Introduction

This document outlines the Password Manager project for the comp1004 module. The following document will outline the project plan, sprints, UML documentation, and challenges and constraints that appeared during the project.

Software Development Lifecycle

For this project I used the agile model with scrum meetings, this allows for projects to be completed efficiently and because it is not an iterative model, steps can be revisited if any issues present themselves.
The steps for the Agile model are the following:

Requirements Analysis stage

This stage consists of analysing the project plan and deciding what requirements the product needs to be considered complete, these requirements consist of functional and non-functional requirements. Functional requirements are features that are integral to the application working as intended and non-functional are features that are separate and provide additional functionality. In the case of my project, the main non-functional requirement I have is the accessibility features like the dark mode.

Design Phase

The design phase consists of making diagrams and project plans to map out the project, in a business, this ensures the project is to the clients standard before the creation can begin, in the case of my project, the designs ensured the project remained coherent and did not stray from the goal outlined previously.

Implementation Phase

This stage occurs after all the designs are completed and consists of creating the actual application using the designs and requirements as a reference. In my project, this took up most of the time as in this stage is a lot that can go wrong, such as errors and other issues with code.

Testing Phase

In this phase, the product is tested by either myself or a user tester to ensure all the features work as intended. User testing is preferable as it is very hard for the developer to use the application as a user. The more people that use the web app, the more likely it is they will run across a complex bug. In the case of my project, testing was carried out by myself and friends and family.

Deployment Phase

This phase consists of publishing the application to its userbase and testing how it responds to running on users' machines, in the case of my project, this phase will not be carried out but my project is ready for pushing if needed, the only thing to do is host it and add higher security.

Maintenance phase

As my application will not be published, this phase does not apply to the project, but In a real-world scenario, this phase consists of maintaining the application and fixing any issues that appear after the deployment.

Sprints

Sprint 1

Outline

For this sprint, the plan was to create the design documentation for the project as it would help me get an idea of the look the site should have and how the different

features would be presented to the end user. During this sprint I layed out the requirements and created the user stories diagrams.

Completed

- A set of wireframes illustrating the design for the website home screen
- A set of UML diagrams outlining the functionality of the web app

Partially completed

- HTML elements for the login feature.

Next Steps

The plan for the next sprint is to develop the navigation features for the website.

Planned features

- Navigation bar functionality

Sprint 2

Outline

For this sprint, the plan was to create the navigation functionality for the website, this was done by hiding the HTML elements with CSS and JavaScript in order to keep the web app as a single page.

Completed

- A working navigation bar complete with hover animations so the user can see what they are selecting.
- Navigation functionality using CSS and JavaScript.
- Adapted the navigation bar names to make them more suitable
- Created a wireframe for the log in page so the HTML could be completed.
- Finished the basic HTML in order to add the functions.

Next Steps

The plan for the next sprint is to create a JSON file and create the page for displaying the data to the user.

Planned features

- Display data from a JSON file to the user.

Sprint 3

Outline

For this sprint, the plan was to create a JSON file and a page for displaying data from the JSON file.

Completed

- Created a JSON file and populated it with example data
- Created a page to display the data from the JSON file

Partially completed

- Data from the JSON file can be viewed in the console but does not display to the user in the display section of the site.

Next Steps

The plan for the next sprint is to create function to display the data in the JSON file.

Planned features

- Display data from a JSON file to the user.
- Allow the application to check if a user already exists.

Sprint 4

Outline

For this sprint, the plan was to create the function to allow the user to view saved passwords and to create the function to check if a user already exists on the system.

Completed

- System checks to see if a user already exists.

Not completed

- Display data from the JSON file to the user as it took time to figure out the JSON functionality.

Next Steps

During this sprint, the functionality to display the JSON information to the user on the site was not completed, I will attempt to complete this in the next sprint. I am considering adding accessibility features in the next sprint.

Planned features

- Display data from a JSON file to the user.
- Accessibility features

Sprint 5

Outline

For this sprint, the plan was to create the functionality for displaying the JSON data to the user as well as some accessibility features like a dark mode. This sprint very short as most of my time was spent learning and preparing for the later Javascript functionality

Completed

- Added a light and dark mode to make the site easier to view for some users.
-

Partially completed

- Data from the JSON file can be viewed in the console but does not display to the user in the display section of the site.

