**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**🙟🕮🙝**

**A logo of hands holding a book and a flame

Description automatically generated**

**TIỂU LUẬN KẾT THÚC HỌC PHẦN**

**ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**Mã học phần: 241PROJ215879**

**TÊN ĐỒ ÁN**

**ỨNG DỤNG HỆ THỐNG TỐI ƯU ĐƯỜNG DẪN NƯỚC**

**Nhóm sinh viên thực hiện: Nhóm 8**

|  |  |
| --- | --- |
| 1. **Võ Hoàng Anh** | **22110104** |
| 1. **Nguyễn Hiếu Nghĩa** | **22110189** |
| 1. **Trần Như Thiện** | **22110235** |

|  |
| --- |
| **Tp.HCM, tháng 11 năm 2024** |

**Giảng viên hướng dẫn: TS.Trần Nhật Quang**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**🙟🕮🙝**

**A logo of hands holding a book and a flame

Description automatically generated**

**TIỂU LUẬN KẾT THÚC HỌC PHẦN**

**ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**Mã học phần: 241PROJ215879**

**TÊN ĐỒ ÁN**

**ỨNG DỤNG HỆ THỐNG TỐI ƯU ĐƯỜNG DẪN NƯỚC**

**Nhóm sinh viên thực hiện: Nhóm 8**

|  |  |
| --- | --- |
| 1. **Võ Hoàng Anh** | **22110104** |
| 1. **Nguyễn Hiếu Nghĩa** | **22110189** |
| 1. **Trần Như Thiện** | **22110235** |

**Giảng viên hướng dẫn: TS. Trần Nhật Quang**

|  |
| --- |
| **Tp.HCM, tháng 11 năm 2024** |

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Ngày……….Tháng………Năm……

**Giảng viên hướng dẫn**

(Ký và ghi rõ họ tên)

**LỜI CẢM ƠN**

Chúng em xin bày tỏ lòng biết ơn sâu sắc đến Thầy Trần Nhật Quang–Giảng viên bộ môn Đồ án Công Nghệ Thông Tin, Trường Đại học Sư Phạm Kỹ Thuật Thành phố Hồ Chí Minh, đã dành thời gian, truyền đạt kiến thức và tâm huyết trong việc hướng dẫn và hỗ trợ chúng em trong quá trình thực hiện đề tài "Ứng dụng hệ thống tối ưu đường dẫn nước."

Thầy Quang đã đóng góp không ngừng sức lực giảng dạy và kiến thức chuyên sâu của mình để giúp chúng em ứng dụng kiến thức của môn “Đồ án Công nghệ thông tin” trong việc xây dựng đồ án.

Sự tận tâm của Thầy đã là nguồn động viên lớn, tạo điều kiện thuận lợi cho quá trình nghiên cứu và phát triển của chúng em.

Chúng em chân thành biết ơn tới Thầy Trần Nhật Quang, người đã là nguồn động viên lớn giúp chúng em vượt qua những thách thức và đạt được kết quả như mong đợi trong đề tài này. Lời cảm ơn này không đủ diễn đạt lòng biết ơn của chúng em, nhưng chúng em hy vọng nó là sự tri ân chân thành của chúng em đối với công lao của Thầy.

**Chúng em xin chân thành cảm ơn!**

**MỤC LỤC**

[**A. MỞ ĐẦU** 6](#_Toc183648593)

[1.Lý do chọn đề tài 6](#_Toc183648594)

[2. Phương pháp nghiên cứu 6](#_Toc183648595)

[**B. NỘI DUNG** 7](#_Toc183648596)

[CHƯƠNG 1: ĐẶC TẢ 7](#_Toc183648597)

[1. Phần mềm dùng làm gì 7](#_Toc183648598)

[2. Dữ liệu, thông tin đầu vào 7](#_Toc183648599)

[3. Các tình huống sử dụng (Use Cases) 7](#_Toc183648600)

[4. Giao diện dự kiến 8](#_Toc183648601)

[CHƯƠNG 2: PHÂN CÔNG NHIỆM VỤ 10](#_Toc183648602)

[CHƯƠNG 3: THIẾT KẾ 11](#_Toc183648603)

[1. Thuật toán 11](#_Toc183648604)

[2. Thiết kế lớp 12](#_Toc183648605)

[3. Thiết kế dữ liệu. 29](#_Toc183648606)

[4. Thiết kế giao diện. 30](#_Toc183648607)

[5. Chức năng chính của hệ thống 32](#_Toc183648608)

[CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ 33](#_Toc183648609)

[1. Các thư viện cần phải cài đặt 33](#_Toc183648610)

[2 Kiểm thử 35](#_Toc183648611)

[CHƯƠNG 5: KẾT LUẬN 40](#_Toc183648612)

**DANH MỤC HÌNH ẢNH**

[Hình 1.Giao diện Map View 30](#_Toc183611870)

[Hình 2.Giao diện Data Input 31](#_Toc183611871)

[Hình 3.Giao diện Result View 31](#_Toc183611872)

[Hình 4.Giao diện Input của Find Shortest Path 31](#_Toc183611873)

[Hình 5.Giao diên Output của Find Shortest Path 32](#_Toc183611874)

[Hình 6.Giao diện Generate Report 32](#_Toc183611875)

**DANH MỤC CÁC BẢNG**

[Table 1.Control Panel 12](#_Toc183648575)

[Table 2.Phương thức trong Control Panel 14](#_Toc183648576)

[Table 3.Data Input 14](#_Toc183648577)

[Table 4.Phương thức trong Data Input 16](#_Toc183648578)

[Table 5.Map View 16](#_Toc183648579)

[Table 6.Phương thức trong Map View 20](#_Toc183648580)

[Table 7.Result View 21](#_Toc183648581)

[Table 8.Phương thức trong Result View 22](#_Toc183648582)

[Table 9.Cost Calculator 23](#_Toc183648583)

[Table 10.Phương thức trong Cost Calculator 24](#_Toc183648584)

[Table 11.Path Finding 24](#_Toc183648585)

[Table 12.Phương thức trong Path Finding 27](#_Toc183648586)

[Table 13.Report Generator 28](#_Toc183648587)

[Table 14.Phương thức trong Report Generator 29](#_Toc183648588)

[Table 15.Thiết kế file connections.json 30](#_Toc183648589)

[Table 16.Thiết kế file coordinates.json 30](#_Toc183648590)

[Table 17.Giao diện 32](#_Toc183648591)

[Table 18.Kiểm thử 40](#_Toc183648592)

