

**In-Use Data Leakage Prevention in
Microsoft Windows Operating System Using
Naïve Bayes Classifier**



By

Danish Alam

Supervised By

Dr. Muhammad Usman

Department of Computer Sciences

Quaid-i-Azam University

Islamabad

(2014 - 2018)

Dedicated to my parents, friends and all students.

ACKNOWLEDGEMENT

This thesis is about development of “Data Leakage Prevention in Windows Operating System Using Naïve Bayes Classifier”, a distributed desktop application for windows operating system. It has been a challenging but great learning experience. It leads to the understanding of basics of machine learning, python language, handling printing tasks and apply remedial action against printing request.

First of all, I would like to express my gratefulness to Allah Almighty, whose blessings I have seen during my time at Quaid-i-Azam University in the form of achievements, success and learning. In addition, I am obligated to a number of people for their sustenance and generous contribution. I especially appreciate my supervisor Dr. Muhammad Usman for guiding me throughout the process. I am also thankful to my friends for encouraging me for completion of this distributed desktop application.

ABSTRACT

The product defines the distributed desktop application known as “In-use Data Leakage Prevention in Microsoft Windows Operating System using Naïve Bayes”. The purpose of the system is to ensure the security of organizational data, which cannot be revealed to outsiders through organization’s employees. In this work, we focused on printing device (printer) through which an important data can print by authorized user and took the printed-paper with him/her outside the organization.

This product performs three main operations based upon data classes, which a printed document belongs. If system finds that the printing document belongs to secret class, then it can cancel the printing request. If system finds that the printing document belongs to confidential class, then system pause the printing request temporarily and ask a question about this printing request from admin, whether admin should allow this request or not. If admin allows this request, system change the status of this printing request from pause to resume else system can cancel the printing request. If system finds that the printing document belongs to non-confidential class, system cannot take any action because all documents belongs to this class are not much important. The system stores all records in database related to printing request. This system is implemented in python language and Tkinter framework is used in this work. All the work is done over client server architecture.

Table of Contents

List of Figures	i
List of Tables	ii
Chapter 1	1
Introduction.....	1
1.1 Problem Definition.....	1
1.2 Proposed Solution	2
1.3 Scope.....	3
1.4 Objectives	3
1.5 Project Organization	3
1.5.1 Software Process Model.....	3
1.5.3 Tools and Techniques	4
1.6 Project Management Plan	4
1.7 Report Structure	6
Chapter 2	8
Requirements Gathering and Analysis	8
2.1 Product Overview	8
2.2 Major Functions	8
2.3 Major Inputs and outputs	8
2.3.1 Major Inputs.....	8
2.3.2 Major Outputs	8
2.4 Definitions, Acronyms and Abbreviations.....	9
2.5 Overview	9
2.6 User Characteristics	9
2.7 Constraints	9
2.8 Assumptions and Dependencies.....	9
2.9 Specific Requirements	10
2.9.1 Functional Requirements	10
2.9.2 External Interface Requirement	10
2.9.3 User Interfaces	10
2.10 Software System Attributes	11
2.10.1 Reliability.....	11
2.10.2 Availability	11
2.10.3 Security	11

2.10.4 Maintainability	12
2.10.5 Portability.....	12
2.10.6 Performance	12
2.11 Product Functions	12
2.11.1 Analyse System.....	12
2.12 Use Case Diagram.....	12
2.13 Use Case Description.....	14
2.13.1 Use case 1: Signup	14
2.13.2 Use case 2: Login.....	15
2.13.3 Use case 3: Configure Parameters	15
2.13.4 Use case 4: Run Agent.....	16
2.13.5 Use case 5: Account Setting.....	16
2.13.6 Use case 6: Search Leak Records	17
2.13.7 Use case 7: Export leak records	17
2.14 Domain Model	18
2.15 Database Requirements.....	19
Chapter 3	21
Software Design Description	21
3.1 Introduction.....	21
3.1.1 Design Overview.....	21
3.1.2 Requirement Traceability Matrix.....	21
3.2 System Architecture Design.....	22
3.2.1 Chosen System Architecture	22
3.2.2 System Interface Description	23
3.3 User Interface Design	23
3.3.1 Description of User Interface Design.....	24
3.4 Sequence Diagram	27
3.4.1 Signup	27
3.4.2 Login	28
3.4.2 Configure Parameters.....	29
3.4.3 Run Agent	29
3.4.4 Account Setting.....	30
3.4.5 Search Leak Records.....	30
3.4.5 Export Leak Records.....	31

3.5 Class Diagram.....	31
Chapter 4	33
Software Implementation Document	33
4.1 Introduction.....	33
4.2 Framework Selection	33
4.3 Language Selection.....	33
4.4 Operating System.....	33
4.5 Algorithms and techniques.....	33
4.5.1 CountVectorizer.....	34
4.5.2 Term Frequency Inverse Document Frequency	34
4.5.3 Multinomial Naïve Bayes	34
4.5.4 Win32print Module.....	34
4.6 Data Classification	34
4.7 System Flow.....	35
4.8 Desktop Application Screenshots	36
4.8.1 Login Interface.....	36
4.8.2 Login Interface with Empty Entry Boxes	37
4.8.3 Login Interface with wrong input	38
4.8.4 Signup Interface	39
4.8.5 Signup Interface Successfully.....	40
4.8.6 Configure Parameter Interface	41
4.8.7 Leaked Records Interface.....	42
4.8.8 Reports Interface.....	43
4.8.9 Account Setting Interface.....	44
4.8.10 Secret Leak prevention Interface	45
4.8.11 Confidential Leak prevention Interface	46
4.8.12 Non-Confidential Leak prevention Interface	47
4.8.13 Client Interface.....	48
Chapter 5	50
Software Test Document	50
5.1 Introduction.....	50
5.1.1 Test Approach.....	50
5.2 Test Plan.....	50
5.2.1 Features to be tested.....	51

5.2.2 Testing Tools and Environment.....	51
5.3 Test Cases	51
5.3.1 Signup	51
5.3.2 Login.....	52
5.3.3 Configure parameters.....	52
5.3.4 Account Setting.....	53
5.3.5 Export Leaked Reports.....	53
5.3.6 Correct detection of data class	53
5.3.7 Capture Print Operation	54
5.3.8 Cancel Print Operation.....	54
5.3.9 Pause Print Operation.....	55
5.3.10 Resume Print Operation	55
Chapter 6	57
Conclusion and Future Enhancements	57
6.1 Summary	57
6.2 Future Enhancements.....	57
References.....	59

List of Figures

Figure 1. 1 Tabular view of the project management plan	5
Figure 1. 2 Timeline view of the project management plan	6
Figure 2. 1 Use Case Diagram	13
Figure 2. 2 Domain Model.....	18
Figure 2. 3 Entity Relationship Diagram	19
Figure 3. 1 Architecture Diagram	23
Figure 3. 2 Login Interface	24
Figure 3. 3 Configure Parameters Interface.....	25
Figure 3. 4 Run Agent Interface	25
Figure 3. 5 Account Setting Interface	26
Figure 3. 6 Generate Reports Interface	26
Figure 3. 7 Signup Sequence Diagram	27
Figure 3. 8 Admin Login Sequence Diagram	28
Figure 3. 9 Configure Parameters Sequence Diagram.....	29
Figure 3. 10 Run Agent Sequence Diagram	29
Figure 3. 11 Account Setting Sequence Diagram.....	30
Figure 3. 12 Search Reports Sequence Diagram	30
Figure 3. 13 Export Leak Records Sequence Diagram.....	31
Figure 3. 14 Class Diagram	32
Figure 4. 1 System Flow Diagram	35
Figure 4. 2 Login Interface	36
Figure 4. 3 Login Interface with empty boxes.....	37
Figure 4. 4 Login Interface with wrong input.....	38
Figure 4. 5 Signup Interface.....	39
Figure 4. 6 Signup with correct inputs.....	40
Figure 4. 7 Configure Parameter Interface	41
Figure 4. 8 Leaked Records Interface	42
Figure 4. 9 Reports Interface	43
Figure 4. 10 Account Setting Interface.....	44
Figure 4. 11 Secret Leak Prevention Interface.....	45
Figure 4. 12 Confidential Leak Prevention Interface.....	46
Figure 4. 13 Non-Confidential Leak Interface.....	47
Figure 4. 14 Client Interface	48

List of Tables

Table 1. 1 Abbreviations.....	9
Table 2. 1 Signup Use case.....	14
Table 2. 2 Login Use case.....	15
Table 2. 3 Configure Parameters Use case.....	15
Table 2. 4 Run Agent Use case.....	16
Table 2. 5 Account Setting Use case.....	16
Table 2. 6 Search Leak Records Use case.....	17
Table 2. 7 Export leak records Use case.....	17
Table 3. 1 Requirement Traceability Matrix.....	22
Table 4. 1 Data Set Details.....	35
Table 5. 1 Signup Test Case.....	51
Table 5. 2 Login Test Case.....	52
Table 5. 3 Configure Parameters Test Case.....	52
Table 5. 4 Account Setting Test Case.....	53
Table 5. 5 Export Leaked Reports Test Case.....	53
Table 5. 6 Correct detection of data class Test Case.....	53
Table 5. 7 Capture Print operation Test Case.....	54
Table 5. 8 Cancel print operation Test Case.....	54
Table 5. 9 Pause print operation Test Case.....	55
Table 5. 10 Resume print operation Test Case.....	55

“Education is not preparation for life; education is life itself.” John Dewey

Chapter 1

Introduction

This chapter first introduces the Data Leakage Prevention (DLP) in windows operating system using Naïve Bayes. It highlights the problem that has been addressed in this work along with the designed and developed solution. It also elaborates project organization and project planning. Finally, this chapter explains the scope and objectives of this project.

1.1 Problem Definition

In general, there are numerous ways through which sensitive data can be revealed to untrusted third parties. According to recent reports, such leaks are incremental in relation to their size and impact [1]. For example, one hacker leaked the account details of over 77 million PlayStation network subscribers, resulting in a total shutdown of PlayStation network services for several weeks and a public apology from the CEO of Sony. Similarly, the giant online shopping website eBay suffered from one of the biggest recorded leaks in history when the names, emails addresses, and personal information of more than 145 million customers were stolen. This disrupted the website’s operations and forced customers to initiate a major account password reset process. Furthermore, in 2010 hundreds of thousands of top secret United States diplomatic cables and military reports were released to the public by an insider and posted on Wikileaks. Websites such as Pastebin and 4chan allowed confidential data, including stolen credit cards details and personal photos to be anonymously spread once leaked [2]. Such data leakage incidents can negatively affect governments, organizations, and individuals. It is crucial to identify sensitive data repositories within an organization. The leaking of confidential data to unauthorized entities can result in various problems for organizations and individuals.

There are three data states, namely, *in use*, *at rest*, and *in transit* that are relevant to data leakage. For data in use case, confidential data can be leaked through channels such as USB port, CD drives, and printed documents. In the case of data at rest, data are in the form of back-up

files, databases, document management systems, and other content repositories. Finally, leaking channels associated with data in transit, such as web services and file sharing, might be extremely challenging, since these channels may be business pre-requisites. To ensure maximum security for data passing through these channels, extensive traffic filtering must be carried out. In order to discuss the data leakage prevention problem, we investigate several factors including data repositories and available data leak channels, i.e., the case of data *in use* is investigated in this work.

1.2 Proposed Solution

A system, namely, DLP is designed and developed in this work which provides monitoring of confidential data in a private network when data is in use. The developed DLP solution accomplishes through the use of a software program known as an agent, which is controlled by the central management capabilities of the overall DLP solution. It categorizes organizational data. The data is classified on the basis of three categories as secret, confidential, and non-confidential. Data is termed as secret when its illegitimate leak would cause serious damage to an organization. This type of data requires special protection. Such as a user wants to print a secret document of the organization, it will restrict to complete that operation. Confidential data is based on the content that is considered as sensitive by organization and that require protection. Manipulation to such data is restricted to irrelevant people. This kind of protection is established through the verification of threat category and user who requests for the printing document intentionally or unintentionally. Non-confidential data is type of data which is public and have no sensitive information. Therefore, such data do not require any special protection such as its screening. DLP solution works for each data classification level. In the case of secret data, the DLP agent would also detect the data leak performed by an authorized user. The agent terminates the intended operation (e.g. print) and sends an alert message to the administrator along with printing details. In the case of confidential data, the DLP agent also detects the data leak caused due to print operation by an authorized user. In this case, the agent checks the user and threat category to confirm that whether the file is allowed to access or not. If admin grants permission to intended user for printing to requested file then it cannot be restricted from the intended operation; otherwise, it cancel the intended operation. In the case of non-confidential data, a DLP agent can detect the data category which is performed by an authorized user and it sends a

message to the administrator about the data category of intended file including details related to printing task.

This system is implemented in Python languages. The application uses xamp server as a database to log leaked information. Administrator can use this logged information to generate reports such as the summary of the leaked details to enhance the internal security of an organization.

1.3 Scope

The functional requirements of the project are classification of data of an organization; monitoring of data as it is accessed, processed and distributed to peripheral devices; and generate alerts messages against unauthorized actions performed by insiders or illegitimate user. The non-functional requirement is detection of data leakage with higher accuracy within reasonable amount of time.

1.4 Objectives

The primary objective of DLP is to prevent the leakage of data. The system is based on the central security categories which identify, monitor, and protect data in-use through data analysis, and performs the remedial actions against an illegitimate user whose information is logged in xamp server.

1.5 Project Organization

1.5.1 Software Process Model

The Spiral process model is used to develop DLP. The spiral model is similar to the incremental model. The spiral model has four phases: Planning, Risk Analysis, Engineering, and Evaluation.

Planning Phase: The system requirements are gathered from the co-principal investigators of the sponsored organization in more than four meetings. Some of the required requirements are monitoring of data, classification of data, detection of data leakage, and prevention of data from leaks with high accuracy in reasonable amount of time.

Risk Analysis: In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype will be produced at the end of the risk analysis phase. If any risk will be found during the risk analysis phase, then alternate solutions will be explored and implemented.

Engineering Phase: The software is developed along with testing at the end of the phase. The objective of this phase is to provide a working solution of DLP.

Evaluation phase: This phase allows the customers to evaluate the output of the project before the project moves to the next spiral. The evaluation is done after the development of the product.

1.5.2 Roles and Responsibilities

This is an industry sponsored project. The roles and responsibilities have been divided into project sponsors and among the team who is working on this project. The responsibilities for the sponsor are budget allocation, provision of project objectives, making major decisions of the project, identifying the scope of project, and providing feedbacks. Responsibilities of the principal investigator and co-principal investigators are to provide technical guidance, conduct of the study effort about the sponsored project, ensuring compliance with the terms and conditions of the sponsored agreement, and managing project funds within the approved budget. The role of the student is to prepare project documentation and develop the software product.

1.5.3 Tools and Techniques

The tools that are used for the implementation of this system are PyCharm IDE, Argo UML, Xamp Server and Microsoft Visio for UML diagrams such as use case diagrams, class diagrams, and domain model. Microsoft Word is used for documentation write-up. For designing a plan of the system, project libre is used. This system is implemented in Python version 3.6.4 (frame work tkinter).

1.6 Project Management Plan

The tabular and timeline views of project plans are given in Figures 1.1 and 1.2 on the next pages.

