**Title: Design and Implementation of a Password Generator Application**

**Introduction:**

The password generator application is designed to generate strong and unique passwords to enhance security for various purposes, such as online accounts, sensitive documents, or any other instances requiring password protection. This project aims to develop a robust and user-friendly password generation tool using Java programming language.

**Functional Requirements:**

**Generate Passwords:** The application should be able to generate a specified number of strong passwords with customizable length.

**Save Passwords to File:** Users should have the option to save generated passwords to a text file for future reference.

**User Input:** Provide a mechanism for users to input the number of passwords they want to generate.

**Exception Handling:** Implement robust error handling mechanisms to handle potential exceptions during password generation and file operations.

**User Interface:** While not explicitly stated in the provided code, a potential functional requirement could be to develop a simple command-line or GUI interface for users to interact with the application.

**Evaluation Criteria:**

**Efficiency:** Evaluate the efficiency of the password generation algorithm in terms of time and space complexity.

**Reliability:** Assess the reliability of the application in generating passwords without errors and saving them to file securely.

**Usability:** Measure the usability of the application interface in terms of user-friendliness and intuitiveness.

**Security:** Evaluate the strength of the generated passwords and the security measures implemented to protect sensitive data, such as the password file.

**Specifications:**

**Programming Language:** Java

**Libraries**: Standard Java libraries for file I/O and random number generation

**IDE:** Any Java-compatible Integrated Development Environment (IDE) such as Eclipse or IntelliJ IDEA

**Input Method:** Command-line interface for simplicity, though a graphical user interface could be implemented for enhanced user experience.

**Output Format:** Text file containing the generated passwords, with each password on a new line.

**Classes:**

**PasswordGenerator:** Responsible for generating strong passwords based on specified criteria.

**FileHandler:** Handles file operations, including saving generated passwords to a text file.

**UserInput:** Manages user input for specifying the number of passwords to generate.

**Conclusion:**

The password generator application presented in this project provides a simple yet effective solution for generating strong and secure passwords. By adhering to functional requirements, implementing robust error handling mechanisms, and considering evaluation criteria, the application aims to meet the needs of users seeking reliable password generation capabilities. With further enhancements and optimizations, such as incorporating additional security features or developing a more intuitive user interface, the application can be further refined to provide even greater value to its users.