A Project-I
reportOn

IMAGE STEGANOGRAPHY USING LSB (LEAST SIGNIFICANT BIT) TECHNIQUE

Submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

Submitted by

Baddam Himabindu(17311A05D2)
Cheedu Sowmya(17311A05D4)
Mannem Nagalaxmi(17311A05F1)

Under the Guidance of
S. Dheeraj
Assistant Professor
Department of CSE

Department of Computer Science & Engineering
SREENIDHI INSTITUTE OF SCIENCE ANDTECHNOLOGY
(AUTONOMUS)
Yamnampet (V), Ghatkesar (M), Hyderabad – 501301, A.P.
2019-2020

Department of Computer Science & Engineering

SREENIDHI INSTITUTE OF SCIENCE ANDTECHNOLOGY

(AUTONOMUS)

Yamnampet (V), Ghatkesar (M), Hyderabad – 501301, A.P.

2019-2020

CERTIFICATE

This is to certify that the project entitled "**Image Steganography Using LSB (Least Significant Bit) Technique**" is being submitted by

Baddam Himabindu	**(17311A05D2)**
Cheedu Sowmya	**(17311A05D4)**
Mannem Nagalxmi	**(17311A05F1)**

in partial fulfillment of the requirements for the award of **BACHELOR OF TECHNOLOGY** to **SNIST, Hyderabad**. This record is a bonafide work carried out by themunder my guidance and supervision. The result embodied in this project report has not been submitted to any other university or institute for the award of any degree of diploma.

Internal guide	**Project Co-ordinator**	**Head of Department**
Mr. S. Dheeraj	Dr. Preethi Jeevan	Dr. Aruna Varanasi
Assistant Professor	Assistant Professor	Professor & HOD
Department of CSE	Department of CSE	Department of CSE

External Examiner:

Date:

DECLARATION

We, **Baddam Himabindu (17311A05D2), Cheedu Sowmya (17311A05D4) and Mannem Nagalxmi (17311A05F1),** students of **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY, YAMNAMPET, GHATKESAR,** studying 4th year 1st semester, **COMPUTER SCIENCE AND ENGINEERING** solemnly declare that the project work, titled **"Image Steganography Using LSB(Least Significant Bit) Technique"**is submitted to **SREENIDHI INSTITUTE OF SCIENCE AND TECHNOLOGY**

for partial fulfillment for the award of degree of Bachelor of technology in **COMPUTER SCIENCE AND ENGINEERING**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

ACKNOWLEDGEMENT

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I would like to express my heart-felt gratitude to my parents without whom I would not have been privileged to achieve and fulfill my dreams. I am grateful to our principal, **Dr. T. Ch. Siva Reddy,** who most ably run the institution and has had the major hand in enabling me to do my project.

I profoundly thank Dr. **ARUNA VARANASI**, Head of the Department of Computer Science & Engineering who has been an excellent guide and also a great source of inspiration to my work.

I would like to thank my internal guide **Mr. S. Dheeraj** for his technical guidance, constant encouragement and support in carrying out my project at college.

The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, I would like thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task.

Baddam Himabindu	**17311A05D2**
Cheedu Sowmya	**17311A05D4**
Mannem Nagalxmi	**17311A05F1**

IMAGE STEGANOGRAPHY USING LSB (LEAST SIGNIFICANT BIT) TECHNIQUE

ABSTRACT

Steganography is a technique for information hiding. It is the art of hiding information within other information in such a way that it is hard or even impossible to identify the existence of any hidden information. There are many different carriers for steganography such as audio, image, video, etc of which, most popular ones are digital images. On the other side, steganalysis aims to expose the presence of hidden secret messages in those stego media. Steganalysis and steganography are the two different sides of the same coin. In practice two properties, undetectability and embedding capacity should be carefully considered when designing a steganographic algorithm. Usually, the larger payload embedded in a cover, the more detectable artifacts would be introduced into the stego. In many applications, the most important requirement for steganography is undetectability, which means that the stegos should be visually and statistically similar to the covers while keeping the embedding rate as high as possible.

INDEX

LIST OF FIGURES

1.INTRODUCTION

1.1 Purpose of the system

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the internet. For hiding secret information in images, there exists a large variety of steganography techniques some are more complex than others and all of them have respective strong and weak points. Different applications may require absolute invisibility of the secret information, while others require a large secret message to be hidden. This project report intends to give an overview of image steganography, its uses and techniques. The primary idea behind developing this project is to protect confidential data from an intruder's counter-attacks and to block the intruder through various levels in his/her attacks. A new tool has been developed with a combination of cryptographic encryption and steganographic encryption for its implementation. The developed steganographic tool has a sender's segment that can take a message, a password and a cover image as input and give a stego-image as output that has message embedded in it. On the other hand, it also has a receiver's segment where the receiver inputs the stego-image and the same password is used by the sender as input to get the sender's message as output. The project is tested with various inputs and made sure that the generated stego-image has no noise are data loss.

1.2 Existing System

Cryptography is technique of securing information and communications through use of codes so that only that person for whom the information is intended can understand it and process it. In cryptography only secret message is hidden, but the fact that secret communication is taking place is not hidden. So the third party can easily suspect the existence of the secret message. In this process even the structure of the data (secret message) is altered. It involves number theory, mathematics etc, which can be complex sometimes.

1.3 Proposed System

To avoid above situations, we propose this model. We propose a model that can hide the data behind the image, using which the third party can’t even suspect the existence of secret information. For this we use the LSB(Least Significant Bit) method. This technique works by replacing some of the information in a given pixel with information from the data in the image. While it is possible to embed data into an image on any bit-plane, LSB embedding is performed on the least significant bit(s). This minimizes the variation in colours that the embedding creates. Thus the output image has very slight difference to the input image which is mostly not recognizable.

Advantages:

- It is simple to understand,
- Easy to implement,
- It results in stego-images that contain hidden data yet appear to be of high visual fidelity.

2. SYSTEM REQUIREMENTS

2.1 Hardware System Configuration: -

Processor	-	Intel i3 or higher
RAM	-	4 GB
Hard Disk	-	30 GB

2.2 Software System Configuration: -

- Operating System: windows
- Python IDE: python 2.7.x and above
- PyCharm IDE

Anaconda Software to be installed.

● For Anaconda—Minimum 3 GB disk space to download and install.

2.3 MODULES OF THE SYSTEM

1. Sender
2. Encoding Module
3. Decoding Module
4. Receiver
5. Least Significant Bit Module

2.3.1 Sender

This is the initial part of the process when one person need to communication the secret information to other person without the involvement or interruption of third party then he will be the sender he needs to give input image which will be later transformed into stego image and secret information as inputs to the algorithm.

2.3.2 Encoding module

In this module the encoding process takes place after sender gives the input. During encoding the secret information will be encrypted inside the input image this is called stego image which acts as the cover for secret data. The algorithm works by taking the first pixel of the image and obtaining its LSB value. This is typically achieved by calculating the modulus 2 of the pixel value. This will return a 0 if then number is even, and a 1 if the number is odd, which effectively tells us the LSB value. We then compare this value with the message bit that we are trying to embed. If they are already the same, then we do nothing, but if they are different then were place the pixel value with the message bit. This process continues whilst there are still values in the message that need to be encoded.

2.3.3 Receiver model

After encoding the stego image will be sent to receiver through any communication channel. It is safe even though this channel is not hidden because the third can't even suspect the existence of secret data. In our project the output stego image can be transferred through E-Mail. Then the receiver gives stego image as input to the algorithm for decoding. To increase the security, we can use a key which will be known only to sender and receiver.

2.3.4 Decoding module

The decoding phase is even simpler. As the encoder replaced the LSBs of the pixel values in c in sequence, we already know the order that should be used to retrieve the data. Therefore, all we need to do is calculate the modulus 2 of all the pixel values in the stego image data. this time we run the loop for length of message instead of length of string. This is because the decoding process is completely separate from the encoding process and therefore has no means of knowing the length of the message. If a key were used, it would probably reveal this information, but instead we simply retrieve the LSB value of every pixel. When we convert this to ASCII, the message will be readable up to the point that the message was encoded, and will then appear as gibberish when we are reading the LSBs of the image data.

2.3.5 Least Significant Bit Module

In a Gray scale image each pixel is represented in 8 bits. The last bit in a pixel is called as Least Significant bit as its value will affect the pixel value only by “1”. So, this property is used to hide the data in the image. If anyone has considered last two bits as LSB bits as they will affect the pixel value only by “3”. This helps in storing extra data. The Least Significant Bit (LSB) steganography is one such technique in which least significant bit of the image is replaced with data bit. As this method is vulnerable to steganalysis so as to make it more secure we encrypt the raw data before embedding it in the image. Though the encryption process increases the time complexity, but at the same time provides higher security also. This approach is very simple. In this method the least significant bits of some or all of the bytes inside an image is replaced with the bits of the secret message. The LSB embedding approach has become the basis of many techniques that hide messages within multimedia carrier data. LSB embedding may even be applied in particular data domains – for example, embedding a hidden message into the color values of RGB bitmap data, or into the frequency coefficients of a JPEG image. LSB embedding can also be applied to a variety of data formats and types. Therefore, LSB embedding is one of the most important steganography techniques in use today.

3. SYSTEM ANALYSIS

3.1 FEASIBILITY STUDY

The task to be accomplished in this stage is to analyze the feasibility of the system that is being developed. The overall plan and the estimates of the project cost are put forth. In the phase of system analysis, one has to analyze whether the system is feasible or not, ensuring it to be feasible. This study is practiced by all the companies to get an assurance that the project that is going to taken up for development is not burden to them. It is necessary to have basic understanding of the important requirements.

Three key considerations involved in the feasibility analysis are

3.1.1 ECONOMICAL FEASIBILITY

3.1.2 TECHNICAL FEASIBILITY

3.1.3 SOCIAL FEASIBILITY

3.1.1 ECONOMICAL FEASIBILITY

This study will check the impact of proposed system that will be finally left on the corporate. This impact will decide whether the project can be handled by the company or not. The economical resources required for the project to be developed are limited in most of the cases. It is important to understand and give a clear idea of the fund that must be invested in the resources required for developing the system. This aspect is mandatory and is to be taken care so that the budget does not exceed the provided fund. In this study, it is the responsibility to make most of the available resources and provide an innovative system to the client.

The proposed system can overcome all the common issues and hence proves its economical feasibility. New data about diseases and their symptoms can be added as and when they are discovered. Thus, upgradation would not be a problem.

The system is economically feasible. No further hardware or software is required since this system's functionality is designed using the pre-existing tools and technologies.

3.1.2 TECHNICAL FEASIBILITY

The technical requirements for the system are to be properly analyzed and evaluated under this study of feasibility.The technical resources are limited and must be used efficiently. The proposal of the technical requirements for the system must be kept limited. If there is high demand for the technical sources, the economical feasibility will turn out to be high. This in turn causes a burden to the client in providing the required criteria. Hence, it is important to understand the need of technical resources properly and use the prior experience to fit the systeminto minimal technical requirements.

The target must be set properly that is to utilize the limited technical resources and gain the maximum project flexibility. This may sound complex but when trying to implement this phase, we get lot of ideas which leads to optimized usage of resources. It also depends on the efficiency of the team involved in the development of the system. More the experience of the team, the greater is the yield. By experience, it means having a general view of the system to be built,along with understanding of the availability of sources technically and skill to make te best out of minimum.

3.1.3 SOCIAL FEASIBILITY

Social feasibility is an important aspect to study. The user acceptance is the ultimate goal to be reached and this study aims at checking the confirmation by the user. The system efficiency requires user training. The user must accept and be satisfied with the look and feel of the system instead of feeling inconvenient. The user has to be trained and must get along with the proposed system as he is the ultimate customer of the product. The standard of the user confidence must be set on a rather high scale as he is the final person to experience the feel of the system. In our system, the doctor is the primary user. It is very important for the doctors not to feel threatened by this application, but rather see it as an intelligent counterpart.

3.2 SOFTWARE ENVIRONMENT

3.2.1 PYTHON

Python is a commonly used high level programming language.The usage of python in the industrydomain has become very common and has gained popularity.The features that make python different from the existing programming languages are:

1. Object oriented
2. Portable
3. Powerful
4. Easy to learn
5. Easy to use

Similar to Java, Python also supports all the object oriented concepts like polymorphism, operator overloading and inheritance. Unlike Java, Python supports multiple inheritance. Itconsists of generators, lambdas and comprehensions. These features make Python language easy to code and learn.

Applications:

1. System Programming
2. Internet Scripting
3. Data Mining
4. Graphical User Interface (GUI)
5. Rapid Prototyping
6. Numeric and Scientific Programming

Python can perform the above using the in-built libraries. Few such libraries are Numpy, Scipy, Matplotlib. These libraries allow Python to perform numerical operations on arrays. They can also help in visualizing the data. In Machine Learning, Python plays a key role in reducing the complexity of algorithm by just coding few lines. The libraries Keras and Tensorflow in Python make the ML model compatible with android studio as well.