	Name	Duration	Start	Finish
	Documentation	203.5 days?	9/15/17 2:00 PM	11/27/17 1:00 PM
	Chapter 1:Project Introduction	85 days?	9/15/17 2:00 PM	10/16/17 4:00 PM
	Introduction	2 days	9/15/17 2:00 PM	9/18/17 9:00 AM
	problem Definition	1 day	9/19/17 4:00 PM	9/20/17 9:00 AM
	Proposed Solution	1 day	9/20/17 4:00 PM	9/21/17 9:00 AM
	Scope	1 day	9/22/17 4:00 PM	9/25/17 9:00 AM
	Objective	1 day	9/25/17 4:00 PM	9/26/17 9:00 AM
	Meeting	1 day	10/2/17 2:00 PM	10/2/17 4:00 PM
	Project Organization	17 days	10/3/17 4:00 PM	10/10/17 9:00 AM
	Software Process Model	1 day	10/3/17 4:00 PM	10/4/17 9:00 AM
	Roles and Responsibilities	1 day	10/5/17 4:00 PM	10/6/17 9:00 AM
	Tools and Techniques	1 day	10/9/17 4:00 PM	10/10/17 9:00 AM
	Project Management Plan	2 days	10/11/17 4:00 PM	10/12/17 11:00 AM
	Meeting	1 day?	10/16/17 2:00 PM	10/16/17 4:00 PM
	Chapter 2:Requirements Gathering and Analysis	59.5 days?	10/17/17 8:00 AM	11/6/17 4:00 PM
	Introduction	59.5 days?	10/17/17 8:00 AM	11/6/17 4:00 PM
	Purpose	0.25 days	10/17/17 8:00 AM	10/17/17 8:30 AM
	Stakeholders	0.25 days	10/17/17 4:00 PM	10/17/17 4:30 PM
	Major Function	0.25 days	10/17/17 6:00 PM	10/18/17 8:30 AM
	Supported Functions	0.25 days	10/18/17 9:00 AM	10/18/17 9:30 AM
	Major Inputs and Major Outputs	0.25 days	10/18/17 4:00 PM	10/18/17 4:30 PM
	Meeting	1 day?	10/23/17 2:00 PM	10/23/17 4:00 PM
	Overview	19.5 days?	10/24/17 8:00 AM	10/30/17 4:00 PM
	Overall Discription	0.25 days	10/24/17 8:00 AM	10/24/17 8:30 AM
	Product Perspective	0.25 days	10/24/17 3:00 PM	10/24/17 3:30 PM
	Product Function	0.25 days	10/25/17 8:00 AM	10/25/17 8:30 AM
	User Characteristic	0.25 days	10/25/17 9:00 AM	10/25/17 9:30 AM
	Constraints	0.25 days	10/25/17 3:00 PM	10/25/17 3:30 PM
	Assumptions and Dependencies	0.25 days	10/25/17 5:00 PM	10/26/17 8:30 AM
	Meeting	1 day?	10/30/17 2:00 PM	10/30/17 4:00 PM
	Specific Requirments	19.5 days?	10/31/17 8:00 AM	11/6/17 4:00 PM
	Functional Requirments	1 day	10/31/17 8:00 AM	10/31/17 10:00 AM
	Non-Functional Requirements	1 day	11/1/17 8:00 AM	11/1/17 10:00 AM
	Use Case Diagram	1 day	11/2/17 8:00 AM	11/2/17 10:00 AM
	Use Cases Description	1 day	11/3/17 8:00 AM	11/3/17 10:00 AM
	Meeting	1 day?	11/6/17 2:00 PM	11/6/17 4:00 PM
	Chapter 3: System Design	23.5 days?	11/7/17 8:00 AM	11/14/17 4:00 PM
	Architectural Diagram	1 day	11/7/17 8:00 AM	11/7/17 10:00 AM
	Class Diagram	1 day	11/8/17 8:00 AM	11/8/17 10:00 AM
	Sequence Diagram	12 days?	11/9/17 8:00 AM	11/13/17 5:00 PM
	Meeting	1 day?	11/14/17 2:00 PM	11/14/17 4:00 PM
	Chapter 4: Implementation	20 days?	11/15/17 8:00 AM	11/21/17 5:00 PM
	Introduction	1 day	11/15/17 8:00 AM	11/15/17 10:00 AM
	Language Selection	1 day	11/16/17 8:00 AM	11/16/17 10:00 AM
	Interfaces	12 days?	11/17/17 8:00 AM	11/21/17 5:00 PM
	Review	14 days	11/22/17 8:00 AM	11/27/17 1:00 PM
	Review 1	6 days	11/22/17 8:00 AM	11/23/17 1:00 PM
	Review 2	6 days	11/24/17 8:00 AM	11/27/17 1:00 PM

Figure 1. 1 Tabular view of the project management plan

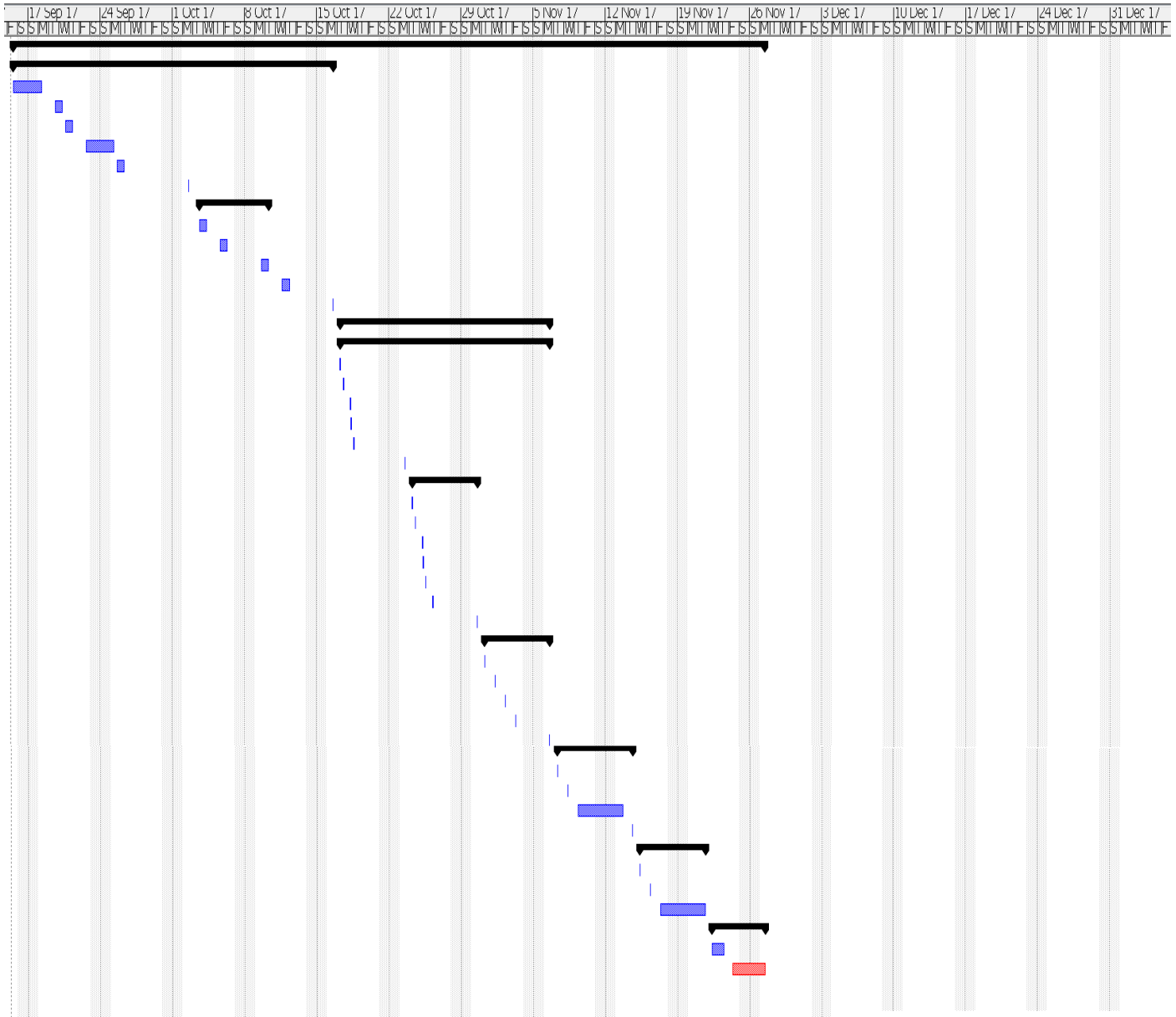


Figure 1. 2 Timeline view of the project management plan

1.7 Report Structure

This chapter has briefly introduced the system. It has also elucidated problem description, proposed solution, scope, objective, and described the organization of the project and project management plan. Chapter 2 describes the functional and non-functional requirement specifications of the system. Chapter 3 provides an overview of the software design. Chapter 4

describes the details of implementation phase, chapter 5 describe the details of testing phase and chapter 6 concluded the whole project.

*“You educate a man; you educate a man. You educate a woman; you educate a generation.”
Brigham Young*

Chapter 2

Requirements Gathering and Analysis

2.1 Product Overview

The developed distributed application is implemented in Python version 3.6.4 (frame work tkinter). Admin [Lab administrator] will manage the system. This product detects data leaks and perform actions against intruder(s) when illegitimate activities are performed within the organizational network and sends alert messages to the admin.

2.2 Major Functions

The major functions of DLP is the classification of organizational data in different categories such as secret, confidential, and non-confidential. Using Naive Bayes multinomial procedure, data monitoring and leak detection of in-use document is detected. After that following possible remedial actions can be taken against in-use data leakage; cancel, pause, and resume the printing operation and sends alert messages to the administrator.

2.3 Major Inputs and outputs

2.3.1 Major Inputs

Major inputs are given below:

- .txt extension files of text.

2.3.2 Major Outputs

The system generates alert messages and perform actions (cancel, pause, resume) against illegitimate activities when:

- Printing confidential documents.
- Printing secret documents.

2.4 Definitions, Acronyms and Abbreviations

Table 1. 1 Abbreviations

Terms	Description
DLP	Data Leakage Prevention
Admin	Lab Administrator
UML	Unified Modeling Language
User	Lab Administrator
System	Desktop Application
End User	Lab Administrator

2.5 Overview

The rest of the chapter focuses on functional, non-functional and performance requirements. The overall functionality of the system, use cases and their description, and domain model have also been elucidated.

2.6 User Characteristics

This is assumed that the administrator have basic knowledge of computer or laptop and knowledge of the desktop application.

2.7 Constraints

Admin must have a computer system configured with a local network. Admin should have login details to logged in to the computer system. The computer system should have minimum of 4 GB RAM and 100 GB hard drive.

2.8 Assumptions and Dependencies

The application is dependent on the availability of a computer system in a network. It is assumed that the admin has a computer with access to the system.

2.9 Specific Requirements

2.9.1 Functional Requirements

The basic functional requirements are given below.

- Classification of data
- Detection of leaked events
- Capture Printing Events of Windows Operating System (Windows X)
- Cancel print task
- Pause print task
- Resume print task
- Searching leaked records based on threat category
- Export summary of leaked records to external .txt extension file

2.9.2 External Interface Requirement

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

2.9.3 User Interfaces

User is able to interact with the intended application using window operating system.

- Administrator Login:
Input: User name and password
Output: Administrator login successfully.
Admin login interface is shown in design chapter.
- Configure Parameters:
Input: Port No
Output: pass its value to the server socket.
Configure parameter interface is shown in the design chapter
- Run agent
Input: Click on run button.
Output: DLP agent starts to detect leaks.
Run agent interface is shown in design chapter.

- Account Setting
 - Input: New password.
 - Output: password change successfully.
 - Account setting interface is shown in design chapter.
- Search Leak Records
 - Input: Threat category, click on search button.
 - Output: All leak records is display on screen.
 - Audit Leak interface is shown in design chapter.
- Export Leak Records:
 - Input: Click on Save as button.
 - Output: All leak records are saved to a file.
 - Export Leak records interface is shown in design chapter.

2.10 Software System Attributes

2.10.1 Reliability

System should be reliable. There should be no occurrence of the failure. The system should be able to work properly all-time, i.e., to the extent to which it works as and when needed. The system should give proper response against every leak performed by insider user.

2.10.2 Availability

System should be available to user at any time. System should monitor data continuously and send notifications to administrator if any illegitimate activity is performed by insider. System should be available within the premises of the organization.

2.10.3 Security

Since this system will be hosted within the organizational network, user should only be able to access the system through his personal computer where the developed application is installed. No other members in the network can access the personal account of the administrator. The system has its own login credentials to use it. Administrator must enter his login credentials before using the system. The system maintains the login sessions of the administrator.

2.10.4 Maintainability

In some cases, maintainability involves continuous improvement in the system, learning from the past in order to improve the ability to maintain systems, or improve the reliability of systems on the basis of maintenance experience. The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.

2.10.5 Portability

This is a network-based desktop application. It runs on Microsoft Windows operating system and gradually its next version will be run on the Linux operating system.

2.10.6 Performance

The system must have strong computing capabilities to handle multiple data classes and respond against illegitimate activity within short period of time. Another performance requirement is the correct identification of the leakage data with the accuracy of over 90%. System should be able to deal with numerous computers at a time.

2.11 Product Functions

2.11.1 Analyze System

Administrator has rights to analyze the whole functionality of the system. Administrator should be able to perform actions such as cancel, resume and pause printing request.

2.12 Use Case Diagram

Administrator can sign up, login, configure parameters, run agent, account setting, search leak records, export leak records to file. The use case diagram is shown in Figure 2.1.

The system use cases include text classification, leak detection, capture print event, cancel print event, pause print event and resume print event. All the system use cases are working when administrator initiate the run agent use case.

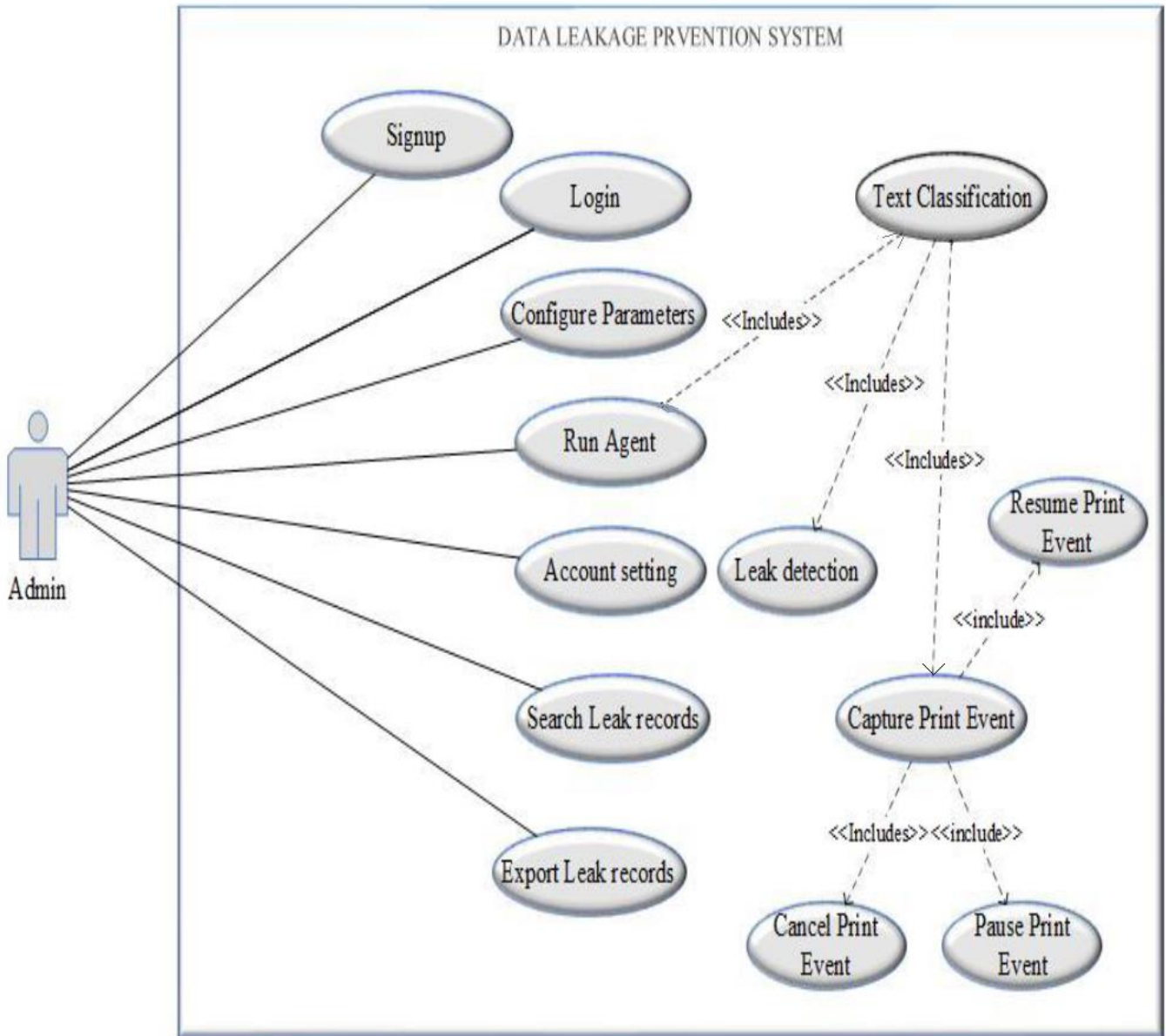


Figure 2. 1 Use Case Diagram

2.13 Use Case Description

2.13.1 Use case 1: Signup

Table 2. 1 Signup Use case

ID	UC1
Name	Signup
Primary Actor	Administrator
Pre-Conditions	Administrator executed the application.
Post-Conditions	Administrator signup successfully.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin opens the DLP's desktop application. 2. Admin sees the desktop application window. 3. Admin click on the signup button. 4. Admin fill up signup details. 5. Admin click on signup button.
Alternative flows or Extensions	<ol style="list-style-type: none"> 1a. if user enters the invalid user name <ol style="list-style-type: none"> 1. System asks user to enter the user name again. 2. if user enters the invalid password. <ol style="list-style-type: none"> 2a. System signals invalid password and ask to re-enter password.
Frequency	Could be nearly continuous

2.13.2 Use case 2: Login

Table 2. 2 Login Use case

ID	UC2
Name	Login
Primary Actor	Administrator
Pre-Conditions	Administrator have valid username and password.
Post-Conditions	Administrator logins successfully.
Main Scenario	<p>Success</p> <ol style="list-style-type: none"> 1. Admin opens the DLP's desktop application. 2. Admin sees the desktop application window. 3. Admin enters the username and password. 4. Admin clicks on the login button.
Frequency	Could be nearly continuous

2.13.3 Use case 3: Configure Parameters

Table 2. 3 Configure Parameters Use case

ID	UC3
Name	Configure Parameters
Primary Actor	Administrator
Pre-Conditions	Administrator successfully login to the system.
Post-Conditions	Administrator configures parameters.
Main Scenario	<p>Success</p> <ol style="list-style-type: none"> 1. Admin open the DLP desktop application. 2. Admin sees desktop application window. 3. Admin selects the configure parameters tab. 4. Admin enter only Port No. 5. Admin press start button to start the server.
Frequency	Could be nearly once.

2.13.4 Use case 4: Run Agent

Table 2. 4 Run Agent Use case

ID	UC4
Name	Run Agent
Primary Actor	Administrator
Pre-Conditions	Administrator logged in successfully.
Post-Conditions	1. Administrator successfully runs DLP agent.
Main Success Scenario	<ol style="list-style-type: none"> 1. Administrator opens DLP system. 2. Administrator enters login credentials and click on login button. 3. Administrator clicks on the run agent button to run the DLP agent.
Frequency	Could be nearly once.

