

SE 3XA3: Test Report
Title of Project

Team #, Team Name
Suhavi Sandhu sandhs11
Shabana Dhayananth dhayanas
Joseph Lu luy89

December 7, 2017

Contents

1	Functional Requirements Evaluation	1
1.1	User Input	1
1.1.1	Master Password	1
1.1.2	Adding Entries	2
1.2	Navigation	2
1.3	Encryption	3
1.4	Database	3
1.5	Password Generation	4
2	Nonfunctional Requirements Evaluation	4
2.1	Look and Feel Requirements	4
2.2	Personalization	4
2.3	Understandability and Politeness Requirements	4
2.4	Performance	5
2.5	Operational and Environmnetal Requirements	5
2.6	Security	5
3	Comparison to Existing Implementation	6
4	Unit Testing	6
5	Changes Due to Testing	6
6	Automated Testing	7
7	Trace to Requirements	7
8	Trace to Modules	7
9	Code Coverage Metrics	7

List of Tables

1	Revision History	ii
----------	-----------------------------------	-----------

List of Figures

Table 1: **Revision History**

Date	Version	Notes
2017/12/06	1.0	Creation

This document discusses the results from testing the product, Password-ProtectionProgram, any changes that were implemented as a result of testing and the extensiveness of the tests performed.

1 Functional Requirements Evaluation

1.1 User Input

1.1.1 Master Password

These test cases verify that the functions related to the master password, a critical part of the system, behave as intended.

The following test cases assume that a master password has not yet been initialized and test for the creation of the master password.

- **FR-MP-1:** A valid input of at least characters having upper and lowercase as well as numbers should let user proceed.
Result: Test pass, user is allowed to proceed.
- **FR-MP-2 to FR-MP-5:** Invalid input for master password should display appropriate error message.
Results: All tests pass and display appropriate error message.

The following test cases assume that a master password has already been initialized and test for logging in with the master password.

- **FR-MP-6:** Upon entry of an empty string, the system should display an error message.
Result: This test originally passed, however, due to a bug found during the testing for update entry, this test was omitted.
- **FR-MP-7, FR-MP-8:** Depending on whether input is correct or incorrect, system lets user proceed or displays error message.
Result: Tests pass

1.1.2 Adding Entries

These tests verify that users are able to add new entries to the database, assuming a connection to the database is already established and user is logged into application.

- **FR-AE-1, FR-AE-2, FR-AE-3:** Originally, the goal of these test cases was to ensure that when adding an entry, a user cannot add an empty password. Upon testing, it became clear that an empty password should be allowed, in the case that the user has not come up with one. Therefore, the test was changed to verify that a Name, rather than a password, should be required.
Results: All test cases pass, allowing users to add entries as long as name is provided, else error message displayed.

1.2 Navigation

The underlying goal for the product was to make it more userfriendly than the original. In doing so, great care was given to the tests designated under navigation. The following tests verify the functionality of the user manual.

- **FR-N-1:** Link to user manual works.
Results: A pdf viewer opens up with the user guide.
- **FR-N-2:** At view entry screen, clicking the question mark should take user to instructions page
Results: Test passes, proving that instructions are readily accessible.
- **FR-N-5:** User should be able to copy a password or username that is in the system database.
Results: Copy button works as intended, test passes
- **FR-N-6, FR-N-7:** test the functionality of the generate button and verify that a string is outputted and can be called random through visual inspection.
Results: All tests pass, output is random to the eye.

- **FR-N-8:** Tests the delete button, ensuring that all entries can be deleted, except master password.
Results: Test passes, deleted entries are immediately removed and the database as well as the interface reflects these changes.
- **FR-N-9:** Tests the update button, ensuring that all entries password and usernames can be changed.
Results: Test passes.
Further discussion: This test led to the discovery of a bug that results from updating a master password to be an empty string. It is a legal action, however, when the user attempts to log back in with the updated password, he/she is prohibited since an empty string was not allowed. A temporary workaround for this bug is to allow empty strings but in the future, there will be unique constraints for updating passwords (i.e cannot be empty).

1.3 Encryption

- **FR-E-1, FR-E-2:** Check that a string being encrypted can ultimately be decrypted to give the original.
Results: All tests pass.

1.4 Database

These tests have to do with the functionality of the database and ensure that it works independently to the rest of the system. Therefore, it is vital to ensure basic methods such as insert, delete, and update work as intended - something that can be proven with the test cases mentioned below.

