PUBLIC OBJECTION BY CRIMINAL TO POLICE HEADQUARTERS SHORT A WAY SET CASUALTY

A PROJECT REPORT

Submitted by

PAVITRAA D S B [211420205106]

KAARTHIKA N [211420205070]

SANGEETHA G [211420205131]

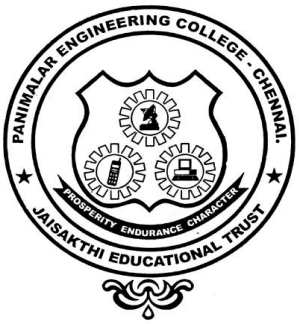
in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY



PANIMALAR ENGINEERING COLLEGE, POONAMALLEE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution,Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"PUBLIC OBJECTION BY CRIMINAL TO POLICE HEADQUARTERS SHORT A WAY SET CASUALTY"** is the bonafide work of **"PAVITRAA D S B(211420205106), KAARTHIKA N (211420205070), SANGEETHA G (211420205131)"** who carried out the project under my supervision.

SIGNATURE

Dr. M. HELDA MERCY M.E., Ph.D.
HEAD OF THE DEPARTMENT

Department of Information Technology

Panimalar Engineering College

Poonamallee, Chennai - 600 123

SIGNATURE

V.PRIYADARSINI, B.Tech, M.tech
SUPERVISOR
ASSISTANT PROFESSOR

Department of Information Technology

Panimalar Engineering College

Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on _____

SIGNATURE

INTERNAL EXAMINER

SIGNATURE

EXTERNAL EXAMINER

DECLARATION

I hereby declare that the project report entitled "**PUBLIC OBJECTION BY CRIMINAL TO POLICE HEADQUARTERS SHORT A WAY SET CASUALTY**" which is being submitted in partial fulfillment of the requirement of the course leading to the award of the 'Bachelor Of Technology in Information Technology ' in **Panimalar Engineering College, Autonomous institution Affiliated to Anna University- Chennai** is the result of the project carried out by me under the guidance of **V.PRIYADARSINI B.Tech, M.Tech ASSISTANT PROFESSOR in the Department of Information Technology**. I further declare that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

(PAVITRAA D S B)

Date**:** **(KAARTHIKA N)**

Place**:** Chennai **(SANGEETHA G)**

It is certified that this project has been prepared and submitted under my guidance.

Date: V .PRIYADARSINI, B.Tech, M.Tech

Place: Chennai (ASSISTANT PROFESSOR / IT)

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion. We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Beloved Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors, Mrs. C. VIJAYA RAJESHWARI and Dr. C. SAKTHI KUMAR, M.E., M.B.A., Ph.D.,** and **Dr. Saranya Sree Sakthi Kumar., B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY, M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our Project coordinator **Mr. M. DILLI BABU**, **M.E.,(Ph.D.,)** Associate Professor, Department of Information Technology for his guidance throughout the course of our project. We also express sincere thanks to our supervisor **V.PRIYADARSINI B.Tech, M.Tech qualification**, Assistant Professor for providing the support to carry out the project successfully. Last, we thank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

TABLE OF CONTENTS

ABSTRACT

The project endeavors to revolutionize the public objection process within criminal cases through the implementation of a robust and secure system. Leveraging the Java programming language and the SHA-256 algorithm of the blockchain, the system is designed to provide a reliable and efficient platform for managing objections. Key components of the system include a user-friendly login and registration interface, as well as specialized modules tailored for use by police headquarters and administrators. Ensuring the authenticity and integrity of objection submissions lies at the core of the system's design. By harnessing the decentralized and immutable nature of blockchain technology, objection records are securely stored and tamper-proof. Additionally, the SHA-256 algorithm is employed to encrypt data, further bolstering the security of objection records and maintaining their integrity The login and registration page facilitate secure access for users, allowing them to submit objections directly to police headquarters. Transparency and accountability are fundamental principles embedded within the system. Each objection submission is recorded on the blockchain, providing a verifiable and transparent record of events. Furthermore, the system extends its functionality to enable individuals to securely access their case files using their mobile number and Aadhar number. By generating a QR code that grants access to the report file, this feature enhances accessibility and empowers individuals to engage more actively in the legal process. In summary, the project aims to modernize and streamline the public objection process in criminal cases through the implementation of a secure and efficient system. By leveraging blockchain technology, encryption algorithms, and specialized modules, the system ensures transparency, accountability, and integrity throughout the objection-handling process, ultimately enhancing trust between stakeholders and optimizing workflow efficiency within the criminal justice framework.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

S.NO	ABBREVIATION	EXPANSION
1.	DB	Database
2.	SMC	Secure Multiparty Computation
3.	MDA	Medical Admin
4.	DBC	Data Base Confidentiality
5.	JVM	Java Virtual Machine
6.	JSP	Java Server Page
7.	SHA	Secure Hash Algorithm
8.	AES	Advanced Encrypted Standard

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:

In the landscape of criminal justice systems, the efficient and transparent handling of public objections stands as a cornerstone for maintaining trust and accountability. Recognizing this critical need, a pioneering project is underway to revolutionize the objection process in criminal cases. This innovative endeavor leverages the power of the Java programming language and the robust SHA-256 algorithm of blockchain technology to introduce a secure and streamlined system.

At the heart of this initiative lies the ambition to address public concerns by ensuring the authenticity, integrity, and transparency of objection submissions. By harnessing blockchain technology, the system aims to instill confidence in stakeholders by providing a verifiable and immutable record of objections. This not only enhances accountability but also fosters trust in the criminal justice framework.

The core architecture of the system revolves around a user-friendly login and registration interface. This interface is pivotal in facilitating secure access for users interested in submitting objections. By prioritizing user experience, this aspect of the system establishes a secure connection between the public and the criminal justice system, thereby fostering greater participation and engagement.

A key highlight of this project is its incorporation of specialized modules tailored for both police headquarters and administrators. These modules are meticulously designed to efficiently handle and process objections, thereby optimizing the overall workflow within the criminal justice framework. Through seamless integration with existing systems, these modules streamline operations and enhance the responsiveness of authorities to public concerns.

The decentralized and immutable nature of blockchain technology introduces an additional layer of security to the objection process. By distributing objection records across a network of nodes, the blockchain ensures that no single entity has control over the data, thereby safeguarding against tampering or manipulation. This not only enhances the integrity of objection records but also strengthens the credibility of the entire system.

Furthermore, the utilization of the SHA-256 algorithm reinforces the security of objection submissions. This cryptographic hash function generates a unique digital fingerprint for each objection, making it virtually impossible for malicious actors to alter or forge objection records. As a result, stakeholders can trust in the authenticity and integrity of the information stored within the system.

Furthermore, the adoption of this innovative objection system has the potential to enhance public perception and confidence in the criminal justice system. By providing a secure and transparent platform for submitting objections, individuals feel empowered to voice their concerns without fear of reprisal or dismissal. This increased transparency and accessibility foster a sense of accountability among authorities, encouraging greater responsiveness to public feedback and concerns.

In conclusion, the project to revolutionize the objection process in criminal cases represents a significant step forward in enhancing transparency, accountability, and trust within the criminal justice framework. By harnessing the power of blockchain technology and the Java programming language, this initiative aims to establish a secure and streamlined system that empowers the public to voice their concerns with confidence.

1.2 SCOPE OF THE PROJECT:

Blockchain Integration: Utilizes the decentralized and immutable nature of blockchain technology to ensure the authenticity, transparency, and security of objection submissions. Each objection is recorded on the blockchain, making the records tamper-proof and verifiable.

SHA-256 Cryptographic Algorithm: Implements the SHA-256 algorithm to provide a robust cryptographic mechanism for securing data and maintaining the integrity of objection records. This algorithm ensures that the information submitted remains secure and unaltered.

User Authentication: Incorporates a secure login and registration system to authenticate users, allowing them to securely access the objection submission platform. This feature enhances the overall security of the system and ensures that only authorized individuals can participate in the objection process.

User-Friendly Interface: Develops an intuitive and user-friendly interface for the login and registration pages, facilitating easy navigation for users. This enhances the overall user experience and encourages active participation in the objection submission process

Specialized Modules: Includes distinct modules designed for the police headquarters and the admin, streamlining the objection handling and processing workflow. These modules are tailored to meet the specific needs and responsibilities of each user role, optimizing efficiency within the criminal justice system.

Efficient Objection Handling: Focuses on creating a system that efficiently manages and processes objections, reducing the time and effort required for resolution. The aim is to enhance the overall responsiveness of the criminal justice system to public concerns.

1.3 OBJECTIVE:

Modernize the Objection Process: Implement a secure and efficient system to modernize the objection process in criminal cases. The objective is to replace outdated and inefficient objection processes with a streamlined and secure digital system.

Enhance Transparency: Increase transparency in the objection process by leveraging blockchain technology, allowing stakeholders, including the public, to independently verify and trust the integrity of objection records.

Ensure Data Security: Implement the SHA-256 cryptographic algorithm to safeguard objection records, ensuring the confidentiality and integrity of the information submitted by users.

Optimize User Experience: Develop a user-friendly interface for the login and registration pages to encourage active participation from the public, making the objection submission process accessible and efficient.

Streamline Workflow: Create specialized modules for the police headquarters and the admin to streamline objection handling and processing, reducing response times and enhancing the overall efficiency of the criminal justice system.

Build Trust: Foster trust between the public and law enforcement authorities by providing a secure, transparent, and accountable platform for submitting and handling objections in criminal cases.

Ensure Accountability: Utilize blockchain technology to create an immutable record of objection submissions, promoting accountability within the criminal justice system and reinforcing public trust in the objection resolution process.

1.4 PROBLEM STATEMENT

In the current landscape of criminal justice systems, the objection process within criminal cases faces numerous challenges, including inefficiencies, lack of transparency, and limited accessibility. Traditional methods often involve manual paperwork, leading to delays, errors, and difficulties in tracking objections. Additionally, there is a lack of secure and transparent platforms for individuals to submit objections and access case files, resulting in a loss of trust and confidence in the legal process. Moreover, concerns regarding the integrity and authenticity of objection submissions persist, as existing systems may be susceptible to tampering or manipulation.

These challenges underscore the need for a modernized and efficient objection-handling system that addresses the shortcomings of existing processes. Such a system should prioritize security, transparency, and accessibility while streamlining objection processing for law enforcement authorities. Furthermore, it should leverage innovative technologies such as blockchain and encryption algorithms to ensure the integrity and authenticity of objection records. Additionally, the system should empower individuals by providing them with secure access to their case files, enabling active participation in the legal process and fostering trust between the public and law enforcement authorities.

Overall, the problem statement encapsulates the need for a comprehensive solution that modernizes the objection process, enhances transparency and accountability, and ensures efficient and secure handling of objections within the criminal justice framework.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY:

TITLE: BLOCKCHAIN-BASED SYSTEM FOR EFFECTIVE POLICE COMPLAINT MANAGEMENT

AUTHORS: Bharath D R, Lynsha Helena Pratheeba HP, Cibiya N E, Dheekshitha S, Divya M N

YEAR: June-2023

With the escalating rates of criminal activities and the persistence of archaic complaint management practices, there is an urgent necessity for an advanced and secure system to handle police complaints. This paper introduces a pioneering solution that leverages blockchain technology to revolutionize police complaint management. The proposed system establishes a decentralized platform that ensures the integrity and immutability of complaint records, mitigating the risks associated with tampering and unauthorized access. By employing encryption and blockchain hashing techniques, the system guarantees the authenticity and timestamping of filed complaints, bolstering the evidentiary value of the records. This novel approach eliminates the reliance on outdated manual processes, allowing complainants to securely file complaints remotely at any time. Furthermore, the decentralized nature of the system eradicates the vulnerability of a single point of failure, enhancing resilience and trust in the complaint management process. This research presents a transformative solution that addresses the prevailing challenges in police complaint management, paving the way for an accountable and transparent law enforcement ecosystem.

TITLE: FIR SYSTEM USING BLOCKCHAIN TECHNOLOGY

AUTHOR: Bharath Kumar V, Dr. Mir Aadil

YEAR: March 2023

In police stations, there are records of crimes. Crime Records are unable to locate crimes and the offenders who committed them. To maintain the crime and criminal data under the current system, an FIR is used. It has less security and makes fraud simple to do. Each time, a manual update has been made to the record. This system's primary goal is to secure data utilizing blockchain technology. Using their authentication credentials, Crime Investigators can view the data from the database. The reports, which are prepared by witnesses and police officers, are accessible to the investigator (writer). Investigators have the authority to edit data (i.e., update, remove, and so on), and this data aids investigators in speeding up their investigations and identifying offenders more quickly. Previous research has focused on the centralized handling of digital evidence, however, if a centralized system server is breached, sensitive operational and investigation data may be exposed. As a result, there is a need to manage digital evidence and investigative information in a distributed system setting using blockchain technology. Performance is reduced when massive amounts of data, such as evidential films, are kept in a blockchain because more data must be processed only once before being generated. As a result, we suggest a three-tier blockchain architecture, with hot and cold blockchains for digital evidence. Information that changes regularly is stored on the hot blockchain, whereas material that does not change, such as files, is saved in the cold blockchain. To assess the system, we compared the storage and inquiry processing performance of digital crime evidence across the multi-level blockchain system's capacities.

TITLE: **POLICE COMPLAINT MANAGEMENT SYSTEM USING BLOCKCHAIN TECHNOLOGY**

AUTHOR: Ishwarlal Hingorani; Rushabh Khara; Deepika Pomendkar; Nataasha Raul

YEAR: JANUARY 2021

The rise in criminal activities across India has become a pressing concern, exacerbated by the prevalence of unreported incidents. Despite the availability of online portals for police to store First Information Reports (FIRs) and Non-Cognizable Reports (NCRs), the persistence of handwritten FIRs as a traditional practice undermines the efficiency of law enforcement processes. To enhance governance, the Crime and Criminal Tracking Network and Systems (CCTNS) initiative was introduced in 2009, aiming to streamline law enforcement procedures on a national scale. However, the centralized nature of the CCTNS, limited to individual states, poses inherent risks of system failure and unauthorized access, necessitating a shift towards decentralization. Under this framework, FIRs filed by law enforcement agencies would undergo encryption and be securely stored in the InterPlanetary File System (IPFS), with corresponding hashes added to the blockchain network. This ensures that complaints are securely managed and protected from unauthorized tampering or deletion. The adoption of a blockchain-based solution offers numerous advantages, including the elimination of opportunities for tampering with FIRs or NCRs and the prevention of unnoticed alterations. By centralizing records within an immutable database, the proposed system enhances transparency, accountability, and the integrity of law enforcement processes, ultimately contributing to a more efficient and trustworthy criminal justice system in India.

TITLE: CRIME PREDICTION AND ANALYSIS

AUTHOR: Pratibha Kumari, Akanksha Gahalot, Lokesh Chouhan

YEAR: FEBRUARY 2020

Crime is a pervasive and distressing aspect of modern society, with a constant stream of criminal incidents unsettling the lives of ordinary citizens. Consequently, preventing these crimes before they occur has become a paramount task. In recent years, artificial intelligence (AI) has emerged as a powerful tool in various domains, including crime prediction. However, the effectiveness of such predictive models hinges on the availability of accurate and comprehensive crime data for reference and analysis. Despite the potential benefits, accurate crime prediction remains a challenging endeavor. Researchers have explored various machine learning algorithms, including K-nearest neighbors (KNN) and decision trees, to develop predictive models capable of identifying and forecasting criminal activities. These models leverage historical crime data to identify spatial and temporal patterns, enabling law enforcement agencies to allocate resources effectively and implement targeted intervention strategies. By harnessing the power of machine learning, law enforcement agencies can enhance their strategic capabilities in crime prevention. Predictive analytics enable authorities to prioritize high-risk areas and proactively deploy resources, thereby reducing the overall crime rate and enhancing public safety. The ultimate goal of crime prediction and analysis methods is to provide actionable insights that empower law enforcement agencies to address the underlying causes of criminal behavior and implement targeted interventions to mitigate future crime. By leveraging machine learning techniques and sophisticated algorithms, police authorities can adopt a data-driven approach to crime prevention, thereby fostering a safer and more secure society for all.

