

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ
Bộ môn: Công Nghệ Thông Tin.



BÀI TẬP KẾT THÚC MÔN HỌC
MÔN HỌC
LẬP TRÌNH PYTHON

Sinh viên:Nguyễn Nguyệt Linh

Lớp:K58KTP.K01

Giáo viên GIẢNG DẠY: . . . TS. Nguyễn Văn Huy.

Link GitHub: https://github.com/NguyenNguyetLinh-Beo/btl_python



Thái Nguyên – 2025

TRƯỜNG ĐHKTCN
KHOA ĐIỆN TỬ

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC: LẬP TRÌNH PYTHON

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên: Nguyễn Nguyệt Linh.....

Lớp: K58KTP.K01.....

Ngành: Kỹ Thuật Máy Tính.....

Giáo viên hướng dẫn: TS. Nguyễn Văn Huy

Ngày giao đề: 20/5/2025 **Ngày hoàn thành:** 10/6/2025

Tên đề tài : Ứng dụng quản lý danh bạ GUI

Đầu bài: Xây ứng dụng quản lý danh bạ đơn giản: thêm, sửa, xóa liên hệ với name, phone, email, lưu vào file JSON.

Tính năng yêu cầu:

- Đọc/ghi JSON với module json.
- Bắt lỗi format JSON, lưu khi thêm/sửa/xóa.
- GUI: Entry, Buttons, Treeview.
- Tìm kiếm theo tên.

GIÁO VIÊN HƯỚNG DẪN
(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày....tháng.....năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

LỜI CAM ĐOAN

Em xin cam đoan bài báo cáo với đề tài “Ứng dụng quản lý danh bạ GUI” là kết quả của quá trình học tập, nghiên cứu và thực hiện của cá nhân em dưới sự hướng dẫn của Thầy Nguyễn Văn Huy trong học phần Lập trình Python.

Toàn bộ nội dung, mã nguồn và kết quả trong báo cáo đều do em tự tìm hiểu và xây dựng, không sao chép từ bất kỳ nguồn nào mà không trích dẫn rõ ràng. Nếu có bất kỳ sai sót hoặc vi phạm nào liên quan đến bản quyền hoặc nội dung trong báo cáo, em xin hoàn toàn chịu trách nhiệm trước nhà trường và giáo viên hướng dẫn.

Sinh viên thực hiện

(*Ký và ghi rõ họ tên*)

Linh

Nguyễn Nguyệt Linh

MỤC LỤC

LỜI CAM ĐOAN	4
MỤC LỤC	5
LỜI NÓI ĐẦU	7
CHƯƠNG 1. GIỚI THIỆU ĐẦU BÀI	8
1.1. Đề tài.....	8
1.2. Tính năng chính của chương trình.....	8
1.3. Thách thức khi thực hiện	8
1.4. Kiến thức được vận dụng	9
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	10
2.1. Ngôn ngữ lập trình Python.....	10
2.1.1. Giới thiệu chung	10
2.1.2. Các đặc điểm nổi bật của Python	10
2.1.3. Ứng dụng của Python trong phát triển GUI	10
2.2. Thư viện Tkinter	10
2.2.1. Giới thiệu.....	10
2.2.2. Các widget cơ bản	11
2.2.3. Quản lý bố cục	11
2.2.4. Cơ chế xử lý sự kiện	11
2.3. Xử lý dữ liệu và ngoại lệ	11
2.3.1. Định dạng và cấu trúc dữ liệu JSON	11
2.3.2. Lưu trữ danh bạ trong file JSON	11
2.3.3. Module json trong Python.....	12
2.3.4. Xử lý lỗi khi đọc/ghi file JSON	12
2.3.5. Kiểm tra và xác thực dữ liệu nhập	12
2.4. Các thuật toán và quy trình xử lý.....	12
2.4.1. Thêm liên hệ (Add)	12
2.4.2. Sửa liên hệ (Edit).....	13
2.4.3. Xóa liên hệ (Delete)	13
2.4.4. Tìm kiếm liên hệ (Search)	13
2.5. Mô hình và kiến trúc phần mềm	13
2.5.1. Mô hình MVC (Model-View-Controller).....	13

2.5.2. Quản lý trạng thái và đồng bộ dữ liệu	13
2.6. Quy trình tổng thể của ứng dụng	14
2.7. Tóm tắt	14
CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH.....	15
3.1. Sơ đồ khối hệ thống.....	15
3.2. Sơ đồ các thuật toán chính	16
3.3. Cấu trúc dữ liệu.....	18
3.4. Chương trình	19
3.5. Tóm tắt	19
CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT LUẬN	21
4.1. Thực nghiệm	21
4.2. Kết luận	22