- **FR-DB-1:** Database can be created
- **FR-DB-2:** Database has already been created
- **FR-DB-3, FR-DB-4:** When multiple entries exist, database should delete them when user deletes and return them when needed:
Results: All tests pass.

1.5 Password Generation

- **FR-PWG-1:** This test case ensures that the criteria for what the team defined as random is achieved by the method.
Results: Test passes. It can be proven that the method generates passwords with exactly 8 characters, including uppercase, lowercase and numbers.

2 Nonfunctional Requirements Evaluation

2.1 Look and Feel Requirements

The following requirements check that the application has a standard interface that is aesthetically pleasing. An example test case below describes generally what was evaluated.

- **NFR-LFR-1:** The user should be able to navigate between screens seamlessly through the use of available buttons.
Results: This test case, like all others in the Look and Feel category, were performed manually by our peers. To ensure that the cases passed, a rating from 1 to 10 is provided from user.
For all the tests the average scores were reasonable.

2.2 Personalization

- **NFR-PSR-1:** The user should be able to specify the category for which they are adding an entry for.
Results: This test was performed manually by peers, who gave feedback in the form of their experience with the personalization aspect as well as a number from 1-10 defining how customizable they felt the application was. In this category, the application received a 7 as the option to add categories is available although there is no sorting by category, something we wish to implement in the future.

2.3 Understandability and Politeness Requirements

- **NFR-UPR-1:** To better understand the needs of users, a survey was conducted in the form of informal interviews with peers that may be

interested in using a password manager.

Results: The outcome of the interview gave insight on the topic of intuitive interfaces. We learned that users are more likely to understand from visual cues rather than text-based as they are more universal. Hence, image symbols were used for as many buttons as possible.

2.4 Performance

- **NFR-PER-1:** To verify the performance of the system, processing time of the `reset_timer` function, a function of the graphical user interface that keeps track of inactivity period, was determined and compared against a `PROCESSING_TIME = 1` second

Result: The test passed, giving a processing time less than `PROCESSING_TIME`, verifying that the performance is up to par with standards.

- **NFR-PER-2:** To verify that error messages are displayed within `ERROR_TIME`, the function for checking an invalid password was timed. Result: The test passed, giving an error time less than `ERROR_TIME`, again, proving that performance is sufficient.

2.5 Operational and Environmental Requirements

The following requirement checks that the application is able to run with Linux and OSX.

- **NFR-OER-1:** The user should be able to run the application with the same functionality as if they are on Windows. Result: All functionality tests passes on other operating systems.

2.6 Security

As the application that we developed being a security applicaiton, testing for security is a high priority of the team.

- **NFR-SR-1:** When the user is creating a new password when starting the application for the first time. They must create the password within the Criteria of set, or the database will not take the password. Result: The test Passes. When entering insufficiently secure passwords, the user can not keep going.

- **NFR-SR-3:** When the application is left inactive for the allocated time given, the application logs out.
Result: The test pass.

3 Comparison to Existing Implementation

4 Unit Testing

5 Changes Due to Testing

There were several changes that were implemented as a result of testing. These changes will be categorized by the main functional requirement that was affected.

1. **FR9** User should be able to change master password:
While manually testing the update master password function, it was noticed that the master password could be updated to be an empty string, which is contradictory to the creation stage where the master password has to be at least 8 characters. As described earlier in the document, a change was made to the password checking module that accepts an empty string.
2. **FR5** User should be able to add entries to the database
While testing the adding entry function through integration of the interface and database, when the entries reached beyond the scrollbar, the scrollbar would not update to increase in size. To fix this, there was code added to refresh the application each time an entry is added. This made it less static than what we originally hoped but ensured that all data would be up to date at all times.

Another change that resulted from this test was that duplicate entries should not be allowed. Originally this was not in the plan but upon seeing that multiple entries could be added with the same name, it was evident that duplicates should not be allowed.

3. **FR7** There should be a link to the user manual
Feedback from those testing the application led to the changes in the

instructions page. The font was made bigger and the steps to use the application were numbered to be more user friendly.

4. **NFR-LFR-1** Look and Feel Requirement - User experience

Some of the feedback that we received regarding user experience was invaluable since the main goal of the application was to increase user-friendliness. Below are the changes that resulted from feedback.

Making Master Password Creation instructions more clear - explicitly define criteria for a valid master password.

When user updates an entry, system should show a message like Changes saved! to show that something occurred behind the scenes.

6 Automated Testing

7 Trace to Requirements

8 Trace to Modules

9 Code Coverage Metrics