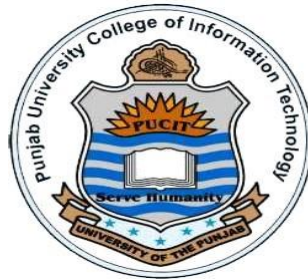


Final Year Design Project

**ANTI-RANSOMWARE SECURITY SOLUTION FOR
MICROSOFT WINDOWS**



By

Abdul Samad	2021-fqr-1
Abdul Rehman	2021-fqr-3
Muhammad Aamir Bakhsh	2021-fqr-22

Under the supervision of

Mr. Muhammad Zeeshan (Lecturer Computer Science)

***Bachelor of Science in Information Technology
(2021-2025)***

**FACULTY OF COMPUTING &
INFORMATION TECHNOLOGY (FCIT),
UNIVERSITY OF THE PUNJAB, LAHORE.**

Anti-Ransomware Security Solution for Microsoft Windows

**A project presented to
University of the Punjab, Lahore**

**In partial fulfilment
Of the requirement for the degree of**

***Bachelors of Science in Information Technology
(2021-2025)***

By

Abdul Samad	2021-fqr-1
Abdul Rehman	2021-fqr-3
Muhammad Aamir Bakhsh	2021-fqr-22

**FACULTY OF COMPUTING &
INFORMATION TECHNOLOGY (FCIT),
UNIVERSITY OF THE PUNJAB, LAHORE**

DECLARATION

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other, we will stand by the consequences. No portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Signature: -----

Abdul Samad - 066678

Signature: -----

Abdul Rehman - 066679

Signature: -----

Muhammad Aamir Bakhsh - 066689

CERTIFICATE OF APPROVAL

It is to certify that the final year design project (FYDP) of BSIT “**Anti-Ransomware Security Solution for Microsoft Windows**” was developed by **ABDUL SAMAD (2021-fqr-1)**, **ABDUL REHMAN (2021-fqr-3)** and **MUHAMMAD AAMIR BAKHSH (2021-fqr-22)** under the supervision of “**Mr. MUHAMMAD ZEESHAN**” in my opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Information Technology.

Signature: -----
FYDP Supervisor:

Signatures (Faculty Advisory Committee (FAC))

Signatures			

Signature: -----
Head of FYDP Coordination Office:

Signature: -----

Chairperson, Department of Information Technology

Dated: _____

Executive Summary

Ransomwares are one of the biggest cyber threats in today's world. It can affect anyone like individuals, students and businesses. These attacks encrypt your important files and demand money to decrypt your data. This can cause problems like losing data, financial loss, and lack of your reputation.

Our project focuses on creating an easy-to-use Anti-Ransomware Security Solution for Microsoft Windows. It is designed in a way so that non-technical users can also use it. This system will check what is happening on your device, and then detects activities like unusual file encryption.

Our project also provides back up feature to back up your important/critical data. It also includes antivirus features to detect viruses and actionable prevention.

Our main goal is to keep everyone safe from these problems.

Acknowledgement

The acknowledgment section is an opportunity to express gratitude and appreciation to individuals and organizations who have contributed to the completion of your Final Year Project (FYP).

Signature: -----

Abdul Samad - 066678

Signature: -----

Abdul Rehman - 066679

Signature: -----

Muhammad Aamir Bakhsh - 066689

Abbreviations

Table of all the abbreviations and acronyms used in your FYP along with their respective brief description.

BL	Backlog list: List of the requirements under consideration
FCIT	Faculty of Computing and Information Technology
FYP	Final Year Project
GUI	Graphical User Interface
UI	User Interface
OS	Operating System

Table of Contents

ANTI-RANSOMWARE SECURITY SOLUTION FOR MICROSOFT WINDOWS	1
DECLARATION	3
CERTIFICATE OF APPROVAL	4
Executive Summary	5
Acknowledgement	6
Abbreviations	7
Table of Contents	8
Chapter 1 Introduction	14
1. Introduction	15
1.1 Problem Statement	15
1.2 Problem Solution	15
1.3 Objectives of the Proposed System	15
1.4 Scope	15
1.5 System Components	16
1.5.1 Module 1: Ransomware Detection and Prevention Module	16
1.5.2 Module 2: Data Backup Module	16
1.5.3 Module 3: Notification Module	16
1.5.4 Module 4: Antivirus Module	16
1.5.5 Module 5: Security Logs Viewer Module	16
1.5.6 Module 6: User Interface (UI) Module	16
1.6 Related System Analysis / Literature Review	16
1.7 Vision Statement	17
1.8 System Limitations and Constraints	17
1.9 Tools and Technologies	17
1.10 Project Deliverables	19
1.11 Project Planning	19
1.12 Summary	20
Chapter 2 Requirements Analysis	21
2. Analysis	22
2.1 User classes and characteristics	22
2.2 Requirement Identifying Technique	22
2.3 Use Case Analysis	22
2.3.1 Use Case #1 (Ransomware Scanning)	22
2.3.2 Use Case #2 (Virus Scanning)	23
2.3.3 Use Case #3 (Data Backup)	23
2.3.4 Use Case #4 (View Logs)	24
2.4 Use Case Diagram	24
2.5 Functional Requirements	26
2.5.1 Functional Requirement 1 – Ransomware Detection	26

2.5.2	Functional Requirement 2 – Ransomware Prevention.....	26
2.5.3	Functional Requirement 3 – User Alert System.....	26
2.5.4	Functional Requirement 4 – Virus Scanning	26
2.5.5	Functional Requirement 5 – Data Backup	26
2.6	Non-Functional Requirements	29
2.6.1	Reliability	29
2.6.2	Usability.....	29
2.6.3	Performance.....	29
2.7	External Interface Requirements.....	29
2.7.1	User Interface Requirements.....	29
2.7.2	Software Interface	29
2.7.3	Hardware Interface	29
2.7.4	Communications Interface.....	30
2.8	Summary.....	30
<i>Chapter 3.....</i>		<i>31</i>
3.	System Design	32
3.1	Design Considerations.....	32
3.2	Design Models.....	32
3.2.1	Class Diagram	32
3.3	Data Design	35
3.3.1	Data Dictionary.....	35
3.3.2	List of Attributes in Ransomware Detection Dataset with Type and Description.....	35
3.3.3	List of System Entities and Data with Types and Descriptions	35
3.3.4	Functions and Functions Parameters.....	36
3.4	User Interface Design.....	37
3.4.1	Screen Images	37
	Dashboard.....	37
	Threats History	37
	Scan Files	38
	Backup Data	38
3.5	Behavioral Model	39
3.5.1	Activity Diagram	39
3.6	Summary.....	40
<i>Chapter 4 Implementation</i>		<i>41</i>
4.	Implementation	42
4.1	Algorithm	42
4.2	External APIs/SDKs.....	44
4.3	Code Repository	45
4.4	Summary.....	45
<i>Chapter 5.....</i>		<i>46</i>
5.	Introduction	47
5.1	Unit Testing (UT).....	47
5.2	Functional Testing (FT)	47
5.2.1	Ransomware Detection.....	47

5.2.2	Ransomware Prevention	48
5.2.3	User Alert System	48
5.2.4	Virus Scanning	48
5.2.5	Data Backup	48
5.3	Integration Testing (IT)	48
5.3.1	Ransomware Detection	48
5.3.2	Ransomware Prevention	48
5.3.3	User Alert System	48
5.3.4	Virus Scanning	48
5.3.5	Data Backup	48
5.4	Performance Testing (PT)	48
5.4.1	Virus Scanning	49
5.4.2	Backup Module	49
5.4.3	User Alerts	49
5.4.4	Anti-Ransomware Tool	49
5.5	Summary	49
Chapter 6.....		51
6.	Introduction	52
6.1	Conversion Method	52
6.2	Deployment	52
6.2.1	Data Conversion	52
6.2.2	Training	52
6.3	Post Deployment Testing	52
6.4	Summary	52
Chapter 7 Conclusion.....		53
7.	Introduction	54
7.1	Evaluation	54
7.2	Traceability Matrix	54
7.3	Conclusion.....	55
7.4	Future Work	55
References.....		56
AppendixA		57
Use case Description Template.....		57
Appendix-A Use Case Description (Fully Dressed Format)		58
Appendix-A Use Case Description (Fully Dressed Format)		59
Appendix-B Coding Standards		60
Appendix-B General Coding Standards & Guidelines		61
1.	Naming Style.....	61
2.	Clear Variable Names	61
3.	Function Names	61
4.	Comments	61
5.	Indentation	61
6.	Separate Code Files.....	61

7. One Task Per Function.....	61
8. Reuse Code	61
9. Error Handling.....	61
10. Logging Actions.....	61
<i>Appendix C Prototype</i>	<i>62</i>
Dashboard.....	63
Threat History	63
Scan files	63
Backup	63
Switch Language	63
Quick Access.....	63

List of Tables

Table 1: Tools and Technologies for Proposed Project	17
Table 2: Project Deliverables	19
Table 3: User Classes	22
Table 4: Ransomware Scanning	22
Table 5: Virus Scanning	23
Table 6: Data Backup	23
Table 7: View Logs.....	24
Table 9: Description of FR-1 – Ransomware Detection	26
Table 10: Description of FR-2 - Ransomware Prevention	27
Table 11: Description of FR-3 – User Alert System.....	27
Table 12: Description of FR-4 – Virus Scanning	28
Table 13: Description of FR-5 – Data Backup	28
Table 14: Dataset Attributes	35
Table 15: System Entities and Data	36
Table 16: System Functions and Function Parameters.....	36
Table 17: Algorithm - Anti-Ransomware	42
Table 18: Algorithm - Antivirus	42
Table 19: Algorithm - Backup Module	44
Table 20: Details of API used in the project.....	44
Table 21: Unit Testing	47
Table 22: Evaluation Matrix.....	54
Table 23: Traceability Matrix.....	54
Table 24: Detailed Use Case Template – Ransomware Scanning	58
Table 25: Detailed Use Case Template – Virus Scanning	59

List of Figures

Figure 1: Gantt Chart.....	20
Figure 2: Use Case Diagram for Anti Ransomware System	25
Figure 3: Class Diagram 1.....	33
Figure 4: Class Diagram 2.....	34
Figure 5: Class Diagram 3.....	34
Figure 6: Dashboard	37
Figure 7: Threat History/Logs	38
Figure 8: Scan Files.....	38
Figure 9: Data Backup.....	39
Figure 10: Activity Diagram	40
Figure 11: Antivirus Scan	49
Figure 12: Backup.....	49
Figure 13: Anti-Ransomware Tool.....	49

Chapter 1

Introduction

1. Introduction

Ransomware is a serious cyber threat that affects people and businesses by encrypting their data. They demand money to decrypt it. These attacks are becoming more common nowadays.

