# TreeSafe Final Report

TREESAFE

PASSWORDS MADE EASY

Team 3

By Caleb Bowden U2166829
Mohammad afaq U2162959
Hassan Ali U2078608
Zak Valli U2162751
Sebastian Astalos U2154374
Omar Elgabbas U2080326

# Table of Contents

# Introduction

1. In today’s world, there are a vast number of online accounts that individuals and organisations use. Subsequently there is a need for an effective password manager. 72% of individuals and organisations use the same password for multiple accounts, this puts organisations at risk of a cyberattack. Here at TreeSafe the team has developed a solution to address the problem of poor passwords & password management by creating a user-friendly application that manages and organises account details for you. Additionally, customers can customise parameters to create unique passwords. All data that is stored within TreeSafe is encrypted using the AES-256 algorithm which is the industry standard for data encryption. On top of the encryption algorithm passwords are salted and hashed. Encryption keys that are used to decrypt your passwords database are stored locally; the password database is also stored locally which means if TreeSafe gets hacked your passwords will not get compromised.

# Merits and limitations of the technology

## Merits of the technology

1. Our password manager has been designed to meet our strict requirements, one of our primary objectives was to implement a highly effective encryption algorithm that could securely encrypt the contents of a text file, ensuring that only the authorised key holder could access it. Our development team accomplished this goal by employing the highly respected AES-256 encryption algorithm, which is widely regarded as computationally secure. By utilising this encryption standard, our password manager ensures that your sensitive data is protected with the highest level of security possible. However the implementation of this algorithm was very hard as our development team was new to C# It took over a month to implement but it was a strict requirement for the project. In the end the algorithm was implemented.

2. In addition to prioritising encryption, our team also placed a strong emphasis on creating a user-friendly interface for our password manager. Our UI design team excelled at crafting a minimalist and intuitive interface that caters to the needs of our users. Our main objective was to create a simple and easy-to-navigate UI that would be accessible to individuals from various backgrounds, enabling them to enhance their online security with ease. By streamlining the user experience and minimising clutter, our password manager ensures that users can quickly and efficiently access their encrypted data, providing an added layer of protection for their sensitive information.

## Limitations of the technology

1. While our technology has proven to be effective in safeguarding user data, it does have a few limitations that we are working diligently to address. Our development team recently identified a major security vulnerability that could potentially allow an attacker to steal the entire database in its unencrypted form while a user is still logged in. This is due to TreeSafe's decryption process, which temporarily stores an unencrypted copy of the database file in %appdata% to facilitate data display on our GUI. While this copied version is removed when the user logs out, it remains vulnerable to exploitation as long as the user is logged in. If a system is infected with malware, an attacker could potentially exploit this vulnerability by using the FTP protocol to steal the raw unencrypted data. Although the development team has devised a solution to address this issue, it will require a significant amount of time and resources to rewrite the data reading process, and we are committed to ensuring that our technology remains secure and effective in protecting user data.

2. The team also did not incorporate the capability to utilise the FTP protocol to enable users to create a self-hosted server to store all of their passwords. Unfortunately, due to unforeseen challenges that emerged during the development process, there was not enough time to complete this feature. Similarly, we were unable to provide compatibility with iOS, Android, macOS, and Linux devices, which is another limitation we recognize.

However, we are committed to continually improving and expanding our technology, and we are exploring options to address these limitations in the future, with a goal of providing the most comprehensive and secure password management solution possible.

## Development and Project Planning

1. In the Agile method, the tasks are split into separate portions, so specific features can be developed and worked on. It’s an iterative approach and a build is created after each iteration so the developments can be tested and worked upon. These iterations are also known as ‘sprints’. This means that the features that are currently being created can be refined and polished as we test and discover improvements in its functionality and design. This is good as the final build had minimal bugs and issues, while also containing the features that were required for our target audience. During this process, we learned how to document and communicate new and existing changes.

2. Features and content are added as time goes on and more builds are created; the final build has the requirements that were set for the game. Implementing new features in each build also allows us to recognize the impact of the added features and understand their value in the game. This is beneficial as, at this stage, we can decide to make any changes with the feature or balances if needed. From this, we learned how to repetitively test software, while creating and receiving immediate useful feedback so that further improvements can be made soon. There is a focus on quick responses to change and requests; requirements can be added and amended after starting the project. This is great because constant testing and builds can help shape the final build in the way we intended while also giving us the flexibility to discuss and make amendments to the game so that it can better align with the end requirements and what our target audience is wanting.

3. Using the Agile method really helped our team organise and work towards features that we found best for our requirements as our development periods passed. Finding the most suitable method was essential for us because we were able to organise and form the best plans for our end goal using the iterative approach of the agile method. Having worked on the Waterfall Methodology in the past, this experience definitely helped us, as a team, explore and utilise more effective methodologies based on the project we are working on. The team received their tasks between the development periods and were able to work on them effectively, which improved organisation. A positive to this is that this method also ensures that our requirements and features are feasible and something that we will be able to implement because splitting the tasks amongst the team allows us to visualise the workload that each team member would receive. This was a learning experience as we all worked on distributing the work based on the skill sets that each of the team members had so that everyone could contribute, and not feel overwhelmed, while also creating the best final project possible.

4. Some downsides of using the agile method include how there were no distinct starting and endpoints for each sprint, which meant measuring progress wasn’t as simple. Due to

this, the tasks are not as stable so sprints may not proceed as smoothly as they could have done with other development methodologies.

## Research

1. When researching different password managers we looked at KeePass and LastPass. KeePass and LastPass are two popular password managers that are used to securely store and manage passwords. While both programs have the same basic purpose, there are some significant differences between them. KeePass has a simple and straightforward interface that is easy to use, while LastPass has a modern and user-friendly interface that makes it easy for users to navigate and manage their passwords. KeePass is a desktop-based password manager that requires installation on a device, while LastPass is a cloud-based password manager that can be accessed from any device with an internet connection. This makes LastPass more convenient for users who need to access their passwords on multiple devices.

2. With this research, it helps us identify the key features and functions that are essential for a password manager. It can guide us in deciding what features to include in the password manager and how to prioritise them. It can also help us understand the strengths and weaknesses of existing password managers, such as KeePass and LastPass. This provides us with valuable insights into what works well and what needs improvement, and can inform the design and development of our own password manager.

3. Overall, this research into different password managers, can help us make informed decisions about the design, development, and implementation of our password manager that meets the needs of users and provides a secure and reliable solution for managing passwords.

## Issues encountered

1. During the development process, the team encountered several issues that required innovative solutions. One of the most significant issues was the implementation of the encryption algorithm. To resolve this, the team decided to store encrypted and unencrypted versions of the data in separate text files. When a user logs into the system, their password key decrypts the data from the encrypted file, creates a temporary file in %appdata%, and reads the unencrypted data onto the table. When a user adds new data, it is saved in the unencrypted folder, and when they refresh the table, the data is encrypted and transferred to the encrypted file. Once the user logs out, the temporary file is deleted to ensure that the data remains secure.

2. Another challenge that the team faced was the use of global variables. As the team was not familiar with C#, they struggled to set up global variables correctly. After conducting extensive research, the team discovered that they could use a "get-set" method in C# to retrieve and modify the variable. To communicate with other pages in the software, the variable needed to be set as public.

3. Effective communication was also a significant hurdle for the team. While using Discord for text communication, team members found it challenging to debug code and explain their ideas clearly through text messages. This often resulted in confusion and delayed progress.

4. Finally, the team faced challenges with GitHub commits. As new users of the platform, the team struggled to understand how to commit their work without compromising the work of other team members. They had to ensure that they were coding on the latest release and not overwrite another team member's work. While there was a solution using branches, the team found it challenging to implement.

