The Information Manager

By: Rushil Desai and Yagnesh Veeraraghavan

Introduction

- Stealing someone's personal information online has become very easy and much more common in recent years.
- Our goal is to create an application using the python programming language that can successfully store a user's information in a safe space that no one else can access except for the owner
- There have been many previous attempts at trying to secure a user's information, but they all store that user's information in a place that can still be accessed by someone

Design

When designing the program, we had a couple of design requirements and constraints:

- The program needed to successfully provide a way of logging in so that the only person who could access the account was only the account owner
- The program needed to store the information in a way that a person couldn't understand it if they did get access to it through an unethical way
- The program needed to distinguish if an account was already created every time it was initialized
- The program needed to make it easy to pick a way to store the information for the user

Solution - 1

- We created a program that can make sure that information can be stored in a space that no one can access except for the owner using many different security measures and encryptions.
- We designed our program so that there were four different files that perform different functions and were used for different things. When combined they would create a well functioning program.
- We tested our program by testing each file separately to make sure they each worked as intended and then tested the whole program multiple times after the code was finished to make sure every part worked perfectly.

Solution - 2

There are four main parts (files) to the program:

- Account Creation
  - This file creates a new user account, sets up all the user info in the system, and creates a reference file for all the information
- Login
  - This file is called every time a user tries to enter the program. It makes sure that the user enters the correct login credentials that they created and verifies if the person trying to log on is actually the user. It will automatically shut down the whole system if any credential is incorrect.
- Creation
  - This file controls everything with the information. The user picks a file or creates a new one. Then they enter their information into the file and it will all be encrypted using a random generated key. This key can also be used to decrypt the file later if the user wants to look at the contents
- Main
  - This is the file that takes all the other files and combines them into one big program that runs smoothly

Account Creation - 1

- The user is automatically redirected here when they open up the program for the first time
- Gathers the user's information by asking for their name, username, password, and email.
- Sends a verification email with a 10-digit security code that needs to be inputted in order for the account to be activated
- Sets up the account by adding the information previously gathered into the system for login later
- Creates a reference file with all the user's information encrypted so no one can see the user's private details
- Initializes a random encryption key that will be used to encrypt and decrypt all the user's information when entered or when read.

