MYSTEGO

A STEGANOGRAPHY TOOL

BATCH NUMBER-11

NISHATH-21311A6237

S.HANSIKA-21311A6247

CH.HRUDAY RAO-21311A6257

UNDER GUIDANCE OF MR.GNANESHWAR

REQUIREMENTS

Hardware

- multi-core processor
- ram - 8GB
- 500 GB Storage

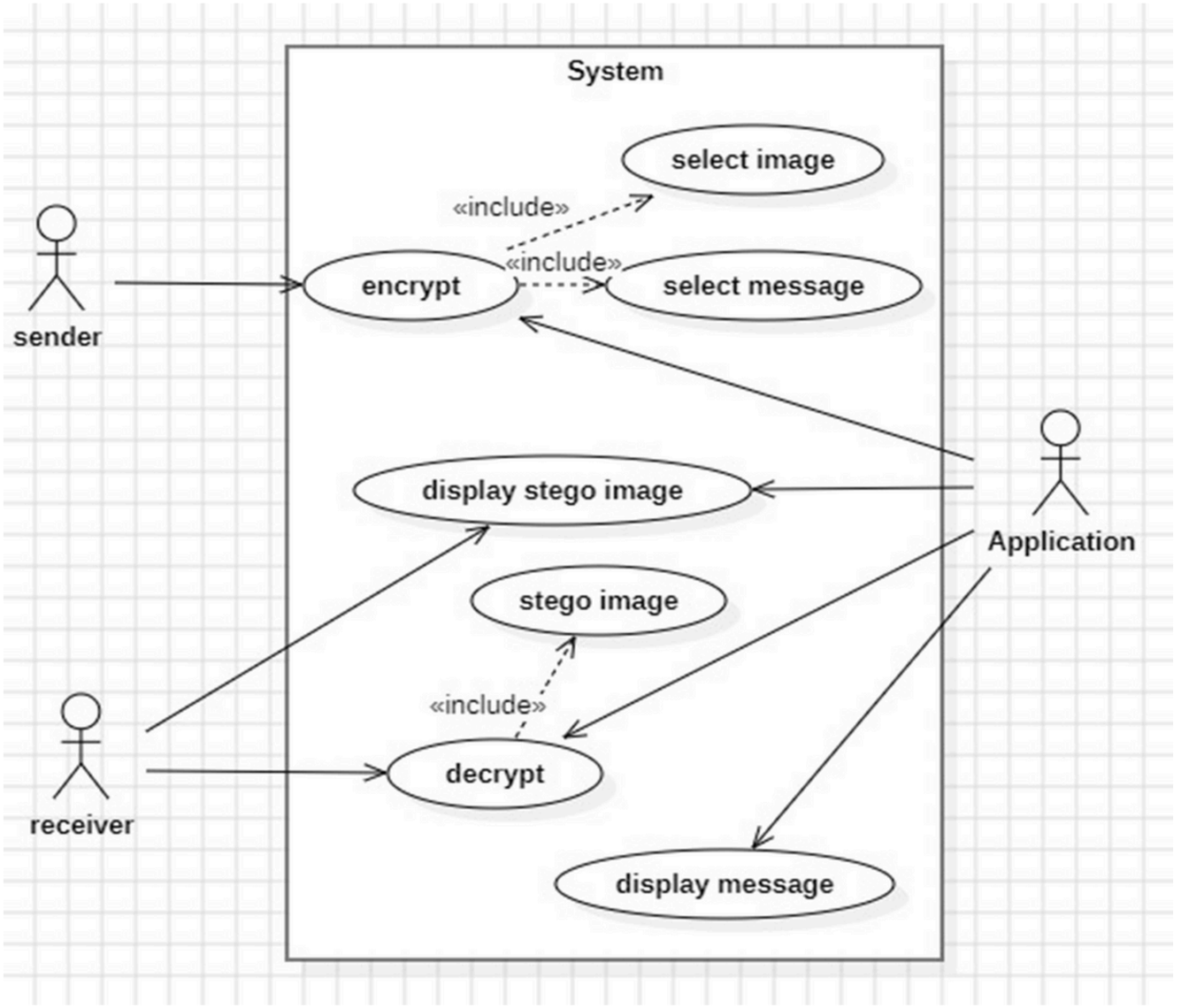
Software

- Tkinter
- Python
- Javascript
- VS Code

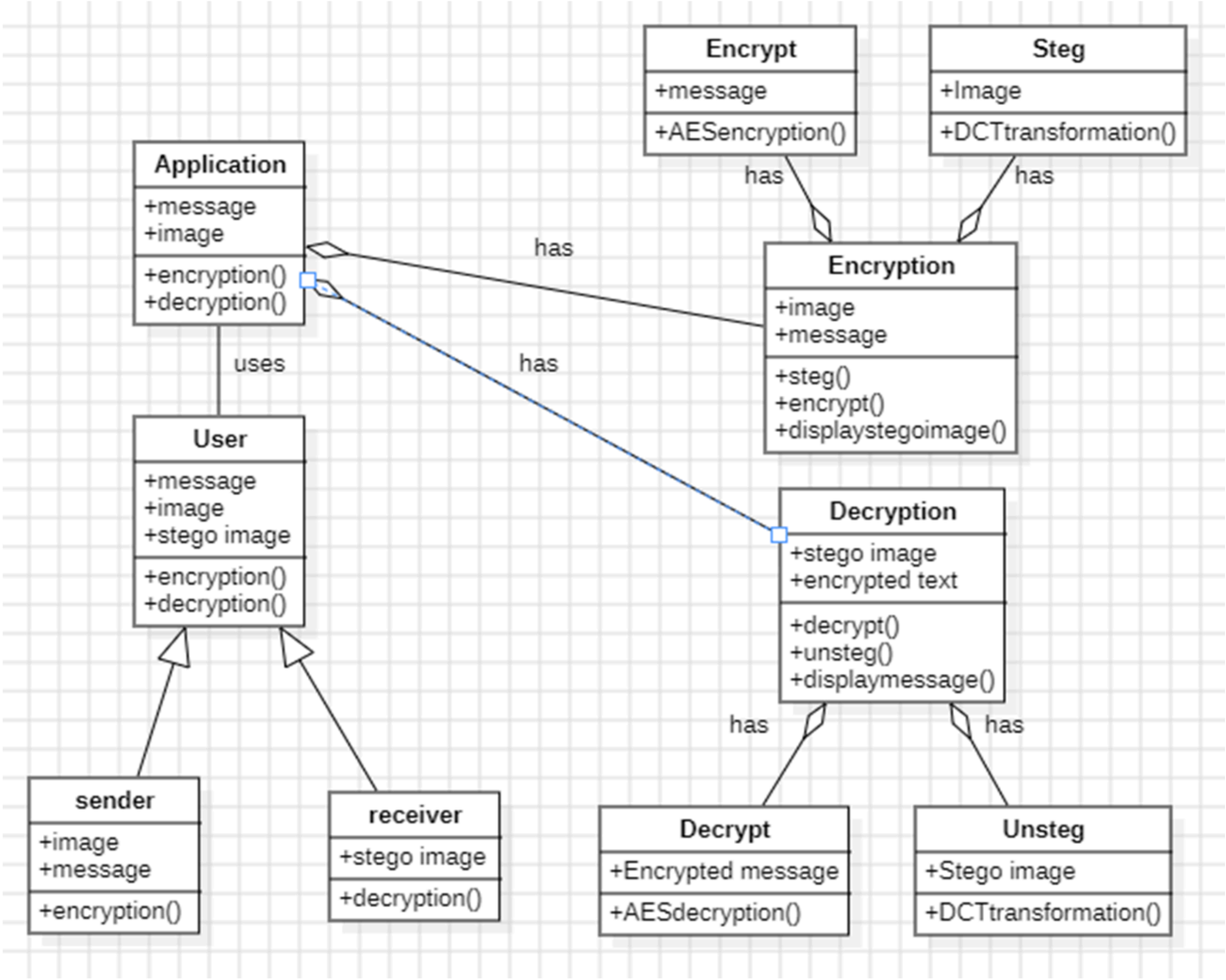
ABSTRACT

Image steganography is a powerful technique for hiding secret information within digital images, ensuring covert and secure communication. By modifying image data in a way that is invisible to the naked eye, it allows for the discreet transmission of sensitive information. This method is crucial for protecting privacy and maintaining confidentiality across various fields, including personal communications, corporate data protection, and government operations. Its applications also extend to digital watermarking for copyright protection and data integrity verification.

