Software Requirements Specification

for

Cryptographically - Secure Password Manager

Version 1.0

Prepared by

Anushka Singh | 2021IMG-014
Avijeet Jain | 2021IMG-018
Yuvraj Kumar | 2021IMG-066

ABV-Indian Institute of Information Technology & Management, Gwalior

20th March 2022

Submitted in partial fulfillment
Of the requirements of
IMIT-2202 Software Engineering Lab

Table of Contents

1. Introduction

1.1. Purpose

The purpose of a cryptographically secured password manager is to provide a secure and convenient way for users to store and manage their passwords and other sensitive information. A password manager typically stores login credentials for various websites and applications, and can also store other types of sensitive information such as credit card numbers, social security numbers, and other personal information.

A cryptographically secured password manager uses strong encryption algorithms to protect the stored data, making it difficult for unauthorized users to access or steal the information. The encryption key used to secure the data is typically derived from a master password chosen by the user, which is itself stored using strong encryption techniques.

Using a password manager can also make it easier for users to create and manage complex and unique passwords for different accounts, reducing the risk of a password breach. With a password manager, users only need to remember one master password to access all of their stored credentials, rather than having to remember multiple passwords for different accounts.

Overall, a cryptographically secured password manager can help users protect their sensitive information and maintain better security practices.

The purpose of this document is to define the software requirements for a cryptographically-secure password manager. The password manager will provide a secure and easy-to-use platform for users to store and manage their passwords.

1.2. Scope

- The scope of a cryptographically secured password manager is primarily to securely store and manage user credentials and other sensitive information.
- A password manager typically allows users to create and store complex and unique passwords for different accounts, and can automatically fill in login forms on websites and applications. Some password managers also have features to generate strong passwords, detect weak passwords, and prompt users to update passwords periodically.
- Additionally, a password manager may offer additional security features, such as two-factor authentication or biometric authentication, to further protect the stored data.

- The scope of a password manager is limited to the storage and management of user credentials and sensitive information. It is not designed to provide protection against other types of attacks, such as malware or phishing attacks. Users should still take other security measures, such as keeping their devices and software up-to-date, avoiding suspicious links and downloads, and being cautious with their personal information.
- Overall, the scope of a cryptographically secured password manager is to provide a secure and convenient way for users to manage their passwords and other sensitive information, reducing the risk of password breaches and improving overall security practices.

1.3. Glossary

Term	Definition
Encryption	In cryptography, encryption is the process of encoding information. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext.
Decryption	The conversion of encrypted data into its original form is called Decryption. It is generally a reverse process of encryption. It decodes the encrypted information so that an authorized user can only decrypt the data because decryption requires a secret key or password.
Private Key	Private key is used for both encrypting and decrypting the sensitive data.
Public Key	Public key is used only for the purpose of encrypting the data in cryptography.
Rating	Measurement on a 10-point scale to rate the user experience with the organization.
Software Requirements Specification	A document that describes all the functions and constraints of a system.
Database	Collection of all the data monitored by this system.
API	Application Programming Interface
HTTPS	HyperText Transfer Protocol Secure

1.4. References

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.

1.5. Document Overview

The next chapter of this document gives an overview of the functionalities of the products. This report focuses on the informal as well as the technical requirements definition of this software and provides an overview of the capabilities of our solution. The system environment is covered in the first portion of this report, which is followed by the functional requirements specifications. Each use case's logical framework and in-depth explanation have been provided. User attributes are also researched.

The non-functional needs are discussed in the following section of this report, which is followed by the requirement specifications. The requirements specifications section contains specifics on the functionality of our product in technical terms that are intended mostly for developers. The external interfaces, which are a thorough description of all the inputs and outputs from the software system, are also reviewed in this paper.

2. Overall Description

2.1. System Environment

The application will be a web app from the user interface's perspective, allowing users to interact with it and use its various capabilities. The website employs a built-in HTTPS module to access the web APIs and raise various HTTPS requests (GET, PUT, POST, and DELETE) to communicate with the MongoDB database, hence the mobile devices that are running the website need to be connected to the internet.

