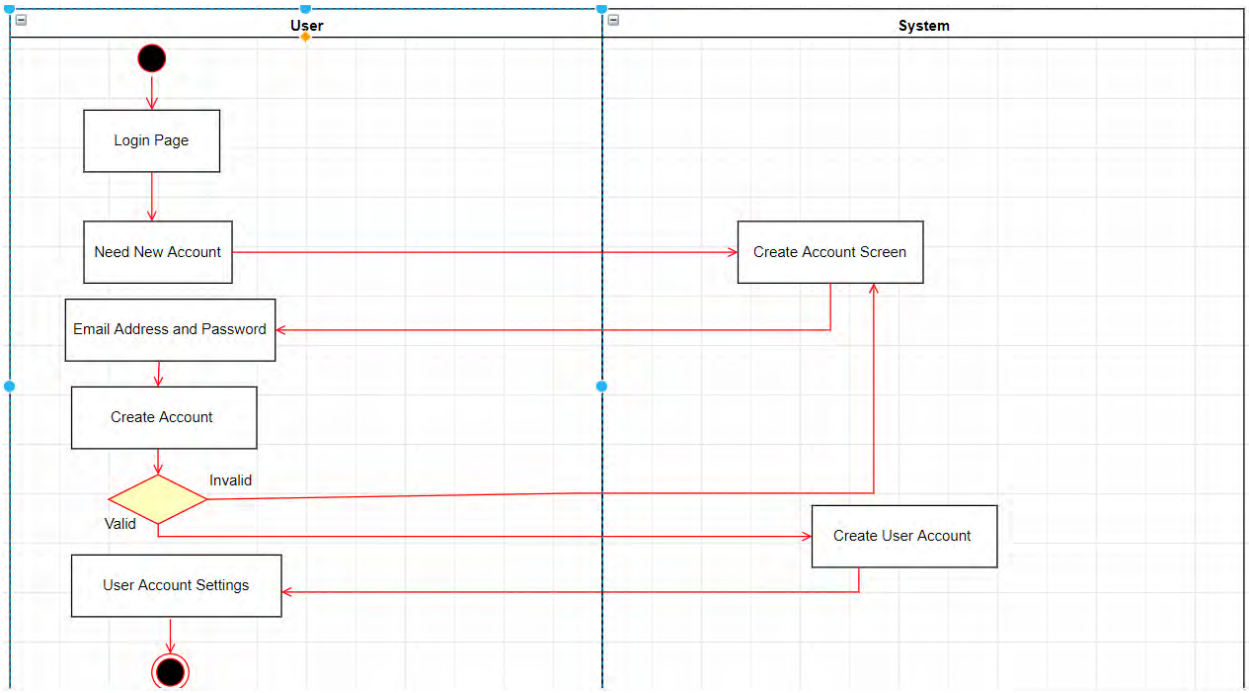

Figure 45 Activity Diagram for Register by Email

Activity Diagram for Register by Phone Number

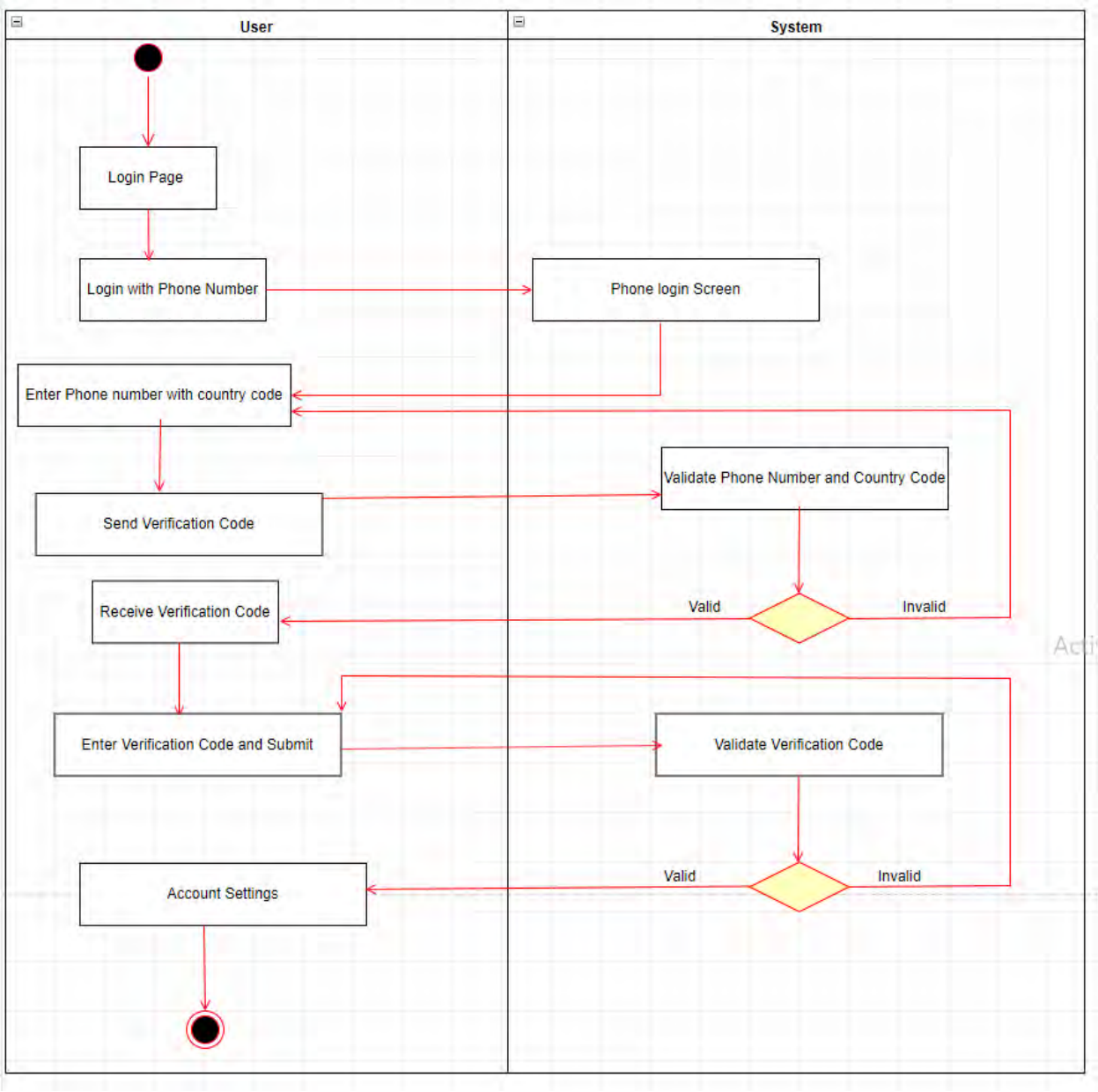

Figure 46 Activity Diagram for Register by Phone Number

Activity Diagrams for Registered User

Activity Diagram for Login

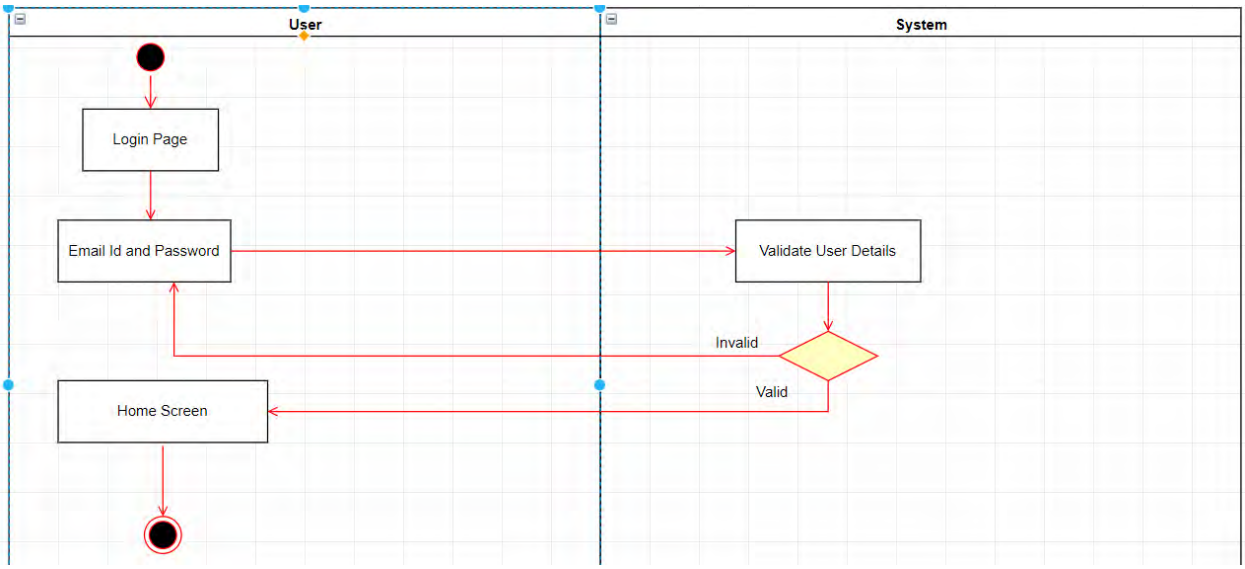

Figure 47 Activity Diagram for Login

Activity Diagram for Set Profile Picture

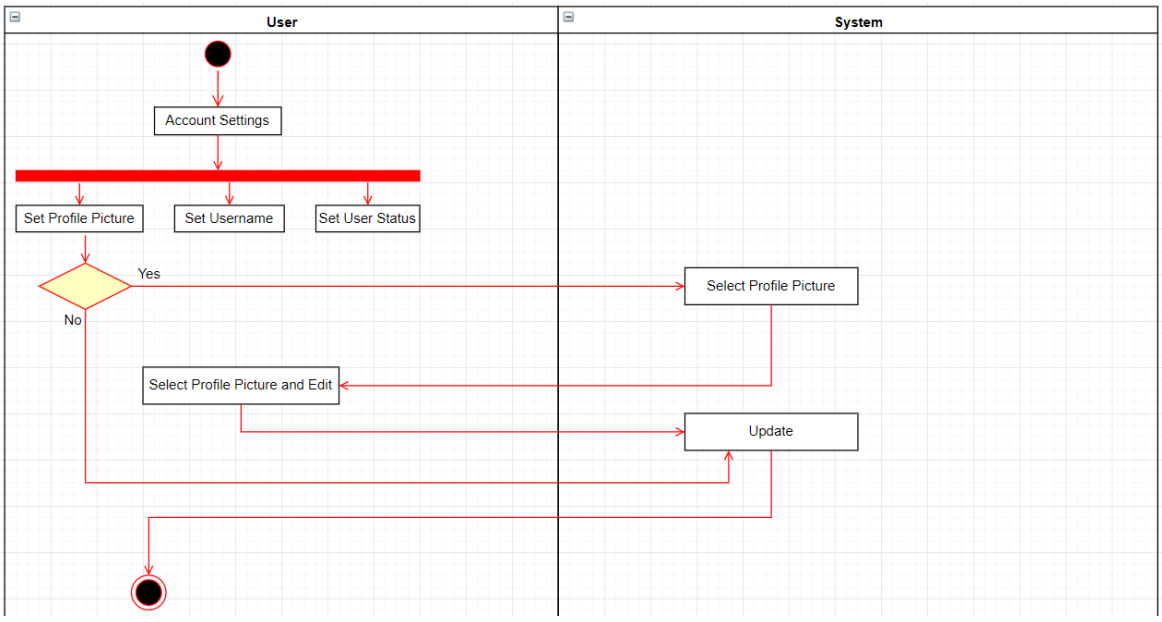

Figure 48 Activity Diagram for Set Profile Picture

Activity Diagram for Set Username

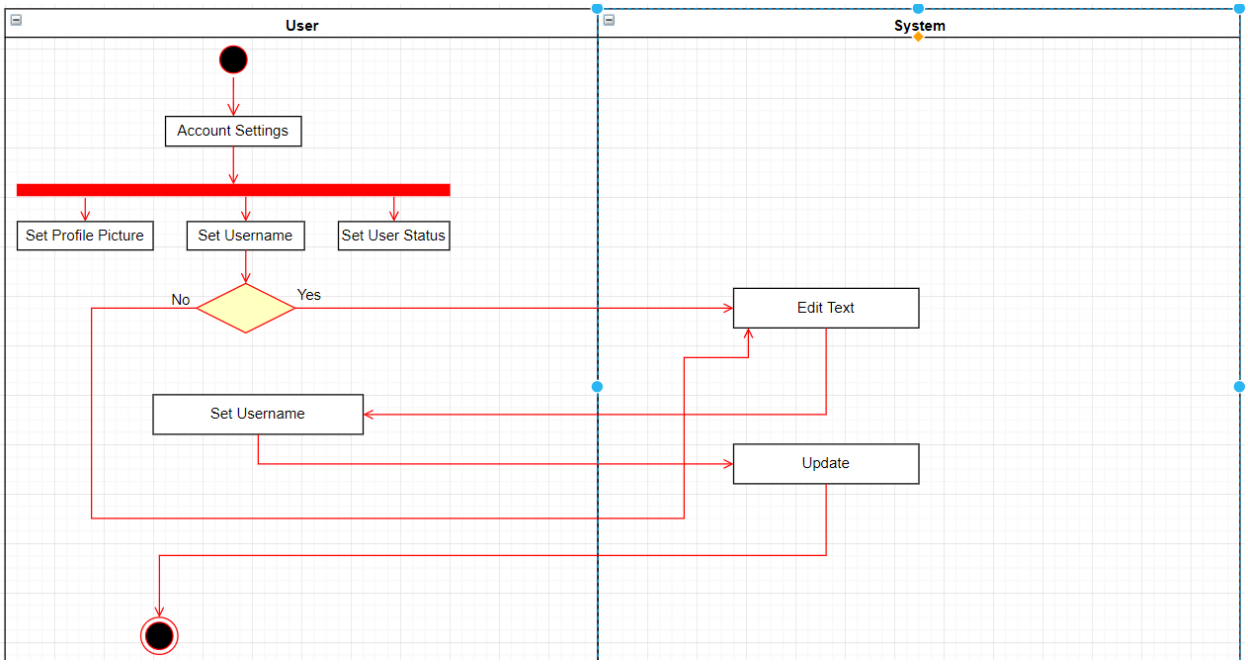

Figure 49 Activity Diagram for Set Username

Activity Diagram for Set User Status

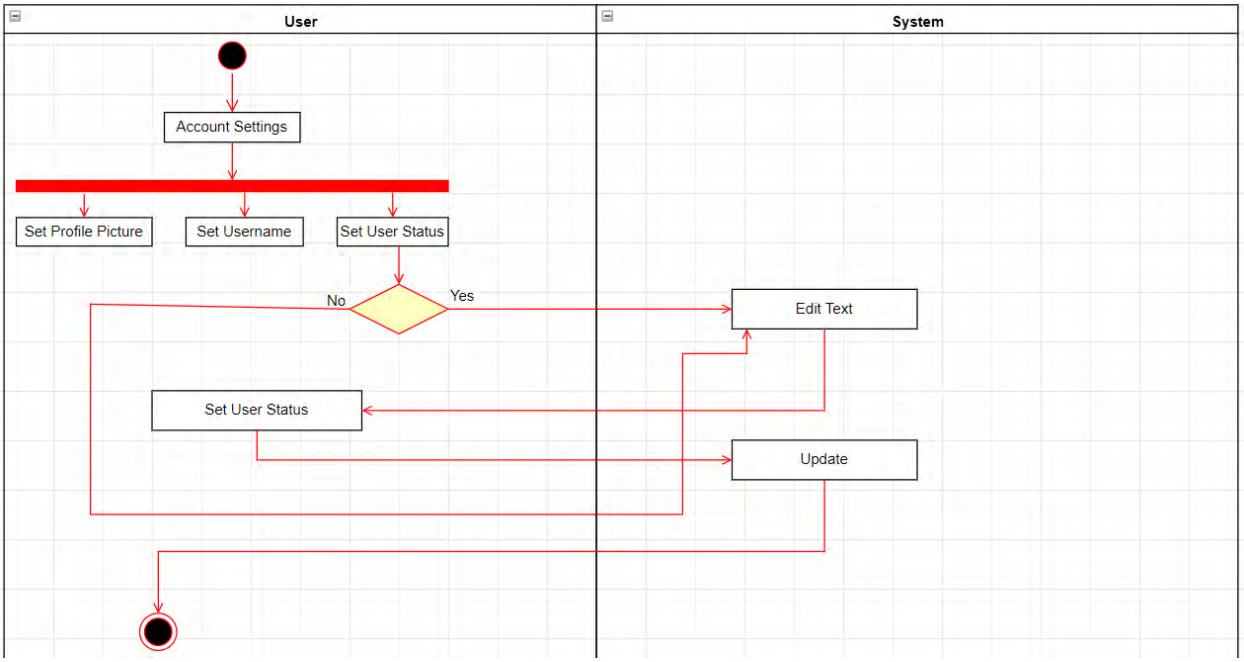

Figure 50 Activity Diagram for Update User Status

Activity Diagram for Load File

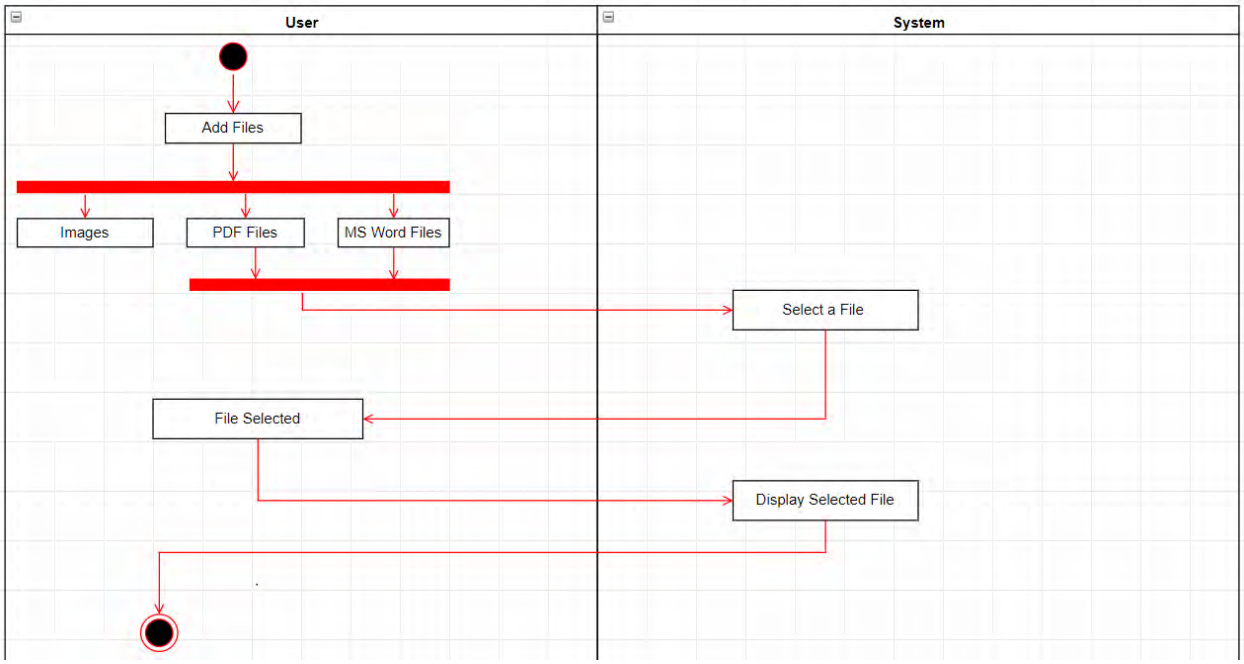

Figure 51 Activity Diagram for Load File

Activity Diagram for Load Image

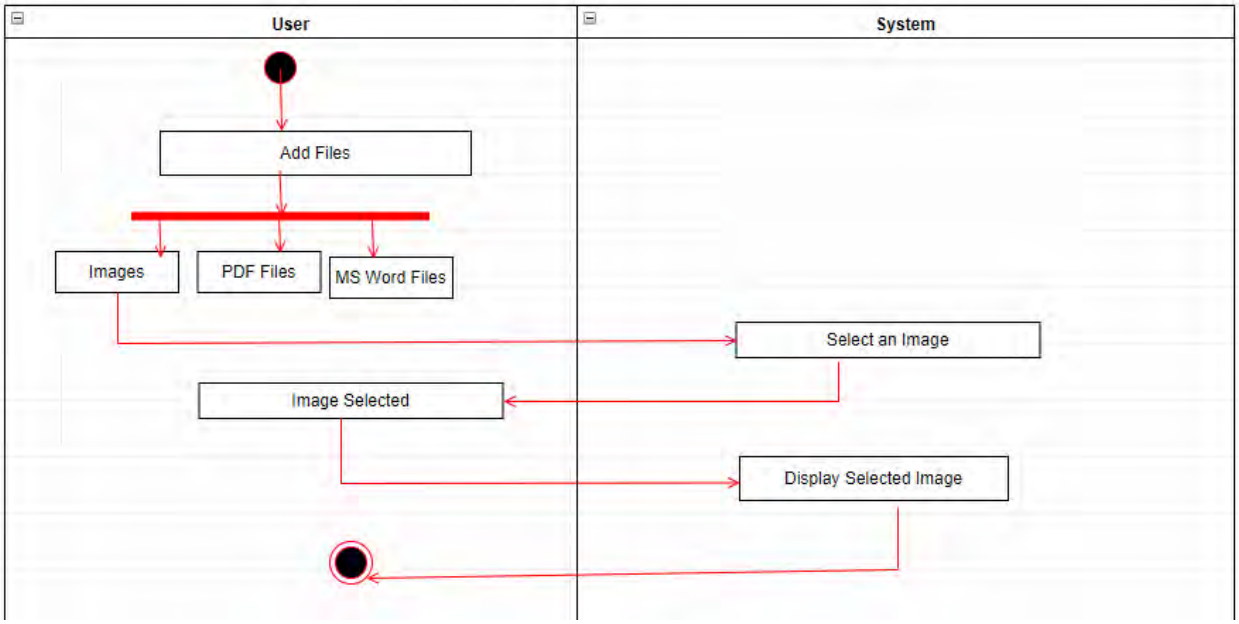

Figure 52 Activity Diagram for Load Image

Activity Diagram for Encryption

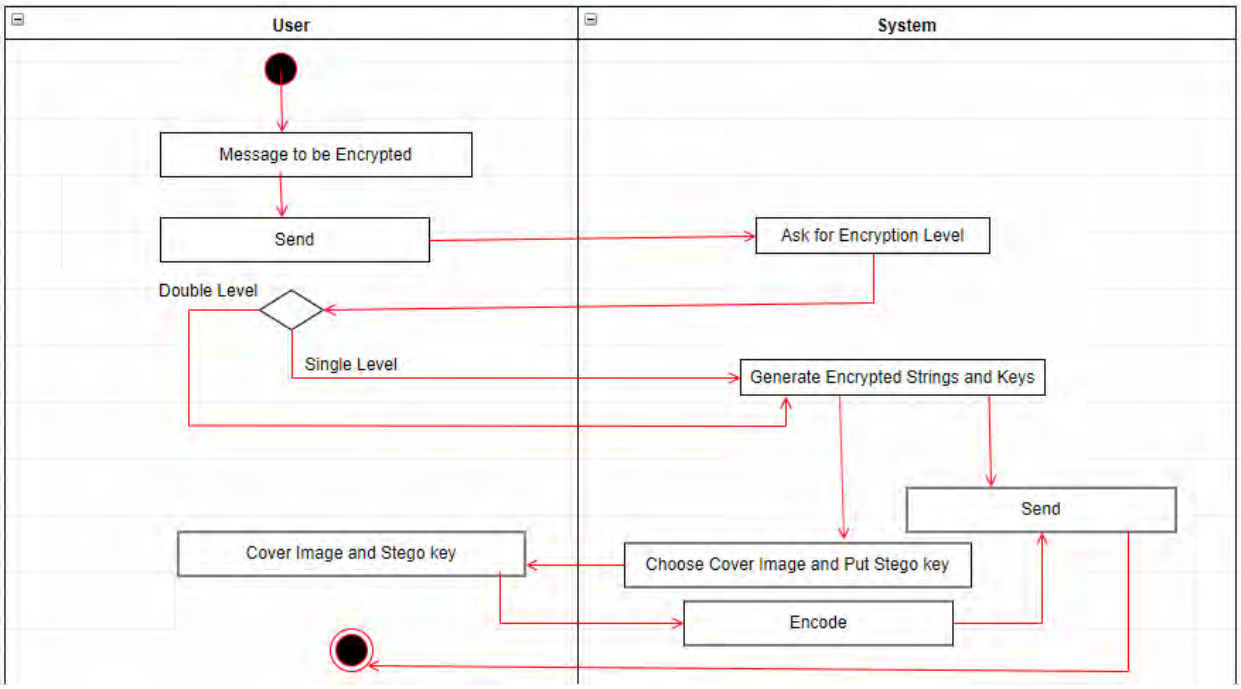

Figure 53 Activity Diagram for Encryption

Activity Diagram for Single Level Decryption

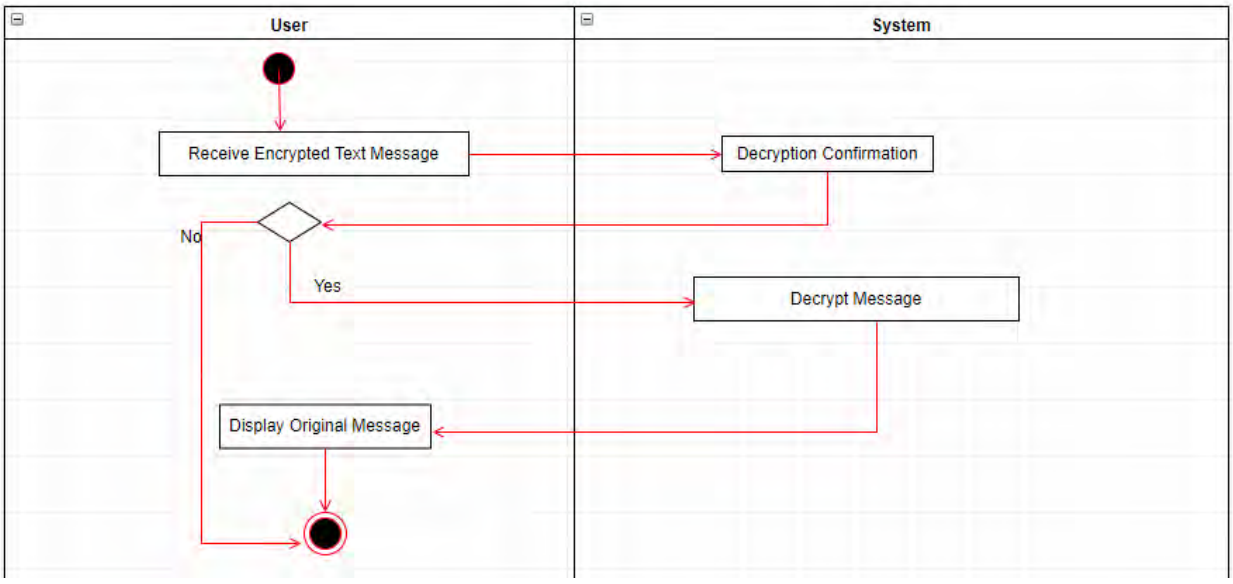

Figure 54 Activity Diagram for Single Level Decryption

Activity Diagram for Double Level Decryption

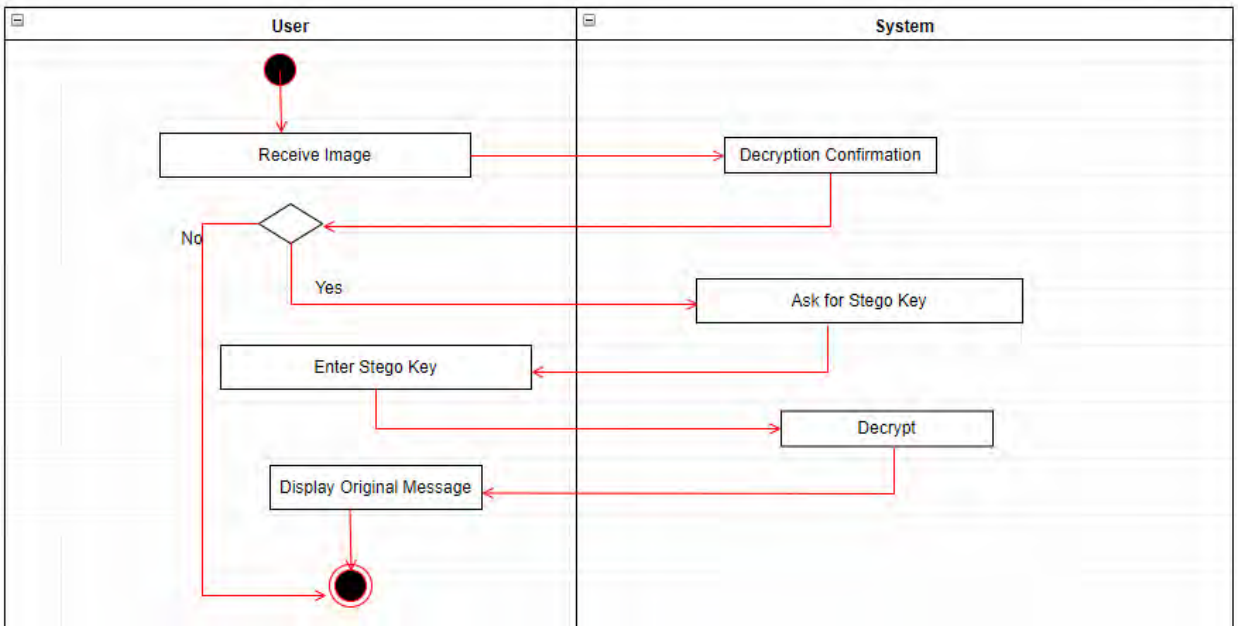