## Target Audience

1. TreeSafe offers a unique and highly secure password management solution that is tailored to meet the needs of both individuals and organisations. Unlike other password management programs, TreeSafe provides a customizable password generator that creates strong and unique passwords for each online account. Additionally, TreeSafe's easy-to-use interface allows users to quickly and securely store their login credentials and access them when needed, without the need to remember each password. Organisations can benefit from TreeSafe's advanced features, such as team sharing and access control, which allow team members to securely share passwords while maintaining control over who has access to sensitive information. With its strong encryption and intuitive design, TreeSafe is the ideal choice for organisations looking to improve their password management systems and protect sensitive data from unauthorised access.

## Team Roles

1. The team leader for this project is Mohammed Afaq, as the team leader he is responsible for coordination and managing the project. He is to ensure that the team stays on track with the workload and meets deadlines. Also, Mohammed is responsible for designing the team logo.

2. The developers for this project are Mohammed Afaq, Caleb Bowden and Sebastian Astalos. Their responsibilities include writing in the report and implementing any software solutions. They will work together and be responsible for troubleshooting and resolving any issues that occur during the development of the product.

3. The researchers for the project are Zak Valli and Sebastian Astalos, they are responsible for researching necessary information to support the proposal and development of our password manager. They will work together to gather and analyse information. Zak and Sebastian are responsible for the report being accurate and up to date.

4. Designers for this project include Caleb Bowen and Omar Elgabbas. They will be responsible for designing the password manager. They will work together to create graphics such as frameworks to help visualise TreeSafe. They will also be responsible in helping to complete the proposal.

5. The minute keeper in this team is Hassan Ali. He is responsible for keeping track of the team's progress during meetings and documenting any decisions. He also is needed to help add information to the proposal.

6. While each team member has been assigned specific roles and responsibilities, the team has agreed to work collaboratively to complete the project. It was decided that everyone will help each other out within the team so all work is completed within the time that was given. The team would all help out within the proposal documentation.

## Constraints and Limitations

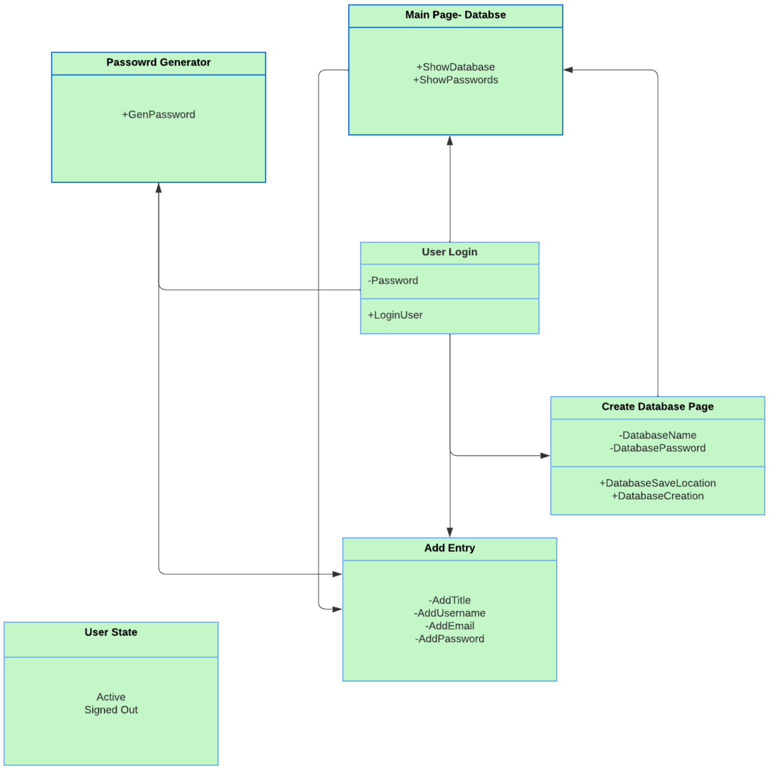
1. The project team faces several limitations that could impact the development of the platform. One of the primary limitations is the team's limited resources and budget, which means that they will have to work within these constraints to ensure that the platform is developed efficiently and within a reasonable timeframe. This will require careful consideration of the cost of hardware and software resources, as well as the time and effort required to develop the platform.

2. Another significant constraint of the project is its complexity, particularly in regards to developing a secure password manager. The team will need to conduct extensive research to gain the necessary expertise in encryption, storing, salting, and hashing

information. Due to the team's lack of experience in this field, this research may take up a considerable amount of time and effort. However, it is essential to prioritise security to ensure that the platform meets the necessary standards and safeguards user data.

## Project planning

### UML Diagram

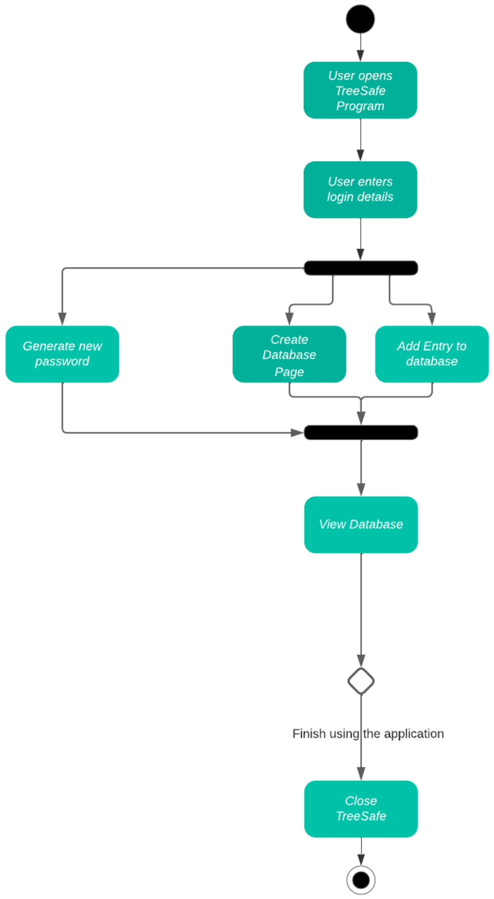


1. This UML diagram contains different classes which show the overall system of our password manager. We first start with the “User Login” class which contains a password attribute, this class is used so only the user can access their database of passwords.

2. We have another class called “Password Manager”, this class is responsible for generating new passwords. This class contains methods for setting the length and complexity the user wants generated. The “Main Class” is used to represent the database page where the user is able to see all their passwords in a certain database in one area with various information for it such as the title, email address, username and password for each entry.

3. The UML diagram includes different relationships between the various classes as it provides a visual representation of the systems architecture

## Activity Diagram



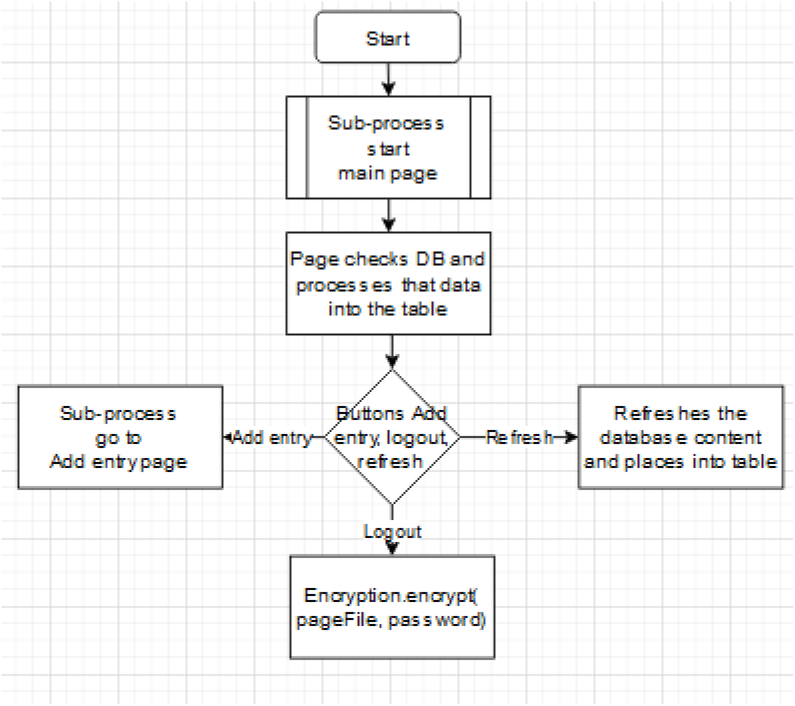
1. The activity diagram shows the flow of activities that occur when a user uses TreeSafe. The first state of TreeSafe is the login area where the user will have to use their credentials to access their saved passwords. From here they can access many other areas of TreeSafe, such as to generate a new password, create a database page to store passwords, or to create an entry for the database. From here the user can view the database page, the app changes to the ‘View Database’ state, here the user is able to see a list of saved passwords.Overall, the activity diagram shows the different states the user can encounter when using TreeSafe.

