

File Encryptor Project Requirements

Jacob Braddock

Cybersecurity

Dr. Michael O'Neil

May 2025

1. Scope of Project

i. Project Use Case

2. Functional

3. Look and Feel

i. Appearance

ii. Style

4. Usability

i. Ease of Use

ii. Personalization and Internationalization

iii. Learning

iv. Understandability and Politeness

v. Accessibility

vi. Convenience

5. Security

i. Access

ii. Integrity

iii. Privacy

iv. Audit

v. Immunity

1. Scope of Project

i. Project Use Case

PUC No	PUC Name	Actors/Users	Input/Output
1	File Selection	Software User	Selected Files (in)
2	File Encryption Process	User Computer	Encrypting Files (in)
3	Distribute Encrypted Files	User Computer	Encrypted Files (out)
4	Create Recovery Key	User Computer	Distribute Key (out)
5	Encrypted File Selection	Software User	Select Files (in)
6	File Decryption	User Computer	Decrypt Files (in)
7	Distribute Unencrypted Files	User Computer	Distribute Files (out)

2. Functional

Volere Requirement Template		
Requirement No.: 2i	Requirement Type: Functional	Use Cases: 1 5
Description: The program will allow for singular or multiple file selection		
Rationale: Allow user free choice of which files should be encrypted or not en masse		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The user should be able to encrypt as many files as possible in a timely fashion but only may decrypt one file at a time with the proper key		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 3	
Dependencies: 2 3 4 6 7	Conflicts:	
History: Last worked on 22 April, 2024	Supporting Materials: 1i PUC	

Volere Requirement Template		
Requirement No.: 2ii	Requirement Type: Functional	Use Cases: 2
Description: The program will encrypt the earlier selected files		
Rationale: To actually allow for the files to be encrypted		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The files encrypted should be sent to a newly created folder within about 30 at max upon the users initial desire to encrpt files.		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 3 4 5 6	Conflicts:	
History: Last worked on 22 April, 2024	Supporting Materials: 1i PUC	

Volere Requirement Template		
Requirement No.: 2iii	Requirement Type: Functional	Use Cases: 3 7
Description: The program will distribute files upon completion of desired task		
Rationale: To allow for the files upon completion of decryption or encryption to be distributed in specific folder		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The files should be sent to a newly created folder within about 30 at max without any major problems		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 1 4 5	Conflicts:	
History: Last worked on 22 April, 2024	Supporting Materials: 1i PUC	

Volere Requirement Template		
Requirement No.: 2v	Requirement Type: Functional	Use Cases: 6
Description: The program will allow for the user to decrypt their files		
Rationale: To allow for the user to have a way to decrypt their files on the same program should they choose		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The decrypted files should be done and sent to the newly created file within about 30 minutes of runtime		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 3	
Dependencies: 5 7	Conflicts:	
History: Last worked on 22 April, 2024	Supporting Materials: 1i PUC	

3. Look and Feel

3i. Appearance

Volere Requirement Template		
Requirement No.: 3i	Requirement Type: Non-Functional	Use Cases:
Description: The program needs to have a sleek, professional look to it		
Rationale: To allow for the user to get the impression of a well designed, business oriented program		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The look should match general standard office standards for a business used program		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: None	Conflicts:	
History: Last worked on 23 April, 2024	Supporting Materials:	

3ii. Style

Volere Requirement Template		
Requirement No.: 3ii	Requirement Type: Non-Functional	Use Cases:
Description: The program needs to look like something an average person trusts		
Rationale: To allow for the user to trust the program instead of assuming said program is malicious		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: 60% or higher of the persons using said program in test case should agree the program is safe in appearance		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 3i	Conflicts:	
History: Last worked on 23 April, 2024	Supporting Materials:	

4. Usability

i. Ease of Use

Volere Requirement Template		
Requirement No.: 4i	Requirement Type: Non-Functional	Use Cases:
Description: The program should generally be very simple for such a complicated task		
Rationale: To allow for the user to simply and quickly understand how to use and what the program is		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: A user at least at the age of 14 should be able to use the program without major problems		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 3ii	Conflicts:	
History: Last worked on 23 April, 2024	Supporting Materials:	

ii. Personalization and Internationalization

Volere Requirement Template		
Requirement No.: 4ii	Requirement Type: Non-Functional	Use Cases:
Description: The program should allow for light/dark mode and allow for english/spanish toggling		
Rationale: To allow for the user to have simple personal customization options for the program that do not interfere with the main function		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The users should generally be able to read all the program in both languages and be able to fully change visibility options		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 3i 3ii	Conflicts:	
History: Last worked on 23 April, 2024	Supporting Materials:	

