Project Report

Project Overview:

The goal of the project is to create a console-based application that allows users to maintain a record of all the PCs in a lab. The application will include information about each PC, including its PC number, the operating system installed on it, and its status. Users will be able to add new PCs to the lab, update information about existing PCs, and remove PCs from the lab.

The application will include several key features. The first is the ability to display information about all the PCs in the lab. Users will also be able to display information about a particular PC by entering its PC number. In addition, the application will include a search functionality that allows users to search for a particular PC and display its information. If the PC is not already in the application, the user will be prompted to add it to the lab.

To ensure that each PC is properly accounted for, the application will include a checking functionality that prevents the user from adding a PC with the same PC number as an existing PC. If the user attempts to add a PC with a duplicate PC number, the application will prompt the user to modify the information of the existing PC or remove it from the lab.

Finally, the application will include a store functionality that allows users to store all the PC information in a text file if they want to maintain a physical copy on their hard drive.

The application will be organized around a menu that displays all the available options, and users will be able to exit the application by selecting the "quit" option. To ensure the code is well-organized and follows best practices, it will be written with a clear structure and proper documentation throughout.

Project Solution Design:

Here is how the project was designed to make the application:

1. **Selecting data structure to store file**: The initial step was to determine the data structure that will store info about all the PCs in the lab. Dictionary was a ideal option for that. In the dictionary, named LAB_PCs, key was PC number and value was a list containing operating system and status.
2. **Menu**: After running the program, it will show user a menu where every option will be available and user can select any one of them at once. After finishing each task users can go back to menu.
3. **Adding Functionality**: To add a new PC, the application will prompt the user to enter the PC number, operating system, and status. Before adding the PC, the application will check whether the PC number is already in the dictionary. If it is, the application will prompt the user to modify the existing PC or remove it from the lab. If not, the new PC will be added to the dictionary.

4. **Updating Functionality**: To update information for an existing PC, the user will be prompted to enter the PC number and the new information (operating system and/or status). The application will then update the information in the dictionary.
5. **Removing Functionality**: To remove an existing PC, the user will be prompted to enter the PC number. The application will then remove the PC from the dictionary.
6. **All PC Display Functionality**: To display information about all the PCs in the lab, the application will iterate over the dictionary and display the PC number, operating system, and status for each PC.
7. **Individual PC Display Functionality**: To display all the information for a particular PC, the user will be prompted to enter the PC number. The application will then display the operating system and status for that PC.
8. **Search Functionality**: To search for a particular PC, the user will be prompted to enter the PC number. If the PC is in the dictionary, the application will display its information. If not, the user will be prompted to add the new PC to the lab.
9. **Store Functionality**: To store all the PC information in a text file, the user will be prompted to enter a file name. The application will then write the dictionary to the file.
10. **Usage of docstring**: Docstrings were used in the first line of each method to understand what the method was doing
11. **Back to menu**: Each method has back to menu option so that users can go back to menu anytime after one particular task is done doing.

Implementation:

LabpcManagement.py

```
class PC :


   def __init__(self,LAB_PCs) :

      """This method is called whenever an object of the class is created"""

      self.LAB_PCs=LAB_PCs


   def update_pc(self,pc_no):

      """This is for updating PC info such as os,status"""


      if pc_no not in self.LAB_PCs:

         print("PC number not found.")

      else:
```

```
        os = input("Installed operating system in the PC: ")
        status = input("Status: ")
        self.LAB_PCs[pc_no]["os"] = os
        self.LAB_PCs[pc_no]["status"] = status
        print("PC information updated successfully.")

    m=int(input("\n\nPress 0 to go back to Menu: "))
    if m==0:
        self.main()

def add_pc(self,pc_no=""):
    """This method is to add new PC in the lab"""
    if pc_no=="":
        pc_no = input("PC no: ")
    if pc_no in self.LAB_PCs:
        print("PC number already exists in the lab")
        print("\nChoose one\n")
        print("1. Modify existing PC")
        print("2. Remove existing PC")
        print("3. Skip")

        select = input("\nChoose one number: ")

        while select!='1' and select!='2' and select!='3':
            select=input("Choose between 1 to 3: ")

        if select == '1':
            self.update_pc(pc_no)
        elif select == '2':
```

```
            self.remove_pc(pc_no)
        elif select == '3':
            self.main()

    else:
        for i in pc_no:
            if i<'0' or i>'9':
                print("\t\tPc number can only be digits\n")
                self.main()

        os = input("The installed operating system in the PC: ")
        status = input("Status : ")
        self.LAB_PCs[pc_no] = {"os": os, "status": status}
        print("PC added successfully.")

    m=int(input("\n\nPress 0 to go back to Menu: "))
    if m==0:
        self.main()

def remove_pc(self,pc_no):
    """This method is for deleting a particular pc from lab"""
    if pc_no not in self.LAB_PCs:
        print("PC number not found.")
    else:
        del self.LAB_PCs[pc_no]
        print("PC removed successfully.")

    m=int(input("\n\nPress 0 to go back to Menu: "))
    if m==0:
```

```
        self.main()

    def show_all_pc(self):
        """This method will show all pc info including os, status"""

        print("\n\nLab PC information\n")
        if not self.LAB_PCs:
            print("No PCs found.")
        else:
            for pc_no,pc_function in self.LAB_PCs.items():
                print(f"PC number:{pc_no}", )
                print(f"Operating system: {pc_function['os']}" )
                print(f"Status: {pc_function['status']}")
                print()

        m=int(input("\n\nPress 0 to go back to Menu: "))
        if m==0:
            self.main()

    def show_pc(self):
        """This is for displaying info of a particular PC"""
        print("\nSearch for a PC info\n")
        pc_number = input("Enter PC number: ")
        if pc_number in self.LAB_PCs:
            for pc_no,pc_function in self.LAB_PCs.items():
                if pc_no==pc_number:
                    print(f"PC number:{pc_no}", )
                    print(f"Operating system: {pc_function['os']}" )
                    print(f"Status: {pc_function['status']}")
```

```
            print()
    else:
        print("\nPC not found.")

    m=input("\n\nPress 0 to go back to Menu: ")
    if m=='0':
        self.main()

def show_functionality(self):
    """This shows a specific function of a particular pc"""
    print("\nSee functionality of a PC\n")
    pc_number = input("Enter PC number: ")
    if pc_number not in self.LAB_PCs:
        print("\nPC not found.")
        print(f"Do you want to add PC no. {pc_number}?")
        print("1. Yes\t 2.No")
        yOn=int(input("Select 1 or 2: "))
        while yOn!=1 and yOn!=2:
            yOn=input("Please select 1 or 2: ")
        if yOn==1:
            self.add_pc(pc_number)
        else:
            self.main()

    else:
        fn=input("Which funtionality you want to see: ")
        fn.lower()
        while fn!='status' and fn!='os':
            fn=input("Choose either 'os' or 'status': ")
```

```
            fn.lower()

        print(f"{fn} for PC number {pc_number} is:", self.LAB_PCs[pc_number][fn])

        m=int(input("\n\nPress 0 to go back to Menu: "))
        if m==0:
            self.main()

def store_pc(self):
    """This method is for storing all the PCs added"""
    filename=input("Your file name(add .txt): ")



    while filename[-4:]!='.txt':
        filename=input("Add .txt at the end of your file name): ")
    try:
        with open(filename, 'a') as file:
            if not self.LAB_PCs:
                print("No PCs found.")
            else:
                for pc_no,pc_function in self.LAB_PCs.items():
                    file.write(f"PC number:{pc_no}\n", )
                    file.writelines(f"OS:{pc_function['os']}\n", )
                    file.writelines(f"Status:{pc_function['status']}\n", )
                    file.write("\n")
                file.close()

                print("\nAll the PCs stored in file\n")
                o=input("To see the stored details press 1: ")
```

```
            if o=='1':
                #print("in")
                with open(filename) as file:
                    for line in file:
                        print(line)
    except FileNotFoundError:
        print("Sorry, file not found")

    m=int(input("\n\nPress 0 to go back to Menu: "))
    if m==0:
        self.main()

def main(self):
    """This method is to display main menu"""

    print("\n\t\t****** Menu ******\n")
    print("\t\tChoose one\n")
    print("\t\t1. Add PC")
    print("\t\t2. Update PC")
    print("\t\t3. Remove PC")
    print("\t\t4. Show all PC")
    print("\t\t5. Search a PC")
    print("\t\t6. See a functionalty of a PC")
    print("\t\t7. Store PC")
    print("\t\t8. Quit")

    select=input("\n\nChoose one number: ")
```

```
        while select<'1' or select>'8':
            select=input("\n\nChoose between 1 to 8 please: ")

        if select=='1':
            self.add_pc()
        elif select=='2':
            select=input("Enter the PC no. you want to update: ")
            self.update_pc(select)
        elif select=='3':
            select=input("Enter the PC no. you want to remove: ")
            self.remove_pc(select)
        elif select=='4':
            self.show_all_pc()
        elif select=='5':
            self.show_pc()
        elif select=='6':
            self.show_functionality()
        elif select=='7':
            self.store_pc()
        else:
            return

LAB_PCs={}
labobj=PC(LAB_PCs)
labobj.main()
```

