A PROJECT REPORT ON

A VIDEO STEGANOGRAPHY METHOD BASED ON TRANSFORM BLOCK DECISION

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

VIKRANT GAIKWAD	**EXAM NO. B190424254**
NIKHIL KOLHE	**EXAM NO. B190424313**
RUTUJA HUNDEKARI	**EXAM NO. B190424275**
SWARAJ KAKADE	**EXAM NO. B190424293**

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

STES'S SINHGAD INSTITUTE OF TECHNOLOGY

KUSGAON (BK), LONAVALA, PUNE- 410401

SAVITRIBAI PHULE PUNE UNIVERSITY
2023 -2024

Sinhgad Institutes

CERTIFICATE

This is to certify that the project report entitles

"A VIDEO STEGANOGRAPHY METHOD BASED ON TRANSFORM BLOCK DECISION"

Submitted by

VIKRANT GAIKWAD EXAM NO. B190424254
NIKHIL KOLHE EXAM NO. B190424313
RUTUJA HUNDEKARI EXAM NO. B190424275
SWARAJ KAKADE EXAM NO. B190424293

Are the bonafide students of this institute and the work has been carried out by him/her under the supervision of **Prof. S.S. WAGH** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

(Prof. S.S.Wagh)
Guide
Department of Computer Engineering

(Dr. S.D. Babar)
Head
Department of Computer Engineering

(External Examiner)

(Dr. M. S. Gaikwad)
Principal
Sinhgad Institute of Technology,
Lonavala, Pune – 410401

Place: Lonavala

Date:

ACKNOWLEDGEMENT

It has been the light of the day due to valuable contribution of certain individuals who are constant guidance, support and encouragement resulted in the realization of our project.

We would like to take this opportunity to thank our internal guide **Prof. S.S.Wagh** for giving us all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful.

We are grateful to **Prof. S. D. Baba**r, Head of Computer Engineering Department, Sinhgad Institute of Technology, Lonavala for his indispensable support, suggestions.

We are thankful to **Dr. M. S. Gaikwad**, Principal of Sinhgad Institute of Technology, Lonavala for providing a healthy environment in the college, which help us in concentrating on our task.

We would also like to thank all the staff members of our department, without whose constructive suggestions and valuable advice, the simple idea, which had borne by us, would not have been able to blossom forth to give such a beautiful bloom. Last but not the least; we are grateful to all our friends and our parents for their direct or in direct constant moral support throughout the course of this project.

VIKRANT ARVIND GAIKWAD
NIKHIL RAJU KOLHE
RUTUJA NITIN HUNDEKARI
SWARAJ RAJESH KAKADE

ABSTRACT

Steganography is an artful technique employed to mask the transmission of information within seemingly innocuous data, ensuring covert communication. Among the plethora of carrier file types, digital photos stand out as the favored medium due to their ubiquity on the web. However, in the realm of concealing sensitive information within video frames, steganography techniques diversify in complexity, each offering distinct advantages and limitations. Some methodologies prioritize complete invisibility of the embedded data, while others accommodate larger payloads, creating a nuanced landscape for information concealment within visual media.

Our final project abstract delves into the realm of video steganography, specifically focusing on a transformative block decision approach. This method enables users to select bits for replacement rather than resorting to the conventional least significant bit (LSB) replacement, thereby enhancing the security and robustness of the concealment process. By leveraging this innovative strategy, our project aims to provide a safer and more efficient means of embedding secret messages within video content, catering to diverse requirements ranging from imperceptible concealment to accommodating larger covert payloads.

The core essence of our project lies in the fusion of advanced stenographic techniques with the dynamic nature of video frames, harnessing the power of transformative block decisions to conceal information seamlessly. This approach not only ensures a higher level of security by deviating from traditional LSB-based methods but also opens avenues for tailored concealment strategies based on user preferences. By bridging the gap between sophisticated concealment needs and user-defined parameters, our project represents a significant advancement in video steganography, promising enhanced versatility and effectiveness in concealing sensitive information within the visual domain.

TABLE OF CONTENTS

LIST OF ABBREVATIONS i
LIST OF FIGURES ii
LIST OF TABLES iii

Sr. No.	Title of Chapter		Page No.
01	Introduction		
1.1	Overview		
1.2	Motivation		
1.3	Problem Definition and Objectives		
1.4	Project Scope & Limitations		
1.5	Methodologies of Problem solving		
02	Literature Survey		
03	Software Requirements Specification		
3.1	Assumptions and Dependencies		
3.2	Functional Requirements		
	3.2.1	System Feature 1(Functional Requirement)	
	3.2.2	System Feature 2(Functional Requirement)	
	3.2.3	System Feature 2(Functional Requirement)	
3.3	External Interface Requirements (If Any)		
	3.3.1	User Interfaces	
	3.3.2	Input & Output	
	3.3.3	Integration	
	3.3.4	Hardware Requirements	
3.4	Nonfunctional Requirements		
	3.4.1	Performance Requirements	
	3.4.2	Safety Requirements	
	3.4.3	Security Requirements	
	3.4.4	Software Quality Attributes	
3.5	System Requirement		
	3.5.1	Database Requirements	
	3.5.2	Software Requirements (Platform Choice)	
	3.5.3	Hardware Requirements	
3.6	Analysis Models: SDLC Model to be applied		
3.7	System Implementation		
04	System Design		
4.1	System Architecture		
4.2	Data Flow Diagram		
4.3	Entity Relationship Diagrams		
4.4	UML Diagrams		
05	Project Plan		
5.1	Stakeholder List		
	5.1.1	Project Type	

		5.1.2	Project Resources	
	5.2	Risk Management		
		5.2.1	Risk Identification	
		5.2.2	Risk Analysis	
		5.2.3	Overview of Risk Mitigation, Monitoring, Management	
	5.3	Project Schedule		
		5.3.1	Project Task Set	
		5.3.2	Task Network	
		5.3.3	Timeline Chart	
	5.4	Team Organization		
		5.4.1	Team structure	
		5.4.2	Management reporting and communication	
06		Project Implementation		
	6.1	Overview of Project Modules		
	6.2	Tools and Technologies Used		
	6.3	Algorithm Details		
		6.3.1	Algorithm 1: LSB	
		6.3.2	Algorithm 1: AES	
07		Software Testing		
	7.1	Type of Testing		
	7.2	Test Strategy		
	7.3	Test Plan		
	7.4	Software to be Tested		
	7.5	Test Cases		
08		Results		
	8.1	Screen Shots		
09		Conclusions		
	9.1	Conclusions		
	9.2	Future Work		
	9.3	Applications Appendix		
		Appendix A: Key Benefits of video steganography, Importance of testing in video Steganography, Challenges & Consideration, Future Trends, and Applications of Video Stegnography. Appendix B: Details of paper publication: name of the conference/journal, comments of reviewers, certificate, and paper. Appendix C: Plagiarism Report of project report.		
		References		

LIST OF ABBREVATIONS

ABBREVIATION	ILLUSTRATION
VPN	Virtual Private Network
IP	Internet Protocol
IDS	Intrusion Detection System
TCP	Transmission Control Protocol

LIST OF FIGURES

FIGURE	ILLUSTRATION	PAGE NO.
1.1	System Overview	3
1.2	System Behavior	5
2.1	TCP Header	11
4.1	Waterfall Model	27
4.2	Timeline Chart	30
4.3	DFD Level – 0	31
4.4	DFD Level – 1	32
4.5	DFD Level – 2	33
4.6	Use case Diagram	34
4.7	Sequence Diagram	35
4.8	ER Diagram	36
4.9	Class Diagram	37
4.10	Component Diagram	38
4.11	Deployment Diagram	39
4.12	State Machine Diagram	40

LIST OF TABLES

TABLE	ILLUSTRATION	PAGE NO.
4.1	Project Plan	29
3.1	Packet Information	47
3.2	Network Error	48
3.3	IP Configuration	48

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

The Greek word ”steganography” denotes obscured writing Stegano means ”covered” in Greek and writing is pictorial. Consequently, steganography not just the practice of concealing data but also the actual actuality of transmitting sensitive in- formation. Using steganography, the insert important information in another manner so that just the recipient is aware of the message’s existence in the past. At one point, the data was shielded by being concealed on the writing desks, a rabbit’s tummy, or the back of wax the slaves’ headshakes. However, the majority of individual’s now a- day’s transfer data using text, graphics, or other visuals. Over the media, video and audio. As a way to safely transmission of sensitive information, multimedia Images, video, and audio files are employed as .The influence of the internet on public life is growing with time. Even if its users are accustomed to using it, the attacker poses a serious threat to sensitive data. Steganography’s driving goal is to enable clandestine communication between two parties. In order to prevent the intruder from discovering the secret information, the secret data (essential information) is hidden in the cover and communicated. The Human Visual system will treat the stegano object as a single unit .Encryption is also carried out to further increase the security of the sensitive data. Perceptual transparency, capacity, and tamper resistance are factors to be taken into account in any stenographic systems. The capacity to conceal the existence of sensitive data is known as perceptual transparency.

The method leverages the transform block decision process in video compression to embed secret data in video frames. This allows for high data hiding capacity while maintaining video quality.

The method randomly selects pixels within video frames using the knight tour algorithm, rather than serially selecting pixels for data embedding. This improves security and robustness compared to basic LSB (least significant bit) techniques.

The method employs a key function encryption to further enhance the security of the data embedding process.

1.2 MOTIVATION

The scope of the project is to limit unauthorized access and supply better security during message transmission. To meet the wants, I exploit the straightforward and basic approach of steganography.

During this project, the proposed approach finds the acceptable algorithm for embedding the info in a picture using steganography which provides the higher security pattern for sending messages through a network. Requirement of this steganography system is that the hider message carried by Stego-media shouldn't be sensible to citizenry. The opposite goal of steganography is to avoid drawing suspicion to the existence of a hidden message. This approach of data hiding technique has recently become important during a number of Application areas.

The motivation behind a video steganography method based on transform block decision lies in the pursuit of enhancing data security and privacy in video communication. By utilizing innovative techniques like transform block decisions, randomized pixel selection, and motion vector preservation, this method aims to achieve high data hiding capacity while maintaining video quality and robustness against attacks.

This video steganography method is to provide a sophisticated, high-capacity, and secure means of embedding secret data in video streams, catering to the increasing need for privacy and confidentiality in digital communication.

1.3 PROBLEM DEFINITION

Steganography is the method of hiding any secret information like password, text and image, audio behind original cover file. Original message is converted into cipher text by using secret key and then hidden into the LSB of original image. The proposed system provides audio-video crypto steganography which is the combination of image steganography and audio steganography using Forensics Technique as a tool to authentication. The main aim is to hide secret information behind image and audio of video file.

1.4 PROJECT SCOPE & LIMITATIONS

The project scope and limitations of a video steganography method based on transform block decision involve defining the boundaries and extent of the project's objectives and constraints. The scope typically outlines what the project aims to achieve, while the limitations highlight the boundaries or restrictions within which the project operates. In the context of a video steganography method based on transform block decision, the scope may include aspects such as the development of a novel steganography technique using transform block decisions to embed data in video frames, enhancing data security and privacy in video communication.

The limitations could involve constraints like the computational complexity of the transform block decision process, potential degradation of video quality due to data embedding, and the need for robustness against various attacks while maintaining high payload capacity.

This project would focus on leveraging transform block decisions to embed data in video frames, ensuring secure communication through steganography techniques tailored for video content. The limitations may revolve around the trade-off between data hiding capacity and video quality, the computational resources required for real-time implementation, and the need to address potential vulnerabilities in the steganography method when applied to video data.

1.5 METHODOLOGIES OF PROBLEM SOLVING IN VIDEO STEGNOGRAPHY

Video steganography involves concealing information within video files to avoid detection. Various methodologies are employed to address the challenges inherent in this process. One common approach is the Least Significant Bit (LSB) insertion, which involves replacing the least significant bits of pixel values in video frames with bits of the secret message. This method is straightforward and quick but can be easily detected through simple statistical or visual analysis. More sophisticated techniques operate in the transform domain, where information is hidden within transformed coefficients of video frames rather than raw pixel values. For instance, the Discrete Cosine Transform (DCT) method embeds data into DCT coefficients, making it more robust against compression but also more computationally demanding. Similarly, the Discrete Wavelet Transform (DWT) offers multi-resolution embedding, enhancing imperceptibility at the cost of increased computational complexity. Another notable technique involves embedding data into motion vectors in compressed video streams, such as those used in MPEG formats.

CHAPTER 2
LITERATURE SURVEY

1. **Paper Name:** Secure Video Steganography Technique using DWT and H.264

   Author: RENUKA B, Dr. N MANJA NAIK

   Abstract: Sharing of mixed media data has turned out to be brisk because of headway in data innovation. In any case, data security breaking has expanded by the innovation progression. The concealing limit and power against assaults are three principle prerequisites used in video steganography technique should think about. In this paper, a secure video steganography calculation using discrete wavelet transform (DWT) space dependent on the motion object detection calculation and H.264 is proposed utilizing Mat lab programming. The mystery message is pre-prepared by applying bit moving and H.264 utilized for encoding the mystery information. To begin with, motion object detection calculation is actualized on host recordings to recognize the locales of enthusiasm for the moving articles. At that point, the information concealing procedure is tasked by inserting mystery message image into the discrete wavelet transform planes of all movement areas in the video relying upon foundation subtraction. Our test result improves the installing limit as well as upgrades its security against different assaults.