Account Creation - 2

```python
username = ""
def accountCreation():
    #Imports
    import os
    import os.path
    import time
    import sys
    import smtplib
    import ssl
    import getpass
    from email.mime.text import MIMEText
    from email.mime.multipart import MIMEMultipart
    from creation import encryptedPassword
    from cryptography.fernet import Fernet

    #Initialization
    print("Thank you for choosing the Information Manager")
    print("This is a great tool for staying organized and keeping Infomation Secure")
    print("Lets get started with creating your account\n")

    #Name
    print("What is your full name?")
    name = input()

    #Username
    while True:
        print("Please type in the username that you want to use.\nYou must remember this username so it is recommended that you save it someplace where you can find it later.")
        username = input()
        if len(username) >= 8:
            print("Great username!")
            break
        else:
            print("Error: " + username + " should be greater than or equal to 8 characters")
            continue
    asterisks = ""
    for i in range(len(username)-4):
        asterisks = asterisks + "*"
    newUsername = username[0:2] + asterisks + username[len(username)-2:len(username)]

    #Password
    while True:
        print("Please type in the password that you wish to use for the account.\nYou must remember this password so it is recommended that you save it someplace where you can find it later.")
        password = input()
        if len(password) >= 8:
            print("Great password!")
            break
        else:
            print("Error: " + password + " should be greater than or equal to 8 characters")
            continue
    while True:
        print("Please confirm your password.")
        passwordTwo = input()
        if password == passwordTwo:
            break
        else:
            print("Error: Passwords do not match")
            continue
    asterisks = ""
    for i in range(len(password)-4):
        asterisks = asterisks + "*"
    newPassword = password[0:2] + asterisks + password[len(password)-2:len(password)]

    #E-mail
    while True:
        try:
            while True:
                try:
                    print("Please enter your email.")
                    email = input()
                    print("Please type in your email again.")
                    emailTwo = input()
                    if email == emailTwo:
                        break
                except:
                    print("Error: emails do not match.")
                    continue
            activation = encryptedPassword(10)
            message = MIMEMultipart("alternative")
            message["Subject"] = "Welcome to the Information Manager!"
```

```python
            message["From"] = "theinformationmanager@gmail.com"
            message["To"] = email

            plaintext = "Hello!\nThank you for choosing the information manager to keep track of all your information.\nThere's just one last step in creating your account.\nThe activation c
            htmltext = """
            <html>
              <body>
                <p style="color:rgb(0,200,255)";"font-family:'Calibri'";"font-size:12px">Hello!<br><br>
                  Thank you for choosing the information manager to keep track of all your information.
                  Please double check that all your information is correct.<br>
                </p>
                <p style="color:rgb(0,200,255)";"font-family:'Calibri'";"font-size:12px">
                   Full Name: <Strong>{fullname}</Strong><br>
                   Username: <Strong>{username}</Strong><br>
                   Password: <Strong>{password}</Strong><br>
                </p>
                <p style="color:rgb(0,200,255)";"font-family:'Calibri'";"font-size:12px">
                  There's just one last step in creating your account.<br><br>
                  The activation code to finish your account creation is below.<br>
                  Use this key to verify your account:<br>
                </p>
                <p style="font-family:'Garamond'";"font-size:20px">
                   <strong>{code}</strong><br>
                </p>
                <p style="color:rgb(0,200,255)";"font-family:'Calibri'";"font-size:12px">
                  Thank you for joining the Information Manager!<br><br>
                </p>
                <p>
                  <img src="https://europeanbusinessmagazine.com/wp-content/uploads/2020/11/0_QxsRUMTDGoTebwvF.jpg" alt="Information Manager">
                </p>
              </body>
            </html>
            """.format(code=activation,fullname=name,username=newUsername,password=newPassword)
            part1 = MIMEText(plaintext, "plain")
            part2 = MIMEText(htmltext, "html")
            message.attach(part1)
            message.attach(part2)

            server = smtplib.SMTP("smtp.gmail.com", port=587)
            context = ssl.create_default_context()
    pWord = password.encode()
    eMail = email.encode()
    key = Fernet.generate_key()
    StrKey = key.decode()
    fernetCode = Fernet(key)
    uName = fernetCode.encrypt(uName)
    uName = uName.decode()
    pWord = fernetCode.encrypt(pWord)
    pWord = pWord.decode()
    eMail = fernetCode.encrypt(eMail)
    eMail = eMail.decode()

    #Information Append
    def accountidWrite(info):
        idOpen = open("accountID.txt", "a")
        idOpen.write(info)
    os.mkdir(os.path.join(os.getcwd(),"Information Files"))
    infopath = os.path.join(os.getcwd(),"Information Files")
    idpath = os.path.join(os.getcwd(),"accountID.txt")
    accountidWrite("\n----------            ----------\nInformation Manager Account ID\n            ----------            \n\n")
    accountidWrite("Name:\n " + name + "\n\n\n")
    accountidWrite("Username:\n " + newUsername + "\n\n\n")
    accountidWrite("Password:\n " + newPassword + "\n\n\n")
    accountidWrite("E-mail:\n " + email + "\n\n\n")
    accountidWrite("Account ID File Path:\n " + idpath + "\n\n\n")
    accountidWrite("Information Files Path:\n " + infopath + "\n\n\n")
    accountidWrite("Source Code:\n\n" + StrKey + "\n\n")
    accountidWrite(uName + "\n\n")
    accountidWrite(pWord + "\n\n")
    accountidWrite(eMail + "\n\n")

    #End
    print("Congratulations!\nYou're Information Manager Account has been created!\nWhenever you need to log in, you will need your username and your password. You will get an email with a l
    sys.exit()
```