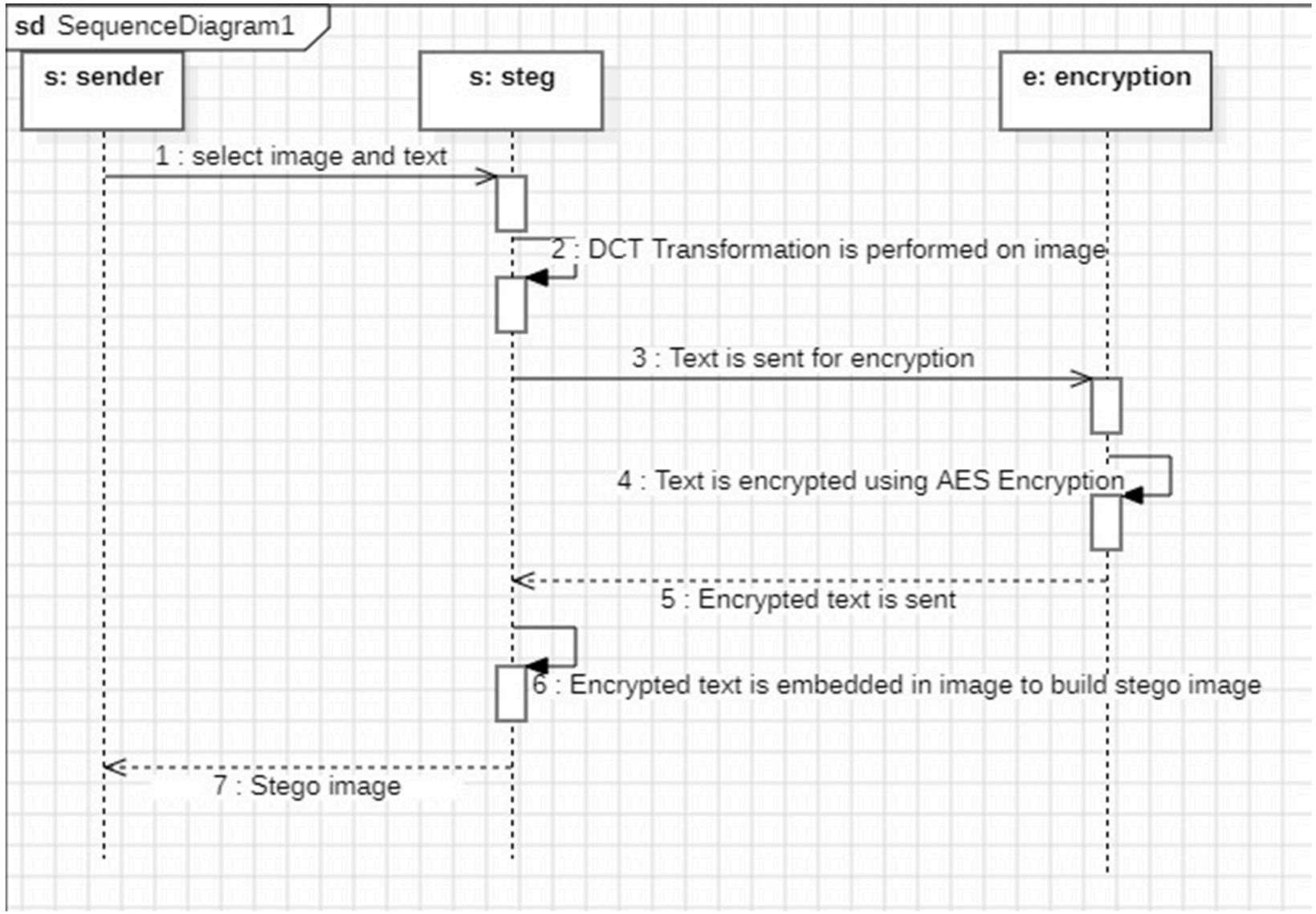
Use Case Diagram:-



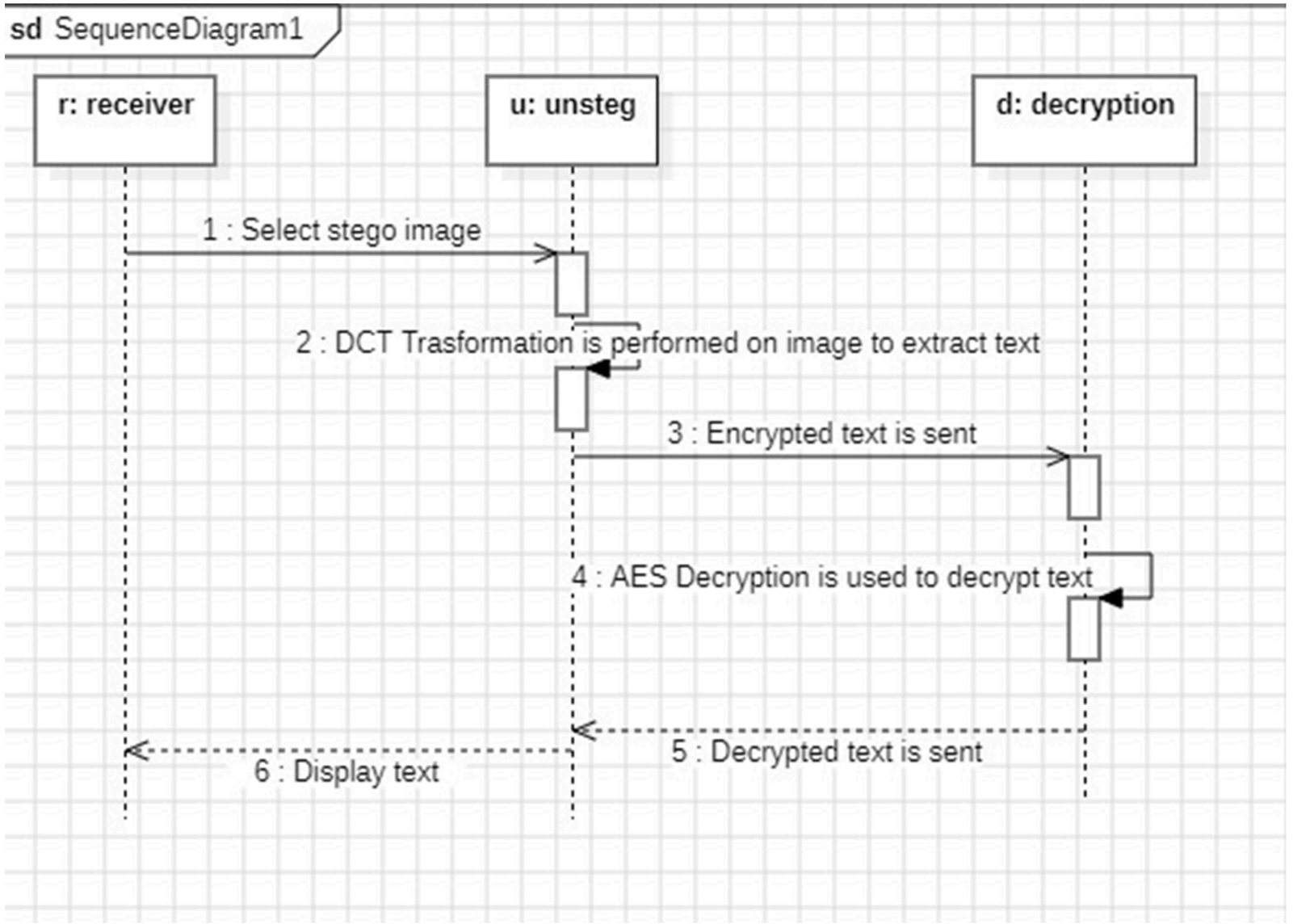
Class Diagram:-



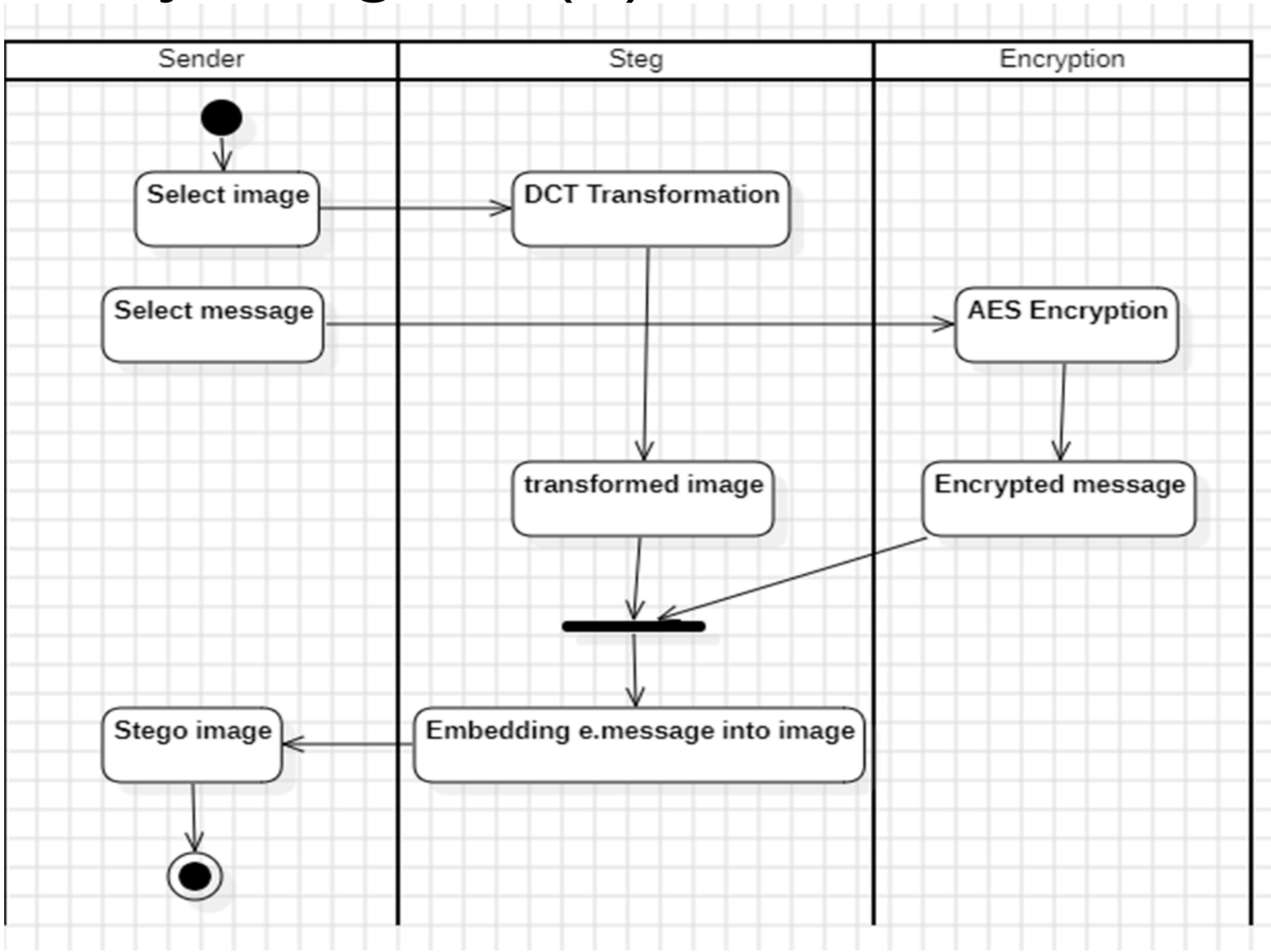
Sequence Diagram(1):-Sender



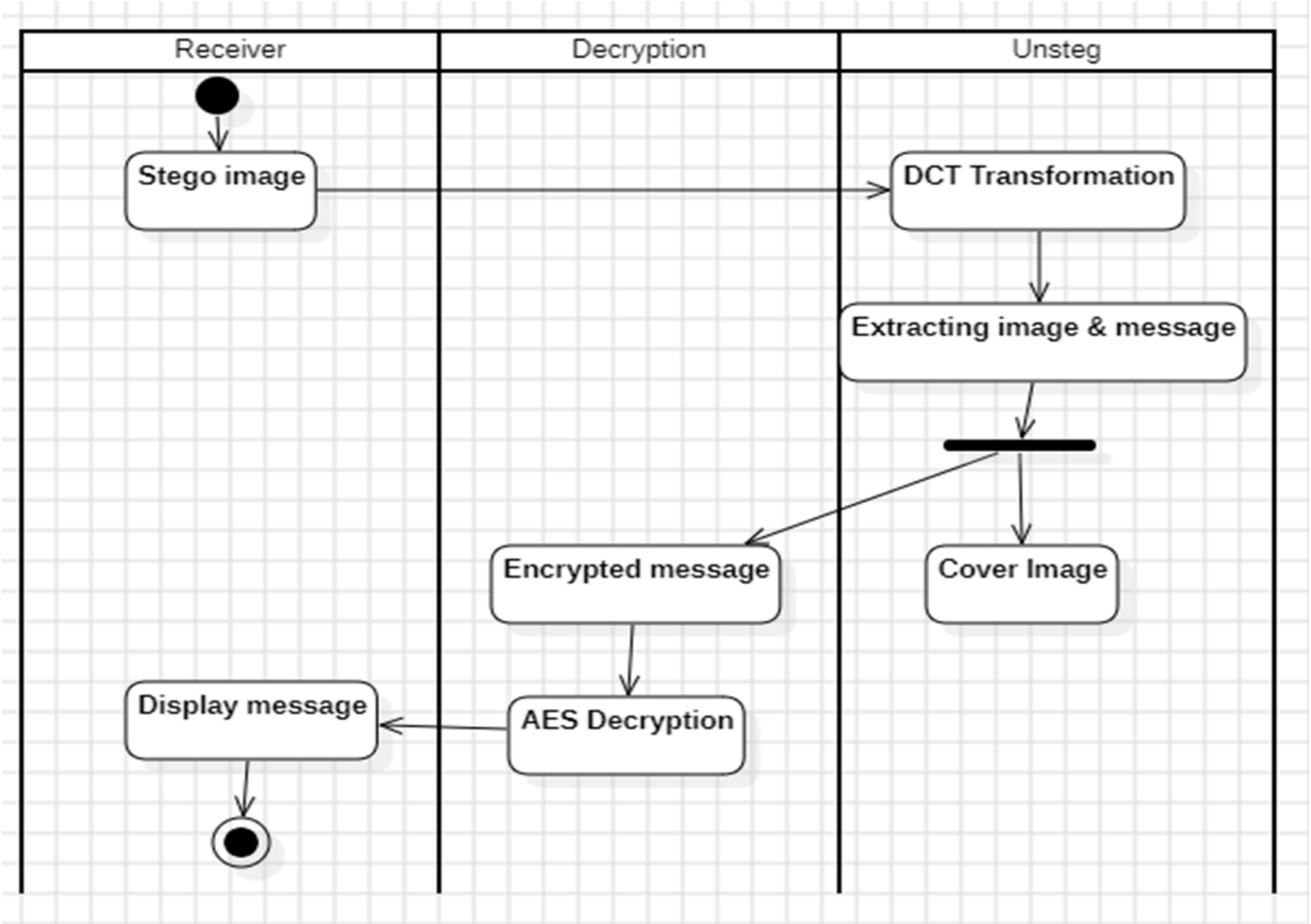
Sequence Diagram(2):-Receiver



Activity Diagram(1):-Sender



Activity Diagram(2):-Receiver



EXISTING SYSTEM

- Least Significant Bit (LSB) Insertion.
- Patchwork.
- Block-based Methods.
- Spread Spectrum Techniques.

LIMITATIONS

- Security.
- Robustness.
- Capacity.
- Statistical Detectability.

PROPOSED SYSTEM

Image steganography using DCT and AES provides robust security by embedding encrypted data within image frequency components, making it less detectable. This technique enhances privacy by securely hiding sensitive information in everyday images. It aids in confidential communication, ensuring that data remains hidden from unauthorized access. Furthermore, it supports digital rights management by embedding copyright information in a tamper-proof manner.

APPROACH

METHODLOGIES

- Preprocessing
- Transmission
- Data Embedding
- Data Extraction

ALGORITHMS

- AES Encryption
- DCT Transformation
- AES Decryption

EXPLANATION

1

AES ENCRYPTION

Encrypt the text message using AES to ensure its confidentiality and security.

2

DCT TRANSFORMATION

Apply DCT to the cover image to convert it from the spatial domain to the frequency domain.

3

DATA EMBEDDING

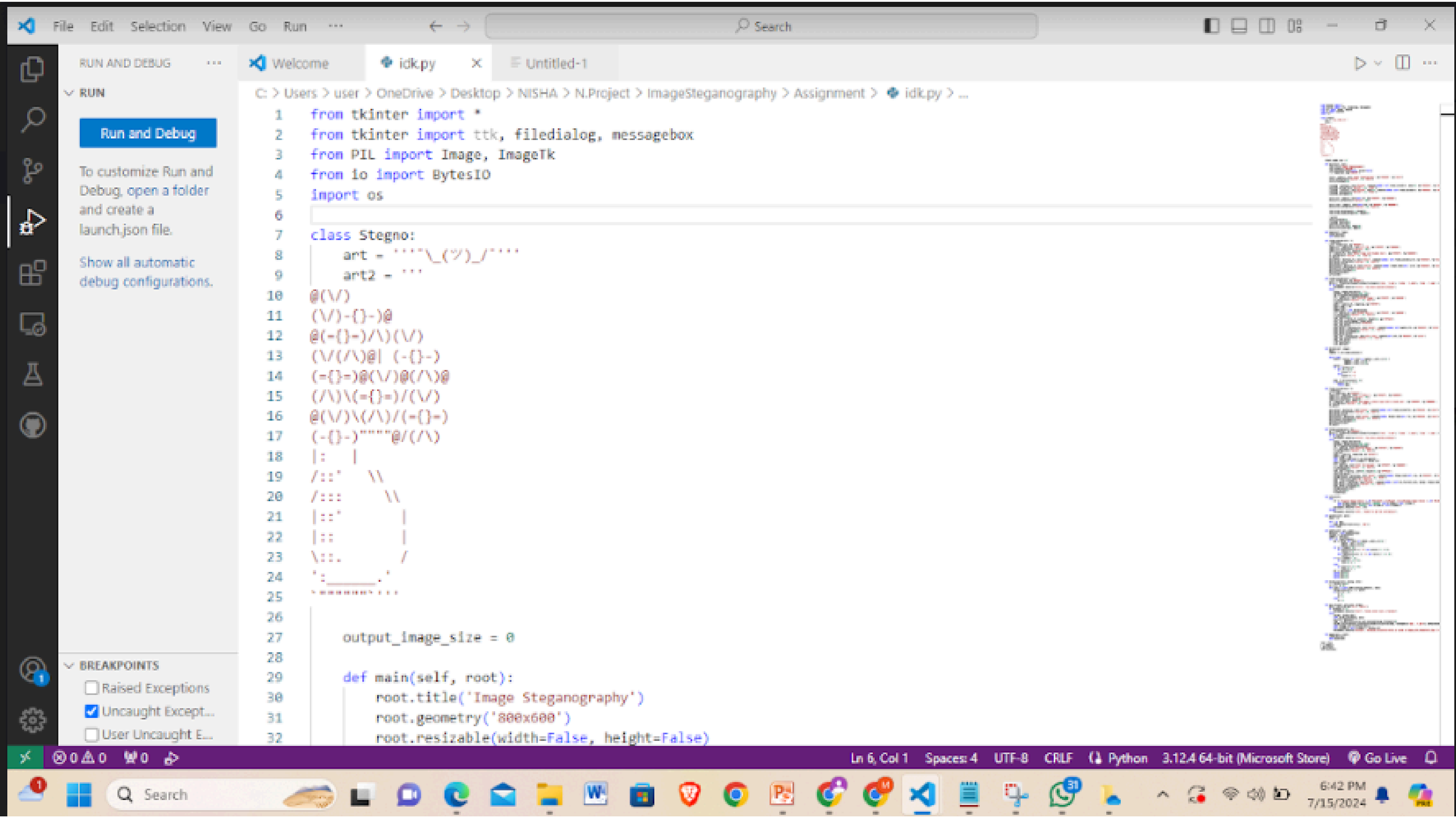
Embed the encrypted text into the DCT coefficients of the image.

4

TRANSMISSION AND EXTRACTION:

Transmit the stego image to the receiver. At the receiver’s end, the DCT is applied again, and the hidden data is extracted and decrypted using AES.

CODE SNIPPETS


File Edit Selection View Go Run
Search
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug, open a folder and create a launch.json file.
Show all automatic debug configurations.
Welcome
idk.py
Untitled-1
C: > Users > user > OneDrive > Desktop > NISHA > N.Project > ImageSteganography > Assignment > idk.py > ...
from tkinter import *
from tkinter import ttk, filedialog, messagebox
from PIL import Image, ImageTk
from io import BytesIO
import os
class Stegno:
art = ''''_(ツ)_/''''
art2 = '''
@(\/)
(\/)-{}-)@
@(-{}-)/\)(\/)
(\/(/\)@| (-{}-)
(-{}-)@(\/)@(/\)@
(/\)\(-{}-)/(\/)
@(\/)\(/\)/(-{}-)
(-{}-)"""@/(/\)
|: |
/::' \\
/::: \\
|::' |
|:: |
\::. /
':_____.'
`"""""""'''
output_image_size = 0
def main(self, root):
root.title('Image Steganography')
root.geometry('800x600')
root.resizable(width=False, height=False)
BREAKPOINTS
Raised Exceptions
Uncaught Except...
User Uncaught E...
Ln 6, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.12.4 64-bit (Microsoft Store)
Go Live
Search
6:42 PM
7/15/2024

File Edit Selection View Go Run ··· Search

RUN AND DEBUG ···

RUN

Run and Debug

To customize Run and Debug, open a folder and create a launch.json file.

Show all automatic debug configurations.

BREAKPOINTS

- [ ] Raised Exceptions
- [x] Uncaught Except...
- [ ] User Uncaught E...

