

Tribhuvan University

**Madan Bhandari Memorial College**

*Kathmandu, Nepal*

**A Project Proposal**

**on**

**“Password Manager Extension for Google Chrome”**

*A Project report submitted in partial fulfillment of the requirement for the degree of  Bachelor of Science (B.Sc.) Computer Science and Information Technology*

Submitted By

**Sarita Sharma  
 Saumyata Nepal  
Shubharaj Thapa  
 Yuvraj Sah**

Submitted To:

Er. Ram Kumar Basnet

Department of Computer Science and Information Technology

New Baneswor, Kathmandu

2025

**ABSTRACT**

The proposed project focuses on developing a Password Manager Extension for Google Chrome, designed to enhance the user experience in managing online credentials securely. With increasing reliance on digital platforms, individuals face challenges in maintaining unique, strong, and secure passwords for multiple accounts. This extension aims to simplify this process by providing two primary functionalities Password Generation and Password Retrieval.

The extension leverages advanced encryption algorithms to ensure data confidentiality and employs multi-factor authentication for secure access. Its lightweight design integrates seamlessly with Google Chrome, enabling users to store, generate, and retrieve passwords with minimal effort. This tool not only reduces the risks of weak or reused passwords but also promotes safer digital practices.

By addressing the core needs of password management, the Password Manager Extension provides an efficient, user-friendly, and secure solution tailored for modern internet users.

**OBJECTIVE**

The objective of the **Password Manager Extension for Google Chrome** is to provide a secure and efficient way for users to manage their passwords directly from their browser. The main objectives are as follows:

1. **Enhance Security:**

Ensure users can generate and store strong, unique passwords using encryption techniques to safeguard sensitive data.

1. **Simplify Password Management:**

Provide an intuitive interface for users to easily generate and retrieve passwords within the Google Chrome browser.

**OVERVIEW**

The Password Manager Extension for Google Chrome is a proposed solution aimed at addressing the challenges of managing online credentials effectively and securely. In today’s digital landscape, where users often juggle multiple accounts across various platforms, maintaining unique and strong passwords is crucial for protecting personal and sensitive data. This extension simplifies the process of password management by integrating two essential features:

**Password Generation:**

* Automatically generates secure and complex passwords.
* Allows customization of password length and character composition (e.g., inclusion of uppercase letters, numbers, symbols).
* Provides users with the option to save the generated passwords securely in an encrypted format.

**Password Retrieval:**

* Enables users to retrieve stored passwords effortlessly through an intuitive interface.
* Ensures security by requiring user authentication, such as a Master Password or Two-Factor Authentication (2FA), to access stored credentials.
* Displays decrypted passwords only after successful verification.

The extension is designed with a user-friendly interface to provide a seamless experience while maintaining strong security standards. By employing encryption to store passwords and implementing robust authentication mechanisms, the tool ensures that user data is protected from unauthorized access.

Key benefits of the Password Manager Extension include:

* Simplified password management for multiple accounts.
* Elimination of weak or reused passwords, enhancing online security.
* Integration within Google Chrome for accessibility and convenience.

This extension is targeted at individuals and professionals who seek a reliable and efficient way to manage their online credentials. With this tool, users can navigate the complexities of modern digital security effortlessly and securely.

**WORKFLOW PROCESS**

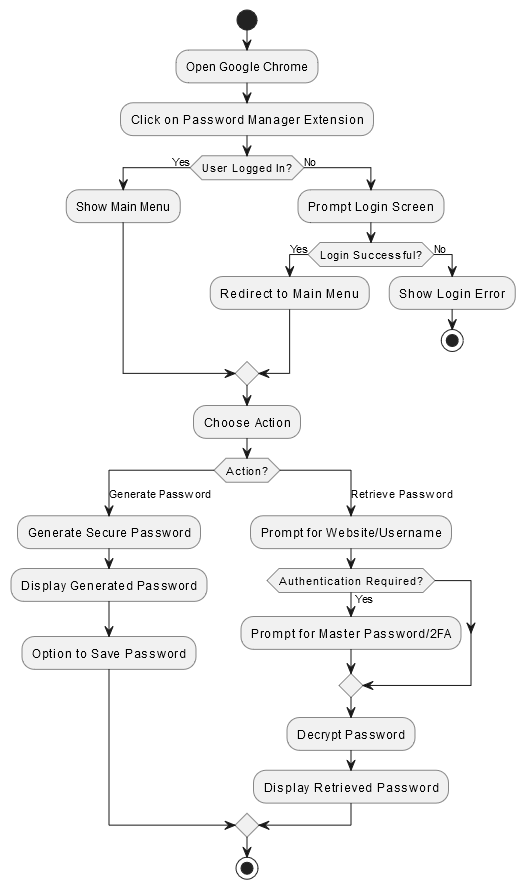


Fig : Workflow Diagram of Password Manager

**Step 1: Accessing the Extension**  
Open Google Chrome and access the password manager extension.

**Step 2: Login Check**  
Check if the user is logged in, and proceed to the main menu or prompt for login.

**Step 3: Login Process**  
Prompt the user to enter credentials, and proceed to the main menu if successful, or show an error message if failed.

**Step 4: Choose Action**  
Present the user with the options to generate or retrieve a password.

**Step 5: Generate Password**  
Generate a secure password if the user selects "Generate Password" and offer an option to save it.

**Step 6: Retrieve Password**  
Prompt the user for the website or username if "Retrieve Password" is selected, and decrypt the password after authentication.

**Step 7: End Process**  
End the process by allowing the user to either exit or log out of the extension.

**Conclusion**

The proposed Password Manager Extension for Google Chrome offers an efficient and secure solution to address the challenges of managing online credentials in the modern digital landscape. By integrating features like password generation, secure storage, and retrieval, the extension simplifies the process of maintaining strong, unique passwords for multiple accounts.

