A
PROJECT REPORT
ON
"Sopho-Wrap: Image Steganography Tool"
B.Tech (CE) SEM-VI Computer Engineering

SUBMITTED BY
1) Mansi.N. Patel (CE-091)
2) Meghana.J. Patel (CE-093)

Under the Guidance of
Prof. M.S. Bhatt

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF TECHNOLOGY
DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD- 387001

CERTIFICATE

This is to certify that the project carried out in the subject of System Design Practices entitled **"Sopho-Wrap: Image Steganoghraphy Tool"** and the record in this report is a work of

Mansi.N. Patel, ROLLNO: CE-091 and ID: 12CEUOS076
Meghana.J. Patel, ROLLNO: CE-093 and ID: 12CEUOG078

of Department of Computer Engineering, Semester VI. They were involved in Project developing during academic year 2014 -2015.

Prof. Malay S. Bhatt
Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University, Nadiad

Prof. C.K.Bhensdadia,
Head, Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University, Nadiad

Acknowledgement

With immense pleasure and commitment we would like to present the project assignment. The nature of project on the development **Sopho-Wrap: Image Steganographic Tool (SWIST)** has given us wide opportunity to think, implement and interact with various aspects of management skills as well as the new emerging facilities and the technology used in architecture and the enhancements given to the students with a boon of spirituality and curricular activities.

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us during this project. This project would not have been materialized without the help from many who asked us good questions and rescued from various red tape crisis.

Theoretical knowledge is of no importance if one doesn't know the way of its implementation. We are thankful to our institute that provided us an opportunity to apply our theoretical knowledge through the project. We feel obliged in submitting this project as part of our curriculum.

We would like to take the opportunity to express our humble gratitude to our guide Prof. Malay Bhatt, under whom we undertook our project. His constant guidance and willingness to share his vast knowledge made us enhance our knowledge and helped us to complete the assigned tasks to perfection. Without his effort, support and an astonishing testing ability this project may not have succeeded.

We are sincerely thankful to Head of CE department, Prof. C.K.Bhensdadia for the unconditional and an unbiased support during the whole session of study and development.

We would also like to thank our institute DDU and respected Vice Chancellor Sir who altogether provide us favourable environment, without them we would not have achieved our goal.

With Sincere Regards,
Mansi Patel
Meghana Patel

INDEX

1. Software Requirement Specification

Our project is Simple Windows form based application.

1.1 Purpose

The purpose of this requirement document is to clearly specify the needs of the users for Image Steganography system. It allows the user to securely transfer a text message by hiding it in the image file such that the intended secret message does not attract attention to itself as an object of scrutiny. It also use as handbook for business analysts, test engineers and project managers. This document will be the major document that will be referred by all the team members involved in this project. This document is also helpful to developers as one hand and validation document for end users. Developers should consult this document and its revisions as the only source of requirements for the project. They should not consider any requirements statements, written or verbal as valid until they appear in this document or its revision.

1.2 Scope:

In Scope:
a. To provide the platform of secured communication.
b. To facilitate encryption of text into image file.
c. To facilitate decryption of encrypted image file into its original text file conveniently.

Not In scope:
This system not provides facility for any other Image Steganography other than *.bmp extensions or any other Steganography (image to image, audio to image, etc.).

1.3 Definitions, acronyms and abbreviation

msg –message
***.bmp** –Bitmap Image File
***.txt** –Text File
SWIST –Sopho-Wrap: Image Steganoghraphy Tool

2. Hardware Requirement:

Any system with Windows 7, Windows 8, and Windows 8.1 supports the project.
Configuration:-1.8 GHz, 256MB Ram, 80 GB HDD, 15”TFT or CRT monitor, Optical Mouse, MM Key board, Serial & Parallel port.

- **Hardware Interfaces**

  The system must basically support certain input and output devices. Their descriptions are as follows :-

	Description of Purpose	Source of Input/output
Mouse	To accept data from user like file	Source of input
Keyboard	To accept file name from user for compression and decompression	Source of input

3. Non-Functional Requirements:

3.1 Reliability

- SWIST shall be available 24 hours a day, 7 days a week.
- SWIST shall be robust enough to have a high degree of fault tolerance. The system should not crash in case of invalid input and shall identify the invalid input and produce a suitable error message.
- SWIST shall be able to recover from hardware failures, power failures and other natural catastrophes.

3.2 Usability

SWIST shall provide an easy-to-use graphical interface similar to other existing systems so that the users do not have to learn a new style of interaction.

- The GUI interface shall be intuitive and easily navigable. Users shall comfortably be able to understand the menu and options provided by SWIST.
- Any notification or error messages generated by SWIST shall be clear and polite.

3.3 Integrity

Only system administer has the right to change system parameters, such as pricing policy etc. The system should be secure.

3.4 Availability

The factors guarantee that the software's availability shall include the proper termination and correct input details.

3.5 Maintainability

The software will be developed by implementing the concept of modularity which in turn reduces the complexity involved in maintaining it. The administrator should have a sound technical knowledge about maintaining the software and further enhancements will be undertaken by the developer.

3.6 Portability

The application is portable which ensures its adaptability for use

4. Functional Requirement:

R1: LSB ALGORITHAM

R1.1 Encryption of file

R1.1.1 Select Cover Image (*.bmp)

Input: File with .bmp extension
Output: Display in Image Preview

R1.1.2 Select Text File

Input: Text file

R1.1.3 Change LSB

R1.1.3.1-Get Pixel Value

Input: Image URL
Output: Pixel value

R1.1.3.2-Get LSB of each Pixel

Input: Pixel value
Output: LSB of each pixel

R1.1.3.3-Get Binary ASCII of Text File

Input: URL of text file
Output: Binary ASCII of entire text file.

R1.1.3.4-Perform LSB change Operation

Input: Pixel value, binary ASCII of char, LSB of each pixel
Output: Resultant value of every pixel's LSB changed

R1.1.3.5-Apply Modification

Input: Resultant value of every pixel's LSB
Output: Encrypted stego-image

R1.2 Decryption of file

R1.2.1 Select Encrypted Image

Input: Encrypted file
Output: Display in Image Preview

R1.2.2 Select Destination File

Input: File location

R1.2.3 Extract Message

Description: Each decoded char is written into destination file by extracting from stego image using LSB algorithm.

R1.2.3.1 -Get Pixel value of Encrypted Image

Input: Encrypted image
Output: Pixel values of stego-image

R1.2.3.2 -Find LSB of each Pixel

Input: Pixel value of stego-image

Output: binary array

R1.2.3.3 -Conversion from Binary to ASCII Char

Input: Binary array
Output: ASCII char

R1.2.3.4 -Extract Text File

Input: Binary ASCII message stream
Output: Text file included message

R2: Change B Byte ALGORITHAM

R2.1 Encryption of File

R2.1.1 Select Cover Image (*.bmp)

Input: File with .bmp extension
Output: Display in Image Preview

R2.1.2 Select Text File

Input: Text file

R2.1.3 Change B byte

R2.1.3.1-Get Pixel Value

Input: Image URL
Output: Pixel value

R2.1.3.2-Get B of each Pixel

Input: Pixel value
Output: B byte of each pixel

R2.1.3.3-Get Binary ASCII of Char

Input: URL of text file

Output: Binary ASCII of char

R2.1.3.4-Perform Operation

Input: Pixel value, binary ASCII of char, B byte of each pixel
Output: Resultant value of every pixel's B byte

R2.1.3.5-Apply Modification

Input: Resultant value of every pixel's B byte

Output: Encrypted image

R2.2 Decryption of File

R2.2.1 Select Encrypted Image

Input: Encrypted file
Output: Display in Image Preview

R2.2.2 Select Destination File

Input: File location

R1.2.3 Extract Message

Description: Each decoded char is written into destination file by extracting from stego image using Change B bit algorithm.

