PROTECTION OF INFORMATION BASED ON SENSITIVITY AND PRIVILEGE LEVELS

CONTINUOUS ASSESSMENT FOR WEEK 6 – CS 3058

T. K. P. Perera

170446L

**Table of Content**

1. [Program Code](#Program_code)………………………………………………………………………………. 3

2. [Description](#Description)………………………………………………………………………………….. 12

3. [Annex](#Annex)……………………………………………………………………………………….. 14

[Configuration File](#Config)……………………………………………………………………… 14

[Data File](#Data)…………………………………………………………………………………14

**Program Code**

*github link :* <https://github.com/kavindaperera/computer-security-python-cmd-medical>

# module imports

import time

import configparser

import getpass

import hashlib

# ====================================================

config\_file = 'config.ini'

data\_file = 'data.ini'

# check username availability

def checkAvailability(username):

config = configparser.ConfigParser()

config.read(config\_file)

sections = config.sections()

check\_section = username

available = True

for s in sections:

if s == check\_section:

print("Username Already Taken")

available = False

return available

# verify username at login

def verifyUsername(username):

config = configparser.ConfigParser()

config.read(config\_file)

sections = config.sections()

check\_section = username

available = False

for s in sections:

if s == check\_section:

available = True

return available

# check patient detail availability

def verifyPatient(patient\_name):

config = configparser.ConfigParser()

config.read(data\_file)

sections = config.sections()

check\_section = patient\_name

available = False

for s in sections:

if s == check\_section:

available = True

return available

# verfity password at login

def verifyPassword(username, password):

config = configparser.ConfigParser()

config.read(config\_file)

en\_password = config.get(username, "password")

if (en\_password == encrypt(password)):

print("\*\*\*password verified\*\*\*")

return True

else:

print("\*\*\*incorrect password\*\*\*")

return False

# md5 encryption for passwords

def encrypt(password):

algorithm = hashlib.md5()

algorithm.update(password.encode())

encrypted = algorithm.hexdigest()

return encrypted

# session

def session(username):

print("Welcome to your account ")

config = configparser.ConfigParser()

config.read(config\_file)

privilege\_level = config.get(username, "privilege\_level")

if privilege\_level == 'admin\_0':

patientSession(username)

else:

staffSession(privilege\_level)

# staff session

def staffSession(privilege\_level):

while True:

print("Options: \n 1.View or Edit Patient Details \n 2.Create New Patient Record \n 3.logout")

option = input("> ")

if option == "3":

print("Logging out...")

time.sleep(2)

break

elif option == "1":

config = configparser.ConfigParser()

config.read(data\_file)

patient\_name = input("Enter patient name: ")

patient\_name = patient\_name + " " + "patient"

if(verifyPatient(patient\_name)):

print("options: \n 1.view details \n 2.edit details")

option = input('> ')

if option == '1':

viewSession(patient\_name, privilege\_level)

elif option == '2':

editSession(patient\_name, privilege\_level)

else:

print("No Records")

elif option == "2":

config = configparser.ConfigParser()

config.read(data\_file)

if (privilege\_level=='admin\_4'):

patient\_name = input("Enter patient name: ")

patient\_name = patient\_name + " " + "patient"

if not (verifyPatient(patient\_name)):

personal\_details = input("Enter personal details: ")

config[patient\_name] = {"personal\_details": personal\_details,

"sickness\_details": "",

"drug\_prescription": "",

"lab\_test\_prescription": ""}

with open('data.ini', 'w') as configfile:

config.write(configfile)

else:

print('Patient details already exist')

else:

print('You do not have permission to create new records')

else:

print("Not an option!")