Login - 1

- The user is automatically redirected here every time the program is initialized after the first time opened
- The user must type in their username correctly in three tries or the program will automatically shut down
- The user must type in their password correctly in three tries or the program will automatically shut down
- The user will get a verification email with a 6-digit code that needs to be inputted into the program correctly in at least three times or the program will automatically shut down
- The user will automatically be redirected to the creation file if they successfully make it past all the security checks

Login - 2

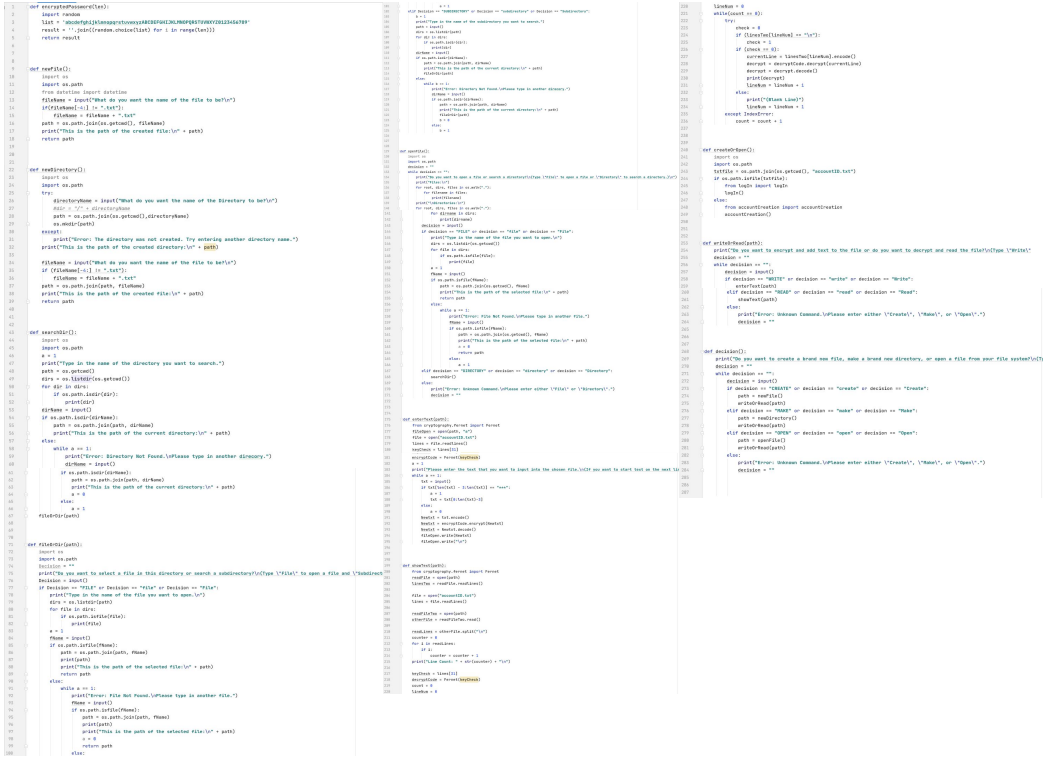
```python
def logIn():
    #Imports
    from cryptography.fernet import Fernet
    import sys
    import smtplib
    import ssl
    from email.mime.text import MIMEText
    from email.mime.multipart import MIMEMultipart
    from creation import encryptedPassword

    #Encryption Conversions
    file = open("accountID.txt")
    lines = file.readlines()
    keyCheck = lines[31]
    unameCheck = lines[33]
    pwordCheck = lines[35]
    emailcheck = lines[37]
    decryptCode = Fernet(keyCheck)
    unameCheck = unameCheck.encode()
    pwordCheck = pwordCheck.encode()
    emailcheck = emailcheck.encode()
    decryptUname = decryptCode.decrypt(unameCheck)
    decryptUname = decryptUname.decode()
    decryptPword = decryptCode.decrypt(pwordCheck)
    decryptPword = decryptPword.decode()
    decryptemail = decryptCode.decrypt(emailcheck)
    decryptemail = decryptemail.decode()
    code = encryptedPassword(6)

    #Username
    print("Welcome Back!\nType in you username to start logging in to your account.")
    uCount = 3
    while True:
        if uCount > 0:
            username = input()
            if (username == decryptUname):
                print("Great!")
                break
            else:
                print("Error: Usernames do not match.")
                if uCount != 1:
                    tries = "tries"
                else:
                    tries = "try"
                print("You have {} {} left...".format(uCount-1,tries))
                uCount = uCount - 1
        else:
            print("The username was not successfully entered in the given amount of attempts.")
            sys.exit()

    #Password
    print("Type in you password to start logging in to your account.")
    pCount = 3
    while True:
        if pCount > 0:
            password = input()
            if (password == decryptPword):
                print("Great!")
                break
            else:
                print("Error: Passwords do not match.")
                if pCount != 1:
                    tries = "tries"
                else:
                    tries = "try"
                print("You have {} {} left...".format(pCount - 1, tries))
                pCount = pCount - 1
```

```python
        else:
            print("The password was not successfully entered in the given amount of attempts.")
            sys.exit()

    #E-mail Creation
    message = MIMEMultipart("alternative")
    message["Subject"] = "Log in to Information manager!"
