A

Major Project Report

on

REVERSIBLE DATA HIDING IN IMAGE

Submitted in Partial Fulfillment of
the Requirements for the Degree
of

Bachelor of Engineering

in

Computer Engineering

to

Kavayitri Bahinabai Chaudhari
North Maharashtra University, Jalgaon

Submitted by

Trupti Rajendra Kulkarni
Rekha Raju Mahajan
Pooja Gunvantrao Devare
Mayur Arun Mali

Under the Guidance of

Miss. Priti Sharma

DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
2023 - 2024

SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY, BAMBHORI, JALGAON - 425 001 (MS)

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the major project entitled *Reversible Data Hiding in Image*, submitted by

Trupti Rajendra Kulkarni
Rekha Raju Mahajan
Pooja Gunvantrao Devare
Mayur Arun Mali

in partial fulfillment of the degree of *Bachelor of Engineering* in *Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

Date: April 16, 2024
Place: Jalgaon

Miss. Priti Sharma
Guide

Dr. Manoj E. Patil
Head

Prof. Dr. G.K.Patnaik
Principal

Acknowledgements

We would like to express our deep gratitude and sincere thanks to all who helped us in completing this project report successfully. Many thanks to almighty God who gave us the strength to do this. Our sincere thanks to Principal Prof. Dr. Girish. K. Patnaik for providing the facilities to complete this Project report. We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this report. A special thanks to Prof. Dr. Manoj E. Patil, Head of the Department, whose help, stimulating suggestions and encouragement, helped us in writing this report.We would also like to thank Miss. Priti Sharma, our project guide, who has devoted her full effort to guiding us and helping us achieve our goals, as well as for her encouragement in keeping our progress on track. I am also sincerely thankful to Prof.Akash D. Waghamare, Incharge of Project, for his valuable suggestions and guidance. We would also like to appreciate the guidance given by other supervisor that has improved our presentation skills by their comments and tips. Last but not the least, we are extremely thankful to our parents and friends without whom it could not reach its successful completion.

Trupti Rajendra Kulkarni
Rekha Raju Mahajan
Pooja Gunvantrao Devare
Mayur Arun Mali

Contents

List of Figures

List of Tables

Abstract

Reversible data hiding was a crucial technique used in various fields, such as healthcare and copyright protection. It allowed embedding information securely in images while maintaining their quality. In a world facing cyber threats and privacy issues, reversible data hiding served as a vital tool to protect information, ensuring secure communication without compromising content. It was particularly valuable in safeguarding digital privacy in personal photos, confidential documents, and medical records. The emphasis on complete reversibility made it a powerful element in privacy-preserving technologies.

The versatility of reversible data hiding addressed specific needs in various domains, emphasizing its applicability in ensuring data security, integrity, and privacy. Beyond privacy, reversible data hiding could also contribute to authentication and tamper detection. By embedding information that served as a digital signature or fingerprint within an image, it became possible to verify the authenticity of the data and detect any unauthorized alterations.

Chapter 1

Introduction

In the dynamic landscape of digital communication and multimedia exchange, the secure integration of additional information within visual content has become a critical imperative. Images, as carriers of rich visual data, play a pivotal role in various applications, from medical imaging to digital forensics and secure communication. Within this realm, the field of reversible data hiding emerges as a sophisticated and theoretically grounded discipline, addressing the delicate balance between concealing supplementary data and preserving the fidelity of the original image.This technique finds application in a spectrum of scenarios, from safeguarding the integrity of medical images to discreetly embedding copyright information in digital media. As our reliance on visual information grows, the ability to seamlessly integrate additional data while ensuring its reversible extraction becomes not just a theoretical pursuit but a practical necessity.

The organization of this Chapter is as follows. Section 1.1 describes Background of the project. Motivation of the project is represented in Section 1.2 Section 1.3 represents Problem statement of the project. Scope of the project is described in Section 1.4. Section 1.5 describes Objective of the project. Section 1.6 describes the selection of life cycle model. Section 1.7 shows the organization of report and finally, the Summary is described in 1.8 Section.

1.1 Background

In the era of digital communication and information exchange, there is an increasing demand for secure and efficient methods of embedding additional data within images. Reversible data hiding techniques play a crucial role in this context, allowing the concealment of information while ensuring the complete recovery of the original image without any loss of quality.Unlike irreversible methods, which introduce permanent modifications to the host data, reversible data hiding aims to embed information in a manner that facilitates the complete reconstruction of the original data upon extraction.

Reversible data hiding, as a subset of information hiding, is distinguished by its fundamental principle of reversibility. Images, as carriers of rich visual information, have witnessed an increased need for information hiding techniques. Information hiding involves the concealment of additional data within a host medium, such as an image, for various purposes including authentication, copyright protection, and secure communication.

1.2 Motivation

The motivation behind the data hiding in images includes secure communication, data protection, and covert transmission. Reversible data hiding in images can be attributed to the need for secure and efficient ways to embed additional information within images without compromising the quality of the original content. It enables confidential information exchange while avoiding detection and safeguards against unauthorized access or tampering. This technique finds applications in fields such as medical imaging, digital forensics, and confidential data transmission.

1.3 Problem Definition

In the rapidly evolving landscape of digital communication and data exchange, the secure integration of additional information within images, while ensuring complete reversibility, is of paramount importance. Reversible data hiding techniques offer a promising avenue for secure communication and information concealment.The challenge in reversible data hiding within images lies in achieving a high data capacity, minimal perceptual distortion, and resistance to attacks, all while ensuring accurate data retrieval.This problem involves developing techniques for seamless embedding confidential or auxiliary data into digital images while maintaining reversibility ensuring the original image can be perfectly restored post-data extraction.

1.4 Scope

The scope for reversible data hiding is vast and spans across multiple industries and applications, including healthcare, copyright protection, secure communication, privacy-preserving technologies, authentication, multimedia content, cloud computing security, forensic applications, and ongoing research and development. It plays a crucial role in addressing concerns related to data integrity, privacy, and secure communication in a variety of digital contexts.

1.5 Objectives

The proposed system is built to hide the data in image using reversible technique for securing the data.
The objectives of this project are as follows:

- Enhance the adaptability of reversible data hiding techniques to diverse image types, ensuring robust performance across different domains and applications.
- Innovate and implement advanced security measures, including encryption techniques, to safeguard the concealed information against unauthorized access and adversarial attacks.
- Address ethical considerations in the development and deployment of reversible data hiding techniques, promoting responsible use and minimizing the potential for misuse.
- Ensure the preservation of human-perceptible image quality, especially in applications where visual interpretation is critical.
- Foster a dynamic approach to reversible data hiding that anticipates and addresses evolving cyber threats, ensuring continued relevance and effectiveness.

1.6 Selection of Life cycle model for development

The software development life cycle model selected for this project is the Iterative Model. Iterative software development involves repetitive cycles of planning, designing, implementing, and testing. Each iteration enhances the software incrementally, allowing for flexibility and adaptation to changing requirements. This model promotes continuous feedback, improving the product gradually while accommodating evolving needs and ensuring a more responsive and resilient development process

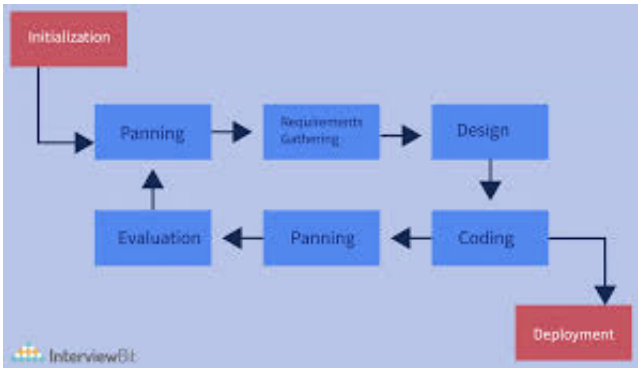

Figure 1.1: Iterative Model

Iteration Model is best suited model for this project.

