CMPT 220 Project 2 Final Writeup

Abstract:

My project is a simple password manager. It is intended to be used to store login and account ID`s and passwords in an organized way while keeping the information safe. It can help with those who have many accounts for many different things and find it hard to keep them all straight in there head.

Intro:

I have upwards of 50 accounts in my name for various reasons that I need to be able to remember and keep all information about them safe and secure, while being able to access them easily. The password manager I have created uses basic encryption to ensure the safety of all the information it collects. To do this all accounts are placed in a JSON file in respect to their fields they belong in and the file is secured.

Detailed System Description:

The system runs as follows:

1. When the program starts, If Accounts.JSON file exits the file will be opened and sent to the decrypt method to be dycrypted
2. Then the login method is called, which prompts the user for the master password, if entered correctly the menu method is called
3. When the menu method is called the integer: userchoice is set equal to the returned value of the menuChoice method.
4. The user is prompted with a menu and depending on the integer returned a method will be called.
5. If the number 1 is entered the write method will be called
    - The user is prompted to enter the Name, username and password
    - If Accounts.JSON file exits, open the file else create the file
    - Add new object to Array of account objects
6. If the number 2 is entered the read method will be called

- If Accounts.JSON file exits the file will be opened
- Converts JSON array into string
- Prints string of all accounts on file

7. If the number 3 is entered the generatePW method is called

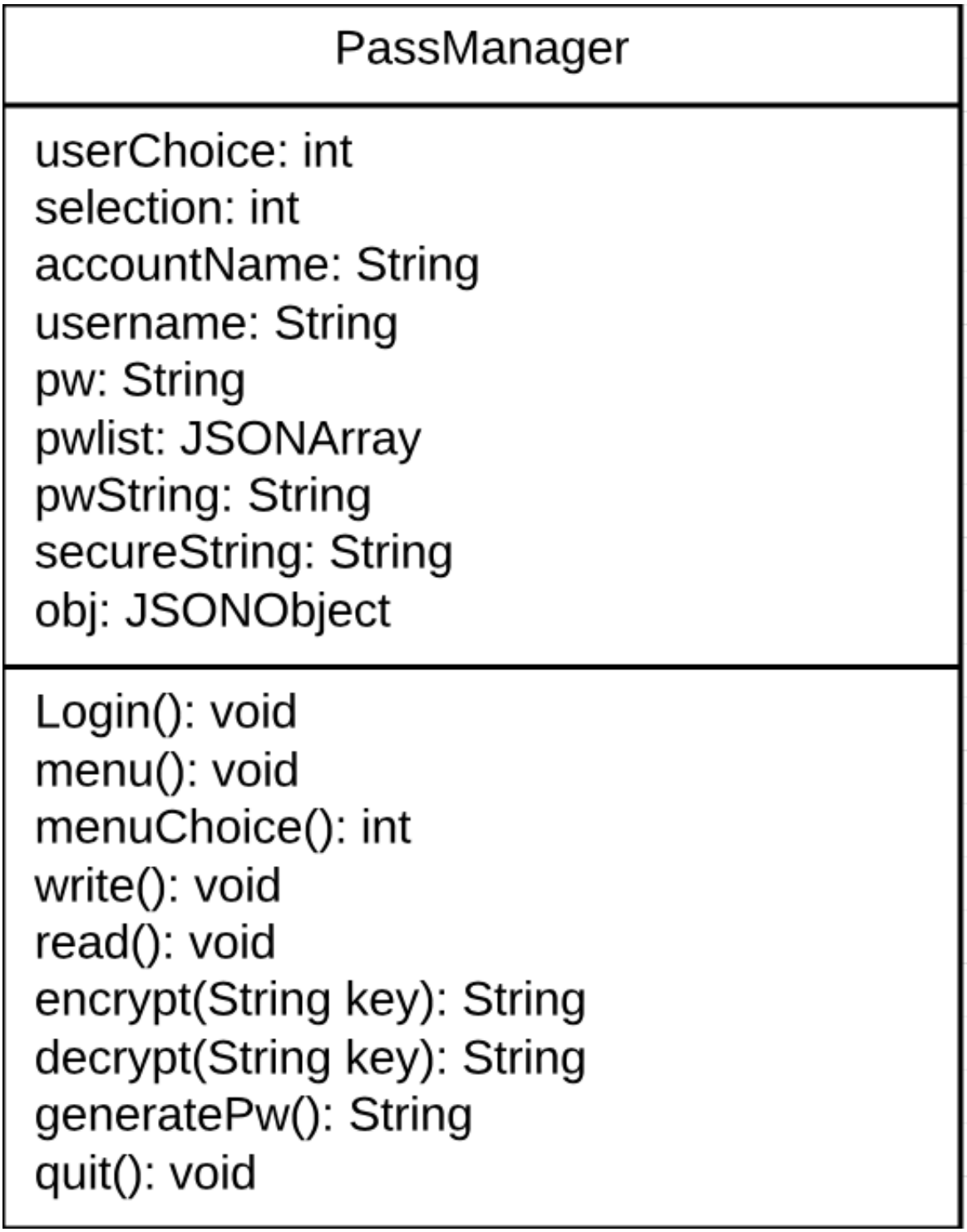
    - a random integer is generated between 0 and 60
    - the integer is used to select a single character from a string that contains all uppercase and lowercase letters as well as numbers
    - 20 Characters are chosen and added to a string for a generated password

8. If the number 4 is entered the quit method is called

    - If Accounts.JSON file exits the file will be opened
    - JSON Array is parsed and the encryption method is called
        - In the encryption method the inputed string is shifted 10 unicode
        - the result is returned

    - The result is saved to the file

UML Diagram



Requirements:

Everyday new accounts are made, average people are now required to remember many account logins and passwords and when to use them. This becomes difficult. The solution to this is to use a password manager that records all account information and passwords and then secures it with a single password. This helps in origination and allows the user to stay on top of the accounts they own

Literature Survey:

There are currently many other working applications that solve the same or similar issue, one of the most well known is 1password which offers a client, web and mobile based app. This application works very much in the same way but has a great GUI and far superior encryption.

User Manuel:

To use the java password manager first run the program and select from the list of options what action you would like to preform. press the number of the key that corresponds to the action desired. To write a new account to the file press “1” then hit enter. To view an account already saved to the file press “2” then hit enter. To generate a new password to be used for an account press “3” then hit enter. Finally to quit the program press “4” then hit enter. Once the user quits the program, the accounts file is encrypted and saved. This prevents any outside users from opening the file without the password.

Improvements

This application in its current state is basic but works well, there are many improvements that could be made.

Possible improvements inclue:

- A more advanced encryption system as the current system uses the most basic level of encryption and could be broken
- The implementation of a GUI
- Parsing the Data from the JSON Array in order to neatly display the information to the user

Conclusion:

This System in its current state is able to create a JSON file in order to save and store an infinite amount of accounts to the file. Additionally a user may go back and view the file at anytime through the program to check passwords. If a user needs a new unused password the program can generate one for them with the password generator method. Finally the program can safely ensure the data is kept secure as it encrypts the data when the program is shutdown.

References/Bibliography:

1. "Simple.JSON." Google Code Archive - Long-term storage for Google Code Project Hosting. N.p., n.d. Web. 07 Apr. 2017. https://code.google.com/archive/p/json-simple/

2. JSON.simple example – Read and write JSON. N.p., n.d. Web. 29 April 2017. https://www.mkyong.com/java/json-simple-example-read-and-write-json/.