2. **Paper Name:** An Improved Video Steganography: Using Random Key-Dependent

   Author: Mohammad A. Alia, Khulood Abu Maria

   Abstract: Steganography is defined as the art of hiding secret data in a non-secret digital carrier called cover media. Trading delicate data without assurance against intruders that may intrude on this data is a lethal. In this manner, transmitting delicate information and privileged insights must not rely on upon just the current communications channels insurance advancements. Likewise should make more strides towards information insurance. This article proposes an improved approach for video steganography. The improvement made by searching for exact matching between the secret text and the video frames RGB channels and Random Key - Dependent Data, achieving steganography performance criteria, invisibility, payload/ capacity and robustness.

3. **Paper Name:** Single level Discrete Wavelet Transform based Video Steganography on Horizontal and Vertical coefficients APPLICATIONS.

   Author: Meenu Suresh1, Dr. I. Shatheesh Sam

   Abstract: This paper proposes a single-level discrete wavelet transform based novel video steganography algorithm. Initially ‘number of carrier frames are chosen to hide the data in this method. After estimating carrier frames, every frame is separated into each R, G and B components which are decomposed using single-level discrete wavelet transform (DWT). For embedding the watermark information the horizontal and vertical coefficients are selected as a small change in these coefficients has negligible effect on the quality of the video frame. The watermark image pixels is shuffled before embedding for which a key is required. The shuffled pixels are grouped in three data matrices. Each data matrix is embedded in the horizontal and

vertical coefficients of RGB frames

. The original diagonal coefficients, embedded coefficients and original approximation coefficients are used to reconstruct the stegano video after embedding.
During the extraction process the watermark is extracted from the horizontal and vertical coefficients of stego-video. PSNR value of 57 dB with an embedding capacity of 60 percent is achieved in the method proposed. The performance of the method proposed in view of video quality as well as embedding capacity outperforms other methods and is justified by the experimental results.

4. **Paper Name:** Video Steganography by Neural Networks Using Hash Function.

Author: GK.Jayasakthi velmurugan, S.Hemavathi

Abstract: Video Steganography is an extension of image steganography where any kind of file in any extension is hidden into a digital video. The video content is dynamic in nature and this makes the detection of hidden data difficult than other steganography techniques. The main motive of using video steganography is that the videos can store large amount of data in it. This paper focuses on security using the combination of hybrid neural networks and hash function for determining the best bits in the cover video to embed the secret data. For the embedding process, the cover video and the data to be hidden is uploaded. Then the hash algorithm and neural networks are applied to form the stegano video. For the extraction process, the reverse process is applied and the secret data is obtained. All experiments are done using MatLab2016a software.

5. **Paper Name:** Data Encryption Decryption Using Steganography

Author: Manohar N1, Peetla Vijay Kumar

Abstract: Video steganography is a method that processes secure communication. When we see the history of steganography, it was hidden in many ways such as tablets covered with wax, written on the stomachs of rabbits. Here in this paper, considering the video steganography methods to perform secure steganography communication. Many methods have been proposed for video steganography but they're no more different types of formats, secured, quality, of the results. So here propose secure steganography methods i.e. secure base LSB method, Neural Networks Fuzzy logic, and check their using PSNR and MSE data of the methods. That data-set has collected is from video streams. And the result was seen with the more formats, more security quality of outputs, accuracy values of PSNR MSE which is better than other proposed methods.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 ASSUMPTIONS AND DEPENDENCIES

The video steganography method based on transform block decision operates under several key assumptions and dependencies. Firstly, the method is designed to work with H.264/AVC compressed video streams, as H.264 is one of the most widely used video codecs, particularly in real-time applications like video conferencing and streaming. The method leverages the motion estimation and transform block decision processes during video compression to embed the secret data, without significantly impacting video quality. Another important assumption is the use of a randomized pixel selection approach, employing the knight tour algorithm, rather than a serial pixel selection. This enhances the security and robustness of the method against steganalysis techniques.

Additionally, the method utilizes a key function encryption to further secure the data embedding process, making it more difficult for unauthorized parties to detect or extract the hidden information. The method operates on a per macro-block basis during the motion estimation sub-pixel refinement stage, enabling real-time performance without the need to wait for an entire frame or group of pictures. Crucially, the method preserves the local optimality, coherency, and consistency of the motion vectors, which are crucial for maintaining video quality and evading steganalysis techniques that target these properties. Finally, the video steganography method based on transform block decision aims to achieve higher embedding rates compared to other state-of-the-art motion vector-based steganography techniques, while still satisfying real-time constraints. This combination of assumptions and dependencies allows the method to provide a secure, high-capacity, and real-time data embedding solution for video communication.

3.2 FUNCTIONAL REQUIREMENTS

Video steganography involves hiding secret data within a video file in such a way that the existence of the hidden data is concealed. Here are the functional requirements for a video steganography system:

1. **Data Embedding:** The system should provide functionality to embed the secret data into the video file. This process typically involves modifying the video frames or adding extra data that does not significantly alter the visual or auditory perception of the video.

2. **Data Extraction:** There should be a mechanism to extract the hidden data from the steganographic video file. This process must accurately retrieve the original secret data without any loss or corruption
3. **Security:** The system should ensure the security and integrity of the hidden data. It should employ robust encryption techniques to prevent unauthorized access or detection of the secret information.

4. **Capacity:** The system should support embedding a sufficient amount of secret data into the video file without significantly degrading the quality of the video. The capacity of the steganographic channel should be high enough to accommodate various types of data, such as text, images, or files.

5. **Imperceptibility:** The modifications made to the video file during the embedding process should be imperceptible to human observers. The steganographic alterations should not introduce noticeable distortions or artifacts that could raise suspicion.

1) **Robustness:** The hidden data should remain intact and recoverable even in the presence of common video processing operations, such as compression, resizing, or format conversion. The steganographic technique should be resilient to unintentional distortions or attacks aimed at destroying or revealing the hidden information.

2) **Compatibility:** The steganography system should be compatible with various video file formats and codecs to ensure broad applicability across different platforms and devices.

3) **Efficiency:** The embedding and extraction processes should be efficient and computationally lightweight, allowing for real-time or near-real-time performance, especially for applications requiring quick processing, such as video streaming or live communication.

4) **Authentication:** The system may include mechanisms for verifying the authenticity and integrity of the steganographic video file to ensure that it has not been tampered with or modified by unauthorized parties.

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 User Interface:

- **Graphical User Interface (GUI):** The system should provide a user-friendly GUI for users to interact with the steganography functionalities.
- **Menu System:** The GUI should have menus or options for embedding, extracting, and managing steganographic data.
- **Progress Indicators**: The interface should display progress indicators during embedding and extraction processes to inform users about the status of their operations.
- **Error Handling:** Clear error messages should be provided in case of incorrect inputs, failures during embedding or extraction, or other issues.

3.3.2 Input and Output:

- **Video Input:** The system should accept various video file formats as input for embedding secret data.
- **Data Input:** Users should be able to input the secret data to be embedded, which may include text, files, or other multimedia content.
- **Output:** The system should produce steganographic video files containing the hidden data, as well as provide extracted secret data as output.

3.3.3 Integration:

- **File System Integration:** The system should allow users to navigate their file systems to select input video files and save steganographic output files.
- **API Integration:** For integration with other systems, the steganography functionalities may be exposed through an application programming interface (API).

3.3.4 Security:

- **Authentication:** If the system includes user accounts, there should be a login interface with appropriate authentication mechanisms.
- **Encryption Keys:** Users should be able to input encryption keys or select encryption algorithms through the interface to enhance the security of the hidden data.

3.3.5 Compatibility:

- **Video Codec Compatibility**: The system should support a variety of video codecs for input and output to ensure compatibility with different video formats.
- **Operating System Compatibility:** The interface should be compatible with popular operating systems such as Windows, macOS, and Linux.

3.3.6 Hardware Requirements:

- **Minimum System Requirements:** The interface should specify the minimum hardware requirements for running the steganography software, including CPU, RAM, and storage space.

3.3.7 Documentation:

- **User Manual:** The system should provide a user manual or help documentation accessible from the interface to guide users on how to use the steganography functionalities effectively.

3.4 NON-FUNCTIONAL REQUIREMENTS

3.4.1 Performance Requirements:

The performance of the functions and every module must be well. Overall performance of the software will enable the users to work efficiently. Performance of encryption of data should be fast. Performance of the providing virtual environment should be fast Safety Requirement.

3.2.2 Safety Requirement:

The application is designed in modules where errors can be detected and fixed easily. This makes it easier to install and update new functionality if required.

3.2.3 Security Requirements:

Users are authenticated using many security phases so reliable security is provided.

3.2.4 Software Quality Attributes:

Our software has many quality attribute that are given below:-

- **Adaptability:** This software is adaptable by all users.
- **Availability:** This software is freely available to all users. The availability of the software is easy for everyone.
- **Maintainability:** After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
- **Reliability:** The performance of the software is better which will increase the reliability of the Software. User Friendliness: Since, the software is a GUI application; the output generated is much user friendly in its behavior.
- **Integrity:** Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
- **Security:** Users are authenticated using many security phases so reliable security is provided. Testability: The software will be tested considering all the aspects.

3.5 SYSTEM REQUIREMENTS

3.5.1 Hardware Requirements:

- Processor: Intel core 5
- Ram size: 8GB
- Hard disk capacity : 500 GB
- Monitor type: 15 Inch shading screen
- Keyboard type: web console

3.5.2 Software Requirements:

- IDE: spyder
- Language: Python
- Documentation: Ms-Office

3.6 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

The Software Development Life Cycle (SDLC) is a framework that outlines the process of developing software from inception to retirement. It consists of several phases, including planning, requirements analysis, design, implementation, testing, deployment, and maintenance. When it comes to video steganography, which is the practice of hiding secret information within a video file, the use of SDLC can be beneficial in several ways:

- **Planning Phase:** In this phase, the objectives of the video steganography project are defined. This includes determining the specific requirements of the steganographic system, such as the types of videos it will support, the level of security required, and the methods of embedding and extracting hidden data.

- **Requirements Analysis:** During this phase, detailed requirements for the video steganography system are gathered from stakeholders. This involves understanding the desired functionality of the system, as well as any constraints or limitations that need to be considered.

- **Design Phase:** In the design phase, the architecture of the video steganography system is developed. This includes designing the algorithms and techniques for embedding and extracting hidden data from video files, as well as designing the user interface and any other components of the system.

- **Implementation Phase:** Once the design is complete, the actual development of the video steganography system begins. This involves writing the code for the embedding and extraction algorithms, as well as any other software components needed to support the system.

- **Testing Phase:** In this phase, the video steganography system is thoroughly tested to ensure that it meets the specified requirements and functions correctly. This includes testing the embedding and extraction algorithms to ensure that hidden data can be successfully embedded into and extracted from video files without causing any noticeable degradation in quality.

- **Deployment Phase:** Once testing is complete and any necessary revisions have been made, the video steganography system is deployed for use. This may involve installing the software on users' computers or integrating it into existing video processing systems.

- **Maintenance Phase:** After deployment, the video steganography system may require ongoing maintenance to address any issues that arise, implement updates or enhancements, and ensure that it continues to meet the needs of its users

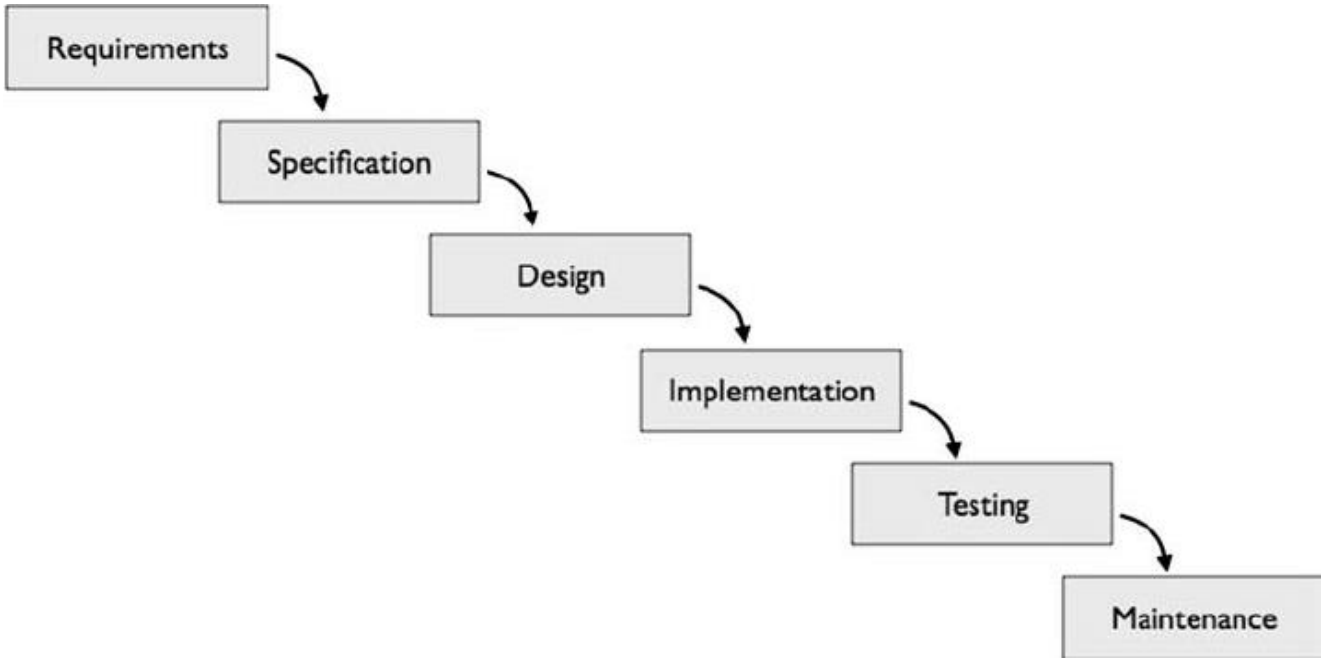


Figure 3.1: Waterfall Model

In developing a video steganography system, applying the Software Development Life Cycle (SDLC) model ensures a structured and systematic approach. Given the specific requirements and security concerns inherent in steganography, a well-chosen SDLC model can help manage the complexity and ensure the successful delivery of a robust solution

3.7 SYSTEM IMPLEMENTATION PLAN

The System Implementation plan table, shows the overall schedule of tasks compilation and time duration required for each task.

