UNIVERSITI POLY-TECH MALAYSIA

UPTM

NAME : **NUR ALEEYA ZAAKIRAH BINTI ZULFAKHA**		
ID NUMBER : **AM2207011634**		
LECTURER NAME : **MOHD AKMAL BIN MOHD AZMER**		SECTION NO : **01**
COURSE CODE : **EMERGING TECHNOLOGIES - SWC 2373**		SUBMISSION DATE : **10 NOVEMBER 2023**
ASSIGNMENT TITLE : **REPORT WEBEX API**		EXTENSION & LATE SUBMISSION : ALLOWED/ **DISALLOWED**
ASSIGNMENT TYPE : **INDIVIDUAL**	% OF ASSIGNMENT MARK: **40%**	RETURNING DATE :

PENALTIES :

1. 10% of the original mark will be deducted for every one-week period after the submission date
2. No work will be accepted after two weeks of the deadline
3. If you were unable to submit the coursework on time due to extenuating circumstances, you may be eligible for an extension
4. Extension will not exceed one week 3.

Declaration: I/We the undersigned confirm that I/we have read and agree to abide by these regulations on plagiarism and cheating. I/we confirm that this work is my/our own. I/we consent to appropriate storage of our work for checking to ensure that there is no plagiarism/academic cheating.

SIGNATURE(s): [illegible]

NAME: (NUR ALEEYA ZAAKIRAH BINTI ZULFAKHA)

This section may be used for feedback or other information

TABLE OF CONTENT

1. Introduction

In today's rapidly evolving digital landscape, effective communication and collaboration are integral to the success of businesses and organizations. Webex from Cisco emerges as a top platform for collaboration, providing a range of features like file sharing, video conferencing, online meetings, and more. The Webex API becomes an essential enabler, enabling developers to expand the platform's capabilities and incorporate them into their own applications, as the demand for seamless integration and customized solutions increases. This article covers the foundations of Webex, dives into the Webex API, and describes how to create third-party applications that improve cooperation.

1.1 What is Webex ?

Cisco created Webex, a flexible platform for collaboration meant to help with online meetings, web conferencing, and communication both inside and across businesses. It offers tools like screen sharing, file sharing, video conferencing, and messaging to help with the difficulties of today's remote work.

1.2 What is Webex API ?

Programmatically accessing the features and data within the Webex platform is possible for developers because of the robust toolset known as the Webex API. It changes the way users interact with the platform by allowing Webex functionalities to be integrated into external apps. To improve efficiency and teamwork, developers can use the Webex API to build custom applications, automate processes, and retrieve data from Webex.

1.3 How can third party apps can be developed with Webex API ?

Creating third-party apps that seamlessly integrate Webex features is necessary to realise Webex's full potential. In order to complete this procedure, a Webex API bridge between Webex and external apps must be built. By adding functionalities like messaging, meetings, and user data, developers may improve their apps and give consumers a more complete collaboration experience.

1.4 How to developed an app from Webex API ?

Access to the Webex API :

The required API credentials, which include a client ID and client secret, must be obtained by developers. In order to authenticate and receive access to the Webex API, these credentials are essential.

Authorization of Users :

An authentication flow is implemented for apps that require user participation. Users can provide the application access to their Webex data using this procedure. This phase guarantees that user-specific data is accessed securely and with authorization.

Application Development :

The application is coded by developers to communicate with the Webex API and accomplish the required functionality. The developer's preferences and the particular needs of the application are taken into consideration while selecting a programming language and framework.

Testing :

The application must be carefully examined and debugged in order to ensure correct operation. To ensure a complete review without affecting the real Webex platform, developers often use sandbox or testing environments to recreate interactions with the Webex API.

Deployment :

After testing is complete, release the application so that users can access it.

Security :

Developers must prioritize adherence to security and compliance guidelines, especially when the application involves handling sensitive data. It is imperative to incorporate robust security protocols to safeguard user data and ensure alignment with relevant regulatory requirements.

In summary, the Webex API provides possibilities to creative apps that meet the changing demands of companies and organizations, while also revolutionizing the way users engage with the platform. Because of its versatility and dedication to security, the Webex API is positioned as a strong tool for developers who want to design customized solutions that improve teamwork in the digital age.

2. Objectives

The objective is to provide a Webex API solving solution that lets users verify their information while in a conference call. After requesting the user's Webex token number, the application offers four options which are checking the Webex server connection, displaying user information, presenting details about five rooms, creating a new room, and sending messages to rooms. Token authentication will be implemented, an intuitive user interface with smooth navigation will be made, error handling will be included for possible problems, and the Webex API will be used for connection, user details retrieval, room information display, room construction, and message sending.

3. Literature Review

3.1 Description of the software and programming language being used

The debugging tool, which is the Python script that is attached, makes strategic use of the Webex Teams SDK in order to create a connection with the Webex API. Python is the script's preferred programming language because of its adaptability and readability. This combination improves the script's overall effectiveness while communicating with Webex Teams services, which is in accordance with its purpose of offering an intuitive command-line interface.

