CS3052
Computer Security

Continuous Assessment 01

Protection of information based on sensitivity and privilege levels

MEDIHELP

Command Line Hospital Management System

Prepared by,
RANMAL J.K.D.
190507U

Table of Contents

Source Code **3**

Description **12**

Privilege value of users 12

Functionalities of users 13

Admin 13

Doctor 13

Nurse 13

Lab Assistant 13

Pharmacist 14

Patient 14

Common Features 14

Other Features and Security options 14

Other Files **15**

Current Config file(Config.csv) 15

Current Data file(data.csv) 16

1. Source Code

GitHub Link: https://github.com/dakshina99/Commandline-Medical-System---MEDIHELP

```
import hashlib
from csv import writer
import re
import pandas as pd
from itertools import islice

# helper functions

def passwordHash(password):
    return hashlib.md5(password.encode()).hexdigest()


def strongPasswordChecker(password):
    password_pattern = "^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-]).{8,}$"
    if re.match(password_pattern, password):
        return True
    else:
        return False

def reqPassword():
    print(
        "Your password must contains,\nat least 8 characters\nAt least 1\nuppercase letter
from: A-Z\n,lowercase "
        "letter from: a-z\n,number from: 0-9\n,any of the special character from: @#$%^&+=")
    password = input("Enter the Password: ").strip()
    while not strongPasswordChecker(password):
        print("Password is not enough strong!")
        password = input("Select a strong Password: ").strip()
    repassword = input("Re-Enter the Password: ").strip()
    while password != repassword:
        print("Password and Re-Password are not matching!")
        print(
            "Your password,\nMust contains at least 8 characters\nMust contains at least
1\nuppercase letters: "
            "A-Z\nlowercase letters: a-z\nnumbers: 0-9\nany of the special characters:
@#$%^&+=")
        password = input("Enter the Password: ").strip()
        while not strongPasswordChecker(password):
            print("Password is not enough strong!")
            password = input("Select a strong Password: ").strip()
        repassword = input("Re-Enter the Password Again: ").strip()
    return password, repassword


def writeUserData(filename, dataArr):
```

```
    with open(filename, 'a', newline='') as users:
        writer_object = writer(users)
        writer_object.writerow(dataArr)


def nth_index(iterable, value, n):
    matches = (idx for idx, val in enumerate(iterable) if val == value)
    return next(islice(matches, n - 1, n), None)


# ---------------------------------------------------------------------------

# User login

def login(filename):
    # read all the user data from the csv file
    df = pd.read_csv(filename)
    userData = {line[0]: list(line[1:]) for line in df.values}

    # refer global variable
    global currentUser

    usertype = ''

    # choose type
    while len(usertype) == 0:
        print("Login as a Patient     : 1\nLogin as a Staff Member : 2")
        try:
            inputVal = int(input("Enter a value: ").strip())
        except:
            print("You have entered a wrong value..! Re enter an acceptable value!!\n")
            continue
        if inputVal not in (1, 2):
            print("You have entered a wrong value..! Re enter an acceptable value!!\n")
            continue
        usertype = 'patient' if inputVal == 1 else 'staff'

    print(f"\n{'Patient' if usertype == 'patient' else 'Staff member'} Login")
    while True:
        username = input(f"Enter {'Patient' if usertype == 'patient' else 'Staff member'}
Username: ").strip()
        password = input("Enter Password: ").strip()

        if username.lower() not in userData.keys() or (
                usertype == 'staff' and userData[username.lower()][2] == 'patient') or (
                usertype == 'patient' and userData[username.lower()][2] != 'patient'):
            print('Username is invalid!\n')
            print("Want to exit? [Y/N]")
            if input().strip().lower() == 'y':
                return False
            else:
```

```
            continue
        if userData[username.lower()][1] != passwordHash(password):
            print("Password Mismatch\n")
            print("Want to exit? [Y/N]")
            if input().strip().lower() == 'y':
                return False
            else:
                continue
        if userData[username.lower()][1] == passwordHash(password):
            currentUser = [username.lower()] + userData[username.lower()]
