AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH

PROJECT*: COMPUTER LAB MANAGEMENT SYSTEM*

Course: Programming with Python

Section: [B]

Date of submission: 01/03/2023

Faulty: Dr. Akinul Islam Jony

Semester: Spring 2022-2023

Submitted by:

Name: Tanvir Chowdhury

ID: 20-42699-1

Project Overview:

This is a computer lab management system coded in python. It is a console-based application that records and keep track of the computer information in a lab. We can register a PC into the system by inserting information such as PC number, PC operating system, PC status. Once data are inserted in the system, we can also update the existing PC information. We may also remove any PC information form the system. The user can also choose to save all the Computer details in a text file to save it on the Hard drive for later use.

This system has multiple editing functionality and user-friendly menu.

Project Solution Design:

START

Loop:

- Show Main Menu Options
- Take user input to select an option
- Press 1 = Call add_pc() function

  (The user input of PC number is checked by= **check_pc()** function. If the PC number already exists the program prompt with options to update the PC number, Remove the PC, Cancel)
- Press 2 = Call search_pc function

  (The search function will search the List for given PC number)
- Press 3 = Call update_pc() function

  (It will check the list with the given PC number and allow the user to update details)
- Press 4= Call remove_pc() function

  (It will match the user input PC number and delete that PC from the system)
- Press 5= Call show_all_pc() function

  (It will fetch the all_pc List and display the details of all PC)
- Press 6= Call save_file() function

  (It will save the contents of the List into a text file in the project directory)
- Press 7= Call quit() function

  (terminates the program)

END

IMPLEMENTATION:

class_module.py :

```
class PC:
    all_pc = [] # stores all instances of the class

    def __init__(self, pc_num, pc_os, pc_status):
        self.number = pc_num
        self.os = pc_os
        self.status = pc_status

        PC.all_pc.append(self)
        print(f"\n New PC Registered\n")  #Adds to the list

    def add_pc(self):  #Adding  New Computer
        print("Enter new PC details:-\n")
        pc_num = input("PC number: ")
        pc_os = input("PC operating system: ")
        pc_status = input("PC status: ")

        pc_exists = PC.check_pc(pc_num)  # check for duplicates

        if pc_exists == 0:
            new_pc = PC(pc_num, pc_os, pc_status)  #Adding PC object to the list
object


        else:
            print("\n PC with same number already exists\n")
            print("To update pc number - press '1'")
            print("TO remove pc - press '2'")
            print("TO cancel - press '3'")

            i = input("\n Waiting for input: ")

            if i == '1':
                new_pc_num = input("\n New pc number: ")
                PC.update_pc(pc_exists, new_pc_num, pc_exists.os,
                                       pc_exists.status)
                new_pc = PC(pc_num, pc_os, pc_status)
            elif i == '2':
                PC.remove_pc(pc_exists)
                new_pc = PC(pc_num, pc_os, pc_stat)
            elif i == '3':
                PC.Main()
            else:
                PC.Main()

    def remove_pc(computer):  #Delete PC
        print(f"\n {computer.number} is deleted")
        PC.all_pc.remove(computer)

    def update_pc(computer, new_pc_num, new_pc_os, new_pc_status):  #Update
Information
        if PC.check_pc(new_pc_num) == 0:
            old_pc_num = computer.number
            computer.number = new_pc_num
            computer.os = new_pc_os
            computer.status = new_pc_status
            print(f"PC {old_pc_num} has been updated\n")
        else:
            print(f"\n The number {computer.number} is already registered\n")

    def check_pc(pc_num):  #Checking whether Pc number is unique
        flag = 1
        for computer in PC.all_pc:
            if computer.number == pc_num:
                flag = 0
                break
        if flag == 0:
            return computer
        else:
            return 0

    def show_all_pc():  #Display all computer information
        if len(PC.all_pc) != 0:
            print("All PC details:-\n")
            for computer in PC.all_pc:
                print(f"PC number: {computer.number}")
                print(f"PC Operating System: {computer.os}")
                print(f"PC Status: {computer.status}\n")
        else:
            print("No PC available")

    def search_pc(pc_num):  # Search for PC number
        flag = 0
        for computer in PC.all_pc:
            if computer.number == pc_num:
                flag = 1
                print("Search results:-\n")
                print(f"PC number: {computer.number}")
                print(f"PC operating System: {computer.os}")
                print(f"PC status: {computer.status}")
                pc_exists = computer
                break
        if flag == 1:
            return pc_exists
        else:
            print(f"No PC found with PC number {pc_num}")
            return 0

    def save_file():  #Saves the Data in the Directory
        try:
            file = "ALL_PC.txt"
            with open(file, 'w') as file_obj:
                for computer in PC.all_pc:

file_obj.write(f"{computer.number},{computer.os},{computer.status}\n")
            print("\nAll PC information has been saved in the file")

        except Exception as e :
            print("\n File saving error!")
```