3.2 Dependencies

The script crucially depends on the webexteamssdk library for communication between the Python script and the Webex API. The inclusion of this library streamlines the interfacing process with Webex Teams services, significantly contributing to the overall efficacy of the application. This heavy reliance on the Webex API underscores its pivotal role as the gateway to Cisco's Webex collaboration platform. The API offers essential endpoints for accessing user information, managing rooms, and sending messages, making it an integral component for the successful execution of the script's functionalities.

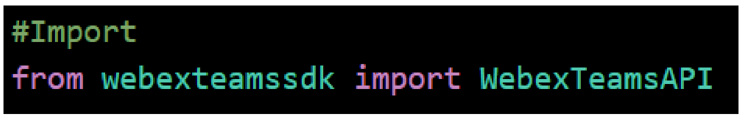
3.3 Architecture between Webex API and Application

The application's architecture is based on web development concepts and makes use of the RESTful endpoints provided by the Webex API. Using the supplied token, the script opens a connection to the Webex API, guaranteeing safe access to Webex resources. The script's different functions such as connecting test, user information retrieval, listing and displaying rooms, adding new rooms, and message sending all maps to separate API calls. The architecture makes sure that commands and data flow clearly.

3.4 How application is connected to Webex API ?

The WebexTeamsAPI class from the webexteamssdk creates the connection between the application and the Webex API. This class offers a more comfortable and Pythonic method of interacting with the Webex API by abstracting away the low-level intricacies of direct HTTP requests, in contrast to regular HTTP connections. The API token facilitates the authentication process and gives the programme easy access to the user's details, room details, and messaging features. The method prioritizes readability and ease of use in the development process, giving developers a structured and clear framework for interacting with the Webex API.

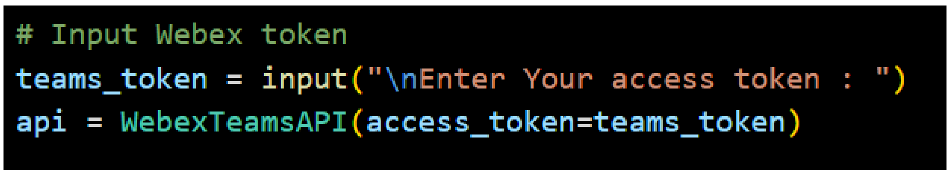
4. Development of the application



```python
#Import
from webexteamssdk import WebexTeamsAPI
```

Figure 4.1

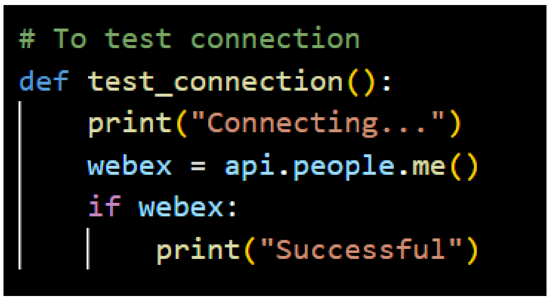
i) Figure 4.1 shows it imports the WebexTeamsAPI class from the webexteamssdk library. This class provides the necessary functionalities to interact with the Webex API. It acts as a bridge, enabling seamless communication between your Python script and the Webex collaboration platform.



```python
# Input Webex token
teams_token = input("\nEnter Your access token : ")
api = WebexTeamsAPI(access_token=teams_token)
```

Figure 4.2

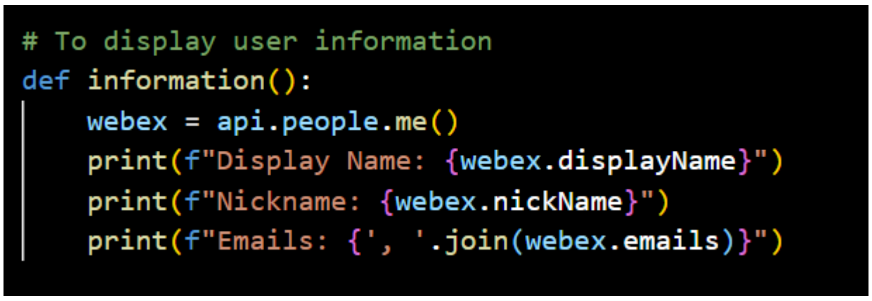
ii) In Figure 4.2 , the code asks the user to enter their Webex access token in these lines, which is the authentication key needed to access the Webex API. After user input is gathered using the input function, the WebexTeamsAPI object is instantiated with the token entered, thereby creating a connection to the Webex API.



```python
# To test connection
def test_connection():
    print("Connecting...")
    webex = api.people.me()
    if webex:
        print("Successful")
```