# **A. MỞ ĐẦU**

### **1.Lý do chọn đề tài**

Dự án "Ứng dụng hệ thống tối ưu đường dẫn nước" được chọn nhằm giải quyết vấn đề tối ưu hóa mạng lưới cấp nước trong các khu vực đô thị, nơi mà việc quản lý hạ tầng nước sạch có ảnh hưởng trực tiếp đến chi phí vận hành và bảo trì. Việc tối ưu hóa các tuyến ống dẫn nước từ các trạm cấp nước đến các điểm tiêu thụ không chỉ giúp giảm thiểu chi phí xây dựng và bảo trì mà còn nâng cao hiệu quả cung cấp nước cho cộng đồng. Với mục tiêu cung cấp một công cụ hỗ trợ cho các kỹ sư và nhân viên quản lý, hệ thống này sẽ sử dụng các thuật toán tìm đường ngắn nhất, phân tích chi phí và tình trạng kết nối, giúp đưa ra các quyết định tối ưu về thiết kế và quản lý mạng lưới cấp nước. Hơn nữa, việc tích hợp bản đồ và giao diện dễ sử dụng thông qua Streamlit và Folium sẽ giúp người dùng dễ dàng theo dõi, quản lý và cập nhật thông tin mạng lưới cấp nước, từ đó cải thiện chất lượng dịch vụ và tiết kiệm chi phí cho các thành phố hoặc khu vực sử dụng.

### **2. Phương pháp nghiên cứu**

- Phương pháp nghiên cứu lý thuyết.

- Phương pháp nghiên cứu ứng dụng trí tuệ nhân tạo (AI).

- Phương pháp thử nghiệm và mô phỏng.

# **B. NỘI DUNG**

## **CHƯƠNG 1: ĐẶC TẢ**

### **Phần mềm dùng làm gì**

Phần mềm này là một công cụ hỗ trợ quản lý và tối ưu hóa mạng lưới cấp nước cho một khu vực hoặc thành phố. Người dùng chính là các kỹ sư hoặc nhân viên quản lý hạ tầng cấp nước, nhằm tìm đường đi ngắn nhất giữa các trạm cấp nước và các điểm sử dụng (hộ gia đình, doanh nghiệp, tòa nhà công cộng), qua đó giảm chi phí xây dựng và bảo trì hệ thống.

**Ngữ cảnh sử dụng**: Phần mềm được sử dụng để lập kế hoạch, thiết kế, và tối ưu hóa mạng lưới cấp nước, đảm bảo cung cấp nước một cách hiệu quả và tiết kiệm.

### **Dữ liệu, thông tin đầu vào**

1. **Vị trí địa lý**: Tọa độ của các trạm cấp nước và các điểm tiêu thụ nước.
2. **Chi phí xây dựng**: Chi phí dự tính cho mỗi đoạn ống giữa các điểm (bao gồm vật liệu, thi công).
3. **Tình trạng kết nối**: Các đoạn kết nối hiện có và trạng thái của chúng (hoạt động, bảo trì, hỏng hóc).

### **Các tình huống sử dụng (Use Cases)**

1. **Tìm đường đi ngắn nhất giữa các trạm**:

* **Mục đích**: Xác định lộ trình có giá thành thấp nhất giữa một trạm cấp nước và các điểm tiêu thụ, giúp tối ưu hóa chi phí.
* **Tính năng**: Nhập vị trí các trạm và điểm tiêu thụ, phần mềm sẽ tự động tính toán và hiển thị lộ trình tối ưu.

1. **Xây dựng và quản lý kết nối mạng lưới**:

* **Mục đích**: Cho phép người dùng tạo và chỉnh sửa các kết nối giữa các điểm.
* **Tính năng**: Thêm, xóa, chỉnh sửa các đoạn ống dẫn, nhập chi phí xây dựng cho mỗi đoạn, cập nhật tình trạng kết nối.

1. **Phân tích chi phí và tối ưu hóa xây dựng**:

* **Mục đích**: Ước tính chi phí xây dựng và tìm giải pháp tối ưu cho mạng lưới.
* **Tính năng**: Tính toán tổng chi phí và đề xuất các kết nối thay thế có chi phí thấp hơn (nếu có).

1. **Báo cáo tổng quan mạng lưới**:

* **Mục đích**: Cung cấp báo cáo chi tiết về mạng lưới và trạng thái của từng đoạn ống.
* **Tính năng**: Xuất báo cáo về tổng chi phí, số lượng điểm kết nối, trạng thái và vị trí các đoạn kết nối.

### **Giao diện dự kiến**

1. **Giao diện chính**:

Sử dụng thư viện streamlit để tạo giao diện web app đồng thời sử dụng thư viện folium để tạo bản đồ web nhằm thể hiện các vị trí nhà máy nước cũng như các điểm cung cấp nước cần đến

1. **Khu vực nhập liệu**:
   * 1. Khung nhập liệu để thêm/sửa thông tin của các điểm kết nối.
     2. Các mục nhập bao gồm tọa độ, chi phí xây dựng, và trạng thái của các đoạn ống.
2. **Cửa sổ kết quả**:
   * 1. Hiển thị thông tin chi tiết về đường đi ngắn nhất, tổng chi phí, và các đoạn ống đề xuất.
     2. Bản đồ và các tọa độ đường đi ống nước được nhập
3. **Bảng điều khiển**:
   * 1. Thanh công cụ để chọn thuật toán tìm đường (Dijkstra hoặc Floyd-Warshall).
     2. Các nút chức năng để lưu mạng lưới, xuất báo cáo, và quản lý kết nối.
4. **Mục cài đặt**:
   * Tùy chọn điều chỉnh chi phí và các yếu tố khác liên quan đến chi phí xây dựng.

## **CHƯƠNG 2: PHÂN CÔNG NHIỆM VỤ**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên SV | Phân công nhiệm vụ | % Đóng góp |
| 01 | Võ Hoàng Anh  22110104 | - Viết các hàm điều khiển ,config, thuật toán, dữ liệu  - Biên soạn Word | 100% |
| 02 | Nguyễn Hiếu Nghĩa  22110189 | Viết các hàm show kết quả, thông tin đầu vào, dữ liệu  - Biên soạn word | 100% |
| 03 | Trần Như Thiện  22110235 | Viết các hàm show bản đồ và các tọa độ, dữ liệu tọa độ  - Biên soạn word | 100% |

## **CHƯƠNG 3: THIẾT KẾ**

### **1. Thuật toán**

Thuật toán tìm đường đi ngắn nhất trong hệ thống quản lý mạng lưới cấp nước sử dụng thuật toán **Dijkstra** để tối ưu hóa lộ trình giữa các trạm cấp nước và các điểm tiêu thụ nước. Hệ thống sẽ tính toán lộ trình có chi phí xây dựng và bảo trì thấp nhất, giúp giảm chi phí cho mạng lưới cấp nước.