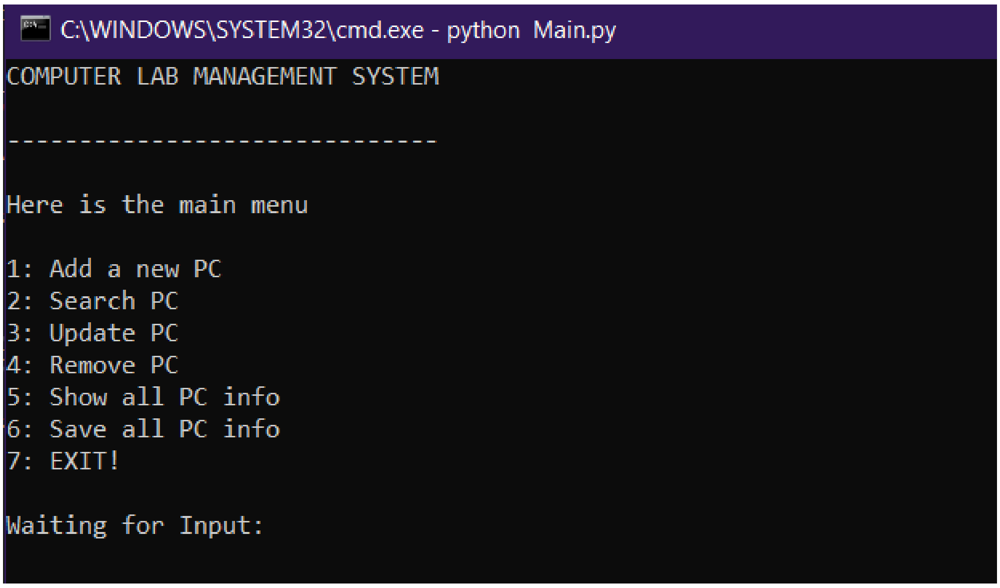
main.py :

```
from class_module import PC  # Import Module

while 1>0:
    print("COMPUTER LAB MANAGEMENT SYSTEM \n")
    print("------------------------------ \n")
    print("Here is the main menu\n")
    print("1: Add a new PC")
    print("2: Search PC")
    print("3: Update PC")
    print("4: Remove PC")
    print("5: Show all PC info")
    print("6: Save all PC info")
    print("7: EXIT!\n")

    choice = input("Waiting for Input: \n")

    if choice == '1':
        PC.add_pc(PC)

    elif choice == '2':
        n = input("Enter PC number to search: ")
        PC.search_pc(n)

    elif choice == '3':
        PC.show_all_pc()
        n = input("Enter PC number to be updated: ")

        pc_exists = PC.search_pc(n)
        if pc_exists != 0:
            print(f"\nFor {n} selected PC")
            new_pc_num = input("\nEnter new PC number: ")
            new_pc_os = input("\Enter new PC'S operating system: ")
            new_pc_stat = input("\Enter new PC'S status: ")
            PC.update_pc(pc_exists, new_pc_num, new_pc_os, new_pc_stat)

    elif choice == '4':
        pc_num = input("\nEnter PC number to remove: ")
        pc_exists = PC.search_pc(pc_num)
        if pc_exists != 0:
            PC.remove_pc(pc_exists)

    elif choice == '5':
        PC.show_all_pc()

    elif choice == '6':
        PC.save_file()

    elif choice == '7':
        quit()
```