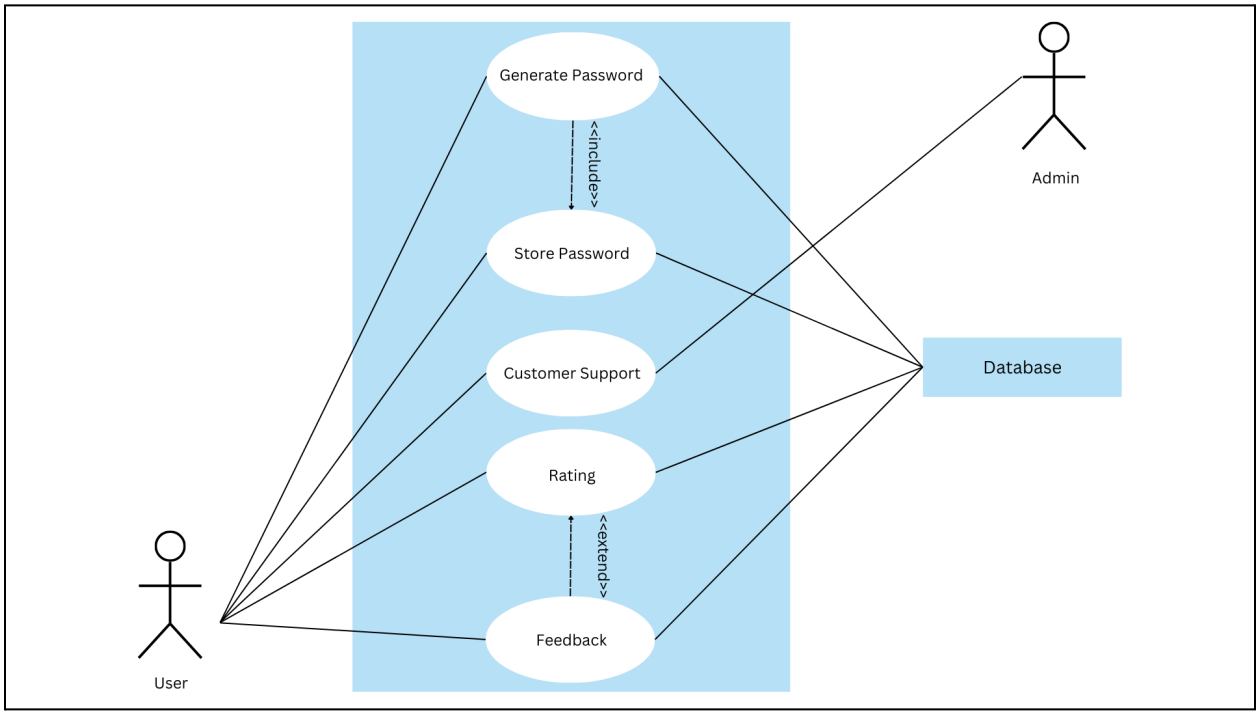


Figure 1 - System Environment

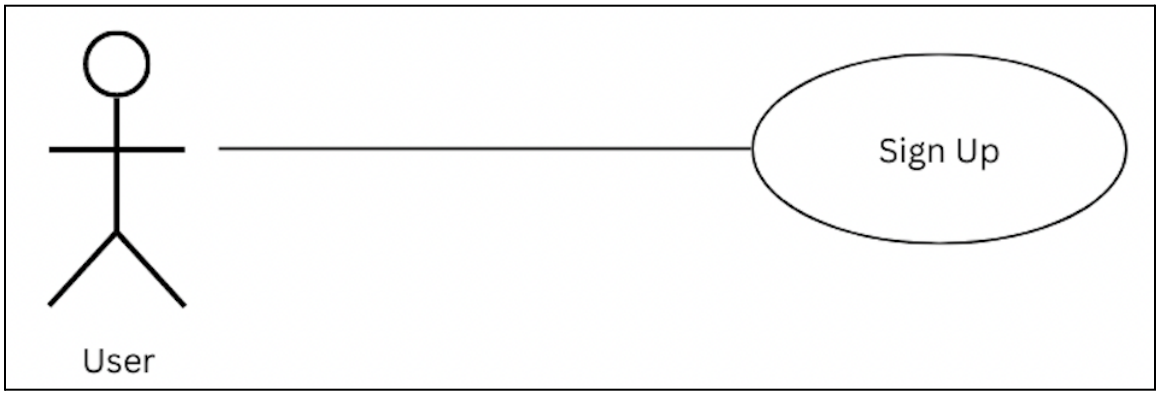
2.2. Functional Requirements Definition

This section outlines the use cases from the perspective of the user of the website.