Figure 55 Activity Diagram for Double Level Decryption

Activity Diagram for Accept Request

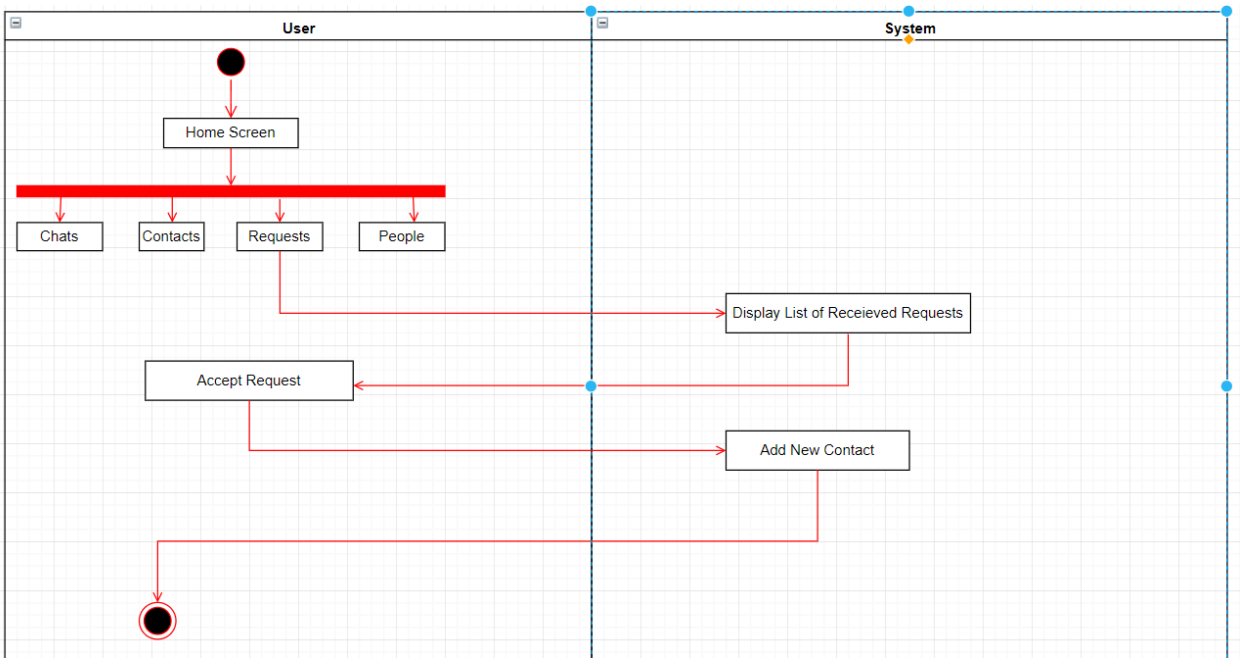

Figure 56 Activity Diagram for Accept Request

Activity Diagram for Send Request

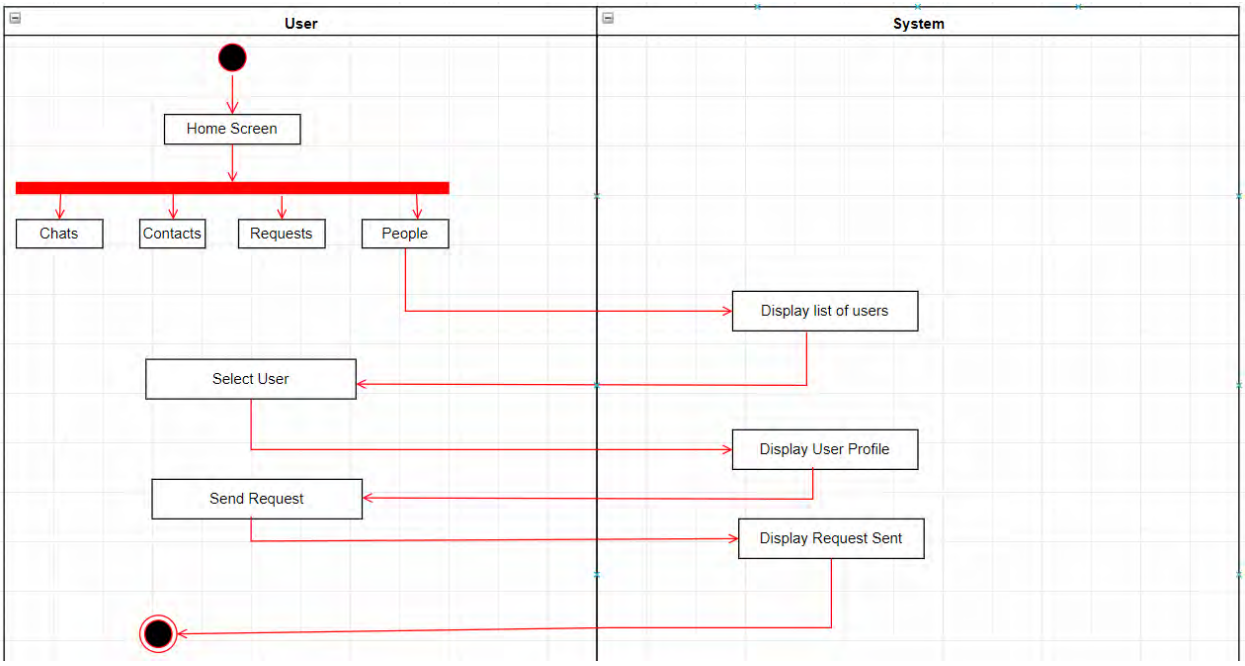

Figure 57 Activity Diagram for Send Request

Activity Diagram for Delete Request

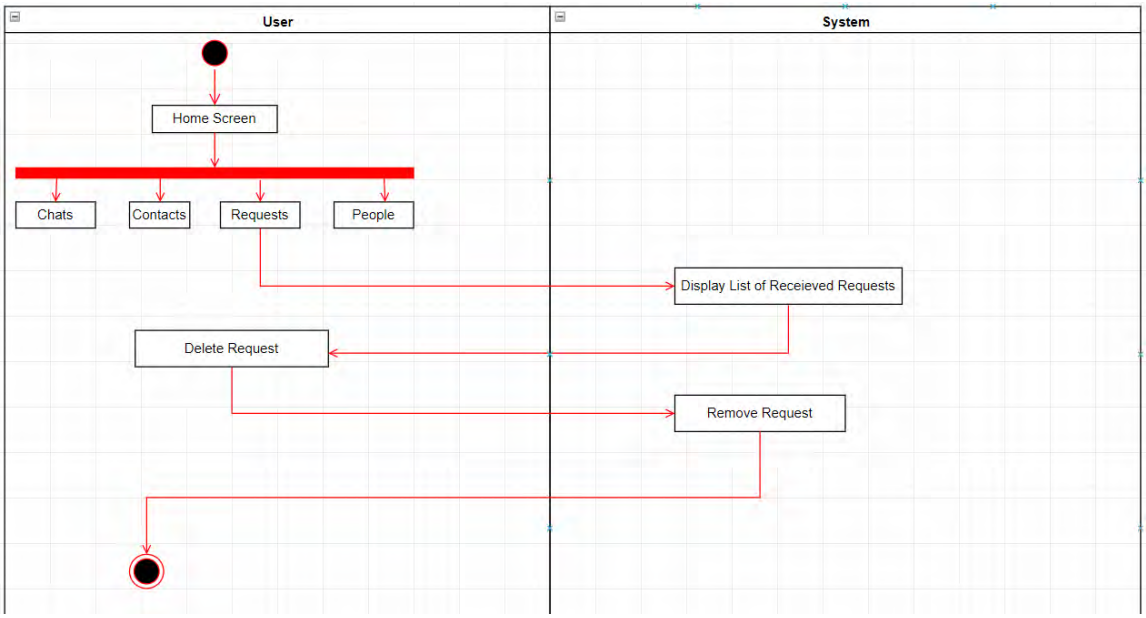

Figure 58 Activity Diagram for Delete Request

Activity Diagram for Delete Chat

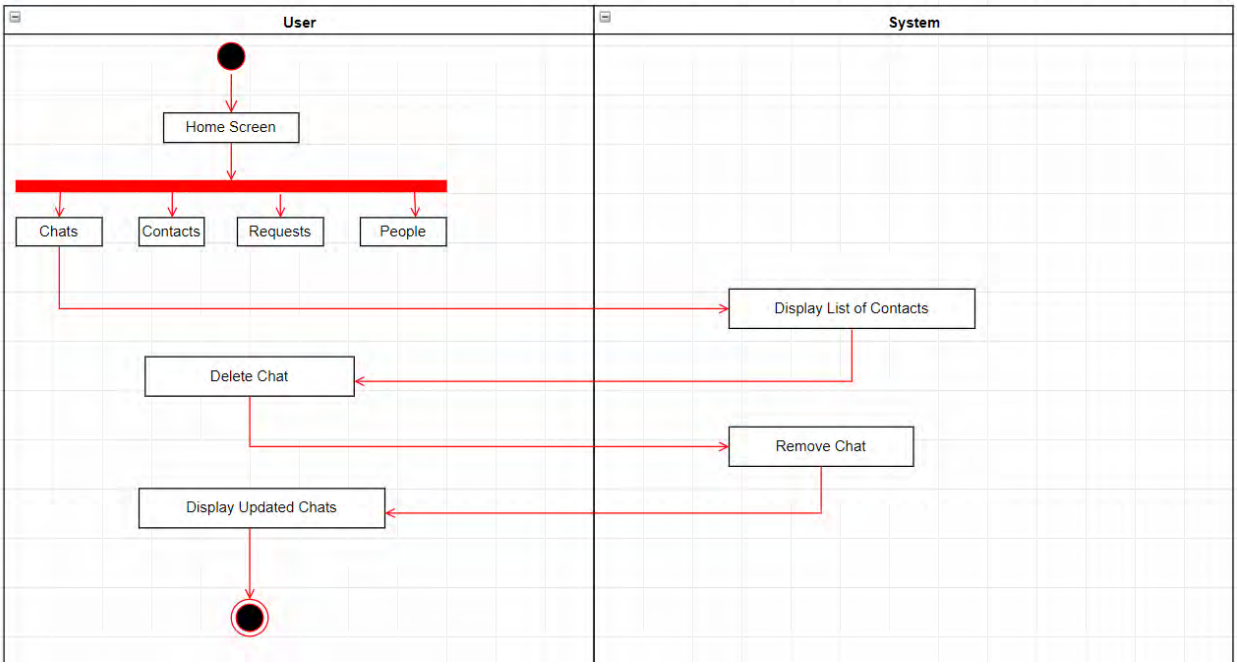

Figure 59 Activity Diagram for Delete Chat

Activity Diagram for Delete Contact

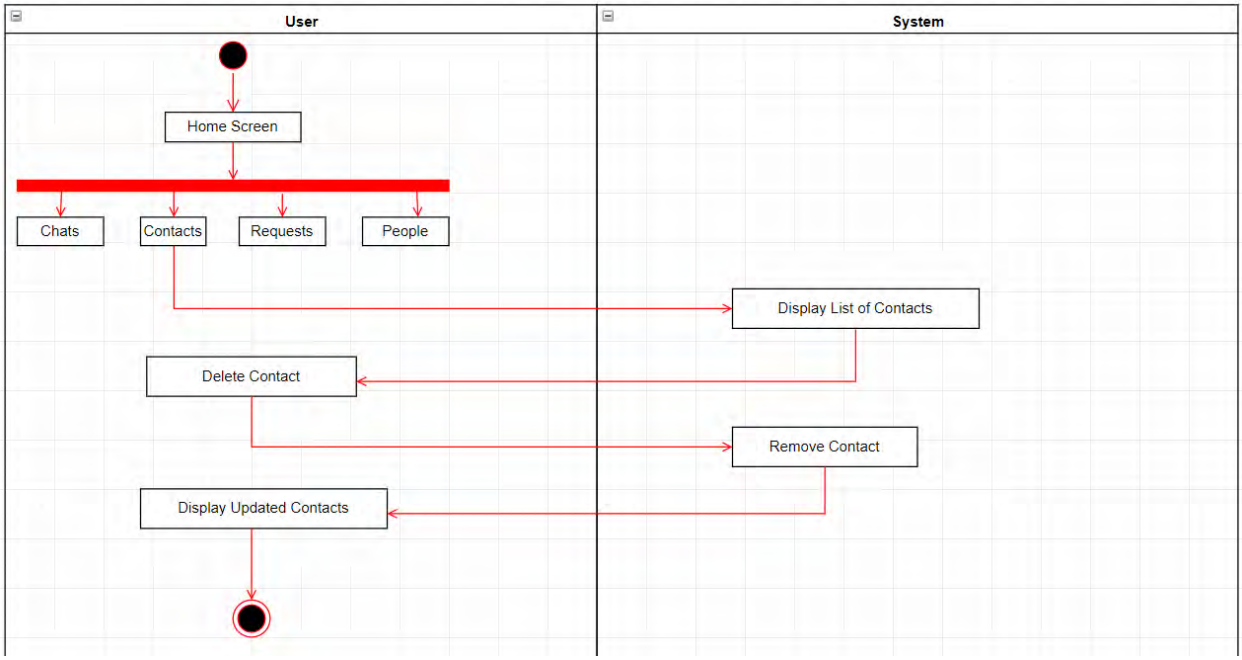

Figure 60 Activity Diagram for Delete Contact

Activity Diagram for Delete Message

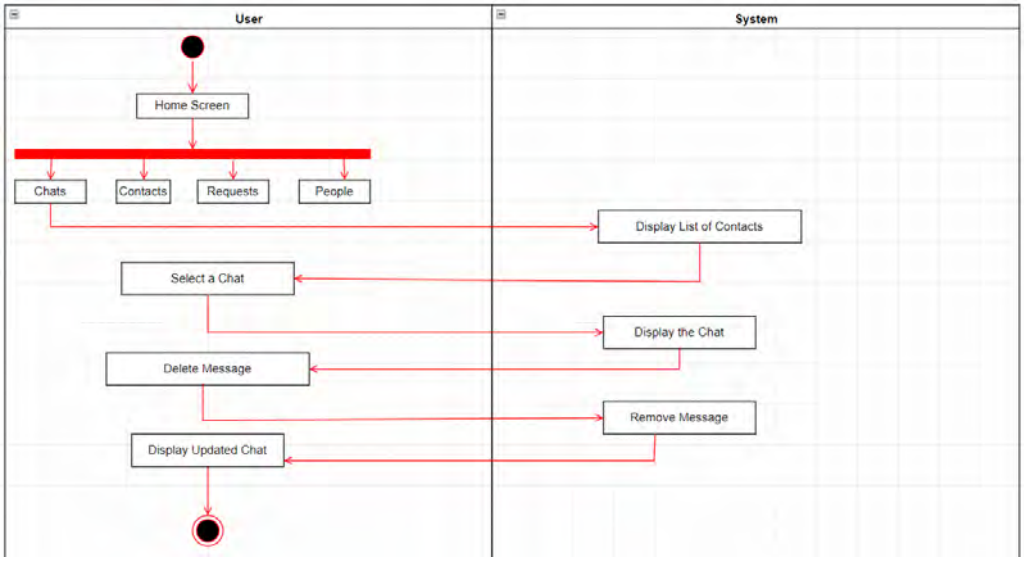

Figure 61 Activity Diagram for Delete Message

Activity Diagram for Search Users

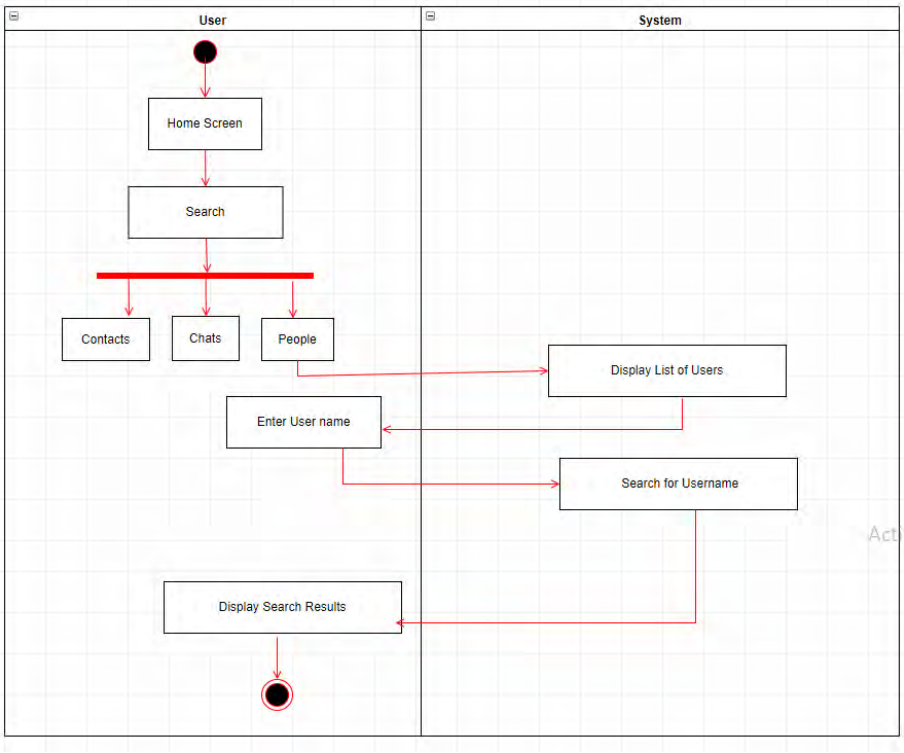

Figure 62 Activity Diagram for Search Users

Activity Diagram for Search Chats

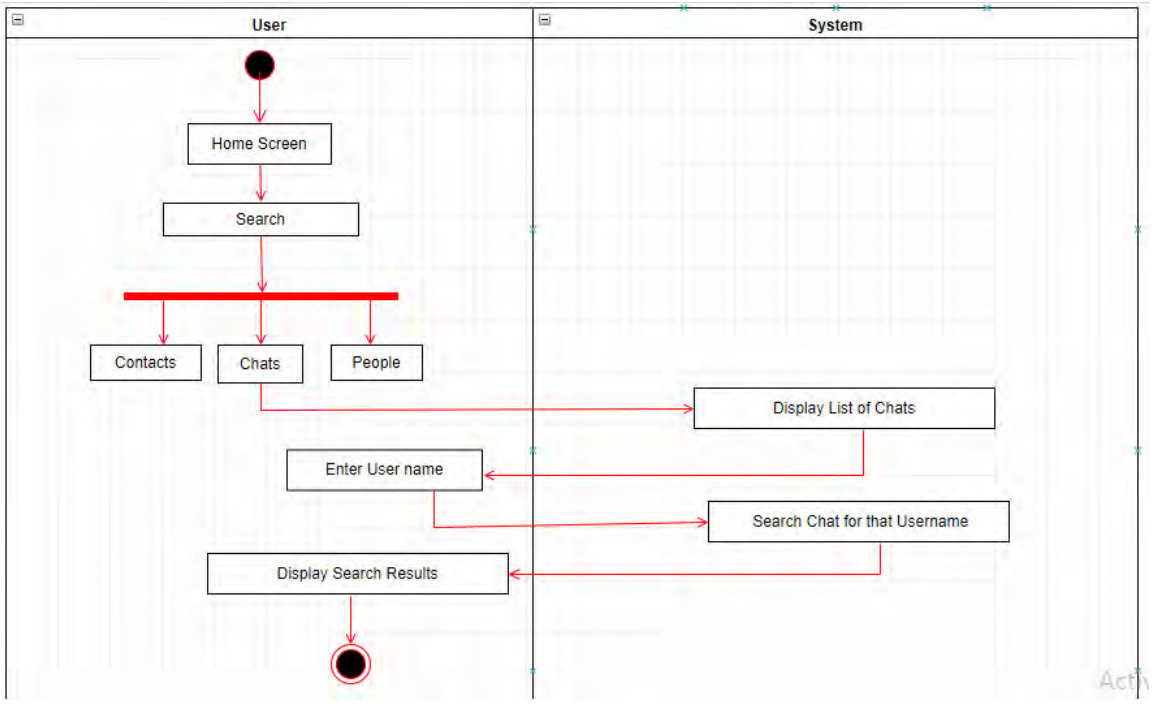

Figure 63 Activity Diagram for Search Chats

Activity Diagram for Search Contacts

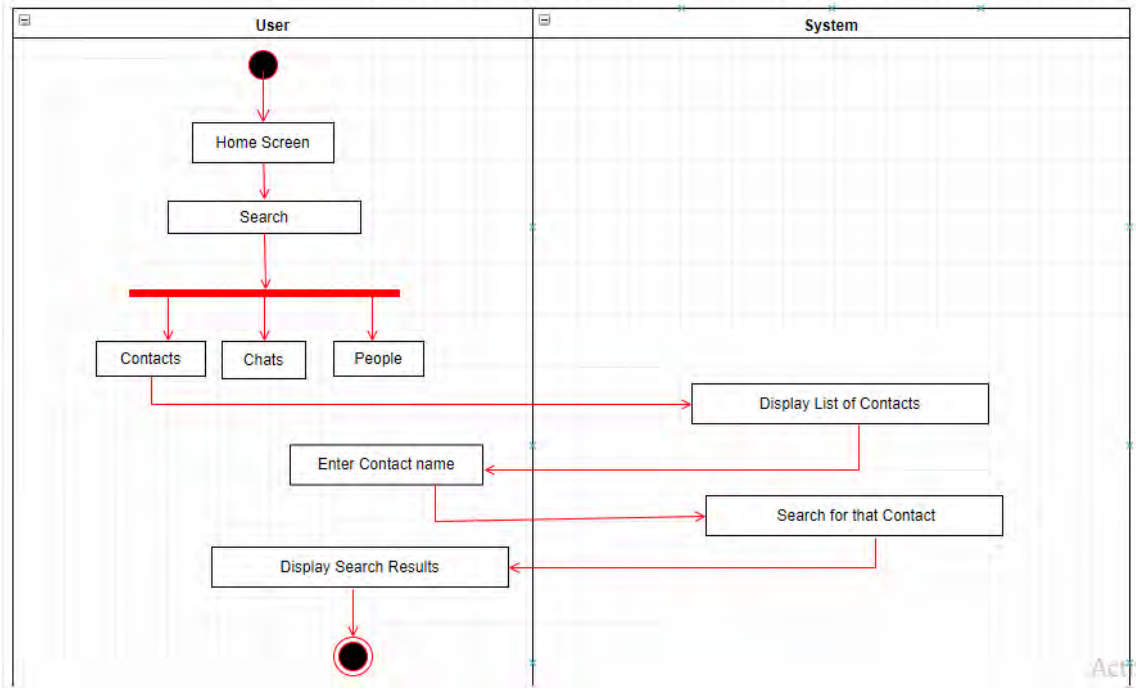

Figure 64 Activity Diagram for Search Contacts

CHAPTER 5

SOFTWARE TEST DOCUMENTATION

5.1 INTRODUCTION

According to ANSI/IEEE 1059 Standard, Testing can be defined as “a process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item”. It is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating or verifying that software meets the business and technical requirements that guided its design and development.