Đầu tiên, hệ thống sẽ tải các kết nối từ tệp dữ liệu chứa thông tin về các trạm cấp nước và các điểm tiêu thụ. Sau khi tải dữ liệu, một đồ thị (graph) được xây dựng, trong đó các điểm là các trạm cấp nước và các điểm tiêu thụ, còn các cạnh là các đoạn ống dẫn với chi phí tương ứng.

Thuật toán Dijkstra sẽ bắt đầu từ trạm cấp nước (điểm bắt đầu) và duyệt qua các điểm kế tiếp, tính toán tổng chi phí cho mỗi lộ trình có thể. Hệ thống sử dụng một hàng đợi ưu tiên (priority queue) để luôn duyệt các điểm có chi phí thấp nhất trước. Quá trình này tiếp tục cho đến khi điểm đến được tìm thấy hoặc không còn điểm nào có thể duyệt.

Khi tìm thấy đường đi ngắn nhất, thuật toán sẽ truy vết lại các điểm đã đi qua từ điểm kết thúc đến điểm bắt đầu, tạo ra một danh sách các điểm nối. Kết quả cuối cùng là lộ trình tối ưu và tổng chi phí xây dựng cho đoạn ống dẫn giữa các trạm.

Thuật toán này không chỉ tìm ra đường đi ngắn nhất, mà còn đảm bảo rằng chỉ các kết nối hoạt động ("active") mới được xét đến, bỏ qua các kết nối đang bảo trì hoặc hỏng hóc, giúp hệ thống duy trì hiệu quả hoạt động trong thực tế.

**Các bước thực hiện:**

1. Tải dữ liệu kết nối từ tệp JSON.
2. Xây dựng đồ thị dựa trên các kết nối có trạng thái "active".
3. Áp dụng thuật toán Dijkstra để tính toán đường đi ngắn nhất.
4. Truy vết đường đi từ điểm đến đến điểm bắt đầu.
5. Trả về kết quả là lộ trình tối ưu và tổng chi phí.

**Lưu ý:** Hệ thống sẽ tính toán lại lộ trình khi có thay đổi trong kết nối hoặc trạng thái của các tuyến ống, đảm bảo rằng lộ trình luôn được tối ưu hóa trong thời gian thực.

### **2. Thiết kế lớp**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên hàm | Mục đích | Tên các SV phụ trách |
| 01 | **Control\_Panel()** | -Dùng để quản lý và hiển thị thông tin về mạng lưới cấp nước  -Dùng để tải và xử lý dữ liệu kết nối mạng lưới  Dùng để thực hiện tính toán đường đi ngắn nhất  **Dùng để tạo báo cáo về kết quả tìm đường**  Dùng để hiển thị bản đồ tương tác | Võ Hoàng Anh |

Table 1.Control Panel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **load\_connections()**  Input: Không có tham số đầu vào.  Output: Trả về dữ liệu kết nối trong định dạng JSON từ tệp connections.json. | Dùng để tải dữ liệu kết nối từ tệp connections.json và trả về dưới dạng danh sách. | Control\_panel.py(10) | Võ Hoàng Anh |
| 02 | **load\_coordinates()**  Input: Không có tham số đầu vào.  Output: Trả về từ điển các tọa độ từ tệp coordinates.json với tên điểm làm khóa và tọa độ tương ứng làm giá trị. | Dùng để tải và trả về danh sách các tọa độ từ tệp coordinates.json | Control\_panel.py(22) | Võ Hoàng Anh |
| 03 | **Def create\_map()**  **Input**:   * path: Danh sách các điểm tạo thành đường đi ngắn nhất. * coordinates: Từ điển các tọa độ của các điểm trong mạng lưới. * connections: Danh sách các kết nối giữa các điểm trong mạng lưới.   **Output**: Trả về đối tượng bản đồ (map) Folium được hiển thị với các kết nối và đường đi ngắn nhất. | Dùng để tạo một bản đồ tương tác với **Folium** | Control\_panel.py(35) | Võ Hoàng Anh |
| 04 | **Def show\_control\_panel()**  **Input**: Không có tham số đầu vào. **Output**: Hiển thị giao diện điều khiển cho người dùng, bao gồm các lựa chọn để tìm đường đi ngắn nhất hoặc tạo báo cáo. | Dùng để hiển thị giao diện điều khiển trong ứng dụng, cho phép người dùng chọn hành động như tìm đường đi ngắn nhất hoặc tạo báo cáo. | Control\_panel.py(73) | Võ Hoàng Anh |

Table 2.Phương thức trong Control Panel

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên hàm | Mục đích | Tên các SV phụ trách |
| 02 | **Data\_input()** | Lớp này dùng để cung cấp giao diện nhập liệu và quản lý thông tin các kết nối trong hệ thống mạng lưới cấp nước.  Người dùng có thể thêm mới hoặc cập nhật các thông tin về kết nối giữa các điểm. | Võ Hoàng Anh |

Table 3.Data Input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **Def load\_coordinates()**  Input: Không có tham số đầu vào.  Output: Trả về danh sách các địa điểm từ tệp coordinates.json trong định dạng JSON. | **Dùng để tải dữ liệu các tọa độ (địa điểm) từ tệp coordinates.json.** | Data\_input.py(5) | Võ Hoàng Anh |
| 02 | **def load\_connections()**  Input: Không có tham số đầu vào.  Output: Trả về danh sách các kết nối từ tệp connections.json trong định dạng JSON hoặc danh sách rỗng nếu tệp không tồn tại. | Dùng để tải dữ liệu các kết nối từ tệp connections.json. | Data\_input.py(10) | Võ Hoàng Anh |
| 03 | **save\_connections(connections)**  Input:  connections: Danh sách các kết nối cần lưu.  Output: Lưu danh sách các kết nối vào tệp connections.json trong định dạng JSON. | Dùng để ghi dữ liệu các kết nối vào tệp connections.json, đảm bảo định dạng đẹp với thụt lề (indent) và mã hóa UTF-8. | Data\_input.py(18) | Võ Hoàng Anh |
| 04 | **show\_input\_form()**  Input: Không có tham số đầu vào.  Output: Hiển thị giao diện người dùng trên Streamlit để thêm hoặc chỉnh sửa kết nối. | Dùng để tạo giao diện nhập liệu cho người dùng với các trường chọn điểm bắt đầu, điểm kết thúc, chi phí, và trạng thái kết nối. | Data\_input.py(23) | Võ Hoàng Anh |

Table 4.Phương thức trong Data Input

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên hàm | Mục đích | Tên các SV phụ trách |
| 03 | **Map\_view()** | Quản lý một hệ thống các điểm tọa độ và các kết nối giữa chúng.  Hiển thị hệ thống trên bản đồ tương tác sử dụng Folium, với các tùy chọn:  Thêm mới điểm tọa độ: Người dùng nhập tên và tọa độ (vĩ độ, kinh độ).  Xóa điểm: Loại bỏ điểm đã chọn và các kết nối liên quan.  Hiển thị bản đồ: Tất cả các điểm và các đường kết nối giữa chúng, với thông tin chi tiết hiển thị qua popup. | Trần Như Thiện |