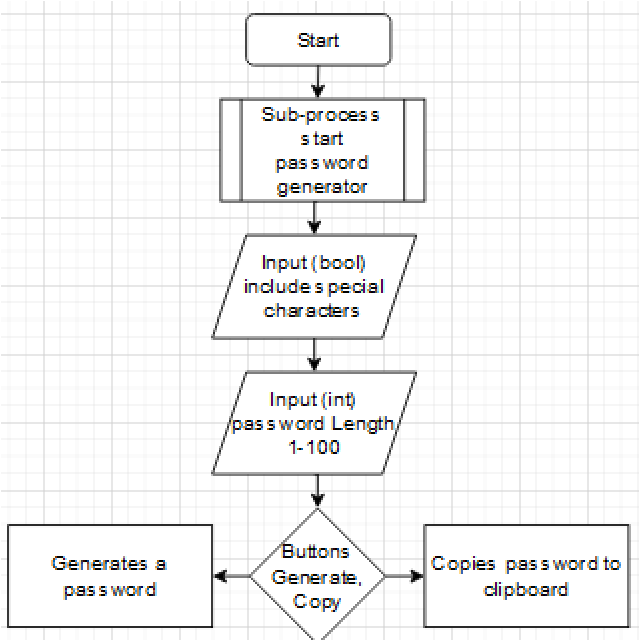
## Flowchart

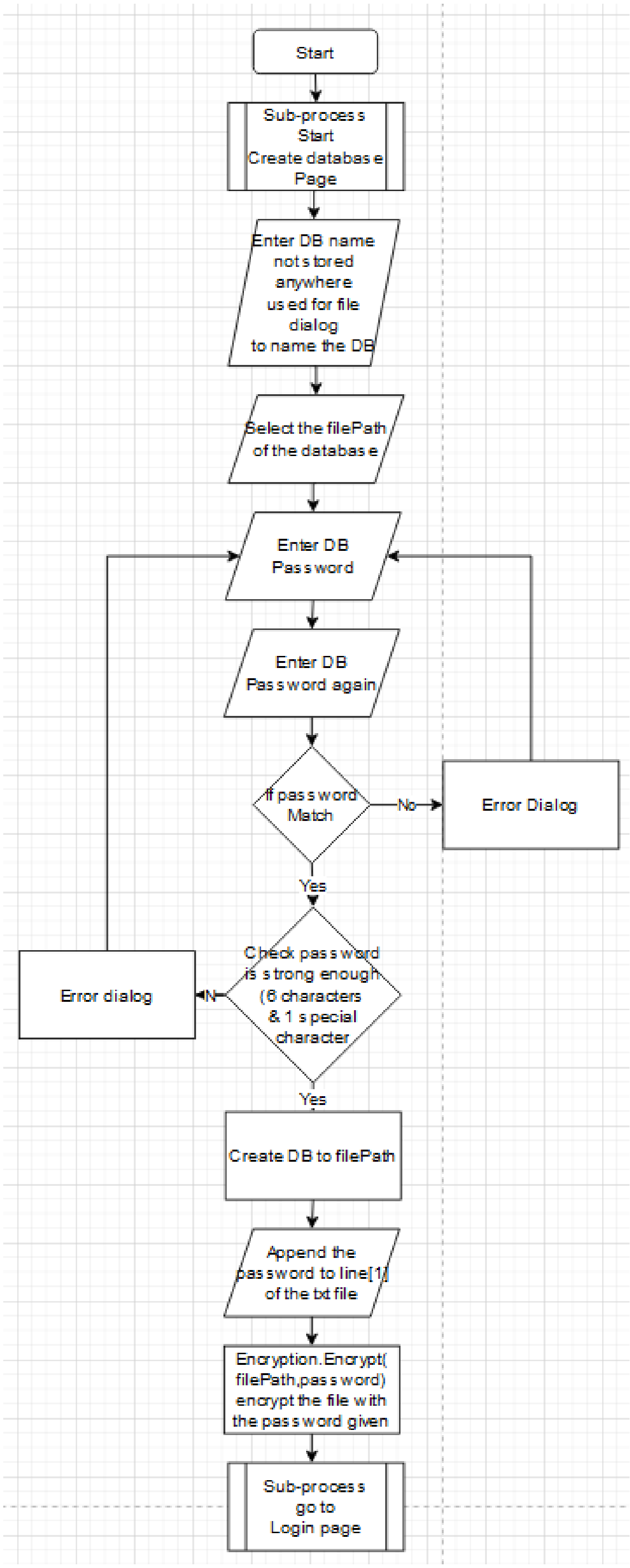
1. Through our experience, we have discovered that creating a flowchart can offer tremendous benefits for various reasons. Primarily, a flowchart provides a clear and concise visual representation of the different steps involved in the application, enabling everyone to comprehend the entire process and identify potential areas for improvement.

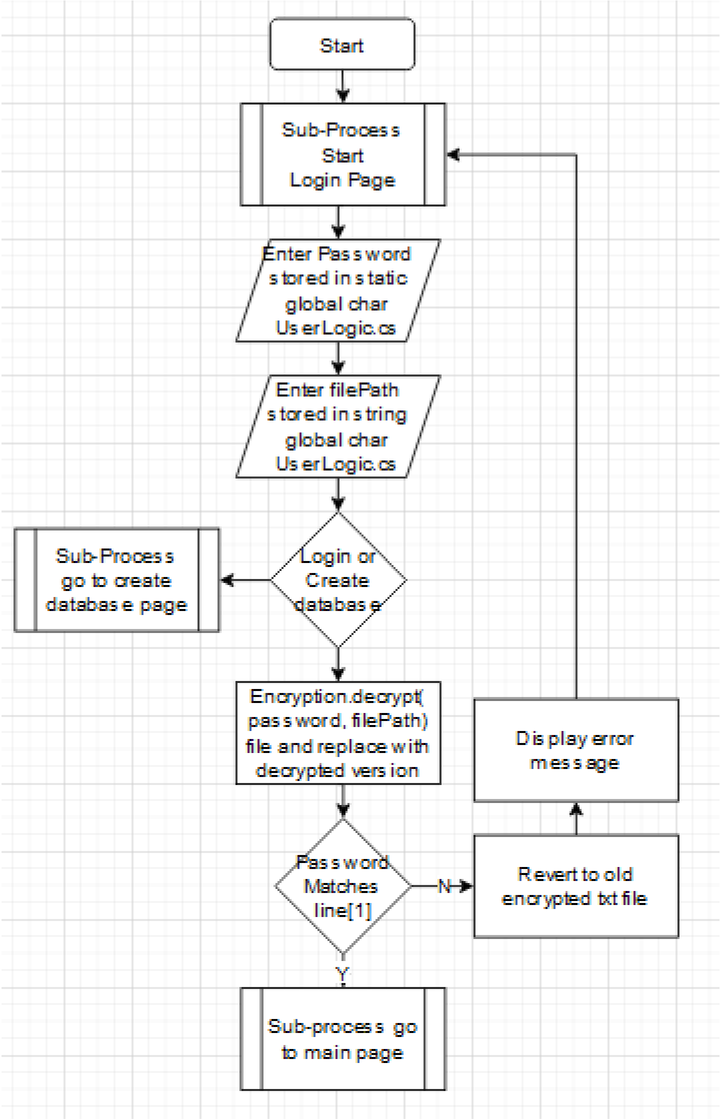
2. Moreover, crafting a flowchart helps to clarify the intricate structure of the password manager, rendering it much easier to follow. By dissecting the various components of the application into distinctive steps and linking them in a logical sequence, we can gain a panoramic understanding of how everything fits together.

