

UNIVERSITY OF MUMBAI

A PROJECT REPORT ON

“Image Steganography”

SUBMITTED BY

Under the guidance of

DECLARATION

I hereby declare that the project entitled, “Image Steganography” done at place where the project is done, has not been in any case duplicated to submit to any other university for the award of any post-graduation degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in fulfillment of the requirements for the award of ----- degree of -
----- to be submitted as ----- project as part of our curriculum.

ABSTRACT

- Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information.
- Android is a software platform and operating system for mobile devices. This is a project that is used to perform secret data transmission by performing encryption of text on images. The sender uses a key to perform encryption and the same key is given to the receiver to decrypt and obtain the data.

INDEX

Sr. No.	Contents	Page No.
1	Introduction	
	1.1 Introduction	8
	1.1.1 Problem definition	8
	1.1.2 Objectives of Project	8
	1.1.3 Scope of Project	8
	1.2 Technical Details	9
	1.2.1 Overview of Front End	9
	1.2.2 Overview of Back End	9
2	System Study and Planning	
	2.1 System Study	10
	2.1.1 Existing System	10
	2.1.2 Disadvantages of Existing system	10
	2.1.3 Proposed System	10
	2.2 System Planning and Schedule	11
	2.2.1 S/W development Model	11
3	System Design	
	3.1 Software Requirement Specification (SRS)	14
	3.1.1 Introduction of SRS	14
	3.1.2 Technology Requirements	15
	3.1.2.1 Hardware to be used	16
	3.1.2.2 Software/tools to be used	16
	3.2 Detailed life Cycle of the Project	17
	3.2.1 Modules	17

	3.2.2 Object Oriented Analysis & Design Diagrams	18
	3.2.2.1 Use Case Diagram	18
	3.2.2.2 Activity Diagram	19
	3.2.2.3 Class Diagram	20
	3.2.2.4 Sequence Diagram	21
	3.2.2.5 Flowchart/DFD/ER diagram	22
	3.2.3 Database	23
	3.2.3.1 Database Table	23
	3.2.4 I/O Screen Layout	24
4	Testing	26
	4.1 Methodologies used for testing	26
	4.2 Types of Testing	26
5	Conclusion	29
6	Future Enhancements	30
7	References	31

Chapter 1: Introduction

1.1 Introduction

1.1.1 Problem Definition:

- Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information.
- Android is a software platform and operating system for mobile devices. This is a project that is used to perform secret data transmission by performing encryption of text on images. The sender uses a key to perform encryption and the same key is given to the receiver to decrypt and obtain the data.
- Encryption of data on images provides a safe and secure transmission of data between the sending and receiving party.
- The data/text which the sender wants to transmit is selected first and then an image is chosen from the current mobile device.
- The chosen text is then encrypted in the image such that the data is not visible to any third party.
- After encryption is performed, the image is sent to receiving party and the receiver decrypts the data using the key given by the sender.
- This decryption process can only be performed by an authenticated receiver using this application

1.1.2 Objective:

Main objective of this application is to provide a secure and secret transmission of text by encrypting it on an image using a key and which can only be decrypted by an authenticated receiver using the same key on the same application.

1.1.3 Scope of Project:

This project is developed for hiding information in any image file. The scope of the project is implementation of steganography tools for hiding information includes any type of information file and image files and the path where the user wants to save Image and extruded file.

1.2: Technical Details

1.2.1 Overview of Front End:

❖ Android Studio:

- Android Studio is the official integrated development environment (IDE) for Android application development.
- There are some features of Android Studio:
 - ✓ Instant Run
 - ✓ Visual Editor/ Layout Editor
 - ✓ Firebase Plugin
 - ✓ APK Analyzer
 - ✓ Extensive testing tools and frameworks

❖ Technology:

- Java:
 - Java is a popular general-purpose programming language and computing platform.
 - Java, the programming language used to develop Android applications.
 - Features of JAVA:
 - ✓ Fast
 - ✓ Reliable
 - ✓ Secure

Chapter 2: System Study and Planning

2.1 System Study:

2.1.1 Existing System:

- The perspective of data security, which has always been an important aspect.
- Encryption of data/text was previously done using hash and cryptographic algorithms, which is written in binary form. Such encryption techniques are visible in a specific form and the intruder who is aware of binary text formats can easily read the text.
- In those times it provided a lot of security but as technology progressed many hackers also came into existence and it could not provide security upto that extent.

2.1.2 Disadvantages of Existing system:

- Time consuming.
- Easy to extract messages.
- Hackers can hack the code.

2.1.3 Proposed System:

- In this project we used many techniques to encrypt and make the data in-visible to any one (not even to the receiver). We used LSB (Least Significant Bit) format, stenography and cryptographic techniques to encrypt the data and the data/text being encrypted on an image will not be visible.
- The sender will use a key for security reasons and the same key should be used by receiving party to decrypt the image and obtain the text written on it.
- The image on which data encryption is being performed can be chosen from the existing device and after all the process completes, that image can be sent through Bluetooth or mail or google drive etc.

