| **Team Number** | 01 | **Section** | 02 |
| --- | --- | --- | --- |
| **Team Members** | 1. Bryson Leshkiw 2. Elmar Imanov 3. Nuha Rushad 4. Ricques Nguyen | | |
| **Software Name** | Personal Password Manager | | |

## Problem Statement

Website security is becoming a bigger issue in recent years due to the increase in the number of websites and people reusing the same password for every website which increases the risks of being hacked. The client requests us to develop a new password manager software as the existing ones are not reliable since older software is more at risk due to there being more time to find vulnerabilities or their own developers selling the information. There have been numerous data breaches in the past and new software needs to be made as studies have shown that people are relying on corrupt software. Groups that are affected by this are everyone who uses any form of technology on the internet.

## Functional Requirements

This product has these features:

1. Ability to handle multiple users.
2. Ability to login, sign up and reset the password.
3. During signup, users must agree to a security question to be able to reset the master password later on.
4. Ability to create accounts to add to the password manager, this includes username, password, and possibly a hint.
5. The password manager can create a password for the user, the criteria to make the password is as follows: maximum/minimum length of generated password can be set by the user, the user has the option of including the number of special characters(!"#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~), and also the option of including capital letters with user specifications.
6. Ability to search passwords by either website/application name or username.
7. Ability to modify password/account.
8. Ability to copy password with the click of a button (copy to clipboard, button version of ctrl+c).
9. The password manager can store username, password, website/application name, email, creation date, expiration date. This will be stored in SQLite (or some kind of database here).
10. The password information will be encrypted before storing into the manager and decrypted before retrieving for the user.
11. The password manager will warn the user about passwords that are about to expire. Passwords will be set to expire after a year in which the user will have to create a new one.
12. This will be a desktop application and will be user friendly.

## References

<https://www.passwordmanager.com/password-manager-trust-survey/>

## Use Cases

1.)

| **Use Case Name** | | Login | |
| --- | --- | --- | --- |
| Goal | | | |
| This use case describes the users ability to login to the password manager | | | |
| Participating Actors | | | |
| User, System | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User opens the application  User clicks “Login” button | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs the manager | | System shows login page |
| 2 | User inputs username and password (master password)  and clicks the login button | | System verifies and finds match |
| 3 | N/A | | System takes the user to the homepage.  System shows user saved passwords and accounts |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| User inputs wrong name or password | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs the manager | | System shows login page |
| 2 | User inputs username and password | | System verifies and finds error |
| *3* | N/A | | System shows message, “Wrong Username/Password or empty field” |
| 4 | N/A | | System stays at login page |
| 5 | User inputs incorrect username and password multiple times (3+ times) | | System suggests to reset password, display message “Try resetting your password” |
| 6 | User clicks cancel | | System returns to main page |

| Alternate Flow of Events | | | |
| --- | --- | --- | --- |
| **Alternate Trigger** | | | |
| User logins and warning for expiring passwords shows up | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | After step 2 on successful login | | System shows home page |
| 2 |  | | System displays warning on expiring passwords |

2.)

| **Use Case Name** | | Signup | |
| --- | --- | --- | --- |
| Goal | | | |
| This use case describes the users ability to sign up for the password manager | | | |
| Participating Actors | | | |
| User, System | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User clicks in “Sign Up” button on the login page | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs the manager | | System shows login page |
| 2 | User clicks on “Sign Up” button | | System shows sign up page |
| 3 | User inputs all information correctly and presses sign up  Note: Security question about resetting will be asked here | | System displays message “sign up successful” and brings user back to login page to login again |
| 4 | N/A | | System stores the account information on the signupUsers database |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| User clicks on cancel | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs the manager | | System shows login page |
| 2 | User clicks on “Sign Up” button | | System shows sign up page |
| *3* | User clicks “Already Registered? Log in” | | System brings user back to login page |

3.)