3.2.3 Python Script

A Python script is a series of commands in a file intended to run like a program. Of course the file can contain functions and import different modules, but the purpose is to run or execute a particular task from the command line or from inside a Python interactive shell. A script also includes a collection of function descriptions first, and then has the main program that will call the functions. This is a very useful feature that enables you to type in a program and have it executed in an interactive environment immediately.

3.2.4 Python Tkinter

The absence of Tkinter from Python would make it less attractive to many users. Tkinter is also calledTki in Python. To be precise, Tkinter is the Python interface for Tk. Tkinter is an acronym for "Tk interface".

Initially, Tkinter had been developed mainly as a GUI extension for the TCL scripting language by John Ousterhout. Its first release was in 1991. In the 1990s, Tk proved to be extremely popular, because it is simpler to learn and use than other toolkits. So it's no wonder a lot of programmers would like to use Tk independently of Tcl.

Listed below are some of the many widgets provided by Tkinter:

- button
- canvas
- checkbutton
- combobox
- entry
- frame
- label
- labelframe
- listbox
- menu

- menubutton
- message
- notebook
- tk_optionMenu
- panedwindow
- progressbar
- radiobutton
- scale
- scrollbar
- separator
- sizegrip
- spinbox
- text
- treeview

It provides the following top-level windows:

- tk_chooseColor - pops up a dialog box for the user to select a color.
- tk_chooseDirectory - pops up a dialog box for the user to select a directory.
- tk_dialog - creates a modal dialog and waits for a response.
- tk_getOpenFile - pops up a dialog box for the user to select a file to open.
- tk_getSaveFile - pops up a dialog box for the user to select a file to save.
- tk_messageBox - pops up a message window and waits for a user response.
- tk_popup - posts a popup menu.
- toplevel - creates and manipulates toplevel widgets.

Tk also provides three geometry managers:

- place - which positions widgets at absolute locations
- grid - which arranges widgets in a grid
- pack - which packs widgets into a cavity

Importing TKinter Module:

1. Import the Tkinter Module.
2. Create the GUI application Main Window.
3. Add one or more of the above-mentioned widgets to the GUI application.
4. Enter the main event loop to take action against each event triggered by the user

4. SYSTEM DESIGN

Systems design is the mechanism used to decide the architecture, components, modules, interfaces, and data of a system to meet specified needs. It is similar to applying system theory toproduct development. Object-oriented methods of research and design are now the techniques most commonly used for the design of computer systems.

4.1 INPUT DESIGN

Input design is a part of overall system design. During the creation of the inputs the main objective is as follows:

- To generate an input system which is cost-effective.
- To get the highest degree of accuracy possible.
- To make sure the user accepts and understands the data.

OBJECTIVES:

1. System analysis helps to discover means to design systems where sub-systems have apparently clear objectives.
2. It helps to achieve inter-compatibility.
3. It develops understanding between complex structures.
4. It constitutes synchronization between objectives and systems.

INPUTMEDIA:

The decision about the input media has to be made at this point. To conclude on the consideration of input media has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy

Taking into consideration the above definition of the input media, it can be assumed that most of the inputs are internal and interactive in nature

Input data is obtained from the user when they select one of the image from their device and enter the secret data.

4.2 OUTPUT DESIGN

It provides the information to the user and is important as it is the only source of output to user. The output that is efficient and correct improves the system relationship with the user which then helps in making decisions. An output is said to be a quality output, when it meets and serves the needs of the end user and displays the information unambiguously. These results are discussed with the user and the other system through outputs. The way displaying the information for the immediate requirement is made in the output design.

The objectives are-

- Design the computer output to serve the user.
- Deliver appropriate output.
- Output is presented to the right user.
- Select methods for re-presenting information.
- Convey information about previous activities, present status.
- Signal and alert important events, opportunities, problems, or warnings.
- Trigger the actions.
- Confirm the actions.

4.3 SYSTEM ARCHITECTURE

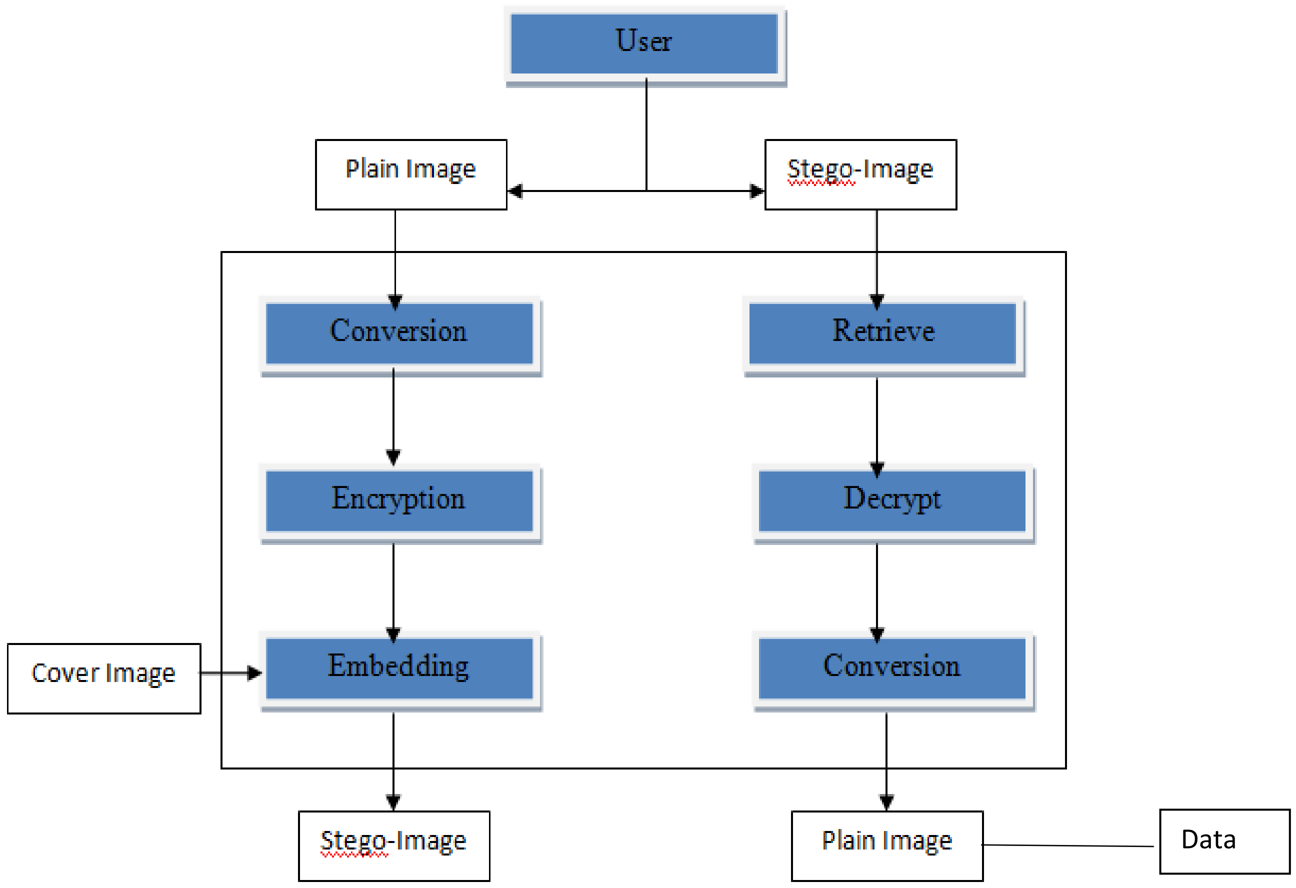


FIG 4.3 ARCHITECTURE OF SYSTEM

The above figure 4.3 shows the proposed architecture of our project “Image Steganography using Least Bit Significant technique”. It shows how our system is designed and shows the flow among various elements throughout the system in an abstract view. In this project, We propose a model that can hide the data behind the image, using which the third party can’t even suspect the existence of secret information. For this we use the LSB (Least Significant Bit) method. This technique works by replacing some of the information in a given pixel with information from the data in the image. While it is possible to embed data into an image on any bit-plane, LSB embedding is performed on the least significant bit(s). This minimizes the variation in colours that the embedding creates. Thus, the output image has very slight difference to the input image which is mostly not recognizable.

4.4 UML Diagrams

4.4.1 USE CASE DIAGRAM

A use case diagram at its simplest is a description of the interaction of a user with the device and a depiction of a use case requirement. A use case diagram will represent the various user styles of a system and the different ways it communicates with the system. Usually, this type of diagram is used in combination with the case for textual use and is often also accompanied by other types of diagrams.

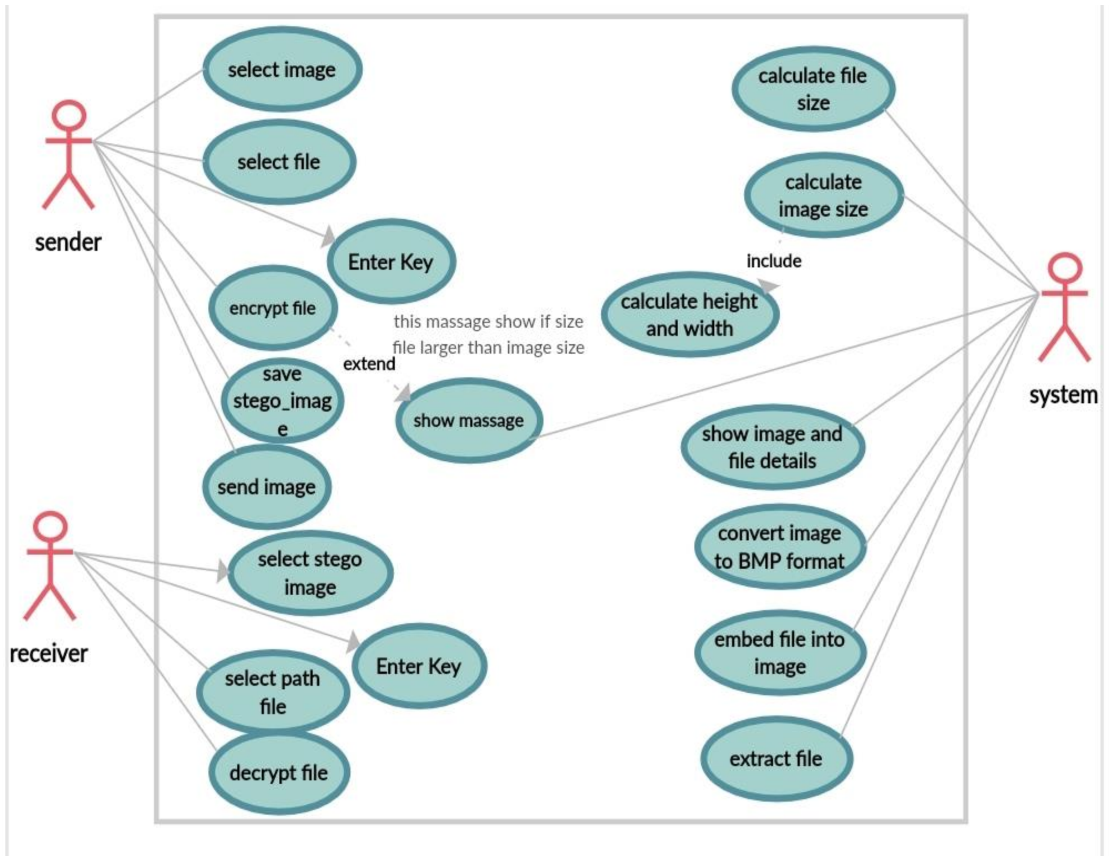


FIG 4.4.1.3.1 USE CASE DIAGRAM

4.4.2 SEQUENCE DIAGRAM

The Figure 4.4.1.3.2 shows the sequence or flow of messages in the system among various objects of the system. The rectangle boxes at top represent objects that are invoked by doctorand the dashed lines dropping from those boxes are life lines which shows existence of the objectup to what time. The boxes on the dashed lines are events and the lines connecting them represent messages and their flow.