Figure 4.3

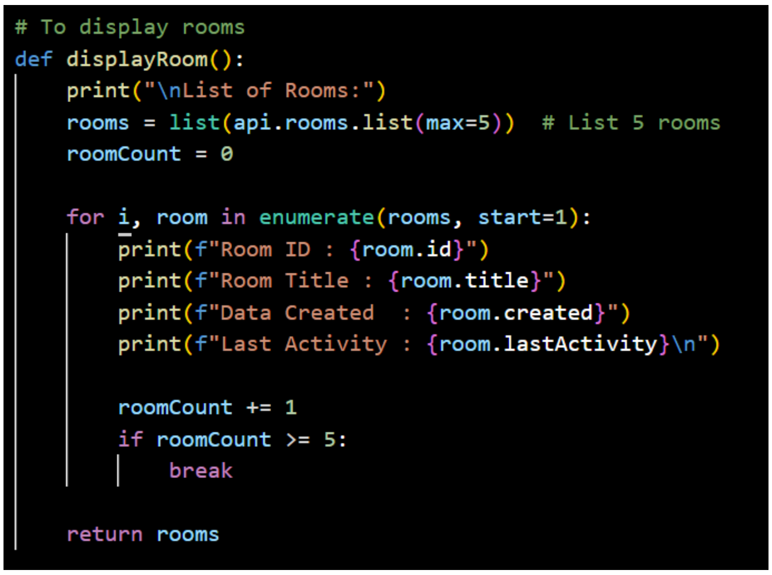
iii) test_connection is for option 0 and it is to confirm that the Webex API is reachable. After printing a "Connecting..." message, it uses api.people.me() to try and get information about the authenticated user; if that works, it prints a confirmation message. This checks to see if the application can connect to the Webex API in the first place.



```python
# To display user information
def information():
    webex = api.people.me()
    print(f"Display Name: {webex.displayName}")
    print(f"Nickname: {webex.nickName}")
    print(f"Emails: {', '.join(webex.emails)}")
```

Figure 4.4

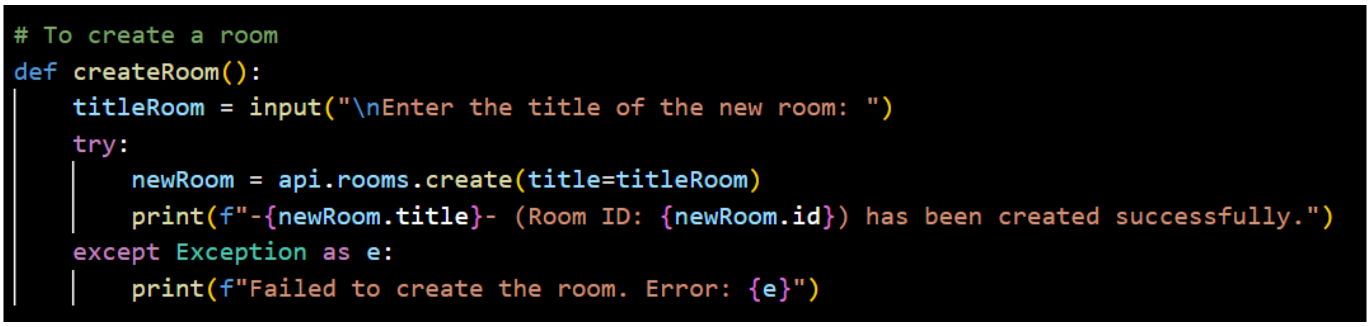
iv) The information function retrieves the user's information using the me() method and prints their display name, nickname, and emails.



```python
# To display rooms
def displayRoom():
    print("\nList of Rooms:")
    rooms = list(api.rooms.list(max=5))  # List 5 rooms
    roomCount = 0

    for i, room in enumerate(rooms, start=1):
        print(f"Room ID : {room.id}")
        print(f"Room Title : {room.title}")
        print(f"Data Created  : {room.created}")
        print(f"Last Activity : {room.lastActivity}\n")

        roomCount += 1
        if roomCount >= 5:
            break

    return rooms
```

FIgure 4.5

v) The displayRoom function calls the listRooms function to get a list of rooms. It then iterates through the rooms, printing details such as Room ID, Room Title, Date Created, and Last Activity for up to 5 rooms.

