Project High-Level Solution

Introduction

In this project, I plan to develop a Windows-based desktop file management application that addresses the critical need for robust personal data security. As desktop computers are integral to individual productivity, the vulnerability of files stored on these systems has become a significant concern. Escalating threats of data breaches and cyber-attacks, including ransomware, malware, and viruses threaten individuals' online safety. As McIntosh et al (2021) acknowledged, traditional file systems do not adequately protect against these threats, including those stemming from software defects. The GuardianVault application is intended to provide a comprehensive solution that not only secures data but also ensures easy accessibility and management for users.

The primary objective of this project is to develop an application that integrates seamlessly with Windows 10 and Windows 11 environments, built on the .NET Framework. The application will feature multi-threaded operations for faster file processing and encryption tasks. The application will utilize Advanced Encryption Standard (AES) encryption and one-way hashing to protect user data and offer user-friendly interfaces that support basic accessibility standards. Users will be able to customize security settings, including encryption levels and access controls, according to their specific needs.

The successful implementation of the GuardianVault application will enhance personal data security for Windows desktop users while setting a new standard for secure file management. By integrating cutting-edge encryption technologies and user-friendly access control capabilities, the application will provide a reliable protection for personal data. I believe the application addresses a significant gap in current secure file management on Windows 11 and Windows 10 platforms. The successful completion of this project will elevating the overall security posture of Windows desktop users.

Solution Description

The GuardianVault app will be a Windows-based application developed using Windows Forms technology and the C# programming language on the .NET platform. The application will run as a standalone solution with no external dependencies and adhere to the principles of Single Page Application (SPA) design. The application use the Advanced Encryption Standard (AES) encryption and one-way hashing components built into the .NET Framework for data protection. The Model-View-Controller (MVC) design pattern will be used to separate the application logic, presentation, and data. The presentation layer of the application will leverage Windows Forms, User Controls, and common Windows Form controls. The MVC pattern will be used to organize the application into distinct components. Models will be used to manage data and business logic; Views will be used to display data and the user interfaces, and Controllers will be used to handle user input and update Models. This separation will increase reusability, testability, and maintainability of the solution.

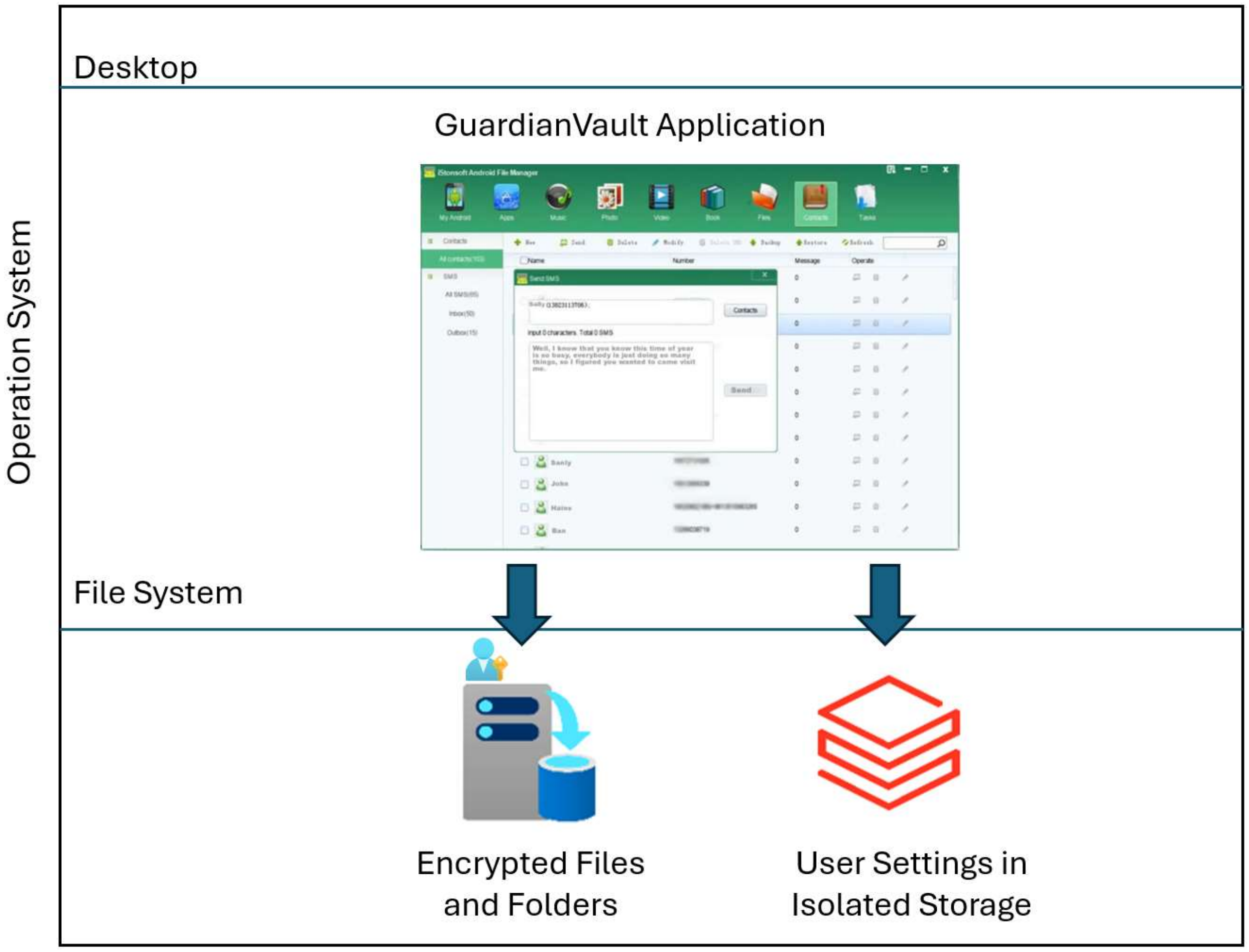
Input will involve user actions such as adding, modifying, and deleting files and folders to be encrypted. The application will process this data by encrypting and decrypting files and folders using AES. The application will use the isolated storage through the *IsolatedStorageFile* class built into the .NET Framework to store and manage specific configuration settings. For desktop applications, isolated storage serves as a data storage method that enhances security and isolation by establishing standard procedures for linking code with stored data (Microsoft, n.d.). The application will maintain an organized list of encrypted folders and files. Output will be the display and management of these encrypted files and folders. The application will be developed using Visual Studio 2022 on .NET 4.8.