- It allow for flexibility and adaptability to changing requirements.
- It includes feedback loops at various stages, which can be beneficial for reversible data hiding.
- In reversible data hiding, the process often involves making incremental changes to the image to embed or extract data.
- Iterative development reduces the risk of large-scale project failures because changes are made in smaller increments
- The iterative model promotes continuous improvement throughout the development process

1.7 Organization of Report

Chapter 1: entitled as Introduction describes the details about Background, Problem Definition, Scope and Objective of the project, Identification of Software Development Process Model and Organization of report.

Chapter 2: entitled as Project Planning and Management consists of details about the Feasibility Study, Risk Analysis, Project Scheduling, Effort Allocation and Cost Estimation of the project.

Chapter 3: entitled as Analysis describes in detail, the Requirement Collection and Identification, H/w and S/w Requirements, Functional and Non-Functional Requirements and a Software Requirements Specification(SRS).

Chapter 4: includes design about System Architecture, Data Flow Diagram and various UML Diagrams.

1.8 Summary

As mentioned in above sections, this project aims to hide the data in image in reservible way mean we can extract our data without any damage . The scopes, objective, etc. are as mentioned above. In the next chapter, project planning and management will be discussed.

Chapter 2

Project Planning And Management

Project planning is a procedural step in project management. It is the practice of initiating, planning, executing, controlling and closing the work team to achieve specific goals. Project planning and management is important because it ensures that the right people do the right things, at the right time. It also ensures the proper project lifecycle.

The organization of this chapter is as below. Section 2.1 shows the Feasibility Study of the project. Risk Analysis of the project is represented in Section 2.2 and Project Scheduling is described in Section 2.3. Section 2.4 and 2.5 describe the Effort Allocation and Cost Estimation respectively. The Summary is mentioned in Section 2.6.

2.1 Feasibility Study

A feasibility study is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

The feasibility of reversible data hiding in images involves assessing technical, operational, economic, legal, security, market, scalability, user acceptance, environmental factors, and conducting a risk analysis.There are few types of feasibility that exists. So, developers should take care of these feasibility and take them into consideration:

2.1.1 Technical Feasibility

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost). It is an evaluation of the hardware and software and how it meets the need of the proposed system.

Technical feasibility for data hiding in images assesses the practicality of implementing the required algorithms and methods. It evaluates factors such as algorithm complexity, computational requirements, compatibility with existing systems, and feasibility of embedding and extracting data without significant degradation to image quality or integrity.

2.1.2 Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. The application is operationally feasible since it is build with the idea for integration with various existing applications and systems. Operational feasibility for data hiding in images assesses the practicality of integrating the technique into existing systems and workflows. It examines factors such as ease of implementation, compatibility with image processing software, and the impact on operational efficiency.

2.1.3 Economical Feasibility

Describes how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, and developmental procedures with company revenue prospectus. As the necessary hardware and the software are easily available in the market at low cost, the initial investment is the only cost incurred and does not need further enhancement. Hence it is economically feasible.

Economic feasibility for data hiding in images entails analyzing the costs involved in development, implementation, and maintenance against the potential value and benefits. Factors such as resource allocation, return on investment, and long-term sustainability must be considered to determine the project's economic viability.

2.2 Risk Analysis

Risk Analysis and Management is a key project management practice to ensure that the least number of surprises occur while your project is underway. While we can never predict the future with certainty, we can apply a simple and streamlined risk management process to predict the uncertainties in the projects and minimize the occurrence or impact of these uncertainties.

Risk analysis for data hiding in images involves identifying potential threats such as data corruption, unauthorized access, and legal issues. Mitigation strategies should address encryption methods, authentication protocols, and compliance with data protection regulations to ensure secure and effective implementation.

2.3 Project Scheduling

Project scheduling refers to the estimated time required for a project from its initiation to its completion. It involves setting deadlines for each task planned in advance for the project's culmination. In project management, scheduling encompasses listing milestones, activities, and tasks from the project's start to its end date. A schedule is a vital tool in project planning and management, incorporating attributes such as budget, task allocation, duration, and resource allocation. It serves as a comprehensive roadmap for the successful execution of the project.

Task	Start Date	End Date
Selection of title	8 Aug 2023	11 Aug 2023
Gathering information	14 Aug 2023	22 Aug 2023
Project discussion	23 Aug 2023	30 Aug 2023
Discussion with Guide	31Aug 2023	1 Sep 2023
First Presentation	2 Sep 2023	2 Sep 2023
Planning/ requirement gathering and analysis	5 Sep 2023	26 Sep2023
Final Presentation	29 Sep 2023	29 Sep 2023
Documentation	1Oct 2023	13 Nov 2023
Implementation	11 Jan 2024	7 Feb 2024
Discussion with Guide	7 Feb 2024	13 Feb 2024
Demo Presentation	24 Feb 2024	24 Feb 2024
Documentation	24 Mar 2024	09 April 2024

Table 2.1: Task Scheduling for the project

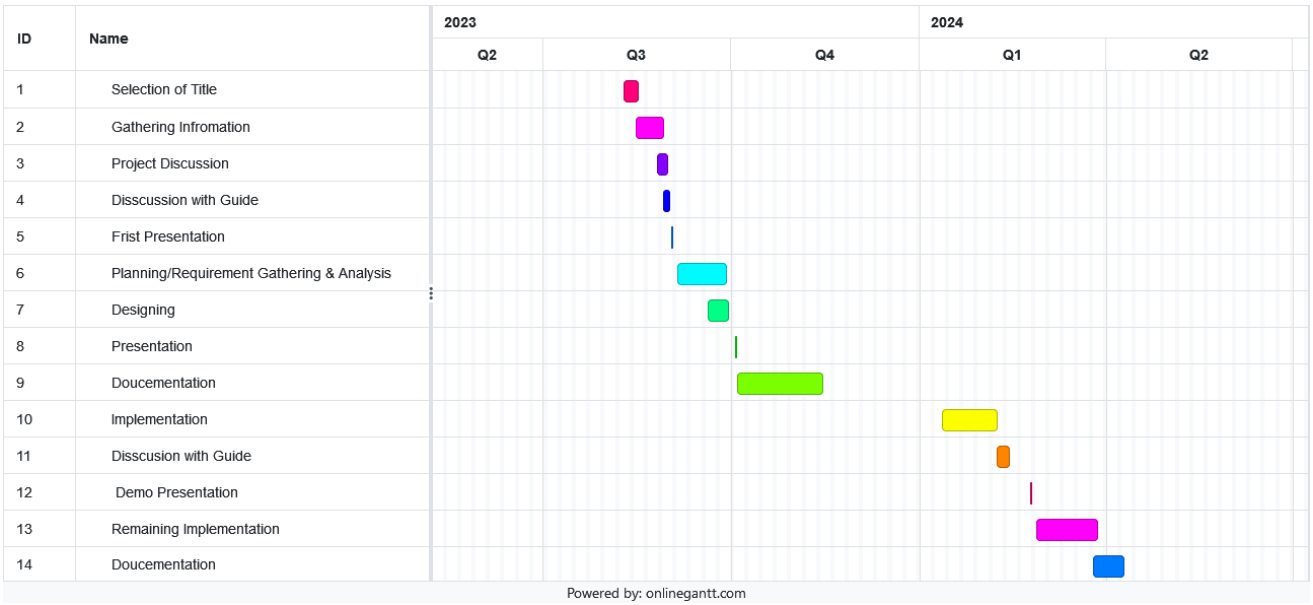

Figure 2.1: Gantt Chart

2.4 Effort Allocation

Effort Allocation is necessary so every team member can give its best to the project. Project was divided into smaller module and task form, for simplification and easy understanding of project overall. Some modules include every team associate's presence to take advantage of team decision taking skills, and some task include some individual member to work on it with precision. We divided the project into 6 modules.