2.2 System Planning and Schedule:

2.2.1 S/W development Model:

Why not Water fall model?

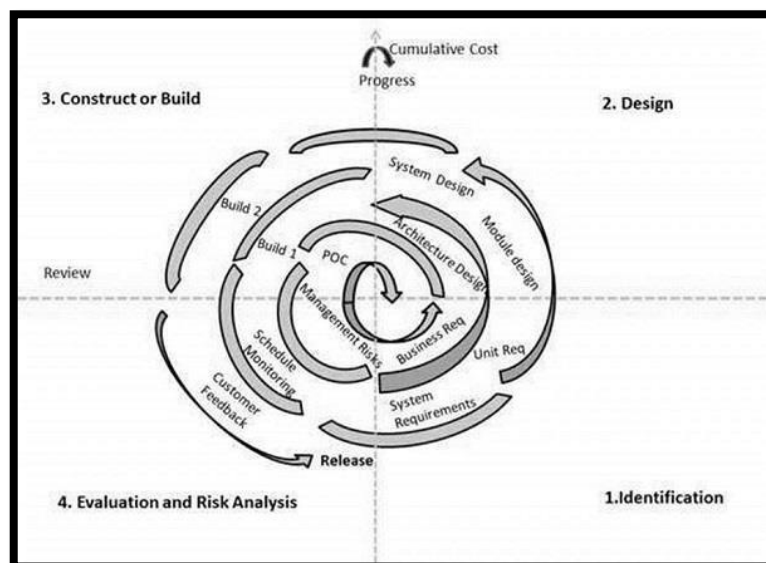
Water fall model can be adopted because in our case because requirements are known in advance but there are some limitations of waterfall model due to which it is not feasible to adopt:

- No parallelism of work.
- Time consuming.

Why Spiral Model?

The development life cycle model chosen for project is spiral model. This Spiral Model is a risk-driven process model generator for software projects. Based on unique risk patterns of given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental waterfall or evolutionary prototyping.

- The spiral model is a risk driven process model generator for software project.
- This provides support for risk handling.
- It is an iterative model.
- It allows incremental releases of product or incremental refinement through each iteration around the spiral



Phases of spiral model are:

1. Identification
2. Design
3. Construct or Build
4. Evaluation and risk analysis

1. Identification:

- This phase consists of gathering the requirements for this project.

2. Design:

- This interface of system is developed keeping in mind that it should be good looking, attract at first sight, easy to understand and self-explanatory.

3. Construction:

- The scope clearly defines the boundaries of proposed system.
 1. The system will bridge the gap between the seller and the customer.
 2. They all can communicate each other and availability of thing will increase.

The main features while executions phases are:

- **Efficiency**

The Efficiency of any system is concerned with the minimum processing time as well as the optimal use of application resource in designing the proposed system; the efficiency factor has been taken well into consideration.

- **User-Friendly Interface**

The interface will be user friendly so that a common user can use It easily. It makes it very easy for user to jump from one section to another. Another uniqueness of design that it is based on fluid interface it can adjust itself accordingly to device in which application is being used.

- **Data Security and Integrity**

Data Security and integrity is our top most priority we will make sure that private data of every single user remains confidential and never be compromised. For this purpose, we will use efficient security mechanisms.

- **Feasibility**

Our application solution is aimed to provide with: Technology and Control.

- **Extensibility**

Key features of proposed solution would be its extensibility. Our solution enables a new level's of remote automation, programmability and extensibility using modern technology.

- **Scalability**

Scalability can be defined as the ease with which a application or components can be modified to fit the problems are. Our system will easily be modified.

4. Evaluation and risk analysis

- Risk Analysis include identifying, estimating and monitoring technical feasibility and management risks such as loss of internet connection or done legacy problems.
- Evaluation is the process which overcomes all the risk analysis and modified the risk analysis and modified the application and adds various data or keeps the application updated.

Chapter 3: System Design

3.1 Software Requirement Specification (SRS):

3.1.1 Introduction of SRS:

❖ External Interface Requirement

The external system is to assume full responsibility for storage functions as well as warehouse management and warehouse control for an entire warehouse. The interfaces in this section are specified by documenting: the name and description of each scheme, source or input, destination or output, range's, accuracy and tolerances, units of measure, timing, delay formats, and display formats and organization and data formats. The user interface required to be developed for the system should be user-friendly and attractive. The interface between the user and the system will be IMP (Icons, Menu, Pointers) keeping in mind that the system is to be run through a web browser. All operations will be off point and click nature with all navigations performed through application specifically buttons and menus:

Buttons: the button is activated when the user will click on the left click of the mouse within the bounds of the button. And thus the action associated with it will be carried out.

Menu: All the operations will be arranged.

❖ Functional requirements:

- Functional requirements are those requirements that are used to illustrate the internal working nature of the application, the description of the system and explanation of each subsystem.
- It consists of what task the application should perform, the processes involve, which data should the application holds and the interfaces with the user.