System Architecture

Figure 1 depicts the high-level system architecture of the GuardianVault application. The illustration showcases the integration and the flow of data between the GuardianVault application and the file system. User interactions with the application, for example, adding or modifying files, trigger encryption processes and securely store encrypted data on the file system.

Figure 1

High-Level Solution Architecture


Desktop
Operation System
GuardianVault Application
File System
Encrypted Files and Folders
User Settings in Isolated Storage

At the highest level, the GuardianVault application operates on the desktop layer of the user's operating system. It is a standalone Windows-based application designed using Windows

Forms technology. The core of the solution is the GuardianVault application itself, which provides a graphical user interface (GUI) that allows users to interact with their files and folders. The interface displays a list of encrypted files and folders, and users can perform operations such as adding, modifying, and deleting these files and folders. The application follows the Single Page Application (SPA) design.

Beneath the desktop layer, the file system layer handles the actual storage and management of encrypted files and folders. The GuardianVault application ensures that all user data is encrypted using the Advanced Encryption Standard (AES) before being saved to the file system. This encrypted data is represented in the diagram by the icon showing a server and a database with a lock and key, indicating secure storage.

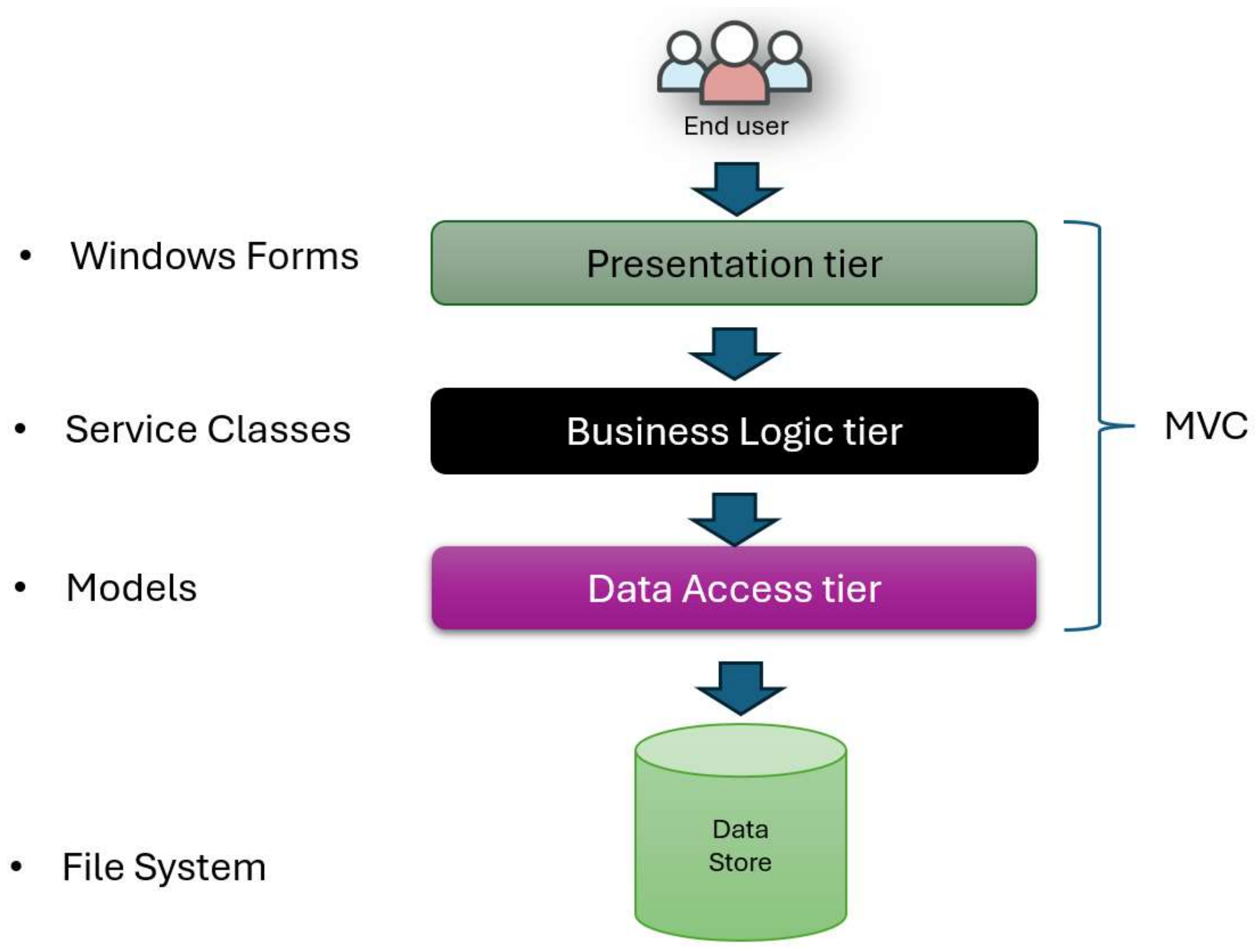
