PROJECT REPORT

TITLE OF THE PROJECT: UNIQUE PASSWORD GENERATOR GUI

ABSTRACT:

UNIQUE PASSWORD GENERATOR is a simple application which can randomly generate passwords. The python implementation of password generator project is using the random and Tkinter module. Using this application we can generate passwords of specified length by the user with the combination of lowercase, uppercase letters, numbers, and special characters.

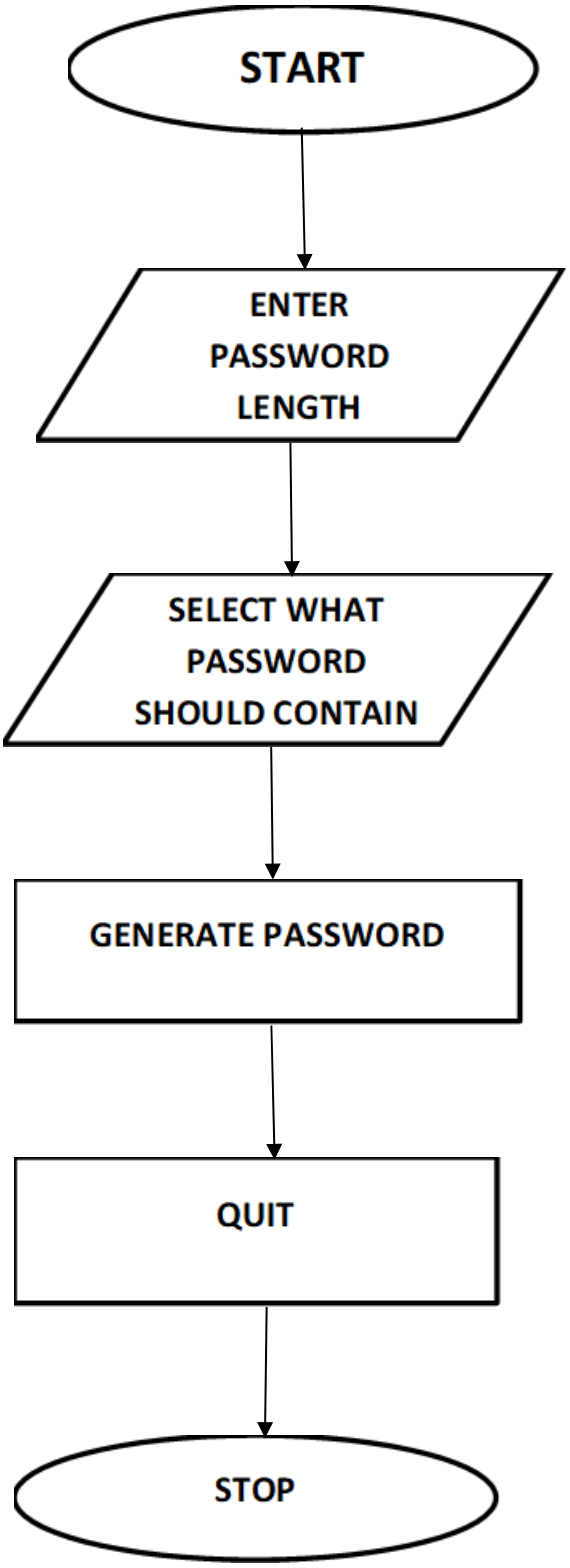
To randomly generate a password, we made a base code that return a unique password by implementing the use of random module. To use widgets and define the application interface, we use the Tkinter module. Tk() function is used to create window. We used Label function to add non-editable text of an application. We placed labels like password, password length, small letters, Capital letters, etc. We also used checkbutton which is an inbuilt function in Tkinter module. The user can select one or more options by clicking the button corresponding to each option, when it is selected it has a value of 1 otherwise zero.

For storing entries intvar() are used. Generate password and quit buttons perform a certain function when the user clicks on it. Using command argument, generator function is linked to the button. In generator function we have 4 lists containing small letters, capital letters, numbers and special characters. If none of the checkbuttons are checked we get a message box with text “Atleast one box needs to be checked”. Similarly, if the password length given by the user is less than 4, a messagebox with text “Password length cannot be less than 4” get displayed. “full_dict” contains the values of choices made by the user. random.shuffle() is used to shuffle the “full_dict” so that user will not get predicted passwords. At the end if

the password length is greater than 8 a message box with information “New password generated successfully.” gets displayed otherwise message box with text “The generated password is weak. Choose a password of length greater than 8.” Gets displayed.

When the user clicks quit button it get closed. root.mainloop() is used to display the window we created using Tk() function.