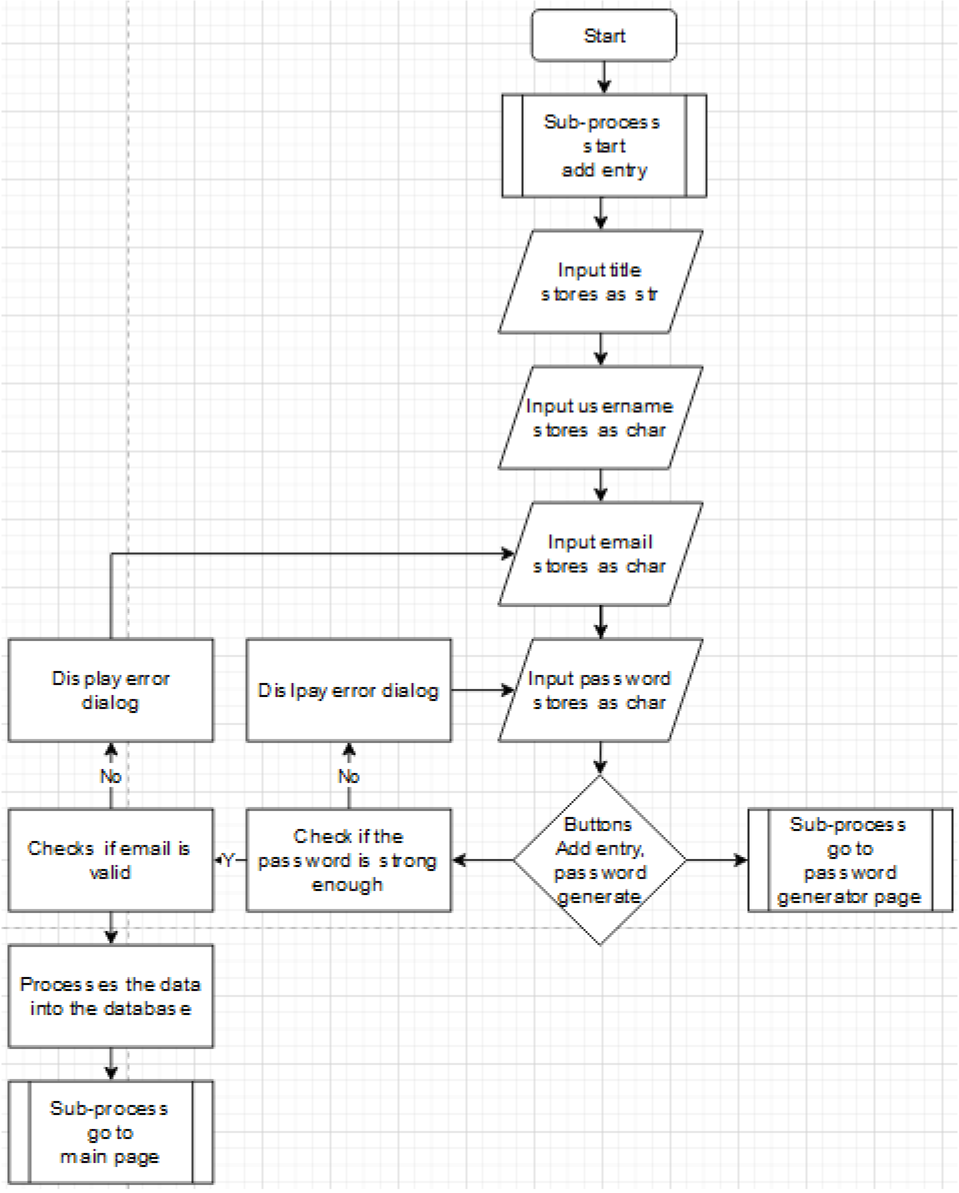
3. When we generate a flowchart, we have also observed that it aids in identifying potential errors or issues in the password manager. By visually illustrating the flow of the application, we can promptly detect and address any potential issues before they escalate into actual problems.


Start
Sub-process start main page
Page checks DB and processes that data into the table
Buttons Add entry, logout, refresh
Add entry
Sub-process go to Add entry page
Refresh
Refreshes the database content and places into table
Logout
Encryption.encrypt( pageFile, password)


Start
Sub-process start password generator
Input (bool) include special characters
Input (int) password Length 1-100
Buttons Generate, Copy
Generates a password
Copies password to clipboard


Start
Sub-process Start Create database Page
Enter DB name not stored anywhere used for file dialog to name the DB
Select the filePath of the database
Enter DB Password
Enter DB Password again
If password Match
No
Error Dialog
Yes
Check password is strong enough (6 characters & 1 special character
N
Error dialog
Yes
Create DB to filePath
Append the password to line[1] of the txt file
Encryption.Encrypt( filePath,password) encrypt the file with the password given
Sub-process go to Login page


Start
Sub-Process Start Login Page
Enter Password stored in static global char UserLogic.cs
Enter filePath stored in string global char UserLogic.cs
Login or Create database
Sub-Process go to create database page
Encryption.decrypt( password, filePath) file and replace with decrypted version
Password Matches line[1]
N
Revert to old encrypted txt file
Display error message
Y
Sub-process go to main page

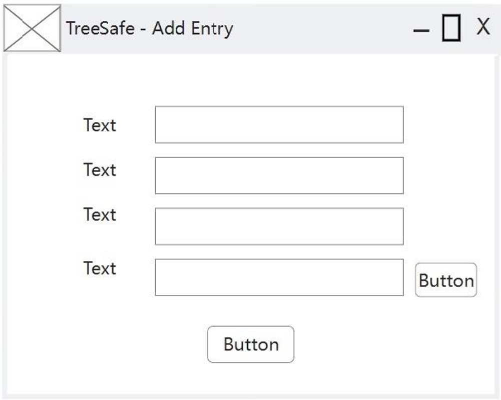

Start
Sub-process
start
add entry
Input title
stores as str
Input username
stores as char
Input email
stores as char
Input password
stores as char
Display error
dialog
Dislpay error dialog
No
No
Checks if email is
valid
Y
Check if the
password is strong
enough
Buttons
Add entry,
password
generate
Sub-process
go to
password
generator page
Processes the data
into the database
Sub-process
go to
main page

## Wireframes

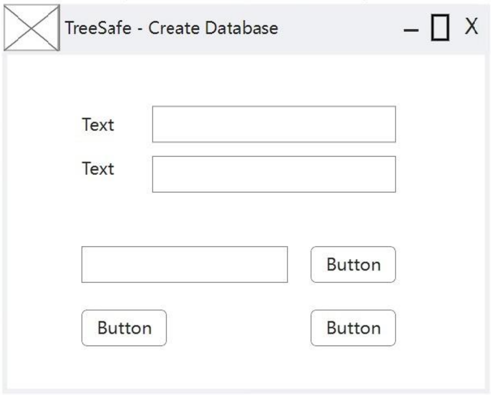
1. In the wireframes we did 3 designs Basic, Intermediate and advanced. We will start with showing the basics then the intermediate and finish by the advanced and explain each single one of them. Firstly a wireframe is basically a design that we make before the program itself or the website which shows how the application is going to look like, it is like taking a screenshot of the application before even developing it to see how it would look like and if it is going to need any changes so we can do it while developing it. Our program is a desktop application which contains a total of 5 pages.