This project focuses on creating a Ransomware Detection and Protection System. It will detect ransomware attacks and it will notify users about them to take action against it whether they want to delete it or not.

Our project also has an antivirus feature that will detect viruses in a user's system. It will detect threats and viruses, and then shows a popup/notification to the user about them. It also has a backup option to back up your important data. This solution is easy-to-use for technical and non-technical users.

1.1 Problem Statement

Ransomware attacks are major issues nowadays that affects students, technical users, non-technical users and businesses. These attacks encrypt important data and after encrypting, they demand heavy ransom/money to decrypt it. This leads to data loss and financial problems because of demanding heavy ransom.

1.2 Problem Solution

Ransomware attacks are increasing quickly these days that affect people and businesses by encrypting their data. Some antivirus tools often fail to detect ransomware in real time. So, a solution is needed that can detect, prevent, and scan those threats and provides a solution on Microsoft Windows.

It will detect and prevent ransomware attacks.

It will detect and prevent from viruses also.

It will notify the users about threats.

It will provide a backup feature to back up your important data.

It is useful for non-technical users too.

1.3 Objectives of the Proposed System

The objective of this project are as follows:

It will detect and prevent ransomware attacks and viruses on Microsoft Windows.

It will notify the user about the threats.

It has integrated antivirus feature.

It has back up feature that will help users to safe their important files.

It has a user-friendly interface so that technical and non-technical users can use it.

1.4 Scope

The Anti-Ransomware Security Solution for Microsoft Windows is a desktop-based application for Microsoft Windows users to detect and prevent ransomware attacks. It also has other features like threat detection and actionable prevention, virus scanning, user alerts and data backup. We designed this application in user-friendly interface so that the people who know about the computer and people who have minimal knowledge about the computer, everyone can use it because it is easy-to-use. It uses Python language and its libraries/technologies. This system is currently limited to Windows and Python, and it will not support mobile and macOS platforms.

1.5 System Components

The system consists of the following components:

1.5.1 Module 1: Ransomware Detection and Prevention Module

It monitors files in real-time.

It will identify encryption patterns.

It will generate alerts when any threat is detected.

It will notify the users whether they want to terminate the processes or no.

It will save logs for future use.

1.5.2 Module 2: Data Backup Module

It allows users to back up their important data.

It supports Google Drive back up.

1.5.3 Module 3: Notification Module

It shows real-time alerts when any threat is detected.

It provides options (actionable) like delete (yes/no).

1.5.4 Module 4: Antivirus Module

It uses ClamAV to scan files.

It detects viruses.

It provides options for actions like delete (yes/no).

1.5.5 Module 5: Security Logs Viewer Module

It allows users to view history of past scans and data upload.

It displays logs details in a readable format.

1.5.6 Module 6: User Interface (UI) Module

It was developed using Python's Tkinter library.

It shows scan options, history, backup controls and also a quick access.

It provides buttons for scanning, view logs, user help and backup.

It has simple design for non-technical users.

1.6 Related System Analysis / Literature Review

Ransomware has become a serious cybersecurity problem these days that encrypts user data and demands money/ransom to decrypt it. It affects all type of people and businesses.

Some research studies, like those by Kharraz et al. (2015), explain that ransomware can be caught by watching unusual behavior like strange file access or very fast encryption of data, we understood the importance of behavior-based detection. MITRE. (2020). T1486: Data Encrypted for Impact. This framework classifies ransomware behavior (e.g., encrypting data, deleting backups), it helped us identify what kind of activity to monitor and how to classify a threat. Subhajit Bhattacharya and Deepak Dakhane, "A Signature-Based Ransomware Detection and Automated Data Backup," International Journal of Information Security and Applications, vol. 8, no. 3, pp. 12–18, 2024. Our project will detect ransomware and also has a backup option to keep

important files safe.

1.7 Vision Statement

Here is the vision statement of our project:

For Microsoft Windows Users

Who need easy and efficient solution against ransomware attacks and viruses

The Anti Ransomware Security Solution for Microsoft Windows

Is a desktop based application

That detects ransomware, scan viruses, and provides backup feature with low user effort

Unlike old programs do not provide comprehensive solution

Our product combines monitoring against ransomware attacks, behavioral analysis, antivirus integration, backup feature in one user friendly solution.

1.8 System Limitations and Constraints

Limitations:

The system may not detect all new ransomware because it has access to limited real world threat data.

The tool may generate false-positive detections when analyzing files that are encrypted by the operating system or other legitimate system-level encryption services.

The system might run a litter slower on old devices.

Constraints:

This project only works on computers that use Windows operating system.

This project is made using Python and its libraries (that are used only in Python) like ClamAV, Tkinter, and Psutil.

1.9 Tools and Technologies

Following are the tools and technologies used in the project:

Table 1: Tools and Technologies for Proposed Project

	Tools	Version	Rationale
	ClamAV	1.4.2	ClamAV is an open-source antivirus engine that is used for scanning files.
	Watchdog	6.0.0	Watchdog is a Python API and shell utilities to monitor file system events.
	Libraries	Version	Rationale
	os	-	It is used for file and directory operations.
	pickle	Python	It is used to save and load Python objects.

**Tools,
Libraries,
And
Technologies**

	Buit-in, no version	
threading	Python Buit-in, no version	It is used to run multiple tasks at the same time.
subprocess	Python Buit-in, no version	It is used to run commands on the computer system.
psutil	7.0.0	It is used to monitor system processes and resource usage.
logging	Python Buit-in, no version	It is used to record messages and errors in the app.
math	Python Buit-in, no version	It is used for mathematical calculations.
datetime	Python Buit-in, no version	It is used to work with date and times.
tkinter	-	It is used to make a desktop app with a user interface.
tkbootstrap	1.12.0	It is used to make the app interface modern.
pillow	11.1.0	Pillow is used to display images.
google-auth-oauthlib	1.2.1	It is used to handle Google sign-in security.
google-api-python-client	2.166.0	It is used to connect client with Google drive.
google-auth	2.38.0	It is used to manage google login.
clamd	1.0.2	Clamd is used to scan files for viruses.
scanners_utils	-	It is used to scan and delete infected files.
Technology	Version	Rationale

Google Drive API	v3	It is used to upload files to Google drive.
OAuth 2.0	2.38.0	It is used to sign-in securely to Google drive.

1.10 Project Deliverables

Table 2: Project Deliverables

Sr. No.	Aspect	Date	Status
1	Project Orientation	04-11-2024	Done
	Project Domain	06-11-2024	Done
	Project Title	25-11-2024	Done
	Theme	30-11-2024	Done
2	Project Proposal	30-12-2024	Done
3	SRS	14-03-2025	Done
	Functional Requirement	08-03-2025	Done
	Use Cases	10-03-2025	Done
	Non-Functional Requirement	12-03-2025	Done
4	SDS	22-03-2025	Done
	Class Diagram	16-03-2025	Done
	Activity Diagram	17-03-2025	Done
	User Interface Design	20-03-2025	Done
5	Implementation	09-05-2025	Done
	Testing and Debugging	03-08-2025	Done
6	Conclusion		
	Conclude FYP	03-09-2025	Done
	Future Work	--	Ongoing

1.11 Project Planning

Gantt Chart representing the plans of carrying out the project activities, representing the milestones as well as resources required for each task.

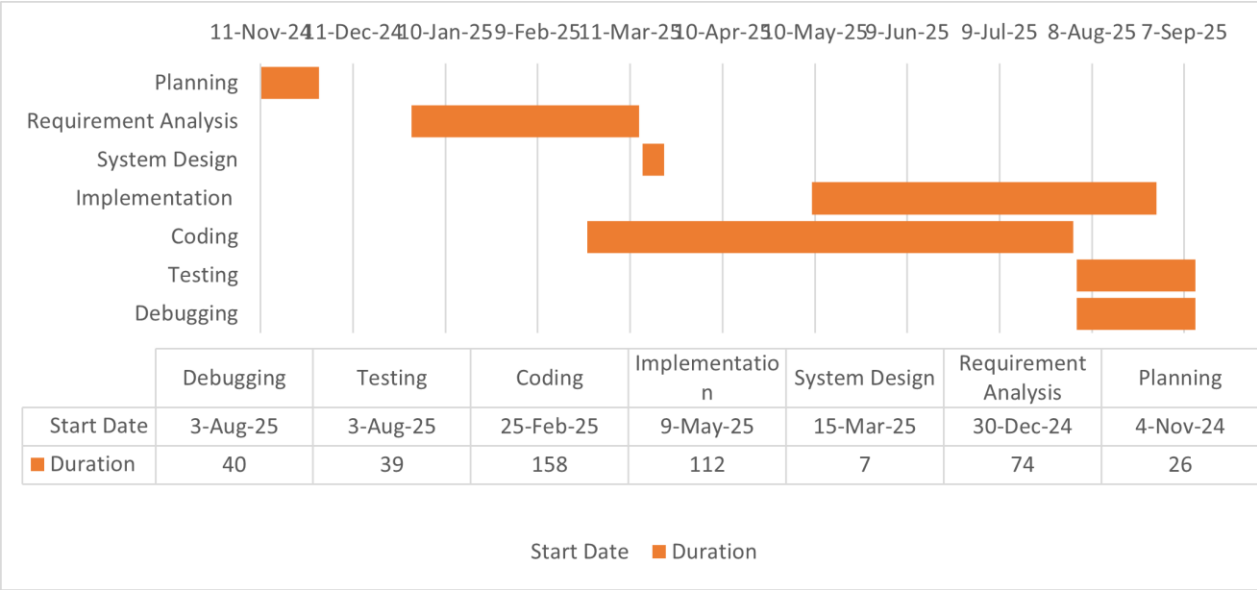


Figure 1: Gantt Chart

1.12 Summary

This chapter discussed the main idea of the project, its purpose, background, and scope. It also discussed the key features of our project that includes antivirus integration, ransomware detection, notifying the user, and data backup. The tools and technologies were defined to show how the project is developed. This chapter is important because it explains the project clearly and it tells what we are going to do next.