R1.2.3.1 -Get Pixel value of Encrypted image

Input: Encrypted image
Output: Pixel values of stego-image

R1.2.3.2 –Find B byte of each pixel

Input: Pixel value of stego-image
Output: Binary array

R1.2.3.3 - Conversion from Binary to ASCII Char

Input: Binary array
Output: ASCII char

R1.2.3.4 -Extract Text File

Input: Binary ASCII message stream
Output: Text file included message

5. USER CHARCTERISTICS

5.1 Knowledge

Any user at any level, operate the system that requires the brief knowledge of the computer and should be able to understand English.
No experience is required for the system user.

5.2 GUI Features:

- The GUI is very Simple and understandable for any nontechnical person.
- Offers robust save options for complete format control of all supported formats.
- The GUI provides the options for the user to select a particular technique through an interactive pull down menu option.
- The parameters related to compressed file and Extracted file like size of file is displayed in GUI.
- GUI provides facility to open extracted file in default application.

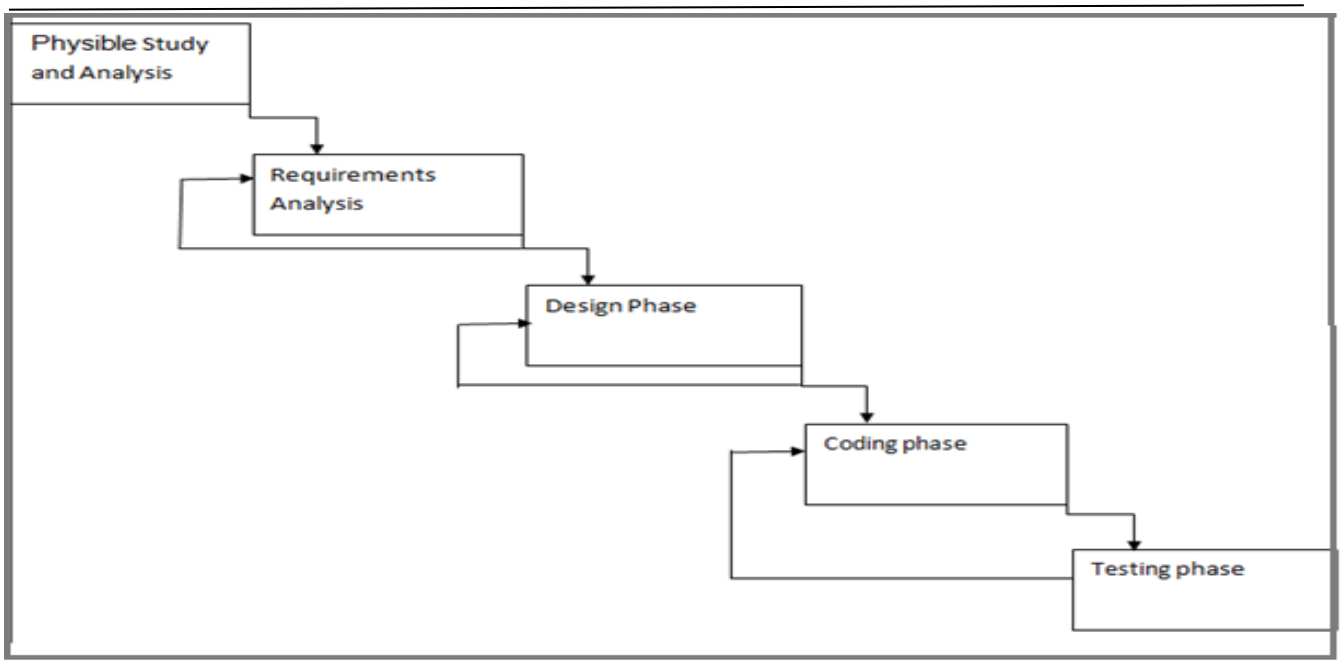
6. Software Process Model

Here for our System the process model which can be well suitable is Iterative Waterfall Model.

In this system we know that what risk can be there so for that this model is suitable. Also this model is easy to understand for inexperienced person. Here, by chance if we have forgotten to add some functionality then also we can add them during any other phase because we can go back to any previous phase, i.e. we have found problem in coding phase then we can correct it by going back to design phase.

7. TimeLine Chart

31/1/15 to				



8. ANALYSIS

Our project Sopho-Wrap: Image Steganography Tool is created using c# higher level language with the help of .NET Frame work.

Analysis of Steganoghraphic technique:

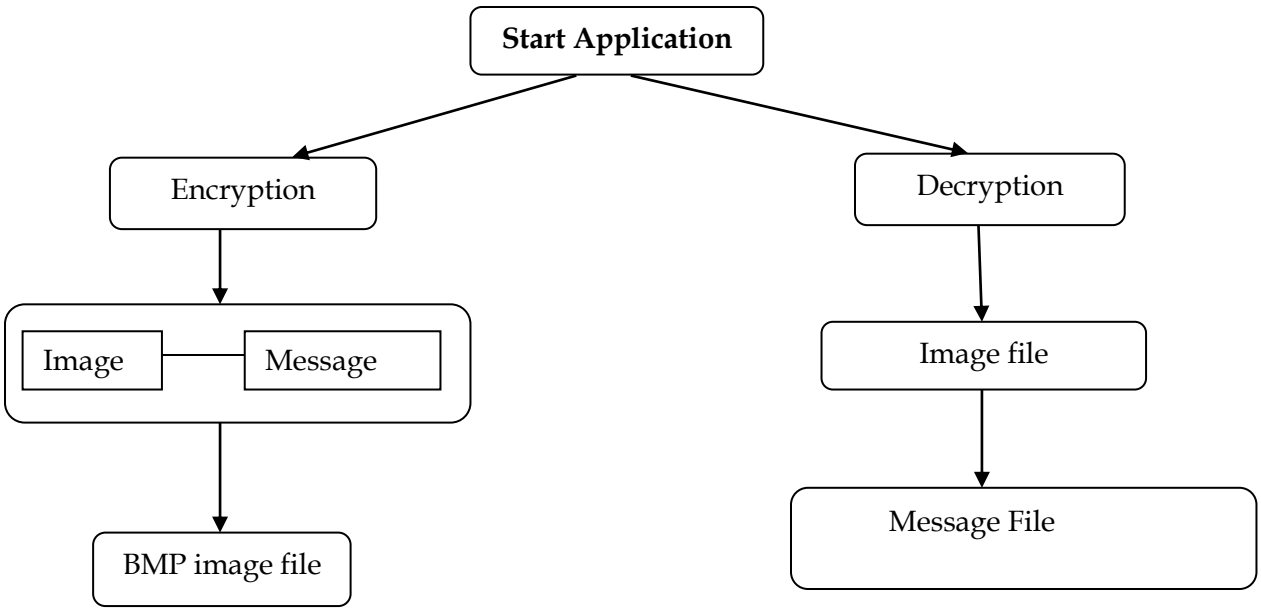
Steganography system requires any type of image file and the information or message that is to be hidden. It has two modules encrypt and decrypt.

Microsoft .Net framework prepares a huge amount of tool and options for programmers that they simples programming. One of .Net tools for pictures and images is auto-converting most types of pictures to BMP format.

The encrypt module is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in destination.

The decrypt module is used to get the hidden information in an image file. It take the image file as an output, and give two file at destination folder, one is the same image file and another is the message file that is hidden it that.

Before encrypting file inside image we must save name and size of file in a definite place of image. We could save file name before file information in LSB layer and save file size and file name size in most right-down pixels of image. Writing this information is needed to retrieve file from encrypted image in decryption state.



In our project we have used two different steganoghraphic techniques:

1) LSB algorithm

2) B Change algorithm

Both techniques are based on a lossless compression.

Lossless data compression is a class of data compression algorithms that allows the original data to be perfectly reconstructed from the compressed data.

1) LSB Algorithm:

It uses a specific method for changing the last bit of each pixel. As a normal pixel is of 24 bit size, it replaces the least significant bit of each pixel with the adequate corresponding bit of ascii character form the message file simultaneously.

