

ASSIGNMENT – 11

Name: **Sudharsan Srinivasan**

UTA ID: **1001755919**

- A python code is written as part of this assignment to validate inputs and compare them with regular expressions.

CODE DESCRIPTION:

- The code executes a typical telephone directory, where the user is provided with the following options

ADD

DELETE USING NAME

DELETE USING PHONE NUMBER

LIST

EXIT

AUDIT (For Linux systems)

ADD:

- This function enables user to add a new user to their phonebook directory. Name and phone number can be added for a user.
- However, while adding these two fields, it is important to follow the allowed regular expression formats for the inputs.
- Random formats will not be accepted by the code. Therefore, it must fall under one of the below expressions as shown in the regex formats below.

```

InputValidation.py
1  import os
2  import csv
3  from datetime import datetime
4  import re
5  import time
6  from datetime import datetime
7  import platform
8
9
10 # Regex for phone numbers
11 phoneNumberF1 = '^\\d{1}\\(\\d{3}\\)\\d{3}-\\d{4}$'
12 phoneNumberF2 = '^\\d{3}\\s\\d{1}\\s\\d{3}\\s\\d{3}\\s\\d{4}$'
13 phoneNumberF3 = '^\\(\\d{3}\\)\\d{3}-\\d{4}$'
14 phoneNumberF4 = '^\\+\\d{2}\\s\\(\\d{2}\\)\\s\\d{3}-\\d{4}$'
15 phoneNumberF5 = '^\\+\\d\\(\\d{3}\\)\\d{3}\\s-\\d{4}$'
16 phoneNumberF6 = '^\\d{5}.\\d{5}$'
17 phoneNumberF7 = '^\\d{3}-\\d{4}$'
18 phoneNumberF8 = '^\\d{3}\\s\\d{1}\\s\\d{3}\\s\\d{3}\\s\\d{4}$'
19 phoneNumberF9 = '^\\d{5}$'
20 phoneNumberF10 = '^\\d{3}\\s\\d{3}\\s\\d{3}\\s\\d{4}$'
21
22 # Regex for names
23 nameF1 = '^([a-zA-Z]+((\\'|\\-|\\.)?([a-zA-Z]))+)?$'
24 nameF2 = '^([a-zA-Z]+((\\(\\),\\(\\s)|\\-|\\.)?([a-zA-Z]))+)((\\s)?([a-zA-Z]))+)?$'
25 nameF3 = '^([a-zA-Z]+((\\s)?([a-zA-Z]))+)?$'
26 nameF4 = '^([a-zA-Z])+?$'
27 nameF5 = '^([\\s]+,?(\\s|^\\s))+?$'
28 nameF6 = '^([a-zA-Z]+((\\(\\),\\(\\s)|\\-|\\.)?([a-zA-Z]))+)?$'
29 nameF7 = '^([a-zA-Z]+((\\s)?(\\'|\\-|\\.)?([a-zA-Z]))+)?$'
30

```

- This regex criteria are applicable for both Name and Phone Number fields.
- As seen below, the add function is implemented. It is also important to see if the platform is not Windows, the log file is also saved.

```

# Function to ADD records
def add():
    try:
        print("Enter Details For A New Record")
        nn = input("Enter Name: ")
        name_check(nn)
        nnp = input("Enter phone number: ")
        phone_check(nnp)
        phone = [nn, nnp]
        list1.append(phone)

        # Log the details if not Windows OS.
        if (platform.system().lower() != 'windows'):
            try:
                log(nn)
            except:
                print('')
                exit()
    except:
        print("Input Error")

```

DELETE USING NAME:

- The idea behind this functionality is to get a name from the user and delete the record that matches the name.
- The criteria for this functionality to work is that the name should already be present in the phonebook database, otherwise the function will return an error.

```

115
116 # Function for DELETE records using NAME
117 def delete_name(name):
118     try:
119         j=0
120         f=0
121         for i in list1:
122             if i[nameParameter]==name:
123                 print("Deleting...")
124                 f=1
125                 j+=1
126             else:
127                 j+=1
128         if f==0 :
129             print ("Record does not exist")
130             exit(1)
131         del list1[j-1]
132         print("Record Deleted- Name : ", name)
133         save_csv()
134
135         # Log the details if not Windows OS.
136         if (platform.system().lower() != 'windows'):
137             try:
138                 log(name)
139             except:
140
141                 print('')
142                 exit()
143
144     except:
145         print("Error Deleting Data")
146

