

MULTIMESSENGER

NAME:SAGAR

Problem Statement — This project aims to let users use any messaging service(including attaching documents) from their terminal .

I. DESIGN, PLAN OF THE PROJECT

Multimessenger:It is a program which will let users message through following by their terminal:

whatsapp,email, twitter

It will also help user to attach documents needed to sent .

we have used pyautogui module and tkinter GUI for creating dioogue box for taking their choice of messaging service from user.

And the core of this all this is modules which is very useful in sending which on being invoked take control of sending message using its methods.

A module is nothing but a file containing definitions and statements.It is a collection of functions,classes and variables.

By using terminal,and giving the required details in terminal ,it will enable you to send message in a very less time.

Specially for programmers this is very useful . A lot of actions can be performed by modifying the basic code-

- i)we can do bulk messaging by using simple for loop.
- ii)we can shedule our message.
- iii)we can Send WhatsApp messages.
- iv)we can Play a YouTube video.
- v)we can Perform a Google Search.
- vi)Get information on a particular topic ,etc.

II.CURRENT STATUS OF THE PROJECT

A. *Current Status of the Project:*

For now our progress:

1)whatsapp messaging-the program part for this is

Perfectly wotking.

Taking the recipient phone no.,message,when to sent, and using sendwhatsmsg method,we are able to send messages.

It is before ensured our system is logged in with our whatsapp.

2)Gmail-

We have used smtp module for making connection with gmail server.by setting up the required port no. and authenticating with our username and password which then proceed by making connection with several mail transfer agent. And finally the end user.

3)Twitter-we are able to make a post on twitter by using Tweepy module and its authetic api.

By using the various security keys,we are and using the method update_status from module for posting on twitter.

B. *How much of the design is converted into code?*

We are now successful in building our multimessenger.

We have developed our program and now we are able to-

- i)send email
 - ii)send message through whatsapp
 - iii)make a post on twitter
-

III. EXPECTED FINAL DESIGN

a. *Abbreviations and Acronyms*

API- **API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software.**

GUI- A graphics-based operating system interface that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system.

TLS-Transport Layer Security (TLS) is a standard internet protocol that encrypts email for privacy and secure delivery. TLS prevents unauthorized access of email when it's in transit over internet connections.

SSL-It is a standard internet protocol for servers and browsers. SSL creates encrypted connections, which

ensures that data transmitted between these systems is secure and private.

SMTP (Simple Mail Transfer Protocol)-A messaging protocol governing electronic-mail transmission in Transmission Control Protocol/Internet Protocol (TCP/IP) networks. It is used to transfer e-mail between computers. It is a server-to-server protocol. SMTP supports only text and cannot handle attachments.

a. Modules and API:

Email-smtplib module

The *smtplib* is a Python library for sending emails using the Simple Mail Transfer Protocol (SMTP). The *smtplib* is a built-in module; we do not need to install it. It abstracts away all the complexities of SMTP.

Whatsapp-pywhatkit

Python offers numerous inbuilt libraries to ease our work. Among them **pywhatkit** is a Python library for sending WhatsApp messages at a certain time, it has several other features too.

Following are some features of pywhatkit module:

1. Send WhatsApp messages.
2. Play a YouTube video.
3. Perform a Google Search.
4. Get information on a particular topic.

Twitter-tweepy

Tweepy is an open source Python package that gives you a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter's models and API endpoints, and it transparently handles various implementation details, such as:

- Data encoding and decoding
- HTTP requests
- Results pagination
- OAuth authentication
- Rate limits
- Streams

Twitter API-

The Twitter API is a set of programmatic endpoints that can be used to understand or build the conversation on Twitter.

This API allows you to find and retrieve, engage with, or create a variety of different resources including the following:

- Tweets
- Users
- Spaces
- Direct Messages
- Lists
- Trends
- Media
- Places

C.Summary:

Step1:

We have imported all the following module for different messaging services -

```
from pyautogui import *  
import pywhatkit  
import tweepy  
import smtplib,ssl
```

step2:

Before stepping further we should take choice from user that what messaging service he wants.

And pyautogui module helps in this.

This module helps us by creating a dialog box with some buttons for the users to click upon.

Step3:

After that we have used the simple if conditions for every corresponding messaging service. The buttons chosen will run the corresponding code under if condition.

Whatsapp messaging algorithm:

```
if(your_mode == "WHATSAPP"):  
    mob_no= int(input('enter no.'))  
    message=input('enter message')  
    time = [int(x) for x in input("time").split(",")]  
    pywhatkit.sendwhatmsg(f'+91{mob_no}',  
message, time[0], time[1])
```

we are using sendwhatsmessage method under pywhatkit module, which takes mobile no, and time for message to send as arguments.