Thus, least significant bit (LSB) insertion is a simple approach to embedding information in image file. The simplest steganography techniques embed the bits of the message directly into least significant bit plane of the cover-image in a deterministic sequence. Modulating the least significant bit does not result in human perceptible difference because the amplitude of the change is small. In this technique, the embedding capacity can be increased by using two or more least significant bits. At the same time, not only the risk of making the embedded message statistically detectable increase but also the image fidelity degrades. The advantage of LSB-based method is easy to implement and high message pay-load.

Although LSB hides the message in such way that the humans do not perceive it, it is still possible for the opponent to retrieve the message due to the simplicity of the technique. Therefore, malicious people can easily try to extract the message from the beginning of the image if they are suspicious that there exists secret information that was embedded in the image.

2) B Change algorithm:

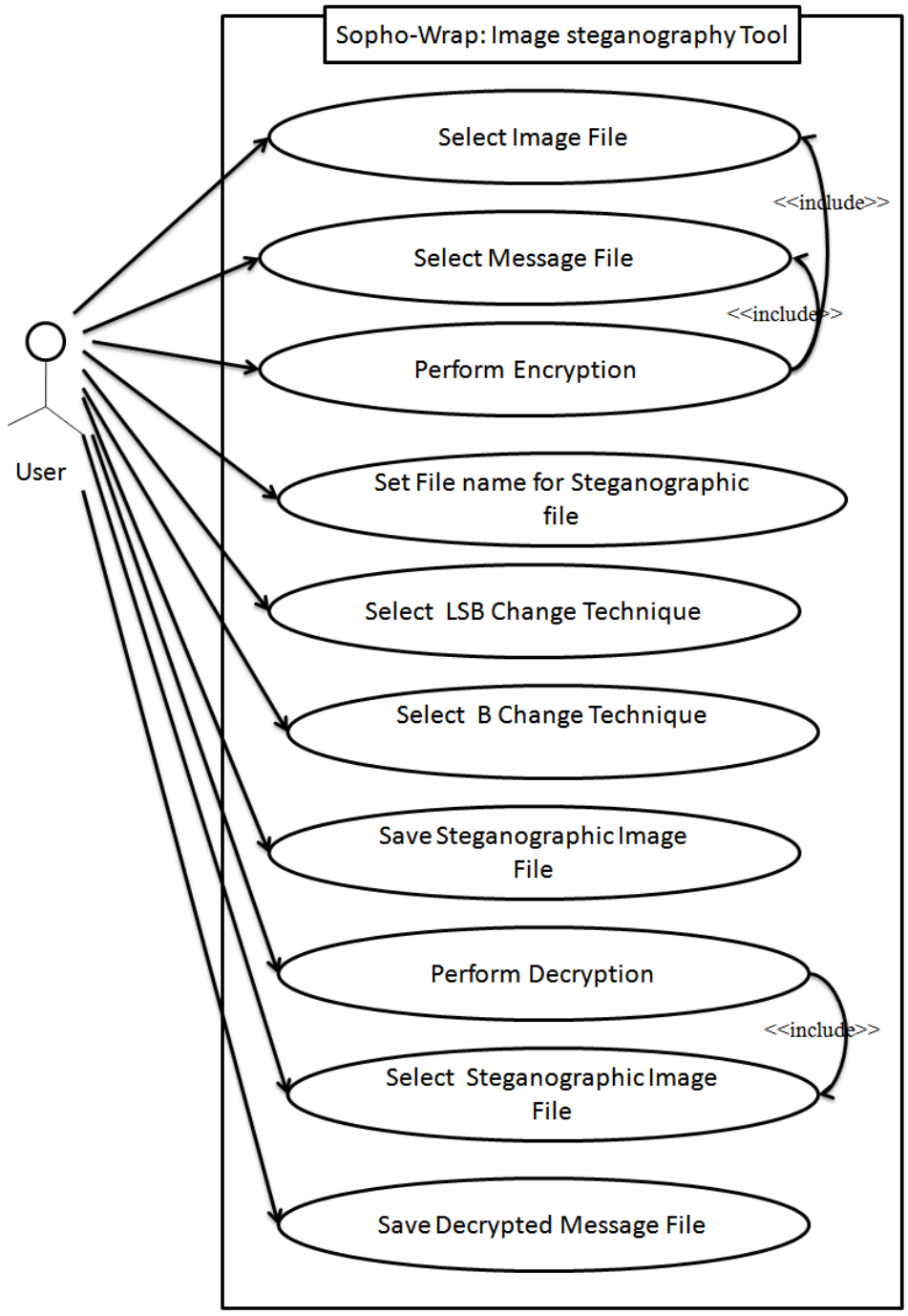
It is a very simple form of image steganographic technique in the Blue color byte of each and every pixel under requirement is changed and replaced by each byte of individual ASCII character of the message file.

Again this technique benefits to be advantageous for the disability of an human eye to be not able to recognize the small change in the image file after performing the steganography to it.

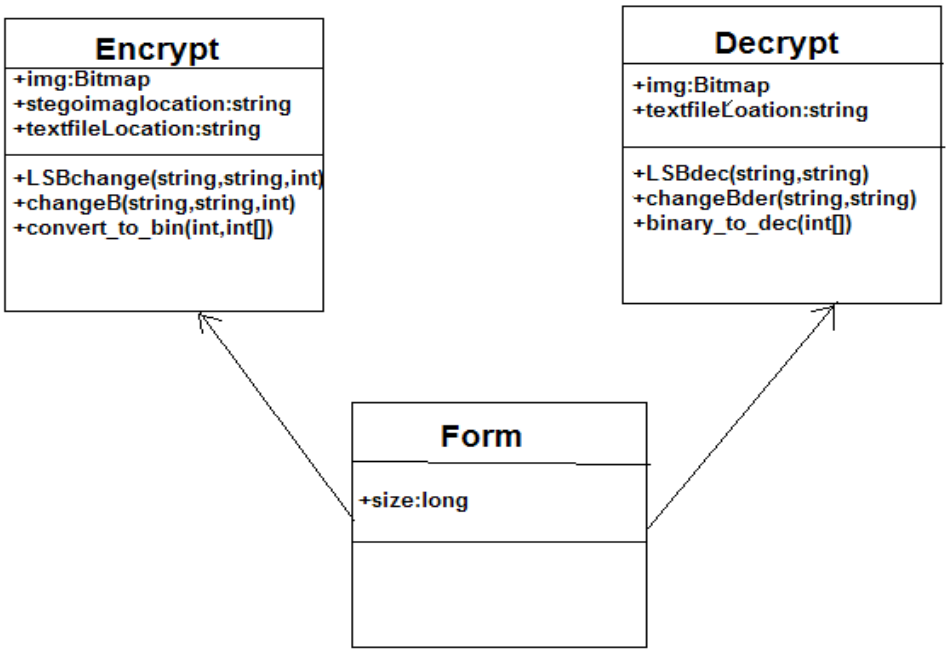
Though, this practice can be easily caught up by any person having some basic knowledge of steganoghraphy. As it directly replaces the whole blue bit of the individual pixel of the image.