The use of advanced encryption techniques and multi-factor authentication ensures that user data remains secure and protected from unauthorized access. Furthermore, the seamless integration with Google Chrome and a user-friendly interface enhances accessibility and convenience for users.

This tool not only empowers individuals to adopt safer digital practices but also contributes to reducing the risks associated with weak or reused passwords. As reliance on digital platforms continues to grow, the Password Manager Extension stands as a critical tool for promoting a more secure and efficient online experience.

**Functional Requirements**

Functional requirements describe what the system should do. These are the specific behaviors, tasks, or functions that the software is expected to perform. They define the inputs, outputs, and processing the system should handle. Essentially, they specify the core features and functionalities that the system must include to fulfill the needs of the users or stakeholders.

Here are the Functional Requirements of **Password Manager Chrome Extension**

1. **User Registration & Authentication**

* Users must be able to create an account with a username and password.
* Login functionality to authenticate users with the stored credentials.
* Forgot password and reset functionality**.**

1. **Password Generation**

* Users should be able to generate strong, random passwords with options like password length, use of special characters, numbers, etc.
* The system should suggest passwords based on security strength (e.g., weak, moderate, strong).

1. **Password Storage & Encryption**

* Passwords should be securely stored in an encrypted format in the local browser storage or cloud, depending on the architecture.
* Users should have the option to store passwords locally or sync across devices via cloud storage.
* Implement encryption algorithms like AES to store passwords securely.

1. **Autofill and Auto-login**

* The extension should be able to autofill login forms on websites with saved credentials.
* Support auto-login functionality when a user visits a saved site.

1. **Password Vault Management**

* Users should be able to view, add, edit, and delete passwords in their vault.
* Passwords should be organized by website/application name, making it easy to search and retrieve.

1. **Two-Factor Authentication (2FA)**

* Users should have the option to enable 2FA for extra security, which can integrate with the password manager.
* The extension should generate one-time codes for 2FA or work with external apps like Google Authenticator.

1. **Password Strength Analyzer**

* Include a password strength indicator when users create or update passwords, providing feedback on its strength.

1. **Backup & Restore Functionality**

* Users should be able to back up their password vault and restore it when required (e.g., in case of data loss or switching devices).

1. **Multi-device Sync**

* Allow users to sync their passwords securely across multiple devices or browsers (e.g., Chrome, Firefox).

1. **Password Expiry Notification**

* Users should receive notifications or reminders about passwords nearing expiry, based on their set intervals.

**Non-Functional Requirement**

Non-functional requirements specify how the system performs its functions, rather than what functions it performs. These requirements define the quality attributes, constraints, and performance goals the system must meet. Non-functional requirements address issues like security, usability, performance, reliability, scalability, and maintainability**.**

Here are the Non-Functional Requirements of **Password Manager Chrome Extension**

1. **Security**

* All sensitive user data, including passwords and personal information, must be encrypted both during transmission and at rest.
* Secure communication channels (e.g., HTTPS) should be used for sync operations, backup, and restore.

1. **Performance**

* The extension must load within 2 seconds when the browser starts.
* Autofill functionality should be executed within 1-2 seconds once a user visits a saved site.
* Password generation should be quick, with minimal delay (less than 1 second).

1. **Scalability**

* The system should be able to handle large vaults with thousands of entries without performance degradation.
* As the number of users increases, the backend infrastructure should support scaling efficiently.

1. **Reliability**

* The extension should work without crashes or failures in common scenarios (password generation, autofill, etc.).
* The system should ensure data integrity, especially with backup and restore operations.

1. **Usability**

* The interface should be simple, intuitive, and easy to use, catering to users of various technical backgrounds.
* Password management operations (viewing, adding, deleting, and editing passwords) should be seamless.
* Clear instructions or tooltips should be available for new users.

1. **Compatibility**

* The extension should be compatible with the latest versions of Chrome (and other popular browsers if applicable).
* It should work seamlessly across different operating systems (Windows, macOS, Linux).

1. **Backup and Recovery**

* The system should offer efficient and reliable backup options to ensure users can restore their data at any time.

1. **Privacy**

* The extension should not collect unnecessary personal data and must comply with data privacy regulations (e.g., GDPR, CCPA).
* No third-party analytics or data tracking unless explicitly consented by the user.

1. **Accessibility**

* The extension should be accessible for users with disabilities (screen reader support, keyboard navigation).
* The design should adhere to accessibility standards (WCAG).

1. **Cross-Browser Support**

* Ensure that the extension works across different browsers (e.g., Chrome, Firefox, Edge, Safari), ensuring the core functionality remains intact.

**Feasibility**

A feasibility study is conducted to evaluate whether a project is practical, viable, and worth developing. It assesses various factors, including technical, economic, legal, operational, and scheduling aspects. Below is a detailed feasibility analysis for your password manager Chrome extension project.

1. **Technical Feasibility**

This assesses whether the project can be developed using the available technology, tools, and expertise.

**Key Considerations:**

* Browser Compatibility – The Chrome extension can be developed using JavaScript, HTML, CSS, and Chrome’s Extension APIs. Additionally, the extension can be made compatible with Firefox and Edge.
* Security & Encryption – Implementing AES-256 encryption for password storage ensures high security.
* Autofill & Auto-login – Chrome provides an Autofill API that allows for filling login fields automatically.
* Storage & Sync – Passwords can be stored in Chrome’s local storage, or users can choose cloud synchronization options.
* Integration with Two-Factor Authentication (2FA) – The extension can support TOTP-based authentication (Google Authenticator, Authy).