2.13.5 Use case 5: Account Setting

Table 2. 5 Account Setting Use case

ID	UC5
Name	Account setting
Primary Actor	Administrator
Pre-Conditions	Administrator successfully login to the system.
Post-Conditions	Administrator changed login details.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin selects the setting account tab. 2. Admin enter new password. 3. Admin presses save changes button.
Frequency	Could be nearly once.

2.13.6 Use case 6: Search Leak Records

Table 2. 6 Search Leak Records Use case

ID	UC6
Name	Search Leak Records
Primary Actor	Administrator
Pre-Conditions	Administrator logged in successfully.
Post-Conditions	1. Administrator run DLP agent successfully.
Main Success Scenario	<ol style="list-style-type: none"> 1. Administrator click on the Leak Records tab. 2. Administrator enter threat category and click on search button. 3. Administrator can then view the records.
Frequency	Could be nearly continuous.

2.13.7 Use case 7: Export leak records

Table 2. 7 Export leak records Use case

ID	UC7
Name	Export leak records
Primary Actor	Administrator
Pre-Conditions	Administrator have access to use the desktop application
Post-Conditions	1. Administrator generates the summary report.
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin starts the desktop application. 2. Admin clicks on the reports tab. 3. Admin click on save as button.
Frequency	Could be nearly continuous

2.14 Domain Model

This model represents different entities such as raw data, data classification, data analyzer, reports, policies and administrator. Raw data entity contains random data in the form of information. Data classification entity contains non-Confidential, confidential and secret data after applying the classification techniques such as Naive Bayes Multinomial procedure. Data analyzer entity is main entity which is looking for the data leaks, capture printing events and applying remedial actions such as cancel, pause, resume printing events. Reports entity contains information about the different leak records. Leak records entity contains specific information about printed request document. Domain model also shows relationship among entities. “Classify By” relation is between the data classification entity and the raw data entity. “Analyze By” relation is between the data classification entity and the raw data analyzer entity. “Store by” relation is between the data analyzer entity and the leaked records entity. “View by” relation is between the leaked records entity and the administrator entity. “Created by” relation is between reports entity and administrator entity. Administrator entity contains the username and password of the administrator for the login purpose.

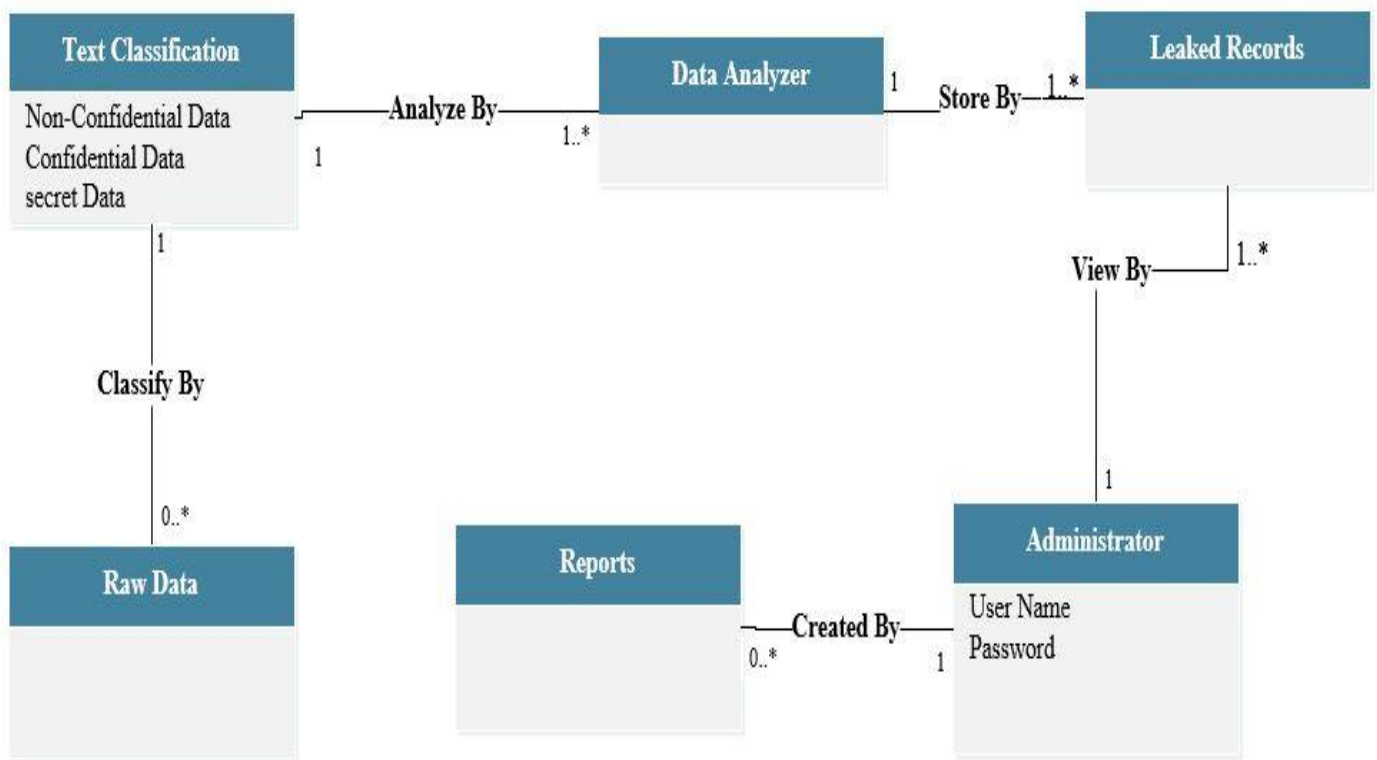


Figure 2. 2 Domain Model

2.15 Database Requirements

There are multiple tables namely administrator, leak records, and reports in the database. Administrator table contains the retirement date, username, password, id, and employment date of the admin. Leak records include all information of illegitimate activities captured and reports table have different information for making summary of leak events. Administrator view the leak records, and make summary reports. Each table contains different attributes, shown in oval shape notations such as in table administrator there are retirement date, username, password, id, and employment date attributes. The Entity Relationship Diagram (ERD) of the system is shown in Figure 2.3.

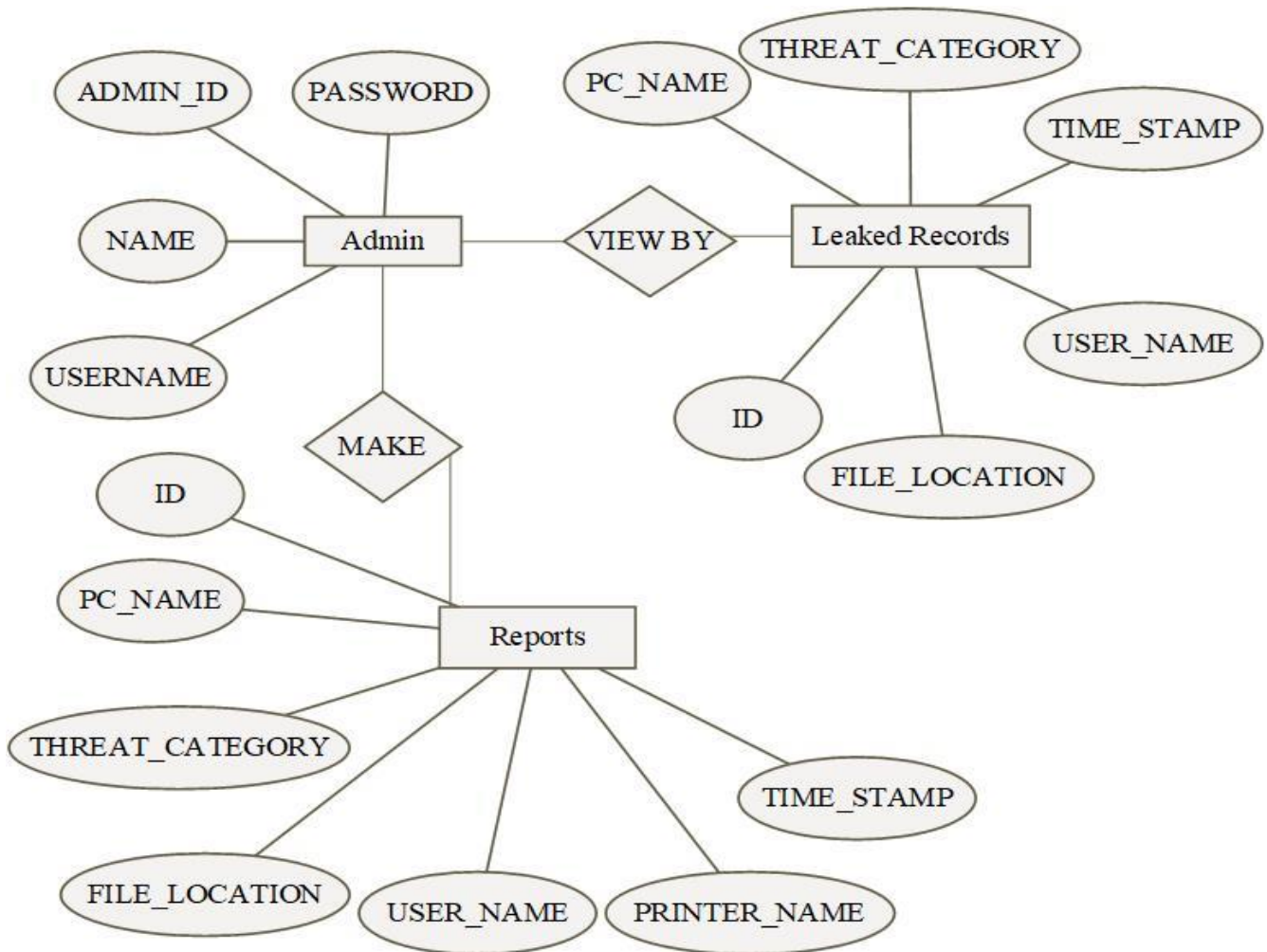


Figure 2. 3 Entity Relationship Diagram

This chapter has provided an overview of the requirement gathering and analysis. The description is based on the functional and non-functional requirements. Next chapter elucidates software design aspects.

“Education is most power weapon which you can use to change the world.” Nelson Mandela

Chapter 3

Software Design Description

This chapter first gives the complete description of the software design. It then elaborates the architectural design and detailed description of the components of the system. Finally, elucidates the user interface design and interaction diagrams such as system sequence and class diagrams.

3.1 Introduction

Software Design Description (SDD) is the representation of a software design which is used for communicating design information of a system to its stakeholders [3]. It shows how the software system will be structured to satisfy the requirements. The SDD is specified into two stages. The first is a preliminary design in which the overall system architecture and data architecture are designed and defined, respectively. In the second stage, the more detailed data structures are defined and algorithms and codes are developed on the basis of the defined architecture.

3.1.1 Design Overview

Design begins with requirement model and at each stage. The software design work product is reviewed for clarity, correctness, and completeness [3]. Software design sits at the technical kernel of software engineering and is applied regardless of the software process model that is used. The requirements translated clearly through designing class diagram, sequence diagram and user interface interactions.

3.1.2 Requirement Traceability Matrix

Requirements traceability matrix is a matrix in which we describe that which requirement is mapping with which sequence diagram, test case, and method of class diagram. The purpose of the traceability matrix is that when requirements have to be updated then one can update that requirement using traceability matrix instead of going through the whole document. It is often used with high-level requirements and detailed requirements of the product to the matching parts

of high-level design, detailed design, test plan, and test cases. Traceability matrix of this system is given in Table 3.1.

Table 3. 1 Requirement Traceability Matrix

Requirement Id	Requirement Name	Sequence Diagram	Test Case	Class Diagram	Interface
UC:1	Signup	Figure 5.1	Table 5. 1	Figure 3. 1	Figure 4. 5
UC:2	Login	Figure 3. 8	Table 5. 1	Figure 3. 2	Figure 4.2
UC:3	Configure Parameters	Figure 3. 9	No	Figure 3. 3	Figure 4. 7
UC:4	Run agent	Figure 3. 4	No	Figure 3. 5	Figure 4.14
UC:5	Account Setting	Figure 3. 6	Table 5. 2	Figure 3. 7	Figure 4.10
UC:6	Search Leak Records	Figure 3. 8	No	Figure 3. 9	Figure 4.8
UC:7	Export Leak Records	Figure 3. 10	Table 5. 3	Figure 3. 11	Figure 4.9

3.2 System Architecture Design

Architectural design defines the relationship between major structural elements of the software. It defines the design patterns that can be used to satisfy the requirements that have been defined for the system. Architecture design entails the manner in which these components interact and the structure of data that are used by the components. Components or modules are generalized to represent major system elements and their interactions.

3.2.1 Chosen System Architecture

The chosen architecture for this system is three tier and client-server architectural pattern shown in figure 3.1. The three-tier architecture is a software architecture pattern in which the user interface (presentation), functional process logic (Application logic), and computer data storage are developed and maintained as independent modules. The presentation (interface) layer focuses on user interfaces, where admin performs different queries. Using the presentation layer, admin can put data to database, configure parameters, add or remove policies, audit leak records,

and manage account setting. The application (business) logic is hidden from the admin, which includes logical flow of the system. Data entered by the admin through the presentation layer can be received by the application logic layer and it manipulates accordingly. The final layer is the database layer, which is connected with the controller class in the application layer. Database can store all necessary information used for generating reports, login credentials, and different policies.

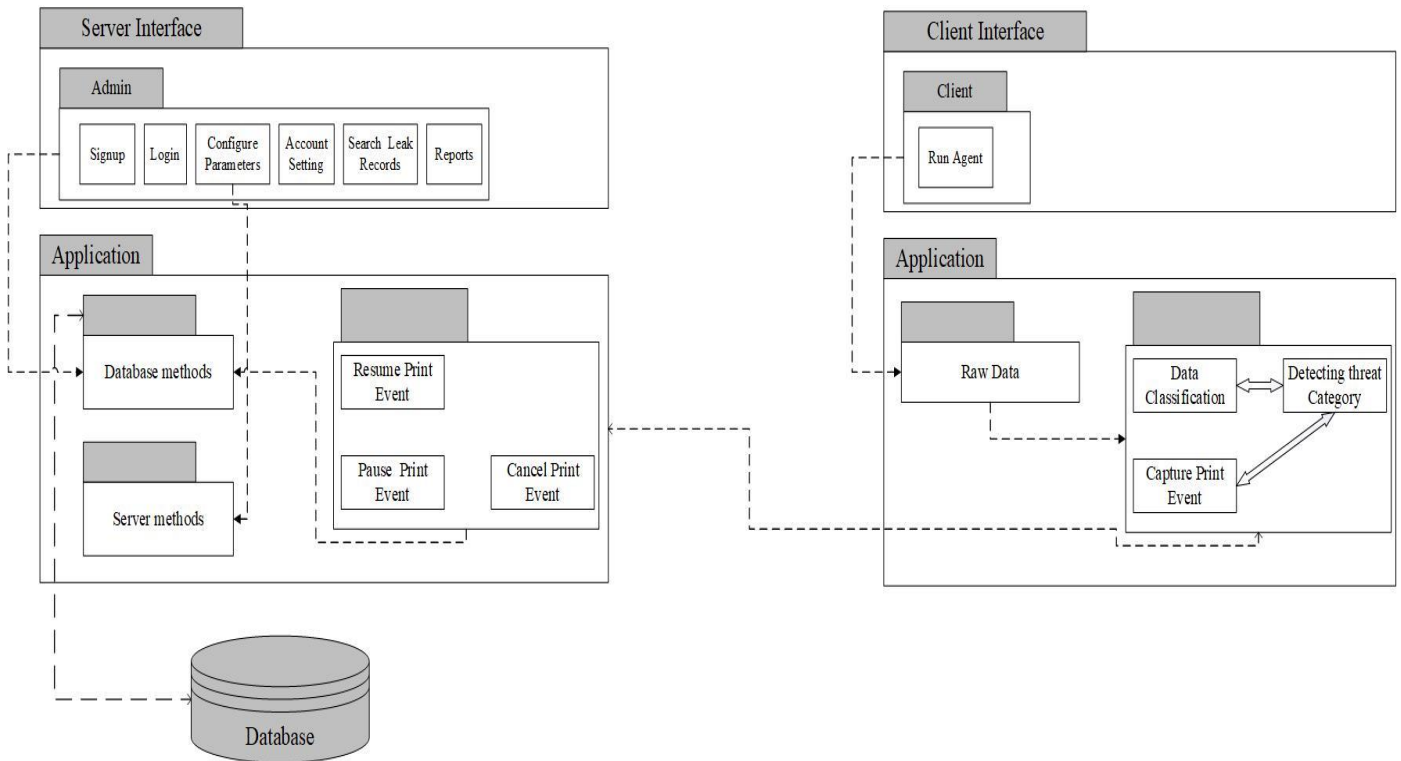


Figure 3.1 Architecture Diagram

3.2.2 System Interface Description

3.3 User Interface Design

User interface design establishes effective communication between a user and a computer. The user interface design begins with the identification of users, tasks, and environmental requirements.

3.3.1 Description of User Interface Design

User interface is the part of the software and is designed in such a way that it is expected to provide the user insight of the system. The following interfaces are given below, login interface, configure parameters interface, run agent interface, add policies interface, remove policies interface, account setting interface, audit leak records interface and generate reports interface. Login interface is used by administrator to login into the system. Administrator provide user name and password to login to the system. Configure parameters interface is used to populate specific type of data that is used in the system. Run agent interface is used to execute the system initially. After that, it continuously run in the background. Add policies interface is used to insert new policies to the database. Remove policies interface is used to delete unnecessary policies from database. Account setting interface is used to update the login credentials. Audit leak records interface is used to search the different leaks records for audit purpose. Generate reports interface is used to generate summary of different data.