9. Design

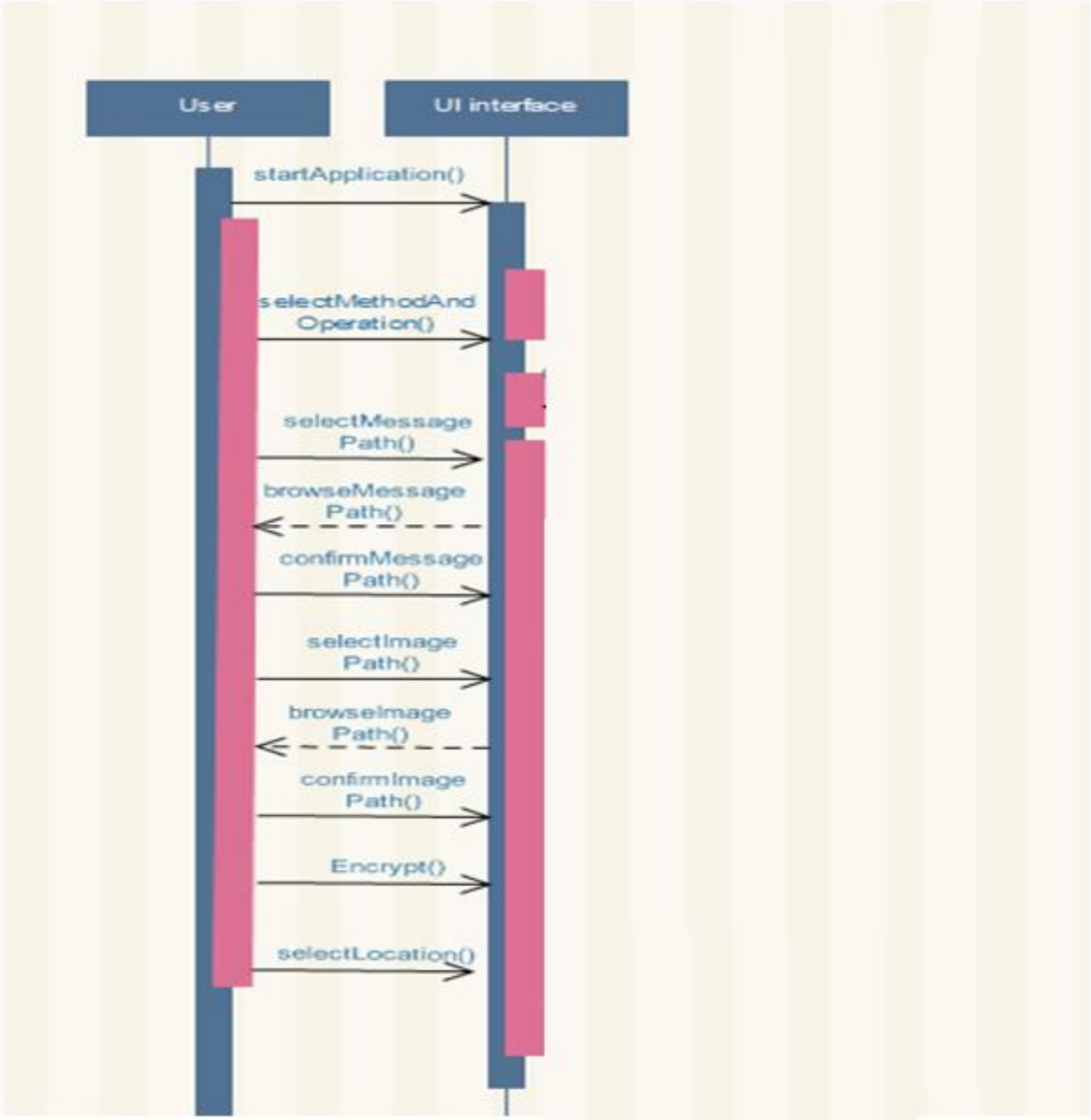
9.1 Uses-Case Diagram



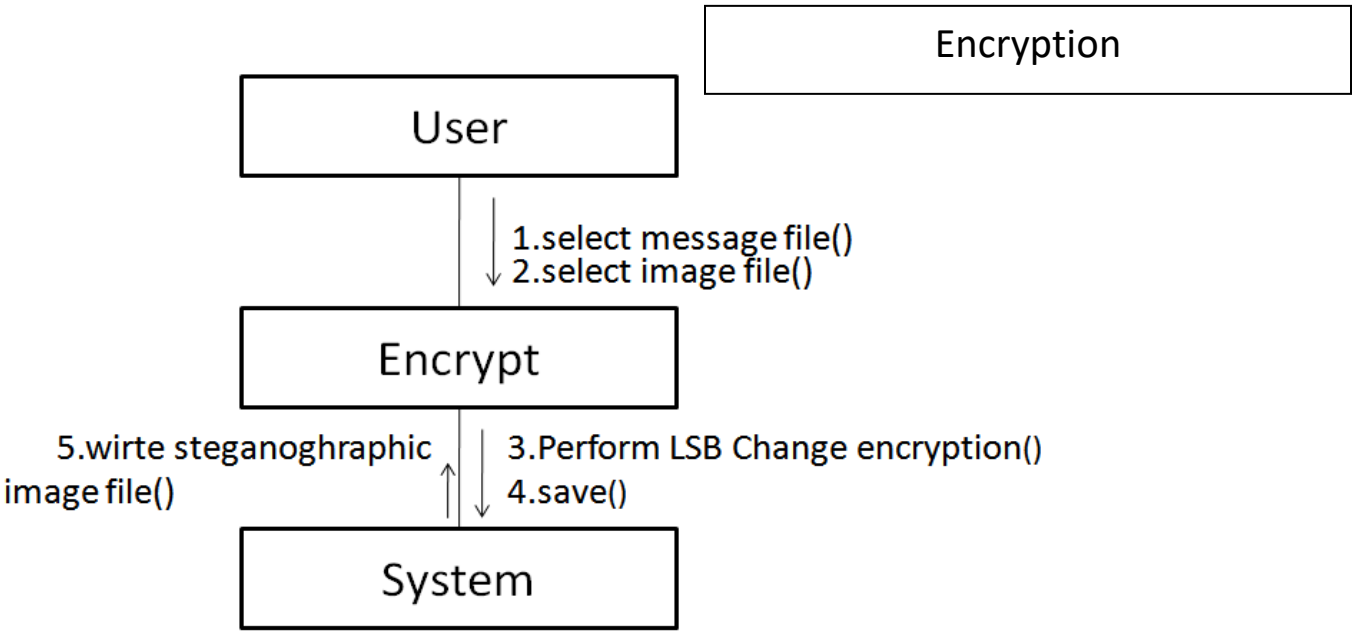
9.2 Class diagram

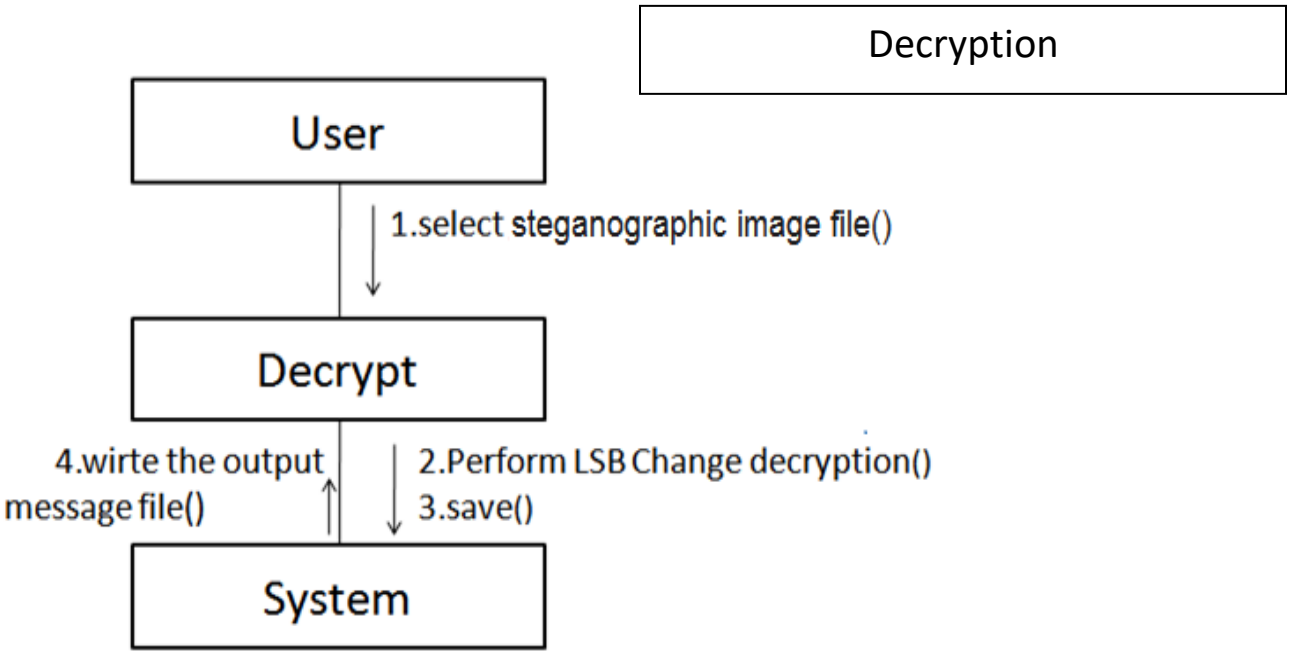


9.3 Sequence diagram



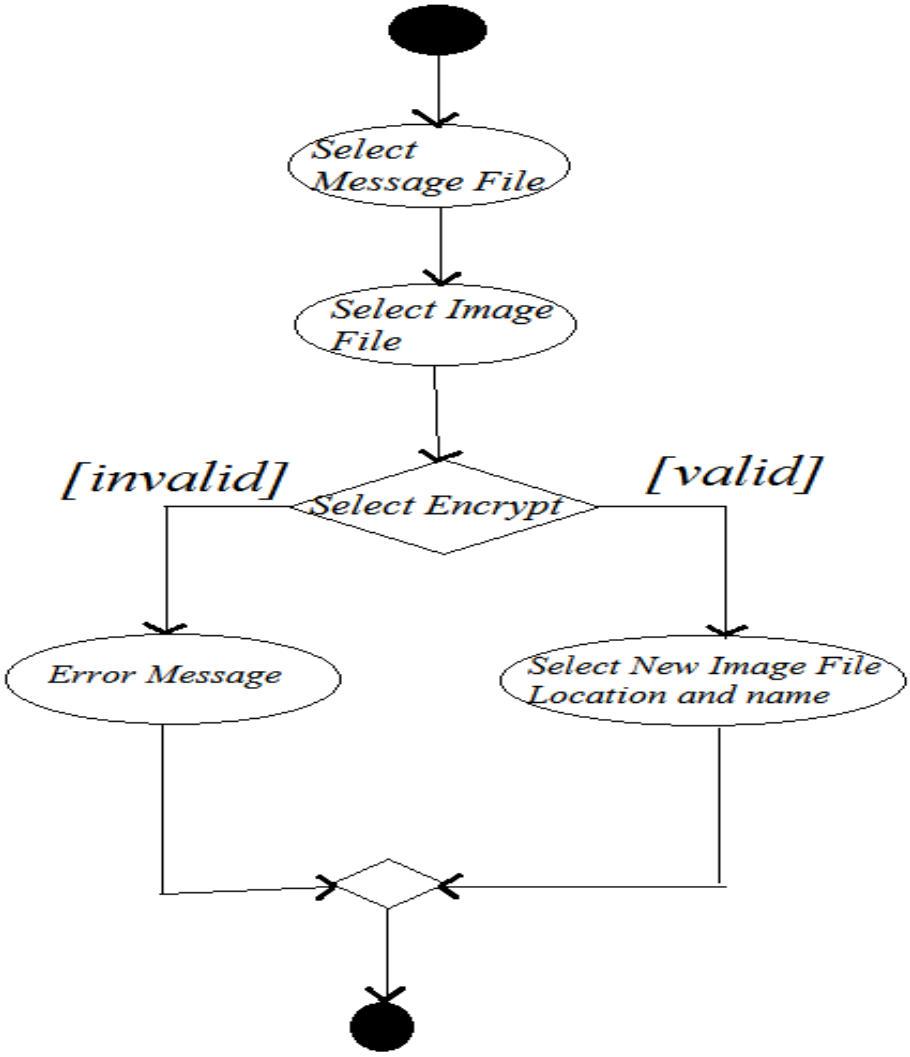
9.4 Collaboration diagram

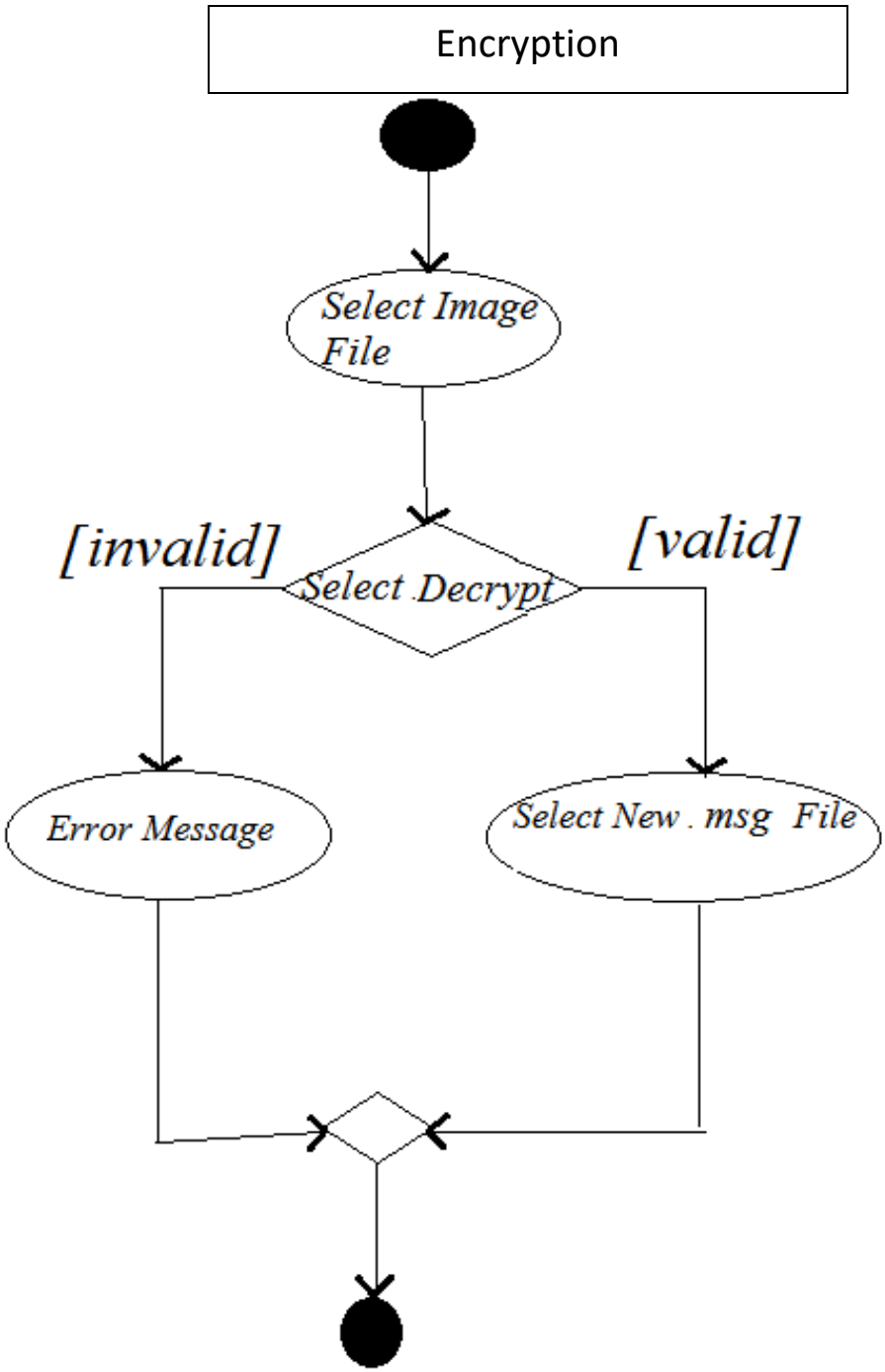




9.4 Activity diagram

User Perspective:




Encryption
Select Image File
[invalid]
Select .Decrypt
[valid]
Error Message
Select New . msg File

Decryption

10 Implementation:

.NET Frame Work is provided by MICROSOFT.

.NET has a well defined DOM for its language and constantly generating things dynamically at run-time is a lot simpler and more efficient.

