

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ - BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÀI TẬP KẾT THÚC HỌC PHẦN MÔN HỌC
PYTHON

NGÀNH : KỸ THUẬT MÁY TÍNH

HỆ : ĐẠI HỌC CHÍNH QUY

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

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 : Trần Thị Thu Hà
MSSV : K225480106009
Lớp : K58KTP.
Giáo viên giảng dạy : TS. Nguyễn Văn Huy
Link GitHub : https://github.com/ThuhaTNUT/BL_Python_UDQLDBGUI.git

THÁI NGUYÊN – 2025

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC: LẬP TRÌNH PYTHON

Sinh viên : Trần Thị Thu Hà **MSSV** : K225480106009

Lớp : K58KTP **Ngành** : Kỹ thuật máy tính

Bộ môn: Công Nghệ Thông Tin

Giáo viên hướng dẫn: TS. Nguyễn Văn Huy

Ngày giao đề : 20/05/2025 **Ngày hoàn thành** : 09/06/2025

Tên đề tài : Xây dựng ứng dụng quản lý danh bạ đơn giản: thêm, sửa, xoá liên hệ với name, phone, email, lưu vào file JSON.

Yêu cầu :

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

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

PHIẾU GHI ĐIỂM
BÀI TẬP KẾT THÚC HỌC PHẦN PYTHON

Sinh viên : Trần Thị Thu Hà

MSSV : K225480106009

Lớp : K58KTP

GVHD: TS. Nguyễn Văn Huy

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

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....
.....
.....
.....

Xếp loại : Điểm :

Thái Nguyên, ngày tháng năm 2025.

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CAM ĐOAN	5
LỜI NÓI ĐẦU	6
DANH MỤC HÌNH ẢNH.....	7
CHƯƠNG 1. GIỚI THIỆU ĐẦU BÀI	8
1.1. Mục tiêu đề tài.....	8
1.2. Các tính năng của chương trình	8
1.3. Các thách thức khi thực hiện	9
1.4. Các kiến thức và công nghệ áp dụng để thực hiện bài toán	9
1.4.1. Ngôn ngữ lập trình python	10
1.4.2. Thư viện Tkinter	10
1.4.3. Thư viện json.....	10
1.4.4. Kiến thức về cấu trúc dữ liệu	11
1.4.5. Xử lý lỗi và kiểm tra dữ liệu đầu vào	11
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	12
2.1. Danh sách (List) trong Python.....	12
2.2. Từ điển (Dictionary) trong Python.....	12
2.3. Xử lý file JSON trong Python	12
2.4. Giao diện đồ họa người dùng (GUI) với Tkinter	13
2.5. Xử lý sự kiện trong GUI.....	13
2.6. Kiểm tra dữ liệu đầu vào (Validation)	13
CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH.....	15
3.1. Sơ đồ khối hệ thống.....	15
3.1.1. Các module chính trong chương trình ứng dụng quản lý danh bạ GUI	15

3.1.2. Biểu đồ phân cấp chức năng.....	15
3.2. Sơ đồ khối các thuật toán chính	16
3.2.1. Sơ đồ khối các thuật toán	16
3.2.2. Phân tích các khối thuật toán	19
3.2.3. Quan hệ đầu vào/ra giữa các khối	20
3.3. Cấu trúc dữ liệu.....	22
3.3.1. Cấu trúc dữ liệu đầu vào	22
3.3.2. Cấu trúc dữ liệu sau tiền xử lý.....	22
3.3.3. Cấu trúc dữ liệu đầu ra	23
3.4. Các hàm chính trong chương trình chính	23
3.4.1. Các hàm chính trong file <i>contacts.py</i>	23
3.4.2. Các hàm chính trong file <i>main.py</i>	24
CHƯƠNG 4. KIỂM THỬ VÀ ĐÁNH GIÁ HỆ THỐNG	26
4.1. Kết quả kiểm thử.....	26
4.2. Kết luận	26
4.2.1. Kết quả sản phẩm đã đạt được	26
4.2.2. Kiến thức đã học được trong quá trình làm đề tài	27
TÀI LIỆU THAM KHẢO	28
LINK VIDEO TRÊN YOUTUBE VÀ GITHUB.....	29

LỜI CAM ĐOAN

Em xin cam đoan rằng đề tài bài tập lớn "Ứng dụng quản lý danh bạ GUI" là kết quả của quá trình nghiên cứu và thực hiện của em.

Toàn bộ nội dung trong đề tài này đều do em tự thực hiện, dựa trên các kiến thức đã học và các nguồn tài liệu tham khảo được liệt kê trong danh mục tài liệu tham khảo. Em không sao chép nội dung của bất kỳ cá nhân hoặc tổ chức nào một cách trái phép. Nếu phát hiện có bất kỳ hành vi gian lận hoặc vi phạm quy định về nghiên cứu khoa học, chúng em xin hoàn toàn chịu trách nhiệm trước nhà trường và pháp luật.

Thái Nguyên, ngày tháng năm 2025

Người thực hiện

(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, việc lưu trữ và quản lý thông tin liên lạc một cách khoa học và tiện lợi đóng vai trò rất quan trọng, đặc biệt là trong môi trường học tập và làm việc. Với mục tiêu giúp người dùng dễ dàng thao tác và truy xuất dữ liệu liên hệ, đề tài “*Ứng dụng quản lý danh bạ GUI*” được xây dựng nhằm cung cấp một giải pháp phần mềm đơn giản, trực quan nhưng đầy đủ chức năng cần thiết như thêm, sửa, xóa và tìm kiếm thông tin liên hệ. Ứng dụng này sử dụng ngôn ngữ lập trình Python kết hợp thư viện Tkinter để xây dựng giao diện đồ họa người dùng, cùng với module json để lưu trữ và truy xuất dữ liệu dưới dạng tệp JSON.