To further enhance security and user customization, the application stores user settings in isolated storage. This ensures that settings such as encryption levels and access controls are securely saved and easily retrievable. The isolated storage mechanism separates user-specific settings from the general application data, reducing the risk of unauthorized access or data corruption.

Application Architecture

Figure 2 shows the layered architecture design of the GuardianVault application and illustrates the separation of concerns and the flow of data between different components of the application. The architecture follows the MVC pattern, which is a well-established design pattern for developing scalable and maintainable applications (FireUp.pro, n.d.).

Figure 2

Application Architecture



The end user interacts with the application through the presentation tier by performing actions such as adding, modifying, encrypting, opening, or deleting files and folders. The presentation tier captures user inputs and forwards them to the business logic tier. For example, if a user adds a file, the presentation layer sends this request to the business logic layer. The business logic tier processes the user request, applying necessary rules and preparing the data for storage or further processing.

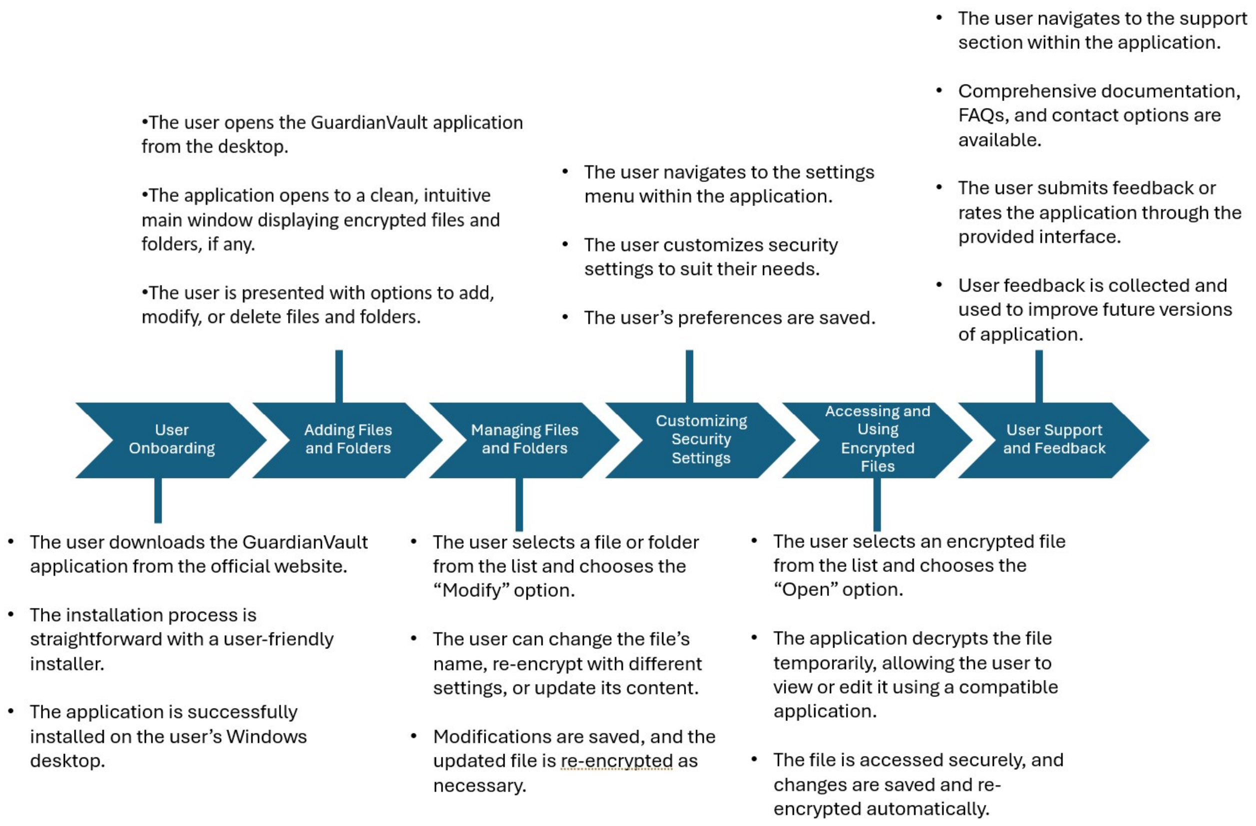
The business logic tier interacts with the data access tier to retrieve, update, or store data. For example, it sends the encrypted file data to be stored in the data store. The data access tier performs the actual data operations on the file system, ensuring that files and folders are securely saved. Once the data operations are complete, the result is sent back through the business logic tier to the presentation tier, which then updates the user interface accordingly.