Interface 1: Login

The figure 3.2 is login interface created while making designing of the proposed system.

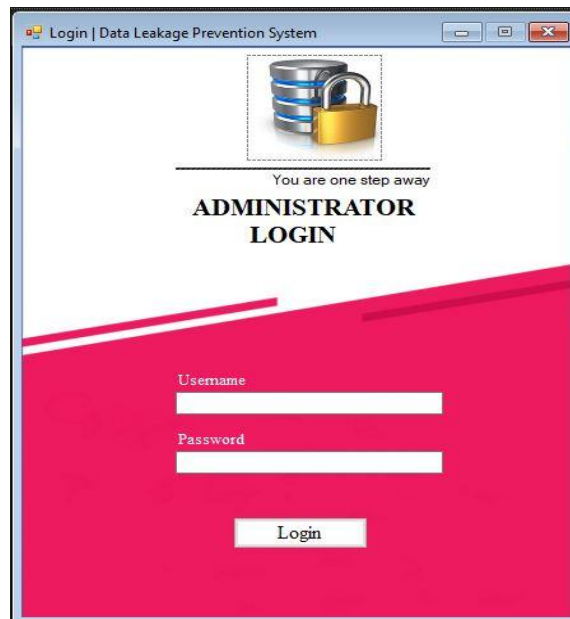


Figure 3.2 Login Interface

Interface 2: Configure Parameters

Figure 3.3, is configure parameters used by administrator while starting the server.

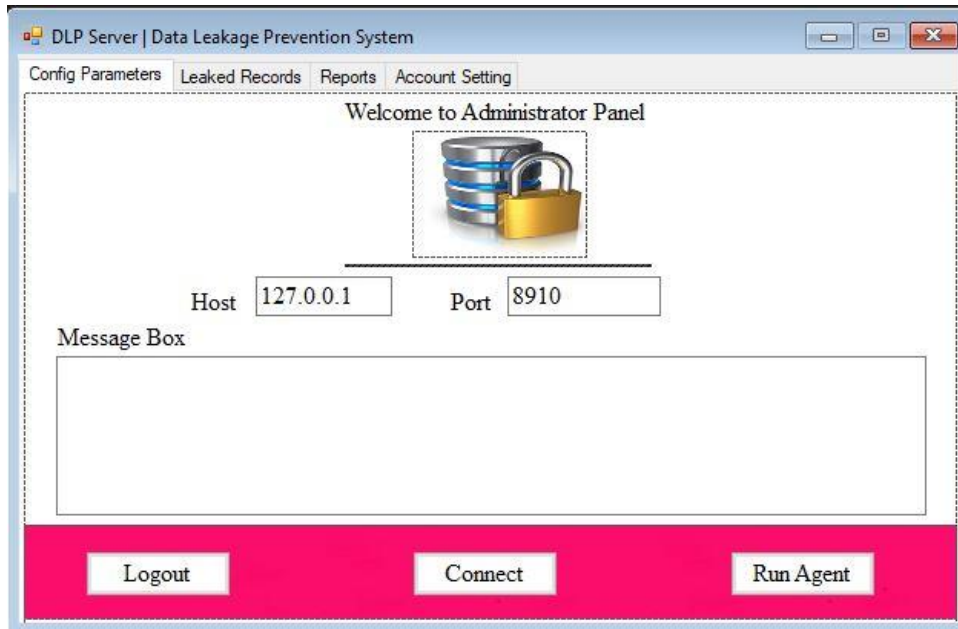


Figure 3.3 Configure Parameters Interface

Interface 3: Run Agent

Figure 3.4, shows run agent interface. Which initiate the system functionality.

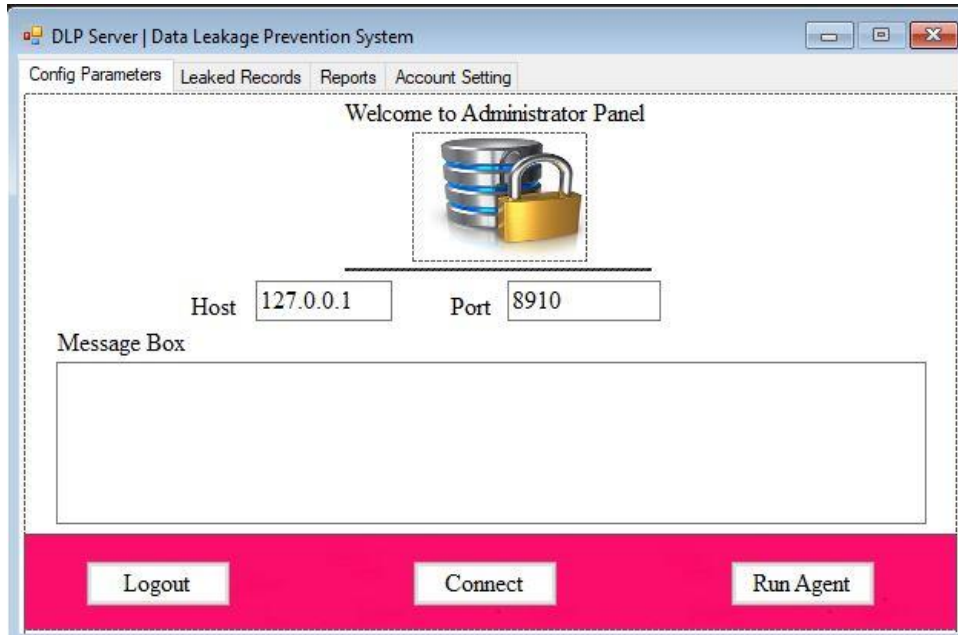


Figure 3.4 Run Agent Interface

Interface 4: Account setting

Figure 3.5, is account setting interface. Admin can change its login credentials.

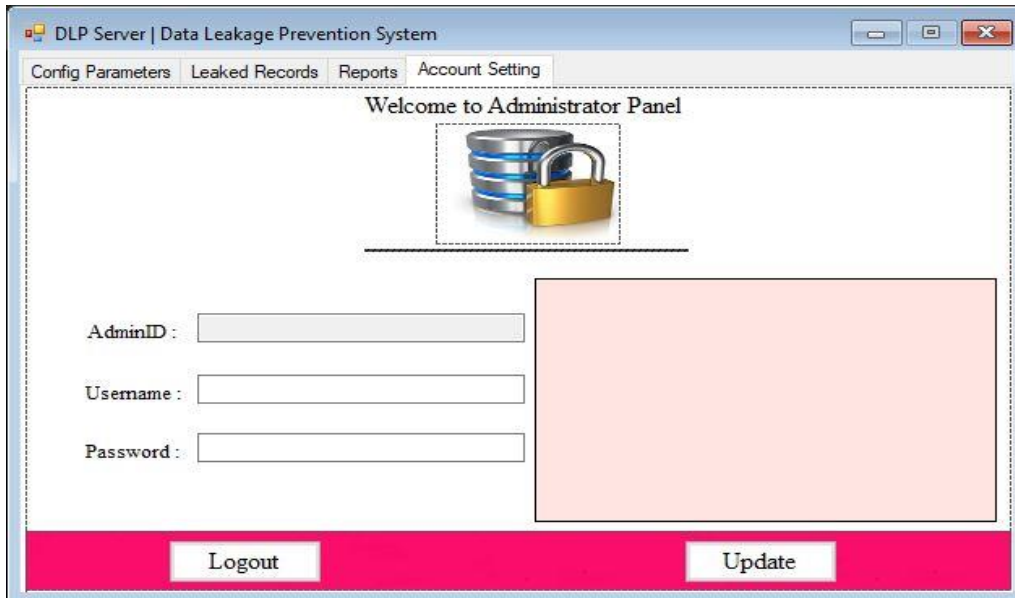


Figure 3.5 Account Setting Interface

Interface 5: Generate Reports

Figure 3.6, is generate reports interface. Admin can search leak records.

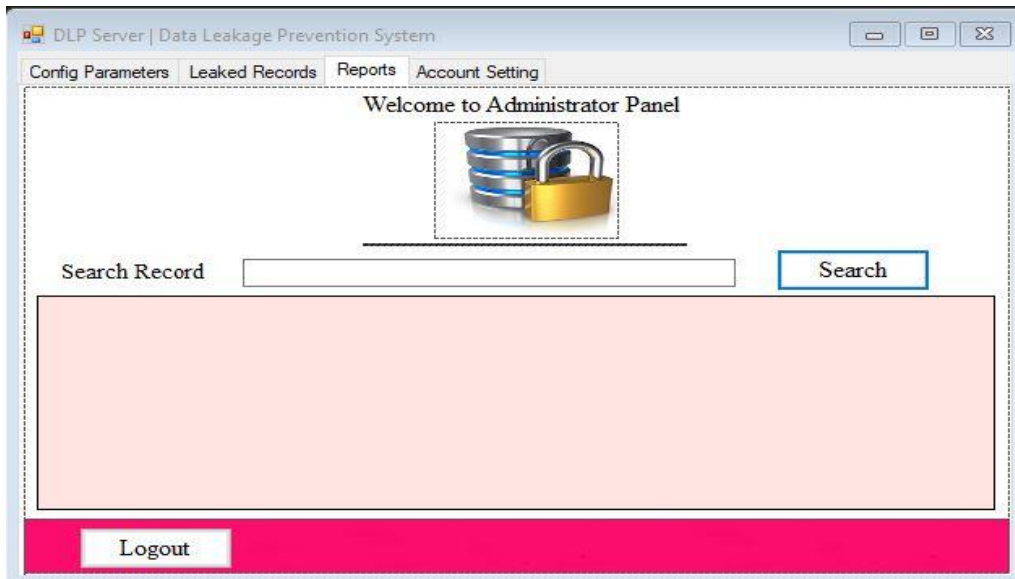


Figure 3.6 Generate Reports Interface

3.4 Sequence Diagram

Sequence diagrams are used to present the interaction between actors and objects in a system and interaction between the objects themselves. A sequence diagram shows the interactions that take place during a particular use case or use case scenario.

3.4.1 Signup

Signup sequence diagram shows that the sequence of interaction takes place when the admin wants to gain access to the system. For this purpose, admin make an account to gain access to the system. Admin fill signup form with following details, name, username, enrolment date and password to the system. The sequence diagram of the admin login processing is giving in Figure 3.7.

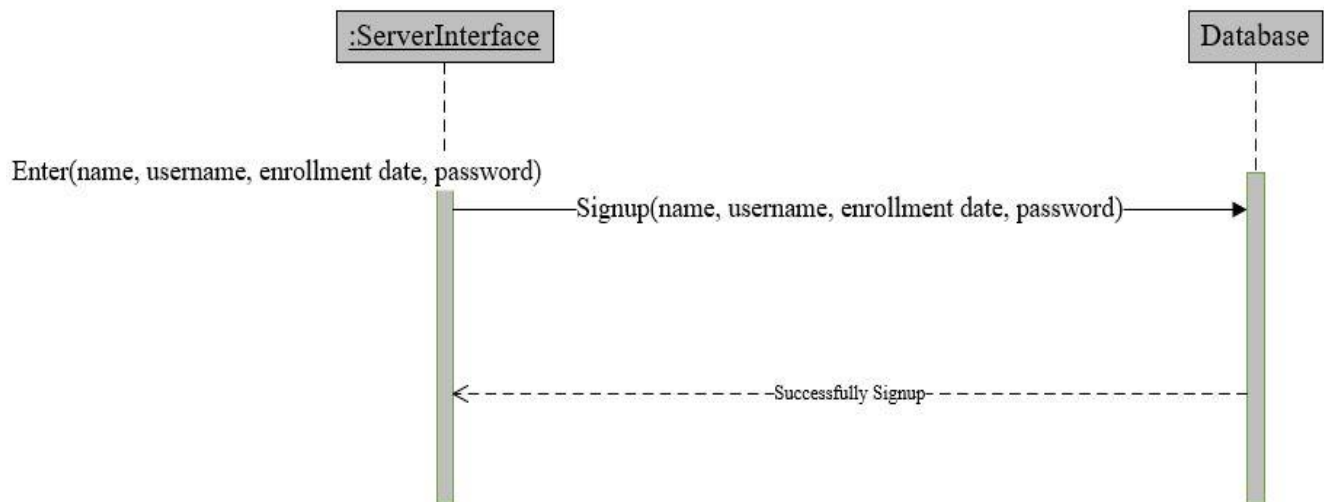


Figure 3.7 Signup Sequence Diagram

3.4.2 Login

Login sequence diagram shows that the sequence of interaction takes place when the admin wants to login to the system. User enters username and password to the system, then system validates username and password. If admin is authorized, then admin will successfully login to the system, otherwise an error message will be displayed to the user. The sequence diagram of the admin login processing is giving in Figure 3.8.

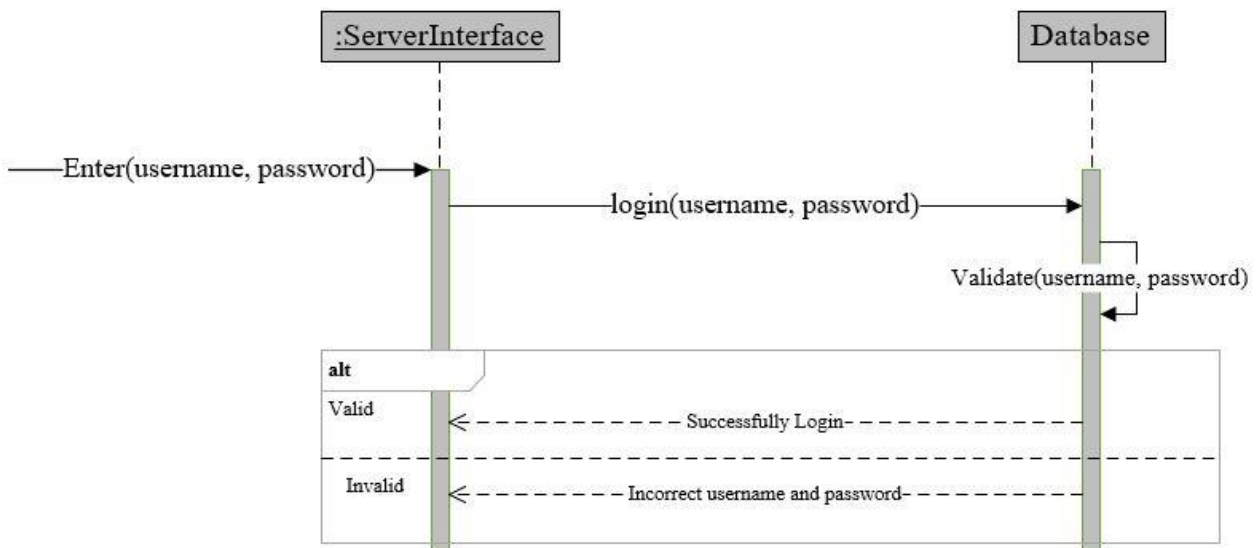


Figure 3.8 Login Sequence Diagram

3.4.2 Configure Parameters

Configure parameters sequence diagram, shown in Figure 3.9, shows the interaction between self-class object. The serverInterface object receives the parameters entered by administrator and it will preserve parameters for future usage in the system.

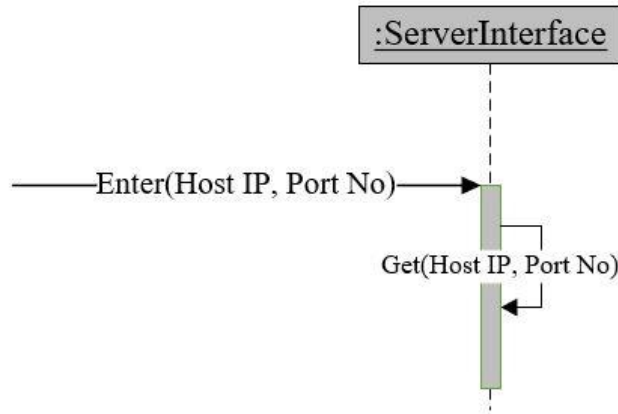


Figure 3.9 Configure Parameters Sequence Diagram

3.4.3 Run Agent

Run agent sequence diagram, shown in Figure 3.10, shows the sequence of methods that how the model method is invoked.

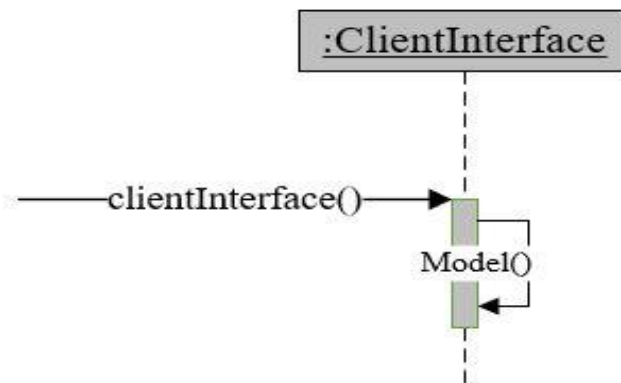


Figure 3.10 Run Agent Sequence Diagram

3.4.4 Account Setting

Account setting sequence diagram, show in Figure 3.11, represents the sequence of methods to change the administrator account details.