5.2 SYSTEM OVERVIEW

The project is an android application which is intended to provide users secure transmission and storage of images, text messages and files using the combination of Hybrid Cryptography (Asymmetric Cryptography and Symmetric Cryptography) and Steganography algorithms, Elliptic Curve Cryptography, Advance Encryption Standard and Least Significant Bit (LSB) respectively. The application will provide the following functionality:

- Non-Registered users can create their account to get registered.
- One time login to application using user’s phone number and also login through email.
- Registered User can set username and set/update user status and profile picture.
- Encrypting images, text messages or files through hybrid cryptography.
- Hiding the encrypted images, text messages or files in another cover image through image steganography.
- Registered User can send/receive the encrypted images, text messages or files.
- Decoding: retrieving encrypted image, text message or file from cover image and then decrypting it.
- Auto saving of encrypted/decrypted data (images/text files).
- Registered users can send or accept requests to or from other registered users.
- Registered users can search people (other users of the application) to send a request, their contacts to start a chat with and their chats to send a message.

- Registered users can delete a request, contact, chat or a particular message.

5.3 TEST APPROACH

The test approach defines how testing will be carried out. The following testing approach will be used:

- **Acceptance Testing**

  User Acceptance Testing (UAT) consists of a process of verifying that a solution works for the user. It determines if the requirements of a specification are met. Software vendors often refer to this as "Beta Testing". It is a Black Box testing technique. The main purpose of this testing is to evaluate the system's compliance with the business requirements and verify if it has met the required criteria for delivery to end users. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.

- **Firebase Test Lab**

  Firebase Test Lab allows to test an android app on real and virtual devices. It helps improve the quality of user's experience. It is great for small apps with bunch of functionality and for larger apps Test Lab offers tons of customization. It can test any app whether it uses Firebase or not. It takes the APK file of the app. It performs a free automated test of a special sort called "robo test" in which Test Lab crawl the application in an intelligent way. It does not require any code or configuration upfront.

