GOVERNMENT OF KARNATAKA
DEPARTMENT OF TECHNICAL EDUCATION

"A project report on"

"DATA LEAKAGE DETECTION PROCESS"

A project report submitted to the Department of Technical Education in partial fulfilment of the requirement for a final Year Diploma in Computer Science and Engineering.

2023-2024

SUBMITTED BY,

BHUVAN S	**563CS21003**
HEMANTH HD	**563CS21008**
MOHAMMED SAAD	**563CS21012**
SAGAR K G	**563CS22703**

GUIDED BY:

Ms. Dakshayani HV
Ms. Archana P
Department of CSE,
PES Polytechnic. Shivamogga.

PESPT
EDUCATION FOR THE REAL WORLD

Department of COMPUTER SCIENCE & Engineering

PES POLYTECHNIC, Shivamogga-577204

2023-2024

GOVERNMENT OF KARNATAKA
DEPARTMENT OF TECHNICAL EDUCATION

"A project report on"

"DATA LEAKAGE DETECTION PROCESS"

A project report submitted to the Department of Technical Education in partial fulfilment of the requirement for a final Year Diploma in Computer Science and Engineering.

2023-2024

SUBMITTED BY,

BHUVAN S	**563CS21003**
HEMANTH HD	**563CS21008**
MOHAMMED SAAD	**563CS21012**
SAGAR K G	**563CS22703**

GUIDED BY :

Ms. Dakshayani HV
Ms. Archana P
Department of CSE,
PES Polytechnic. Shivamogga.

PESPT
EDUCATION FOR THE REAL WORLD

Department of COMPUTER SCIENCE & Engineering

PES POLYTECHNIC, Shivamogga-577204

2023-2024

“CANDIDATE DECLARATION"

BHUVAN S	**563CS21003**
HEMANTH HD	**563CS21008**
MOHAMMED SAAD	**563CS21012**
SAGAR K G	**563CS22703**

We, **Bhuvan S, Hemanth HD, Mohammed Saad, and Sagar KG** the students of the Diploma in **Computer Science and Engineering** Department bearing Register Numbers **563CS21003, 563CS21008, 563CS21012, 563CS22703** of **PES Polytechnic**, hereby declare that I owe full responsibility for the information, results, and conclusions provided in this project work titled **“DATA LEAKAGE DETECTION PROCESS**” submitted to Board of Technical Examinations, Government of Karnataka for the award of Diploma in Computer Science and Engineering. To the best of my knowledge, this project work has not been submitted in part or full elsewhere in any other institution/organization for awarding any certificate/diploma/degree.

I have completely taken care in acknowledging the contribution of others in this academic work. I further declare that in case of any violation of intellectual property rights and particulars declared, found at any stage. I, as the candidate will be solely responsible for the same.

Signature of the Candidate

Date:________________

Place:________________

Name: ___________________________

Reg No: ___________________________

Signature of the Candidate

Name: ___________________________

Reg No: ___________________________

Signature of the Candidate

Name: ___________________________

Reg No: ___________________________

Signature of the Candidate

Name: ___________________________

Reg No: ___________________________

GOVERNMENT OF KARNATAKA

DEPARTMENT OF TECHNICAL EDUCATION

PES POLYTECHNIC

NH – 206, Sagara road, Shivamogga – 577204

Department of Computer Science and Engineering

BONAFIDE CERTIFICATE

Certified that this project report **“Data Leakage Detection Process”** is the Bonafide work of “**Bhuvan S, Hemanth HD, Mohammed Saad, Sagar KG**” bearing Register No **“563CS22703, 563CS21003, 563CS21008, 563CS21012, 563CS22703”** of this institution who carried out the project work under my supervision.

Signature of the Guide:	Signature of HOD:
Ms. Dakshayani HV, / Ms. Archana P	**Mrs. Vanitha D,**
Lecturer, Dept of CSE,	**HOD, Dept of CSE,**
PES Polytechnic, Shimoga.	**PES Polytechnic, Shimoga**

GOVERNMENT OF KARNATAKA
DEPARTMENT OF TECHNICAL EDUCATION

PES POLYTECHNIC

NH – 206, Sagara Road, Shivamogga – 577204

Department of Computer Science and Engineering

CERTIFICATE

Certified that this project report entitled "**Data Leakage Detection Process"** is being submitted by **Sagar KG, Mohammed. Saad, Bhuvan S, Hemanth HD "** Reg No "**563CSS22703, 563CS21012, 563CS21003, 563CS21008 "** student of **PES Polytechnic** in partial fulfilment for the award of **Diploma of Computer Science and Engineering during the year 2023-24** is the record of student's work carried out under my/our guidance. It is certified that all corrections/suggestions indicated for internal Assessment have been incorporated in the Report and One copy of it is deposited in the Polytechnic library.

The project report has been approved as it satisfies the academic requirements concerning the Project work prescribed for the said Diploma.

It is further understood that by this Certificate the undersigned do not endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the project only for the purpose for which it is submitted.

Signature of Guide:	Signature of HOD:	Signature of Principal:
________________	____________________	____________________
Ms. Dakshayani HV **Ms. Archana P**	**Mrs. Vanitha D,**	**Prof. Gowtham JK**
Lecturer, Dept of CSE,	HOD, Dept of CSE,	**PRINCIPAL**
PES Polytechnic,	PES Polytechnic,	PES Polytechnic,
Shimoga.	Shimoga.	Shimoga.

Examiners Name and Sign and Date: ___________________________

ACKNOWLEDGEMENT

No work is complete with due recording being given to the person who made it possible. My project work report is no exception. I would like to place on record, my profound gratitude for those who have mattered most in the successful completion of the report.

Good guidance takes a long way in achieving our goals. We are grateful to our guide **Ms. Dakshayani HV and Ms. Archana P** Lecturer of Computer Science and Engineering, for guidance in preparing the project report.

We sincerely thank **Mrs. Vanitha D,** HOD of the Computer Science and Engineering Diploma for her guidance. Encouragement and timely advice. We are very grateful for her support for the entire project.

We would like to express our deep gratitude to our respected principal, **Prof. Gowtham JK,** who is the source of encouragement for all the students in general and us in particular in completing this project.

Last but not least, Thank our faculty and non-teaching staff for PES Polytechnic for their support.

BHUVAN S	**563CS21003**
HEMANTH HD	**563CS21008**
MOHAMMED SAAD	**563CS21012**
SAGAR K G	**563CS22703**

ABSTRACT

SQL injection remains a pervasive threat to the security of web applications and databases, enabling attackers to gain unauthorized access to sensitive data and compromise the integrity of systems. This paper presents a comprehensive approach for detecting and mitigating SQL injection vulnerabilities, focusing on real-time detection and minimizing false positives. Leveraging advanced techniques such as input validation, error message analysis, and automated testing tools, our solution identifies potential vulnerabilities within web applications and intercepts malicious SQL queries before they can compromise the database. By integrating seamlessly with existing security infrastructure and prioritizing scalability and performance, our approach enhances the security posture of organizations, mitigating the risk of data breaches resulting from SQL injection attacks.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. For example, a hospital may give patient records to researchers who will devise new treatments. Similarly, a company may have partnerships with other companies that require the sharing of customer data. Another enterprise may outsource its data processing, so data must be given to various other companies. We call the owner of the data the distributor and the supposedly trusted third parties the agents. Then further datawill be provided by the distributor to the trusted third party of the enterprise using this application. This application will further monitor if in case any data has been leaked by the agent of the enterprise.

1.2 BRIEF DESCRIPTION

This will describe the overall mathematical model of the project and the settheory used in the project.

1.2.1 Project Problem Statement:

To build an application that helps in **Detecting the data** that has been leaked. It also helps in finding the **Guilt of an Agent** from the given set of agents that has leaked the data using **Probability Distribution**.

1.2.2 Problem Description:

The objective is to develop a comprehensive solution for detecting and mitigating SQL injection vulnerabilities, thereby safeguarding sensitive data from unauthorized access and potential leakage. This solution must address the following key challenges:

1. Vulnerability Identification: Identify and analyze potential SQL injection vulnerabilities within web applications, including both known and emerging attack vectors.

2. Real-time Detection: Implement mechanisms for real-time detection of SQL injection attempts, capable of intercepting and blocking malicious queries before they can compromise the database.

3. Minimizing False Positives: Develop techniques to minimize false positive detections, ensuring that legitimate user input is not mistakenly flagged as malicious.

4. Scalability and Performance: Design a scalable and efficient detection system capable of handling large volumes of traffic without impacting the web application's performance.

5. Integration with Existing Infrastructure: Ensure seamless integration with existing security infrastructure, including web application firewalls (WAFs) and intrusion detection/prevention systems (IDS/IPS).

1.3 PROBLEM DEFINITION

Our goal is to detect when the distributor’s sensitive data has been leaked by agents, and if possible, to identify the agent that leaked the data.

1.4 APPLYING SOFTWARE ENGINEERING APPROACH

The incremental model is an evolution of the waterfall model. The product is designed, implemented, integrated, and tested as a series of incremental builds. It is a popular model of software evolution used by many commercial software companies and system vendors. An incremental software development model may apply to projects where Software Requirements are well defined, but realization may be delayed.

The basic software functionality is required early

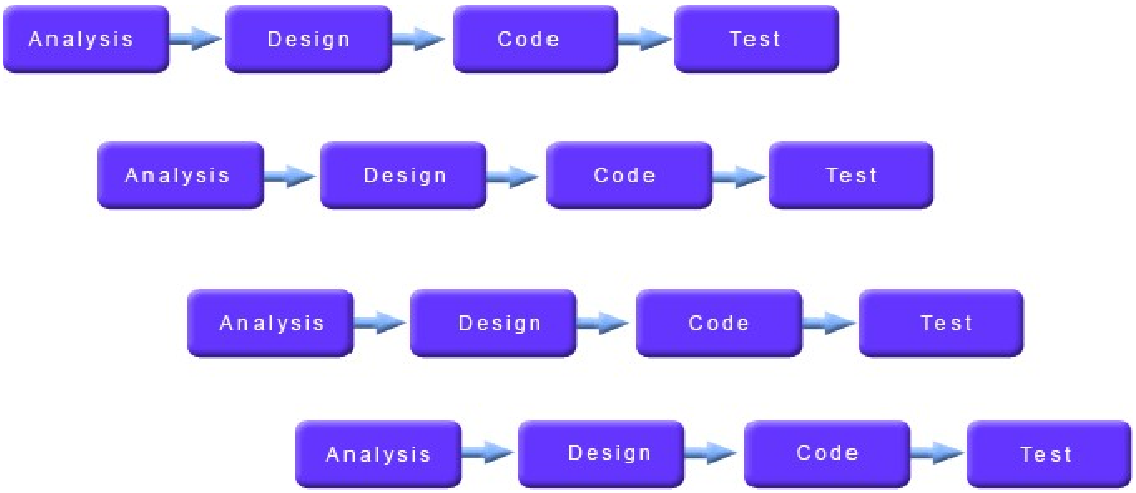


Fig 1.1 System Software Engineering Approach

ADVANTAGES:

- Generates working software quickly and early during the software life cycle.
- More flexible - less costly to change scope and requirements.
- Easier to test and debug during a smaller iteration.
- Easier to manage risk because risky pieces are identified and handled during iteration.

Disadvantages:

- Each phase of an iteration is rigid and does not overlap each other.
- Problems may arise about system architecture because not all requirements are gathered up front for the entire software life cycle.