TITLE: CRIME PREDICTION USING PATTERNS AND CONTEXT

AUTHOR: Oscar Figueroa, Marcos E. Orchard, Nelson Baloian, Rául Manásevich

YEAR: April 2017

Science fiction has long envisioned the ability to predict future crimes, a concept now within reach thanks to computer algorithms utilizing diverse data sources. These algorithms approximate the time and risk maps for various types of felonies, including home burglaries, armed robberies, and violent thefts. This predictive capability equips law enforcement with valuable insights, enabling targeted patrolling efforts aimed at reducing crime rates.

Our proposed crime prediction solution, tailored for Chilean large cities, adopts a novel approach comprising three distinct software modules utilizing different algorithms. These modules collaboratively generate predictions, leveraging their unique methodologies to enhance accuracy. Through testing on historical data, the system has demonstrated satisfactory performance, deemed suitable for operational deployment by law enforcement.

An intriguing finding from our experiments is that while individual modules exhibit inferior performance compared to the integrated system, their collective output surpasses that of any single algorithm. This underscores the potential of leveraging diverse algorithms to exploit distinct features within the data, enhancing overall predictive capabilities. By embracing this cooperative integration approach, our solution offers law enforcement agencies a powerful tool for preemptive crime prevention and resource allocation.

TITLE: SOCIAL NETWORK ANALYSIS AND ITS CONTRIBUTION TO RESEARCH ON CRIME AND CRIMINAL JUSTICE

AUTHOR: Aili Malm, Martin Bouchard

YEAR: NOVEMBER 2016

Network analysis has enabled criminologists to explore a myriad of new questions, delving into the intricate interplay between individuals, groups, and criminal activities. Moreover, it has enhanced the understanding of longstanding inquiries by providing more nuanced and comprehensive analyses. Through the application of network analysis, criminologists have unearthed novel evidence shedding light on the complexities of criminal behavior and its underlying mechanisms. Law enforcement agencies and other practitioners have leveraged network analysis techniques to identify key actors and hotspots of criminal activity, informing targeted intervention strategies. By mapping out criminal networks and their interactions, practitioners can better allocate resources and implement tailored interventions, thereby enhancing the effectiveness of crime control efforts. However, despite its promise, the utility of network data in understanding crime patterns is not without limitations. Recognizing these limitations is essential for refining methodologies and advancing our understanding of crime dynamics. In conclusion, the development of network analysis techniques has revolutionized both research on crime and the practice of crime control. By enabling a deeper understanding of criminal networks and their behaviors, network analysis has empowered criminologists and practitioners alike to develop more effective strategies for crime prevention and control. However, ongoing efforts to address the limitations of network data are essential for maximizing the utility of these techniques in combating crime effectively.

CHAPTER 3

PROJECT DESCRIPTION

3.1 EXISTING SYSTEM

The study involves utilizing personal information and GPS trajectories to extract features, with a focus on representing individuals and locations as nodes in a network. The objective is to classify crime risks based on these features, creating a Multi-dimension Fusion Information Graph (FIG) that incorporates data from various sources. The proposed FIGAT model, which is built on graph neural networks, effectively addresses the limitations of existing studies that overlook individual GPS trajectory data when inferring crime locations or groups. These studies overlook the comprehensive impact on crime patterns resulting from the internal dynamics among offenders, locations, and time. In contrast, our study introduces Fusion Information Graph Attention Networks (FIGAT) to classify individuals into high and low-risk groups by analyzing personal movement time series and location trajectories. To overcome challenges related to independent crime behavior and information fusion loss, FIGAT introduces a multi-dimensional fusion Information Graph that combines semantic correlation features with traditional individual basic features, time features, and location features. Through a multi-relation graph attention layer, FIGAT leverages semantic relationships and node information to accurately categorize individuals into high and low-risk groups. The evaluation of FIGAT involves analyzing 14,625,884 GPS trajectories from 1038 individuals, collected by a real-world public safety department.

DISADVANTAGE :

Reliance on Personal Information and GPS Trajectories: FIGAT heavily relies on personal information and GPS trajectories for crime risk classification.

This dependence may limit its effectiveness if the data quality is compromised or if individuals provide inaccurate or incomplete information.

3.2 PROPOSED SYSTEM

This project introduces an innovative Java-based system designed to streamline the process of public objections in criminal cases, utilizing the robust SHA-256 algorithm of blockchain technology. The system is meticulously crafted to provide a secure and efficient platform for managing objections while ensuring data integrity and authenticity. At its core, the system features user-friendly login and registration pages, facilitating seamless access for users keen on submitting objections. These interfaces are crucial components in establishing a secure connection between the public and the criminal justice framework. Additionally, specialized modules are integrated to cater to the distinct needs of police headquarters and administrative functions, optimizing workflow efficiency within the system. One of the standout features of this project is its utilization of blockchain technology. By leveraging blockchain's decentralized and immutable nature, the system ensures the security of objection submissions. Each objection is securely recorded on the blockchain, making it tamper-proof and verifiable. This not only enhances the authenticity of objection records but also fosters transparency and accountability throughout the objection process. Furthermore, the system addresses the need for individuals affected by a case to access relevant files from the headquarters. Through a streamlined process, users can submit their mobile number and Aadhar number to retrieve case files. This initiates the generation of a QR code, providing a secure passkey for downloading the necessary report file. This feature enhances accessibility and empowers individuals to engage more actively in the legal process. In summary, this project represents a significant step forward in modernizing the objection process within criminal cases. By leveraging Java programming language and blockchain technology, the system provides a secure, transparent, and efficient platform for managing objections. With user-friendly

interfaces and specialized modules, coupled with the innovative use of blockchain, the system enhances trust, transparency, and accountability within the criminal justice framework, ultimately contributing to a more just and equitable legal system.

ADVANTAGE:

High Security: Utilizes blockchain and SHA-256 algorithm for robust security, protecting objection submissions from unauthorized access or tampering.

Data Integrity: Maintains the integrity of objection records through blockchain's immutable ledger, ensuring transparency and trustworthiness in the objection-handling process.

Efficient Workflow: Streamlines objection handling with specialized modules for police headquarters and admin functions, optimizing operational efficiency within the criminal justice framework.

Centralized Case File Access: Enables secure access to case files from police headquarters using a mobile number and Aadhar number, simplifying the retrieval process for further legal proceedings.

Tamper-Proof Records: Once recorded on the blockchain, objection submissions become tamper-proof, ensuring the preservation of data integrity and preventing unnoticed alterations or manipulations.

Enhanced Trust and Accountability: The transparent and accountable nature of the system fosters trust between the public and law enforcement authorities, promoting greater confidence in the objection-handling process and overall criminal justice framework.

3.3 MODULES:

MODULE NAMES:

- PUBLIC
- LOCAL POLICE
- OTHER POLICE STATION
- POLICE HEADQUARTERS
- PUBLIC COMPLAINT ABOUT CRIME
- LOCAL POLICE UPLOAD REPORT

3.4 MODULE DESCRIPTION:

1. **PUBLIC:**

In this application public doesn't need any register and login process. They are straightforwardly moving forward to the public home page and can use all features.

2. **LOCAL POLICE:**

Local police need to enter a username and secret word to open their landing page after that they can see all grumblings

3. **OTHER POLICE STATION**

Nearby police need to enter a username and secret words to open their greeting page after that they can see all grumblings and contrast with their data set assuming any case record matches that case they quickly send that case report to the neighborhood police station.

4. **POLICE HEADQUARTERS:**

Headquarters is one of the models of this application it has a secret username and secret phrase to open their point of arrival. It will keep the Case records securely.

5. PUBLIC COMPLAINT ABOUT CRIME:

Public wants to make a complaint about a crime they should fill out the complaint form. That complaint form contains a few input tags. the public has to enter personal details and crime details.

6. LOCAL POLICE UPLOAD REPORT:

Nearby police need to enter the username and secret phrase to open their point of arrival and afterward if get any data from different stations make the prompt move and report to the base camp about wrongdoing subtleties.

3.5 MODULE DIAGRAM:

1. PUBLIC:

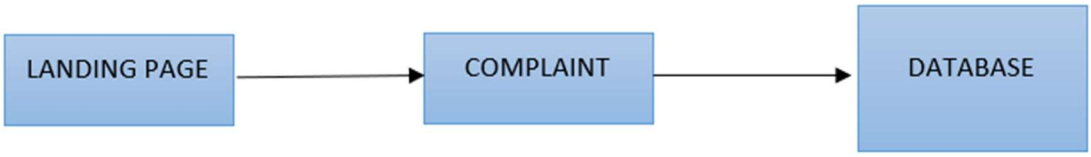


fig3.1 Public Module

2. LOCAL POLICE:

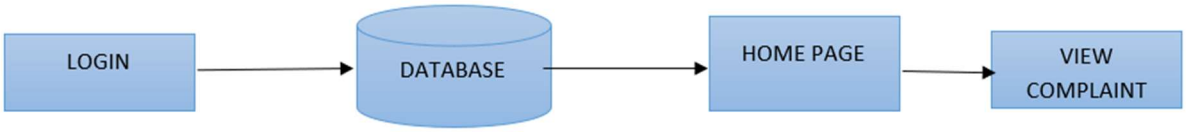


fig3.2 Local Police Module

3. OTHER POLICE STATIONS:

fig3.3 Other Police Station Module

4. POLICE HEADQUARTERS:

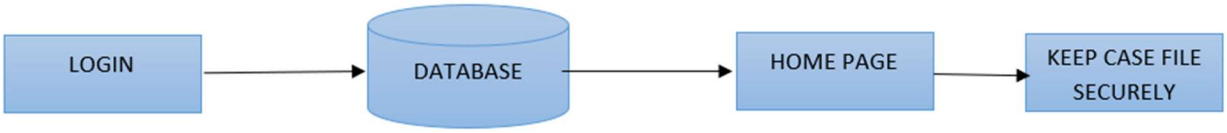


fig3.4 Police Headquarters Module

5. PUBLIC COMPLAINT ABOUT CRIME:

fig3.5 Public Complaint About Crime Module

6. LOCAL POLICE UPLOAD REPORT:

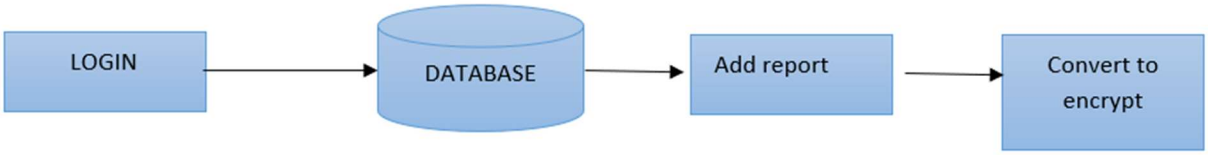


fig3.6 Local Police Upload Report Module

CHAPTER 4

REQUIREMENTS

REQUIREMENTS ENGINEERING

4.1 GENERAL:

These are the requirements for doing the project. Without using these tools and software we can't do the project. So we have two requirements to do the project. They are

1. Hardware Requirements.
2. Software Requirements.

4.2 HARDWARE REQUIREMENTS:

The hardware requirements outlined serve as foundational specifications for the implementation of the system. These requirements are essential components of a contract that outlines the parameters for developing the system. They provide a clear and comprehensive overview of the necessary hardware specifications that will support the system's functionality. For software engineers, these hardware requirements act as a starting point for designing the system. By understanding the capabilities and limitations of the hardware, engineers can make informed decisions about the design and development of the software components. This ensures that the system is optimized to perform efficiently on the specified hardware configuration.

PROCESSOR:	PENTIUM IV 2.6 GHz, Intel Core 2 Duo.
RAM:	4GB DD RAM
MONITOR:	15" COLOR
HARD DISK:	40 GB

4.3 SOFTWARE REQUIREMENTS:

The software requirements document serves as a comprehensive specification outlining the functionalities and features expected from the system. It encompasses both the definition and specification of requirements, detailing what the system should do rather than how it should be implemented. This document acts as a blueprint for the development process, providing a clear roadmap for software engineers to follow. In terms of content, the software requirements document typically includes a detailed description of the system's functional and non-functional requirements. Functional requirements outline the specific actions and behaviors the system must perform to meet user needs, while non-functional requirements specify criteria related to system performance, security, usability, and other quality attributes. The software requirements document is invaluable for various aspects of the development process The SRS provides a more detailed and formalized description of the system's functionality, serving as a reference for developers throughout the development lifecycle. Moreover, the software requirements document plays a crucial role in estimating costs, planning team activities, and tracking progress. By clearly defining the scope and objectives of the project, it enables stakeholders to make informed decisions regarding resource allocation and project management. Additionally, it facilitates effective communication among team members, ensuring that everyone is aligned with the project goals and objectives.

Front End	:	J2EE (JSP, SERVLETS) JAVASCRIPT
Back End	:	MY SQL 5.5
Operating System	:	Windows 07
IDE	:	Eclipse

CHAPTER 5

SYSTEM DESIGN

DESIGN ENGINEERING

5.1 GENERAL

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of the projects. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into a representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into a finished product.

5.2 ARCHITECTURE DIAGRAM

An architectural diagram serves as a visual representation of a software system's structure, illustrating the components, relationships, constraints, and boundaries within the system. It provides a high-level abstraction of the system's architecture, allowing stakeholders to gain a comprehensive understanding of its design and functionality.

Architectural diagrams are essential tools in software development as they offer insights into the physical deployment of the system and its evolution roadmap. They enable stakeholders to visualize the system's overall structure and identify potential areas for improvement or optimization.

The architectural diagram presented below illustrates the overarching design of the system implemented in the project. By depicting the various components and their interactions, it offers a holistic view of the system's architecture, guiding development efforts and facilitating effective communication among project stakeholders. The below architecture diagram depicts the overall system implemented in the project.

