Planning (508 words)

The Scenario:

I am an IB computer science HL student. My adviser for this project is my computer science teacher, Mr. Baker. My client is an investor; his name is Franklin. He relies on the internet and applications to make profit. The main problem he has when working is that it gets difficult for him to track the many usernames and passwords he inputs for his work. Overall, he wants an application that can manage his passwords. I proposed a password management system that stores his passwords behind a single, "*master*" password, so Franklin will only have to remember one password. This product will solve Franklin's problem because it alleviates the stress of having to keep track of various usernames and passwords mentally through an automated alternative.

Initial Consultation with Client and/or Adviser:

In order to arrive at my solution, I had to maintain full clarity on the problem Franklin has. To bring clarity, I conducted an interview with Franklin to figure out what issues he has with his work. Through the consultation, I found out he currently uses Notepad for storing his login information, and he wants a change to a password management tool that makes the login process simpler. He said he wants the application to allow him to create, read and update his passwords in a secure manner. Also, he wants the application to allow him to copy and paste login information into a webpage for logging in. The culmination of everything he told me allowed me to arrive at a solution. Our thorough discussion is provided in Appendix A.

The Proposed Product:

The solution I proposed to Franklin is a password management system that securely stores his passwords. This application will allow for Franklin to alleviate the stress of remembering passwords. The programming language I have chosen to use is Java, for Java has four attributes that makes it a very powerful object-oriented language: abstraction, polymorphism, inheritance and encapsulation. Abstraction is where only essential information is used to generalize certain concepts. This method is used in object-oriented programming languages to allow comprehensibility and readability for software developers. Abstraction will benefit me in this project because it will allow me to conceptualize the program I'm creating. Moreover, polymorphism is the ability for a name to refer to objects of many different class types. Polymorphism is used to manipulate objects in unique ways without knowledge of their class types. The concept of polymorphism will be beneficial for me because it increases reusability and ensures readability of code. Additionally, inheritance is where classes use the attributes and behaviors of other classes. This concept will allow me to maintain organization and focus when writing code. Lastly, encapsulation is where attributes of a class can be restricted to that particular class or not. This function is beneficial for me because it ensures flexibility for different tasks in my coding interface. Moreover, I naturally chose to use a graphical user interface to provide usability. I will be using the Java Swing library to implement a graphical user interface (GUI).

Specific Performance (Success) Criteria:

In order to ensure I focus on my client's problem, I have posed three potential cases for success in this project:

i. **Best-Case Requirements -** The application is fully functional, and it meets all supplementary features:
    - The application meets all typical and minimal requirements.
    - The application allows the user to access a website through the password viewing window.

- The application allows the user to copy password information into their clipboard.

ii. **Typical-Case Requirements -** The application is fully functional, and it meets some supplementary features:

- The application meets all minimal requirements.
- The application utilizes a GUI.
- The application allows the user to update and delete existing passwords.
- The passwords are decrypted and encrypted for reading and writing, respectively.

iii. **Minimal-Case Requirements -** The application is partially functional:

- The application possesses a command line interface (CLI).
- The application allows the user to create and read passwords.
- The application allows the user to access their productivity system through a hashed "*master*" password.
- The application stores the user's passwords through a "*.txt*" file.