**Challenges:**

* Security Vulnerabilities – The extension must prevent keylogging, phishing attacks, and data leaks.
* Cross-Browser Compatibility – Adapting to Firefox and Edge might require additional configurations.
* Data Backup & Restore – Secure implementation of data export/import must be ensured.

1. **Economic Feasibility**

This evaluates whether the project is financially viable in terms of development costs and potential revenue.

**Cost Estimation:**

Development Costs:

* Coding & Testing – No cost (self-developed)

Potential Revenue Streams:

* Freemium Model – Offer basic password storage for free and charge for premium features (multi-device sync)
* Donations/Open Source Contributions – the project is open source, crowdfunding and community support could help.

1. **Legal Feasibility**

This determines whether the project complies with laws and regulations.

Key Legal Considerations:

* Data Protection & Privacy Laws – Must comply with GDPR (Europe), CCPA (California), and other global privacy laws.
* Security Compliance – Must follow OWASP best practices for secure password management.
* Google Chrome Web Store Policies – The extension must meet Google's guidelines for listing on the Chrome Web Store.

1. **Operational Feasibility**

This examines whether the project can be successfully developed and used in a real-world scenario.

Key Considerations:

* User Experience (UX) – The extension is simple and easy to use for non-technical users.
* Security & Reliability – Provides secure, offline access, and support multi-device usage.
* Customer Support & Updates – Regular updates and a help desk (FAQs, email support) is available.

Challenges:

* Users forgetting their master password (solution: password recovery via secret key or backup codes).

**Hardware And Software Used**

For the development and deployment of my password manager Chrome extension, I have carefully selected the necessary hardware and software components to ensure smooth functionality, security, and efficiency. Below is a detailed overview of the resources I am using in this project.

1. **Hardware Requirements**

The project requires specific hardware for both development and end-user usage.

**A. Development Hardware (My Setup)**

During the development phase, a system with the following specifications was used to ensure smooth coding, testing, and debugging:

***Processor****:* Intel Core i5/i7 or AMD Ryzen 5/7 (or higher)

***RAM:*** 8GB (Recommended: 16GB for better multitasking)

***Storage:*** SSD with at least 256GB (Recommended: 512GB for faster performance)

***Graphics Card****:*  Integrated GPU (Intel UHD Graphics or equivalent)

***Operating System:*** Windows 10/11, macOS, or Linux

***Internet Connection:*** Stable connection for API testing and cloud integration

**B. End-User Hardware (For Users)**

The extension is designed to work seamlessly on most modern devices, ensuring accessibility for users.

***PC/Laptop***: Any system that supports Google Chrome

***Internet Connection:*** Required for cloud sync, but the extension will function offline for locally stored passwords

**2. Software Requirements**

To develop, test, and maintain Password Manager Chrome extension, various software tools and frameworks are used.

**A. Development Tools**

The following tools are chosen for writing and debugging the code:

***Google Chrome (Latest Version):*** For testing and running the extension

***Chrome Developer Tools***: To debug and analyze performance

***Visual Studio Code (VS Code):*** preferred code editor for development

***Node.js & npm:*** For managing dependencies

***Git & GitHub:*** To track code changes and collaborate efficiently

**B. Programming Languages**

The extension is being developed using a combination of the following languages:

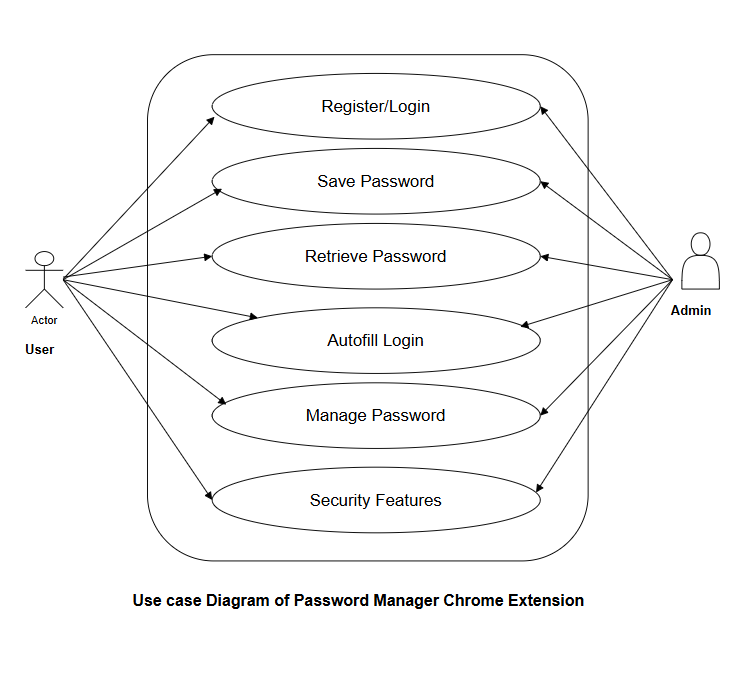
**JavaScript (ES6+):** Core functionality and logic

**HTML & CSS:** Designing the user interface

**JSON:** Data storage and API responses

**Use Case Diagram**

A Use Case Diagram is a visual representation of the interactions between users and the system. It helps identify the system’s functionalities, the actors involved, and how they interact with different features.



The use case diagram provided visually represents the functionalities of a password manager Chrome extension, showing the interactions between the User, Admin, and the system.

**Actors in the Diagram**

* **User** – The primary actor who interacts with the password manager extension to perform various actions.
* **Admin** – The secondary actor responsible for managing the overall system, ensuring security, and overseeing user activities.