Windows is the dominating Operating System on client computers. The best GUI frameworks for Windows applications are Win forms and WPF together with .NET Framework. The best programming language to work with the .NET Framework and its APIs is C#. Java is also an alternative for this. And C++ is an older language without automatic memory management. C# is similar to C++ but has automatic memory management and you don't have to work with pointers.

C# is a very good language if:

We want to do general purpose object oriented development. It's a classic, statically typed OOP language.

We are targeting Microsoft platforms only .C# as a language is nicer than Java in various ways (better syntax for properties, value types, reified generics etc.).

C# has native garbage-collection and it creates more dynamic and flexible relationships between classes. Here, interactions between classes are required in order to create an appropriate, clear, understandable solution of the problem.

It also contains huge standard library with so much useful stuff that’s well-implemented and easy to use.

Instead of a lot of noise like EJB, private static class implementations, etc…which are generally used by JAVA, here we get elegant and friendly native constructs such as Properties and Events.

It's deeply integrated with Windows.

It has dynamic variables support.

Encryption:

Class Name	Method Name	Description
Encrypt	LSBchange	It replaces the least significant bit of every pixel with the corresponding individual bit of ascii character from the message file.
Encrypt	changeB	It replaces Blue byte of every pixel with the corresponding byte of ascii character from the message file.
Encrypt	Convert_to_bin	It converts every character of the message file into its equivalent ascii value in binary.

Decryption:

Class Name	Method Name	Description
Decrypt	LSBdec	It extracts the least significant bit of every pixel with the corresponding individual bit of character to the message file.
Decrypt	changeBdecr	It extracts Blue byte of every pixel with the corresponding byte of character to the message file.
Decrypt	BinaryTo_decimal	It converts equivalent ascii value in binary to decimal ascii.

11. Test Case

E1= <File which is encrypted in LSB Change method>

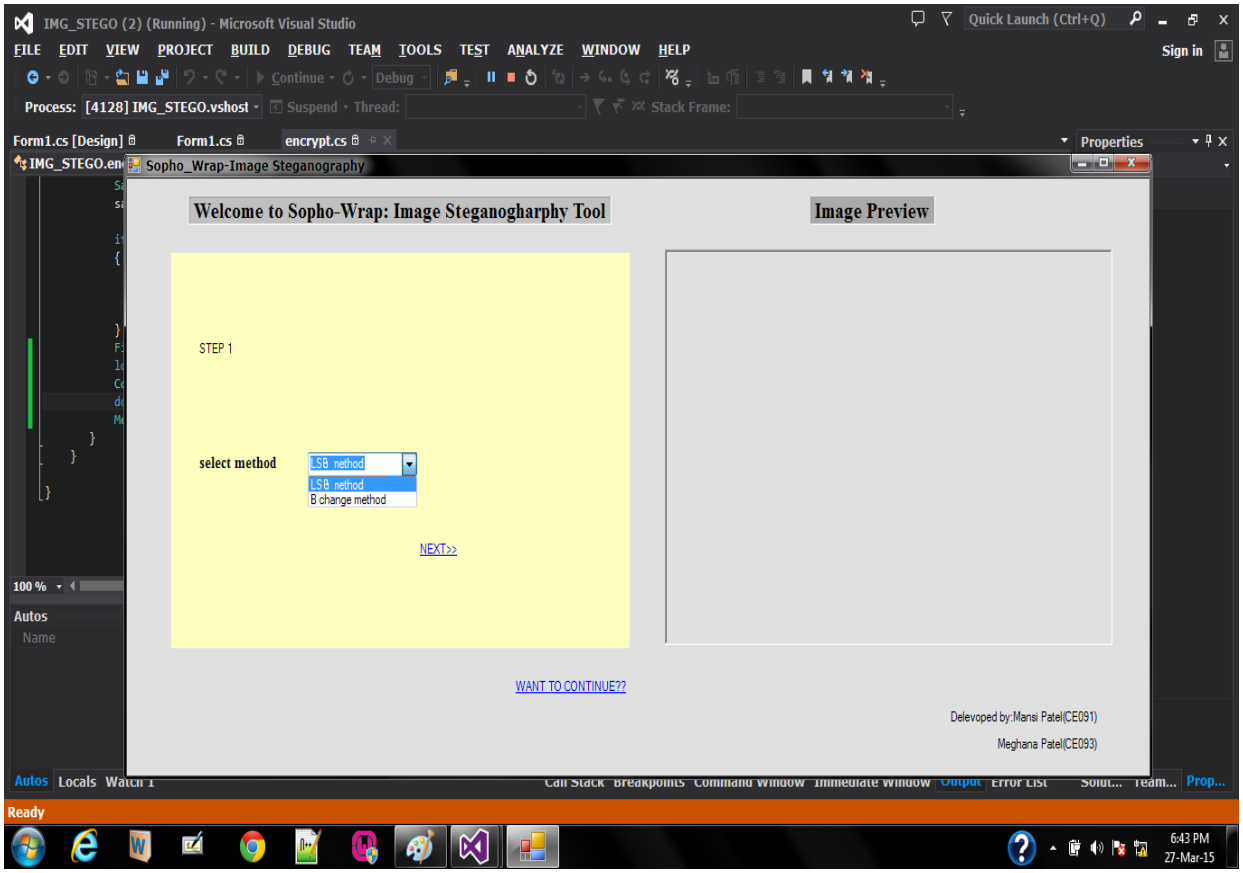
E2= <File which is encrypted by B Change method>

E3= <File which is not encrypted>

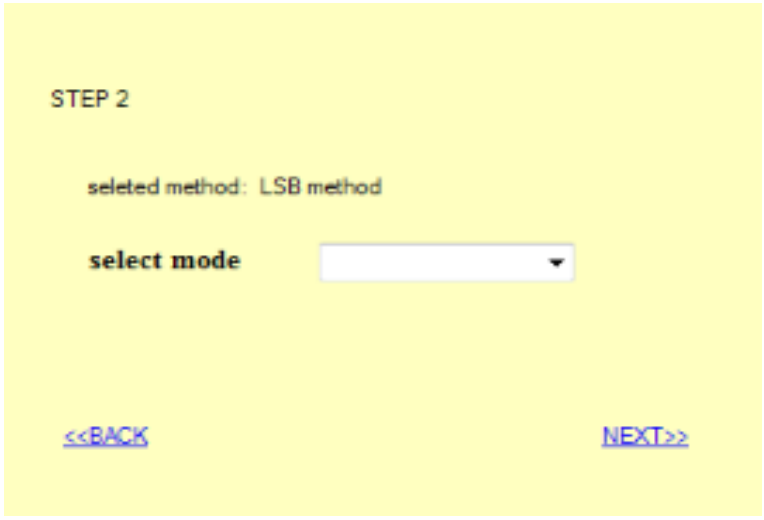
Black Box Test case of Sopho-Wrap: Image Steganoghraphy Tool:

Index	Equivalent Class	Input	Expected output	Actual Output
1.	I1<-E3	a.txt, img.bmp (no selection of LSB/B change method) User clicks on Encrypt button	Error	Error
2.	I2<-E1	a.txt, img.bmp (text size <255 bytes and image size >11KB) (LSB change method) User clicks on Encrypt button	Encryption of file	Encryption of file
3.	I3<-E2	a.txt, img.bmp (text size <255 bytes and image size >11KB) (B change method) User clicks on Encrypt button	Encryption of file	Encryption of file
4.	I4<-E1	a.txt, img.bmp (text size <255 bytes and image size >11KB) (LSB change method) User clicks on Encrypt button	Encryption of file	Size of Input Files size does not satisfy the constraint.
5.	I5<-E2	a.txt, img.bmp (text size <255 bytes and image size >11KB) (B change method) User clicks on Encrypt button	Encryption of file	Size of Input Files size does not satisfy the constraint.
6.	I6 <-E1	S_img.bmp (LSB change technique) User clicks on decrypt button	Decryption Of file	Decryption Of file
7.	I7<-E2	S_img.bmp (LSB change technique) User clicks on decrypt button	Decryption Of file	Decryption Of file
8.	I8<- E3	S_img.bmp (not selection of LSB/B change technique) User clicks on decrypt button	Decryption Of file	Error

