CS301.1 Master Vault IDD Report

Written by Conor Cook

Working with Nicki Choo

Supervisor: Rouwa Yalda

April 19th, 2024

Contents

[Goals & Objective 3](#_Toc164110162)

[Updated Timeline 3](#_Toc164110163)

[Feature List 4](#_Toc164110164)

[New Features 4](#_Toc164110165)

[Improvements 4](#_Toc164110166)

[Team Members Responsibilities 4](#_Toc164110167)

[Tools & Technologies 5](#_Toc164110168)

[Figma 5](#_Toc164110169)

[Git Hub 5](#_Toc164110170)

[Trello 5](#_Toc164110171)

[Visual Studios Code 6](#_Toc164110172)

[Python 6](#_Toc164110173)

[Flask 6](#_Toc164110174)

[Flask is a lightweight and flexible web development framework for Python. It simplifies building web applications with its minimalistic design and extensive libraries. 6](#_Toc164110175)

[MongoDB 6](#_Toc164110176)

[SRS Document 7](#_Toc164110177)

[Functional Requirements 7](#_Toc164110178)

[Save Different Login Details 7](#_Toc164110179)

[Account Types 7](#_Toc164110180)

[Search and Filter 7](#_Toc164110181)

[Non-Functional Requirements 7](#_Toc164110182)

[Secure/Lock Passwords 7](#_Toc164110183)

[256-AES Encryption 8](#_Toc164110184)

[Secure Login 8](#_Toc164110185)

[Move MasterVault Online 8](#_Toc164110186)

[Paraphrase Generator 8](#_Toc164110187)

[Diagrams 9](#_Toc164110188)

[UI/UX Modelling 12](#_Toc164110189)

[Lo-fi Wireframe 12](#_Toc164110190)

[Usability Testing 14](#_Toc164110191)

[High-fi Wireframe 15](#_Toc164110192)

[Usability Testing 15](#_Toc164110193)

[References 16](#_Toc164110194)

[Appendix 1 16](#_Toc164110195)

# Goals & Objective

**Goals:**

* Enhance security measures to protect user data from unauthorized access.
* Offer diverse password customization options.
* Deploy application online for public use.

**Objectives:**

* Implement 256-AES encryption to secure user passwords and sensitive data.
* Introduce animal ID authentication feature to enhance login security.
* Add paraphrase generator to improve our password generator.
* Expand current MasterVault for family password managements.

## Updated Timeline

Fig 1: Updated Timeline

## Feature List

### New Features

* Saving different login details
* Adding types of account options
* Search & Filter
* Secure/lock passwords

### Improvements

* 256-AES Encryption
* Securer login
* Move MasterVault online
* Paraphrase Generator

## Team Members Responsibilities

For the development of this product, we decided to divide the work into back-end development and front-end development. I (Conor) will work on the back-end development, while Nicki will work on front-end development. However, we will not exclusively work on front and back-end. These are just the main area each of us will focus on.

Conor

* 256 AES Encryption
* Moving app online

Nicki

* About Us Page
* Animal ID

Conor & Nicki:

* Paraphrase Generator
* Family Accounts

## Tools & Technologies

### Figma

(Field, 2016)

We used Figma for making the Lo-fi and High-fi wireframe, as well as for User Testing.

### Git Hub

(Wanstrath et al., 2008)

GitHub will be used for code collaboration. It has been a stable product for collaboration, and we have plenty of experience using it in the past.

### Trello

(Cannon-Brookes & Farquhar, 2011)

Trello is a project management tool. It allows us to have a visual representation of tasks using lists and cards. It allows us to assign work to teammates and track the progress of one another and the progress of the project.

### Visual Studios Code

(Microsoft, 2021)

Visual Studios Code is a programming software that has many inbuild features and downloadable extensions that will be vital for development.

### Python

Python's simplicity, extensive libraries, and built-in cryptography make it ideal for developing a password manager across multiple platforms. It also has a code library for web development

### Flask

### Flask is a lightweight and flexible web development framework for Python. It simplifies building web applications with its minimalistic design and extensive libraries.

### MongoDB

(P. Ryan, 2009)

MongoDB is a powerful NoSQL database, perfect for storing and managing unstructured data efficiently. Its flexibility makes it ideal for web applications, providing fast and reliable data storage.

# SRS Document

## Functional Requirements

### Save Different Login Details

Users should have the ability to save different login details aside from the traditional email and password. They should have the ability to store dates, pins and more for ease of access.