**Use Cases (Functionalities)**

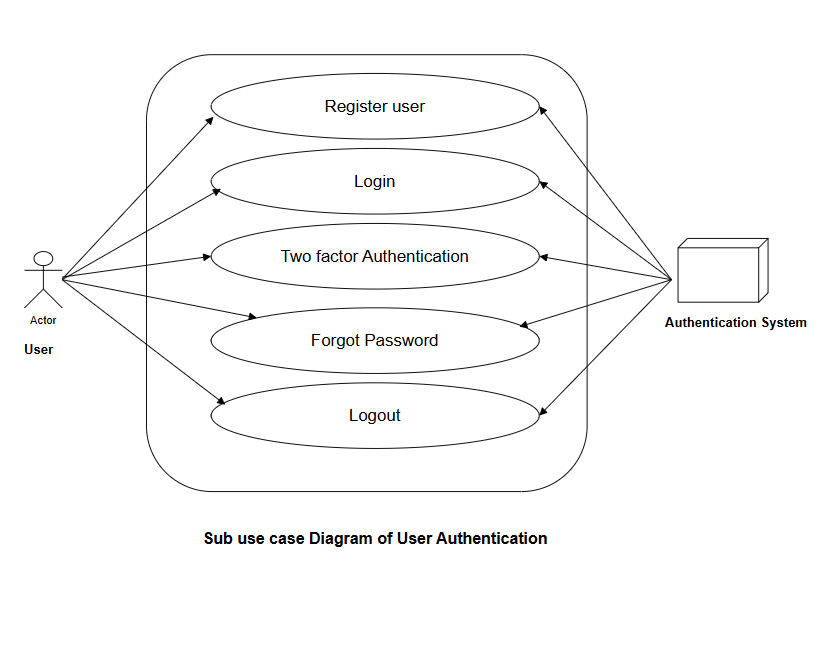
The system provides six main functionalities (use cases), as shown in the diagram:

* **Register/Login** – Allows users to create an account or log in to access the password manager.
* **Save Password** – Enables users to securely store their passwords for different websites.
* **Retrieve Password** – Allows users to access stored passwords when needed.
* **Autofill Login** – Automatically fills in login credentials on websites where passwords are saved.
* **Manage Password** – Users can update, edit, or delete their saved passwords.
* **Security Features** – Includes additional security options such as two-factor authentication (2FA), encryption, and logout features.

**Relationships and Interactions**

* Both User and Admin can interact with all the use cases.
* User performs essential password-related actions like saving, retrieving, and managing passwords.
* Admin oversees security and ensures the system functions properly.

**Sub Use case diagrams**

1. **User Authentication**

**Actors in the Diagram**

* **User** – The primary actor who interacts with the authentication system to access the password manager.
* **Authentication System** – The system that handles user verification, login security, and account recovery.

**Use Cases (Functionalities)**

**Register User –**

* Allows a new user to create an account.
* The system verifies and stores user credentials securely.

**Login –**

* Users enter their credentials to access the password manager.
* The system validates the credentials before granting access.

**Two-Factor Authentication (2FA) –**

* An additional security layer that requires a verification code (e.g., via email or authenticator app).
* Enhances account security to prevent unauthorized access.

**Forgot Password –**

* If a user forgets their password, they can reset it using email or security questions.
* The authentication system verifies the user’s identity before allowing a reset.

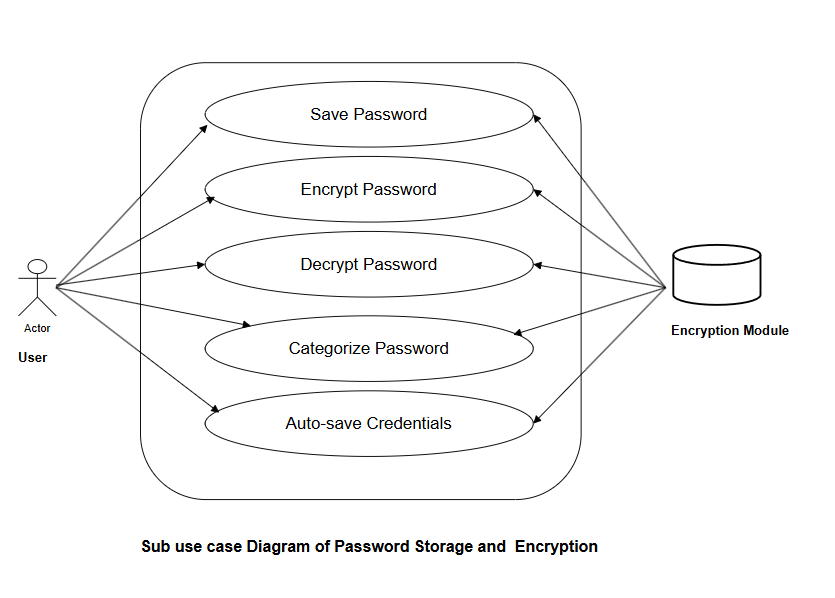
**Logout –**

* Users can securely log out of their accounts to prevent unauthorized access.
* The system ensures session termination after logging out.

**Relationships and Interactions**

* The User initiates all authentication processes.
* The Authentication System verifies credentials, manages security features, and handles password recovery.
* All use cases are directly linked to the authentication system, ensuring security in user access.

1. **Password Storage and Encryption**



**Actors in the Diagram**

* **User** – The primary actor who saves and manages passwords within the extension.
* **Encryption Module** – A system component responsible for encrypting and decrypting stored passwords to ensure security.

**Use Cases (Functionalities)**

**Save Password** –

* + Users can store passwords securely in the password manager.
  + The system ensures that passwords are saved in an encrypted format.

**Encrypt Password** –

* + Before storing passwords, they are **encrypted** using cryptographic techniques.
  + Ensures passwords are stored securely and protected from unauthorized access.