APPLICATION OVERVIEW:

Main menu:

This is the startup menu which shows the executable features of the application

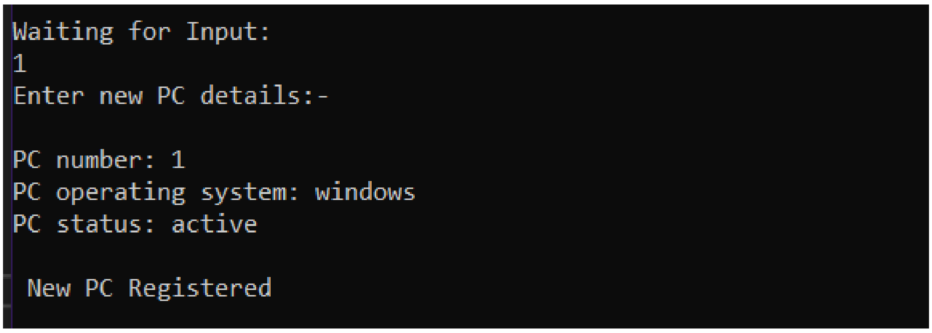


C:\WINDOWS\SYSTEM32\cmd.exe - python Main.py

```
COMPUTER LAB MANAGEMENT SYSTEM

------------------------------

Here is the main menu

1: Add a new PC
2: Search PC
3: Update PC
4: Remove PC
5: Show all PC info
6: Save all PC info
7: EXIT!

Waiting for Input:
```

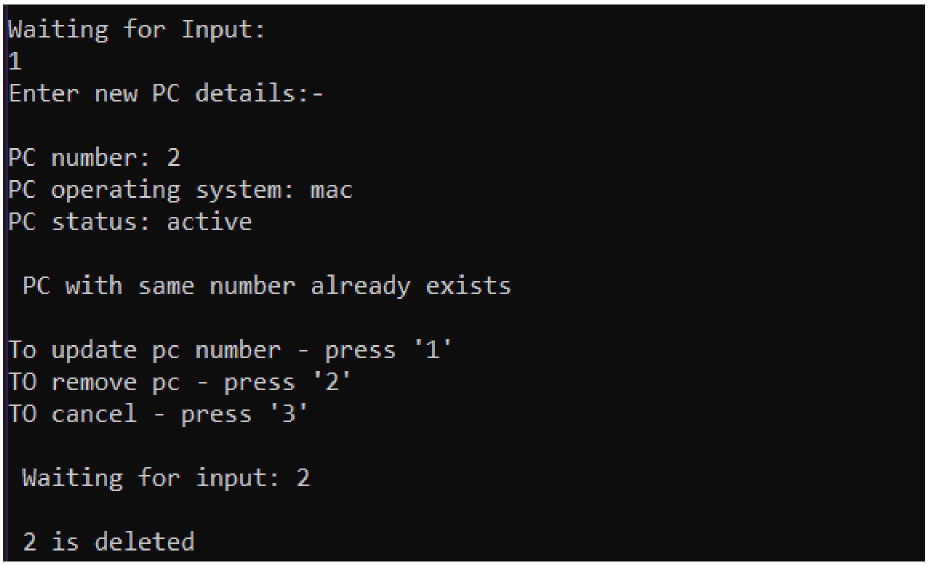
1. Adding a new PC:

The user can add a new PC into the system by inserting information such as, PC number, PC Operating system and PC status



```
Waiting for Input:
1
Enter new PC details:-

PC number: 1
PC operating system: windows
PC status: active

 New PC Registered
```