### Account Types

This feature allows users to create different account types within the password manager, such as personal and family accounts. With a family account, members can share saved passwords among authorized family members***.***

### Search and Filter

Users should have the ability to search for specific login details or apply filters to quickly locate desired accounts among their saved information.

## Non-Functional Requirements

### Secure/Lock Passwords

The password manager should provide mechanisms to securely store and lock passwords, preventing unauthorized access to sensitive information.

### 256-AES Encryption

Utilizing 256-bit Advanced Encryption Standard (AES) encryption ensures that stored data is highly secure and protected against unauthorized access or data breaches.

### Secure Login

This feature addresses the inadequacies of the previous login system by enhancing security measures. The goal is to significantly improve upon the previous login system, ensuring that user authentication processes are robust, resilient, and capable of effectively safeguarding sensitive information from unauthorized access.

### Move MasterVault Online

Transitioning the MasterVaultonline ensures accessibility from anywhere with an internet connection, improving convenience for users.

### Paraphrase Generator

This feature generates complex and unique paraphrases based on a keyword provided by the user, which can be used as strong passwords for their accounts, enhancing security by creating passwords that are difficult to guess or crack through brute-force attacks.

## Diagrams

Fig 2: Paraphrase Generator Activity Diagram

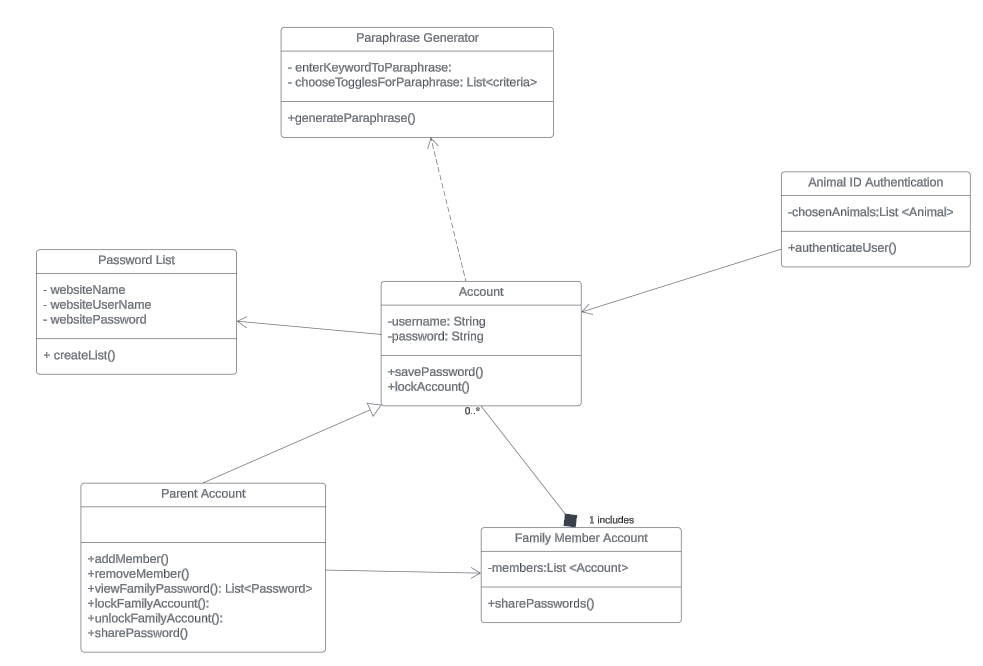


Fig 3: MasterVault Class Diagram

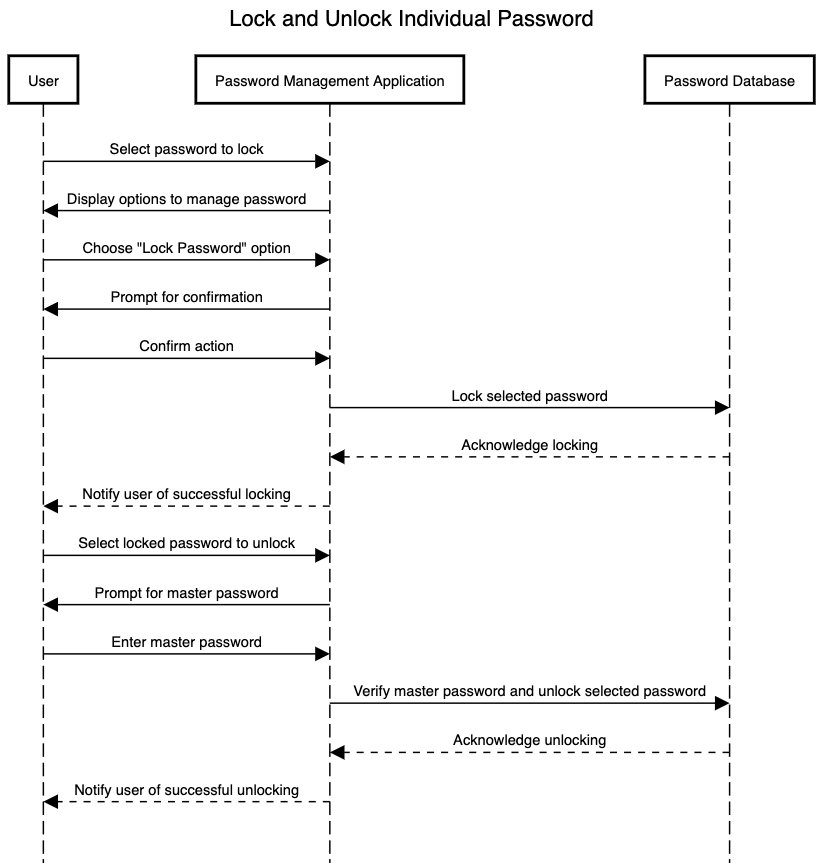


Fig 4: Lock Password Sequence Diagram

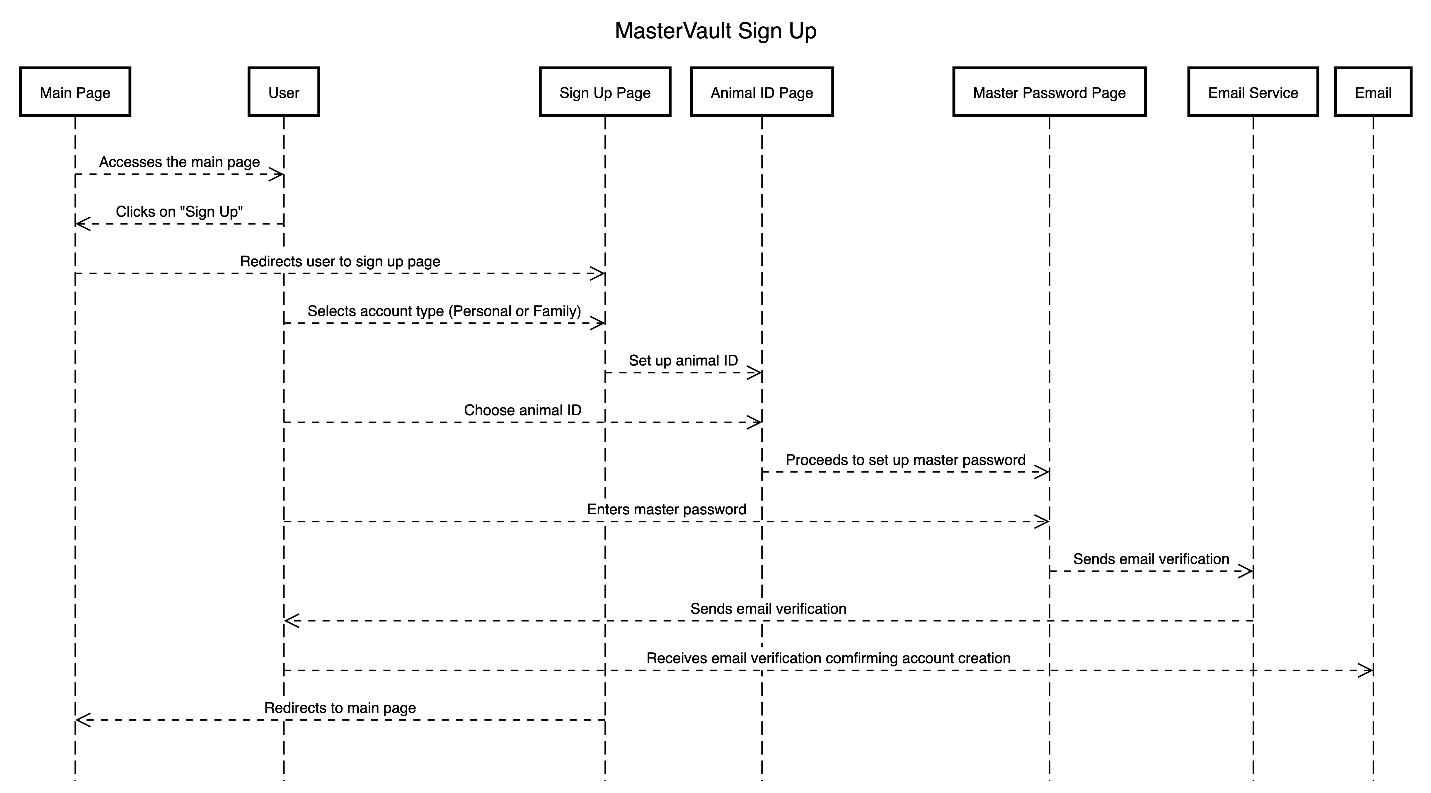


Fig 5: Sign Up Sequence Diagram

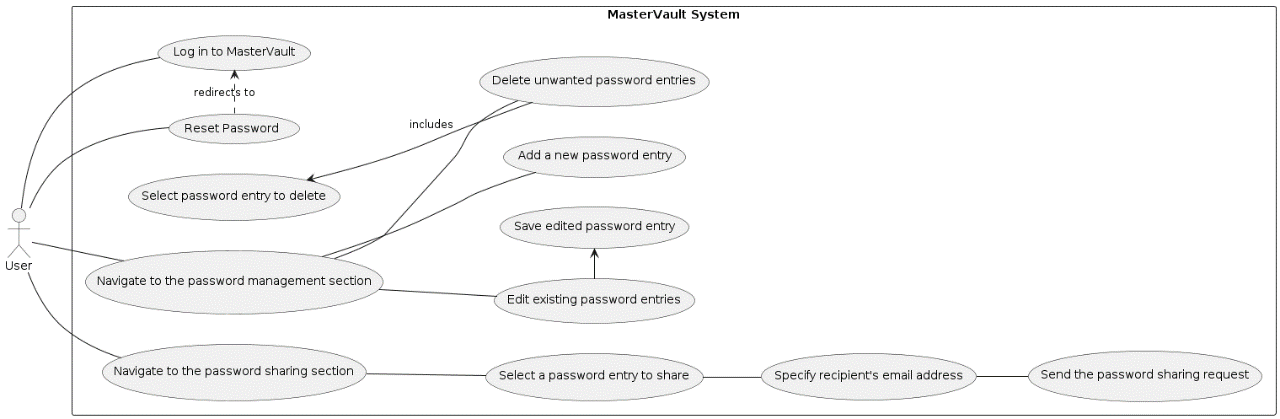


Fig 6: System Use Case Diagram