- 1. Gathering of Information
- 2. Planning/Requirement Analysis
- 3. Selection of Life Cycle
- 4. Planning and Management
- 5. Analysis Design UML

Module	Trupti	Rekha	Pooja	Mayur
Gathering of information	✓	✓	✓	✓
planning / requirement analysis		✓	✓	
selection of life cycle	✓			✓
planning and management	✓	✓		
analysis and design			✓	✓

Table 2.2: Effort Allocation

2.5 Cost Estimation

2.6 Cost Estimation

The basic COCOMO estimation model is given by the following expressions:

Software project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.22	2.5	0.33
Embedded	3.6	1.20	2.5	0.32

$$\text{Effort} = \text{a x } (KLOC)^{b}\text{PM} \text{ ..... Eq.(1)}$$

$$\text{Tdev} = \text{c x } (EFFORT)^{d}\text{Months} \text{ ..... Eq.(2)}$$

$$\text{Productivity} = \frac{KLOC}{PM}\text{.....Eq.(3)}$$

Where,

* Equation 2.1 shows estimation of development of effort, expressed in person months (PM).

* Equation 2.2 shows Tdev is the estimated time to develop the software, expressed in months,

* KLOC is the estimated size of the software product expressed in Kilo Lines of Code

KLOC = 830

Software Project Type = Organic

Calculation:

E= a $(KLOC)^{b}$

Time= c$(Effort)^{d}$

$$\text{Person required} = \text{Effort}/\ \text{Time}$$
$$E = 2.4(830)^{1.05}$$
$$E = 4.96PM$$
$$\text{Time} = 2.5(4.96)^{0.38}$$
$$\text{Time} = 4.5$$

The salary of software engineers is Rs.20,000/- per month. The cost required to develop the product is cost = 4.5 × 20000 cost = 90,000
Thus, Rs.90,000/- is the total cost to develop the system

2.7 Summary

The project, is hence found to be feasible since there is a balance of resources required and the cost incurred. The project will be able to easily integrate with other required systems.The project demonstrates seamless integration capabilities with other essential systems.

Chapter 3

Analysis

The development of computer-based information system includes the system analysis phase which produces or enhances the data model which itself is to creating or enhancing a database.There are a number of different approaches to system analysis.The analysis is the process which is used to analyze, refine and scrutinize the gathered information of entities in order to make consistence and unambiguous information. Analysis activity provides a graphical view of the entire System. System Analysis is the process of gathering and interpreting facts, diagnosing problems and using the facts to improve the system. System analysis chapter will show overall system analysis of the concept, description of the system, meaning of the system. System analysis is the study of sets of interacting entities, including computer system analysis.

The organization of this Chapter is as follows. Section 3.1 represents Requirement Collection and Identification. Software Requirement and Specification are described in the Section 3.2. Section 3.3 describes summary of the chapter.

3.1 Requirement Collection and Identification

Requirement collection is the process which is used to gather, analyze, and documentation and reviews the requirements. Requirements describe what the system will do in place of how. In practical application, most projects will involve some combination of these various methods in order to collect a full set of useful requirements. Requirements collection is initiated when the project need is first identified and the project "solution" is to be proposed. Requirements refinement continues after the project is "selected" and as the scope is defined, aligned and approved.

To gather and identify requirements for a project involving reversible data hiding in images, define project objectives, collect user and technical requirements, address security, legal, and integration considerations, establish performance expectations, assess scalability

and flexibility needs, define testing and validation criteria, specify documentation requirements, and identify training needs for relevant personnel. This comprehensive approach ensures a clear understanding of project goals and a roadmap for successful implementation.

3.2 Software Requirements Specifications (SRS)

Software Specification will provide a broad understanding of the requirement specification of this system. Also, understand features of this system along with the requirements. Software Requirement Specification documents guide the developers in the development process and it will help to reduce the ambiguity of the requirements provided by the end-user. It's used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

3.2.1 Product Feature

The product features are high level attributes of a software or product such as software performance, user-friendly interface, security portability, etc. These attributes are defined according to the product, in this case, a software product.
They are as follows:

- Intuitive User Interface: Design a user-friendly interface that allows users to easily upload, manage, and extract hidden data from images.
- Encryption and Security Measures: Implement robust encryption algorithms to secure embedded information and protect against unauthorized access.
- Watermarking Options: Include features for digital watermarking to assist in copyright protection and ownership identification.
- Password Protection: Integrate password protection options to enhance the security of embedded data and restrict access to authorized users.
- User Authentication and Authorization: Include robust user authentication and authorization mechanisms to control access to sensitive functionalities.

3.2.2 Operating Environment

The software will operate within the following environment:

- Operating System: Windows 7 or above/Linux/MacOS
- Any system with at least 4GB RAM

- System with processor Intel CORE i3 or later

3.2.3 Assumption

- Adequate Technical Infrastructure: Assuming the availability of the necessary technical infrastructure, including hardware and software, to support the implementation of reversible data hiding.
- User Familiarity:Assuming users have a basic understanding of image processing concepts and are familiar with using similar tools or software.
- It is assumed that the user will have a working internet connection with sufficient internet speed
- Compatibility with Image Formats: Assuming compatibility with commonly used image formats and an ability to adapt to changes in image processing standards.

3.2.4 Functional Requirements

Functional requirements are the functions which are expected from the software or platform. Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.
Functional requirements are as follows:

- Image Upload and Selection:Users should be able to upload and select images for data hiding.
- Data Embedding and Extraction:The system must embed data within selected images and extract hidden data when required.
- Encryption and Security:Implement encryption algorithms to secure the embedded data and ensure the security of the reversible data hiding process.
- Password Protection:Implement password protection options to restrict access to embedded data.

3.2.5 Non-Functional Requirements

Non-functional Requirement is mostly quality requirement. That stipulates how well the portal does, what it has to do. Other than functional requirements in practice, this would entail detail analysis of issues such as availability, security, usability and maintainability.
Non-functional requirements are as follows:

- The system should have low latency and high throughput when embedding and extracting data from images.
- The solution should be scalable to handle a growing number of images and increased data volumes.
- The system should operate reliably without errors, ensuring consistent data embedding and extraction.
- The reversible data hiding service should be available for use within agreed-upon timeframes, minimizing downtime

3.2.6 External Interfaces

External interfaces for a project involving reversible data hiding in images are the points of interaction and communication between the system and external entities, including users, other systems, or external services.Some aspects to consider in terms of external interfaces:

- User interface
  Provide a user-friendly interface for users to interact with the reversible data hiding system, allowing them to upload, manage, and extract hidden data from images.
- Integration with Image Editing Tools
  Enable integration with popular image editing software or tools, streamlining the workflow for users who are accustomed to using specific applications.
- Authentication Mechanisms
  Include robust authentication mechanisms to control access to the system, ensuring that only authorized users or systems can interact with the reversible data hiding functionalities
- Software Interface

3.3 Summary

In the chapter, Analysis was presented which included the hardware and software requirements, functional and non-functional requirements and the software requirements specification(SRS) as well. In the next chapter, Design is described along with various UML diagrams.