| **Use Case Name** | | Reset | |
| --- | --- | --- | --- |
| Goal | | | |
| This use case describes the users ability to reset the master password | | | |
| Participating Actors | | | |
| User, System | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User will press on “Forgot password?” text on the login page | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs the manager | | System shows login page |
| 2 | User presses “Forgot password?” text | | System shows reset page,system asks for username, security question, security answer, and the new password |
| 3 | User inputs their username, security question and answer, and the new password | | System verifies the information |
| 4 | If the User answers the security question correctly and gives the correct username | | System then saves the new password in the database |
| 5 | User presses the reset button | | After the reset button is pressed the password is saved and the System returns the user to the login page to enter the new password |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| User clicks on “Create an Account” in the bottom left at any time during reset page progress | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User presses “Forgot password?” on the homepage | | System shows reset page |
| 2 | After step 1: User presses “Create an Account” | | System sends the user to the create account page |
| *3* | After Step 2: User presses “Already registered? Log in?” | | System sends the user back to the login page |

| **Alternate Trigger** | | | |
| --- | --- | --- | --- |
| User inputs wrong username/email on reset page | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs the manager | | System shows login page |
| 2 | User presses “Forgot password?” on the login page | | System shows reset page,system asks for username, security question and answer, and the new password |
| *3* | User inputs incorrect username/email | | System verifies and finds an error, sends the warning “wrong username/security answer” |

4.)

| **Use Case Name** | | Copy | |
| --- | --- | --- | --- |
| Goal | | | |
| This use case describes the users ability to copy their passwords into the clipboard for ease of access | | | |
| Participating Actors | | | |
| User, System | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User clicks the “Copy” button to copy a password to the clipboard | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs the manager | | System shows login page |
| 2 | User inputs username and password | | System logs into account |
| 3 | User clicks the “Generate password” button | | System goes to the generate password page |
| 4 | User fills in the text fields for length of password,number of special characters, and number of capital letters. User then clicks the “generate” button. | | System reads the text fields and generates a password based on the user’s given criteria. |
| 5 | User clicks on the “copy” button next to the generated password. | | System copies the password from the text field a saves it to the clipboard |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| N/A | | | |

5.)

| **Use Case Name** | | Search | |
| --- | --- | --- | --- |
| Goal | | | |
| Users will be able to search for passwords based on searching application username or website. | | | |
| Participating Actors | | | |
| User, System | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User types in the search bar with the name of a username or website | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs the manager | | System shows login page |
| 2 | User logins successfully | | System verifies and shows saved information |
| 3 | User clicks on search bar and types the name of a website or username | | System searches for the website or username from the list and finds it |
| 4 |  | | System shows the user the account information they searched for |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| Search bar can’t find what user is inputting | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | On step 3: user inputs website into search bar that doesn’t exist | | System shows search results, which come up empty |
| 2 |  | | System show message “Search is empty” |

6.)

| **Use Case Name** | | Modify | |
| --- | --- | --- | --- |
| Goal | | | |
| User is able to edit info in password manager | | | |
| Participating Actors | | | |
| User, System | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User presses edit button when looking at account information | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs manager | | System shows login page |
| 2 | User logins successfully | | System verifies and shows saved information |
| 3 | User clicks on a website to view info | | System shows website info, which is username and password |
| 4 | User clicks on edit button | | System shows prompt or text box to edit info |
| 5 | User puts in correct format of new information | | System saves (encrypts) and sends user back to website info page |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| User inputs wrong format of new information | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | At step 5: User puts in wrong account info (ex. wrong password length with wrong amount of capital letters) | | System gives error message “Invalid input please try again” and makes user try again |

7.)

| **Use Case Name** | | Add an Account | |
| --- | --- | --- | --- |
| Goal | | | |
| User is able to add new accounts into the list | | | |
| Participating Actors | | | |
| User, System | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User presses the “add an account” button on the accounts page | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs manager | | System shows login page |
| 2 | User logins successfully | | System verifies and shows saved information |
| 3 | User clicks the “add an account” button | | System takes the user to the add an account page |
| 4 | User fills in all of the fields and clicks the “submit” button | | System verifies that all of the fields are filled and saves the information |
| 5 | N/A | | System returns the user to the account list page and adds the new account to the list |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| User does not fill all of the text fields | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs manager | | System shows login page |
| 2 | User logins successfully | | System verifies and shows saved information |
| 3 | User clicks the “add an account” button | | System takes the user to the add an account page |
| 4 | User clicks the “submit” button without filling all of the text fields | | System gives the user the warning message “one or more empty fields. please fill all..” |

| Alternate Flow of Events | | | |
| --- | --- | --- | --- |
| **Alternate Trigger** | | | |
| User clicks the home button | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs manager | | System shows login page |
| 2 | User logins successfully | | System verifies and shows saved information |
| 3 | User clicks the “add an account” button | | System takes the user to the add an account page |
| 4 | User clicks the “home” button in the top left | | System takes the user back to the list of accounts page |