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, lập trình ứng dụng giao diện đồ họa đã và đang trở thành một trong những kỹ năng quan trọng đối với sinh viên ngành Kỹ thuật Máy tính. Trong đó, Python là một ngôn ngữ lập trình được ưa chuộng nhờ cú pháp ngắn gọn, dễ tiếp cận và hệ sinh thái thư viện phong phú. Đặc biệt, thư viện tkinter tích hợp sẵn trong Python cho phép lập trình viên nhanh chóng xây dựng các ứng dụng có giao diện đồ họa thân thiện, trực quan và dễ sử dụng.

Được sự hướng dẫn tận tình của thầy Nguyễn Văn Huy trong học phần Lập trình Python, em đã lựa chọn thực hiện đề tài “Ứng dụng quản lý danh bạ GUI” với mục tiêu vận dụng các kiến thức đã học vào việc xây dựng một phần mềm quản lý danh bạ đơn giản. Thông qua đề tài này, em có cơ hội củng cố lại các kiến thức về lập trình hướng đối tượng, xử lý sự kiện, thao tác với file dữ liệu định dạng JSON và thiết kế giao diện đồ họa bằng thư viện tkinter.

Chương trình Quản lý danh bạ GUI cho phép người dùng nhập thông tin liên hệ gồm họ tên, số điện thoại và email, sau đó thực hiện các chức năng thêm, sửa, xóa liên hệ và lưu trữ dữ liệu vào file `contacts.json`. Đồng thời, ứng dụng còn hỗ trợ hiển thị danh sách liên hệ dưới dạng bảng ngay trên giao diện, giúp người dùng dễ dàng theo dõi và thao tác. Việc sử dụng file JSON để lưu trữ giúp dữ liệu được bảo toàn và dễ dàng khôi phục khi mở lại ứng dụng.

Việc thực hiện đề tài không chỉ giúp em nắm vững hơn về ngôn ngữ lập trình Python và thư viện tkinter, mà còn rèn luyện thêm kỹ năng phân tích, thiết kế, tổ chức mã nguồn cũng như giải quyết vấn đề — những năng lực quan trọng cho công việc chuyên môn sau này. Mặc dù sản phẩm còn một số hạn chế nhất định, nhưng kết quả đạt được đã phần nào thể hiện được nỗ lực học tập và tinh thần sáng tạo của bản thân.

Cuối cùng, em xin gửi lời cảm ơn chân thành tới thầy Nguyễn Văn Huy đã tận tình hướng dẫn, hỗ trợ và tạo điều kiện thuận lợi để em hoàn thành tốt đề tài này. Em rất mong nhận được những ý kiến đóng góp quý báu để tiếp tục hoàn thiện và nâng cao chất lượng sản phẩm trong thời gian tới.

CHƯƠNG 1. GIỚI THIỆU ĐẦU BÀI

1.1. Đề tài

Trong bối cảnh công nghệ phần mềm ngày càng phát triển, việc quản lý thông tin cá nhân, liên hệ khách hàng hay bạn bè qua các ứng dụng máy tính đã trở nên phổ biến và cần thiết. Bài tập này yêu cầu sinh viên xây dựng một ứng dụng quản lý danh bạ đơn giản với giao diện đồ họa (GUI) bằng ngôn ngữ Python, sử dụng thư viện tkinter.

Ứng dụng cho phép người dùng:

- Nhập thông tin liên hệ gồm tên (name), số điện thoại (phone) và email qua các ô nhập liệu.
- Thêm mới, sửa, xóa các liên hệ.
- Hiển thị danh sách các liên hệ dưới dạng bảng (Treeview).
- Tìm kiếm liên hệ theo tên.
- Dữ liệu được lưu trữ vào file contacts.json để đảm bảo tính bền vững khi tắt chương trình.

1.2. Tính năng chính của chương trình

Theo đề bài, chương trình yêu cầu các chức năng sau:

- Nhập thông tin liên hệ: name, phone, email.
- Hiển thị danh sách liên hệ trong bảng Treeview.
- Thêm, sửa, xóa liên hệ từ danh sách.
- Tìm kiếm liên hệ theo tên.
- Dữ liệu được lưu vào file JSON và tự động đọc lại khi mở chương trình.
- Bắt lỗi khi nhập sai định dạng JSON hoặc khi thao tác thêm/sửa/xóa.
- Giao diện trực quan với các thành phần: Entry, Button, Treeview.