Next Steps

The plan for the next sprint is to create the log in functionality, this means that when the user logs in to an account that exists in the JSON file, the display will change to show them the features available to them, like the page to add and change a Password.

Planned features

- Display data from a JSON file to the user.
- Allow the application to check if a user already exists.

Sprint 6 (final)

Outline

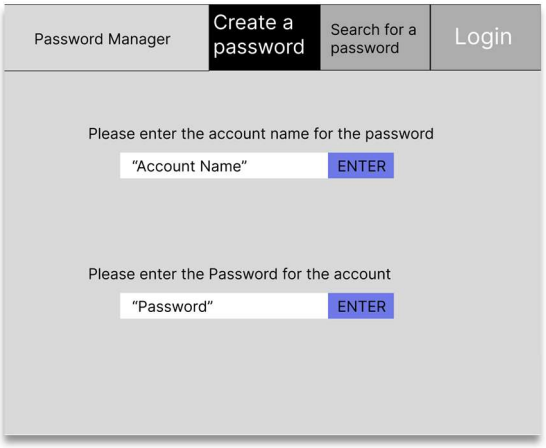
The main bulk of the features were created In this sprint as everything that needed to be completed first was already completed. These features were integral to the applications functionality but they proved difficult to implement, for this reason I chose to leave them till last.

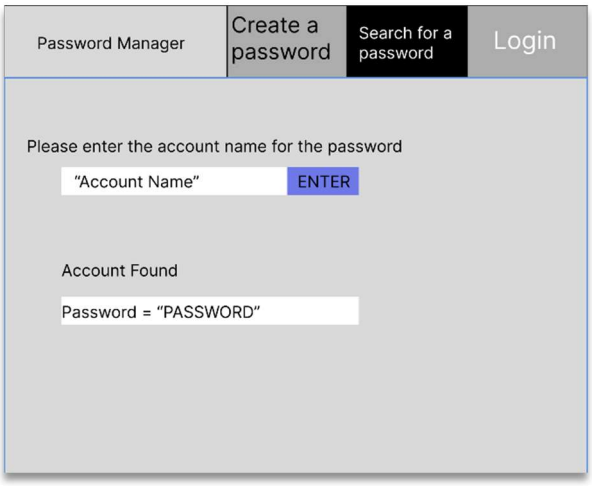
Completed

- Added the ability to see the data saved under a username in the json file and have it display on the site.
- Added the ability to add a new user/ signup
- Added the password strength feature when adding new passwords
- Added the ability to log in to the site
- Added a JSON export function

Design Document

Below are three wireframe diagrams created during the design phase of the project. This design has been created with user ease in mind, Large buttons have been chosen on the navigation bar and a clear separation between the navigation buttons and the page content has been chosen to avoid clutter.





Project Vision and Background

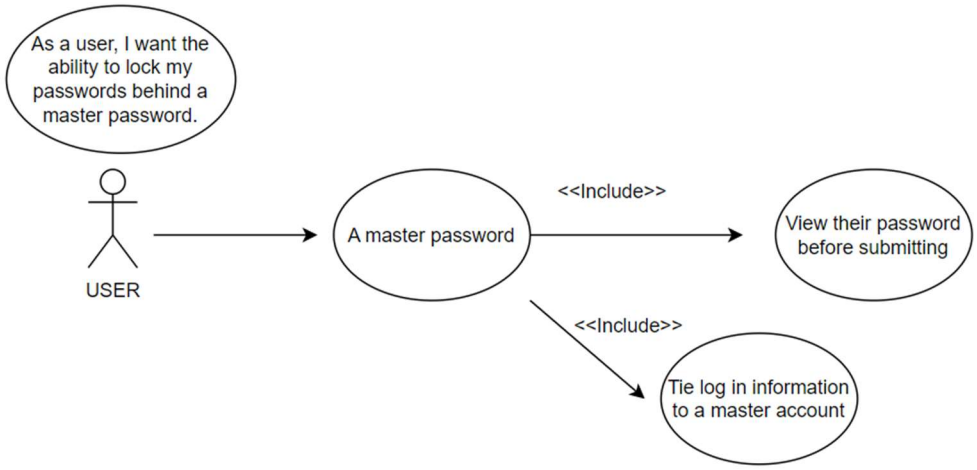
The purpose of this project is for users to create and store strong passwords in order to keep themselves safe online. The latest security standards for passwords can make passwords hard to remember, especially when it is strongly recommended not to reuse passwords. This password manager aims to help users store passwords, the usernames they are linked to, and the website they are for. The main vision for the project is to make it less tedious for users to keep their online accounts safe.