  Type of APK Files:

  - ❖ Debug APKs as they don't need to be formally signed. It is the easiest way to get started with Test Lab.
  - ❖ Incremental builds with instant run does not work because everything needed is not in a single APK.

5.4 TEST PLAN

A test plan outlines the strategy that will be used to test an application, the resources that will be used, and the test environment in which testing will be performed and the time which will be spent on testing.

Features to be tested

The features to be tested are from the user's perspective:

- Login
- Update Profile Picture
- Set Username
- Update Status
- Load an Image
- Load File
- Send Message
- Encrypt Text Message
- Encrypt Text Message Twice
- Encrypt Image
- Encrypt Image Twice
- Encrypt File
- Encrypt File Twice
- Decrypt Text Message
- Decrypt Text Message Twice
- Decrypt Image
- Decrypt Image Twice
- Decrypt File
- Decrypt File Twice
- Delete Chat
- Delete Message
- Change Settings

- Delete Contact
- Delete Request
- Send Request
- Accept Request
- Search
- Register by Email
- Register by Phone Number

Features not to be tested

Features not to be tested are from the developer's perspective:

- How much power is used by the processor?
- How much memory is consumed by the system?
- Maintainability of the system.
- Software risk factors like system vulnerabilities, efficiency weakness etc.

Testing Tools and Environment

A Testing Environment is a setup of software and hardware for the testing teams to execute test cases. As it is beta testing so following testing tool or testing environment is required:

- Android device (cell phone or tablet) with minimum API level 28.
- Internet connection

5.6 TEST CASES

The test cases for the Fusion system are:

Test Case 1

Table 39 TC-1

ID	TC-1
Purpose	Check that the registered user can successfully login the account.
Setup	1. Application is installed successfully. 2. User is registered.
Instructions	1. Open the application. 2. Enter email and password.
Expected Results	User is logged in.
Actual Result	User is logged in.
Verdict	Pass

Test Case 2

Table 40 TC-2

ID	TC-2
Purpose	Check that a non-registered user can successfully register by using phone number.
Setup	1. Open the application. 2. Go to phone number registration.
Instructions	1. Enter the phone number with correct country code. 2. Enter the 6 digit verification code.
Expected Results	User has been registered.
Actual Result	User has been registered.
Verdict	Pass

Test Case 3

Table 41 TC-3

ID	TC-3
Purpose	Check that a non-registered user can successfully register by using email and password.
Setup	3. Open the application. 4. Go to need new account.
Instructions	3. Enter the valid email address. 4. Enter the valid password.
Expected Results	User has been registered.
Actual Result	User has been registered.
Verdict	Pass

Test Case 4

Table 42 TC-4

ID	TC-4
Purpose	Check that a registered user can successfully update a profile picture.
Setup	1. Log in to application. 2. Go to settings.
Instructions	1. Select profile picture. 2. Select source to choose picture from. 3. Select a picture. 4. Edit the picture that is rotate, flip, crop etc. 5. Update the picture.
Expected Results	Profile picture has been updated.
Actual Result	Profile picture has been updated.
Verdict	Pass

Test Case 5

Table 43 TC-5

ID	TC-5
Purpose	Check that a registered user can successfully set a username.
Setup	1. Log in to application. 2. Go to Settings.
Instructions	1. Type username. 2. Update the username.
Expected Results	Username has been set.
Actual Result	Username has been set.
Verdict	Pass

Test Case 6

Table 44 TC-6

ID	TC-6
Purpose	Check that a registered user can successfully update a status.
Setup	1. Log in to application. 2. Go to Settings.
Instructions	1. Type user status like software developer. 2. Update user status.
Expected Results	User status has been updated and displayed.
Actual Result	User status has been updated and displayed.
Verdict	Pass

Test Case 7

Table 45 TC-7

ID	TC-7
Purpose	Check that a registered user can successfully insert an image.
Setup	1. Log in to application. 2. Select the particular contact to send the image to.
Instructions	1. Select the image.
Expected Results	Image has been loaded successfully.
Actual Result	Image has been loaded successfully.
Verdict	Pass

Test Case 8

Table 46 TC-8

ID	TC-8
Purpose	Check that a registered user can successfully insert a file.
Setup	1. Log in to application. 2. Select the particular user to send the file to.
Instructions	1. Select the file.
Expected Results	File has been loaded successfully.
Actual Result	File has been loaded successfully.
Verdict	Pass

Test Case 9

Table 47 TC-9

ID	TC-9
Purpose	Check that a registered user can send a message to some other user.
Setup	1. Log in to application. 2. Select the particular user to send message to. 3. Write the message. 4. Encrypt the message.
Instructions	1. Send the encrypted message.
Expected Results	Message has been sent successfully.
Actual Result	Message has been sent successfully.
Verdict	Pass

Test Case 10

Table 48 TC-10

ID	TC-10
Purpose	Check that a typed text message is encrypted successfully by hybrid cryptography (1^{st} level of encryption).
Setup	1. Log in to application. 2. Select the particular user to send message to. 3. Write the message.
Instructions	1. Encrypt the message.
Expected Results	Message has been encrypted.
Actual Result	Message has been encrypted.
Verdict	Pass

Test Case 11

Table 49 TC-11

ID	TC-11
Purpose	Check that an encrypted text message is successfully embedded into a cover image (2nd level of encryption).
Setup	1. Log in to application. 2. Select the particular user to send message to. 3. Write the message. 4. Encrypt the message.
Instructions	1. Send the encrypted message. 2. Select double encryption option. 3. Select the cover image. 4. Embed the encrypted message in cover image.
Expected Results	Encrypted message has been embedded in cover image.
Actual Result	Encrypted message has been embedded in cover image.
Verdict	Pass

Test Case 12

Table 50 TC-12

ID	TC-12
Purpose	Check that inserted image is encrypted successfully by hybrid cryptography (1st level of encryption).
Setup	4. Log in to application. 5. Select the particular user to send image to. 6. Select the image.
Instructions	2. Encrypt the image.
Expected Results	Image has been encrypted.
Actual Result	Image has been encrypted.
Verdict	Pass

Test Case 13

Table 51 TC-13

ID	TC-13
Purpose	Check that the encrypted image string is successfully embedded into selected cover image (2nd level of encryption).
Setup	5. Log in to application. 6. Select the particular user to send image to. 7. Select the image. 8. Encrypt the image.
Instructions	5. Send the encrypted image. 6. Select double encryption option. 7. Select the cover image. 8. Embed encrypted image in cover image.
Expected Results	Encrypted image has been embedded in cover image.
Actual Result	Encrypted image has been embedded in cover image.
Verdict	Pass

Test Case 14

Table 52 TC-14

ID	TC-14
Purpose	Check that the inserted file is encrypted successfully by hybrid cryptography (1st level of encryption).
Setup	7. Log in to application. 8. Select the particular user to send file to. 9. Select the file.
Instructions	3. Encrypt the file.
Expected Results	File has been encrypted.
Actual Result	File has been encrypted.
Verdict	Pass

Test Case 15

Table 53 TC-15

ID	TC-15
Purpose	Check that the encrypted file string is embedded successfully into selected cover image (2^{nd} level of encryption).
Setup	9. Log in to application. 10. Select the particular user to send file to. 11. Select the file. 12. Encrypt the file.
Instructions	9. Send the encrypted file. 10. Select double encryption option. 11. Select the cover image. 12. Embed the encrypted file in cover image.
Expected Results	Encrypted file has been embedded in cover image.
Actual Result	Encrypted file has been embedded in cover image.
Verdict	Pass

Test Case 16

Table 54 TC-16

ID	TC-16
Purpose	Check that the encrypted text message is extracted from the stego image and is then decrypted successfully.
Setup	1. Log in to application. 2. Receive the stego image.
Instructions	1. Decrypt the stego image. 2. Decrypt the extracted encrypted message string.
Expected Results	Text message has been decrypted.
Actual Result	Text message has been decrypted.
Verdict	Pass

Test Case 17

Table 55 TC-17

ID	TC-17
Purpose	Check that the encrypted image string is extracted from stego image and is then decrypted successfully.
Setup	1. Log in to application. 2. Receive the stego image.
Instructions	1. Decrypt the stego image. 2. Decrypt the encrypted image string.
Expected Results	Encrypted image has been decrypted.
Actual Result	Encrypted image has been decrypted.
Verdict	Pass

Test Case 18

Table 56 TC-18

ID	TC-18
Purpose	Check that the encrypted file string is extracted from stego image and is then decrypted successfully.
Setup	1. Log in to application. 2. Receive the stego image.
Instructions	1. Decrypt the stego image. 2. Decrypt the encrypted file string.
Expected Results	Encrypted file has been decrypted.
Actual Result	Encrypted image has been decrypted.
Verdict	Pass

Test Case 19

Table 57 TC-19

ID	TC-19
Purpose	Check that the selected chat is deleted successfully.
Setup	1. Log in to application.
Instructions	1. Select the chat to be deleted. 2. Delete the chat
Expected Results	The selected chat has been deleted successfully.
Actual Result	The selected chat has been deleted successfully.
Verdict	Pass

Test Case 20

Table 58 TC-20

ID	TC-20
Purpose	Check that the selected message is deleted successfully.
Setup	1. Log in to application. 2. Open the chat from which message is to be deleted.
Instructions	1. Select the message to be deleted. 2. Delete the message.
Expected Results	The selected message has been deleted successfully.
Actual Result	The selected message has been deleted successfully.
Verdict	Pass

Test Case 21

Table 59 TC-21

ID	TC-21
Purpose	Check that settings are updated successfully
Setup	1. Log in to application.
Instructions	1. Go to settings. 2. Make changes in the settings. 3. Update the settings.
Expected Results	Settings have been updated successfully.
Actual Result	Settings have been updated successfully.
Verdict	Pass

Test Case 22

Table 60 TC-22

ID	TC-22
Purpose	Check that selected contact is deleted successfully.
Setup	1. Log in to application. 2. Go to contacts.
Instructions	1. Delete the contact.
Expected Results	The contact has been deleted successfully.
Actual Result	The contact has been deleted successfully.
Verdict	Pass

Test Case 23

Table 61 TC-23

ID	TC-23
Purpose	Check that request is deleted successfully.
Setup	1. Log in to application. 2. Go to requests.
Instructions	1. Delete the request.
Expected Results	The request has been deleted successfully.
Actual Result	The request has been deleted successfully.
Verdict	Pass

Test Case 24

Table 62 TC-24

ID	TC-24
Purpose	Check that request is accepted successfully.
Setup	1. Log in to application. 2. Go to requests.
Instructions	1. Accept the request.
Expected Results	The contact has been added successfully.
Actual Result	The contact has been added successfully.
Verdict	Pass

Test Case 25

Table 63 TC-25

ID	TC-25
Purpose	Check that request is sent successfully.
Setup	1. Log in to application. 2. Go to people section.
Instructions	1. Search the user to send request to. 2. Send the request.
Expected Results	The request has been sent successfully.
Actual Result	The request has been sent successfully.
Verdict	Pass

Test Case 26

Table 64 TC-26

ID	TC-26
Purpose	Check that search returns the accurate results.
Setup	1. Log in to application. 2. Go to search section.
Instructions	1. Select the parameter to make search. 2. Enter the query.
Expected Results	The search results have been returned successfully.
Actual Result	The search results have been returned successfully.
Verdict	Pass

5.6 FIREBASE TEST LAB

Robo Test

The robo test automatically explores app on a wide array of devices to find defects and report any crashes that occur. It does not require to write app tests.

1. Upload APK File

The first step in Robo Test is to upload APK file of the application. After its upload the test start running automatically on a real Google device at one of Google data centers.

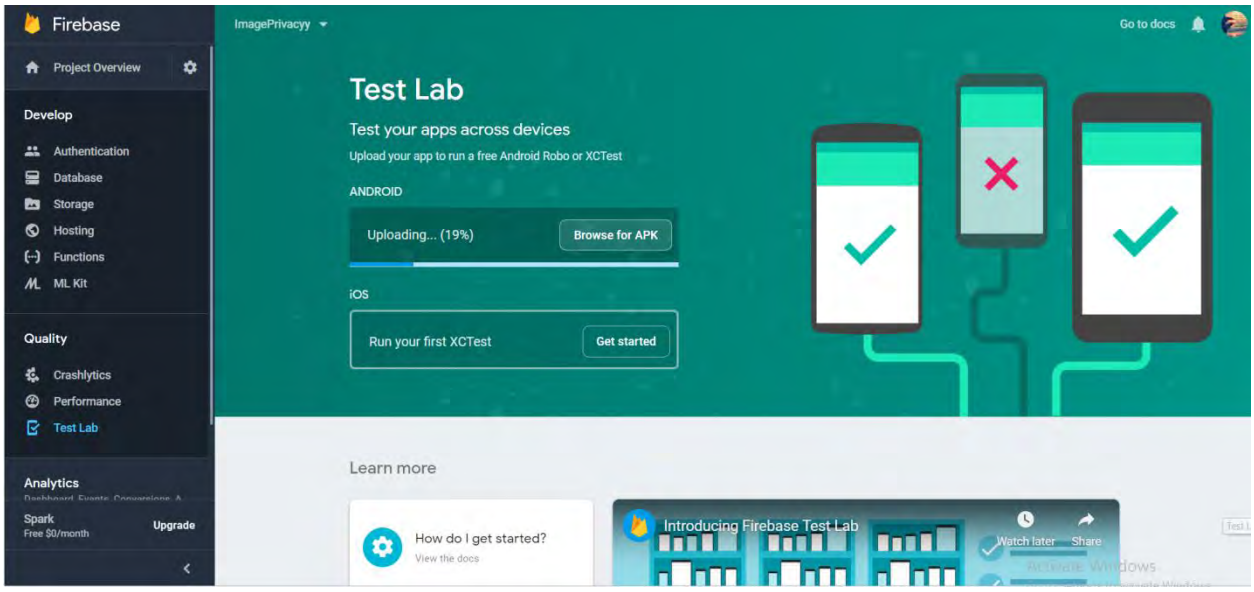

Figure 65 Upload APK File

2. Email Update

The test lab sends an email update about the test when the test gets completed or whether or not it resulted in any crashes.