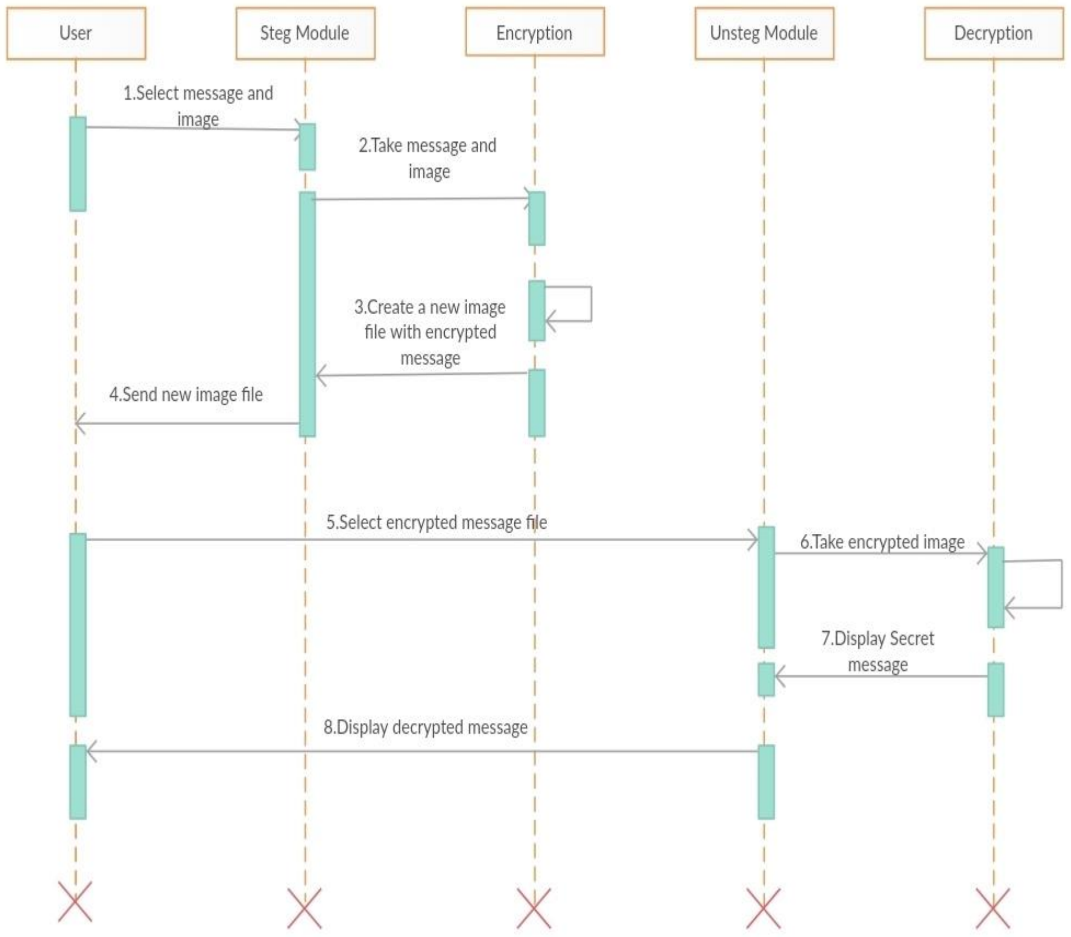


FIG 4.4.1.3.2 SEQUENCE DIAGRAM

4.4.3 CLASS DIAGRAM

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

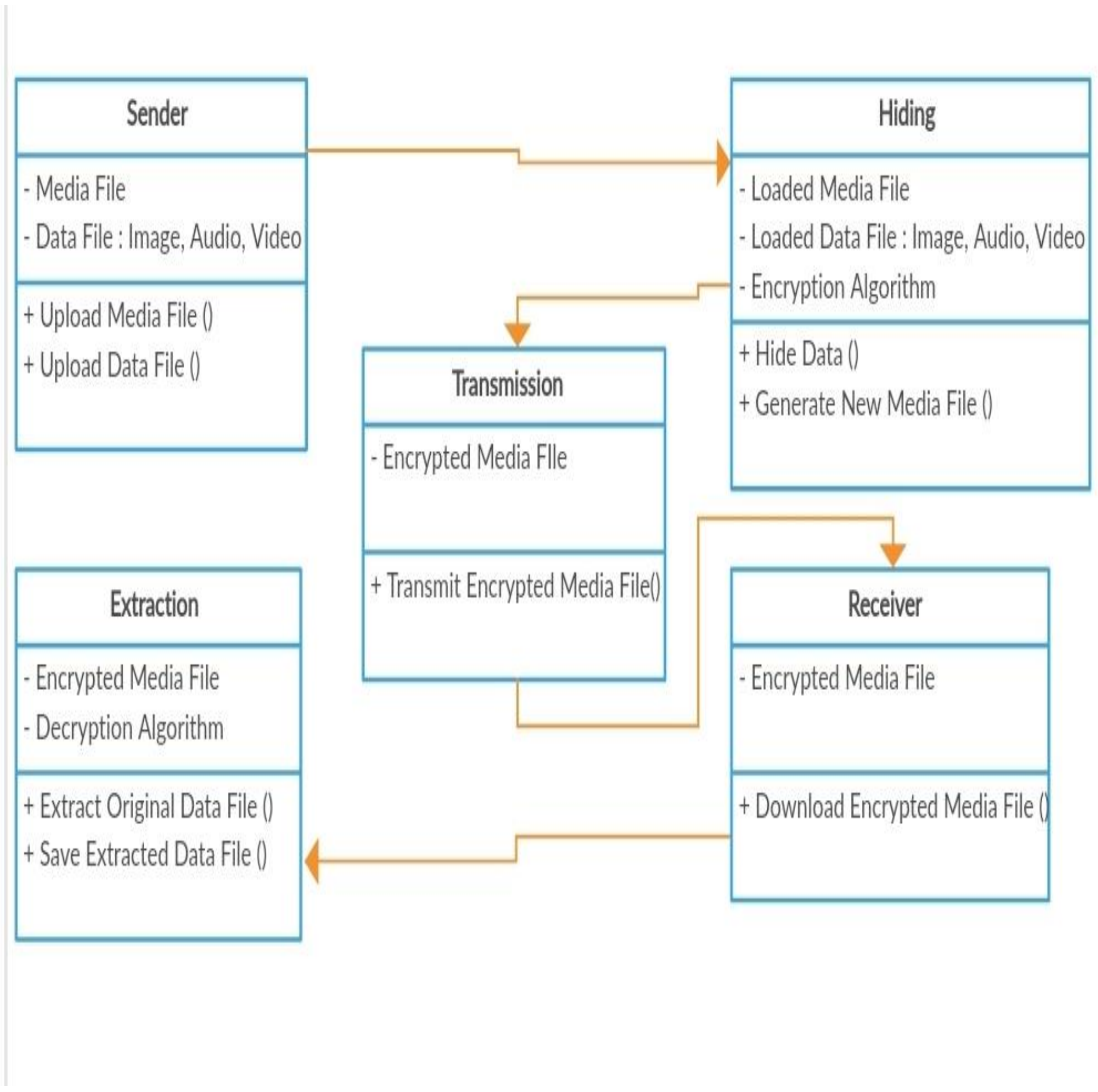


FIG 4.4.1.3.3 CLASS DIAGRAM

5.IMPLEMENTATION

Implementation is one of the most critical tasks in a project where one needs to be careful as all the attempts made during the project have to be very interactive. Implementation is the most crucial stage in achieving an effective system and giving the users trust that the new system is workable and efficient.

5.1 SYSTEM MODULES WITH ALGORITHM STEPS

1. Sender module
2. Encoding module
3. Receiver module
4. Decoding module
5. LSB Algorithm

1. Sender Module

<u>Steps:</u>

1. Upload the Image

2. Enter the Secret data

2. Encoding Module

<u>Steps:</u>

1) Apply LSB on input data

2) Encrypt the data

3) Conversion of image to stego-image

4) Embedding

3. Receiver Module

<u>Steps:</u>

1) Receive the stego image
2) Upload the stego image
3) Decryption

4. Decoding module

1) Retrieve the stego image

2) Decrypt the secret data

3) Display the secret data

5. LSB Module

1) Run the algorithm

2) Get the least significant bits in the pixels

3) Covert the data into binary format

4) Embed the binary bits into pixels least significant bits.

5) Send the stego-image.

5.2 CODE

```
from tkinter import *

from tkinter import ttk
```

```
import tkinter.filedialog

from PIL import ImageTk

from PIL import Image

from tkinter import messagebox

from io import BytesIO

import  os

import webbrowser


new=1

url="https://www.gmail.com"

class Stegno:


    output_image_size = 0




    def main(self,root):


        root.title('ImageSteganography')

        root.geometry('700x700')

        root.resizable(width =False, height=False)
```

```
f = Frame(root)


title = Label(f,text='Image Steganography')

title.config(font=('courier',33))

title.grid(pady=10)


b_encode = Button(f,text="Encode",command= lambda :self.frame1_encode(f), padx=14)

b_encode.config(font=('courier',14))

b_decode = Button(f, text="Decode",padx=14,command=lambda :self.frame1_decode(f))

b_decode.config(font=('courier',14))

b_decode.grid(pady = 12)

b_share=Button(f,text="Share",padx=14,command=self.openweb)

b_share.config(font=('courier',14))

b_share.grid(pady=12)




root.grid_rowconfigure(1, weight=1)

root.grid_columnconfigure(0, weight=1)
```

```
    f.grid()

    title.grid(row=1)

    b_encode.grid(row=2)

    b_decode.grid(row=3)

    b_share.grid(row=4)



def home(self,frame):

        frame.destroy()

        self.main(root)

 def openweb(self):

    webbrowser.open(url,new=new)


 def frame1_decode(self,f):

    f.destroy()

    d_f2 = Frame(root)


    l1 = Label(d_f2, text='Select Image with Hidden text:')

    l1.config(font=('courier',18))

    l1.grid()

    bws_button = Button(d_f2, text='Select', command=lambda :self.frame2_decode(d_f2))
```

```
        bws_button.config(font=('courier',18))

        bws_button.grid()

        back_button = Button(d_f2, text='Cancel', command=lambda : Stegno.home(self,d_f2))

        back_button.config(font=('courier',18))

        back_button.grid(pady=15)

        back_button.grid()

        d_f2.grid()



    def frame2_decode(self,d_f2):

        d_f3 = Frame(root)

        myfile = tkinter.filedialog.askopenfilename(filetypes = ([('png', '*.png'),('jpeg',
'*.jpeg'),('jpg', '*.jpg'),('All Files', '*.*')]))

        if not myfile:

            messagebox.showerror("Error","You have selected nothing !")

        else:

            myimg = Image.open(myfile, 'r')

            myimage = myimg.resize((300, 200))

            img = ImageTk.PhotoImage(myimage)

            l4= Label(d_f3,text='Selected Image :')

            l4.config(font=('courier',18))

            l4.grid()

            panel = Label(d_f3, image=img)
        panel.image = img

        panel.grid()

        hidden_data = self.decode(myimg)

        l2 = Label(d_f3, text='Hidden data is :')

        l2.config(font=('courier',18))

        l2.grid(pady=10)

        text_area = Text(d_f3, width=50, height=10)

        text_area.insert(INSERT, hidden_data)

        text_area.configure(state='disabled')

        text_area.grid()

        back_button = Button(d_f3, text='Cancel', command= lambda :self.page3(d_f3))

        back_button.config(font=('courier',11))

        back_button.grid(pady=15)

        back_button.grid()

        show_info = Button(d_f3,text='More Info',command=self.info)

        show_info.config(font=('courier',11))

        show_info.grid()

        d_f3.grid(row=1)

        d_f2.destroy()


    def decode(self, image):
```

```
    data = ''
    imgdata = iter(image.getdata())


    while (True):
        pixels = [value for value in imgdata.__next__()[:3] +
                  imgdata.__next__()[:3] +
                  imgdata.__next__()[:3]]
        binstr = ''
        for i in pixels[:8]:
            if i % 2 == 0:
                binstr += '0'
            else:
                binstr += '1'


        data += chr(int(binstr, 2))
        if pixels[-1] % 2 != 0:
            return data


def frame1_encode(self,f):
    f.destroy()
    f2 = Frame(root)
```

```
        l1= Label(f2,text='Select the Image in which \nyou want to hide text :')

        l1.config(font=('courier',18))

        l1.grid()


        bws_button = Button(f2,text='Select',command=lambda : self.frame2_encode(f2))

        bws_button.config(font=('courier',18))

        bws_button.grid()

        back_button = Button(f2, text='Cancel', command=lambda : Stegno.home(self,f2))

        back_button.config(font=('courier',18))

        back_button.grid(pady=15)

        back_button.grid()

        f2.grid()



    def frame2_encode(self,f2):

        ep= Frame(root)

        myfile = tkinter.filedialog.askopenfilename(filetypes = ([('png', '*.png'),('jpeg',
'*.jpeg'),('jpg', '*.jpg'),('All Files', '*.*')]))

        if not myfile:

            messagebox.showerror("Error","You have selected nothing !")

        else:
```

```
        myimg = Image.open(myfile)

        myimage = myimg.resize((300,200))

        img = ImageTk.PhotoImage(myimage)

        l3= Label(ep,text='Selected Image')

        l3.config(font=('courier',18))

        l3.grid()

        panel = Label(ep, image=img)

        panel.image = img

        self.output_image_size = os.stat(myfile)

        self.o_image_w, self.o_image_h = myimg.size

        panel.grid()

        l2 = Label(ep, text='Enter the message')

        l2.config(font=('courier',18))

        l2.grid(pady=15)

        text_area = Text(ep, width=50, height=10)

        text_area.grid()

        encode_button = Button(ep, text='Cancel', command=lambda : Stegno.home(self,ep))

        encode_button.config(font=('courier',11))

        data = text_area.get("1.0", "end-1c")

        back_button = Button(ep, text='Encode', command=lambda :
[self.enc_fun(text_area,myimg),Stegno.home(self,ep)])

        back_button.config(font=('courier',11))
        back_button.grid(pady=15)

        encode_button.grid()

        ep.grid(row=1)

        f2.destroy()




    def info(self):

        try:

            str = 'original image:-\nsize of original image:{}mb\nwidth: {}\nheight: {}\n\n' \

                  'decoded image:-\nsize of decoded image: {}mb\nwidth: {}' \

                '\nheight: {}'.format(self.output_image_size.st_size/1000000,

                                    self.o_image_w,self.o_image_h,

                                    self.d_image_size/1000000,

                                    self.d_image_w,self.d_image_h)

            messagebox.showinfo('info',str)

        except:

            messagebox.showinfo('Info','Unable to get the information')

    def genData(self,data):

        newd = []



        for i in data:
```

```
        newd.append(format(ord(i), '08b'))

    return newd


def modPix(self,pix, data):

    datalist = self.genData(data)

    lendata = len(datalist)

    imdata = iter(pix)

    for i in range(lendata):



        pix = [value for value in imdata.__next__()[:3] +

               imdata.__next__()[:3] +

               imdata.__next__()[:3]]



        for j in range(0, 8):

            if (datalist[i][j] == '0') and (pix[j] % 2 != 0):



                if (pix[j] % 2 != 0):

                    pix[j] -= 1



            elif (datalist[i][j] == '1') and (pix[j] % 2 == 0):

                pix[j] -= 1
        if (i == lendata - 1):
            if (pix[-1] % 2 == 0):
                pix[-1] -= 1
        else:
            if (pix[-1] % 2 != 0):
                pix[-1] -= 1

        pix = tuple(pix)
        yield pix[0:3]
        yield pix[3:6]
        yield pix[6:9]

def encode_enc(self,newimg, data):
    w = newimg.size[0]
    (x, y) = (0, 0)

    for pixel in self.modPix(newimg.getdata(), data):


        newimg.putpixel((x, y), pixel)
```

```
            if (x == w - 1):

                x = 0

                y += 1

            else:

                x += 1


    def enc_fun(self,text_area,myimg):

        data = text_area.get("1.0", "end-1c")

        if (len(data) == 0):

            messagebox.showinfo("Alert","Kindly enter text in TextBox")

        else:

            newimg = myimg.copy()

            self.encode_enc(newimg, data)

            my_file = BytesIO()

            temp=os.path.splitext(os.path.basename(myimg.filename))[0]

            newimg.save(tkinter.filedialog.asksaveasfilename(initialfile=temp,filetypes = ([('png',
'*.png')]),defaultextension=".png"))

            self.d_image_size = my_file.tell()

            self.d_image_w,self.d_image_h = newimg.size

            messagebox.showinfo("Success","Encoding Successful\nFile is saved as
Image_with_hiddentext.png in the same directory")
```

```
    def page3(self,frame):

        frame.destroy()

        self.main(root)


root = Tk()


o = Stegno()

o.main(root)


root.mainloop()
```