Chapter 2

Requirements Analysis

2. Analysis

Here, we will explain why we created Anti Ransomware Security Solution and how it will help users. It includes all the important features of our project, like ransomware detection and prevention, scanning files, sending notifications, and data backup. We will also describe functional and non-functional requirements. This chapter helps to understand our system.

2.1 User classes and characteristics

Following are the different types of users who are expected to use this system:

Table 3: User Classes

User Class	User Characteristics
General Users	Basic computer knowledge, need for protection from cyber threats, no technical expertise required.
Developer	Developers can view information about the application.

2.2 Requirement Identifying Technique

We used Use Case Analysis technique to identify the system requirements. This method outlines the interaction between the user and the system, and focuses on detecting ransomware threats.

2.3 Use Case Analysis

A use case describes how users interact with the system to achieve a certain goal. It describes the interaction between users and the system.

2.3.1 Use Case #1 (Ransomware Scanning)

Table 4: Ransomware Scanning

<i>UC Identifier</i>	<i>UC1</i>
<i>Requirements Traceability</i>	<i>It is related to ransomware detection and prevention.</i>
<i>Purpose</i>	<i>It detects and prevents ransomware activity based on file behavior.</i>
<i>Priority</i>	<i>High</i>
<i>Preconditions</i>	<i>The system is running and monitoring file behavior and active processes.</i>
<i>Post conditions</i>	<i>System will detect ransomware, notifies the user and show actions about it.</i>
<i>Actors</i>	<i>User</i>
<i>Extends</i>	<i>Action on threat, if a threat is detected during real-time monitoring.</i>
<i>Main Success Scenario</i>	<i>1. System monitors files and process in real-time. 2. It detects abnormal encryption patterns or high entropy. 3. User is alerted immediately via popup.</i>
<i>Alternate Flows</i>	<i>If no threat is detected, it will continue monitoring.</i>

Exceptions	<i>Detection may fail due to unknown ransomware variant.</i>
Includes	<i>Logging threat details. It will notify the user.</i>

2.3.2 Use Case #2 (Virus Scanning)

Table 5: Virus Scanning

UC Identifier	UC2
Requirements Traceability	<i>It is related to virus detection and file security.</i>
Purpose	<i>It will scan for viruses and other threats.</i>
Priority	<i>High</i>
Preconditions	<i>Clamd service should run in the background.</i>
Post conditions	<i>System will report whether the file is safe or infected.</i>
Actors	<i>User</i>
Extends	<i>Action on threat – If a file is found to be infected.</i>
Main Success Scenario	<i>1. User will be able to scan files manually and automatically. 2. System checks the file against viruses. 3. Then the user is notified about the results.</i>
Alternate Flows	<i>No alternate flows.</i>
Exceptions	<i>Scan failure due to system error or corrupted file.</i>
Includes	<i>None.</i>

2.3.3 Use Case #3 (Data Backup)

Table 6: Data Backup

UC Identifier	UC3
Requirements Traceability	<i>It is related to file safety.</i>
Purpose	<i>It allows user to manually and scheduled back up their important files.</i>
Priority	<i>Medium</i>
Preconditions	<i>User will select files to back up and cloud storage access is available.</i>
Post conditions	<i>Files are successfully stored in cloud storage.</i>
Actors	<i>User</i>
Extends	<i>Backup files – backup is successfully created.</i>
Main Success Scenario	<i>1. User will select files for backup. 2. After selecting the files, system will upload files to Google Drive.</i>
Alternate Flows	<i>If Google Drive is unavailable, data will not backup.</i>

Exceptions	<i>Network failure, insufficient storage.</i>
Includes	<i>None</i>

2.3.4 Use Case #4 (View Logs)

Table 7: View Logs

UC Identifier	UC4
Requirements Traceability	<i>This is related to system monitoring.</i>
Purpose	<i>It allows users to check the actions.</i>
Priority	<i>Medium</i>
Preconditions	<i>Log files exist.</i>
Post conditions	<i>User can view past security-related events.</i>
Actors	<i>User</i>
Extends	<i>None</i>
Main Success Scenario	1. All users will have access to logs. 2. System will display logs.
Alternate Flows	<i>If no logs exist, then it will show an empty message.</i>
Exceptions	<i>Log corruption, access failure.</i>
Includes	<i>None</i>

2.4 Use Case Diagram

Here is the use case diagram that shows how the system will interact with its users. Diagram is placed on the next page to make it clear and provide enough space. This diagram includes all the main actions and interactions between the system and its users.

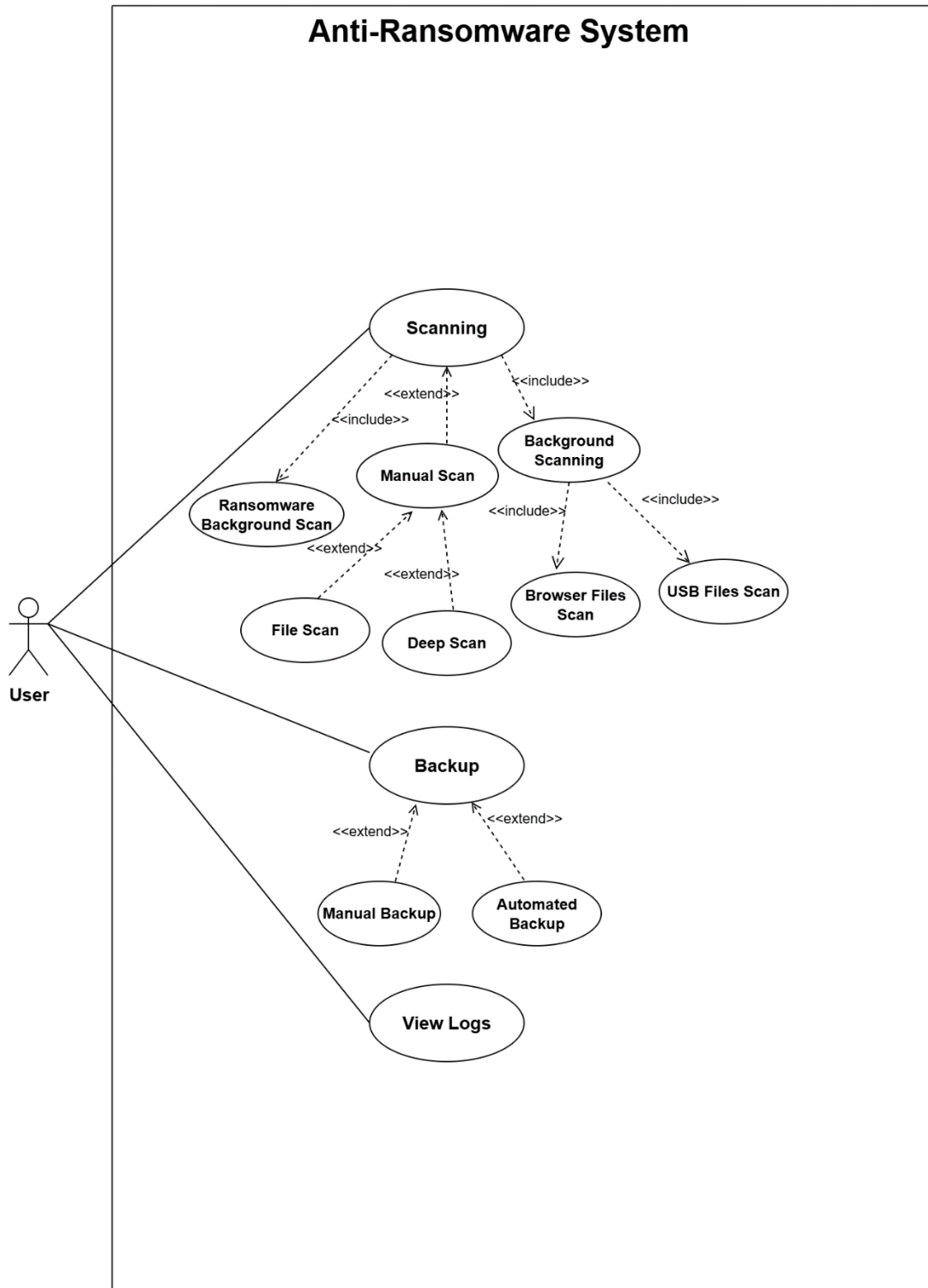


Figure 2: Use Case Diagram for Anti Ransomware System

2.5 Functional Requirements

This section describes the functional requirements of the system expressed in the natural style. The requirements are grouped by system features.

2.5.1 Functional Requirement 1 – Ransomware Detection

The system monitors files to detect ransomware activities in real-time. It should analyze encryption patterns, and it will generate alerts when any threat is detected. It notifies the user through popup box. All detected threats should be logged for a record.