Table 5.Map View

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **read\_coordinates()**  Input: Không có tham số.  Output: Danh sách các tọa độ từ file coordinates.json. Nếu file không tồn tại, trả về danh sách rỗng. | Tải dữ liệu các tọa độ đã lưu từ file JSON. | Map\_view.py(8) | Trần Như Thiện |
| 02 | **read\_connections()**  Input: Không có tham số.  Output: Danh sách các kết nối từ file connections.json. Nếu file không tồn tại, trả về danh sách rỗng. | Tải dữ liệu các kết nối giữa các điểm từ file JSON. | Map\_view.py(16) | Trần Như Thiện |
| 03 | **save\_coordinates(coordinates)**  Input: Danh sách các tọa độ.  Output: Ghi danh sách tọa độ vào file coordinates.json. | Lưu danh sách tọa độ đã chỉnh sửa hoặc thêm mới | Map\_view.py(46) | Trần Như Thiện |
| 04 | **save\_connections(connections)**  Input: Danh sách các kết nối.  Output: Ghi danh sách kết nối vào file connections.json. | Lưu danh sách kết nối đã chỉnh sửa hoặc thêm mới. | Map\_view.py(52) | Trần Như Thiện |
| 05 | **def show\_map()**  Input: Không có tham số.  Output: Hiển thị bản đồ với các marker và đường kết nối. | Đọc tọa độ và kết nối từ file JSON.  Hiển thị các điểm trên bản đồ dưới dạng marker.  Vẽ đường nối giữa các điểm dựa vào trạng thái (active/maintenance) và chi phí. | Map\_view.py(65) | Trần Như Thiện |
| 06 | **def get\_coordinates\_by\_name(name, coordinates)**  Input:  name: Tên điểm cần tìm.  coordinates: Danh sách các tọa độ.  Output: Trả về tọa độ [lat, lon] của điểm, hoặc None nếu không tìm thấy. | Tìm tọa độ của một điểm cụ thể dựa trên tên. | Map\_view.py(27) | Trần Như Thiện |
| 07 | **draw\_connection(map\_obj, coord\_from, coord\_to, cost, status)**  Input:  map\_obj: Đối tượng bản đồ folium.Map.  coord\_from: Tọa độ điểm bắt đầu.  coord\_to: Tọa độ điểm kết thúc.  cost: Chi phí của kết nối.  status: Trạng thái (active hoặc maintenance).  Output: Vẽ một đường nối trên bản đồ với màu sắc và thông tin trạng thái | Hiển thị đường kết nối giữa các điểm. | Map\_view.py(34) | Trần Như Thiện |
| 08 | **add\_new\_coordinate()**  Input: Người dùng nhập Tên, Vĩ độ, và Kinh độ.  Output:  Thêm tọa độ mới vào file coordinates.json.  Thông báo thành công hoặc lỗi nếu thiếu thông tin. | Thêm một điểm mới vào danh sách tọa độ. | Map\_view.py(99) | Trần Như Thiện |
| 09 | **delete\_coordinate(selected\_name)**  Input: Tên điểm cần xóa (selected\_name).  Output:  Xóa điểm khỏi coordinates.json.  Xóa mọi kết nối liên quan khỏi connections.json. | Loại bỏ một điểm và các kết nối liên quan. | Map\_view.py(117) | Trần Như Thiện |
| 10 | **Def delete\_existing\_coordinate()** Input: Người dùng chọn điểm từ danh sách.  Output:  Hiển thị danh sách các điểm.  Xóa điểm đã chọn khi nhấn nút. | Cung cấp giao diện trực quan các điểm có thể xóa tọa độ | Map\_view.py(133) | Trần Như Thiện |
| 11 | **Def add\_marker()**  Input: Bản đồ, tọa độ [lat, lon], và nội dung popup.  Output: Thêm marker vào bản đồ nếu tọa độ hợp lệ, hoặc báo lỗi nếu không. | Thêm một marker (điểm đánh dấu) vào bản đồ Folium tại tọa độ chỉ định. | Map\_view.py(58) |  |

Table 6.Phương thức trong Map View

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên hàm | Mục đích | Tên các SV phụ trách |
| 04 | **Result\_view()** | Giúp quản lý mạng lưới hiệu quả: Theo dõi trạng thái, chi phí, và hiệu suất của mạng lưới.  Tối ưu hóa quy trình: Đảm bảo các thông tin quan trọng được tổng hợp và trình bày rõ ràng, dễ hiểu cho việc ra quyết định. | Nguyễn Hiếu Nghĩa |

Table 7.Result View

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **Def load\_report()**  Input: Không có tham số.  Output:  Trả về nội dung file network\_report.json dưới dạng dictionary (được tải bằng json.load).  Nếu file không tồn tại hoặc không đọc được, chương trình có thể gặp lỗi (cần xử lý ngoại lệ). | **Đọc báo cáo được lưu trong thư mục data/report/.** | Result\_view.py(5) | Nguyễn Hiếu Nghĩa |
| 02 | **def call\_report()**  nput: Không có tham số.  Output:  Hiển thị thông tin từ báo cáo trong giao diện Streamlit:  Ngày tạo báo cáo (date).  Tổng chi phí (total\_cost).  Danh sách các kết nối mạng lưới (network\_connections). | Lấy dữ liệu từ file JSON và hiển thị một cách rõ ràng trong giao diện. | Result\_view.py(9) | Nguyễn Hiếu Nghĩa |
| 03 | **show\_results()**  Input:Không có tham số.  Output:Gọi hàm generate\_report(từmodulecore.report\_generator) để tạo báo cáo.  Sau đó, hiển thị báo cáo bằng cách gọi call\_report. | Tích hợp quy trình tạo báo cáo và hiển thị kết quả trong giao diện. | Results\_view.py(23) | Nguyễn Hiếu Nghĩa |

Table 8.Phương thức trong Result View

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên hàm | Mục đích | Tên các SV phụ trách |
| 05 | **Cost\_Calculator()** | Phân tích chi phí mạng lưới: Tính toán tổng chi phí giúp người quản lý hiểu được mức độ tốn kém để duy trì hoặc xây dựng mạng lưới. | Võ Hoàng Anh |