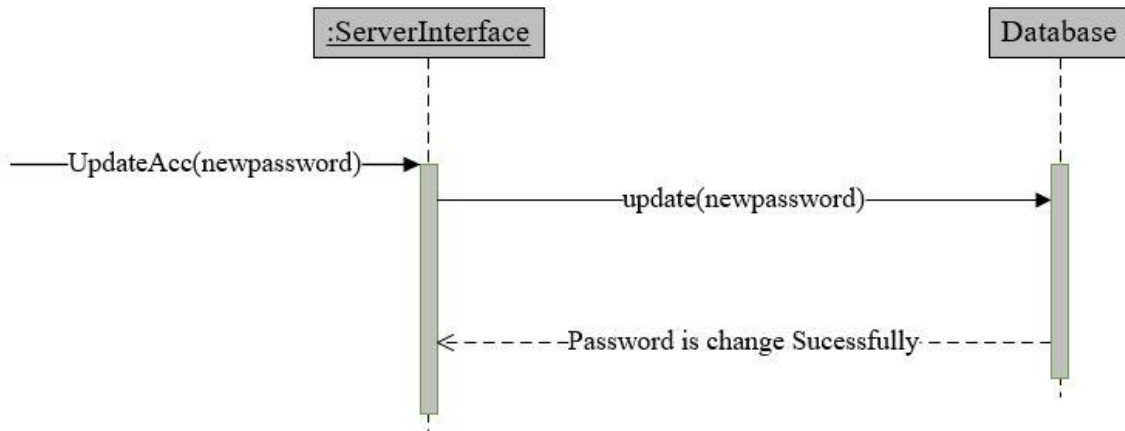


Figure 3.11 Account Setting Sequence Diagram

3.4.5 Search Leak Records

Search leak records sequence diagram, shown in Figure 3.12, shows interaction between ServeInterface class object and database class. The ServerInterface object receives the threat category and it fetches all the leak records from database.

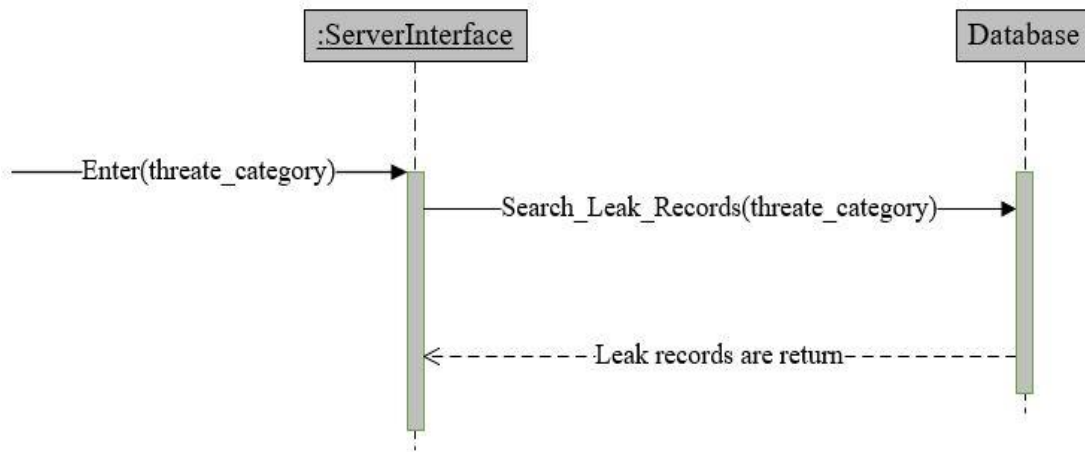


Figure 3.12 Search Reports Sequence Diagram

3.4.5 Export Leak Records

Search leak records sequence diagram, shown in Figure 3.13, shows interaction between ServerInterface class object and database class. The ServerInterface object receives the threat category and it fetches all the leak records from database.

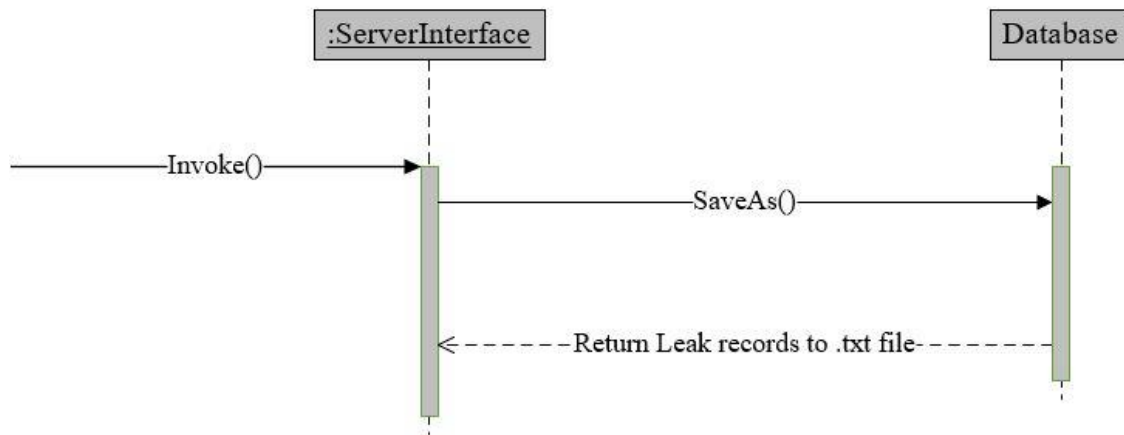


Figure 3.13 Export Leak Records Sequence Diagram

3.5 Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system, but also for constructing executable code of the software application [4]. It is also known as a structural diagram.

Class diagram represents classes which can interact with each other to perform the functionality of the system. LoginInterface class have association relation with ServerInterface class. ClientInterface class have dependency relation with ServerInterface class.

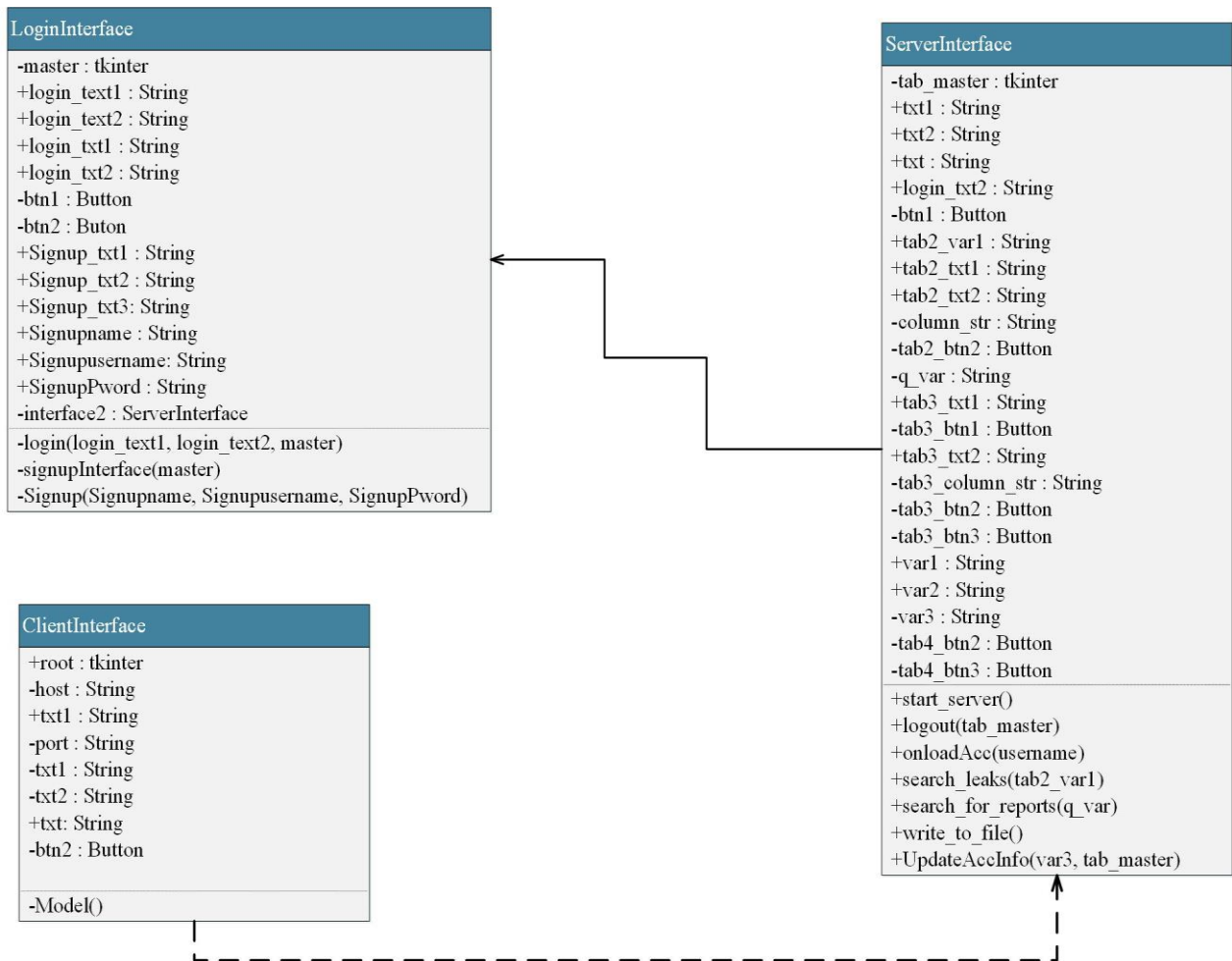


Figure 3.14 Class Diagram

This chapter has provided a comprehensive description of the software design. The detailed descriptions regarding architecture design, components of the system, and user interfaces are provided. Finally, interactions among objects are shown by the interaction diagram and relationship between the instances is shown by the class diagram. Implementation techniques discussed in the next chapter.

*“You educate a man; you educate a man. You educate a woman; you educate a generation.”
Brigham Young*

Chapter 4

Software Implementation Document

4.1 Introduction

In this chapter, the framework and language selection for the project has been discussed. It also includes screenshots of implemented system.

4.2 Framework Selection

The framework used for this system is developed in Tkinter python 3.6.4. It supports for making graphical user interfaces, provide modules; related to windows operating system, and give support for using machine learning algorithms. This development tool is preferred because it is a powerful in terms of training machines, easy to use, have built-in modules for complex structures.

4.3 Language Selection

- **Python**

Because it is easy to use, its syntax is elegant, and provide with large standard libraries. Achieving core functionality of the system includes machine learning, capturing printing events, and perform remedial actions are being possible because it can also provide Microsoft windows operating system kernel level settings.

4.4 Operating System

This application will run only on Microsoft Windows Operating Systems.

4.5 Algorithms and techniques

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms. Therefore, we are using scikit learn API's for this system. The purpose of making this system is to provide internal security of an organizational content using these algorithms.

4.5.1 CountVectorizer

It converts collection of text documents to a matrix of token counts. This implementation produces a sparse representation of the counts. [5]

4.5.2 Term Frequency Inverse Document Frequency

The goal of such an algorithm is to rank documents according to their relevance to a query. TF measures how relevant is a word for a specified document. IDF measures how relevant is word according the full corpus of documents. [6]

4.5.3 Multinomial Naïve Bayes

It estimates the conditional probability of a particular word given a class as the relative frequency of term in documents belonging to class. The variation takes into account the number of occurrences of term in training documents from class including multiple occurrences. [7]

4.5.4 Win32print Module

A module encapsulating the Windows printing API. This API includes different methods for handling printer related operations. [8]

4.6 Data Classification

- Supervised Classification

Classification is the task of choosing the correct class or category for a given input. In basic classification tasks, each input is considered in isolation from all other inputs, and the set of classes or categories are defined in advance.

- Categories for Classification

- The data is classified on the basis of three classes or threat categories as secret, confidential, and non-confidential.
- Data is termed as *secret* when its illegitimate leak would cause serious damage to an organization. This type of data requires special protection.
- *Confidential* data is based on the content that is considered as sensitive by organization that also requires protection.
- *Non-confidential* data is type of data which is public and have no sensitive information. Therefore, such data do not require any special protection.

- Data Sets Details

Table 4. 1 Data Set Details

S.No	Threat Categories	Content Type	Total Files	Training Files	Testing Files
1.	Secret	Business	510	408	102
2.	Confidential	Politics	417	335	82
3.	Non-Confidential	Technology	401	321	80

4.7 System Flow

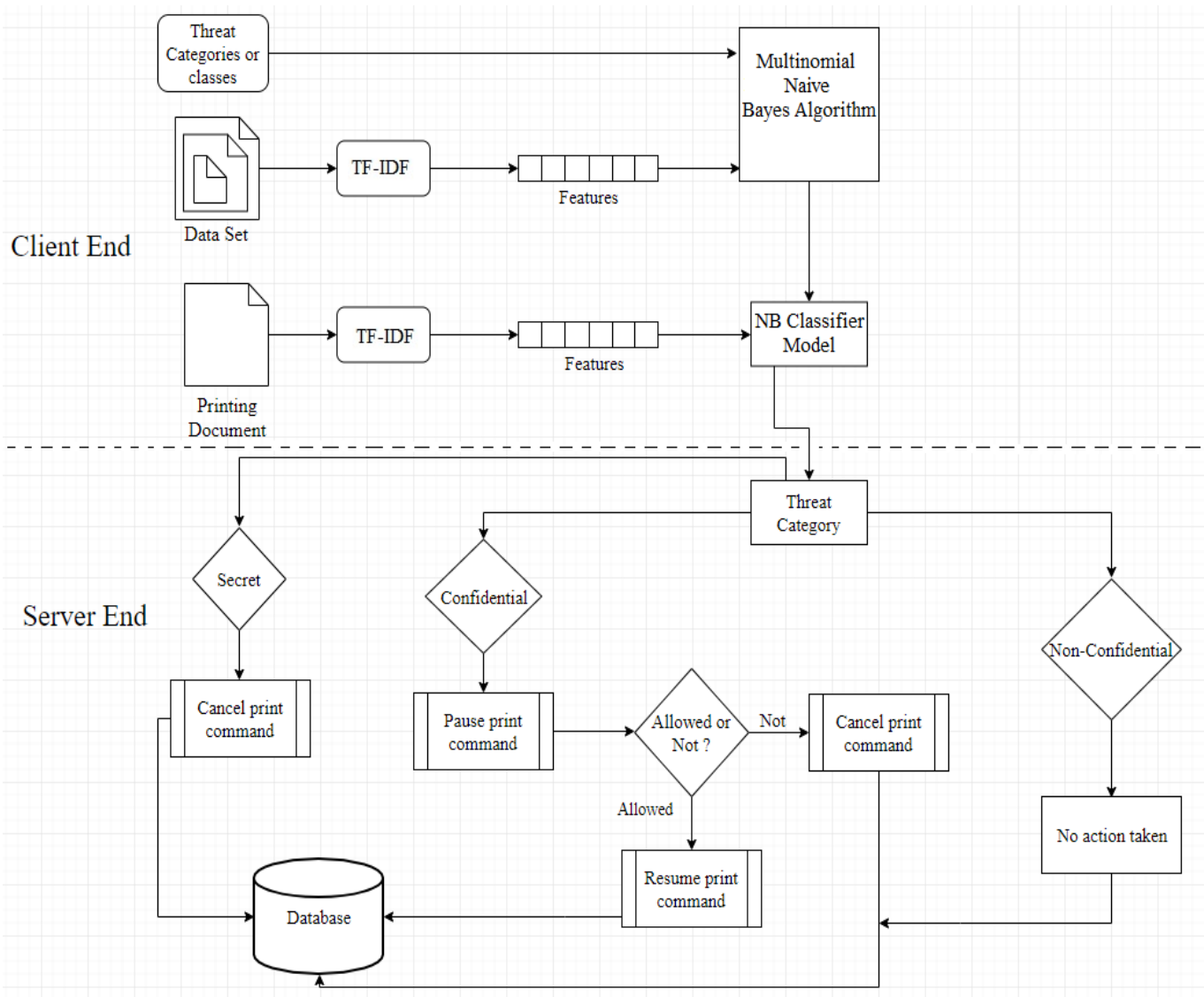


Figure 4. 1 System Flow Diagram

4.8 Desktop Application Screenshots

The Application Screen Shots provide an idea of the system; Initially the application starts with a login screen, after login authentication; all other screens are accessible except signup.

4.8.1 Login Interface

Figure 4.2, is login interface for administrator. Administrator enters login credentials to access its functionality.

