Encryption Program

Prepared By: Jacob Braddock

December 9th, 2024

1) Introduction:

The goal of the test plan is to go over the scope, ideals, and various other parts of the Senior Project. The project in question is File Encryption Program which should be designed as presentable, easy to use by anyone, and most importantly, able to encrypt and decrypt files at the whims of what is chosen.

2) Scope:

The program will be tested to handle most common files that exist, primarily document or image-based files that easily can be encrypted and decrypted by the chose encrypting algorithm. The files should range in size at a reasonable amount to make sure the program can handle large sized files and lower sized ones as well, without taking a delayed amount of time, exceeding expected runtime results. The expected process should also not damage the file in any way, shape, or form.

3) Test Strategy:

Testing will be broken down on a step-by-step basis for each individual part of the program written and should fall into a testing of several categories. These categories include:

- **Functional Testing** - Verifies that each feature works according to specifications, including file encryption, file decryption, file type handling, error handling, and user interface functionality.

- **Performance Testing** - Measures the program's response times for various file sizes to ensure encryption and decryption operations are completed within an acceptable timeframe.
- **Usability Testing** - Ensures the interface is easy to use, especially for non-technical users, with clear labels, minimal steps to encrypt/decrypt, and appropriate error messages.
- **Security Testing** - Evaluates the security of the encryption algorithm, ensuring data is securely encrypted, and only decrypted by authorized means.
- **Compatibility Testing** - Ensures the application performs consistently across different operating systems (Windows, macOS, Linux).

4) Test Environment:

The testing environment will range from at least three different computers and operating systems to make sure the program can run on all three. A modern Macbook Pro running on the latest Apple OS, an old 2012 Samsung portable laptop running on Linux, and a common PC running on the latest version of Windows.

5) Test Schedule and Estimation:

Since there has been some delay in the progress of the Senior Project, the expected test schedule will need to occur on a weekly basis. Each week or two should result in a new portion of the program completed, tested to some degree, and presentable. This is to hopefully give the Test Schedule ample time to be worked on and properly tested.

6) Test Deliverables:

Deliverables should be actual programs with working examples able to be shown, either in person or by video. These displays of progress should be followed by a small summary of what the part does, what was done to achieve said part, and testing done to achieve the desired result. The summary should also include what is intended to be worked on next.

7) Exit Criteria:

Testing is complete once:

- All high and critical priority test cases have passed.
- No critical or high-severity defects remain unresolved.
- Functional, performance, usability, and security requirements are met to a satisfactory level.
- All deliverables are reviewed and approved by stakeholders.

8) Suspension and Resumption Criteria:

The testing portion should be suspended if the student becomes encumbered with other classes, the project shifts in priority otherwise, or there is a critical problem with the code that must be addressed before the testing of the code may continue. Resumption begins when the issues of before are resolved.

9) Responsibilities:

The responsibility of the student is to fully create, test, debug, and be able to present a program. This should become documented and well observed by the student to be able back trace and prevent any problems that should arise from the past. It is on the student reach set deadlines, amend them as needed, and make sure that all documentation is up to date.

10) Risk and Contingencies:

Due to the nature of the product, the risks of the test primarily are low resources, compatibility on devices, and the ability to naturally and accurately do the desired task. The process of encryption is a powerful tool and can create major device problems within an instant and proper measures should be taken. It can also be a massive security risk and must be managed as well.

11) Test References:

This test plan is based on the past created documents including the WBS, the Requirements Document, and all past created documents relating to the mentioned project.

12) Review and Approvals:

The finished testing and product will be presented to the advisor for them to review and critique. Ultimately, they will have final word on the project and its endeavors and whether a part is acceptable or not.

Test Results Checklist

Test Category	**Test Description**	**Test Results**
Functional Testing		
1.1 File Encryption	Verify that the file encryption process works correctly for various file types (documents, images, etc.).	**Worked perfectly**
1.2 File Decryption	Verify that encrypted files can be decrypted successfully and return to their original form.	**Works perfectly**

1.3 File Type Handling	Test program's ability to handle various file formats (e.g., PDF, JPG, PNG, DOCX).	**Struggles on large files, encrypts most known files**
1.4 Error Handling	Ensure proper error messages are displayed for unsupported file types, incorrect passwords, etc.	**Displays**
1.5 User Interface	Ensure that the program's UI is intuitive, and all buttons/labels are functional.	**Working**
Performance Testing		
2.1 Encryption Speed	Test the time taken to encrypt files of varying sizes (small, medium, large).	**Fast**
2.2 Decryption Speed	Test the time taken to decrypt files of varying sizes (small, medium, large).	**Fast**
2.3 Program Responsiveness	Ensure the program doesn't freeze or slow down excessively during operations.	**No Freezes**
Usability Testing		
3.1 Ease of Use	Test the ease with which a non-technical user can encrypt and decrypt files.	**All clear**
3.2 User Instructions	Check if the program provides clear, easy-to-understand instructions and error messages.	**All Clear**
3.3 Navigation Simplicity	Test the program's navigation flow (minimal steps to complete encryption/decryption).	**All clear**
Security Testing		
4.1 Algorithm Strength	Verify the strength of the encryption algorithm.	**Strong**
4.2 Data Integrity	Verify that the original data is not altered or corrupted during encryption or decryption.	**Strong**
Overall Testing		
5.1 Test Documentation	Ensure all tests are documented and meet requirements outlined in the test plan.	**Documented**

5.2 Test Report	Verify that a final test report is generated summarizing all tests and results.	**Written**