Chapter 4

Design

Design is the activity to design and model the various component of software system. The system design provides the understanding and procedural details necessary for implementing the system. Design is helpful for a better understanding of the project. It contains the UML diagrams, data flow diagrams. UML is a modeling language which is used to document the object-oriented analysis and design.

The organization of this Chapter is as follows. Section 4.1 describes the system architecture of the project. DFD of the project are represented in Section 4.2. Section 4.3 represents UML Diagrams (Use case, Class, Sequence, Component, Deployment, State chart, Activity diagram, Class Diagram, Component Diagram, etc.) of the project. Finally, the Summary is described in last Section 4.4

4.1 System Architecture

Systems Architecture is a generic discipline to handle objects (existing or to be created) called "systems", in a way that supports reasoning about the structural properties of these objects. The system architecture is the conceptual model that defines the structure, behavior and more views of a system.

An architecture description is a formal description and representation of a system. It provides broad understanding of the portal. In the system architecture database provide the functionality like get information, select criteria, etc. to users.

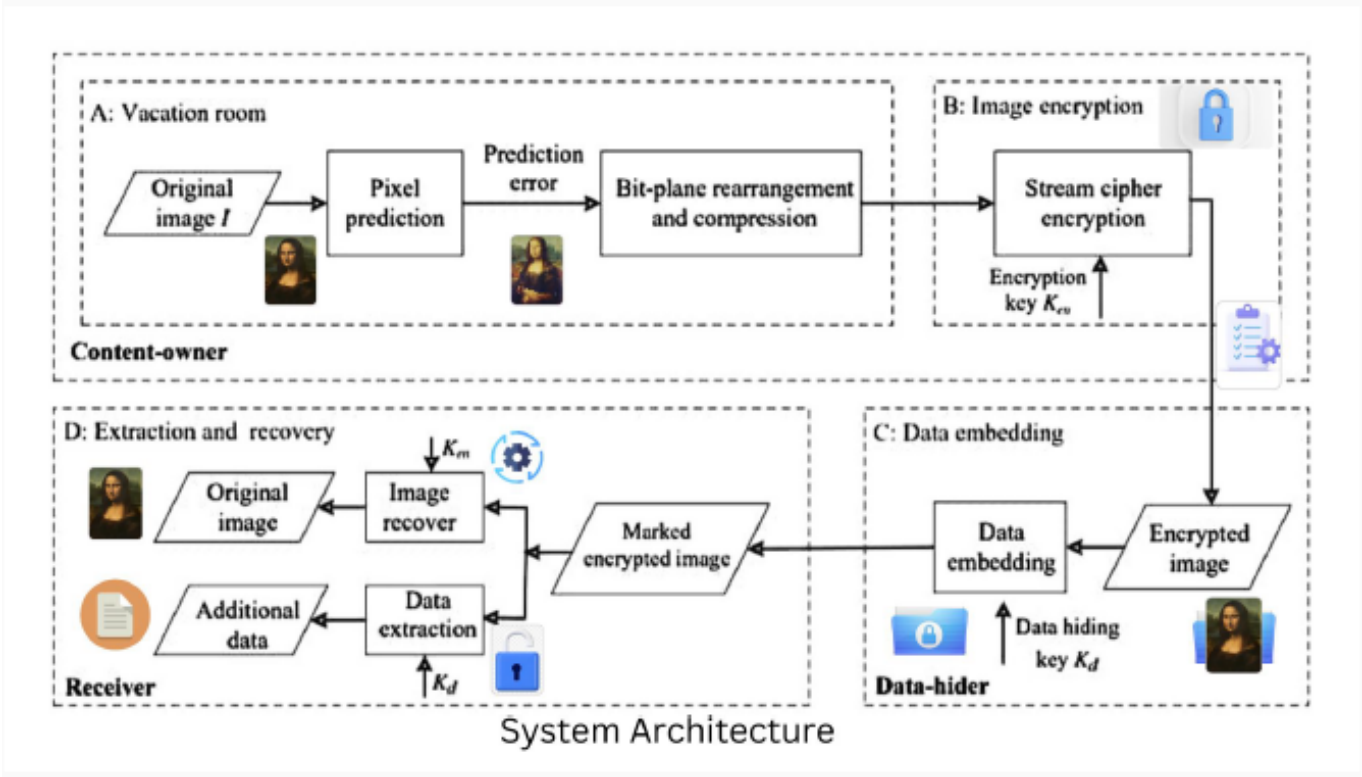

Figure 4.1: System Architecture

4.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the 'flow' of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

4.2.1 Level 0 DFD

Level 0 contains one input and one output. The system provides information to the user means system is input and the user is output. Figure 4.2 shows Level 0 DFD of project.

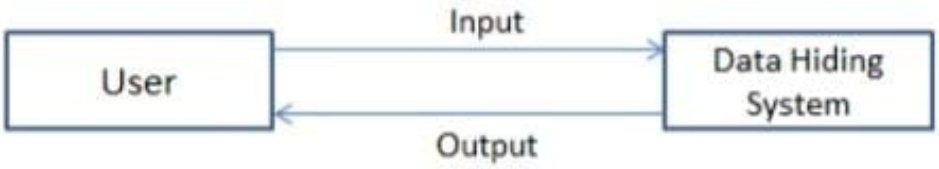

Figure 4.2: 0 Level Data Flow Diagram

4.2.2 Level 1 DFD

A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an "exploded view" of the context diagram. Figure 4.3 shows Level 1 DFD of project.

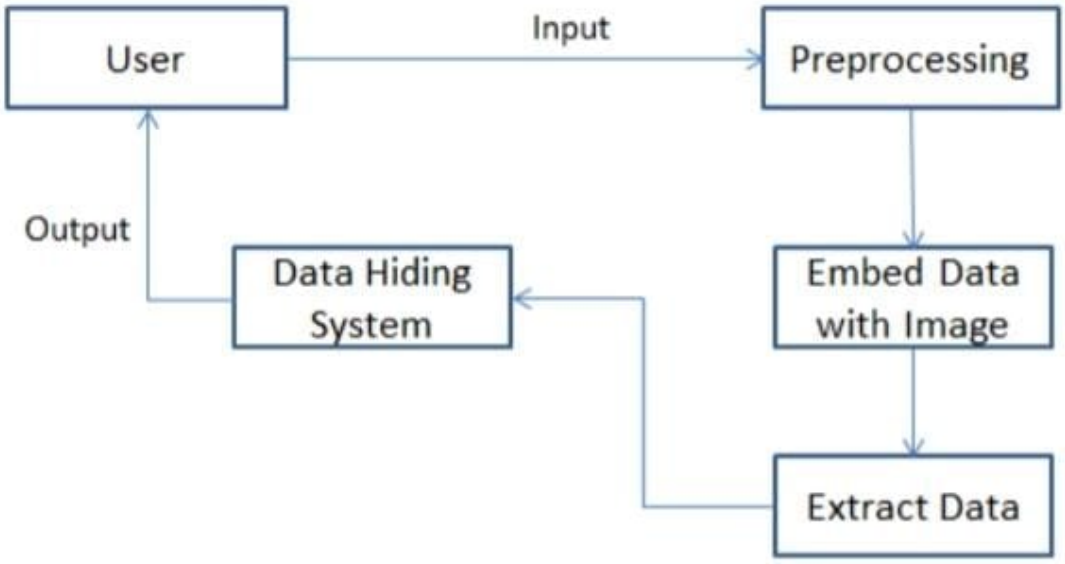

Figure 4.3: 1 Level Data Flow Diagram

4.3 UML Diagrams

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

4.3.1 Use Case Diagram

Use case diagram shows the interaction between Use case which represents system functionality and actor which represent the people or system.