            print('\nsuccessfully logged in!\n')
            return True


# User signup

def signUp(filename):
    # read all the user data from the csv file
    df = pd.read_csv(filename)
    userData = {line[0]: list(line[1:]) for line in df.values}

    global currentUser

    if currentUser and int(currentUser[4]) > 8:
        while (True):
            print(
                "Register a new Admin: 1\nRegister a new Doctor: 2\nRegister a new Nurse:
3\nRegister a new Lab "
                "Assistant: 4\nRegister a new Pharmacist: 5")
            type = input("Choose User Type: ")
            if type not in ('1', '2', '3', '4', '5'):
                print("You have entered a wrong value..! Re enter an acceptable value!!\n")
                print("Want to exit? [Y/N]")
                if input().strip().lower() == 'y':
                    break
                else:
                    continue
            role = ['admin', 10] if type == '1' else (['doctor', '8'] if type == '2' else (
                ['nurse', '7'] if type == '3' else (['labassistant', '6'] if type == '4' else (
                    ['pharmacist', '5'] if type == '5' else 'Invalid input!'))))
            username = input("Enter the Username: ").strip()
            while username.lower() in userData:
                print("Username Already exists..! Choose another username.")
                username = input("Enter the Username Again: ").strip()
            password, repassword = reqPassword()
            try:
                writeUserData(filename, [username.lower(), username,
passwordHash(password)] + role)
                userData[username.lower()] = [username, passwordHash(password)] + role
                print("Account successfully Registered!")
                print("Want to exit? [Y/N]")
```

```
            if input().strip().lower() == 'y':
                break
        except FileNotFoundError:
            print("Account creation is failed. Try Again!")
else:
    usertype = ""
    # choose type
    while len(usertype) == 0:
        print("Signup as a Patient      : 1\nSignup as a Staff Member : 2")
        try:
            inputVal = int(input("Enter a value: ").strip())
        except:
            print("You have entered a wrong value..! Re enter an acceptable value!!\n")
            continue
        if inputVal not in (1, 2):
            print("You have entered a wrong value..! Re enter an acceptable value!!\n")
            continue
        usertype = 'patient' if inputVal == 1 else 'staff'

    if usertype == "staff":
        print("\nStaff member Signup")

        while (True):
            code = input("Enter the Access Code: ")
            if code == "doc@reg1234":
                print("\nDoctor Signup")
                role = ['doctor', '8']
            elif code == "regnur*2234":
                print("\nNurse Signup")
                role = ['nurse', '7']
            elif code == "labassist(1212)reg":
                print("\nLab Assistant Signup")
                role = ['labassistant', '6']
            elif code == "pharm&reg12":
                print("\nPharmacist Signup")
                role = ['pharmacist', '5']
            else:
                print("\nYou have entered a wrong Access code!")
                print("Want to exit? [Y/N]")
                if input().strip().lower() == 'y':
                    break
                else:
                    continue

            username = input("Enter the Username: ").strip()
            while username.lower() in userData:
                print("Username Already exists..! Choose another username.")
                username = input("Enter the Username Again: ").strip()
            password, repassword = reqPassword()
            try:
                writeUserData(filename, [username.lower(), username,
```

```
passwordHash(password)] + role)
                userData[username.lower()] = [username, passwordHash(password)] + role
                print("Successfully Signed up!")
                break
            except FileNotFoundError:
                print("Account creation is failed. Try Again!")

        if usertype == 'patient':
            print("\nPatient Signup")

            while (True):
                username = input("Enter the Username: ").strip()
                while username.lower() in userData:
                    print("Username Already exists..! Choose another username.")
                    username = input("Enter the Username Again: ").strip()