    message["From"] = "theinformationmanager@gmail.com"
    message["To"] = decryptemail
    plaintext = "Welcome back! Someone is trying to sign into your infomation manager account. To enter your account and access your information, enter this code into the program."
    htmltext = """
                <html>
                  <body>
                    <p style="color:rgb(0,200,255)";"font-family:'Calibri'";"font-size:12px">Hello!<br><br>
                       Welcome back!
                       Someone is trying to sign into your infomation manager account.
                       To enter your account and access your information, enter this code into the program.<br>
                    </p>
                    <p style="font-family:'Garamond'";"font-size:20px">
                        <strong>{code}</strong><br>
                    </p>
                    <p style="color:rgb(0,200,255)";"font-family:'Calibri'";"font-size:12px">
                       Thank for using the information manager.<br><br>
                    </p>
                    <p>
                       <img src="https://europeanbusinessmagazine.com/wp-content/uploads/2020/11/0_QxsWlMTDGmTebavF.jpg" alt="Information Manager">
                    </p>
                  </body>
                </html>
                """.format(code=code)
    part1 = MIMEText(plaintext, "plain")
    part2 = MIMEText(htmltext, "html")
    message.attach(part1)
    message.attach(part2)

    server = smtplib.SMTP("smtp.gmail.com", port=587)
    context = ssl.create_default_context()
    server.ehlo()
    server.starttls(context=context)
    server.starttls(context=context)
    server.ehlo()
    server.login("theinformationmanager@gmail.com", "inf0RulZ!")
    server.sendmail("theinformationmanager@gmail.com", decryptemail, message.as_string())
    server.quit()

    #E-mail
    print("Type in your 6-digit code from the email to start logging in to your account.")
    cCount = 2
    while True:
        if cCount > 0:
            Code = input()
            if (Code == code):
                print("Great!")
                break
            else:
                print("Error: The code does not match.")
                if cCount != 1:
                    tries = "tries"
                else:
                    tries = "try"
                print("You have {} {} left...".format(cCount - 1, tries))
                cCount = cCount - 1
        else:
            print("The code was not successfully entered in the given amount of attempts.")
            sys.exit()
```

Creation - 1

- The user will pick if they want to create a file, create a directory or open a file.
  - If the user wants to create a file, they will provide a file name and be redirected to the next step
  - If the user wants to create a directory, they will provide a directory name and make a new file inside that directory
  - If the user wants to open a file, they can choose to open a file in the current directory or continue to browse subdirectories until they find a file
- The user can now select if they want to add and encrypt text to the chosen file or read and decrypt it