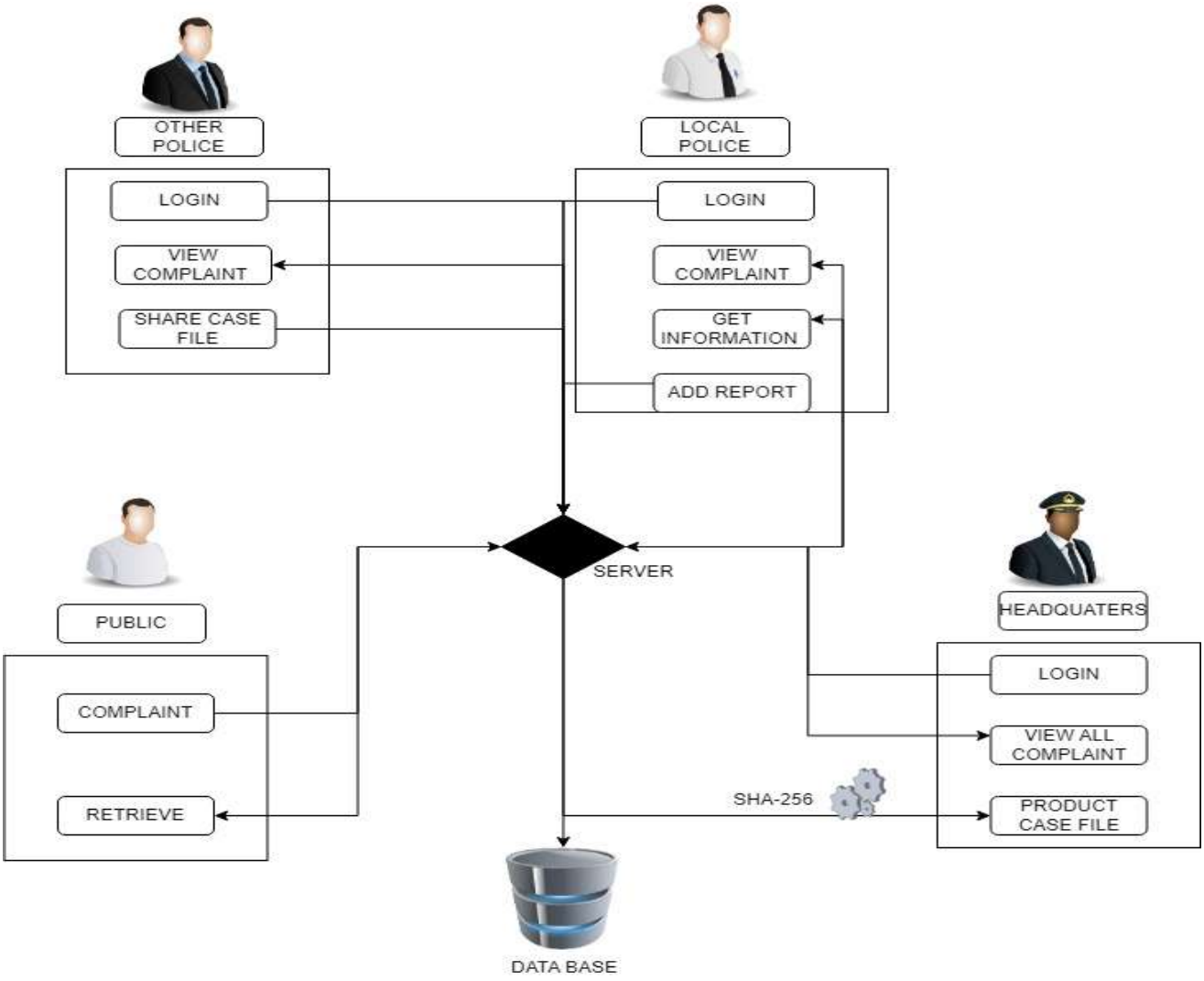


fig5.1 Architecture Diagram

EXPLANATION:

The systems architect establishes the basic structure of the system, we propose a Hash code Solomon algorithm, and we can put a small part of data in the local machine and fog server to protect privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machines, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is a powerful supplement to the existing cloud storage scheme

5.3 UML DIAGRAMS

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. If we look around, we will realize that diagrams are not a new concept but it is used widely in different forms in different industries. We prepare UML diagrams to understand the system in a better and simpler way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system. You can also create your own set of diagrams to meet your requirements. Diagrams are generally made incrementally and iteratively. There are two broad categories of diagrams and they are again divided into subcategories –

1)Structural Diagrams

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, that form the main structure and are therefore stable. These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are − Class diagram Object diagram Component diagram Deployment diagram

2) Behavioural Diagrams

Any system can have two aspects, static and dynamic. So, a model is considered complete when both aspects are fully covered. Behavioral diagrams capture the dynamic aspect of a system. The dynamic aspect can be further described as the changing/moving parts of a system. UML has the following five types of behavioral diagrams

5.3.1 USE-CASE DIAGRAM:

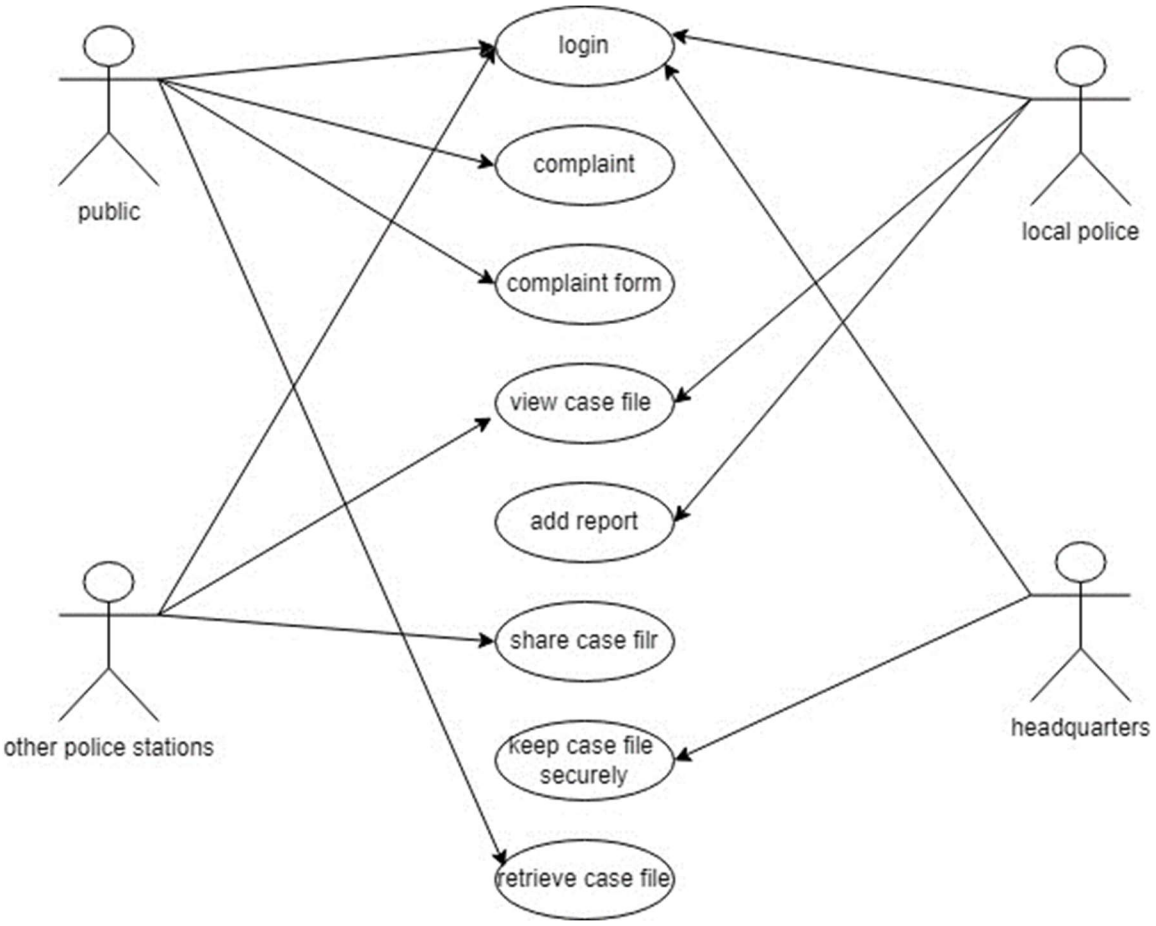


fig5.2 Use case Diagram

EXPLANATION:

The use case diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematic application and for detailed modeling translating the models into programming code. For this in our component diagram first propose a data In this proposed method we are using the Hash-Solomon Code Algorithm to encrypt the data.

5.3.2 STATE DIAGRAM:

fig5.3 State Diagram

EXPLANATION:

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. In our state diagram first, propose for in our component diagram propose data in this proposed method we are using the Hash-Solomon Code Algorithm to encrypt the data.

5.3.3 ACTIVITY DIAGRAM:

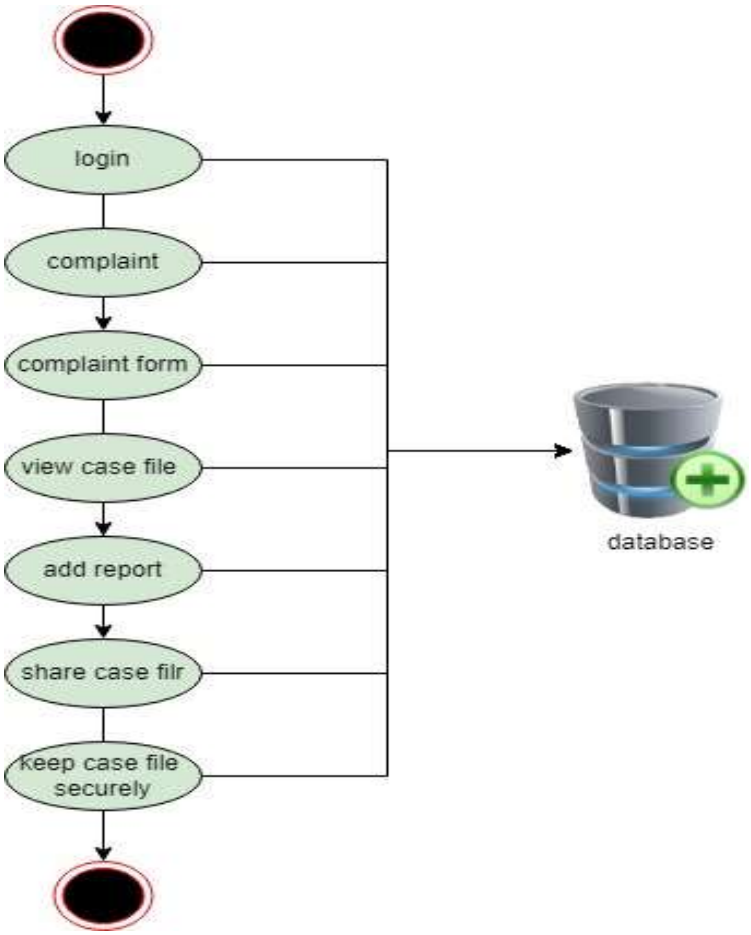


fig5.4 Activity Diagram

EXPLANATION:

Activity diagrams are loosely defined diagrams to show workflows of stepwise activities and actions, with support for choice, iteration, and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways, UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams(DFDs)from structural development.

5.3.4 CLASS DIAGRAM:

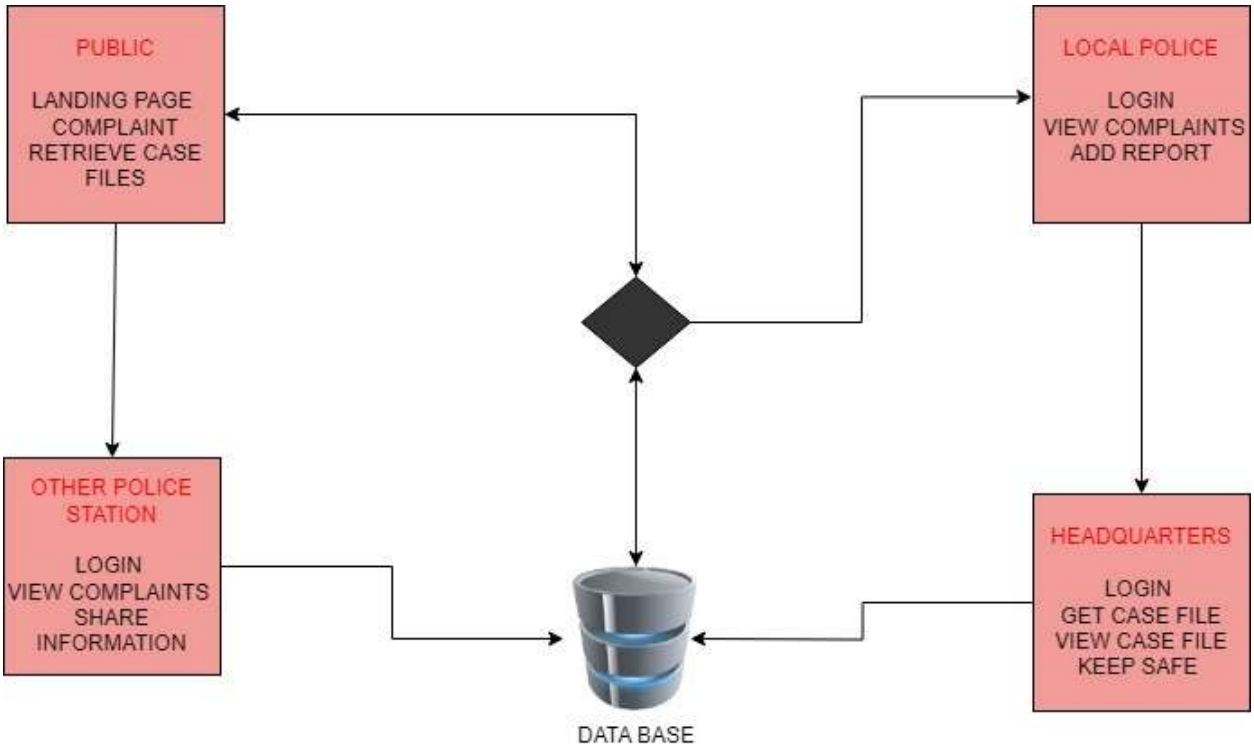


fig5.5 Class diagram

EXPLANATION:

A class diagram is a static structure diagram depicting a system's architecture by illustrating its classes, attributes, and relationships. These classes represent the main objects and interactions within the application, showcasing the system's structural elements. By visualizing the relationships between classes and their attributes, the class diagram provides a clear overview of the system's organization and functionality. It serves as a fundamental tool in software development, aiding developers in understanding the system's design and facilitating communication among team members. Overall, the class diagram is instrumental in documenting and analyzing the structure of a system, guiding the development process, and ensuring the coherence and efficiency of the final product.

5.3.5 SEQUENCE DIAGRAM:

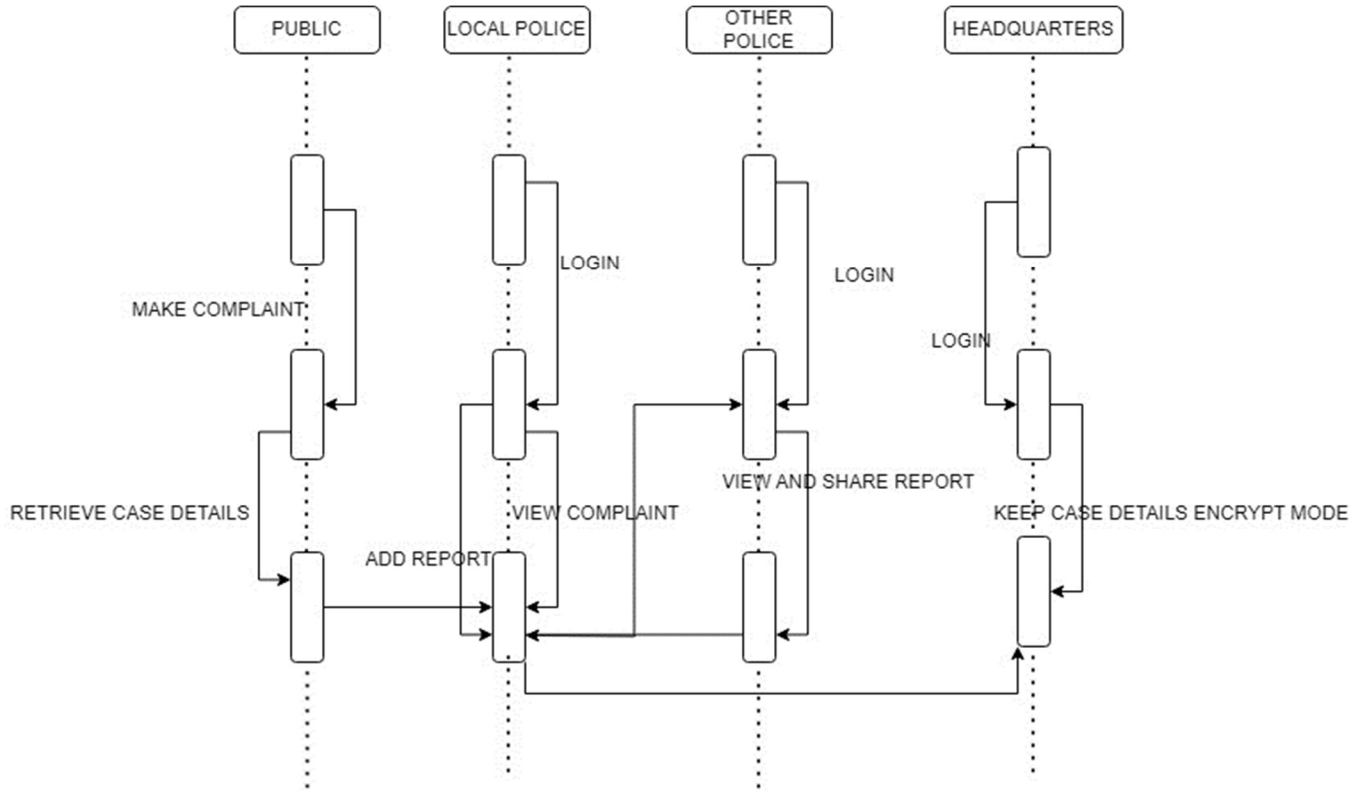


fig5.6 Sequence diagram

EXPLANATION:

In our sequence diagram, we delineate the sequential flow of processes and interactions within our system. Initially, the sequence diagram depicts the initiation of a proposal process. Following this, subsequent steps unfold sequentially, illustrating how various components and actors interact with one another to fulfill the proposed action. Within our component diagram, we propose a data component to encapsulate the storage and manipulation of information within the system. This data component serves as a central repository for storing crucial data elements and facilitating communication between different parts of the system. In implementing this proposed method, we leverage the Hash-Solomon Code Algorithm to encrypt the data stored within the system.