```

DELETE USING PHONE NUMBER:

- The idea behind this functionality is similar to delete using name, where we get phone number from the user and delete the record that matches the name.
- The criteria for this functionality to work is that the phone number should already be present in the phonebook database, otherwise the function will return an error.

```

0
1 # Function to DELETE records using PHONE NUMBER
2 def delete_phone(number):
3
4     try:
5         j = 0
6         f=0
7         for i in list1:
8             if i[phoneParameter] == number:
9                 print("Deleting Record....")
10                f=1
11                j+=1
12            else:
13                j+=1
14        if f==0 :
15            print("Record does not exist in the database")
16            exit(1)
17        del list1[j]
18        print("Deleting.....")
19        print("Record Deleted Number : ", number)
20        save_csv()
21    except:
22        print("")
23

```

LIST:

- This function is to list all the records that have been added to the phonebook directory.

- It is to be noted that this function will not return any values if we choose to use this, without adding any records to our phonebook directory.

```

37
38 # Function for displaying LIST of records.
39 def list():
40     try:
41         display2(headers, "")
42         j = 1
43         for i in list1:
44             display2(i, j)
45             j += 1
46         print()
47         try:
48             log2()
49         except:
50             print("")
51             main()
52
53     except:
54         print("Error Displaying Data")
55

```

EXIT:

- As the name suggests, it stop the execution of the program and exits.

AUDIT:

- As the code runs functions such as adding/deleting records, a **log.txt** file is created, which contains the UserID of the system (assuming it's a root user), the records that are used and the corresponding timestamp of the record.
- It is important to note that the file will not get created unless the user is a **privileged root user**.

```

176 # Function to write into LOG file
177 def log(name):
178
179     # Store the current system time and the RUID.
180     t1 = datetime.now()
181     ruid = os.getuid()
182
183     try:
184
185         if(is_root()):
186             f = open("log.txt", "a")
187             f.write("\nRUID: " + str(ruid) + ' ' + 'Name: ' + name + ' ' + "Time-Stamp: " + str(t1))
188             f.close()
189
190         # Error if the user isn't root.
191         else:
192             print("Unable to write into log file. Not a root user")
193             main()
194     except:
195         print("")
196

```

COMPILATION/BUILD INSTRUCTIONS:

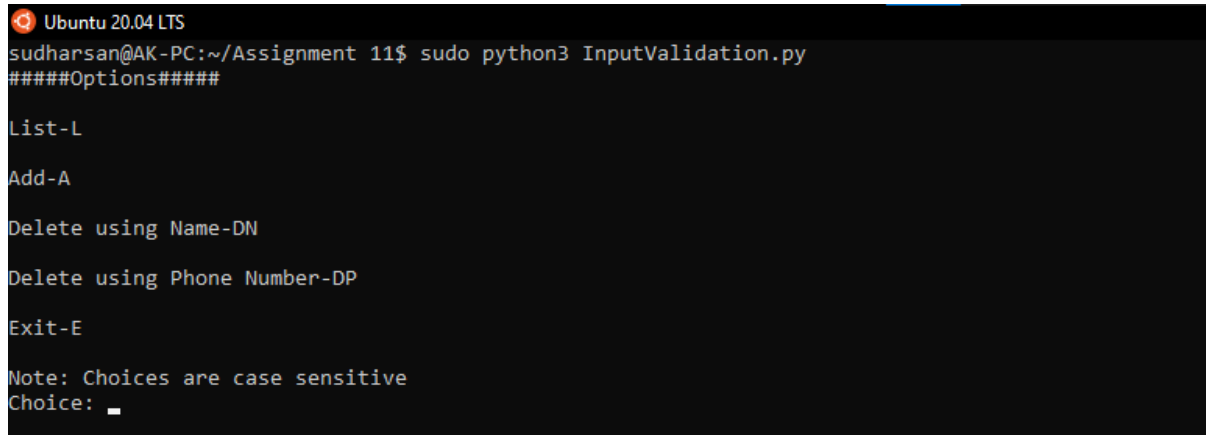
- The code is written in python with extension as .py for the file. The code can be run from the command prompt directly by navigating to the folder/directory which contains the py file.
- Further, since this file is cross platform, it can also be used to run through code editors such as **Visual Code**, **Eclipse** etc., but we are sticking to execution using **Command Line Interface** itself, since it's much easier and more efficient.