2.2.1 Use Case for Authentication

2.2.1.1 Use Case: Authenticating the users - Sign Up

UML Diagram:



Brief Description

The User accesses PassMan website homepage, creates a new account, and activates the account using activation mail sent.

Step-By-Step Description

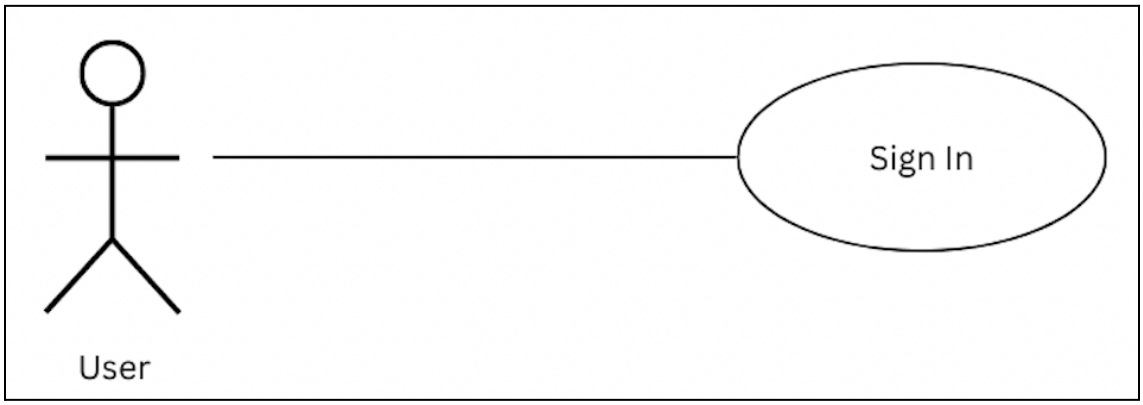
Before this use case can be initiated, the user has already accessed the homepage of PassMan.

1. The user chooses the "Sign Up" button from the homepage.
2. The System displays E-mail Sign Up option.
   - 2.1. The user enters their desired email ID
   - 2.2. The user confirms the password.
   - 2.3. The System sends an activation link to the entered email Id.
   - 2.4. The user activates the account with the activation link.
   - 2.5. The System displays the account activation successful message.
   - 2.6. The System reloads the homepage for the user to Log In.

Xref: 3.1.1

2.2.1.2 Use Case: Authenticating the users - Sign In

UML Diagram:



Brief Description
The User accesses PassMan website homepage, has already made and activated his account using activation mail sent and then logs in using the login credentials.

Step-By-Step Description
Before this use case can be initiated, the user has already made an account on the PassMan website using the signup.

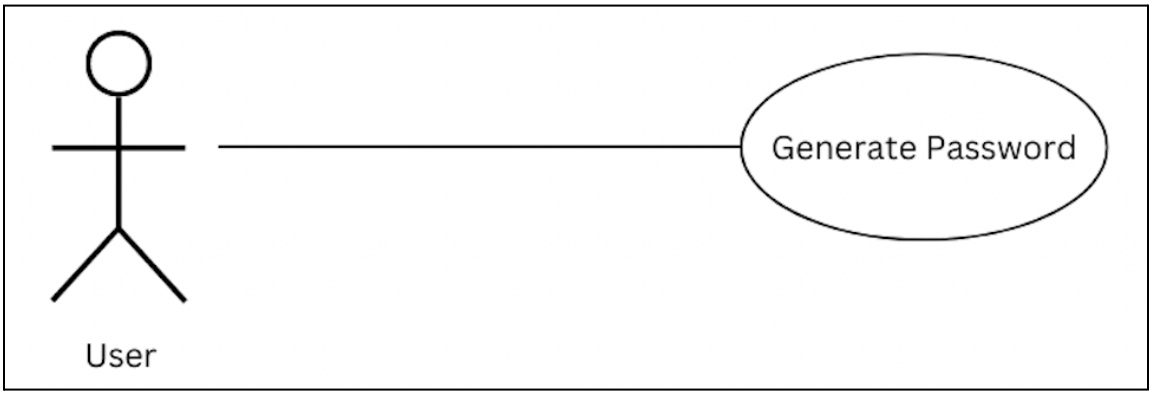
1. The user chooses the "Sign In" button from the homepage, after successfully making an account.
    1.1. The user enters their desired email ID
    1.2. The user clicks the Login button.
    1.3. If the user provides the correct credentials, the user is signed in to this account.