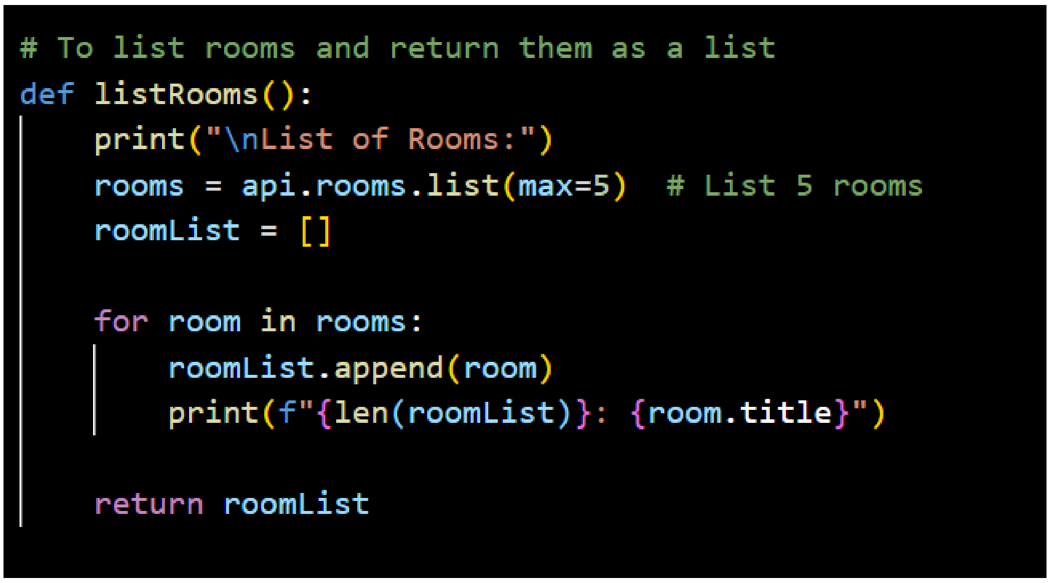


```
# To create a room
def createRoom():
    titleRoom = input("\nEnter the title of the new room: ")
    try:
        newRoom = api.rooms.create(title=titleRoom)
        print(f"-{newRoom.title}- (Room ID: {newRoom.id}) has been created successfully.")
    except Exception as e:
        print(f"Failed to create the room. Error: {e}")
```

Figure 4.6

vi) The createRoom function prompts the user to enter the title of a new room.

It attempts to create the room using the rooms.create() method and prints a success message with the room's title and ID upon success. If an error occurs, it prints an error message.



```
# To list rooms and return them as a list
def listRooms():
    print("\nList of Rooms:")
    rooms = api.rooms.list(max=5)  # List 5 rooms
    roomList = []

    for room in rooms:
        roomList.append(room)
        print(f"{len(roomList)}: {room.title}")

    return roomList
```

Figure 4.7

vii) The listRooms function retrieves a list of rooms (up to a maximum of 5) using the rooms.list() method. It prints the titles of the rooms along with their corresponding numbers and returns the list of rooms.

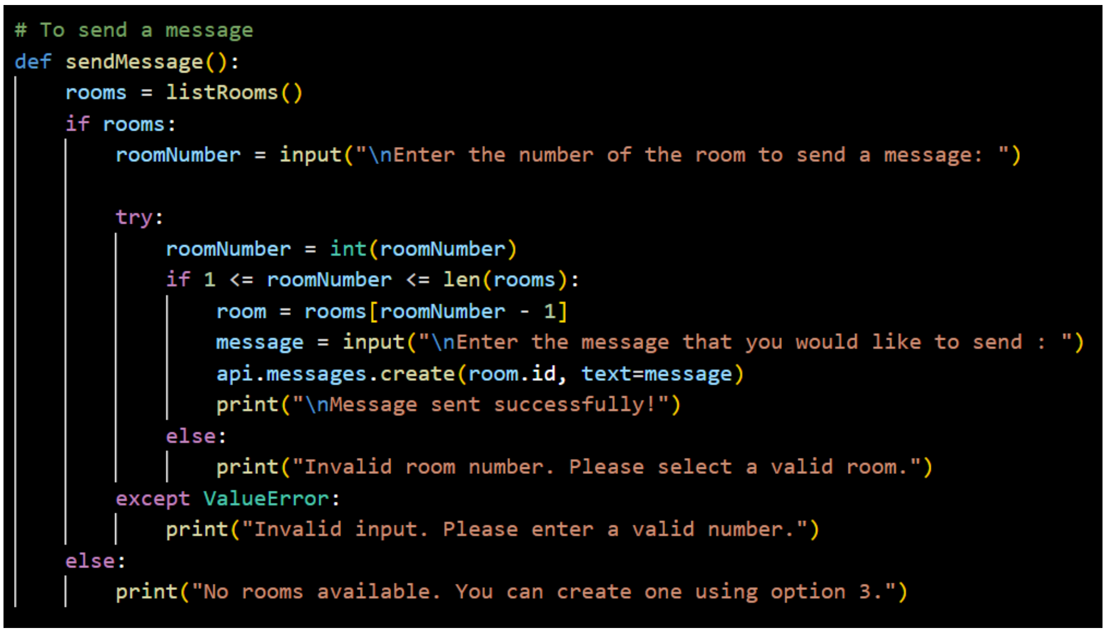

```python
# To send a message
def sendMessage():
    rooms = listRooms()
    if rooms:
        roomNumber = input("\nEnter the number of the room to send a message: ")

        try:
            roomNumber = int(roomNumber)
            if 1 <= roomNumber <= len(rooms):
                room = rooms[roomNumber - 1]
                message = input("\nEnter the message that you would like to send : ")
                api.messages.create(room.id, text=message)
                print("\nMessage sent successfully!")
            else:
                print("Invalid room number. Please select a valid room.")
        except ValueError:
            print("Invalid input. Please enter a valid number.")
    else:
        print("No rooms available. You can create one using option 3.")
```

Figure 4.8

viii) The sendMessage function allows the user to choose a room, input a message, and send it. It calls the listRooms function to get a list of rooms, validates user input, and uses the messages.create() method to send the message. It handles errors such as invalid room numbers or input values.