CHAPTER 2
LITERATURE SURVEY

2.1 DATA LEAKAGE DETECTION

We consider applications where the original sensitive data cannot be perturbed. Perturbation is a very useful technique where the data is modified and made “less sensitive” before being handed to agents. For example, one can add random noise to certain attributes, or one can replace exact values with ranges. However, in some cases, it is important not to alter the original distributor’s data. For example, if an outsourcer is doing our payroll, he must have the exact salary and customer bank account numbers. If medical researchers will be treating patients (as opposed to simply computing statistics), they may need accurate data for the patients. Traditionally, leakage detection is handled by watermarking, e.g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. Watermarks can be very useful in some cases, but again, they involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious. In this paper, we study unobtrusive techniques for detecting leakage of a set of objects or records. Specifically, we study the following scenario: After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place. (For example, the data may be found on a website, or may be obtained through a legal discovery process.)

At this point, the distributor can assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Using an analogy with cookies stolen from a cookie jar, if we catch Freddie with a single cookie, he can argue that a friend gave him the cookie. But if we catch Freddie with 5 cookies, it will be much harder for him to argue that his hands were not in the cookie jar. If the distributor sees “enough evidence” that an agent leaked data, he may stop doing business with him or may initiate legal proceedings. In this paper, we develop a model for assessing the “guilt” of agents. We also present algorithms for distributing objects to agents, in a way that improves our chances of identification.

2.2 DATA MINING

Data Mining (the analysis step of the knowledge discovery in databases process, or KDD), a relatively young and interdisciplinary field of computer science is the process of discovering new patterns from large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The goal of data mining is to extract knowledge from a data set in a human-understandable structure and involves database and data management, data preprocessing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of found structure, visualization, and online updating.

2.2.1 The Scope of Data Mining

Data mining derives its name from the similarities between searching for valuable business information in a large database — for example, finding linked products in gigabytes of store scanner data — and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material or intelligently probing it to find exactly where the value resides. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

- **Automated prediction of trends and behaviors.**

Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data — quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings.

- **Automated discovery of previously unknown patterns**

. Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors. Data mining techniques can yield the benefits of automation on existing software and hardware platforms and can be implemented on new systems as existing platforms are upgraded and new products are developed. When data mining tools are implemented on high-performance parallel processing systems, they can analyze massive databases in minutes. Faster processing means that users can automatically experiment with more models to understand complex data. The high speed makes it practical for users to analyze huge quantities of data. Larger databases, in turn, yield improved predictions.

2.3 WATERMARKING

Digital Watermarking describes methods and technologies that hide information, for example, a number or text, in digital media, such as images, video, or audio. The embedding takes place by manipulating the content of the digital data, which means the information is not embedded in the frame around the data. The hiding process has to be such that the modifications of the media are imperceptible. For images, this means that the modifications of the pixel values have to be invisible. Furthermore, the watermark must be either robust or fragile, depending on the application. By "robust" we mean the capability of the watermark to resist manipulations of the media, such as lossy compression (where compressing data and then decompressing it retrieves data that may well be different from the original, but is close enough to be useful in some way), scaling, and cropping, just to enumerate some. In some cases, the watermark may need to be fragile. "Fragile" means that the watermark should not resist tampering, or would resist only up to a certain, predetermined extent The first applications that came to mind were related to copyright protection of digital media. In the past duplicating artwork was quite complicated and required a high level of expertise for the counterfeit to look like the original. However, in the digital world, this is not true. Now almost anyone can duplicate or manipulate digital data and not lose data quality. Similar to the process when artists creatively signed their paintings with a brush to claim copyrights, artists of today can watermark their work by hiding their names within the image.

In this scenario, digital watermarking may be useful to set up controlled audio distribution and to provide efficient means for copyright protection, usually in collaboration with international registration bodies.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 INTRODUCTION

In this project, we develop a model for assessing the "guilt" of agents. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker. Finally, we also consider the option of adding "fake" objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agents. In a sense, the fake objects act as a type of watermark for the entire set, without modifying any individual members. If it turns out that an agent was given one or more fake objects that were leaked, then the distributor can be more confident that the agent was guilty.

We define unobtrusive techniques for detecting leakage of a set of objects or records. Specifically, we study the following scenario: After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place. (For example, the data may be found on a website, or may be obtained through a legal discovery process.) At this point, the distributor can assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means.

Using an analogy with cookies stolen from a cookie jar, if we catch Freddie with a single cookie, he can argue that a friend gave him the cookie. But if we catch Freddie with five cookies, it will be much harder for him to argue that his hands were not in the cookie jar. If the distributor sees "enough evidence" that an agent leaked data, he may stop doing business with him or may initiate legal proceedings.

3.1.1 Objective

In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. For example, a hospital may give patient records to researchers who will devise new treatments. Similarly, a company may have partnerships with other companies that require the sharing of customer data. Another enterprise may outsource its data processing, so data must be given to various other companies. We call the owner of the data the distributor and the supposedly trusted third parties the agents.

Our goal is to detect when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data.

3.1.2 Project Scope

The project aims to introduce an easy and efficient solution that is capable of detecting data leakage in the system and if possible the agent details who did it: o Agent/Distributor Data Communication o Finding Guilty Agent and their details We consider applications where the original sensitive data cannot be perturbed. Perturbation is a very useful technique where the data are modified and made less sensitive before being handed to agents. For example, one can add random noise to certain attributes, or one can replace exact values with ranges. However, in some cases, it is important not to alter the original distributor's data. For example, if an outsourcer is doing our payroll, he must have the exact salary and customer bank account numbers. If medical researchers will be treating patients (as opposed to simply computing statistics), they may need accurate data for the patients.

we study the following scenario: After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place. (For example, the data may be found on a website, or may be obtained through a legal discovery process.) At this point, the distributor can assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Using an analogy with cookies stolen from a cookie jar, if we catch Freddie with a single cookie, he can argue that a friend gave him the cookie. But if we catch Freddie with five cookies, it will be much harder for him to argue that his hands were not in the cookie jar If the distributor sees "enough evidence" that an agent leaked data, he may stop doing business with him or may initiate legal proceedings.

3.1.3 User Classes and Characteristics.

- Administrator who will be given all the privileges about the application activities such as blocking the guilty agent using probability distribution table. He can also see the shared files and manages the whole database
- Distributor who will send data T to different agents U . He will also accept requests from the different agents about the particular type of files and will distribute it accordingly
- Agents who will receive the data from the distributor D. The agent can also request the particular type of text files. These agents are continuously monitored by the administrator and distributor in case they leak the data.

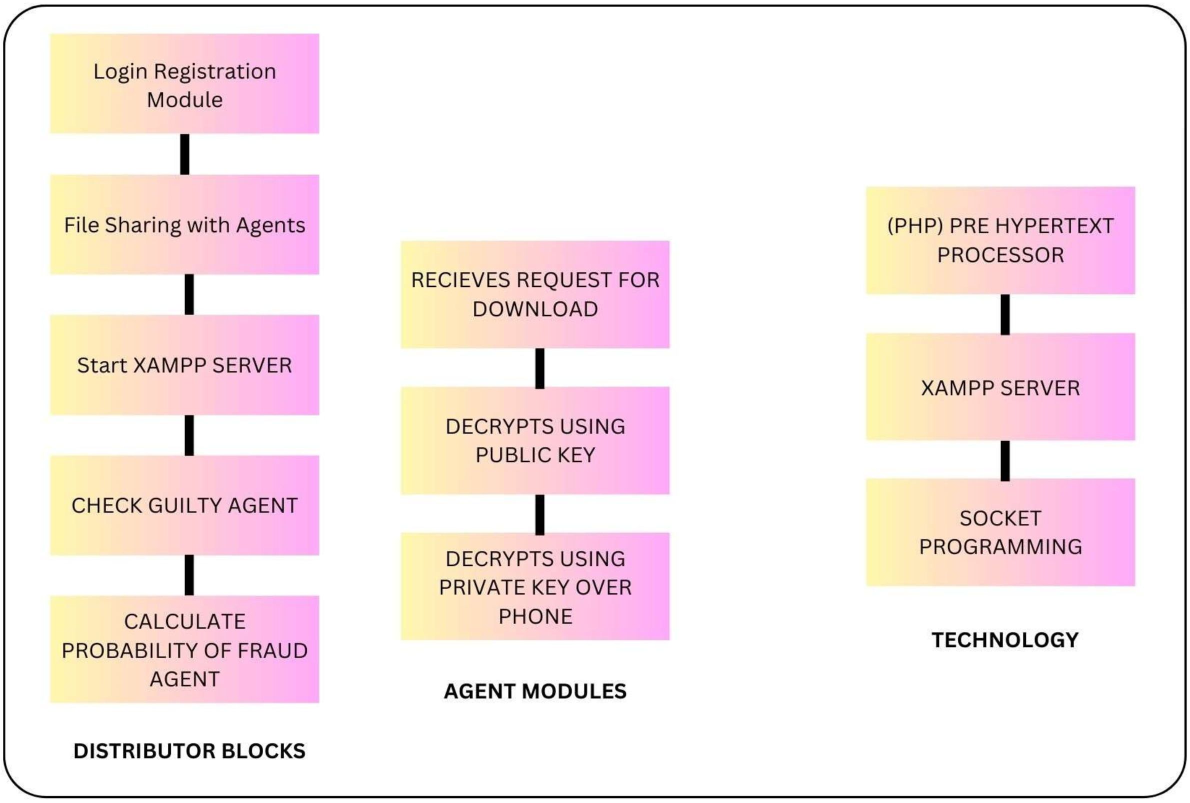

Login Registration Module
File Sharing with Agents
Start XAMPP SERVER
CHECK GUILTY AGENT
CALCULATE PROBABILITY OF FRAUD AGENT
DISTRIBUTOR BLOCKS
RECIEVES REQUEST FOR DOWNLOAD
DECRYPTS USING PUBLIC KEY
DECRYPTS USING PRIVATE KEY OVER PHONE
AGENT MODULES
(PHP) PRE HYPERTEXT PROCESSOR
XAMPP SERVER
SOCKET PROGRAMMING
TECHNOLOGY

Fig 3.1 Data Leakage Detection Overview

3.1.4 Operating Environment

To develop this system, we are using PHP language and its different technology.

3.1.4.1 PHP (Hypertext Preprocessor)

PHP, which stands for Hypertext Preprocessor, is a widely used server-side scripting language primarily designed for web development. It is a powerful and versatile language that is particularly well-suited for creating dynamic and interactive websites. PHP code is embedded directly into HTML documents and executed on the server, generating dynamic content that is then sent to the client's web browser.

One of the key features of PHP is its ability to interact with databases, making it an ideal choice for building dynamic websites that require data storage and retrieval. PHP supports a wide range of database systems, including MySQL, PostgreSQL, and SQLite, allowing developers to create dynamic web applications with complex data-driven functionality.

PHP is also known for its ease of use and flexibility, with a simple and intuitive syntax that is easy to learn and understand. It offers a wide range of built-in functions and libraries for common tasks such as file handling, string manipulation, and form processing, reducing the need for developers to write code from scratch.

3.1.4.2 XAMPP SERVER

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It stands for Cross-platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P). XAMPP is designed to facilitate web development and testing environments on a local computer, allowing developers to create and test dynamic web applications without the need for a live server.

The XAMPP package includes several components:

2. MySQL Database Server: MySQL is a powerful relational database management system (RDBMS) widely used for storing and managing structured data. XAMPP includes MySQL to enable developers to create and interact with databases locally.

3. PHP: PHP is a server-side scripting language used for developing dynamic web applications. XAMPP includes PHP, allowing developers to write PHP scripts and execute them on the server to generate dynamic content.

XAMPP provides developers with a convenient and easy-to-use solution for setting up a local web server environment. It is available for multiple operating systems, including Windows, macOS, and Linux, making it accessible to a wide range of developers. With XAMPP, developers can quickly create, test, and debug web applications without the need for an internet connection or access to a remote server. It is particularly useful for beginners learning web development or experienced developers looking to prototype and test new ideas in a local environment before deploying them to a live server.

3.1.4.3 Socket Programming

A socket is an endpoint of a two-way communication link between two programs running on the network.

The socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent. Java provides a set of classes, defined in a package called java.net, to enable the rapid development of network applications. Key classes, interfaces.