Sr. No.	Name/Title	Start Date	End Date
1	Preliminary Survey	17/08/23	25/08/23
2	Introduction and Problem Statement	25/08/23	27/08/23
3	Literature Survey	27/08/23	01/09/23
4	Problem Statement	01/09/23	10/09/23
5	Software Requirement and Specification	10/10/23	27/10/23
6	System Design	27/10/23	10/11/23
7	Partial Report Submission		
8	Architecture Design		
9	Implementation		
10	Deployment		
11	Testing		
12	Paper Publish		
13	Report Submission		

CHAPTER 4
SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The system architecture of the video steganography method based on transform block decision is designed to operate within the framework of H.264/AVC compressed video streams. This architecture utilizes the motion estimation and transform block decision processes inherent in video compression to embed secret data effectively.

By implementing a randomized pixel selection strategy using the knight tour algorithm, the method enhances security and resilience against steganalysis techniques. Additionally, a key function encryption mechanism is integrated to further fortify the data embedding process, increasing the difficulty for unauthorized entities to detect or extract hidden information. Operating on a per macro-block basis during the motion estimation sub-pixel refinement stage enables real-time performance without the need to process entire frames or groups of pictures.

The system architecture also prioritizes the preservation of local optimality, coherency, and consistency of motion vectors to maintain video quality and evade steganalysis attempts targeting these characteristics. Ultimately, the architecture aims to achieve superior embedding rates compared to existing motion vector-based steganography methods while meeting real-time constraints, offering a robust and efficient solution for secure data embedding in video communication.

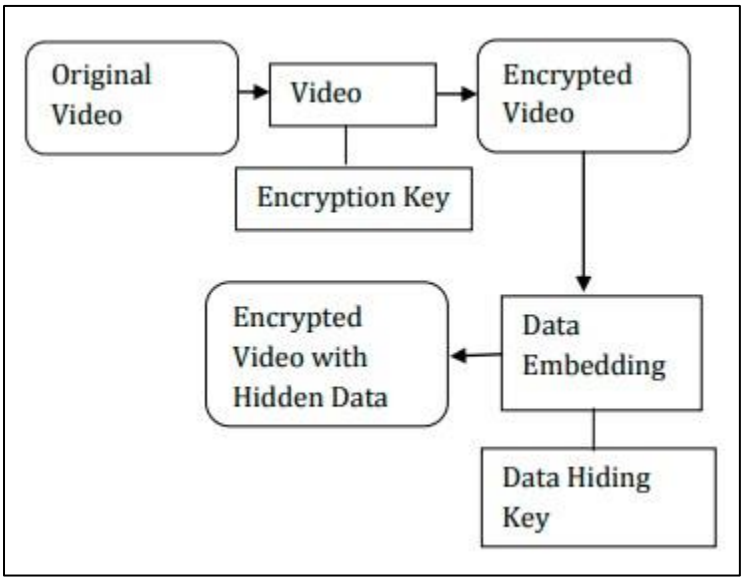


Figure 4.1: System Architecture

4.2 DATA FLOW DIAGRAMS

In Data Flow Diagram, we Show that flow of data in our system in DFD0 we show that base DFD in which rectangle present input as well as output and circle show our system, In DFD1 we show actual input and actual output of system input of our system is text or image and output is rumor detected likewise in DFD 2 we present operation of user as well as admin.

Data Flow Diagram Level 0:

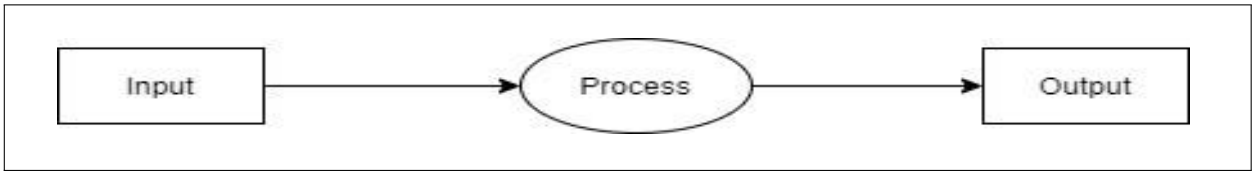


Figure 4.2: Data Flow Diagram Level 0

Data Flow Diagram Level 1:

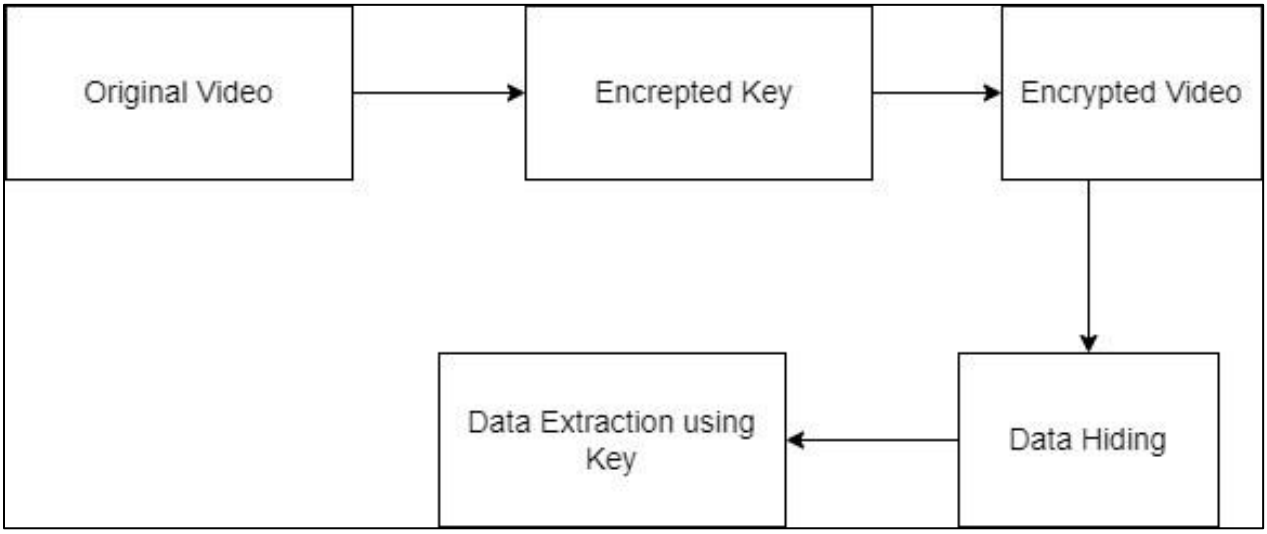


Figure 4.3: Data Flow Diagram Level 1

Data Flow Diagram Level 2:

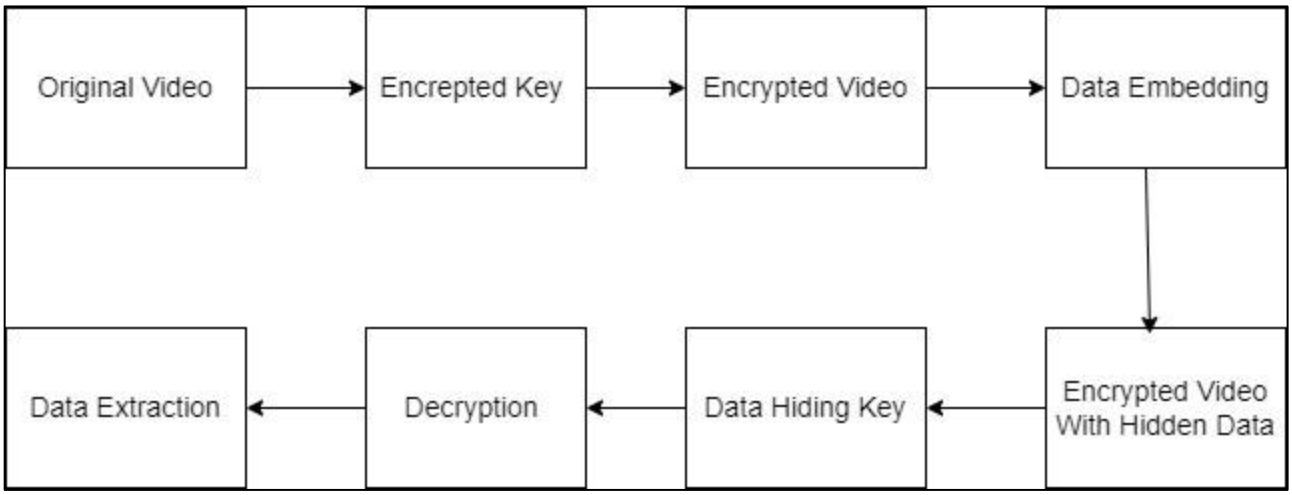


Figure 4.4: Data Flow Diagram Level 2

4.3 ENTITY RELATIONSHIP DIAGRAM

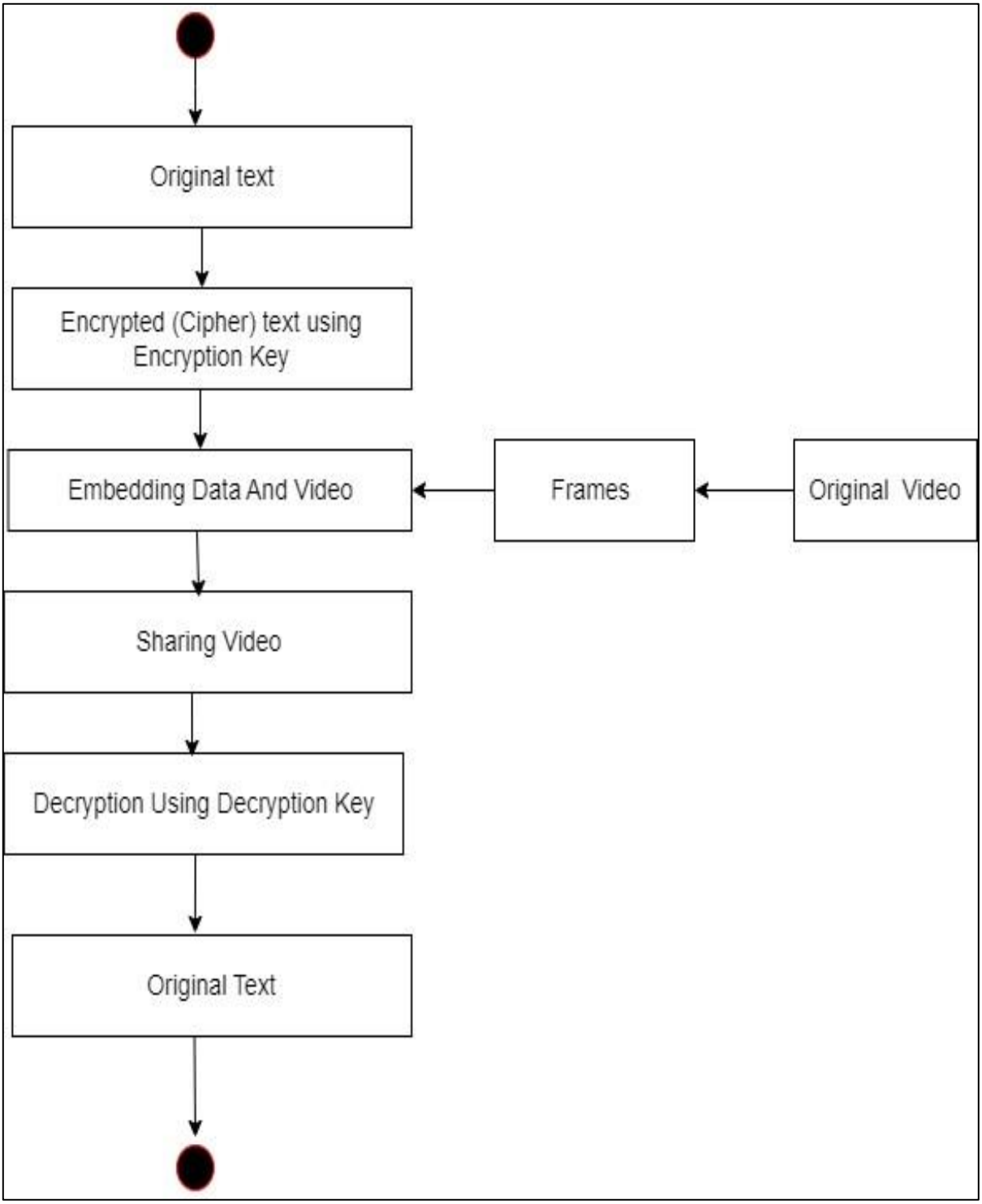


Figure 4.5: Entity Relationship Diagram

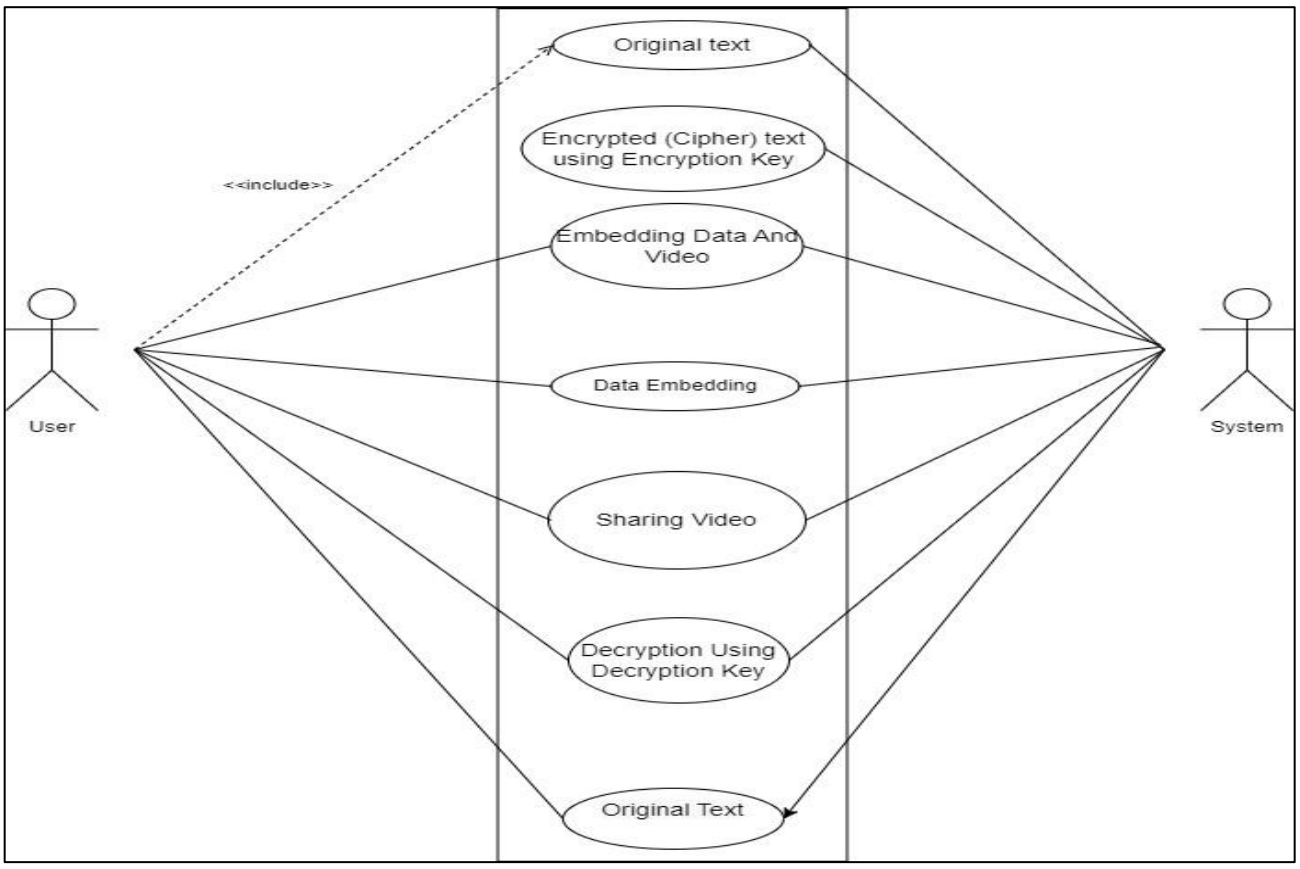
4.4 UML DIAGRAMS

Unified Modeling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of a software intensive system. UML is process independent, although optimally it should be used in process that is use case driven, architecture-centric, iterative, and incremental. The Number of UML Diagram is available.

- Use case Diagram.
- Activity Diagram.
- Sequence Diagram.
- Class Diagram.

4.4.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.



Figures 4.6: Use Case diagram

4.4.2 Activity Diagram

Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control they can also include elements showing the flow of data between activities through one or more data stores.

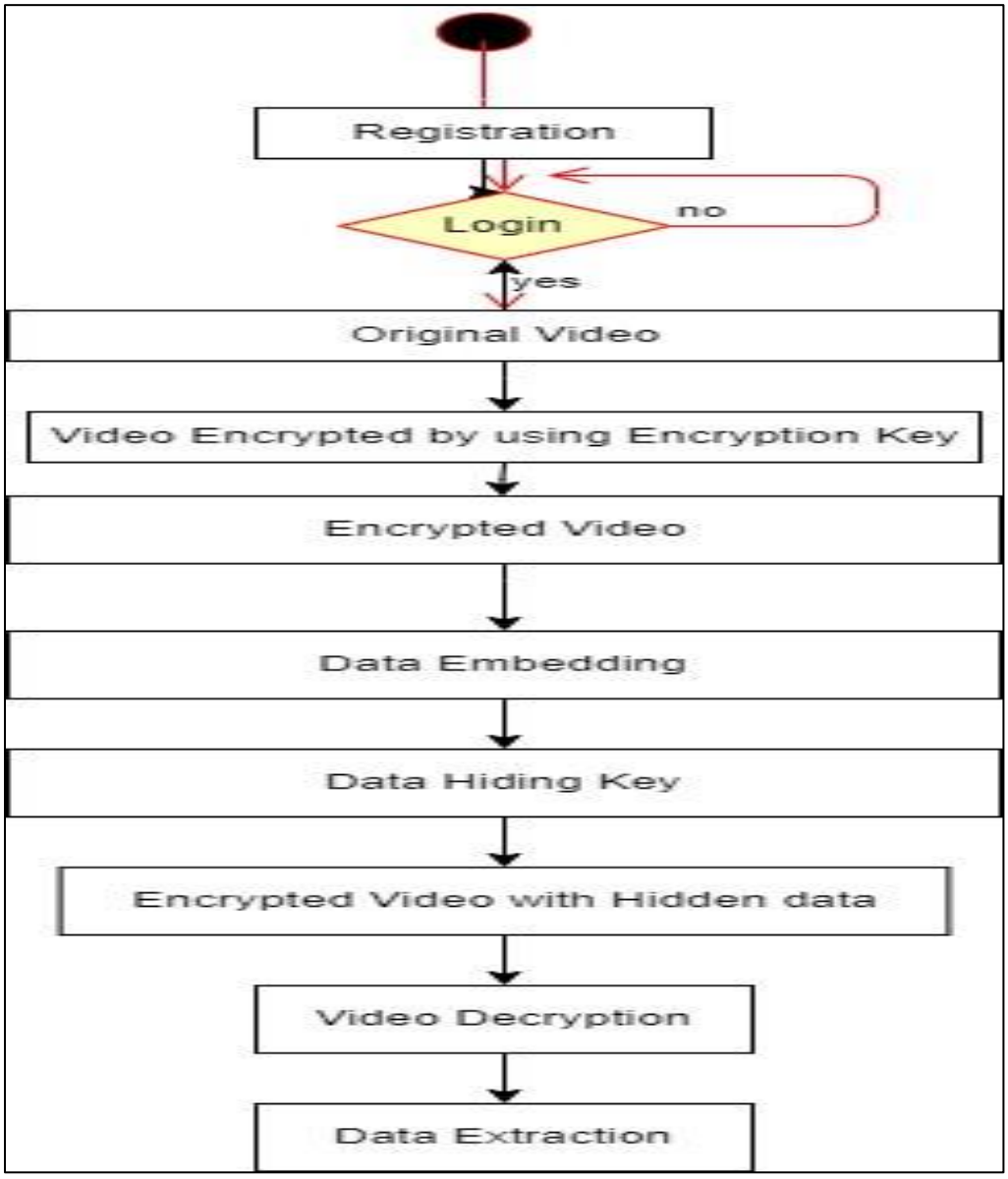


Figure 4.8: Activity Diagram

4.4.3 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged be- tween the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

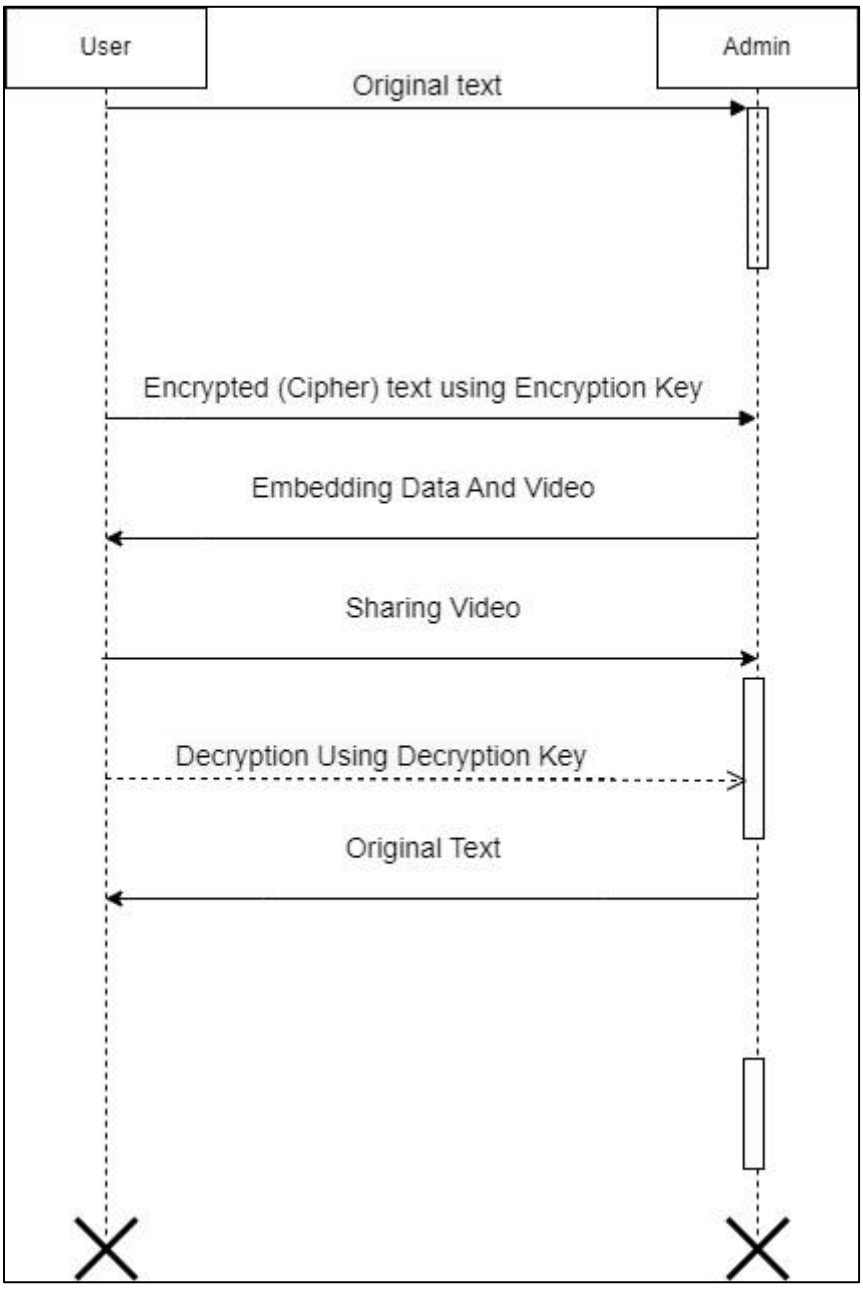


Figure 4.9: Sequence diagram

4.4.1 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relation- ships among objects.

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.