### Basic Wireframes

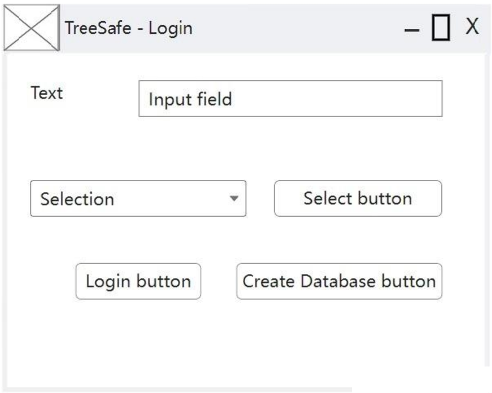
Add entry Page



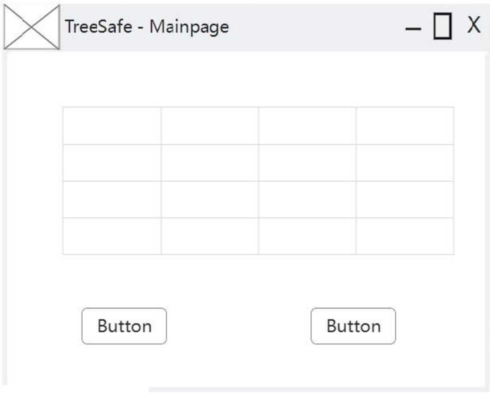
Create database Page



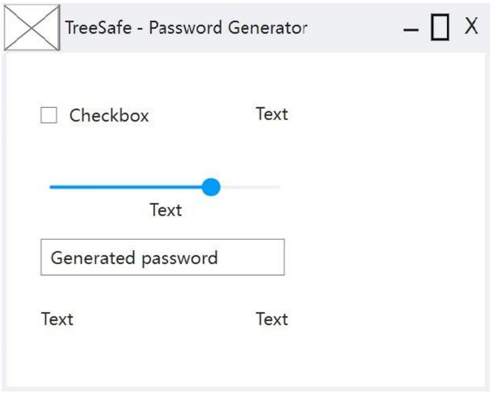
Login Page



Main page



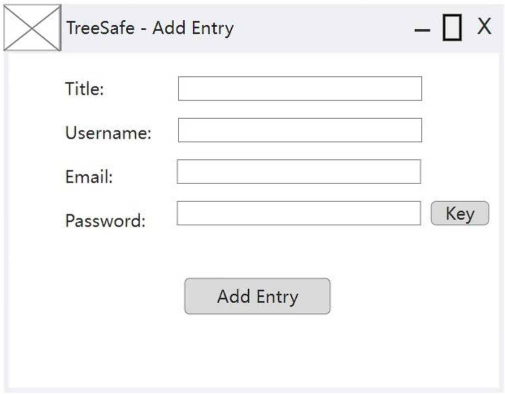
Password generator page



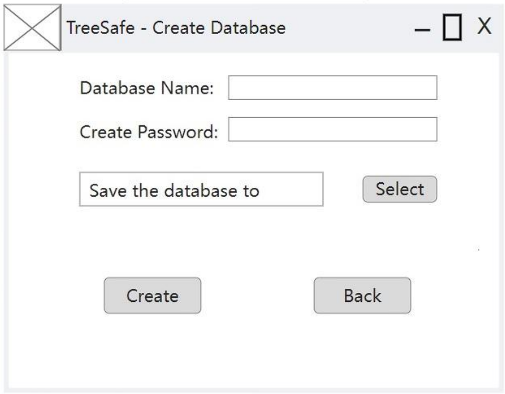
2. These were the basic wireframes, the square with an x inside should be the logo, it does not contain any real thing, just a real basic look of the program to see where we are and then start developing the application. It does not contain any information or test cases.

## Intermediate Wireframes

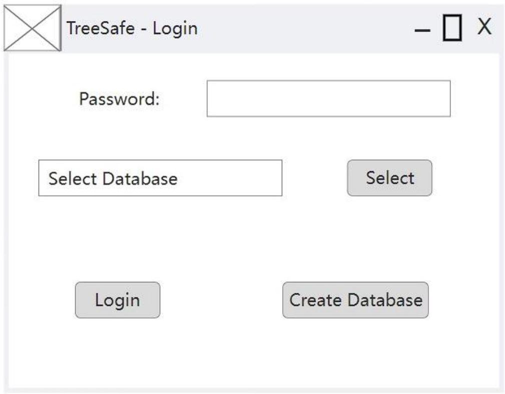
Add Entry Page



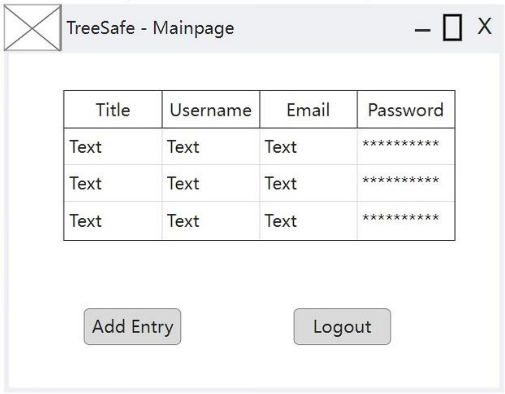
Create Database Page



Login Page

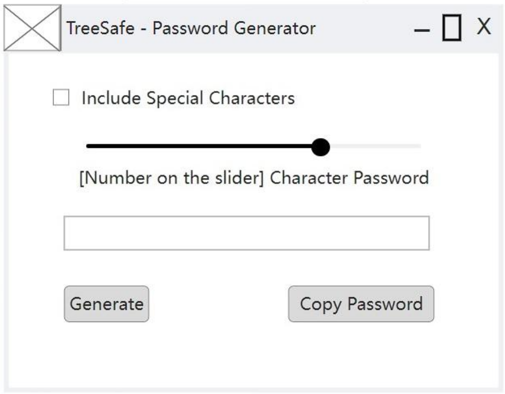


Mainpage



| Title | Username | Email | Password |
|---|---|---|---|
| Text | Text | Text | ********** |
| Text | Text | Text | ********** |
| Text | Text | Text | ********** |

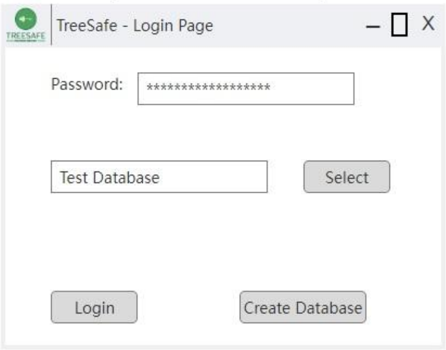
Password Generator Page


TreeSafe - Password Generator
– ☐ X
□ Include Special Characters
[Number on the slider] Character Password
Generate
Copy Password

3. These are the intermediate ones, with this one we show the real actual button for example how they would look like, how the data in the table would appear and how each and every single aspect of the application look and work together. But it does not yet have real data or test cases, and it does not contain the logo.

<u>Advanced Wireframes</u>

Login Page

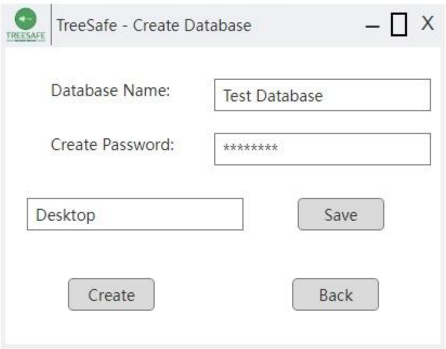

TreeSafe - Login Page

Password: ******************

Test Database

When you first start the application it shows you the login page, Where you have a drop down menu to select which database you want to use now and type the database password so that you can access it. With a Create Database option if you want to create a new database.