❖ Non-functional requirements:

- It describes aspects of the application that are concerned with how the application provides the non-functional requirements i.e., it specifies the criteria that can be used to judge the system attributes:
 - ✓ **Portability:** The system is developed for secured purpose, so it can't be portable.
 - ✓ **Availability:** This application will be available only until the application on which it is install, is running.
 - ✓ **Scalability:** Applicable.

3.1.2 Technology Requirements:

1] Hardware requirement: -

- **Processor:** Intel dual core or above
- **Processor Speed:** 1.0GHZ or above
- **RAM:** 4GB RAM or above
- **Hard Disk:** 500GB hard disk or above

2] Software Requirement:-

- **Language:** Java JDK, Android SDK, Android Mobile Device

3] Functional Requirements:-

Users of the Image Steganography, must be provided the following functionality:

- Browsing image for encoding.
- Entering secret message.
- Entering secret password.
- Saving the encoded image.
- Taking saved encoded image for decoding.
- Entering secret key for decoding.
- Decoding the encoded image.

4] Non-functional requirements:-

Performance criteria:

Time:

The elapsed time between the encoding and decoding of message should be as minimum as possible.

User-friendly:

Our Image Steganography application should be more users friendly. The user interface should be kept simple and uncluttered. Since the different type of people will interact with this process so our project should be very easy to them to understand.

Flexibility:

Our project should be so flexible that whenever we want to make changes in it very easily it can be done.

Portable:

Our project should be portable on any platform and available on applications easily and at a faster speed than others.

Reusable:

All the information should be easily get processed so that many users can interact with us very easily and very fast without any information destroyed.

3.1.2.1 Hardware to be used:

- **Processor:** Intel dual core or above
- **Processor Speed:** 1.0GHZ or above
 - **RAM:** 4GB RAM or above
 - **Hard Disk:** 500 GB hard disk or above

3.1.2.2 Software/tools to be used:

- **Language:** Java JDK, Android SDK, Android Mobile Device

3.2 Detailed life Cycle of the Project

3.2.1 Modules:

A module is a collection of source files and build settings that allow you to divide your project into discrete units of functionality. Your project can have one or many modules, and one module may use another module as a dependency. You can independently build, test, and debug each module.

Android Studio offers a few distinct types of module:

Android app module

Provides a container for your app's source code, resource files, and app level settings such as the module-level build file and Android Manifest file. When you create a new project, the default module name is "app".

In the **Create New Module** window, Android Studio offers the following types of app modules:

- Phone & Tablet Module
- Wear OS Module
- Android TV Module
- Glass Module

Feature module

Represents a modularized feature of your app that can take advantage of Play Feature Delivery. For example, with feature modules, you can provide your users with certain features of your app on-demand or as instant experiences through Google Play Instant.

Library module

Provides a container for your reusable code, which you can use as a dependency in other app modules or import into other projects. Structurally, a library module is the same as an app module, but when built, it creates a code archive file instead of an APK, so it can't be installed on a device.

In the Create New Module window, Android Studio offers the following library modules:

- **Android Library:** This type of library can contain all file types supported in an Android project, including source code, resources, and manifest files. The build result is an Android Archive (AAR) file that you can add as a dependency for your Android app modules.
- **Java Library:** This type of library can contain only Java source files. The build result is an Java Archive (JAR) file that you can add as a dependency for your Android app modules or other Java projects.