Xref : 3.1.1.2

2.2.2 Use case to Store, Generate and Delete Password

2.2.2.1 Use Case: Generate Password

UML Diagram:



Brief Description
The User accesses PassMan website and based on their requirements generates a password for themselves.

Step-By-Step Description

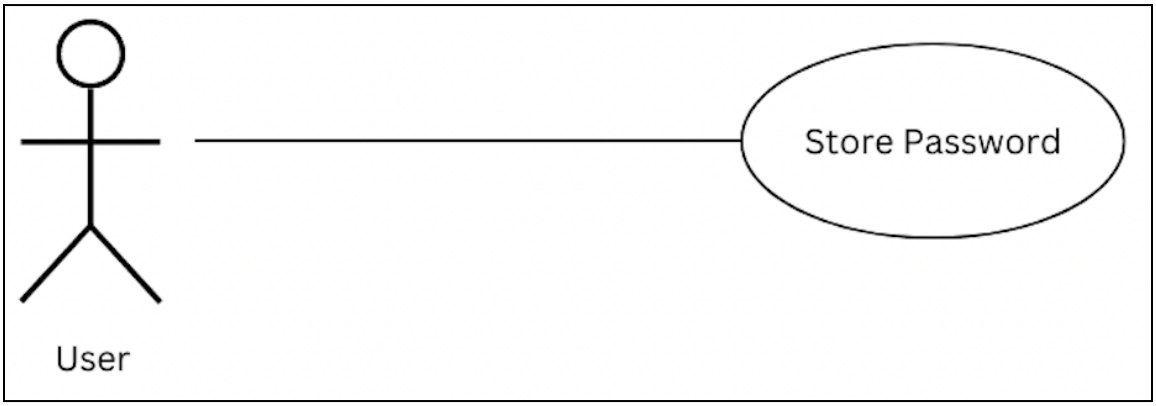
Before this use case can be initiated, the user is on the website's main page and the user gets an option to generate a strong password.

1. Once a password is generated, the user can use and save the generated password.
2. Furthermore, the user gets an option to regenerate the current password if wanted.

Xref: 3.3

2.2.2.2 Use Case: Store Password

UML Diagram:



Brief Description

The User accesses the PassMan website and based on their requirements stores the generated password in the database.

Step-By-Step Description

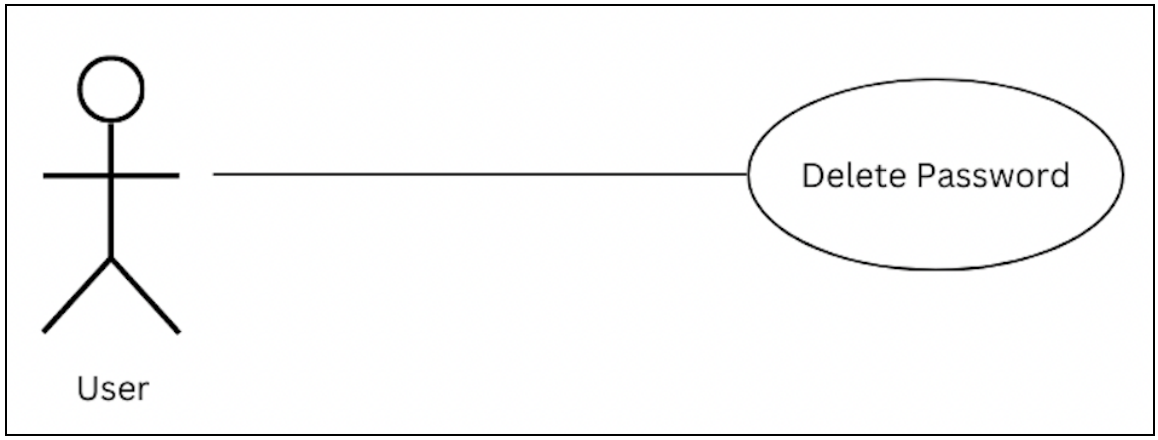
Before this use case can be initiated, the user generates a password and a window pops up if the user wants to store that password.