**Decrypt Password** –

* + When a user retrieves a saved password, the system **decrypts** it securely.
  + Only authorized users can access the decrypted version.

**Categorize Password** –

* + Users can organize stored passwords into categories (e.g., Social Media, Banking, Work Accounts).
  + Helps in efficient password management.

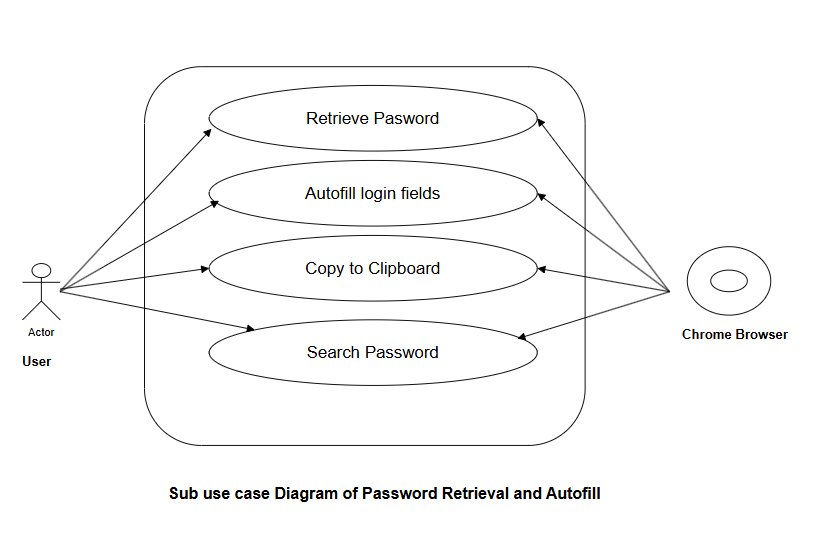
**Auto-save Credentials** –

* + The extension automatically detects login credentials and prompts users to save them.
  + Reduces manual effort and enhances convenience for users.

**Relationships and Interactions**

* The User interacts with all the use cases to manage passwords securely.
* The Encryption Module is responsible for encrypting and decrypting passwords, ensuring that stored data remains protected.
* Each use case is linked to the Encryption Module, which ensures the security of password storage and retrieval.

1. **Password Retrieval and Autofill**

****

**Actors in the Diagram**

* **User** – The primary actor who interacts with the password manager to access stored credentials.
* **Chrome Browser** – The platform where the password manager operates, facilitating autofill and retrieval features.

**Use Cases (Functionalities)**

**Retrieve Password –**

* Allows the user to retrieve stored passwords from the password manager.
* The system securely decrypts and displays the password when requested.

**Autofill Login Fields –**

* Automatically fills in login credentials on websites when users visit known login pages.
* Reduces manual entry and improves convenience.

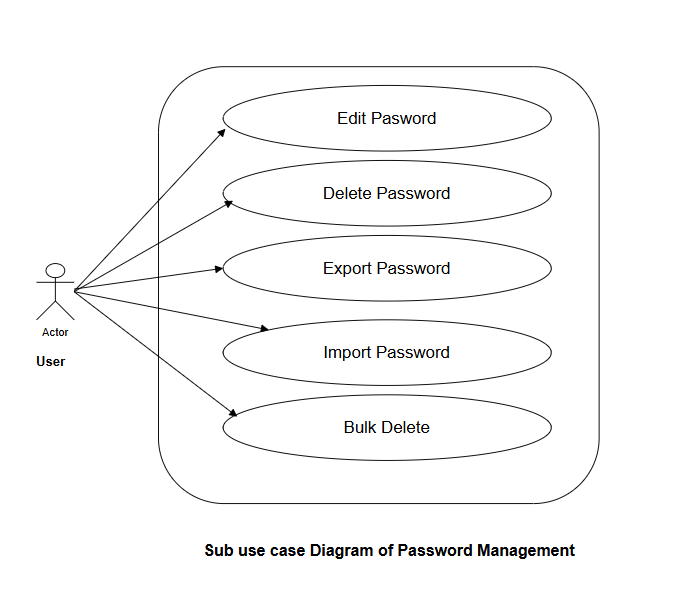
**Copy to Clipboard –**

* Enables users to copy a stored password for pasting elsewhere.
* Ensures that users can use their passwords without directly viewing them.

**Search Password –**

* Users can search for specific saved credentials using account names or website details.
* Helps in quick retrieval when multiple passwords are stored.
* Relationships and Interactions
* The User interacts with the system to retrieve and manage saved passwords.
* The Chrome Browser integrates with the password manager to autofill login fields automatically.
* Each use case enhances password accessibility while maintaining security and ease of use.

1. **Password Management**



**Actors in the Diagram**

* **User** – The primary actor who interacts with the system to manage saved passwords.

**Use Cases (Functionalities)**

**Edit Password –**

* Allows users to update or modify an existing saved password.
* Useful when changing credentials on a website.

**Delete Password –**

* Enables users to remove a specific stored password from the system.
* Helps in managing and organizing stored credentials.

**Export Password –**

* Allows users to export stored passwords in a file format for backup or transfer purposes.
* Typically requires authentication for security.

**Import Password –**

* Users can import passwords from an external source or file.
* Helps in migrating data from other password managers.