.

3.1.5 Product Features

Product features are:

1. Agent/Distributor Sign up/log in

2. Distributor distributes files to agents

3. File Encryption using public and private key

4. Guilty Agent Report and their machine details

5. Agent requests for the particular text file.

6. Probability distribution of the agents.

7. The administrator can block the guilty agent.

3.1.6 Design and Implementation Constraints

Design Constraints

a. Error Recognition: Error should be easily recognized and solved.

General Constraints

a. . Network Speed – Files need to be transmitted for any transaction using triple AES encryption, it introduces an additional overhead on network bandwidth.

User Documentation

a. Currently we will be using Java to develop Data Leakage detection techniques.

b. Files will be public/private key encrypted.

3.1.7 Assumption and Dependencies

Assumptions:

a. Agent must have basic knowledge of the internet.

b. Agent should log in from a single workstation/machine.

Dependencies:

a. Speed of the network

b. Size of the file

3.2 System Features

The main focus of this project is the data allocation problem: how can the distributor “intelligently” give data to agents to improve the chances of detecting a guilty agent?

As illustrated in Fig, there are four instances of this problem we address, depending on the type of data requests made by agents and whether “fake objects” are allowed.

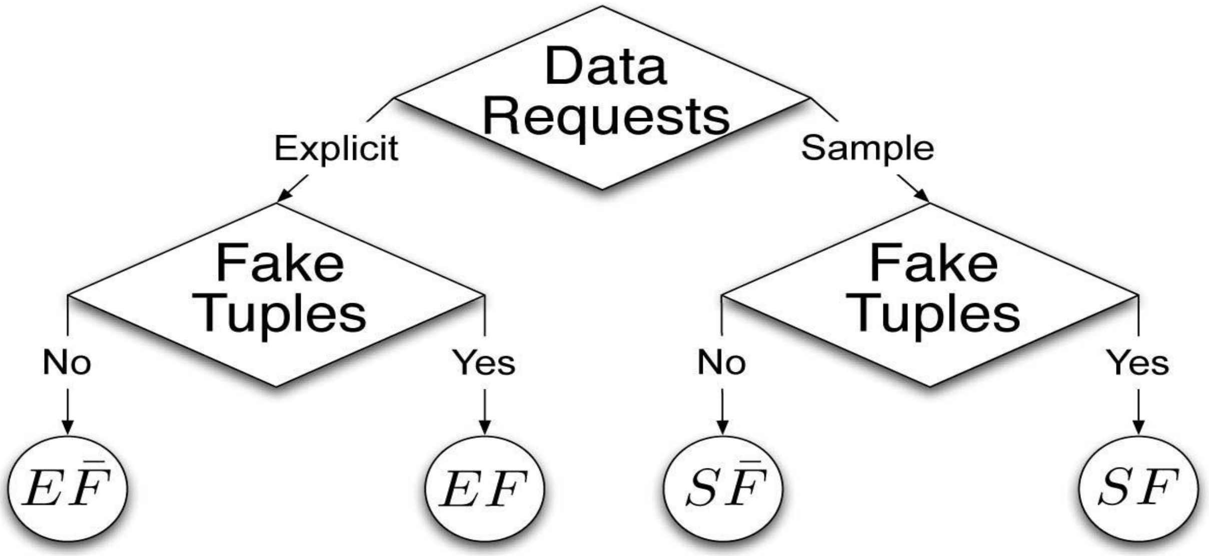


Fig 3.2 System Feature

We handle two types of requests - sample and explicit. The distributor generates fake objects that are not in set T, to look like real objects and distribute them to agents with T objects. This helps to detect agents that leak data. However, fake objects may impact the correctness of what agents do, so they may not always be allowed. We perturb the set of distributor objects by adding fake elements. In some applications, fake objects may cause fewer problems than perturbing real objects.

For example, if the distributed data objects are medical records and the agents are hospitals, even small modifications to the records of actual patients may be undesirable. However, the addition of some fake medical records may be acceptable since no patient matches these records, and hence, no one will ever be treated based on fake records. Our use of fake objects is inspired by the use of "trace" records in mailing lists.

The distributor creates and adds fake objects to the data that he distributes to agents. Fake objects must be created carefully so that agents cannot distinguish them from real objects. The distributor may be limited in how many fake objects he can create, and he may want to limit the number of fake objects received by each agent to not arouse suspicions and not adversely impact the agents' activities.

The distributor's constraint is to satisfy agents' requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to detect any agent who leaks any portion of his data. We consider the constraint as strict, and the distributor may not deny serving an agent request or provide agents with different perturbed versions of the same objects. Our detection objective is ideal and intractable. We use instead the following objective: maximize the chances of detecting a guilty agent that leaks all his data objects

3.2.1 Data Allocation Problem

The main focus of this project is the data allocation problem: how can the distributor "intelligently" give data to agents to improve the chances of detecting a guilty agent?

1. The distributor will be sending data that will be encrypted using public and private keys to agents; any agent who wants to download the file has to log in to the system and download the file.
2. As the File is in encrypted format, the agent will have to enter the private key to open the file in the software.

3.2.3 Probability Calculation For Guilty Agent

Once we know that the file was leaked, we will have to calculate the probability of the agent being guilty.

3.3 Hardware Interfaces

- 2.4 GHZ, 80 GB HDD for installation.
- 512 MB memory.
- Users can use any PC-based browser clients with IE 5.5 upwards. Internet service provider for group mailing with static IP.

Software Interfaces

- PHP (Hypertext Preprocessor)
- XAMPP SERVER
- Socket Programming
- Triple DES algorithm

3.4 FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behaviors or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture.

3.4.1 Performance Requirements

The system should process all Internet banking requests in parallel for various users to give a quick response time and should complete the process as a whole in one go. UI should not hang for any inquiry from the client.

3.4.2 Safety requirements

Data safety must be ensured by arranging for a secure and reliable transmission media. The source and destination information must be entered correctly to avoid any misuse or malfunction.

3.4.3 Security Requirements

1. User password must be stored in encrypted form for security reasons
2. All the user details shall be accessible to only high authority persons.
3. Access will be controlled with usernames and passwords.
4. Files cannot be accessed from the file system, the only way to view them is through private key decryption from software.

3.5ANALYSIS MODELS

3.5.1 Data Flow Diagrams

Level 0:

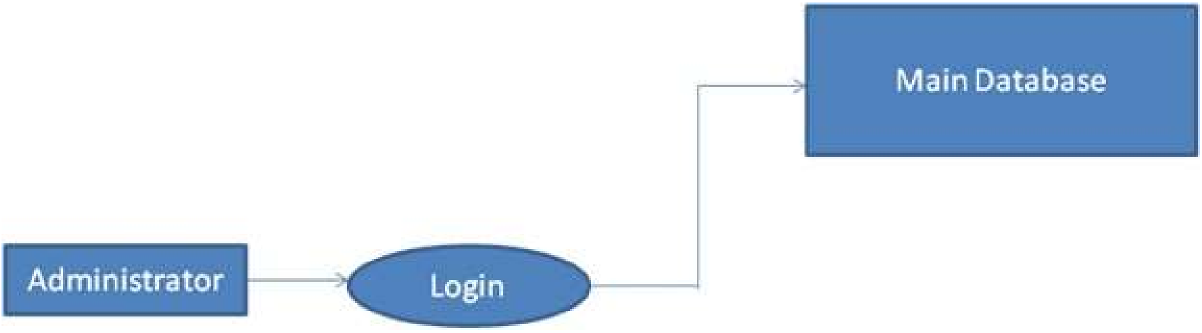


Fig 3.3 Data Allocation

Level 1:

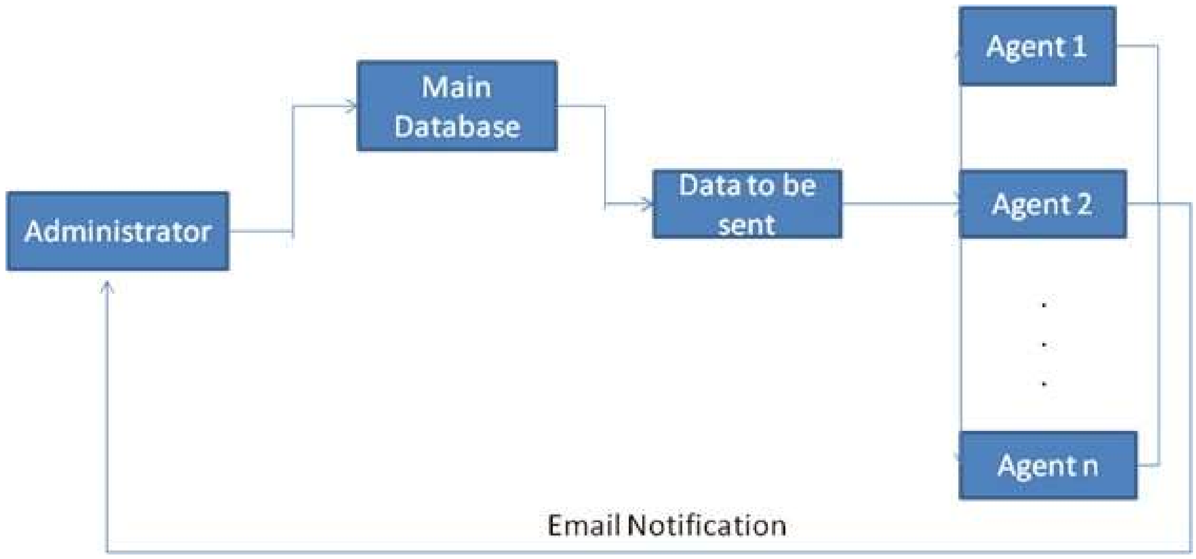


Fig 3.4 Data Distribution

Level 2:

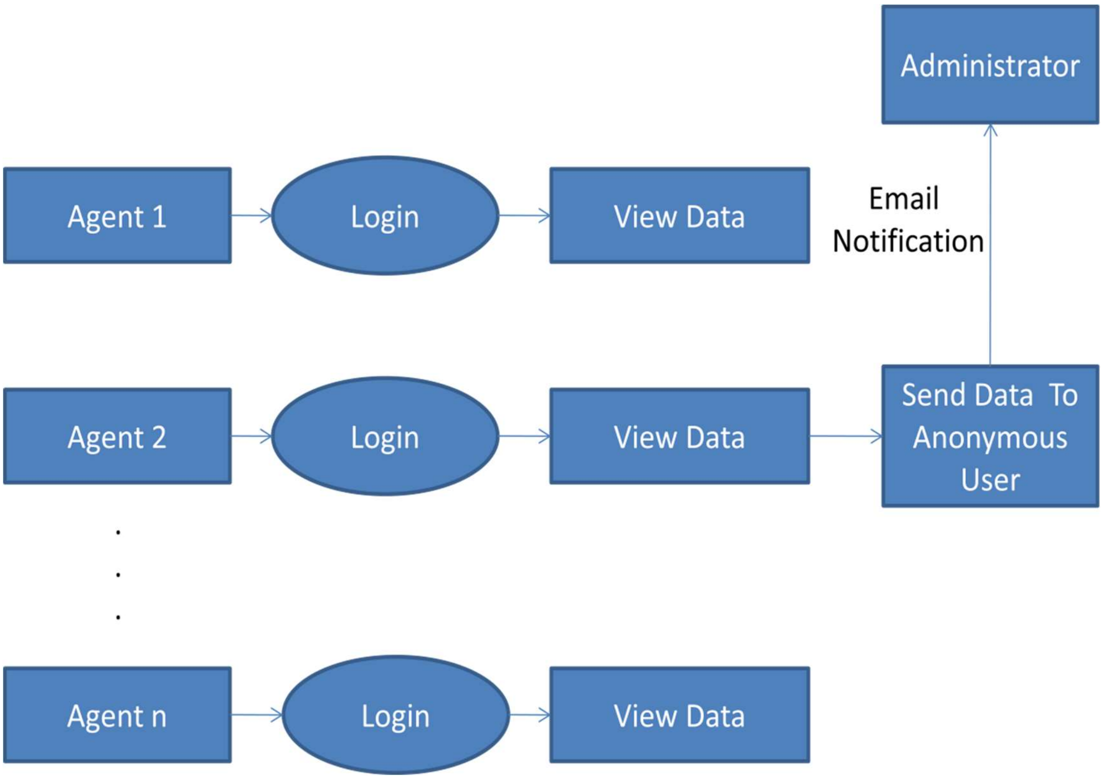


Fig 3.5 Data Leakage and Detection

3.5.2 Class Diagrams

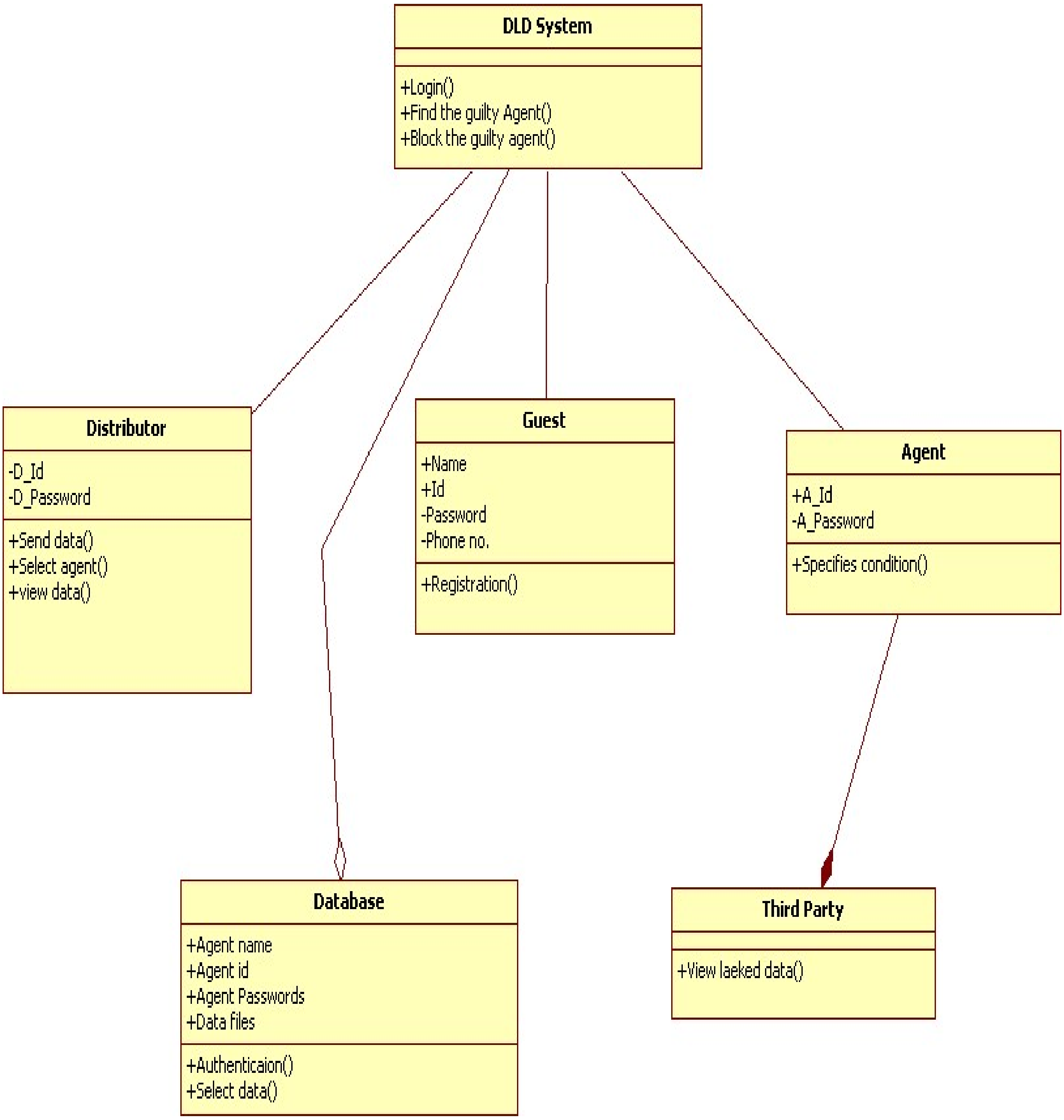


Fig 3.6 Class Diagram

3.5.3 ENTITY RELATIONSHIP DIAGRAM

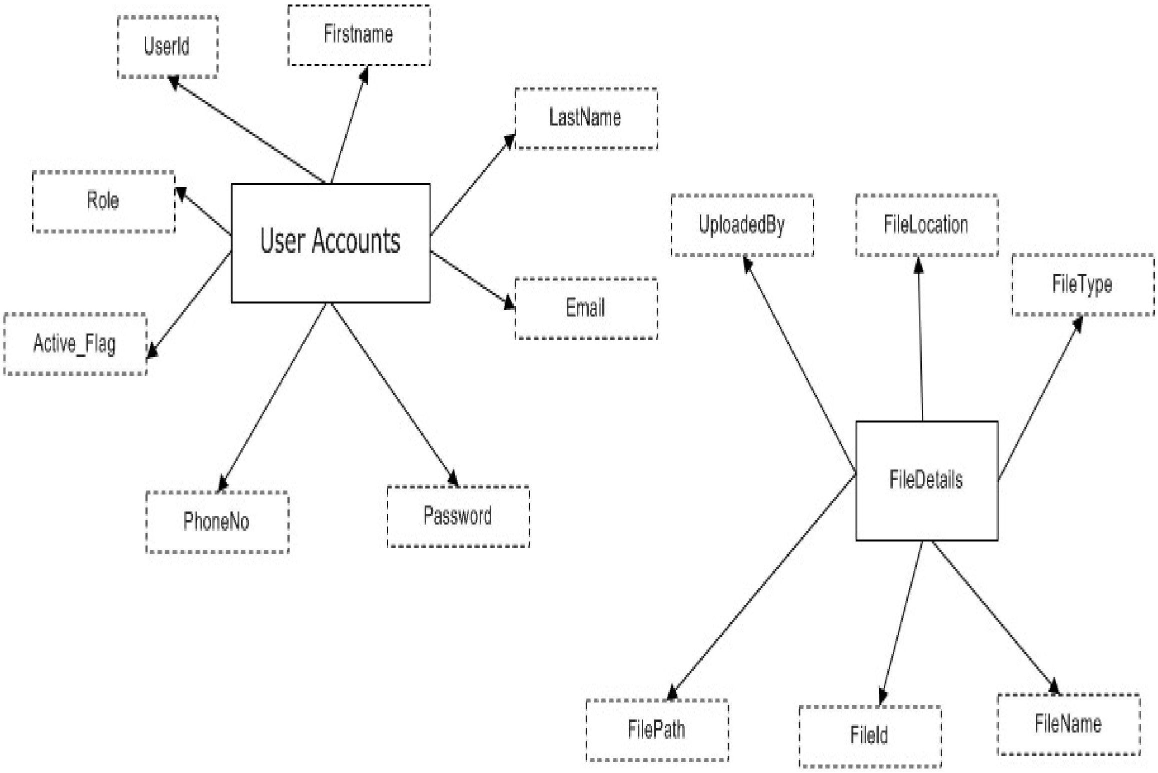


Fig 3.7 Entity relationship diagram

CHAPTER 4

SYSTEM DESIGN

Design is concerned with identifying software components and specifying relationships among components. It specifies software structure and provides blue blueprint for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner the interaction between parts is minimal and specified. The design will explain software components in detail; this will help the implementation of the system.

Implementation is the stage of the project when the theoretical design is turned into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover, and evaluation of changeover methods.

EXPECTED INPUT

Input design is the process of converting user-originated inputs to computer-based formats. Input design is one of the most expensive phases of the operation of a computerized system and is often the major problem of a system.

EXPECTED OUTPUT

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system based on which they evaluate the usefulness of the application.

4.1 SYSTEM ARCHITECTURE

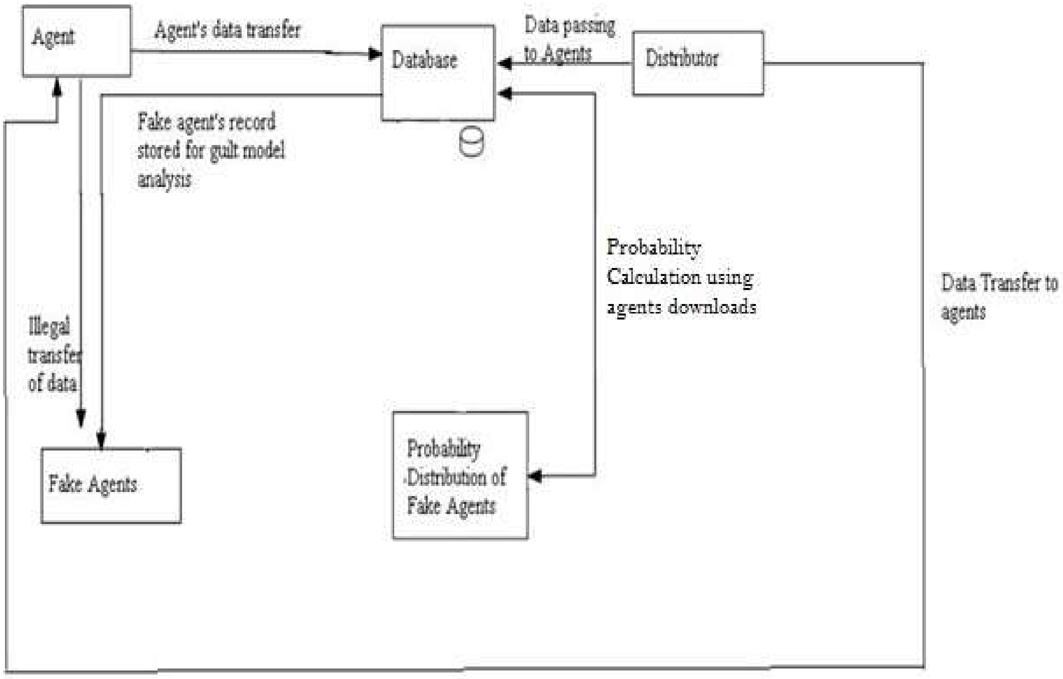

Agent
Agent's data transfer
Database
Data passing to Agents
Distributor
Fake agent's record stored for gult model analysis
Probability Calculation using agents downloads
Data Transfer to agents
Illegal transfer of data
Fake Agents
Probability Distribution of Fake Agents

Fig 4.1 System Architecture

The working of the whole system is as follows:

1. The distributor logins into the system.

2. The distributor uploads the Data [eg.text files] into the Database.

3. The agent asks for the particular file or the distributor uploads all the files for the agents accordingly along with the private key after Login into the system.

4. Agents will download the files according to their needs [Sample request or Explicit request].

5. If any agents leak the data to a third party [Fake Agents] the distributor will check for the leaked data and will find the file that has been leaked.