1. Once a password is stored, the user can access it using the provided private key.
2. Furthermore, the user gets an option to regenerate the current password if wanted.

Xref: 3.2

2.2.2.3 Use Case: Delete Password

UML Diagram:



Brief Description

The User accesses the PassMan website and based on their requirements can delete an earlier stored password in the server's database.

Step-By-Step Description

Before this use case can be initiated, the user is on the website's main page and the user gets an option to delete a stored password.

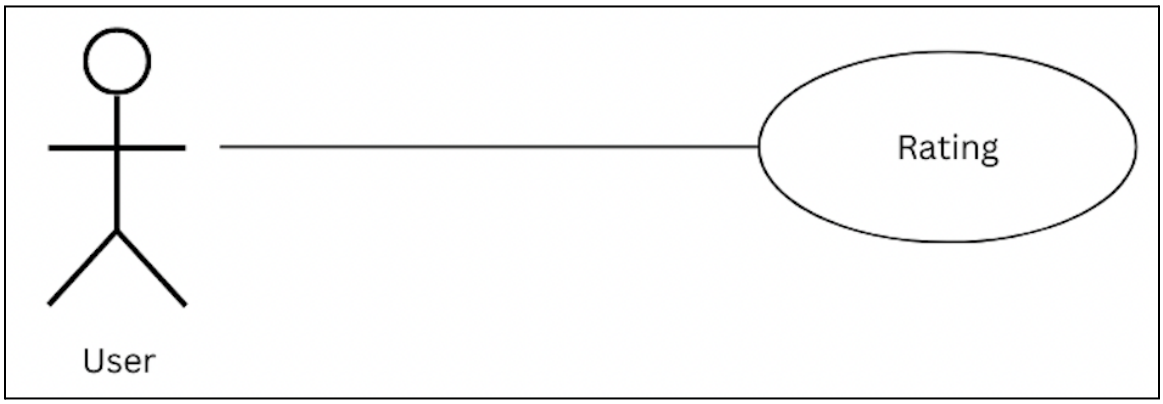
1. User has to select the password that needs to be deleted.
2. Users get a pop-up window to confirm the deletion process or cancel it otherwise.

Xref: 3.5

2.2.3 Use case for Rating and Feedback

2.2.3.1 Use Case: Rating the website - Rating

UML Diagram:



Brief Description

The User accesses PassMan website homepage and based on their experience rates the website. And as per their wish tells about their experience on the website.

Step-By-Step Description

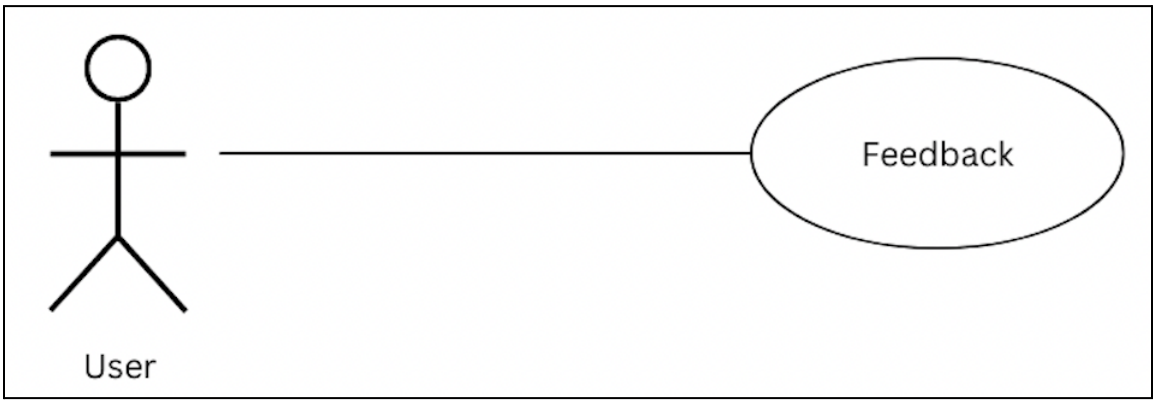
Before this use case can be initiated, the user is on the website after login and a pop up appears to rate the website.