Step4:

Twitter message algorithm:

```
# authentication of consumer key and secret
auth = tweepy.OAuthHandler(consumer_key,
consumer_secret)

# authentication of access token and secret
auth.set_access_token(access_token,
access_token_secret)
api = tweepy.API(auth)

# update the status
status=input('enter the status you want to set')

api.update_status(status),
update_status method under API method is used for
finally updating the status.
```

We have used api for twitter messaging.

We have got our API by its official website which is completely authentic.

NOTE:when we will be using API ,actually we will handover our work to some third program on behalf of us without giving access of our important details to site providing api

But using module may give the program control over browser for some actions, like selenium automation and pywhatkit.

Step5:

Gmail algorithm:

```
if(your_mode == "GMAIL"):
```

```
    try:
```

```
port = 465 # For SSL
```

```
smtp_server = "smtp.gmail.com"
```

```
sender_email = "sagaradhrit@gmail.com"
```

```
reciever=input('reciever email id')
```

```
receiver_email = reciever
```

```
password = 'abcd'
```

```
sub=input('enter the subject')
```

```
mes=input('enter the message')
```

```
args=(sub,mes)
```

```
message = """\
```

```
Subject: {0}
```

```
{1}""".format(*args)
```

```
context = ssl.create_default_context()
```

```
with smtplib.SMTP_SSL(smtp_server, port,
context=context) as server:
```

```
server.login(sender_email, password)
```

```
server.sendmail(sender_email, receiver_email,
message)
```

```
print('message sent asuccessfully')
```

```
except:
```

```
print('an error occured')
```

First we are connecting to gmail server using port no. 465 and authenticating our account using username and password.

Finally using its sendmail method under smtplib module we are succesful in sending mail.

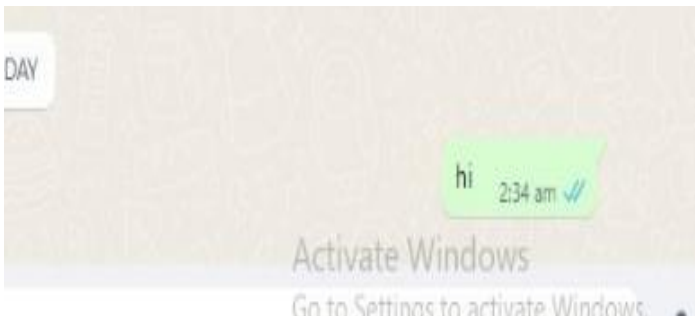
At last we have tried to handle the exception so that our program doesn't terminate in the case some error occurred.

IV SNAPSHOTS:

Whatsapp-

```
elif(your_mode == "WHATSAPP"):
    mob_no= int(input('enter no. '))
    message=input('enter message')
    time = [int(x) for x in input("time").split(",")]
    pywhatkit.sendwhatmsg(f'+91{mob_no}', message, time[0], time[1])
```

```
enter no. 8287764290
enter message hi
time 02,34
In 69 Seconds WhatsApp will open and after 15 Seconds Message will be Delivered!
```



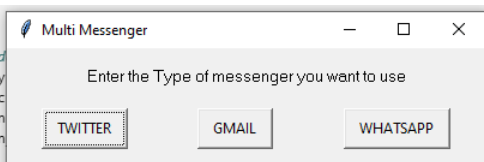
TWITTER-

```
try:
    # personal d
    consumer_key
    consumer_sec
    access_token
    access_token

    # authentication of consumer key and secret
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)

    # authentication of access token and secret
    auth.set_access_token(access_token, access_token_secret)
    api = tweepy.API(auth)

    # update the status
    status=input('enter the status you want to set')
    api.update_status(status)
    print('status set on twitter')
```



Gmail-

```
print('message sent asuccessfully')
except:
    print('an error occurred')
```

```
reciever email id sagaradhrit@gmail.com
enter the subject hi
enter the message hi
message sent asuccessfully
```



Subject: hi <sagaradhrit@gmail.com>

to bcc: me ▼

hi

REFERENCES

- [1] Python 3.10.6 documentation
- [2] <https://docs.python.org/3/library/tkinter.html>
- [3] <https://developer.twitter.com/en/docs/twitter-api>
- [4] <https://selenium-python.readthedocs.io/>
- [5] <https://pypi.org/project/pywha>

19 Following 0 Followers

Tweets

Tweets & replies



sagar @sagar07530533 · 16s

Hello Everyone !