4.2 UML Diagrams

4.2.1 Use Case Diagram

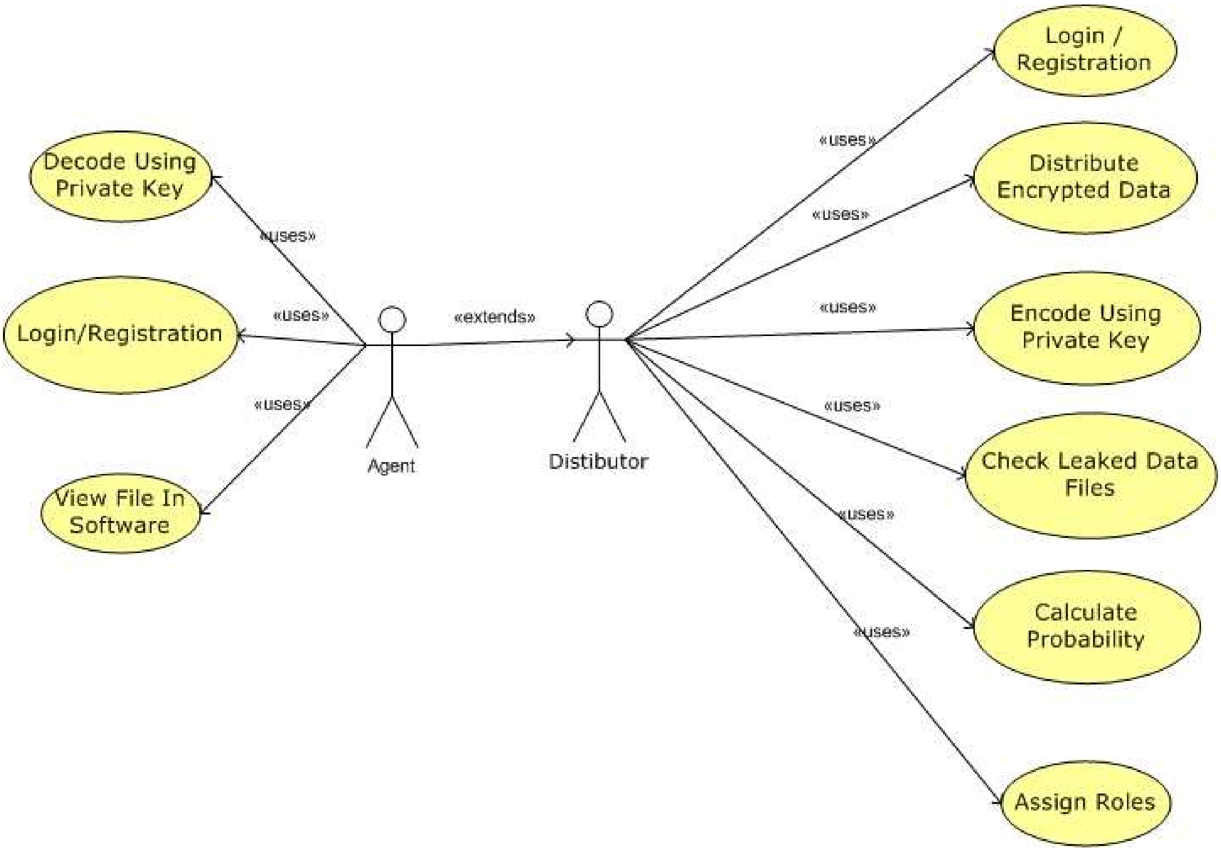


Fig 4.2 Use case Diagram

4.2.2 Sequence diagram

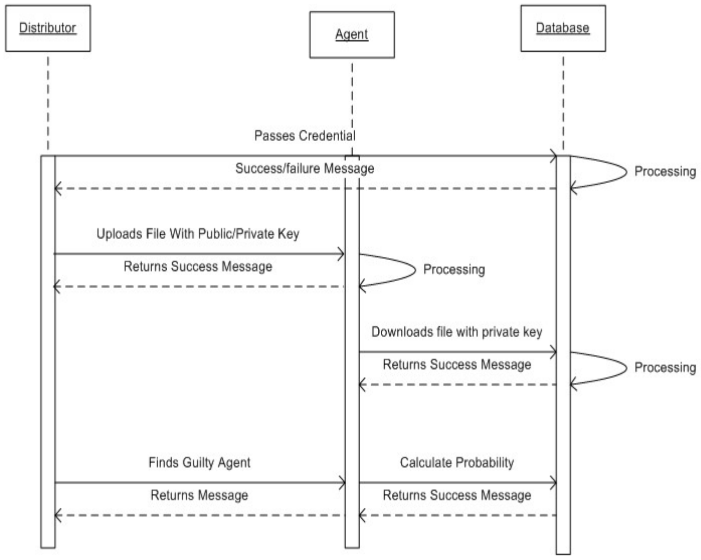


FIG 4.3: Sequence Diagram

4.2.3 Activity Diagram

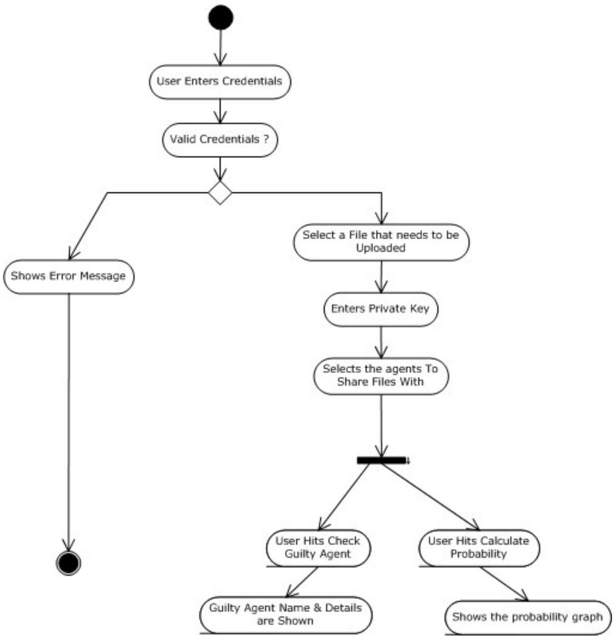

User Enters Credentials
Valid Credentials ?
Shows Error Message
Select a File that needs to be Uploaded
Enters Private Key
Selects the agents To Share Files With
User Hits Check Guilty Agent
User Hits Calculate Probability
Guilty Agent Name & Details are Shown
Shows the probability graph

FIG 4.4: Activity Diagram

4.2.3 Component Diagram

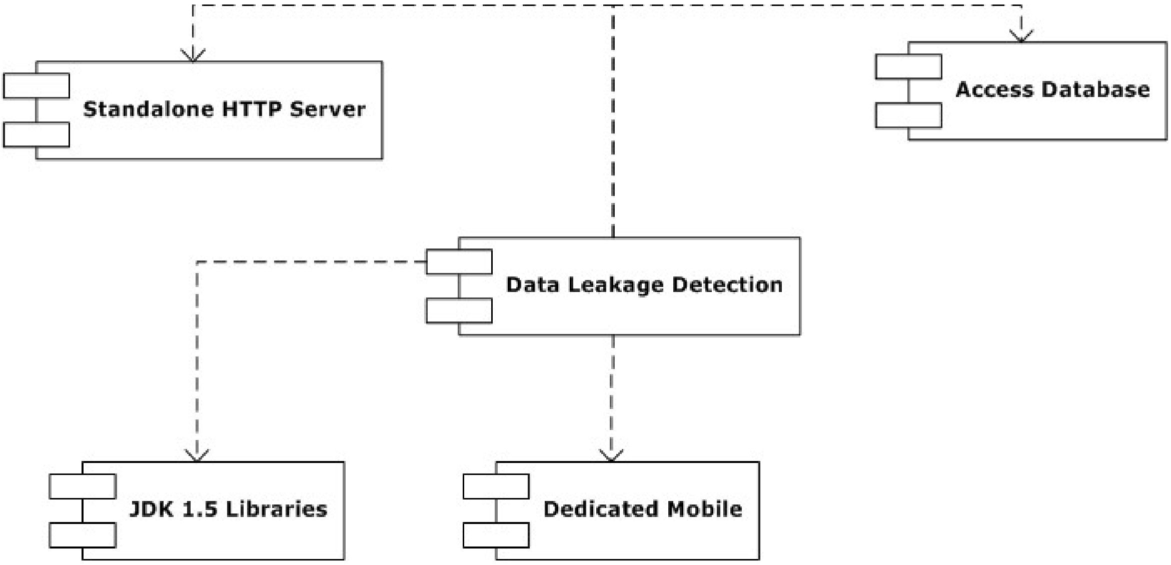


Fig 4.5 Component Diagram

4.2.4 Work Flow Diagram

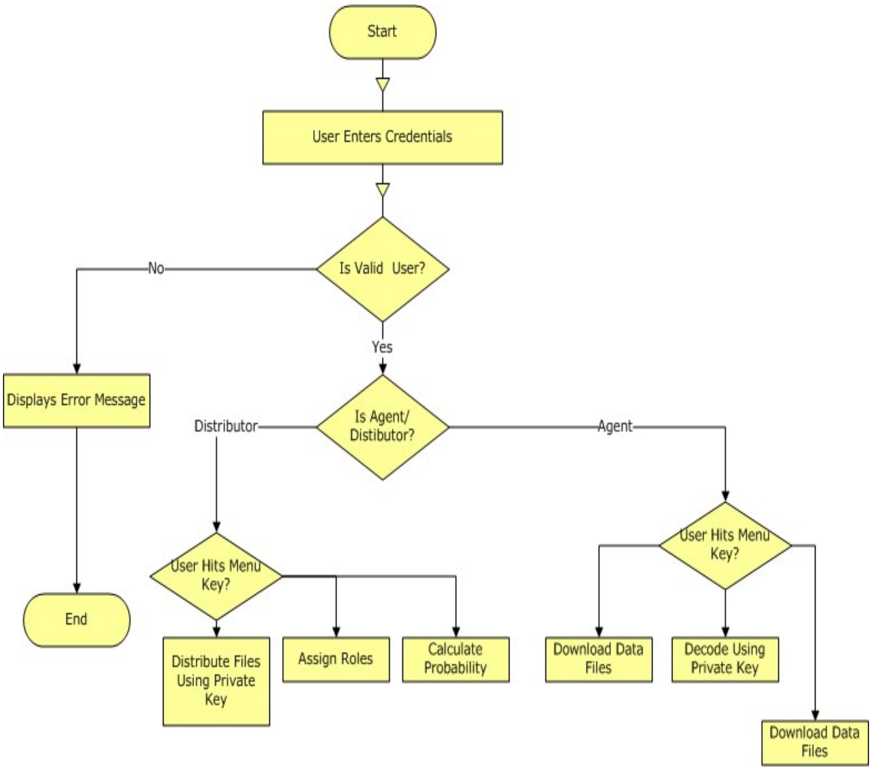


Fig 4.6 Work Flow Diagram

CHAPTER 5

TECHNICAL SPECIFICATIONS

5.1 TECHNOLOGY USED IN THE PROJECT

5.1.1 MYSQL

MySQL is the world's most used relational database management system (RDBMS) that runs as a server providing multi-user access to several databases. The SQL phrase stands for Structured Query Language. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open-source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python"

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools or download MySQL front-ends from various parties that have developed desktop software and web applications to manage MySQL databases, build database structures, and work with data records.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium-scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open-source tools such as phpMyAdmin.

5.1.2 XAMPP SERVER

XAAMP is a free and open-source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

Installing XAMPP takes less time than installing each of its components separately. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another.

It is offered in both a full, standard version and a smaller version.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. In practice, however, XAMPP is sometimes used to serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others.

5.1.3 PHP

PHP, which stands for Hypertext Preprocessor, is a widely used server-side scripting language primarily designed for web development. It is a powerful and versatile language that is particularly well-suited for creating dynamic and interactive websites. PHP code is embedded directly into HTML documents and executed on the server, generating dynamic content that is then sent to the client's web browser.

One of the key features of PHP is its ability to interact with databases, making it an ideal choice for building dynamic websites that require data storage and retrieval. PHP supports a wide range of database systems, including MySQL, PostgreSQL, and SQLite, allowing developers to create dynamic web applications with complex data-driven functionality.

PHP is also known for its ease of use and flexibility, with a simple and intuitive syntax that is easy to learn and understand. It offers a wide range of built-in functions and libraries for common tasks such as file handling, string manipulation, and form processing, reducing the need for developers to write code from scratch.