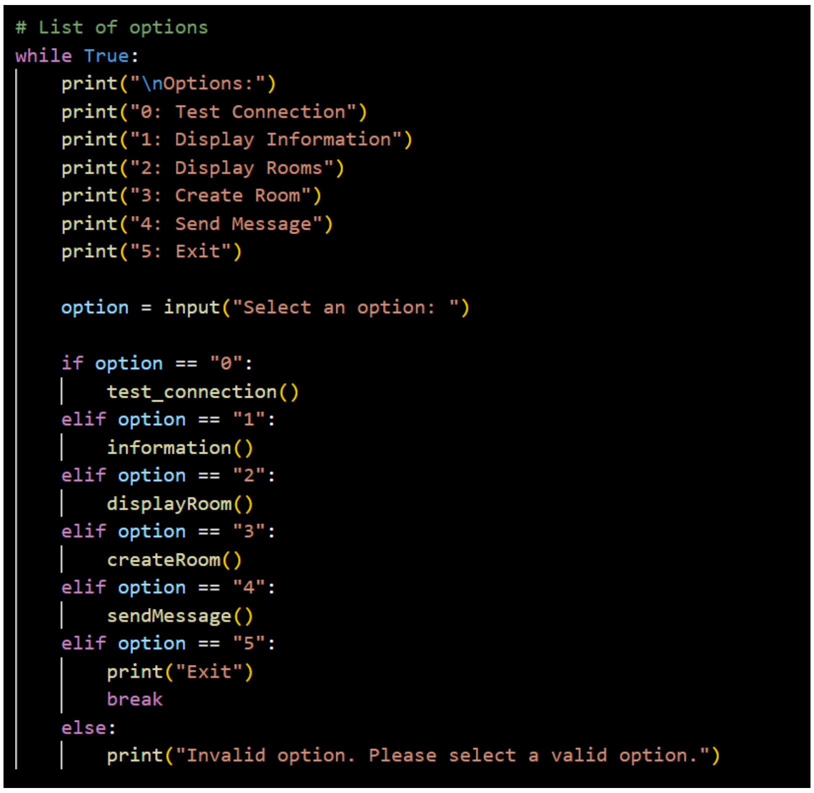
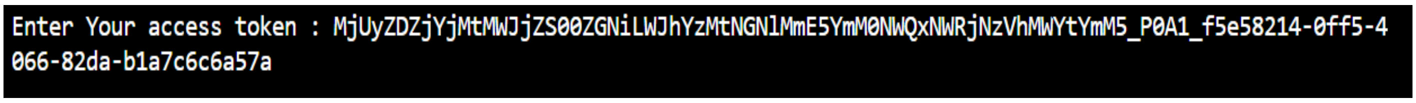

```python
# List of options
while True:
    print("\nOptions:")
    print("0: Test Connection")
    print("1: Display Information")
    print("2: Display Rooms")
    print("3: Create Room")
    print("4: Send Message")
    print("5: Exit")

    option = input("Select an option: ")

    if option == "0":
        test_connection()
    elif option == "1":
        information()
    elif option == "2":
        displayRoom()
    elif option == "3":
        createRoom()
    elif option == "4":
        sendMessage()
    elif option == "5":
        print("Exit")
        break
    else:
        print("Invalid option. Please select a valid option.")
```

Figure 4.9

ix) The script uses an infinite loop to present a menu of options to the user. Based on the user's selection, it calls the corresponding function. It also allows users to exit by choosing option 5. For invalid options, it prints a message guiding users to select a valid option.

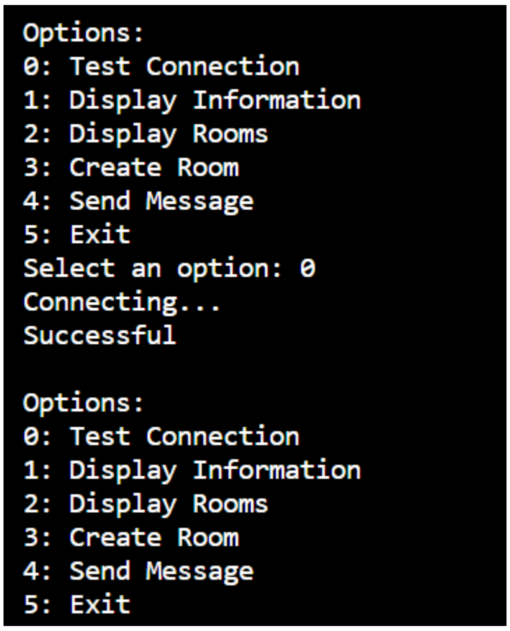
5. Testing of the apps



```
Enter Your access token : MjUyZDZjYjMtMWJjZS00ZGNiLWJhYzMtNGNlMmE5YmM0NWQxNWRjNzVhMWYtYmM5_P0A1_f5e58214-0ff5-4
066-82da-b1a7c6c6a57a
```