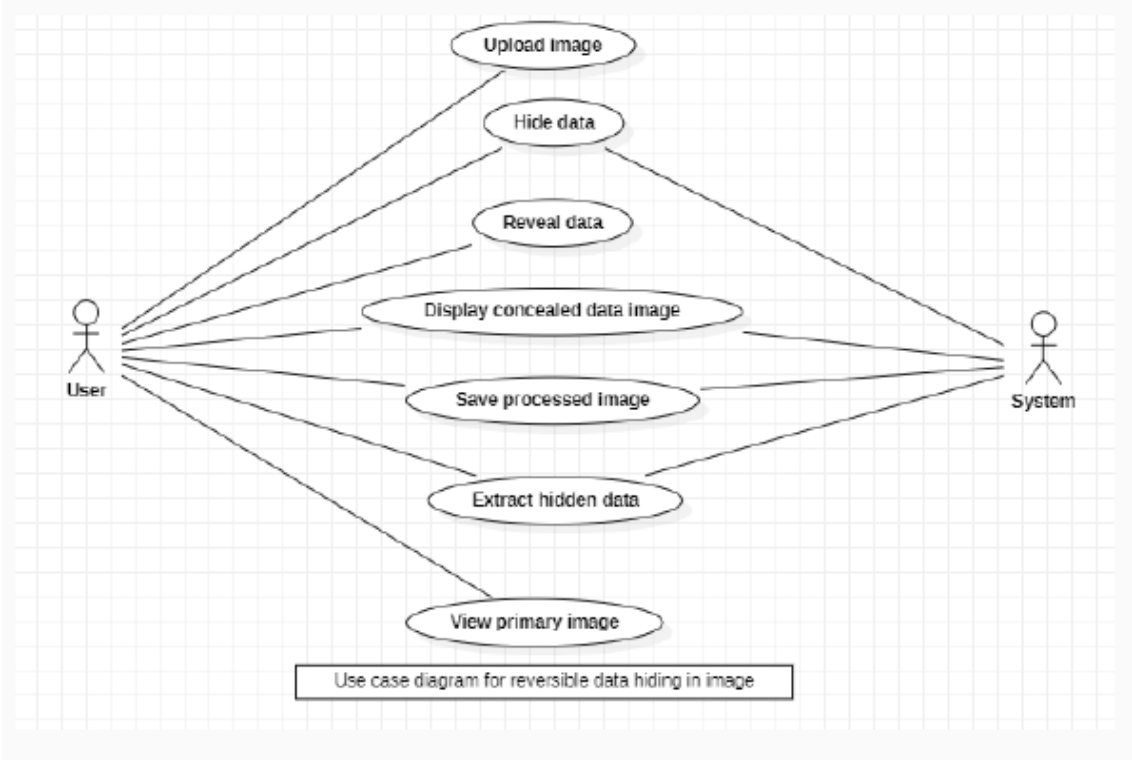

Figure 4.4: Use Case Diagram

4.3.2 Sequence Diagram

The sequence diagram shows the flow of functionality through Use case. A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

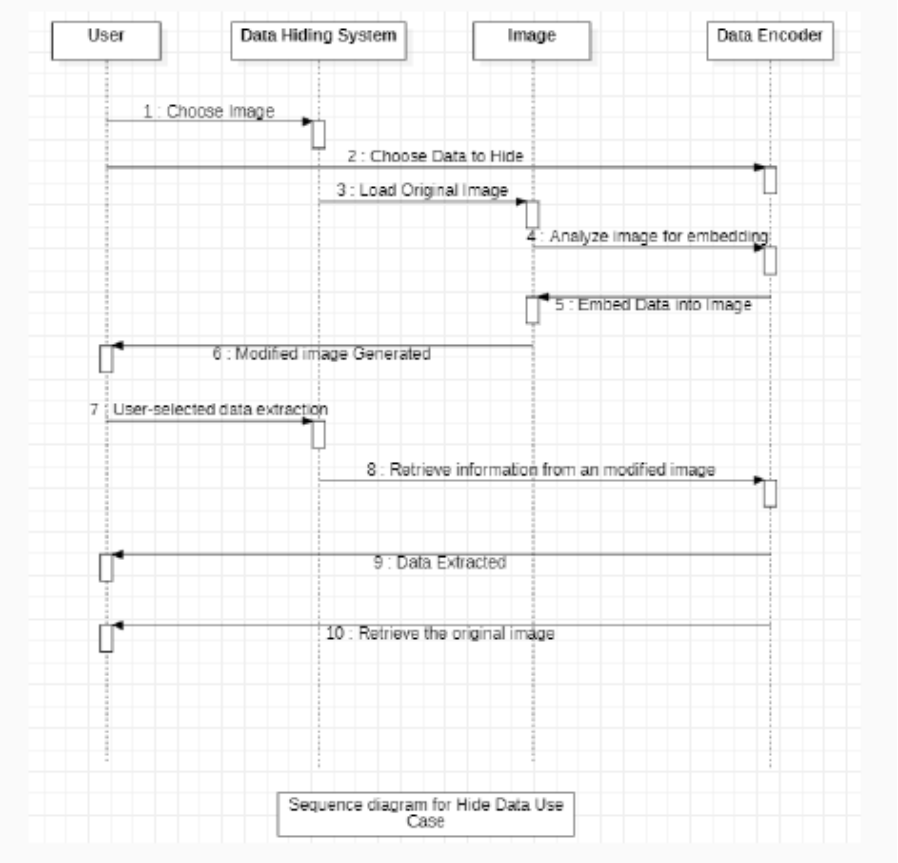

Figure 4.5: Sequence Diagram

4.3.3 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

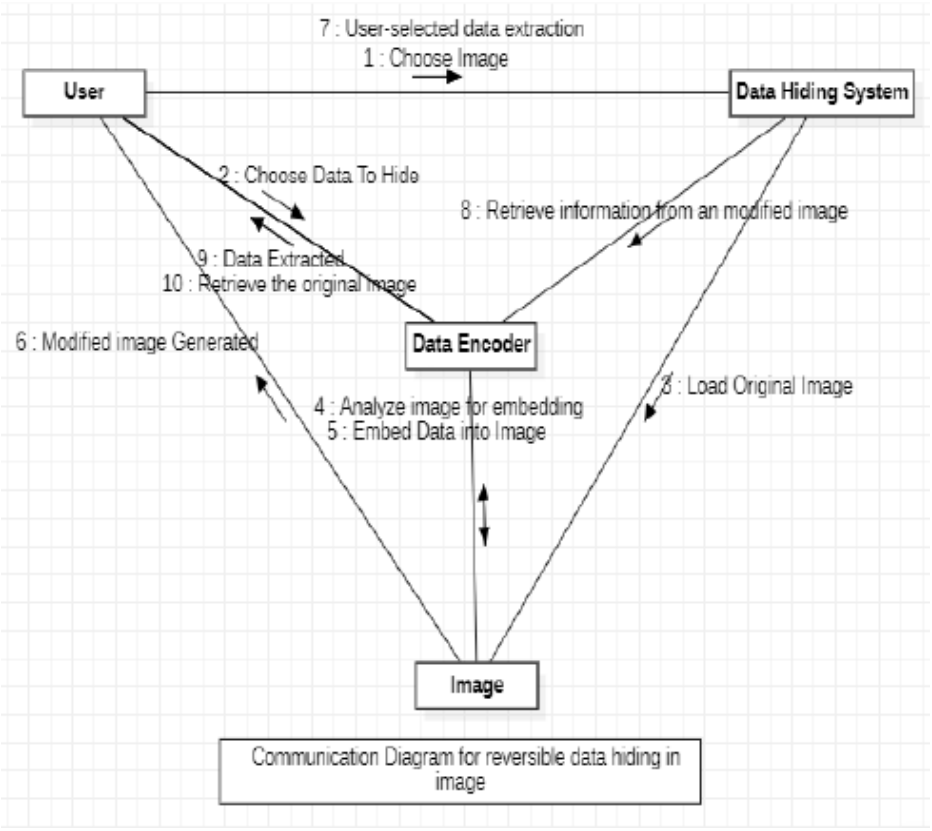

Figure 4.6: Collaboration Diagram

4.3.4 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.