User Journey

Figure 3 illustrates the user journey for the GuardianVault application, detailing each step from user onboarding, adding, and managing files and folders, customizing security settings, accessing encrypted files, to receiving user support and providing feedback.

Figure 3

User Journey



User Onboarding
Adding Files and Folders
Managing Files and Folders
Customizing Security Settings
Accessing and Using Encrypted Files
User Support and Feedback

User Onboarding:
- The user downloads the GuardianVault application from the official website.
- The installation process is straightforward with a user-friendly installer.
- The application is successfully installed on the user's Windows desktop.

Adding Files and Folders:
- The user opens the GuardianVault application from the desktop.
- The application opens to a clean, intuitive main window displaying encrypted files and folders, if any.
- The user is presented with options to add, modify, or delete files and folders.

Managing Files and Folders:
- The user selects a file or folder from the list and chooses the "Modify" option.
- The user can change the file's name, re-encrypt with different settings, or update its content.
- Modifications are saved, and the updated file is re-encrypted as necessary.

Customizing Security Settings:
- The user navigates to the settings menu within the application.
- The user customizes security settings to suit their needs.
- The user's preferences are saved.

Accessing and Using Encrypted Files:
- The user selects an encrypted file from the list and chooses the "Open" option.
- The application decrypts the file temporarily, allowing the user to view or edit it using a compatible application.
- The file is accessed securely, and changes are saved and re-encrypted automatically.

User Support and Feedback:
- The user navigates to the support section within the application.
- Comprehensive documentation, FAQs, and contact options are available.
- The user submits feedback or rates the application through the provided interface.
- User feedback is collected and used to improve future versions of application.

Wireframes

This section showcases initial design mockups for the application's user interface, providing a glimpse into the early stages of our design process. Figure 4 shows the window where users enter their credentials to securely authenticate and access the application.