2.5.2 Functional Requirement 2 – Ransomware Prevention

The system prevents ransomware attacks after detecting suspicious behavior. When it finds any suspicious activity, it will show a popup to the user to take actions on it. It also saves this action in the log for record or future use.

2.5.3 Functional Requirement 3 – User Alert System

When any threat is detected, our system will notify the user through popup box with some actionable prevention buttons like (delete (yes/no)). It will alert the user when the threat is detected without any delay.

2.5.4 Functional Requirement 4 – Virus Scanning

This system also has an antivirus feature that scan the system for viruses. Users has an option to scan files manually when they want, also they scan automatically for browser and USB files with some actionable prevention buttons like (delete yes/no).

2.5.5 Functional Requirement 5 – Data Backup

The system also has a feature of data backup. By using this feature, users will back up their important data to secure it. It also ensures that users will not lose access to their critical data.

Table 8: Description of FR-1 – Ransomware Detection

Identifier	FR-1
Title	Ransomware Detection
Requirement	The system will detect ransomware by monitoring file encryption patterns. If any threat is detected, it notifies the users about the threat through popup box including some actionable buttons.
Source	User Activity
Rationale	It is essential for preventing data loss and ensuring system security.
Business Rule (if required)	The system should follow basic security guidelines to detect ransomware attacks.
Dependencies	This requirement only works if the system detect threats.

Priority	High
-----------------	------

Table 9: Description of FR-2 - Ransomware Prevention

Identifier	FR-2
Title	Ransomware Prevention
Requirement	The system will allow users to stop threats or processes after detection. It will show a popup to take action against it.
Source	Real-time Detection, User Response
Rationale	It stops ransomware attack before it locks the files, so that less damage and loss happen.
Business Rule (if required)	The system asks the user before stopping anything. It will save the action also.
Dependencies	It works when the system finds any suspicious activity.
Priority	High

Table 10: Description of FR-3 – User Alert System

Identifier	FR-3
Title	User Alert System
Requirement	The system will notify users immediately when the threat is detected. The system will show popup box including threat details and some actionable buttons.
Source	System Logs, Threat Detection Module
Rationale	It helps the users to stop attacks quickly.
Business Rule (if required)	It gives simple and clear alerts that without disturbing the user. It helps the users to take the right steps.
Dependencies	It works with the ransomware detection and alert system.

Priority	High
-----------------	------

Table 11: Description of FR-4 – Virus Scanning

Identifier	FR-4
Title	Virus Scanning
Requirement	The system will scan files to detect viruses. If the system finds something bad, it will show actionable popups to keep the system safe.
Source	Threat Database
Rationale	It helps to keep the system safe from different viruses and malwares.
Business Rule (if required)	N/A
Dependencies	The antivirus needs threat detection to work well.
Priority	High

Table 12: Description of FR-5 – Data Backup

Identifier	FR-5
Title	Data Backup
Requirement	The system allows users to create backup of their important files to prevent their data in case of any attack.
Source	User Input
Rationale	It helps users to create a safe copy of their important data in case of attack.
Business Rule (if required)	The backup system should keep backup files safe, so that they cannot be encrypted by ransomware.
Dependencies	It works by using manual and real time backup.

Priority	Medium
-----------------	--------

2.6 Non-Functional Requirements

This section includes system quality requirements like speed, ease of use, security and reliability. The system detects ransomware attacks quickly, better performance, secure, and it is simple for users.

2.6.1 Reliability

The system should work properly without crashing or stopping in normal use. It must detect ransomware and should not miss serious threats. The system should be able to run for a long time without any errors. It will save logs for future reference.

2.6.2 Usability

The system should be very easy to use. People should understand how to use it without any help. All the messages and buttons in the system must be clear to understand. Even users who don't know much about computers should be able to use it easily.

2.6.3 Performance

The system will detect threats and response to them in some seconds after the threat is detected. This system must be optimized to use less resources of CPU.

2.7 External Interface Requirements

This section explains how the system will work with users and on other devices. It will support ransomware detection, virus scanning, alerts, backup, and logs for security events.

2.7.1 User Interface Requirements

This section explains how the User Interface (UI) looks and works so that it is easy for the users to use it properly.

It should include:

The design should be simple and easy to use.

The layout should be clean.

Same fonts, button names, and different colors should be used in the system.

The screen design will be neat and it will fit properly on all displays.

Buttons like back, go to main menu, will be shown on every screen.

The design should not confuse users, everything will be easy to find.

The system will alert the user, shows popup, when any threat is detected.

2.7.2 Software Interface

The system will be developed using Python, and it uses Tkinter for UI Interface.

The system will provide security features and will not depend on any external tool.

2.7.3 Hardware Interface

The system requires a minimum of 4GB RAM, a quad-core processor, and at least 2GB of

available storage space for optimal performance.

2.7.4 Communications Interface

Internet connection is required for Google Drive upload and authentication.

It will open a browser window for Google login.

It uses Google Drive API for secure file transfer.

2.8 Summary

The requirements for this project were identified using Use Case Analysis. This helped us understand how users will interact with the Anti-Ransomware Security Solution. It ensured that the system can find and block threats, send security alerts, and handle backups.

Chapter 3

Design and Architecture

3. System Design

The Anti Ransomware Security Solution for Microsoft Windows works within any computer-based environment. It will detect and remove threats. It will also protect important data. The system also has antivirus feature to improve security.

3.1 Design Considerations

Assumptions and Dependencies: The system will store real-time logs. It works independently. It has detection and prevention system that does not require access to other networks.

Limitations: It may work slow and it will need more time for detection when the system is slow. This tool might show a false warning if a file is locked or protected by Windows or any other encryption service.

Risks: There are new ransomware types that will not detect by it.

3.2 Design Models

In this project, we used the class diagram to show how different parts of the Anti-Ransomware Solution, like file scanning, threat detection, and file uploading, are connected and work together.

3.2.1 Class Diagram

The class diagram shows the main parts of our app, like scanning files, detecting threats, and uploading to Google Drive. It helps us understand how everything works together. It is placed on the next page for better visibility.

