Safe-Key

Team 4

Ruth Jimenez, Andrew Le, Jerstine Medrano, and Carter Rath

Software Requirements Specification Document

Version: (1) **Date: (10/13/2021)**

Table of Contents

1 Purpose

For anyone that has multiple accounts such as for social media, banks, devices, etc., passwords can come as a challenge. Users want to create a unique password but at the same time one difficult enough that others can't access. For multiple accounts, you have the option of having a different password for every login you have. The pro is that each password is different and unique, the con? Forgetting several passwords and having to reset it or getting locked out of the accounts. Then begins the displeasure of resetting the password. In order to avoid remembering every single password or getting locked out, some found using the same password for everything as a solution. While it is a good idea at first, there is a greater risk there. If an unauthorized user obtains your password, they then have access to all your accounts. Between having the same password and having a different password for every account. Having a diverse amount is safer. How do we then get over the problematic idea of remembering every single password?

2 Scope

With the need for unique passwords and a secure hub for storing them, we have created the system called Safe-Key. The purpose of Safe-Key's software is to provide a secure and convenient hub to store any number of passwords that the user desires. The way Safe-Key will create a secure password storage is through the use of encryption by taking the user's passwords, encrypting them, and storing them in the software's password hub. The goals of Safe-Key are to provide organizational methods of storing specific passwords that pertain to social media or even more important passwords, that way the user will not lose track of which password accesses what service they want to use. Another goal is to create the utmost password storage security to ensure the privacy and security of potential users. The benefits of having an encrypted password storage are the dependable method of storing the user's passwords in a hub that only the user can access with their own specialized key or pin, which adds another layer of security with our software. Another feature of our software is the lockdown feature: when there are too many failed attempts at accessing the hub storing all of the passwords, the hub automatically goes on lockdown until the user is allowed to try accessing the passwords once again. Finally, the last potential benefit of our software is the optional self-destruct feature where Safe-Key deletes the encrypted passwords when there have been too many failed attempts at entering the software's hub.

3 User characteristics

The types of users may vary widely. In general, our users are anyone who wish to have a safe and secure way to store their passwords. Since using the same password for different accounts makes it easier for hackers to access information, many people create passwords unique to each of their accounts. This is a smart security measure to take, but it is difficult to remember multiple passwords, so they need a place to store them for reference.

3.1 Key users

Our key users would most likely be those with disabilities that impact memory and the elderly. The age for this group of users would be teenage years and older. As mentioned previously, it is a highly recommended security measure to take by creating different passwords for each online account. However, this program will be made very simple to use since we know these will be our key users and they are not very familiar with technology. They are most likely all novice-level users and do not know about the encryption process that is behind the scenes of the program. In a world that is constantly

advancing in software and technology, we do not want these users to view this as another challenging and unfamiliar program they must use. We want them to view this program as a helping tool that makes their technology use easier and more secure. With this information, we will focus on creating a simple design for the user interface with very few steps and provide thorough directions for those who need it.

3.2 Secondary users

On the other hand, our secondary users are quite different from the key users. These secondary users may be gamers or anyone who has multiple accounts to where it would be impossible to remember all the different passwords they created. Again, the age for this group of users could be teenage years and older. Since they are frequent users of different software, they are most likely a journeyman, or even a master with technological experience. With this, using our program may be easy for them and they would have no problems. They probably would not need to view a tutorial or have steps to guide them through the process. However, the key users are our biggest priority, so we will still have directions. As a compromise, we will give these secondary users an option to skip the tutorial and go straight to entering their passwords.

3.3 Unimportant users

Lastly, the unimportant users would be just anyone looking for a convenient and safe way to store their passwords. This group of users could be classified as unimportant because there is no accurate way to categorize them since they could be anybody who knows how to use a computer. A more specific group who misuse the product could be hackers trying to learn the encryptions and get through the security we provide. Their technological experience is most likely master-level, and their age could be 18 and older. If they were to gain access to these passwords, it would be very bad for us since our users trust us to store them safely. With this in mind, we need to make sure our software is as secure as we can possibly make it with our experience.

4 Product perspective

4.1 System Context

4.2 User interfaces

1. *It is required that the system provides password encryption confirmation to the user*
2. *It is required that the system must provide the optional feature of deleting the encrypted passwords after a failed number of attempts to enter the system.*

3. *It is required that the user is provided backup or recovery options if something were to happen to their computer or mobile device.*
4. *It is required that the user is provided the recovery options in the event that they cannot verify themselves into their account, or the passwords are bound to be deleted if that feature is enabled.*

4.3 Software interfaces

The system must use mySQL as a way of storing the encrypted passwords for each individual user in an organized and retrievable way. Communication with the DB will be through ODBC connections as well as there is a driver that allows for easy readability and usage. The reason why we are using mySQL is to avoid paying a server fee to house all the data for the encrypted passwords. The system will also be implemented on GitHub as they have open-source websites or pages that students can use so purchasing a subscription plan will not be needed.