EXECUTION OF THE CODE:

- Since, we are using Command Line Interface, we navigate to the folder containing the file and then use the below command

sudo python3 InputValidation.py

- Here, we run the file using sudo command because we are running the file as a root user with privileges to make sure we get Audit logs for our activity.

A terminal window titled 'Ubuntu 20.04 LTS' showing the execution of a Python script. The prompt is 'sudharsan@AK-PC:~/Assignment 11\$' and the command entered is 'sudo python3 InputValidation.py'. The output shows a menu of options: 'List-L', 'Add-A', 'Delete using Name-DN', 'Delete using Phone Number-DP', and 'Exit-E'. A note states 'Note: Choices are case sensitive' and the prompt 'Choice: ' is followed by a cursor.

```
Ubuntu 20.04 LTS
sudharsan@AK-PC:~/Assignment 11$ sudo python3 InputValidation.py
#####Options#####

List-L
Add-A
Delete using Name-DN
Delete using Phone Number-DP
Exit-E

Note: Choices are case sensitive
Choice: _
```

ASSUMPTIONS:

Some of the assumptions made for this assignment are given below:

- We assume that the user knows the correct format of the name and phone number to input and store into the phonebook.
- If List operation is chosen first by the user, we assume that we already have records.csv file present in the folder, which already contains some records in it.
- The user enters the phone number and name in the same format for deleting a record, as they did for adding a record.

PROS/CONS:**PROS:**

- Since the code is cross-platform, it is easy to manage and can be run using various software and not restricted to just one.
- Can be implemented easily using Python.

CONS:

- The code will throw error if the format of name or phone number does not match the formats specified in the code as per the requirement for the assignment.

SCREENSHOTS OF CODE EXECUTION:**Running the Code:**

```
Ubuntu 20.04 LTS
sudharsan@AK-PC:~/Assignment 11$ sudo python3 InputValidation.py
#####Options#####

List-L

Add-A

Delete using Name-DN

Delete using Phone Number-DP

Exit-E

Note: Choices are case sensitive
Choice: _
```

Choosing ADD function:

- Since initially we don't have any records to display, we use ADD function to add records to the phonebook directory.

```
Ubuntu 20.04 LTS
sudharsan@AK-PC:~/Assignment 11$ sudo python3 InputValidation.py
#####Options#####

List-L

Add-A

Delete using Name-DN

Delete using Phone Number-DP

Exit-E

Note: Choices are case sensitive
Choice: A
Enter Details For A New Record
Enter Name: ak

Name Accepted
Enter phone number: 68228-30833

Phone Accepted
```

- Multiple records are then added to the phonebook directory.

```
Ubuntu 20.04 LTS
#####Options#####

List-L

Add-A

Delete using Name-DN

Delete using Phone Number-DP

Exit-E

Note: Choices are case sensitive
Choice: A
Enter Details For A New Record
Enter Name: ashwin

Name Accepted
Enter phone number: 68228-30834

Phone Accepted
```

```
Ubuntu 20.04 LTS
#####Options#####

List-L

Add-A

Delete using Name-DN

Delete using Phone Number-DP

Exit-E

Note: Choices are case sensitive
Choice: A
Enter Details For A New Record
Enter Name: sudharsan

Name Accepted
Enter phone number: 68228-30835

Phone Accepted
```

```
Ubuntu 20.04 LTS
#####Options#####

List-L

Add-A

Delete using Name-DN

Delete using Phone Number-DP

Exit-E

Note: Choices are case sensitive
Choice: A
Enter Details For A New Record
Enter Name: Lili

Name Accepted
Enter phone number: 68228-30836

Phone Accepted
```

Choosing LIST function:

- Now that records are added to the phonebook, we can display the records using List option as seen below.

```
Ubuntu 20.04 LTS
#####Options#####

List-L

Add-A

Delete using Name-DN

Delete using Phone Number-DP

Exit-E

Note: Choices are case sensitive
Choice: L

  Name                Phone Number
1  ak                 68228-30833
2  ashwin             68228-30834
3  sudharsan          68228-30835
4  Lili               68228-30836
```