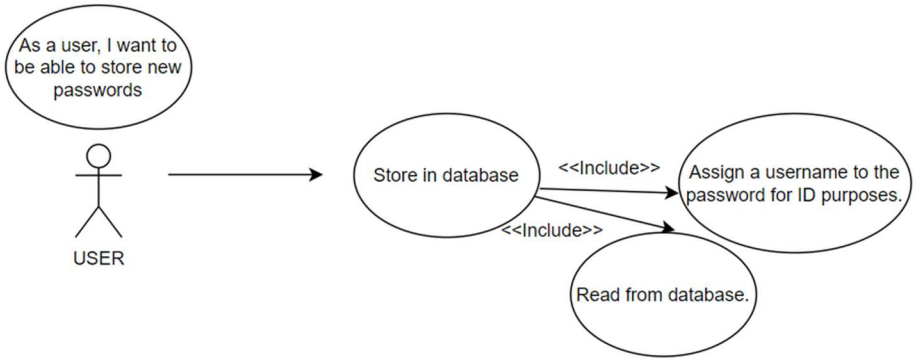
Due to the nature of the web application, it is important we consider the legal and ethical impact, if passwords are not kept secure it would be a breach of GDPR which can lead to fines, loss of business license, and even legal prosecution. Although the project will not be made available to the general public, the security of the product has been considered throughout its development. As mentioned in the sprints, a hashing feature was attempted but was not deemed feasible as node.js could not be used and therefore local storage was the alternative used which has no real security features. If the application were to be developed further for a public audience, more security features, like hashing, would be created. With a web application like a password manager, failure to comply with GDPR would also raise some ethical concerns as user passwords can be used to access payment accounts and other sensitive information so for these reasons security is integral to this project.

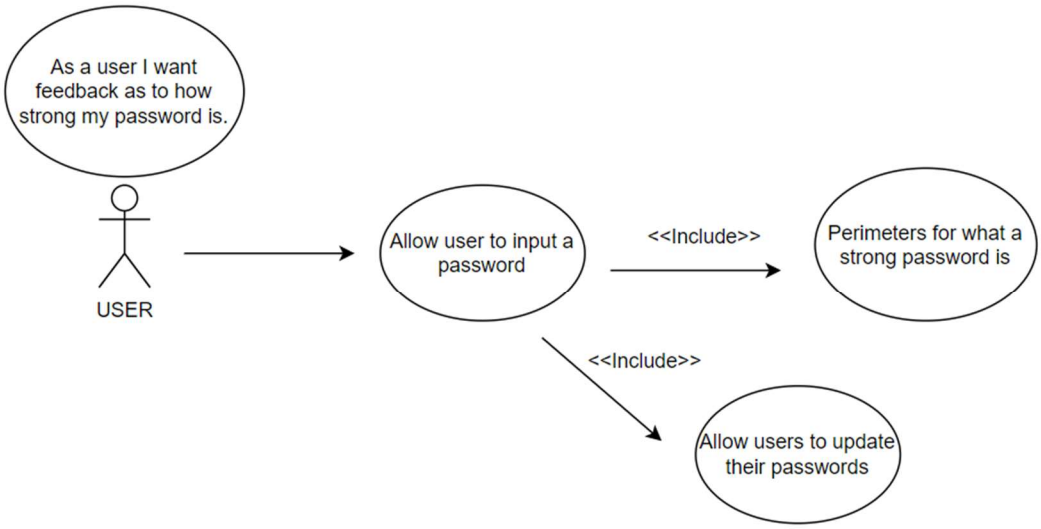
The design for this project was kept simple purposefully, a simple, quick, and responsive UI is needed as users, in my experience, want access to their passwords as easily and quickly as possible. The navigation bar is large with clear buttons and the navigation is not filled with animations that overall reduce the responsiveness of the application.