Application Overview:

1. Menu:

```
****** Menu ******

Choose one

1. Add PC
2. Update PC
3. Remove PC
4. Show all PC
5. Search a PC
6. See a functionalty of a PC
7. Store PC
8. Quit


Choose one number:
```

2. Add PC:
   a.

```
Choose one number: 1
PC no: 1
The installed operating system in the PC: mac
Status : good
PC added successfully.
```

   b.

```
Choose one number: 1
PC no: 1
PC number already exists in the lab

Choose one

1. Modify existing PC
2. Remove existing PC
3. Skip

Choose one number:
```

3. Update PC:
   a.

```
Choose one number: 2
Enter the PC no. you want to update: 1
Installed operating system in the PC: windows
Status: good
PC information updated successfully.
```

b.

```
Choose one number: 2
Enter the PC no. you want to update: 3
PC number not found.
```

4. Remove PC

```
Choose one number: 3
Enter the PC no. you want to remove: 1
PC removed successfully.


Choose one number: 3
Enter the PC no. you want to remove: 1
PC number not found.
```

5. Show all

a.

```
Choose one number: 4


Lab PC information

PC number:1
Operating system: mac
Status: good

PC number:2
Operating system: windows
Status: bad
```

b. When there is no PC in the lab

```
Choose one number: 4


Lab PC information

No PCs found.
```

6. Search a PC:

a.

```
Choose one number: 5

Search for a PC info

Enter PC number: 1
PC number:1
Operating system: mac
Status: good
```

b.

```
Choose one number: 5

Search for a PC info

Enter PC number: 3

PC not found.
```

7. See a functionality of a PC:
   a.

```
Choose one number: 6

See functionality of a PC

Enter PC number: 1
Which funtionality you want to see: os
os for PC number 1 is: mac
```

   b.

```
Choose one number: 6

See functionality of a PC

Enter PC number: 1
Which funtionality you want to see: r
Choose either 'os' or 'status': status
status for PC number 1 is: good


Press 0 to go back to Menu:
```

8. Store PC:

```
Choose one number: 7
Your file name(add .txt): Store.txt

All the PCs stored in file

To see the stored details press 1: 1
in
PC number:1

OS:mac

Status:good
```

9. Quit:

```
****** Menu ******

Choose one

1. Add PC
2. Update PC
3. Remove PC
4. Show all PC
5. Search a PC
6. See a functionalty of a PC
7. Store PC
8. Quit


Choose one number: 8
PS C:\Users\USER\AppData\Local\Programs\Python\Python3
11\work>
```

10. Anything else:

```
                ****** Menu ******

                Choose one

                1. Add PC
                2. Update PC
                3. Remove PC
                4. Show all PC
                5. Search a PC
                6. See a functionalty of a PC
                7. Store PC
                8. Quit


Choose one number: 9


Choose between 1 to 8 please: e


Choose between 1 to 8 please: 8
PS C:\Users\USER\AppData\Local\Programs\Python\Python3
11\work>
```