Choosing Delete using Name function:

- Now that ADD and LIST function ensures that we have records in the phonebook, we can go ahead and delete a record from it. We delete the record using name. In this case, we delete the record with name **Lili**

- It is important to note that the name format should be the same as that of the one which was used while adding the record, otherwise the database would throw an error saying that the Record is not found in the database.

```
Ubuntu 20.04 LTS
#####Options#####
List-L
Add-A
Delete using Name-DN
Delete using Phone Number-DP
Exit-E
Note: Choices are case sensitive
Choice: DN
Please Enter The Name Of The Record To Delete Lili
Name : Lili
Deleting...
Record Deleted- Name : Lili
#####Options#####
List-L
Add-A
Delete using Name-DN
Delete using Phone Number-DP
Exit-E
Note: Choices are case sensitive
Choice: L
      Name                Phone Number
1   ak                    68228-30833
2   ashwin                 68228-30834
3   sudharsan              68228-30835
```

Choosing Delete using Phone Number function:

- For this function to work, we choose the corresponding DP option and enter the phone number in the same format as when it was added.

```
Ubuntu 20.04 LTS
#####Options#####

List-L

Add-A

Delete using Name-DN

Delete using Phone Number-DP

Exit-E

Note: Choices are case sensitive
Choice: DP
Please Enter The Phone Number Of The Record To Delete 68228-30835
Phone number : 68228-30835
Deleting Record....
```

```
Ubuntu 20.04 LTS
#####Options#####

List-L

Add-A

Delete using Name-DN

Delete using Phone Number-DP

Exit-E

Note: Choices are case sensitive
Choice: L
      Name                Phone Number
1  ak                    68228-30833
2  ashwin                68228-30834
```

Choosing EXIT function:

- Finally, we use the EXIT function to exit the program and stop execution.

```
Ubuntu 20.04 LTS
#####Options#####
List-L
Add-A
Delete using Name-DN
Delete using Phone Number-DP
Exit-E

Note: Choices are case sensitive
Choice: L
      Name                Phone Number
  1  ak                   68228-30833
  2  ashwin               68228-30834

#####Options#####
List-L
Add-A
Delete using Name-DN
Delete using Phone Number-DP
Exit-E

Note: Choices are case sensitive
Choice: E
Exiting...

sudharsan@AK-PC:~/Assignment 11$
```

AUDIT LOG:

- As we perform functions such as ADD/DELETE, a corresponding **log.txt** file gets created and it keeps track of the functions that are carried out.

```
*log.txt - Notepad
File Edit Format View Help

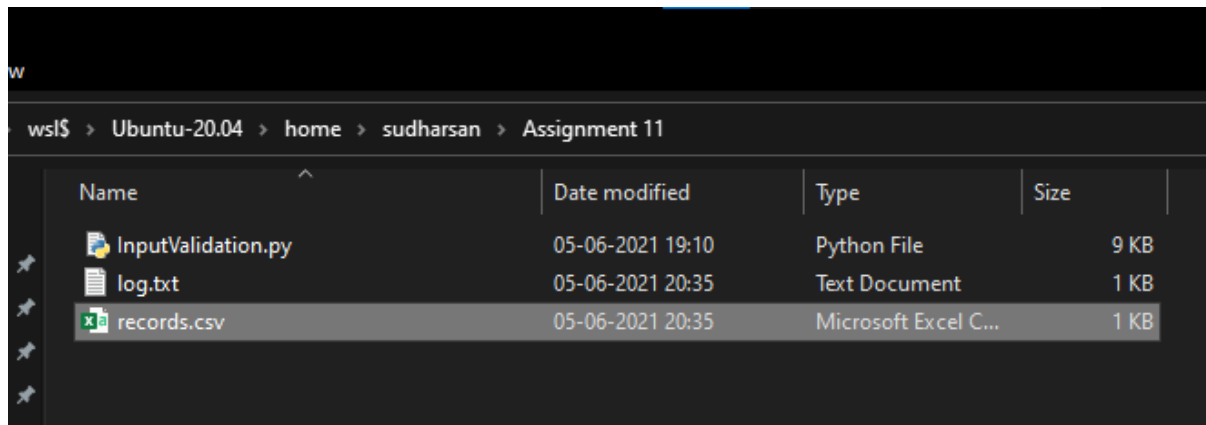
RUID: 0 Name: ak Time-Stamp: 2021-05-06 20:26:36.253160
RUID: 0 Name: ashwin Time-Stamp: 2021-05-06 20:26:46.226680
RUID: 0 Name: sudharsan Time-Stamp: 2021-05-06 20:27:01.795664
RUID: 0 Name: Lili Time-Stamp: 2021-05-06 20:27:15.859149
RUID: 0 Time-Stamp: 2021-05-06 20:29:24.643297
RUID: 0 Name: Lili Time-Stamp: 2021-05-06 20:31:59.419608
RUID: 0 Time-Stamp: 2021-05-06 20:32:04.432596
RUID: 0 Time-Stamp: 2021-05-06 20:34:38.111813
RUID: 0 Time-Stamp: 2021-05-06 20:35:51.966359
```

