Gmail Client Python

Email Automation – by Aastha Goyal

Goals

Create a Python application that lets users interact with their Gmail accounts through Command Line Interface for performing the following tasks –

- Creating drafts
- Checking mails
- Sending mails
- Adding labels
- Downloading mails/attachments
- Adding emails to labels

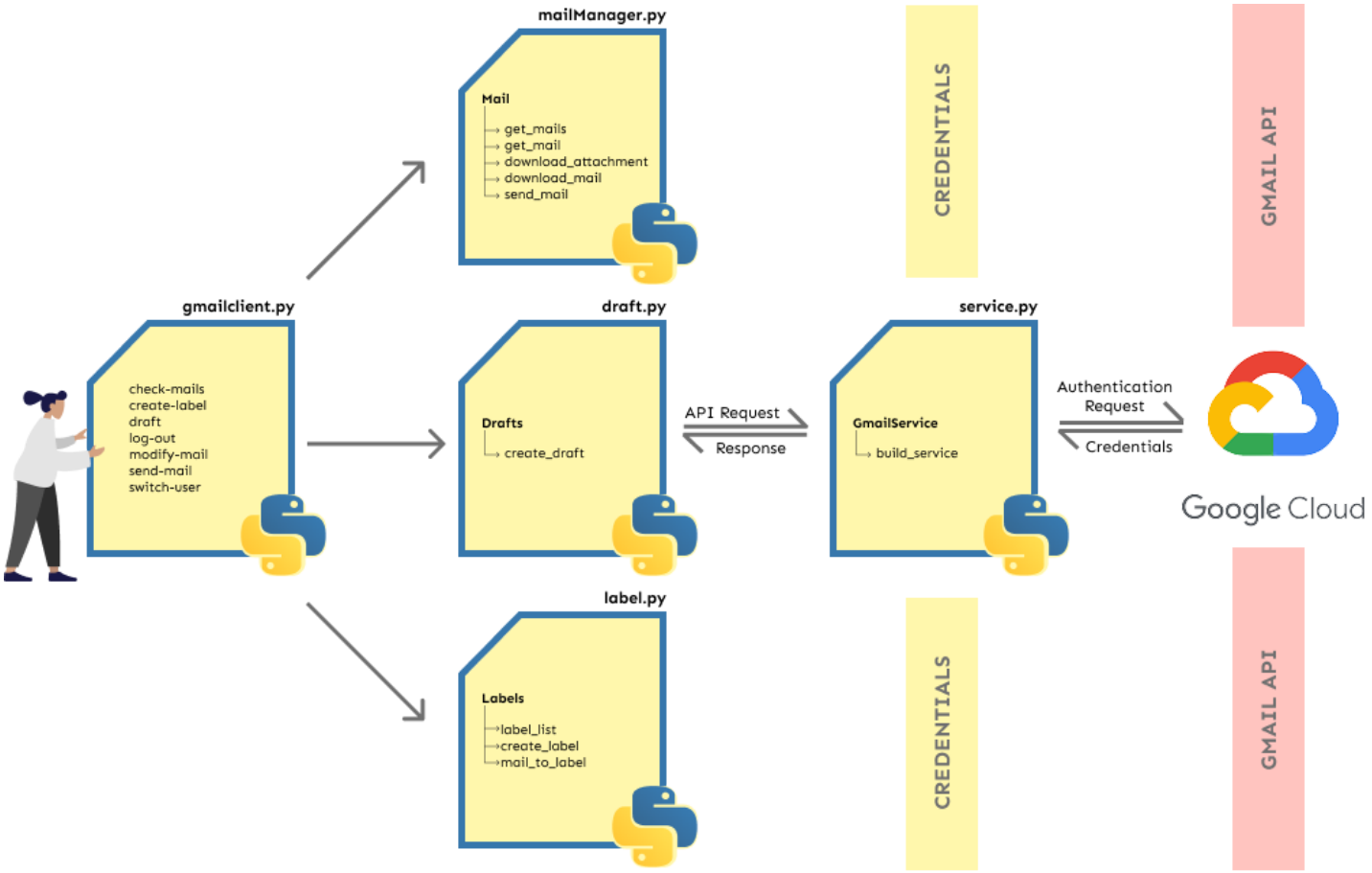
Pre-requisites

1. Get started as a Workspace developer
    a. Create a project on the Google Cloud Platform.
    b. Enable API – `API & Services > Library > Gmail API`
    c. Register App – `API & Services > Consent for OAuth`
    d. Get Credentials – `API & Services > Credentials > Create Credentials (OAuth2) > Application Type(Desktop App) > Download as credentials.json`
2. Python Libraries
    a. `threading` → Module that provides the mechanism to implement concurrency and makes it easier to synchronize multiple threads. In this project, it is used to implement Progress Bar functionality to perform network calls and UI rendering on separate threads.
    b. `typer` → This library makes it simpler to implement Command Line Interface.
    c. `rich` → This library helps in enhancing the UI by styling the terminal. Inside this project, it is used to create tables, show emoticons, and enrich the text with colors and different font styles.
3. A knack for programming.

Disclaimer

1. **The app on the Google Cloud Platform is not published yet**. So for any new user, first the creator will need to add the user's email as a testing email on the Cloud Platform.
2. **How to install it?** Unzip the shared file, then run the following command "`pip install -r requirements.txt`"

Modulation



Implementation

1. **Getting started –** `python3 gmailclient.py --help`
   Enlists all the functions available to the users along with a short description of the functions.

```
$ python3 gmailclient.py --help
Usage: gmailclient.py [OPTIONS] COMMAND [ARGS]...

Options:
  --install-completion [bash|zsh|fish|powershell|pwsh]
                                  Install completion for the specified shell.
  --show-completion [bash|zsh|fish|powershell|pwsh]
                                  Show completion for the specified shell, to
                                  copy it or customize the installation.

  --help                          Show this message and exit.

Commands:
  check-mails   To see list of emails
  create-label  To create labels
  download      To download mail or attachments
  draft         To create a draft
  log-out       To log out of the app
  modify-mail   To add mail to a label
  send-mail     To send mails
  switch-user   To switch user
```

2. **Create Draft messages –** Creates a draft message from the details provided.
   `python3 gmailclient.py draft --help`
   Enlists all the parameters available for creating the draft messages.

```
$ python3 gmailclient.py draft --help
Usage: gmailclient.py draft [OPTIONS] BODY

Arguments:
  BODY  [required]

Options:
  --to TEXT
  --by TEXT
  --subject TEXT
  --attachment TEXT  [default: ]
  --help             Show this message and exit.
```

Arguments

|--------> **--body**	Body text	[Required]
|--------> **--to**	Receiver's mail id	[Optional]
|--------> **--by**	Sender's Name	[Optional]
|--------> **--subject**	Subject of the mail	[Optional]
|--------> **--attachment**	List of attachments	[Optional]

Complete Command –

`python3 gmailclient.py draft "Draft Body" --to "receiver@email.com" --by "Sender's Name" --subject "Draft subject" --attachment "path/to/file1.png" --attachment "path/to/file2.pdf"`

```
$ python3 gmailclient.py draft "this is an automated draft" --to "aasthagoyalk23@gmail.com" --by "aasthagoyal" --subject "This is an automated amil"
--attachment "my_file.png"

✅ Draft Created
```

DraftId	To	By	Subject	Body	Attachment
r-8399469332431359156	aasthagoyalk23@gmail.com	aasthagoyal	This is an automated amil	this is an automated draft	✅

3. **Sending Emails –** Send mails as per the user's specifications.
   `python3 gmailclient.py send-mail --help`
   Enlists all the details about the send-mail function.

```
$ python3 gmailclient.py send-mail --help
Usage: gmailclient.py send-mail [OPTIONS] TO...

Arguments:
  TO...  [required]

Options:
  --by TEXT
  --subject TEXT
  --body TEXT
  --attachment TEXT  [default: ]
  --help             Show this message and exit.
```

Arguments

|--------> **--to**	Receiver's mail id	[Required]
|	(can be more than one mail ids for sending to multiple persons)	
|--------> **--by**	Sender's Name	[Optional]
|--------> **--subject**	Subject of the mail	[Optional]
|--------> **--body**	Body text	[Optional]
|--------> **--attachment**	List of attachments	[Optional]