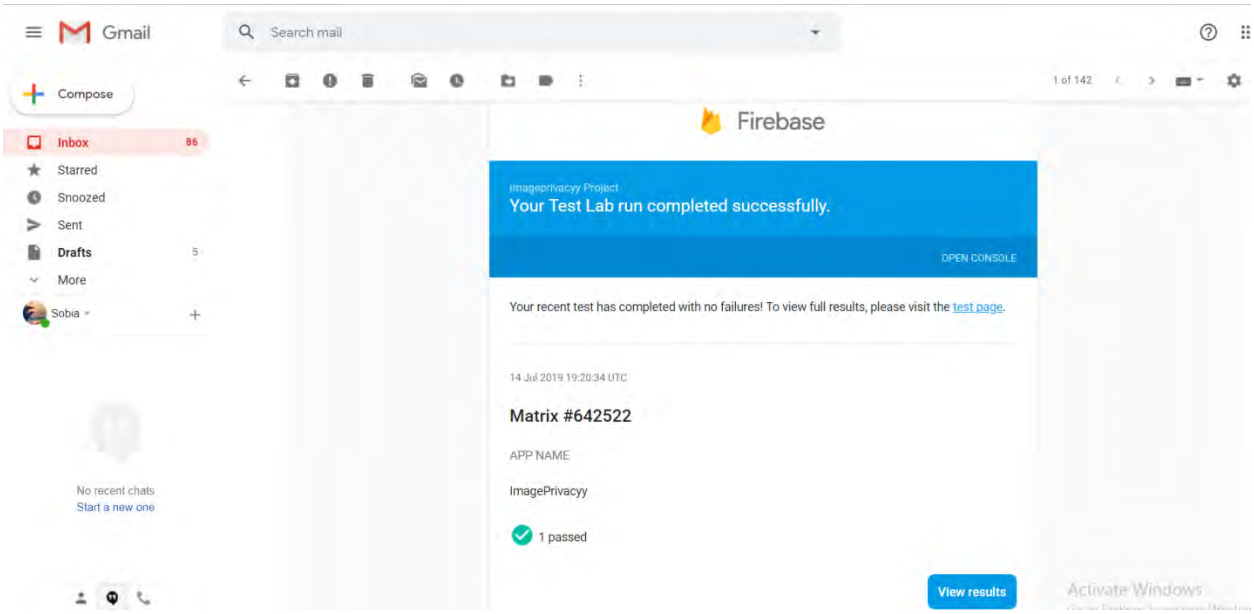

Figure 66 Email Update

3. Dashboard

The dashboard opens when the "view results" is clicked. It shows the path of where the robo test went.

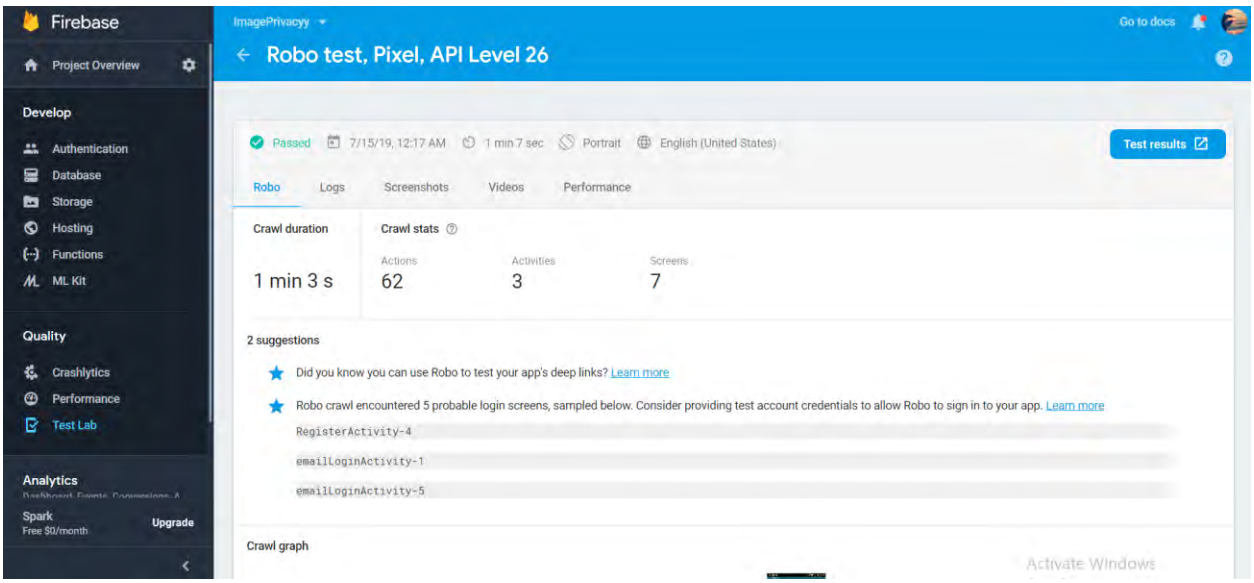

Figure 67 Dashboard

4. Crawl Graph

The robo test generates a crawl graph to show how long it crawled and where in the application.

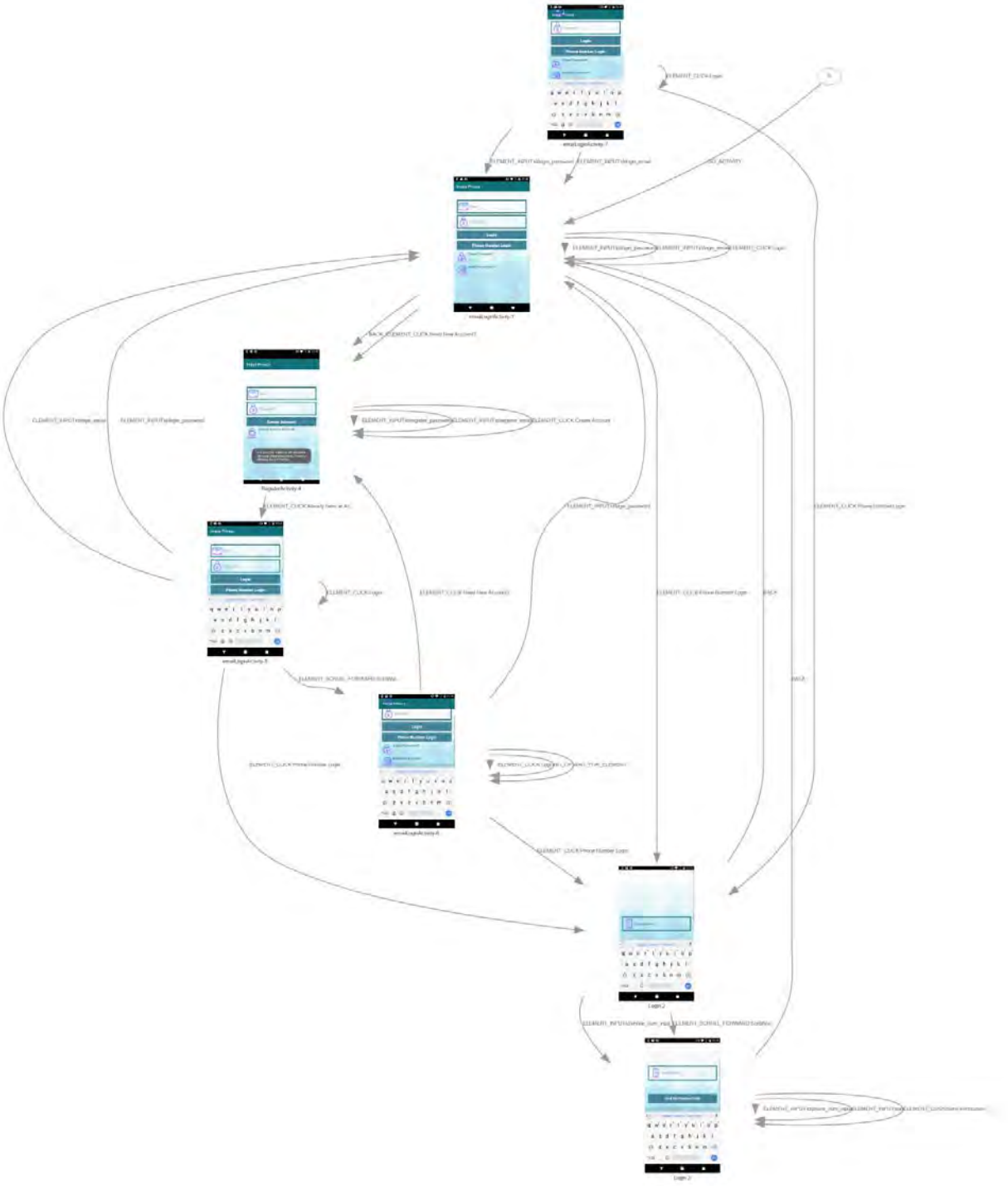

Figure 68 Crawl Graph

5. Screenshots

Robo test generate the screenshots of unique screens which robo found. It shows the interactions between the app and robo.

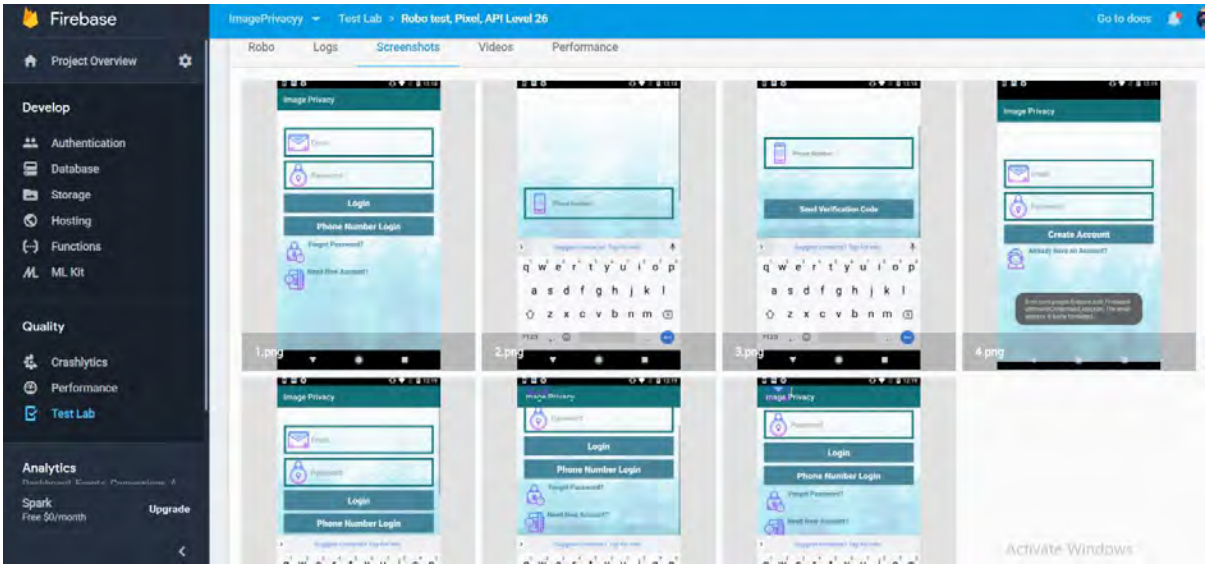

Figure 69 Screenshots

6. Log

Robo test generates the full log cat output of the app while it was being tested.

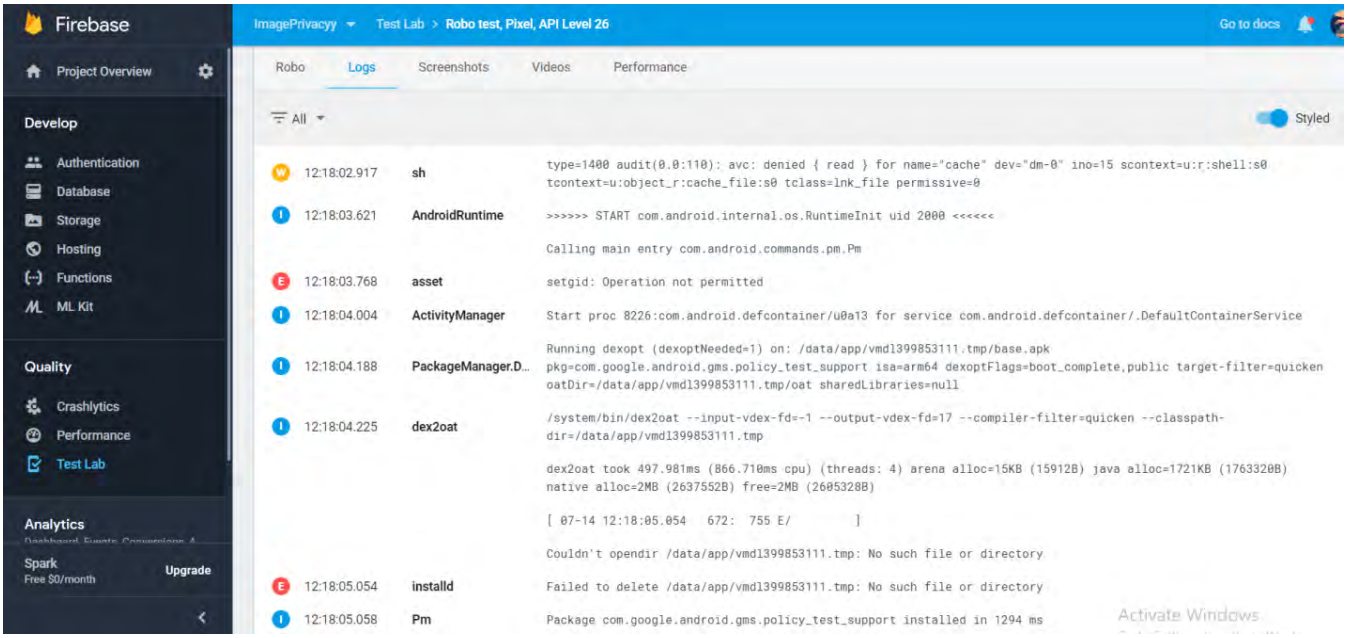

Figure 70 Test Log

7. Video

Robo test generates the video of the test.