5.3.6 COLLABORATION DIAGRAM

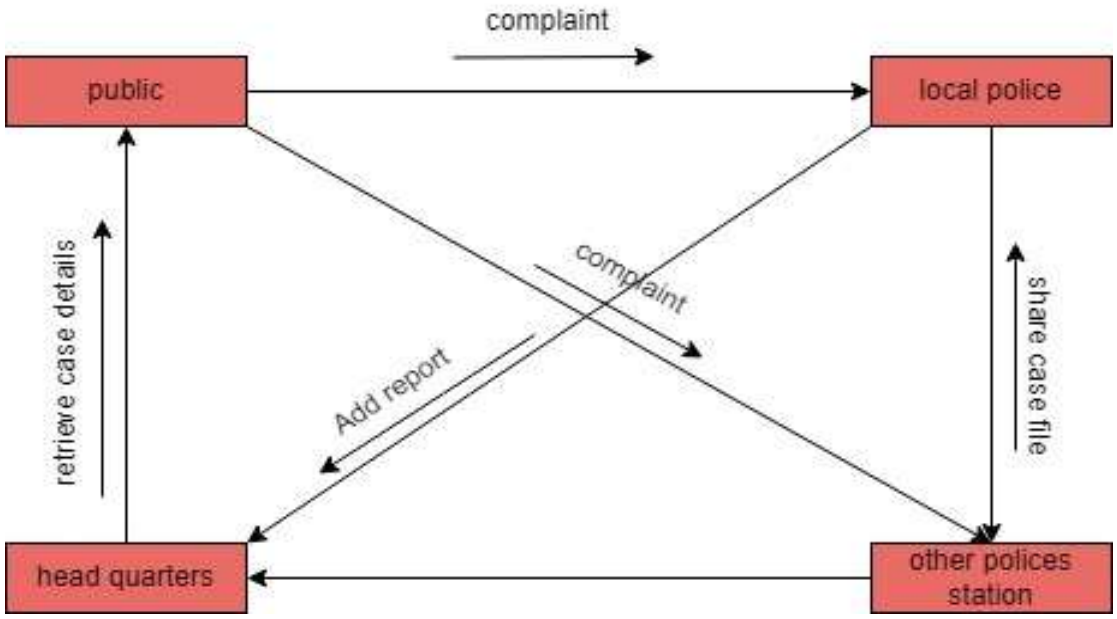


fig5.7 Collaboration diagram

EXPLANATION:

A collaboration diagram visually represents the objects and relationships engaged in an interaction, depicting the sequence of messages exchanged among these objects. It serves as a breakdown of a class or use case diagram, illustrating the flow of communication between classes or instances during an interaction. Each system operation pertinent to the current development cycle is depicted through a separate diagram within the collaboration diagram set. Through this graphical representation, developers gain insight into the dynamic behavior of the system, facilitating clear communication and understanding of the interaction flow. Collaboration diagrams play a vital role in the iterative development process, aiding in the decomposition of complex interactions and ensuring the seamless coordination of system components throughout the development lifecycle.

5.3.7 DATAFLOW DIAGRAM

A data flow diagram(DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

Level 1:

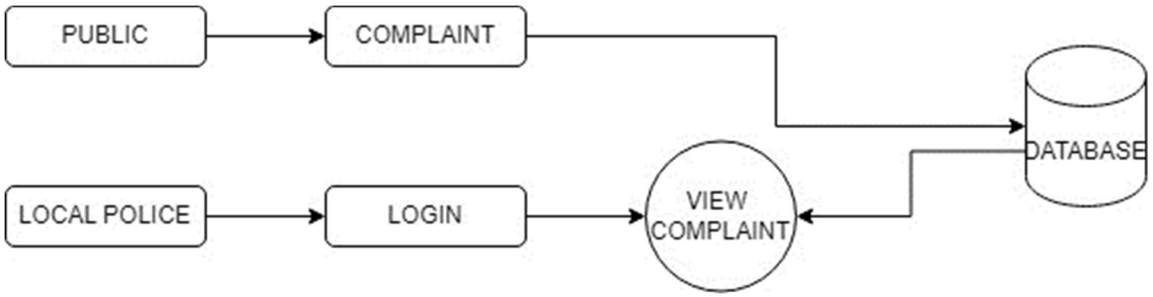


fig5.8 DFD Level 1

Level 2:

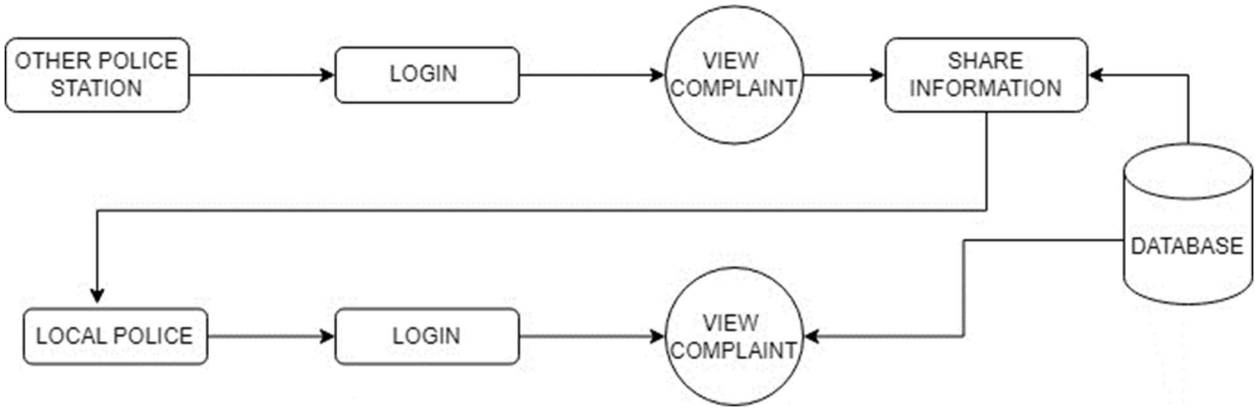


fig5.9 DFD Level 2

Level 3:

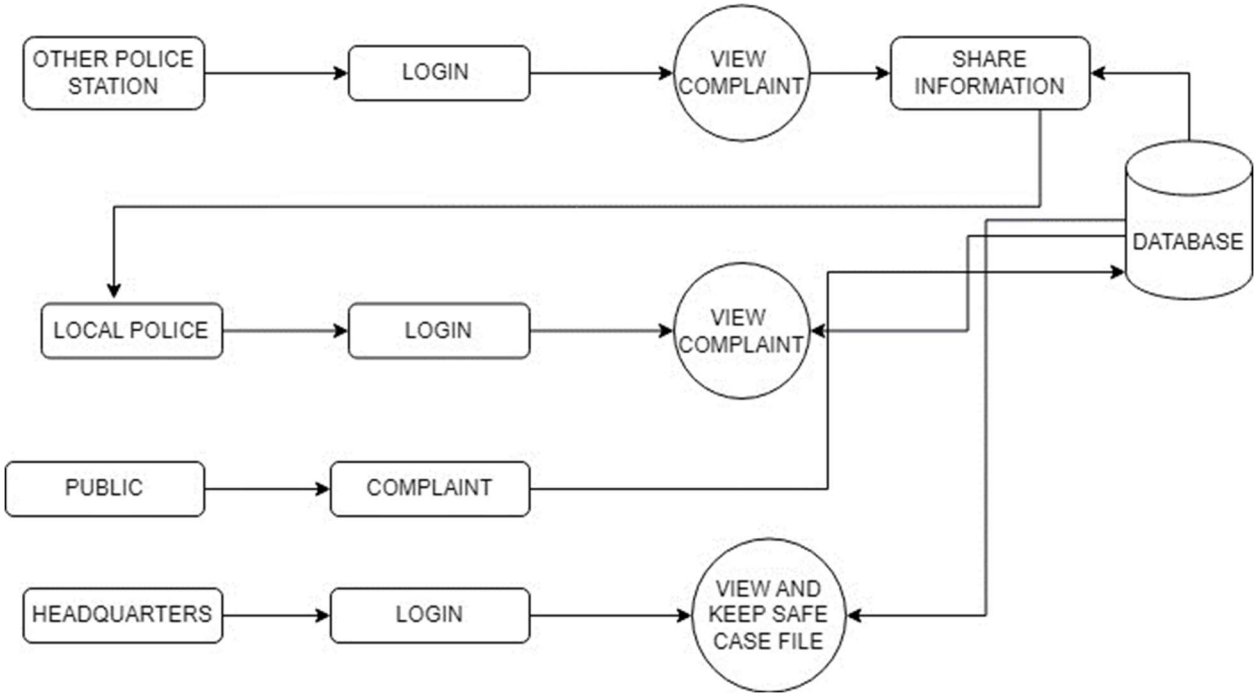


fig5.10 DFD Level 3

Level 1 DFD:

Often the first level below the context diagram, the Level 1 DFD provides a more detailed view of the system, showing major processes and data stores. It decomposes the system into manageable subsystems or processes, illustrating how data flows between them.

Subsequent Levels:

These lower-level DFDs offer increasingly granular insights into the system's data flow, helping developers and analysts understand the intricacies of data movement and processing.

5.3.8 ER-DIAGRAM:

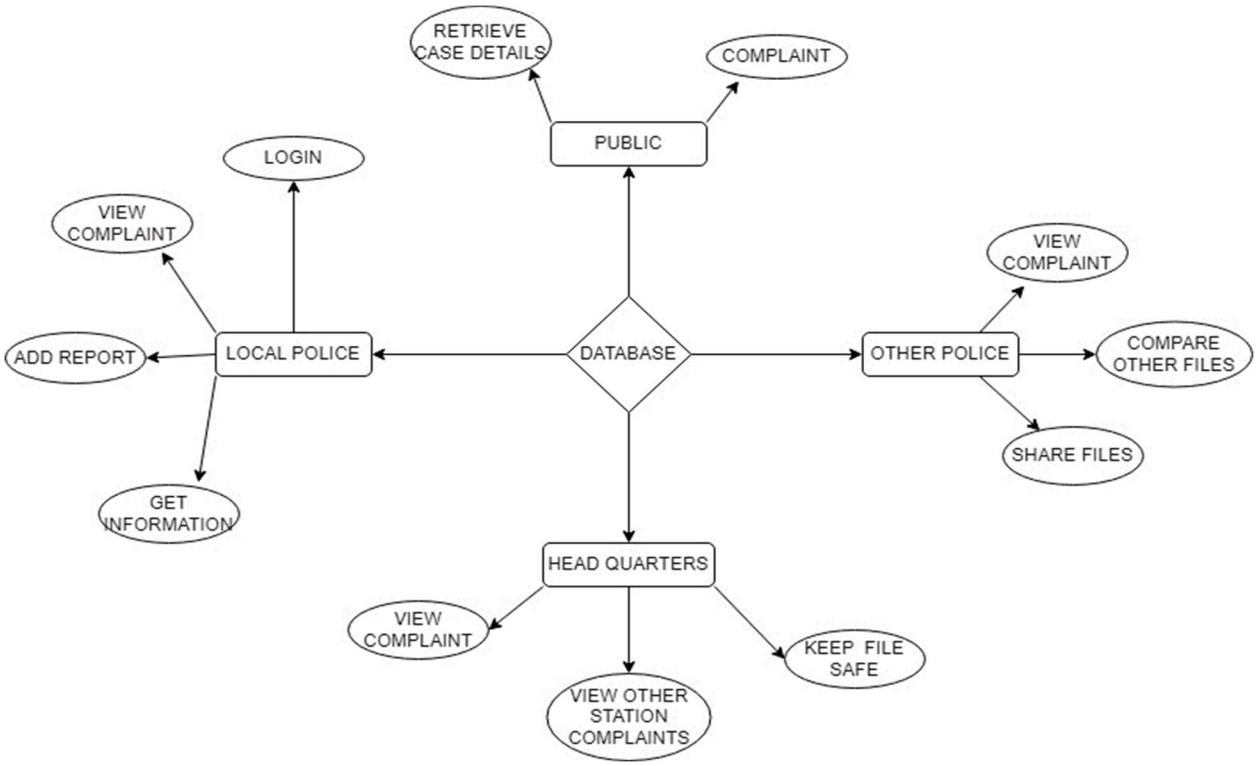


fig5.11 E-R Diagram

EXPLANATION:

In an ER diagram, entities are depicted as rectangles, each representing a distinct entity within the system. For instance, in the provided diagram, "Student" and "College" are entities. They exhibit a many-to-one relationship, signifying that multiple students can attend a single college. While relationships between entities will be explored further, the primary focus at this stage is to grasp the concept of entities and their representation in the diagram. This foundational understanding lays the groundwork for comprehending the complex relationships and interactions within the system.

CHAPTER 6

TECHNIQUES

6.1 TECHNIQUES

SHA-256

SHA-256, part of the SHA-2 (Secure Hash Algorithm 2) family, is a cryptographic hash function widely used in various security applications. It generates a 256-bit (32-byte) hash value, typically represented as a 64-character hexadecimal string, from input data of arbitrary size. This hash value serves as a unique digital fingerprint for the input data, facilitating secure data verification and integrity checking. SHA-256 operates by executing a series of mathematical operations on input data blocks, ultimately producing a fixed-length hash value that is computationally infeasible to reverse-engineer or reproduce. The primary purpose of SHA-256 is to ensure data integrity and authenticity. When a piece of data is hashed using SHA-256, the resulting hash value is unique to that specific input data. Even a minor alteration in the input data, such as a single-bit change, results in a significantly different hash value. This property enables users to verify the integrity of data by comparing its hash value before and after transmission or storage. If the hash values match, it indicates that the data has not been tampered with. Conversely, a mismatch indicates potential data corruption or unauthorized modification.

SHA-256 is also widely employed in digital signatures and cryptographic protocols to verify the authenticity of messages and documents. In digital signature schemes, a sender generates a hash value of the message using SHA-256 and then encrypts the hash value with their private key to create a digital signature. The recipient can then decrypt the digital signature using the sender's public key and compare the decrypted hash value with the hash value of the received message. If the two hash values match, it provides cryptographic proof of the message's origin and integrity, as only the sender possessing the private key could have generated the

signature. Furthermore, SHA-256 plays a crucial role in password hashing and authentication mechanisms. Instead of storing user passwords directly, systems often store their SHA-256 hash values. When a user attempts to log in, their entered password is hashed using SHA-256, and the resulting hash value is compared with the stored hash value. This approach enhances security by mitigating the risks associated with storing plaintext passwords and ensures that even if the stored hash values are compromised, the original passwords remain undisclosed.