3.2.2 Object Oriented Analysis & Design Diagrams:

3.2.2.1) Use Case Diagram:-

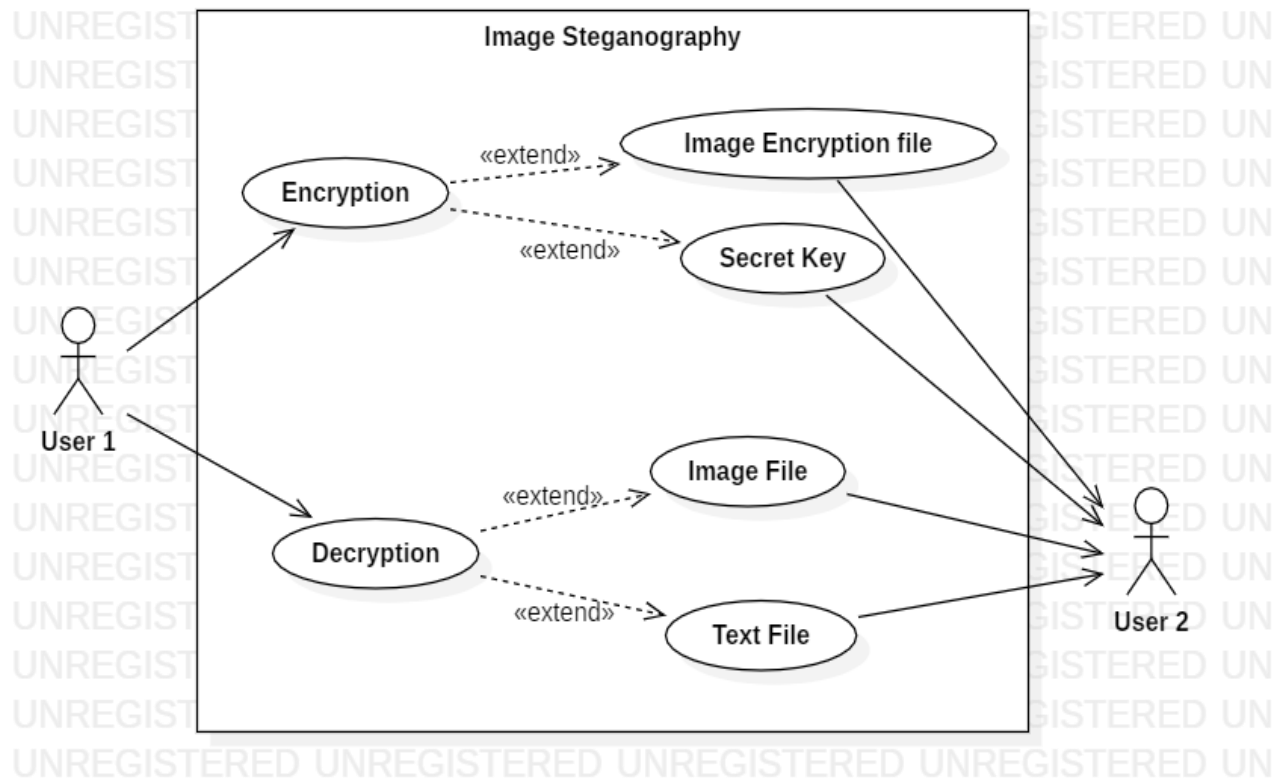


Figure: -Use Case diagram for Image Steganography

Assumptions:-

Actors:

The Actors of the system are User 1 and User 2

Use cases:

I have identified a set of use cases based on the functionalities and goals of the application.

- **Encryption**- This use case denotes a set of actions required for user to encrypt the image and create an Image encryption file and a secret key.
- **Decryption**-This use case denotes a set of actions required by another user to decrypt the message from the encrypted image using a secret key.

3.2.2.2) Activity Diagram:-

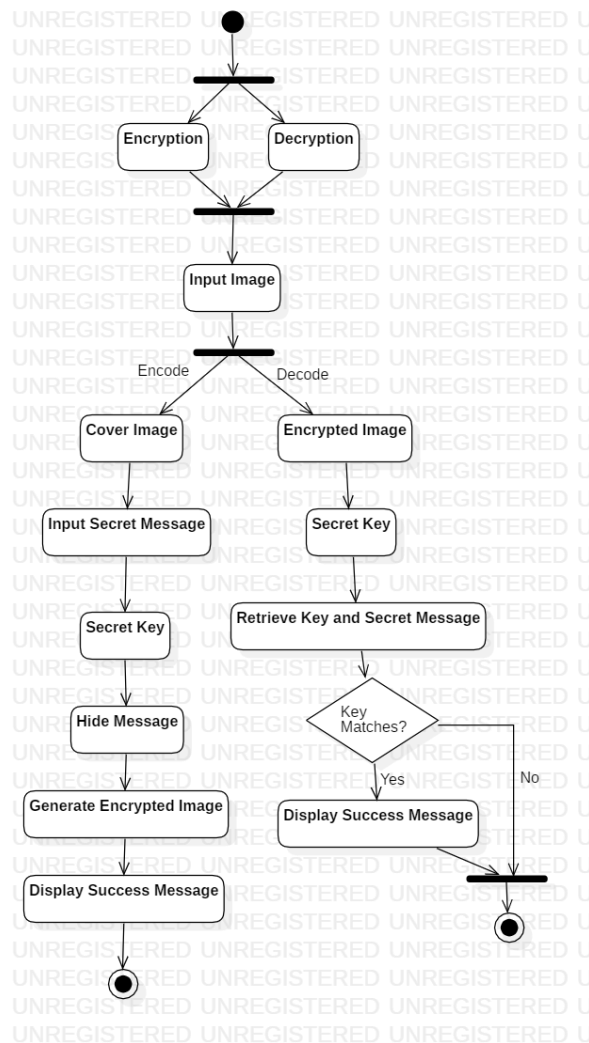


Figure: -Activity diagram for Image Steganography

Assumptions:-

- Firstly, the user starts with input the image file.
- Secret message is validated to determine whether the secret message is entered or not.
- The system only proceed if it receive the correct data.
- Secret Key and Secret Message is needed to generate encrypted image.
- Key is also validated whether the key is entered or not. After user select the output folder, the system may generate the Encrypted Image.
- Meanwhile, Encrypted image and the same Secret key is needed to retrieve the message back.
- If the key is not matched, the process end, else the secret message is displayed.

