**1. Introduction**

The Password Manager for Windows is a desktop application developed to securely manage passwords and sensitive account information on the Windows platform. This document outlines the functional and non-functional requirements, technical architecture, and design considerations for the application.

**2. Scope**

The scope of the project includes the development of a password manager desktop application for Windows operating systems. The application will allow users to store, retrieve, update, and delete their account credentials securely. It will implement encryption, password hashing, and salting techniques to ensure the security of stored data. The application will not rely on server-side rendering or data storage; all data will be stored locally on the user's device.

**3. Functional Requirements**

* User Authentication: Users will be required to authenticate themselves with a master password to access the password manager.
* Account Management: Users can perform CRUD operations (Create, Read, Update, Delete) on their stored account credentials.
* Password Generation: The application will provide a built-in password generator to create strong and secure passwords.
* Password Checking: Password checkers will be integrated to evaluate the strength and security of user-provided passwords.
* Encryption: All stored data will be encrypted using industry-standard encryption algorithms.
* Password Hashing and Salting: User passwords will be securely hashed and salted before storage to prevent unauthorized access.
* SQL Injection Prevention: Prepared statements will be used to prevent SQL injection attacks, and user inputs will be sanitized and validated before database operations.
* Brute-Force Attack Prevention: The application will implement measures to prevent brute-force attacks, such as limiting the number of login attempts and implementing CAPTCHA or delay mechanisms.

**4. Non-functional Requirements**

* Security: The application must adhere to high-security standards, including encryption, password hashing, salting, and measures to prevent SQL injection and brute-force attacks.
* Performance: The application should provide fast response times for user interactions, even with large amounts of stored data.
* Usability: The user interface should be intuitive and user-friendly, ensuring ease of use for all types of users.
* Compatibility: The application should be compatible with Windows operating systems, including Windows 10 and 11.
* Reliability: The application should be reliable and robust, minimizing the risk of data loss or corruption.
* Rapid Development: WinForms will allow for rapid, easy development of the application.

**8. Testing and Quality Assurance**

* The application will undergo thorough testing, including unit testing, integration testing, and user acceptance testing.
* Quality metrics and acceptance criteria will be defined to ensure the application meets the specified requirements, including security measures against SQL injections and brute-force attacks.

**9. Deployment and Maintenance**

* The application will be deployed as a standalone desktop application for Windows as an executable file.
* On-going maintenance and support will be provided to address any issues or updates, including security patches and enhancements related to SQL injection prevention and brute-force attack mitigation.

**11. Additional Details: Password Reset Options and Logging**

* Password Reset Options: The application will provide users with the option to reset their master password in case they forget it. This process will involve verifying the user's identity through a secure method such as email verification.
* Logging: The application will implement logging mechanisms to record user activities, system events, and security-related incidents. Logged information will include login attempts, password changes, and database operations. Logs will be stored securely and can be accessed by authorized administrators for auditing and troubleshooting purposes.

**12. Class Structure**

* **Master Account**
* **User Account**
* **Password Utils (Checker and Generator)**
* **Database Utils**