Figure 5.1

- The user is prompted to input their Webex access token.



```
Options:
0: Test Connection
1: Display Information
2: Display Rooms
3: Create Room
4: Send Message
5: Exit
Select an option: 0
Connecting...
Successful

Options:
0: Test Connection
1: Display Information
2: Display Rooms
3: Create Room
4: Send Message
5: Exit
```

Figure 5.2

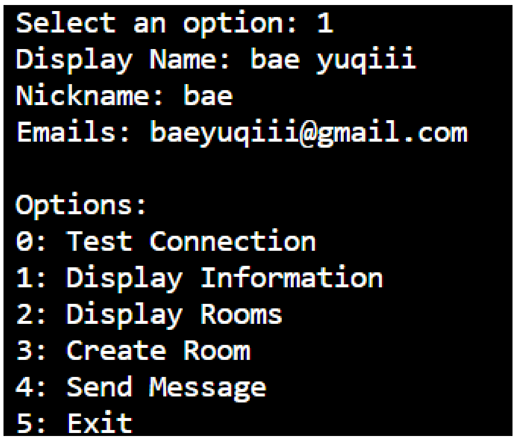
- The script presents a menu of options numbered from 0 to 5.

- The user is given the following choices from 0-5

- The user selects option 0 to test the connection.

- The script prints "Connecting..." and then "Successful," indicating a successful connection to Webex.

- The user is given the option again.



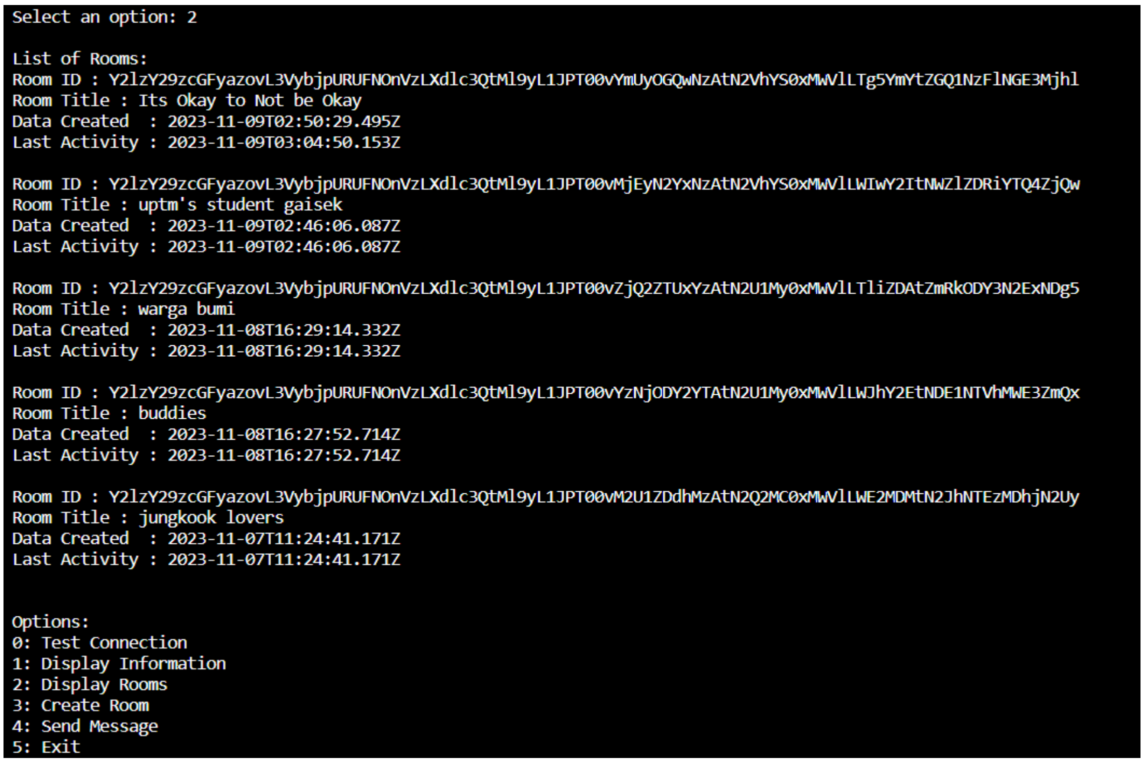
```
Select an option: 1
Display Name: bae yuqiii
Nickname: bae
Emails: baeyuqiii@gmail.com

Options:
0: Test Connection
1: Display Information
2: Display Rooms
3: Create Room
4: Send Message
5: Exit
```

Figure 5.3

- The user selects option 1 to display their Webex information.