Figure 4

Sign-in Window

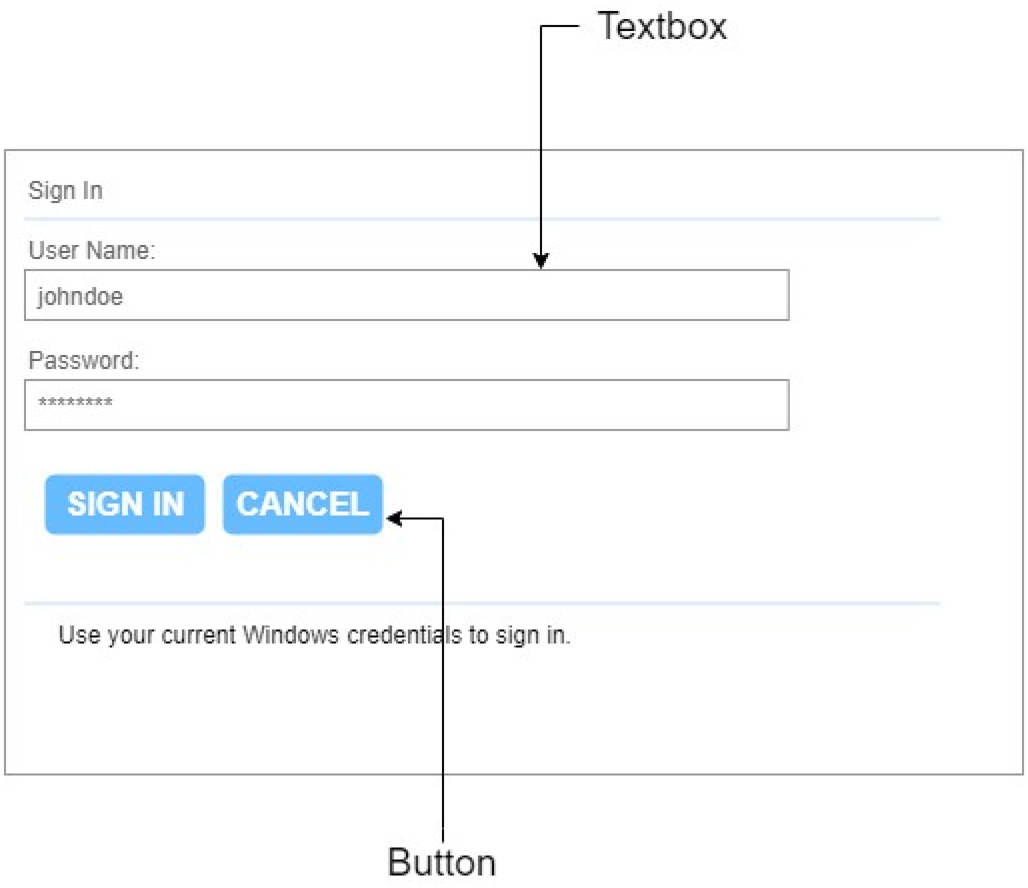
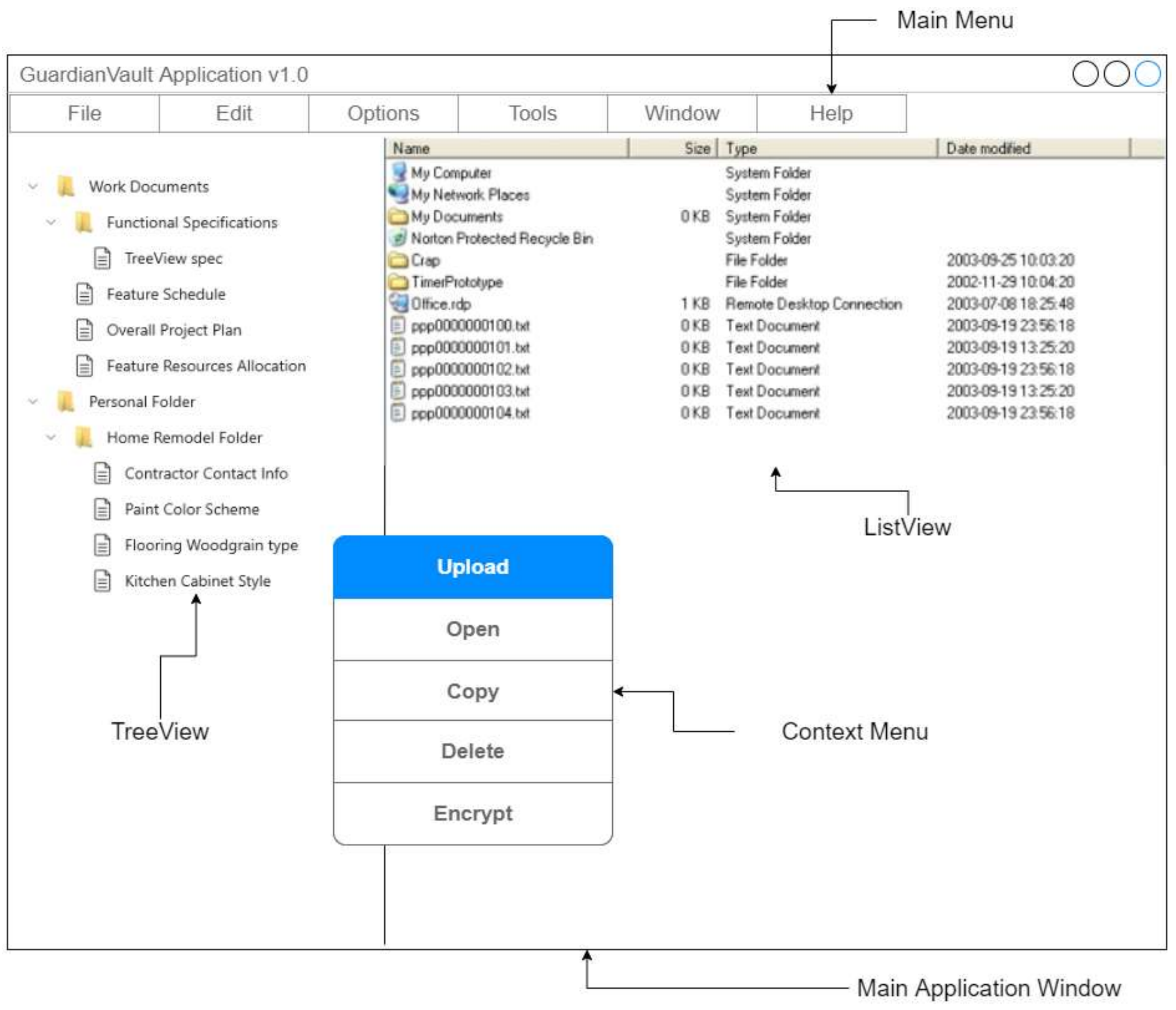