Class Diagram 1

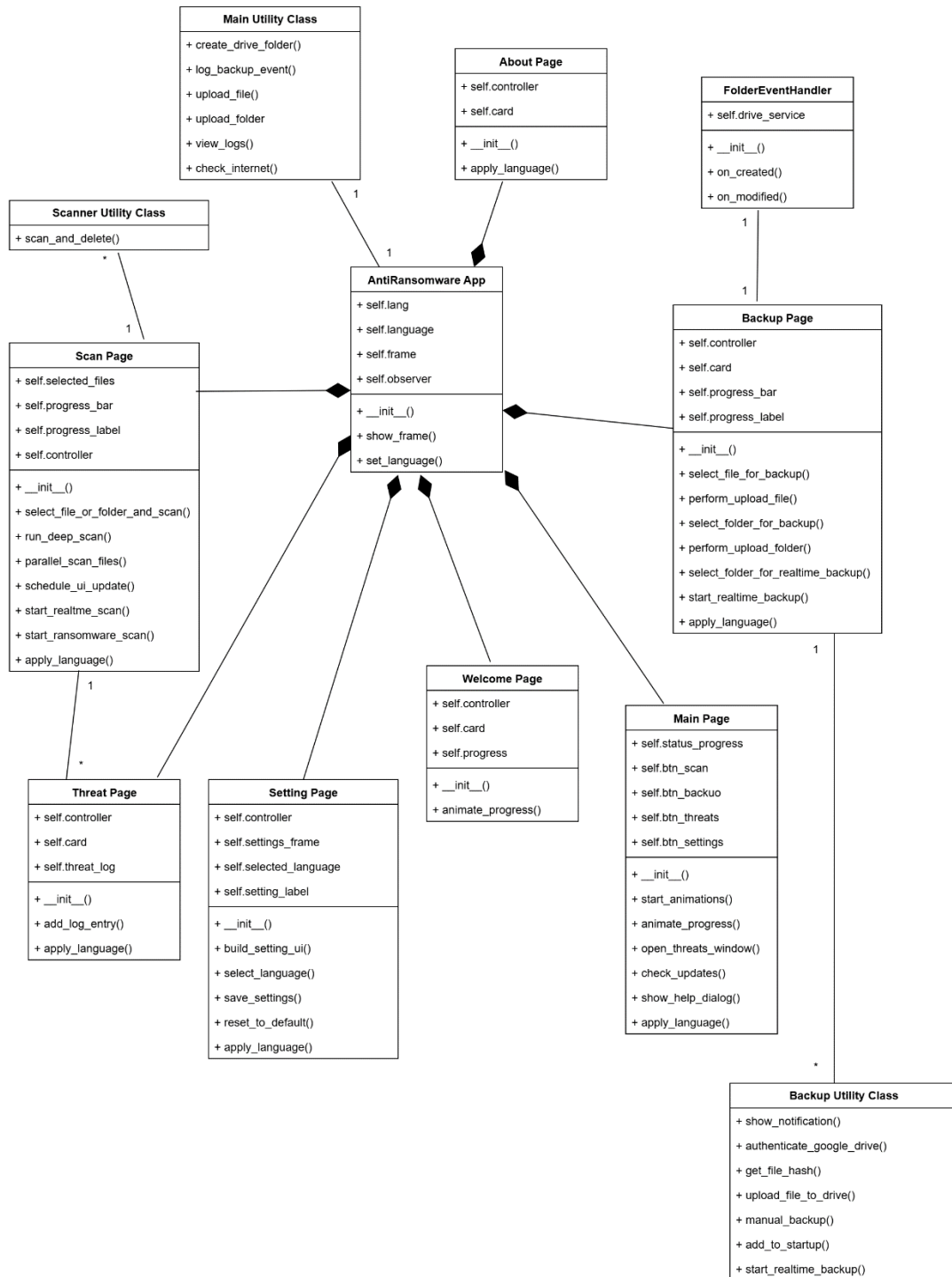


Figure 3: Class Diagram 1

Class Diagram 2

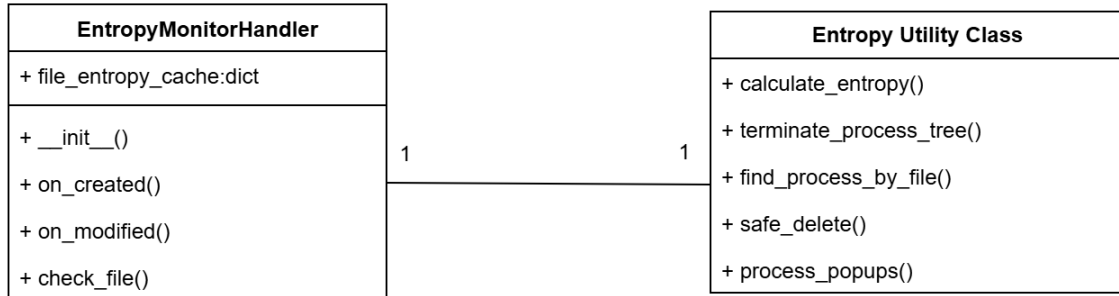


Figure 4: Class Diagram 2

Class Diagram 3

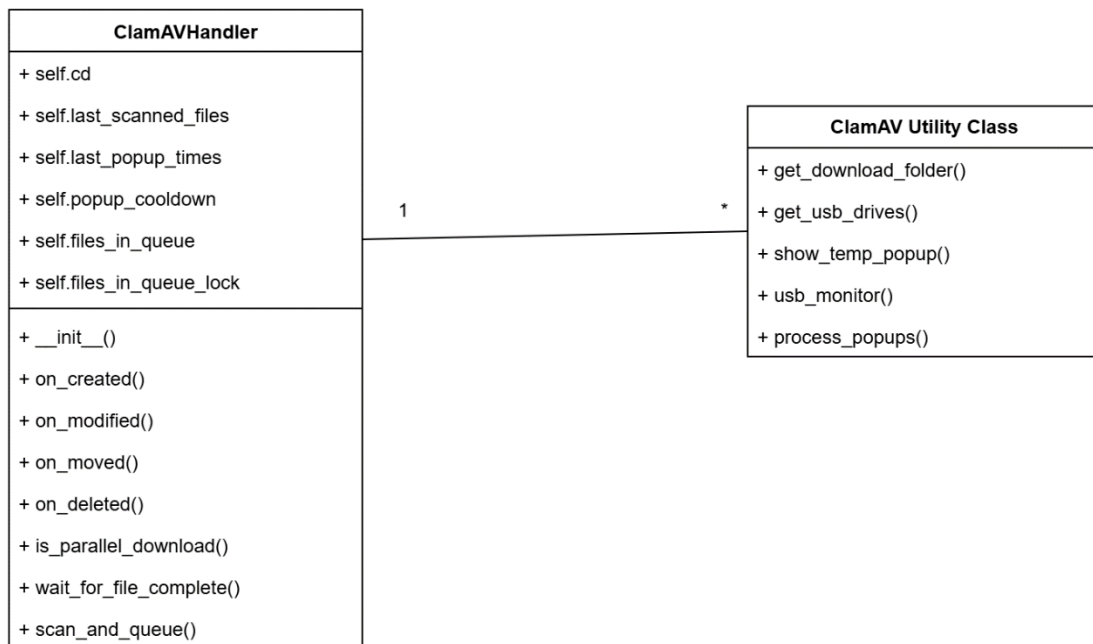


Figure 5: Class Diagram 3

3.3 Data Design

In our project, the system saves all the information about ransomware. This helps it to find dangerous activity and stop them. When a threat is found, it will notify the user and save it in logs that what kind of threat it was. The system also shows notifications or popup box when threat is detected. It also keeps a record of virus scans. There is an option of backing up important files.

3.3.1 Data Dictionary

A Data Dictionary is a list that explains all the important data used in the system. It tells what each type of data is, what it does, and how it is connected to other data. In our Anti-Ransomware Security System, the Data Dictionary helps to organize things like threat records, backup details, user alerts, and file recovery information. This makes the system more organized, and helps it work better in finding and stopping ransomware.

3.3.2 List of Attributes in Ransomware Detection Dataset with Type and Description

Here is the list of common attributes with data types and characteristics that will help to understand ransomware detection data, threat logs, encryption, and user alerts.

Table 13: Dataset Attributes

Data Attributes	Data Type	Description
SCOPES	list[str]	It is a list of permissions that are needed to access Google Drive.
root	tk.Tk	Root is the main app window.
progress_bar	ttkb.Progressbar	It shows the progress when files are uploading.
progress_label	ttkb.Label	It shows the upload progress in percentage.
threats_text	tk.Text	It displays scan results and threats found.
frames	dict	It stores different pages/screen of the app.
logo_img	ImageTk.PhotoImage	It is a logo image shown on the app screen.
observer	Observer	It watches for real-time changes in files and folders.
gui_callback	Callable	It is a function that runs when suspicious files are found.
service	Resource	It is used to interact with Google Drive for uploads.

3.3.3 List of System Entities and Data with Types and Descriptions

Our system has different parts that work together to find, and stop ransomware attacks. Each part takes care of own data so that everything can work properly.

Table 14: System Entities and Data

Data Attribute	Data Type	Description
File Hashes	list[str]	It is a list of unique codes used to identify scanned files.
Suspicious Files	list[str]	These are files that look unsafe or potentially harmful.
Ransomware Signatures	list[str]	These are known patterns used to spot ransomware.
Scan Logs	list[str]	These are records of past scans.
Detected Threats	list[str]	These are all threats found during scans.
Upload Queue	list[str]	These are files that are waiting to be uploaded.
Upload Progress	int	It shows how much of the upload is done (in percent).
Logo Image	image	It is the logo shown on the app screen.
File System Events	list[str]	It tracks file changes (like creation or deletion) in real-time.
Callback Function	function	It automatically runs when a threat is found.
Google Drive Service	service	It is used to upload files and perform tasks on Google Drive.

3.3.4 Functions and Functions Parameters

Functions are tasks that take input and give results. Parameters are values used inside the functions. Hyperparameters are settings that control how the function works.

Table 15: System Functions and Function Parameters

Functions	Function Parameters	Return Type	Description
upload_file()	file_path: str, drive_folder_id : str	bool	It uploads a file to the Google Drive folder.
track_upload_progress()	total_files: int, uploaded_files: int	None	It updates the progress bar based on file upload status.
scan_for_threats()	file_path: str	list[str]	It scans the file for ransomware.
monitor_directory()	folder_path: str	None	It monitors a folder for changes or new files.
update_log()	message: str	None	It updates the log with a new message.
authenticate_google_drive()	credentials_path: str	Resource	It authenticates the user and returns a Google Drive API resource object.

3.4 User Interface Design

This Anti Ransomware Security Solution has a simple and easy-to-use interface. It helps users to find and stop ransomware and other threats. Everything that is important shown clearly on the screen.

It includes the following:

Scan Button: This button will allow users to scan files.

Threats History: This allows users to view logs of the current records.

Backup your Data: This button has three main options, the first one is Upload Folder, the second one is Upload Files, and the third one is Real-time backup. Users backup their important files or folders when they feel that they are not safe.

3.4.1 Screen Images

This section shows screenshots of our project Anti-Ransomware Security Solution. It includes the dashboard, threat history, file scanning interface, switch language, and backup to help understand how the application works visually.

Dashboard

This is the main dashboard of our solution. It shows the user interface where threat logs, scanning and uploading process can be managed easily.

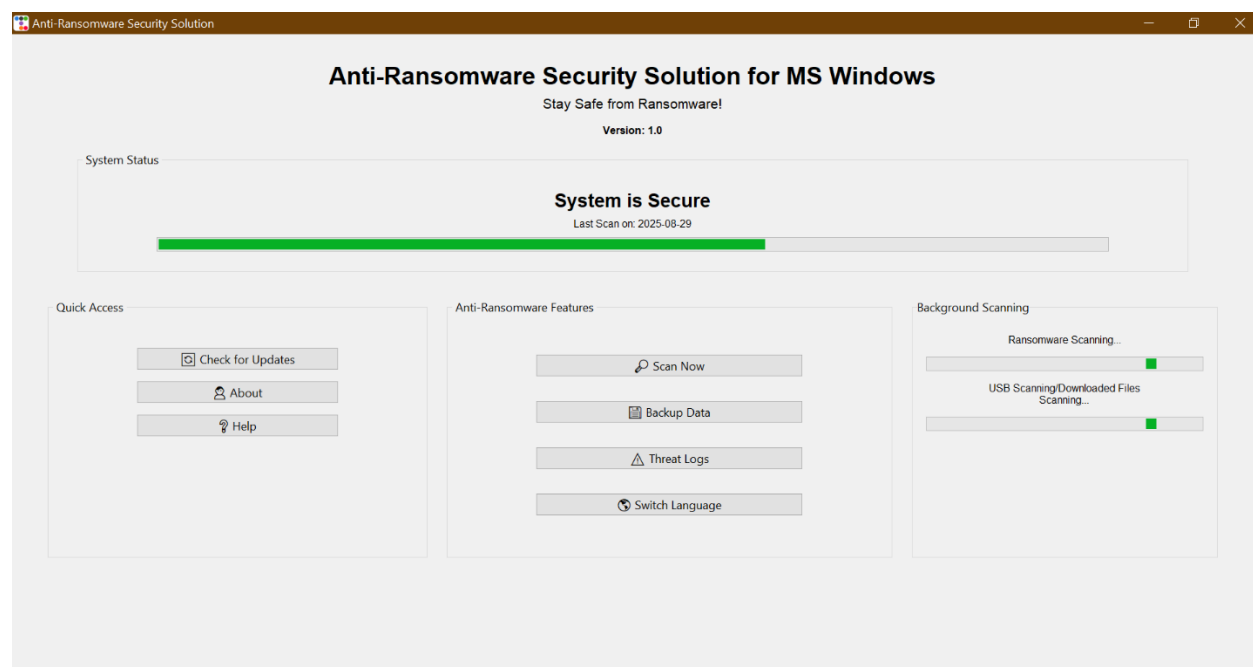


Figure 6: Dashboard

Threats History

This section keeps a record of all threats that found during scans.

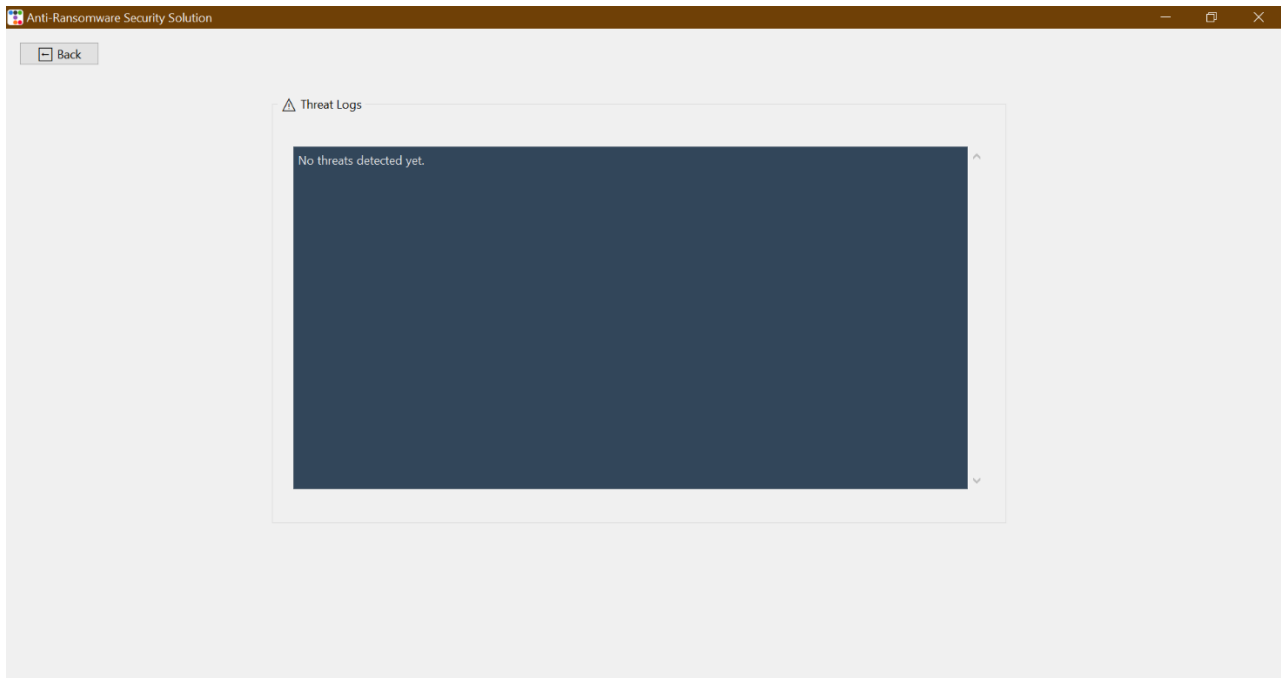


Figure 7: Threat History/Logs

Scan Files

This section helps users to scan files to check for any threat.

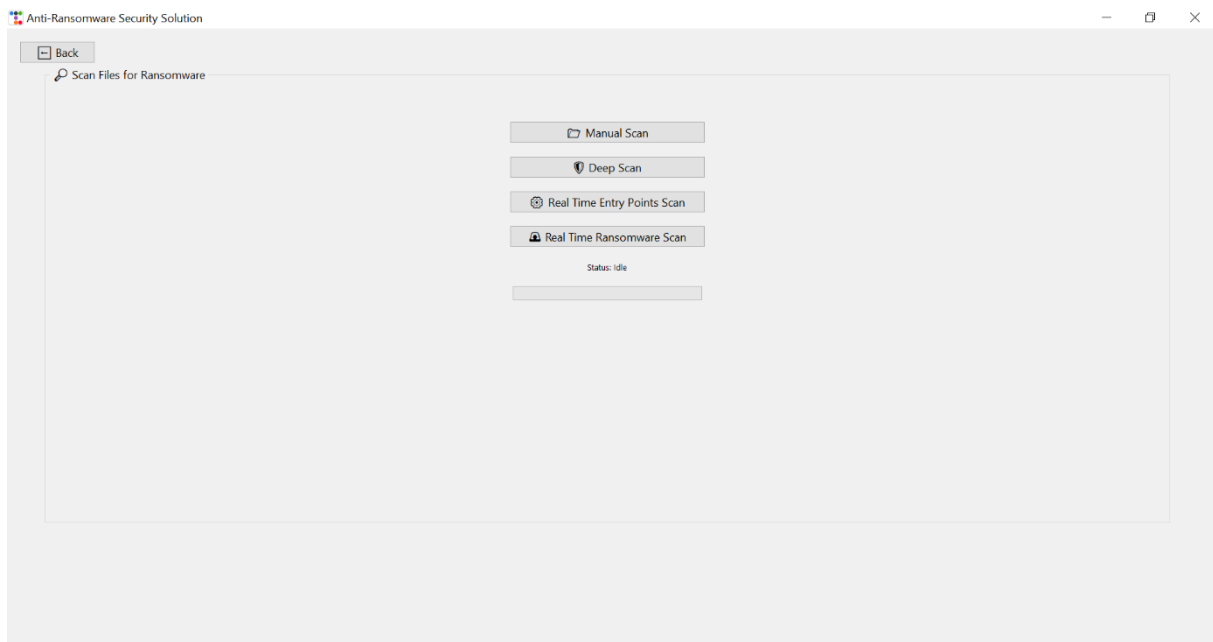


Figure 8: Scan Files

Backup Data

This allows users to backup their important data such as files or folders to keep them safe from

any kind of threat.

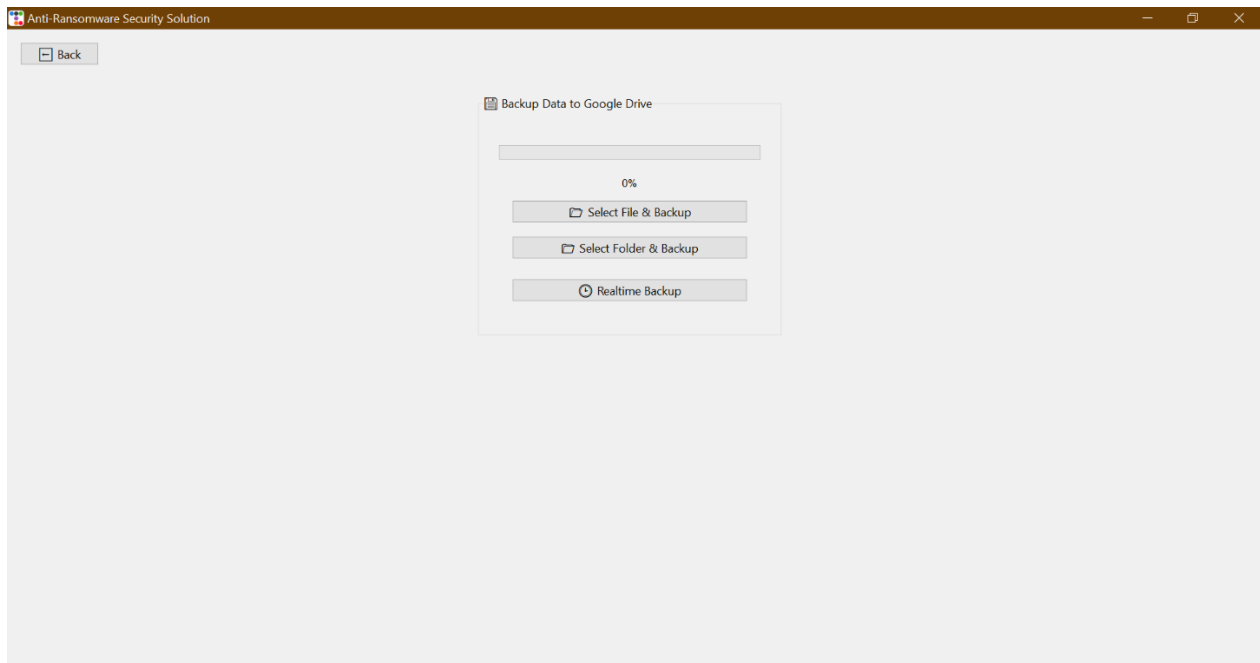


Figure 9: Data Backup

3.5 Behavioral Model

The behavioral model shows that how our system behaves and it reacts to different events, like scanning files, detecting threats, and uploading to Google Drive. It helps us understand the sequence of actions and how the system reacts to user input.

3.5.1 Activity Diagram

The activity diagram shows the sequence of actions or steps that the system performs in different processes, like scanning for threats or uploading files. It helps to understand how tasks are done. It is also placed on the next page for better visibility and space.

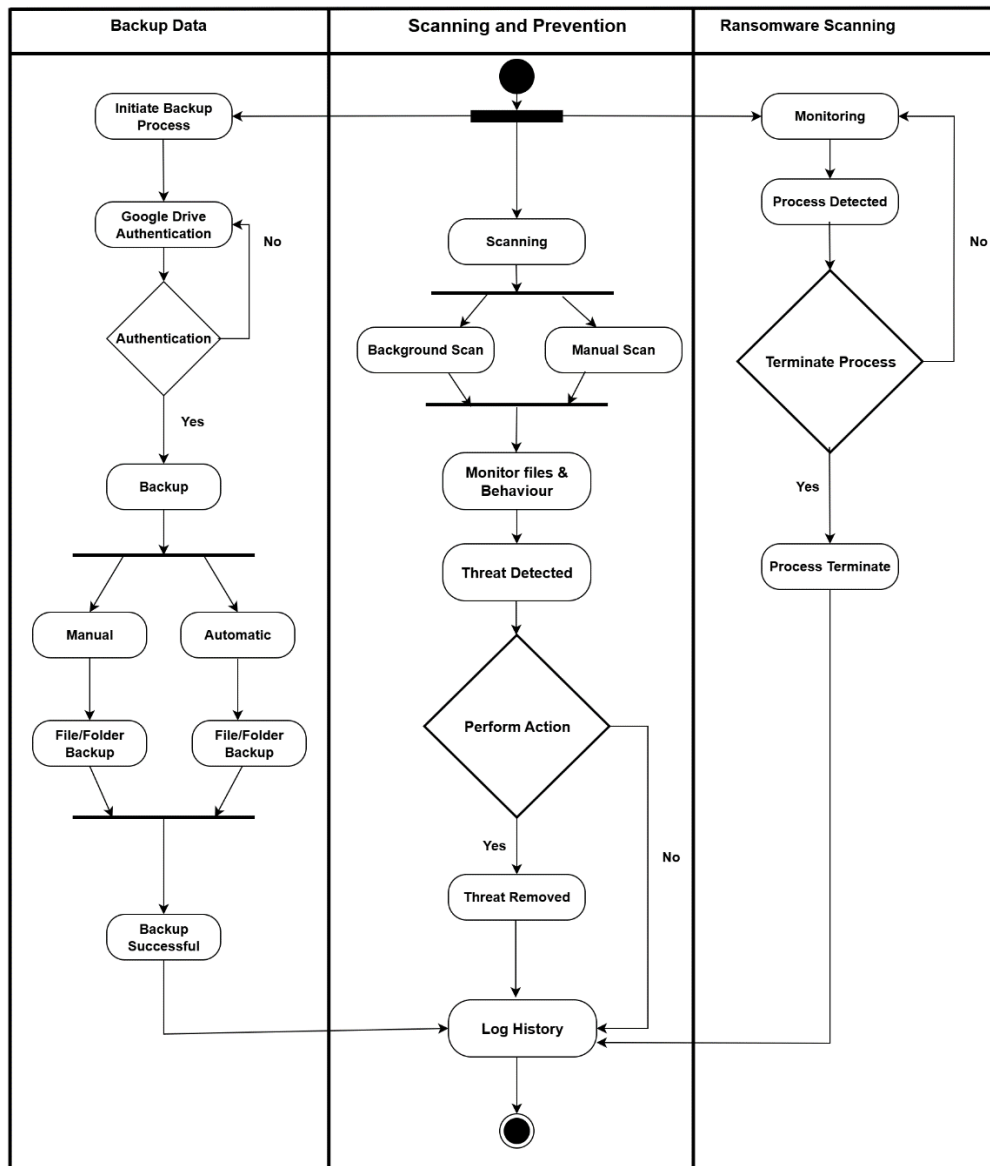


Figure 10: Activity Diagram

3.6 Summary

This chapter explains the design and architecture of the Anti-Ransomware Security Solution for Microsoft Windows. It outlines the diagrams, key decisions, and the structures that are used to create a system, like detecting, preventing ransomware threats. This design confirms the system is user-friendly and it performs well.

Chapter 4

Implementation

4. Implementation

This chapter explains how the system was created using Python. It shows how all the parts like scanning files, uploading to Google Drive, and showing alerts, were built and connected. It also tells about the tools and libraries used to make everything work.

4.1 Algorithm

The algorithm that are used in our project helps to detect ransomware by scanning files, checking their behavior and matching them with known threat patterns. The antivirus functionality is also included to identify harmful viruses and other malwares.

Table 16: Algorithm - Anti-Ransomware

Algorithm 1: Anti-Ransomware Module
<div>Start</div> <div>If application is installed in the system</div> <div> Anti-Ransomware tool is running in background</div> <div> Monitor active processes and file activities continuously</div> <div> For each detected activity:</div> <div> If behavior matches known ransomware patterns then:</div> <div> Temporarily stop the suspicious process</div> <div> Alert the user:</div> <div> "Suspicious behavior detected in process with process id</div> <div> Show options: Terminate Process or Allow to Continue</div> <div> If user selects Terminate then:</div> <div> Kill the process and delete the file</div> <div> Log that action</div> <div> Else:</div> <div> Continue that process</div> <div> Else:</div> <div> Continue monitoring</div> <div>stop</div>

Table 17: Algorithm - Antivirus

Algorithm 2: Antivirus Module
<div>Start</div> <div>If application is installed in the system</div> <div> Antivirus engine (clamd) is running in the background</div>

```

Monitor entry points: Downloads from browser, USB drive files
For each monitored_path in monitored_paths do:
    If new file is added
        scan file using ClamAV
        If result = Infected then:
            Alert user
        Log as threat detected
        Show options: Delete File or Ignore
        If user selects Delete then:
            Delete file
            Log that action as certain file deleted
        Else:
            Log as no threat detected
    Run scheduled scans (every time when new file is added) for USB and browser
Downloads
End While

While application is running do:
    If user clicks "Scan" on UI then:
        User will select file
        scan file using ClamAV
        If it is Infected then:
            Alert user
            Log that action as threat detected
            Show options: Delete File or Ignore
            If user selects Delete then:
                Delete file
                Log that action as certain file deleted
            Else:
                Log as not threat detected
        End while
    If user click on "Deep Scan" on UI then:
        Antivirus scan all folders and the files of home directory
    Alert the user for threats when scan complete
    Show options: delete threats or ignore
    If user selects Delete then:
        Delete files
        Log that action as certain files deleted
    Else:
        Log as no threat detected
    End while
stop

```

Table 18: Algorithm - Backup Module

Algorithm 3: Backup Module
<p>Start</p> <p>While application is running</p> <p> If user select manual backup for file\folder</p> <p> Application will ask for sign in to google account</p> <p> If user sign in and allow the application to access google account</p> <p> File\Folder will upload to google drive</p> <p> Then show message “file\folder uploaded successfully”</p> <p> If user selects “Realtime Backup” (incremental backup) for a critical folder:</p> <p> Then ask user to choose the important folder</p> <p> Check if first-time full backup has been taken</p> <p> If not, upload all files to google backup (Full Backup)</p> <p> Save the backup log</p> <p> Start real-time monitoring of critical folder</p> <p> When a new file is created</p> <p> Upload the file to Google Drive.</p> <p> Log the backup.</p> <p> After each backup:</p> <p> Save the backup log</p> <p> Show message: “files uploaded successfully”</p> <p>End</p>

4.2 External APIs/SDKs

Describe the third-party APIs/SDKs used in the project implementation in the following table. Few examples of APIs are provided in the table.

Table 19: Details of API used in the project

Name of API and Version	Description of API	Purpose of usage	List down the API endpoint/function/class in which it is used
Google Drive API – v3	It is used for file upload to Google Drive.	It uploads backup to Google Drive.	GoogleDriveBackup.upload_file()

google-api-python-client – 2.166.0	It connects Python to Google Services.	It connects Python code to Google APIs.	GoogleDriveBackup class
OAuth 2.0 (google-auth) – 2.38.0	It is used to handle Google login.	It authenticates user for drive access.	GoogleDriveAuth.authenticate()
OAuth 2.0 (google-auth-oauthlib) – 1.2.1	It enables browser-based login.	It asks the user for permission and completes the OAuth login process.	GoogleDriveAuth.authenticate()
ClamAV Engine – 1.4.2	It scans files for threats.	It scans files for malwares/ransomware.	clamav_scan.scan_file().scan_folder()

4.3 Code Repository

We are using Git and GitHub work to manage our project. The parts of the project that have been completed so far, we have uploaded them to the GitHub repository. As more parts are developed, they will also be uploaded step by step. This helps us save each step and keeping everything organized.

Git Repository Link:

<https://github.com/its22647/ARSS> (Private – Code Repository)

<https://github.com/maan43190-ops/Anti-ransomware-security-solution/releases/tag/v1.0.2> (Application - Public Repository)

4.3.1 Metrics of the Git Repository: The following metrics will be monitored to ensure effective use of the Git repository:

Commits: Many commits have been made so far, that shows work is being done.

Branches: There is only one branch right now, so everything is done in the main project area.

Pull Requests: Pull requests have been created. This means changes are added directly without review steps for now.

Issues: There are no open or closed issues.

Contributors: We are 3 members. We have uploaded and working on it.

Code Reviews: Pull requests have been created, and code reviews have been completed.

4.4 Summary

This chapter shows that how we created different parts of the Anti-Ransomware system. It includes how the system works, its features, and where the code is saved. New features will be added and updated step by step.

Chapter 5

Testing and Evaluation

5. Introduction

Testing ensures that the Anti-Ransomware Security Solution works as expected in different situations. This chapter covers unit, functional, integration, and performance testing conducted on all the major features of the system, including ransomware detection, antivirus, backup, threat logs, and admin access control.

5.1 Unit Testing (UT)

Table 20: Unit Testing

Test Case ID	Requirement ID	Title	Description	Objective	Precondition
T1	FR-1	Ransomware Detection	Our application tests file behavior.	It makes sure that the system can correctly detect ransomware.	It makes sure file activity is being monitored.
T2	FR-2	Ransomware Prevention	It prevents the ransomware attacks.	It makes sure that the system can prevent ransomware attacks correctly.	It makes sure that the ransomware is detected.
T3	FR-3	User Alert System	It alerts the users.	It makes sure that the system will notify the users about the threats.	When a threat is detected or not, the user is connected to the system, and the backup feature is running.
T4	FR-4	Virus Scanning	It scan files.	It ensures that threat is being detected.	User will select the files and clamd service is running.
T5	FR-5	Data Backup	It backup user's critical data.	It secures user's data in case of any suspicious activity or attack.	Internet connection is required.

5.2 Functional Testing (FT)

In this phase, the functionality of each module is tested to ensure that the system meets its requirements.

5.2.1 Ransomware Detection

The ransomware module was tested for real-time behavior monitoring and entropy detection. It successfully detected suspicious activities.

5.2.2 Ransomware Prevention

The ransomware prevention was tested by implementing the actionable options like terminating the process to prevent the ransomware attack.

5.2.3 User Alert System

The alert system was tested for notifications and popup during threat detection, and backups. All the alerts were almost accurate according to the condition.

5.2.4 Virus Scanning

The antivirus module was tested for scanning. It detected viruses for manual and automated scanning. The system responded correctly.

5.2.5 Data Backup

The backup module was tested for manual backup and automated backup. All backups were successful.

5.3 Integration Testing (IT)

Integration testing checks that all modules work together properly, ensuring the system functions as a whole.

5.3.1 Ransomware Detection

Our system detected suspicious activities and showed results quickly. All parts worked together smoothly.

5.3.2 Ransomware Prevention

The ransomware prevention was tested by implementing the actionable options like terminating the process to prevent the ransomware attack.

5.3.3 User Alert System

When threats are detected and backup feature is running, alerts appear on time that informs the user immediately.

5.3.4 Virus Scanning

The antivirus scan files, showed alerts, protected through actionable buttons, and saved logs. Everything worked properly.

5.3.5 Data Backup

In our project, the backup features worked correctly. Each backup was saved securely in the cloud (Google Drive).

5.4 Performance Testing (PT)

To evaluate the speed and stability of each feature, performance testing was conducted. Our goal was to measure response time and reliability.

5.4.1 Virus Scanning

Our project scanned files and folders within 15 seconds.

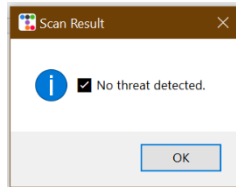


Figure 11: Antivirus Scan

5.4.2 Backup Module

Our tool uploaded to cloud in around about 10-20 seconds.

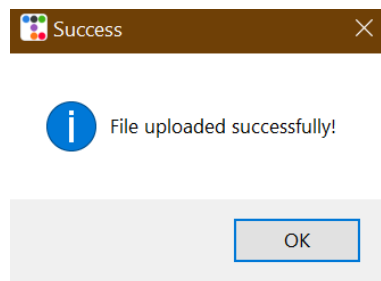


Figure 12: Backup

5.4.3 User Alerts

In our project, the alerts are showing accurately without any specific delay.

5.4.4 Anti-Ransomware Tool

Our tool detects ransomware behavior within the range of 5-10 seconds.

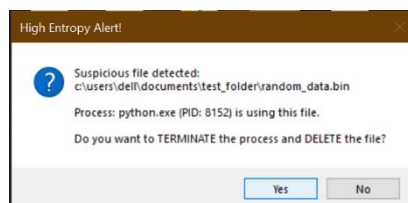


Figure 13: Anti-Ransomware Tool

5.5 Summary

All the features of Anti-Ransomware Security Solution for Microsoft Windows were successfully tested.

Unit tests confirms that each module works correctly.

Functional and Integration Testing shows that all the modules of our project worked together smoothly.

Performance Testing proves that our system is almost fast and reliable.

Overall, the system meets its objectives effectively.

Chapter 6

System Conversion

6. Introduction

This chapter explains how our newly developed system is deployed and made ready for use.

6.1 Conversion Method

We used pilot conversion method.

First, system has tested and deploy on limited machines.

Second, scan behavior, threat handling and other application functionalities has been observed.

6.2 Deployment

Download the installer (.exe) from GitHub.

Install the application on the system.

Launch the application.

Start using the app for scanning, backup, and threat protection.

6.2.1 Data Conversion

The project did not replace an existing solution or application. Scan and backup will be set (as default).

6.2.2 Training

A user guide will be developed:

End user help: How to use application and steps for how to make the system or data protected from malwares.

6.3 Post Deployment Testing

Our application monitors background ransomware detection.

It checks and monitors the viruses through antivirus.

It uploads files and folders manually and automatically.

We verify the functionalities and features from users.

6.4 Summary

The application has been deployed using pilot-first approach then testing in real conditions, adjusting for the performance, security and usability.

Chapter 7

Conclusion

7. Introduction

This chapter concludes the project, evaluate the project objectives & goals and highlights future work.

7.1 Evaluation

Make a list of the objectives specified in Chapter 1 and trace whether they have been implemented in your developed FYDP.

Table 21: Evaluation Matrix

Objectives	Status
Detect and prevent ransomware in real time	Completed
Integrate antivirus to scan files and USBs	Completed
Provide file/folder backup feature	Completed
Maintain log/history	Completed
Offer user-friendly interface	Completed
Educate users via help section	Completed

7.2 Traceability Matrix

This table has main features of our project. This includes its design, code, and test. It helps us that all important parts like detection, prevention, backup, and alerts are fully developed and tested properly.

Table 22: Traceability Matrix

Requirement ID	Requirement Description	Design Specification	Code	Test ID
FR-1	Ransomware Detection	Anti-Ransomware Tool	entropy.py	TI, FT1
FR-2	Ransomware Prevention	Anti-Ransomware Tool	entropy.py	T2, FT2
FR-4	Virus Scanning	ClamAV Integration	scanner_utils.py, real_time_clamscan.py, main.py	T4, FT4
FR-5	Data Backup	Google Drive Backup	backup.py, main.py	T5, FT5

7.3 Conclusion

The Anti-Ransomware Security Solution for Microsoft Windows successfully met all its objectives. It is a desktop application that offers real-time ransomware detection and prevention, antivirus integration for manually and automated scanning, and backup (automated and manual). Our tool is user-friendly and develop to help against different viruses and ransomware attacks.

7.4 Future Work

It can be integrated with AI for better ransomware protection.

It can be introduced in multi-language UI.

It can be extended in other environments like Linux.

References

- [1] W. R. D. L. B. E. K. A. Kharraz, "Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks," *Detection of Intrusions and Malware, and Vulnerability Assessment*, 2015.
- [2] M. A. M. S. Z. M. S. B. A. S. Al-Rimy, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Computers & Security*, 2018.
- [3] L. M.-G. R. M. E. C. L. D. Sgandurra, "Automated Analysis of Ransomware: Benefits, Limitations and Use for Detection," *IEEE Transactions on Information Forensics and Security*, 2016.

Appendix A

Use case Description Template

Appendix-A Use Case Description (Fully Dressed Format)

The Table A-1 below indicates a comprehensive use case template.

Table 23: Detailed Use Case Template – Ransomware Scanning

Use Case ID	UC-1
Use Case Name	Ransomware Scanning
Actors	User
Description	It will monitor file behavior. It will make sure that the system can detect and actionable prevention against ransomware successfully.
Trigger	Real-time detection of suspicious encryption.
Preconditions	The system is running and monitoring file behavior and active processes.
Postconditions	The system will detect ransomware, notifies the user and shows actions.
Normal Flow	<ol style="list-style-type: none">1. System starts real-time monitoring.2. It monitors file behaviors and active processes.3. When any threat is detected, it will show popup and actions to the users.4. All the actions and threats will be logged.
Alternative Flows	If no threat is detected, it will continue monitoring.
Business Rules	The system should follow basic security guidelines to detect ransomware attacks.
Assumptions	The system has permission to monitor files and processes in real-time.

Appendix-A Use Case Description (Fully Dressed Format)

The Table A-2 below indicates a comprehensive use case template.

Table 24: Detailed Use Case Template – Virus Scanning

Use Case ID	UC-2
Use Case Name	Virus Scanning
Actors	User
Description	It detects the virus attacks and provide actionable prevention.
Trigger	Files or folders should be selected and automated service should be running.
Preconditions	Clamd service should run in the background.
Postconditions	The system will report whether the file is safe or infected.
Normal Flow	<ol style="list-style-type: none">1. System detects viruses.2. User will notify through a popup.3. User choose actions.4. All the actions are logged in a record.
Alternative Flows	No alternate flows.
Business Rules	The system shall use ClamAV to scan files for viruses.
Assumptions	The system remains safe and stable during this operation.

Appendix-B

Coding Standards

Appendix-B General Coding Standards & Guidelines

These are the main coding rules and practices followed while developing the Anti-Ransomware Security Solution.

1. Naming Style

We used camelCase and snake_case in code.

2. Clear Variable Names

We used names like SCOPES that show their purpose clearly.
We avoided random or short names.

3. Function Names

Functions named by what they do.
Examples: check_files(), upload_files().

4. Comments

Important logic was explained using short comments.

Example:

```
# Upload file to Google Drive
def upload_file(file_path):
```

5. Indentation

4-space indentation used properly in all code blocks.

6. Separate Code Files

Code was split into files, like:

backup.py, for backups
detection.py, for scanning
main.py, for interface and functionality

7. One Task Per Function

Each function/class was written for one specific job only.

8. Reuse Code

Repeated logic (e.g., logging) was written once and reused.

9. Error Handling

Try/except blocks used in risky places (e.g., upload, scanning).

10. Logging Actions

Used Python's logging to save:

Threats found
Actions taken
Errors

Appendix C

Prototype

Appendix-C Application Prototype

The following prototype is part of the implemented system.

Dashboard

In the dashboard, central panel displaying scan, threat history, backup, help button and more options.

Threat History

This shows logs of current actions.

Scan files

This is file selection screen for scan.

Backup

In backup, we have three options like backup files, backup folders and real-time backup.

Switch Language

Users can switch language from English to Urdu.

Quick Access

In this, we have buttons of about and help.

These interfaces developed using Tkinter and Ttkbootstrap for modern UI experience. They were designed to be simple, usable by both technical and non-technical users.