Table 9.Cost Calculator

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **load\_connections()**  Input:  Không có tham số truyền vào trực tiếp.  Sử dụng biến toàn cục NETWORK\_FILE\_PATH làm đường dẫn đến file JSON chứa dữ liệu mạng lưới.  Output:  Một danh sách chứa các kết nối mạng lưới (nếu file tồn tại).  Trả về danh sách rỗng nếu file không tồn tại hoặc xảy ra lỗi đọc file. | Đọc dữ liệu kết nối mạng lưới từ file JSON, đảm bảo xử lý trường hợp không tìm thấy file mà không làm gián đoạn chương trình. | Cost\_caculator.py(4) | Võ Hoàng Anh |
| 02 | **calculate\_total\_cost()**  Input:Khôngcó tham số truyền vào trực tiếp.Phụ thuộc vào dữ liệu từ hàm load\_connections.  Output:  Trả về tổng chi phí của tất cả các kết nối (dạng số thực hoặc nguyên).Trả về 0 nếu không có kết nối nào. | Tính toán tổng chi phí của toàn bộ mạng lưới kết nối dựa trên dữ liệu trong file JSON. | Cost\_caculator.py(11) | Võ Hoàng Anh |

Table 10.Phương thức trong Cost Calculator

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên hàm | Mục đích | Tên các SV phụ trách |
| 06 | **Path\_finding()** | Tối ưu hóa chi phí trong mạng lưới: Tìm đường đi ngắn nhất qua các điểm trong hệ thống cấp nước, giảm thiểu chi phí xây dựng và vận hành.  Quản lý trạng thái kết nối: Bỏ qua các kết nối không khả dụng giúp phản ánh trạng thái thực tế của hệ thống. | Võ Hoàng Anh |

Table 11.Path Finding

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **load\_connections()**  Input:  Không có tham số truyền vào trực tiếp.  Sử dụng biến toàn cục NETWORK\_FILE\_PATH làm đường dẫn đến file JSON chứa dữ liệu mạng lưới.  Output:  Trả về một danh sách chứa các kết nối mạng lưới được đọc từ file JSON.  Nếu file không tồn tại hoặc xảy ra lỗi trong quá trình đọc file, hàm trả về danh sách rỗng. | Đọc dữ liệu kết nối mạng lưới từ file JSON.  Đảm bảo xử lý trường hợp không tìm thấy file hoặc lỗi đọc file một cách an toàn, không làm gián đoạn chương trình. | Path\_finding.py(8) | Võ Hoàng Anh |
| 02 | **find\_shortest\_path(start, end)**  Input:  start: Điểm bắt đầu trong mạng lưới.  end: Điểm kết thúc trong mạng lưới.  Output:  Trả về một danh sách các điểm tạo thành đường đi ngắn nhất từ start đến end và tổng chi phí của đường đi đó**.** | Tính toán đường đi ngắn nhất giữa hai điểm trong mạng lưới kết nối bằng cách sử dụng thuật toán Dijkstra.  Xây dựng đồ thị từ dữ liệu kết nối đọc được từ file JSON. | Path\_finding.py(12) | Võ Hoàng Anh |
| 03 | **build\_graph(connections)**  Input:  connections: Danh sách các kết nối trong mạng lưới, mỗi kết nối bao gồm thông tin start, end, và cost.  Output:  Trả về một đồ thị dưới dạng từ điển, trong đó mỗi điểm sẽ có một danh sách các điểm kết nối và chi phí tương ứng**.** | Xây dựng đồ thị từ danh sách các kết nối trong mạng lưới, sử dụng dữ liệu về các kết nối mạng.  Đảm bảo kết nối được tính cho cả hai chiều (đồ thị vô hướng). | Path\_finding.py(23) | Võ Hoàng Anh |
| 04 | **find\_shortest\_path(start,end,connections)**  Input:  start: Điểm bắt đầu.  end: Điểm kết thúc.  Connections: Danh sách các kết nối trong mạng lưới, với thông tin start, end, cost, và status.  Output:  Trả về một đường đi ngắn nhất từ start đến end, loại bỏ các kết nối có trạng thái là ‘maintenance’.  Tổng chi phí của đường đi ngắn nhất**.** | Tính toán đường đi ngắn nhất từ start đến end, sử dụng thuật toán Dijkstra và loại bỏ các kết nối không hoạt động (trạng thái là ‘maintenance’). | Path\_finding.py(37) | Võ Hoàng Anh |

Table 12.Phương thức trong Path Finding

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên hàm | Mục đích | Tên các SV phụ trách |
| 06 | **Report\_generator()** | Đọc dữ liệu kết nối mạng lưới: Chương trình đọc từ tệp JSON chứa thông tin các kết nối trong mạng lưới (được xác định trong NETWORK\_FILE\_PATH). Dữ liệu này bao gồm các thông tin như: điểm bắt đầu, điểm kết thúc, chi phí, và trạng thái của từng kết nối.  Chuẩn bị báo cáo:  Tạo danh sách các kết nối hợp lệ từ dữ liệu đọc được, lọc bỏ các kết nối không hợp lệ.  Tính toán tổng chi phí của các kết nối và lưu lại trong báo cáo.  Tạo và lưu báo cáo: Chương trình tạo một báo cáo JSON với thông tin về:  Thời gian tạo báo cáo.  Tổng chi phí của tất cả các kết nối mạng lưới.  Danh sách chi tiết các kết nối, bao gồm các thông tin về điểm bắt đầu, điểm kết thúc, chi phí, và trạng thái kết nối. | Nguyễn Hiếu Nghĩa |

Table 13.Report Generator

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **Def generate\_report()**  Input:  Không có tham số đầu vào trực tiếp.  Dữ liệu đầu vào được lấy từ hai file JSON:  NETWORK\_FILE\_PATH chứa các kết nối mạng lưới.  REPORT\_FILE\_PATH là nơi lưu báo cáo đầu ra dưới dạng JSON.  Output:  Hàm trả về một báo cáo JSON chứa thông tin kết nối mạng lưới, tổng chi phí của các kết nối và thời gian báo cáo được tạo.  Báo cáo này được lưu tại REPORT\_FILE\_PATH.. | Tạo báo cáo chứa tổng chi phí của tất cả các kết nối mạng lưới và thông tin chi tiết về các kết nối.  Đảm bảo báo cáo được lưu trữ và dễ dàng truy xuất cho các mục đích quản lý và phân tích. | Report\_generator.py(5) | Nguyễn  Hiếu Nghĩa |

Table 14.Phương thức trong Report Generator

### **3. Thiết kế dữ liệu.**

a) Dữ liệu vị trí tọa độ( tên địa điểm, kinh độ, vĩ độ):

File coordinates.json sẽ chứa các dữ liệu của tên địa điểm bao gồm kinh độ và vĩ độ của địa điểm đó. File sẽ cung cấp đầu vào cho hàm map\_view để hiện tọa độ các điểm trên bản đồ.

File được thiết kế như sau:

|  |
| --- |
| {          "name": "Nhà máy nước Thủ Đức",          "coordinates": [              10.852960874075226,              106.77441278660831          ]      } |

b) Dữ liệu về điểm bắt đầu , điểm kết thúc, mức giá và trạng thái hiện tại của đường ống nước:

File connections.json sẽ cung cấp đầu vào cho hàm data\_input để xây dựng các đường đi cũng như tìm ra đường tối ưu nhất thông qua thuộc tính cost được hiện ra ở hàm control\_panel.py

File được thiết kế như sau:

|  |
| --- |
| {          "start": "Xưởng thực tập nguội",          "end": "Xưởng thực tập nghề",          "cost": 10000,          "status": "maintenance"      }, |

Table 15.Thiết kế file connections.json

c) Dữ liệu đầu ra cho biết các thuộc tính hiện có:

File coordinates.json sẽ cung cấp dữ liệu đầu ra cho hàm result\_view nhằm thể hiện được các thuộc tính hiện có của đường ống nước đang muốn xét đến.

File được thiết kế như sau:

|  |
| --- |
| {              "start": "Khu C",              "end": "Khu D",              "cost": 200,              "status": "active"          }, |

Table 16.Thiết kế file coordinates.json

### **4. Thiết kế giao diện.**

Giao diện chính được tạo trên streamlit.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên cửa sổ | Mục đích | Các quyết định khi thiết kế |
| 1 | Map View    Hình 1.Giao diện Map View | Giao diện show map và các địa điểm của hệ thống ống dẫn nước(thêm và xóa các địa điểm). | Trần Như Thiện.  Cần một giao diện bản đồ bên cạnh thanh công cụ tiện cho việc thêm xóa các tọa độ mới. |
| 2 | Data Input    Hình 2.Giao diện Data Input | Giao diện hiển thị nơi nhập các thông tin để add connections. | Võ Hoàng Anh.  Giao diện riêng để thêm các kết nối và giá trị cost của các kết nối |
| 3 | Result View    Hình 3.Giao diện Result View | Giao diện hiển thị báo cáo kết quả | Nguyễn Hiếu Nghĩa.  Giao diện riêng để theo dõi các kết quả. |
| 4 | Find Shortest Path    Hình 4.Giao diện Input của Find Shortest Path | Giao diện hiển thị nơi chọn điểm đầu và điểm cuối để tìm ra đường đi ngắn nhất | Võ Hoàng Anh.  Giao diện nằm bên trên bản đồ tiện cho việc chọn điểm. |
| 5 | Find Shortest Path    Hình 5.Giao diên Output của Find Shortest Path | Giao diện hiển thị bản đồ đường đi ngắn nhất giữa 2 điểm đã chọn từ input | Võ Hoàng Anh.  Giao diện output ngay dưới input tiện cho việc theo dõi đường đi ngắn nhất ngay sau khi nhập |
| 6 | Generate Report    Hình 6.Giao diện Generate Report | Giao diện xuất file báo cáo | Võ Hoàng Anh.  Giao diện riêng để người dùng lấy về file báo cáo. |

Table 17.Giao diện

### **5. Chức năng chính của hệ thống**

**Map\_view:**

Cho phép người dùng có thể nhập vào 1 địa điểm mới thông qua kinh độ và vĩ độ.

Input: Tên địa điểm , kinh độ, vĩ độ.

Cho phép người dùng có thể xóa 1 địa điểm đang có trong hệ thống thông qua tên địa điểm. Người dùng chỉ việc chọn tên địa điểm có trong thanh selectbox và nhấn nút xóa.

Show vị trí tọa đọa mà người dùng vừa nhập trên bản đồ folium.

**Data Input:**

Cho phép người dùng nhập vào các thuộc tính của một đường ống nước( điểm bắt đầu, kết thúc, cost, active).

Từ những dữ liệu người dùng nhập này hệ thống sẽ show ra đường ống vừa được thiết kế bên trang Map View (biểu diễn dưới dạng 1 đường nối liền 2 điểm đầu và kết thúc, màu xanh ý chỉ cho đường có trạng thái active và đường màu đỏ biểu thị cho trạng thái maintenance).

Nhờ những dữ liệu được cung cấp này hệ thống sẽ tìm được đường ống có tổng giá trị cost thấp nhất có thể.

**Result View:**

Cho phép người dùng có thể xem được các thuộc tính mà 1 hệ thống hiện tại đang có( điểm bắt đầu, kết thúc, cost, trạng thái).

Từ những dữ liệu này người dùng có thể dễ dàng quan sát được giá cả và trạng thái của từng đường đi từ đó đảm bảo được dữ liệu phân tích.

**Control Panel:**

Cho phép người dùng chọn điểm bắt đầu(các trạm nước đang có trong hệ thống) và điểm kết thúc(điểm tiêu thụ nước bất kỳ) sau đó bấm nút Find Path hệ thống sau đó sẽ sử dụng thuật toán Dijsktra và tìm được đường có giá cả thấp nhất và show ra bản đồ.

Đồng thời ở phần này người dùng có thể in ra bản báo cáo của đường đi tốt nhất vừa tìm được ra thành file text thông qua nút Generate Report.

## **CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ**

### **1. Các thư viện cần phải cài đặt**

**• Streamlit**

Streamlit là một thư viện Python mạnh mẽ giúp xây dựng các ứng dụng web tương tác một cách nhanh chóng và dễ dàng. Nó đặc biệt phù hợp cho các nhà khoa học dữ liệu và lập trình viên Python để tạo ra các dashboard hoặc ứng dụng phân tích dữ liệu mà không cần phải viết mã web phức tạp. Streamlit hỗ trợ các tính năng như hiển thị đồ thị, bảng, hình ảnh, và các thành phần giao diện người dùng (UI) dễ dàng thông qua cú pháp Python thuần túy.

Cách cài đặt: *pip install streamlit*.

**• NetworkX**

NetworkX là một thư viện Python dùng để tạo, thao tác và nghiên cứu các cấu trúc đồ thị và mạng lưới. Thư viện này hỗ trợ việc xây dựng đồ thị có trọng số, không trọng số, có hướng và vô hướng. Nó cung cấp các thuật toán tối ưu hóa như tìm đường đi ngắn nhất, phát hiện cộng đồng, phân tích kết nối và nhiều công cụ mạnh mẽ khác cho việc làm việc với đồ thị.

Cách cài đặt: *pip install networkx*.

**• Geopy**

Geopy là thư viện Python giúp làm việc với các dịch vụ địa lý, như tính toán khoảng cách giữa các địa điểm, dịch chuyển địa lý và lấy tọa độ từ các địa chỉ. Geopy hỗ trợ nhiều dịch vụ geocoding khác nhau như Google Maps, Nominatim (OpenStreetMap), và nhiều hơn nữa, giúp dễ dàng truy vấn và xử lý thông tin địa lý.  
Cách cài đặt: *pip install geopy*.