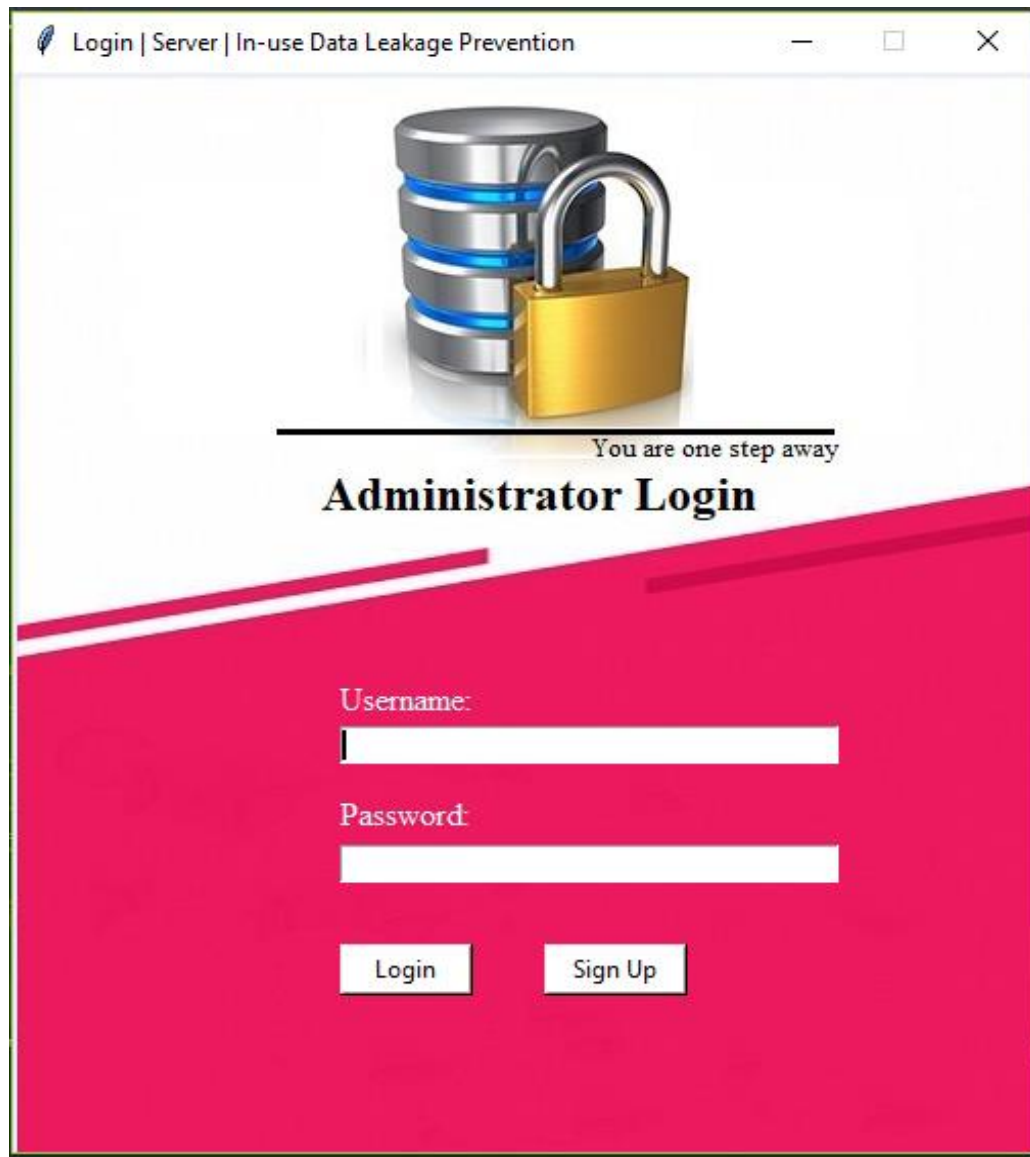


Figure 4. 2 Login Interface

4.8.2 Login Interface with Empty Entry Boxes

Figure 4.3, is login interface. Administrator click on login button without filling login credentials.



Figure 4. 3 Login Interface with empty boxes

4.8.3 Login Interface with wrong input

Figure 4.4, is login interface. Administrator enters wrong login credentials. System prevent to access its functionality. Show prompt message for wrong login credentials.

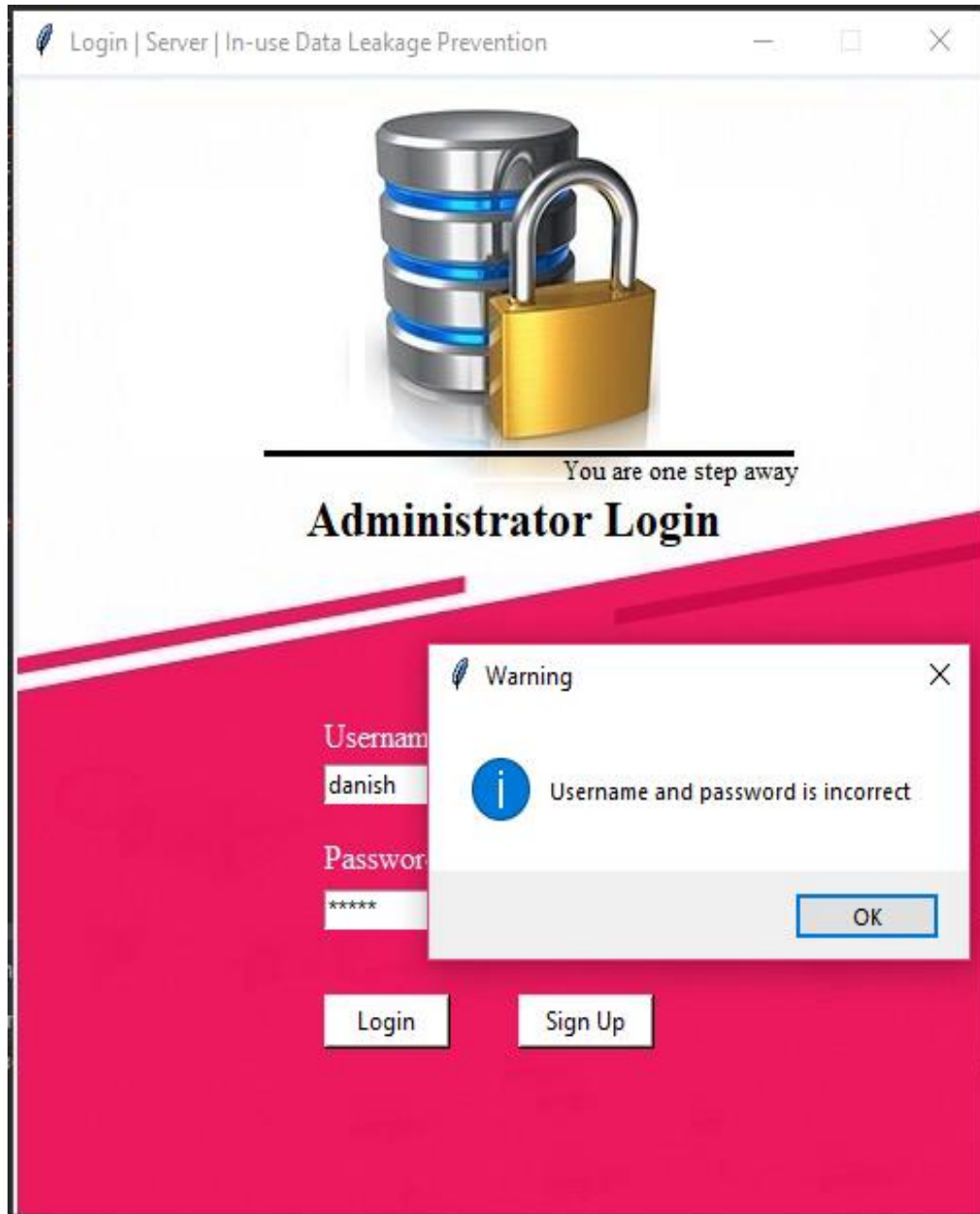
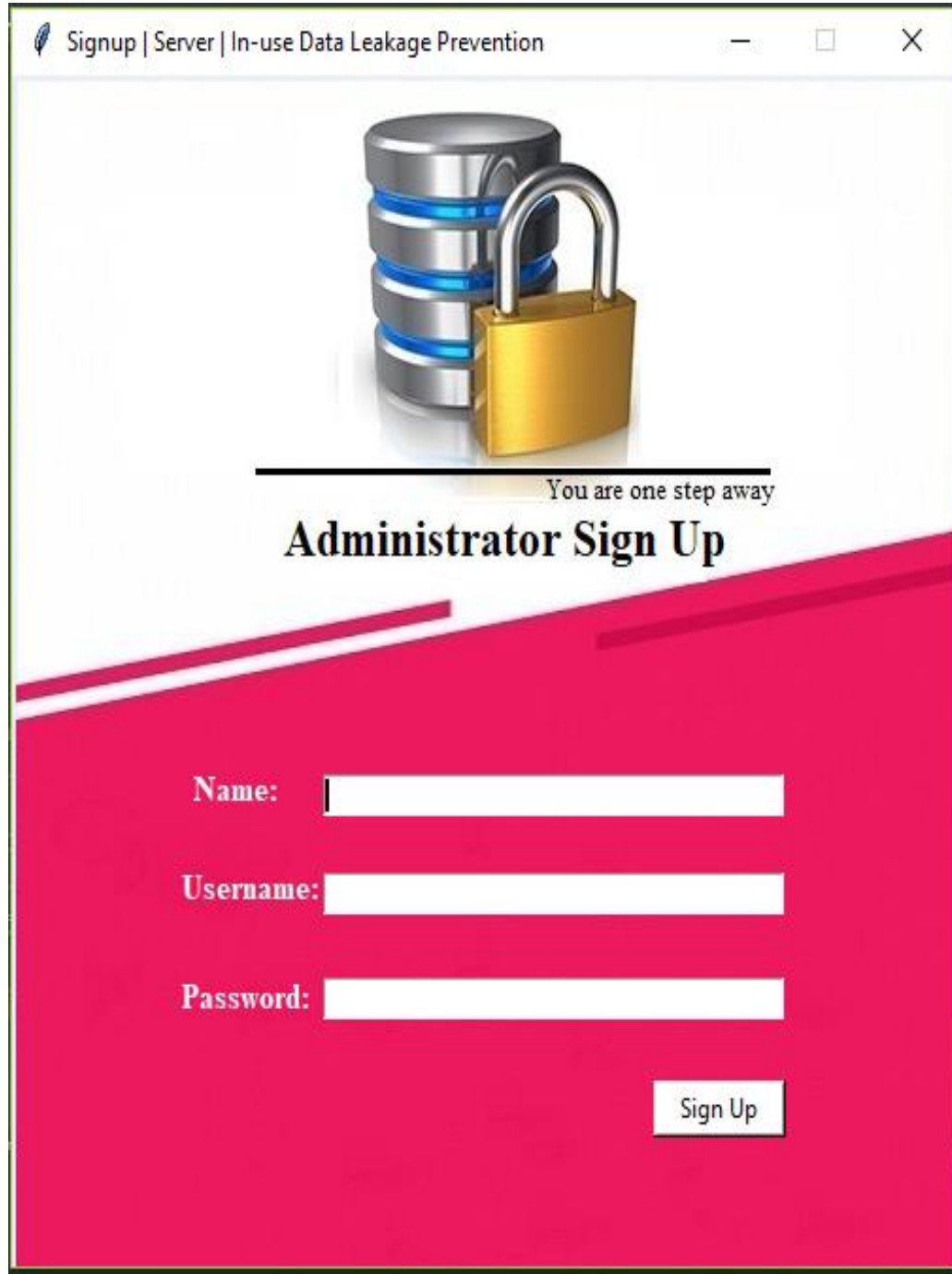


Figure 4. 4 Login Interface with wrong input

4.8.4 Signup Interface

Figure 4.5, is signup interface. Administrator enters his personal information to create login account.



Signup | Server | In-use Data Leakage Prevention

You are one step away

Administrator Sign Up

Name:

Username:

Password:

Sign Up

Figure 4. 5 Signup Interface

4.8.5 Signup Interface Successfully

Figure 4.6, is signup interface. Administrator enters his personal information to create login account.



Figure 4. 6 Signup with correct inputs

4.8.6 Configure Parameter Interface

Figure 4.7, is configure parameter interface. Administrator enters only server port no on which it listens.

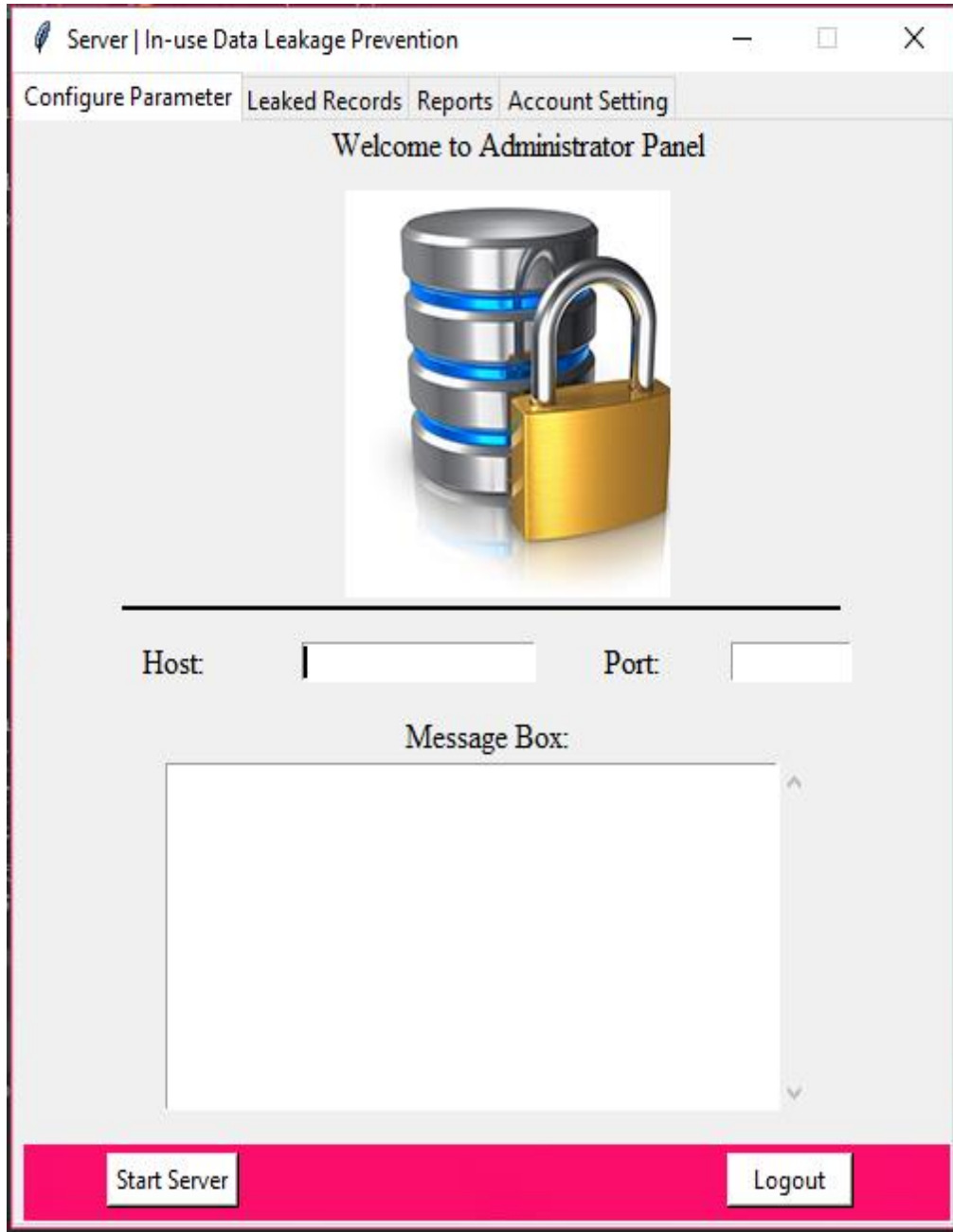


Figure 4. 7 Configure Parameter Interface

4.8.7 Leaked Records Interface

Figure 4.8, is leaked records interface. Administrator can search for logged information about printing task performed by authorize users.

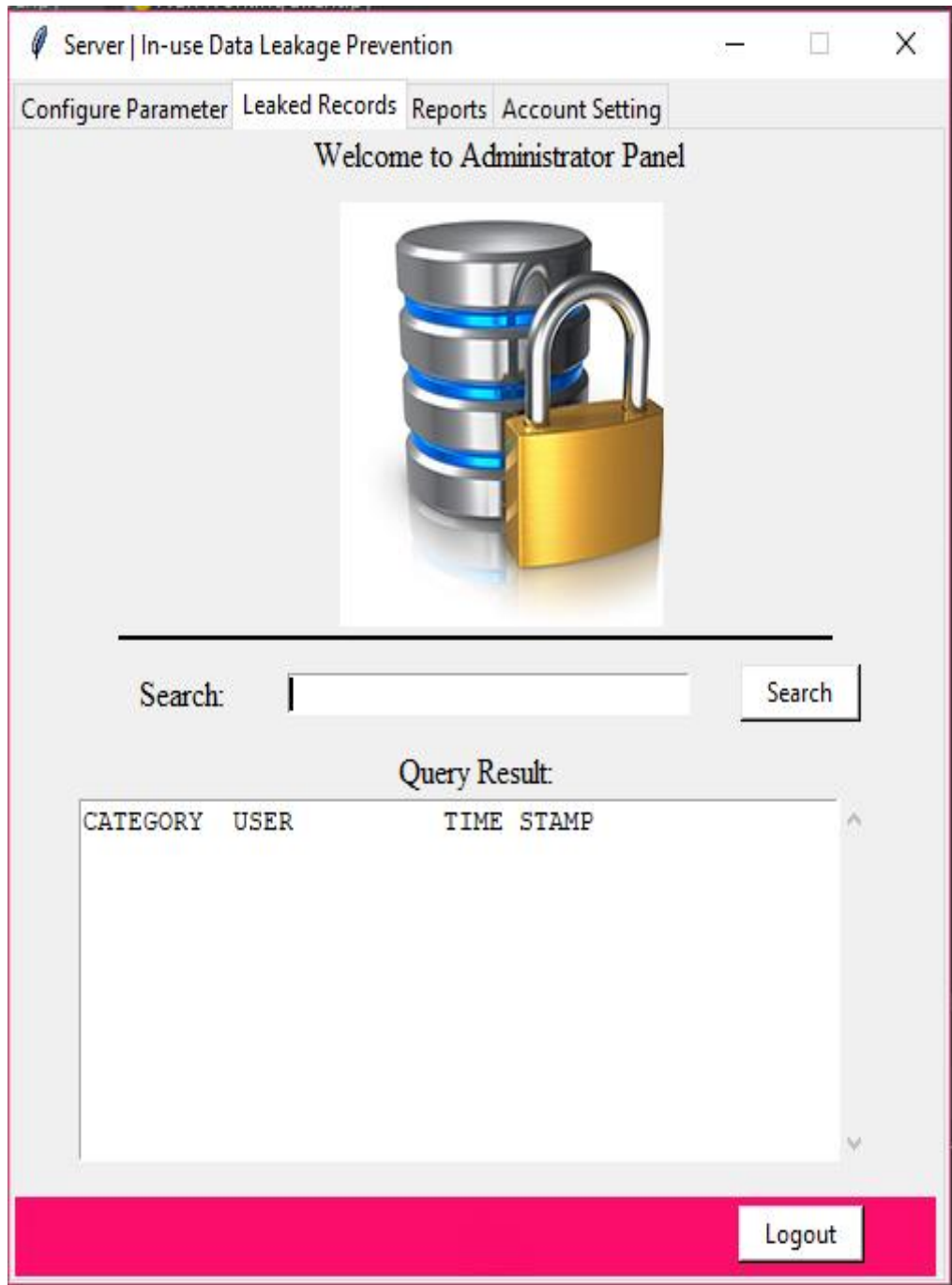


Figure 4. 8 Leaked Records Interface

4.8.8 Reports Interface

Figure 4.9, is reports interface. Administrator can search and export logged information to external file for further use.