continue

# view only session

def viewSession(patient\_name, pl):

config = configparser.ConfigParser()

config.read(data\_file)

while True:

print("View Options: \n 1.personal details \n 2.sickness details \n 3.drug prescriptions \n 4.lab test prespriptions \n 5.back")

option = input('> ')

if option == '1':

print("\n=====================")

print(config.get(patient\_name, "personal\_details"))

print("===================== \n")

continue

elif option == '2':

if (pl=='admin\_0' or pl=='admin\_1' or pl=='admin\_2' or pl=='admin\_5'):

print("\n=====================")

print(config.get(patient\_name, "sickness\_details"))

print("===================== \n")

else:

print('You do not have permission to view this section')

continue

elif option == '3':

if (pl=='admin\_0' or pl=='admin\_1' or pl=='admin\_2' or pl=='admin\_5'):

print("\n=====================")

print(config.get(patient\_name, "drug\_prescription"))

print("===================== \n")

else:

print('You do not have permission to view this section')

continue

elif option == '4':

if (pl=='admin\_0' or pl=='admin\_1' or pl=='admin\_2' or pl=='admin\_3'):

print("\n=====================")

print(config.get(patient\_name, "lab\_test\_prescription"))

print("===================== \n")

else:

print('You do not have permission to view this section')

continue

elif option == '5':

break

# edit file session

def editSession(patient\_name, pl):

config = configparser.ConfigParser()

config.read(data\_file)

while True:

print("Select a section to Edit: \n 1.personal details \n 2.sickness details \n 3.drug prescriptions \n 4.lab test prespriptions \n 5.back/save")

option = input('> ')

if option == '2':

if (pl == "admin\_1" or pl == "admin\_2"):

details\_old = config.get(patient\_name, 'sickness\_details')

print('Old Record: ', details\_old)

print("Enter New Details:")

details\_new = input('> ').strip()

mode = input("Do you want to keep old records (Y/N): ")

while True:

if (mode == 'y' or mode == 'Y'):

details = details\_old + ', ' + details\_new

break

elif (mode == 'n' or mode == 'N'):

details = details\_new

break

else:

print("(Y/N)??")

continue

config.set(patient\_name, 'sickness\_details', details)

print(config.get(patient\_name, 'sickness\_details'))

print('Updated Successfully')

else:

print('You do not have permission to edit this section')

elif option == '1':

if (pl == "admin\_4"):

details\_old = config.get(patient\_name, 'personal\_details')

print('Old Record: ', details\_old)

print("Enter New Details:")

details\_new = input('> ').strip()

mode = input("Do you want to keep old records (Y/N): ")

while True:

if (mode == 'y' or mode == 'Y'):

details = details\_old + ', ' + details\_new

break

elif (mode == 'n' or mode == 'N'):

details = details\_new

break

else:

print("(Y/N)??")

continue

config.set(patient\_name, 'personal\_details', details)

print(config.get(patient\_name, 'personal\_details'))

print('Updated Successfully')

else:

print('You do not have permission to edit this section')

elif option == '3':

if (pl == "admin\_1" or pl == "admin\_2"):

details\_old = config.get(patient\_name, 'drug\_prescription')

print('Old Record: ', details\_old)

print("Enter New Details:")

details\_new = input('> ').strip()

mode = input("Do you want to keep old records (Y/N): ")

while True:

if (mode == 'y' or mode == 'Y'):

details = details\_old + ', ' + details\_new

break

elif (mode == 'n' or mode == 'N'):

details = details\_new

break

else:

print("(Y/N)??")

continue

config.set(patient\_name, 'drug\_prescription', details)

print(config.get(patient\_name, 'drug\_prescription'))

print('Updated Successfully')

else:

print('You do not have permission to edit this section')

elif option == '4':

if (pl == "admin\_1" or pl == "admin\_2" or pl == "admin\_3"):

details\_old = config.get(patient\_name, 'lab\_test\_prescription')

print('Old Record: ', details\_old)

print("Enter New Details:")

details\_new = input('> ').strip()

mode = input("Do you want to keep old records (Y/N): ")

while True:

if (mode == 'y' or mode == 'Y'):

details = details\_old + ', ' + details\_new

break

elif (mode == 'n' or mode == 'N'):

details = details\_new

break

else:

print("(Y/N)??")