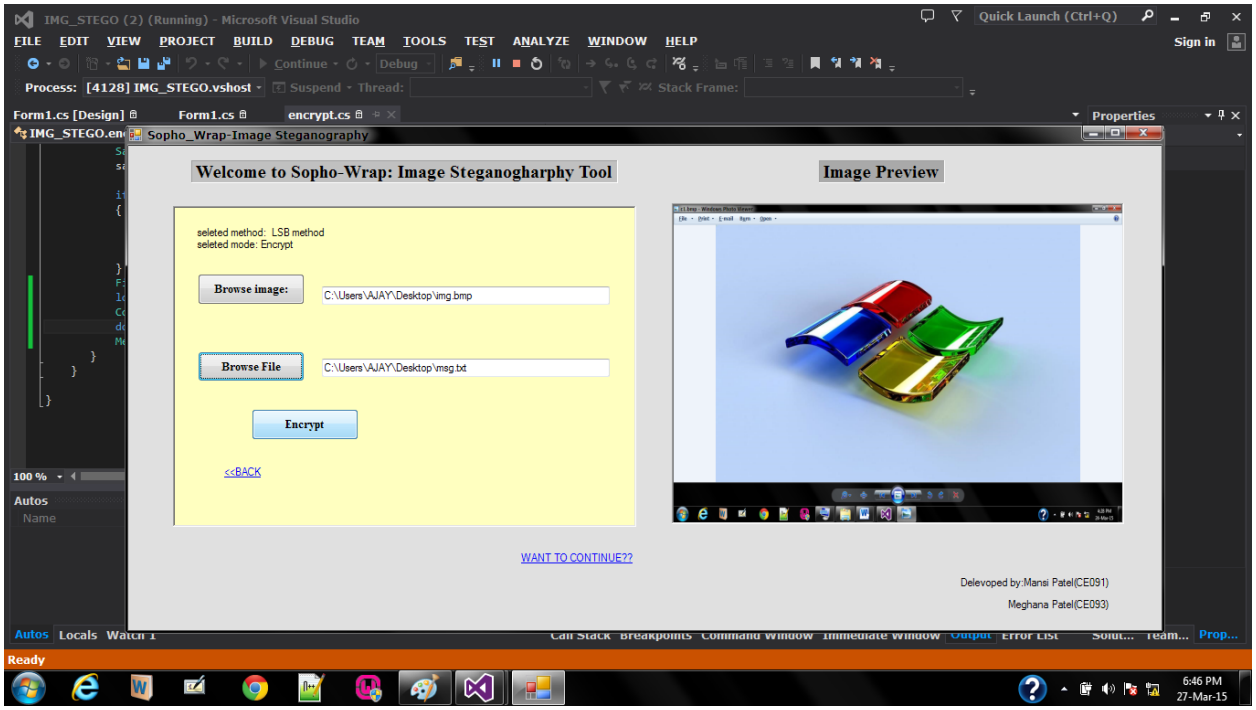
12. Screen Layouts:



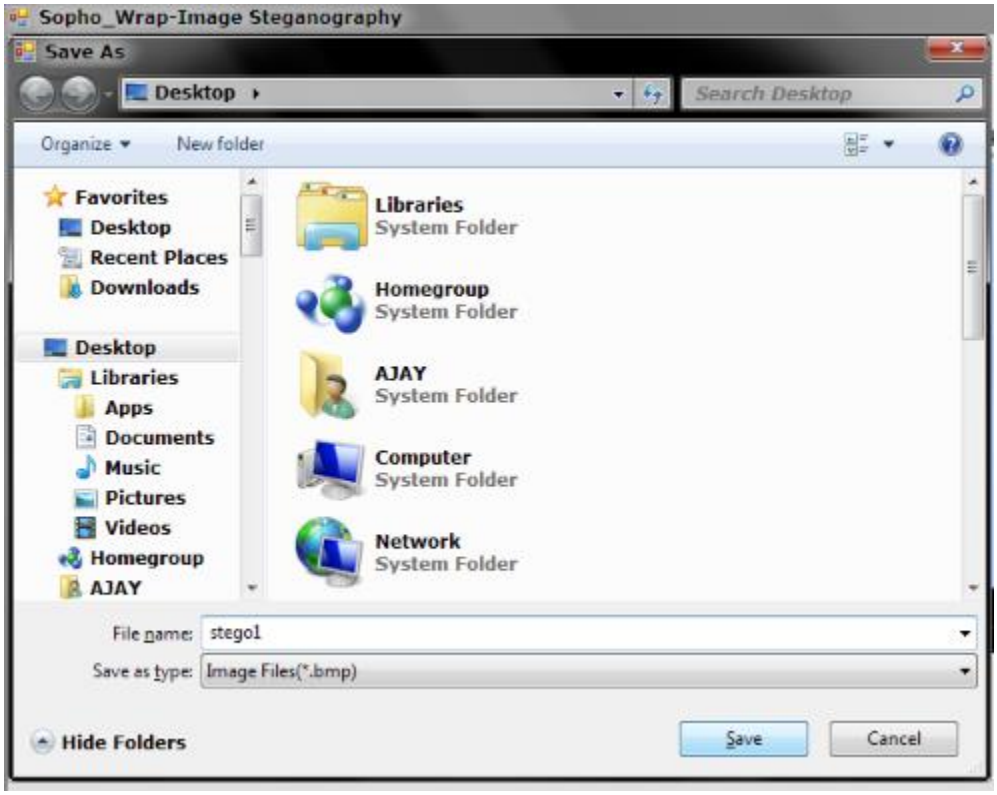
(I) Method Selection.



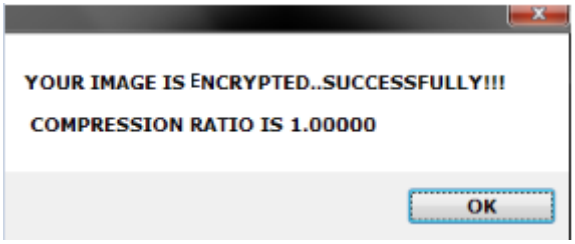
(II) Mode Selection.



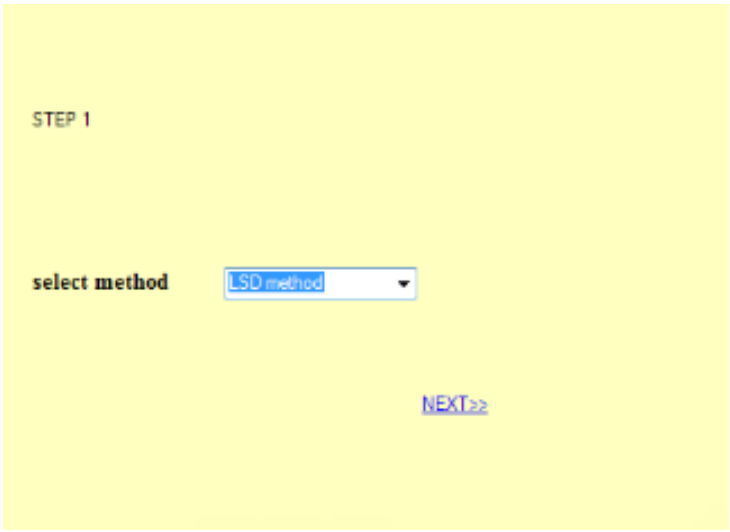
(III) Message and Image Selection.