6. OUTPUT SCREENS

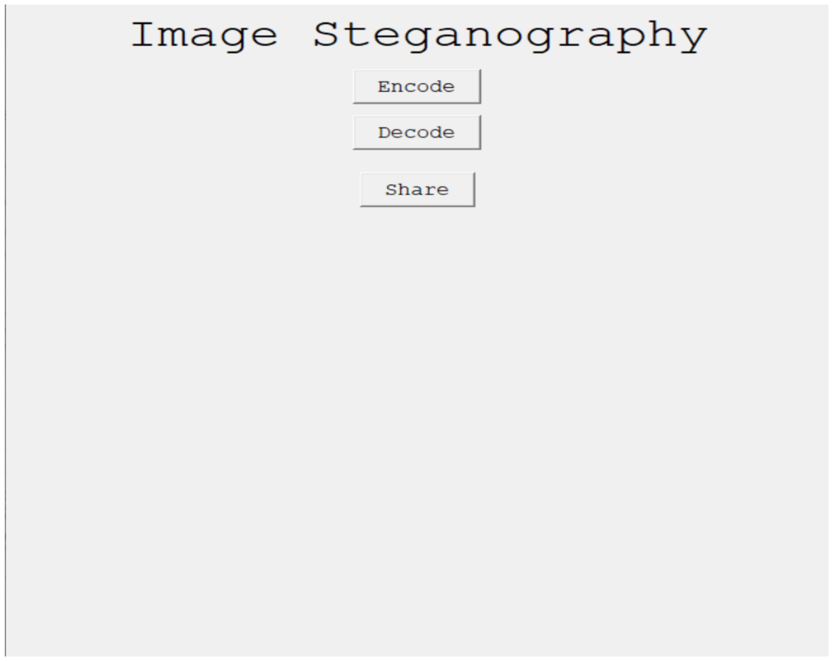


FIG 6.1 OUTPUT OVERVIEW

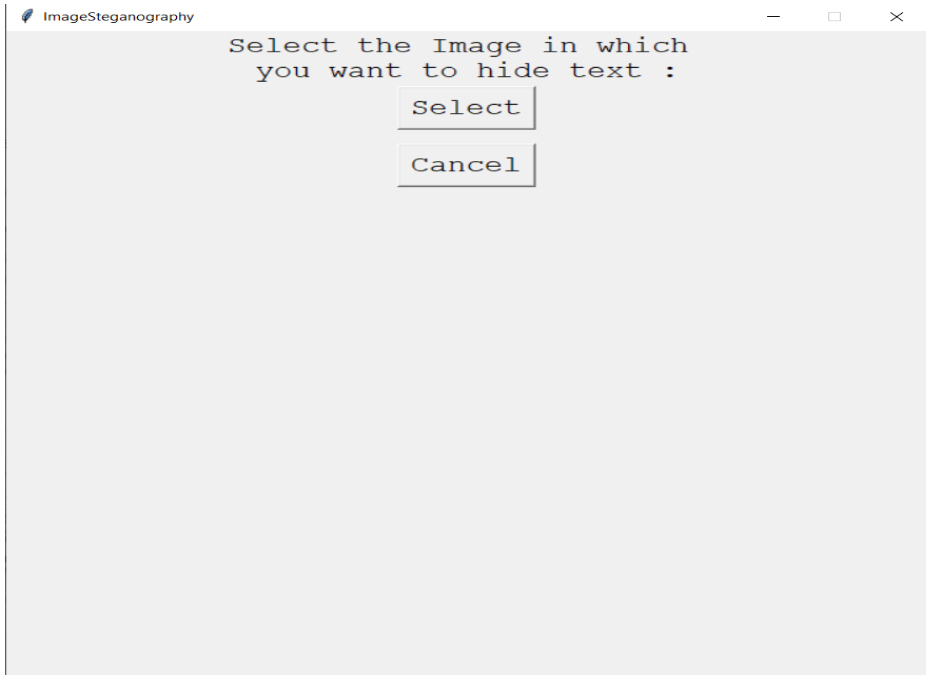
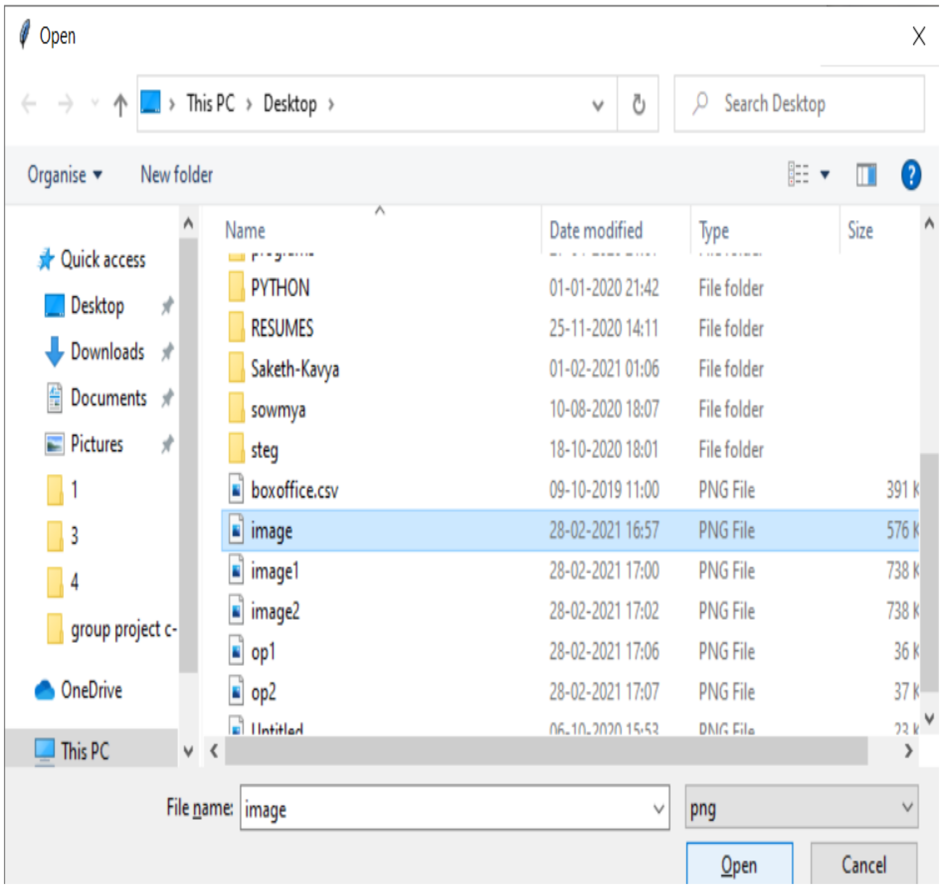


FIG 6.2 ENCODING

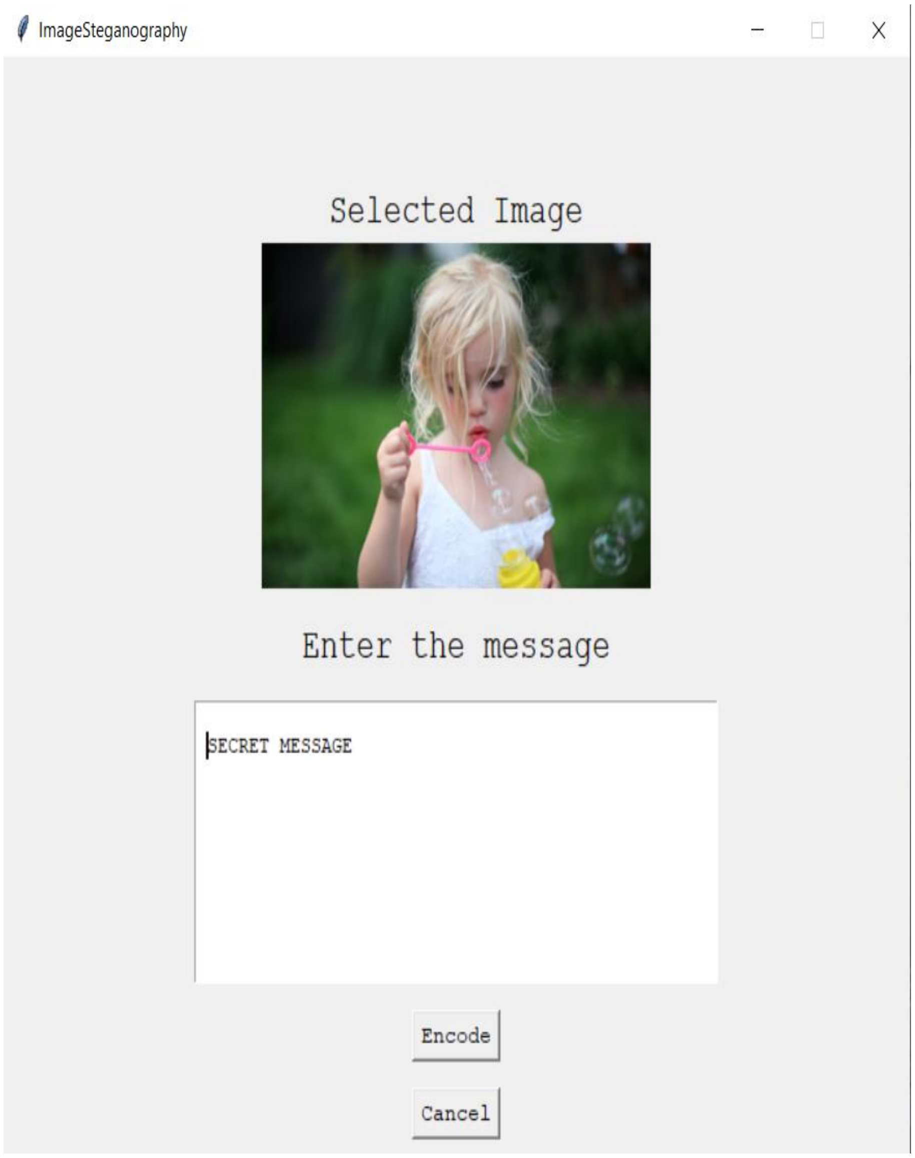


Name	Date modified	Type	Size
PYTHON	01-01-2020 21:42	File folder	
RESUMES	25-11-2020 14:11	File folder	
Saketh-Kavya	01-02-2021 01:06	File folder	
sowmya	10-08-2020 18:07	File folder	
steg	18-10-2020 18:01	File folder	
boxoffice.csv	09-10-2019 11:00	PNG File	391 K
image	28-02-2021 16:57	PNG File	576 K
image1	28-02-2021 17:00	PNG File	738 K
image2	28-02-2021 17:02	PNG File	738 K
op1	28-02-2021 17:06	PNG File	36 K
op2	28-02-2021 17:07	PNG File	37 K
Untitled	06-10-2020 15:53	PNG File	23 K