# UI/UX Modelling

## A screenshot of a login form Description automatically generatedLo-fi Wireframe

Fig 10: Create Master Password

Fig 8: Login Master Password

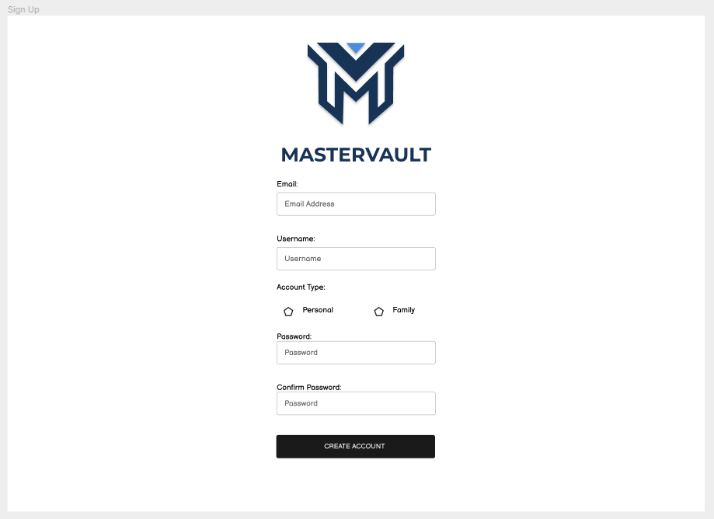


Fig 9: Sign Up

Fig 7: Login

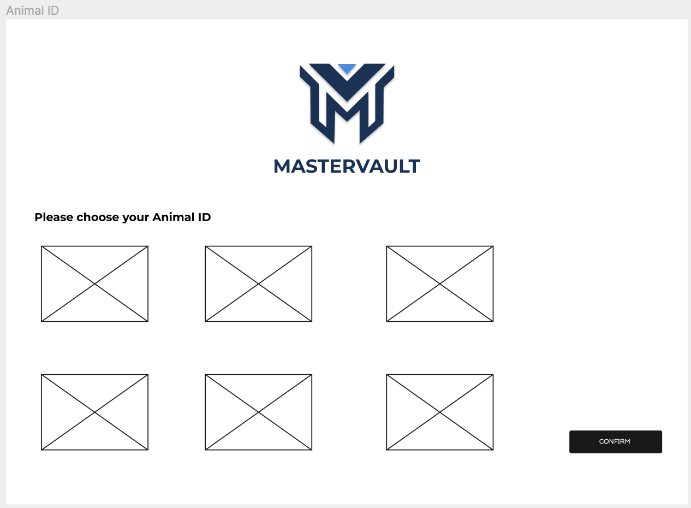
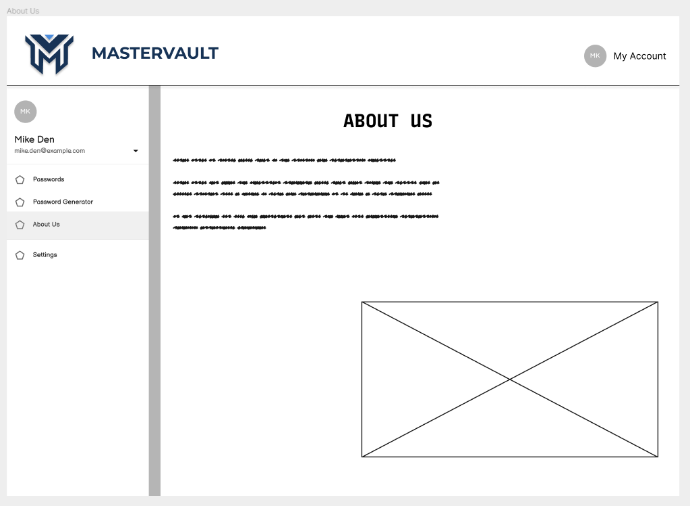
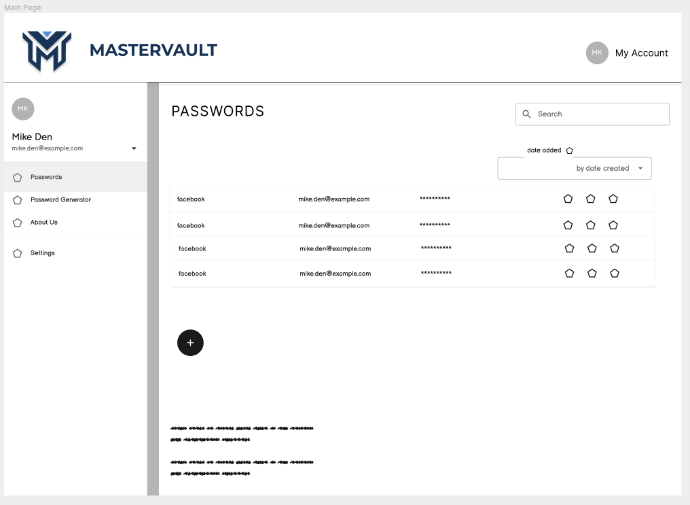
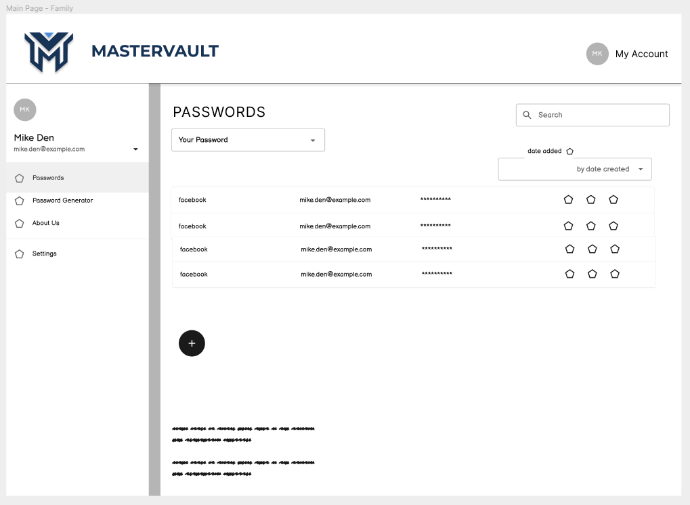
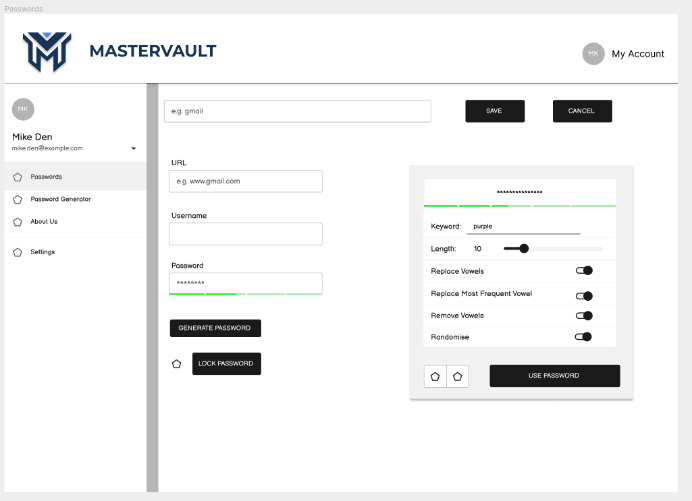
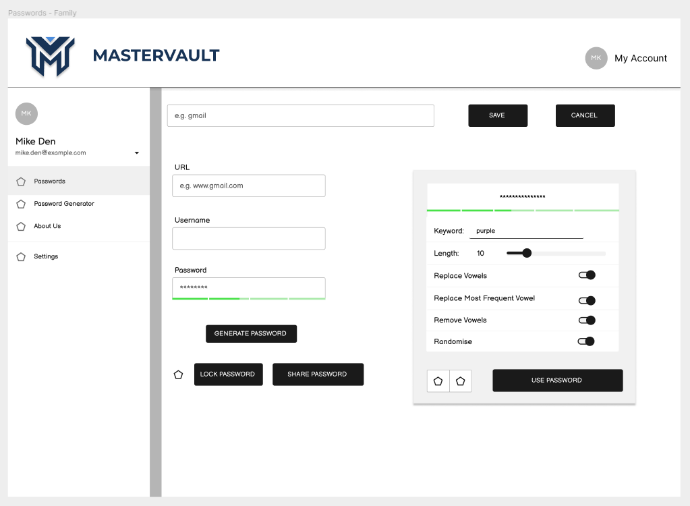