1.3. Thách thức khi thực hiện

Trong quá trình thực hiện đề tài, một số thách thức đáng chú ý gồm:

- Xử lý dữ liệu đầu vào: Đảm bảo người dùng nhập đầy đủ và đúng định dạng cho các trường name, phone, email.

- Kiểm tra và xử lý lỗi khi thao tác với file JSON: Cần kiểm tra tính hợp lệ của file, đồng thời tránh lỗi khi file không tồn tại hoặc dữ liệu sai cú pháp.
- Thiết kế giao diện hợp lý, thân thiện: Bố trí các thành phần sao cho dễ thao tác, dễ nhìn.
- Liên kết giữa giao diện và logic xử lý: Cần hiểu rõ cơ chế sự kiện trong tkinter và cách kết hợp các phương thức trong class để thực thi lệnh.

1.4. Kiến thức được vận dụng

Để hoàn thành đề tài này, sinh viên vận dụng các kiến thức sau:

- Lập trình Python cơ bản: Biến, hàm, class, list, dict, vòng lặp, điều kiện, xử lý ngoại lệ.
- Lập trình giao diện với thư viện tkinter: Sử dụng các widget như Tk, Label, Entry, Button, Treeview, messagebox.
- Xử lý dữ liệu JSON: Đọc và ghi file JSON bằng module json.
- Tư duy thiết kế phần mềm: Thiết kế giao diện người dùng trực quan và tổ chức mã nguồn rõ ràng, dễ bảo trì.
- Xử lý sự kiện (event handling) và cập nhật giao diện đồng bộ với dữ liệu.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Ngôn ngữ lập trình Python

2.1.1. Giới thiệu chung

Python là ngôn ngữ lập trình bậc cao, thông dịch (interpreted), đa mục đích, hỗ trợ lập trình hướng đối tượng (OOP) và lập trình thủ tục (procedural programming). Được phát triển bởi Guido van Rossum vào cuối những năm 1980, Python đã trở thành ngôn ngữ phổ biến nhờ cú pháp đơn giản, dễ đọc, dễ học và khả năng mở rộng cao.

2.1.2. Các đặc điểm nổi bật của Python

- **Đa nền tảng:** Python có thể chạy trên hầu hết các hệ điều hành như Windows, Linux, macOS.
- **Cú pháp rõ ràng, thân thiện:** Python ưu tiên viết mã ngắn gọn, dễ đọc, giúp giảm thiểu lỗi.
- **Thư viện chuẩn phong phú:** Python cung cấp nhiều thư viện chuẩn hữu ích hỗ trợ nhiều lĩnh vực, từ xử lý chuỗi, file đến phát triển GUI, khoa học dữ liệu.
- **Hỗ trợ OOP và các mô hình lập trình khác:** Python cho phép khai báo lớp, kế thừa, đa hình, đóng gói dữ liệu giúp tổ chức mã nguồn rõ ràng, dễ bảo trì.
- **Thông dịch:** Python thực thi mã lệnh từng dòng, giúp thuận tiện cho việc kiểm thử, phát triển nhanh.

2.1.3. Ứng dụng của Python trong phát triển GUI

Python kết hợp với các thư viện GUI (Tkinter, PyQt, wxPython...) tạo ra ứng dụng giao diện người dùng thân thiện. Trong đồ án này, Tkinter được chọn vì tính sẵn có, dễ học và tích hợp sẵn với Python.

2.2. Thư viện Tkinter

2.2.1. Giới thiệu

Tkinter là thư viện chuẩn của Python dùng để phát triển giao diện đồ họa đơn giản đến trung bình. Tkinter là “wrapper” cho Tcl/Tk GUI toolkit, cho phép xây dựng các cửa sổ, widget và xử lý sự kiện.

2.2.2. Các widget cơ bản

- **Tk():** Khởi tạo cửa sổ chính ứng dụng.
- **Label:** Hiển thị văn bản tĩnh.
- **Entry:** Ô nhập liệu cho phép người dùng nhập text.
- **Button:** Nút bấm thực hiện hành động khi được nhấn.
- **Treeview (thuộc ttk):** Hiển thị bảng dữ liệu dạng cây hoặc bảng nhiều cột, thích hợp để hiển thị danh sách liên hệ.
- **Radiobutton:** Cho phép người dùng chọn một trong nhiều lựa chọn.
- **Messagebox:** Hộp thoại thông báo (lỗi, cảnh báo, xác nhận).

2.2.3. Quản lý bố cục

Tkinter hỗ trợ ba phương pháp bố trí widget phổ biến:

- **pack():** Đơn giản, xếp widget theo trật tự.
- **grid():** Chia giao diện thành các hàng, cột, bố trí widget theo lưới.
- **place():** Định vị widget theo tọa độ cụ thể (x, y).

Trong đồ án, grid() được ưu tiên vì linh hoạt và dễ tổ chức form nhập liệu và bảng danh bạ.

2.2.4. Cơ chế xử lý sự kiện

Mỗi widget có thể gán một hàm callback để xử lý sự kiện, ví dụ khi nhấn nút, thay đổi dữ liệu, chọn mục. Cơ chế này giúp giao diện tương tác động với người dùng.

2.3. Xử lý dữ liệu và ngoại lệ

2.3.1. Định dạng và cấu trúc dữ liệu JSON

JSON (JavaScript Object Notation) là định dạng dữ liệu dạng văn bản đơn giản, dễ đọc, dễ ghi và phổ biến trong trao đổi dữ liệu. JSON cấu thành từ các kiểu dữ liệu như đối tượng (dictionary), mảng (list), số, chuỗi, boolean, null.

2.3.2. Lưu trữ danh bạ trong file JSON

Trong ứng dụng quản lý danh bạ, dữ liệu lưu trữ được cấu trúc theo danh sách các đối tượng liên hệ. Ví dụ:

[

```
{ "name": "Nguyen Van A", "phone": "0123456789", "email":  
"nguyenvana@example.com" },  
  
{ "name": "Tran Thi B", "phone": "0987654321", "email":  
"tranthib@example.com" }  
  
]
```

2.3.3. Module json trong Python

- **json.load(file):** Đọc dữ liệu JSON từ file và chuyển thành đối tượng Python (list/dict).
- **json.dump(data, file, indent=4):** Ghi dữ liệu Python vào file JSON với định dạng dễ đọc.

2.3.4. Xử lý lỗi khi đọc/ghi file JSON

- Lỗi file không tồn tại (FileNotFoundError): Khi file JSON chưa có, chương trình cần tạo file mới hoặc bắt lỗi và khởi tạo danh bạ rỗng.
- Lỗi định dạng JSON (json.JSONDecodeError): Nếu file bị hỏng hoặc dữ liệu không đúng chuẩn, cần thông báo và khôi phục dữ liệu.

2.3.5. Kiểm tra và xác thực dữ liệu nhập

Để tránh lỗi trong quá trình lưu trữ và sử dụng, chương trình cần kiểm tra:

- Tên không được để trống.
- Số điện thoại chỉ chứa số và có độ dài hợp lý.
- Email có định dạng chuẩn (có thể dùng biểu thức chính quy regex). Việc kiểm tra giúp nâng cao chất lượng dữ liệu, tránh lỗi trong quá trình xử lý sau này.

2.4. Các thuật toán và quy trình xử lý

2.4.1. Thêm liên hệ (Add)

- Thu thập dữ liệu từ form nhập.
- Kiểm tra hợp lệ dữ liệu.
- Thêm dữ liệu vào danh sách liên hệ (list).
- Cập nhật file JSON lưu trữ.

- Cập nhật giao diện hiển thị (Treeview).

2.4.2. Sửa liên hệ (Edit)

- Lấy dữ liệu liên hệ được chọn trên giao diện.
- Cho phép chỉnh sửa các trường thông tin.
- Kiểm tra hợp lệ dữ liệu mới.
- Cập nhật danh sách liên hệ và file JSON.
- Làm mới bảng hiển thị.

2.4.3. Xóa liên hệ (Delete)

- Xác nhận người dùng muốn xóa liên hệ.
- Xóa liên hệ khỏi danh sách.
- Cập nhật file JSON.
- Cập nhật giao diện.

2.4.4. Tìm kiếm liên hệ (Search)

- Nhận từ khóa tìm kiếm (thường là tên hoặc một phần tên).
- Duyệt danh sách lọc các liên hệ có tên chứa từ khóa (không phân biệt hoa thường).
- Hiển thị kết quả tìm kiếm trên giao diện.

2.5. Mô hình và kiến trúc phần mềm

2.5.1. Mô hình MVC (Model-View-Controller)

- **Model:** Lớp quản lý dữ liệu (danh bạ, đọc/ghi JSON, xử lý dữ liệu).
- **View:** Giao diện người dùng (form nhập, bảng hiển thị).
- **Controller:** Xử lý sự kiện người dùng, gọi Model cập nhật dữ liệu, làm mới View.

Áp dụng mô hình này giúp phân tách rõ ràng chức năng, dễ dàng bảo trì và mở rộng.