Thông qua giao diện gồm các ô nhập liệu (Entry), các nút chức năng (Button), và bảng hiển thị danh bạ (Treeview), người dùng có thể dễ dàng thao tác với thông tin liên hệ. Hệ thống còn hỗ trợ kiểm tra dữ liệu đầu vào, bắt lỗi định dạng file JSON, đồng thời đảm bảo đồng bộ giữa dữ liệu trên giao diện và file lưu trữ. Bên cạnh đó, chức năng tìm kiếm theo tên cũng được tích hợp để hỗ trợ truy xuất nhanh chóng. Đề tài không chỉ giúp sinh viên chúng em rèn luyện kỹ năng lập trình và tư duy thiết kế phần mềm mà còn là tiền đề để phát triển các ứng dụng quản lý dữ liệu lớn hơn sau này.

DANH MỤC HÌNH ẢNH

Hình 3.1. Biểu đồ chức năng của ứng dụng quản lý danh bạ GUI

Hình 3.2. Sơ đồ khối thuật toán đọc/ghi file json

Hình 3.3. Sơ đồ khối thuật toán thêm liên hệ

Hình 3.4. Sơ đồ khối thuật toán sửa liên hệ

Hình 3.5. Sơ đồ khối thuật toán xóa liên hệ

Hình 3.6. Sơ đồ khối thuật toán tìm kiếm

Hình 4.1. Giao diện chính của ứng dụng

CHƯƠNG 1. GIỚI THIỆU ĐẦU BÀI

1.1. Mục tiêu đề tài

Đề tài "*Ứng dụng quản lý danh bạ GUI*" là đề tài hướng tới xây dựng một phần mềm giao diện đồ họa (GUI) đơn giản để quản lý danh bạ liên lạc. Ứng dụng này cho phép người dùng có thể thực hiện các chức năng cơ bản như thêm, sửa, xóa các liên hệ, với các thông tin bao gồm: tên (name), số điện thoại (phone), và email. Dữ liệu được lưu trữ dưới dạng file JSON, giúp đảm bảo tính đơn giản và linh hoạt trong việc lưu/đọc dữ liệu.

1.2. Các tính năng của chương trình

Ứng dụng quản lý danh bạ GUI được xây dựng nhằm hỗ trợ người dùng lưu trữ và quản lý thông tin liên hệ (tên, số điện thoại, email) một cách thuận tiện và trực quan thông qua giao diện đồ họa. Các tính năng chính của chương trình bao gồm:

- *Thêm liên hệ mới*: Người dùng có thể nhập thông tin tên, số điện thoại và email vào các ô nhập liệu (Entry). Sau khi nhấn nút “Thêm” (Add), hệ thống sẽ kiểm tra dữ liệu hợp lệ rồi lưu vào danh sách, đồng thời cập nhật nội dung bảng hiển thị (Treeview) và file contacts.json
- *Sửa thông tin liên hệ*: Khi người dùng chọn một liên hệ từ bảng danh bạ và sửa thông tin trong các ô nhập, sau đó nhấn nút “Sửa” (Edit) thì chương trình sẽ cập nhật các thông tin mới trong danh sách liên hệ và lưu lại vào file JSON.
- *Xóa liên hệ*: Chức năng này cho phép người dùng chọn và xóa liên hệ nào đó ra khỏi danh sách bằng cách chọn liên hệ và sau đó nhấn nút delete. Sau khi xóa liên hệ, danh sách và file contacts.json sẽ được cập nhật lại tương ứng
- *Tìm kiếm theo tên*: Tính năng này cho phép người dùng có thể tìm kiếm liên hệ một cách nhanh chóng và chính xác bằng cách nhập một từ khóa tên liên hệ vào ô tìm kiếm sau đó chương trình sẽ lọc và hiển thị các liên hệ có tên

chứa từ khóa đó trong bảng danh bạ.

1.3. Các thách thức khi thực hiện

a. Thiết kế giao diện

Giao diện người dùng (GUI) cần đảm bảo thân thiện, dễ hiểu và bố cục hợp lý. Việc sử dụng Tkinter để sắp xếp các thành phần như Entry, Button, Treeview đòi hỏi sự tỉ mỉ trong căn chỉnh, đặc biệt là đảm bảo hiển thị tốt trên các độ phân giải màn hình khác nhau.

b. Quản lý dữ liệu với định dạng JSON

Việc đọc/ghi file JSON yêu cầu phải đảm bảo đúng định dạng dữ liệu. Nếu file contacts.json bị thay đổi thủ công hoặc lỗi, chương trình cần có cơ chế kiểm soát lỗi để không gây mất dữ liệu hoặc treo ứng dụng khi khởi động.

c. Đồng bộ dữ liệu giữa giao diện và bộ nhớ

Sau mỗi thao tác thêm, sửa, xóa, danh sách trên giao diện cần được cập nhật chính xác và đồng bộ với dữ liệu trong bộ nhớ (danh sách list) và file JSON. Việc này đòi hỏi xây dựng quy trình xử lý rõ ràng, tránh trùng lặp và lỗi logic.

d. Kiểm tra và xử lý đầu vào

Chương trình cần kiểm tra dữ liệu đầu vào từ người dùng: tránh để trống, tránh trùng lặp, kiểm tra định dạng email và số điện thoại, đồng thời thông báo lỗi một cách rõ ràng nếu có sai sót.

e. Tổ chức chương trình theo hướng module hóa

Việc chia chương trình thành các module như contacts.py (xử lý logic) và gui.py (giao diện) đòi hỏi phải hiểu rõ nguyên tắc tách biệt trách nhiệm và quản lý giao tiếp giữa các phần

1.4. Các kiến thức và công nghệ áp dụng để thực hiện bài toán

Để xây dựng được một ứng dụng quản lý danh bạ GUI thì chương trình cần

sử dụng các kiến thức và công nghệ sau:

1.4.1. Ngôn ngữ lập trình python

Ngôn ngữ lập trình Python là một ngôn ngữ lập trình bậc cao, thông dịch, hướng đối tượng và có cú pháp đơn giản dễ học. Ngôn ngữ này có các đặc điểm nổi bật như:

- Dễ học, dễ đọc vì cú pháp của nó gần giống với ngôn ngữ tự nhiên
- Mã nguồn mở
- Đa nền tảng: chạy được trên nhiều hệ điều hành như Windows, macOS, Linux
- Có thư viện phong phú: có nhiều thư viện hỗ trợ GUI và hỗ trợ nhiều lĩnh vực như AI, web, game, xử lý ảnh,...