                password, repassword = reqPassword()
                try:
                    writeUserData(filename, [username.lower(), username,
passwordHash(password), 'patient', 3])
                    userData[username.lower()] = [username, passwordHash(password), 'patient',
3]
                    print("Successfully Signed up!")
                    break
                except FileNotFoundError:
                    print("Account creation is failed. Try Again!")


# Change password
def changePassword(filename):
    # reading the csv file
    df = pd.read_csv(filename)

    userData = {line[0]: list(line[1:]) for line in df.values}

    global currentUser

    cpass = input("enter the current Password: ").strip()
    while passwordHash(cpass) != currentUser[2]:
        print("Password is wrong!\n")
        print("Want to exit? [Y/N]")
        if input().strip().lower() == 'y':
            return
        else:
            cpass = input("Enter the current password: ").strip()

    print("!!New password!!\n")
    password, repassword = reqPassword()

    # updating the column value
    df.loc[list(userData.keys()).index(currentUser[0]), 'Password'] =
passwordHash(password)
```

```
    df.to_csv(filename, index=False)

    print("Password Successfully changed!")


# View MEDI logs according to the sensitivity levels
def viewData(filename):
    global currentUser

    df = pd.read_csv(filename)
    data = {line[0]: [] for line in df.values}
    for line in df.values:
        data[line[0]] += [list(line[1:])]
    if int(currentUser[4]) > 4:
        patient = input("Enter patient username: ").strip()
        while patient not in data:
            print("Invalid Patient details or no logs!\n")
            print("Want to exit? [Y/N]")
            if input().strip().lower() == 'y':
                return ''
            else:
                patient = input("Enter patient username: ").strip()
        print(f"patient:- {patient}\n")

        if int(currentUser[4]) == 8:
            n = 1
            for line in data[patient]:
                print(
                    f"({n})\nSickness Details:- {line[0]}\nDrug Prescription:- {line[1]}\nLab Test
Prescription:- {line[2]}\nChanneled Doctor:- {line[3]}\n")
                n += 1
            return patient
        elif int(currentUser[4]) == 7:
            n = 1
            for line in data[patient]:
                print(
                    f"({n})\nSickness Details:- {line[0]}\nDrug Prescription:- {line[1]}\nLab Test
Prescription:- {line[2]}\n")
                n += 1
            return patient
        elif int(currentUser[4]) == 6:
            n = 1
            for line in data[patient]:
                print(f"({n})\nLab Test Prescription:- {line[2]}\n")
                n += 1
        elif int(currentUser[4]) == 5:
            n = 1
            for line in data[patient]:
                print(f"({n})\nDrug Prescription:- {line[1]}\n")
                n += 1
```

```
    else:
        n = 1
        for line in data[currentUser[1]]:
            print(
                f"({n})\nSickness Details:- {line[0]}\nDrug Prescription:- {line[1]}\nLab Test
Prescription:- {line[2]}\nChanneled Doctor:- {line[3]}\n")
            n += 1
    return ''


def viewCons(filename):
    global currentUser

    df = pd.read_csv(filename)
    data = {line[4]: [] for line in df.values}
    for line in df.values:
        data[line[4]] += [list(line[:4])]

    if currentUser[1] not in data:
        print("No consultations!\n")
    else:
        for line in data[currentUser[1]]:
            print(
                f"Patient:- {line[0]}\nSickness Details:- {line[1]}\nDrug Prescription:-
{line[2]}\nLab Test Prescription:- {line[3]}\n")


def addLog(filename, filename2):
    df = pd.read_csv(filename)
    dff = pd.read_csv(filename2)

    data = {line[0]: [] for line in df.values}
    for line in df.values:
        data[line[0]] += [list(line[1:])]

    userData = {line[1]: list(line[1:]) for line in dff.values}

    patient = input("Enter patient username: ").strip()
    while patient not in userData:
        print("Invalid Patient details!\n")
        print("Want to exit? [Y/N]")
        if input().strip().lower() == 'y':
            return
        else:
            patient = input("Enter patient username: ").strip()

    wp = [patient]
    if int(currentUser[4]) > 7:
        wp.append(input("Enter Sickness details: ").strip())
        wp.append(input("Enter Drug Prescription: ").strip())
        wp.append(input("Enter Lab Test Prescription: ").strip())
```

```
        wp.append(currentUser[1])
    elif int(currentUser[4]) == 7:
        wp.append(input("Enter Sickness details: ").strip())
        wp.append('')
        wp.append('')
        wp.append(input("Enter channeled doctor username: ").strip())

    with open(filename, 'a', newline='') as data:
        writer_object = writer(data)
        writer_object.writerow(wp)
    print("successfully logged!")


def editLog(filename, patient):
    # reading the csv file
    df = pd.read_csv(filename)
    dataset = list(list(zip(*list(map(list, df.values))))[0])
    data = {line[0]: [] for line in df.values}
    for line in df.values:
        data[line[0]] += [list(line[1:])]

    global currentUser

    num = input("select the number of the log to edit: ").strip()

    while len(data[patient]) < int(num) or int(num) < 1:
        print("Entered value is wrong!")
        print("Want to exit? [Y/N]")
        if input().strip().lower() == 'y':
            return
        else:
            num = input("select the number of the log to edit: ").strip()

    if int(currentUser[4]) == 8:
        while True:
            print("Change Sickness details: 1\nChange Drug Prescription: 2\nChange Lab Test
prescription: 3\n")
            nu = input("select a number: ").strip()
            while int(nu) > 3 or int(nu) < 1:
                print("Entered value is wrong!")
                print("Want to exit? [Y/N]")
                if input().strip().lower() == 'y':
                    return
                else:
                    nu = input("select a number: ").strip()
            if nu == '1':
                sd = input("Enter New Sickness details: ").strip()
                df.loc[nth_index(dataset, patient, int(num)), "sickness_details"] = sd
            elif nu == '2':
                sd = input("Enter New Drug Prescription: ").strip()
                df.loc[nth_index(dataset, patient, int(num)), "drug_prescription"] = sd
```

```
        elif nu == '3':
            sd = input("Enter New Lab Test Prescription: ").strip()
            df.loc[nth_index(dataset, patient, int(num)), "lab_prescription"] = sd
        print("Log successfully edited!")
        print("Want to exit? [Y/N]")
        if input().strip().lower() == 'y':
            break
    elif int(currentUser[4]) == 7:
        print("Change Sickness details: 1\n")
        nu = input("select a number: ").strip()
        while int(nu) != 1:
            print("Entered value is wrong!")
            print("Want to exit? [Y/N]")
            if input().strip().lower() == 'y':
                return
            else:
                nu = input("select a number: ").strip()
        if nu == '1':
            sd = input("Enter New Sickness details: ").strip()
            df.loc[nth_index(dataset, patient, int(num)), "sickness_details"] = sd

    df.to_csv(filename, index=False)


# globals
currentUser = []
userDataFile = 'config.csv'
dataFile = 'data.csv'

print("Greetings from MEDIHELP!\n")

while True:
    print("Login to the system: 1\nRegister to the system: 2\nExit: 99\n")
    val = input("Enter the value: ")
    if val == '1':
        val = login(userDataFile)
        while val:
            print(f"\nUsername:- {currentUser[1]}\n")

            if int(currentUser[4]) > 8:
                print("Register a staff member: rm")
            if int(currentUser[4]) == 8:
                print("View my consultations: mc")
            if int(currentUser[4]) > 6:
                print("Add new medical log: al")
            if int(currentUser[4]) != 10:
                print("View Medical Details: vm")
            print("Change Password: cp\nLogout: lo")

            val1 = input("Enter the value: ")
```

```
        if val1 == 'cp':
            changePassword(userDataFile)
        if val1 == 'rm' and int(currentUser[4]) > 8:
            signUp(userDataFile)
        if val1 == 'mc' and int(currentUser[4]) == 8:
            viewCons(dataFile)
        if val1 == 'al' and int(currentUser[4]) > 6:
            addLog(dataFile, userDataFile)
        if val1 == 'vm' and int(currentUser[4]) != 10:
            patient = viewData(dataFile)
            if patient:
                print("Want to edit a log? [Y/N]")
                if input().strip().lower() == 'y':
                    editLog(dataFile, patient)
        if val1.lower() == 'lo':
            currentUser = []
            break

    elif val == '2':
        signUp(userDataFile)
    elif val == '99':
        print("Bye!! Have a nice day!")
        break
```

2. Description

This is a digital hospital system named MEDIHELP, which has two main user types. They are, “Patient” and “Staff”. There are 5 different staff member types in the system. Each user type has a unique privilege value and they can access different types of functions depending on their privilege values.

Privilege value of users

User Type	Privilege value
Admin	10
Doctor	8
Nurse	7
Lab Assistant	6
Pharmacist	5
Patient	3

Table 01

Functionalities of users

Different types of users have different functionalities in this system. High-privileged users have access to highly sensitive methods. All the functionalities related to the users are shown below.

1. Admin
    - Admin can log in to the system as a staff member. Only an admin can register another admin to the system. Admin has the highest privilege value in this system. MEDIHELP admin has privileges to register any new staff member to the system.
2. Doctor
    - Doctor has the second highest privilege value in this system. The doctor can access any medical detail in this system. Hence, the doctor is the highest privileged medical officer in this system.
    - A doctor can register in the system by himself/herself or the admin can register a doctor in the system. If a doctor wants to register in this system as a doctor, he/she needs to enter the appropriate access code in the signup portal(table 02).
    - A doctor can log in to the system as a staff member by entering the username and password.
    - The doctor can view all the medical logs of a patient by entering the patient's username. All the sickness details, drug prescriptions, lab test prescriptions, and consulted doctors can be accessed by the doctor. A doctor can edit the sickness details, drug prescription, or lab test prescription of an already existing medical log.
    - The doctor has a functionality to view their consultation history through the system.
3. Nurse
    - The nurse has a higher privilege level than the lab assistant and a lower privilege level than the doctor.
    - Nurses can sign up in the system by themselves or the admin can register nurses to the system. Nurses need to enter the appropriate access code when they sign up in the system(table 02).
    - Nurses can view the medical history of a patient by entering the patient’s username. But, they can view only the sickness details, drug prescriptions, and lab test prescriptions. Nurses do not have the privilege to view the doctor's details in a medical log.
    - Nurses can edit the sickness details of an already existing medical log. Any other details of a medical log cannot be modified by the nurses.
4. Lab Assistant
    - The lab assistant has a higher privilege level than the pharmacist and a lower privilege level than the nurse.
    - Lab assistants can sign up in the system by themselves or the admin can register lab assistants to the system. Lab assistants need to enter the appropriate access code when they sign up in the system(table 2).
    - Lab assistants can view the medical history of a patient by entering the patient’s username. But, they can view only the lab test prescriptions. Lab assistants do not have the privilege to view the other details in a medical log. They cannot modify any medical log.

5. Pharmacist
    - The pharmacist has a higher privilege level than the patient and a lower privilege level than the lab assistant.
    - Pharmacists can sign up in the system by themselves or the admin can register lab assistants to the system. Pharmacists need to enter the appropriate access code when they sign up in the system(table 2).
    - Pharmacists can view the medical history of a patient by entering the patient's username. But, they can view only the drug prescriptions. Pharmacists do not have the privilege to view the other details in a medical log. They cannot modify any medical log.
6. Patient
    - The patient has the lowest privilege level in this system.
    - Patients can only sign up in the system by themselves. Patients need to add their personal details to the system when they log in to the system.
    - Patients can view their medical history. All the sickness details, drug prescriptions, lab test prescriptions, and consulted doctors can be accessed by the patient of their medical logs.
7. Common Features
    - All the users can change their login passwords in the system.
    - Users can log out from the system. All the global variables are cleared after they log out from the system.

User Type	Access code
Doctor	doc@reg1234
Nurse	regnur*2234
Lab Assistant	labassist(1212)reg
Pharmacist	pharm&reg12

Table 02

Other Features and Security options

- All the usernames must be unique. Neither of them can have the same values. This is validated at the user signup.
- All the passwords must contain,
    - At least 8 characters
    - At least 1 uppercase letter(A-Z)
    - At least 1 lowercase letter(a-z)
    - At least 1 number(0-9)
    - any of the special character from @#$%^&+="
- All the passwords are hashed using "md5(Message Digest Method 5)".
- All the inputs are sanitized and wrong inputs are handled.

3. Other Files

Current Config file(Config.csv)

usernameLower	Username	Password	Type	privilege Level
admin	admin	251800da8d338eb82819105d5f3c7629	admin	10
testpatient1	TestPatient1	251800da8d338eb82819105d5f3c7629	patient	3
testpatient2	TestPatient2	251800da8d338eb82819105d5f3c7629	patient	3
testdoctor1	TestDoctor1	251800da8d338eb82819105d5f3c7629	doctor	8
testdoctor2	TestDoctor2	251800da8d338eb82819105d5f3c7629	doctor	8
testnurse1	TestNurse1	251800da8d338eb82819105d5f3c7629	nurse	7
testnurse2	TestNurse2	251800da8d338eb82819105d5f3c7629	nurse	7
testlabassist1	TestLabAssist1	251800da8d338eb82819105d5f3c7629	labassistant	6
testlabassist2	TestLabAssist2	251800da8d338eb82819105d5f3c7629	labassistant	6
testpharm1	TestPharm1	251800da8d338eb82819105d5f3c7629	pharmacist	5
testpharm2	TestPharm2	251800da8d338eb82819105d5f3c7629	pharmacist	5

Current Data file(data.csv)

patient	sickness_details	drug_prescription	lab_prescription	doctor
TestPatient 1	COVID 19	Pandol	PCR Test	TestDoctor 1
TestPatient 2	Dengue	Samahan	Full Count Blood Test	TestDoctor 1
TestPatient 2	Chickenpox	Sitavig, Zovirax	Whole infected cell test	TestDoctor 2