3. The user chooses a rating for the platform as per their wish.
4. And then will be asked if they want to give feedback to the website.

Xref

2.2.3.2 Use Case: Rating the website - Feedback

UML Diagram:



Brief Description

The user accesses and rates the PassMan website homepage and gives a feedback as per the choice.

Step-By-Step Description

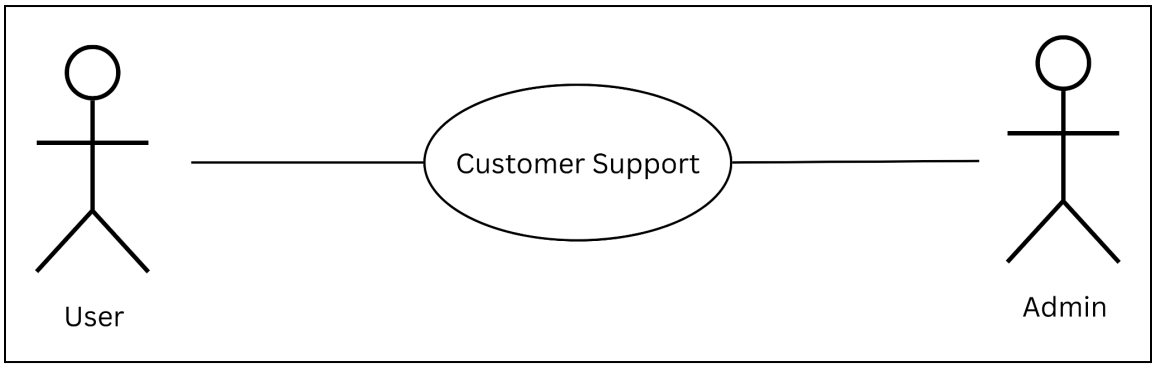
Before this use case can be initiated, the user has already rated the website and is required to give the feedback as per the like.

1. The user writes the feedback and clicks on the submit button.

Xref

2.2.4. Use Case: Customer Support

UML Diagram:



Brief Description

By using this feature, users can resolve all their dilemmas regarding anything on the platform.

Step-By-Step Description

1. All the users registered are visible to the admin and it ensures their authenticity by verifying whether all the documents needed for authentication have been uploaded at the time of registration.
2. The users can ask a query, or about any issue that they are facing using the website.
3. And the issues will be looked at and their queries will be resloved via communicating with them.

Xref

2.3 User Characteristics

The user Characteristics of our users include:

- Security-conscious:
  Users of a password manager are typically security-conscious individuals who are concerned about protecting their personal and sensitive information. They understand the importance of using strong, unique passwords for each account and recognize the risks of reusing passwords or storing them in insecure locations.
- Tech-savvy:

Users of a password manager may be technically inclined individuals who are comfortable using software tools and have a good understanding of basic computer security principles. They may also be early adopters of new technology and eager to try out new software solutions.

- Multiple online accounts:
  Users of a password manager typically have multiple online accounts across various websites and applications, which can make it challenging to remember and manage all of their login credentials.

- Busy schedule:
  Users of a password manager may have busy schedules and limited time to manage their passwords manually. A password manager can help save time by automating the process of generating, storing, and filling in passwords.

- Trustworthy:
  Users of a password manager need to be trustworthy individuals who are careful with their master password and other sensitive information. They must also be willing to follow security best practices, such as using two-factor authentication and keeping their software up-to-date.

2.4 Non-Functional Requirements

PassMan will be on a server with high speed Internet capability. The speed of the user's connection will depend on the hardware used rather than characteristics of this system.
Since, PassMan is a website supported for modern web browsers, no changes need to be made specifically for the operating system compatibility as all three windows, Linux and MacOS supports modern web browsers.

3. Functional Requirements

3.1 User Authentication

The password manager will require users to authenticate using a master password. The master password will be hashed and stored securely in the system. The password manager will prompt the user to create a new master password if it is their first time using the software.