4.4 Hardware interfaces and Memory constraints

- The terminal output should support full-screen of up to 1200x800 pixel.
- The terminal output should support user input through keyboard and mouse
- Because most standard laptops/desktop computers have at least 4G of RAM, it is required that the design footprint should not exceed 4G.

4.5 Deployment requirements

Java Runtime Environment or Java Development Kit are required to run the application. It is also required to have internet access because it will allow our system to interact with mySQL and retrieve the stored, hashed passwords, to the website to be decoded for the user upon verification.

5 Assumptions and Dependencies

- It is assumed that users will have access to a computer with a keyboard to type their passwords.
- It is assumed that users will have at least some basic knowledge of using software and are at a novice level or above.
- It is assumed that users know what a password is and have a list of their own passwords to type into our program.
- It is assumed that users have a capability to follow directions given by the program to use it.
- It is assumed that we will have a secure and effective encryption selection.

6 Specific requirements

6.1 System Functional Requirements

R.6.1.1: As a user, I want to be able to save an unlimited amount of passwords

R.6.1.2: As a user, I want my passwords encrypted as much as possible not just minimal encryption.

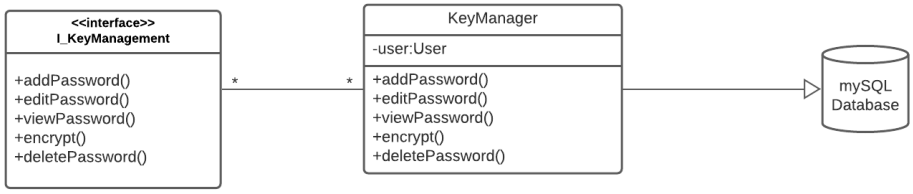
R.6.1.3: As a user, I want to be comfortable securing my information or passwords, so I don't get hacked or lose that information.

R.6.1.4: As a user, I want to be able to recover my document if I, for whatever reason, cannot verify myself, so I do not lose all my information.

R.6.1.5: As a user, I want to be insured that my passwords will be saved after I exit the program, so I do not need to gather and enter them in again.

R.6.1.6: As a user, I want to be reminded of old passwords to see if any particular accounts are still active or if I can delete them.

6.2 Logical Database Requirements



6.3 Software System Attributes

6.3.1 Usability

R.3.1.1: The system should be accessible at all times (24/7)

R.3.1.2: The system should be compatible with computers and other smaller devices such as cell phones

R.3.1.3: The system should be clear so people of any age find it easy to navigate

R.3.1.4: The system should display a set of instructions as another courtesy so users know that they must enter a password and that it will be encrypted.

6.3.2 Performance

R.3.2.1: The system shall support multiple users at the same time

R.3.2.2: The system shall allow a user to document as many passwords

R.3.2.3: The system shall prompt all passwords per user request

R.3.2.4: The system shall take less than 1 second to encrypt a password

R.3.2.5: The system shall encrypt every password entered

6.3.3 Reliability/Dependability

R.3.3.1: When a password is entered the system shall save the password once “save password” button is pressed.

R.3.3.2: When exiting out, the user shall be able to go back into the system and view all the passwords they have saved thus far.

R.3.3.3: When a password is a couple years old, the system will notify the user and the user may delete if password is no longer active

R.3.3.4: When updating the system, all previous user data entered will not be erased

6.3.4 Security

R.3.4.1: System will utilize SHA-251 as a cryptographic security

R.3.4.2: System shall have a history which will allow it to display when passwords are entered

R.3.4.3: System shall prompt authenticated users through a verification process to access information.

R.3.4.4: System shall encrypt all data entered.

R.3.4.5: System shall only access data and will ask for permission which the user can accept or deny

6.3.5 Maintainability

R.3.5.1: The code shall be well organized and commented to allow other contributors to understand the work of their peers.

R.3.5.2: Variable will be commented to avoid confusion with any similar variable

R.3.5.3: Before entering a loop, contributors will comment what they do

R.3.5.4: If a portion of coded is needed in more than two parts, it will be turned into a method to avoid unnecessary repetition.