Creation - 2



Main - 1

- Combines all the other three pieces of the program so the program flows and functions smoothly
- Decides whether the user has made an account and executes the Account Creation file if not. Otherwise, it will execute the Login file
- Executes the Creation file after so that the user can pick their file
- Redirects the user to entering or reading the text in a file after they have chosen one
- Closes and shuts down the program when the user has finished

Main - 2

```
#The Information Manager - Rushil Desai and Yagnesh Veeraraghavan
print("Welcome to the Information Manager.")
print("Here you can manage all your information in a secure and organized space")
print("If you have an account, you will need to verify yourself")
print("If you are new, You will need to create an account")
print("Created By: Rushil Desai and Yagnesh Veeraraghavan\n")
def main():
    from creation import createOrOpen
    from creation import decision
    createOrOpen()
    decision()
main()

```

Documentation

- We wanted to create a program that had multiple layers of security for logging in so we decided to make the user make a username and password as usual; on top of that though, we added a verification email service that will send an automated email with a random code to the user
- We wanted the user to have multiple options when selecting their files, so we made options for creating new files, creating new directories, opening files, and opening files in deeper subdirectories
- We wanted the text file containing information to be completely unreadable to anyone without the program so we made the program so that every time a user enters information into a file, it will automatically be encrypted using a randomly generated key and then added to the file so no one can read what the contents are.

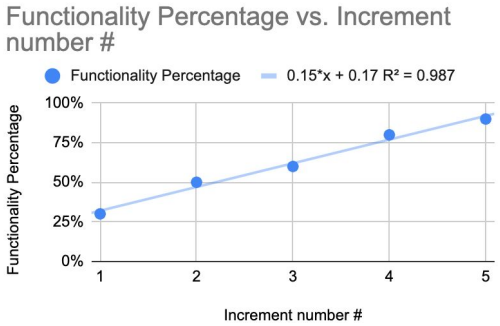
Development

- The program meets our requirement of having a secure way of logging in with a username, password, and an email security code
- The program meets our requirement of making sure no one can access files with information in them without logging in with the app because all the files are automatically encrypted into text that isn't understandable to anyone.
- Our program is better than current information manager technology because we have much more security layers and protection against stolen information. We also make the process of logging on and entering information as quick and simple as possible.

Evaluation

- Our engineering goal for the program was to make a safe and secure place where a user could store their information and no one else could access it. The program also had to be easy to use and make the process of entering information quick.
- We did meet and accomplish our engineering goal because we added three security features when logging in, made the whole process easy and fast for the user, and encrypted all the files so the information could only be accessed using our program.

Increment number #	Functionality Percentage
1	30%
2	50%
3	60%
4	80%
5	90%



Evidence - 1

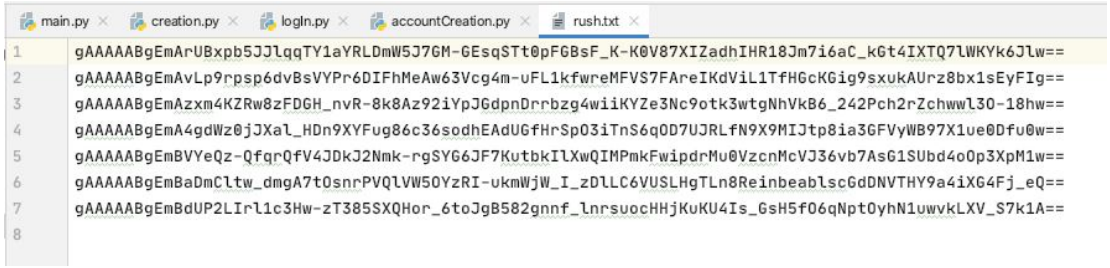

Welcome to the Information Manager!
Inbox ×
theinformationmanager@gmail.com
to me
Hello!
Thank you for choosing the information manager to keep track of all your information. Please double check that all your information is correct.
Full Name: Rushil Desai
Username: ru*******ai
Password: ru*******ai
There's just one last step in creating your account.
The activation code to finish your account creation is below.
Use this key to verify your account:
jmLXUYFIJ
Thank you for joining the Information Manager!
BIG DATA