3.1.1. Sign UP

Use Case Name	Sign Up
XRef	Section 2.2.1.1
Trigger	The user clicks on the sign up button
Precondition	The user is on the landing page of the website
Basic Path	i. The user enters their desired email ID ii. The User enters the following details depending on their role: ● **Parent/Guardian** 1. E-mail ID 2. First Name 3. Last Name 4. Address 5. New Password ● **Creche Facility Provider** 1. E-mail Id 2. First Name 3. Last Name 4. Creche Location 5. Creche Name 6. New Password iii. The user confirms the password. iv. The System sends an activation link to the entered email Id.

	v. The user activates the account with the activation link. vi. The System displays the account activation successful message. vii. The System reloads the homepage for the user to Log In.
Postcondition	The user's account is created.
Exception Paths	The user can quit this process at any time.

3.1.2. Sign IN

Use Case Name	Sign In
XRef	Section 2.2.1.2
Trigger	The user clicks on the sign in button
Precondition	The user is on the landing page of the website
Basic Path	i. The user enters their desired email ID ii. The User enters the following details depending on their role: ● **Parent/Guardian** 1. E-mail ID 2. Password ● **Creche Facility Provider** 1. E-mail Id 2. Password ● **Admin** 1. Email-id 2. Password The admin's email id and password is already provided. The admin has to just sign in using those credentials. iii. The user clicks the Login button. iv. If the user provides the correct credentials, the user is signed in to this account.

Alternative Paths	**a. Google Sign In** i. The user selects Google Sign-In ii. The System displays the list of Google Accounts iii. The user selects their desired account iv. The user's account is created successfully v. The user is signed up and is signed in to the dashboard with the Google account
Postcondition	The user is logged in to the website and is taken to the homepage of the website.
Exception Paths	If the user has not created their account then they will no tbe able to login, so this use case will be abandoned. Moreover, the user can exit this process at any time.

3.2 Secure Password Storage / Add password

The password manager will use encryption to store passwords securely. The encryption algorithm used will be RSA-256. Passwords will be stored in an encrypted format, and only the user with the private key will be able to decrypt and view them.

Use Case Name	Add a Password
XRef	Section 2.2.2.2
Trigger	The user clicks on the add passowrd button
Precondition	The user is already logged in and is present on the home page.
Basic Path	1. The user clicks on the add password button. 2. The user will have to enter the site url the username/ the gamil id he has made an account with and the password

Alternative Paths	None
Postcondition	The user has logged in.
Exception Paths	The creche facility provider can quit this process at any time.

3.3 Password Generation

The password manager will have a feature to generate strong passwords for users. The user will be able to select the length and complexity of the password generated.

Use Case Name	Password Generation
XRef	Section 2.2.2.1
Trigger	The user clicks on the generate new password button
Precondition	The user is on the home of the website
Basic Path	Before this use case can be initiated, the user is on the website's main page and the user gets an option to generate a strong password. 1. Once a password is generated, the user can use and save the generated password. 2. Furthermore, the user gets an option to regenerate the current password if wanted.
Postcondition	The user has generated a new password and is asked for the option to save the generated password or regenerate the current password.

3.4 Password Import

The password manager will allow users to import and export their password data. The import and export process will use encryption to ensure the security of the password data.

Use Case Name	Passwords Import
XRef	Section 2.2.3.3
Trigger	The user provider clicks on the view password button
Precondition	The user is already logged in and is present on the desired all passwords.
Basic Path	1. By this point, the user has already added the passwords it wants to view. 2. Using this feature, the user can choose the passwords he wants to view and then view them using the view button. 3. The user will be asked to eneter the private key to view the encrypted passwords.
Alternative Paths	None
Postcondition	The user is asked for private key and the password is then shown.
Exception Paths	The user can quit this process at any time.

3.5 Password Deletion

The password manager will allow users to delete their excitingly stored passwords.

Use Case Name	Delete a Password
XRef	Section 2.2.2.3
Trigger	The creche facility provider clicks on the delete button
Precondition	The creche facility provider is already logged in and is present on the desired page.
Basic Path	1. By this point, the user has already added the passwords.

	2. Using this feature, the user can choose the passwords he wants to delete and then delete them using the delete button.
Alternative Paths	None
Postcondition	The user is asked for confirmation and the password is then deleted.
Exception Paths	The user can quit this process at any time.

4. Non-Functional Requirements

4.1 Performance

The password manager should be able to handle a large number of passwords without any performance issues. It should be able to store and retrieve passwords quickly and efficiently.