2.5.2. Quản lý trạng thái và đồng bộ dữ liệu

Khi người dùng thao tác (thêm, sửa, xóa), trạng thái danh bạ cần được cập nhật đồng bộ:

- Cập nhật bộ nhớ trong (list liên hệ).
- Cập nhật file JSON.
- Cập nhật giao diện (Treeview). Quá trình đồng bộ này cần đảm bảo nhất quán để tránh mất mát dữ liệu hoặc lỗi giao diện.

2.6. Quy trình tổng thể của ứng dụng

1. Khởi động ứng dụng:

- Load dữ liệu từ file contacts.json nếu tồn tại.
- Hiển thị danh sách liên hệ lên Treeview.

2. Thao tác người dùng:

- Nhập hoặc sửa thông tin trong các ô Entry.
- Nhấn nút tương ứng (Add/Edit/Delete).
- Chương trình kiểm tra, cập nhật danh bạ và lưu file JSON.
- Hiển thị kết quả trên giao diện, cập nhật Treeview.

3. Tìm kiếm liên hệ:

- Nhập tên hoặc một phần tên.
- Danh sách trên Treeview tự động lọc theo từ khóa.

4. Xử lý lỗi và thông báo:

- Nếu dữ liệu không hợp lệ, hiện hộp thoại cảnh báo.
- Bắt lỗi khi thao tác file hoặc lỗi định dạng JSON.

2.7. Tóm tắt

Chương này trình bày cơ sở lý thuyết quan trọng, làm nền tảng cho việc xây dựng ứng dụng quản lý danh bạ GUI. Các kiến thức về Python, Tkinter, xử lý dữ liệu JSON và các thuật toán quản lý danh bạ được giới thiệu chi tiết, giúp đảm bảo tính khả thi và hiệu quả khi triển khai thực tế.

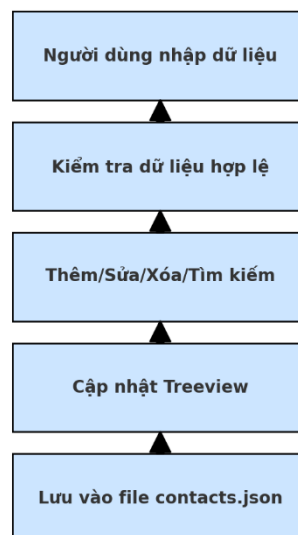
CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Sơ đồ khối hệ thống

Dưới đây là sơ đồ khối mô tả các module chính trong chương trình Quản lý danh bạ GUI:

- **Nhập dữ liệu:** Nhận thông tin liên hệ từ người dùng (Họ tên, Số điện thoại, Email).
- **Lưu dữ liệu:** Lưu dữ liệu liên hệ vào file contacts.json.
- **Xử lý dữ liệu:** Thêm mới, sửa, xóa và tìm kiếm liên hệ.
- **Hiển thị dữ liệu:** Hiển thị danh sách liên hệ qua Treeview.
- **Reset dữ liệu:** Xóa ô nhập và làm mới giao diện.

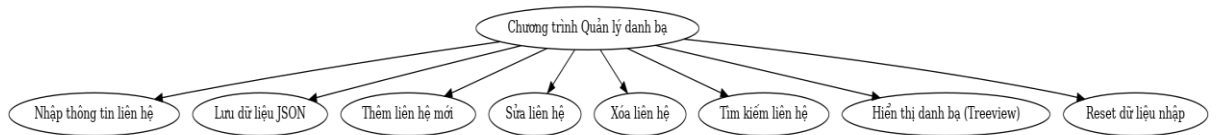
Sơ đồ khối chương trình Quản lý danh bạ GUI



3.1.1. Biểu đồ phân cấp chức năng

- **Chương trình quản lý danh bạ**
 - Nhập thông tin liên hệ
 - Lưu dữ liệu JSON
 - Thêm liên hệ mới
 - Sửa thông tin liên hệ

- Xoá liên hệ
- Tìm kiếm liên hệ
- Hiển thị danh bạ trên giao diện
- Reset dữ liệu nhập



3.2. Sơ đồ các thuật toán chính

Các thuật toán chính có trong ứng dụng Quản lý danh bạ GUI:

► Nhập dữ liệu

- **Chức năng:** Nhận thông tin liên hệ từ người dùng.
- **Đầu vào:** Họ tên, Số điện thoại, Email.
- **Xử lý:**
 - Đọc giá trị từ các ô nhập liệu.
 - Kiểm tra tính hợp lệ (trống, sai định dạng số điện thoại/email).
- **Đầu ra:** Thông tin hợp lệ hoặc thông báo lỗi.

► Thêm liên hệ

- **Chức năng:** Thêm liên hệ mới vào danh sách và lưu vào file JSON.
- **Đầu vào:** Thông tin liên hệ.
- **Xử lý:**
 - Kiểm tra dữ liệu.
 - Thêm vào danh sách.
 - Ghi lại vào file JSON.

- Cập nhật Treeview.
- **Đầu ra:** Treeview hiển thị danh sách mới.

► **Sửa thông tin liên hệ**

- **Chức năng:** Sửa thông tin của liên hệ đã chọn.
- **Đầu vào:** Thông tin mới.
- **Xử lý:**
 - Lấy chỉ số liên hệ được chọn.
 - Cập nhật dữ liệu trong danh sách và JSON.
 - Cập nhật Treeview.
- **Đầu ra:** Treeview mới.

► **Xoá liên hệ**

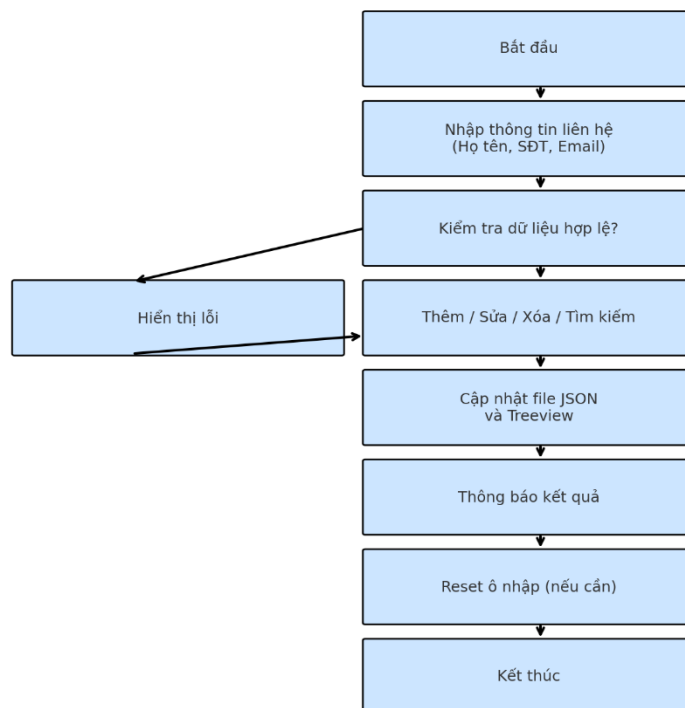
- **Chức năng:** Xoá liên hệ đang được chọn.
- **Đầu vào:** Vị trí dòng trong Treeview.
- **Xử lý:**
 - Xoá trong danh sách.
 - Ghi lại file JSON.
 - Cập nhật Treeview.
- **Đầu ra:** Danh sách mới.

► **Tìm kiếm liên hệ**

- **Chức năng:** Tìm và hiển thị các liên hệ chứa từ khoá nhập vào.
- **Đầu vào:** Từ khoá tìm kiếm.
- **Xử lý:**
 - Lọc danh sách theo từ khoá.
 - Cập nhật Treeview chỉ hiển thị kết quả phù hợp.
- **Đầu ra:** Treeview lọc kết quả.

► Reset dữ liệu

- **Chức năng:** Xoá sạch ô nhập liệu.
- **Đầu vào:** Nhấn nút Reset.
- **Xử lý:**
 - Xoá nội dung các ô Entry.
- **Đầu ra:** Giao diện trắng.



3.3. Cấu trúc dữ liệu

Trong chương trình, dữ liệu được tổ chức như sau:

- **contacts:** Danh sách các liên hệ dạng list chứa các dictionary:

```
contacts = [
    {"name": "Nguyen Van A", "phone": "0123456789", "email":
    "a@gmail.com"},
]
```

- **contacts.json:** File lưu trữ dữ liệu dạng JSON.

- Các widget điều khiển:
 - **entry_name, entry_phone, entry_email**: các ô Entry nhập dữ liệu.
 - **search_entry**: ô tìm kiếm.
 - **tree**: Treeview hiển thị danh sách liên hệ.
 - **result_var**: StringVar lưu kết quả thông báo.
 - Các nút: Add, Edit, Delete, Search, Reset.

3.4. Chương trình

Chương trình gồm các hàm chính:

- **load_contacts()**: Đọc file JSON và load vào danh sách.
- **save_contacts()**: Ghi danh sách vào file JSON.
- **add_contact()**: Thêm liên hệ mới.
- **edit_contact()**: Sửa thông tin liên hệ được chọn.
- **delete_contact()**: Xoá liên hệ.
- **search_contact()**: Tìm kiếm theo từ khoá.
- **reset_fields()**: Xoá dữ liệu nhập.
- **update_treeview()**: Cập nhật Treeview khi dữ liệu thay đổi.