Nhờ các ưu điểm trên dễ triển khai và có nhiều thư viện hỗ trợ GUI, xử lý dữ liệu nên toàn bộ chương trình quản lý danh bạ GUI được viết bằng Python

1.4.2. Thư viện Tkinter

Tkinter là thư viện tiêu chuẩn của Python được dùng để xây dựng giao diện đồ họa người dùng (GUI – Graphical User Interface). Nó là giao diện Python của bộ công cụ Tk GUI được tích hợp sẵn trong hầu hết các bản Python nên ta không cần phải cài đặt thêm.

Trong đề bài xây dựng ứng dụng quản lý danh bạ GUI ta sẽ sử dụng thư viện chuẩn Tkinter này để xây dựng giao diện đồ họa cho ứng dụng bao gồm:

- *Entry*: cho phép nhập thông tin liên hệ.
- *Button*: thao tác thêm, sửa, xóa.
- *Treeview*: hiển thị danh bạ dưới dạng bảng.

1.4.3. Thư viện json

JSON là một thư viện chuẩn trong Python được dùng để mã hóa và giải mã dữ liệu theo định dạng JSON. Nó là định dạng phổ biến được dùng để trao đổi thông tin giữa các hệ thống với nhau.

Trong đề tài ứng dụng quản lý danh bạ GUI này sẽ không thể thiếu được thư viện json, nó được dùng để lưu trữ và đọc dữ liệu từ file contacts.json với định dạng chuẩn. Việc này giúp lưu thông tin liên hệ một cách lâu dài và dễ dàng chia sẻ hoặc sao lưu.

1.4.4. Kiến thức về cấu trúc dữ liệu

Chương trình sử dụng danh sách (list) và từ điển (dictionary) để quản lý tập hợp các liên hệ. Mỗi liên hệ được biểu diễn bằng một từ điển với các khóa name, phone, email.

1.4.5. Xử lý lỗi và kiểm tra dữ liệu đầu vào

Áp dụng các kỹ thuật kiểm tra dữ liệu đầu vào (validation), xử lý ngoại lệ (exception handling) để đảm bảo chương trình hoạt động ổn định, không bị lỗi khi gặp đầu vào không hợp lệ.

Tóm tắt chương:

Chương này giới thiệu tổng quan về đề tài “Ứng dụng quản lý danh bạ GUI”, một chương trình đơn giản giúp người dùng thêm, sửa, xóa và tìm kiếm thông tin liên hệ (gồm tên, số điện thoại, email) thông qua giao diện đồ họa sử dụng thư viện Tkinter. Dữ liệu được lưu trữ dưới dạng file JSON nhằm đảm bảo tính đơn giản và linh hoạt.

Các tính năng chính gồm: thêm liên hệ mới, chỉnh sửa, xóa và tìm kiếm theo tên. Trong quá trình phát triển, chương trình gặp nhiều thách thức như thiết kế giao diện thân thiện, đảm bảo đồng bộ dữ liệu giữa giao diện và bộ nhớ, xử lý dữ liệu JSON ổn định, và kiểm tra dữ liệu đầu vào chặt chẽ.

Đề tài áp dụng các công nghệ và kiến thức như: ngôn ngữ Python, thư viện Tkinter (tạo giao diện), thư viện json (xử lý dữ liệu), cấu trúc dữ liệu (list, dictionary) và kỹ thuật kiểm tra lỗi/ngoại lệ để xây dựng ứng dụng hiệu quả, dễ sử dụng và dễ mở rộng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Trong chương trình quản lý danh bạ giao diện đồ họa (GUI), các kiến thức lý thuyết và công nghệ liên quan bao gồm: cấu trúc dữ liệu Python, kỹ thuật xử lý file JSON, thiết kế giao diện người dùng với thư viện Tkinter, và xử lý sự kiện trong GUI. Các nội dung cơ sở lý thuyết chính được trình bày như sau:

2.1. Danh sách (List) trong Python

Khái niệm: List là một cấu trúc dữ liệu cơ bản trong Python, cho phép lưu trữ một dãy các phần tử có thể thay đổi (mutable).

Ứng dụng trong chương trình: Danh sách được sử dụng để lưu trữ tất cả các liên hệ trong danh bạ. Mỗi phần tử trong list là một từ điển (dictionary) chứa thông tin một người dùng.

Ví dụ:

```
contacts = [  
    {"name": "Nguyen Van A", "phone": "0123456789", "email": "a@gmail.com"},  
    {"name": "Le Thi B", "phone": "0987654321", "email": "b@yahoo.com"}  
]
```

2.2. Từ điển (Dictionary) trong Python

Khái niệm: Dictionary là một cấu trúc dữ liệu ánh xạ khóa - giá trị (key-value), dùng để lưu trữ các cặp dữ liệu có liên kết.

Ứng dụng trong chương trình: Mỗi liên hệ được đại diện bằng một dictionary với các trường: name, phone, email.

2.3. Xử lý file JSON trong Python

Khái niệm: JSON (JavaScript Object Notation) là định dạng lưu trữ dữ liệu dạng văn bản, phổ biến trong các ứng dụng web, nó được dùng để lưu trữ và truyền dữ liệu trong các ứng dụng.

Thư viện sử dụng: json là thư viện tích hợp sẵn trong Python để thao tác với định dạng JSON.

Ứng dụng trong chương trình:

- Lưu danh bạ vào file contacts.json.
- Đọc lại dữ liệu khi khởi động chương trình.

2.4. Giao diện đồ hoạ người dùng (GUI) với Tkinter

Khái niệm: Tkinter là thư viện chuẩn trong Python dùng để xây dựng giao diện đồ hoạ (GUI).

Các thành phần chính:

- Tk(): Khởi tạo cửa sổ chính.
- Entry: Nhập dữ liệu.
- Button: Nút nhấn thực hiện chức năng.
- Treeview: Hiển thị bảng danh bạ.
- Label: Hiển thị tiêu đề hoặc thông báo.

Ứng dụng trong chương trình quản lý danh bạ GUI là tạo giao diện chính cho phép người dùng thêm, sửa, xoá và tìm kiếm thông tin liên hệ.

2.5. Xử lý sự kiện trong GUI

Khái niệm: Sự kiện là các hành động xảy ra do người dùng (như click chuột, nhập dữ liệu, chọn dòng).

Ứng dụng trong chương trình: Mỗi nút (Button) được gắn với một hàm xử lý. Khi người dùng nhấn nút, chương trình gọi hàm tương ứng để thực hiện chức năng như thêm, xoá, cập nhật.

2.6. Kiểm tra dữ liệu đầu vào (Validation)

Khái niệm: Validation là quá trình kiểm tra dữ liệu do người dùng nhập vào có hợp lệ hay không.

Ứng dụng trong chương trình:

- Kiểm tra các ô nhập liệu không được để trống.
- Kiểm tra định dạng email có đúng không.
- Kiểm tra số điện thoại chỉ chứa chữ số.

Tóm tắt chương:

Trong chương này trình bày các cơ sở lý thuyết được sử dụng trong quá trình xây dựng ứng dụng quản lý danh bạ giao diện đồ họa (GUI). Cụ thể, chương trình sử dụng danh sách (list) để lưu trữ nhiều liên hệ và từ điển (dictionary) để biểu diễn thông tin chi tiết của từng liên hệ như tên, số điện thoại và email. Dữ liệu được lưu trữ dưới định dạng JSON thông qua thư viện json, giúp chương trình dễ dàng ghi và đọc dữ liệu từ file contacts.json. Giao diện người dùng được thiết kế bằng thư viện Tkinter, bao gồm các thành phần như Entry (nhập liệu), Button (nút chức năng), Treeview (bảng hiển thị), và Label (nhãn mô tả), hỗ trợ người dùng tương tác trực quan. Bên cạnh đó, chương trình cũng xử lý các sự kiện GUI, gắn các hành động của người dùng với các hàm xử lý chức năng tương ứng như thêm, sửa, xóa và tìm kiếm. Cuối cùng, chương trình tích hợp kiểm tra tính hợp lệ của dữ liệu đầu vào (validation) để đảm bảo thông tin nhập vào đúng định dạng, không bị trống hoặc sai cú pháp, từ đó tăng độ ổn định và tin cậy của hệ thống

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Sơ đồ khối hệ thống

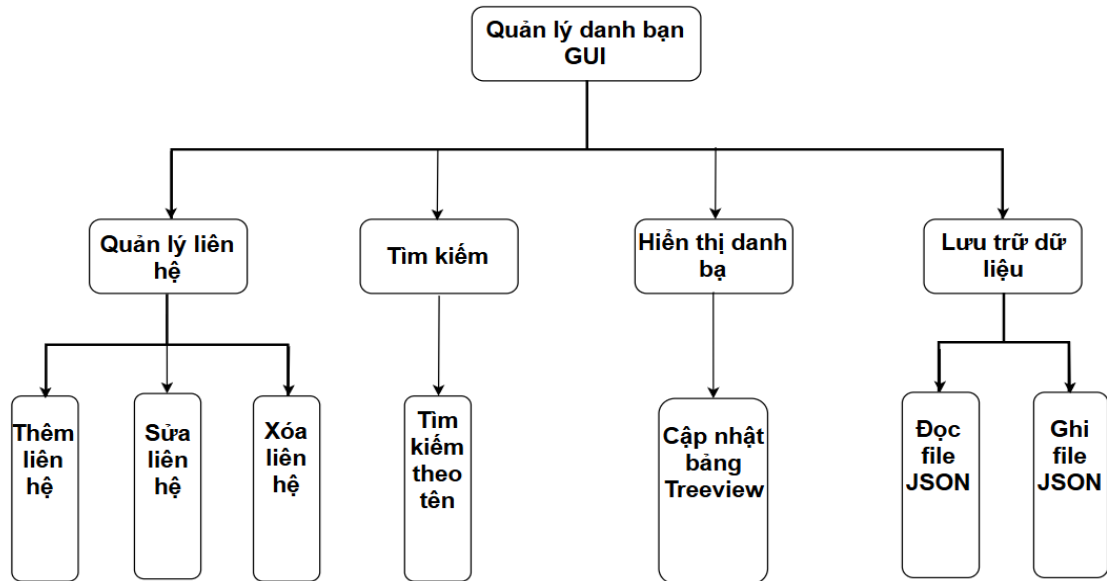
3.1.1. Các module chính trong chương trình ứng dụng quản lý danh bạ GUI

Chương trình quản lý danh bạ GUI được thiết kế gồm có 2 module chính là:

- Giao diện người dùng (GUI module) – main.py: module này sẽ có chức năng hiển thị giao diện, tiếp nhận và xử lý các thao tác của người dùng và hiển thị kết quả ra màn hình bao gồm có các Entry nhập liệu, nút Button để thực hiện các chức năng thêm, sửa, xóa và bảng Treeview để hiển thị danh bạ
- Xử lý dữ liệu (Logic module) – contacts.py: module này sẽ thực hiện các chức năng xử lý nghiệp vụ như khi người dùng sử dụng chức năng thêm, sửa, xóa, tìm kiếm, lưu và đọc file JSON và quản lý các liên hệ được lưu trong bộ nhớ và file