In addition to its role in server-side scripting, PHP can also be used for command-line scripting and standalone applications. It is supported by a large and active community of developers, who contribute to its ongoing development and maintenance through open-source collaboration.

Overall, PHP is a powerful and versatile language that is widely used in web development for creating dynamic, interactive, and data-driven websites and web applications. Its simplicity, flexibility, and wide range of features make it a popular choice among developers for building a wide variety of web-based projects.

5.2 Coding Files

1. HTML

```
<!DOCT
YPE
html>
            <html lang="en">
            <head>
            <meta charset="utf-8">
            <title>BizPage Bootstrap Template</title>
            <meta content="width=device-width, initial-scale=1.0" name="viewport">
            <meta content="" name="keywords">
            <meta content="" name="description">
            <!-- Favicons -->
            <link href="img/favicon.png" rel="icon">
            <link href="img/apple-touch-icon.png" rel="apple-touch-icon">
            <!-- Google Fonts -->
            <link
            href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,700
            ,700i|Montserrat:300,400,500,700" rel="stylesheet">
            <!-- Bootstrap CSS File -->
            <link href="lib/bootstrap/css/bootstrap.min.css" rel="stylesheet">
            <!-- Libraries CSS Files -->
            <link href="lib/font-awesome/css/font-awesome.min.css" rel="stylesheet">
            <link href="lib/animate/animate.min.css" rel="stylesheet">
            <link href="lib/ionicons/css/ionicons.min.css" rel="stylesheet">
            <link href="lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
            <link href="lib/lightbox/css/lightbox.min.css" rel="stylesheet">
            <!-- Main Stylesheet File -->
            <link href="css/style.css" rel="stylesheet">
            <!-- =======================================================
            Theme Name: BizPage
            Theme URL: https://bootstrapmade.com/bizpage-bootstrap-business-template/
            Author: BootstrapMade.com
            License: https://bootstrapmade.com/license/
            ======================================================= -->
            </head>
            <body>
            <!--==========================
            Header
            ============================-->
            <header id="header">
            <div class="container-fluid">
            <div id="logo" class="pull-left">
            <h1><a href="#intro" class="scrollto">BizPage</a></h1>
            <!-- Uncomment below if you prefer to use an image logo -->
            <!-- <a href="#intro"><img src="img/logo.png" alt="" title="" /></a>-->
            </div>
            <nav id="nav-menu-container">
            <ul class="nav-menu">
            <li class="menu-active"><a href="#intro">Home</a></li>
            <li><a href="#about">File</a></li>
            <li><a href="#services">Agent</a></li>
            <li><a href="#portfolio">ChangePassword</a></li>
            <li><a href="#team">Logout</a></li>
            </nav><!-- #nav-menu-container -->
            </div>
            </header><!-- #header -->
            <!--==========================
            Intro Section
            ============================-->
            <section id="intro">
            <div class="intro-container">
            <div id="introCarousel" class="carousel slide carousel-fade" data-
            ride="carousel">
            <ol class="carousel-indicators"></ol>
            <div class="carousel-inner" role="listbox">
            <div class="carousel-item active">
            <div class="carousel-background">
            <!<img src="img/intro-carousel/123.jpeg" alt=""> ></div>
            <div class="carousel-container">
            <div class="carousel-content">
            <form id="form_id" method="post" name="myform" >
            <label><h3>Sharing Details :</h3></label><br><br>
            <label>Data Request Description:</label>
            <input type="text" name="username" id="username" required/>
            <i class="validation"><br><br>
            <label>Select Region :</label>
            <input type="password" name="password" id="password" required/><br><br>
```

```
<label>Select Destributor :</label>
<input type="password" name="password" id="password" required/>
<br><br><br>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</section><!-- #intro -->
<div class="container">
<div class="copyright">
&copy; Copyright <strong>BizPage</strong>. All Rights Reserved
</div>
<div class="credits">
<!--
All the links in the footer should remain intact.
You can delete the links only if you purchased the pro version.
Licensing information: https://bootstrapmade.com/license/
Purchase the pro version with a working PHP/AJAX contact form:
https://bootstrapmade.com/buy/?theme=BizPage
-->
Designed by <a href="https://bootstrapmade.com/">BootstrapMade</a>
</div>
</div>
</footer><!-- #footer -->
<a href="#" class="back-to-top"><i class="fa fa-chevron-up"></i></a>
<!-- Uncomment below i you want to use a preloader -->
<!-- <div id="preloader"></div> -->
<!-- JavaScript Libraries -->
<script src="lib/jquery/jquery.min.js"></script>
<script src="lib/jquery/jquery-migrate.min.js"></script>
<script src="lib/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="lib/easing/easing.min.js"></script>
<script src="lib/superfish/hoverIntent.js"></script>
<script src="lib/superfish/superfish.min.js"></script>
<script src="lib/wow/wow.min.js"></script>
<script src="lib/waypoints/waypoints.min.js"></script>
```

```
<script src="lib/counterup/counterup.min.js"></script>
<script src="lib/owlcarousel/owl.carousel.min.js"></script>
<script src="lib/isotope/isotope.pkgd.min.js"></script>
<script src="lib/lightbox/js/lightbox.min.js"></script>
<script src="lib/touchSwipe/jquery.touchSwipe.min.js"></script>
<!-- Contact Form JavaScript File -->
<script src="contactform/contactform.js"></script>
<!-- Template Main Javascript File -->
<script src="js/main.js"></script>
</body>
</html>
```

2. PHP

```
<?php
   // Enter your host name, database username, password, and database
name.
   // If you have not set the database password on localhost then set it empty.
   $con = mysqli_connect("localhost","root","","myproject");
   // Check connection
   if (mysqli_connect_errno()){
      echo "Failed to connect to MySQL: " . mysqli_connect_error();
   }
?>
```

3. HTML/PHP

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8"/>
    <title>Registration</title>
    <link rel="stylesheet" href="css/style1.css"/>
</head>
<body>
<?php
    require('db.php');
    // When form submitted, insert values into the database.
    if (isset($_REQUEST['LoginId'])) {
        // removes backslashes
        $LoginId = stripslashes($_REQUEST['LoginId']);
        //escapes special characters in a string
        $LoginId = mysqli_real_escape_string($con, $LoginId);
        $UserPassword = stripslashes($_REQUEST['UserPassword']);
        $UserPassword    = mysqli_real_escape_string($con, $UserPassword);
        $FName = stripslashes($_REQUEST['FName']);
        $FName = mysqli_real_escape_string($con, $FName);

                  $LName = stripslashes($_REQUEST['LName']);
        //escapes special characters in a string
        $LName = mysqli_real_escape_string($con, $LName);
        $CellNumber    = stripslashes($_REQUEST['CellNumber']);
        $CellNumber    = mysqli_real_escape_string($con, $CellNumber);
        $EmailAddress = strip slashes($_REQUEST['EmailAddress']);
        $EmailAddress = mysqli_real_escape_string($con, $EmailAddress);
```

```
            $rollId = strip slashes($_REQUEST['rollId']);
        //escapes special characters in a string
        $rollId = mysqli_real_escape_string($con, $rollId);
        $city    = strip slashes($_REQUEST['city']);
        $city    = mysqli_real_escape_string($con, $city);
        $pincode = stripslashes($_REQUEST['pincode']);
        $pincode = mysqli_real_escape_string($con, $pincode);
        $create_datetime = date("Y-m-d H:i:s");
        $query      = "INSERT into `user info` (LoginId, UserPassword, FName, LName,
CellNumber,  EmailAddress, rolled, city, Pincode, create_datetime)
                VALUES ('$LoginId', '" . md5($UserPassword) . "', '$FName', '$LName',
'$CellNumber',  '$EmailAddress', '$rollId', '$city', '$pincode', '$create_datetime')";
        $result   = mysqli_query($con, $query);
        if ($result) {
            echo "<div class='form'>
                <h3>You are registered successfully.</h3><br/>
                <p class='link'>Click here to <a href='login.php'>Login</a></p>
                </div>";
        } else {
            echo "<div class='form'>
                <h3>Required fields are missing.</h3><br/>
                <p class='link'>Click here to <a href='registration.php'>registration</a>
again.</p>
                </div>";
        }
    } else {
?>
    <form class="form" action="" method="post">
        <h1 class="login-title">Registration</h1>
        <input type="text" class="login-input" name="LoginId" placeholder="Username"
required />
```

```
      <input type="password" class="login-input" name="UserPassword"
placeholder="UserPassword">
      <input type="text" class="login-input" name="FName" placeholder="Fast Name">
              <input type="text" class="login-input" name="LName" placeholder="Last
Name" required />
      <input type="text" class="login-input" name="CellNumber" placeholder="Phone
Number">
      <input type="text" class="login-input" name="EmailAddress" placeholder="Email">
             <input type="text" class="login-input" name="rollId" placeholder="Role Id"
required>