User stories and Associated Use Case Scenarios

The user story diagrams below show the main functional requirements the user can expect in the final product, if these features were not present in the final product, the project would not be considered complete.







Name	Add a new web account
Short Description	The user can add a new web account, this consists of a username, password and website name.
Precondition	User has made a master account
Post Condition	Password is added to storage to be viewed and used later.
Error Situations	The user leaves an input field blank and the user does not use a password that meets the security requirements.
System state in the event of an error	Passwords must contain at least 8 characters, at least one upper and lower case, at least one special character and at least one number
Actors	User
Triggers	User wishes to add a new web account for storage.
Standard Process	1. User creates an account. 2. User logs into their account 3. User navigates to the submit new password section 4. User enters the valid information 5. User presses the submit button
Alternative Process	6' Error displays 7' User retries

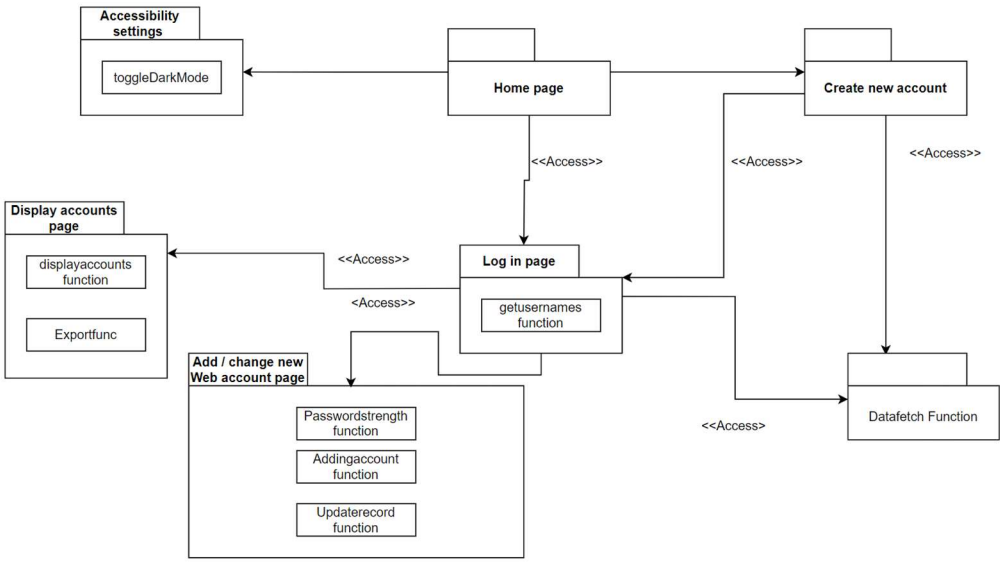
Name	Make a new Master Account
Short Description	Make a new account to tie the passwords to the user
Precondition	User has opened the site
Post Condition	New user account is created
Error Situations	If the username is the same as one already used, the system will alert the user.
System state in the event of an error	User already exists
Actors	New User
Triggers	New user needs to make an account.
Standard Process	1. User opens site 2. User navigates to the account creation tab 3. User enters a username 4. User enters a password 5. User presses submit.
Alternative Process	6' Account is not created 7' User tries again

Name	Update web account information
Short Description	Update an already existing web account with new details

Precondition	User has an account with a web account already submitted
Post Condition	Web account details are updated
Error Situations	The user enters a website that does not already exist on their account
System state in the event of an error	Site information does not update
Actors	Existing User
Triggers	User wishes to update their username and password for a web account
Standard Process	1. User creates an account. 2. User logs into their account 3. User navigates to the submit new password section 4. User enters the valid information 5. User presses the update button
Alternative Process	6' Account is not updated 7' User tries again