continue

config.set(patient\_name, 'lab\_test\_prescription', details)

print(config.get(patient\_name, 'lab\_test\_prescription'))

print('Updated Successfully')

else:

print('You do not have permission to edit this section')

elif option == '5':

break

config.write(open('data.ini', 'w'))

# patient session

def patientSession(username):

while True:

print("Options: \n 1.view history \n 2.logout")

option = input("> ")

if option == "2":

print("Logging out...")

time.sleep(2)

break

elif option == "1":

if(verifyPatient(username)):

viewSession(username,'admin\_0')

else:

print("No Records")

else:

print("Not an option")

#staff verification

def staffVerification(v\_code):

config = configparser.ConfigParser()

config.read(config\_file)

sections = config.sections()

return (config.get("staff verification", v\_code,fallback='1234567'))

# register

def register():

while True:

username = input("New username: ").strip()

print("Enter User Type: ")

print("Options: staff | patient")

user\_type = input("> ")

if user\_type == "patient":

privilege\_level = "admin\_0"

username = username + " " + "patient"

if not (checkAvailability(username)):

continue

break

elif user\_type == "staff":

username = username + " " + "staff"

if not (checkAvailability(username)):

continue

while True:

print("Enter provided verfication code:")

v\_code = input("> ").strip()

staff\_type = staffVerification(v\_code)

##print(staffVerification(v\_code))

user\_type = staff\_type

if staff\_type == "doctor":

print("Account type:", staff\_type)

privilege\_level = "admin\_1"

break

elif staff\_type == "nurse":

print("Account type:", staff\_type)

privilege\_level = "admin\_2"

break

elif staff\_type == "lab":

print("Account type:", staff\_type)

privilege\_level = "admin\_3"

break

elif staff\_type == "receptionist":

print("Account type:", staff\_type)

privilege\_level = "admin\_4"

break

elif staff\_type == "pharmacy":

print("Account type:", staff\_type)

privilege\_level = "admin\_5"

break

else:

print("please check the verification code..")

continue

break

else:

print(user\_type + " is not an option")

continue

while True:

password = getpass.getpass(prompt="New password: ")

if not len(password) > 0:

print("Password can't be blank")

continue

else:

confirm\_password = getpass.getpass(prompt="Confirm Password: ")

if (password == confirm\_password):

print("Passwords Matched...")

break

else:

print("Not Matching.Please Re-enter...")

continue

print("Creating account...")

en\_password = encrypt(password)

config = configparser.ConfigParser()

config.read('config.ini')

config[username] = {"password": en\_password,

"user\_type": user\_type,

"privilege\_level": privilege\_level}

with open('config.ini', 'w') as configfile:

config.write(configfile)

time.sleep(1)

print("Account has been created")

# Login

def login():

while True:

username = input("Username: ")

if not len(username) > 0:

print("Username can't be blank")

continue

else:

print("Enter User Type: ")

print("Options: staff | patient")

user\_type = input("> ")

if (user\_type == 'staff') or (user\_type == 'patient'):

username = username + " " + user\_type

if(verifyUsername(username)):

print("\*\*\*user verfied\*\*\*")

while True:

password = getpass.getpass(prompt="Password: ")

if(verifyPassword(username, password)):

session(username)

break

else:

continue

break

else:

print("invalid username or user type")

continue

else:

print(user\_type + " is not an option")

continue

# Start

print("Welcome to the Medical Data Processing System. Please Register or Login.")

while True:

print("Options: register | login | exit")

option = input("> ")

if option == "login":

login()

elif option == "register":

register()

elif option == "exit":

break

else:

print(option + " is not an option")

# Exit

print("Shutting down...")

time.sleep(1)

**Description**

There are two **user types** in the system. Users are identified using those two user types. Staff user type have 4 sub staff types: **doctor, nurse, lab, receptionist and pharmacy**. When a user is registered username, password, user type and privilege level fields are saved in the config.ini file. Format of the username is created by considering the user type. If staff user type is selected staff user needs to enter a unique verification code provided to them by the administration. Sub staff types will be assigned according to that verification code by the system. Verification details are saved in the config.ini file under [staff verification]. This will avoid patients from creating staff type accounts and staff members from creating other staff type accounts.