Log in to Information manager!
Inbox ×
theinformationmanager@gmail.com
to me
Hello!
Welcome back! Someone is trying to sign into your information manager account. To enter your account and access your information, enter this code into the program:
CUpqdI
Thank for using the information manager.
BIG DATA

Evidence - 2

```
----------          ----------
Information Manager Account ID
          ----------

Name:
 Rushil Desai


Username:
 ru********ai


Password:
 ru********ai


E-mail:
 rushilcraft@gmail.com


Account ID File Path:
 /Volumes/Seagate/Programming/Python/Python Programs/Information Manager/accountID.txt


Information Files Path:
 /Volumes/Seagate/Programming/Python/Python Programs/Information Manager/Information Files


Source Code:

am0p07fI4Na4Av4OibclbmterdOBkUu2e7cYWDcXYgU=

gAAAAABf_5iU20qcFIkm6M7wLNEFdmtS7ExLYYKj_Iwi9vglSb3NAmZaV5E9VGGzJh1BiMmCGnYiNQfh-rvdsgmz47soSSQFCA==

gAAAAABf_5iUWrB0vBQEnzaJ9KbZRQm6fY-4b4qVMI79tHQtydFoKKQG2BvTTnzZJpfDGGQRHYllxF-LovJTMW-9RdnhsKiJLg==

gAAAAABf_5iUooJZKpeYVHbsgcqUYtUqPze0HVN8HgdHyx-uNwiygSIPRZcehZVSM_MuJFDFS5QwEaZIOor7bVSGNNaMoviNXUvoLJNEP-wsPN1S3ZbBLZg=
```

Evidence - 3


main.py
creation.py
logIn.py
accountCreation.py
rush.txt
1 gAAAAABgEmArUBxpb5JJlqqTY1aYRLDmW5J7GM-GEsqSTt0pFGBsF_K-K0V87XIZadhIHR18Jm7i6aC_kGt4IXTQ7lWKYk6Jlw==
2 gAAAAABgEmAvLp9rpsp6dvBsVYPr6DIFhMeAw63Vcg4m-uFL1kfwreMFVS7FAreIKdViL1TfHGcKGig9sxukAUrz8bx1sEyFIg==
3 gAAAAABgEmAzxm4KZRw8zFDGH_nvR-8k8Az92iYpJGdpnDrrbzg4wiiKYZe3Nc9otk3wtgNhVkB6_242Pch2rZchwwl3O-18hw==
4 gAAAAABgEmA4gdWz0jJXal_HDn9XYFug86c36sodhEAdUGfHrSpO3iTnS6qOD7UJRLfN9X9MIJtp8ia3GFVyWB97X1ue0Dfu0w==
5 gAAAAABgEmBVYeQz-QfqrQfV4JDkJ2Nmk-rgSYG6JF7KutbkIlXwQIMPmkFwipdrMu0VzcnMcVJ36vb7AsG1SUbd4oOp3XpM1w==
6 gAAAAABgEmBaDmCltw_dmgA7tOsnrPVQlVW5OYzRI-ukmWjW_I_zDlLC6VUSLHgTLn8ReinbeablscGdDNVTHY9a4iXG4Fj_eQ==
7 gAAAAABgEmBdUP2LIrl1c3Hw-zT385SXQHor_6toJgB582gnnf_lnrsuocHHjKuKU4Is_GsH5fO6qNptOyhN1uwvkLXV_S7k1A==
8