iii. Learning

Volere Requirement Template		
Requirement No.: 4iii	Requirement Type: Non-Functional	Use Cases:
Description: The program should be extremely easy to pick up and learn without major hiccups		
Rationale: To allow for the user to learn how to use the program within a reasonable amount of time		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The users should generally be able to use all the functions and know what they do within about 10 minutes of program runtime		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 3ii	Conflicts:	
History: Last worked on 23 April, 2024	Supporting Materials:	

iv. Understandability and Politeness

Volere Requirement Template		
Requirement No.: 4iv	Requirement Type: Non-Functional	Use Cases:
Description: The program will be very general and easy to understand in wording		
Rationale: The program should promote general, easy to understand context of what the program is going to do for the user		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The user should have general idea what to do upon opening the program or a few seconds after		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 3i	Conflicts:	
History: Last worked on 23rd April, 2024	Supporting Materials:	

v. Accessibility

Volere Requirement Template		
Requirement No.: 4v	Requirement Type: Non-Functional	Use Cases:
Description: The program shall be usable by a partially disabled persons		
Rationale: The program should be easy to use for those with a partial or minor disability just like with those without		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The user should have a general ease of use that a majority of partially disabled users can use		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 4i 4ii	Conflicts:	
History: Last worked on 23rd April, 2024	Supporting Materials:	

vi. Convenience

Volere Requirement Template		
Requirement No.: 4vi	Requirement Type: Non-Functional	Use Cases:
Description: The program shall be usable upon download without need of extra needs to use said program		
Rationale: The program should be simple and without needing extra paramters to use		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The program should be approved of its simplicity by a majority of test users		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 4i 4ii	Conflicts:	
History: Last worked on 23rd April, 2024	Supporting Materials:	

5. Security

5i. Access

Volere Requirement Template		
Requirement No.: 5i	Requirement Type: Non-Functional	Use Cases:
Description: The program should only be accessible in its use and files to those who specifically are using it		
Rationale: The program should maintain that only the user of the program has access to the files on their device		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The program should only keep the files and keys on the person's device and no where else		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: None	Conflicts:	
History: Last worked on 23rd April, 2024	Supporting Materials:	

5ii. Integrity

Volere Requirement Template		
Requirement No.: 5ii	Requirement Type: Non-Functional	Use Cases:
Description: The program should only be able to be used for file encryption in an ethical manner		
Rationale: To prevent the abuse of said program for a malicious intent such as ransomware		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The program should unable to be used in other programs in every attempt to try.		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: None	Conflicts:	
History: Last worked on 23rd April, 2024	Supporting Materials:	

5iii. Privacy

Volere Requirement Template		
Requirement No.:	5iii	Requirement Type: Non-Functional Use Cases:
Description: The program should notify the user of the inability for this program to do anything except the intended function		
Rationale: To prevent the breaking of trust of users that may use this program		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The program should be approved by at least 3 different computer specialists for its safety		
Customer Satisfaction Rating:	5	Customer Dissatisfaction Rating: 5
Dependencies:	5i 5ii	Conflicts:
History: Last worked on 23rd April, 2024	Supporting Materials:	

5iv. Audit

Volere Requirement Template		
Requirement No.:	5iv	Requirement Type: Non-Functional Use Cases:
Description: The program should follow the General Accepted Auditing Standards or other auditing rules in place		
Rationale: To prevent the breaking of legal laws or other rules that may be within place		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The program should be approved by all of the college's auditors		
Customer Satisfaction Rating:	5	Customer Dissatisfaction Rating: 5
Dependencies:	2i-v 5i-v	Conflicts:
History: Last worked on 23rd April, 2024	Supporting Materials:	

5v. Immunity

Volere Requirement Template		
Requirement No.: 5v	Requirement Type: Non-Functional	Use Cases:
Description: The program should unable to be affected by a majority of malicious programs from the outside		
Rationale: To prevent interference from outside malicious programs		
Source: Jacob Braddock - Student/Programmer		
Fit Criterion: The program should be able to withstand most common malicious programs or uncorruptable		
Customer Satisfaction Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies: 5i-v	Conflicts:	
History: Last worked on 23rd April, 2024	Supporting Materials:	