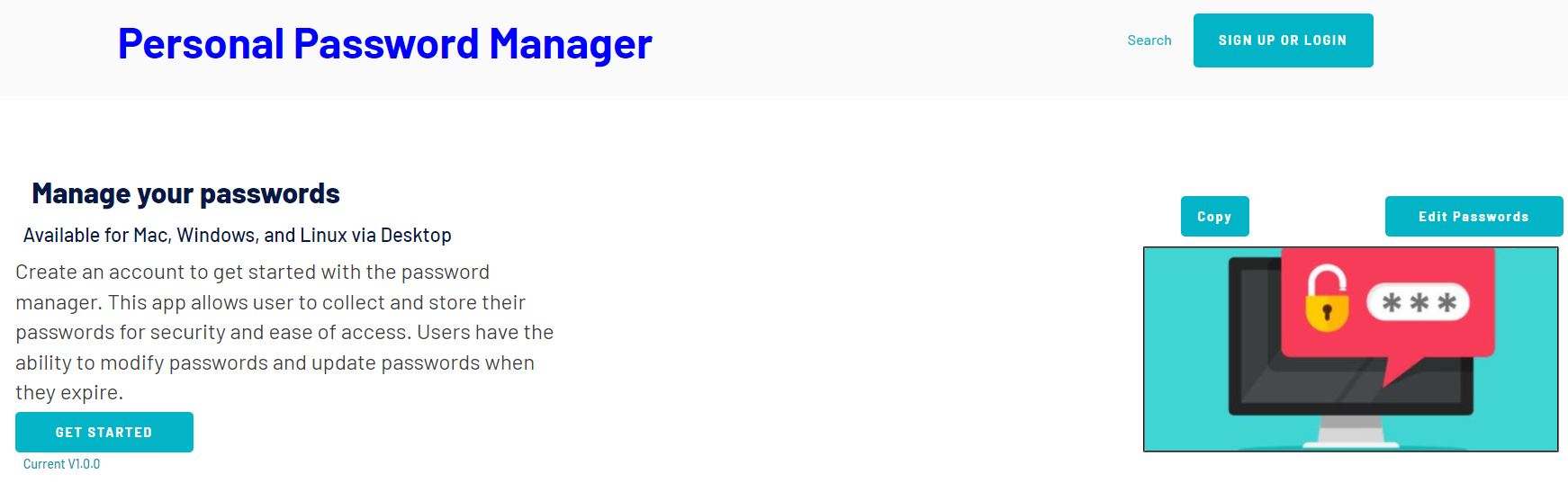
8.)