SYSTEM ARCHITECTURE:



CODE:

```
from tkinter import *

from tkinter import messagebox

import random


root = Tk()

root.geometry("500x555")

root.title("UPG Utility")

root.resizable(False,False)


def generator():

    #this is the base code for the unique password generator

    full_dict = []

    raw_pass = []


    l1 = [chr(a) for a in range(97,123)]

    l2 = [chr(a) for a in range(65,91)]

    l3 = [chr(a) for a in range(48,58)]

    l4 = ["@","#","$","%"]



    if(smallValue.get() == 0 and capValue.get() == 0 and numValue.get() == 0 and
specialValue.get() == 0):
```

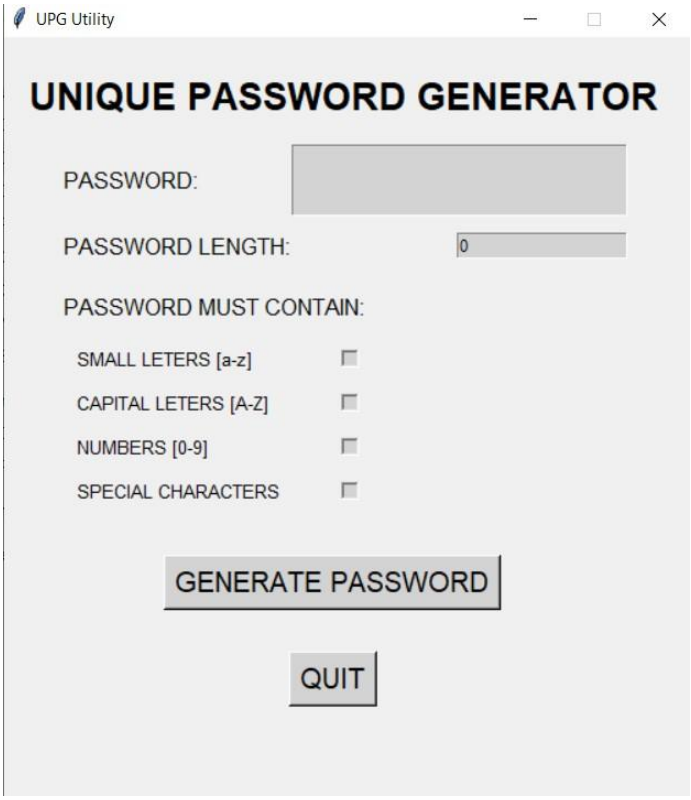
```
        messagebox.showerror("Unique Password Status","At least one box needs to be checked.")
    elif(passlengthValue.get()<4):
        messagebox.showerror("Unique Password Status","Password Length cannot be less than 4.")
    else:
        if(smallValue.get() == 1):
            raw_pass.append(random.choice(l1))
            full_dict.extend(l1)

        if(capValue.get() == 1):
            raw_pass.append(random.choice(l2))
            full_dict.extend(l2)

        if(numValue.get() == 1):
            raw_pass.append(random.choice(l3))
            full_dict.extend(l3)

        if(specialValue.get() == 1):
            raw_pass.append(random.choice(l4))
            full_dict.extend(l4)

        random.shuffle(full_dict)
```

```
    x = int(passlengthValue.get())

    shuffle_pass = []

    final_pass = ""

    start_val = smallValue.get() + capValue.get() + numValue.get() + specialValue.get()


    for _ in range(start_val,x):

        raw_pass.append(random.choice(full_dict))


    shuffle_pass = random.sample(raw_pass,len(raw_pass))

    final_pass = final_pass.join(map(str,shuffle_pass))

    print(final_pass)

    password.delete('1.0', 'end') #clear the output text text widget

    password.insert(END, final_pass)

    if(passlengthValue.get()>=8):

        messagebox.showinfo("Unique Password Status","New Password Generated Successfully.")

    else:

        messagebox.showwarning("Unique Password Status","The Generated Password is weak.\nChoose a password of length greater than 8.")



#Heading

Label(root,text="UNIQUE PASSWORD GENERATOR", font="comicsansms 20 bold").grid(row=0, column=1, padx=(15,0), pady=(25,15))
```