3.2.2.3) Class Diagram:-

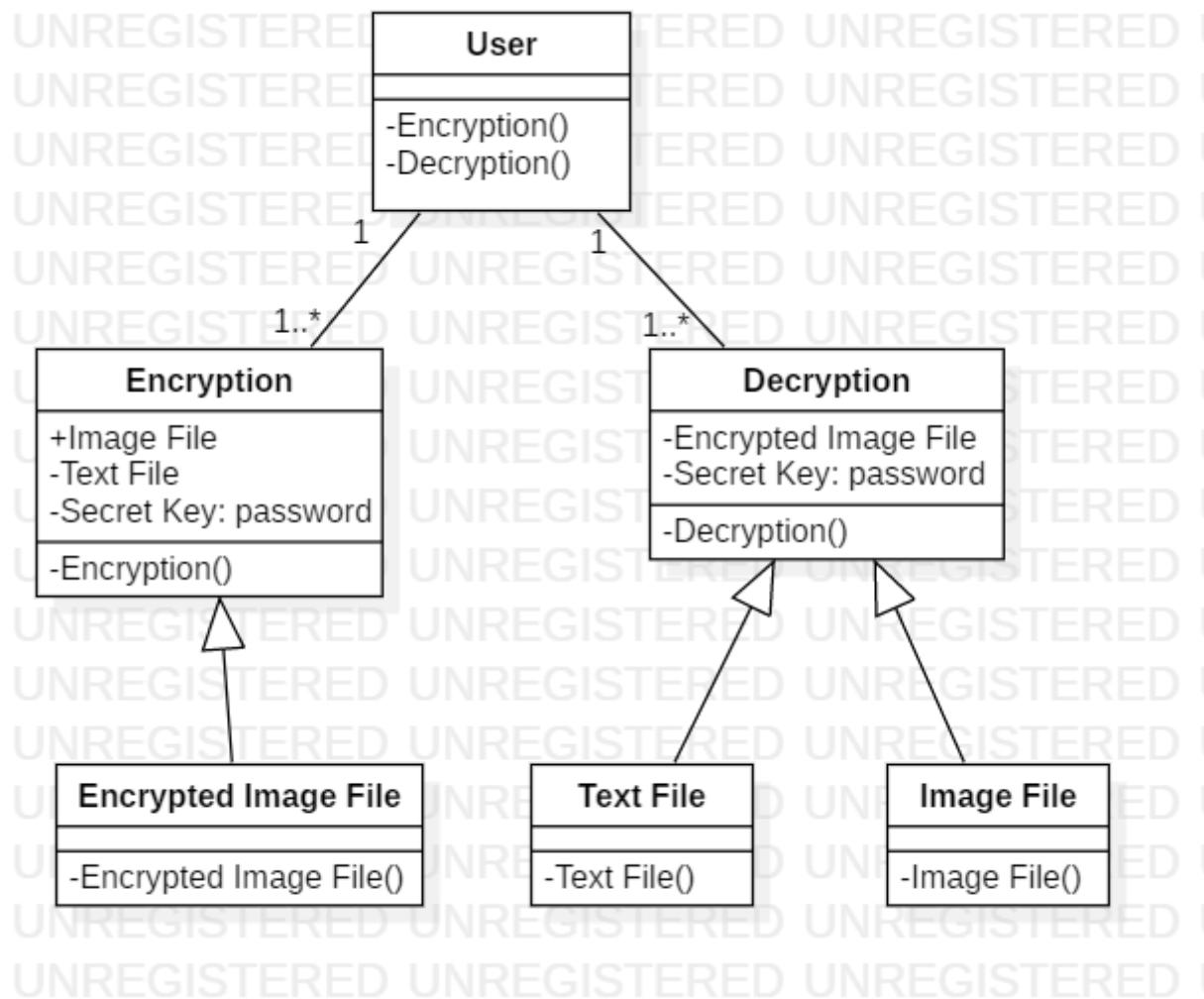


Figure: -Class diagram for Image Steganography

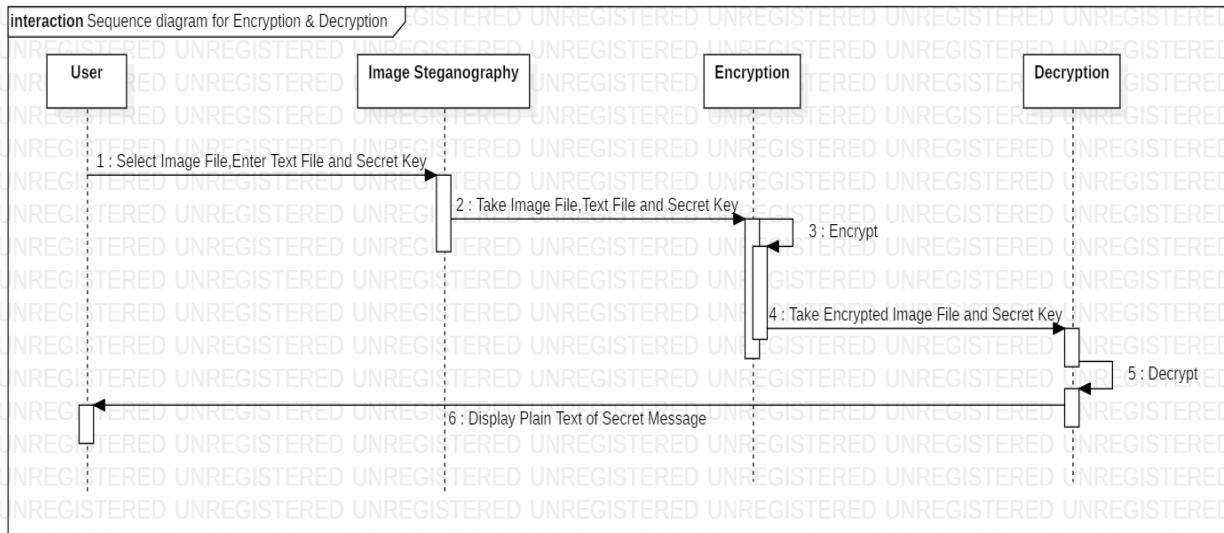
Assumptions:-

The following diagram is an example of Image Steganography Application. It describes a particular aspect of the entire application.

- First of all, User and Encryption, Decryption have a one-to-many relationship between each other.
- User are identified as the element of the system. They have a one-to-many relationship
- Encryption class is an abstract class and it has one concrete class (Generalization relationship) Encrypted Image File.
- Decryption class is an abstract class and it has two concrete classes (Generalization relationship) Text File and Image File.

3.2.2.4) Sequence Diagram:-

A. Sequence diagram for Encryption & Decryption



Assumptions:

A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner

- For instance, the UML diagram describes the sequences of messages of a (simple) Image Steganography.