3.1.2. Biểu đồ phân cấp chức năng

Biểu đồ phân cấp chức năng này sẽ thể hiện được các chức năng chính của chương trình và thông qua đó sẽ hiểu được các chức năng của chương trình sẽ làm được gì, dưới đây là biểu đồ phân cấp chức năng của chương trình:

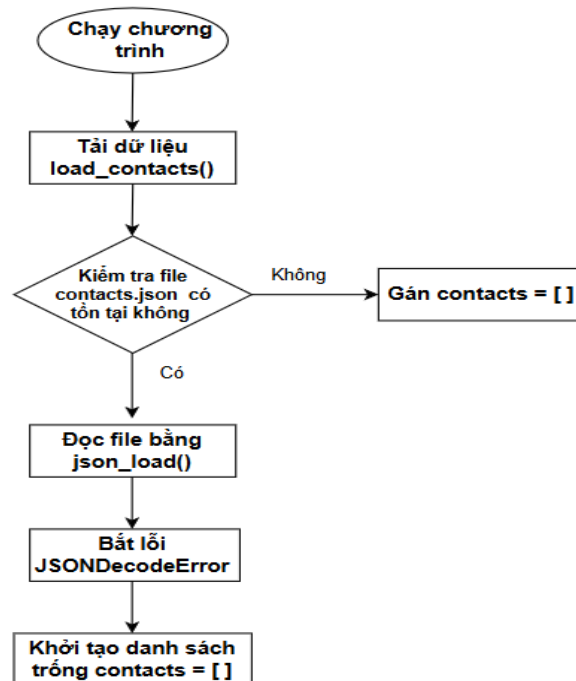


Hình 3.1. Biểu đồ chức năng của ứng dụng quản lý danh bạ GUI

3.2. Sơ đồ khối các thuật toán chính

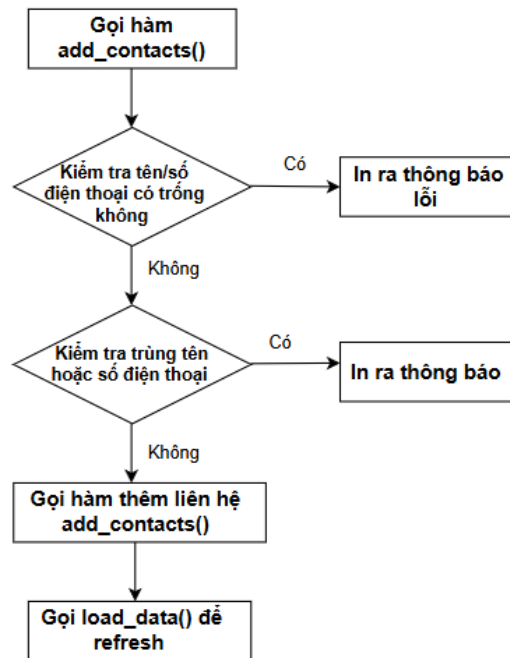
3.2.1. Sơ đồ khối các thuật toán

a. Sơ đồ khối thuật toán đọc/ghi file json



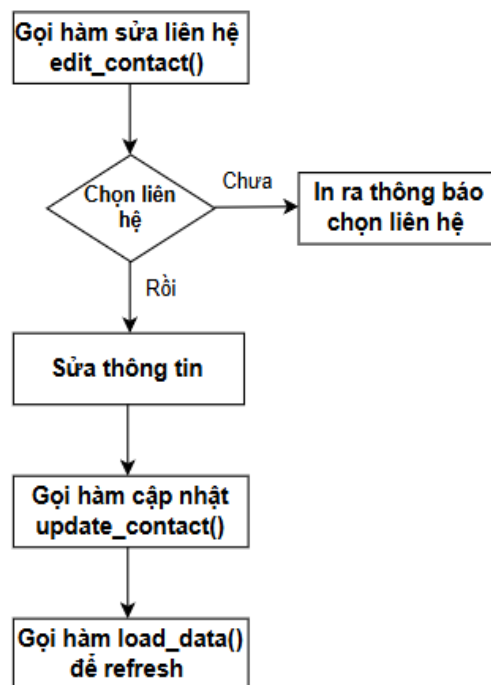
Hình 3.2. Sơ đồ khối thuật toán đọc/ghi file json

b. Sơ đồ khối thuật toán thêm liên hệ



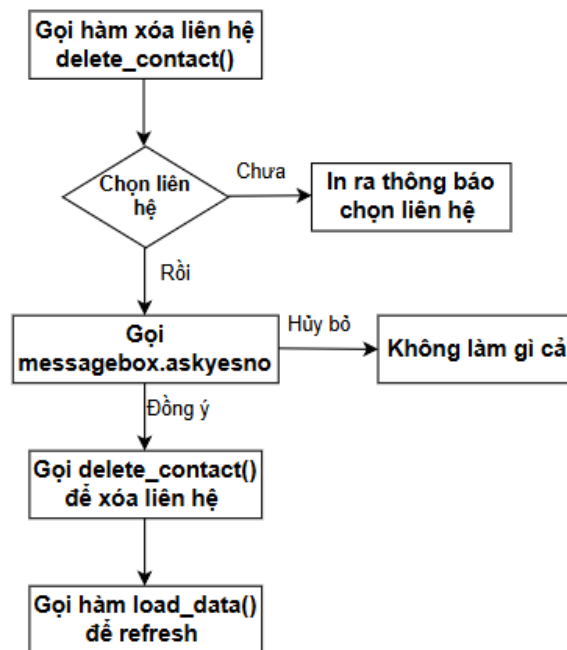
Hình 3.3. Sơ đồ khối thuật toán thêm liên hệ