Create Database Page


TreeSafe - Create Database

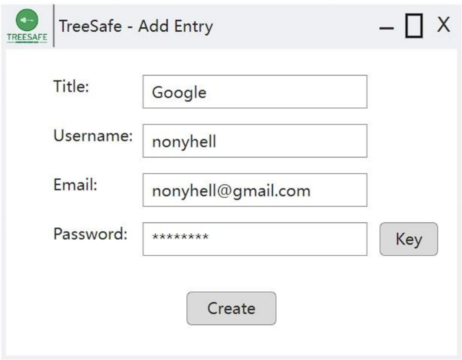
Database Name: Test Database

Create Password: ********

Desktop

Here is the Create Database window where you first type the database name that you like to choose and create a password to make your data secure, then a drop down menu to choose where you want to save your database and a save button after you choose. A create button in case you have made everything you need or a back button in case you do not want to make a new database anymore which would take you back to the login page.

Add Entry Page


TreeSafe - Add Entry

Title: Google

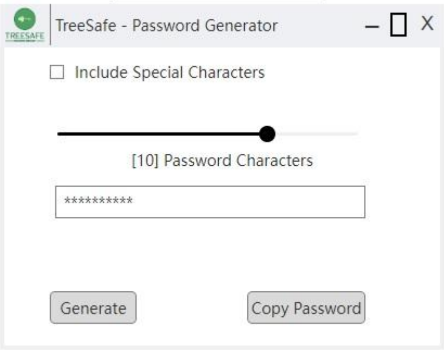
Username: nonyhell

Email: nonyhell@gmail.com

Password: ********

When you go to the Add Entry window it gives you 4 input fields where you type the title, username, and Email. Then when it comes for the password you can use a password that you already have or press the key button which takes you to the next window which generates you a password. With a create button which adds that entry to the database.

Password Generator Page



TreeSafe - Password Generator

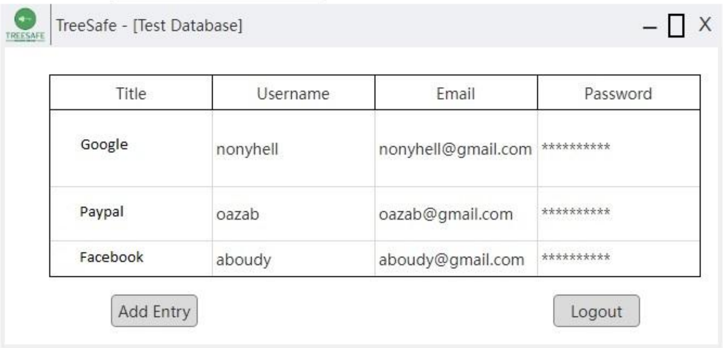
☐ Include Special Characters

[10] Password Characters

**********

Here is the password generator. The first option you have here is a checkbox to check whether you want to include special characters in the generated password or not. Then we made a slider that you use to determine the number of characters in the password to make it easier for the user with an output below showing what the current characters are now. An output which shows you the generated password with a copy password button so that you can use it straight away and a generate button which each time you press on it, it generates a new password.

Main Page



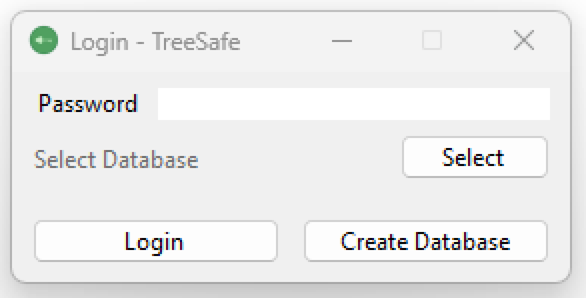
TreeSafe - [Test Database]

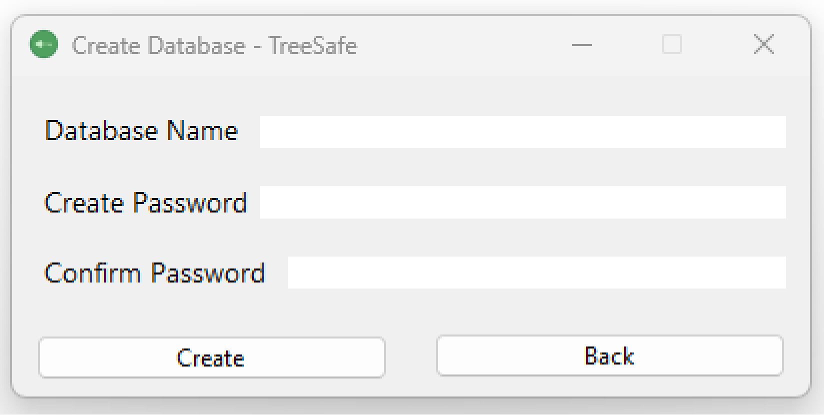
| Title | Username | Email | Password |
|---|---|---|---|
| Google | nonyhell | nonyhell@gmail.com | ********** |
| Paypal | oazab | oazab@gmail.com | ********** |
| Facebook | aboudy | aboudy@gmail.com | ********** |

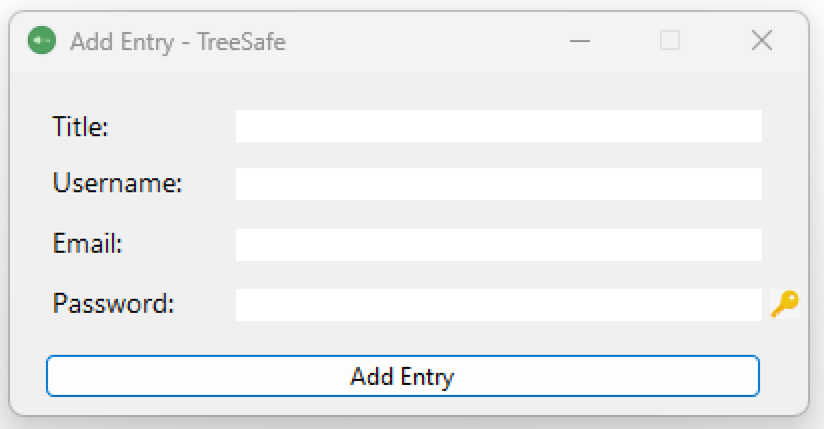
1. Here is the mainpage whether you used an already existing database or just created a new one. It consists of a table that represents the data in your database, The data contains the title which is the name of the website or application you use, Then your username in that website or application, Then your Email that use for signing in because you may have multiple Emails so that way it is easy to track which is which and finally the password which stores each single special password that you can copy and use it as simple as that. With the header containing the logo and the name of the database which in our test case is called (Test Database). As well as a button for adding an existing entry which takes you to the next window, and a logout button which you use after you finish.

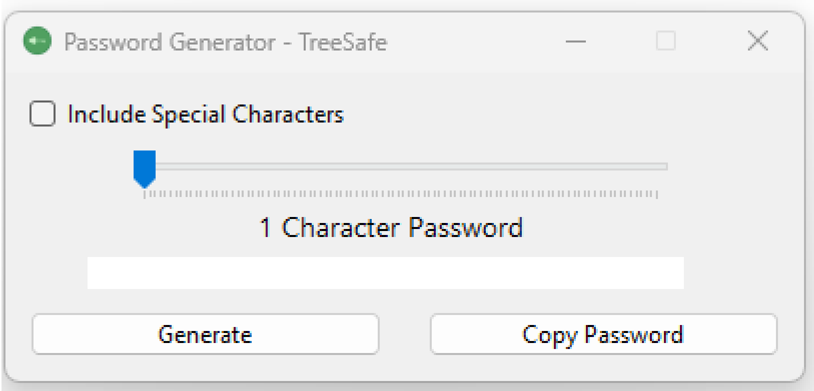
2. After you finish everything you need with the application you can just close it using the x button in the top right corner, or go back to the mainpage to logout from your database and then start working on another database or even create a new one. And that was a brief description of how our application would look like and work with screenshots and test cases to make it as real as possible.