One of the key strengths of SHA-256 lies in its cryptographic security properties. It is designed to be collision-resistant, meaning that it is highly unlikely for two different inputs to produce the same hash value. This property ensures the reliability of SHA-256 for various security applications, including digital signatures, data integrity verification, and password hashing.

Despite its robust security features, SHA-256 is not immune to certain attacks, such as brute-force attacks and collision attacks. As computing power advances, brute-force attacks become more feasible, potentially compromising the security of SHA-256 hash values. Additionally, although collision attacks against SHA-256 are computationally challenging, theoretical vulnerabilities exist, prompting ongoing research and development of more secure cryptographic algorithms. In addition to its widespread adoption in cryptographic applications, SHA-256 is renowned for its computational efficiency and speed. Its algorithmic design and optimized implementation enable SHA-256 to generate hash values quickly, making it suitable for real-time data processing and high-performance computing environments. This efficiency is particularly crucial in scenarios where rapid hashing of large volumes of data is required, such as in network security protocols, digital signatures, and blockchain technology.

Moreover, SHA-256's resistance to cryptographic attacks adds to its appeal as a secure hashing algorithm. While no cryptographic algorithm is entirely immune to attacks, SHA-256's robust design and extensive analysis make it highly resistant to known vulnerabilities. This resistance provides a strong foundation for building secure systems and protocols, instilling confidence in the integrity and confidentiality of data hashed using SHA-256. As a result, SHA-256 remains a cornerstone of modern cryptographic practices, serving as a critical component in securing digital assets and ensuring the trustworthiness of online transactions and communications. In conclusion, SHA-256 stands as a cornerstone of modern cryptographic protocols and security mechanisms, offering robust data integrity, authenticity verification, and password hashing capabilities. Its widespread adoption across diverse industries underscores its importance in safeguarding digital assets and ensuring secure communication and transactions in an increasingly interconnected world. However, ongoing advancements in computing technology necessitate continuous vigilance and innovation to mitigate emerging security threats and uphold the integrity of cryptographic standards. The SHA-256 algorithm involves several steps to generate a 256-bit hash value from input data. Here are the main steps involved in the SHA-256 hashing process:

Padding: The input message is padded to ensure its length is congruent to 448 modulo 512. Padding typically involves appending a single '1' bit followed by a sequence of '0' bits, with the last 64 bits representing the length of the original message in bits.

Message Parsing: The padded message is divided into blocks of 512 bits each. If the message length is not an exact multiple of 512 bits, additional padding is added to complete the last block.

Initialization: Initialize the hash values (also known as "IVs") with the first 32 bits of the fractional parts of the square roots of the first 8 prime numbers. These values serve as initial parameters for the hashing process.

Message Schedule: Create a message schedule array, expanding the 512-bit block into a 64-entry array of 32-bit words. This array is derived from the original message block and undergoes various transformations during the hashing process.

Compression: Perform a series of compression rounds, where the message schedule array undergoes a series of bitwise logical operations, including XOR, AND, OR, and bit shifting. Each compression round involves processing a chunk of the message schedule array using a specific set of constant values and bitwise operations.

Finalization: After completing all compression rounds for each message block, the final hash value is derived by concatenating the resulting hash values from each block. The resulting 256-bit hash value represents the unique digital fingerprint of the input message.

Output: The final hash value is typically represented as a 64-character hexadecimal string, providing a compact and standardized representation of the hashed data.

AES ALGORITHM

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm widely adopted for securing sensitive data in various applications. AES is considered one of the most secure and efficient encryption algorithms available today, offering robust protection against unauthorized access and data breaches. In

this comprehensive guide, we will explore the history, principles, operation, and applications of the AES algorithm.

History and Development:

The need for a new encryption standard arose in the late 1990s due to vulnerabilities found in existing encryption algorithms, such as the Data Encryption Standard (DES). To address these concerns, the National Institute of Standards and Technology (NIST) initiated a public competition in 1997 to select a new encryption algorithm. After extensive evaluation and analysis of candidate algorithms, AES was selected as the successor to DES in 2001. The selection process involved rigorous testing and scrutiny from cryptographic experts worldwide, ensuring that AES met stringent security criteria.

Principles of AES:

AES is a symmetric encryption algorithm, meaning the same key is used for both encryption and decryption. It operates on fixed-size blocks of data, typically 128 bits in length, and supports key sizes of 128, 192, or 256 bits. The AES algorithm consists of several rounds of substitution, permutation, and mixing operations, known as the SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations. These transformations ensure that even small changes in the input data or key produce significant changes in the output, enhancing the algorithm's resistance to cryptographic attacks.

Operation of AES:

The AES encryption process begins with key expansion, where the original encryption key undergoes a series of transformations to generate round keys for each

encryption round. The input data is then divided into blocks and processed through multiple rounds of encryption. Each round consists of four main operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey. These operations manipulate the data and key to produce a ciphertext block, which is the encrypted form of the input data.

Key Features and Strengths:

AES offers several key features and strengths that contribute to its widespread adoption and security:

Strong Security: AES is resistant to known cryptographic attacks, including brute-force, differential, and linear cryptanalysis. Its design and implementation have undergone extensive scrutiny and analysis by cryptographic experts worldwide, ensuring its security against a wide range of threats.

Efficiency: AES is highly efficient in terms of computational complexity and resource utilization. It can encrypt and decrypt data quickly, making it suitable for real-time applications and systems with stringent performance requirements.

Versatility: AES supports multiple key sizes, allowing users to choose the appropriate level of security based on their specific requirements. It can be implemented in software, hardware, and embedded systems, making it adaptable to a wide range of platforms and devices.

Standardization: AES has been standardized by organizations such as NIST and the International Organization for Standardization (ISO), providing a globally recognized and interoperable encryption solution. Its widespread adoption and standardization promote compatibility and ease of integration in various applications and industries.

Applications of AES:

AES is utilized in numerous applications across various industries to secure sensitive data and communications. Some common applications of AES include:

Data Encryption: AES is used to encrypt sensitive data, such as financial transactions, personal information, and confidential documents, to prevent unauthorized access and ensure data privacy.

Network Security: AES is employed in secure communication protocols, such as SSL/TLS, IPsec, and SSH, to encrypt data transmitted over networks and protect against eavesdropping and data interception.

File and Disk Encryption: AES is used to encrypt files, folders, and disk volumes to prevent unauthorized access and safeguard sensitive information stored on computers, servers, and storage devices.

Cryptocurrency and Blockchain: AES is utilized in cryptocurrency wallets and blockchain networks to encrypt private keys, transaction data, and digital signatures, ensuring the security and integrity of decentralized financial transactions.

Furthermore, AES offers a high level of versatility and adaptability, making it suitable for a wide range of applications and environments. It can be implemented efficiently in both software and hardware, catering to diverse computing platforms and devices Moreover, AES has undergone extensive scrutiny and analysis by cryptographic experts worldwide, ensuring its security and resilience against known cryptographic attacks. Its selection as the standard encryption algorithm by reputable organizations such as the National Institute of Standards and Technology (NIST) and the International Organization for Standardization (ISO) further attests to its reliability and trustworthiness. As a result, AES has become the de facto encryption standard adopted by governments, businesses, and individuals globally, providing a

robust foundation for securing digital assets and ensuring the confidentiality, integrity, and authenticity of sensitive information.

In summary, the Advanced Encryption Standard (AES) is a symmetric encryption algorithm renowned for its strong security, efficiency, versatility, and widespread adoption. Its robust design and implementation make it a cornerstone of modern cryptographic practices, serving as a vital tool for securing sensitive data, communications, and transactions in various industries and applications.

The Advanced Encryption Standard (AES) algorithm operates through several distinct steps to encrypt and decrypt data securely. Here's an overview of the main steps involved in the AES encryption and decryption processes:

AES Encryption Steps:

Key Expansion:

The encryption key undergoes a key expansion process to generate a set of round keys.

Each round key is derived from the original encryption key and is used in the subsequent encryption rounds.

Initial Round:

The plaintext data is divided into fixed-size blocks, typically 128 bits (16 bytes) each. The initial round of encryption involves adding the round key to the plaintext block.

Rounds:

AES operates through a series of encryption rounds, with the number of rounds determined by the key size.

Each round consists of four main operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey.

In the SubBytes operation, each byte of the plaintext block is replaced with a corresponding byte from a substitution table (S-box).

The ShiftRows operation cyclically shifts the rows of the plaintext block to create a diffusion effect.

MixColumns operates on the columns of the block, using matrix multiplication to further diffuse the data.

Finally, AddRoundKey XORs the block with the round key derived from the expanded key.

Final Round:

The final encryption round excludes the MixColumns operation to simplify the process.

It consists of SubBytes, ShiftRows, and AddRoundKey operations using the final round key.

Output:

After completing all encryption rounds, the resulting ciphertext block represents the encrypted form of the input plaintext.

AES Decryption Steps:

Key Expansion:

The decryption process begins with the expansion of the original encryption key to generate the round keys used in decryption.

Initial Round:

The decryption process starts with an initial round, similar to encryption, where the round key is added to the ciphertext block.

Rounds:

AES decryption proceeds through a series of rounds, mirroring the encryption rounds but in reverse order.

Each round involves inverse versions of the encryption operations: InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey.

Final Round:

The final decryption round excludes the InvMixColumns operation and includes only InvSubBytes, InvShiftRows, and AddRoundKey.

Output:

Upon completing all decryption rounds, the resulting plaintext block represents the original input data.

By following these steps, AES ensures secure encryption and decryption of data while maintaining confidentiality and integrity. Its robust design and efficient operation make it a widely adopted encryption standard in various industries and applications.

CHAPTER 7

IMPLEMENTATION

7.1 GENERAL

This chapter describes the implementation of a search-based application. It deals with the source code for the main viewpoint of Anonymous Database Management.

7.2 IMPLEMENTATION

Web Content:

User main.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

<style>

*,

*::before,

*::after {

  box-sizing: border-box;

  padding: 0;

  margin: 0;}

body {
```

```
  background: url(https://images.unsplash.com/photo-1578526932643-
e04af68c3e8b?q=80&w=1470&auto=format&fit=crop&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D);

  height: 100vh;

  background-size: cover;

  background-position: center;

  background-repeat: no-repeat;

   font-size: 30px;}

header {

  position: relative;

  background: hsla(0, 0%, 0%, .6);

  padding: 1rem 0;

  z-index: 999;}

.header-container {

  display: flex;

  align-items: center;

  justify-content: space-between;

  width: min(90%, 800px);

  margin-inline: auto;

  color: #fff;}

nav ul {

  display: flex;

  list-style: none;

  gap: 2rem;}

nav a {
```

```
  text-decoration: none;
  color: #fff;}
nav a:hover {
  color: #4FC3F7;}
.hamburger {
  display: none;
  cursor: pointer;}
@media (max-width: 600px) {
  .toggle {
    transition: ease-in-out 550ms;
    transform: translate(0px);
    opacity: 1;
    display: block;  }
  .header-container {
    width: 100%;
    padding: 0 1rem;}
  nav ul {
    flex-direction: column;
    position: absolute;
    top: 100%;
    left: 0;
    width: 100%;
    background: hsla(0, 0%, 0%, .5);
    backdrop-filter: blur(10px);
```

```
    transform: translateX(-500px);
    opacity: 0;  }
  nav li {
    padding: 1rem;
    cursor: pointer;}
  nav li:hover {
    background: hsla(0, 0%, 0%, .7); }
  .hamburger {
    display: block; }}
</style>
</head>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css"
integrity="sha512-
z3gLpd7yknf1YoNbCzqRKc4qyor8gaKU1qmn+CShxbuBusANI9QpRohGBreCFkKxLhei6S9CQXFEbbKuqLg0D
A==" crossorigin="anonymous" referrerpolicy="no-referrer" />
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/brands.min.css"
integrity="sha512-
W/zrbCncQnky/EzL+/AYwTtosvrM+YG/V6piQLSe2HuKS6cmbw89kjYkp3tWFn1dkWV7L1ruvJyKbLz73Vlgfg=
=" crossorigin="anonymous" referrerpolicy="no-referrer" />
<body>
  <header>
    <div class="header-container">
      <span>Public</span>
      <nav id="nav">
        <ul id="nav-ul">
          <li><a href="">Home</a></li>
          <li><a href="complaintpage.jsp">Complaint</a></li>
```