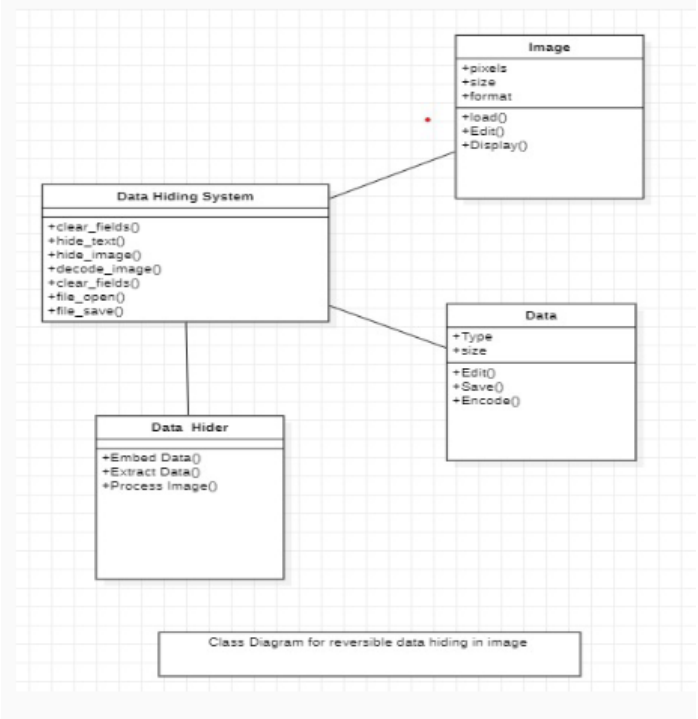

Figure 4.7: Class Diagram

4.3.5 State Chart Diagram

The name of the diagram itself clarifies the purpose of the diagram and other details.It describes different states of a component in a system. The states are specific to component/object of a system. A State Chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

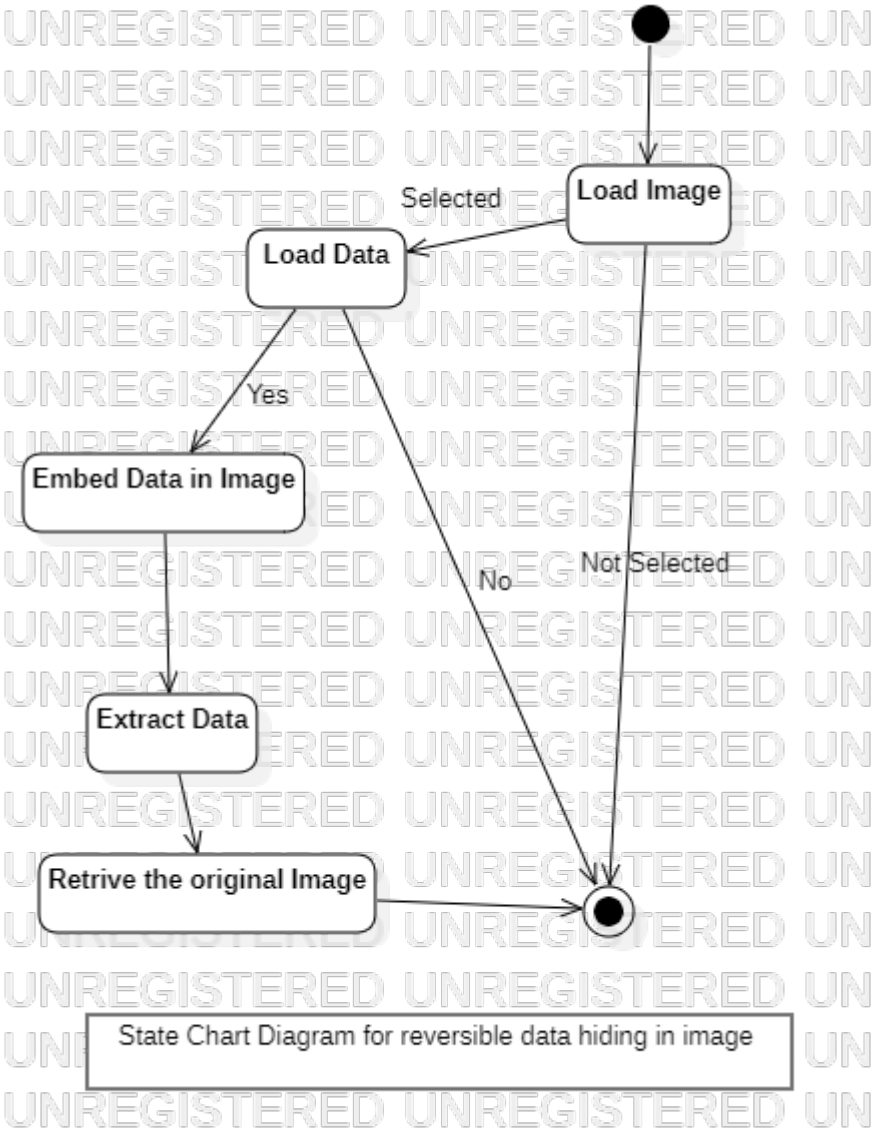

Figure 4.8: State Chart Diagram

4.3.6 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.

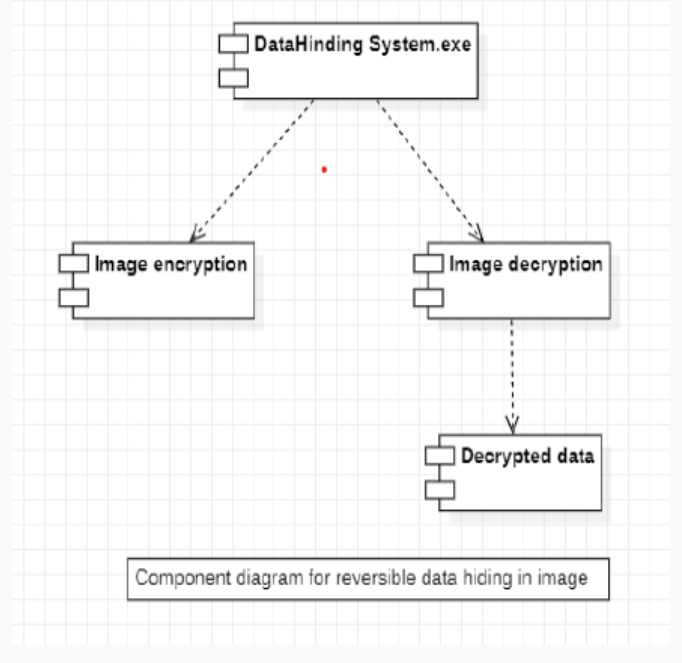

Figure 4.9: Component Diagram

4.3.7 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middle ware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.