Evidence - 4

```
/usr/local/bin/python3.8 "/Volumes/Seagate/Programming/Python/Python Programs/Information Manager/main.py"
Welcome to the Information Manager.
Here you can manage all your information in a secure and organized space
If you have an account, you will need to verify yourself
If you are new, You will need to create an account
Created By: Rushil Desai and Yagnesh Veeraraghavan

Thank you for choosing the Information Manager
This is a great tool for staying organized and keeping Infomation Secure
Lets get started with creating your account

What is your full name?
Rushil Desai
Please type in the username that you want to use.
You must remember this username so it is recommended that you save it someplace where you can find it later.
rushildesai
Great username!
Please type in the password that you wish to use for the account.
You must remember this password so it is recommended that you save it someplace where you can find it later.
rushildesai
Great password!
Please confirm your password.
rushildesai
Please enter your email.
desai.desai.r.r@gmail.com
Please type in your email again.
desai.desai.r.r@gmail.com
An email has been sent to you with an activation code for your account.
Please enter that activation code exactly how it is displayed below.
jmtLXUYFIJ
Congratulations!
You're Information Manager Account has been created!
Whenever you need to log in, you will need your username and your password. You will get an email with a login code.
Restart the program and log in to start inputting information.
Thank you for registering and we hope this program helps you.

Process finished with exit code 0
```

```
/usr/local/bin/python3.8 "/Volumes/Seagate/Programming/Python/Python Programs/Information Manager/main.py"
Welcome to the Information Manager.
Here you can manage all your information in a secure and organized space
If you have an account, you will need to verify yourself
If you are new, You will need to create an account
Created By: Rushil Desai and Yagnesh Veeraraghavan

Welcome Back!
Type in you username to start logging in to your account.
rushildesai
Great!
Type in you password to start logging in to your account.
rushildesai
Great!
Type in your 6-digit code from the email to start logging in to your account.
ogM7Yw
Great!
```

```
Do you want to create a brand new file, make a brand new directory, or open a file from your file system?
(Type "Create" to open a file, "Make" to open a directory, or "Open" to search a directory.)
create
What do you want the name of the file to be?
rush
This is the path of the created file:
/Volumes/Seagate/Programming/Python/Python Programs/Information Manager/rush.txt
Do you want to encrypt and add text to the file or do you want to decrypt and read the file?
(Type "Write" to open a add text or "Read" to decrypt the text.
write
Please enter the text that you want to input into the chosen file.
(If you want to start text on the next line, put a "***" at the end of the current line.)
My***
name***
is***
Rushil***
I***
love***
data
```

```
Do you want to create a brand new file, make a brand new directory, or open a file from your file system?
(Type "Create" to open a file, "Make" to open a directory, or "Open" to search a directory.)
open
Do you want to open a file or search a directory?
(Type "File" to open a file or "Directory" to search a directory.)

files:

main.py
account2.txt
accountCreation.py
logIn.py
rush.txt
creation.py
accountCreation.cpython-38.pyc
main.cpython-36.pyc
logIn.cpython-38.pyc
userCheck.cpython-38.pyc
creation.cpython-38.pyc
Information Manager.iml
misc.xml
modules.xml
aws.xml
.gitignore
workspace.xml
encodings.xml
profiles_settings.xml

Directories:

__pycache__
.idea
Information Files
inspectionProfiles
File
Type in the name of the file you want to open.

main.py
account2.txt
accountCreation.py
logIn.py
rush.txt
creation.py
rush.txt
This is the path of the selected file:
/Volumes/Seagate/Programming/Python/Python Programs/Information Manager/rush.txt
Do you want to encrypt and add text to the file or do you want to decrypt and read the file?
(Type "Write" to open a add text or "Read" to decrypt the text.
read
Line Count: 7

My
name
is
Rushil
I
love
data
```

Test Demonstration

The Code

https://github.com/rush-bot/InformationManager


Thank
you