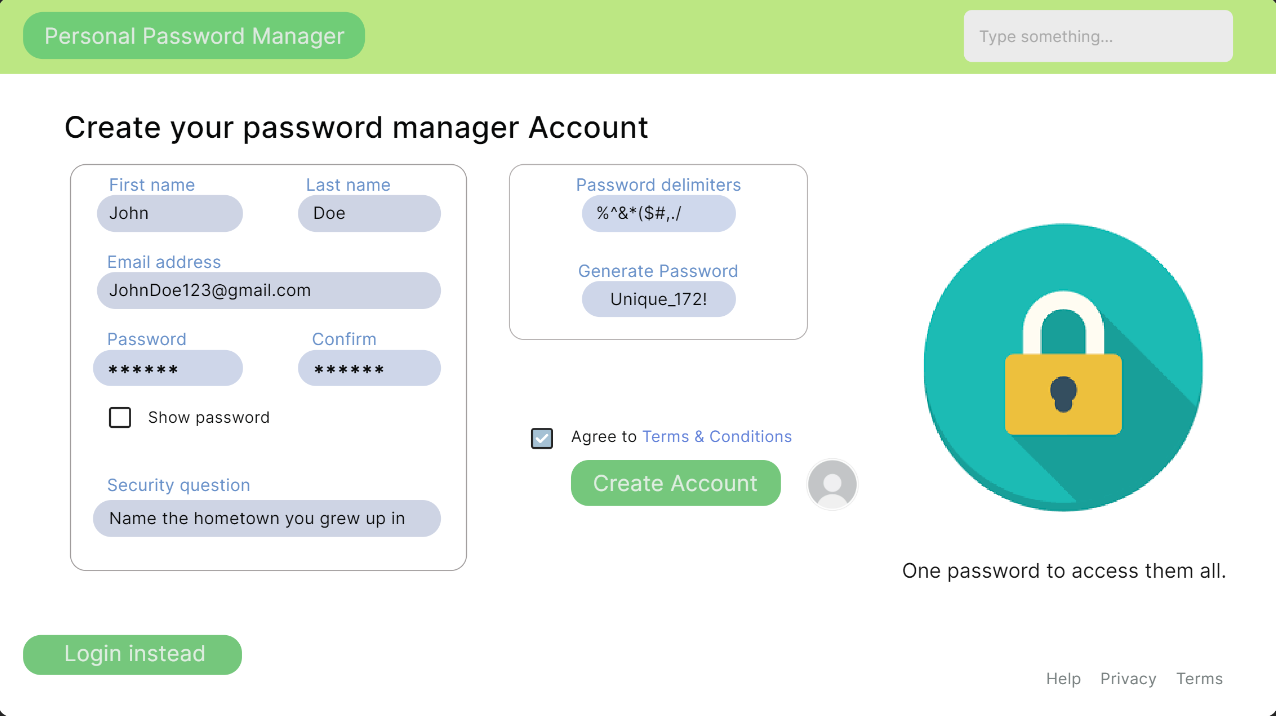
| **Use Case Name** | | Delete an Account | |  |
| --- | --- | --- | --- | --- |
| Goal | | | |  |
| User is able to delete accounts from the table | | | |  |
| Participating Actors | | | |  |
| User, System | | | |  |
| Glossary | | | |  |
| N/A | | | |  |
| Primary Flow of Events | | | |  |
| Trigger | | | |  |
| User clicks on an account and clicks the delete an account button | | | |  |
| **Steps** | **Action** | | **System Response** |  |
| 1 | User runs manager | | System shows login page |  |
| 2 | User logins successfully | | System verifies and shows saved information |  |
| 3 | User clicks on an account from the list | | System highlights that specific account and all of its info |  |
| 4 | User clicks the “delete an account” button | | System removes that account from the list and no longer displays it |  |
| Alternate Flow of Events | | | |  |
| **Alternate Trigger** | | | |  |
| User logs out | | | |  |
| **Steps** | **Action** | | **System Response** |  |
| 1 | User runs manager | | System shows login page |  |
| 2 | User logins successfully | | System verifies and shows saved information |  |
| 3 | User clicks on the “Log Out” button | | System signs the user out returning to the login page |  |