```
<!--          <li><a href="casestatus.jsp">Reports</a></li>
 -->
       </ul>
     </nav>
     <span class="hamburger" id="button"><i class="fa-solid fa-bars"></i></span>
   </div>
  </header>
<script>
const btn = document.getElementById('button');
const nav = document.getElementById('nav-ul');
btn.addEventListener('click', function() {
  nav.classList.toggle ('toggle');
});
</script>
</body>
</html>
```

Complaintpage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>\
```

```
<style>*{
        padding: 0;
        margin: 0;
        border: 0;}
body, html {
        color: #373C40;
        font-family: Verdana, Arial, Helvetica, sans-serif;
        height: 100%;
        background-color: #f0f0f0;
        margin: 10px;}
body {
        font-size: 70%;}
label {
        padding: 7px 0 7px 0;
        font-weight: 500;
        font-size: 10pt;}
h1 {
        font-weight: 200;
        color: #888888;
        font-size: 17pt;
        text-align: center;}
form.register {
        margin: 20px auto 0px auto;
        background-color: #fff;
```

```
        padding: 20px;
        -moz-border-radius: 20px;
        -webkit-border-radius: 20px;}
.form-group {
        font-size: 8pt;
        margin: 0;
        color: gray;
        padding: 7px 0 7px 0;
        font-weight: 500;
        font-size: 10pt;}
form.register legend {
   color: #abda0f;
   padding:2px;
   margin-left: 10px;
   font-weight:bold;
   font-size: 14px;
   font-weight:100;
   margin-bottom: 5px;}
form.register input {
        border: 1px solid black;
        padding:4px 20px;}
form.register input[type=radio] {
        margin-left: 7px;}
form.register select {
```

```
    border: 1px solid #E1E1E1;
    width: 130px;
    margin-bottom:3px;
    color: #505050;
    margin-right:5px;}
form.register select.date {
    width: 40px;}
input:focus, select:focus {
    background-color: #efffe0;}
div.agreement{
    margin-left:15px;}
div.agreement label{
    text-align:left;
    margin-top:3px;}
#registerButton {
    background: #abda0f;
    padding: 8px 10px 8px;
    color: #fff;
    text-decoration: none;
    -moz-border-radius: 5px;
    -webkit-border-radius: 5px;
    -moz-box-shadow: 0 1px 3px rgba(0,0,0,0.5);
    -webkit-box-shadow: 0 1px 3px rgba(0,0,0,0.5);
    text-shadow: 0 -1px 1px rgba(0,0,0,0.25);
```

```
    cursor: pointer;
    float:left;
    font-size:18px;
    margin:10px;}
#updatedProfile {
        display: none;
        margin: 20px auto 0px auto;
        background-color: #fff;
        padding: 20px;
        -moz-border-radius: 20px;
        -webkit-border-radius: 20px;}
#updatedProfile h2 {
        color: #abda0f;
    padding:2px;
    margin-left: 10px;
    font-weight:bold;
    font-size: 14px;
    font-weight:100;
    margin-bottom: 5px;}
#updatedProfile h3 {
        color: #abda0f;
    padding:2px;
    margin-left: 10px;
    font-weight:bold;
```

```
    font-size: 12px;
    font-weight:100;
    margin-bottom: 5px;}
body{
background-color: orange;}
button{
background-color:green;
padding:6px 15px;
color:white;
border-radius:6px;}
</style>
</head>
<body>
<div class="container">
                    <div id="newUser" class="col-sm-6 col-sm-offset-3">
                        <!-- New User Profile -->
                        <form name="newUser" class="register" method="post" action="complaintdata">
                            <h1>Complaint Form</h1>
                            <fieldset class="row1">
                               <legend>Personal Details</legend>
                               <!-- EMAIL -->
                               <div class="form-group">
                                    <label>Name *</label>
                                    <input type="text" name="name" class="form-control">
```

```
    <label>Address *</label>
        <input type="text" name="address" class="form-control">
  </div>
  <!-- PASSWORD -->
  <div class="form-group">
        <label>Email*</label>
        <input type="email" name="email" class="form-control">
    <label>Phone No *</label>
        <input type="number" name="pnumber" class="form-control">
  </div>
</fieldset>
<fieldset class="row2">
        <legend>Crime Details</legend>
        <!-- NAME -->
        <div class="form-group">
                <label>Criminal Name *</label>
                <input type="text" name="cname" class="form-control" >
                        <br><br>
                <label>Crime Street *</label>
        <input type="text" name="street" class="form-control" required>
        </div>
        <!-- PHONE NUMBER -->
        <div class="form-group">
                <label>Pin Crime Location *</label>
```

```
                    <input type="text" name="location" class="form-control" required >

                </div>

<br><br>

                <!-- ADDRESS -->

                    <div class="form-group">

                        <label>Date *</label>

                        <input type="date" name="date" class="form-control">

                        <label>Time *</label>
        <input type="time" name="time" class="form-control"><br><br>

                                <label>Description</label>

                                <textarea name="des"></textarea>


                        <!-- <label>Country *</label>

                        <select name="country">

                            <option>

                            </option>

                            <option value='United State'>United State</option>

                        </select> -->

                    </div>

                </fieldset>

                <br><br>

                    <div>

                        <label>Crime Type *</label>
            <!-- <input type="date" name="city" class="form-control"> -->

        <select name="ctype" id="cars">
```

```
  <option value="Murder">Murder</option>
  <option value="Domestic Violence">Domestic Violence</option>
  <option value="Fraud">Fraud</option>
  <option value="Illegal drug trade">Illegal drug trade</option>
</select>
                                        </div>
<br><br>
                                    <fieldset class="row3">
                                        <!-- GENDER -->
                                        <div class="form-group">
                                            <label>Gender :
                                        <input type="radio" name="Gender" value="male"> Male
                                            <input type="radio" name="Gender" value="female"> Female
                                            </label>
                                            <br>
                                        </div>
                                            </fieldset>
                                            <br>
                                            <br>
                                        <!-- BIRTHDAY -->
                                        <!-- <div class="form-group">
                                            <label>Birthdate :</label>
                                            <select name="date">
                                               <option value="1">01
```

```
</option>
<option value="2">02
</option>
<option value="3">03
</option>
<option value="4">04
</option>
<option value="5">05
</option>
<option value="6">06
</option>
<option value="7">07
</option>
<option value="8">08
</option>
<option value="9">09
</option>
<option value="10">10
</option>
<option value="11">11
</option>
<option value="12">12
</option>
<option value="13">13
```

```
</option>
<option value="14">14
</option>
<option value="15">15
</option>
<option value="16">16
</option>
<option value="17">17
</option>
<option value="18">18
</option>
<option value="19">19
</option>
<option value="20">20
</option>
<option value="21">21
</option>
<option value="22">22
</option>
<option value="23">23
</option>
<option value="24">24
</option>
<option value="25">25
```

```
    </option>
    <option value="26">26
    </option>
    <option value="27">27
    </option>
    <option value="28">28
    </option>
    <option value="29">29
    </option>
    <option value="30">30
    </option>
    <option value="31">31
    </option>
</select>
<select name="month">
    <option value="1">January
    </option>
    <option value="2">February
    </option>
    <option value="3">March
    </option>
    <option value="4">April
    </option>
    <option value="5">May
```

```
                                        </option>
                                        <option value="6">June
                                        </option>
                                        <option value="7">July
                                        </option>
                                        <option value="8">August
                                        </option>
                                        <option value="9">September
                                        </option>
                                        <option value="10">October
                                        </option>
                                        <option value="11">November
                                        </option>
                                        <option value="12">December
                                        </option>
                                    </select>
                                <input name="year" type="text" size="4" maxlength="4"/>e.g 1986
                                </div>
                        </fieldset> -->
            <!--        <fieldset class="row4">
<legend>Terms and Mailing</legend>
 <div class="agreement">
    <input type="checkbox" name="agreement">
    <label>*  I accept the <a href="#">Terms and Conditions</a></label>
```

```
                    </div>

                    <div class="agreement">

                        <input type="checkbox">

                        <label>I want to receive personalized offers by your site</label>

                    </div>

                    <div class="agreement">

                        <input type="checkbox">

                        <label>Allow partners to send me personalized offers and related services</label>

                    </div>

                </fieldset> -->

                <div>

                  <button>Submit</button>

                </div>

              </form>

            </div> <!-- New User -->

        <script>

        // get users data

        function userRegistration(form) {

                // if any form field isn't filled don't go on

                if(!form.userEmail.value || !form.userPassword.value || !form.userFirst.value ||
!form.userLast.value || !form.userPhone.value || !form.street.value || !form.city.value || !form.country.value ||
!form.userGender.value || !form.date.value || !form.month.value || !form.year.value) {

                        if(form.userEmail.value !== form.userRepeatEmail.value || form.userPassword.value !==
form.userRepeatPassword.value) {

                            // VALIDATE EMAIL ADDRESS

                            var emailInput = form.userEmail.value
```

```
                var regEx = /^[a-z0-9._%-]+@[a-z0-9.-]+\.[a-z]{2,5}$/
                var testing = (regEx.test(emailInput))
                if( testing !== true) {
                        alert('Please entered a validated email')
                        form.userEmail.focus();
                        return false;}
                // ACCEPT TERMS
                if(!form.agreement.checked) {
                        alert("Please indicate that you accept the Terms and Conditions")
                        form.agreement.focus();
                        return false;}
                alert('Sorry your Email or Password is unmatched')
                form.userEmail.focus();
                form.userPassword.focus();
                return false;}
        alert('Missing Data')
        return}
// store all form data in an object, selecting by elements
var registeredUser = {
        email : form.userEmail.value,
        firstname : form.userFirst.value,
        lastname : form.userLast.value,
        phone : form.userPhone.value,
        street : form.street.value,
```

```
                city : form.city.value,
                country : form.country.value,
                gender : form.userGender.value,
                date : form.date.value,
                month : form.month.value,
                year : form.year.value,}
            // passing registered user info
            displayUser(registeredUser)}
        // display User Information
        function displayUser(registeredUser) {
            console.log(registeredUser)
            // selecting elements
            var userEmail = document.getElementById('userEmail'),
                userName = document.getElementById('userName'),
                userPhone = document.getElementById('userPhone'),
                userAddress = document.getElementById('userAddress'),
                userGender = document.getElementById('userGender'),
                userBirthday = document.getElementById('userBirthday')
            // add text into elements
            userEmail.innerText = "Email: " + registeredUser.email
            userName.innerText = "Name: " + registeredUser.firstname + " " + registeredUser.lastname
            userPhone.innerText = "Phone: " + registeredUser.phone
            userAddress.innerText = "Address: " + registeredUser.street + " " + registeredUser.city + " " +
registeredUser.country
            userGender.innerText = "Gender: " + registeredUser.gender
```

```
                userBirthday.innerText = "Birthday: " + registeredUser.month + " " + registeredUser.date + " " +
registeredUser.year

                // hide registration form

                document.getElementById('newUser').style.display = "none"

                // display new register form

                document.getElementById('updatedProfile').style.display = "block"}

        // add click function to button

        document.getElementById('registerButton').addEventListener('click', function() {

                userRegistration(document.newUser)

        }, false)

        </script>

</body>

</html>
```

Reportform.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

   pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

<style>

.container {

   width:600px;
```

```
    margin:34px auto -34px;
    border: 3px solid #E9BF7B ;}
.head .title h1{
    text-align: center;
    font-weight: bold;
    font-size: 43px;
    font-family: Arial, Helvetica, sans-serif;
    padding:16px 0 0 0;
    color:white;}
.head .title{
    background-color: #E9BF7B;
    height:83px;
    margin: -29px 0 0 0;}
form .first .clear\{
    clear:both;}
.first,.last,.email,.pass,.cpass{
    margin:12px 0 13px 20px;
    padding:12px 0 0 0;}
 .first label,
.last label,
.email label,
.pass label,
.cpass label{
    font-weight:bold;
```

```
    font-family: Arial, Helvetica, sans-serif;
    color:#E9BF7B;}
.first input,
.last input,
.email input,
.pass input,
.cpass input{
    width:90%;
    height:33px;
    margin: 11px 0 3px 0;
    padding: 0 0 0 10px;
    background-color: #D8D8D8;
    border: none;}
.submit{
    margin:0 0 0 19px;}
.submit input[type="submit"]{
    background-color:#E9BF7B;
    color:white;
    font-weight: bold;
    margin:0 0 20px 0;
    font-family: Arial, Helvetica, sans-serif;
    border-radius: 25px;
    font-size: 20px;
    width:160px;
```

```
    height:33px;
    text-align:center;
    padding:4px 10px 2px 10px;}
 button{
    background-color:#FFE4B5;
    color:black;
    padding: 5px 15px;
    cursor:pointer;
    border-radius:10px;}
button:hover{
background-color:red;}
form{
    margin:-12px 0 0 0px;
    background-color: white;}
</style>
</head>
<body>
<%String id=session.getAttribute("idd").toString();
String image=request.getParameter("image");
 %>
        <div class="head">
            <div class="container">
                <div class="title">
                    <h1>Report Form!</h1>
```

```
        </div>
    </div>
</div>
<div class="content">
    <div class="container">
            <form action="Reportform" method="post" enctype="multipart/form-data">
            <input type="hidden" name="cid" value="<%=id%>">
            <%-- <%String id=request.getParameter("id"); %>
             <input type="text" name="id" value="<%=id%>"> --%>
                <div class="first">
                    <label for="first">Name:</label>
                    <div class="clear"></div>
                    <input type="text" name="name" id="first" placeholder="enter name" required>
                </div>
                <div class="last">
                    <label for="last">Crime</label>
                    <div class="clear"></div>
                    <input type="text" name="crime" id="last" placeholder="enter crime name" required>
                </div>
                <div class="email">
                    <label for="email">Crime Street</label>
                    <div class="clear"></div>
                    <input type="text" name="street" id="email" placeholder="enter street" required>
                </div>
```

```
                <div class="pass">
                   <label for="pass">Criminal Age</label>
                   <div class="clear"></div>
                   <input type="number" name="age" id="pass" placeholder="enter age" required>
                </div>
                <input type="hidden" name="image" value="<%=image%>">
                 <div class="pass">
                   <label for="pass">Complainer Email</label>
                   <div class="clear"></div>
                   <input type="email" name="cemail"  placeholder="Enter complainer Email" required>
                </div>
                <div class="cpass">
                   <label for="Cpass">Case File</label>
                   <div class="clear"></div>
                   <input type="file" name="file"  placeholder="enter again password" required>
                </div>
                <div class="submit">
                   <button>Submit</button>
                </div>
             </form>
          </div>
       </div>
<button><a href="localcomplaintviwe1.jsp">back</a></button>
</body>
```

```
</html>
```

Requestpage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>

<%@ page import="dbcon.dbcon" %>

<%@ page import="java.sql.PreparedStatement"%>

<%@ page import="java.sql.*"%>

<%@ page import="javax.swing.*"%>

<%@ page import="Servlet.mail1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

   <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

   <title>Insert title here</title>

   <style>

      body {

         display: flex;

         align-items: center;

         justify-content: center;

         height: 100vh;

         margin: 0;}

      fieldset {

         border: 3px solid #ccc;

         border-radius: 5px;

         padding: 20px;
```

```
            text-align: center;}
        #qrcode-container {
            text-align: center;}
        button{
        padding:10px 20px;}
        a{
        text-decoration: none;}
    </style>
    <script src="https://cdn.rawgit.com/davidshimjs/qrcodejs/gh-pages/qrcode.min.js"></script>
</head>
<body>
<%
String id=request.getParameter("id");
System.out.println("id: "+id);
try {
    String status="request";
    Connection con1=dbcon.create();
    java.sql.Statement st=con1.createStatement();
    st.executeUpdate("UPDATE investication.reportform r SET status='"+status+"' where id='"+id+"' ");}
catch(Exception e) {
    response.sendRedirect("error.jsp?inval id");
    System.out.println(e);}
int b=0;
JFrame f;
```

```
f=new JFrame();
String email=JOptionPane.showInputDialog(f,"Enter Email");
String adhar=JOptionPane.showInputDialog(f,"Enter Adhar No");
System.out.println("adhar  "+adhar);
System.out.println("email  "+email);
if(adhar.isEmpty()) {
   f=new JFrame();
   JOptionPane.showMessageDialog(f,"Invalid Adhar Number");   }
else {
   int min = 22300;
   int max = 43200;
   b = (int)(Math.random()*(max-min+1)+min);
   System.out.println("bb  "+b         );
%>
<fieldset>
   <legend>Scan the QR Code</legend>
   <div id="qrcode-container">
      <div id="qrcode"></div>
   </div>
   <br><br>
   <button><a href="finalreport.jsp">Go</a></button>
</fieldset>
<%-- <script>
   // Generate a QR code with QRCode.js
```

```
    var bValue = "<%= b %>"; // Use the dynamic value for the QR code content
    var qrcode = new QRCode(document.getElementById("qrcode"), {
        text: bValue,
        width: 228,
        height: 228 });
</script> --%>
<script>
    // Generate a QR code with QRCode.js
    var bValue = "<%= b %>"; // Use the dynamic value for the QR code content
    var qrcode = new QRCode(document.getElementById("qrcode"), {
        text: bValue,
        width: 228,
        height: 228    });
    // Remove any tooltip or title attributes
    document.getElementById("qrcode").removeAttribute("title");
</script>
<%}
session.setAttribute("session", id);
try {
    Connection con1=dbcon.create();
    java.sql.Statement st=con1.createStatement();
    st.executeUpdate("UPDATE investication.reportform r SET filekey = '"+b+"', adhar = '"+adhar+"' WHERE id =
'"+id+"'");}
catch(Exception e) {
    response.sendRedirect("error.jsp?inval id");
```

```
    System.out.println(e);}
%>
</body>
</html>
```

Choosephoto.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
<style>
.picture-container {
    position: relative;
    cursor: pointer;
    text-align: center;}
.picture {
    width: 106px;
    height: 106px;
    background-color: #999999;
    border: 4px solid #CCCCCC;
    color: #FFFFFF;
    border-radius: 50%;
```

```
    margin: 5px auto;
    overflow: hidden;
    transition: all 0.2s;
    -webkit-transition: all 0.2s;}
.picture-src {
    width: 100%;
            height: 100%;}
.picture:hover {
    border-color: #4caf50;}
.picture input[type="file"] {
    cursor: pointer;
    display: block;
    height: 100%;
    left: 0;
    opacity: 0 !important;
    position: absolute;
    top: 0;
    width: 100%;}
.choice {
    text-align: center;
    cursor: pointer;}
.choice input[type="radio"], .choice input[type="checkbox"] {
    position: absolute;
    left: -10000px;
```

```
    z-index: -1;}
.choice .icon {
    text-align: center;
    vertical-align: middle;
    height: 106px;
    width: 106px;
    border-radius: 50%;
    color: #999999;
    margin: 5px auto;
    border: 4px solid #CCCCCC;
    transition: all 0.2s;
    -webkit-transition: all 0.2s;
            overflow: hidden;}
.choice .icon:hover {
    border-color: #4caf50;}
.choice.active .icon {
    border-color: #2ca8ff;}
.photo{
margin:50px 560px;}
button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 6px 12px;
```