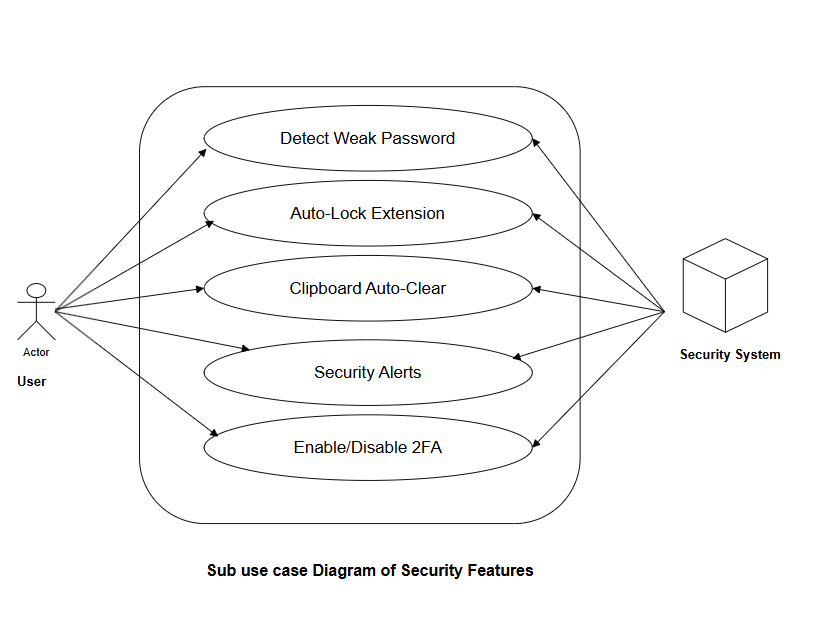
**Bulk Delete –**

* Users can delete multiple saved passwords at once.
* Enhances convenience when removing old or unnecessary credentials.

**Relationships and Interactions**

* The User interacts with all password management functionalities.
* This use case diagram ensures users can efficiently manage stored credentials by editing, deleting, exporting, and importing passwords.

1. **Security Features**



**Actors in the Diagram**

* User – The primary actor who interacts with the security features to enhance protection.
* Security system- Monitors and ensures Data Protection

**Use Cases (Functionalities)**

**Detect Weak Password –**

* Analyzes stored passwords and notifies users if a password is weak or easily guessable.
* Encourages users to create stronger passwords.

**Auto-Lock Extension –**

* Automatically locks the password manager after a specified period of inactivity.
* Prevents unauthorized access when the user is away.

**Clipboard Auto-Clear –**

* Automatically clears copied passwords from the clipboard after a certain time.
* Prevents sensitive data from being exposed to other applications.

**Security Alerts –**

* Sends notifications when suspicious activity is detected.
* Alerts users of potential breaches, unauthorized access, or compromised credentials.

**Enable/Disable 2FA (Two-Factor Authentication) –**

* Allows users to enable or disable Two-Factor Authentication (2FA) for added security.
* Provides an extra layer of protection against unauthorized logins.
* Relationships and Interactions
* The User interacts with all security features.
* The Security System ensures secure handling and processing of these features.

**Relationships and Interactions**

* The User interacts with all security features.
* The Security System ensures secure handling and processing of these features.

**Class Diagram**

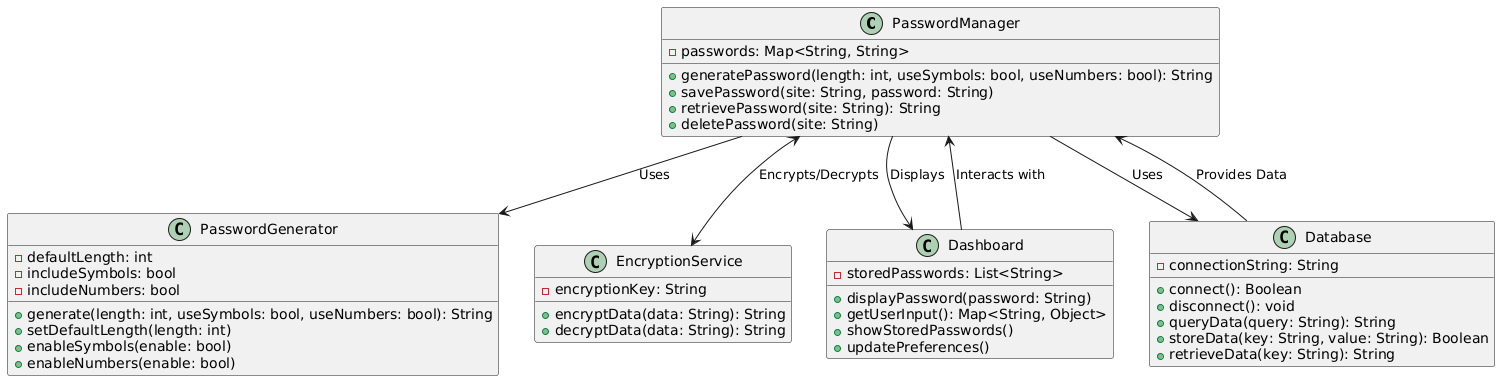


Fig: Class Diagram of the Password Extension

A class diagram is a type of UML diagram that visually represents the structure of a system by showing its classes, attributes, methods, and relationships between objects. It helps in designing the architecture of a software system by illustrating how different components interact.

The given class diagram represents a Password Manager System with five main classes: PasswordManager, PasswordGenerator, EncryptionService, Dashboard, and Database. The PasswordManager class acts as the central component, handling password generation, retrieval, and storage. It utilizes the PasswordGenerator class to create secure passwords based on user-defined criteria and the EncryptionService class to encrypt and decrypt sensitive data before storage. The system interacts with a Database to store and retrieve encrypted passwords securely. Additionally, the Dashboard class is responsible for presenting information to the user, including stored passwords and user input. The relationships between these classes ensure smooth data flow, secure encryption, and efficient password management.

