SOFTWARE REQUIREMENTS SPECIFICATION

for

Shared Password Manager System

Release 1.0

Version 1.0 approved

Prepared by The Better Team

Contents

List of Figures 3

List of Tables 4

1 INTRODUCTION 6
- 1.1 Purpose . . . 6
- 1.2 Document Conventions . . . 6
- 1.3 Acronyms . . . 7
- 1.4 Project Scope and Product Features . . . 7

2 OVERALL DESCRIPTION 8
- 2.1 Product Perspective . . . 8
- 2.2 Product Features . . . 8
- 2.3 User Classes and Characteristics . . . 11
- 2.4 Operating Environment . . . 11
- 2.5 Design and Implementation Constraints . . . 12

3 NONFUNCTIONAL REQUIREMENTS 13
- 3.1 Performance Requirements . . . 13
- 3.2 Safety Requirements . . . 13
- 3.3 Security Requirements . . . 13
- 3.4 Software Quality Attributes . . . 14

4 STANDARDS AND REFERENCES 15
- 4.1 References . . . 15

List of Figures

2.1 Regular User Diagram 9
2.2 View individual credential screen 10
2.3 Use Case Diagram 11

List of Tables

Revision History 5

Acronym table 7

4.1 Applicable Standards 15

Revision History

Date	Description	Revised by

1 INTRODUCTION

1.1 Purpose

This document exists to inform stakeholders of the functions and specification requirements of the Shared Password Manager Software System. The goal is to explain how the system shall hold a collection of passwords for various systems and keep track of them in a single, secure, accessible place. It also details the credentials, usernames, passwords, timestamps for the last time a password was updated, and notes for the password and system that each user is given.

1.2 Document Conventions

This document conforms to the IEEE Standards Style Manual. Therefore, "shall," "should," "may," and typographic conventions are utilized based on the definitions by IEEE standards.

1.3 Acronyms

Acronym	Meaning
NFR	nonfunctional requirements
OpenBSD	Open Berkeley Software Distribution
SQL	Structured Query Language
NIST	National Institute of Standards and Technology
UI	User Interface
HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
AES	Advanced Encryption Standard

1.4 Project Scope and Product Features

We propose a Shared Password Manager Application that stores password credentials. The system will use a website application environment to provide functionality for the three types of users to interact with the system: admin, regular, and view-only. To use the application, a user will first login to their account. The Application will then display options dependent on their privilege level. For admin users, they will have the ability to add/remove users, while regular users will have the functionality to view, add, and edit entered credentials. Additionally, View-only users will only have privilege to view credentials. The User Data will be stored securely in an SQLite database using a server.

2 OVERALL DESCRIPTION

2.1 Product Perspective

This project solves the problem of managing a collection of shared credentials and accessing those credentials securely. It will also create a role-specific interface to interact with data based on the class of user (admin, regular, and view-only).

2.2 Product Features

List of product features
1. Functional Features
- Management system shall store credentials in a shared manner
- There shall be three classes of Users: Admin, Regular, View-only
- Admins shall have the ability to create, edit, and delete users
- Regular users shall be able to add, edit, view, and delete credentials
- View-only users shall only be able to view credentials

2. Performance Features
- The website shall feature a simple universal login screen for all users
- After login, users shall be shown a screen displaying their credentials, see Figure 2.2.
- The website shall load dynamically depending on the user type
- The website shall have a button to add or edit a credential from the list
- The Admin user shall have an extra button to view a user list
- Users shall be able to view each of their stored credentials and their stored information, see Figure 2.1
- Users should be able to view a particular credential by clicking on an in-

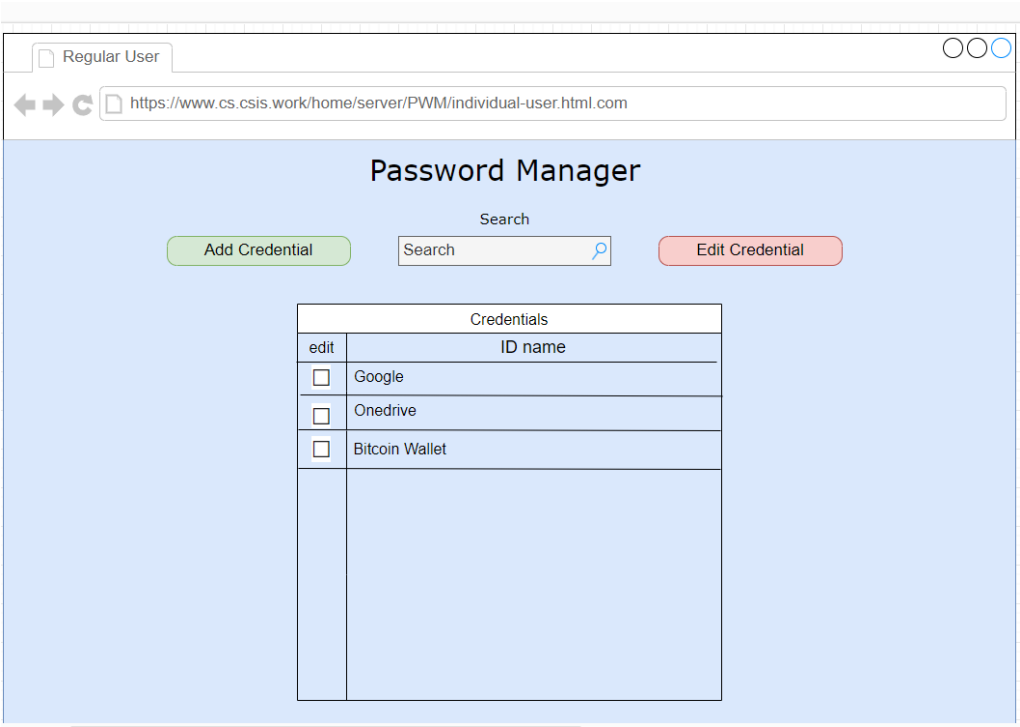

Figure 2.1: Regular User Diagram

dividual credential and viewing that credential, see Figure 2.2 - View-only users shall have the add and edit credential buttons greyed out.
- If the user exits the session, they shall be required to log back in
3. Aesthetic Features
- The website shall present an interface that offers... reference chapter 7?
- The website shall offer high affordance to features: add and edit credential
- The website shall not overwhelm the users with harsh color patterns
- Buttons a particular user cannot use shall be greyed out

4. Complementary Features
- A search bar shall allow users to search their account's list of stored credentials by name.

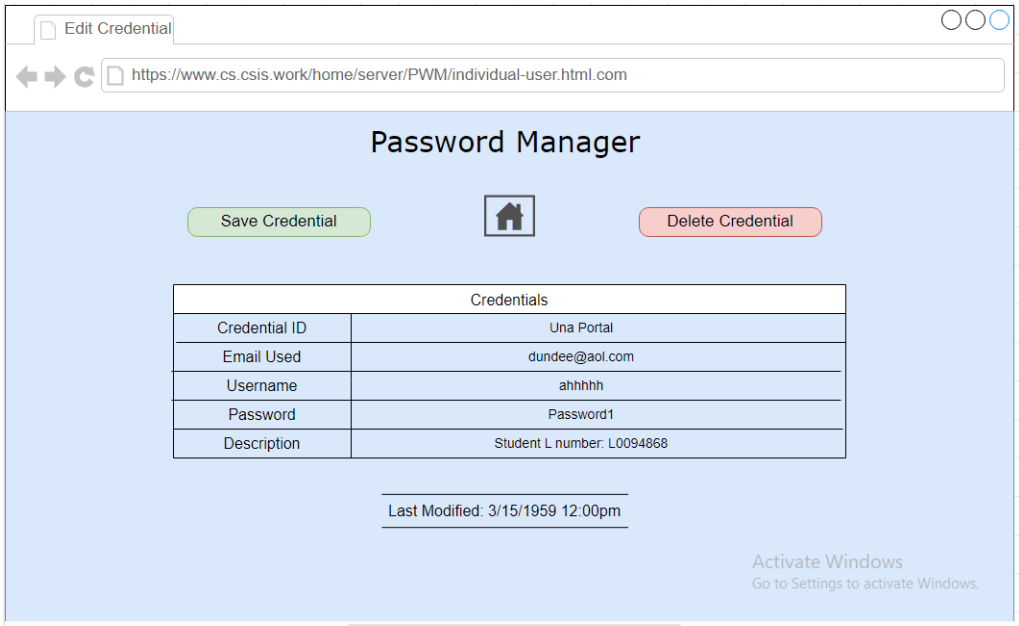

Figure 2.2: View individual credential screen

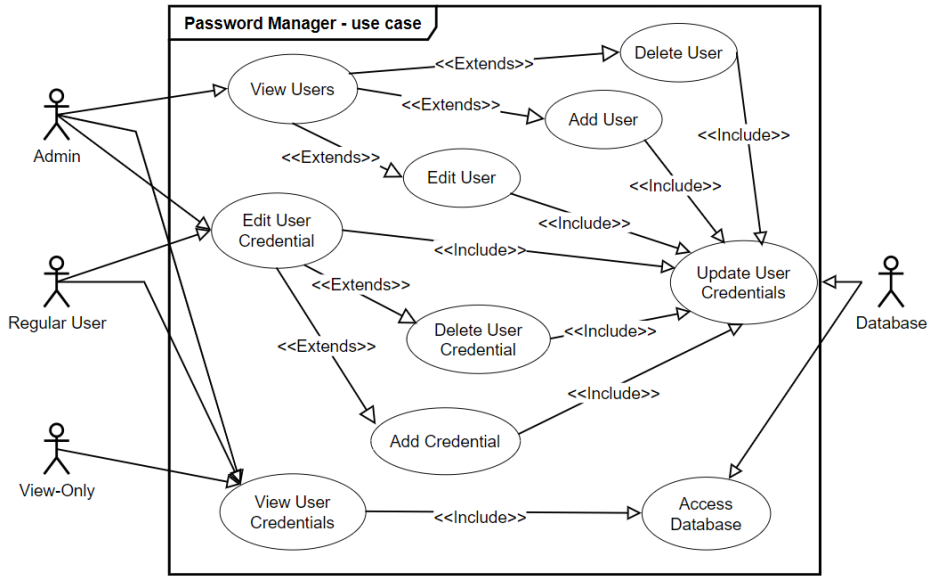

Figure 2.3: Use Case Diagram

2.3 User Classes and Characteristics

There are three different classes of users interacting with the software. These three users include an admin, who can create, edit, and delete users; a regular user who can edit, view, and delete credentials; and a view-only user who can only view the credentials. A database will send and receive information to manage the storage and access of user credentials.

Figure 2.3 illustrates the use case diagram that represents the different users and their interaction with the system.

2.4 Operating Environment

The software shall run on an OpenBSD server and shall have at least 512MB of disk space allocated. The software shall utilize a SQLite database, therefore a minimum of 512MB of RAM needs to be allocated. The server shall

allocate a maximum of 2GB for storage of the data file for the SQLite database.

2.5 Design and Implementation Constraints

The finished product shall run on an OpenBSD server. The system shall be developed in the C++ language. The system shall make use of the web toolkit "witty."

3 NONFUNCTIONAL REQUIREMENTS

3.1 Performance Requirements

(R1.1) The software shall support 10 users while maintaining a maximum response time of 2 seconds.
(R1.2) The software shall exhibit response times of between 0-2 seconds after user input.
(R1.3) Unplanned extended downtime shall not exceed 1 minute each week.
(R1.4) The software shall support a minimum of two users altering credentials at the same time.

3.2 Safety Requirements

This software does not require any safety requirements due to its design and intended use.

3.3 Security Requirements

(R3.1) Admin passwords shall not be obtainable by any user but that specific admin.
(R3.2) Data log access shall only be available to Admin users.
(R3.3) All data transfers containing user data shall be encrypted.
(R3.4) Sessions shall expire after an inactivity of 30 minutes.

(R3.5) The software shall enforce password authentication requirements following the NIST Digital Identity Guidelines (SP 800-63B), as specified in the Applicable Standards (Table 4.1).
(R3.6) Accounts shall lock after three repeated failed login attempts.
(R3.7) For any database login, a minimum of AES encryption shall be used before authentication.
(R3.8) The software shall prevent data loss of user credentials by completing daily backups of all files to a secondary location.
(R3.9) All interactions with the database shall access the database with the least privilege possible for the least amount of time.

3.4 Software Quality Attributes

(R4.1) The software shall be divided into a minimum of 2 components that can be modified individually.
(R4.2) Components shall allow the ability to update without affecting other parts of the system.
(R4.3) Software shall be accompanied by documentation.
(R4.4) The software shall protect against unauthorized access and data breaches.

4 STANDARDS AND REFERENCES

Table 4.1: Applicable Standards

Standard	**Application in Project**
IEEE Standard for Requirements Engineering (IEEE Std 29148-2011)	Applicable standards for this project include the IEEE Standard for Requirements Engineering (IEEE Std 29148-2011). Therefore, "shall," "should," "may," and typographic conventions are utilized based on the definitions by IEEE standards.
NIST Digital Identity Guidelines (SP 800-63B)	Secure password standards for this project include the NIST Digital Identity Guidelines (SP 800-63B) and should enforce Single-Factor Cryptographic Software password complexity requirements for the software. Password management guidelines from NIST are utilized.

4.1 References

[1] SQL Maestro Group, "SQLite Admin Tool - SQLite Database Browser by SQL Maestro Group," Sqlmaestro.com, 2021. (accessed Oct. 01, 2024) https://www.sqlmaestro.com/products/sqlite/maestro/help/00_04_00_system_requirements/
[2] "Basic Installation," OpenBSD Handbook, 2024. (accessed Oct. 01, 2024). https://www.openbsdhandbook.com/installation/ [2] NIST Pass-

word Guidelines, "NIST Special Publication 800-63-3 Digital Identity Guidelines". (accessed Oct. 01, 2024). Ehttps://pages.nist.gov/800-63-3/sp800-63a.html