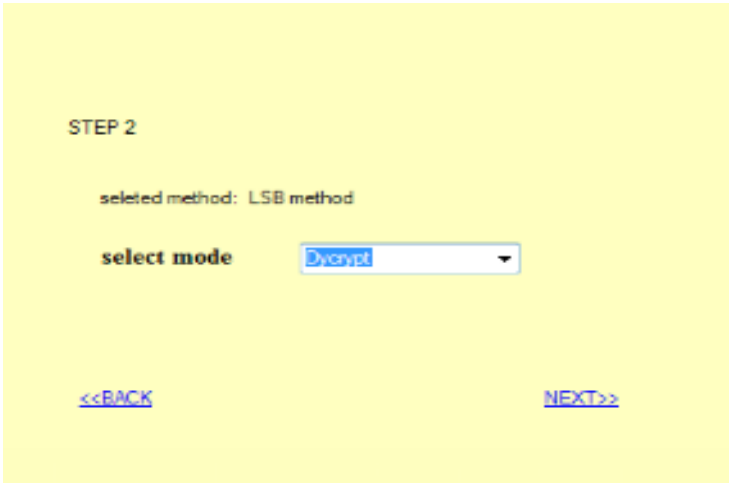
(IV) Steganoghraphic Image and Location Selection.



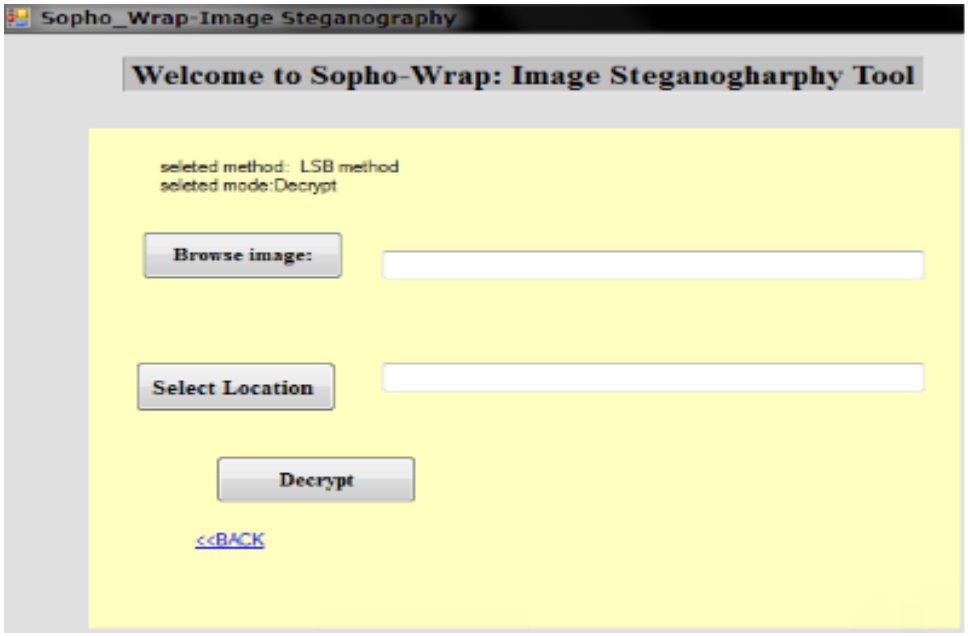
(V) Confirmation of Encryption and displays the computed compression ratio of simple image to the steganographic image. Close the application.



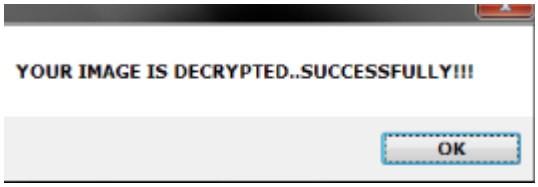
(VI) Method Selection.



(VII) Mode Selection.

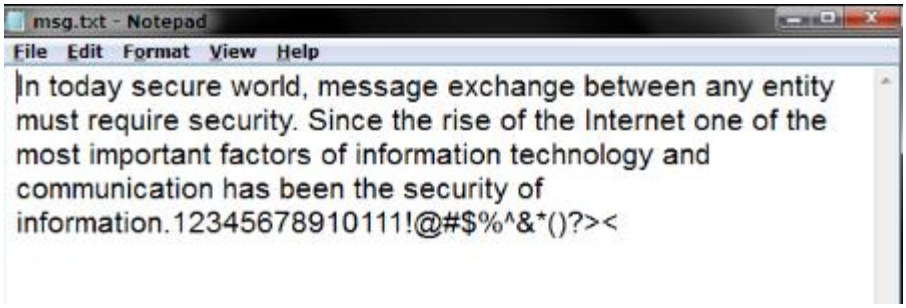


(VIII) Steganoghraphic Image and New Message File Selection.

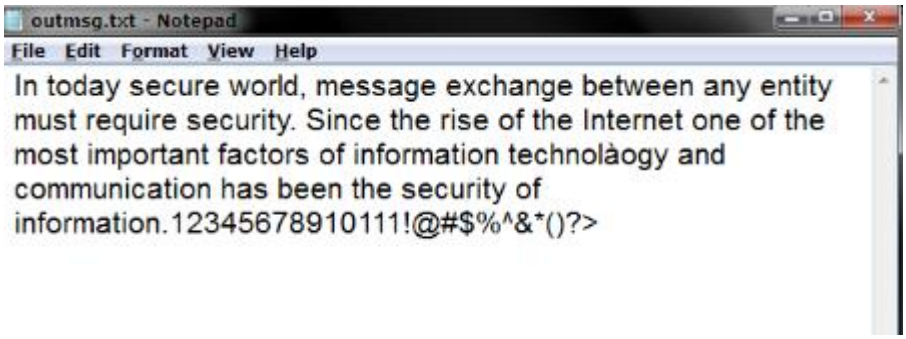


(IX) Confirmation of Successful Decryption.

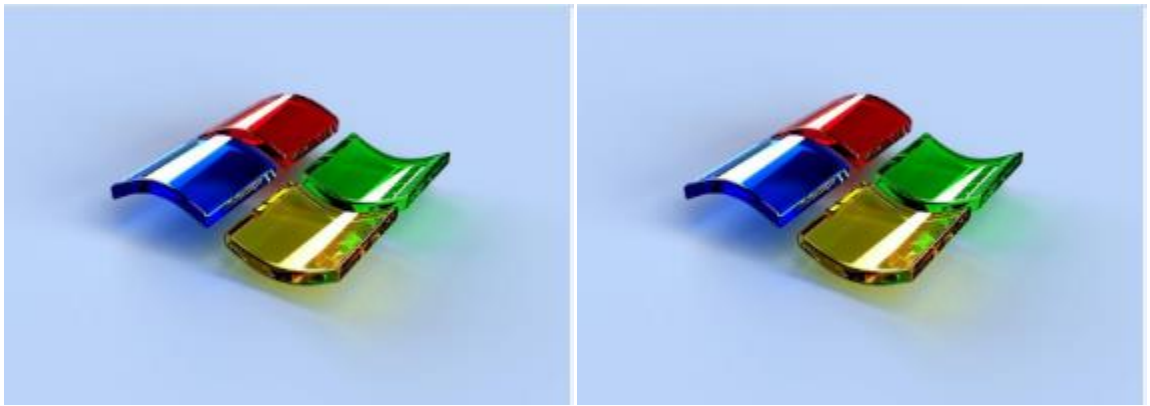
- **Message before performing steganoghraphy.**



- **Message after performing steganoghraphy.**



- **Image before and after performing steganoghraphy.**



13. Conclusion:

We have implemented Sopho-Wrap: Image Steganoghraphic Tool aimed to have secured communication. This Tool can be used for hiding the text message in the image. Also, the message which is to be concealed is in an encrypted form, so as to support safe steganoghraphy. Simultaneously, the decryption of that image would be difficult for an eaves dropper.

14. Future Extension:

Now a day, image steganography is broadly used in steganography field. So there is lot to do as per improvement is concerned. Time Complexity of these algorithms can be reduced. We can use the algorithms, B change and LSB change with other cryptographic algorithms and steganographic algorithms which can reduce the space and time complexity and more vitally increase the sustained level of security. Moreover, the scope to be extended is high level of complex algorithms which cannot be easily disclosed by novice users. As the ultimate goal for Image Steganography, is to preserve the security purpose.

14. Bibliography:

Websites:

1) www.msdn.microsoft.com
2) www.stackoverflow.com
3) www.wikipedia.org
4) www.codeproject.com
5) www.programmertoprogrammer.com