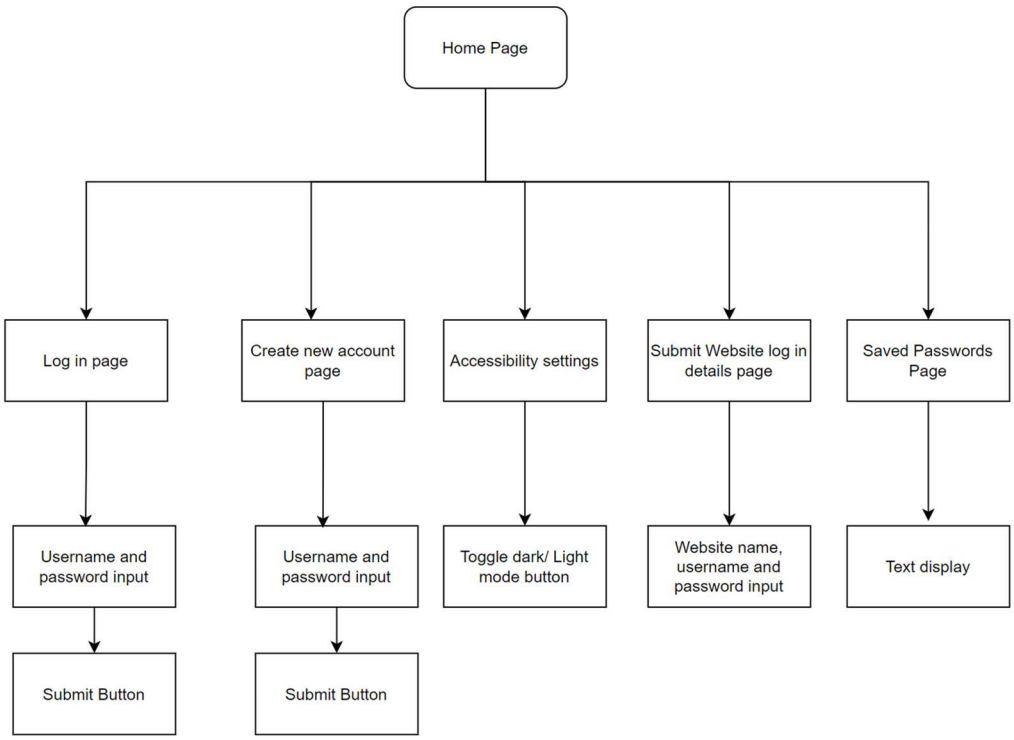
Architecture

The diagram below is a Package diagram that shows how the site interacts and shares information with its components, giving us an idea of how the logic of the site works.



Sitemap

The diagram below is a sitemap, it shows each “page” of the application and what it contains.



Noted issues and constraints

This project went smoothly overall but it was not without challenges, the biggest challenge was learning 3 new programming languages, although HTML and CSS are quite straightforward at this level, JavaScript proved a challenge; It took me longer than I would have liked to begin developing the web application as I spent a large amount of time learning and understanding JavaScript and how to apply it to my project. Understanding and deconstructing a programming language is one thing but applying it to your own project with no template is another, this caused a significant delay in development but I am certain the challenge was overcome and I am extremely happy with the final product.

Some of the programming was more difficult than others, specifically the JSON file functionality, trying to find a workaround to not using Node.js took a significant amount of time and even when I had an idea of how the logic would work, it took me a while to figure out how to implement it into my project. A large amount of the information online about storing and accessing information for a website use node.js so It took a long time of asking lecturers and searching the DLE to find information I could use in my project.

During the development of the project, my time management skills were tested, I found it easier to continue learning new programming techniques instead of applying what I knew to

the project. If I were to do a project like this again, I would spend more time making small incremental progress rather than developing knowledge and applying it in bulk in order to make the workload more manageable and the project less daunting to work on.

Managing my time with the other projects this year proved harder than I thought, making sure I left enough time to develop my knowledge for this module whilst doing the other coursework was a challenge, especially when the deadline for this project seemed so far away. To combat this, I found going to the lab sessions when I could, even for just the first two hours helped me keep on track and keep developing my knowledge. I also found that setting myself a time of day to work as the deadline approached helped me progress at a faster pace, spending some of the mornings understanding what it is I need to do and then adding what I can to the project in the afternoon helped me make small progress rather than no progress at all.

Overall, I am very happy with the project but ideally, I would have implemented a hashing system for security however due to the limitations of the project and what we could use it would have not been completely functional especially as a local storage server was needed instead of node.js. If I were to develop this project for a public audience, I would spend more time developing the security and ensuring there are no GDPR breaches.

Github repo link

https://github.com/Harvey-Moth/Comp-1004-Project/tree/main

References