Problem Statement

Design and development of electronic mail system (Read, write, send and delete operations)

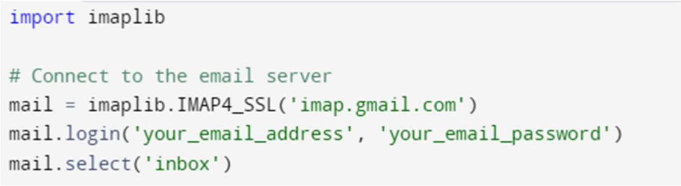
Question demands the development of an electronic mail system that can perform essential operations such as reading, writing, sending, and deleting messages. The main objective of the project is to design and develop a reliable email system that can be used by individuals and organizations to communicate effectively.

Solution of problem statement

Designing and developing an electronic mail system in Python requires several components to be implemented such as reading, writing, sending, and deleting emails. In this answer, we will discuss the steps to create each of these functionalities.

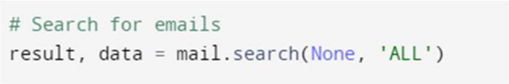
1. Reading Emails:

To read emails, we will use the built-in Python module named "IMAPLIB" which allows us to connect to an email server and retrieve emails. To use this module, we will first establish a connection to the email server using the following code:



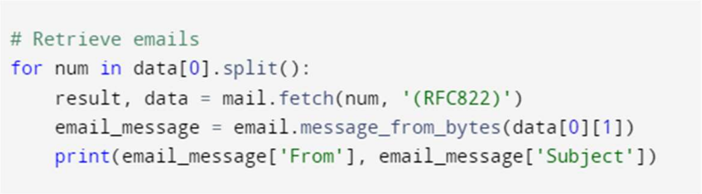
```
import imaplib

# Connect to the email server
mail = imaplib.IMAP4_SSL('imap.gmail.com')
mail.login('your_email_address', 'your_email_password')
mail.select('inbox')
```

After connecting to the server, we will select the inbox and search for the emails using the search method provided by the IMAPLIB module:



```
# Search for emails
result, data = mail.search(None, 'ALL')
```

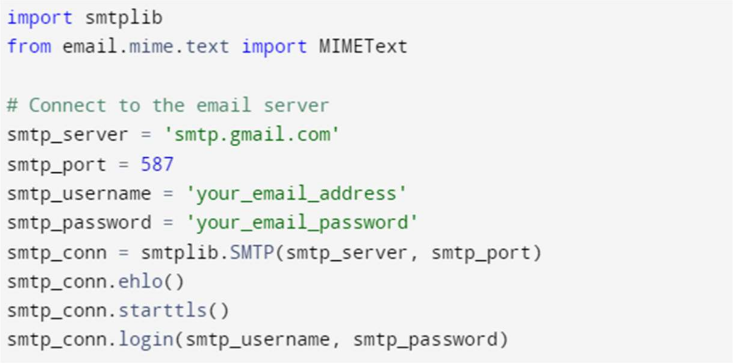
The result will contain the status of the search operation, and data will contain the email IDs of all the emails in the inbox. We can then iterate through these IDs and retrieve the email using the following code:



```
# Retrieve emails
for num in data[0].split():
    result, data = mail.fetch(num, '(RFC822)')
    email_message = email.message_from_bytes(data[0][1])
    print(email_message['From'], email_message['Subject'])
```

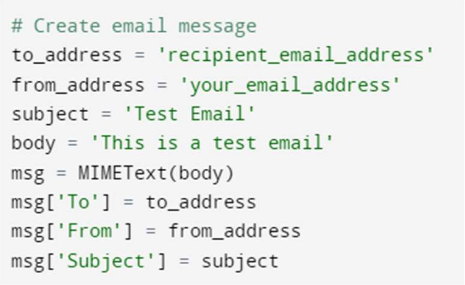
2. Writing Emails:

To write emails, we will use the built-in Python module named "smtplib" which allows us to send emails using Simple Mail Transfer Protocol (SMTP). To use this module, we will first establish a connection to the email server using the following code:



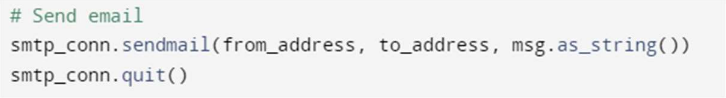
```python
import smtplib
from email.mime.text import MIMEText

# Connect to the email server
smtp_server = 'smtp.gmail.com'
smtp_port = 587
smtp_username = 'your_email_address'
smtp_password = 'your_email_password'
smtp_conn = smtplib.SMTP(smtp_server, smtp_port)
smtp_conn.ehlo()
smtp_conn.starttls()
smtp_conn.login(smtp_username, smtp_password)
```

After connecting to the server, we will create an email message using the MIMEText class provided by the email module:



```python
# Create email message
to_address = 'recipient_email_address'
from_address = 'your_email_address'
subject = 'Test Email'
body = 'This is a test email'
msg = MIMEText(body)
msg['To'] = to_address
msg['From'] = from_address
msg['Subject'] = subject
```

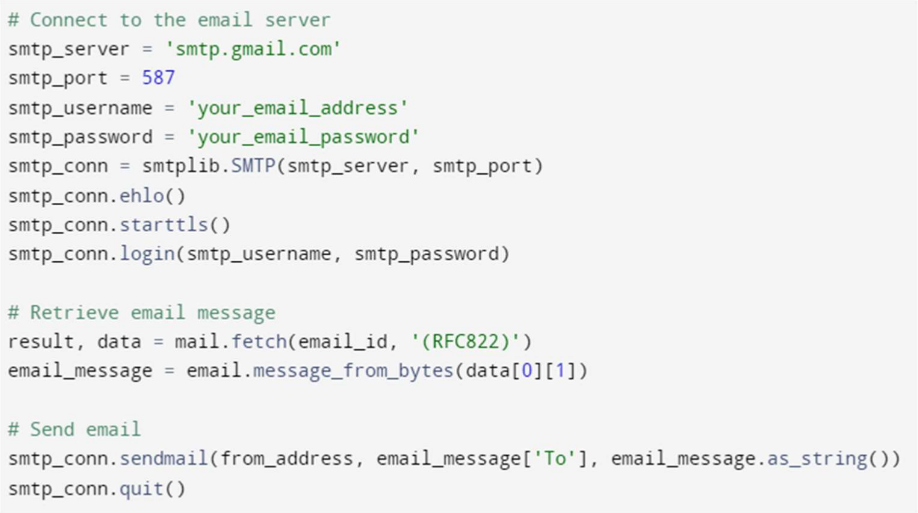
We can then send the email using the following code:



```python
# Send email
smtp_conn.sendmail(from_address, to_address, msg.as_string())
smtp_conn.quit()
```

3. Sending Emails:

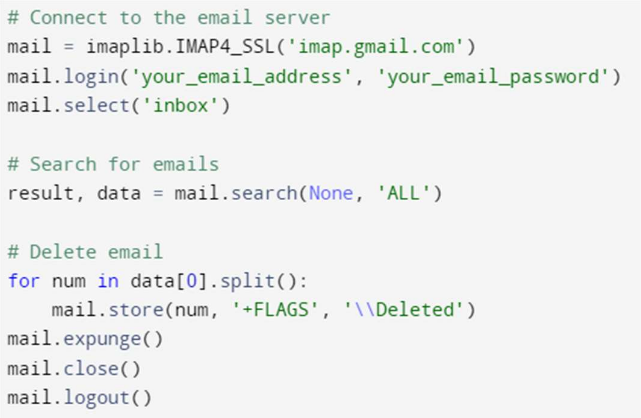
To send emails, we will use the built-in Python module named "smtplib" which allows us to send emails using Simple Mail Transfer Protocol (SMTP). The process of sending emails is similar to writing emails, except that we will use the message ID to retrieve the email message from the server:



```
# Connect to the email server
smtp_server = 'smtp.gmail.com'
smtp_port = 587
smtp_username = 'your_email_address'
smtp_password = 'your_email_password'
smtp_conn = smtplib.SMTP(smtp_server, smtp_port)
smtp_conn.ehlo()
smtp_conn.starttls()
smtp_conn.login(smtp_username, smtp_password)

# Retrieve email message
result, data = mail.fetch(email_id, '(RFC822)')
email_message = email.message_from_bytes(data[0][1])

# Send email
smtp_conn.sendmail(from_address, email_message['To'], email_message.as_string())
smtp_conn.quit()
```

4. Deleting Emails:

To delete emails, we will use the built-in Python module named "IMAPLIB" which allows us to connect to an email server and retrieve emails. The process of deleting emails involves marking the email as deleted and then expunging it from the server. Here is the code to delete an email:

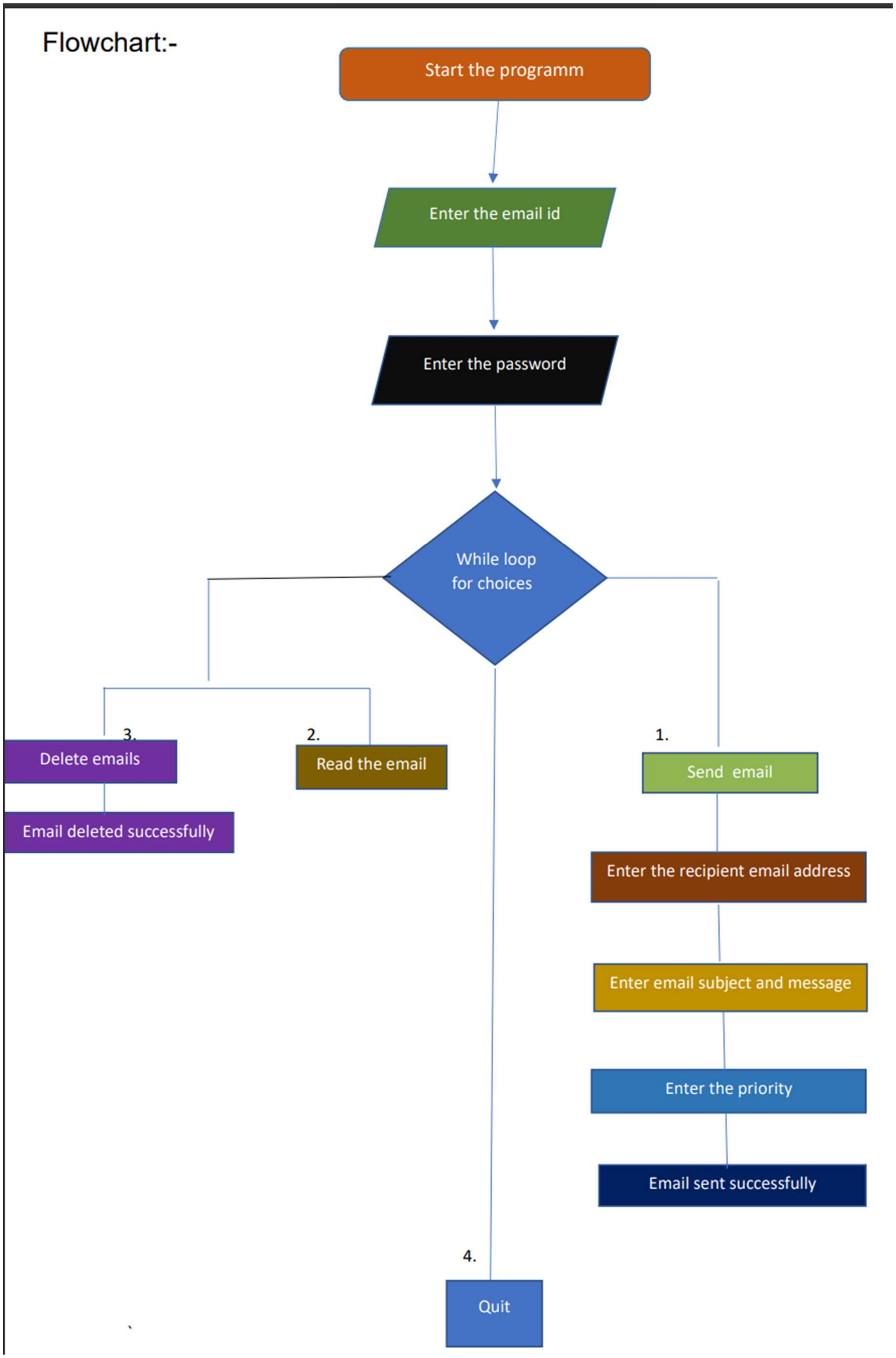


```
# Connect to the email server
mail = imaplib.IMAP4_SSL('imap.gmail.com')
mail.login('your_email_address', 'your_email_password')
mail.select('inbox')

# Search for emails
result, data = mail.search(None, 'ALL')

# Delete email
for num in data[0].split():
    mail.store(num, '+FLAGS', '\\Deleted')
mail.expunge()
mail.close()
mail.logout()
```

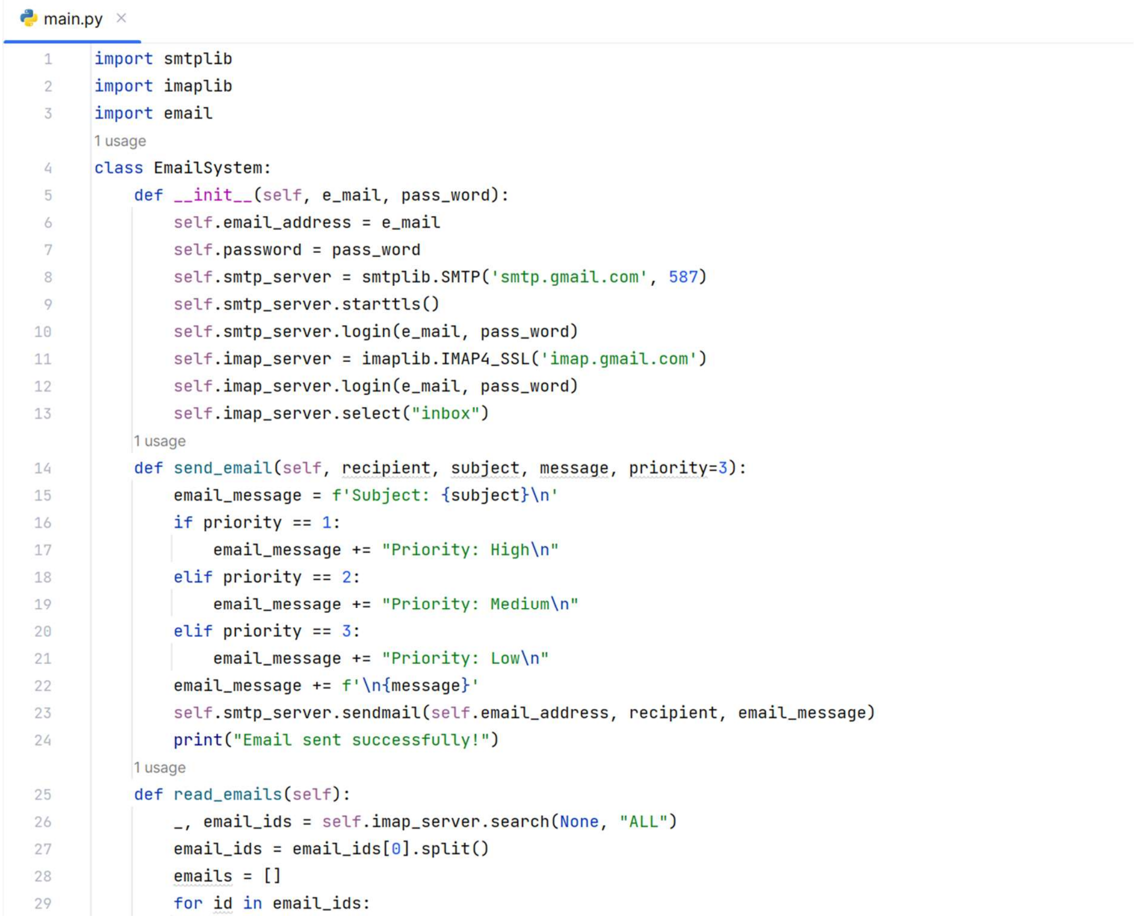
In the above code, we first establish a connection to the email server and select the inbox. We then search for the emails and mark them as deleted using the store method provided by the IMAPLIB module. Finally, we expunge the deleted emails from the server and close the connection.

Overall, designing and developing an electronic mail system in Python involves using the built-in modules such as IMAPLIB and smtplib to connect to the email server, retrieve, write, send, and delete emails.

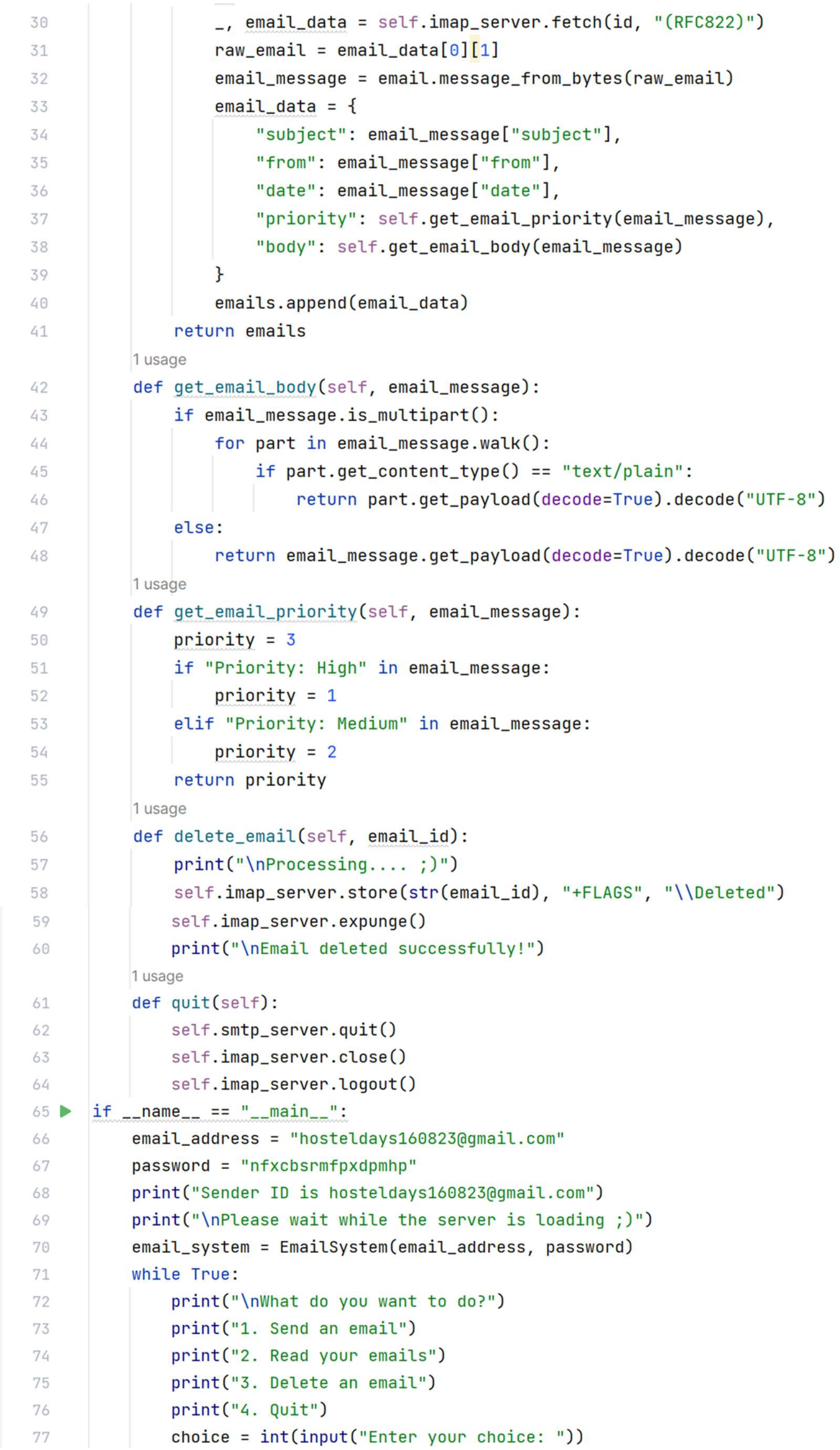
Flowchart:-


Start the programm
Enter the email id
Enter the password
While loop for choices
3.
Delete emails
Email deleted successfully
2.
Read the email
1.
Send email
Enter the recipient email address
Enter email subject and message
Enter the priority
Email sent successfully
4.
Quit

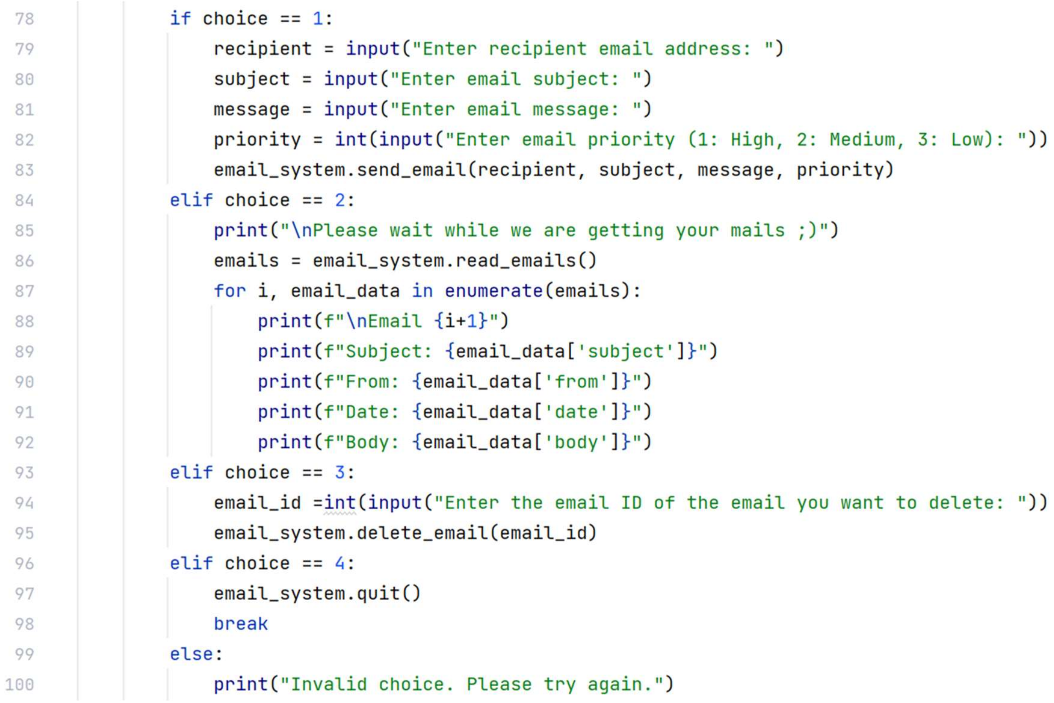
Executed Code :



```python
import smtplib
import imaplib
import email
class EmailSystem:
    def __init__(self, e_mail, pass_word):
        self.email_address = e_mail
        self.password = pass_word
        self.smtp_server = smtplib.SMTP('smtp.gmail.com', 587)
        self.smtp_server.starttls()
        self.smtp_server.login(e_mail, pass_word)
        self.imap_server = imaplib.IMAP4_SSL('imap.gmail.com')
        self.imap_server.login(e_mail, pass_word)
        self.imap_server.select("inbox")
    def send_email(self, recipient, subject, message, priority=3):
        email_message = f'Subject: {subject}\n'
        if priority == 1:
            email_message += "Priority: High\n"
        elif priority == 2:
            email_message += "Priority: Medium\n"
        elif priority == 3:
            email_message += "Priority: Low\n"
        email_message += f'\n{message}'
        self.smtp_server.sendmail(self.email_address, recipient, email_message)
        print("Email sent successfully!")
    def read_emails(self):
        _, email_ids = self.imap_server.search(None, "ALL")
        email_ids = email_ids[0].split()
        emails = []
        for id in email_ids:
```



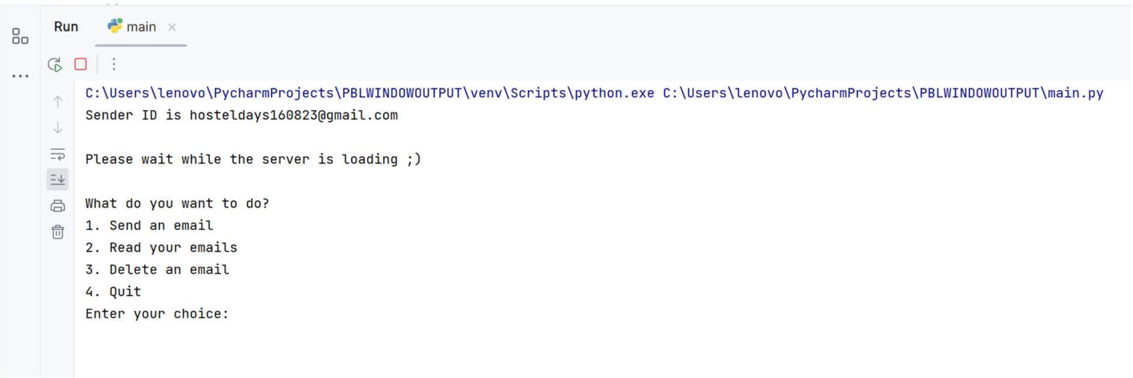
```
            _, email_data = self.imap_server.fetch(id, "(RFC822)")
            raw_email = email_data[0][1]
            email_message = email.message_from_bytes(raw_email)
            email_data = {
                "subject": email_message["subject"],
                "from": email_message["from"],
                "date": email_message["date"],
                "priority": self.get_email_priority(email_message),
                "body": self.get_email_body(email_message)
            }
            emails.append(email_data)
        return emails
    1 usage
    def get_email_body(self, email_message):
        if email_message.is_multipart():
            for part in email_message.walk():
                if part.get_content_type() == "text/plain":
                    return part.get_payload(decode=True).decode("UTF-8")
        else:
            return email_message.get_payload(decode=True).decode("UTF-8")
    1 usage
    def get_email_priority(self, email_message):
        priority = 3
        if "Priority: High" in email_message:
            priority = 1
        elif "Priority: Medium" in email_message:
            priority = 2
        return priority
    1 usage
    def delete_email(self, email_id):
        print("\nProcessing.... ;)")
        self.imap_server.store(str(email_id), "+FLAGS", "\\Deleted")
        self.imap_server.expunge()
        print("\nEmail deleted successfully!")
    1 usage
    def quit(self):
        self.smtp_server.quit()
        self.imap_server.close()
        self.imap_server.logout()
if __name__ == "__main__":
    email_address = "hosteldays160823@gmail.com"
    password = "nfxcbsrmfpxdpmhp"
    print("Sender ID is hosteldays160823@gmail.com")
    print("\nPlease wait while the server is loading ;)")
    email_system = EmailSystem(email_address, password)
    while True:
        print("\nWhat do you want to do?")
        print("1. Send an email")
        print("2. Read your emails")
        print("3. Delete an email")
        print("4. Quit")
        choice = int(input("Enter your choice: "))
```



```python
        if choice == 1:
            recipient = input("Enter recipient email address: ")
            subject = input("Enter email subject: ")
            message = input("Enter email message: ")
            priority = int(input("Enter email priority (1: High, 2: Medium, 3: Low): "))
            email_system.send_email(recipient, subject, message, priority)
        elif choice == 2:
            print("\nPlease wait while we are getting your mails ;)")
            emails = email_system.read_emails()
            for i, email_data in enumerate(emails):
                print(f"\nEmail {i+1}")
                print(f"Subject: {email_data['subject']}")
                print(f"From: {email_data['from']}")
                print(f"Date: {email_data['date']}")
                print(f"Body: {email_data['body']}")
        elif choice == 3:
            email_id =int(input("Enter the email ID of the email you want to delete: "))
            email_system.delete_email(email_id)
        elif choice == 4:
            email_system.quit()
            break
        else:
            print("Invalid choice. Please try again.")
```

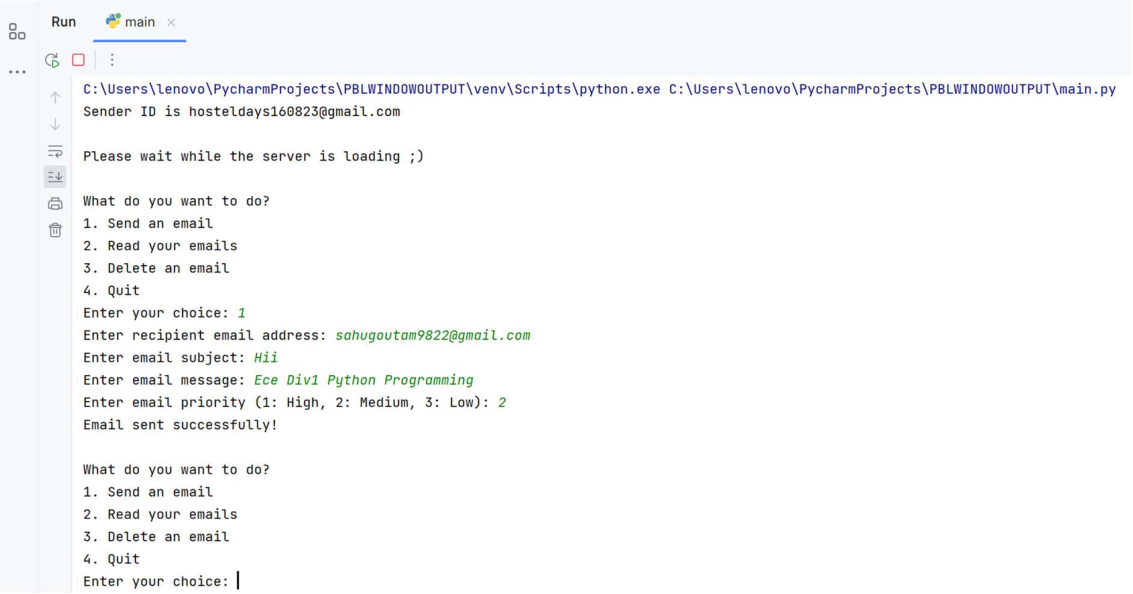
Output :

#Choice Interface



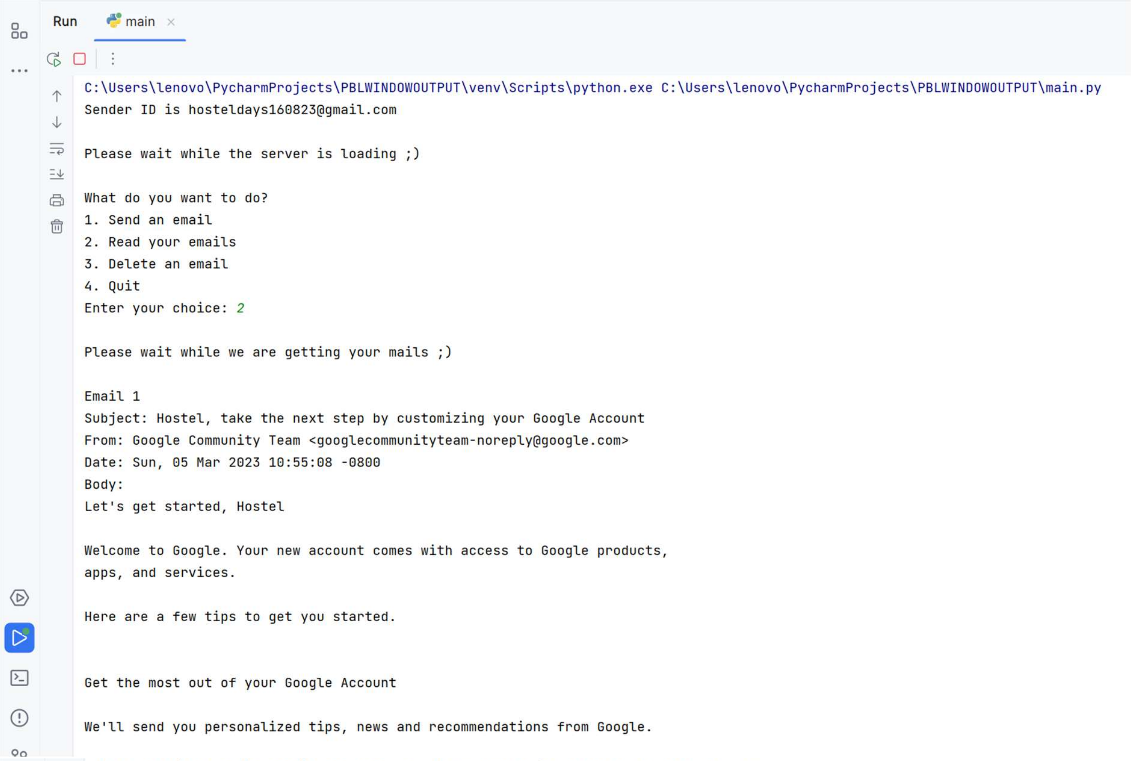
```
C:\Users\lenovo\PycharmProjects\PBLWINDOWOUTPUT\venv\Scripts\python.exe C:\Users\lenovo\PycharmProjects\PBLWINDOWOUTPUT\main.py
Sender ID is hosteldays160823@gmail.com

Please wait while the server is loading ;)

What do you want to do?
1. Send an email
2. Read your emails
3. Delete an email
4. Quit
Enter your choice:
```

#If ‘1’ is the choice

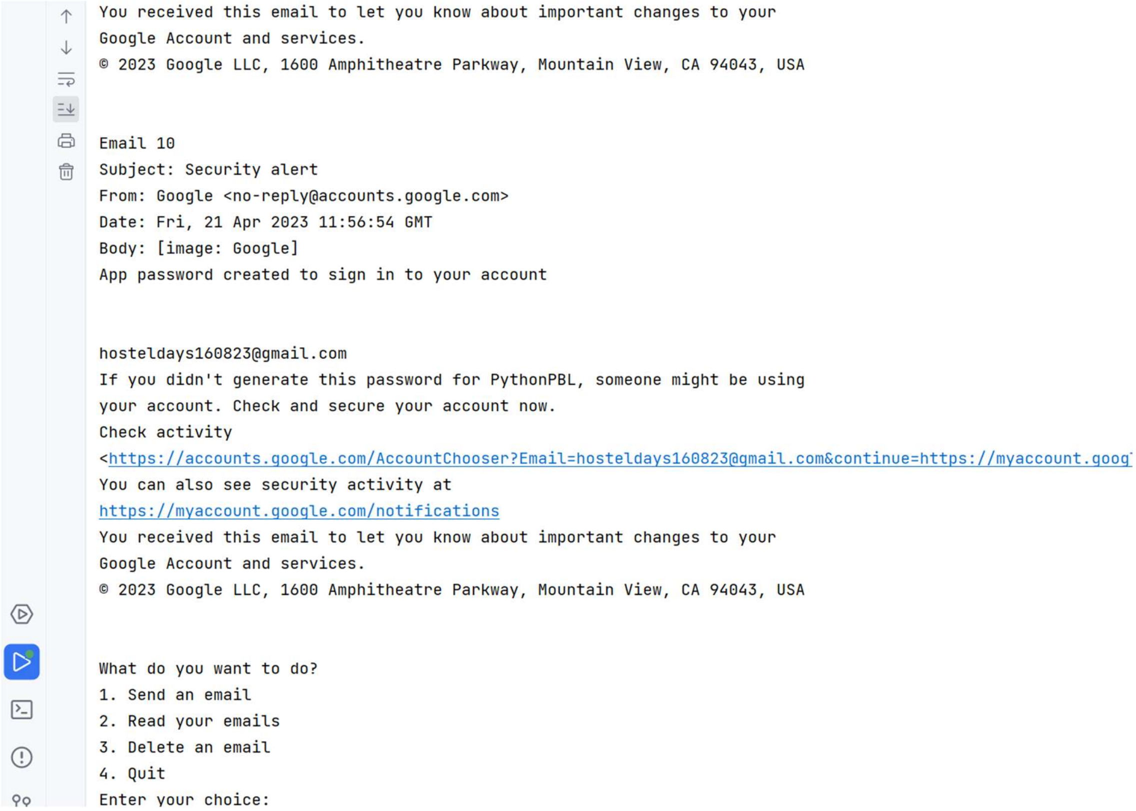


```
C:\Users\lenovo\PycharmProjects\PBLWINDOWOUTPUT\venv\Scripts\python.exe C:\Users\lenovo\PycharmProjects\PBLWINDOWOUTPUT\main.py
Sender ID is hosteldays160823@gmail.com

Please wait while the server is loading ;)

What do you want to do?
1. Send an email
2. Read your emails
3. Delete an email
4. Quit
Enter your choice: 1
Enter recipient email address: sahugoutam9822@gmail.com
Enter email subject: Hii
Enter email message: Ece Div1 Python Programming
Enter email priority (1: High, 2: Medium, 3: Low): 2
Email sent successfully!

What do you want to do?
1. Send an email
2. Read your emails
3. Delete an email
4. Quit
Enter your choice:
```

#If ‘2’ is the choice

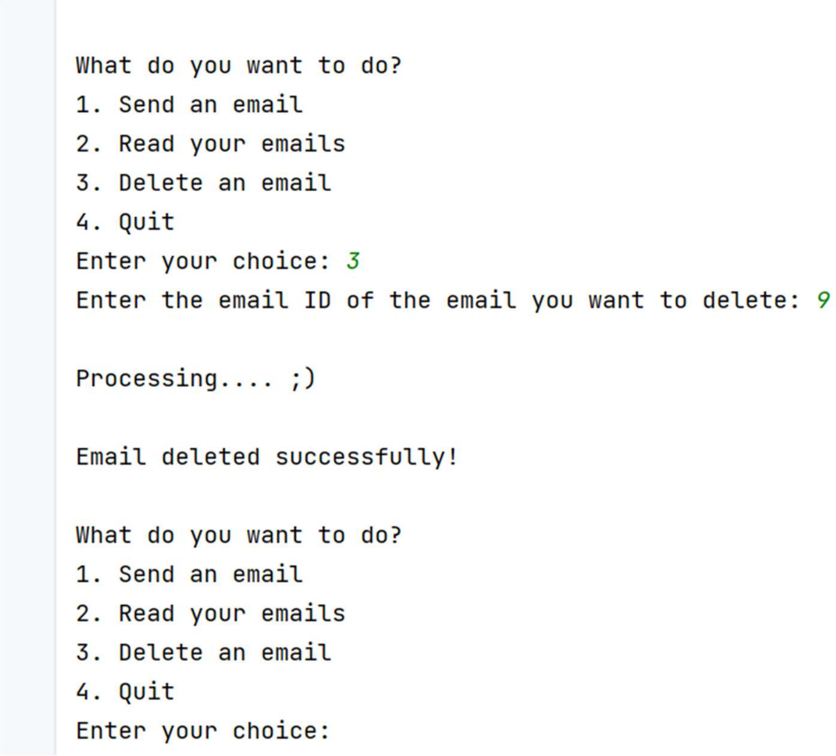


```
C:\Users\lenovo\PycharmProjects\PBLWINDOWOUTPUT\venv\Scripts\python.exe C:\Users\lenovo\PycharmProjects\PBLWINDOWOUTPUT\main.py
Sender ID is hosteldays160823@gmail.com

Please wait while the server is loading ;)

What do you want to do?
1. Send an email
2. Read your emails
3. Delete an email
4. Quit
Enter your choice: 2

Please wait while we are getting your mails ;)

Email 1
Subject: Hostel, take the next step by customizing your Google Account
From: Google Community Team <googlecommunityteam-noreply@google.com>
Date: Sun, 05 Mar 2023 10:55:08 -0800
Body:
Let's get started, Hostel

Welcome to Google. Your new account comes with access to Google products,
apps, and services.

Here are a few tips to get you started.


Get the most out of your Google Account

We'll send you personalized tips, news and recommendations from Google.
```



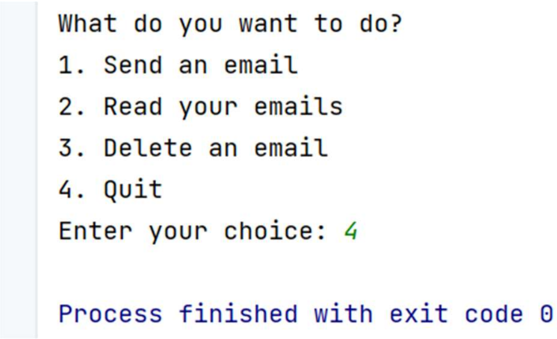
```
You received this email to let you know about important changes to your
Google Account and services.
© 2023 Google LLC, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA


Email 10
Subject: Security alert
From: Google <no-reply@accounts.google.com>
Date: Fri, 21 Apr 2023 11:56:54 GMT
Body: [image: Google]
App password created to sign in to your account


hosteldays160823@gmail.com
If you didn't generate this password for PythonPBL, someone might be using
your account. Check and secure your account now.
Check activity
<https://accounts.google.com/AccountChooser?Email=hosteldays160823@gmail.com&continue=https://myaccount.goog
You can also see security activity at
https://myaccount.google.com/notifications
You received this email to let you know about important changes to your
Google Account and services.
© 2023 Google LLC, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA


What do you want to do?
1. Send an email
2. Read your emails
3. Delete an email
4. Quit
Enter your choice:
```

#If ‘3’ is the choice



```
What do you want to do?
1. Send an email
2. Read your emails
3. Delete an email
4. Quit
Enter your choice: 3
Enter the email ID of the email you want to delete: 9

Processing.... ;)

Email deleted successfully!

What do you want to do?
1. Send an email
2. Read your emails
3. Delete an email
4. Quit
Enter your choice:
```

#If ‘4’ is the choice



```
What do you want to do?
1. Send an email
2. Read your emails
3. Delete an email
4. Quit
Enter your choice: 4

Process finished with exit code 0
```

Conclusion

After doing this Project Based Learning activity we understood the concept of Python Programming, we also used control statement and exceptional handling by using this we developed an application of email system.

In this way CO2 , CO3 and CO6 are achieved