             <input type="text" class="login-input" name="city" placeholder="City">
      <input type="text" class="login-input" name="pincode" placeholder="Pincode">
      <input type="submit" name="submit" value="Register" class="login-button">
             <input type="Reset" value="Reset" onclick="validate()"/>
      <p class="link">Already have an account? <a href="login.php">Login here</a></p>
   </form>
<?php
   }
?>
</body>
</html>
```

CHAPTER 6

SOFTWARE TESTING

6.1 INTRODUCTION

Testing is a process used to uncover errors and ensure that defined input will produce actual results that agree with the required results. A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software.

The strategy for our project is as follows:

- Testing will begin at the component level and will work outward towards the integration of the entire system.
- Appropriate testing techniques will be used at different points in time.
- Debugging is an activity that will go hand in hand with testing.

6.2 TEST CASES

LOGIN PAGE TEST CASES:

SR.NO.	STEP DESCRIPTION	EXPECTED RESULTS
1	Go to the User ID field and without enteringdata in that field press the "Enter" key.	it should prompt the message " Please enter'User ID' "
2)	Go to the login screen enter "User ID" andwithout entering Password tries to click on the "OK" button.	it should prompt the message "Please enter'Password'".

3)	Go to the login screen enter "User ID" and enter the wrong Password tries to click on the "OK" button.	it should prompt the message " Please enter Proper 'User ID' and 'Password'".
4)	Go to the Login screen enter all required data and press "OK" Button.	The software window will open.

Table 6.1 Login Page test cases

REGISTRATION FORM TEST CASES

SR.NO.	STEP DESCRIPTION	EXPECTED RESULTS
1	enter first name and last name blank	it should prompt the message "First name andlast name should not be blank”
2)	Enter the contact no less than 10 digit	it should prompt the message "enter 10 digitcontact no”
3)	Enter not valid email ID	it should prompt the message "Enter validemail ID”
4)	Enter already existing login ID andpassword	it should prompt the message “login ID is inuse”.

Table 6.2 Registration Form Test Cases

AGENT PAGE TEST CASES:

SR.NO.	STEP DESCRIPTION	EXPECTED RESULTS
1	If the agent clicks on change password	It should open a password change window
2)	If the agent enters a new password correctly.	it should prompt the message "passwordchange successfully”.
3)	If the agent clicks on download files.	It should open a download file window.
4)	At the time of downloading files if the userenters the wrong Encryption key.	it should prompt the message “Enter correctEncryption key”.
5)	At the time of downloading files, the userenters the correct Encryption key.	File contents should be displayed in the text boxarea.
6)	If the agent does not select any file and click onthedownload button	It should prompt the message “You have notselected any file”
7)	If the agent clicks on send request	It should open a request window
8)	If the agent does not select any distributor at thetime of sending a request	It should prompt the message “You have notselected any distributor “.
9)	If the agent clicks on Log out	It should go to the login page.

Table 6.3 Agent page test cases

DISTRIBUTOR PAGE TEST CASES:

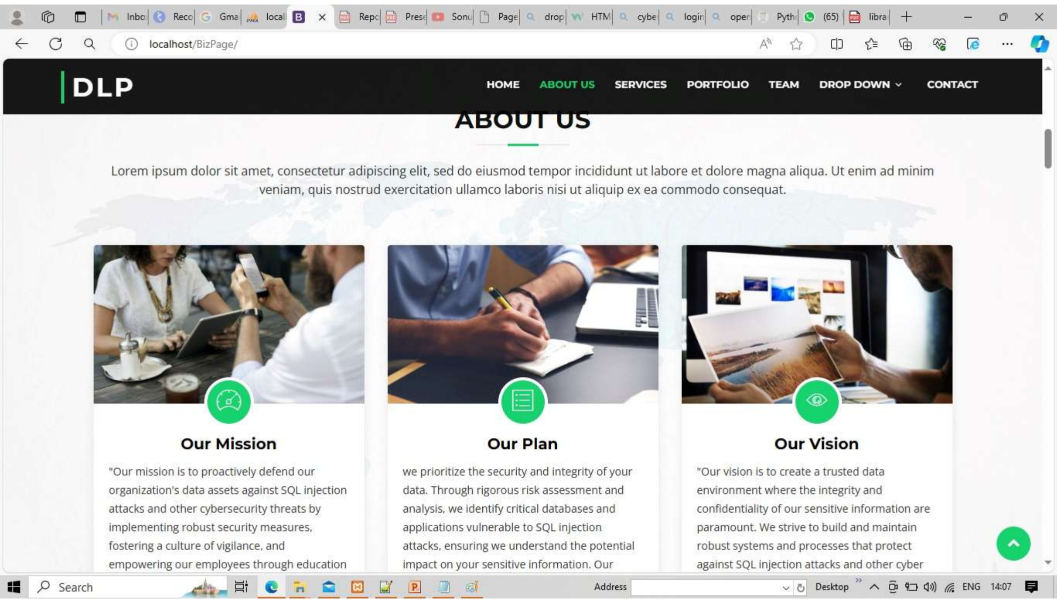
SR.NO.	STEP DESCRIPTION	EXPECTED RESULTS
1	If the distributor clicks on change password	It should open a password change window
2)	If the distributor enters the new passwordcorrectly.	it should prompt the message "passwordchange successfully”.
3)	If the distributor clicks on upload the file	It should open an upload window.
4)	If the distributor does not select an agent	It should prompt the message “Select agent”
	If the distributor does not enter the Encryption key.	It should prompt the message “EnterEncryption key”
4)	Click on the share button	The file should be shared with no error and shared file details should be displayed below.
5)	If the distributor clicks on find probability	It should open a probability window.
6)	If the distributor clicks on Log out	It should go to the login page.

Table 6.4 Distributor Form Test Cases

CHAPTER 7

RESULTS(SNAPSHOTS OF RESULTS)

1) HOME PAGE



ABOUT US

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Our Mission

"Our mission is to proactively defend our organization's data assets against SQL injection attacks and other cybersecurity threats by implementing robust security measures, fostering a culture of vigilance, and empowering our employees through education

Our Plan

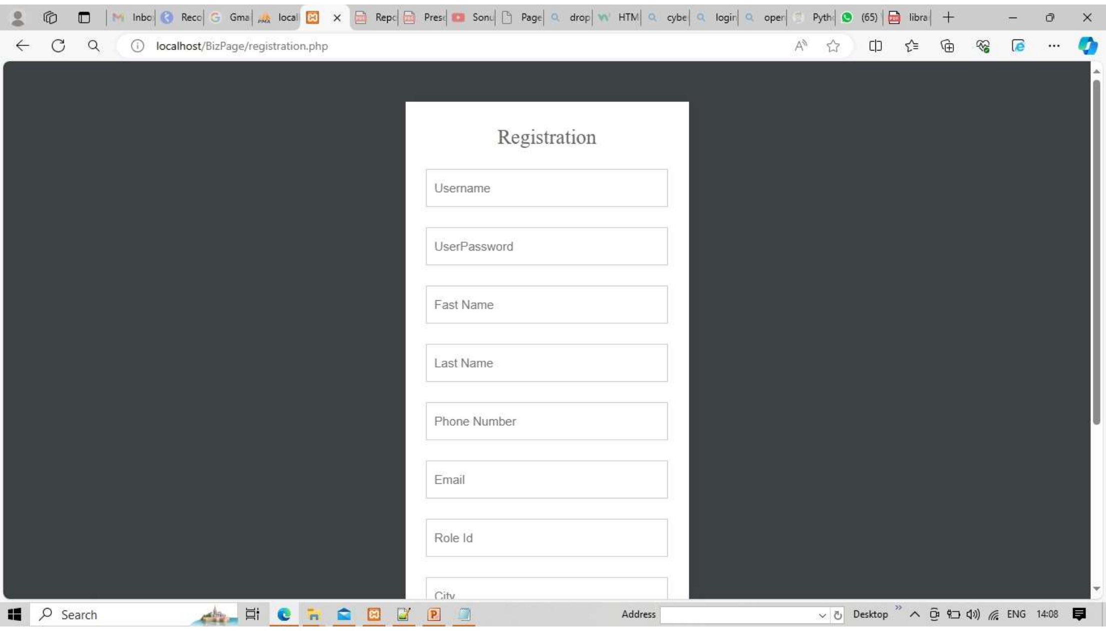
we prioritize the security and integrity of your data. Through rigorous risk assessment and analysis, we identify critical databases and applications vulnerable to SQL injection attacks, ensuring we understand the potential impact on your sensitive information. Our

Our Vision

"Our vision is to create a trusted data environment where the integrity and confidentiality of our sensitive information are paramount. We strive to build and maintain robust systems and processes that protect against SQL injection attacks and other cyber

Fig 7.1 Home Page

2) REGISTRATION PAGE



Registration

Username

UserPassword

Fast Name

Last Name

Phone Number

Email

Role Id

City

Fig 7.2 Registration Page

3) ADMINISTRATOR

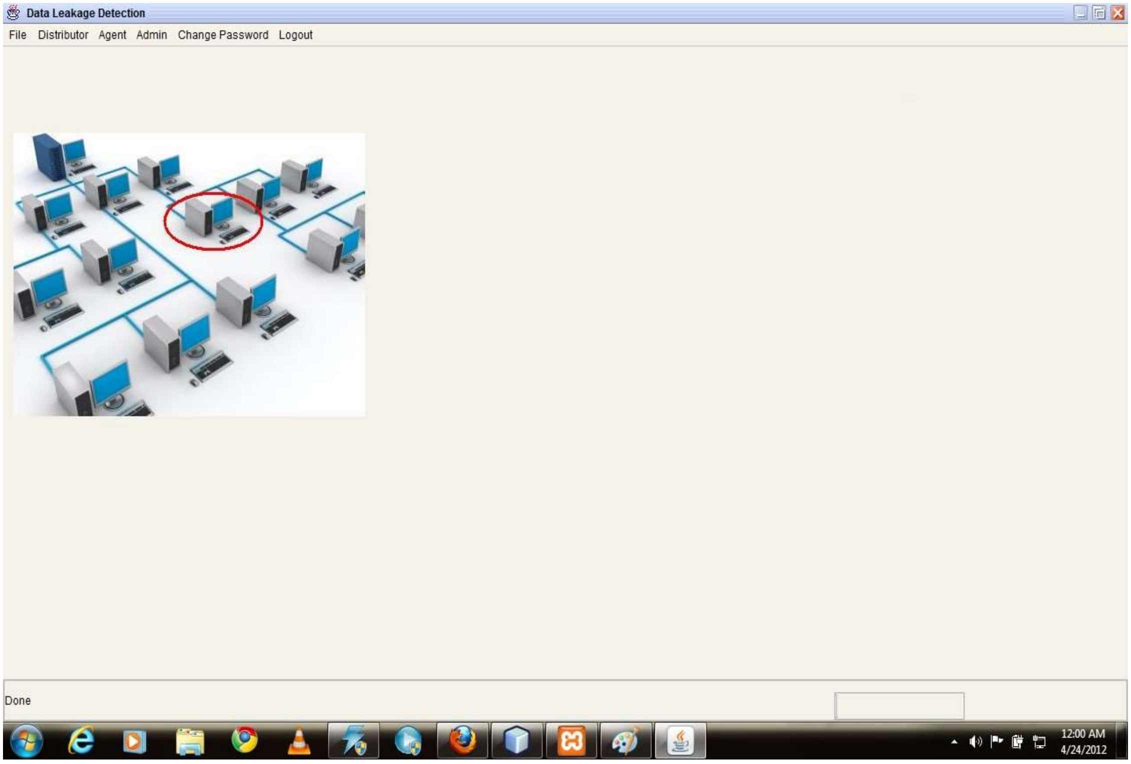
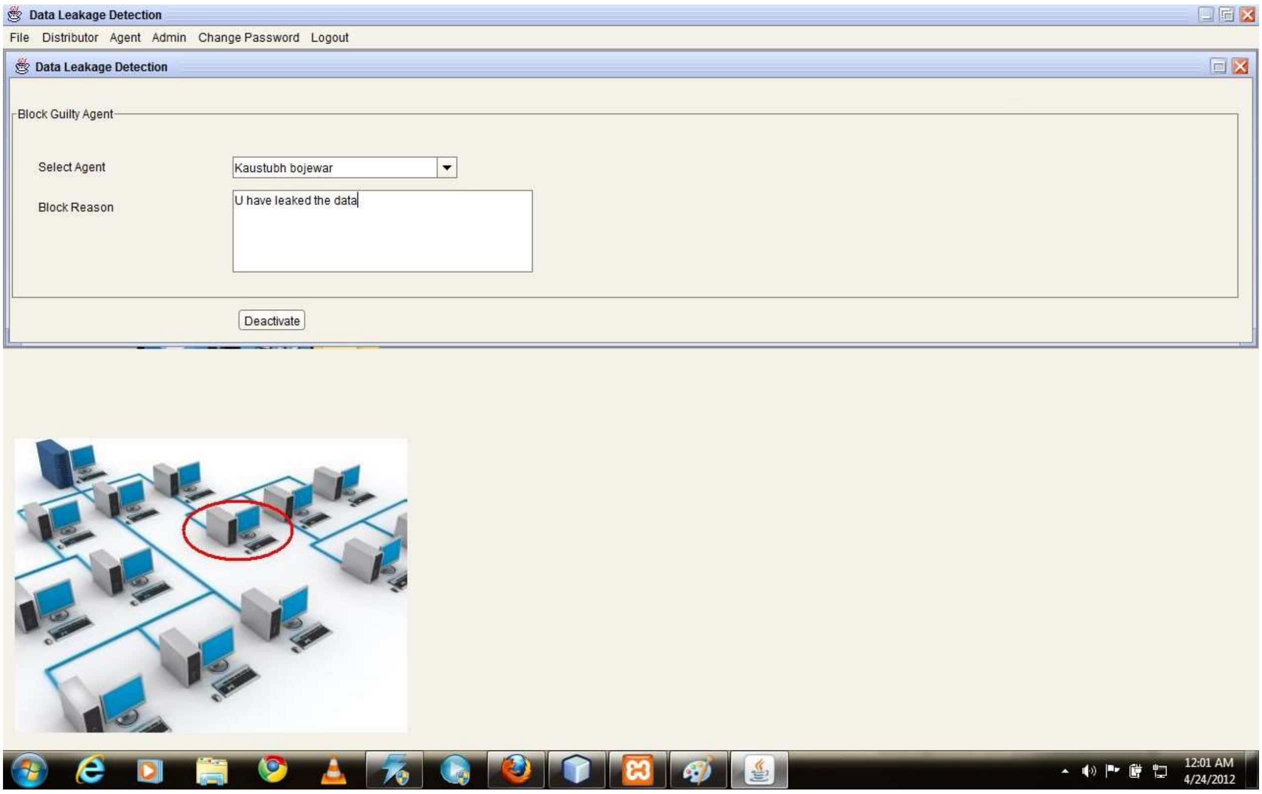


Fig 7.3 Administrator form

This form will be visible to the Administrator. As he has the right to view everything going on in the application, every menu including file, distributor, agent, admin, and changepassword will be visible to him. He can see the probability distribution table and accordingly block the guilty agent.

4) ADMINISTRATOR BLOCKING AGENT:



Block Guilty Agent

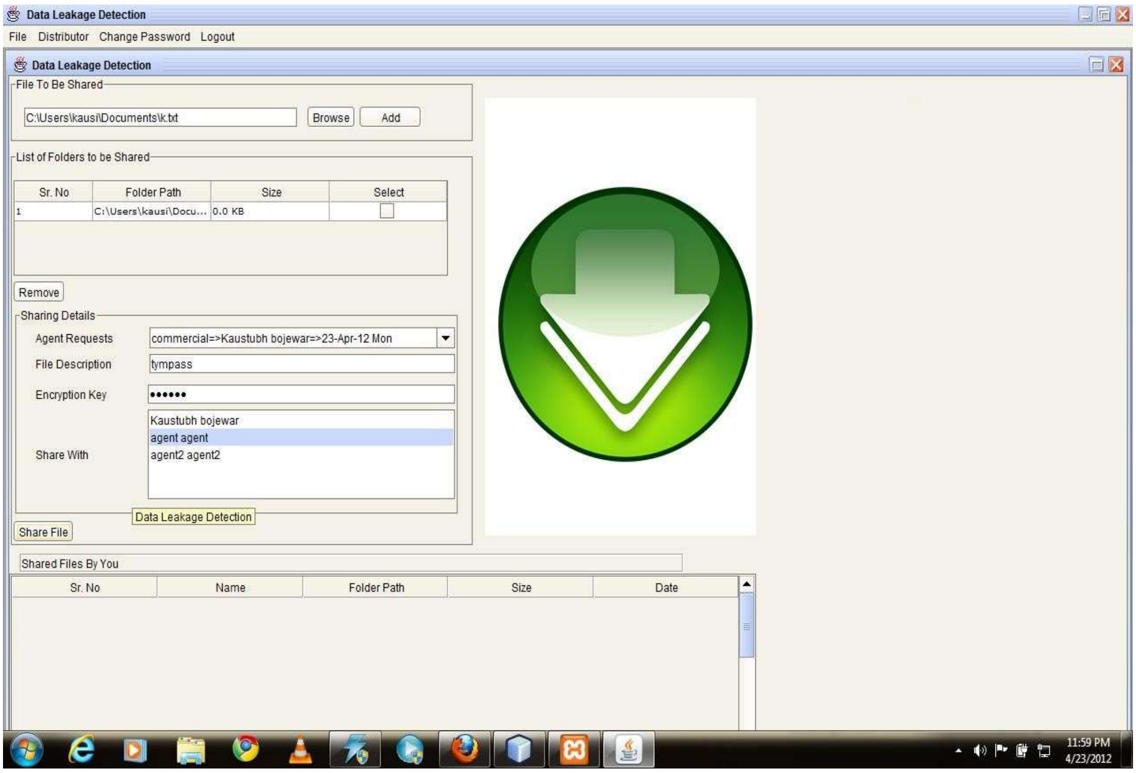
Select Agent: Kaustubh bojewar

Block Reason: U have leaked the data

Fig 7.4 Administrator blocking agent form

This form shows how the Administrator blocks the agent who has leaked the data. He willfirst select the agent from the drop-down list of the agents and in the next text box, he willgive a reason why he had blocked the agent

5) DISTRIBUTOR UPLOAD FILE:



File To Be Shared: C:\Users\kausi\Documents\k.txt

List of Folders to be Shared