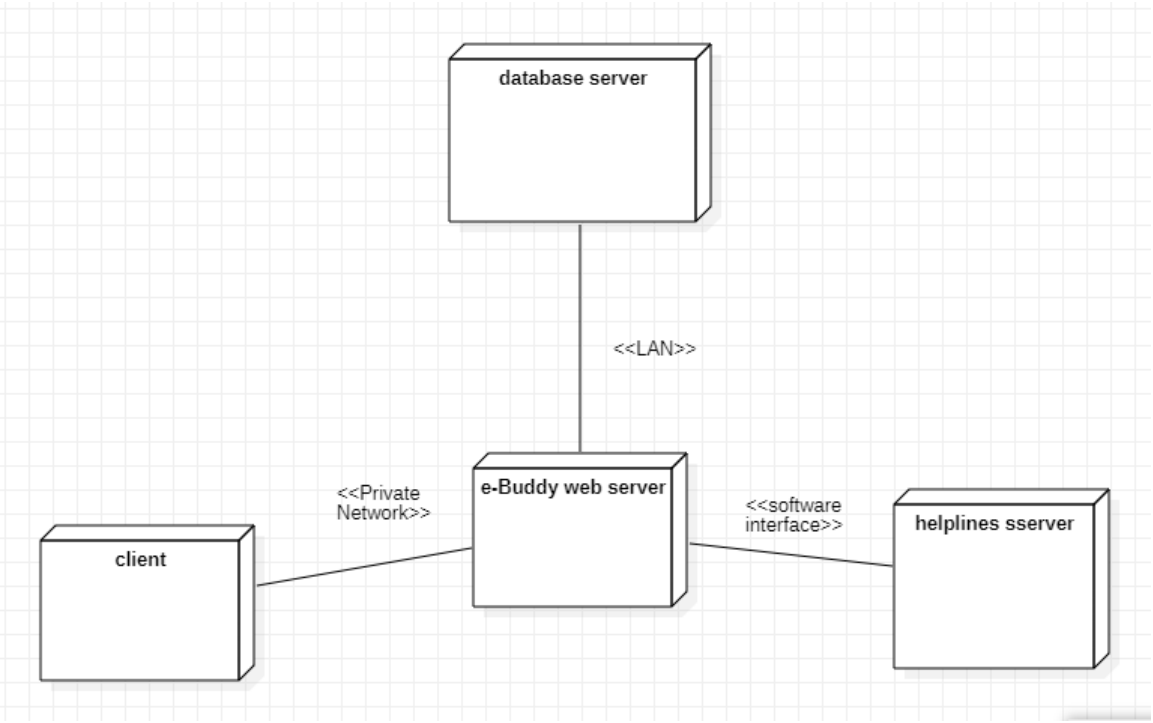

Figure 4.10: Deployment Diagram

4.4 Summary

Detailed design of project has been described in this chapter including the Data Flow Diagrams and the UML Diagram explaining all the design details of the project. Conclusion of the project has been explained in the next chapter.

Chapter 5

Implementation

Important phase in system development is the successful implementation of the new system design. Implementation includes all those activities that take place to convert from the old system to the new system. Project Implementation is a practice of executing or carrying out a project under a certain plan in order to complete this project and produce desired results. Such a practice encompasses all processes and activities involved in getting the project plan fulfilled and accomplishing project goals and objectives.

Section 5.1 describes the Algorithm. Software and hardware for development is discussed in detail in Section 5.2. Section 5.3 describes the various Modules in Project

5.1 Steps

Following are the steps that should be carried out for the execution of data hiding system

1.For Encryption

- Start: Display Homepage
- Upload the image for encryption
- Give the data which you want to hide in image
- Click on encrypt button
- Encrypted image is ready to send

2.For Decryption

- Start: Display Homepage
- Upload the encrypted image
- Click on decrypt button
- Image is decrypt and we can reval the data

5.2 Software and Hardware for development in detail

- Python is a high-level, versatile programming language known for its simplicity and readability. It emphasizes code readability and allows developers to express concepts in fewer lines of code compared to other languages. Python supports multiple programming paradigms and has a vast ecosystem of libraries for various tasks.

- Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. We chose Django for this project due to its built-in features for authentication, URL routing, and database interaction, which streamline web development. Its scalability and extensive documentation make it ideal for building robust web applications efficiently.

5.2.1 Hardware Requirements

- **Dedicated PC/Mobile Phone**: For user, it is necessary to connect to the website. To visit the website, user required the dedicated PC,laptop, tablet or Mobile phone. Similarly, this PC also required to developer while developing the System.

- **Internet**: For connecting end user to the developed system, Internet is required. Into the internet nowadays, the 4G and 5g speed Internet is best option for reaching the website because it generates the interest into the user and view contents of the website in a efficient manner.

5.3 Modules in project

There are only two module

1. **Encryption:**

   Encryption is the process of converting plaintext or original data into an encoded format (ciphertext) that can't be easily understood by unauthorized parties. In the context of data hiding in images, encryption is typically applied to the data that you want to hide before embedding it into the image. This ensures that even if the image is intercepted or accessed by unauthorized users, they won't be able to extract the hidden data without the decryption key. Common encryption techniques used in such projects include symmetric encryption algorithms like AES (Advanced Encryption Standard) or asymmetric encryption algorithms like RSA.

2. **Decryption:** Decryption is the process of converting the encrypted data (ciphertext) back into its original, plaintext form. In the context of data hiding in images, decryption is performed after the image with the hidden data is retrieved. The decryption algorithm utilizes a decryption key to reverse the encryption process and reveal the original data. The decryption key is essential for decrypting the data correctly. It must be kept secure and only shared with authorized parties who need to access the hidden information. The decryption process ensures that the hidden data remains confidential and can be accessed only by those who have the appropriate decryption key.

Chapter 6

Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Software Testing is a process of verifying and validating whether the program is performing correctly with no bugs. It is the process of analyzing or operating software for the purpose of finding bugs. It also helps to identify the defects / aws / errors that may appear in the application code, which needs to be fixed. Testing not only means fixing the bug in the code, but also to check whether the program is behaving according to the given specifications and testing strategies. This Chapter is organized as follows.

Section 6.1 describes Black Box Testing and white Box Testing. Section 6.2 describes Manual Testing and Automated Testing. Test Cases Identification and Execution describe in Section 6.3. Finally, summary of the chapter is given in last section

6.1 Black Box and White Box Testing

The following methodologies are used for testing.

6.1.1 Black Box Testing

Black Box testing also known as Behavioral Testing, is a software testing method in which the internal structure/design implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories

- Incorrect or missing functions
- Interface errors

- Errors in data structures or external database access
- Behaviour or performance errors
- Initialization and termination errors

Our focus is on ensuring that the software effectively hides data within images without compromising the image quality or integrity. We will test for various scenarios, such as embedding different types of data into images, checking for errors in the hiding process, and verifying that the hidden data can be successfully extracted without corruption. Our testing will encompass both functional aspects, like verifying correct data embedding and extraction, as well as non-functional aspects, such as performance and reliability under different conditions.

6.1.2 White Box Testing

White Box Testing is also known as Clear Box Testing, Open Box Testing, Glass Box Testing,Transparent Box Testing, Code-Based Testing or Structural Testing. It is a software testing method in which the internal structure, design, implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.This method is named so because the software program, in the eyes of the tester, is like a white or transparent box; inside which one clearly sees.

White box testing for data hiding in images involves examining the internal structure of the image processing software. We'll analyze the source code to ensure that the data hiding algorithm is implemented correctly and efficiently.Through white box testing, we aim to validate the correctness and robustness of the data hiding implementation within the software.

6.2 Manual and Automated Testing

Following methodologies are used for testing.

6.2.1 Manual Testing

It is the oldest and most rigorous types of testing it is performed by human sitting in front of a computer carefully going through applications screens, trying various usage and input combinations, comparing the results to the expected behavior and recording their

observations about project. There are certain ways of manual testing first of all test cases are written then they are executed and then report is generated according to test result.

