Design Requirements

2.2.1 Functions

The application must be able to:

- Generate strong, unique passwords based on user preferences.
- Store passwords securely in an encrypted format.
- Retrieve passwords for the user after secure authentication.
- Authenticate users through a secure master password or other secure methods.
- Validate password strength based on length, complexity, and uniqueness criteria.
- Delete stored credentials securely upon user request.
- Import/Export password data in a secure manner for backup or transfer.
- Test internal logic and functions using unit, integration, and system testing strategies.
- Log user actions (optionally and securely) for audit and debugging purposes.

2.2.2 Objectives

The design of the application should be:

- Secure - to protect user data from unauthorized access.
- Reliable - to function consistently under different conditions.
- User-friendly - to be easily navigable and understandable by users with minimal training.
- Efficient - to operate with minimal resource usage while maintaining performance.
- Maintainable - to allow future modifications or updates with ease.
- Testable - to support thorough verification through various testing methods.
- Portable - to work across multiple platforms or systems if needed.

- Private - to ensure the user’s privacy and data rights are respected.

2.2.3 Constraints

The project will adhere to the following constraints:

- The password storage must be encrypted using industry-standard algorithms
- The application must authenticate users before allowing access to stored passwords.
- The application must operate within a local system environment (no cloud dependency).
- The design must follow test-driven development (TDD) methodology.
- At least four design constraints (economic factors, security compliance, reliability, societal impact) must be addressed.
- The entire application must be developed and tested within two months.