Fig 16: Password View (Family)

Fig 15: Password View

Fig 14: Password List (Family)

Fig 13: Password List

Fig 12: About Us

Fig 11: Animal ID

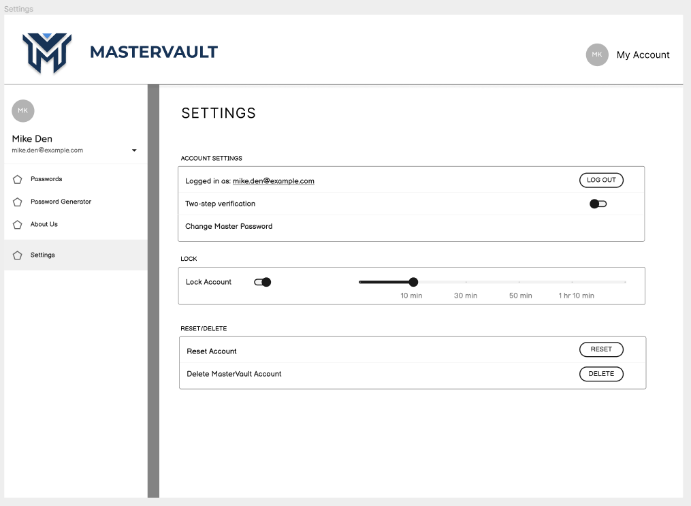
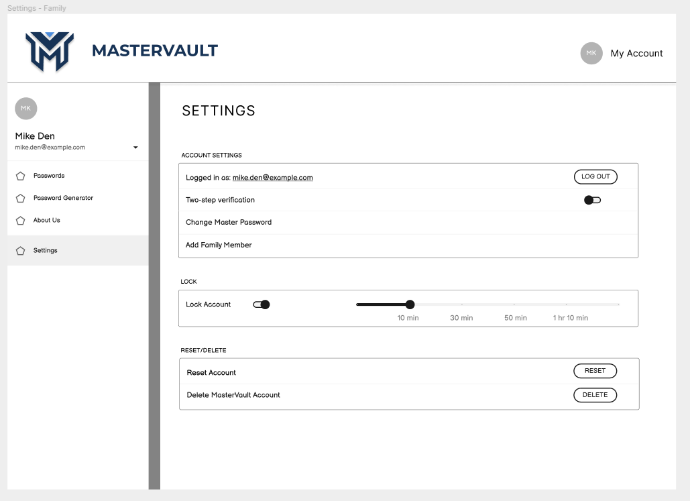
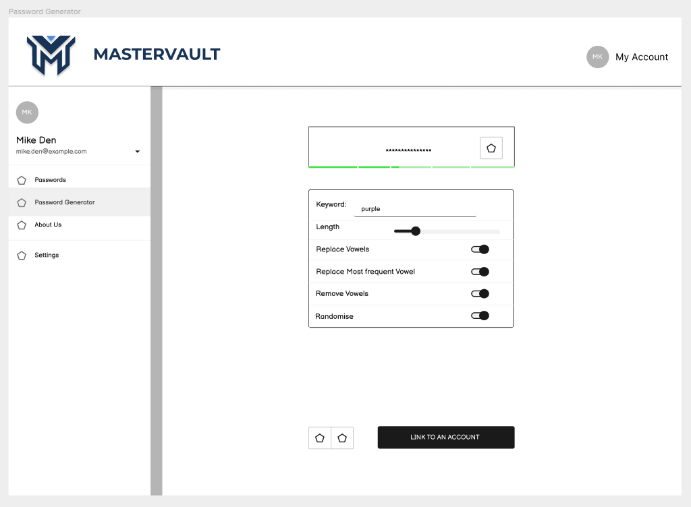
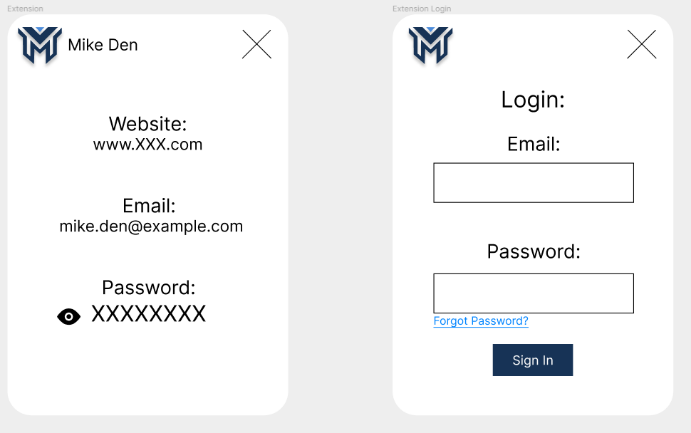


Fig 20: Extentions

Fig 19: Paraphrase Generator

Fig 18: Settings (Family)

Fig 17: Settings

## Usability Testing

## High-fi Wireframe

# References

Field, D. (2016). *The Collaborative Interface Design Tool*. Figma. <https://www.figma.com/>

Wanstrath, C., Preston-Werner, T., & Hyett, P. J. (2008). *Let’s build from here*. GitHub. <https://github.com/>

Cannon-Brookes, M., & Farquhar, S. (2011). *Trello brings all your tasks, teammates, and tools together*. Trello. <https://trello.com/>

Microsoft. (2021, November 3). *Visual studio code - code editing. redefined*. RSS. <https://code.visualstudio.com/>

P. Ryan, K. (2009, February 11). *The developer Data Platform*. MongoDB. <https://www.mongodb.com/>

# Appendix 1

Fig 1: Updated Timeline Page 3

Fig 2: Paraphrase Generator Activity Diagram Page 9

Fig 3: MasterVault Class Diagram Page 10

Fig 4: Lock Password Sequence Diagram Page 11

Fig 5: Sign Up Sequence Diagram Page 12

Fig 6: System Use Case Diagram Page 12

Fig 7: Login Page 1

Fig 8: Login Master Password Page 1

Fig 9: Signup Page 1

Fig 10: Create Master Password Page 1

Fig 11:

Fig 12:

Fig 13:

Fig 14:

Fig 15:

Fig 16:

Fig 17:

Fig 18:

Fig 19:

Fig 20: