

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

----- ∞  ∞ -----



BÁO CÁO ĐỒ ÁN MÔN HỌC
MẠNG MÁY TÍNH

THỰC HÀNH:
SOCKET

Giảng viên hướng dẫn : Chung Thùy Linh
Nhóm sinh viên thực hiện : Nhóm 1.2 – Lớp 22CTT3

Thành phố Hồ Chí Minh - 12/2023

MỤC LỤC

Contents

1. Thông tin thành viên:.....	2
2. Yêu cầu đồ án:	2
3. Phân công công việc:	3
4. Đánh giá:	4
5. Môi trường lập trình và framework hỗ trợ:.....	5
6. Danh sách hàm sử dụng:	10
7. Kịch bản giao tiếp của chương trình:.....	15
8. Hướng dẫn sử dụng:.....	16
i. Khởi động Thunder Owl	16
ii. Thao tác gửi email.	17
iii. Thao tác tải và xem email.	20
VIDEO CHI TIẾT CÁCH HƯỚNG DẪN SỬ DỤNG THUNDER OWL ..	24
TÀI LIỆU THAM KHẢO:	25

1. Thông tin thành viên:

STT	Họ và tên	MSSV	Email	SĐT
1	Nguyễn Minh Hưng	22120123	hungm0434@gmail.com	0769552756
2	Trịnh Hoàng Khang	22120154	trinhhoangkhangvl@gmail.com	0825228120
3	Hà Đức Huy	22120133	haduchuy122004@gmail.com	0944226408

2. Yêu cầu đề án:

- **Gửi được email (dùng giao thức SMTP):** To, CC, BCC (lưu ý khi gửi BCC thì những người nhận liên quan (To, CC) sẽ không biết là có gửi BCC, vì khi đọc email sẽ không hiện user nào được BCC), gửi file đính kèm email.
- **Tải được email từ Mail server (dùng giao thức POP3):** Tải đủ và đúng. Không tải lại những email đã tải trước đó. Sinh viên tự đưa ra giải pháp để đảm bảo yêu cầu này cả khi tắt/ mở lại Mail Client.
- **Cho phép người dùng phân loại email vào các folder cụ thể theo cấu hình của file config** (phân loại theo địa chỉ người gửi, theo subject, hoặc từ khóa trong nội dung email). Mặc định email được tải về client luôn thuộc folder Inbox nếu không phù hợp bất kỳ kết quả filter nào.
- **Cho phép lọc các từ khóa spam từ subject và nội dung email**, di chuyển các spam email vào thư mục Spam khi lấy email về từ Mailbox của Server
- **Quản lý được trạng thái email (đã đọc/ chưa đọc).** Sinh viên tự đưa ra giải pháp để đánh dấu, đảm bảo vẫn đúng khi tắt/mở chương trình Mail Client của mình
- **Nâng cao:** Tự động tải email trong mailbox về mỗi 10s một lần, cấu hình thời gian tự động tải email này trong file config

3. Phân công công việc:

STT	Họ và tên	MSSV	Nhiệm vụ
1	Nguyễn Minh Hưng	22120123	<ul style="list-style-type: none"> - Làm giao diện mail client (Code chính). - Thực hiện các thao tác: <ul style="list-style-type: none"> • Gửi được attached file. <ul style="list-style-type: none"> ○ Gửi được file .txt ○ Gửi được file có các định dạng: *.docx, *.pdf, *.jpg... ○ Gửi được nhiều file đính kèm trong cùng 1 email. • Xử lý lọc email. <ul style="list-style-type: none"> ○ Lọc email dựa vào địa chỉ người gửi ○ Lọc email dựa trên subject ○ Lọc email dựa vào nội dung ○ Di chuyển email vào thư mục cụ thể / spam • Xử lý và điều chỉnh các cài đặt với text mail (style, color, font, find, ...) • Thực hiện chèn ảnh (gif, ...) vào trong text mail.
2	Trịnh Hoàng Khang	22120154	<ul style="list-style-type: none"> - Làm giao diện mail client. (Code chính) - Thực hiện các thao tác: <ul style="list-style-type: none"> • Tổ chức file config: <ul style="list-style-type: none"> ○ Tổ chức file config dạng có cấu trúc (*.xml, *.sql, *.json...) • Quản lý trạng thái email (đã đọc / chưa đọc) • Tải được email về mail client từ mailbox của Server
3	Hà Đức Huy	22120133	<ul style="list-style-type: none"> - Làm giao diện mail client. (Đóng góp) - Thực hiện thao tác: <ul style="list-style-type: none"> • Gửi email với TO, CC, BCC • Xử lý tự động tải email từ Mailbox theo thời gian cấu hình trong config file. • Kiểm tra chương trình và viết báo cáo. • Kiểm chứng giao thức đảm bảo dữ liệu truyền đi.

4. Đánh giá:

STT	Yêu cầu	Đánh giá	Ghi chú
1	Gửi được Email với TO và CC <ul style="list-style-type: none"> Gửi email đến một hoặc nhiều địa chỉ người nhận (TO) 0,5đ Gửi email đến một hoặc nhiều địa chỉ người nhận (CC) 0,5đ 	100%	Không
2	Gửi được Email với BCC	100%	Không
3	Gửi được attached file (dung lượng file <= 3MB) <ul style="list-style-type: none"> Gửi được file *.txt: 0,5đ Gửi được các định dạng file như: *.docx, *.pdf, *.jpg, *.zip,... 1đ Gửi được nhiều file đính kèm trong cùng 1 email 0,5đ Có giới hạn dung lượng file gửi: 0,5đ 	100%	Không giới hạn dung lượng file gửi.
4	Tải được các email về mail client từ mailbox của Server <ul style="list-style-type: none"> Tải nội dung email không kèm file 1đ Tải nội dung email có kèm file 1,5đ Tải nội dung email có kèm file, có lưu file xuống máy cục bộ của client 2đ 	100%	Lưu file về phải tự đặt tên mới cho file, không giữ tên file cũ. Nếu muốn lấy tên file cũ thì phải nhập đúng tên cũ.
5	Quản lý trạng thái email (đã đọc/ chưa đọc)	100%	Không
6	Xử lý lọc email: <ul style="list-style-type: none"> Lọc email dựa trên địa chỉ người gửi 0,5đ Lọc email dựa trên subject 0,25đ Lọc email dựa trên nội dung 0,25đ Đưa email vào thư mục cụ thể/ Spam 0,5đ 	100%	Lọc được dựa theo một danh sách các keyword tùy ý (mỗi keyword chr một từ).
7	Xử lý tự động tải email từ Mailbox theo thời gian cấu hình trong config file	100%	Không
8	Tổ chức file config: <ul style="list-style-type: none"> Không tổ chức file config: 0đ Có sử dụng file config dạng *.txt 0,5đ Có sử dụng file config dạng file có cấu trúc (*.xml, *.sql, *.json,...) 1đ 	100%	Không

5. Môi trường lập trình và framework hỗ trợ:

- **Môi trường lập trình:** Python 3.12.1
- **Framework hỗ trợ:** Không sử dụng
- **Các thư viện đã sử dụng:**

Lưu ý:

- Nếu thư viện không phải thư viện con của một thư viện khác (VD: `import tkinter as tk...`) thì sử dụng toàn bộ các hàm liên quan đến thư viện.
- Nếu thư viện là một thư viện con của thư viện khác (VD: `from tkinter import filedialog...`) thì sử dụng những hàm chức năng chỉ liên quan đến thư viện con.
- Các hàm và thư viện dưới đây có thể vẫn chưa liệt kê đầy đủ.

S T T	Thư viện sử dụng	Thư viện con sử dụng	Tải xuống thư viện trong cmd (Window)	Mục đích sử dụng.
1	tkinter	Không	Không	Tạo giao diện người dùng đồ họa (GUI)
		filedialog		Cung cấp các hộp thoại để mở tệp, lưu tệp và chọn thư mục. Thư viện cho phép nhóm tương tác chọn các tập tin hoặc thư mục từ máy tính của nhóm. Nhóm sử dụng 2 hộp thoại: <ul style="list-style-type: none"> • filedialog.askopenfilename(): Mở hộp thoại để người dùng chọn tệp tin cần mở và trả về đường dẫn của tệp tin đã chọn • filedialog.asksaveasfilename(): Mở hộp thoại để người dùng chọn nơi lưu tệp tin và trả về đường dẫn của tệp tin đã chọn
		simpledialog		Cung cấp một cách đơn giản để yêu cầu người dùng nhập thông tin nhỏ thông qua việc hiển thị hộp thoại. Nhóm sử dụng 1 hộp thoại: <ul style="list-style-type: none"> • simpledialog.askstring(title, prompt): Mở hộp thoại với tiêu đề và thông báo yêu cầu người dùng nhập một chuỗi.
2	PIL	Image	pip install pillow	Image trong thư viện PIL (Python Imaging Library) là một thư viện Python phổ biến được sử dụng để làm việc với hình ảnh. Image cung cấp một tập hợp các chức năng và lớp để mở, thao tác, tạo và lưu các định dạng hình ảnh khác nhau. Nhóm sử dụng 1 chức năng của thư viện: <ul style="list-style-type: none"> • image.open(file_path): Mở các định dạng hình ảnh khác nhau (JPEG, PNG, BMP, GIF, v.v.)

3	tkcalendar	Calendar	pip install tkcalendar	Cung cấp một widget dành cho lịch để có thể chọn ngày tháng trong ứng dụng GUI của nhóm.
4	tkinter.colordrchooser	askcolor	Không	Thực hiện nhập hàm “askcolor” từ module “colorchooser” trong thư viện tkinter. Hàm này giúp mở một hộp thoại cho người dùng chọn một màu từ bảng màu và trả về giá trị màu được chọn dưới dạng một bộ giá trị RGB hoặc dạng chuỗi hex.
5	socket	Không	Không	<p>Thư viện cung cấp một giao diện cấp thấp để truyền thông mạng. Nó cho phép tạo ra các socket, là các điểm cuối được sử dụng để gửi và nhận dữ liệu qua mạng. Socket cho phép giao tiếp giữa các thiết bị khác nhau trên mạng sử dụng các giao thức khác nhau như TCP (Transmission Control Protocol) và UDP (User Datagram Protocol).</p> <ul style="list-style-type: none"> • Creating Sockets: Module socket cho phép bạn tạo ra các loại socket khác nhau: <ul style="list-style-type: none"> ○ socket.socket(): Tạo một socket mới. ○ Nhóm có thể xác định loại socket. Ví dụ: <ul style="list-style-type: none"> ▪ socket.AF_INET cho IPv4 ▪ socket.AF_INET6 cho IPv6 ○ Nhóm cũng có thể xác định giao thức của socket <ul style="list-style-type: none"> ▪ socket.SOCK_STREAM cho TCP ▪ socket.SOCK_DGRAM cho UDP. • Socket Methods: Sau khi tạo socket, nhóm có thể sử dụng các phương thức khác nhau để truyền thông: <ul style="list-style-type: none"> ○ socket.bind(): Liên kết socket với một giao diện mạng cụ thể và số cổng. ○ socket.listen(): Đưa socket vào chế độ máy chủ để chấp nhận các kết nối đến (TCP). ○ socket.connect(): Khởi tạo một kết nối đến một socket khác (chế độ client). ○ socket.accept(): Chấp nhận một kết nối đến và trả về một đối tượng socket mới và địa chỉ của client (TCP). ○ socket.send(), socket.recv(): Gửi và nhận dữ liệu thông qua socket (cho TCP). ○ socket.sendto(), socket.recvfrom(): Gửi và nhận dữ liệu thông qua socket (cho UDP).
6	json	Không	Không	<p>Thư viện cung cấp các công cụ để làm việc với dữ liệu JSON (JavaScript Object Notation). JSON cũng là một định dạng dữ liệu phổ biến để truyền thông tin giữa các hệ thống.</p> <p>Nhóm sử dụng các thao tác sau của thư viện “json”:</p> <ul style="list-style-type: none"> • json.dump(): Chuyển đổi một đối tượng Python thành một chuỗi JSON. • json.load(): Chuyển đổi một chuỗi JSON thành đối tượng Python.

7	base64	Không	pip install pybase64	<p>Đây là một thư viện tiêu chuẩn cung cấp chức năng mã hóa và giải mã dữ liệu theo chuẩn Base64 và là một phương pháp biểu diễn dữ liệu nhị phân dưới dạng chuỗi ASCII.</p> <p>Nhóm sử dụng các thao tác sau của thư viện “base64”</p> <ul style="list-style-type: none"> • base64.b64encode(data): Nhận dữ liệu nhị phân đầu vào và trả về chuỗi theo chuẩn Base64 dưới dạng đối tượng bytes • base64.b64decode(encoded_data): Giải mã một chuỗi byte được mã hóa theo chuẩn Base64 trở lại dạng nhị phân gốc của nó. • Ngoài ra nhóm cũng sử dụng thêm thao tác sau cho các thao tác ở trên: <ul style="list-style-type: none"> ◦ *.decode(): Giải mã byte thành chuỗi theo chuẩn bất kì. Nếu không có bất kì ràng buộc theo chuẩn nào, mặc định hàm sẽ giải mã theo chuẩn “utf- 8”
8	pathlib	Path	pip install pathlib	<p>Thư viện “pathlib” cung cấp một cách tiếp cận hướng đối tượng để xử lý đường dẫn hệ thống tệp tin và các hoạt động liên quan đến chúng. “Path” bên trong thư viện “pathlib” sử dụng để biểu diễn đường dẫn hệ thống tệp tin theo cách hướng đối tượng.</p>
9	customtkinter	Không	pip install customtkinter	<p>Là một thư viện mô phỏng hoặc mở rộng các chức năng của thư viện chuẩn “tkinter” trong Python. Thư viện này bao gồm các cải tiến, các thành phần giao diện người dùng bổ sung, cải thiện bố cục hoặc các hàm tiện ích để đơn giản hóa các công việc thường gặp liên quan đến phát triển giao diện đồ họa (GUI).</p> <p>Nhóm sử dụng các thao tác sau của thư viện “customtkinter”:</p> <ul style="list-style-type: none"> • customtkinter.CTkFrame(): Tạo ra một khung (frame) tùy chỉnh trong giao diện người dùng (GUI) • customtkinter.CTkScrollableFrame(): Tạo một khung có khả năng cuộn (scrollable frame) trong giao diện người dùng (GUI) • customtkinter.CTkLabel(): Tạo một nhãn (label) trong giao diện người dùng (GUI) • customtkinter.CTkToplevel(): Tạo một cửa sổ độc lập (toplevel window) trong giao diện người dùng (GUI) • customtkinter.CTkButton(): Tạo một nút (button) trong giao diện người dùng (GUI) • customtkinter.CTkImage(): Hàm hoặc lớp tượng trưng nhằm để làm việc với hình ảnh (images) trong giao diện người dùng (GUI). • customtkinter.CTkEntry(): Tạo một ô nhập liệu (entry) trong giao diện người dùng sử dụng (GUI) • customtkinter.CTk(): Để tạo một cửa sổ giao diện người dùng (GUI). • customtkinter.set_appearance_mode (“dark”): Chuyển chế độ giao diện sang chế độ tối (dark mode)

				<ul style="list-style-type: none"> • customtkinter.set_default_color_theme (“blue”): Thiết lập chủ đề màu sắc mặc định của ứng dụng thành màu xanh (blue theme).
10	re	Không	Không	<p>Là một thư viện chuẩn cung cấp các công cụ cho việc làm việc với các biểu thức chính quy (regular expressions). Công cụ này cho phép thực hiện các phép tìm kiếm, thay thế và xử lý các chuỗi văn bản dựa trên các mẫu được xác định bằng biểu thức chính quy.</p> <p>Nhóm sử dụng các thao tác sau của thư viện “re”:</p> <ul style="list-style-type: none"> • re.compile(): Sử dụng để biên dịch biểu thức chính quy thành một đối tượng biểu thức chính quy (regular expression object). Điều này cho phép tái sử dụng biểu thức chính quy đã biên dịch để thực hiện nhiều phép tìm kiếm hoặc thay thế mà không cần phải biên dịch lại biểu thức từ đầu. • re.escape(): Để thoát các ký tự được đề cập trong một chuỗi sao cho chúng không được xử lý như một phần của biểu thức chính quy.
11	io	Không	Không	<p>Là một thư viện được sử dụng để làm việc với input/output dữ liệu. Nó cung cấp các đối tượng và hàm để thực hiện các hoạt động đọc, ghi và xử lý dữ liệu từ các nguồn khác nhau như file, chuỗi, stream và các đối tượng khác.</p> <p>Nhóm sử dụng thao tác sau của thư viện “io”:</p> <ul style="list-style-type: none"> • io.BytesIO(): Cho phép đọc và ghi dữ liệu từ một đối tượng byte string như nếu nó là file.
12	os	Không	Không	<p>Là thư viện cung cấp các hàm để tương tác với hệ thống máy tính (Operating System - OS). Điều này bao gồm các chức năng để thực hiện các thao tác như tạo, xóa, di chuyển và thay đổi tên các tệp tin và thư mục, cũng như truy cập các biến môi trường và thực hiện các thao tác tương tác với hệ điều hành.</p> <p>Nhóm sử dụng các thao tác sau của thư viện “os”:</p> <ul style="list-style-type: none"> • os.path.basename(file_path): Trả về phần tên của tệp tin hoặc thư mục từ đường dẫn được cung cấp trong đối số file_path
13	datetime	datetime	Không	<p>Nhóm sử dụng lớp “datetime” từ hàm “datetime” với chức năng cho phép thực hiện các hoạt động như lấy thời gian hiện tại, tạo các đối tượng ngày tháng cụ thể và phân tích cú pháp từ chuỗi thành đối tượng thời gian.</p>

14	time	Không	Không	Cung cấp các hàm liên quan đến thời gian. Ta có thể sử dụng để thực hiện các thao tác liên quan đến thời gian như đo thời gian thực hiện một chương trình, tạo độ trễ (delay), hoặc xử lý thời gian trong các ứng dụng của bạn.
15	threading	Không	Không	Cung cấp các công cụ để tạo và quản lý các luồng (threads) đồng thời trong một chương trình. Luồng là một dạng của quá trình (process) nhưng nhẹ hơn và chúng chia sẻ cùng không gian bộ nhớ, cho phép các tác vụ được thực hiện song song.

- Các thư viện ở trên được biểu diễn như sau:

```
import tkinter as tk
from tkinter import filedialog, simpledialog
from PIL import Image, ImageTk
import io
from tkcalendar import Calendar
import threading
import time
import os
from tkinter.colorchooser import askcolor
import socket
import json
import base64
from pathlib import Path
import customtkinter
from customtkinter import *
import re
from datetime import datetime
```

6. Danh sách hàm sử dụng:

S T T	Tên Hàm	Chức năng	Hoạt động
1	download_email_every_1_minute_thread_function_()	Tự động email 10 giây mỗi lần	Mỗi 10 giây thì chương trình sẽ tự động tải xuống các email.
2	get_date()	Trả về thời gian hiện tại, biểu diễn dạng sau: %a, %d %b %Y %H:%M:%S Ví dụ: Thu, 14 Dec 2023 21:03:41.	Sử dụng hàm có sẵn datetime.now() để lấy thời gian thực tế.
3	fixTextForButton()	Điều chỉnh text để nút hiển thị text đúng vị trí mong muốn	Dựa vào chiều dài của chuỗi text và nút để điều chỉnh vị trí cho phù hợp
4	resolveFile()	Lưu file về máy local	Sử dụng hàm có sẵn filedialog.asksaveasfilename() để chọn thư mục lưu file.
5	resolveMail()	Giải mã để hiển thị bố cục, thành phần mail đúng mail đã gửi	Với từng mục cần giải mã vào database json để lấy các dữ liệu liên quan bao gồm thuộc tính, vị trí, dữ liệu ảnh, ... để giải mã và hiển thị
6	resolve_image()	Tạo ảnh từ data image và các thuộc tính như vị trí, chiều cao, chiều dài để hiển thị và đưa về đúng vị trí trong mail.	Sử dụng hàm image.open(io.BytesIO(image_data)) để mở ảnh gốc từ image_data, sau đó chỉnh size ảnh bằng hàm resize() và cho vào hộp text bằng hàm image_create()
7	change_style_resolve()	Dùng để hiển thị style của text sao cho đúng style của text từ mail gửi	Dùng để hiển thị style của text sao cho đúng style của text từ mail gửi, sử dụng database json để lấy tên style và vị trí chính xác
8	change_font_resolve	Dùng để hiển thị font chữ của text sao cho đúng font của text từ mail gửi	Dùng để hiển thị font của text sao cho đúng font của text từ mail gửi, sử dụng database json để lấy font và vị trí chính xác
9	change_color_resolve()	Dùng để hiển thị màu của text sao cho đúng màu của text từ mail gửi	Dùng để hiển thị màu của text sao cho đúng style của text từ mail gửi, sử dụng database json để lấy mã màu

			và vị trí chính xác
10	getFolderMessage()	Tạo hai cửa sổ để lưu các thư đúng folder và cửa sổ để hiển thị mail.	Dùng CTKToplevel(window) tạo cửa sổ window phụ từ cửa sổ window.
11	disable()	Bỏ chức năng tự mở rộng của frame	Tắt grid_propagate
12	send_data_to_server()	Gửi toàn bộ data file json tạm lên server	Dùng SMTP, gửi toàn bộ data file json tạm lên server
13	is_valid_string()	Kiểm tra, xâu hợp lệ nếu tồn tại keyword trong xâu	Kiểm tra, xâu hợp lệ nếu tồn tại keyword trong xâu
14	recvall()	Nhận toàn bộ data từ socket	Nhận data từ socket một lượng 1024 byte mỗi lần, nhận đến khi hết data
15	get_data_from_server()	Lấy mail của user về	Dùng POP3, lấy email của user về từ server
16	process_data_so_we_can_convert_to_json_file()	Xử lý data của hàm Get_data_from_server trước khi chuyển thành json	Cắt bỏ dòng đầu và dòng cuối của data vì do SMTP dòng đầu dính đoạn +OK và dòng cuối dính dấu chấm
17	write_the_data_received_to_the_temporary_json_file_to_help_us_read_easier()	Viết data sau khi process lên file json tạm	Cần viết lên mail json tạm để có thể kiểm soát được từ khóa và lọc thư dễ dàng hơn, file json tạm sẽ bị xóa sạch sau bước này
18	clean_the_temporary_json_file_after_we_done_with_it()	Xóa sạch file json tạm	Xóa sạch file json tạm
19	choose_which_mail_box_on_user_config()	Chọn mail box dựa trên từ khóa trong file config	Chọn mail box dựa trên từ khóa trong file config, dùng kết hợp hàm is_valid_string()
20	after_decided_the_mail_box_now_we_take_the_buffer_json_and_add_it_to_the_database()	Thêm data của mail vừa nhận vào database	Thêm data của mail vừa nhận vào database
21	check_to_see_how_many_mail_are_there_in_the_acount_of_the_user()	Check xem có bao nhiêu mail trong mail_box trên server của user	Tạo một đoạn giao tiếp SMTP, dùng lệnh LIST để xem danh sách mail trên server

22	get_the_number_of_mail_the_user_already_has()	Check xem có bao nhiêu mail đã tải	Kiểm tra bằng database
23	get_all_the_mail_from_sever_that_has_not_been_downloaded()	Tải mail từ server, chỉ tải những mail đã chưa tải	Tải mail từ server, chỉ tải những mail đã chưa tải
24	on_entry_click()	Tạo hoặc thay đổi sự kiện nào đó khi click chuột vào widget	Tạo hoặc thay đổi sự kiện nào đó khi click chuột vào widget
25	on_entry_leave()	Tạo hoặc thay đổi sự kiện nào đó khi ngừng click chuột vào widget	Tạo hoặc thay đổi sự kiện nào đó khi ngừng click chuột vào widget
26	load_and_resize_image()	Tải ảnh từ path để đưa vào image object	Mở path của ảnh sau đó dùng customtkinter để chỉnh lại chiều cao và rộng của ảnh
27	remove_file_mail()	Loại file đính kèm tùy chọn khỏi danh sách file	Loại file đính kèm tùy chọn khỏi danh sách file
28	remove_file_window()	Mở cửa sổ mới để chuẩn bị thao tác remove_file_mail	Mở cửa sổ mới để chuẩn bị thao tác remove_file_mail
29	cut_action()	Thực hiện cắt một chuỗi đã chọn trong text mail	Sử dụng hàm có sẵn event_generate("<<Cut>>") để cắt một chuỗi đã chọn trong text mail
30	copy_action()	Thực hiện copy một chuỗi đã chọn trong text mail	Sử dụng hàm có sẵn event_generate("<<Copy>>") để copy một chuỗi đã chọn trong text mail
31	paste_action()	Thực hiện paste một chuỗi đã chọn trong text mail	Sử dụng hàm có sẵn event_generate("<<Paste>>") để paste một chuỗi đã chọn trong text mail
32	select_all_action()	Hightlight tất cả các text để chọn tất cả nội dung trong text mail	Thêm tag để đánh dấu từ đầu textbox tới cuối textbox
33	find_action()	Tìm kiếm một chuỗi nào đó trong text mail	Sử dụng hàm có sẵn search() để tìm kiếm
34	find_replace_action()	Tìm kiếm và thay thế một chuỗi nào đó trong text mail	Sử dụng hàm có sẵn search() để tìm kiếm và thay thế

35	open_edit_window()	Mở cửa sổ edit để thực hiện các thao tác edit (Cut, copy, ...)	Mở cửa sổ edit để thực hiện các thao tác edit (Cut, copy, ...)
36	zoomIn_action()	Tăng kích thước của một chuỗi nào đó	Tăng kích thước của một chuỗi nào đó
37	zoomOut_action()	Giảm kích thước của một chuỗi nào đó	Giảm kích thước của một chuỗi nào đó
38	reset_action()	Đưa chuỗi về lại kích thước ban đầu, mặc định là size = 12	Đưa chuỗi về lại kích thước ban đầu, mặc định là size = 12
39	open_view_window()	Mở cửa sổ view để thực hiện các thao tác như Zoom In, Zoom Out, ...	Mở cửa sổ view để thực hiện các thao tác như Zoom In, Zoom Out, ...
40	getIndexImage()	Lấy vị trí của ảnh khi ảnh được chèn vào text mail để lưu vào json	Vị trí của ảnh chính là vị trí của con trỏ chuột trong text mail
41	insert_image()	Thực hiện thao tác chèn ảnh vào text mail	Mở file path của ảnh sau đó chỉnh lại kích thước phù hợp để chèn vào text mail
42	change_font()	Thay đổi font của chuỗi đã chọn trong text mail	Thay đổi font của chuỗi đã chọn trong text mail
43	change_style()	Thay đổi style của chuỗi đã chọn bao gồm ("Bold", "Italic", "Strikethrough", ...)	Thay đổi style của chuỗi đã chọn bao gồm ("Bold", "Italic", "Strikethrough", ...)
44	apply_tag()	Dùng để lấy tên của style muốn chọn	Dùng để lấy tên của style muốn chọn
45	text_color_action()	Thay đổi màu cho chuỗi đã chọn	Thay đổi màu cho chuỗi đã chọn
46	open_format_window()	Mở cửa sổ format để thực hiện các thao tác như chỉnh style, font, color, ...	Mở cửa sổ format để thực hiện các thao tác như chỉnh style, font, color, ...
47	close_action()	Đóng cửa sổ mail hiện tại	Đóng cửa sổ mail hiện tại
48	saveAs_action()	Lưu lại nội dung mail hiện tại	Lưu lại nội dung mail hiện tại
49	open_file_window()	Mở cửa sổ file để thực hiện các thao tác như Save, Save as, Close,	Mở cửa sổ file để thực hiện các thao tác như Save, Save as, Close,

50	json__fully__complete__now_send_the_json_file_to_server()	Gửi file json lên server	Gửi temp_email_json lên server.
51	button_toolbar_clicked()	Thực hiện in ra nút đã chọn trong thanh toolbar và thực hiện các event khi click vào	Thực hiện in ra nút đã chọn trong thanh toolbar và thực hiện các event khi click vào
52	button_clicked()	In ra tên nút đã chọn	In ra tên nút đã chọn
53	on_button_click()	In ra tên nút đã chọn	In ra tên nút đã chọn
54	create__button__with__image()	Chèn icon vào nút	Chèn icon vào nút
55	create__button__with__image__senDown()	Tạo các nút có thể làm thành dropdown menu	Tạo các nút có thể làm thành dropdown menu
56	attach_file()	Đính kèm file vào text mail	Đính kèm file vào text mail
57	newMessage()	Mở cửa sổ mới để thực hiện gửi mail	Mở cửa sổ mới để thực hiện gửi mail
58	toggle_additional_buttons()	Tạo dropdown menu cho các nút	Tạo dropdown menu cho các nút
59	create_second_part()	Tạo 2 frame khi mới bắt đầu chạy code	Tạo 2 frame khi mới bắt đầu chạy code
60	create_mail_subframe()	Tạo subframe	Tạo subframe
61	create_buttons_frame()	Tạo một frame cho các nút	Tạo một frame cho các nút

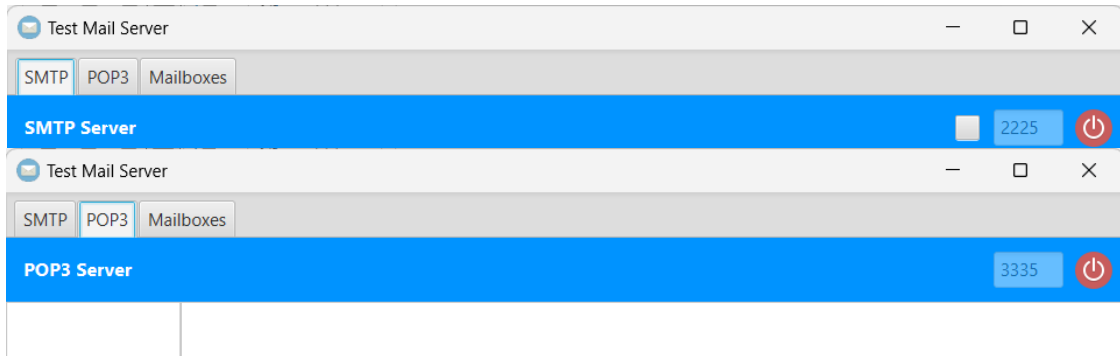
7. Kịch bản giao tiếp của chương trình:

- Giao thức trao đổi giữa client và server: SMTP và POP3
- Cấu trúc thông điệp: Nội dung thư sẽ được lưu trong file json, file này sẽ có các thông tin: Người gửi, người nhận, tiêu đề, nội dung thư, file đính kèm...
- Kiểu dữ liệu của thông điệp: json (được mã hóa base64 trước khi gửi lên)
- Cách tổ chức cơ sở dữ liệu: Là một file json chứa các thông tin các user, mỗi user có 5 hòm thư. Thư sau khi được nhận sẽ lưu vào một trong năm hòm thư.
- Chi tiết:
 - Sau khi người dùng soạn thư xong (điền đầy đủ các thông tin To, From, Subject, Cc, Bcc, ... edit nội dung chính, thêm các file đính kèm), tất cả thông tin đó sẽ được lưu tạm thành 1 file json (Temp_email.json).
 - File json ở trên sẽ được mã hóa và gửi lên server để đến với người nhận bằng giao thức SMTP
 - Bên nhận sau khi nhấn nút Download, ứng dụng sẽ kiểm tra và tải về những email mà user chưa có trong database bằng giao thức POP3. Data của email đó sẽ được lưu tạm vào một file json, file đó sẽ qua một bước kiểm tra để xác định xem email nên vào hộp thư nào.
 - Sau khi xác định hòm thư, data của email sẽ được tích hợp vào database đúng với user.

8. Hướng dẫn sử dụng:

i. Khởi động Thunder Owl

- **Bước 1:** Đảm bảo đã dùng câu lệnh *pip install* để tải các thư viện đã được đề cập ở trên.
- **Bước 2:** Khởi động Test Mail Server với SMTP Port là 2225 và POP3 Port là 3335.



Hình 1: Khởi động Test Mail Server với SMTP Port 2225 và POP3 Port 3335

- **Bước 3:** Mở Visual Studio Code, mở thư mục **Source**, chạy file *Thunder_Owl.py*. Khi chạy xong sẽ được giao diện dưới đây.

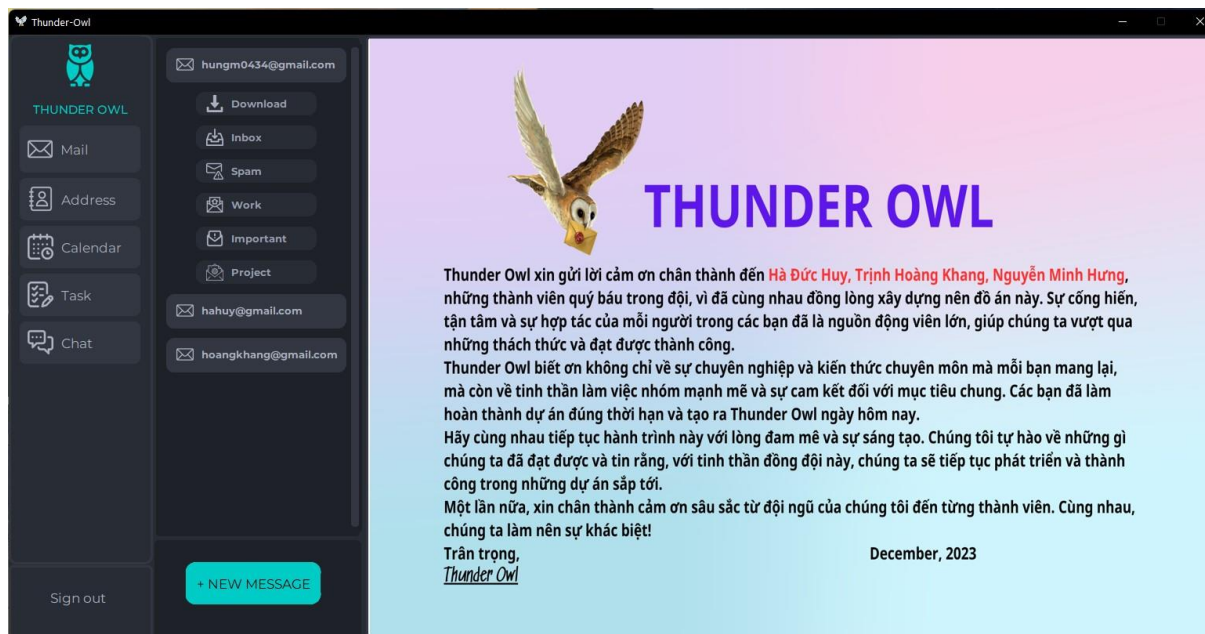


Hình 2: Giao diện Thunder Owl

Hiện tại nhóm chỉ mới phát triển phần Mail nên các nút khác như *Address*, *Calendar*, *Task*, *Chat* sẽ chưa thể sử dụng được. Các thao tác sau đây sẽ chỉ được thực hiện với nút *Mail*.

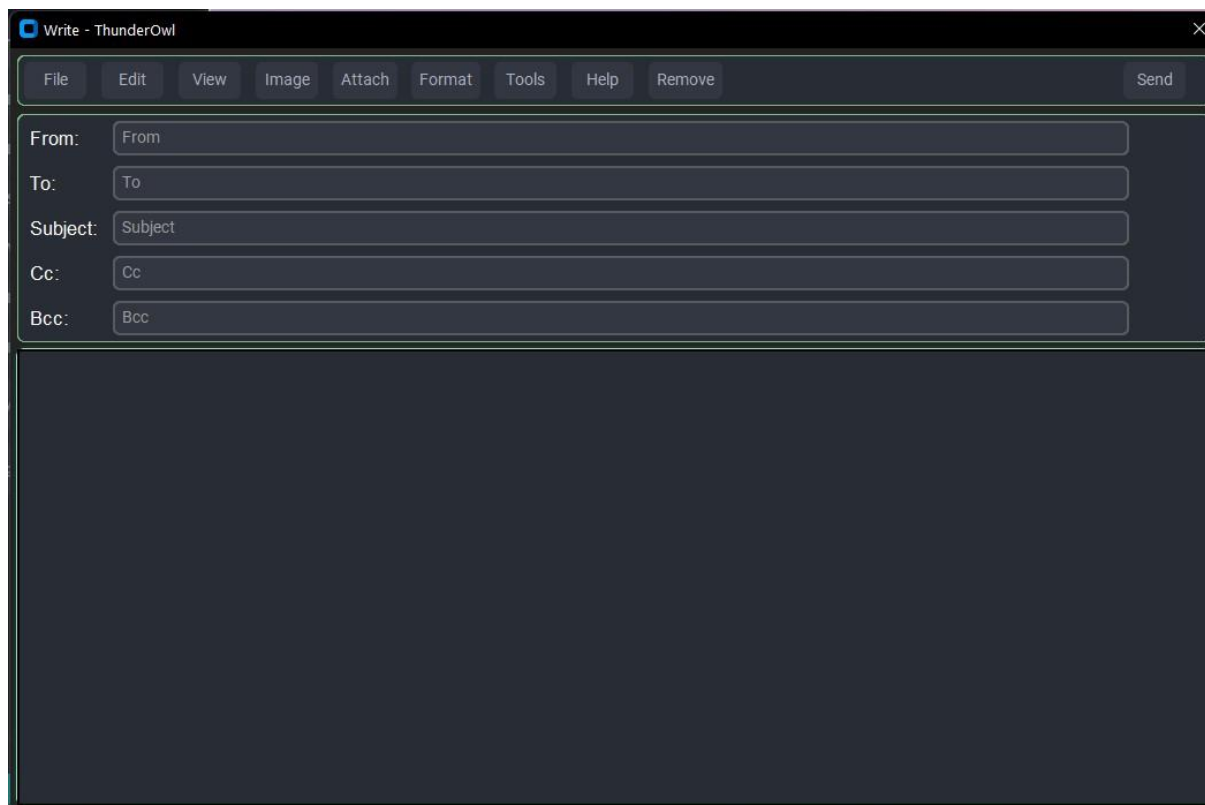
ii. Thao tác gửi email.

- **Bước 1:** Click vào nút Mail để tiến hành thao tác gửi mail. Khi đó sẽ hiện ra giao diện như sau:



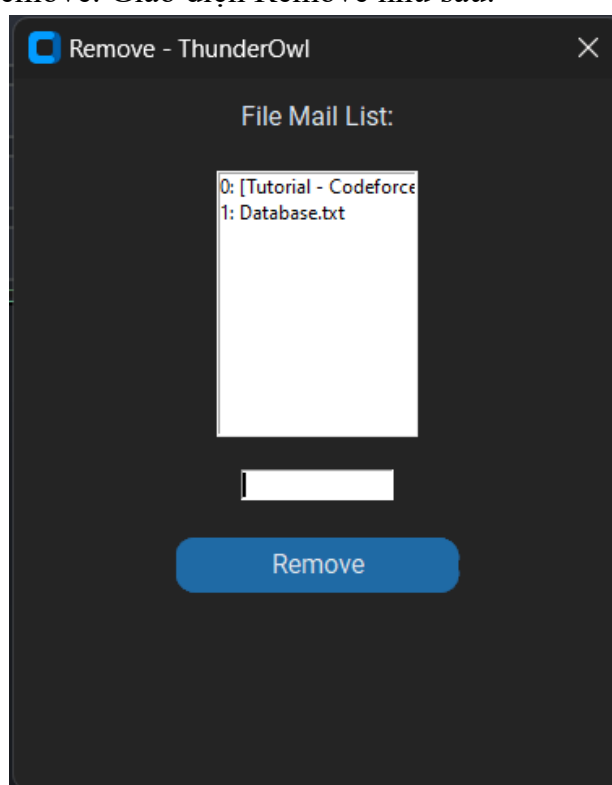
Hình 4: Giao diện mục Mail của Thunder Owl

- **Bước 2:** Chọn NEW MESSAGE, lúc này sẽ hiện ra giao diện gửi email như sau:



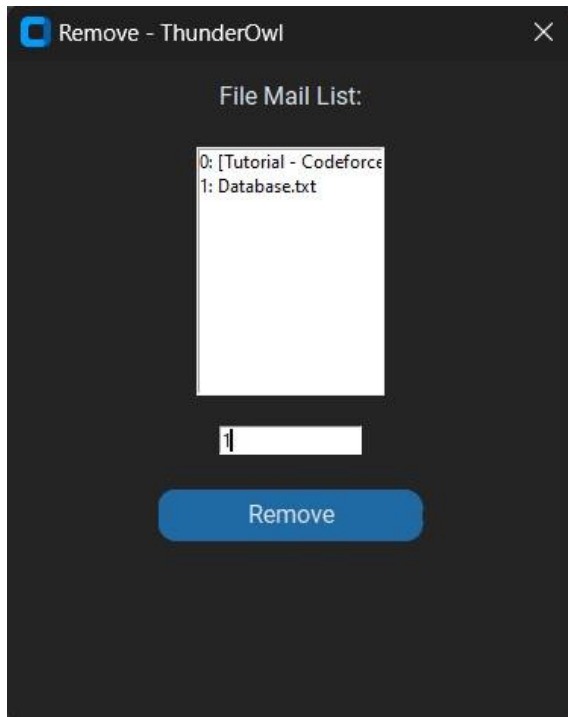
Hình 5: Giao diện mục NEW MESSAGE của Thunder Owl

- **Bước 3:** Soạn thư theo bất kì định dạng nào bạn mong muốn. Có nhiều thao tác để bạn chọn:
 - Soạn thảo văn bản: Trước hết, bạn hãy soạn thảo văn bản bình thường. Sau đó bôi đen văn bản bạn muốn chỉnh sửa. Bạn có thể soạn thư theo nhiều định dạng khác nhau tại mục Format:
 - Phong chữ: Chọn Format → Font. Khi đó sẽ có nhiều font chữ cho bạn lựa chọn như Arial, Terminal,...
 - Kiểu chữ: Chọn Format → Style. Có rất nhiều kiểu chữ cho bạn chọn như Bold, Italic, Underline, ...
 - Màu sắc: Chọn Format → Color. Bạn có thể chọn tùy ý màu sắc nào cho văn bản của bạn, làm cho mail của bạn trở nên đẹp hơn.
 - Chèn ảnh *.png, *.gif, ...: Để nội dung email trở nên sinh động hơn, bạn có thể chèn hình ảnh với nhiều dạng khác nhau bằng cách chọn Image, sau đó chọn một ảnh bất kì trong máy của bạn. Lưu ý rằng khi chèn ảnh xong thì ***nhấp nút phải chuột một lần bên phải ảnh được chèn*** để có thể tiếp tục soạn nội dung email của bạn.
 - Đính kèm file: Bạn có thể đính kèm file bất kì cho email bằng cách sử dụng hộp lệnh Attach. Sau đó chọn file bất kì với nhiều định dạng như *.pdf, *.zip, *.png, *.docx, ... Mỗi lần đính kèm chỉ được một lần nên nếu muốn đính kèm nhiều file thì sẽ thực hiện thao tác này nhiều lần.
 - Xóa file đính kèm: Bạn có thể xóa file đính kèm bằng cách sử dụng hộp lệnh Remove. Giao diện Remove như sau:

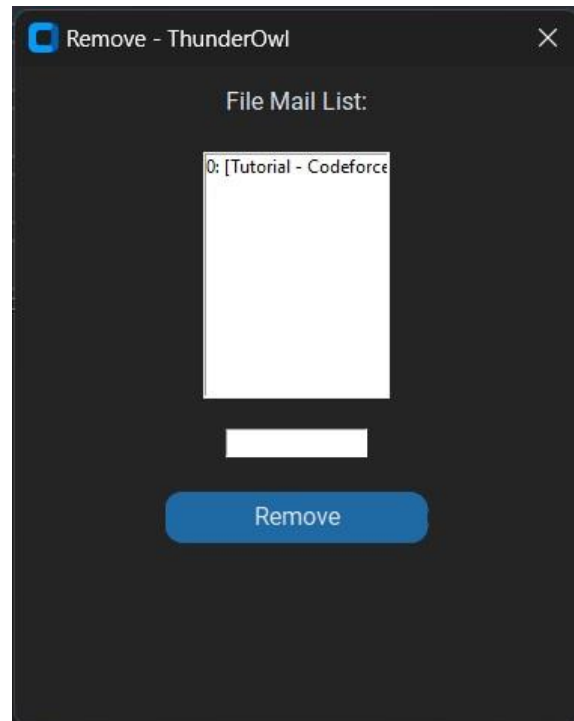


Hình 6 : Giao diện Remove trong Thunder Owl

Bạn muốn bỏ file nào thì chỉ cần nhập số thứ tự của file đó trong hộp File Mail List. Ví dụ cần loại bỏ file có số thứ tự 1 trong hộp File Mail List thì ta sẽ gõ 1 vào trong khoảng trống ở dưới → Remove. Sau đó thoát ra và vào lại, lúc này chỉ còn file có số thứ tự 0.

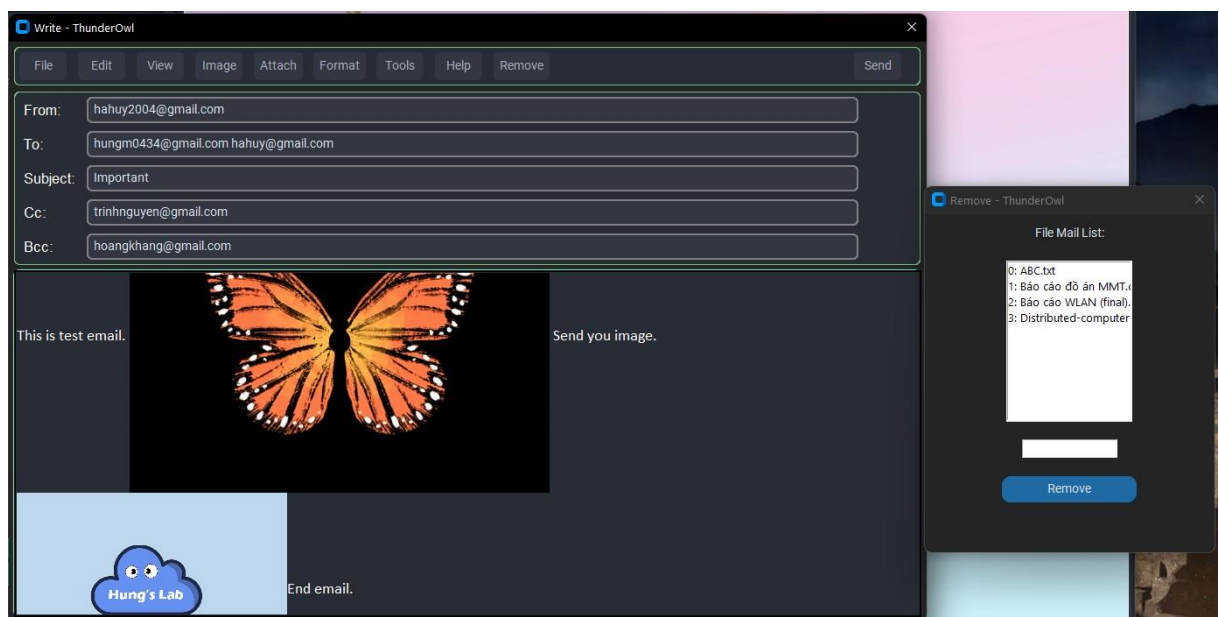


Hình 7 : File Mail List trước khi xóa



Hình 8 : File Mail List sau khi xóa

- **Bước 4:** Ta ấn nút Send, đợi một thời gian rồi thoát ra. Lúc này bạn đã gửi mail thành công.



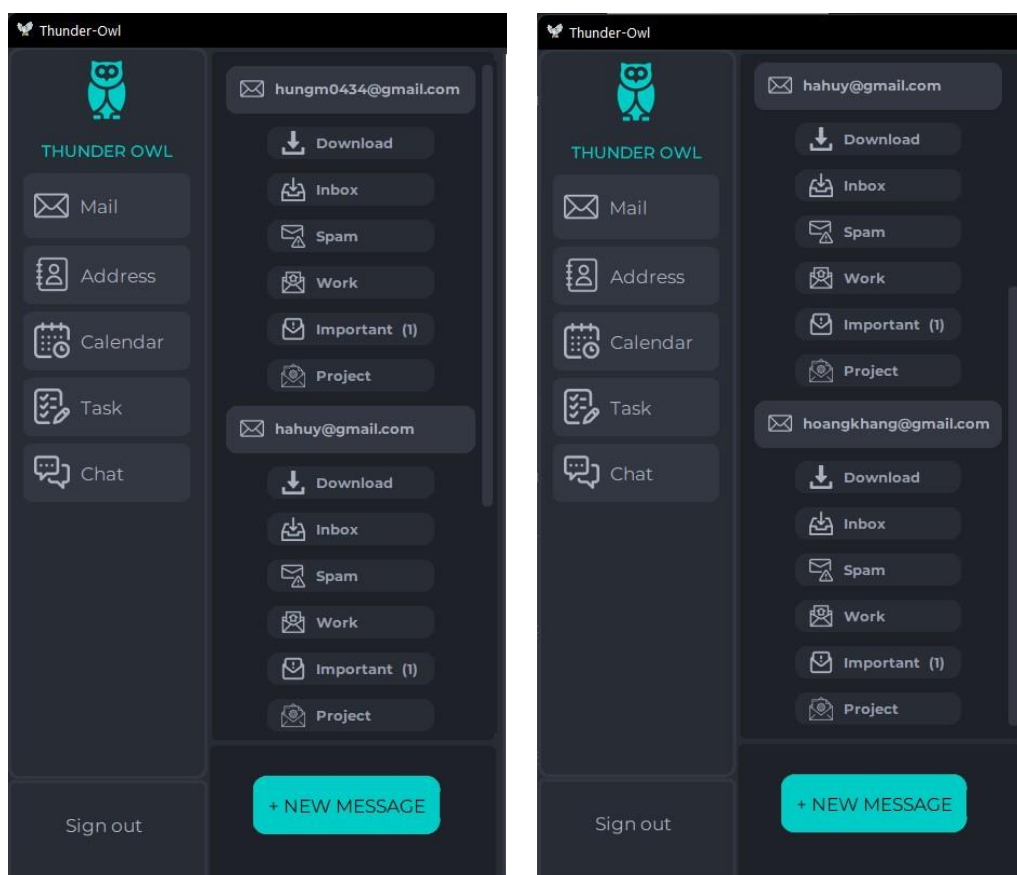
Hình 9: Gửi email trên Thunder Owl

Như bạn thấy thì email sẽ có dạng như sau:

- **From:** haduchuy122004@gmail.com
- **To:** hungm0434@gmail.com hahuy@gmail.com
- **Subject:** Important
- **Cc:** trinhnguyen@gmail.com
- **Bcc:** hoangkhang@gmail.com
- **Content:** Như trên, gồm đoạn văn bản và chèn 1 file *.gif và một file *.png.
- **File đính kèm:** Gồm 3 file, gồm 1 file *.txt, 1 file *.pdf và 1 file *.docx.

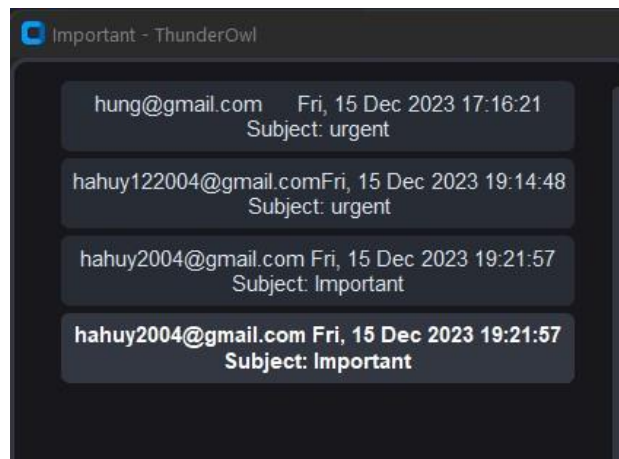
iii. Thao tác tải và xem email.

- **Bước 1:** Sau khi gửi mail xong thì tại các mail được nhận ta nhấn nút Download để tải các mail đó về. Nếu bạn không muốn làm thế thì sẽ có chế độ tự động tải email sau mỗi 10s.
- **Bước 2:** Sau khi tải xong thì ta sẽ kiểm tra các hòm thư của các email. Do mail được gửi ở trên có **Subject: Important** nên thư này sẽ được đưa vào hòm thư **Important**. Kiểm tra ở hòm thư Important, ta thấy hiện kí hiệu (1), tức là có một email chưa đọc, nghĩa là email bạn vừa soạn đã được đưa vào hòm thư Important theo địa chỉ gửi được ghi trong email.



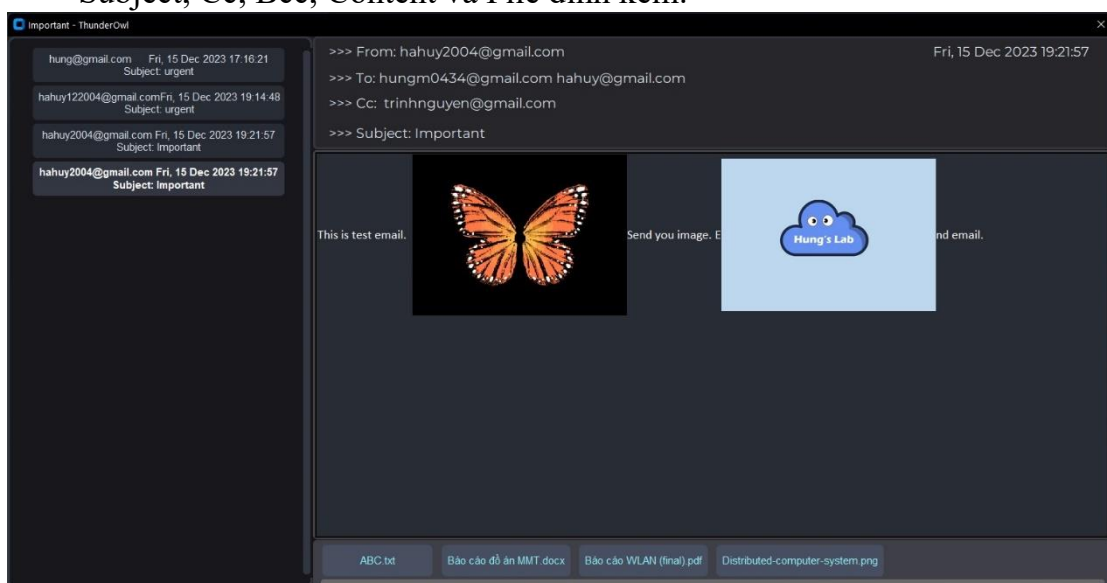
Hình 10: Thư đã được gửi thành công và nằm trong hòm thư Important

- **Bước 3:** Ta tìm kiếm trong hòm thư Important. Khi đó thư mới được gửi sẽ được sáng lên, tức là thư đây chưa được đọc. Còn các thư không sáng lên tức là thư đã được đọc trước đó rồi.



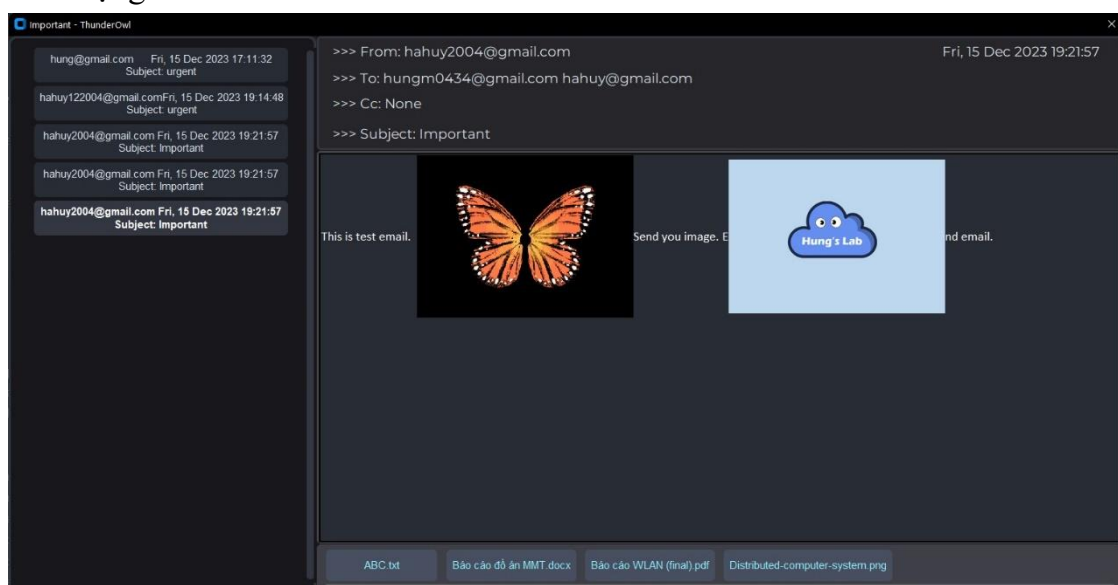
Hình 11: Giao diện hòm thư Important

Khi chọn vào thư đầy thì sẽ thấy đầy đủ thông tin được gửi như là To, From, Subject, Cc, Bcc, Content và File đính kèm.



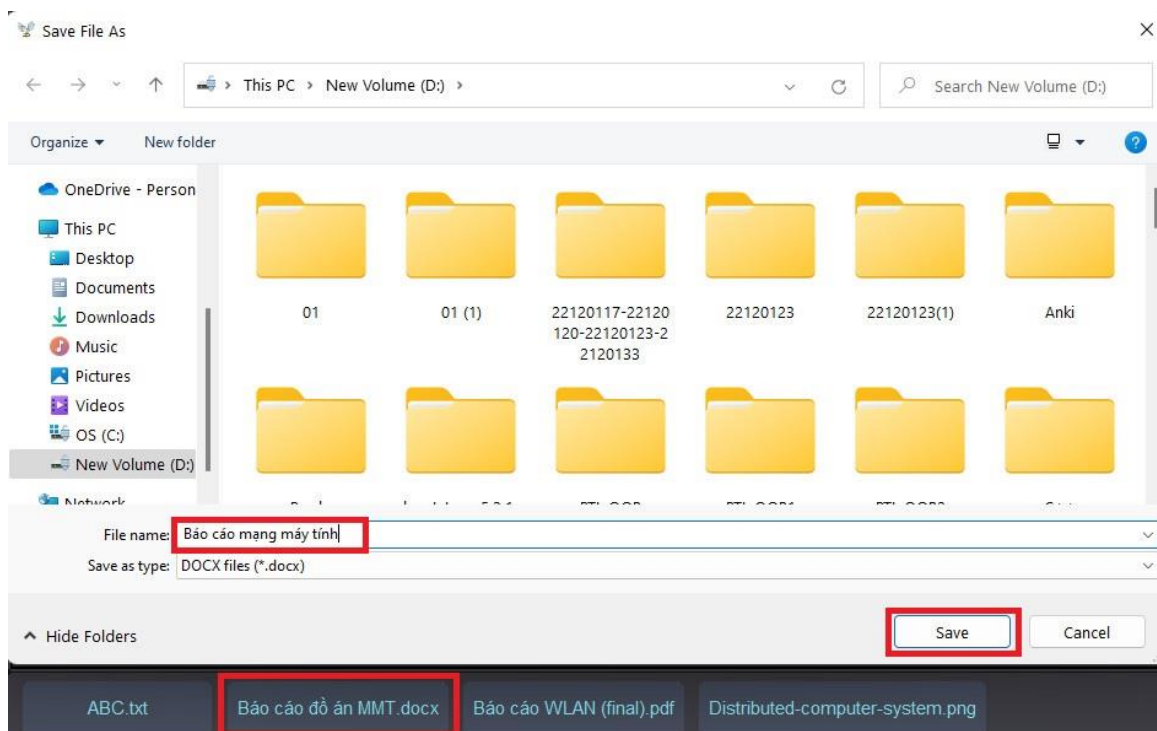
Hình 12: Nội dung chi tiết email được gửi.

Đặc biệt với Bcc thì sẽ không thấy được những người được gửi thư đầy với dạng Cc.



Hình 13: Nội dung chi tiết email được gửi dưới dạng Bcc.

- **Bước 4:** Nếu bạn muốn tải các file được đính kèm trong email đó thì bạn chỉ cần nhấp vào file đấy, chọn đường dẫn và lưu file. Ví dụ ở đây sẽ lưu file có tên là Báo cáo đồ án MMT.docx



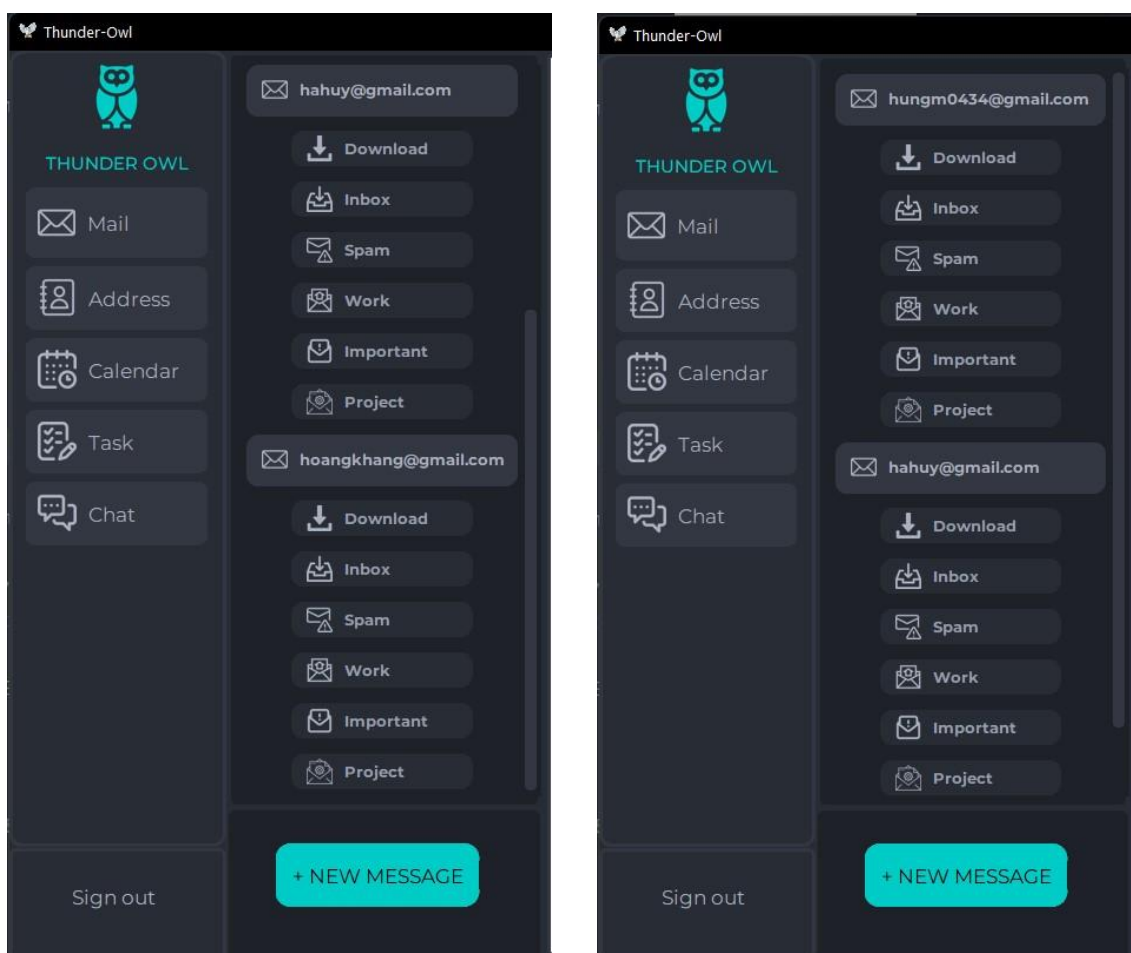
Hình 14: Lưu file về máy tính

Bạn đi đến nơi đã lưu file ở trên, thì lúc này sẽ thấy file bạn lưu tồn tại ở đó. Bạn có thể lưu nhiều file đính kèm trong email đó về nhiều nơi khác nhau trong máy tính



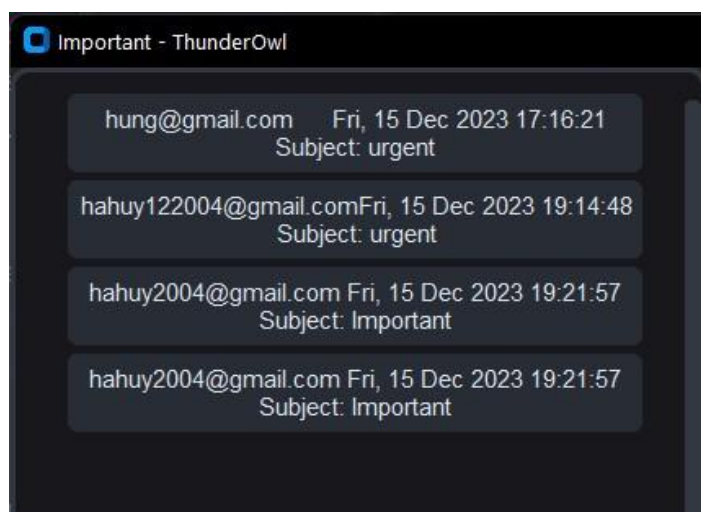
Hình 15: File được lưu xuất hiện tại nơi được lưu.

- **Bước 5:** Sau khi xem xong mail, bạn thoát ra ngoài, nhấp vào mục Mail để load lại, rồi mở các hộp thư để kiểm tra thì ở hộp thư Important sẽ không còn kí hiệu (1) nữa, tức là đã không còn thư nào chưa xem nữa.



Hình 16: Thư đã được xem rồi và hộp thư Important không còn đánh dấu nữa

Ta kiểm tra chi tiết hơn trong hộp thư thì thấy rằng thư vừa đọc không còn sáng lên nữa. Tức là thư được đánh dấu đã đọc.



Hình 17: Thư được đánh dấu *Đã xem* trong hộp thư Important.

VIDEO CHI TIẾT CÁCH HƯỚNG DẪN SỬ DỤNG THUNDER OWL:

Source: [Thunder Owl.mp4](#)

TÀI LIỆU THAM KHẢO:

- [1]. C.T.Linh, N.T.Quan, “Huong Dan.docx”.
- [2]. ChatGPT.