Giao diện người dùng (GUI):

- Xây dựng bằng Tkinter:
 - **Label + Entry**: Nhập thông tin.
 - **Button**: Chức năng Thêm, Sửa, Xoá, Tìm, Reset.
 - **Treeview + Scrollbar**: Hiển thị danh sách.
 - **Messagebox**: Thông báo lỗi và kết quả.

3.5. Tóm tắt

Chương trình được thiết kế chia module theo hướng:

- Giao diện độc lập.
- Xử lý dữ liệu riêng.
- Tương tác với file JSON tách biệt.

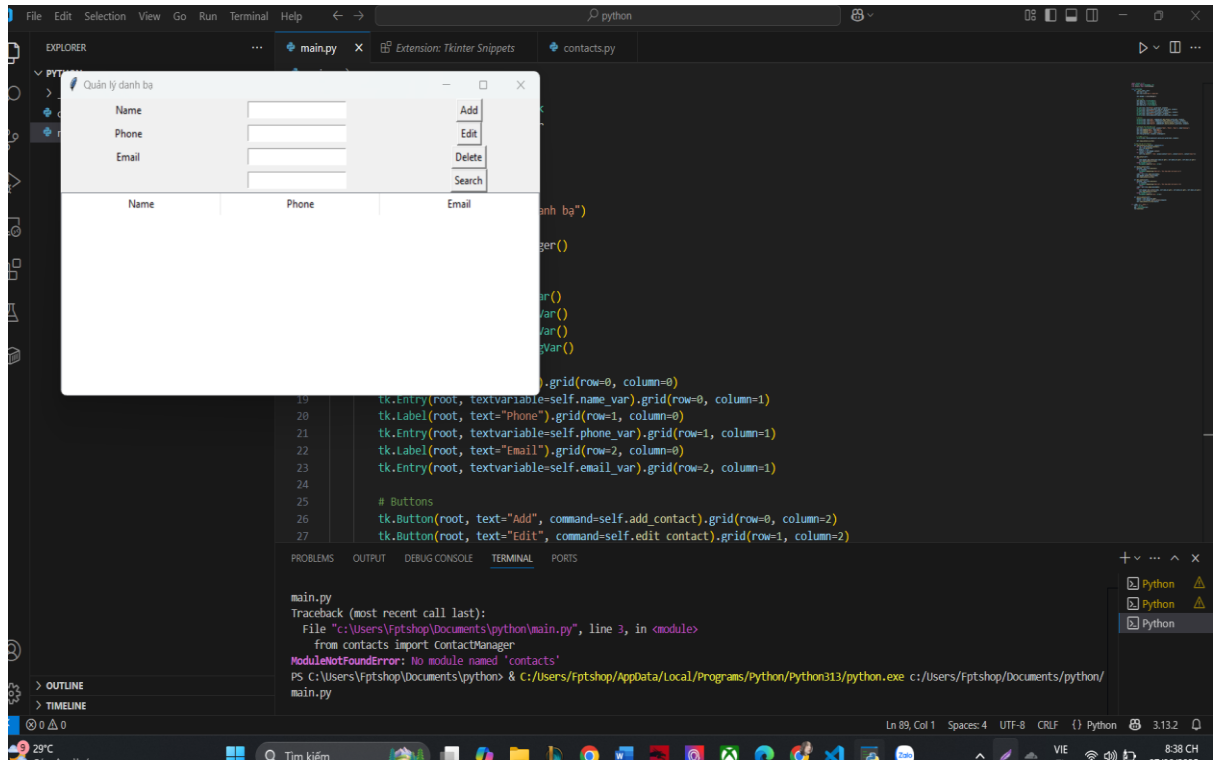
Giúp dễ mở rộng, dễ bảo trì, và thân thiện với người dùng.