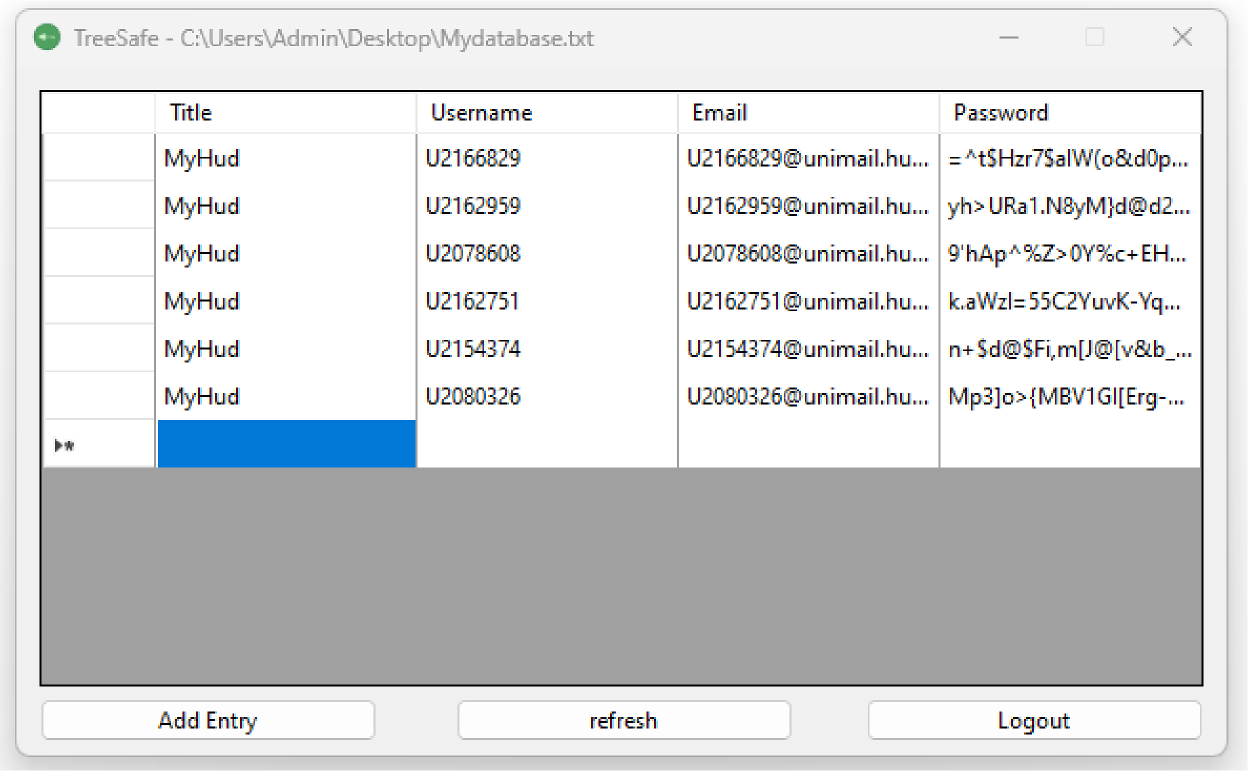
# Final Product











TreeSafe - C:\Users\Admin\Desktop\Mydatabase.txt

| | Title | Username | Email | Password |
|---|---|---|---|---|
| | MyHud | U2166829 | U2166829@unimail.hu... | =^t$Hzr7$alW(o&d0p... |
| | MyHud | U2162959 | U2162959@unimail.hu... | yh>URa1.N8yM}d@d2... |
| | MyHud | U2078608 | U2078608@unimail.hu... | 9'hAp^%Z>0Y%c+EH... |
| | MyHud | U2162751 | U2162751@unimail.hu... | k.aWzl=55C2YuvK-Yq... |
| | MyHud | U2154374 | U2154374@unimail.hu... | n+$d@$Fi,m[J@[v&b_... |
| | MyHud | U2080326 | U2080326@unimail.hu... | Mp3]o>{MBV1Gl[Erg-... |
| ▸* | | | | |

# Conclusion

1. In conclusion, our report emphasises the necessity for a reliable password manager, given the plethora of online accounts utilised by both individuals and organisations. To address this issue, TreeSafe has developed a user-friendly password manager that not only organises account details, but also facilitates the creation of unique passwords. TreeSafe's use of the industry-standard AES-256 algorithm ensures that all stored data is encrypted, and the password database is stored locally, preventing potential compromise in the event of a security breach.

2. Despite its many benefits, TreeSafe's technology has certain limitations. For example, a recently discovered vulnerability could potentially allow an attacker to obtain the unencrypted database in its entirety. Additionally, TreeSafe does not currently support the use of the FTP protocol, preventing users from creating a self-hosted server to store their passwords. However, the team behind TreeSafe remains committed to addressing these issues and improving their technology to better meet the needs of their users.

3. The development and project planning of TreeSafe utilised the Agile method, enabling the team to focus on features that best served their requirements. The iterative approach of the agile method allowed the team to identify and address issues quickly, resulting in builds with minimal bugs and other problems. By remaining responsive to change and user feedback, TreeSafe has created a technology that offers a comprehensive and secure password management solution, with ongoing efforts to improve and expand its capabilities.In conclusion the team has successfully developed software that meets the target requirements set by us.

# Meetings

The meetings were decided to be held on every consecutive Wednesday at 3pm to last for a period of 45 minutes.

Attendees (6):

Name: Caleb

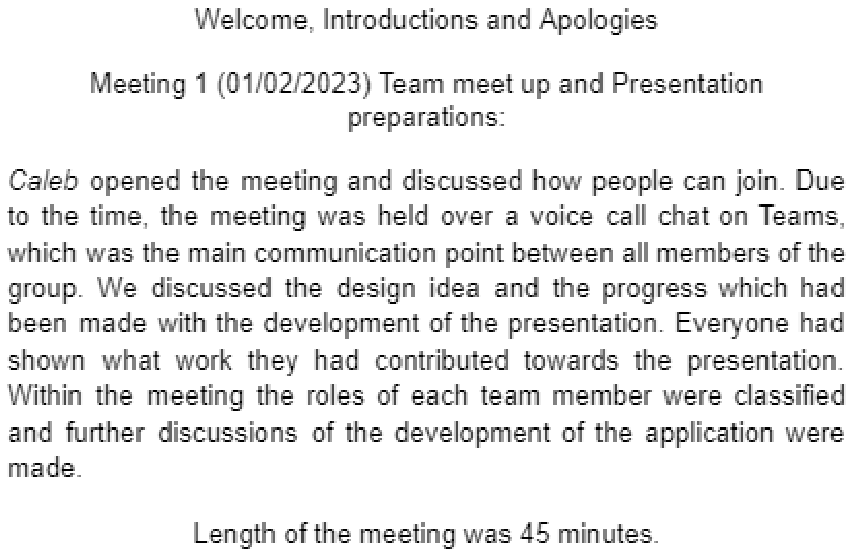
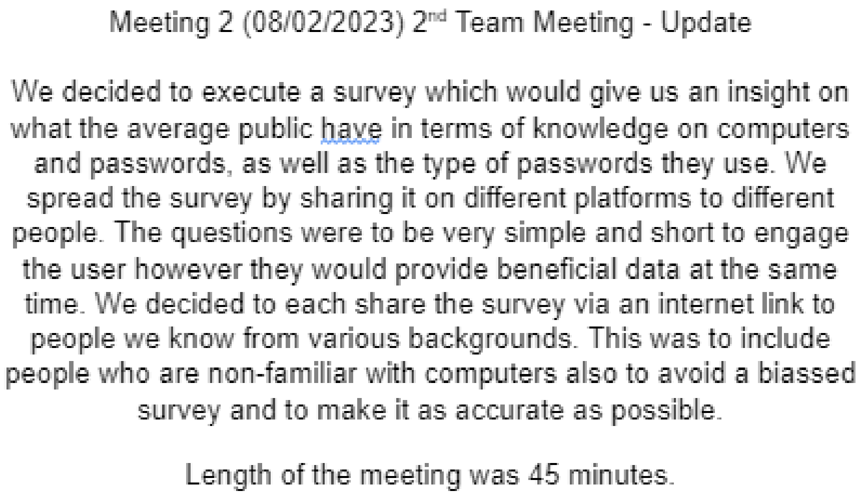
Name: Hassan