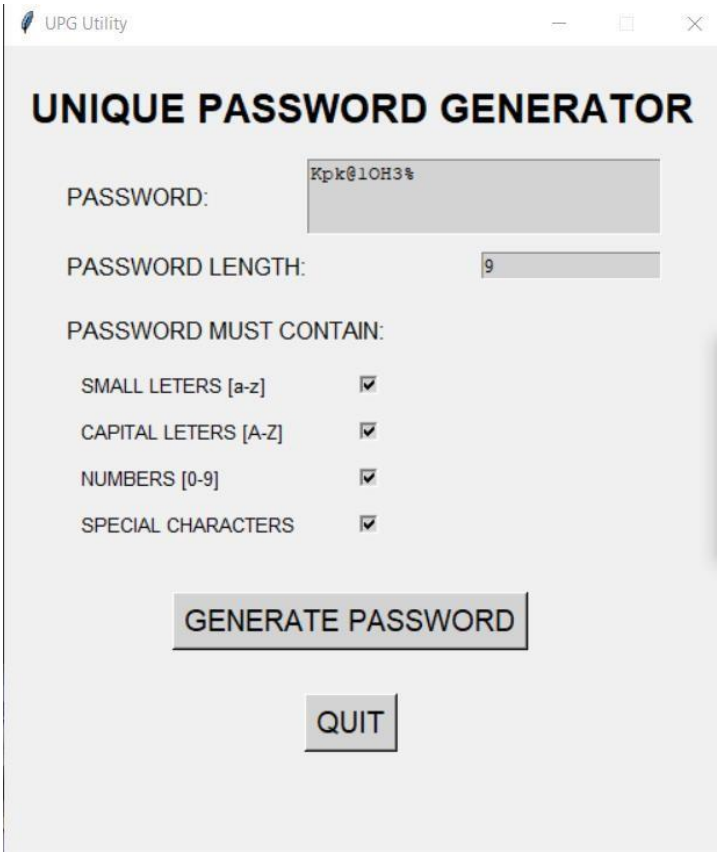
```
#Texts for the application

p = Label(root, text="PASSWORD:",font="comicsansms 12").grid(row=1, column=1, padx=(40,0), sticky=W)

p_length = Label(root, text="PASSWORD LENGTH:",font="comicsansms 12").grid(row=2, column=1, padx=(40,0), pady=(10,0), sticky=W)

Label(root, text="PASSWORD MUST CONTAIN:",font="comicsansms 12").grid(row=3, column=1, padx=(40,0), pady=(20,10), sticky=W)

#Tkinter variables for storing entries

passlengthValue = IntVar()

smallValue = IntVar()

capValue = IntVar()

numValue = IntVar()

specialValue = IntVar()

#Entry and Password display for our program

plengthEntry = Entry(root, textvariable=passlengthValue, bg = "light grey").grid(row=2, column=1, pady=(10,0), padx=(0,25), sticky=E)

password = Text(root, height = 3, width = 30, bg = "light grey")

password.grid(row=1, column=1, padx=(0,25), sticky=E)


#Checkboxes

Label(root, text= "SMALL LETERS [a-z]",font="comicsansms 10").grid(row=6, column=1, padx=(50,0), pady=5, sticky=W)

Label(root, text= "CAPITAL LETERS [A-Z]",font="comicsansms 10").grid(row=7, column=1, padx=(50,0), pady=5, sticky=W)
```

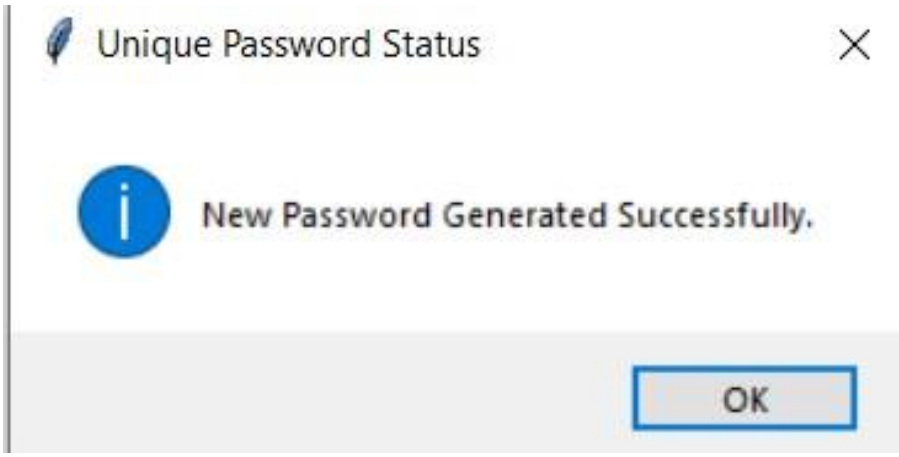
```
Label(root, text= "NUMBERS [0-9]",font="comicsansms 10").grid(row=8, column=1, padx=(50,0), pady=5, sticky=W)

Label(root, text= "SPECIAL CHARACTERS",font="comicsansms 10").grid(row=9, column=1, padx=(50,0), pady=5, sticky=W)

lowercase = Checkbutton(root, variable=smallValue, selectcolor="light grey").grid(row=6,column=1,padx=(30,0))

uppercase = Checkbutton(root, variable=capValue, selectcolor="light grey").grid(row=7,column=1,padx=(30,0))

numbers = Checkbutton(root, variable=numValue, selectcolor="light grey").grid(row=8,column=1,padx=(30,0))

specialChars = Checkbutton(root, variable=specialValue, selectcolor="light grey").grid(row=9,column=1,padx=(30,0))


#Buttons

Button(root, text="GENERATE PASSWORD", font="comicsansms 15", bg="light grey", command=generator).grid(row=10,column=1,pady=30)

Button(root, text="QUIT", font="comicsansms 15", bg="light grey", command=root.destroy).grid(row=11,column=1)


root.mainloop()
```

UI DESIGN:



OUTPUT:





CONCLUSION:

We have successfully created python password generator. This code is a concise method to generate unique password according to the length and the selections given by the user. This python project also provided an introduction to a random module that can be used to generate a random number or return a random element from an array. Also, an introduction to Tkinter is also provided for using simple widgets.