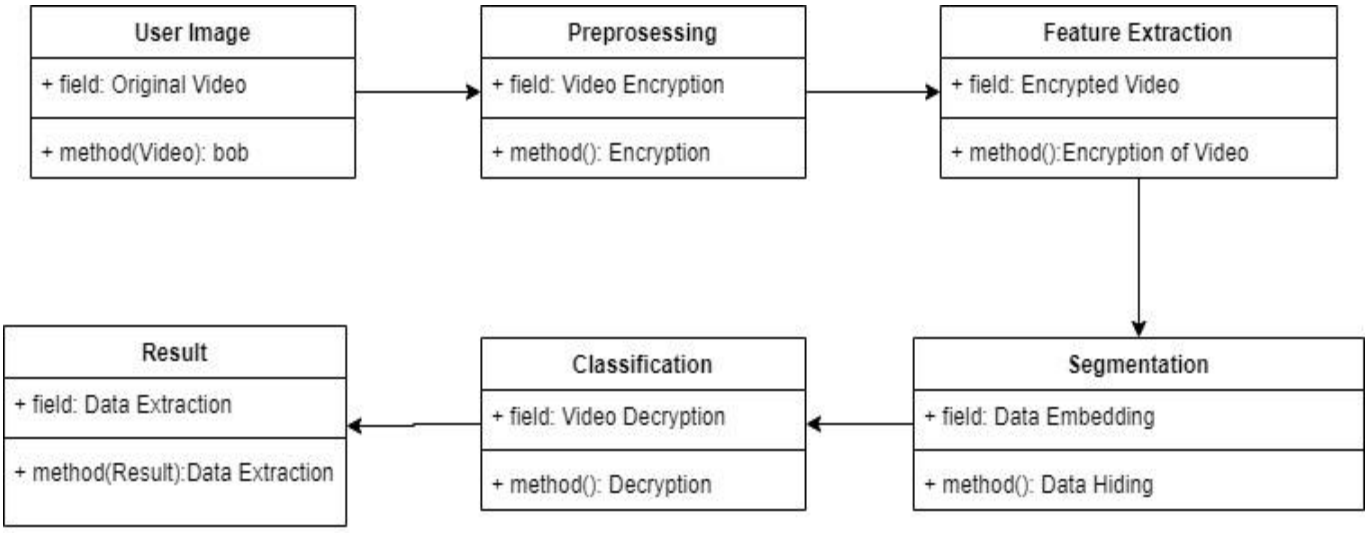


Figure 4.10: Class Diagram

CHAPTER 5
PROJECT PLAN

In developing a video steganography system, applying the Software Development Life Cycle (SDLC) model ensures a structured and systematic approach. Given the specific requirements and security concerns inherent in steganography, a well-chosen SDLC model can help manage the complexity and ensure the successful delivery of a robust solution.

5.1 STAKEHOLDER LIST

SR.NO	STAKEHOLDER	HOSPITALS
1.	Project Type	Hospital
2.	Customer	Patient
3.	User	Any one person

5.2 RISK MANAGEMENT

5.2.1 Identifying Risks

- Detection Risk: The risk that the hidden information could be detected by unauthorized parties using steganalysis tools.
- Data Integrity Risk: The risk that the hidden data could be corrupted or altered unintentionally during transmission or processing.
- Performance Risk: The risk that embedding data into video files could degrade the performance or quality of the video, making it suspicious or unusable.
- Security Risk: The risk that steganography methods could be compromised, leading to unauthorized access or leakage of hidden information.
- Legal and Ethical Risks: The risk that using steganography could violate legal statutes or ethical guidelines, especially if used for malicious purposes.

5.2.2 Assessing Risks

- Likelihood: Determine the probability of each risk occurring based on the chosen steganographic method, the environment in which it will be used, and the capabilities of potential adversaries.
- Impact: Evaluate the potential consequences of each risk, considering factors such as the sensitivity of the hidden information, the importance of video quality, and regulatory implications.
- Vulnerability Analysis: Identify specific vulnerabilities in the stenographic process, such as weaknesses in the algorithm used, susceptibility to certain types of steganalysis, or points of exposure during video transmission or storage.

5.2.3 Mitigating Risks

- Choosing Robust Techniques: Use advanced and well-researched steganographic methods that are less prone to detection and degradation, such as those employing sophisticated embedding algorithms and error-correction mechanisms.
- Encryption: Encrypt the hidden information before embedding it into the video to provide an additional layer of security in case the steganographic cover is compromised..
- Quality Control: Implement measures to maintain video quality and prevent noticeable degradation that could arouse suspicion. Techniques include adaptive embedding that balances data capacity and video quality.

- Regular Audits: Conduct regular audits and testing to detect potential vulnerabilities and to ensure that the steganographic methods remain secure against evolving threats.
- Legal Compliance: Ensure that the use of steganography complies with all relevant laws and regulations, and that ethical guidelines are followed to avoid misuse.

5.2.4 Monitoring and Response

- Continuous Monitoring: Implement monitoring systems to detect any attempts to analyze or tamper with steganographic content.
- Incident Response Plan: Develop a response plan to address incidents where hidden information is detected, exposed, or otherwise compromised. This should include steps for containment, investigation, and remediation.
- Training and Awareness: Provide training for stakeholders on the risks associated with steganography and the measures in place to mitigate them.

1	**RISK ID**	**1**
2	**RISK DESCRIPTION**	**DESCRIPTION 1**
3	**CATEGORY**	**DEVELOPMENT ENVIRONMENT**
4	**SOURCE**	**SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT**
5	**PROBABILITY**	**LOW**
6	**IMPACT**	**HIGH**
7	**RESPONSE**	**MITIGRATE**
8	**STRATEGY**	**STRATEGY**
9	**RISK STATUS**	**OCCURED**

.

1	**RISK ID**	**2**
2	**RISK DESCRIPTION**	**DESCRIPTION 2**
3	**CATEGORY**	**REQUIREMENTS**
4	**SOURCE**	**SOFTWARE DESIGN SPECIFICATION DOCUMENT REVIEW**
5	**PROBABILITY**	**LOW**
6	**IMPACT**	**HIGH**
7	**RESPONSE**	**MITIGRATE**
8	**STRATEGY**	**BETTER TESTING WILL RESOLVE THIS ISSUE**
9	**RISK STATUS**	**IDENTIFIED**

5.2.1 Project Task Set

Major tasks in the project stages are:

- Task 1: Correctness
- Task 2: Availability
- Task 3: Integrity

5.3.2 Task Network

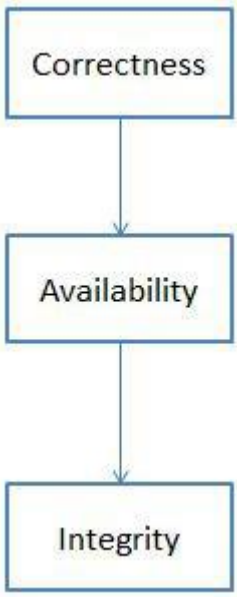


Figure 6.1 Task Network

5.2.2 TimeLine Chart

NO.	TASK	DURATION (DAYS)
1.	Group Formation	4
2.	Decide Area Of Interest	4
3.	Search Topic	5
4.	Topic Selection	5
5.	Sanction Topic	5
6.	Search Related Information	12
7.	Understanding Concept	7
8.	Search Essential Documents (IEEE & white paper, software)	6
9.	Problem definition	2
10.	Literature Survey	5
11.	SRS	14
12.	Project Planning	2

- **TEAM ORGANISATION**

5.4.1 Team Structure

The team structure for the project is identified. Roles are defined. Our team have three members. We select this topic after discussing with each other. All the members performing all the task whatever tasks are assign to the members.

5.4.2 Management reporting and communication

For developing this project, first finalize the project topic after reviewing the multiple project topics. After that we gather the requirements about this project. Then we make the synopsis, SRS, PPT and report for sem1. For all above requirements, our team member and our guide discuss with each other. Every time we maintain all the details about whatever activities are performed by us.

CHAPTER 6
PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

1. **Preprocessing Module:** Acquires and prepares the video for embedding, including extracting frames and converting formats if needed.

2. **Embedding Module:** Prepares the data to be hidden through compression and encryption, then embeds it into the video frames using various steganographic algorithms.

3. **Post-Processing Module**: Reconstructs the video, ensures quality, and applies compression if necessary.

4. **Transmission Module:** Ensures secure transmission of the steganographic video over networks.

5. **Extraction Module:** Receives the video, extracts frames, and retrieves the hidden data.

6. **Post-Extraction Module:** Decrypts, decompresses, and validates the hidden data.

7. **Analysis and Security Module:** Tests for steganalysis resistance, conducts security audits, and evaluates performance.

8. **User Interface Module:** Provides interfaces for embedding and extracting data and tools for monitoring.

9. **Documentation and Reporting Module:** Produces user and technical documentation and generates performance and security reports.

10. **Compliance and Legal Module:** Ensures adherence to legal and ethical standard

6.2 TOOLS AND TECHNOLOGIES USED

6.2.1 Libraries

1. **Pandas:** Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library.

2. **NumPy:** NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

3. **Import cv2:** All packages contain Haar cascade files. Cv2.data.haarcascades can be used as a shortcut to the data folder.

4. **Pillow:** Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors.

5.

6.2.2 Pandas

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant white space. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python was created in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release."[30] No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported. Python interpreters are available for many operating systems. A global community of programmers develops and maintains Python, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, man- ages and directs resources for Python and Python development.

6.2.3 Anaconda

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and MacOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only anaconda, Python, the packages they depend on, and a small number of other packages. Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the anaconda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI). The big difference between anaconda and the pip

package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

When pip installs a package, it automatically installs any dependent Python pack- ages without checking if these conflict with previously installed packages [citation needed]. It will install a package and any of its dependencies regardless of the state of the existing installation [citation needed]. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensor flow. In some cases, the package may appear to work but produce different results in detail

6.2.4 Spyder

Spyder is a powerful scientific environment written in Python, for Python, and de- signed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package. Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console, in your own software.

Features:-

- **Editor:**
  Work efficiently in a multi-language editor with a function/class browser, real- time code analysis tools (pyflakes, pylint, and pycodestyle), automatic code completion (jedi and rope), horizontal/vertical splitting, and go-to-definition.

- **Interactive console:**
  Harness the power of as many IPython consoles as you like with full workspace and debugging support, all within the flexibility of a full GUI interface. Instantly run your code by line, cell, or file, and render plots right in line with the output or in interactive windows.

- **Documentation viewer:**
  Render documentation in real-time with Sphinx for any class or function, whether external or user-created, from either the Editor or a Console.

6.3 ALGORITHM DETAILS

6.3.1 Least Significant Bit (LSB) Algorithm:

In video steganography, the Least Significant Bit (LSB) algorithm is used to hide information within the video frames. The technique takes advantage of the large amount of data in video files and the minor visual impact of changing the least significant bits of the pixel values. The LSB algorithm modifies the pixel values of the frames in such a way that the hidden message is imperceptible to human viewers.

- LSB algorithm is widely used in steganography to hide information within images, audio files, and video frames by modifying the least significant bits of the pixel values.
- The process involves converting the secret message into binary format, embedding the message by modifying the LSB of the cover media, and extracting the message by reading the LSBs of each byte in the cover media.
- The capacity for hiding data depends on the size of the cover media, and imperceptibility is maintained by making minimal changes to the original media.
- Advantages include simplicity, high fidelity maintenance, and high data-hiding capacity, while vulnerabilities to steganalysis and lack of robustness against processing operations are notable disadvantages.
- Video steganography using the LSB algorithm follows a similar process, with the capacity depending on the number of frames and resolution, and imperceptibility being affected by significant alterations to the video content.

6.3.2 Advanced Encryption Standard (AES) Algorithm:

In video steganography, the Advanced Encryption Standard (AES) can be used to encrypt the message before embedding it into the video. This approach adds a layer of security, ensuring that even if the hidden data is detected, it cannot be easily read without the correct decryption key. Combining AES with video steganography leverages the strengths of both cryptography and steganography to protect sensitive information.

- AES is a symmetric key encryption algorithm used worldwide for securing sensitive data, offering different key lengths and rounds for varying levels of security.
- AES encryption involves dividing input data into fixed-size blocks, key expansion, initial and main rounds of transformations, and a final round to produce encrypted ciphertext.
- AES can be integrated into video steganography projects by encrypting the secret message into binary format and embedding it into video frames' least significant bits (LSBs).
- The combination of AES with video steganography enhances security by leveraging the strengths of both cryptography and steganography to protect sensitive information.
- The decryption process for AES involves extracting the encrypted data from the stego-video, retrieving the LSBs to obtain the encrypted message, and decrypting it to recover the original message.

CHAPTER 7
SOFTWARE TESTING

7.1 TYPES OF TESTING

To ensure the effectiveness, reliability, and security of video steganography systems, various types of testing are conducted. These tests are crucial for validating that the steganography methods used can securely hide and retrieve information without compromising the quality of the video or being easily detectable. Here are the main types of testing involved in video steganography:

- **Unit Testing:** It is the testing of individual software units of the application .it is done after the complexion of an individual unit before integration. Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

- **Regression Testing:** Regression testing is a software testing practice that ensures an application still functions as expected after any code changes, updates, or improvements. Regression testing is responsible for the overall stability and functionality of the existing features.

- **Smoke Testing:** Smoke Testing comes into the picture at the time of receiving build software from the development team. The purpose of smoke testing is to determine whether the build software is testable or not. It is done at the time of "building software." This process is also known as "Day 0". It is a time-saving process. It reduces testing time because testing is done only when the key features of the application are not working or if the key bugs are not fixed. The focus of Smoke Testing is on the workflow of the core and primary functions of the application.

- **System Testing:** System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

- **Integration Testing:** Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- **Quality Testing:** Ensure that the video quality is not significantly degraded by the embedding process. PSNR (Peak Signal-to-Noise Ratio): Measure the ratio between the maximum possible power of a signal and the power of corrupting noise. SSIM (Structural Similarity Index): Assess the similarity between the original and the steganography video.

7.2 TEST STRATERGY

Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

1. White-box testing: In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

2. Black-box testing: Black-box testing treats the software as a quot;black boxquot;, examining functionality without any knowledge of internal implementation. The testers are only aware of what the software is supposed to do, not how it does it.