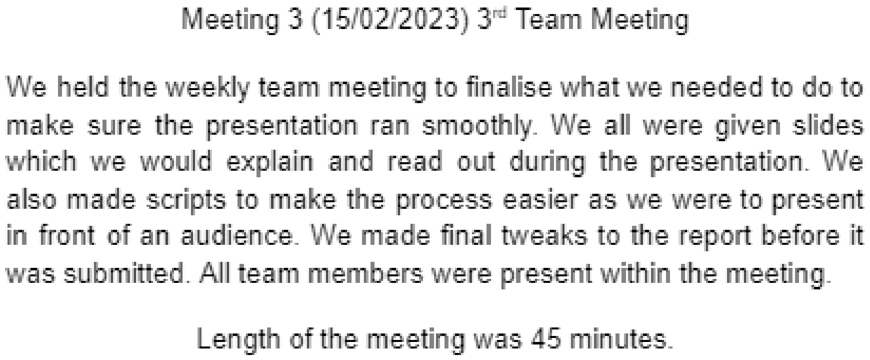
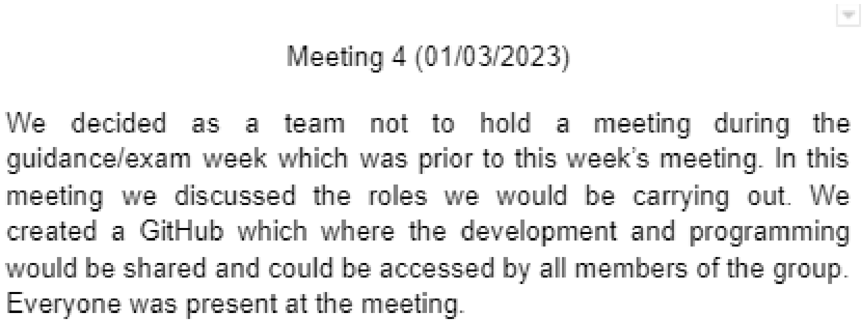
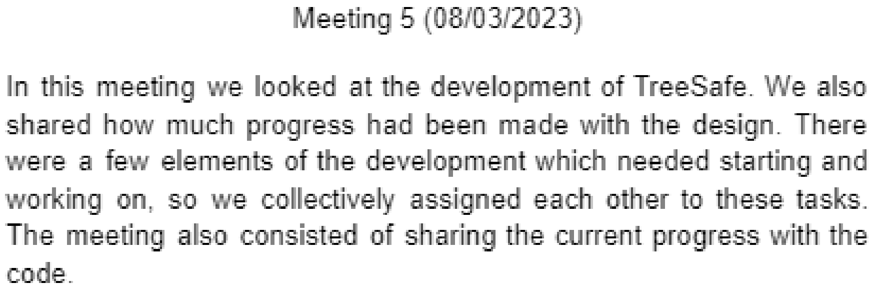
Name: Afaq

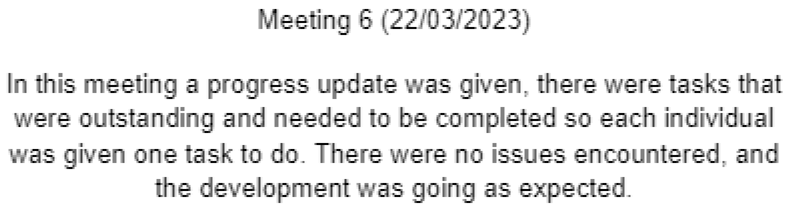
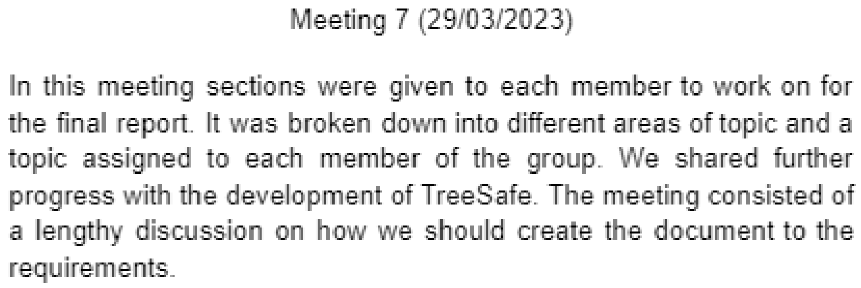
Name: Seb

Name: Zak

Name: Omar

| No | Item | Action |
|---|---|---|
| 1.0 | <br>Welcome, Introductions and Apologies<br>Meeting 1 (01/02/2023) Team meet up and Presentation preparations:<br>*Caleb* opened the meeting and discussed how people can join. Due to the time, the meeting was held over a voice call chat on Teams, which was the main communication point between all members of the group. We discussed the design idea and the progress which had been made with the development of the presentation. Everyone had shown what work they had contributed towards the presentation. Within the meeting the roles of each team member were classified and further discussions of the development of the application were made.<br>Length of the meeting was 45 minutes. | All to note |
| 2.0 | <br>Meeting 2 (08/02/2023) 2nd Team Meeting - Update<br>We decided to execute a survey which would give us an insight on what the average public have in terms of knowledge on computers and passwords, as well as the type of passwords they use. We spread the survey by sharing it on different platforms to different people. The questions were to be very simple and short to engage the user however they would provide beneficial data at the same time. We decided to each share the survey via an internet link to people we know from various backgrounds. This was to include people who are non-familiar with computers also to avoid a biassed survey and to make it as accurate as possible.<br>Length of the meeting was 45 minutes. | |

| | | |
|---|---|---|
| 3.0 | <br>Meeting 3 (15/02/2023) 3rd Team Meeting<br>We held the weekly team meeting to finalise what we needed to do to make sure the presentation ran smoothly. We all were given slides which we would explain and read out during the presentation. We also made scripts to make the process easier as we were to present in front of an audience. We made final tweaks to the report before it was submitted. All team members were present within the meeting.<br>Length of the meeting was 45 minutes. | All to note |
| 4.0 | <br>Meeting 4 (01/03/2023)<br>We decided as a team not to hold a meeting during the guidance/exam week which was prior to this week's meeting. In this meeting we discussed the roles we would be carrying out. We created a GitHub which where the development and programming would be shared and could be accessed by all members of the group. Everyone was present at the meeting. | All to note<br>Action TG |
| 5.0 | <br>Meeting 5 (08/03/2023)<br>In this meeting we looked at the development of TreeSafe. We also shared how much progress had been made with the design. There were a few elements of the development which needed starting and working on, so we collectively assigned each other to these tasks. The meeting also consisted of sharing the current progress with the code. | |
| 6.0 | | |

| | | |
|---|---|---|
| | <br>Meeting 6 (22/03/2023)<br>In this meeting a progress update was given, there were tasks that were outstanding and needed to be completed so each individual was given one task to do. There were no issues encountered, and the development was going as expected. | |
| 7.0 | <br>Meeting 7 (29/03/2023)<br>In this meeting sections were given to each member to work on for the final report. It was broken down into different areas of topic and a topic assigned to each member of the group. We shared further progress with the development of TreeSafe. The meeting consisted of a lengthy discussion on how we should create the document to the requirements. | |