- This diagram represents a User performing the encryption and decryption process.
- The dotted lines extending downwards indicate the timeline.
- Time flows from top to bottom.
- The arrows represent messages from an actor or object to other objects.

For example, the user sends image file, text file and secret key for encryption process.

3.2.2.5) Flowchart diagram:-

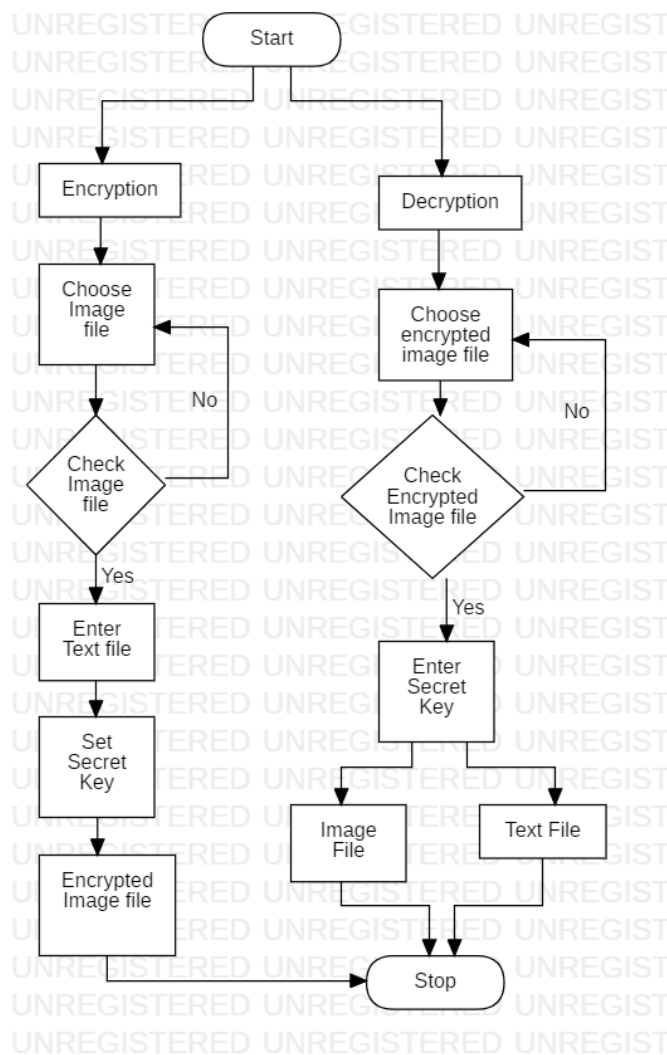


Figure: -Flow Chart diagram for Image Steganography

Assumptions:-

- Firstly, the user starts with input the image file.
- Secret message is validated to determine whether the secret message is entered or not.
- The system only proceed if it receive the correct data.
- Secret Key and Text File is needed to generate encrypted image.
- Key is also validated whether the key is entered or not then the system may generate the Encrypted Image.
- Meanwhile, Encrypted image and the same Secret key is needed to retrieve the message back.
- If the key is not matched, the process end, else the secret message is displayed.