3. Grey-box testing: Grey-box testing involves having knowledge of internal data structures and algorithms for purposes of designing tests, while executing those tests at the user, or black-box level. The tester is not required to have full access to the software.

7.3 TEST PLAN

To test this application we are going with proper sequencing of testing like unit, integration, validation, GUI, Low level and High level test cases, major scenarios likewise. We will go with the GUI testing first and then integration testing. After integration testing performs the high level test cases and major scenarios which can affect the working on the application. We will perform the testing on the data transmitted using the various inputs and outputs and validate the results. It also intends to cover any deviations that the project might take from the initially agreed Test Strategy in terms of scope, testing methodology, tools, etc. This test plan covers details of testing activities for this project and scope.

7.4 SOFTWARE TO BE TESTED

1. Edraw Max: It enables students, teachers and business professional store liable create and publish various kinds of diagram store present any ideas. With this application users can easily create professional- looking flow charts, organizational charts, network diagrams, business presentations, building plans, mind maps, science illustration, fashion designs, UML diagrams and much more.

2. Star UML: Star UML is a fully fledged, open source, UML modeling tool that supports the ability to create software designs, from basic concepts, through to the coded solution. The user should be aware that this tool is more complex than a simple UML diagram editing tool, in that, through the use of the model Drive Architecture (MDA) standard, the tool supports complex modeling which is realizable in code.

7.5 TEST CASES

1. **GUI Testing:**

Test Case	**Login Screen-Sign up**
Objective	**Click on sign up button then check all required mandatory fields with leaving all fields blank**
Expected Result	**All required mandatory fields should display with symbol "*". Instruction line "* field(s) are mandatory" should be displayed.**
Test Case	**Create a Password >> Test Box** **Confirm a Password >> Text Box**
Objective	**Check the validations message for password and confirm password field.**
Expected Result	**Correct validation message should be displayed accordingly or "Password and Confirm password should be same" in place of "Password mismatch"**

Figure 7.1: GUI Testing

2. **Login Test Cases:**

Test Cases ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test Case Criteria (P/F)
001	**Enter the wrong username or password click on submit button.**	**Username or Password**	**Error Comes**	**Error should come**	**P**
002	**Enter the Correct username or password click on submit button.**	**Username or Password**	**Accept**	**Accept**	**P**

Figure 7.2: Login Test Cases

3. **Registration Test Cases:**

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test Case Criteria (P/F)
001	Enter the number in username, middle name, and last name field.	Number	Error Comes	Error Should comes	P
001	Enter the character in username, middle name, and last name field.	Character	Accept	Accept	P
002	Enter the invalid email id format in email id field.	Kkgmail, com	Error Comes	Error should comes	P
002	Enter the valid email id format in email id field.	kk@gmail.com	Accept	Accept	P
003	Enter the invalid digit number in phone number field	99999	Error Comes	Error Should Comes	P
003	Enter the 10 digit number in phone number field	9999999999	Accept	Accept	P

Figure 7.3: Registration Test Cases

4. **System Test Cases:**

Test Case ID	Test Case	Test Case I/P	Actual Result	Expected Result	Test Case Criteria (P/F)
001	Store Xml File	Xml file	Xml file store	Accepted	P
002	Parse the Xml file for conversion	Parsing	File get parse	Accepted	P
003	Attribute identification	Check individual Attribute	Identify Attribute	Accepted	P
004	Weight Analysis	Check weight	Analyze weight of individual attribute	Accepted	P
005	Tree Formation	Form them-tree	Formation	Accepted	P
006	Cluster Evaluation	Check Evaluation	Should check cluster	Accepted	P
007	Algorithm Performance	Check Evaluation	Should work algorithm properly	Accepted	P
008	Query Formation	Check Query Correction	Should check query	Accepted	P