Figure 5 shows the main application window and the UI components of the GuardianVault application.

Figure 5

Main Application Windows

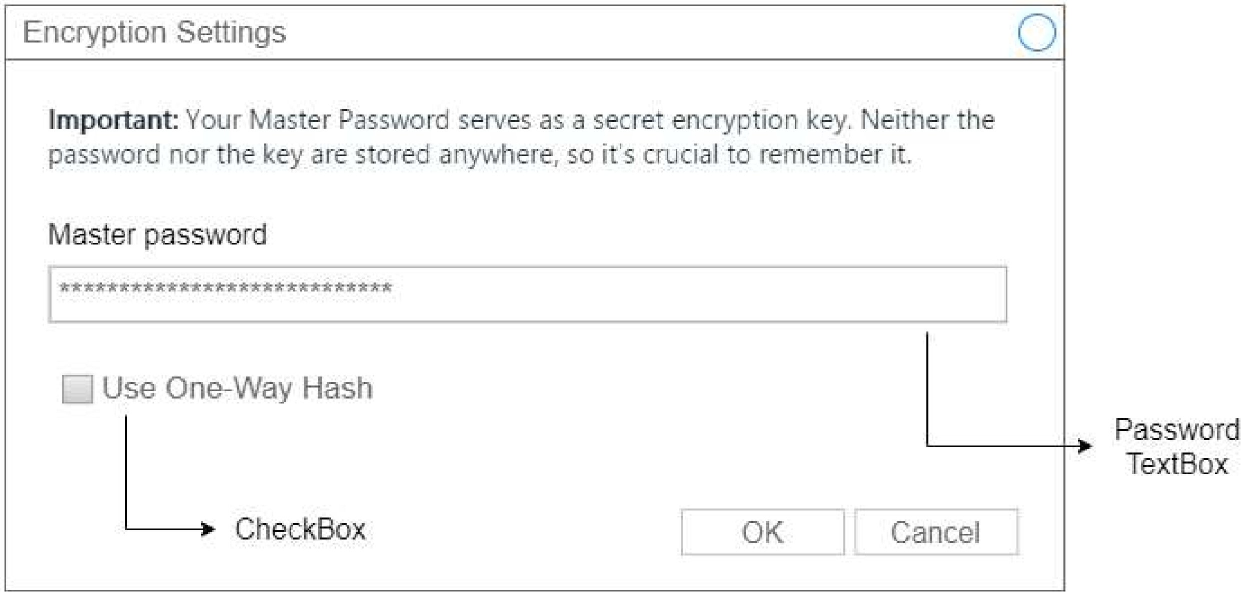


Name	Size	Type	Date modified
My Computer		System Folder	
My Network Places		System Folder	
My Documents	0 KB	System Folder	
Norton Protected Recycle Bin		System Folder	
Crap		File Folder	2003-09-25 10:03:20
TimerPrototype		File Folder	2002-11-29 10:04:20
Office.rdp	1 KB	Remote Desktop Connection	2003-07-08 18:25:48
ppp0000000100.txt	0 KB	Text Document	2003-09-19 23:56:18
ppp0000000101.txt	0 KB	Text Document	2003-09-19 13:25:20
ppp0000000102.txt	0 KB	Text Document	2003-09-19 23:56:18
ppp0000000103.txt	0 KB	Text Document	2003-09-19 13:25:20
ppp0000000104.txt	0 KB	Text Document	2003-09-19 23:56:18

Figure 6 displays the Encryption Settings window, where users can configure their master password and utilize one-way hashing to enhance data security.

Figure 6

Encryption Settings Window



Encryption Settings

Important: Your Master Password serves as a secret encryption key. Neither the password nor the key are stored anywhere, so it's crucial to remember it.