Complete Command –

`python3 gmailclient.py send-mail “receiver@email.com” --by “Sender’s Name” --subject “Mail subject” –body “Mail Body” --attachment “path/to/file1.png” --attachment “path/to/file2.pdf”`

```
$ python3 gmailclient.py send-mail "aasthagoyalk23@gmail.com" "ashagoyal7898@gmail.com" --by "aasthagoyal" --subject "This is an automated mail" --bo
dy "This is an automated mail" --attachment "my_file.png"

progress |                                        | 100.0% complete


 ✅ Mail Sent
```

MailId	To	By	Subject	Body	Attachment
1810b4370f25a3f7	aasthagoyalk23@gmail.com	aasthagoyal	This is an automated mail	This is an automated mail	✅
1810b4375cd68d78	ashagoyal7898@gmail.com	aasthagoyal	This is an automated mail	This is an automated mail	✅

4. **Checking Emails –** Checking daily emails (form inbox[primary] for now) according to the user’s view, i.e., any number of emails from any page.
   `python3 gmailclient.py check-mails --help`

```
$ python3 gmailclient.py check-mails --help
Usage: gmailclient.py check-mails [OPTIONS]

Options:
  --limit INTEGER  [default: 10]
  --page INTEGER   [default: 1]
  --help           Show this message and exit.
```

Arguments

`|--------> --limit`	the numer of mails user wants to see at once	[Optional]
`|--------> --page`	from which page user wants to the mails	[Optional]

(page1 contains the ‘limit’ number of mails, according to that user can specify his requirements)

Complete Command –

`python3 gmailclient.py check-mails –limit 5 –page 2`

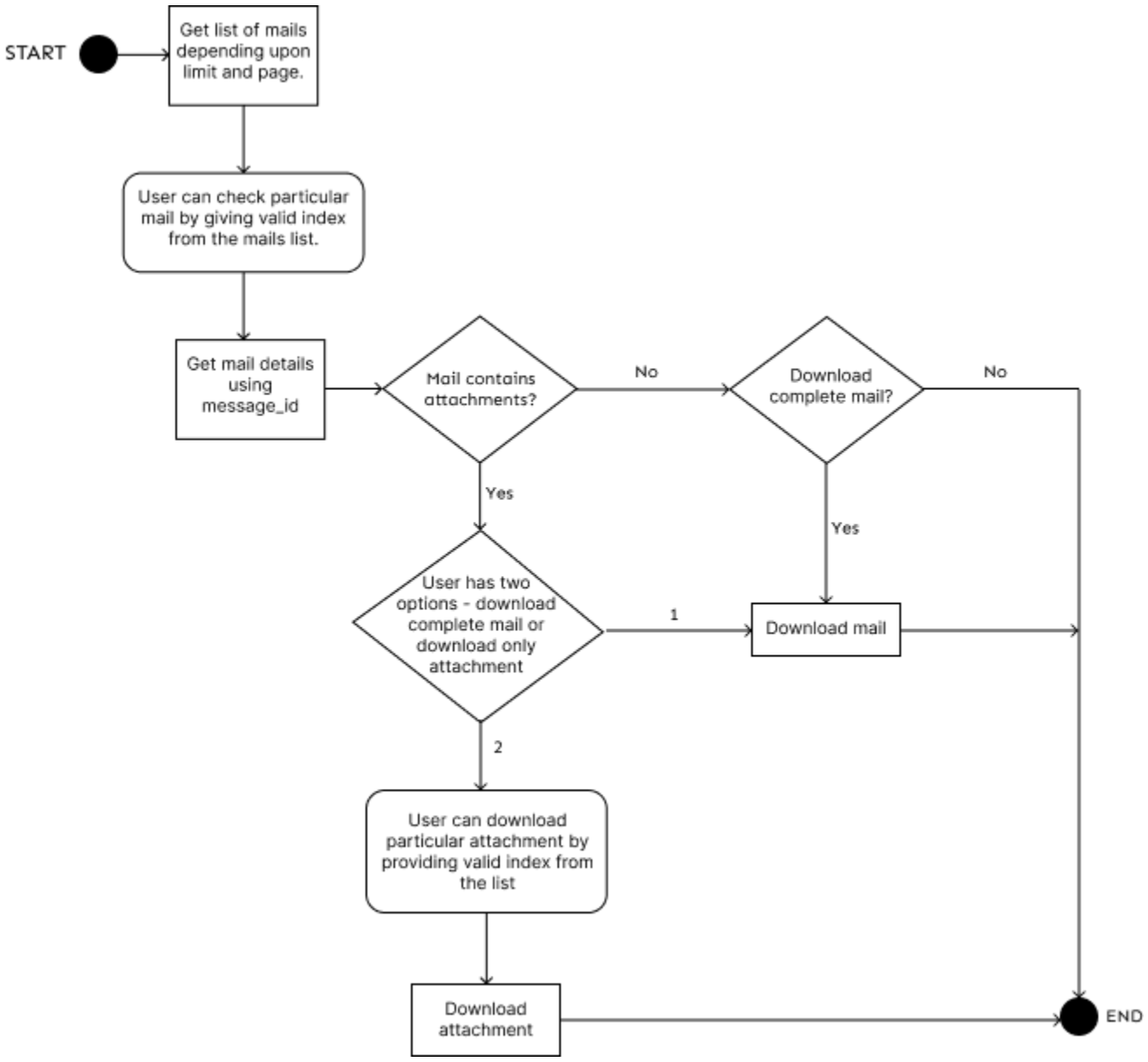
```
$ python3 gmailclient.py check-mails --limit 5 --page 1

progress |                                        | 100.0% complete
```

#	Id	From	Date	Subject
1	18116a80515fe6fa	CDSL e-Voting <donotreply.evoting@cdslindia.co.in>	Tue, 31 May 2022 01:58:35 +0530 (IST)	e-Voting FOR TRIVENI ENG&IN RE 1 will be OPEN FROM 03-06-2022 10:00:00 TO 05-06-2022 17:00:00 AND Meeting ON 06-06-2022 11:30:00
2	18116578f31c2f70	Unacademy <team@unacademy.com>	Tue, 31 May 2022 00:30:46 +0530	Aastha, here's your chance to win up to 90% scholarship 🥳
3	181162a1d4015f4c	LinkedIn <jobs-noreply@linkedin.com>	Mon, 30 May 2022 18:11:06 +0000 (UTC)	Want to get your application to Adobe noticed?
4	18115eb1671a39bf	Aastha Goyal <aasthagoyalk23@gmail.com>	Mon, 30 May 2022 22:32:05 +0530	this is an automated mail
5	18115dade8c82461	NASA <hq-newsletter@nasa.gov>	Mon, 30 May 2022 12:42:13 -0400 (EDT)	Reminder: With a Parachute Landing in New Mexico, Starliner Completes Flight Test

5. **Downloading -** For downloading complete mail or attachment
   `python3 gmailclient.py download --help`

```
$ python3 gmailclient.py download --help
Usage: gmailclient.py download [OPTIONS]

Options:
  --limit INTEGER  [default: 10]
  --page INTEGER   [default: 1]
  --help           Show this message and exit.
```

Workflow



Arguments

`|--------> --limit`	the numer of mails user wants to see at once	[Optional]
`|--------> --page`	from which page user wants to the mails	[Optional]

(`page1` contains the `'limit'` number of mails, according to that user can specify his requirements)

```
$ python3 gmailclient.py download --limit 4 --page 2

progress |                                        | 100.0% complete
```

#	Id	From	Date	Subject
1	1810a04c2001ba33	"Amazon.in" <shipment-tracking@amazon.in>	Sat, 28 May 2022 09:34:52 +0000	Your Amazon.in order #403-6723510-9393954 of 1 item has been dispatched
2	1810a0499916e908	Amazon Pay balance <AmazonPay-balance@amazon.in>	Sat, 28 May 2022 09:34:41 +0000	Your cashback of ₹91.18 is here!
3	18109c6b7160998b	"Amazon.in" <shipment-tracking@amazon.in>	Sat, 28 May 2022 08:27:06 +0000	Your Amazon.in order #403-9676368-1837166 of 1 item has been dispatched
4	18109c3832771b96	Amazon Pay balance <AmazonPay-balance@amazon.in>	Sat, 28 May 2022 08:23:36 +0000	Your cashback of ₹8.82 is here!

```
Give Serial number to check the mail :
```

```
Give Serial number to check the mail :  2
```

Mail Content
FROM: Amazon Pay balance <AmazonPay-balance@amazon.in>
DATE: Sat, 28 May 2022 09:34:41 +0000
SUBJECT: Your cashback of ₹91.18 is here!
BODY: Amazon Pay Gift Card ID - 6014860614883133 Expiry Date - 28 May 2023 Hi Aastha Goyal, Yay! Here's ₹91.18 cashback! You've got this cashback for Trance Home Linen 100% Cotton 200TC Elasticated

```
To download the mail [Y/N]?
```

```
To download the mail [Y/N]?  y
Give File a name:  my_mail

✅ Mail Downloaded
```

Attachment present

```
$ python3 gmailclient.py download --limit 2

progress |                                        | 100.0% complete
```

#	Id	From	Date	Subject
1	18115eb1671a39bf	Aastha Goyal <aasthagoyalk23@gmail.com>	Mon, 30 May 2022 22:32:05 +0530	this is an automated mail
2	18115dade8c82461	NASA <hq-newsletter@nasa.gov>	Mon, 30 May 2022 12:42:13 -0400 (EDT)	Reminder: With a Parachute Landing in New Mexico, Starliner Completes Flight Test

```
Give Serial number to check the mail :  1

┌──────────────────────────────────────────────┐
│                 Mail Content                 │
├──────────────────────────────────────────────┤
│ FROM: Aastha Goyal <aasthagoyalk23@gmail.com>│
├──────────────────────────────────────────────┤
│      DATE: Mon, 30 May 2022 22:32:05 +0530   │
├──────────────────────────────────────────────┤
│       SUBJECT: this is an automated mail     │
├──────────────────────────────────────────────┤
│                    BODY:                     │
├──────────────────────────────────────────────┤
│                  1. todo.png                 │
└──────────────────────────────────────────────┘


Choose from following options:
1 -> To download the mail
2 -> To download the attachment
Choice:  2
 Give Attachment numbers to download:   1

 ✅ Attachment Downloaded
```

6. **Labels –** Users can create labels according to requirements
   `python3 gmailclient.py create-label --help`

```
$ python3 gmailclient.py create-label --help
Usage: gmailclient.py create-label [OPTIONS] LABEL_NAME

Arguments:
  LABEL_NAME  [required]

Options:
  --label-visibility TEXT
  --messagelist-visibility TEXT
  --help                          Show this message and exit.
```

Arguments

|--------> **--Label-name**	Name of the label	[Required]
|--------> **--lable-visibility**	the label should be visible to all users or not	[Optional]
|--------> **--messagelist-visibility**	from which page user wants to the mails	[Optional]

Complete Command –
`python3 gmailclient.py create-label “Label name”`

```
$ python3 gmailclient.py create-label "Custom Label"

 ✅ Label Created
```

7. **Modify mails –** Users can change the labels of mails
   `python3 gmailclient.py modify-mail --help`

```
$ python3 gmailclient.py modify-mail --help
Usage: gmailclient.py modify-mail [OPTIONS]

Options:
  --limit INTEGER  [default: 10]
  --page INTEGER   [default: 1]
  --help           Show this message and exit.
```

Arguments

|--------> **--limit**	the numer of mails user wants to see at once	[Optional]
|--------> **--page**	from which page user wants to the mails	[Optional]

(page1 contains the **'limit'** number of mails, according to that user can specify his requirements)

```
$ python3 gmailclient.py modify-mail --limit 5

progress |                                        | 100.0% complete
```

#	Id	From	Date	Subject
1	18116a80515fe6fa	CDSL e-Voting <donotreply.evoting@cdslindia.co.in>	Tue, 31 May 2022 01:58:35 +0530 (IST)	e-Voting FOR TRIVENI ENG&IN RE 1 will be OPEN FROM 03-06-2022 10:00:00 TO 05-06-2022 17:00:00 AND Meeting ON 06-06-2022 11:30:00
2	18116578f31c2f70	Unacademy <team@unacademy.com>	Tue, 31 May 2022 00:30:46 +0530	Aastha, here's your chance to win up to 90% scholarship 🥳
3	181162a1d4015f4c	LinkedIn <jobs-noreply@linkedin.com>	Mon, 30 May 2022 18:11:06 +0000 (UTC)	Want to get your application to Adobe noticed?
4	18115eb1671a39bf	Aastha Goyal <aasthagoyalk23@gmail.com>	Mon, 30 May 2022 22:32:05 +0530	this is an automated mail
5	18115dade8c82461	NASA <hq-newsletter@nasa.gov>	Mon, 30 May 2022 12:42:13 -0400 (EDT)	Reminder: With a Parachute Landing in New Mexico, Starliner Completes Flight Test

```
Give Serial number to check the mail :
```

```
Give Serial number to check the mail :  4
```

Mail Content
FROM: Aastha Goyal <aasthagoyalk23@gmail.com>
DATE: Mon, 30 May 2022 22:32:05 +0530
SUBJECT: this is an automated mail
BODY:
1. todo.png

```
Give label names to add:  spam
Give label names to remove or can be left blank:

✅ Mail modified
```

Functions Description

To better understand the actual internal working of the application, I have added pseudocode for all the important functions. Along with the pseudocode, a link to the official documentation is also provided for reference.

1. Internal working of the `draft()` for creating and saving drafts. This function calls `users.drafts.create()` method of Gmail API. More information regarding the request and response body can be found in the official documentation.

```
# draft.py - helper function that calls Gmail API
def create_draft():
      # converts all arguments to MIMEMultipart message
      if contains attachments:
            # add attachments to message
      encode_message()
      # create a draft message of the following format
      {
        'message': {
           'raw': encoded_message
        }
      }
      # call users.drafts.create() method of Gmail API
      return created_draft, error

#gmailclient.py - function to interact with user
def draft():
      create_draft()
      if error is None:
            create_table()
      else:
            error_handler()
```

2. Internal working of `send_mail()` which is used to send the mail. This function call makes use of `users.messages.send()` method provided by Gmail API. More information regarding the directions for use can be found in the official documentation. Here concurrency is also implemented using the threading library.

```
# mailManager.py - helper function to send mails
def send_mail():
        # converts all arguments to MIMEMultipart message
        if contains attachments:
                # add attachments to message
        encode_message()
        # create a draft message of the following format
        {
           'raw': encoded_message
        }
        # use Gmail API
        sent_mail = users.messages.send()
        return sent_mail, error

#gmailclient.py - function to interact with user
def send_mail():
         # call methods on separate threads
         send_mail()                # thread 1
         start_progress_bar()       # main thread

         data, error = thread1.joins()
         if error is None:
                # mail sent successfully
                create_table()
        else:
                error_handler()
```

3. Internal working of `download()` function, that gives users the ability to either download the complete mail or any attachment if present. For understanding the functionality, the following links can be helpful -
    a. Link1 - Get the mail
    b. Link2 - Download the attachment

```
# mailManager.py - helper function to read mail
def get_mail(msg_id):
        # call Gmail API
        mail = users.messages.get()
        return mail, error
# mailManager.py - helper function to download mail
def download_mail():
        # downloads mail as .eml file
        mail = users.messages.get()
        encode_mail()
        # write mail data to a file
        file = open(filename.eml)
        file.write(encoded_data)
        return download_status, error
# mailManager.py - helper function to download attachment
def download_attachment():
        data = users.attachments.get()
        encode_data()
        # write data to a binary file
        file = open(filename.eml)
        file.write(encoded_data)
        return error

# gmailclient.py - Function to interact with user
def download():
        mails = check_mails()    # show list of message list

        # ask user to choose one mail from given list
        choice = show_menu()
        mail = read_mail(mails[choice])

        if mail has attachment:
                # ask user to download mail or attachment
                download_mail()
                download_attachment()
        else:
                # ask user to download mail
                download_mail()

       if error is None:
                # downloaded successfully
       else:
                error_handler()
```

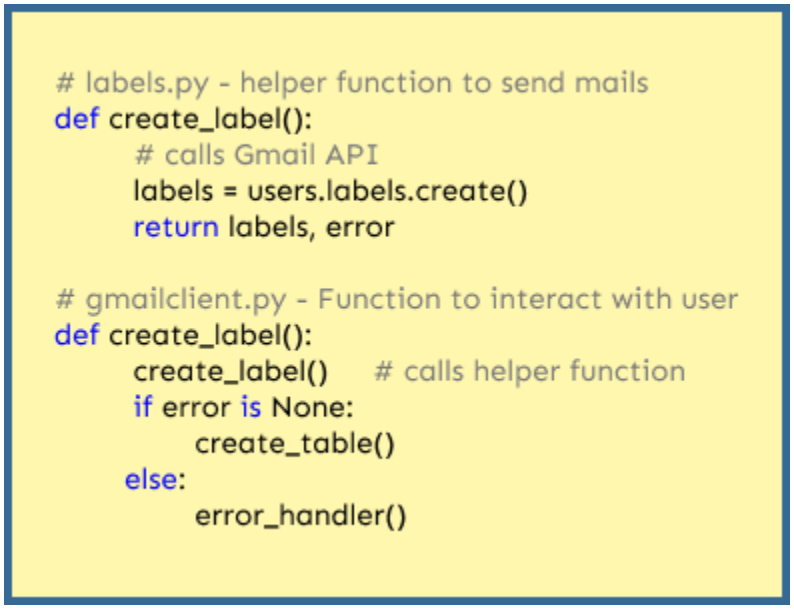
4. Internal working of `check_mails()` which allows users to see their email from the *primary* category. This function calls `users.messages.list()` method provided from the Gmail API. It makes the network call on a separate thread to synchronously show the progress bar as well as fetch the data. More information regarding the API used can be found in the official documentation.

```
# mailManager.py - helper function to get list of all mails
def get_mails():
        # find messages on given page
        for i in range(page):
                messages = users.messages.list(next_page_token)
                next_page_token = messages['nextPageToken']
        # messages only contains message_id
        # get message details using id
        data = []
        for message in messages:
                msg = users.messages.get(message['id'])
                data.append(msg)
        return data, error

# gmailclient.py - Function to interact with user
def check_mails():
        # call methods on separate threads
        get_mails()                 # thread 1
        start_progress_bar()    # main thread

        data, error = thread1.joins()
        if error is None:
                create_table()
        else:
                error_handler()
```

5. Internal working of `modify_mail()` that allows user to change the label of a mail. It makes use of `users.messages.modify()` provided by the Gmail API. For information regarding the function, also see the official documentation.

```
# labels.py - helper function to send mails
def mail_to_label():
        # calls Gmail API
        labels = label_list()
        if add_label, remove_label in labels:
                # get label ids
                label_ids.append(id)
        # call Gmail API
        modify = users.messages.modify()
        return modify, error

# gmailclient.py - Function to interact with user
def modify_mail():
        # show user list of mails
        mail_list = check_mails()

        # ask user to choose one mail
        choice = show_menu()
        mail = read_mail(mail_list[choice])

        add_label = input()
        remove_label = input()
        mail_to_label(add_label, remove_label)

        if error is None:
                # mail modified successfully
        else:
                error_handler()
```

6. Internal working of `create_label()` that allows users to create new labels for their mailbox. It makes use of `users.labels.create()` provided by the Gmail API. For more information regarding the usage of this particular API, you can visit the official website.



```
# labels.py - helper function to send mails
def create_label():
        # calls Gmail API
        labels = users.labels.create()
        return labels, error

# gmailclient.py - Function to interact with user
def create_label():
        create_label()     # calls helper function
        if error is None:
                create_table()
        else:
                error_handler()
```

Future Scopes

- UI enhancement → **Showing file browser for attachment and downloads**.
  Having used the app for quite some time, I realized that giving a file path is not very intuitive rather using a file browser to select locations would be a better option.
- Feature
  - **Allowing mail creation using contents of a text file (.txt, .docx)**.
    There was a use case where I wanted to add tables in the mail body which is not a trivial task to perform using a terminal. In such a case it would be better to directly get the contents from a file.
  - **Allowing users to reply to a particular mail** as of now the user can only send new mails.