Only the receptionist can create a new patient data record. Receptionist only have the permission to enter personal details of the patient, when creating a new data record. Patients can view his/her medical history but cannot edit them.

***Viewing Sensitivity of data***

* Patient can view only his/her personal medical history.
* Any staff member can view the personal details on patients.
* Doctors, nurses and pharmacy staff types can view sickness details of any patient.
* Doctors, nurses and pharmacy staff types can view drug prescriptions of any patient.
* Doctors, nurses and lab staff types can view lab test prescription of any patient.

***Editing Sensitivity of data***

* Only the receptionist can edit personal details of patients.
* Only doctors and nurses can edit sickness details of patients.
* Only doctors and nurses staff types can edit drug prescriptions of patients.
* Doctor, nurse and lab staff types can edit lab test prescriptions of patients.
* Users can keep or erase previous data after editing each fields they have access to.

***Format of the usernames in config file***

* + For patient: **[username] patient**
  + For staff: **[username] staff**
  + No two staff members can have the same username.
  + No two staff patients can have the same username.

***User privilege levels***

* Patient: admin\_0
* Doctor: admin\_1
* Nurse: admin\_2
* Lab: admin\_3
* Receptionist: admin\_4
* Pharmacy: admin\_5

**Current User Accounts**

1. ***Patient\_1***
   * *Username: kavinda*
   * *Password: kavinda*
2. ***Patient\_2***
   * *Username: bruce*
   * *Password: kavinda*
3. ***Doctor***
   * *Username: kavinda*
   * *Password: kavinda*
4. ***Nurse***
   * *Username: diana*
   * *Password: kavinda*
5. ***Lab***
   * *Username: fernando*
   * *Password: fernando*
6. ***Receptionist***
   * *Username: jennifer*
   * *Password: kavinda*
7. ***Pharmacy***
   * *Username: desilva*
   * *Password: desilva*

NOTE: Run the code in command line, so the passwords will not be echoed.

**Annex**

***Configuration File***

[staff verification]

1234567 = error

adihknx = doctor

jdby7cz = nurse

udbysnx = lab

bsiyc8x = receptionist

invy6sz = pharmacy

[bruce patient]

password = 93e3b03050327fbae23c3489f0878d6f

user\_type = patient

privilege\_level = admin\_0

[kavinda patient]

password = 93e3b03050327fbae23c3489f0878d6f

user\_type = patient

privilege\_level = admin\_0

[kavinda staff]

password = 93e3b03050327fbae23c3489f0878d6f

user\_type = doctor

privilege\_level = admin\_1

[diana staff]

password = 93e3b03050327fbae23c3489f0878d6f

user\_type = nurse

privilege\_level = admin\_2

[fernando staff]

password = cebdd715d4ecaafee8f147c2e85e0754

user\_type = lab

privilege\_level = admin\_3

[jennifer staff]

password = 93e3b03050327fbae23c3489f0878d6f

user\_type = receptionist

privilege\_level = admin\_4

[desilva staff]

password = d75f8ced74e2b73f1bb22aacb8b43b6d

user\_type = pharmacy

privilege\_level = admin\_5

**Data file**

[kavinda patient]

personal\_details = name: Mr.Kavinda Perera, age: 65

sickness\_details = Dengue, Fever

drug\_prescription = Panadol

lab\_test\_prescription = FBC

[perera patient]

personal\_details = name: Mr.Perera, age: 65

sickness\_details = Fever

drug\_prescription = Panadol

lab\_test\_prescription = Liver Profile, FBR, CPR

[silva patient]

personal\_details = name: Mr.Silva, age: 26

sickness\_details = Dengue

drug\_prescription =

lab\_test\_prescription =