Figure 7.4: System Test Cases

CHAPTER 8
RESULTS

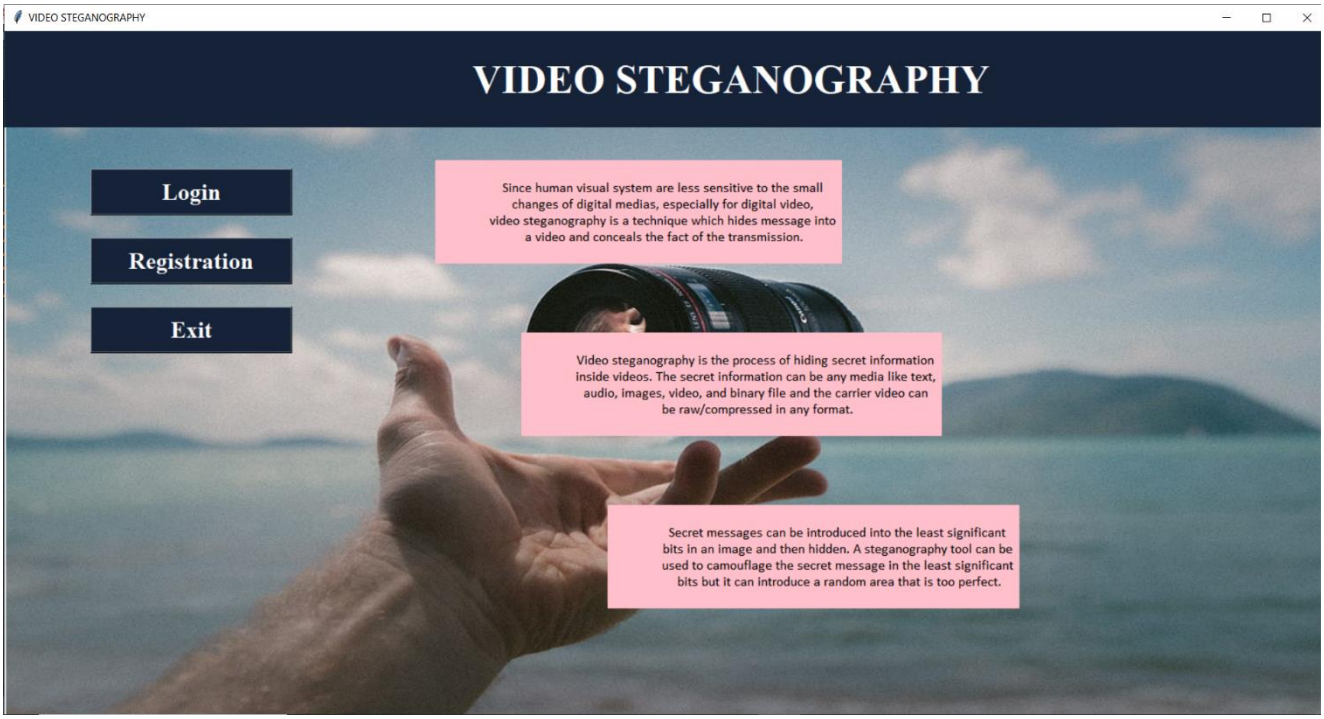


Figure 8.1: Output 1

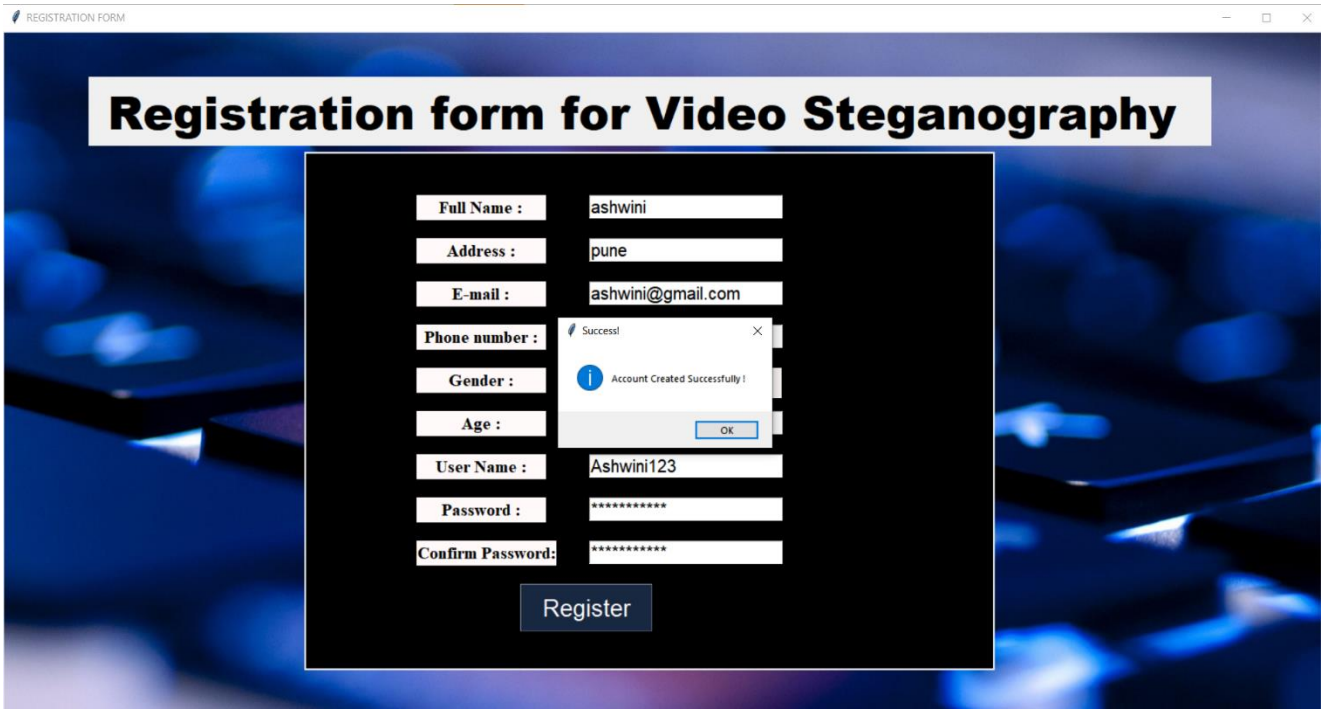


Figure 8.2: Output 2

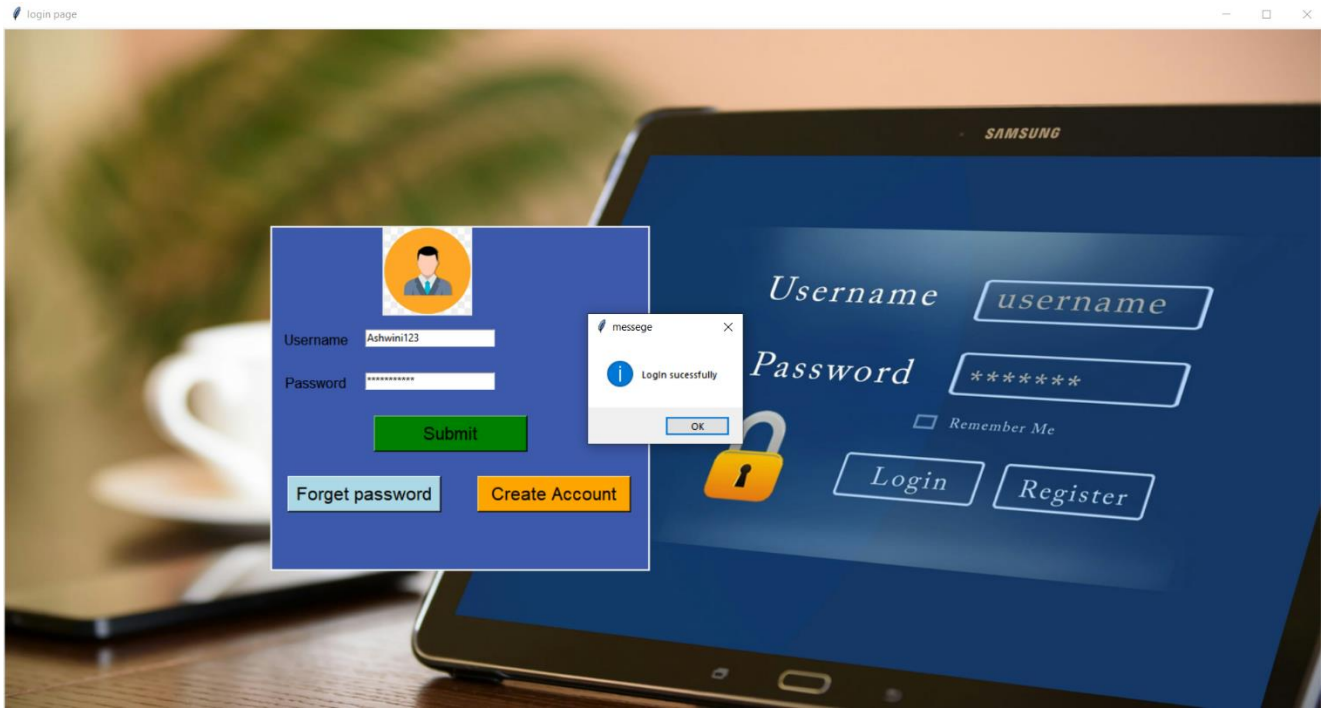


Figure 8.3: Output 3

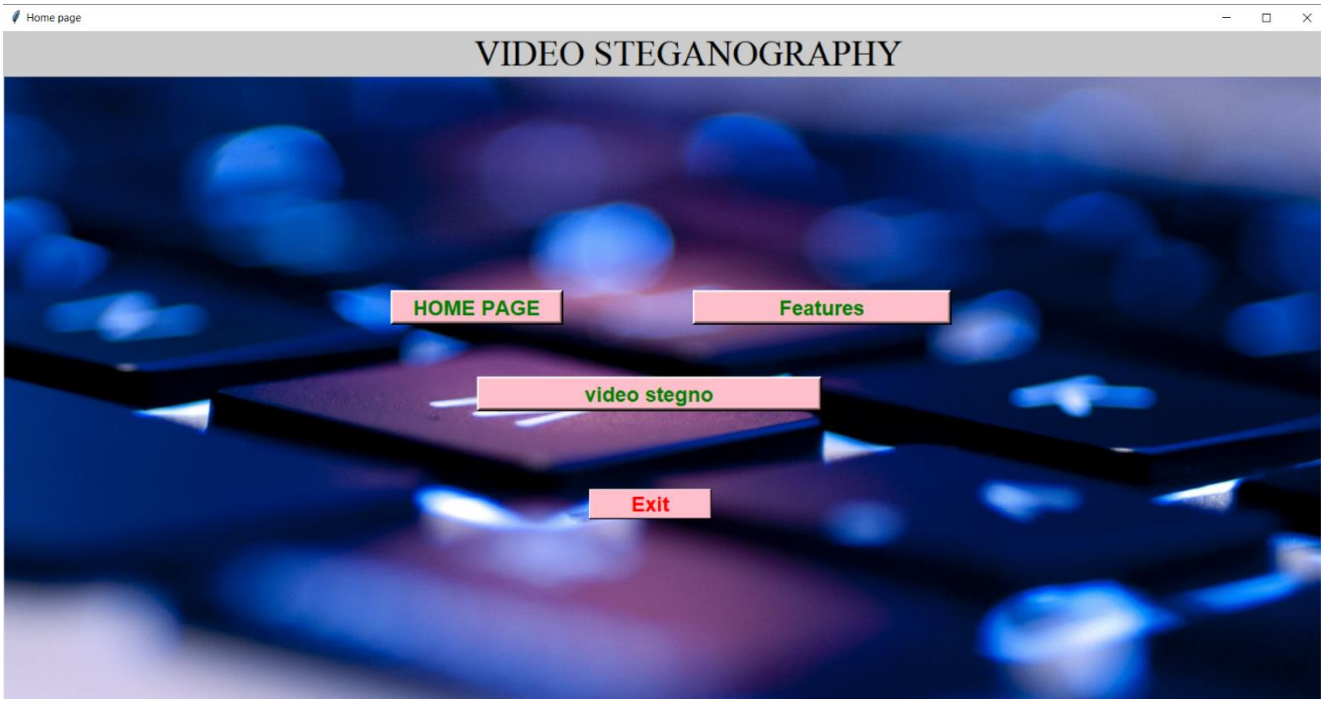


Figure 8.4: Output 4

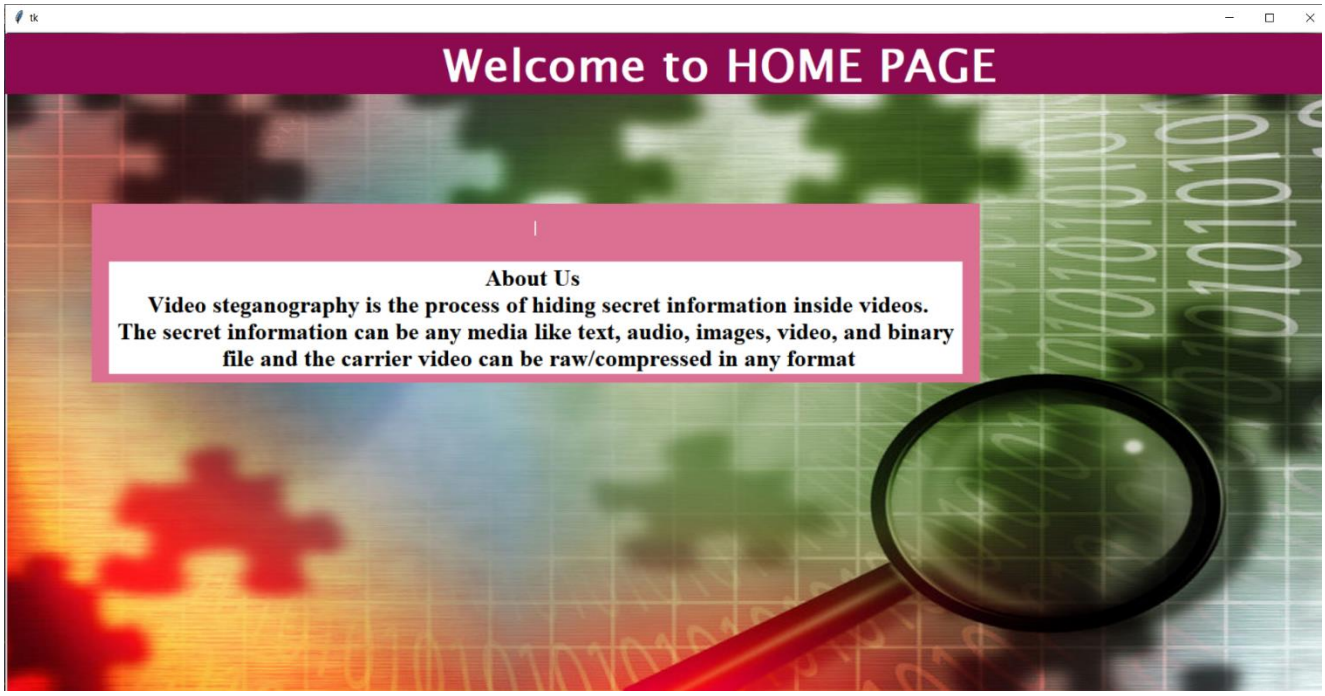


Figure 8.5: Output 5

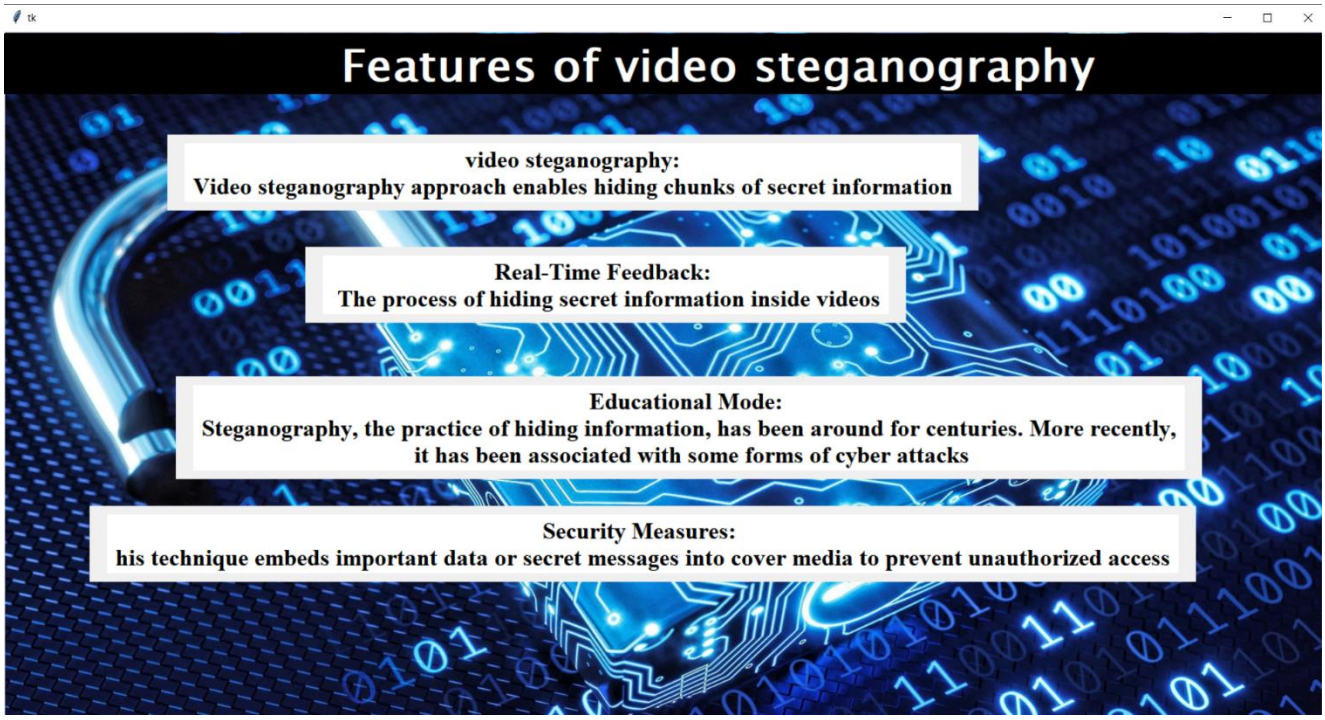


Figure 8.6: Output 6

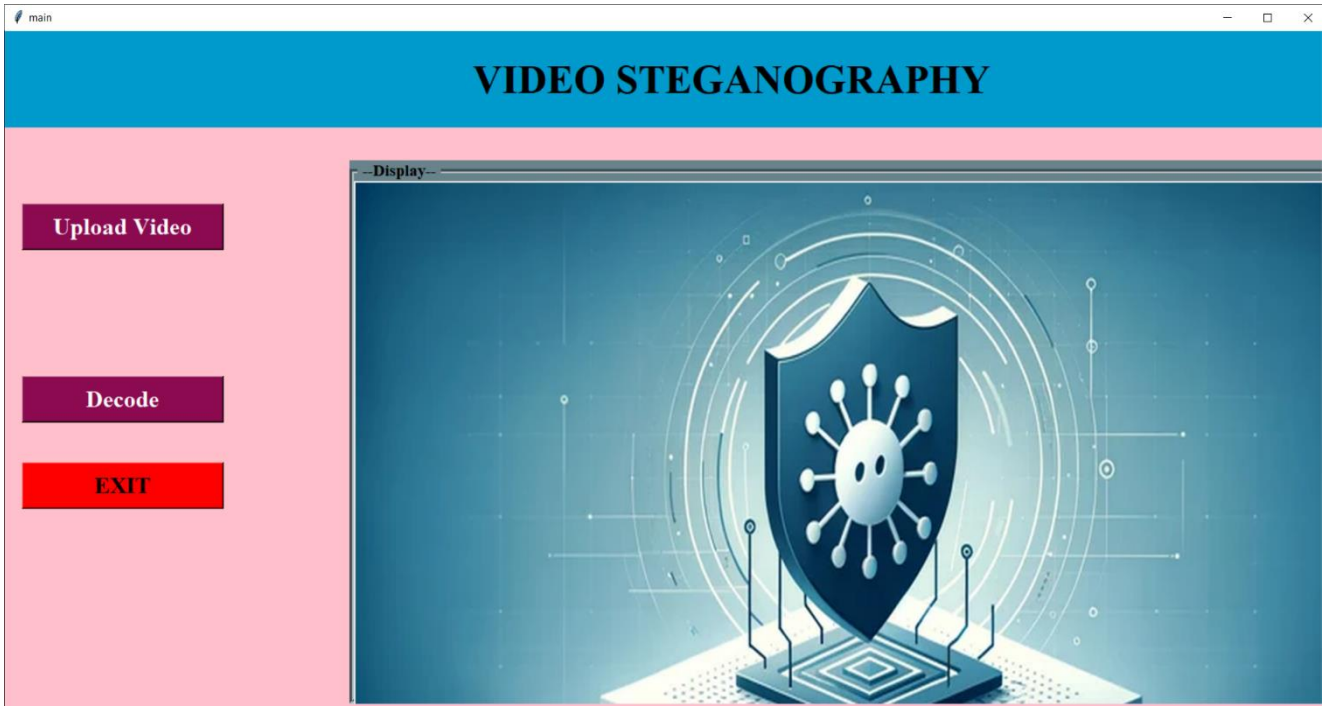


Figure 8.7: Output 7

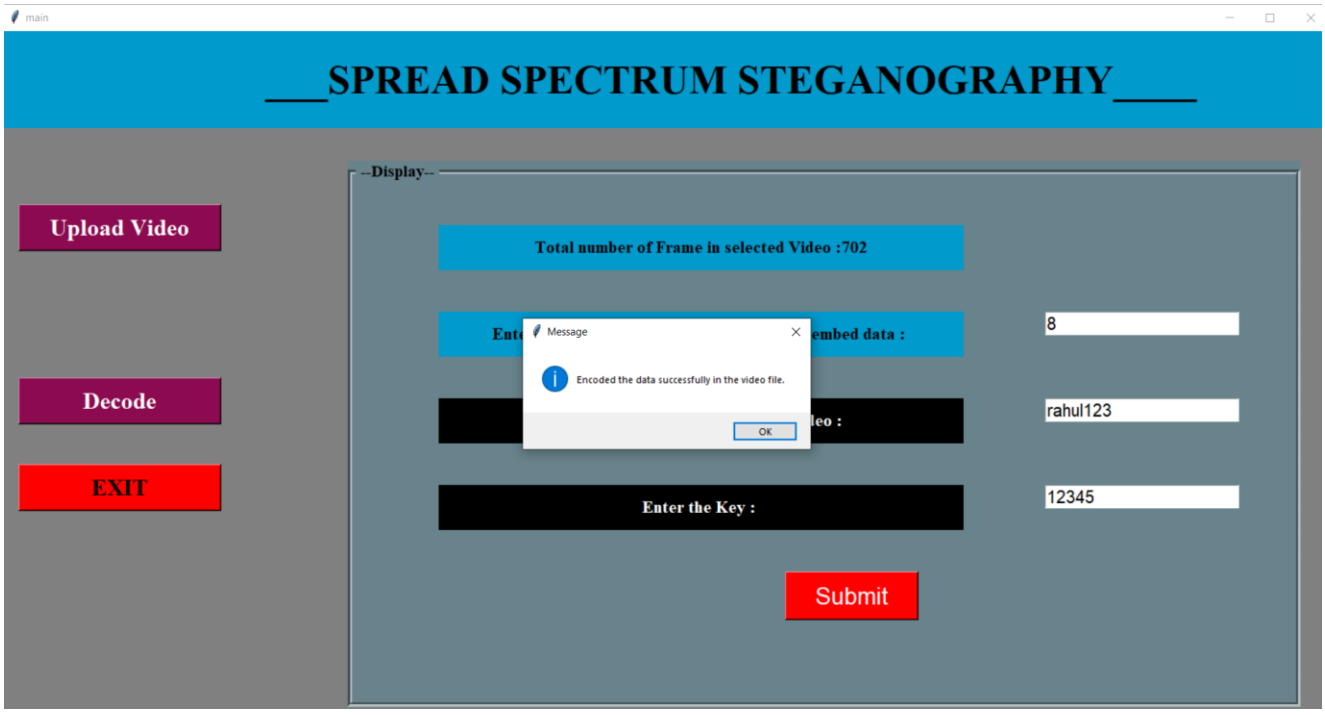


Figure 8.8: Output 8

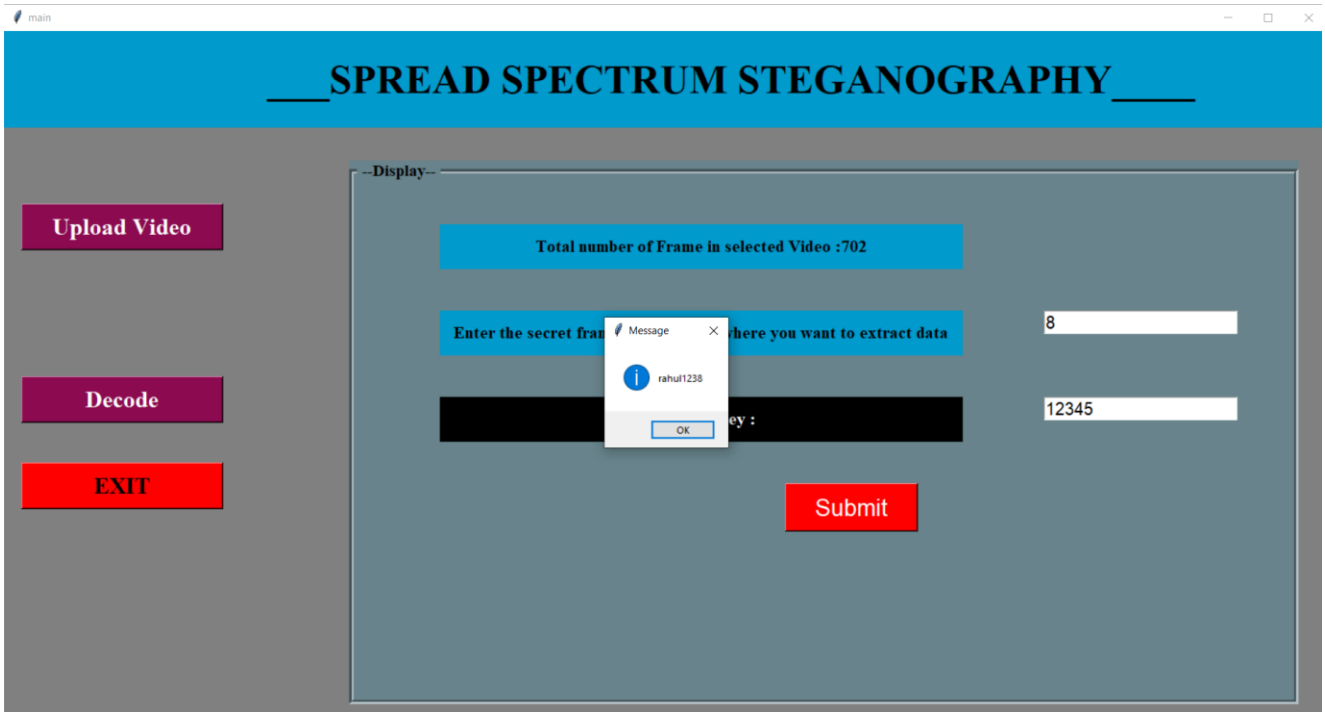


Figure 8.9: Output 9

CHAPTER 9
CONCLUSION

APPENDIX A:

A.1 Key Benefits of Video Steganography:

- **Enhanced Security**: The combination of data hiding and encryption (e.g., using AES) ensures that even if the hidden data is detected, it remains unreadable without the decryption key.
- **Confidential Communication:** Video steganography enables secure communication channels, essential for sensitive information exchange in fields like military, government, and corporate sectors.
- **Digital Watermarking:** It provides a means to embed copyright information, helping to protect intellectual property and combat digital piracy.
- **Minimal Impact on Quality**: Advanced steganographic techniques, such as LSB (Least Significant Bit) insertion, can embed data without significantly degrading video quality, making the hidden data imperceptible to the human eye.

A.2 Importance of Testing in Video Steganography:

To ensure the reliability and robustness of video steganography systems, comprehensive testing is crucial. A well-defined testing strategy includes:

- **Functionality Testing:** Ensuring that data can be accurately embedded and extracted without errors.
- **Performance Testing:** Assessing the impact of the steganographic process on video performance, including embedding and extraction times and playback quality.
- **Robustness Testing:** Evaluating the system's resilience to various attacks and manipulations, such as compression, noise addition, and frame dropping.
- **Steganalysis Testing:** Determining the detectability of the hidden data using statistical analysis and machine learning techniques.
- **Security Testing:** Ensuring the cryptographic security of the hidden data, focusing on encryption strength and key management.
- **Quality Testing:** Measuring the impact on video quality using metrics like PSNR and SSIM.
- **Usability Testing:** Evaluating the user-friendliness and overall user experience of the steganographic system.
- **Compliance Testing:** Ensuring adherence to relevant legal requirements and industry standards.

A.3 Challenges and Considerations:

- **Capacity vs. Security Trade-off:** Increasing the capacity of hidden data often comes at the expense of security, as it may make the hidden information more detectable.
- **Real-time Processing:** Implementing steganographic methods that can operate in real-time is essential for applications like video conferencing and live streaming.

A.4 Future Trends:

- **Deep Learning in Steganalysis:** Continued advancements in deep learning techniques are expected to lead to more sophisticated steganalysis methods capable of detecting hidden information with higher accuracy.
- **Privacy-Preserving Technologies:** With growing concerns about data privacy, video steganography will play a crucial role in developing privacy-preserving solutions for secure data communication and storage.
- **Block chain Integration:** Integrating steganography with block chain technology can offer tamper-proof and transparent methods for securely transmitting and verifying hidden information.
- **Quantum Steganography:** Research in quantum steganography aims to leverage the principles of quantum mechanics to develop ultra-secure methods for hiding information in quantum states.

A.5 Applications of Video Steganography:

- **Secure Communication:** Used in military and intelligence operations for covert communication and information exchange without raising suspicion.
- **Copyright Protection:** Digital watermarking techniques are employed to embed copyright information and ownership details in video content, aiding in copyright enforcement and protection against piracy.
- **Forensic Analysis:** Steganalysis techniques are applied in digital forensics to detect hidden information in videos for investigation purposes, such as identifying perpetrators and analyzing criminal activities.
- **Medical Imaging:** In the medical field, steganography is utilized for securely transmitting sensitive medical images and patient records while ensuring patient privacy and confidentiality.

A.5 Conclusion:

Video steganography represents a sophisticated approach to secure communication in the digital age, where data privacy and confidentiality are paramount. By embedding secret information within video files, steganography leverages the vast amount of data in multimedia to create an additional layer of security. This technique ensures that sensitive information can be transmitted covertly, avoiding detection by unauthorized parties.