- The script prints the display name, nickname, and emails associated with the Webex account.

- The user is given the option again.



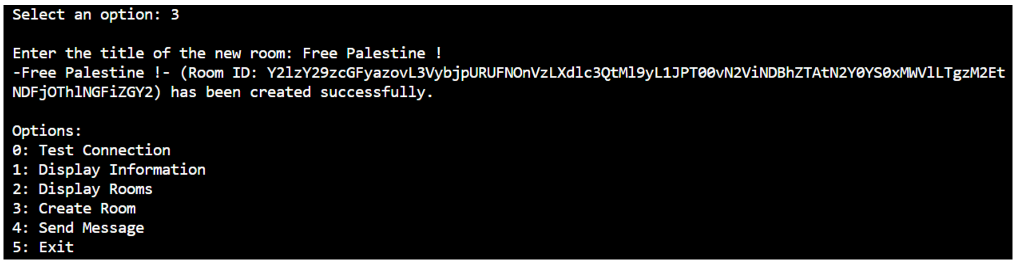
```
Select an option: 2

List of Rooms:
Room ID : Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vYmUyOGQwNzAtN2VhYS0xMWVlLTg5YmYtZGQ1NzFlNGE3Mjhl
Room Title : Its Okay to Not be Okay
Data Created  : 2023-11-09T02:50:29.495Z
Last Activity : 2023-11-09T03:04:50.153Z

Room ID : Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vMjEyN2YxNzAtN2VhYS0xMWVlLWIwY2ItNWZlZDRiYTQ4ZjQw
Room Title : uptm's student gaisek
Data Created  : 2023-11-09T02:46:06.087Z
Last Activity : 2023-11-09T02:46:06.087Z

Room ID : Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vZjQ2ZTUxYzAtN2U1My0xMWVlLTliZDAtZmRkODY3N2ExNDg5
Room Title : warga bumi
Data Created  : 2023-11-08T16:29:14.332Z
Last Activity : 2023-11-08T16:29:14.332Z

Room ID : Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vYzNjODY2YTAtN2U1My0xMWVlLWJhY2EtNDE1NTVhMWE3ZmQx
Room Title : buddies
Data Created  : 2023-11-08T16:27:52.714Z
Last Activity : 2023-11-08T16:27:52.714Z

Room ID : Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vM2U1ZDdhMzAtN2Q2MC0xMWVlLWE2MDMtN2JhNTEzMDhjN2Uy
Room Title : jungkook lovers
Data Created  : 2023-11-07T11:24:41.171Z
Last Activity : 2023-11-07T11:24:41.171Z


Options:
0: Test Connection
1: Display Information
2: Display Rooms
3: Create Room
4: Send Message
5: Exit
```

Figure 5.4

- The user selects option 2 to display a list of rooms.

- The script prints information about the first 5 rooms, including Room ID, Room Title, Date Created, and Last Activity.

- The user is given the option again.



```
Select an option: 3

Enter the title of the new room: Free Palestine !
-Free Palestine !- (Room ID: Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vN2ViNDBhZTAtN2Y0YS0xMWVlLTgzM2Et
NDFjOThlNGFiZGY2) has been created successfully.

Options:
0: Test Connection
1: Display Information
2: Display Rooms
3: Create Room
4: Send Message
5: Exit
```

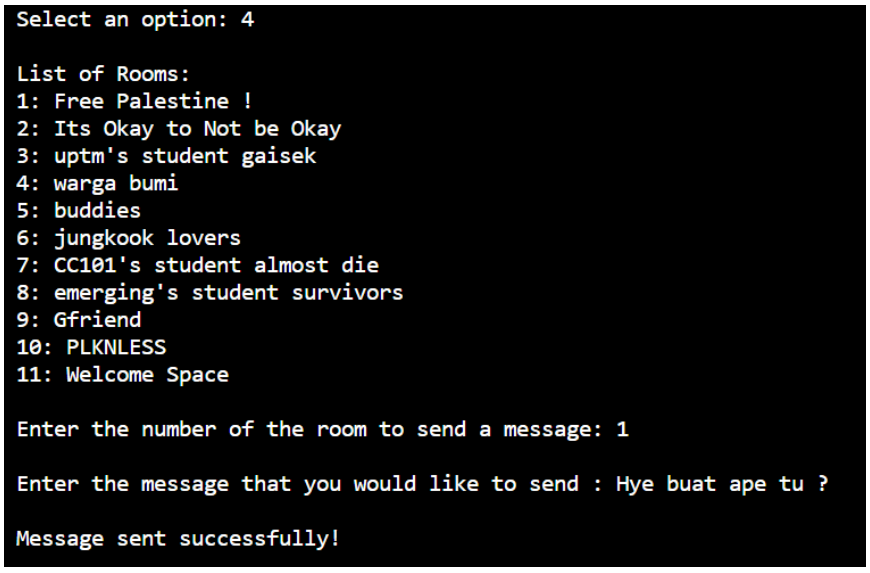
Figure 5.5

- The user selects option 3 to create a new room.