**Sequence Diagram**

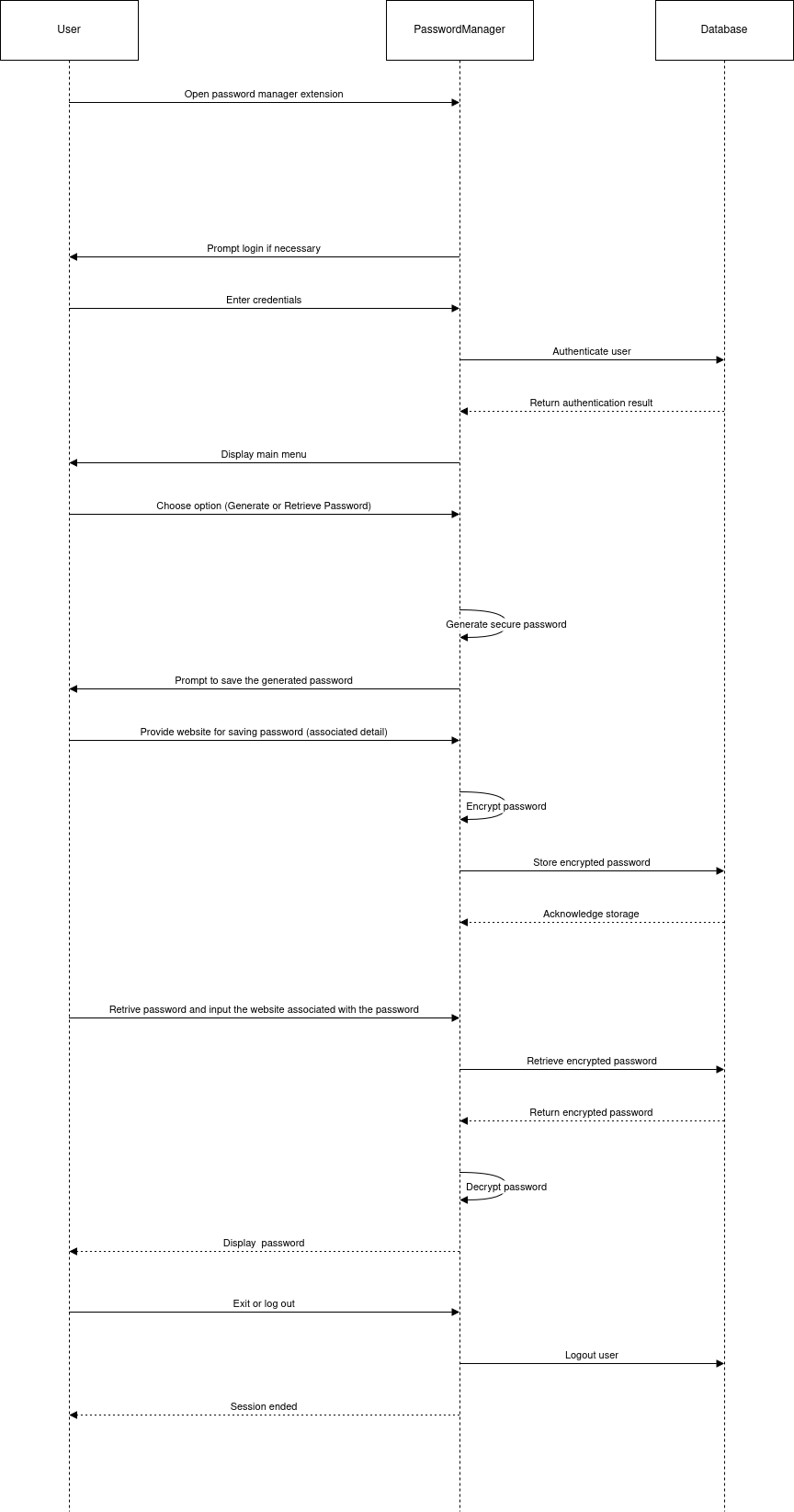


Fig: Sequence Diagram of the Password Extension

A sequence diagram is a type of UML diagram that represents the chronological flow of interactions between system components. It shows the order of messages exchanged between participants to accomplish a specific task.

This sequence diagram illustrates the interactions between the User, PasswordManager, and Database in a password management system. The process includes user authentication, password generation and storage, password retrieval, and user logout.

1. **User Authentication**:
   * The user opens the password manager, which checks their login status with the database.
   * If login is required, the user enters credentials, and the password manager verifies them with the database.
2. **Password Generation & Storage**:
   * The user requests a password generation.
   * The password manager creates a secure password and prompts the user to save it.
   * If saved, the password is encrypted and stored in the database.
3. **Password Retrieval**:
   * The user requests a stored password.
   * The password manager fetches the encrypted password from the database, decrypts it, and displays it to the user.
4. **User Logout**:
   * The user logs out, and the password manager ends the session by notifying the database.