Video steganography is a powerful tool for secure communication, offering a unique blend of data hiding and encryption. Through meticulous testing and continuous improvement, steganographic systems can provide robust security solutions that protect sensitive information and ensure data privacy in an increasingly connected world. The ongoing advancement of steganographic techniques and their integration with emerging technologies will further enhance their effectiveness and applicability in various domains.

APPENDIX B: Details of the papers referred:

1. **Paper Name:** Secure Video Steganography Technique using DWT and H.264
   Authors: RENUKA B, Dr. N MANJA NAIK
   Abstract: This paper proposes a secure video steganography algorithm utilizing discrete wavelet transform (DWT) and H.264 encoding. The method involves motion object detection to identify regions of interest in the video, followed by embedding the secret message image into the DWT planes of motion regions. The technique enhances embedding capacity and security against various attacks.

2. **Paper Name:** An Improved Video Steganography: Using Random Key-Dependent
   Authors: Mohammad A. Alia, Khulood Abu Maria
   Abstract: This paper presents an improved approach for video steganography, focusing on achieving performance criteria such as invisibility, payload/capacity, and robustness. The method involves searching for exact matches between secret text and video frames' RGB channels, along with utilizing Random Key-Dependent Data for enhanced security.

3. **Paper Name:** Single level Discrete Wavelet Transform based Video Steganography on Horizontal and Vertical coefficients APPLICATIONS.
   Authors: Meenu Suresh1, Dr. I. Shatheesh Sam
   Abstract: This paper proposes a novel video steganography algorithm based on single-level discrete wavelet transform (DWT). It involves embedding watermark information into horizontal and vertical coefficients of video frames after decomposition using DWT. The method achieves high PSNR value and embedding capacity, outperforming other methods.

4. **Paper Name:** Video Steganography by Neural Networks Using Hash Function.
   Authors: GK.Jayasakthi velmurugan, S.Hemavathi
   Abstract: This paper explores video steganography using a combination of hybrid neural networks and hash functions for security. The method utilizes neural networks and hash algorithms to determine optimal embedding positions in the cover video. Experimental validation is conducted using MATLAB software.

5. **Paper Name:** Data Encryption Decryption Using Steganography
   Authors: Manohar N1, Peetla Vijay Kumar
   Abstract: This paper discusses secure communication through video steganography methods. It compares different methods, including secure base LSB method and Neural Networks Fuzzy logic, based on PSNR and MSE metrics. Experimental results indicate improved security, quality, and accuracy compared to other methods.

APPENDIX C: PLAGIARISM REPORT:

This report aims to provide an analysis of the originality of the content related to video steganography. By checking for instances of copied text from external sources, we ensure the integrity and authenticity of the material. The following sections detail the findings from various plagiarism detection tools and manual checks.

The content was checked against multiple sources, including academic papers, online articles, and published books. The plagiarism detection software used includes Turnitin, Grammarly, and Copyscape.

REFERENCES

[1] Bhargava, S., Mukhija, M. (2019). HIDE IMAGE AND TEXT USING LSB, DWT AND RSA BASED ON IMAGE ST EGANOGRAPHY. ICTACT Journal on Image Video Processing, 9(3)

[2] Srilakshmi, P., Himabindu, C., Chait Anya, N., Muralidhar, S. V., Sumanth, M. V., Vinay, K. (2018). TEXT EMBEDDING USING IMAGE ST EGANOG- RAPHY IN SPATIAL DOMAIN. International Journal of Engineering Technology, 7(3.6), 14.

[3] Krishnaveni, N. (2018). IMAGE STEGANOGRAPHY USING LSB EM- BEDDING WIT H CHAOS. International Journal of Pure and Applied Mathematics, 118(8), 505-509.

[4] Karanjit Kaur Baldip Kaur (2018). "DWT-LSB Approach for Video Steganography using Artificial Neural Network". In International Advanced Research Journal in Science, Engineering and Technology, IARJSET.

[5] Mehdi Boroumand, Mo Chen Jessica Fridich (2018). "Deep Residual Net- work for Steganalysis of Digital Images". 2018 IEEE.

[6] Anamika Saini, Kamaldeep Joshi, Kirti Sharma Rainu Nandal (2017). "An Analysis of LSB Technique in Video Steganography using PSNR and MSE". In International Journal of Advanced Research in Computer Science. IJARCS.

[7] Ramadhan J. Mustafa and Khaled M. Elleithy Eman Abdelfattah (2017). "Video Steganography Techniques: Taxonomy, Challenges and Future Directions. 2017 IEEE

[8] M. Dalal and M. Juneja, "A robust and imperceptible steganography technique for SD and HD videos," Multimed Tools Appl, vol. 78, no. 5, pp. 5769–5789, Mar. 2019, doi: 10.1007/s11042-018-6093-3

[9] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. S. Ho, and K.-H.Jung, "Image steganography in spatial domain: A survey," Signal Processing: Image Communication, vol. 65, pp. 46–66, Jul. 2018, doi: 10.1016/j.image.2018.03.012.