- The script prompts the user to enter the title of the new room.

- It then prints a success message with the newly created room's title and ID.

- The user is given the option again.



```
Select an option: 4

List of Rooms:
1: Free Palestine !
2: Its Okay to Not be Okay
3: uptm's student gaisek
4: warga bumi
5: buddies
6: jungkook lovers
7: CC101's student almost die
8: emerging's student survivors
9: Gfriend
10: PLKNLESS
11: Welcome Space

Enter the number of the room to send a message: 1

Enter the message that you would like to send : Hye buat ape tu ?

Message sent successfully!
```

Figure 5.6

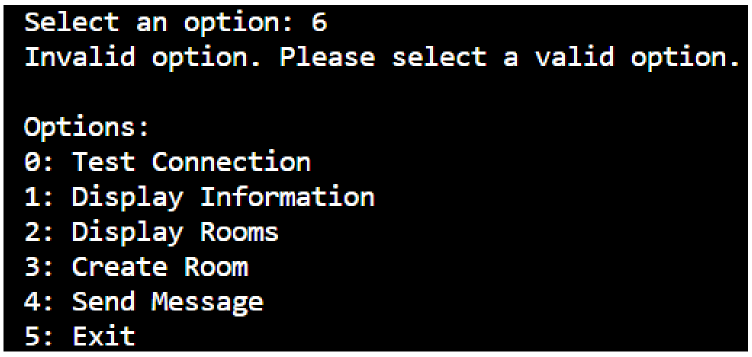
- The user selects option 4 to send a message.

- The script displays a list of rooms and prompts the user to enter the number of the room to send a message.

- The user inputs the message .

- The script prints "Message sent successfully!"

- The user is given the option again.



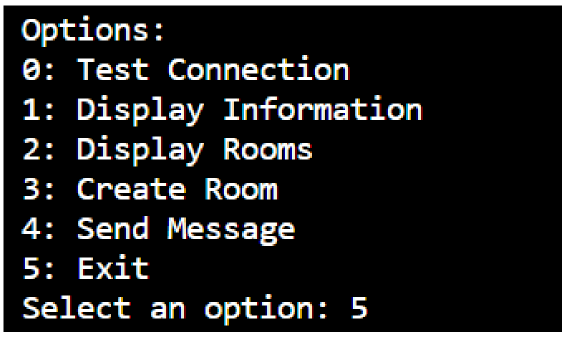
```
Select an option: 6
Invalid option. Please select a valid option.

Options:
0: Test Connection
1: Display Information
2: Display Rooms
3: Create Room
4: Send Message
5: Exit
```

Figure 5.7

- The user inputs an invalid option, 6.

- The script responds with "Invalid option. Please select a valid option."

- The user is given the option again.



```
Options:
0: Test Connection
1: Display Information
2: Display Rooms
3: Create Room
4: Send Message
5: Exit
Select an option: 5
```

Figure 5.8

- The user selects option 5 to exit the application.

- The script prints "Exit," and the loop terminates, ending the script.

6. Conclusion

We have effectively used the Webex API to create a reliable troubleshooting tool during this project. The Webex collaboration platform is seamlessly connected to the Python script via the Webex Teams SDK. Our application has all of the necessary functions, such as sending messages, listing rooms, creating rooms, connecting tests, and displaying user information. Python is used to ensure readability and flexibility of the code. We emphasize the importance of the Webexteamssdk library and the Webex API, and we construct our application in a modular fashion to make it easier to maintain.With a user-friendly menu-driven interface, we have provided an effective and comprehensive tool for troubleshooting within the Webex environment, complete with robust error handling and user guidance.

7. Reference

I. *Cisco Webex*. (2023, November 7). Wikipedia. https://en.wikipedia.org/wiki/Cisco_Webex

II. *Configure Third-Party Integrations on a Webex Site*. (n.d.). Webex Help Center. https://help.webex.com/en-us/article/k6j1db/Configure-Third-Party-Integrations-on-a-Webex-Site

III. *Messages - Create a Message*. (n.d.). Webex for Developers. https://developer.webex.com/docs/api/v1/messages/create-a-message

IV. *Webex App | Set up your Personal Room*. (n.d.). Webex Help Center. https://help.webex.com/en-us/article/e7l52d/Webex-App-%7C-Set-up-your-Personal-Room

V. *Rooms - Create a Room*. (n.d.). Webex for Developers. https://developer.webex.com/docs/api/v1/rooms/create-a-room

8. Github Link

https://github.com/baeyuqiii/SWC2373-EMERGING-TECHNOLOGY