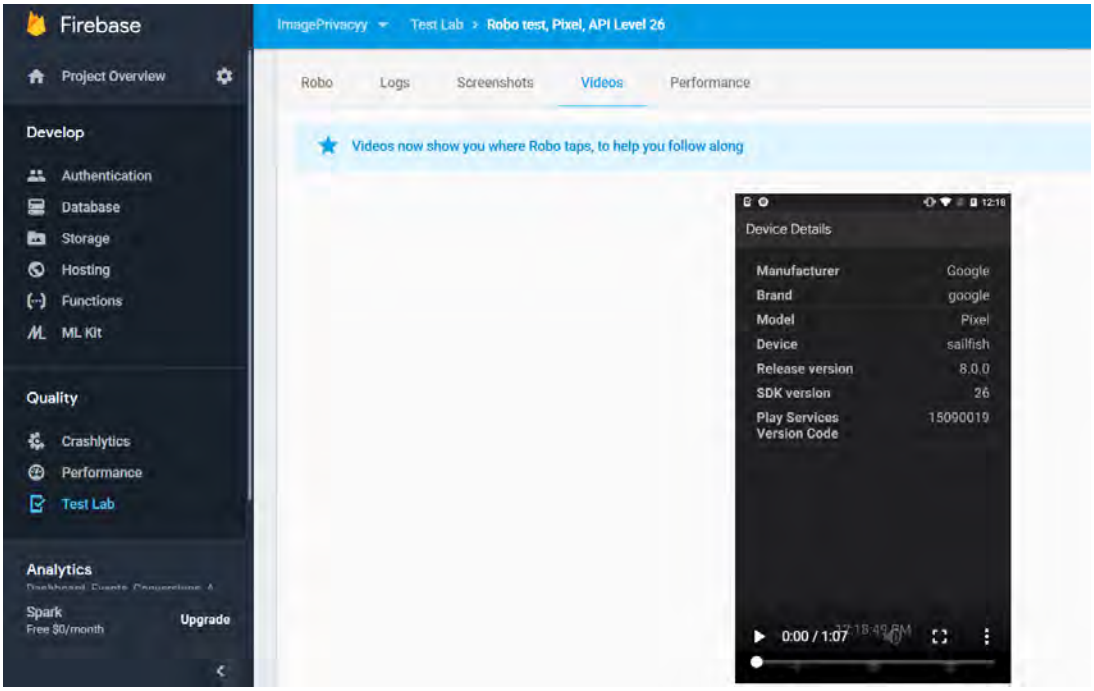

Figure 71 Video

8. Performance

The performance section displays some other performance metrics collected while app was being test.

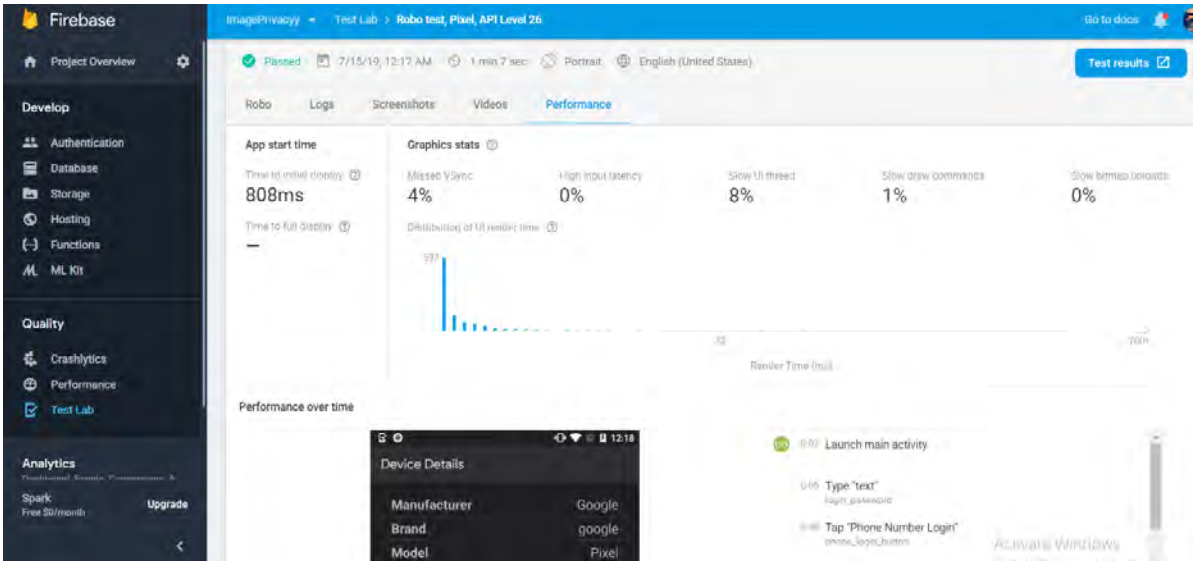

Figure 72 Performance 1

Figure 73 Performance 2

Customized Robo Test

We can customize a robo testing by selecting devices, API Levels, orientation, and locales to run tests on. I selected the Google Pixel 2 with the API level 28 and Galaxy S6 Samsung with API level 23, among virtual devices I selected Nexus 6 Motorola with API levels 22 and 25, among tablets I selected Nexus 9 HTC with API levels 25 and 22, the orientation I selected is Portrait and the locale is English (US). So I have selected over all 6 devices, 1 orientation and 1 locale. I start these 6 tests. The results are that one test is passed and the remaining 5 are skipped. The test that is passed is on Pixel 2 Google with API level 28 and the failed tests are with API level less that 28 that's because I mentioned that the minimum API level for the app should be 28 in the Constraints section.

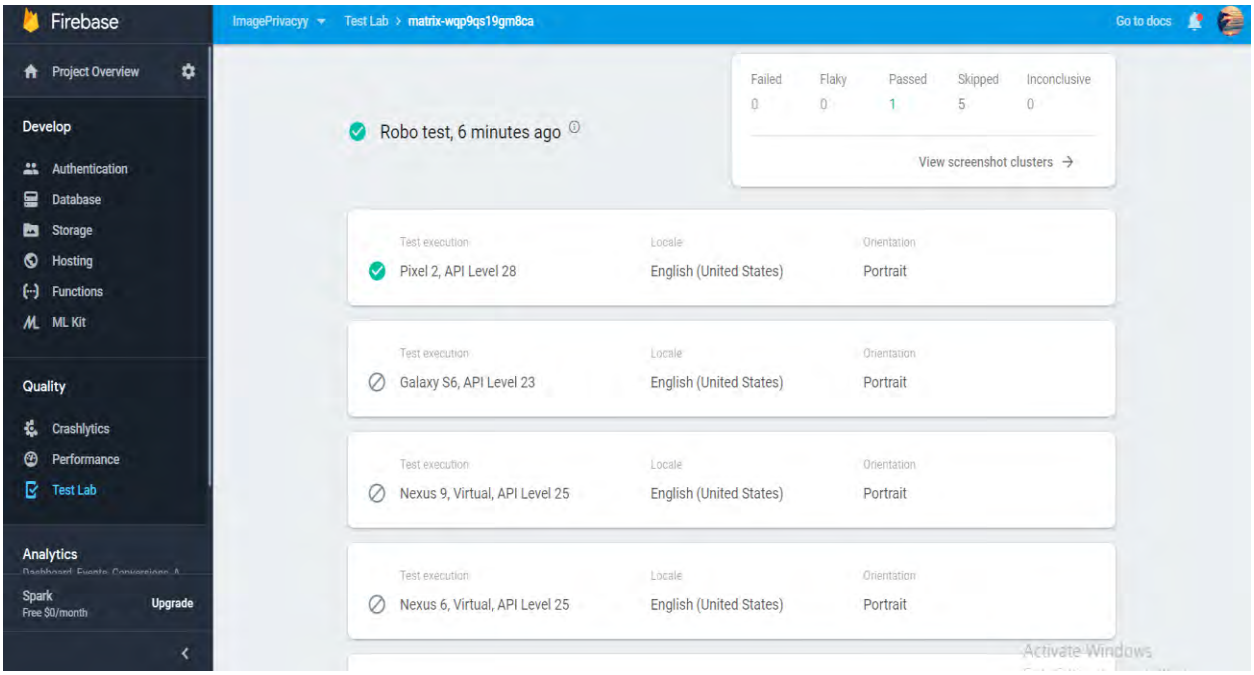

Figure 74 Customized Robo Test Result

CHAPTER 6

SOFTWARE IMPLEMENTATION DOCUMENTATION

6.1 INTRODUCTION

This document describes the project implementation for developing the Fusion application.

Language Selection

Java:

Java is used for the implementation of Fusion application. Java is an object oriented language that enables fewer dependencies in implementation.

Firebase:

Used for database. Firebase is a real-time, no relational and No SQL database. It stores data in JSON format and in tree structure.

Tools Selection

Tools that are used in the implementation of Fusion are:

Android Studio (IDE):

Android Studio is used for the implementation of android apps.

Node JS:

Node JS is an open source, cross platform runtime environment for developing server side and networking implementations. Node.js applications are written in JavaScript. In the fusion app Node JS v10.11.0 is used for push notifications.

Resources

Firebase Cloud Messaging (FCM):

Used for push notifications in android apps.

Glide:

Glide is a fast and efficient image loading library for android focused on smooth scrolling. In Fusion app Glide is used for smooth scrolling of a list of images, text messages, files, chats, contacts, people and requests, and to fetch and display images.

Espresso:

Espresso is a testing framework for android. Espresso is a part of the Android Support Repository since its 2.0 release.