3.2.3 Database

3.2.3.1) Database Table:-

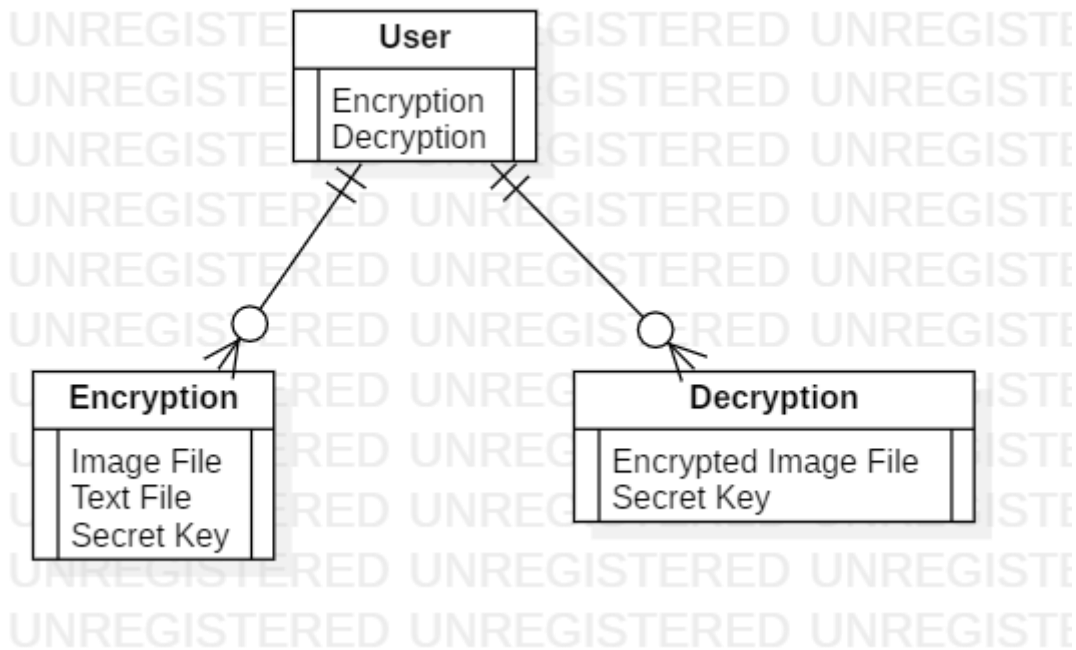


Figure: -Database Relationship diagram for Image Steganography

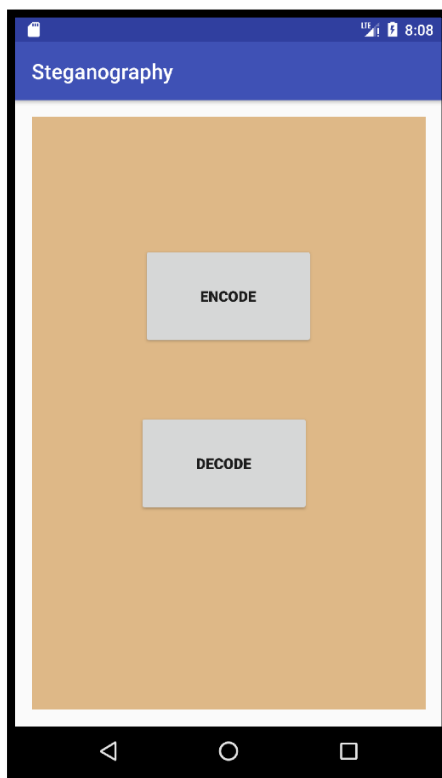
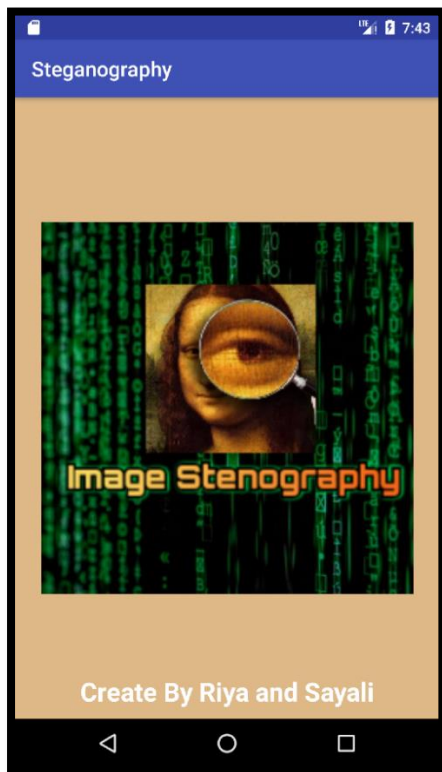
Assumptions:-

The following diagram is an example of Image Steganography Application. It describes the database relationship diagram, but our system has no database.