6.2.2 Automated Testing

Automation testing is a Software testing technique to test and compare the actual outcome with the expected outcome. This can be achieved by writing test scripts or using any automation testing tool. Test automation is used to automate repetitive tasks and other testing tasks which are difficult to perform manually. The benefit of manual testing is that it allows a human mind to draw insights from a test that might otherwise be missed by an automated testing program. Automated testing is well-suited for large projects; projects that require testing the same areas over and over; and projects that have already been through an initial manual testing process.

6.3 Test Case Identification and Execution

6.3.1 Test Cases

Test Case is the set of inputs along with the expected output and actual output some additional information.

ID	Scenario	Input	Expected Output	Actual output	Result
1	Original image file in PNG format	Image	Image upload Successfully	Image upload Successfully	Pass
2	Data to be hidden in text format	text	Data Hidden Successfully	Data Hidden Successfully	Pass
3	Original image file with high resolution. Large volume of data to be hidden	Image/text	Modified image file with embedded data that maintains the original resolution and visual quality	Modified image with embedded large data	Pass
4	Original image file with significant detail and color variation, along with embedded diverse data types, such as text, an audio clip	Image and combination of text and an audio clip	Modified image file with embedded data that retains the original image's detail and color fidelity.	Modified image with embedded diverse data	Fail
5	Original image file in JPEG,JPG format	Image	Image upload Successfully	Image format is invaild	Fail

Chapter 7

Result and Discussion

This chapter discusses about the results of the project. We successfully implemented a reversible data hiding technique in image steganography. The technique involved embedding data within digital images while maintaining the ability to fully recover both the original image and the hidden data. The results indicated that the encryption process incurred minimal overhead, enabling efficient embedding of data within images without significant degradation in image quality.The reversible data hiding technique, included secure communication, digital watermarking, and information embedding in multimedia content. By integrating the technique into various applications, we demonstrated its versatility and potential for enhancing data security and privacy in diverse domains. the results of the data hiding in image project underscored the effectiveness and practical utility of the reversible data hiding technique.

7.1 Output

7.1.1 Home Page

Our project focuses on steganography, hiding data within images. Our homepage offers an Encryptor to embed data and a Decryptor to extract it. Users can explore covert communication through our intuitive interface.

Figure 7.1: Home Page

7.1.2 Encryption

Our project focuses on steganography, hiding data within images. Our homepage offers an Encryptor to embed data and a Decryptor to extract it. Users can explore covert communication through our intuitive interface.

Figure 7.2: Encryption

7.1.3 Decryption

The Decryptor page simplifies the process of revealing hidden data from encrypted images. Users can effortlessly upload their encoded images and initiate decryption. With a user-friendly interface, our Decryptor ensures secure and convenient access to concealed information, providing a seamless experience for users.

Figure 7.3: Decryption

7.1.4 Explore

The Explore page is your gateway to understanding steganography. Discover how data hides within images through curated resources and examples. Dive into the world of covert communication with ease.

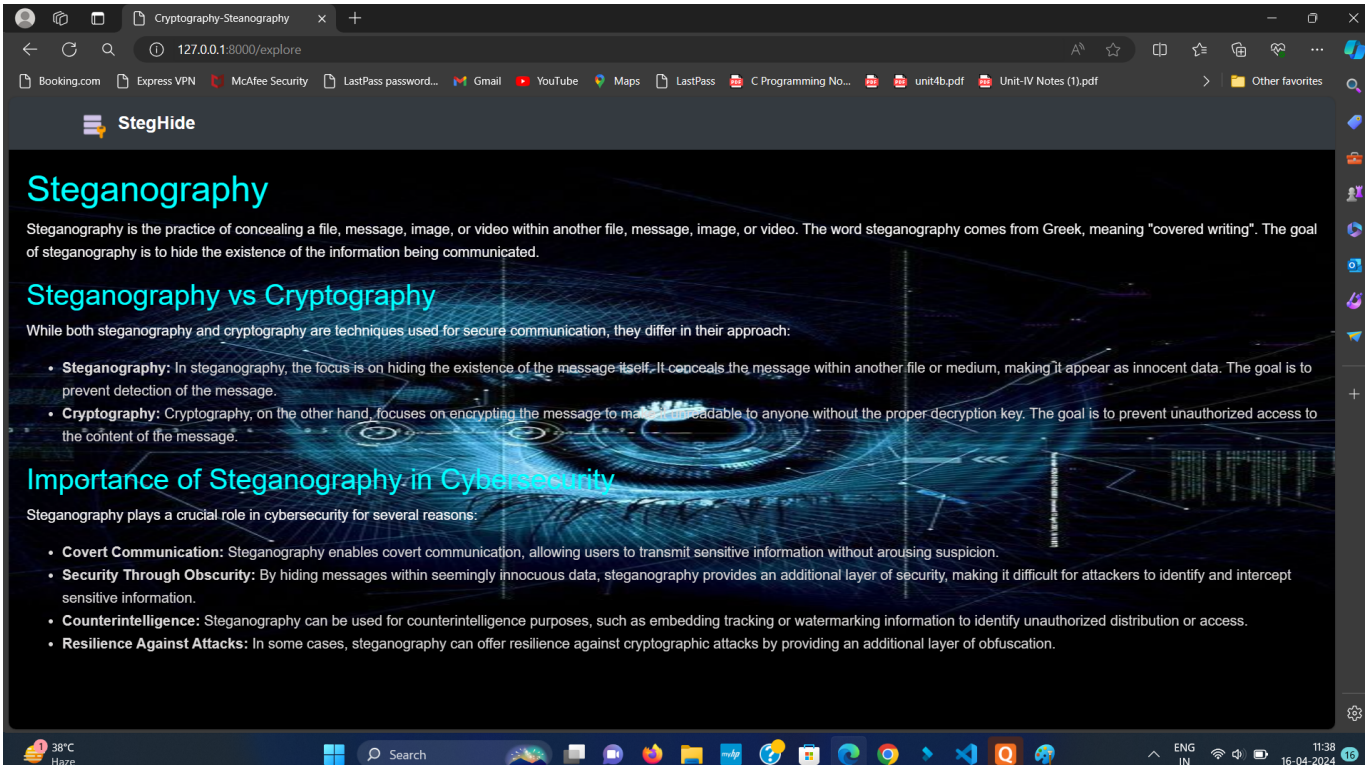

Figure 7.4: Explore Page

Chapter 8

Conclusion and Future Work

Reversible data hiding in image steganography offers a valuable technique for concealing data within images while preserving the ability to fully recover both the original image and the hidden data. This approach has a wide range of applications in secure communication, watermarking, and information embedding, making significant contributions to data security and privacy. By embedding data in images in a reversible manner, users can ensure that the integrity of both the image and the hidden data remains intact, even after multiple rounds of processing or transmission. This capability is essential in scenarios where maintaining the fidelity of the original image and the confidentiality of the hidden data are paramount concerns.

Moving forward, exploring encryption in different formats of images presents an intriguing avenue for further research and development in reversible data hiding projects.

1. Exploring Various Image Formats
2. Performance Optimization
3. Security Analysis

Bibliography

[1] Zhaoxia Yin, "Reversible Data Hiding in Encrypted Images based on Pixel Prediction and Bit-plane Compression", IEEE August 2015

[2] Zhenyu Guan, "DeepMIH: Deep Invertible Network for Multiple Image Hiding", IEEE, January 2023

[3] R. Gurunath, "Social Media and Steganography: Use, Risks and Current Status", IEEE November 2021

[4] Nandini Subramanyam, "Image Steganography: A Review of the Recent Advances", IEEE February 2021