- As seen in the above **log.txt** file, the first four lines correspond to the ADD function, where we added 4 records to the phonebook.

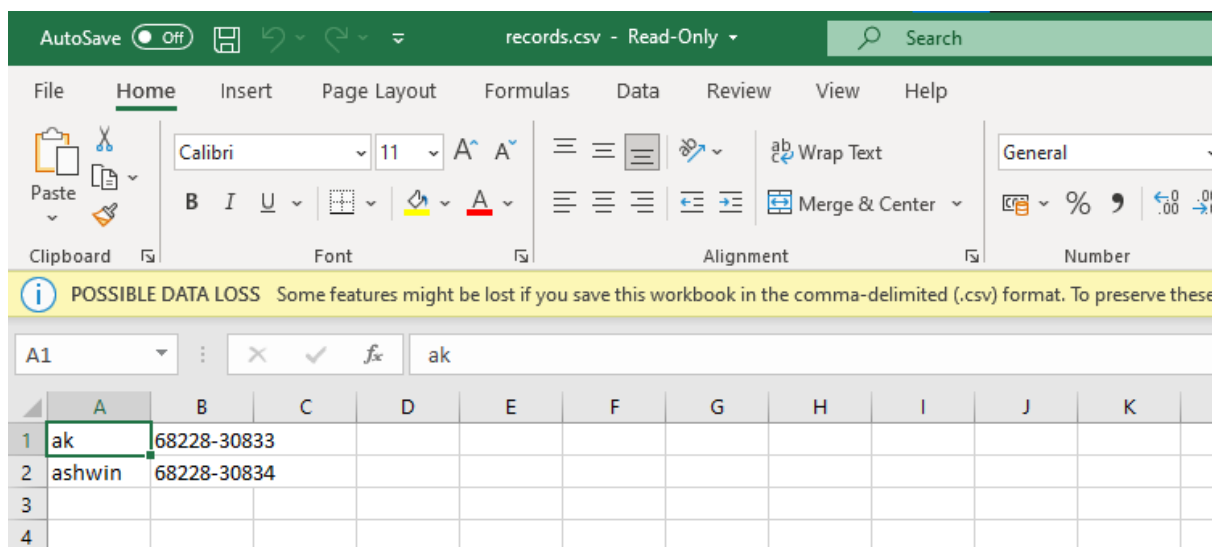
- The 5th line denotes the DN option we selected to delete a record using Name. The next line (6th) denotes the record name we entered to get it removed.
- The next two lines (7th and 8th) indicates similar functionality, but in this case, it is for DP option and then followed by the phone number entered.
- The final line (9th) corresponds to the Exit option finally chosen.

RECORDS FILE:

- After execution of the program and exiting, we see that there is a **records.csv** file created in the same directory as that of the python file.



- This file contains the log of all the records in a table format, with the fields, **Name and Phone Number**.



- We only have 2 records when we end the program because, we added 4 records using ADD function, then deleted a record using DN and DP function, leaving us with 2 records, which are display in the above csv file.

REFERENCES:

- <https://stackoverflow.com/questions/2385701/regular-expression-for-first-and-last-name>
- <https://regexper.com/>
- https://www.tutorialspoint.com/java/java_regular_expressions.htm