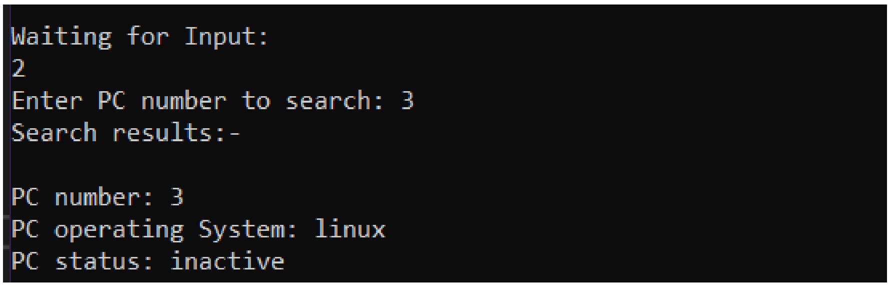
If the user wants to add a PC with already existing PC number. The program will prompt with options to update that PC number or remove that PC from the system



```
Waiting for Input:
1
Enter new PC details:-

PC number: 2
PC operating system: mac
PC status: active

 PC with same number already exists

To update pc number - press '1'
TO remove pc - press '2'
TO cancel - press '3'

 Waiting for input: 2

 2 is deleted
```

2. Search for PC:

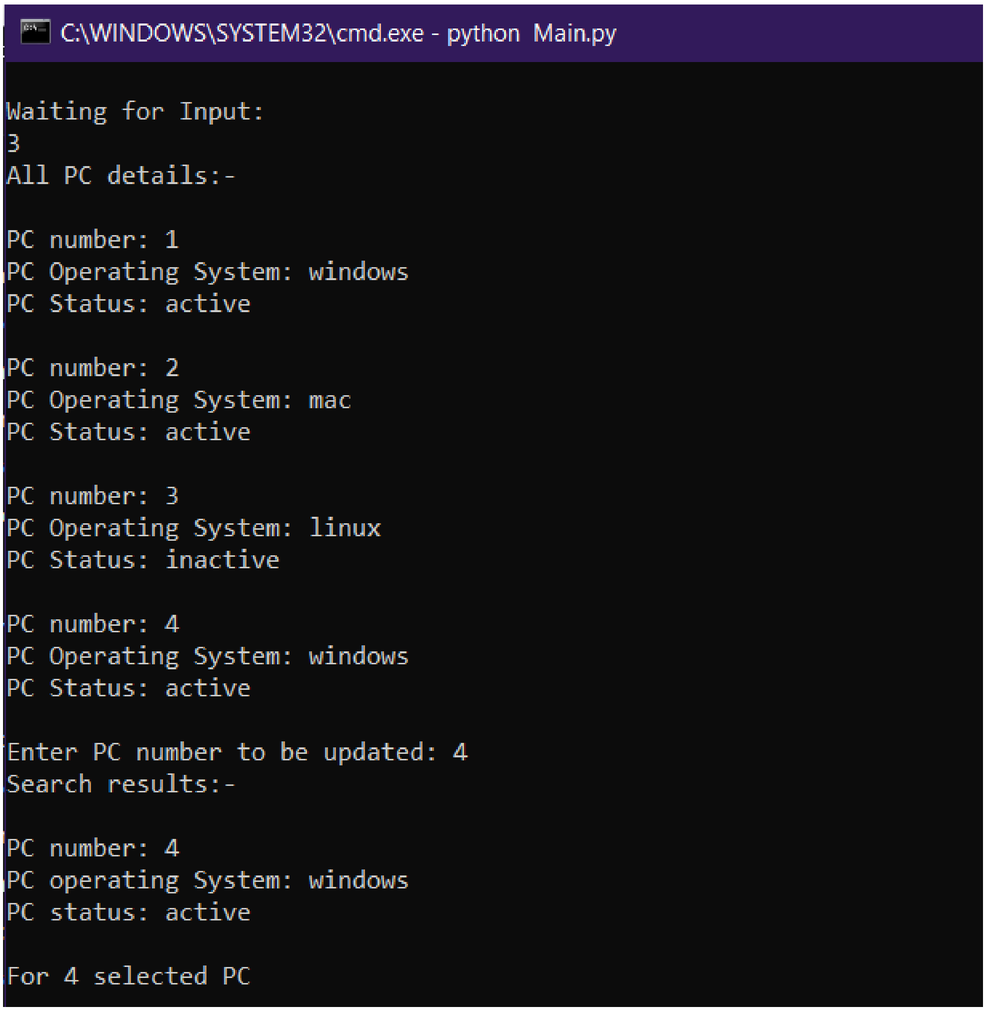
As the details of all computers are kept in a list as objects, user can search for specific PC by the PC number. The system implements linear search to look for the requested PC



```
Waiting for Input:
2
Enter PC number to search: 3
Search results:-

PC number: 3
PC operating System: linux
PC status: inactive
```