c. Sơ đồ khối thuật toán sửa liên hệ



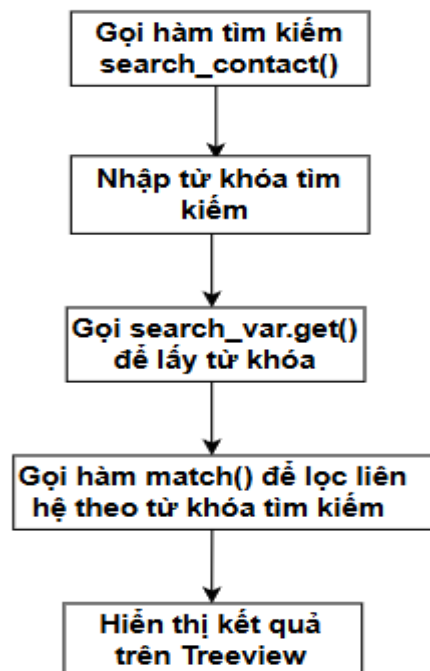
Hình 3.4. Sơ đồ khối thuật toán sửa liên hệ

d. Sơ đồ khối thuật toán xóa liên hệ



Hình 3.5. Sơ đồ khối thuật toán xóa liên hệ

e. Sơ đồ khối thuật toán tìm kiếm liên hệ



Hình 3.6. Sơ đồ khối thuật toán tìm kiếm

3.2.2. Phân tích các khối thuật toán

a. Thuật toán đọc dữ liệu từ file json

- Hàm load_contacts(): được gọi khi khởi động chương trình để nạp danh sách liên hệ từ file JSON
 - Kiểm tra file contacts.json xem có tồn tại chưa để tránh lỗi nếu file chưa tồn tại thì khởi tạo file mới
 - Đọc file bằng json.load(): Đọc danh sách liên hệ từ file JSON
 - Bắt lỗi JSONDecodeError: để kiểm tra nếu file JSON bị hỏng sẽ không làm sập chương trình và tự động khởi tạo danh sách rỗng contacts = []
- ➔ **Luồng xử lý:** Kiểm tra file ➔ Đọc file JSON ➔ Bắt lỗi decode ➔ Gán danh sách rỗng nếu lỗi

b. Thuật toán thêm liên hệ

- Hàm add_contacts(): Hàm này sẽ được gọi khi người dùng nhấn nút thêm liên hệ với validation dữ liệu đầu vào
 - Kiểm tra trống (tên hoặc số điện thoại) để đảm bảo dữ liệu đầu vào không bị thiếu
 - Kiểm tra trùng tên hoặc số điện thoại để tránh thêm liên hệ bị trùng lặp
 - add_contacts(): Gọi đến lớp xử lý dữ liệu để thêm vào danh sách và ghi lại file JSON
 - Hàm load_data() được gọi để cập nhật lại Treeview sau khi thêm để phản ánh dữ liệu mới đã được thêm
- ➔ **Luồng xử lý:** Validate ➔ Kiểm tra trùng lặp ➔ Thêm liên hệ ➔ Lưu liên hệ ➔ Refresh UI

c. Thuật toán sửa liên hệ

- Hàm edit_contacts() được gọi khi người dùng chọn một liên hệ và bấm nút sửa
- Kiểm tra có chọn liên hệ chưa để tránh khi chưa có mục nào được chọn
- Cập nhật lại thông tin từ Entry: các thông tin đã sửa được ghi lại
- Hàm update_contact(): Ghi lại dữ liệu vào danh sách và file JSON
- Gọi hàm load_data() để refresh lại danh sách Treeview

➔ **Luồng xử lý:** Kiểm tra `selected_index` → Sửa lại thông tin → Lưu → Refresh UI

d. Thuật toán xóa liên hệ

- Hàm `delete_contact()`: hàm này được gọi khi người dùng bấm nút xoát
 - Kiểm tra đã chọn liên hệ hay chưa để tránh xóa nhầm liên hệ hoặc khi chưa lựa chọn liên hệ
 - Gọi hộp thoại xác nhận `messagebox.askyesno` để đảm bảo người dùng có thật sự muốn xóa liên hệ không
 - Gọi hàm xóa liên hệ `delete_contact()` từ danh sách để thực hiện thao tác xóa và lưu lại vào JSON
 - Gọi `load_data()` để cập nhật lại giao diện hiển thị danh sách sau khi xóa liên hệ
- ➔ **Luồng xử lý:** Kiểm tra `selected_index` → Xác nhận xóa → Xóa liên hệ → Lưu → Refresh UI

e. Thuật toán tìm kiếm liên hệ

- hàm `search_contact()`: Gọi khi người dùng nhập từ khóa vào ô tìm kiếm và bấm nút tìm kiếm
 - Dữ liệu sẽ được lấy từ ô tìm kiếm `search_var.get()`: dữ liệu người dùng nhập vào
 - Gọi hàm `match()` để lọc danh sách liên hệ theo từ khóa
 - Hiển thị danh sách kết quả trên Treeview
- ➔ **Luồng xử lý:** Nhập từ khóa tìm kiếm → Tìm → Lọc tên bắt đầu từ từ khóa tìm kiếm → Hiển thị kết quả ra Treeview

3.2.3. Quan hệ đầu vào/ra giữa các khối

a. Quan hệ đầu vào/ra giữa các khối thuật toán đọc dữ liệu file JSON

- Đầu vào: File `contacts.json`
- Xử lý:
 - Kiểm tra file `contact.json` có tồn tại không
 - Đọc và phân tích nội dung file JSON