4.2 User Interface

The user interface should be intuitive and easy to use. It should be designed to provide a seamless user experience.

4.3 Security

The password manager should be designed to be secure from any potential threats. It should have measures in place to prevent unauthorized access to user data.

5. System Architecture

The password manager will be designed using a client-server architecture. The client application will be installed on the user's device, while the server application will store and manage the encrypted password data. The client application will communicate with the server application using HTTPS protocols.

5.1 User Interface

The password manager will have a user-friendly interface. The interface will provide easy access to all the features of the password manager. It will be designed to be intuitive and easy to use.

5.2 Security Considerations

- The password manager will have the following security measures in place:
- Encryption of password data using AES-256
- Use of HTTPS protocols for communication between the client and server applications
- Password hashing and salting for user authentication
- Multi-factor authentication for added security
- Use of the latest security standards and protocols

5.3 Performance Requirements

1. Encryption and decryption speed

   The password manager must be able to quickly encrypt and decrypt passwords using a strong cryptographic algorithm to ensure security. The performance of the encryption and decryption process should be fast enough to avoid slowing down the user experience.

2. Database read and write speed

   The password manager must be able to read and write data from the database quickly, as users will be accessing and modifying their password data frequently. Proper indexing and query optimization can help improve database performance.

3. User authentication and authorization speed

   The user authentication and authorization process should be fast and secure to ensure that only authorized users can access the password manager. Properly implementing authentication and authorization using secure protocols such as OAuth or JWT can help improve performance.

4. User interface speed

   The user interface should be responsive and fast, with minimal lag time when loading data. Proper caching and asynchronous loading of data can help improve user interface speed.

5. Scalability

   The password manager should be able to handle a growing number of users and passwords without slowing down or becoming unstable. Implementing load balancing and horizontal scaling techniques can help improve scalability.

6. Security

   The password manager should be designed with security in mind, and all security measures should be carefully implemented and regularly audited to ensure maximum protection against attacks.

7. Backup and recovery

   The password manager should have a reliable backup and recovery system in place, with backups taken regularly to prevent data loss in case of system failures or disasters.

5.4 Testing Requirements

The password manager will undergo extensive testing to ensure that the ecrytio

Here are some of the testing requirements that you may consider for your password manager:

1. Functional Testing

   Functional testing is a type of testing that checks whether our password manager functions as expected. This type of testing covers the following scenarios:

   - Creating a new account
   - Adding a new password

- Editing an existing password
- Deleting a password
- Generating a password
- Copying a password to the clipboard
- Searching for a password
- Logging in and out of the password manager

2. Security Testing

Since our password manager is meant to be cryptographically secured, we ensured that it is protected against common security threats. Some of the security testing requirements for our password manager include

- Checking for SQL injection vulnerabilities
- Checking for cross-site scripting (XSS) vulnerabilities
- Testing for weak passwords and password reuse
- Ensuring secure password storage and retrieval
- Ensuring secure communication between the client and server
- Testing for other security vulnerabilities like buffer overflows, format string attacks, etc.

3. Performance Testing

Performance testing checks whether our password manager can handle a large number of users and passwords without slowing down. Some of the performance testing requirements for your password manager may include:

- Testing the response time of the password manager under normal and peak loads
- Testing the memory usage of the password manager
- Testing the scalability of the password manager

4. Usability Testing

Usability testing checks whether our password manager is user-friendly and easy to use. Some of the usability testing requirements for your password manager may include:

- Testing the user interface of the password manager
- Checking whether the user interface is intuitive and easy to navigate
- Testing the accessibility of the password manager

5. Compatibility Testing

   Compatibility testing checks whether our password manager works as expected across different browsers, devices, and operating systems. Some of the compatibility testing requirements for your password manager may include:

   - Testing the password manager on different browsers like Chrome, Firefox, Safari, etc.
   - Testing the password manager on different devices like desktops, laptops, tablets, and smartphones.
   - Testing the password manager on different operating systems like Windows, Mac OS, Linux, etc.

These are just some testing requirements we have considered for our password manager. To ensure your password manager is secure and reliable, it's important to design and implement a comprehensive testing strategy that covers all aspects of the password manager's functionality, security, performance, usability, and compatibility.