**TRƯ**

**ỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

**KHOA ĐI**

**ỆN TỬ**

**BÀI T**

**ẬP LỚN**

**Môn:Python**

**NGÀNH**

**:**

**K**

**Ỹ THUẬT MÁY TÍNH**

**H**

**Ệ**

**:**

**Đ**

**ẠI HỌC CHÍNH QUY**

**THÁI NGUYÊN**

**–**

**2025**



*Biểu mẫu BTL01*

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: Lê Ngọc Tú

Lớp: K58KTP

Giáo viên GIẢNG DẠY: TS.Nguyễn Văn Huy

Link GitHub : https://github.com/LNtu2004/BTL-python

**Thái Nguyên – 2025**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN**  *Biểu mẫu BTL02* | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***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:* *Lê Ngọc Tú*

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

*Giáo viên hướng dẫn:* Nguyễn Văn Huy

*Ngày giao đề : 20/05/2025 Ngày hoàn thành* : 8/06/2025

*Tên đề tài :* ***Ứng dụng quản lý danh bạ GUI***

*Yêu cầu :*

|  |
| --- |
| **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)*

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 6](#_Toc200446779)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 6](#_Toc200446780)

[CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH 7](#_Toc200446781)

[CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN 8](#_Toc200446782)

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

* Trong thời đại số hóa hiện nay, việc quản lý thông tin cá nhân một cách hiệu quả đang trở nên ngày càng quan trọng. Từ cá nhân đến doanh nghiệp, việc lưu trữ và tra cứu danh bạ liên hệ một cách nhanh chóng, trực quan có thể tiết kiệm rất nhiều thời gian và công sức. Đề tài "Ứng dụng Quản lý Danh bạ GUI" được lựa chọn nhằm xây dựng một phần mềm đơn giản nhưng đầy đủ chức năng cơ bản như thêm, sửa, xoá và tìm kiếm liên hệ, với dữ liệu được lưu trữ trong tệp JSON.
* Ứng dụng được xây dựng bằng ngôn ngữ Python, sử dụng thư viện Tkinter để tạo giao diện đồ họa và module json để xử lý lưu trữ dữ liệu. Đây không chỉ là một bài tập mang tính học thuật mà còn có giá trị thực tiễn cao, giúp người học rèn luyện kỹ năng lập trình hướng đối tượng, làm quen với giao diện người dùng và kỹ thuật lưu trữ dữ liệu bền vững.
* Thách thức chính của đề tài nằm ở việc đồng bộ hóa giữa giao diện và dữ liệu nền, đảm bảo tính ổn định khi thao tác, đồng thời nâng cao trải nghiệm người dùng thông qua giao diện trực quan và dễ sử dụng.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

**2.1. Giao diện đồ họa (Tkinter)**

Tkinter là thư viện giao diện mặc định trong Python, cung cấp các thành phần GUI như Label, Entry, Button, Treeview... Với ưu điểm dễ học và tích hợp sẵn, Tkinter là lựa chọn lý tưởng để xây dựng các ứng dụng nhỏ đến trung bình. Trong đề tài này, Tkinter được dùng để tạo form nhập liệu và bảng hiển thị danh bạ.

**2.2. Quản lý dữ liệu với JSON**

JSON (JavaScript Object Notation) là định dạng lưu trữ dữ liệu dạng văn bản nhẹ, dễ đọc và ghi. Python hỗ trợ sẵn module `json` để chuyển đổi giữa kiểu dữ liệu Python (list, dict) và chuỗi JSON. Trong bài này, JSON giúp lưu trữ danh bạ ở dạng tệp tin để có thể truy xuất lại khi mở chương trình.

**2.3. Treeview – Hiển thị bảng dữ liệu**

Treeview là widget đặc biệt trong Tkinter, cho phép hiển thị dữ liệu dạng bảng nhiều cột. Đây là công cụ lý tưởng để trình bày danh sách liên hệ một cách trực quan và dễ thao tác. Người dùng có thể chọn dòng để sửa, xóa, hoặc tìm kiếm nhanh thông tin.

**2.4. Lập trình hướng đối tượng**

Toàn bộ logic quản lý danh bạ được gói gọn trong class `ContactManager`, đảm nhiệm việc nạp, lưu, cập nhật và tìm kiếm dữ liệu. Việc tách riêng phần xử lý dữ liệu và phần giao diện giúp chương trình rõ ràng, dễ bảo trì và mở rộng về sau.

# CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

**3.1. Kiến trúc tổng thể**

Ứng dụng được chia thành hai phần chính:

- `contacts.py`: xử lý logic quản lý danh bạ (class `ContactManager`)

- `app.py`: tạo giao diện, xử lý tương tác người dùng (class `ContactApp`)

**3.2. Biểu đồ phân cấp chức năng**

- Nhập liệu (Tên, SĐT, Email)

- Thêm liên hệ

- Sửa liên hệ

- Xoá liên hệ

- Tìm kiếm theo tên

- Lưu & nạp dữ liệu từ file JSON

**3.3. Giao diện người dùng**

Giao diện sử dụng các thành phần Tkinter:

- `Entry`: nhập tên, số điện thoại, email

- `Button`: thao tác thêm, sửa, xoá, tìm

- `Treeview`: hiển thị danh sách liên hệ dạng bảng

**3.4. Cấu trúc dữ liệu**

Dữ liệu được lưu dưới dạng JSON như sau:

[

{"name": "Nguyen Van A", "phone": "0123456789", "email": "a@gmail.com"},

{"name": "Le Thi B", "phone": "0988123456", "email": "b@yahoo.com"}

]

**3.5. Các hàm chính**

- `load\_contacts()`: đọc dữ liệu từ file khi khởi động

- `save\_contacts()`: ghi dữ liệu vào file mỗi khi thao tác

- `add\_contact()`: thêm mới liên hệ

- `update\_contact()`: cập nhật liên hệ được chọn

- `delete\_contact()`: xoá liên hệ được chọn

- `search\_contacts()`: lọc theo tên người dung

# CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN

**4.1. Thực nghiệm**

a) Kiểm thử chức năng

| STT | Tác vụ | Dữ liệu đầu vào | Kết quả mong đợi |

|-----|------------------|--------------------------------|-----------------------------------|

| 1 | Thêm liên hệ | Nguyen, 0123, a@gmail.com | Hiển thị trên bảng, lưu JSON |

| 2 | Sửa số điện thoại| Đổi 0123 thành 0999 | JSON được cập nhật |

| 3 | Xoá liên hệ | Chọn dòng và xoá | Dòng biến mất, JSON cập nhật lại |

| 4 | Tìm kiếm | Nhập "Nguyen" | Hiển thị đúng kết quả tìm được

b) Nhận xét

- Chương trình hoạt động đúng chức năng.

- File JSON được cập nhật chính xác sau mỗi thao tác.

- Giao diện đơn giản, dễ sử dụng.

- Việc tách riêng giao diện và logic xử lý giúp dễ mở rộng sau này.

**4.2. Kết luận**

Qua quá trình xây dựng chương trình, em đã nắm vững kỹ năng tổ chức mã nguồn theo hướng đối tượng, sử dụng Tkinter để xây dựng giao diện, đồng thời thành thạo việc lưu trữ dữ liệu bằng JSON. Đề tài không quá lớn nhưng đòi hỏi sự cẩn thận trong xử lý logic và tương tác người dùng.

Nếu được phát triển thêm, chương trình có thể tích hợp các tính năng nâng cao như:

- Xuất danh bạ ra Excel hoặc CSV

- Gửi email trực tiếp từ ứng dụng

- Thêm ảnh đại diện cho từng liên hệ

- Lưu trữ dữ liệu online (Firebase, SQLite...)

Em cảm thấy đề tài này vừa sức, vừa thực tiễn và cực kỳ hữu ích để ứng dụng vào công việc thật.