**• Matplotlib**

Matplotlib là thư viện vẽ đồ thị phổ biến trong Python. Nó cho phép người dùng tạo ra các đồ thị tĩnh, động, và tương tác chẳng hạn như đồ thị đường, đồ thị thanh, đồ thị phân tán, biểu đồ tròn, và nhiều dạng đồ thị khác. Matplotlib hỗ trợ tùy chỉnh cao và có thể xuất các đồ thị dưới nhiều định dạng file như PNG, PDF, SVG.  
Cách cài đặt: *pip install matplotlib*.

**• Numpy**

Numpy là một thư viện quan trọng trong Python để tính toán khoa học. Nó cung cấp các cấu trúc dữ liệu như mảng (arrays) và các hàm toán học hỗ trợ phép toán ma trận và đại số tuyến tính. Numpy giúp tối ưu hóa các phép toán số học trên mảng và ma trận, và hỗ trợ các phép tính số học nhanh chóng với hiệu suất cao.  
Cách cài đặt: *pip install numpy*.

**• Folium**

Folium là một thư viện Python dành cho việc tạo và hiển thị bản đồ tương tác, sử dụng Leaflet.js, một thư viện JavaScript. Nó hỗ trợ các tính năng như tạo các điểm đánh dấu (markers), vẽ đường, khu vực, và biểu đồ dữ liệu địa lý trực quan. Folium rất hữu ích khi cần tạo bản đồ với dữ liệu không gian (geospatial data) và có thể tích hợp với các dịch vụ như OpenStreetMap và Google Maps.

Cách cài đặt: *pip install folium*.

**• JSON**

JSON (JavaScript Object Notation) là một định dạng dữ liệu nhẹ được sử dụng để trao đổi dữ liệu giữa các hệ thống. Trong Python, thư viện json cho phép dễ dàng chuyển đổi giữa các kiểu dữ liệu Python (như dictionary, list) và định dạng JSON. Nó hỗ trợ các thao tác như mã hóa (encoding), giải mã (decoding), đọc và ghi dữ liệu JSON từ các file hoặc chuỗi.

Cách cài đặt: Thư viện json là một thư viện chuẩn của Python, không cần cài đặt thêm.

### **2 Kiểm thử**

Mô tả các tình huống kiểm thử:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | **Tình huống:** Kiểm tra giao diện hiển thị danh sách các điểm  **- Dữ liệu vào:** file coordinates.json và file connections.json tồn tại và có dữ liệu hợp lệ.  Connection.json:  {          "start": "Khu A",          "end": "Tòa nhà trung tâm",          "cost": 5000,          "status": "active"      },  Coordinates.json:     {"name": "Nhà máy nước Thủ Đức", "coordinates": [10.852960874075226, 106.77441278660831]},  **- Kết quả dự kiến:**  + Dropdown "Select start point" hiển thị: ["Nhà máy nước Thủ Đức  ", "Tòa E"].  + Dropdown "Select end point" hiển thị: ["Tòa E", "Khu C"].  + Không có thông báo lỗi trên giao diện. | Xác minh rằng danh sách các điểm trong giao diện được tải đúng từ tệp dữ liệu JSON. | Kiểm tra dữ liệu hiển thị cơ bản là bước đầu để đảm bảo tính toàn vẹn và chính xác của hệ thống. |
| 2 | **Tình huống:** Người dùng chọn điểm bắt đầu và kết thúc hợp lệ.  **- Dữ liệu đầu vào:**  + Start: “Nhà máy nước Thủ Đức”.  + End: “ Khu C”  + file connections.json và coordinates.json  **- Kết quả đự kiến:**  + Khi nhấn nút Find Path, gọi đến hàm find\_shortest\_path() và trả về là:  (["Nhà máy nước Thủ Đức", "Xưởng máy", "Khu C"], 1100)  + Giao diện hiển thi:   * Shortest Path: Nhà máy nước Thủ Đức → Xưởng máy → Khu C * Total Cost: 1100 | Kiểm tra tính chính xác của thuật toán tìm đường đi ngắn nhất và sự đồng bộ giữa dữ liệu hiển thị và thuật toán. | Quyết định sử dụng dữ liệu đầu vào nhỏ và có kết quả dự đoán được nhằm giảm thiểu yếu tố phức tạp, tập trung vào việc kiểm tra thuật toán. |
| 3 | **Tình huống:** Hiển thị bản đồ  **- Dữ liệu vào**:   * Đường đi ngắn nhất: ["Nhà máy nước Thủ Đức", "Xưởng máy", "Khu C"] * File coordinates.json và connections.json như tình huống 1   - **Kết quả dự kiến**:   * Bản đồ Folium hiển thị:   + Đường đi tất cả các kết nối (màu xanh dương).   + Đường đi ngắn nhất (màu đỏ, nét dày hơn).   + Marker với màu sắc phân biệt:     - A: Xanh dương (bắt đầu).     - B: Cam (trung gian).     - C: Xanh lá cây (kết thúc). | Đảm bảo bản đồ hiển thị đúng cả hệ thống kết nối và đường đi ngắn nhất một cách trực quan. | + Đường đi ngắn nhất được vẽ màu đỏ và nét dày hơn để nổi bật hơn các kết nối khác, giúp người dùng dễ dàng nhận biết.  + Quyết định sử dụng các marker (điểm đánh dấu) với màu sắc phân biệt cho điểm bắt đầu, trung gian, và kết thúc để tăng tính rõ ràng.  + Việc kết hợp cả tooltip và popup trên bản đồ giúp cải thiện trải nghiệ người dùng khi theo dõi thông tin chi tiết. |
| 4 | **Tình huống:** Tệp dữ liệu bị lỗi hoặc thiếu  **- Dữ liệu vào**:   * Tệp connections.json hoặc coordinates.json bị thiếu hoặc chứa JSON không hợp lệ.   - **Kết quả dự kiến**:   * Khi nhấn nút **Find Path**, xuất hiện thông báo lỗi:   "Connections or coordinates data is missing!"   * Không thực hiện thêm bất kỳ thao tác nào khác. | Đảm bảo ứng dụng xử lý tốt các lỗi liên quan đến tệp dữ liệu. | Hiển thị thông báo lỗi cụ thể như "Connections or coordinates data is missing!" giúp người dùng hiểu rõ nguyên nhân thay vì chỉ hiển thị lỗi chung chung hoặc để ứng dụng dừng đột ngột. Việc không thực hiện thêm thao tác nào nếu gặp lỗi dữ liệu giúp ngăn chặn hệ thống chạy với dữ liệu không hợp lệ, đảm bảo an toàn và ổn định. |
| 5 | **Tình huống:** Người dùng nhập giá trị tọa độ ngoài phạm vi cho phép hoặc bỏ trống một trường.  **- Dữ liệu vào:**   * Latitude: -91.000000 * Longitude: -181.000000 * Bỏ trống trường Name   **-Kết quả dự kiến:**   * Khi nhấn **Add** sẽ hiển thị thông báo lỗi "Vui lòng nhập đủ thông tin." hoặc "Giá trị không hợp lệ." | Đảm bảo chỉ cho phép thêm tọa độ hợp lệ và xử lý đúng trường hợp thiếu thông tin hoặc dữ liệu ngoài phạm vi | Kiểm tra giúp đảm bảo dữ liệu đầu vào chính xác trước khi thêm vào file JSON. Việc nhập thông tin không hợp lệ sẽ bị từ chối với thông báo lỗi cụ thể |
| 6 | **Tình huống:** Xóa điểm không hợp lệ  **- Dữ liệu vào**:   * Select coordinates to delete: Không chọn tên. * Select coordinates to delete: Chọn một điểm đã bị xóa trước đó   - **Kết quả dự kiến**:   * Khi nhấn nút **Delete**, xuất hiện thông báo lỗi "Vui lòng chọn điểm hợp lệ để xóa!” | Đảm bảo không cho phép xóa điểm khi người dùng chưa chọn đúng tên hoặc chọn tên không tồn tại | Việc không kiểm tra đầu vào trước khi xóa có thể gây lỗi hoặc làm ứng dụng treo. Test case này đảm bảo chỉ xử lý yêu cầu xóa khi tên điểm hợp lệ |
| 7 | **Tình huống:** Nhập một điểm đã có tên trên bản đồ  **- Dữ liệu vào**:   * PointA (đã tồn tại trong coordinates.json) bao gồm Latitude, Longitude hợp lệ.   - **Kết quả dự kiến**:   * Khi nhấn nút **Add**, xuất hiện thông báo lỗi "Tên đã được sử dụng” | Đảm bảo mỗi điểm trên bản đồ có một tên duy nhất | Việc thêm tên trùng lặp mà không kiểm tra có thể làm sai dữ liệu hoặc gây lỗi trong các chức năng liên quan |
| 8 | **Tình huống:** Thêm connection với điểm đầu và điểm cuối trùng nhau  **- Dữ liệu vào**:   * Select start point: Nhà máy nước Thủ Đức. * Select end point: Nhà máy nước Thủ Đức.   - **Kết quả dự kiến**:   * Hiển thị thông báo lỗi: *"Điểm bắt đầu và điểm kết thúc không được trùng nhau."* * Không có thay đổi trong file connections.json | Đảm bảo không cho phép thêm kết nối khi điểm bắt đầu và kvết thúc giống nhau | Việc connection với điểm đầu và điểm cuối giống nhau có thể gây ra dư thừa dữ liệu, dẫn đến các lỗi không mong đợi. |
| 9 | **Tình huống:** Nhập giá trị cost không hợp lệ  **- Dữ liệu vào**:   * Cost of connection: -10.   - **Kết quả dự kiến**: Hiển thị thông báo lỗi: *"Giá trị chi phí phải lớn hơn hoặc bằng 0."* Không có thay đổi trong file connections.json | Đảm bảo chỉ cho phép nhập giá trị dương cho chi phí kết nối. | Việc nhập giá trị âm sẽ gây ra sai xót trong việc tính toán đường đi tối ưu nhất |