File name: image

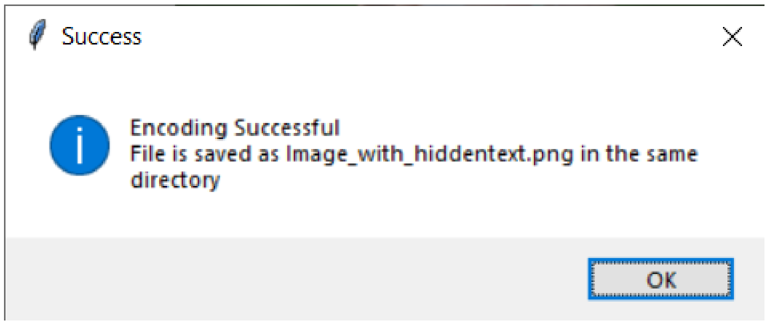
png

FIG 6.3 SELECT INPUT IMAGE



SECRET MESSAGE

FIG 6.4 ENTER SECRET DATA



Encoding Successful
File is saved as Image_with_hiddentext.png in the same directory

FIG 6.3 ENCODING OUTPUT

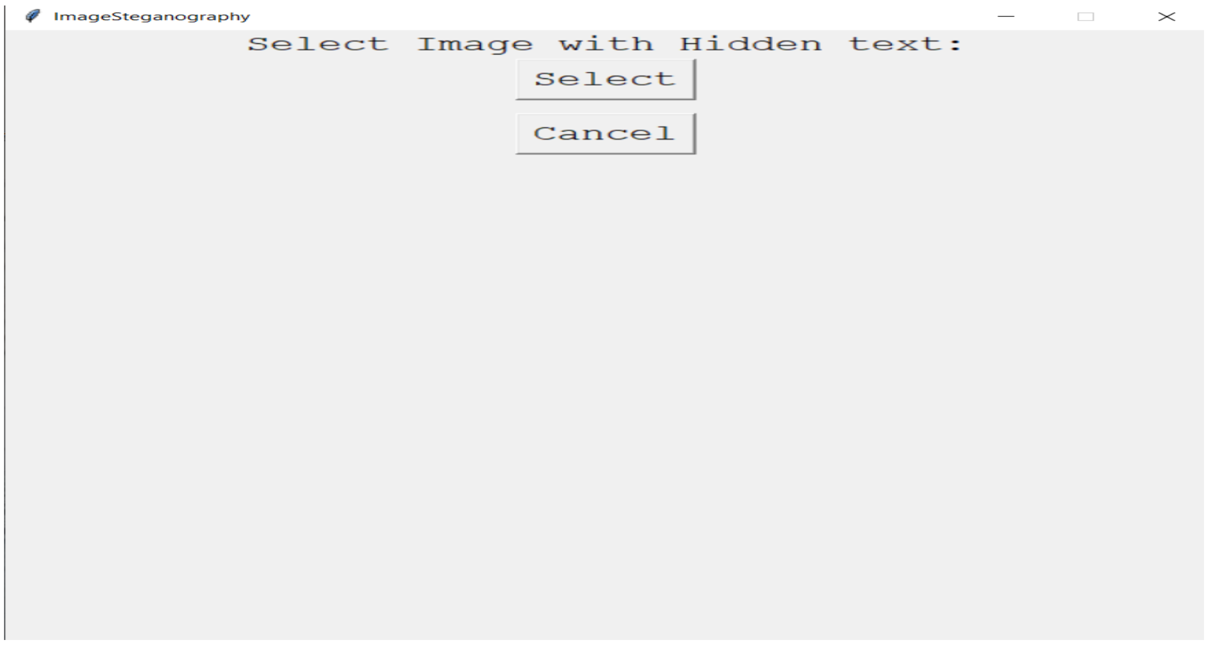
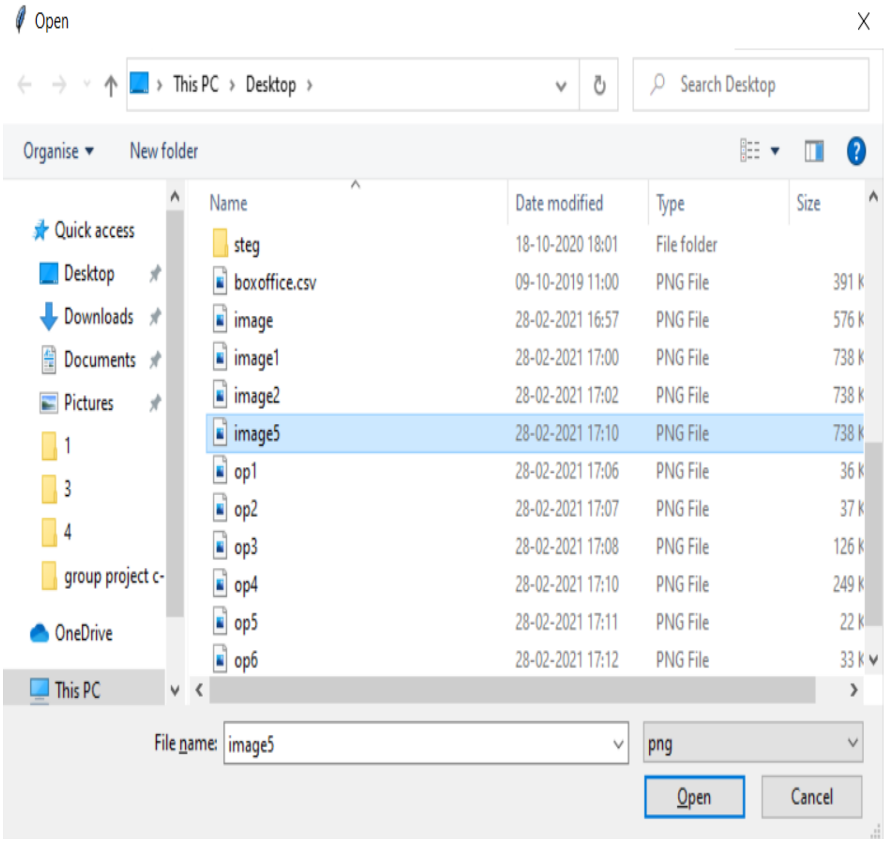


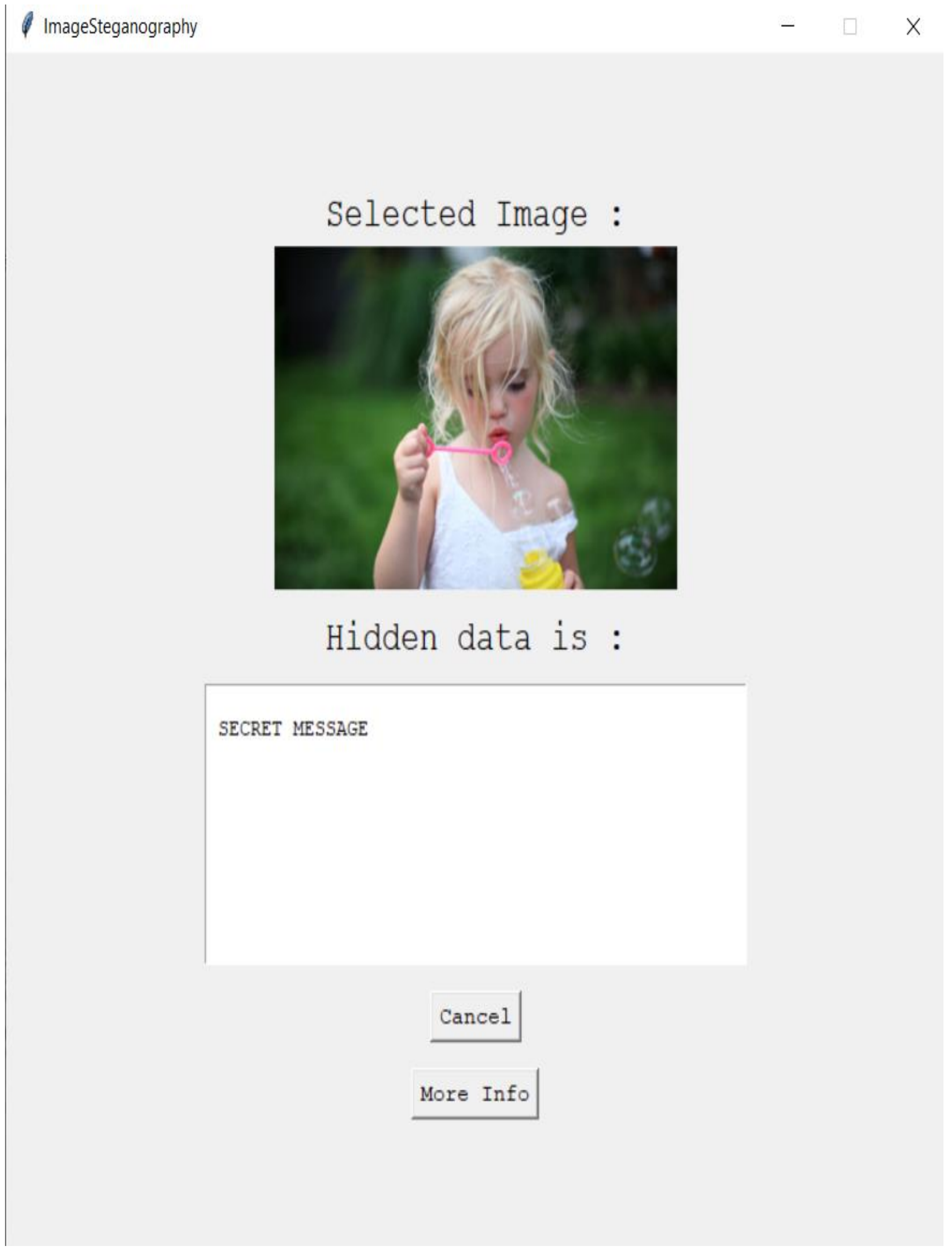
FIG 6.5 DECODING



Name	Date modified	Type	Size
steg	18-10-2020 18:01	File folder	
boxoffice.csv	09-10-2019 11:00	PNG File	391 K
image	28-02-2021 16:57	PNG File	576 K
image1	28-02-2021 17:00	PNG File	738 K
image2	28-02-2021 17:02	PNG File	738 K
image5	28-02-2021 17:10	PNG File	738 K
op1	28-02-2021 17:06	PNG File	36 K
op2	28-02-2021 17:07	PNG File	37 K
op3	28-02-2021 17:08	PNG File	126 K
op4	28-02-2021 17:10	PNG File	249 K
op5	28-02-2021 17:11	PNG File	22 K
op6	28-02-2021 17:12	PNG File	33 K

File name: image5
png

FIG 6.6 SELECT STEGO IMAGE



SECRET MESSAGE

FIG 6.7 DECODING OUTPUT

6. SYSTEM TESTING

Testing can be understood as the process of evaluating the software functionality with aninitiative that is intended to end up understanding and knowing whether the software that is developed has lived up to the expectations and requirement specifications. This entire procedure will help the developer know the pitfalls of the system before delivering it to the user. Once the defects are known, the team will strive hard to make sure they are cleared as soon as possible andthe product is ready to be delivered. The software must be entirely reliable as expected by the client and the quality check must take place thoroughly with the help of testing. It is simply analyzing the software to detect differences between existing software and the developed software. The main aim of testing is to identify the errors and their spots so that they can be rectified. Work product may contain several defects which are to be recognized with the process of testing. Testing provides tools to check the individual components functionality sub- assemblies, assemblies and the final product. The results of testing are put into action later on during maintenance also

Testing is a method used to identify defects. By using Testing, it is possible to detect any potential defect or flaw of the method in a work product.

Testing checks the functionality of the software also functionalities of assemblies, subassemblies and the finished product. It is the process of checking whether the developed software reaches all the user requirements or not. Testing is useful for the software to function accurately. If the testing done well then the final product doesn’t fail to achieve its functionalities. The testing can be performed in many ways and there many types in testing. Each type of testing have its own specific requirements.

The aim of the test is to detect software failures in order to resolve and correct these detections. Testing is a process that which analyses the functionalities of software under some specific conditions .It does not require that the software can be performed well in all the conditions. Software testing includes testing of source code, syntax and semantics of functions and procedures in that code and also execution of source code in all environment and all differentconditions. Testing tests whether the code does what it actually needs to do and whether it does what it is supposed to do or not. In the software environment Testing is different from development. The organization has separate team members for the testing itself. The testing can be done either manually or automatically by using testing tools.The knowledge the test team receives is used to correct the program operation.

Testing Objectives

- Revealing the errors that are not found by the developers in the former stage is the primary objective of testing ,systematically and with less and in less time.
- An undiscovered error will be revealed sooner by a successful test ,
- A good test case does the job of finding the error which has highest probability of deviating the software.
- The confirmation of the software quality and the reliability is essential.

Defects and Failures

Not all software bugs are caused by coding errors while designing the code. Sometimes the defects are caused by unrecognized requirements of the programmer. If the programmer doesn't understand the requirements of the customer properly then it may be defect in the software. In this type of defects the code may be correct but actual functionality of the software changes. Non-functional specifications such as testability, scalability, reliability, usability, efficiency, and protection are a common source of requirement gap.

The software faults trigger the following process. A programmer makes an mistake, which results in a defect(fault, bug) in the source code of the software. If the code with these is executedthen it may leads to wrong results and the entire system produce wrong results and it leads to failures of the system.Sometimes the failures cannot be detected if there is a dead code.This type of failures only detected by change the environment of the system.These failures are expensive.

E.g. the program running on a new hardware device, modifications to source data or the connection with specific applications. A single defect may lead to a large array of symptoms of failure.

Compatibility

Another significant cause of software failure is compatibility. The developed software should be compatible with another program or new operating system.This failure happens when a programmer does not consider the operating system. Programmers only consider the latest operating system in which they operate. Programmers only consider the programs for, or testing software to, the latest operating system for which they have access or otherwise under the perfect circumstances they have learned.If different users want to use in different environment it leads to software failure.This means the software only works under some specific conditions and only in the same machine with some particular hardware and software combinations only.These types of failures are very hard to predict because these are known to the developers only after releasing the final product.A software programmer may not able to detect these type of failures because lack of availability and access permissions.These failures can be detected by experienced programmers who are able to interact with several type of older versions of operating systemsand several older machines.To Controls these type of failures the organizations should have very experienced programmers and the older versions of machines with it.It could be viewed as a prevention-oriented approach that is in line with the current research process proposed by Dave Gelperin and William C.Hetzel.

Input Conditions And Preconditions

The main percussion of software testing is that even if the product is simple it should be tested with all input combinations and preconditions which is not feasible.Testing the simple product with all the combinations is very expensive and a time taking process. And also in some cases the product may be simple but the defects are large.Significantly, qualitative non-functional dimensions, such as usability, valuableness, outputs, compatibilities, and reliability (how to be compared with what is expected to do), can be very subjective, which can be of inadequate interest to one individual.

Static vs Dynamic Testing

There are two forms of software testing. One is static testing and the other is dynamic testing. Updates, walkthroughs or source code checks are called static testing. And the programmed code and the collection of software creation are applied, and then the first phase is used and then the test stage starts. This can actually start before the software is 100 percent complete to test other code parts (modules or separate functions).

Spreadsheet systems, for example, are often checked by their very design – on the fly during the construction process when some calculation or text manipulation is seen interactively directly after each formula is entered.

Software Verification and Validation

Software testing is used in tandem with verification and validation.

Verification: Have we designed the right program (i.e., does it meet specification?) that is based on method.

Validation: Have we developed the appropriate applications (i.e. is this what the client wants?)It is Product-based.

The software Testing Team

Since the testing is also a critical part of the production of software, separate testing teams are needed for testing. The testing team will do testing. Until the 1950s the testing was commonly used, but it was also a separate occupation due to the importance of research. The different roles of software testing are often different depending on the times and priorities of software testing. The different roles of software testing are: test manager, test designer, test developer, test administrator, hardware tester.

Software Quality Assurance (SQA)

Software testing is an important aspect of Software Quality Assurance . In SQA, the software quality is reviewed by the team members. The program will meet all end user requirements to achieve high quality. Specialists and auditors take a look at the program and its phase of growth. We analyze the method of software engineering and change the process if it needs means it can completely meet the requirements.

DESIGN OF TEST CASES AND SCENARIOS

At each step of the software development process, there are chances that various errors may occur. Checking is performed at the output of each step. Each module and sub modules are testedfor errors at the output of each phase.

TESTING STRATEGIES

Test cases are designed for the intention of detecting errors. A test case is a collection of data that the system must interpret as standard input. The test is designed for this system to verify if the changes are being made correctly. The other test cases are designed to verify the situation where no data is available for modification for a particular condition.

Code Testing ----------------------> Unit Testing------------------> System Testing

7.1 TYPES OF TESTS

7.1. Unit testing

Unit testing involves testing of individual units in the project. The entire project is divide into several units and each unit In turn is tested to check if the user requirements are fulfilled. Unlike other testings, developer performs unit testing. It includes designing of all the possible and hidden test cases which will evaluate the programming logic to know if it is working appropriatley, and the inputs of the program give right outputs. It is the level-1 testing technique.It is performed after Integration Testing is done. Unit testing is comprised of 2 parts:

1)Manual Testing

2)Automation Testing

Manual testing: It is a type of testing performed by the testers and this testing procedure doe not involve any automation tools. The entire testing process is carried out manually which is time taking and inflexible, Automation testing: a type of testing where special kinds of software tools are used for executing test cases and predicted and actual test cases results.

Few techniques in Unit Testing :

1.Statement coverage

2.Branch coverage

3.Decision coverage

4.Condition coverage

5.Finite state machine coverage

7.2. Integration testing

Integration is the step that is carried out after unit testing is performed. The main aim is to see that the modules are integrated properly based on testing interfaces between modules. The prior integration part is for individual elements and after integration, the role of integration testing is to verify if the whole unit which is built from the individual units is working as per the stated constraints. This testing is driven by all the events involved respectively and the focus isessentially on the output of the fields. This testing will encourage the revealing of problems that occur when the system is checked as a whole. Various approaches of performing integration testing are :

1)Big Bang Approach and

2)Incremental Approach

A : Top Down Approach

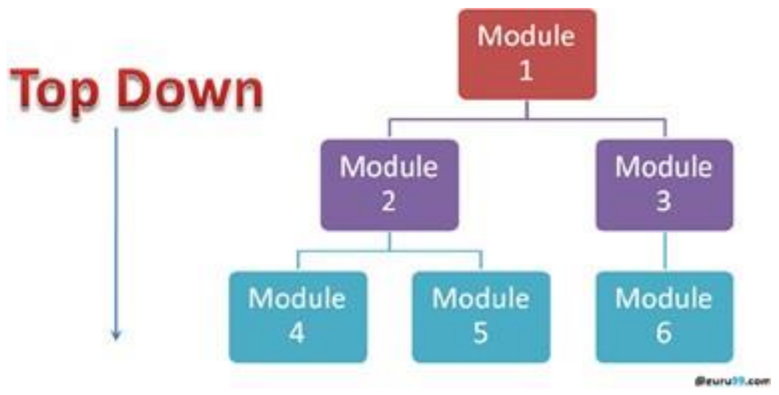


FIG 7.2 Approach for Top Down

B : Bottom Up Approach

C : Sandwich Approach

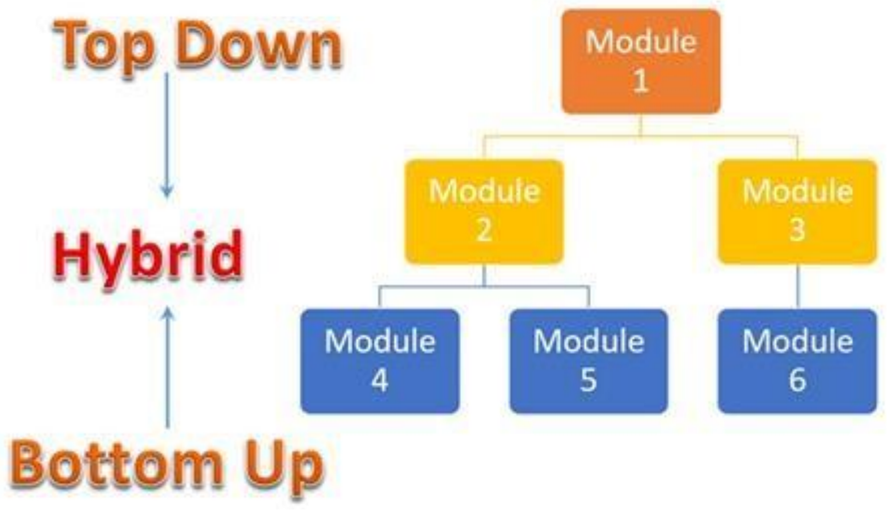


FIG 7.3 Hybrid Approach

System Testing

System testing is also called End-to-end testing. It ensures that the final complete integrated software system meets the requirements. It verifies that the testing of each input is checked to retrieve every possible outputs.It is performed by the professional testing agent.The configuration of the system is tested to know if it is functioning as expected. It falls under the class of Black-box testing. Few types of system testing are :

1.Usability

2.Load

3.Regression

4.Recovery

5.Migration

6.Functional

7.Hardware/Software

7.3Acceptance Testing

User Acceptance Testing is performed by the end users/clients to verify and make sure that all the tasks work well in the software application and in less time and efficiently. It can be considered as an analytic type of testing for a given project, it requires prominent involvement ofthe final user of the system. Apart from this, it also checks whether the functional requirements are met right.

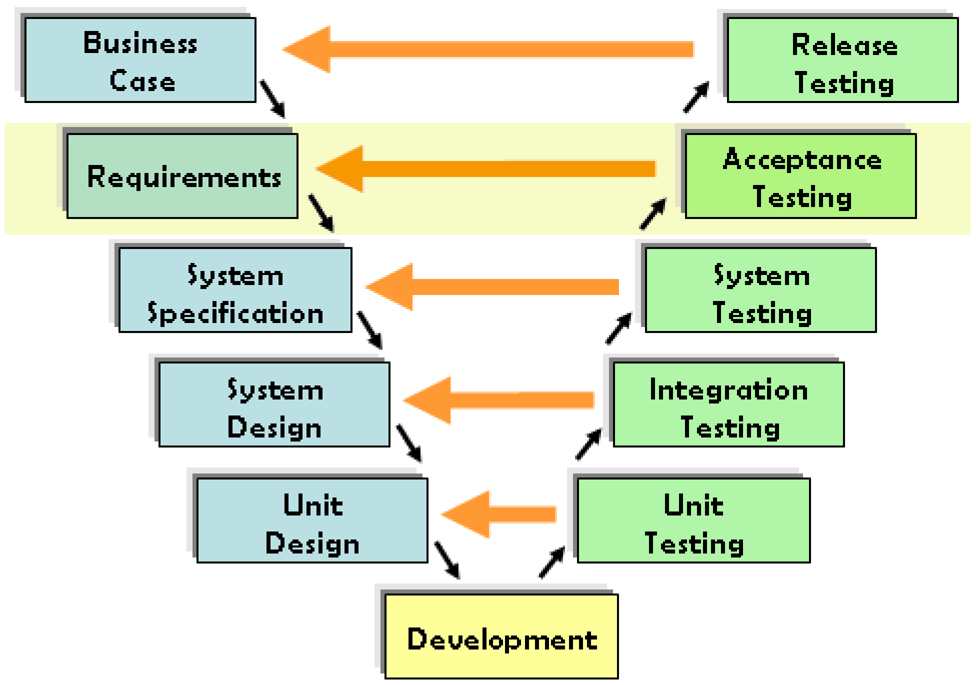


FIG 7.4 Stages of Testing

White Box Testing

White Box Testing shows the flow of working of the whole application. This is a kind of testing process which involves the tester who is expected to have the conscience of the overall structure, internal functioning knowledge and language of software. If not all, at least one of the aspects must be known by software testers. Code is being optimized. The developer performs white-box-testing. It requires a lot of knowledge on the coding part and other parts of the application. It can be done in 2 steps:

1. First, understanding the source code

2. Creating test cases and executing those test cases

Few tools which help to perform white-box-testing are:

a.ParasoftJtest

b.EclEmma

c.NUnit

d.PyUnit

e.HTMLUnit

f.CppUnit

Black Box Testing

Black Box Testing is a type of testing which is mostly based on software requirements and specifications. All these testing reports/results will be mentioned in the document so called “SRS” (Software Requirement Specification) document. The working of software is least bothered in the black box test. It is just concernced with the type of input given and responds to the output. The tester performs black-box-testing. It doesn’t require any deep knowledge on any part of the application. Types of black-box-testing are:

1.Functional

2.Non-functional

3.Regression

Few techniques of black-box-testing :

a.Equivalence Class Testing

b.Boundary Value Testing

c.Decision Table Testing

Test Results: All the test cases mentioned below passed successfully. No defects encountered.

7.5 Test Cases

TEST CASES FOR ENCODING MODULE

S. NO	Test case ID	Test steps	Expected output	Actual output	Result
1.	ID-1	Load the input image	image should be loaded without any errors	Input image is loaded	pass
2.	ID-2	Enter the data	Data should be loaded without any errors	Data notloaded due to invalid path	Fail
3.	ID-3	Encrypt the data	Data should be encrypted	Data is not encrypted	Fail
4.	ID-4	Embed the data into image	Data should be embedded without errors	Data embedded	pass
5.	ID-5	Check the ouput	Should give stego image same plain image	Difference in stego and plain image	Fail

TEST CASES FOR DECODING MODULE

S.NO	Test case ID	Test steps	Expected output	Actual output	Result
1.	ID-1	Upload the stego-image	Stego-image should be uploaded without errors	Stego-image uploaded without errors	pass
2.	ID-2	Decrypt the data from image	Data should be decrypted	Decryption successful	pass
3.	ID-3	Check the output	Hidden data should be displayed as output	Hidden data not displayed	fail

8. CONCLUSION

It is observed that through LSB Substitution Steganographic method, the results obtained in data hiding are pretty impressive as it utilizes the simple fact that any image could be broken up to individual bit-planes each consisting of different levels of information. It is to be noted that as discussed earlier, this method is only effective for bitmap images as these involve lossless compression techniques. Also, in this project grey-scale images have been used for demonstration. But this process can also be extended to be used for color images where, bit plane slicing is to be done individually for the top four bit-planes for each of R, G, B of the message image, which are again to be placed in the R, G, B planes of the cover image, and extraction is done similarly.

9. FUTURE ENHANCEMENT

As a part of security, the pixels of the cover image are filtered both according to their position and the threshold limit. Because of this, the space availability of data insertion could become very less. Therefore, the embedding information should be small for successful embedding. New ideas could be developed on increasing the space availability in the cover image to insert as much data as possible.

8.BIBILOGRAPHY

1.Jin-Suk Kang, Yonghee You, Mee Young Sung. “Steganography using Block-based

Adaptive Threshold”. *Computer Science 2007*.

2. Rengarajan A., Mahalakshmi V., Narendran S., Chandrasekar M., John B Rayappan. “Modulation of Hiding Intensity by Channel Intensity Stego by Pixel Commando” 2012.

3. Amirtharajan R., John B Rayappan. “Tri-Layer Stego for Enhanced Security – A

Keyless Random Approach” 2009.

4. Ushll S., Sathish K., Boopathybagan K., . “A Secure Triple Level Encryption

Method using Cryptography and Steganography” 2011.

5. Ruth M. Davis. *The Data Encryption Standard in Perspective 1999.*

6. Manikandan, G., G.D.R. Krishnan and N. Sairam,. “A unified block and stream cipher based file encryption”. *J. Global Res. Comp. Science 2011*

7. Domenico, B. and I. Luca,. “Image based Steganography and cryptography”.

APPENDIX A: PYTHON

Python is a commonly used, general-purpose programming language, developed by Guido van Rossum, and first published in 1991. .History of Python The Python programming programming language was conceived in the late 1980s and started to be introduced in December 1989 by Guido van Rossum at the CWI (Centrum Wiskunde&Informatica, National Research Institute for Mathematics and Computer Science) in the Netherlands as a successor to the ABC programming language capable of handling exceptions and interfacing with Amoeba operating systemVan Rossum is the principal author of Python, and the title given to him by the Python community, Benevolent Dictator for Life (BDFL), represents his continuing central role in determining the course of Python. Python was called Flying Circus for the BBC TV series Monty Python. Python is derived from many other languages including shell and other scripting languages such as ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix and others. Python retains copyright. As with Perl, source code for Python is now available under the General Public License (GPL).

DEFINITION OF PYTHON

General purpose programming language incorporating paradigms oriented to processes, functions and objects. Python is a free programming language, it is an open source language. Canbe used for autonomous software writing and scripting applications.

FEATURES OF PYTHON

• Open source general-purpose language.

• Object Oriented, Procedural, Functional

• Easy to interface with C/ObjC/Java/Fortran

• Easy-ishto interface with C++ (via SWIG)

• Great interactive environment

Python is a scripting language of high quality, interpreted, collaborative and object-oriented. Python is designed to be extremely legible. It also uses English keywords, where punctuation is used as other languages, and has less syntactic constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not

need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive − You can actually sit at a Python prompt and interact with the

interpreter directly to write your programs.

Python is Object-Oriented − Python supports Object-Oriented style or technique of

programming that encapsulates code within objects.

Python is a Beginner;s Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

APPLICATIONS OF PYTHON

What can you do with Python?

1. Systems programming

- you can write portable, maintainable system

administration tools.

- Some of the python tools can search files, directories, launch other programs.

- Python standard library comes with POSIX bindings which makes support easier for all the OS Tools.

2. GUI

- Python has a standard object oriented interface to TkGUI API called Tkinter.

- Tkinter is used to implement portable GUIs with native look and feel.

- A free extension package PMW adds advanced widets to tkinter toolkit.

- wxPython GUI API based on C++ library an alternative tool for constructing portable GUIs

- Dabo toolkit for GUI

- QT

- PyQT

- GTK

-PyGTK

- MFC

- PyWIN32

-.NET

- Iron Python

-Swings

- Jython or JPype

3. *Internet scripting*

- Python comes with standard internet modules.

- Python can perform a variety of networking tasks.

- Extracting information sent to server-side CGI scripts

- Transfer files by FTP

- Parse and generate XML and JSON documents.

- Send receive compose and parse email

- fetch web pages by URL

- to generate HTML files we can use HTMLGen

- mod_python runs python efficiently on Apache web server

- Jython - used for coding server side applets.

- Django, TurboGears, web2py, pylons, Zope, WebWare support for quick construction of full-featured and production quality websites with python.

- Silverlight tool of microsoft.

4. *Component Integration*

- Integrating a C library into python enables python to test and launch the library's components, and embedding python in a product enables onsite customizations to be coded without having to recompile the entire product. SWIG and SIP tools use cython for scripts.

5. *Database Programming*

- There is a python interface for all commonly used relational databases

- Sybase, Oracle, informix, ODBC, MySQL, PostgreSQL,SQLite. portable database API

- used to access SQL Database systems Pickle module

- non-sql databases can be accessed, it allows programs to easily save and restore entire python objects to files and file like objects

- ZODB, Durus provide OODB for python scripts

PyMango an interface to MangoDB

Pythons Json module to create Json style document database

6. *Rapid prototyping*

- In python, we can initially prototype programs, and then transfer selected components to a compiled language like C or C++. Look the same, as Python and C.

7. *Numeric and Scientific Programming*

- NumPy - Matlab , is a numeric programming tool that can often replace existing code written in C C++. Numeric tools for python support animation, 3D visualization, parallel processing

- SciPy - Provides library for scientific programming. 8. Gaming, images ,data mining, Robotics.

Game programming and multimedia with pygame, cgkit, pyglet, PySoy, Panda3D, and other Windows, Linux, and more Serial port contact with PySerial Image Processing with PIL and its newer Pillow Clone, PyOpenGL, Blender, Maya, and more.

20 Python libraries

1. Requests: This is the most famous http library written by kennethreitz.

2. Scrapy: It is a must-have library for web-scraping.

3. wxPython: It is generally used as a gui toolkit for python. It can also be used in the place of tkinter.

4. Pillow: A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and

is a must have for anyone who works with images.

5. SQLAlchemy: A database library.

6. BeautifulSoup: It is an xml and html parsing library is very useful for

beginners.

7. Twisted: The most important tool for any network application developer. It has a very

beautifulapi and is used by a lot of famous python developers.

8. NumPy: It provides some advanced math functionalities to python.

9. SciPy: It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.

10. Matplotlib: A numerical plotting library. It is very useful for any data scientist or any data analyzer.

11. Pygame: This library helps in 2d game development.

12. Pyglet: A 3d animation and game creation engine. This is the engine in which the famous python port of minecraft was made

13. pyQT: A GUI toolkit for python.

14. pyGtk: Another python GUI library. It is the same library in which the famous Bittorrent client is created.

15. Scapy: A packet sniffer and analyzer for python made in python.

16. pywin32: A python library which provides some useful methods and classes for interacting with windows.

17. nltk. Natural Language Toolkit : It is a very useful library if you want to manipulate strings. But it’s capacity is beyond that.

18. Nose: A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.

19. SymPy: SymPy can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.

20. IPython:.Ithas completion, history, shell capabilities, and a lot more.

Keras

Keras is a deep learning, lightweight Python library that can run on top of either Theano or TensorFlow. It was designed to make implementation for research and development of deep learning models as simple and easy as possible.

It runs on Python 2.7 or 3.5 and, despite the underlying frameworks, can execute smoothly on GPUs and CPU. It is published under the MIT license which is permissive.

Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:

1.Modularity: A model can be seen as a series, or as a graph alone. All of a deep learning model's problems are discrete components which can be combined in arbitrary ways.

2.Minimalism: The library offers enough for an result, no frills and maximizes readability.

3.Extensibilty: ntentionally, new features are simple to incorporate and use within the system, intended for trial testing and the development of new ideas.

4.Python: No different files with Formats of custom files. All at Python is native.

APPENDIX B: UNIFIED MODELING LANGUAGE

The Unified Modeling Language (UML) is defines as a visual modeling language and it can be used for general-purpose. By general purpose, we mean that it can help us to know the detailing about output and artifacts of a software system. We can use UML to construct the artifacts of the system after the specification of artifacts. Once the construction is done, we can visualize and document the artifacts related to software system. It catches the pulse of the system that must be constructed which involves making decisions and understanding everything about the system.

The basic purpose of this language is to have a proper understanding about the system that must be developed. Next comes the remaning tasks which include designing the software. Once this is done configuring the system must be done properly. The final step is maintaining and controlling the information about the system. All the above functionalities collectively form the purpose of UML. It is developed with an intention of involving it in the various procedures that take place within an organization. The methods for development, stages of various software lifecycles and several domains reuired for applications will make of it.

Current object oriented processes which are used for developing the software is supported by UML. This language will not have a standard specification and does not guarantee a fixed procedure for the development, rather this specification is more of iterative development.Visual modeling tools which contain several functionalities like report writers and code generators will make a contribution to commence the purpose of UML. It also invoves several meaningful concepts which are technically termed as semantic concepts and various notations for each of the components present in it. There are also many rules which help us to figure out the representation easily. It consists of 4 parts namely dynamic, static, organizational and environmental parts. It mainly aims at understanding the prior experience with the system and to impart novel and innovative software practices which have proved to be best in several cases.

Several features of the system like static and dynamic behavior, structure of the system and information are captured by UML. System can be described as distinct objects whoch interact among themselves and provides an ultimate solution which is desired by the user. UML will help in modeling such systemswith all the necessary objects. There exits difference between the behavior of the system with respect to the requirements.

Various objects that are essential for implementing a system which should meet the user requirements are defined by the static structure. Apart from this, italso defines the relationships that exist between the defined objects. This description will help to understand what exactly the user needs. Dynamic behavior is different from the static structure. It completely involves in the history of objects and defining the communications that is required among objects to meet the goal.

Involving various separate viewpoints which are related to modelin a system will be of great help for everyone to understand the deep purpose of the system.

UML consists of the organizational constructs which are helpful for organizing the models into different packages relevantly. This arrangement can be useful for the software teams involved in the development of project to break the entire large system into manageable parts which can be easy to work with. This partition made helps the team to dealwith the dependencies among the packages which means they can understand and control such dependencies.Thisdivison also helps in managing the various units of the given model which can be versioned and fit into critical development environment. It also consists of constructs which serve the purpose of representation of the decisions of implementation and for organizing run-time elements into components.

Unlike other programming languages, UML is entirely a different genre language which is distinct in its features and working. It can be termed as a modeling language just as the name suggests.There are various application tools and online tools which can be used to model the system that id to be developed. We can perform 2 processes namely forward engineering and reverse engineering. Forward engineering involves creating an application or a system with the constraints provided by the user and based on the requirements. Reverse engineering is a process of developing the model from the code that is provided.

The application tools or the online tools for UML can generate the codes with the help of code generators from the model developed . This code can be generated in a variety of programming languages.The same tools can be used to again to construct the model from the code proving that the UML tools can be two way purpose.

UML is not so formal language like the existing high level languages. It is not aimed atproving the theorms or such. The ease with this language is that it is simple and easy to understand and can be used for multiple purposes which lack in the contemporary other languages. The notationsand behavior are quite simple and basic which makes it the pouplar modeling language inspite ofmany such exsiting languages. Layouts for graphical interface, several circuit deisgns, the domain of rule based systems and many other domains may require yet more specialized tool with a special language.

The purpose of UML is not to model the random and continuous system which are generally found in physics and engineering. Rather it aims at providing a model which can be suitable for any general purpose systems.