- Bắt lỗi nếu file rỗng hoặc hỏng
- Đầu ra: Trả về danh sách liên hệ dạng list hoặc gán contacts = [] nếu bị lỗi. Dữ liệu này sẽ dùng để hiển thị giao diện cho Treeview

b. Quan hệ đầu vào/ra giữa các khối thuật toán thêm liên hệ

- Đầu vào: Dữ liệu thông tin liên hệ nhập từ người dùng (Tên, SĐT, Email)
- Xử lý: Hệ thống sẽ kiểm tra xem dữ liệu bị rỗng hay trùng không, sau đó sẽ gọi add_contacts() để thêm và lưu file
- Đầu ra: File contacts.json và giao diện Treeview đã được cập nhật và hiển thị liên hệ

c. Quan hệ đầu vào/ra giữa các khối thuật toán sửa liên hệ

- Đầu vào: liên hệ được chọn từ Treeview và thông tin sửa được nhập từ người dùng
- Xử lý: Kiểm tra chọn dòng, gọi update_contact() để cập nhật file
- Đầu ra: file contacts.json đã được cập nhật và hiển thị lại dữ liệu ra Treeview

d. Quan hệ đầu vào/ra giữa các khối thuật toán xóa liên hệ

- Đầu vào: Liên hệ được chọn từ Treeview
- Xử lý: Kiểm tra chọn dòng, xác nhận xóa, gọi delete_contact() để xóa liên hệ khỏi file JSON
- Đầu ra: File contacts.json và Treeview được cập nhật sau khi xóa liên hệ

e. Quan hệ đầu vào/ra giữa các khối thuật toán tìm kiếm liên hệ

- Đầu vào: Từ khóa tìm kiếm được người dùng nhập vào
- Xử lý: Gọi hàm match() để lọc các liên hệ phù hợp
- Đầu ra: các liên hệ được lọc và hiển thị ra Treeview

3.3. Cấu trúc dữ liệu

3.3.1. Cấu trúc dữ liệu đầu vào

Dữ liệu danh bạ được lưu trữ dưới dạng danh sách liên hệ (list of dictionary), mỗi liên hệ bao gồm các trường sau:

- Name: họ tên của người liên hệ
- Phone: số điện thoại liên hệ

Định dạng tệp: dữ liệu được lưu và đọc dưới dạng JSON (contacts.json) bằng thư viện json trong python

Ví dụ cấu trúc dữ liệu đầu vào:

```
[  
  {  
    "name": "Nguyen Van A",  
    "phone": "0909123456"  
  },  
  {  
    "name": "Tran Thi B",  
    "phone": "0988123456"  
  }  
]
```

3.3.2. Cấu trúc dữ liệu sau tiền xử lý

- Dữ liệu sau khi đọc từ file JSON sẽ được xử lý và kiểm tra trước khi hiển thị:

- Kiểm tra định dạng số điện thoại
- Loại bỏ dòng trống hoặc giá trị thiếu
- Tránh trùng lặp số điện thoại khi thêm

- Các trường dữ liệu sau khi xử lý gồm:

- Name: chuỗi ký tự không rỗng
- Phone: chuỗi số điện thoại hợp lệ không trùng lặp

3.3.3. Cấu trúc dữ liệu đầu ra

- Dữ liệu sau khi thêm, sửa, xóa sẽ được ghi lại vào file JSON (contacts.json) theo định dạng danh sách đối tượng (list of dictionaries)

Ví dụ:

```
[  
  {  
    "name": "Pham Van C",  
    "phone": "0911222333"  
  }  
]
```

- Dữ liệu cũng được xuất trực tiếp trên Treeview GUI để người dùng dễ dàng theo dõi. Và các thao tác như tìm kiếm không làm thay đổi dữ liệu gốc mà chỉ làm thay đổi phần hiển thị

3.4. Các hàm chính trong chương trình chính

Ứng dụng quản lý dạng bạ bằng GUI được chia thành hai phần chính là module xử lý dữ liệu – file contacts.py để quản lý danh sách liên hệ, đọc/ghi file JSON và giao diện người dùng GUI – file chính main.py sử dụng thư viện Tkinter để tạo form nhập liệu, bảng hiển thị và các nút chức năng

3.4.1. Các hàm chính trong file contacts.py

- Hàm *load_contacts(filename)*: Hàm này có chức năng đọc dữ liệu từ file JSON vào chương trình với tham số là tên file JSON và trả về danh sách liên hệ (list of dict), nếu file JSON lỗi hoặc không tồn tại thì sẽ trả về danh sách rỗng
- Hàm *thêm liên hệ add_contact()*: Hàm này có chức năng thêm một liên hệ mới vào danh sách, khi thêm nó sẽ kiểm tra xem liên hệ mới có bị trùng tên hoặc số điện thoại hay không sau đó liên hệ mới sẽ được cập nhật ghi lại vào file
- Hàm *sửa liên hệ edit_contact()*: Hàm sửa liên hệ có chức năng sửa thông tin liên

hệ tại chỉ số index, sau đó kiểm tra dữ liệu có hợp lệ không, nếu có thì thông tin sẽ được cập nhật lại vào danh sách và file contacts.json

- *Hàm xóa liên hệ delete_contact()*: Hàm xóa liên hệ có chức năng xóa liên hệ tại vị trí index khỏi danh sách và sau đó cập nhật lại file và hiển thị ra Treeview
- *Hàm tìm kiếm liên hệ search_contacts()*: Hàm tìm kiếm có chức năng tìm kiếm các liên hệ trong danh sách và trả về các liên hệ có chứa từ khóa tìm kiếm và hàm này không phân biệt chữ hoa hay chữ thường

3.4.2. Các hàm chính trong file main.py

- *Hàm load_data_to_Treeview()*: hàm này có chức năng hiển thị danh sách các liên hệ lên bảng Treeview và được gọi sau mỗi lần cập nhật danh sách hoặc tìm kiếm
- *Hàm on_add_click()*: có chức năng lấy dữ liệu từ ô nhập và gọi add_contact() sau đó sẽ hiển thị lại danh sách và ghi kết quả vào file JSON
- *Hàm on_edit_click()*: có chức năng lấy dòng đang chọn trong Treeview sau đó người dùng sẽ sửa nội dung thông tin và cập nhật lại file
- *Hàm on_delete_click()*: có chức năng xóa dòng đang chọn trong Treeview và danh sách sau đó cập nhật lại file và giao diện hiển thị trên Treeview
- *Hàm on_search_click()*: có chức năng lọc danh sách theo tên được nhập vào trong ô tìm kiếm và hiển thị danh sách liên hệ khớp với tên tìm kiếm trong Treeview

Tóm tắt chương:

Trong chương này trình bày quá trình thiết kế và xây dựng chương trình quản lý danh bạ với giao diện đồ họa người dùng (GUI). Ứng dụng được chia thành hai module chính: module giao diện (main.py) đảm nhận việc hiển thị, tiếp nhận và xử lý thao tác từ người dùng; và module xử lý logic (contacts.py) phụ trách các chức năng như thêm, sửa, xóa, tìm kiếm, lưu và đọc dữ liệu từ file JSON. Chương này cũng mô tả các biểu đồ phân cấp chức năng và sơ đồ khối thuật toán nhằm minh họa quy trình xử lý cho từng chức năng chính như đọc/ghi file, thêm,

sửa, xóa và tìm kiếm liên hệ. Bên cạnh đó, chương phân tích chi tiết luồng xử lý và quan hệ đầu vào/ra giữa các khối thuật toán, đảm bảo tính logic và sự liên kết giữa giao diện và dữ liệu. Dữ liệu được tổ chức dưới dạng danh sách các từ điển, mỗi từ điển là một liên hệ gồm các trường như tên và số điện thoại, được lưu trữ và xử lý bằng định dạng JSON. Cuối chương là phần giới thiệu các hàm chính trong hai file mã nguồn: `contacts.py` chứa các hàm xử lý dữ liệu, và `main.py` đảm nhiệm hiển thị giao diện và kết nối các thao tác người dùng với các chức năng tương ứng. Nhờ đó, chương trình hoạt động trơn tru, dễ sử dụng và có khả năng xử lý hiệu quả các tác vụ quản lý danh bạ cơ bản

CHƯƠNG 4. KIỂM THỬ VÀ ĐÁNH GIÁ HỆ THỐNG

4.1. Kết quả kiểm thử

Trong quá trình phát triển ứng dụng quản lý danh bạ với giao diện người dùng GUI, em đã tiến hành kiểm thử toàn bộ các chức năng chính của hệ thống như thêm, sửa, xóa, tìm kiếm. Tất cả các tính năng chính này đều hoạt động ổn định và cho ra kết quả đúng như mong đợi, giao diện người dùng thì thân thiện, dễ sử dụng và hiển thị thông tin rõ ràng, không phát hiện lỗi trong quá trình kiểm thử

Dưới đây là kết quả kiểm thử sau khi chạy chương trình và có các liên hệ đã được thêm vào danh bạ

Hình 4.1. Giao diện chính của ứng dụng

4.2. Kết luận

4.2.1. Kết quả sản phẩm đã đạt được

Sau quá trình tìm hiểu và triển khai, ứng dụng Quản lý danh bạ GUI đã được xây dựng thành công với các chức năng cơ bản thêm, sửa, xóa, tìm kiếm liên hệ

và lưu trữ dữ liệu vào file json. Giao diện người dùng thân thiện và trực quan với các thành phần như Entry, button và Treeview giúp người dùng thao tác dễ dàng và hiệu quả. Dữ liệu được lưu trữ và đồng bộ hóa kịp thời sau mỗi lần hành động của người dùng, hệ thống cũng xử lý tốt các lỗi đầu vào, lỗi định dạng và đảm bảo an toàn dữ liệu trong quá trình xử dụng.

4.2.2. Kiến thức đã học được trong quá trình làm đề tài

Trong quá trình thực hiện đề tài này, em đã củng cố và nâng cao kiến thức về lập trình python đặc biệt là cách làm việc với file json và xử lý dữ liệu theo cấu trúc list – directory. Ngoài ra, em còn học được cách xây dựng giao diện đồ họa người dùng (GUI) bằng thư viện Tkinter, quản lý sự kiện và tổ chức mã nguồn theo hướng module để dễ dàng bảo trì. Bên cạnh đó kỹ năng tư duy thuật toán, thiết kế cấu trúc dữ liệu và khả năng kiểm soát luồng xử lý cũng được cải thiện. Những kinh nghiệm này cũng giúp cho em có một nền tảng quan trọng cho các môn học lập trình sau này

Tóm tắt chương:

Chương 4 trình bày kết quả kiểm thử và đánh giá hệ thống. Tất cả chức năng chính như thêm, sửa, xóa, tìm kiếm liên hệ đều hoạt động ổn định, giao diện thân thiện và dữ liệu được lưu trữ an toàn. Qua quá trình thực hiện, em đã nâng cao kỹ năng lập trình Python, xử lý file JSON và xây dựng giao diện GUI bằng Tkinter.

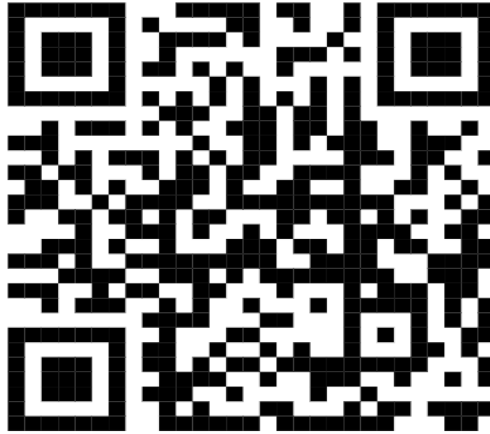
TÀI LIỆU THAM KHẢO

1. Python Programming for the Absolute Beginner- 3rd Edition.pdf
2. W3Schools. *Python File Handling*.
https://www.w3schools.com/python/python_file_handling.asp
3. Real Python. *Python GUI Programming With Tkinter*.
<https://realpython.com/python-gui-tkinter/>
4. Stack Overflow. *Python Tkinter Examples and Discussions*.
<https://stackoverflow.com/>

LINK VIDEO TRÊN YOUTUBE VÀ GITHUB

1. Link video: <https://www.youtube.com/watch?v=iWQoMGKgLvI>

- Mã QR video:



2. Link github: https://github.com/ThuhaTNUT/BTL_Python_UDQLDBGUI.git

- Mã QR github:

