xVersion: 1.0

Note: This document is intended for internal academic presentation purposes.

Software Requirements Specification (SRS)

*Topic : “Secure storage and controlled access of files within a directory”*

**Developed by: (Team – 7) 2nd Semester ‘P’ Section**

1. **Yadunandana Reddy M**
2. **Yashas Sadananda**
3. **Vidushi Singh**
4. **Yashwanth Aditya**

**Date: [23-05-2025]**

# 1. Introduction

## 1.1 Purpose

The purpose of this project is to simulate a secure digital vault using puzzles and delays, while offering no real protection or encryption. It provides the illusion of security, serving as a novelty or satirical tool also in Design and Implementation Constraints, add the fact that the authentication stuff resides in that config file

This system:

* Requires **password-based authentication** to access files.
* Implements **security questions** for identity verification.
* Uses a **puzzle challenge** as an additional security measure.
* Maintains and lists **stored files** for access.

This system does not support:

* File encryption
* Multi-user functionality
* Graphical User Interface (GUI)

## 1.3 Definitions, Acronyms, Abbreviations

| **Term** | **Definition** |
| --- | --- |
| **SRS** | Software Requirements Specification |
| **Vault** | A protected storage system for files |
| **CLI** | Command Line Interface |
| **Puzzle Authentication** | A challenge-based authentication mechanism |
|  |  |

# 2. Overall Description

## 2.1 Product Perspective

The Secure Vault is a **standalone CLI-based application** with no external dependencies like databases or network connectivity.

## 2.2 User Classes and Characteristics

|  |  |
| --- | --- |
| User Type | Description |
| General User | Basic knowledge of file systems and command-line interface |

## 2.3 Operating Environment

**-Operating System:** Windows 10 or higher

**-Language:** C

**-Compiler:** GCC via MinGW / Code::Blocks

## 2.4 Design and Implementation Constraints

-Works only on **Windows OS**

-Authentication includes **passwords, security questions, and puzzles**

-Files must reside in the **vault\_files directory**

## 2.5 Assumptions and Dependencies

- Users should not rename or move files manually within the vault directory.  
- The system assumes single-user access.

# 3. Specific Requirements

## 3.1 Functional Requirements

* **FR1:** The system shall **prompt the user for authentication** before access.
* **FR2:** The system shall store files in a **secure vault directory**.
* **FR3:** Users shall be required to **solve a puzzle** before accessing the vault.
* **FR4:** Users must **answer security questions** to verify identity.
* **FR5:** The system shall **list available files** in the vault.
* **FR6:** Users can **open and view file contents**.
* **FR7:** The system shall maintain **configuration settings** in a file.

## 3.2 Non-Functional Requirements

* **NFR1:** The system shall validate **user input** at every step.
* **NFR2:** The program shall respond to user commands **within 1 second**.
* **NFR3:** The system shall be lightweight and run on **systems with 4GB RAM or less**.
* **NFR4:** Passwords and sensitive input **shall not be displayed while typing** (future enhancement).

## 3.3 External Interface Requirements

- User Interface: CLI-based menu system.  
- Hardware Interface: Standard keyboard and monitor.  
- Software Interface: Windows OS, file system.

# 4. System Features

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Multi-step Authentication** | Requires passwords, puzzles, and security questions to access vault and ability to reset authentication file |
| **File Listing** | Displays available files for selection |
| **File Viewing** | Allows users to open and read file contents |
| **Configuration Management** | Stores user credentials securely |

# 5. Appendices

## Appendix A: Future Enhancements

* **Secure hashed password storage**
* **Encryption of stored files**
* **Graphical User Interface (GUI)** for usability improvements

## Appendix B: Sample Output

Welcome to the Secure Vault!

1. Unlock Vault

2. Reset Vault

3. Exit

# Team Contributions

Member 1 - Name: Yadunandana Reddy M SRN : PES2UG24CS605

**Contribution:**

Took primary responsibility for structuring and integrating the complete codebase. Played a key role in coordinating individual modules, particularly contributing to the list\_vault\_files\_and\_select and unlock functions, and ensured the final solution was cohesive and fully functional.

Member 2 - Name: Yashas Sadananda SRN : PES2UG24CS610

Contribution:

Focused on implementing robust file handling mechanisms within the vault system. Contributed to developing secure methods for file listing, access, and content display, ensuring stability and accuracy in vault operations.

Member 3 - Name: Vidushi Singh SRN : PES2UG24CS581

Contribution:

Developed the command-line interface and implemented the security authentication flow including password verification, puzzle challenge, and security questions.

Member 4 - Name: Yashwanth Aditya SRN : PES2UG24AM190

Contribution:

Handled the password setup, reset functionality, and managing configuration file interactions for storing and retrieving vault credentials.