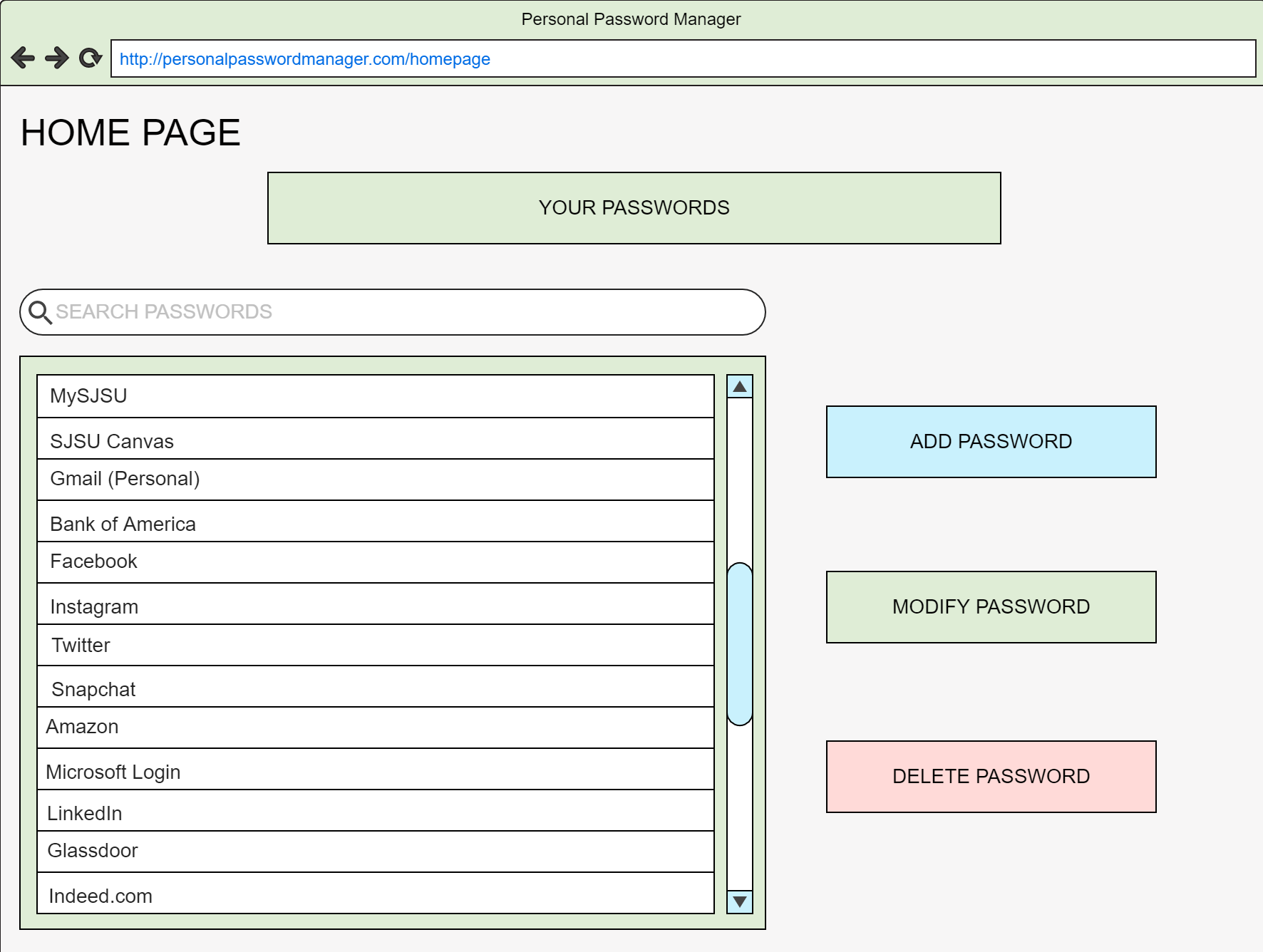
9.)

| **Use Case Name** | | Generate password | |
| --- | --- | --- | --- |
| Goal | | | |
| User is able to generate their own passwords based on the criteria they give the application | | | |
| Participating Actors | | | |
| User, System | | | |
| Glossary | | | |
| N/A | | | |
| Primary Flow of Events | | | |
| Trigger | | | |
| User presses the “generate password” button in the account list page | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs manager | | System shows login page |
| 2 | User logins successfully | | System verifies and shows saved information |
| 3 | User clicks the “generate password” button | | System takes the user to the generate password page |
| 4 | User fills in the text fields for length of password,number of special characters, and number of capital letters. User then clicks the “generate” button. | | System reads the text fields and generates a password based on the user’s given criteria. |
| Alternate Flow of Events | | | |
| **Alternate Trigger** | | | |
| User clicks the home button on the password generator page | | | |
| **Steps** | **Action** | | **System Response** |
| 1 | User runs manager | | System shows login page |
| 2 | User logins successfully | | System verifies and shows saved information |
| 3 | User clicks the “generate password” button | | System takes the user to the generate password page |
| 4 | User clicks the home button in the top left | | System returns the user to list of accounts page |

## Mockups







## Glossary

| N/A | N/A |
| --- | --- |
|  |  |
|  |  |