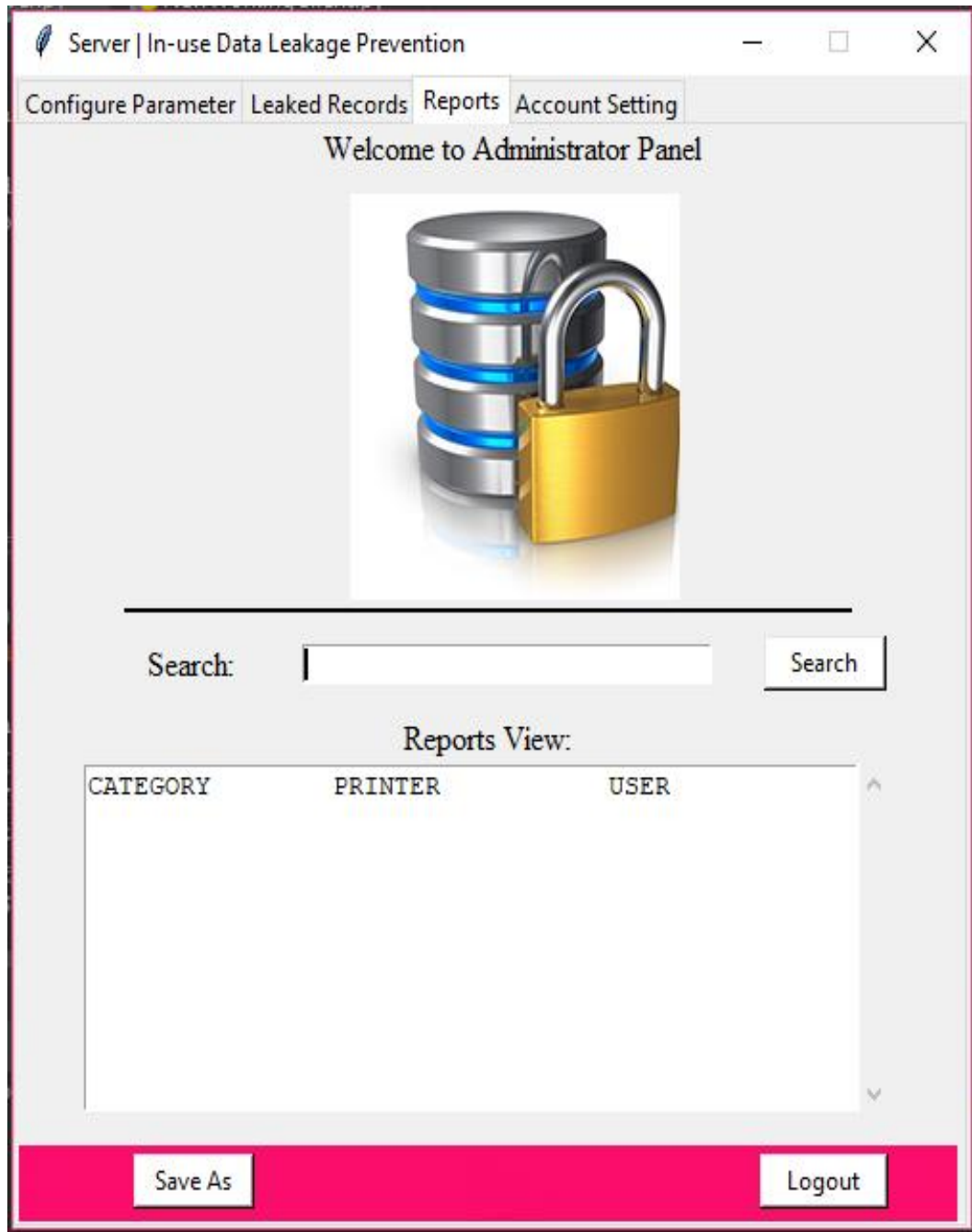


Figure 4. 9 Reports Interface

4.8.9 Account Setting Interface

Figure 4.10, is account setting interface. Administrator can able to change login password.

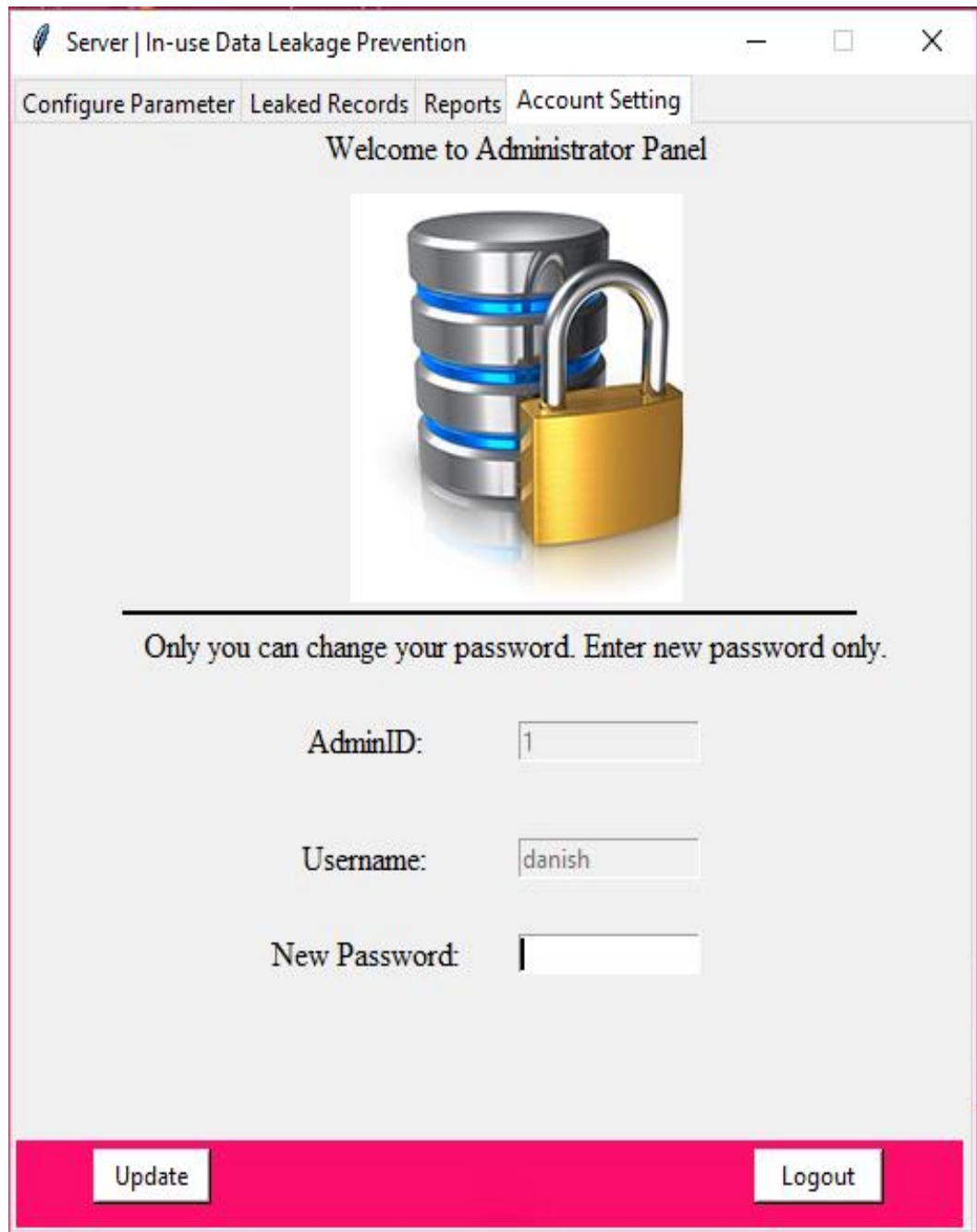


Figure 4. 10 Account Setting Interface

4.8.10 Secret Leak prevention Interface

Figure 4.11, is secret leak prevention interface. Sever alert prompt have details related to printing secret document and it has been cancelled by the system.

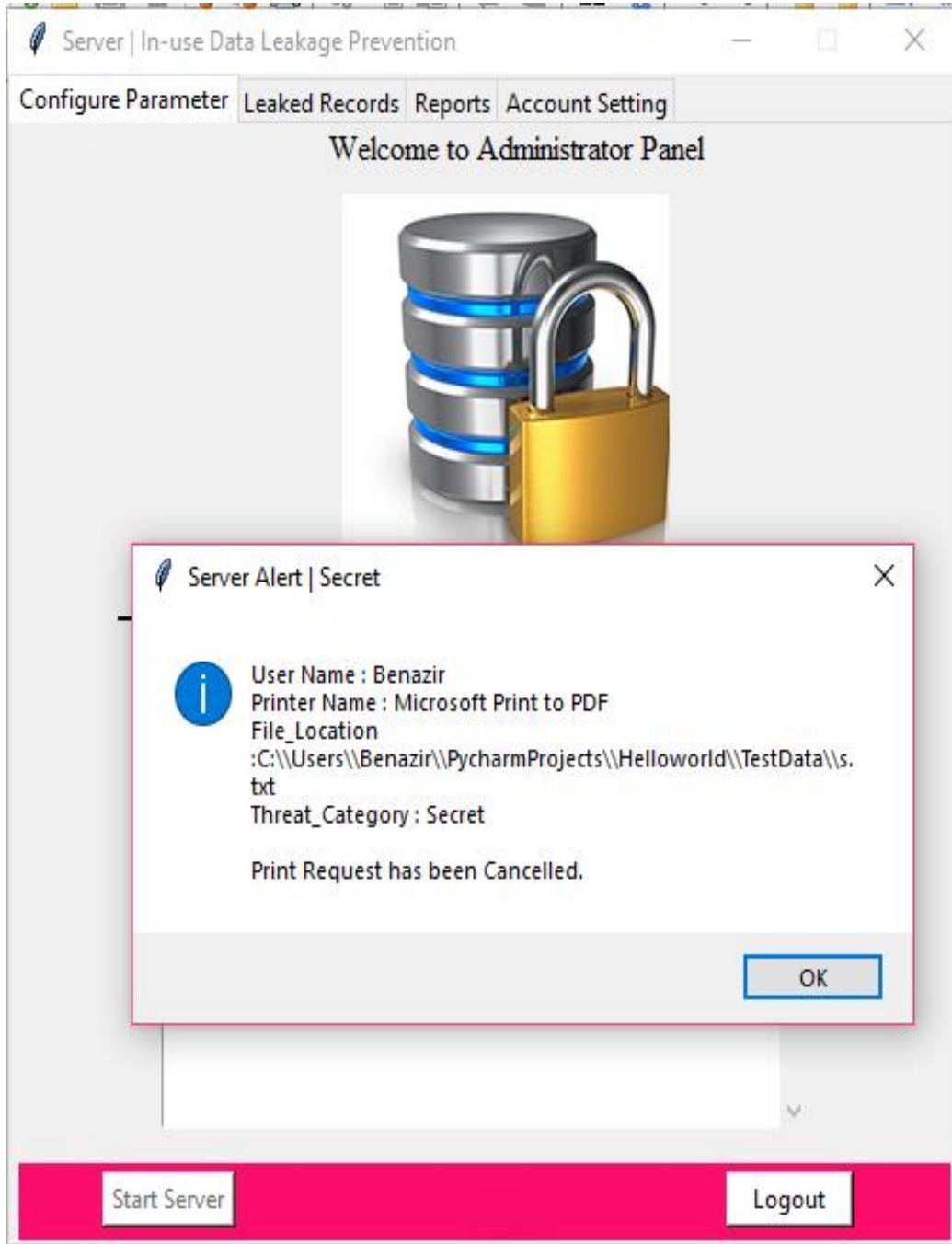


Figure 4. 11 Secret Leak Prevention Interface

4.8.11 Confidential Leak prevention Interface

Figure 4.12, is confidential leak prevention interface. Sever alert prompt have details related to printing confidential document and it has been paused by the system and ask administrator, whether this document is allowed to print or not. If administrator click on Yes button, paused status is change to resume and if administrator click on No button, printing request will be cancelled.

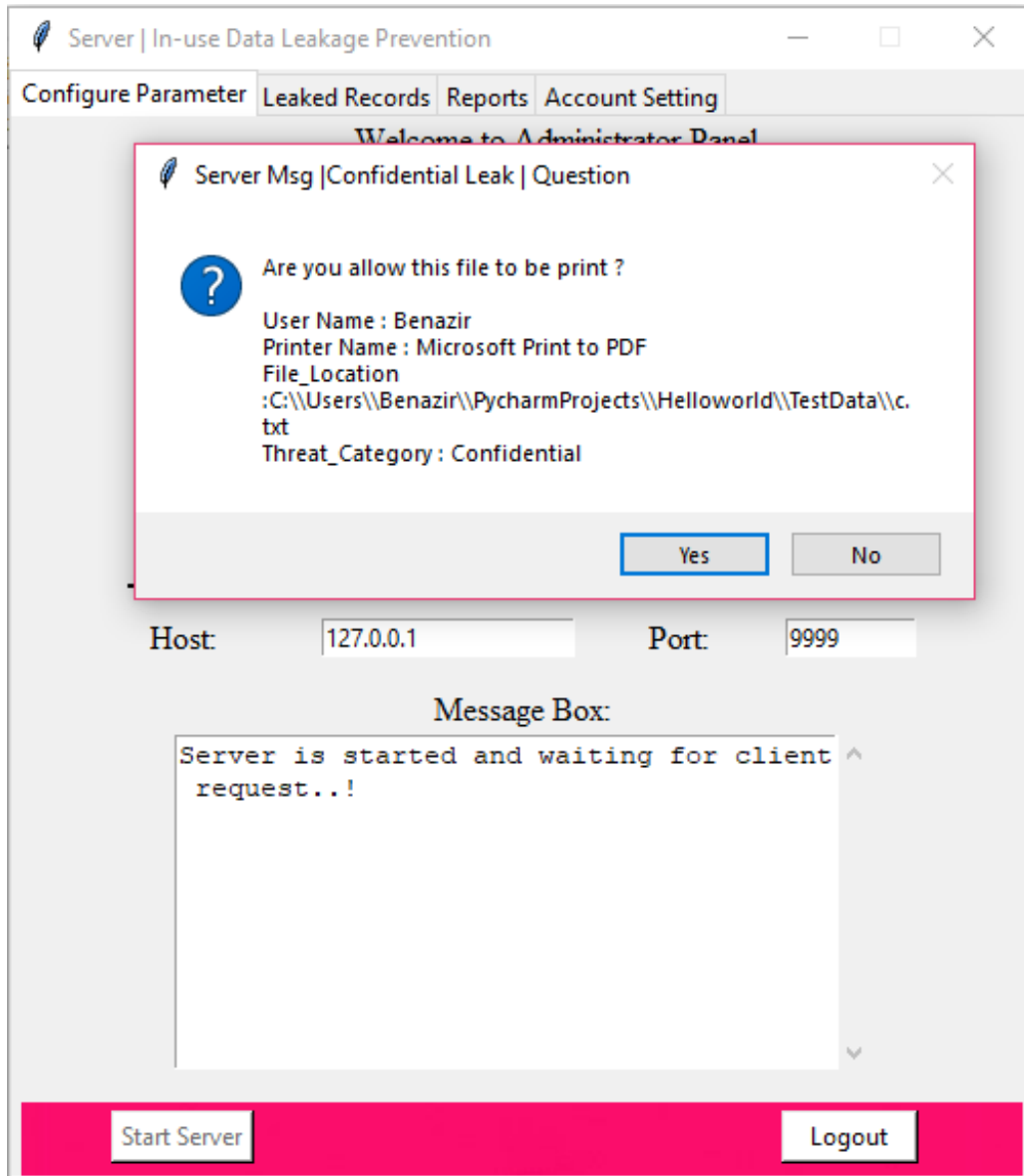


Figure 4. 12 Confidential Leak Prevention Interface

4.8.12 Non-Confidential Leak prevention Interface

Figure 4.13, is non-confidential leak prevention interface. Sever alert prompt have details related to printing non-confidential document and not action taken for this threat category by the system.