**Activity Diagram**

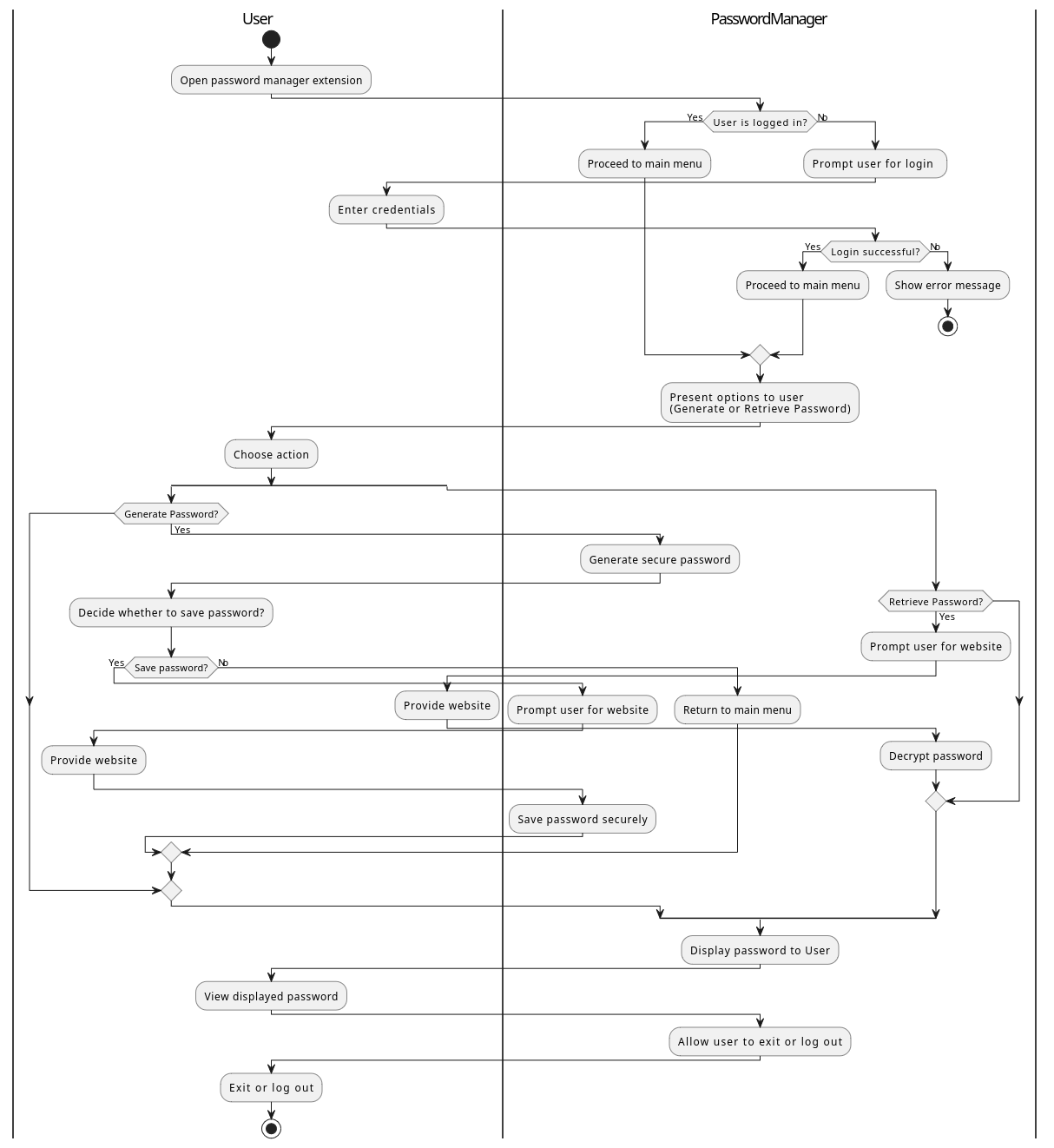


Fig: Activity Diagram of the Password Extension

An activity diagram is a UML diagram that visually represents the flow of control and actions in a system. It focuses on depicting workflows, processes, and decision points, making it useful for understanding the sequence of activities in an application.

The activity diagram for the Password Manager System illustrates the step-by-step workflow for user interactions, starting from accessing the password manager extension to managing passwords securely.

1. User Authentication:
   * The user opens Google Chrome and accesses the password manager extension.
   * The system checks if the user is already logged in.
   * If the user is not logged in, they are prompted to enter their credentials.
   * If authentication is successful, the system proceeds to the main menu; otherwise, it displays an error message.
2. Password Management Options:
   * After successful login, the system presents two main options: **Generate Password, Retrieve Password**
   * The user selects an action.
3. Generating a Password:
   * If the user selects Generate Password, the system creates a secure password.
   * The user decides whether to save the password.
   * If the user chooses to save it, they provide a website name, and the system securely stores the password.
4. Retrieving a Password:
   * If the user selects Retrieve Password, they provide a website or username.
   * The system decrypts the stored password and displays it to the user.
5. Final Actions:
   * After managing passwords, the user can choose to exit or log out.
   * The system ends the session accordingly.

**Component / Deployment Diagram**

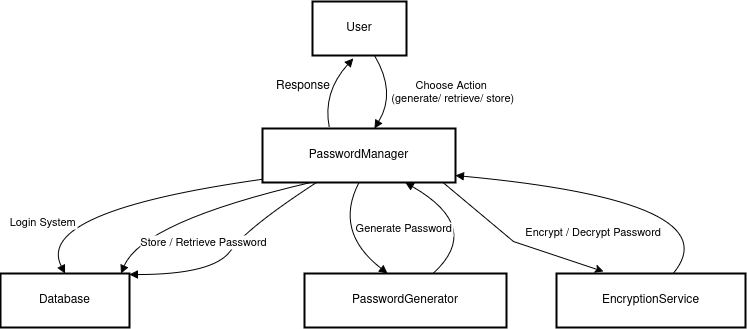


Fig: Deployment Diagram of the Password Extension

A deployment diagram is a UML diagram that models the physical architecture of a system. It shows how different components (software and hardware) interact and how they are deployed on servers, databases, and devices. It is useful for understanding system infrastructure, network communication, and dependencies.

The deployment diagram for the Password Manager System illustrates the interaction between key components, including the User, Password Manager, Database, Password Generator, and Encryption Service. The Password Manager serves as the central component, handling user authentication, password generation, and retrieval. It communicates with the Database to store and retrieve encrypted passwords, while the Encryption Service ensures secure encryption and decryption. The Password Generator creates strong passwords based on user preferences. This architecture ensures secure data handling, efficient password management, and seamless user experience across client and server interactions.