```
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size 16px;
  transition-duration: 0.4s;
  cursor: pointer;
   border-radius: 12px;}
button {
  background-color: white;
  color: black;
  border: 2px solid #4CAF50;
 border-radius: 12px;
 margin:10px 72px;}
button:hover {
  background-color: red;
  color: white;
  border-radius: 12px;}
</style>
</head>
<body>
<%
String id=request.getParameter("id");
System.out.println("iddd"+id);
%>
```

```
<form action="imageuploader" method="post" enctype="multipart/form-data">

<!-- <h4>Logo:</h4> -->

<div class="col-md-2 col-sm-offset-5 ">

  <div class="picture-container">

    <div class="picture">

      <img src="IMAGE/queationmark.jpg" class="picture-src" id="wizardPicturePreview" title=""/>

     <input type="file" name="photo" id="wizard-picture" aria-invalid="false" class="valid" accept="image/*"
required/> </div>

    <h2>Upload Criminal Photo</h2> </div></div>

 <div class="photo">

 <input type="hidden" value="<%=id%>" name="id">

 <button>Submit</button> </div>

</form>

<script>

$("#wizard-picture").change(function() {

  readURL(this);});

function readURL(input) {

  if (input.files && input.files[0]) {

    var reader = new FileReader();

    reader.onload = function(e) {

      $('#wizardPicturePreview').attr('src', e.target.result).fadeIn('slow');}

    reader.readAsDataURL(input.files[0]);}}

</script>

</body>

</html>
```

CHAPTER 8

RESULTS

HOMEPAGE:

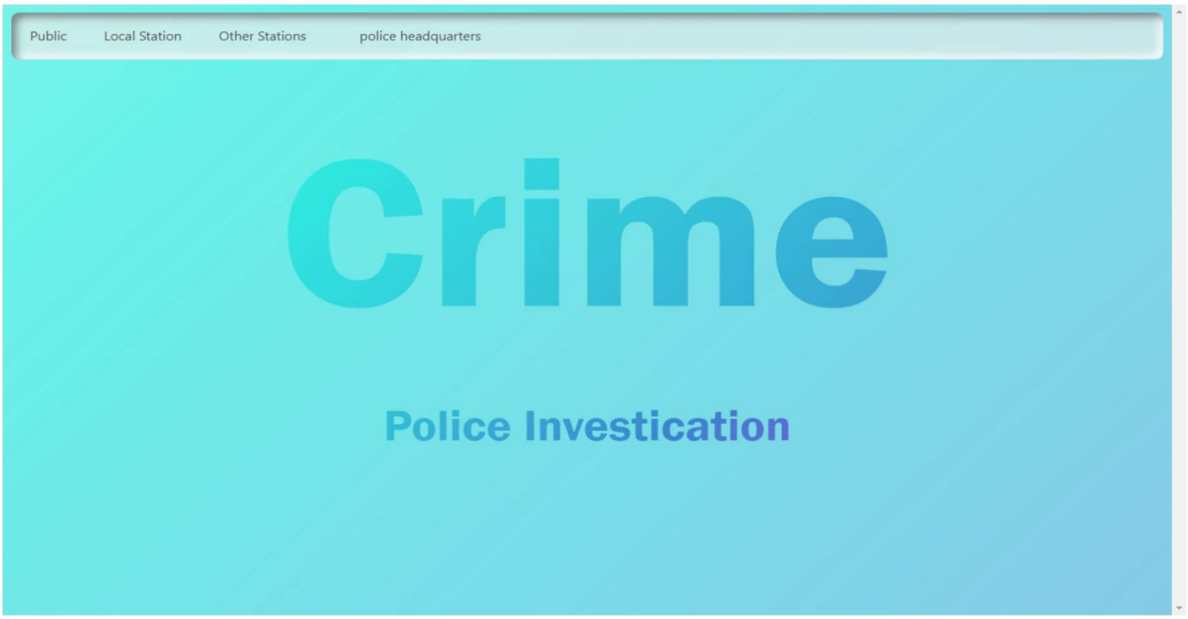


fig 8.1 Homepage

PUBLIC LANDING PAGE

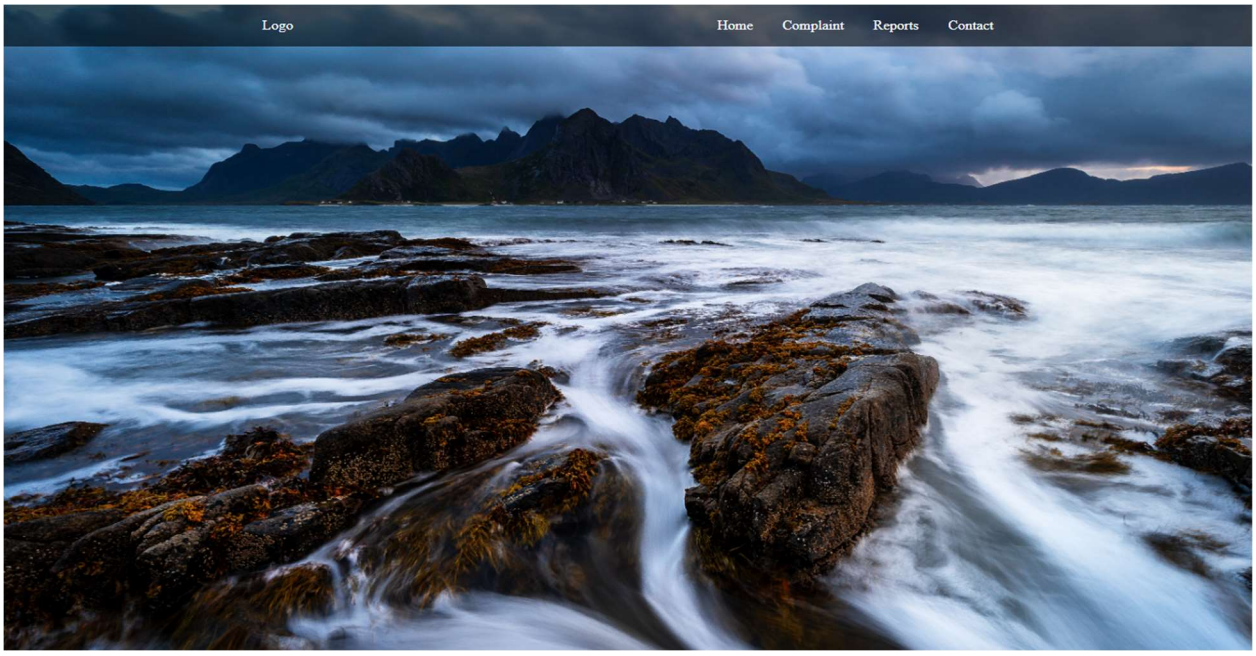


fig 8.2 Public Landing Page

PUBLIC COMPLAINT PAGE:

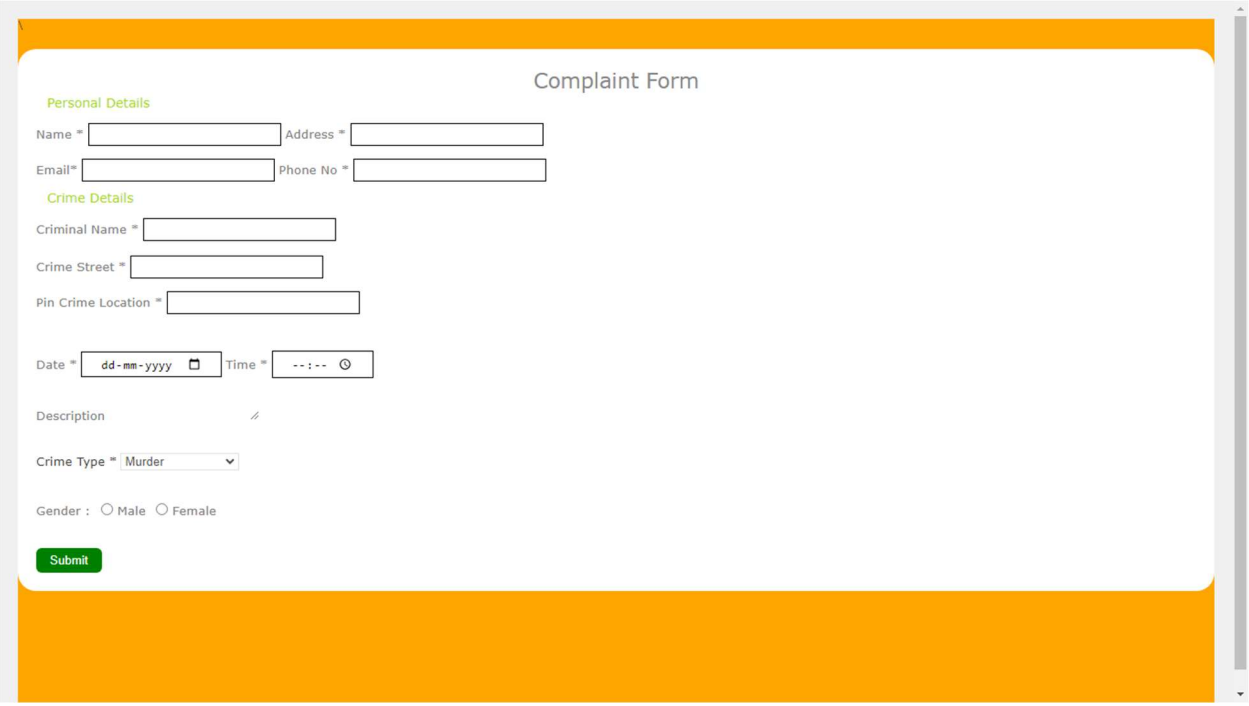


fig 8.3 Public Complaint Page

LOCAL POLICE LOGIN PAGE:

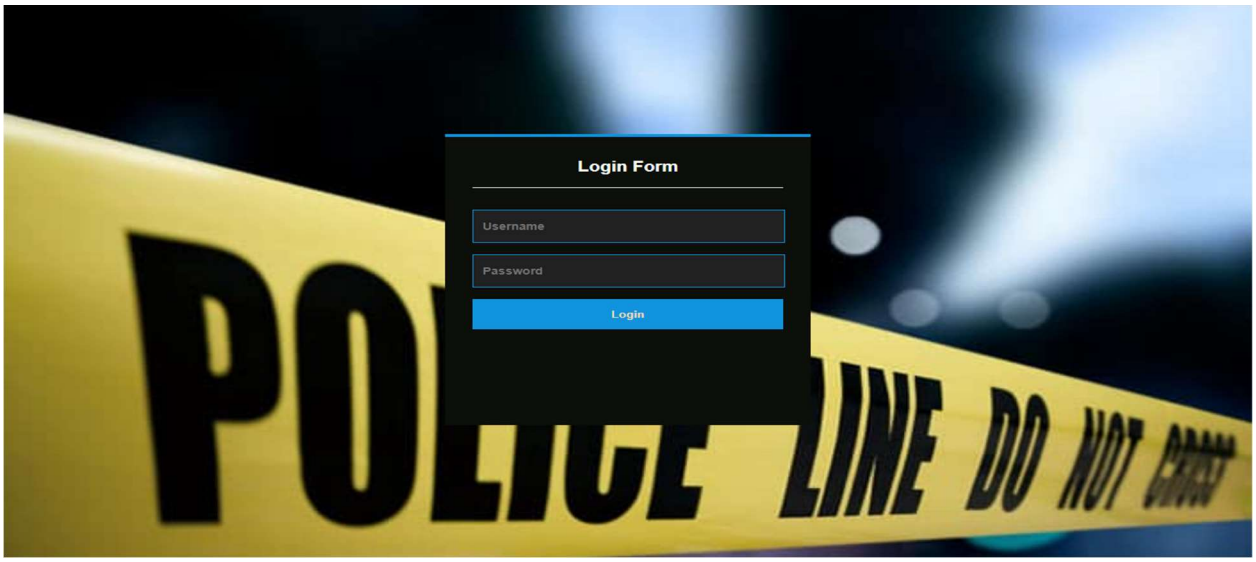


fig 8.4 Local Police Login Page

LOCAL STATION VIEW COMPLAINT:

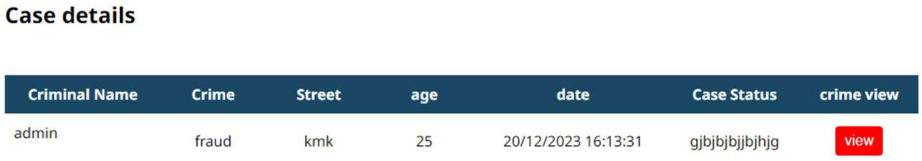


fig 8.5 Local Station View Complaint

OTHER POLICE STATION LOGIN PAGE:

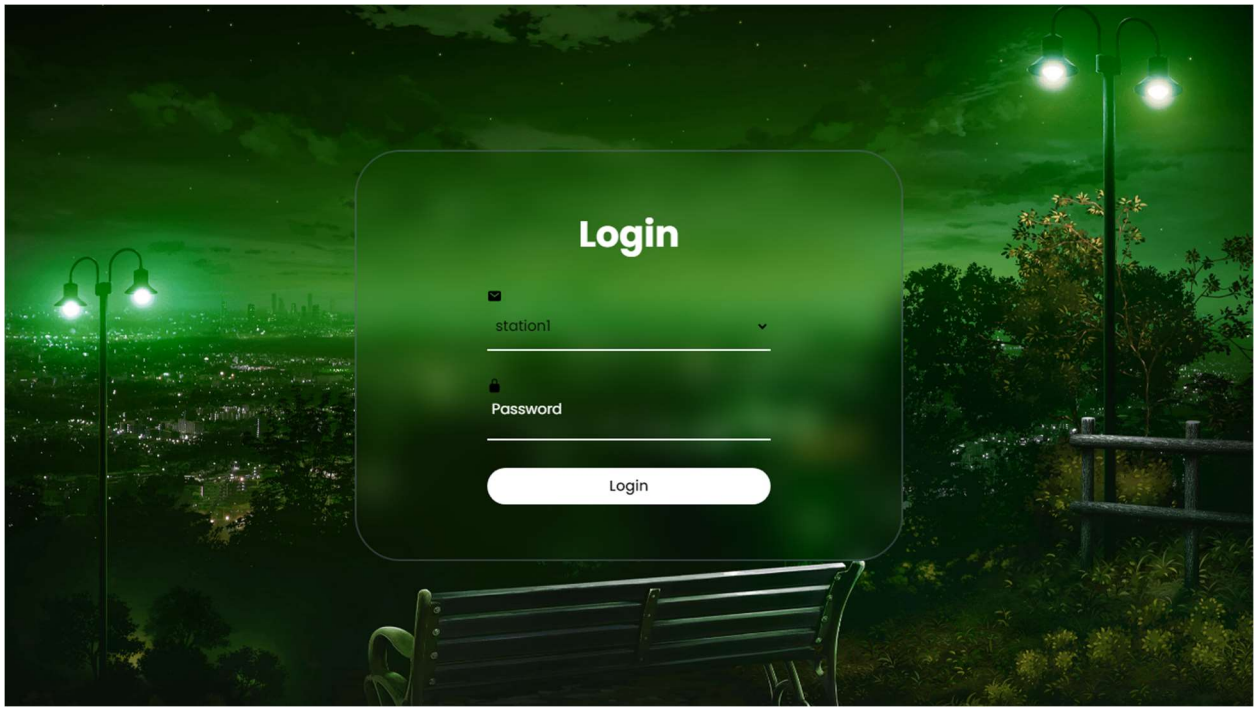


fig 8.6 Other Police Station Login page

HEADQUARTERS LOGIN PAGE:

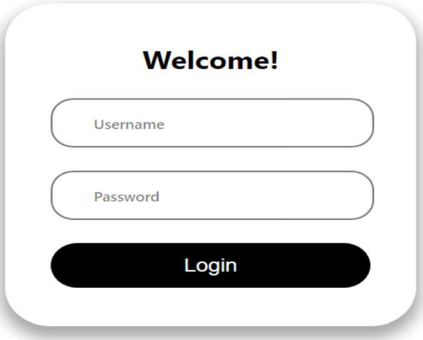


fig 8.7 Headquarters Login page

OTHER POLICE STATION LANDING PAGE:

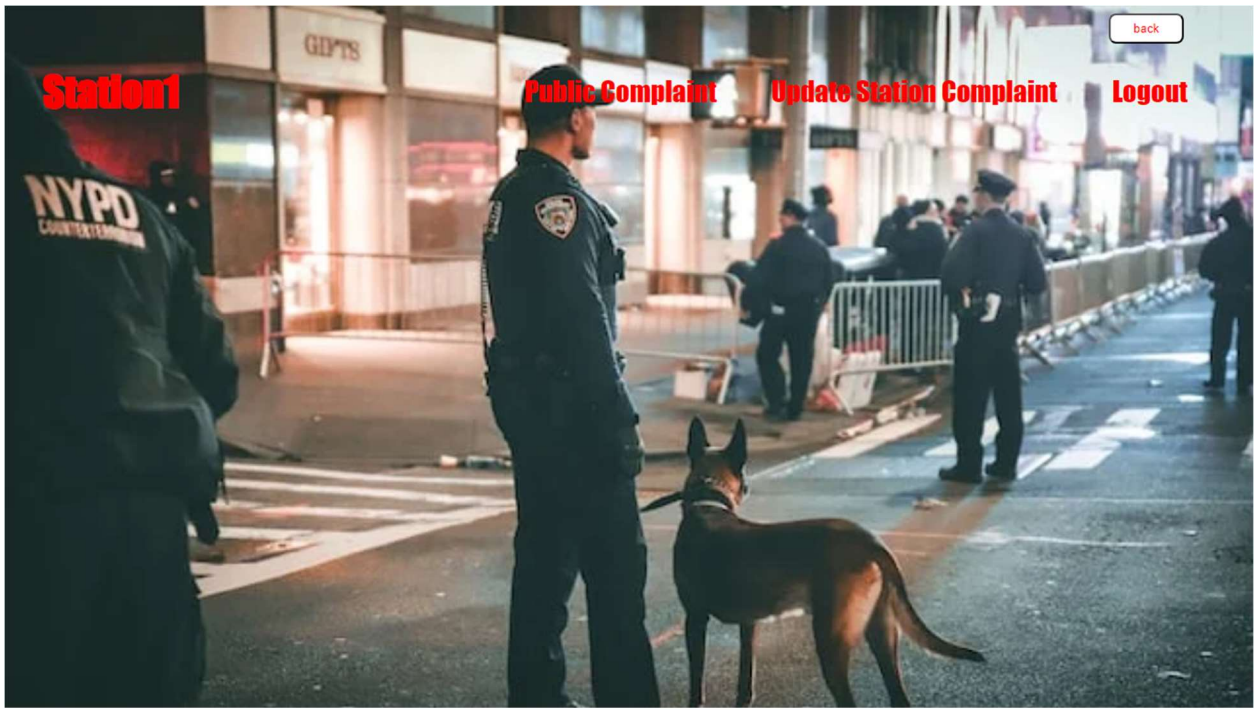


fig 8.8 Other Police Station Landing Page

LOCAL POLICE STATION LANDING PAGE:

fig 8.9 Local Police Station Landing page

OTHER POLICE SHARE CASE FILE:

fig 8.10 Other Police Share Case File

CHAPTER 9

TESTING

SOFTWARE TESTING:

9.1. FEASIBILITY STUDY

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success.

In its simplest terms, the two criteria to judge feasibility are the cost required and the value to be attained. As such, 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.

There are 3 types of Feasibility

- Economical feasibility
- Technical feasibility
- Operational feasibility

9.1.1. ECONOMICAL FEASIBILITY

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. to estimate whether the new system will perform adequately or not.

9.1.2. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources.

9.1.3 OPERATIONAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

9.2. SYSTEM TESTING

The software, which has been developed, has to be tested to prove its validity. Testing is considered to be the least creative phase of the whole cycle of system design. In the real sense, it is the phase, which helps to bring out the creativity of the other phases and makes it shine.

VARIOUS LEVELS OF TESTING

1. White Box Testing
2. Black Box Testing
3. Unit Testing
4. Functional Testing
5. Performance Testing
6. Integration Testing
7. Validation Testing

8. System Testing

9. Output Testing

10. User Acceptance Testing

9.2.1 WHITE BOX TESTING

White-box testing, sometimes called glass-box, is a test case design method that uses the control structure of the procedural design to derive test cases. Using White Box testing methods, we can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure their validity.

9.2.2. BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure, or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

In this testing by knowing the internal operation of a product, a test can be conducted to ensure that “all gears mesh”, that is the internal operation performs

according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

9.2.3. UNIT TESTING

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. 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. It is the testing of individual software units of the application.it is done after the completion of an individual unit before integration.

9.2.4. FUNCTIONAL TESTING

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does. Functional testing differs from system testing in that functional testing "verifies a program by checking it against ... design document(s) or specification(s)", while system testing "validates a program by checking it against the published user or system requirements" (Kane, Falk, Nguyen 1999, p. 52). Functional testing typically involves five steps. The identification of functions that the software is expected to perform

9.2.5. PERFORMANCE TESTING

In general, testing is performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate, or verify other quality attributes of the system, such as scalability, reliability, and resource usage.

Performance testing is a subset of performance engineering, an emerging computer science practice that strives to build performance into the implementation, design, and architecture of a system.

9.2.6. INTEGRATION TESTING

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready. All the errors found in the system are corrected for the next phase.

The purpose of integration testing is to verify the functional, performance, and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success, and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

9.2.7. VALIDATION TESTING

Verification and Validation are independent procedures that are used together to check that a product, service, or system meets requirements and specifications and that it fulfills its intended purpose. These are critical components of a quality management system such as ISO 9000. The words "verification" and "validation" are sometimes preceded by "Independent" (or IV&V), indicating that the verification and validation are to be performed by a disinterested third party.

It is sometimes said that validation can be expressed by the query "Are you building the right thing?" and verification by "Are you building it right?". In practice, the usage of these terms varies. Sometimes they are even used interchangeably.

9.2.8. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated (called *assemblages*) or between any of the *assemblages* and the hardware. It seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

9.2.9. OUTPUT TESTING

After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered in two ways. One is on screen and another is in printed format. The output comes as the specified requirements by the user. Hence output testing does not result in any correction in the system.

9.2.10. USER ACCEPTANCE TESTING

User acceptance of a system is the factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of development and making changes wherever required.

- Input screen design.
- Output screen design.
- Online message to guide users.
- Format of the ad-hoc reports and other outputs.

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using the test data. While testing the system by using test data errors are again uncovered and corrected.

9.3 TEST CASES AND TEST RESULTS

CASE ID	TESTCASE DESCRIPTION	EXPECTED RESULT	OBTAINED RESULT	STATUS
1	Enter correct passkey	Download case file	Download case file	PASS
2	Enter the wrong passkey	Sorry enter the correct passkey	Sorry enter the correct passkey	PASS
3	Upload .pdf or.docx file in report form	Submitted successfully	Submitted successfully	PASS
4	Upload other than .pdf or.docx file in report form	The error page should be shown	The error page should be shown	PASS
5	Upload .jpg,.jpeg, or .png image in other police statice inform module	Submitted successfully	Submitted successfully	PASS
