ASSIGNMENT #2

Course: Scripting

INFT1104 - 01

Date: 11/15/2024

by

Mihir Mane

100947380

Professor:

Sohaib Mohiuddin

A black and green logo

Description automatically generated

Durham College of Applied Arts and Technology

Cloud Computing

2024

**Introduction:**

A local computing problem that I want to address includes typing messages on WhatsApp without having to save the number of the recipient. At the moment, in WhatsApp for example, one cannot directly send a message or media to a person they have no saved as a contact, thus for people who engage in different exchanges with many persons for a single or temporary period of time, this becomes rather cumbersome. This automation program gets around that requirement by using the pywhatkit library to send a message or an image to any number, through threading to do multiple sends at once. That is why with the help of threading several tasks can be performed in parallel, the procedure of sending messages to several contacts simultaneously is also more convenient. This application would be most useful for owners of small businesses, or for individuals, who might need to communicate with several clients or counterparts. The script will use both pywhatkit for sending messages via WhatsApp Web together with selenium and the threading Python library for handling multiple operations. But the convenience it provides in the organization of communication is quite invaluable, so it is a worthy challenge.

**WhatsApp message Automation:** [Github Link (whatsappauto.py)](https://github.com/StunnerMnM7/cosc-1104-scripting/blob/main/whatsappauto.py)

The majority of individuals, whether in practice or post interviews, should send reminder emails after meetings, applications, or project completion; however, managing the process can be tedious and often reminiscent of the game of telephone. This could be easily done by a python program that reads from a list containing contacts and follow up notes and sends out pre-scheduled emails depending on certain conditions such as the number of days since the last contact. Through threading, the program was able to work on multiple mails in a single go, which would prove beneficial to users who have to deal with multiple follow-ups on the same subject with different recipients. This solution would be most beneficial for people, who have many clients or projects, as it increases one’s efficiency without the need of keeping track of time. This would require practical knowledge of the smtplib for sending the emails, pandas for reading the csv files and proper error handling for managing the server connection, it is moderately complex since it involves lot of real time usage of the concepts but it is highly beneficial.

**Email follow-up Automation:** [Github Link (emailauto.py)](https://github.com/StunnerMnM7/cosc-1104-scripting/blob/main/emailauto.py)

**Reflection:**

Working on these two automation programs for WhatsApp and email follow-ups was both engaging and challenging, as they offered practical solutions to common communication tasks. I found the problems to be suitably challenging because they required me to work with libraries that I hadn’t explored in depth before, like pywhatkit for WhatsApp and smtplib for emails. The most challenging aspect was configuring these libraries to handle concurrent message sending using threading, which made the code more efficient but required careful management to avoid overlapping or incomplete sends.

For learning about the libraries, I used official documentation and several online tutorials to understand the specific functions of pywhatkit and smtplib, as well as examples from GitHub and Stack Overflow. These resources were essential for troubleshooting and optimizing the code, especially for threading operations and managing variables dynamically. The most valuable takeaway from this assignment was seeing how automation could save time in real-world scenarios and realizing the potential for Python to solve day-to-day tasks efficiently.

Testing these programs involved running the code with a variety of test cases, including different phone numbers and email addresses, as well as handling both text and image messages. I feel confident in the reliability of both programs but would like to add a feature to log the status of each sent message or email for easier tracking. Overall, I enjoyed this assignment because it combined coding with practical applications, offering immediate, tangible benefits for repetitive tasks. It also inspired me to explore more automation possibilities with Python.

Here’s a more structured reflection addressing each question directly:

1. **Did you pick a suitably challenging problem to solve? Or was it too easy, or too hard?**  
   I chose problems that were suitably challenging because they required me to go beyond simple automation by incorporating threading and handling multiple external libraries (pywhatkit, smtplib, time, threading, datetime and selenium). While not overly difficult, these tasks pushed me to learn new techniques and apply them practically.
2. **What was the most challenging aspect of solving this problem?**  
   The most challenging part was implementing threading to ensure that multiple messages or emails could be sent concurrently without causing errors or interruptions. Managing the flow of tasks through threading involved learning how to handle multiple instances and ensuring each action completed before the next began.
3. **What resources did you use to learn about the libraries (or any other new features) you used?**  
   I relied on several resources, including the official documentation for pywhatkit and smtplib, Stack Overflow, and a few blog tutorials that explained threading with examples. GitHub was also helpful for finding code snippets and seeing how other developers approached similar problems.
4. **What was the most valuable thing you learned from this assignment?**  
   The most valuable takeaway was seeing the practical applications of Python automation for everyday tasks. This assignment showed me how powerful automation can be, not only in improving efficiency but also in reducing the need for repetitive manual steps.
5. **How did you test your completed product? How confident are you that it works reliably?**  
   I tested the program by running it with multiple contact numbers and email addresses, sending both text and image messages for WhatsApp, and scheduling a variety of follow-up emails. I feel medium to high success in its reliability, although I would like to add logging functionality to record each successful send, which would improve tracking.
6. **Is there something you would still like to add to this, or something it makes you want to try next?**  
   I would like to add error handling for failed sends and logging for a record of each message or email sent. This would add an extra layer of functionality and make it easier to track and troubleshoot any issues.
7. **Did you enjoy the experience of working on this assignment? Why?**  
   I enjoyed this assignment because it provided immediate, practical solutions to communication challenges. The experience taught me how Python can streamline routine tasks, and it inspired me to think of other areas where I can apply automation to improve efficiency.