Master password

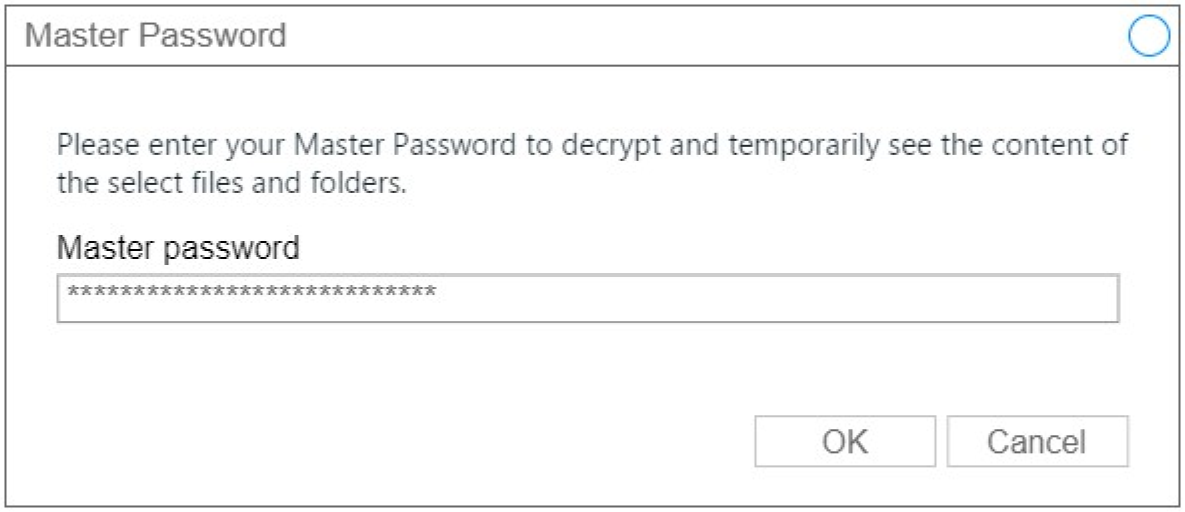
****

Use One-Way Hash

Figure 7 shows the Master Password Confirmation window, where users verify their master password before decrypting, downloading, or opening files and folders.

Figure 7

Master Password Confirmation Window



Master Password

Please enter your Master Password to decrypt and temporarily see the content of the select files and folders.

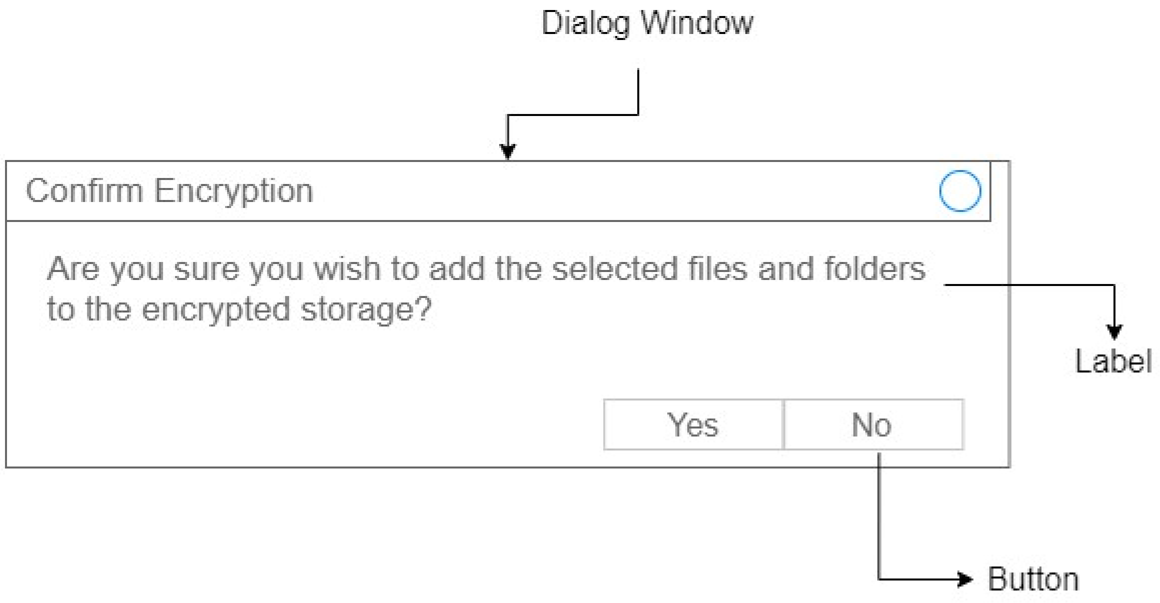
Master password

****

Figure 8 displays the window where users confirm their intention to add files and folders to the encryption storage.

Figure 8

Add Files and Folders Confirmation Window.