6	Upload other than .jpg,.jpeg or .png files in other police station inform module	Sorry Upload only image files	Sorry Upload only image files	PASS
7	Entering a valid email ID in complaint registration	Email notification sent	Email notification sent	PASS
8	Entering an invalid email ID in complaint registration	Email notification is not sent	Email notification is not sent	PASS
9	Enter a valid email ID and Aadhar number	QR code is generated	QR code is generated	PASS
10	Enter an invalid email ID or Aadhar number	QR code is not generated	QR code is not generated	PASS

Table 9.1 Test-case Report

CHAPTER 10

CONCLUSION AND FUTURE WORK

CONCLUSION:

This project presents a comprehensive solution designed to streamline and enhance the public objection process within criminal cases. Leveraging the Java programming language, the SHA-256 algorithm, and blockchain technology, the system establishes a secure and efficient platform. One of the key strengths of the system lies in its utilization of blockchain technology, which ensures the authenticity and integrity of objection records. By leveraging the decentralized and immutable nature of blockchain, combined with the robust cryptographic mechanisms of the SHA-256 algorithm, the system creates a tamper-proof and verifiable record for each objection submission.

Moreover, the system includes specialized modules tailored for both the police headquarters and the admin, optimizing the handling and processing of objections. These modules are meticulously crafted to efficiently manage objection submissions, thereby enhancing the overall workflow within the criminal justice framework. Additionally, the incorporation of blockchain technology adds an extra layer of security, safeguarding objection records against unauthorized access or manipulation.

Furthermore, the system's ability to generate tamper-proof and verifiable objection records has far-reaching implications for the criminal justice system. It provides law enforcement authorities with reliable and credible evidence, strengthening their ability to address public objections with greater efficiency and accuracy. Additionally, by digitizing and centralizing objection records, the system offers a more systematic and organized approach to managing objections, reducing the likelihood of errors or discrepancies.

In essence, this project represents a significant advancement in addressing public objections within criminal cases.

FUTURE ENHANCEMENTS:

- Development of a practical database system.
- Optimization of protocols for efficiency in terms of message exchange quantity and size.
- Implement using two or more algorithms.
- Integration of Machine Learning for Objection Analysis
- Expansion to Include Multi-Language Support
- Implementation of Advanced Authentication Mechanisms
- Integration with External Systems for Seamless Data Exchange
- Development of Mobile Applications for On-the-Go Access
- Create dedicated mobile applications for iOS and Android platforms, providing users with convenient access to objection submission and tracking functionalities.
- Utilize mobile push notifications to keep users informed about the status of their objections in real time, enhancing user engagement and satisfaction.

CHAPTER 11

REFERENCES

REFERENCE OR BIBLIOGRAPHY:

[1]AN IMPLEMENTATION OF BLOCKCHAIN TECHNOLOGY IN FORENSIC EVIDENCE MANAGEMENT G.Vasavi[1], Dr. G Kalpana [2] M.Tech[1], Professor of CSE [2] Department of Computer Science and Engineering, IJCRT | Volume 11, 8 August **2023**

[2]BLOCKCHAIN-BASED SYSTEM FOR EFFECTIVE POLICE COMPLAINT MANAGEMENT Lynsha Helena Pratheeba HP**, Associate professor,** Bharath D R, Cibiya N E, Dheekshitha S, Divya M N, Department of Computer Science and Engineering, Journal of Emerging Technologies and Innovative Research, Volume 11 | Issue 3 June **2023**

[3]Blockchain-driven Evidence Management System by Shyam Mehta; K. Shantha Kumari; Paras Jain; Harshal Raikwar; Shubham Gore**,** 3rd International Conference on Artificial Intelligence and Signal Processing (AISP),01 June **2023**

[4]FIGAT: Accurately Classify Individual Crime Risks With Multi-Information by Fusion Wenbo Xu, Peiyi Han, Shaoming Duan, and Chuanyi Liu, IEEE Transactions on Services Computing pp. 1890-1903, vol. 16, June **2023**

[5]FIR SYSTEM USING BLOCKCHAIN TECHNOLOGY Bharath Kumar V, Dr. Mir Aadil, International Journal of Advanced Research in Computer and Communication Engineering Vol. 12, Issue 3, March **2023**

[6]Derick Anderes, Edward Baumel, Christian Grier, Ryan Veun and Shante Wright, **"The Use of Blockchain within Evidence Management Systems"**. IJCRT | Volume 10, Issue 6 June **2022**

[6]Impact of Crime Reporting System to Enhance Effectiveness of Police Service, KN Jayasinghe #1, MPL Perera#2, International Journal of Computer Trends and Technology Volume 69 Issue 5, May **2021**

[7]The Application of Blockchain of Custody in Criminal Investigation Process, Fu-Ching Tsai*, Department of Criminal Investigation, Central Police University, Taoyuan City 33304, Taiwan Volume 192, **2021**

[8]Revathy Sathyaprakasan Pratheeksha Govindan, Samina Alvi, Lipsa Sadath, Sharon Philip and Nrashant Singh, "An Implementation of Blockchain Technology in Forensic Evidence Management". 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE),**2021**

[9]D. Kim, S. Y. Ihm, and Y. Son, "Two-level blockchain system for digital crime evidence management", Sensors, vol. 21, no. 9, pp. 3051, **2021.**

[10]Police Complaint Management System Using Blockchain Technology, Ishwarla Hingorani, Rushabh, Khara, Deepika, Pomendkar, Nataasharaul, **2020**

[11]S. Rao, S. Fernandes, S. Raorane and S. Syed, "**A Novel Approach for Digital Evidence Management Using Blockchain**", Proceedings of the International Conference on Recent Advances in Computational Techniques (IC-RACT), June **2020**.

[12]S. Leible, S. Schlager, M. Schubotz and B. Gipp, "**A review on blockchain technology and blockchain projects fostering open science**", *Frontiers in Blockchain*, pp. 16**, 2019**.

[13]A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "**On blockchain and its integration with IoT: Challenges and opportunities**," *Future Gener. Comput. Syst.*, vol. 88, pp. 173–190, **2018.**

[14]M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "**Bubbles of trust: A decentralized blockchain-based authentication system for IoT,**" Comput. Secur., vol. 78, pp. 126–142, **2018.**

[15]S. Chhabra, G. Gupta, M. Gupta, and G. Gupta, "**Detecting fraudulent bank checks**," in *Proc. IFIP Int. Conf. Digit. Forensics*, **2017**, pp. 245–266.

[16] Sunil Yadav, Meet Timbadia, Ajit Yadav, Rohit Vishwakarma, and Nikhilesh Yadav," **Crime pattern detection, analysis and prediction**", International Conference on Electronics, Communication and Aerospace Technology(ICECA), **2017**

[17]Sunil Yadav, Meet Timbadia, Ajit Yadav, Rohit Vishwakarma, and Nikhilesh Yadav," **Crime pattern detection, analysis, and prediction**," International Conference on Electronics, Communication and Aerospace Technology(ICECA), **2017**

[18]K. Christidis and M. Devetsikiotis, "**Blockchains and smart contracts for the Internet of Things**", *IEEE Access*, vol. 4, pp. 2292–2303, **2016**.

[19]Ubon Thansatapornwatana**," A Survey of Data Mining Techniques for Analyzing Crime Patterns**", Second Asian Conference on Defense Technology ACDT, IEEE, Jan **2016**, pp. 123–128

[20] H. Adel, M. Salheen, and R. Mahmoud, "**Crime about urban design. Case study: the greater Cairo region**," Ain Shams Eng. J., vol. 7, no. 3, pp. 925-938, **2016**.

[21]J. Jones and D. McCoy, "**The check is in the mail: Monetization of Craigslist buyer scams**," in Proc.APWGSymp. Electron. Crime Res., **2014**, pp. 25–35.

[22]C.H. Yu, "**Crime Forecasting Using Data Mining Techniques**," in 11th International Conference on Data Mining Workshop, **2011,** pp. 779-786.

[23]C.-D. Chen and L.-T. Huang, "**Online deception investigation: Content analysis and cross-cultural comparison,**" Int. J. Bus. Inf., vol. 6, no. 1, pp. 91–111, **2011**.