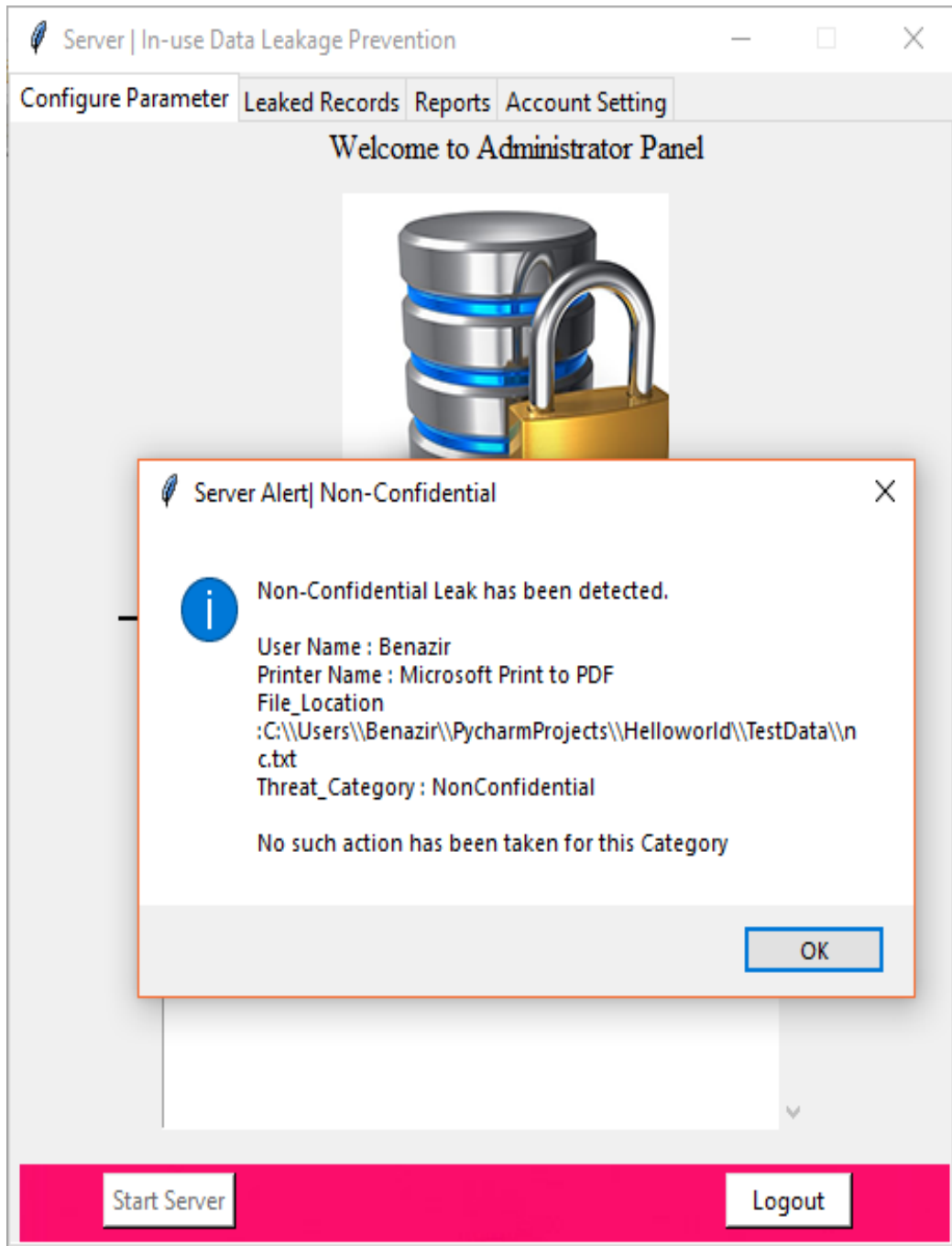


Figure 4. 13 Non-Confidential Leak Interface

4.8.13 Client Interface

Figure 4.14, is client end interface. Client end is also known as an Agent.

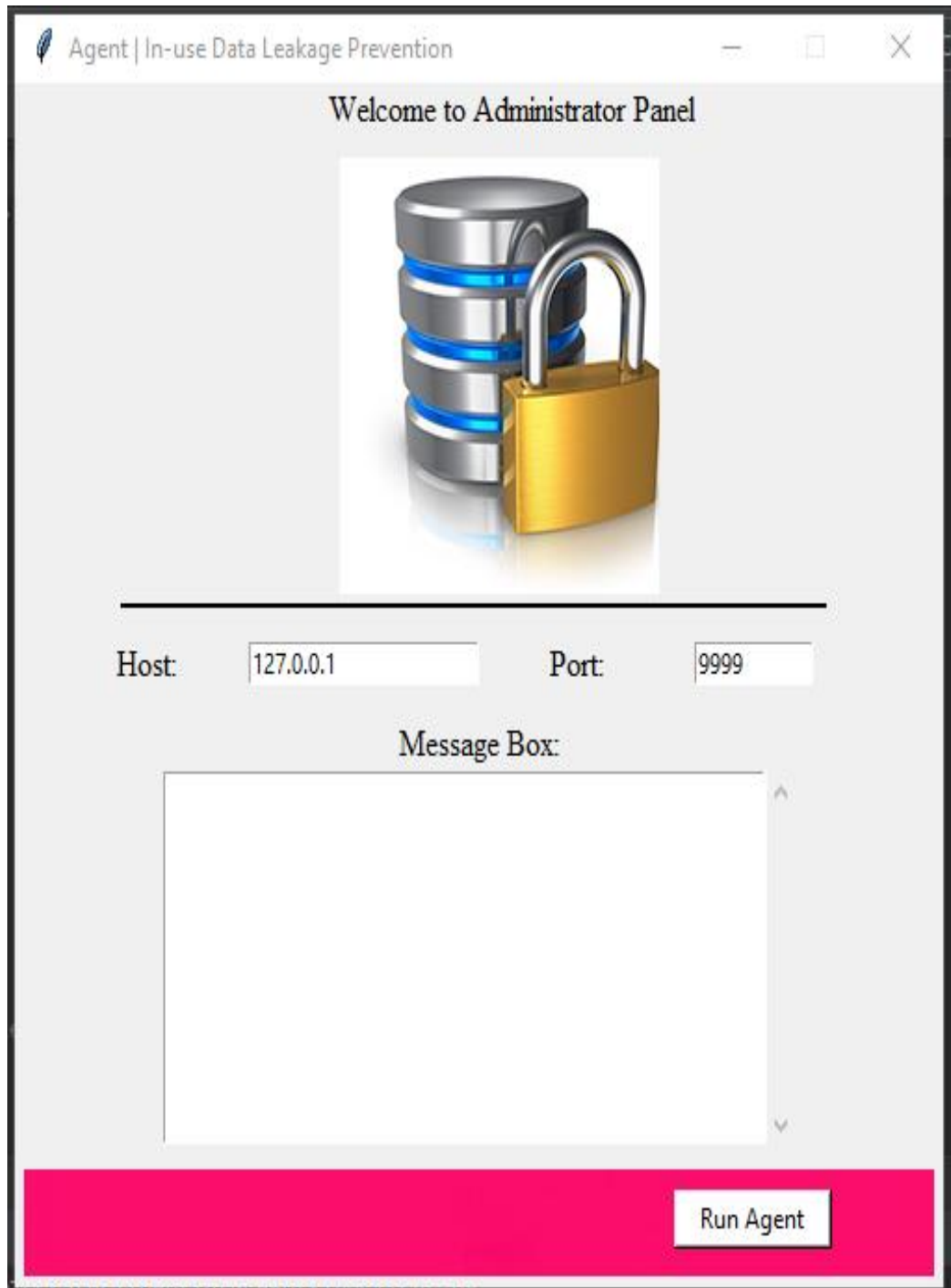


Figure 4. 14 Client Interface

This chapter has provided a description of the software implementation document. The descriptions regarding choose framework, language, algorithms, data classification, working system flow and interfaces of developed system. Software testing techniques have discussed in the next chapter.

*“Education is the ability to listen to almost anything without losing your self-confidence.”
Robert*

Chapter 5

Software Test Document

This chapter explains the software testing process. It further elaborates the acceptance test cases which are used to test the functional and non-functional requirements after the development of the software.

5.1 Introduction

Software test document involves the documentation of objects that should be developed before or during the testing of software. Software testing is the process of assessing a system or its components with the intent to find whether it satisfies the specified requirements or not.

5.1.1 Test Approach

Manual testing includes testing software manually without using any automated tool or any script [9]. The tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. Here in this work, the tester is admin. There are different stages for manual testing such as unit testing, system testing, and user acceptance testing. Admin only develop acceptance test plans, acceptance test cases, or acceptance test scenarios to test software to ensure the reliability of the system.

Test-first development is an approach to development where tests are written before the code to be tested. Small code changes are made and the code is refactored until all tests execute successfully.

5.2 Test Plan

Test planning is an activity that ensures that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project. Test plan determines the scope and the risks that need to be tested and are not to be tested. Deciding fail and pass criteria.

5.2.1 Features to be tested

All features of the system, which were defined in the software requirement specifications, need to be tested.

- Correct detection of data class (Secret, confidential, non-confidential)
- Capture print operation
- Cancel print operation
- Pause print operation
- Resume print operation
- Search Leak records
- Export Leak records
- Login successfully
- Signup successfully
- Password Change

5.2.2 Testing Tools and Environment

A testing environment is a setup of software and hardware on which testing of the newly built product is performed. This consists of the physical setup which includes computer system which have windows operating system, minimum 4 GB of RAM, and minimum 100 GB of hard drive. The developed application is properly connected with the logical database system.

5.3 Test Cases

A test case is a document, which has a set of test data, preconditions, expected results, and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

5.3.1 Signup

Table 5. 4 Signup Test Case

ID	T1
Description	Admin signup to the system for accessing the core functionality.
Tester	Admin
Setup	<ol style="list-style-type: none"> 1. Admin open desktop application 2. Go to Signup form
Instructions:	<ol style="list-style-type: none"> 1. Enter name, username, and password: Huzaifa, huzaifa12, and aa12 2. Click on signup button.

Expected Results	Show signed up successfully pop up alert.
Verdict	Passed

5.3.2 Login

Table 5. 5 Login Test Case

ID	T2
Description	Admin login to the system. It shows that login of admin can only be possible if username and password are correct.
Tester	Admin
Setup	<ol style="list-style-type: none"> 1. Admin open desktop application 2. Go to login form
Instructions:	<ol style="list-style-type: none"> 3. Enter Username and password: huzaifa12 and aa12 4. Click on login button.
Expected Results	Show pop up message “You logged in successfully”
Verdict	Passed.

5.3.3 Configure parameters

Table 5. 6 Configure Parameters Test Case

ID	T3
Description	The purpose of this test case is whether the input parameters will be preserved correctly.
Tester	Admin
Setup	<ol style="list-style-type: none"> 1. Admin open desktop application 2. Admin go to configure parameters tab
Instructions:	<ol style="list-style-type: none"> 1. Enter host and port no
Expected Results	The given parameters for the system are correctly input to ServerInterface.
Verdict	Passed

5.3.4 Account Setting

Table 5. 7 Account Setting Test Case

ID	T4
Description	To check whether system can update new password changes to database. Administrator have rights to edit login credentials.
Tester	Admin
Setup	<ol style="list-style-type: none"> 1. Admin open desktop application 2. Admin go to account setting tab.
Instructions:	<ol style="list-style-type: none"> 1. Enter new password. 2. Click on update button.
Expected Results	Desired login updates change successfully to database.
Verdict	Passed

5.3.5 Export Leaked Reports

Table 5. 8 Export Leaked Reports Test Case

ID	T5
Description	After detecting the data leaks. Its report will be useful for future enhancements within an organization. Therefore, all the reports have been maintained in database and admin can only store it to file.
Tester	Admin
Setup	<ol style="list-style-type: none"> 1. Admin open desktop application 2. Admin go to reports tab
Instructions:	<ol style="list-style-type: none"> 1. Click on save as button
Expected Results	Desired file will be created successfully.
Verdict	Passed

5.3.6 Correct detection of data class

Table 5. 9 Correct detection of data class Test Case

ID	T6
Description	Raw data is classified into three categories as non-confidential, confidential and secret. The files are classified into these three categories.
Tester	Admin

Setup	1. Admin open client end interface
Instructions:	1. Click on Run agent button 2. Give print to a file
Expected Results	file is correctly identified to its true category.
Verdict	Passed

5.3.7 Capture Print Operation

Table 5. 10 Capture Print operation Test Case

ID	T7
Description	Capture printing event in windows operating system. When a legitimate user wants to print some documents, the system captures its sequence wise operation.
Tester	Admin
Setup	1. Admin open desktop application 2. Admin go to client interface
Instructions:	1. Click on run agent button. 2. Open a document on the laptop. 3. Give command to print.
Expected Results	It captures the following print event attributes; print operation, Job id, printer name, user name and file location.
Verdict	Passed

5.3.8 Cancel Print Operation

Table 5. 11 Cancel print operation Test Case

ID	T8
Description	Cancel printing event in windows operating system, when a legitimate user wants to print some secret documents, the system captures its sequence wise operation.
Tester	Admin
Setup	1. Admin open desktop application 2. Go to client interface
Instructions:	1. Click on run agent button 2. Open a secret document. 3. Give command to print.

Expected Results	If threat category is found secret, it should cancel the print request.
Verdict	Passed

5.3.9 Pause Print Operation

Table 5. 12 Pause print operation Test Case

ID	T9
Description	Pause printing event in windows operating system, when a legitimate user wants to print some document, the system captures its sequence wise operation.
Tester	Admin
Setup	<ol style="list-style-type: none"> 1. Admin open desktop application 2. Go to client interface
Instructions:	<ol style="list-style-type: none"> 1. Click on run agent button 2. Open a confidential document. 3. Give command to print.
Expected Results	If threat category is found confidential, system pause the print request, it should ask from admin whether the requested print is allowed for printing. Admin not allowed for this threat category and system cancel the print request.
Verdict	Passed

5.3.10 Resume Print Operation

Table 5. 13 Resume print operation Test Case

ID	T10
Description	Resume printing event in windows operating system, when a legitimate user wants to print some document, the system captures its sequence wise operation.
Tester	Admin
Setup	<ol style="list-style-type: none"> 1. Admin open desktop application 2. Go to client interface
Instructions:	<ol style="list-style-type: none"> 1. Click on run agent button 2. Open a confidential document. 3. Give command to print.
Expected Results	If threat category is found confidential, system pause the print request, it should ask from admin whether the requested print is allowed for printing. Admin allowed for this threat category and

	system resume the print request.
Verdict	Passed

This chapter has provided the description of software testing of the system. It has elaborated the test case approach, test plan, and test cases. The next chapter further explained conclusion and future enhancements in the system.

“Education is most power weapon which you can use to change the world.” Nelson Mandela

Chapter 6

Conclusion and Future Enhancements

6.1 Summary

The product defines the distributed desktop application known as “In-use Data Leakage Prevention in Microsoft Windows Operating System using Naïve Bayes Classifier”. The purpose of the system is to ensure the security of organizational data which cannot be revealed to outsiders through organization’s employees. In this work, we focused on printing device (printer) through which important data can be printed by authorized users and the printed paper can be taken outside of the organization. This product performs three main operations based upon data classes which a printed document belongs. If the system finds that the printing document belongs to the secret class, then it can cancel the printing request. If the system finds that the printing document belongs to the confidential class, the system pauses the printing request temporarily and asks a question about this printing request from the admin, whether the admin should allow this request or not. If the admin allows the request, then the system changes the status of the printing request from pause to resume, else the system can cancel the printing request. If the system finds that the printing document belongs to the non-confidential class, then the system cannot take any action, because all documents belong to that class are not important. The records of the printing request have been maintained in the database.

6.2 Future Enhancements

The system enhancements and future work will be done by:

- Including a policy module. Policy is basically changing of file rights and apply remedial actions (such as cancel print, pause print, and resume print). When a client repeatedly gives a print to a secret document, the system automatically changes the file permissions for that client e.g. previous permission on secret file (read, write), after changing permission on secret file (remove both permission)
- Making the system as a service for the window operating system.

- Ensure portability of the current system.
- Include similar mechanism to the system through which data leakage can be blocked.

The understanding and the code written for this system would act as an aid for further development on the similar concepts. This would serve as a helpful guide for students and novice distributed system developers who want to develop such similar applications. Such as making controlling security mechanism for copying data through USB ports and CD-drives.

References

- [1] Detecting data semantic: A data leakage prevention approach by Sultan Alneyadi, Elankayer Sithirasanen, Vallipuram Muthukkumarasamy School of Information and Communication Technology, Griffith University Gold Coast, Australia.
- [2] S. Matic, A. Fattori, D. Bruschi, and L. Cavallaro, "Peering into the Muddy Waters of Pastebin," ERCIM News: Special Theme Cybercrime and Privacy Issues, p. 16, 2012.
- [3] Chapter 13. Design Concepts and Principles, Software Engineering A Practitioner's Approach by R.S. Pressman.
- [4] 14.5 Class Diagrams Software engineering A lifecycle approach by P. Mohapatra Software Engineering A Practitioner's Approach by R.S. Pressman,
- [5] sklearn.feature_extraction.text.CountVectorizer
- [6] sklearn.feature_extraction.text.TfidfVectorizer
- [7] sklearn.naive_bayes.MultinomialNB
- [8] <http://timgolden.me.uk/pywin32-docs/win32print.html>
- [9] Chapter 8. Software Testing Ian_Sommerville _Software_Engineering_9th_edition.