Welcome | idk.py | Untitled-1

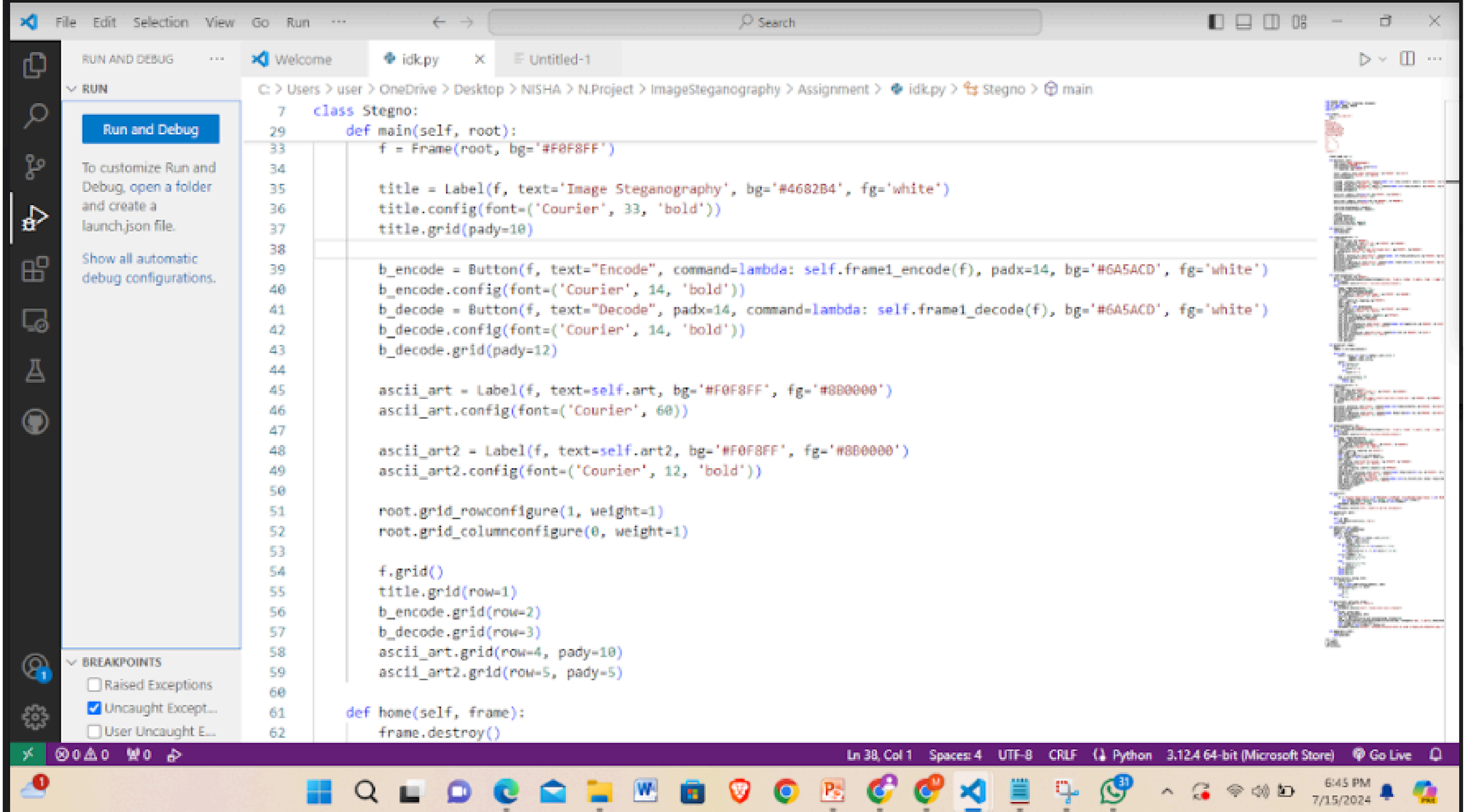
C: > Users > user > OneDrive > Desktop > NISHA > N.Project > ImageSteganography > Assignment > idk.py > ...

```python
from tkinter import *
from tkinter import ttk, filedialog, messagebox
from PIL import Image, ImageTk
from io import BytesIO
import os

class Stegno:
    art = '''"\_(ツ)_/"'''
    art2 = '''
@(\/)
(\/)-{}-)@
@(-{}-)/\)(\/)
(\/(/\)@| (-{}-)
(-{}-)@(\/)@(/\)@
(/\)\(-{}-)/(\/)
@(\/)\(/\)/(-{}-)
(-{}-)"""""@/(/\)
|:   |
/::'   \\
/:::     \\
|::'       |
|::        |
\::.       /
':_____.'
`"""""""`'''

    output_image_size = 0

    def main(self, root):
        root.title('Image Steganography')
        root.geometry('800x600')
        root.resizable(width=False, height=False)
```