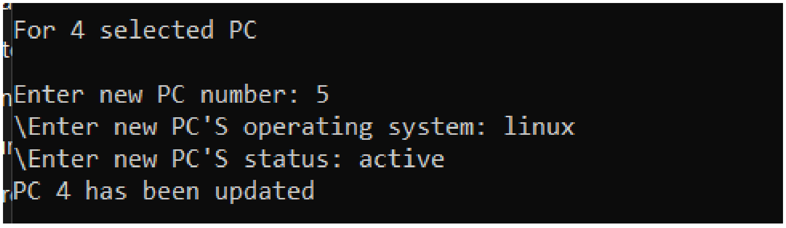
3. Updating PC details

User can update the existing PC details in the system, the prompt will show all available PC and can choose the which one to update by the PC number.User can update and save the details of that PC



C:\WINDOWS\SYSTEM32\cmd.exe - python Main.py

```
Waiting for Input:
3
All PC details:-

PC number: 1
PC Operating System: windows
PC Status: active

PC number: 2
PC Operating System: mac
PC Status: active

PC number: 3
PC Operating System: linux
PC Status: inactive

PC number: 4
PC Operating System: windows
PC Status: active

Enter PC number to be updated: 4
Search results:-

PC number: 4
PC operating System: windows
PC status: active

For 4 selected PC
```

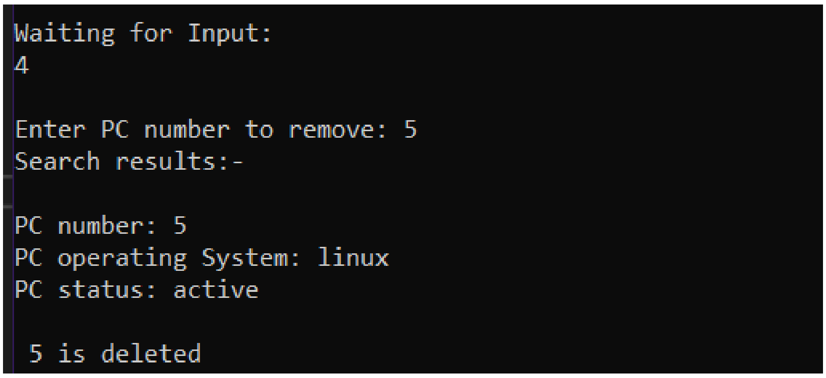


```
For 4 selected PC

Enter new PC number: 5
\Enter new PC'S operating system: linux
\Enter new PC'S status: active
PC 4 has been updated
```

The selected PC gets updated.