6.2 APPLICATION SCREENSHOTS

Interface for Create Account

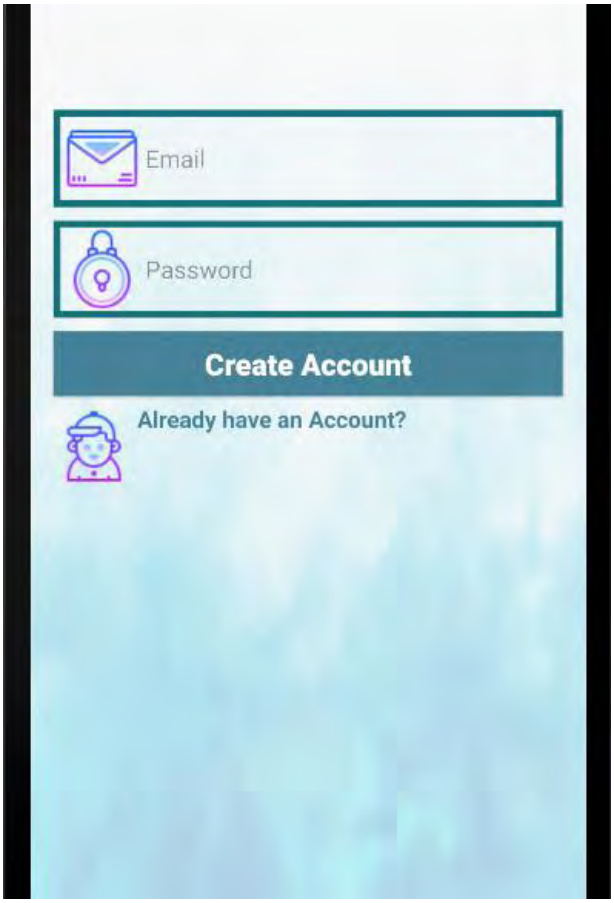

Figure 75 Interface for Create Account

Interface for Phone Number Login

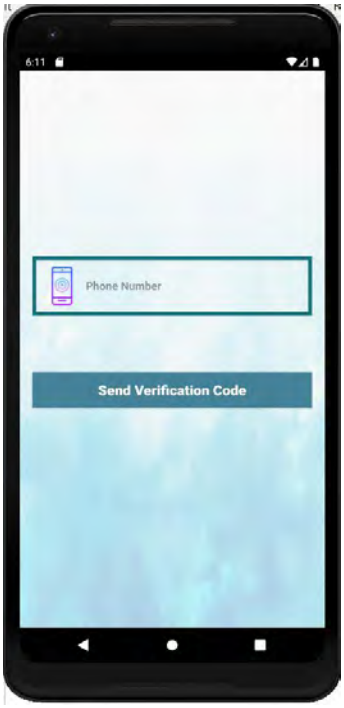

Figure 76 Interface for Phone Number Login

Interface for Login

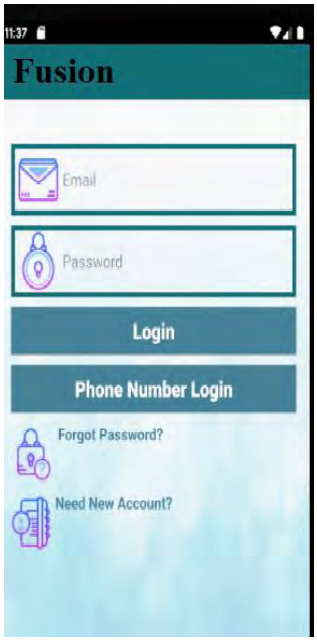

Figure 77 Interface for Login

Interface for Sign In

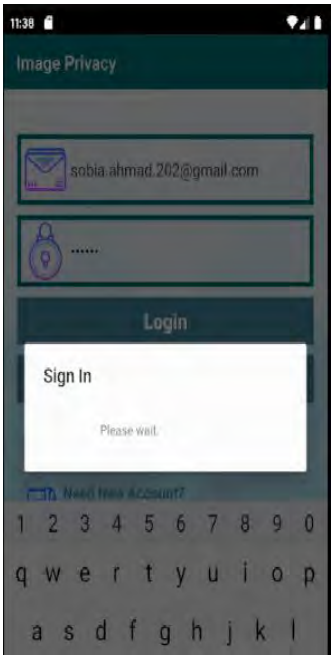

Figure 78 Interface for Sign In

Interface for Account Settings

Figure 79 Interface for Account Settings

Interface for Chat List

Figure 80 Interface for Chat List

Interface for People

Figure 81 Interface for People

Interface for Individual Chat

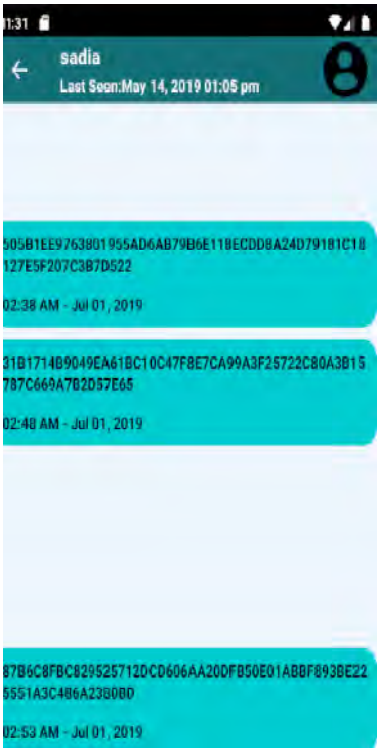

Figure 82 Interface for Individual Chat

Interface for Add Files

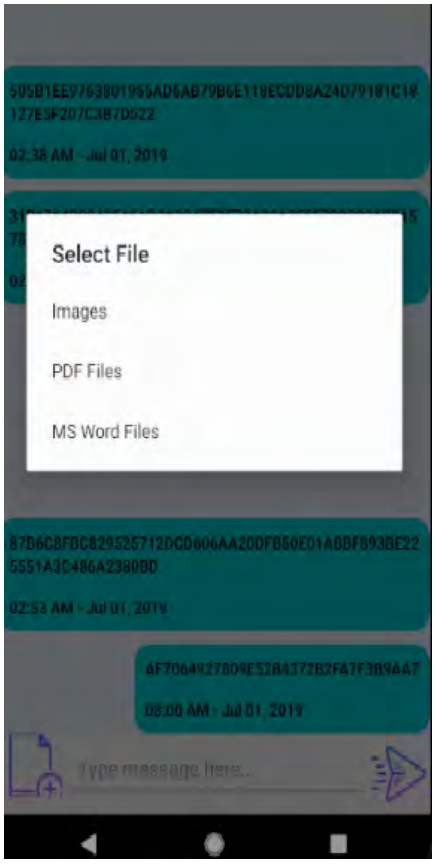

Figure 83 Interface for Add Files

Interface for Encryption

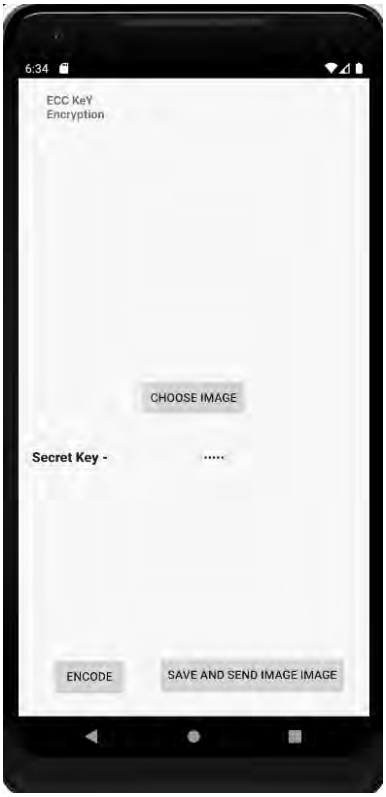

Figure 84 Interface for Encryption

Interface for Decryption

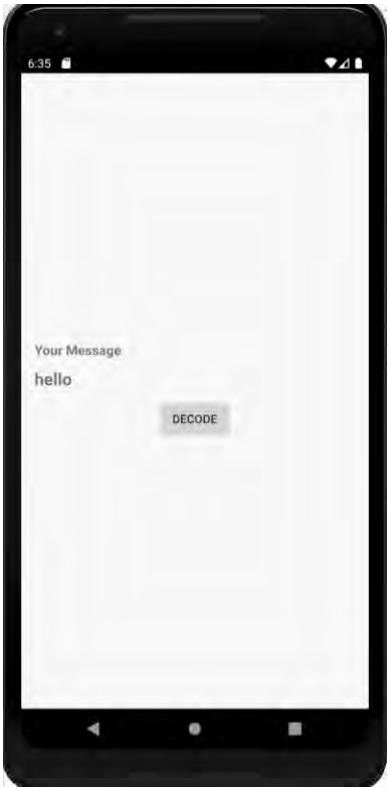

Figure 85 Interface for Decryption

Interface for Search

Figure 86 Interface for Search

Interface for User Profile

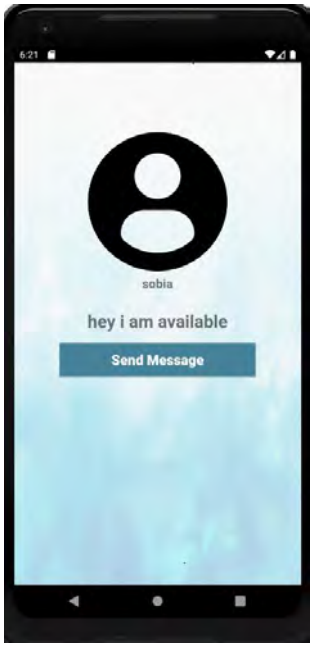

Figure 87 Interface for User Profile

Interface for Sent Request

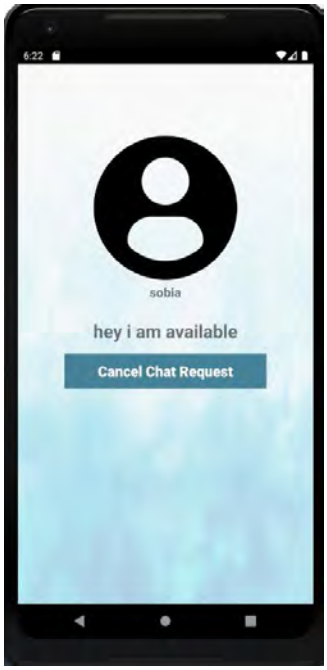

Figure 88 Interface for Sent Request

Interface for Received Request

Figure 89 Interface for Received Request

CHAPTER 7

CONCLUSIONS AND FUTURE ENHANCEMENTS

7.1 INTRODUCTION

This document describes the project conclusions and future enhancements- what type of new features can be added with time.

Summary

This application allows users to communicate with each other confidentially using combination of hybrid cryptography and steganography. The contents of the messages are only between sender and receiver.

7.2 Conclusions

- We are now able to send text messages, images and files using 1st level of encryption (hybrid cryptography).
- We are also able to send images, text messages and files using 2nd level of encryption (hybrid cryptography + steganography).
- We are able to decrypt receive images, text messages or files whether it is encrypted by 1st level or by 2nd level of encryption.
- We are able to add/ delete contact.
- We are able to update user profile.
- We are able to search other users of the application, contacts and chats.

7.3 Future Enhancements

- Send more than one image or file at a time.
- System selects the cover image randomly and embeds in it without asking user to choose.
- Different Algorithms will be implemented to give user a choice to choose from.

Table 65 APPENDIX: LIST OF ACRONYMS

Acronym	Description
SRS	Software Requirements Specifications
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
ECC	Elliptical Curve Cryptography
RSA	Rivest Shamir Adleman
SDD	Software Design Document
RTM	Requirements Traceability Matrix
UI	User Interface
ANSI	American National Standard Institute
UAT	User Acceptance Test

APPENDIX B: REFRENCES

BOOKS:

[1]. W. Stallings, Network Security Essentials Applications and Standards, 4/e, Prentice Hall, 2011.

[2]. William Stallings, Cryptography and Network Security Principle and Practice, Sixth Edition, 2014.

[3]. Multimedia and Web Technology, Authors Reeta Sahoo and Gagan Sahoo.

[4]. C. Larmen Applying UML and Patterns An Introduction to Object Oriented Analysis and Design and Iterative Development, 3rd ed., Massachusetts: Pearson Education, 2005.

[5]. Roger S. Pressman, Software Engineering – A Practitioner's Approach, McGraw Hill, 7th Edition, 2010

RESEARCH PAPERS:

[1]. Anis Cherid (2018). "Asymmetric and Symmetric Cryptography to Secure Social Network Media Communication: The case of android-based E-Learning software," Universitas Mercu Buana , Jakarta, Indonesia.

[2]. Fatima Maikudi Abubakar and Shitu Abdullahi Lame (2015). "The Role of Cryptography in Information and Data Security," A.D. Rufa'I College for Legal and Islamic Studies Misau, Bauchi State, Nigeria.

[3]. Omar Badeea Baban, J.E. kamalasekaran (2017). "Sybil Attack Prevention in WSN Using Elliptic Curve Cryptography and Genetic Algorithm," Sinhgad College of Engineering, Pune, India.

[4]. Bhinal Chauhan, Shubhangi Borikar, Shamali Aote, Prof. Veena Katankar (2018). "A Survey on Image Cryptography Using Lightweight Encryption Algorithm," RTMNU, Nagpur, India.

[5]. Ramesh Yegireddi, R Kiran Kumar (2016). "A Survey on Conventional Encryption Algorithms of Cryptography," AndhraPradesh, India.

[6]. Pratap Chandra Mandal (2012). "Evaluation of Performance of the Symmetric key Algorithms: DES, 3DES, AES and Blowfish," Kolkata, India.

[7]. V. Kumara Swamy, Prabhu Benakop (2017). "Predominance of Blowfish over 3DES Symmetric Key Algorithm For Secure Integrated Circuits using Verilog HDL," Telangana, India.

[8]. Mansoor Ebrahim, Shujaat Khan, Umer Bin Khalid (2013). "Symmetric Algorithm Survey: A Comparitive Analysis," Karachi, Pakistan.

[9]. Madhumita Panda (2016). "Performance Analysis of Encryption Algorithms for Security," India.

[10]. Jangala. Sasi Kiran M. Anusha, A. VijayKumar, M. Kavya (2016). "Cryptography: The Science of Secure Communication," India.

[11]. Vishal R. Pancholi, Dr. Bhadresh P.Patel (2015). "Cryptography: Comparative Studies of Different Symmetric Algorithms," India.

[12]. Faiqa Maqsood, Muhammad Mumtaz Ali, Muhammad Ahmed, Munam Ali Shah (2017). "Cryptography: A Comparative Analysis for Modern Techniques," Pakistan.

[13]. Shivani Sharma, Yash Gupta (2017). “Study on Cryptography and Techniques,” India.

[14]. J. Athena, V. Sumathy (2017). “Survey on Public Key Cryptography Scheme for Securing Data in Cloud Computing,” India.

[15]. P. Rajkumar, R. Kar, A. K. Bhattacharjee, H. Dharmasa (2012). “A Comparative Analysis of Steganographic Data Hiding within Digital Images,” India.

[16]. Minati Mishra, Priyadarsini Mishra, Flt. Lt. Dr. M. C. Adhikary (2012). “Digital Image Data Hiding Techniques: A Comparative Study,” India.

[17]. Firas Sabah Salih Al-Turaihi (2016). “Three Levels of Protection by using Cryptography and Steganography,” Babylon University.

[18]. Alpa Agath, Chintan Sidpara, Darshan Upadhyay (2018). “Critical Analysis of Cryptography and Steganography,” India.

[19]. Priyanka Jagota (2015). “Image Steganography: A Review,” Punjab.

[20]. IEEE Std 830-1998 (Revision of IEEE Std 830-1993): IEEE Recommended Practice for Software Requirements Specifications.

[21]. Amandeep Kaur, Rupinder Kaur and Navdeep Kumar (2015). “A Review on Image Steganograophic Techniques,” India.

[22]. Prof. Amee J.Mankad and Makwana Sameer K (2017). “Comparison and Analysis of F5 Algorithm and PVD Technique base Steganography for Data Hiding in terms of MSE and PSNR Parameters,” India.

[23]. Ratnakirti Roy and Suvamoy Changder (2016). “Quality Evaluation of Image Steganography Techniques: A Heuristic based Appraoch,” India.

[24]. Ms. Nikita N Chintawar, Ms. Sonali J Gajare, Ms. Shruti V Fatak, Ms. Sayali S Shinde and Prof. Gauri Virkar (2016). “Enhancing Cloud Data Security Using Elliptic Curve Cryptography,” Pune, India.

WEBSITES:

[1]. http://www.iaaf.uwa.edu.au, last accessed 25 October 2017.

[2]. https://books.google.com.pk , last accessed 5 November 2017.

[3]. https://www.wikipedia.org/ , last accessed 8 November 2017.

[4]. http://www.omnisecu.com , last accessed 1 December 2017.

[5]. https://casecurity.org , last accessed 15 January 2018.

[6]. https://www.yubico.com , last accessed 25 January 2018.

[7]. https://cryptography.io , last accessed 10 February 2018.

[8]. http://searchsecurity.techtarget.com , last accessed 3 March 2018.

[9]. https://en.wikipedia.org/wiki/Data_Encryption_Standard , last accessed 25 June 2019.

[10]. https://www.izenda.com/5-benefits-3-tier-architecture , last accessed 12 July 2019.

[11]. https://android.jlelse.eu/android-mvp-for-beginners-25889c500443 , last accessed 12 July 2019.