Sr. No	Folder Path	Size	Select
1	C:\Users\kausi\Docu...	0.0 KB	□

Sharing Details

Agent Requests: commercial=>Kaustubh bojewar=>23-Apr-12 Mon

File Description: tympass

Encryption Key: ••••••

Share With: Kaustubh bojewar
agent agent
agent2 agent2

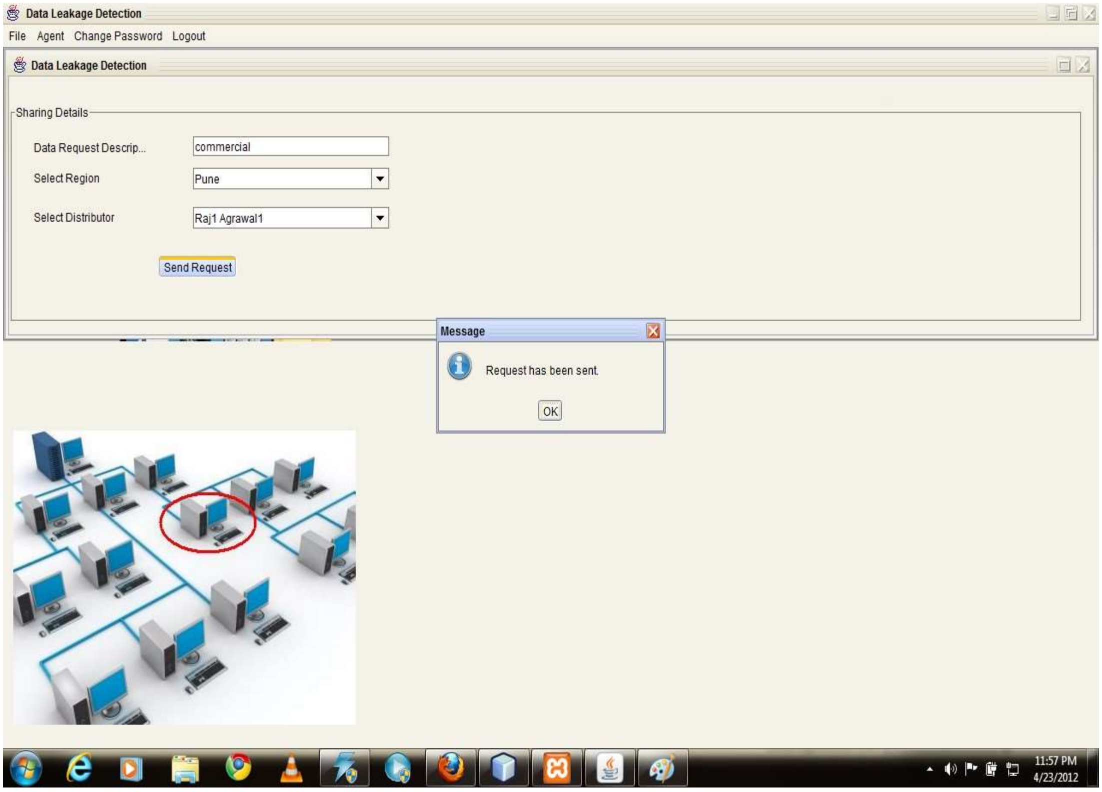
Shared Files By You

Sr. No	Name	Folder Path	Size	Date

Fig 7.5 Distributor Form

This is one of the main forms of our project. This form is for the distributor. In both the main modules distributor distributes the files to the agent. In the first algorithm, the distributor decides which data is to be sent and in the second algorithm, the agents send the request to the distributor, and accordingly distributor sends the data to the particular agent through this form.

6) AGENT SEND REQUEST



Data Request Descrip... commercial

Select Region Pune

Select Distributor Raj1 Agrawal1

Request has been sent.

Fig 7.6 **Agent Send Request**

One more thing that the agent can do is he can send the request to the distributor asking for the particular type of files. In that case, this form is used. Where he has to provide a request and send it to the distributor.

CONCLUSION AND FUTURE SCOPE

Conclusion

In a perfect world, there would be no need to hand over sensitive data to agents that may unknowingly or maliciously leak it. And even if we had to hand over sensitive data, in a perfect world we could watermark each object so that we could trace its origins with absolute certainty. However, in many cases, we must indeed work with agents that may not be 100% trusted, and we may not be certain if a leaked object came from an agent or some other source, since certain data cannot admit watermarks. Despite these difficulties, we have shown it is possible to assess the likelihood that an agent is responsible for a leak, based on the overlap of his data with the leaked data and the data of other agents, and based on the probability that objects can be “guessed” by other means. Our model is relatively simple, but we believe it captures the essential trade-offs. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor’s chances of identifying a leaker. We have shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is a large overlap in the data that agents must receive.

It includes the investigation of agent guilt models that capture leakage scenarios that are not studied in this paper. For example, what is the appropriate model for cases where agents can collude and identify fake tuples? A preliminary discussion of such a model is available. Another open problem is the extension of our allocation strategies so that they can handle agent requests in an online fashion (the presented strategies assume that there is a fixed set of agents with requests known in advance).

Future Scope

The developer of an application can never be carried out to the fullest extent in a stipulated time, the main reason why revisions of the application are always introduced over time. This application being restricted to one-time development will have no revision done, hence certain areas that can be enhanced are pointed out.

REFERENCES

[1] “Data Leakage Detection” Panagiotis Papadimitriou, Student Member, IEEE, and Hector Garcia-Molina, Member, IEEE

[2] R. Agrawal and J. Kiernan, “Watermarking Relational Databases,” Proc. 28th Int’l Conf. Very Large Data Bases (VLDB ’02), VLDB Endowment, pp. 155-166, 2002.

[3] P. Bonatti, S.D.C. di Vimercati, and P. Samarati, “An Algebra for Composing Access Control Policies,” ACM Trans. Information and System Security, vol. 5, no. 1, pp. 1-35, 2002.

[4] P. Buneman and W.-C. Tan, “Provenance in Databases,” Proc. ACM SIGMOD, pp.1171-1173, 2007.

[5] Y. Cui and J. Widom, “Lineage Tracing for General Data Warehouse Transformations,” The VLDB J., vol. 12, pp. 41-58, 2003.

[6] V.N. Murty, “Counting the Integer Solutions of a Linear Equation with Unit Coefficients,” Math. Magazine, vol. 54, no. 2, pp. 79-81,1981.

[7] A programming guide to java certification.

[8] Cryptography and Network Security by Williams Stallings.

APPENDIX A

GLOSSARY

Watermarking

A watermark is a recognizable image or pattern in paper that appears as various shades of lightness/darkness when viewed by transmitted light (or when viewed by reflected light, atop a dark background), caused by thickness or density variations in the paper.

Encryption

It is the process of converting plain text into ciphertext which is not readable in general form using the encryption algorithm for security purposes.

Class diagrams:

Class diagrams use classes and interfaces to capture details about the entities that make up your system and the static relationships between them. Class diagrams are one of the most commonly used UML diagrams, and they vary in detail from fully fleshed-out and able to generate source code to quick sketches on whiteboards and napkins.

Component diagrams:

Component diagrams show the organization and dependencies involved in the implementation of a system. They can group smaller elements, such as classes, into larger, deployable pieces. How much detail you use in component diagrams varies depending on what you are trying to show. Some people simply show the final, deployable version of a system, and others show what functionality is provided by a particular component and how it realizes its functionality internally.

Deployment diagrams:

Deployment diagrams show how your system is executed and assigned to various pieces of hardware. You typically use deployment diagrams to show how components are configured at runtime.

Activity diagrams:

Activity diagrams capture the flow from one behavior or *activity*, to the next. They are similar in concept to a classic flowchart but are much more expressive.

Sequence diagrams:

Sequence diagrams are a type of interaction diagrams that emphasize the type and order of messages passed between elements during execution. Sequence diagrams are the most common type of interaction diagram and are very intuitive to new users of UML.

Use case diagrams:

Use case diagrams to capture functional requirements for a system. They provide an implementation-independent view of what a system is supposed to do and allow the modeler to focus on user needs rather than realization details