4. **Remove PC**

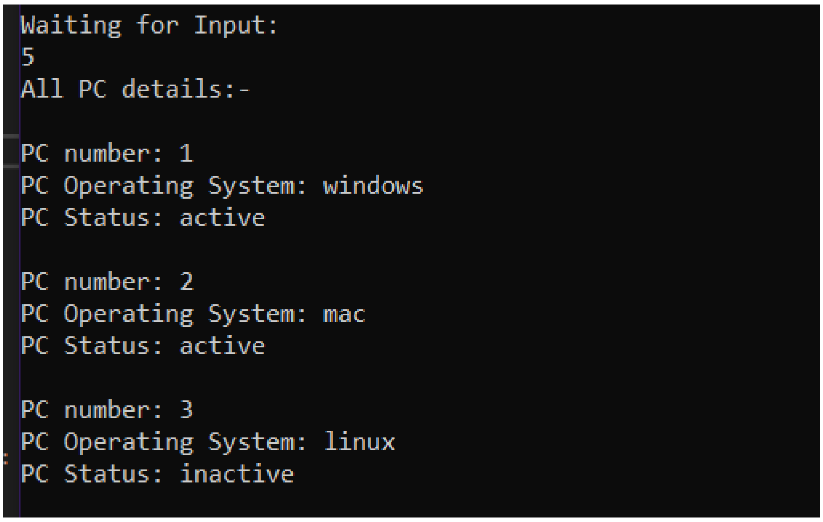
The user call also remove existing PC from the system by the Remove functionality



```
Waiting for Input:
4

Enter PC number to remove: 5
Search results:-

PC number: 5
PC operating System: linux
PC status: active

 5 is deleted
```

5. **Display all PC details:**

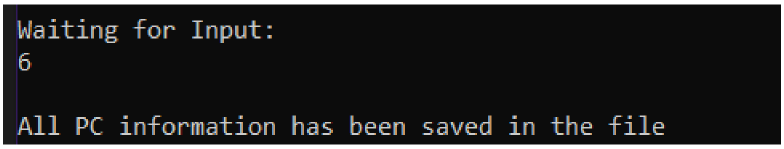
This function shows the detailed list of all the PC registered in the system



```
Waiting for Input:
5
All PC details:-

PC number: 1
PC Operating System: windows
PC Status: active

PC number: 2
PC Operating System: mac
PC Status: active

PC number: 3
PC Operating System: linux
PC Status: inactive
```

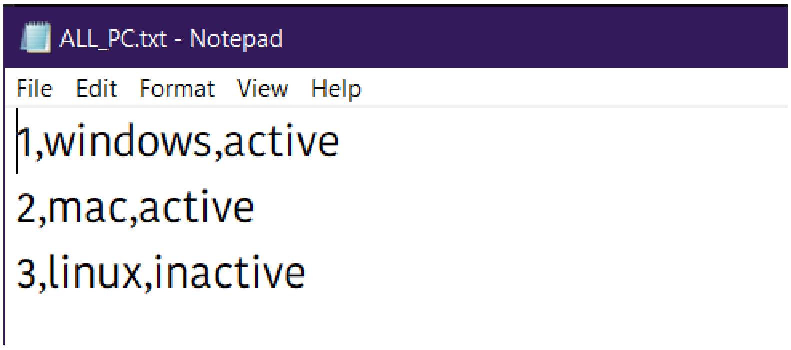
6. **Save all PC details in a text file**

This function saves the data on the hard drive in a text file : ALL_PC.txt



```
Waiting for Input:
6

All PC information has been saved in the file
```

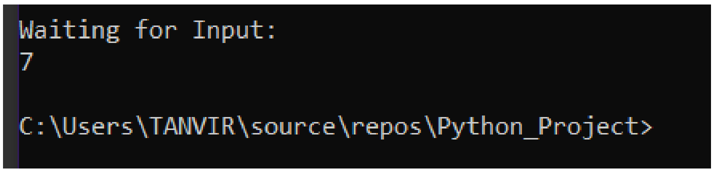
The file is stored in the Project Directory



```
1,windows,active
2,mac,active
3,linux,inactive
```

7. **Exit:**

This option stop the execution of the program



```
Waiting for Input:
7

C:\Users\TANVIR\source\repos\Python_Project>
```