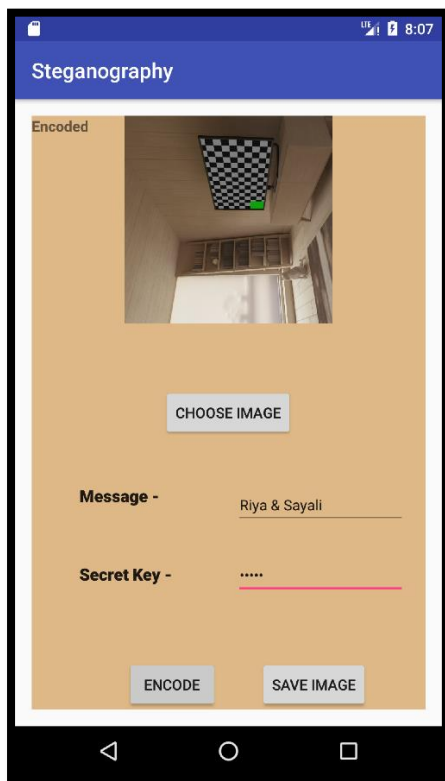
- First of all, User and Encryption, Decryption have a one-to-many relationship between each other.
- User are identified as the element of the system. They have a one-to-many relationship

3.2.4 I/O Screen Layout:-

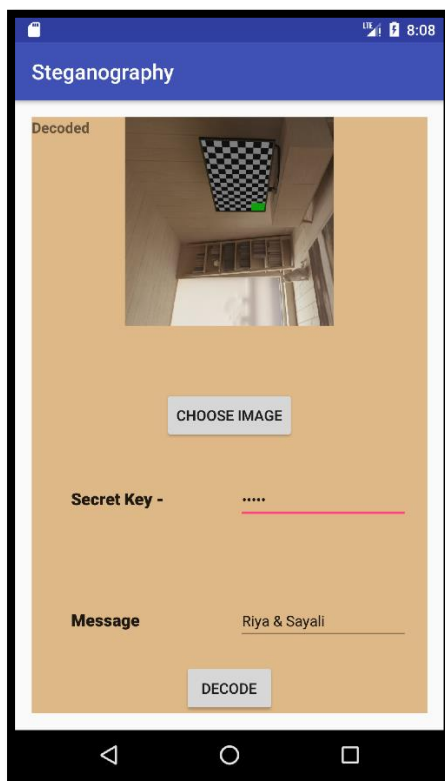
Splash Screen:-



Encryption Screen:-



Decryption Screen:-



Chapter 4: Testing

4.1 Methodologies used for testing

Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.

System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Testing Objectives:

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has a probability of finding an as yet undiscovered error
3. A successful test is one that uncovers an undiscovered error

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

- White box testing.
- Black box testing.

White-box testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Block-box testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information

domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

All these phases go through the process of software testing levels. There are mainly three testing levels which are as follows:

1. Unit Testing

2. Integration Testing

3. System Testing

- 1. Unit Testing:**

- A Unit is a smallest testable portion of system or application which can be compiled, linked, loaded, and executed. we tested each module separately.
- The aim is to test each part of the software by separating it. It checks that component are fulfilling functionalities or not. we tested all the modules which is working properly or not.

- 2. Integration Testing:**

- Integration means combining. In this testing phase, different application modules are combined and tested as a group to make sure that integrated system is ready for system testing.
- Integrating testing checks the data flow from one module to other modules. we tested all the activities in the manner of integration testing.

- 3. System Testing:**

- System testing is performed on a complete, integrated system. It allows checking system's compliance as per the requirements.
- It tests the overall interaction of components. It involves load, performance, reliability and security.
- System testing is the final testing to verify that the system meets the specification. It evaluates both functional and non-functional need for the testing.
- In system testing we checked all encryption and decryption process are workable, properly or not, and ensuring that the all activities are correct or not.

Test Cases:

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions on their true and false sides.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structure to assure their validity.

The test case specification for system testing has to be submitted for review before system testing commences.

Test Case table: -

1) Test Report:

Test Cases for User:

Test case ID	Test scenario	Operator action	Actual result	Remark
T01	Encryption	Enter Image, Text and secret key	Encrypted Image.	Test Successful.
T02	Decryption	Enter Encrypted Image, and secret key	Decrypted image with Text.	Test Successful.

Chapter 5: Conclusion

- To secure the private information or the confidential information from the attacks the proposed system is developed.
- Whenever we send the confidential files to the receiver there might be the chance to use this information by the third party person without the senders permission.
- For the solution for this there is cryptography and the steganography processes are invented.
- In the proposed system we used the Image steganography process, in that we used the image as a hiding source.
- We can encrypt as well as decrypt the file with the help of image.

Chapter 6: Future Enhancement

- This mobile application is used to secure the confidential or the private information from the other persons.
- It is very useful because of the main feature of this application i.e Secret key password which is known to sender and receiver only.
- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Because it is based on object-oriented design, any further changes can be easily adaptable.

Chapter 7: References

Website: -

Sr. No	Website Link	Visited date
1	www.Neonprojects.com	19 May 2022
2	www.w3schools.com	21 May 2022
3	www.stackoverflow.com	1 June 2022
4	www.uml.com	5 June 2022
5	https://github.com	18 June 2022