CHƯƠNG 4: THỰC NGHIỆM VÀ KẾT LUẬN

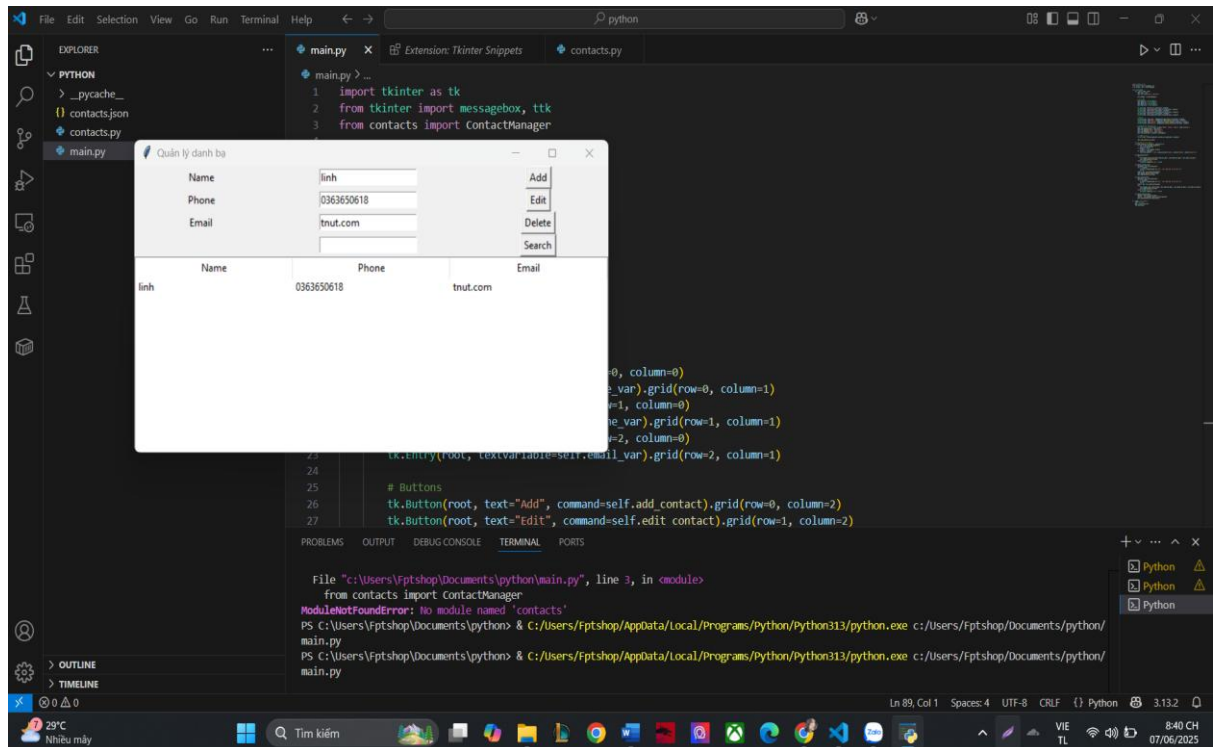
4.1. Thực nghiệm

Chương trình đã được chạy thử và kiểm tra các tính năng chính:

Giao diện:



Nhập và nhận được kết quả:



4.2. Kết luận

Qua quá trình thiết kế và xây dựng, chương trình **Quản lý Danh bạ GUI bằng Python** đã hoàn thiện với đầy đủ các chức năng cơ bản như: thêm mới, sửa, xoá, tìm kiếm liên hệ, lưu trữ vào file JSON, hiển thị danh sách liên hệ bằng giao diện trực quan và hỗ trợ người dùng thao tác dễ dàng.

Những điểm đạt được:

- Giao diện rõ ràng, trực quan và dễ sử dụng nhờ thư viện **tkinter**.
- Xử lý lỗi hợp lý: kiểm tra dữ liệu nhập vào (trống, sai định dạng số điện thoại, email).
- Tổ chức mã nguồn rõ ràng, chia module hợp lý, dễ bảo trì và mở rộng.
- Dữ liệu lưu trữ vào file JSON giúp ứng dụng lưu giữ thông tin lâu dài qua các phiên chạy.
- Phản hồi tức thời và thông báo rõ ràng khi người dùng thao tác.

Hạn chế:

- Chưa hỗ trợ phân loại nhóm liên hệ (gia đình, bạn bè, công việc...).
- Giao diện còn đơn giản, ít lựa chọn về màu sắc, chủ đề và chưa hỗ trợ hiển thị ảnh đại diện.
- Chưa có chức năng sắp xếp danh sách liên hệ.

Hướng phát triển:

- Bổ sung tính năng phân nhóm liên hệ và sắp xếp theo tên, số điện thoại.
- Nâng cấp giao diện với nhiều màu sắc, chủ đề và hỗ trợ avatar cho từng liên hệ.
- Hỗ trợ xuất dữ liệu sang các định dạng khác như CSV, Excel.
- Thêm tính năng đồng bộ hóa dữ liệu với dịch vụ lưu trữ đám mây hoặc cơ sở dữ liệu online.