File Edit Selection View Go Run
Search
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug, open a folder and create a launch.json file.
Show all automatic debug configurations.
BREAKPOINTS
Raised Exceptions
Uncaught Except...
User Uncaught E...
Welcome
idk.py
Untitled-1
C: > Users > user > OneDrive > Desktop > NISHA > N.Project > ImageSteganography > Assignment > idk.py > Stegno > main
class Stegno:
def main(self, root):
f = Frame(root, bg='#F0F8FF')
title = Label(f, text='Image Steganography', bg='#4682B4', fg='white')
title.config(font=('Courier', 33, 'bold'))
title.grid(pady=10)
b_encode = Button(f, text="Encode", command=lambda: self.frame1_encode(f), padx=14, bg='#6A5ACD', fg='white')
b_encode.config(font=('Courier', 14, 'bold'))
b_decode = Button(f, text="Decode", padx=14, command=lambda: self.frame1_decode(f), bg='#6A5ACD', fg='white')
b_decode.config(font=('Courier', 14, 'bold'))
b_decode.grid(pady=12)
ascii_art = Label(f, text=self.art, bg='#F0F8FF', fg='#8B0000')
ascii_art.config(font=('Courier', 60))
ascii_art2 = Label(f, text=self.art2, bg='#F0F8FF', fg='#8B0000')
ascii_art2.config(font=('Courier', 12, 'bold'))
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)
ascii_art.grid(row=4, pady=10)
ascii_art2.grid(row=5, pady=5)
def home(self, frame):
frame.destroy()
Ln 38, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit (Microsoft Store) Go Live
6:45 PM
7/15/2024

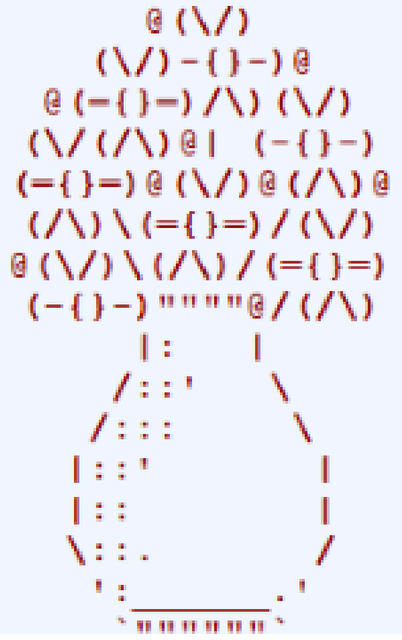
OUTPUT SNIPPETS

Image Steganography

Encode

Decode

¯_(ツ)_/¯



' \ (° Ω °) / '

Select the Image in which you want to hide text:

Select

Cancel

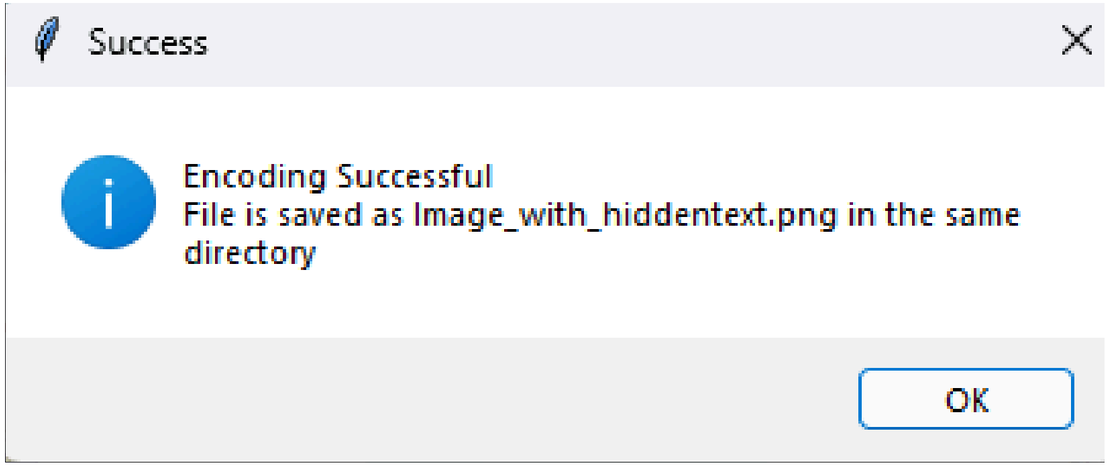
Selected Image:

Enter the message:

Image Steganography

Encode

Cancel



٩(^‿^)۶

Select Image with Hidden text:

Select

Cancel

Selected Image:

Hidden data is:

Image Steganography

Cancel

More Info

ADVANTAGES

- Enhanced Security
- Data Integrity
- Versatility
- Wide Range of Applications
- Low Overhead
- Ease of Use
- Stealth and Subtlety
- Complementary to Other Security Measures

USE CASES

- Secure Communications
- Digital Rights Management (DRM)
- Data Concealment
- Digital Watermarking
- Authentication and Integrity Verification
- Covert Channels
- Steganographic File Systems
- Censorship Circumvention
- Confidential Information Storage
- Intellectual Property Protection

CONCLUSION

In summary, this steganography project demonstrates how to securely hide information within digital media, such as images, audio, and video files. By effectively embedding and extracting hidden messages, steganography enhances security and privacy without attracting attention. The project highlights key benefits, including covert communication, data integrity, and versatile applications across various fields. When combined with encryption, it provides a robust, layered security approach. Steganography is a valuable tool in cybersecurity, enabling secure communication and data protection in today's digital landscape.

REFERENCES

- Image Steganography Articles
- GitHub

BIBLIOGRAPHY

- GitHub Katzenbeisser, S., & Petitcolas, F. A. (1999). "Information Hiding Techniques for Steganography and Digital Watermarking." Artech House.
- Comprehensive coverage of various steganography and digital watermarking techniques.

THANK YOU