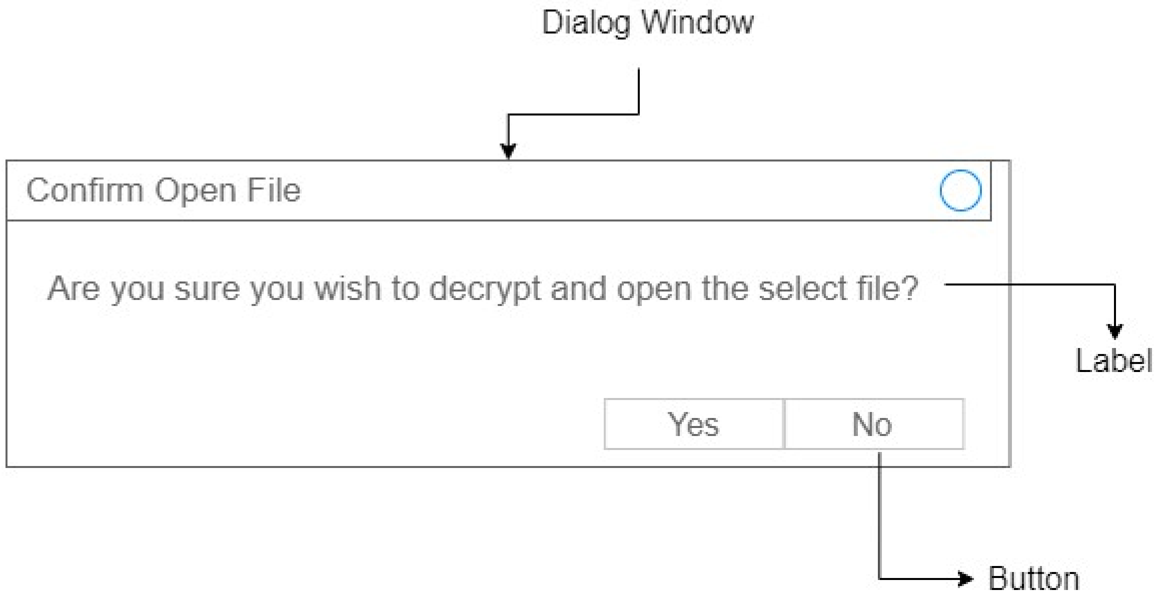

Confirm Encryption

Are you sure you wish to add the selected files and folders to the encrypted storage?

Figure 9 shows the confirmation window for decrypting and opening a selected file.

Figure 9

Open File Confirmation Window

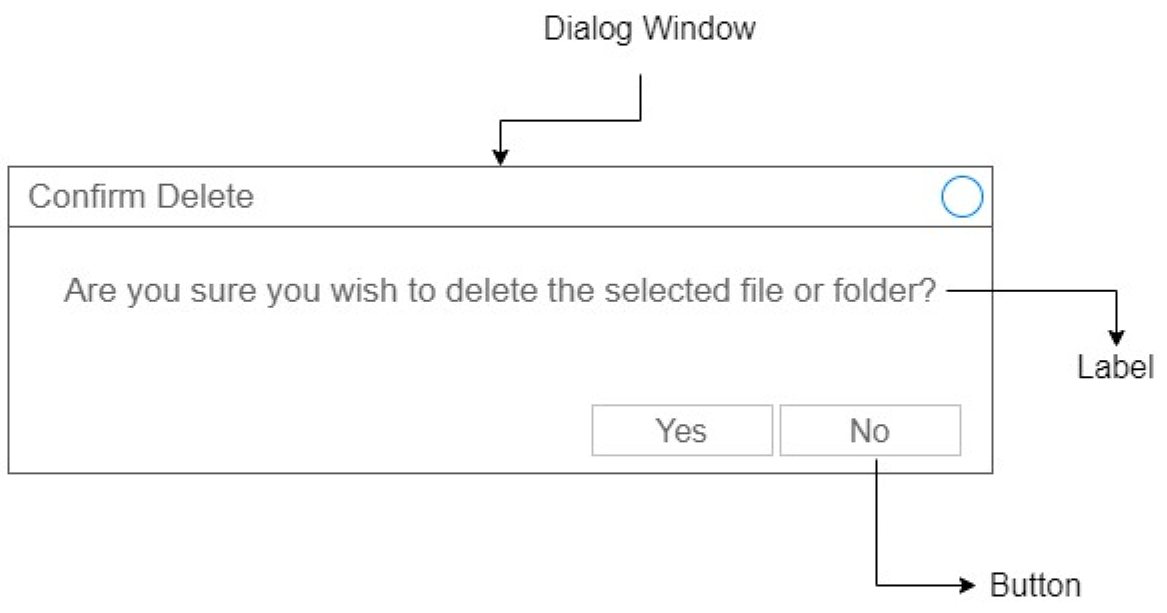

Confirm Open File

Are you sure you wish to decrypt and open the select file?

Figure 10 shows the confirmation window that appears before deleting a file or folder.

Figure 10

Delete Confirmation Window


Dialog Window
Confirm Delete
Are you sure you wish to delete the selected file or folder?
Label
Yes
No
Button