Building Blocks of the UML

There are 3 concepts which are termed to be the most important ones shown below:

·Building Blocks

·Rules

·Common Mechanisms

The below vocabulary of the language will strengthen the building blocks:

·Things

·Relationships

·Diagrams

From the vocabulary specified above, thing are the utmost important ones to be known for knowing the basics of the language.

1. Things in the UML

Things in this language are categorized ito 4 types which are stated below:

·Structural things

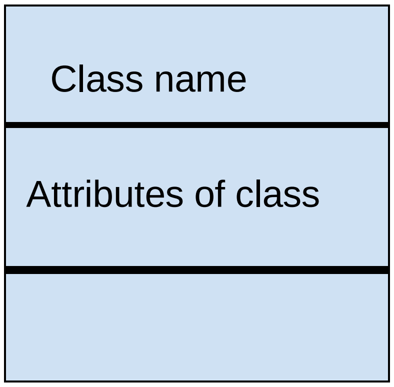
·Behavioral things

·Grouping things

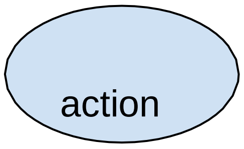
·Annotational things

Structural things: These are mpstly the static parts of a model. They represent the elements which are either conceptual or physical elements. Forrmally we can say that they are the nouns of UML models. These things use several notations to represent which are explained below:

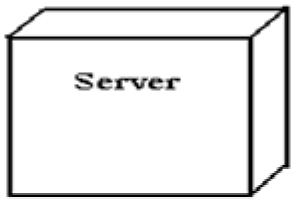
i.Class : It is defined as the description of a set of objects. These objects have many things in common and share the attributes, different operations, several relationships and semantics of the system. The representation of a class is as follows.



i.Use case : It is used to structure the behavioral things in a model.It is basically defined as the description of a sequence of actions that a system performs which results in a value to a particular actor. It is realized by collaboration which is defined in the following pages. It is represented as an ellipse shown below.

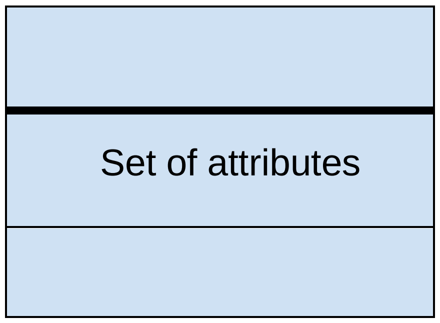


i.Node: It is a physical element and usually exists at runtime.It represents any element or a resource of the system which has an importance at some space in the project and generally it is a representation of computational resources. It also represents the elements which occupy at least some memory. The representation of node is shown below.



Node

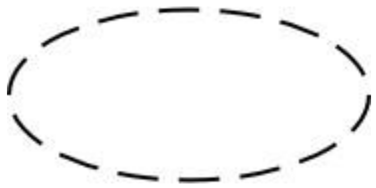
ii.Interface: It defines the extermalyisible behavior of an element. It defines the set of operations but never the set of implementations. These operations specify the service of a class. The representation must contain interface keyword within it. Graphically it is represented as a circle with a name



Interface

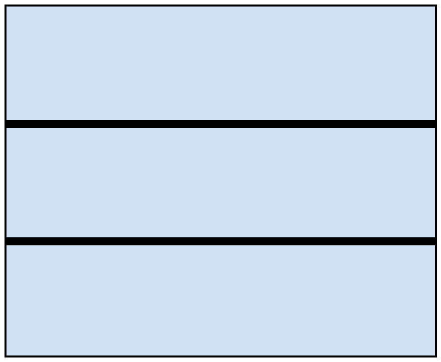
iii. *Collaboration:*

It has structural as well as behavioral dimensions. It is represented as a dashed ellipse. It basically defines an interaction and is a set of roles and all these come together to promote the cordial and cooperative behavior.



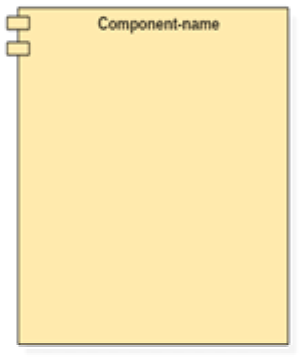
Collaboration

iv. Active Class: It can initiate the control activity. It is a class in general which has few processes ranging from one to many and same with the threads as well.



Active class

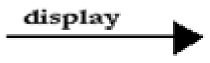
v. Component: It represents physical packaging of logical elements such as classes, interfaces and collaborations. Realization of group of interfaces is provided by the component. It is replacable in general. Component representation has 2 little boxes on the left which differentiates it from the rest.



Component

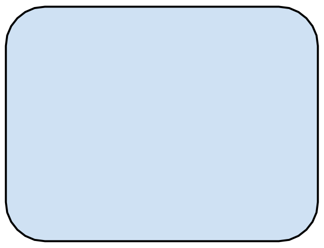
Behavioral things: These models are dynamic parts of the UML. It specifies about the elements which are dependent on time. In general we can say that they rare the verb forms of the language. The behavioral thingsthat we use to represent the models are the following:

i.**Interaction:** It consisits of a group of messages which are communicated among the groups of objects. The meassages are thus exchanged to perform a task that is given. An arrow symbol which is generally a ray is used to represent the interaction.



Interaction

ii. **State Machine:** It represents states of applications at different times. Each state has its own procedure during a particular time. A sequence of several states happenduring the entire lifetime which are specified by the state machine. The representation is a rectangle rounded in the corners.



Relationships in the UML:

They are used to represent the connection and interaction between the things which are specified in the above information. It provides a link between two things that makes us understand the correlation among the various units in the system. There are 4 different types of relationships in UML. They are:

·Dependency

·Association

·Generalization

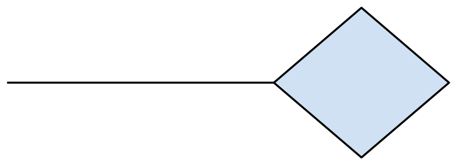
·Realization

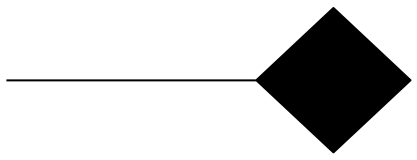
Each relationship has its own significance in the system. Irrespective of the other aspects like the notations of the things, attributes and operations within them, relationships connect these things according to the user requirements specified.

Dependency: As the name suggests, the things are dependent on one another. It can be coined as a semantic relationship which exists between two things. If one of the things change, the other thing will have to automatically reflect the change. The semantics of the dependent thing is modified. Represented by dashed ray.

Dependency

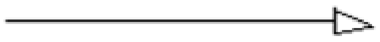
A**ssociation:**It can be presented as a single link or a collection of links among various objects. It is further classified as aggregation and composition. Aggregation represents loose bond that means the part can exist individually irrespective of the whole. Composition is a strong bond where the part cannot exist without the whole. This concept is generally used in most of the dailythings we go through. A simple line is association. Aggregation is a line with its end containinga rhombus. Composition is similar to the above but contains a shaded rhombus.





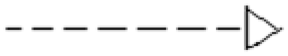
Composition

G**eneralization:** It also includes specialization. This is a kind of relationship in which one element is a parent and another element is the child.Specialization goes more specific from the general description and generalization combines all the elements under one name.



Generalization

Realization: The classifiers are involved in this relation. One of the classifier specifies what is tobe done and the other classifier which is participating in the relation will assure the functioning of contract. The representation is similar to the above one except the dotted lines.



Realization

APPENDIX C: GOOGLE COLAB

Google Colab is one of the most famous cloud services for seasoned data scientists, researchers, and software engineers. While Google Colab seems easy to start, some things are difficult to use. In this guide, you will learn:

Why Colab

Creating a new notebook

Import Notebooks from GitHub/local machine

Google Drive with Colab

Keyboard shortcuts for Colab

Change Language (Python 3 -> Python 2)

Select GPU or TPU

Load Data from Drive

Load Data from Github Repository

Importing External Datasets such as from Kaggle

Download Packages

Bash commands in Colab

Why Colab

There are several benefits of using Colab over using your own local machines. Some of the benefits of Colab are

You do not require to do an environment setup. It comes with important packages pre-installed and ready to use.

Provides browser-based Jupyter Notebooks.

Free GPUStore Notebooks on Google Drive

Importing Notebooks from Github

Document code with Markdown

Load Data from Drive

and much more

Creating a new notebook

To create a new Notebook on Colab, open https://colab.research.google.com/, and it will automatically show your previous notebooks and give an option to create a new notebook**.**

Here you can click on NEW NOTEBOOK to start a new notebook and start running your code in it. By default, it is a Python 3 notebook. I will show you later how to make a new Python 2 notebook.

Alternatively, if for some reason you do not see this prompt, or you canceled it, You can make a new notebook from "File > New Notebook"**.**

All the notebooks you create in Google Colab, are by default stored in your Google Drive. There is a folder in your drive named "Colab Notebooks" where you can find all your notebooks created from Google Colab.

To open a notebook from Drive in Colab, right-click on the desired notebook and "Open With > Google Colaboratory". Similarly, if you are in a notebook in Colab, and want to see it in the drive, you can do it using "File > Locate to Drive", which will redirect you to the notebook position in your Drive. Keyboard Shortcuts for Colab

Most of the keyboard shortcuts of Colab are similar to those of the Jupyter Notebook. Let's look at some important ones.

1. Shortcut to show all the shortcut keys.
- You can see all the shortcut keys with *ctrl + M + H* the key. When looking for any specific shortcut, you can search here.
2. New Code Block
- You can add a new code block by *Ctrl + M + B*.
3. Convert Cell to Markdown
- You can convert a cell to Markdown by using *Ctrl + M + M* shortcut key.

4. Convert from Markdown to Code

- To convert from Markdown to Code, you can use *Ctrl* + *M* then quickly press *Y*. I can say it as *Ctrl* + *M* followed by *Y*. Remember not to press them altogether but follow the order.

There are many other great shortcut keys that you can search using *Ctrl* + *M* + *H* key.

Change Language from Python 3 to Python 2

Now with the official end of support of Python 2, Python 2 is no longer available at Colab. Now for some reason, someone sends you a python 2 code, or you have to check something in Python 2 quickly, all you can do is to go to these links

- http://bit.ly/colabpy2
- http://colab.fan/py2

which will eventually redirect you to this https://colab.research.google.com/notebook#create=true&language=python2 link from where you can test your Python 2 code.

Hardware Selection (GPU or TPU)

Colab's biggest advantage is that it provides free support to GPU and TPU. You can easily select GPU or TPU for your program by Runtime > Change runtime type.

After clicking on the Change runtime type, you will be prompted to select GPU or TPU.

You can also use your local GPU with Colab Interface. In order to do that, do the following steps.

Start a Jupyter notebook instance at your local machine.

Copy its URL with Token

Click on the arrow here(It may be written 'connect' in some cases)

Click on connect to local runtime

After clicking, you will see this prompt

Paste the link here, and connect it

Load Data from Drive

You can easily load data from Google Drive by mounting it to the notebook. To do this, type the following code in your notebook.

```
from google.colab import drive
drive.mount('gdrive')
```

It will give you a link to open,

Go to the link

Login to your Google Account

Copy the code

Paste it in notebook

Now if you see in your “Files” section, you will find your ‘gdrive’.

Let's say you have uploaded is under the Untitled folder.

You can get it by

```
myfile = open('gdrive/My Drive/Untitled folder/dataset1.csv)
```

And you get your file.

Load data from Github

Let’s say your data set is in Github, and you want to load it. You can do it simply by downloading it as you download it normally in your machine

```
!git clone REPOLINK
%cd REPONAME
```

If it is in zip format, you can unzip it by

```
!unzip GPR_radargrams.zip
%cd GPR_radargrams
```

And now you have access to data. Later in this tutorial, I will show you how to use bash commands, after that this part will make more sense and easy to use for you.

Import Dataset from Kaggle

Step 1 is to get a Kaggle API Token. Go to My account > API > Create New API Token. Go to API and click on Create New API token. You will receive a JSON file which you can save it.

Install Kaggle Library and Import Google Colab Files in your notebook.

```
from google.colab import files
!pip install -q kaggle
```

Upload your Kaggle API JSON file to Colab.

```
uploaded = files.upload()
```

Go to Dataset you want to download, and click on copy API command, under 3 dots.

Copy the command and put a '!' before it and run it on Colab, i.e.,

```
!kaggle datasets download -d ruchi798/malnutrition-across-the-globe
```

and it will download the dataset for you.

Download Packages

Although all important packages such as Tensorflow, PyTorch, Numpy, Pandas, etc are installed, you can install further packages or upgrade current ones.

To download the packages, you can simply use the command through which you download datasets in your local machine with '!' in the start.

Let's say you want to download the pillow package in your Colab (though it will be downloaded by default), just type the following code and run the cell.

```
!pip install pillow
```

If it is already downloaded, it will show

otherwise, it will download it for you.

To update a package, let's say TensorFlow, use following

```
!pip install tensorflow --upgrade
```