Table 18.Kiểm thử

## **CHƯƠNG 5: KẾT LUẬN**

Dự án "Ứng dụng hệ thống tối ưu đường dẫn nước" được chọn nhằm giải quyết vấn đề tối ưu hóa mạng lưới cấp nước trong các khu vực đô thị, nơi mà việc quản lý hạ tầng nước sạch có ảnh hưởng trực tiếp đến chi phí vận hành và bảo trì. Việc tối ưu hóa các tuyến ống dẫn nước từ các trạm cấp nước đến các điểm tiêu thụ không chỉ giúp giảm thiểu chi phí xây dựng và bảo trì mà còn nâng cao hiệu quả cung cấp nước cho cộng đồng.

Dự án đã được chúng em hoàn thành được khoảng 85% so với mục tiêu ban ban đầu đề ra. Vẫn còn một vài chỗ chúng em vẫn chưa thể hoàn thiện như những gì dự tính.

Trong quá trình làm chúng em đã gặp 1 vài khó khăn mà cần vài ngày mới có thể giải quyết được.

+ Ban đầu chúng em dự tính sẽ sử dụng API của GGMap và OpenStreetMap để hiện bản đồ và các tọa độ được thêm vào.Tuy nhiên ý tưởng đó đã không thực hiện được vì không có sẳn keyAPI nên chúng em đã chuyển qua dùng thư viện tạo bản đồ có sẳn được hỗ trợ trong công cụ Python và đã thực hiện được việc show bản đồ và các tọa độ được chỉ điểm.

+ Mặc dù đã có được bản đồ và các tọa độ nhưng chúng em không thực hiện được thao tác click chọn 1 điểm có trong bản đồ để lấy tọa độ mình muốn mà không cần phải nhập khi ứng dụng chạy trên streamlit thì không thực hiện được thao tác này.

+ Dữ liệu khi nhập các điểm và hệ thống ban đầu sẽ được lưu ở đâu cũng là 1 khó khăn lớn chúng em đã gặp phải. Nhờ vào việc tìm hiểu của cả 3 thành viên nhóm mà chúng em đã đưa ra quyết định sử dụng file json để có thể lưu các dữ liệu tọa độ và các thuộc tính khác của đường ống nước mà khi chạy chương trình có thể tự động cập nhật theo các thao tác của người dùng đang thực hiện, giúp dữ liệu được linh hoạt hơn và dễ dùng hơn.

+ Trong quá trình thực hiện Giao Diện của dự án cũng là một trong những vẫn đề khiến chúng em phải suy nghĩ rất nhiều. Sau khi thực hiện bằng cách dùng thư viện Tkinter nhưng không thành công thì chúng em đã đưa ra quyết định sẽ sử dụng thư viện streamlit và giao diện trên đó.

* **Ý tưởng phát triển đồ án**

+ Có thể thực hiện việc click vào bản đồ để có thể tối ưu hóa việc chọn tọa độ khi người dùng không biết chính xác tọa độ của điểm cần tính toán.

+ Các đường ống nước có thể chạy dọc theo các đường có sẳn trên bản đồ thông qua GPS.

+ Có thể phát triển thành app có dữ liệu và giao diện người dùng

+ Dùng GPS để đo được chiều dài đường ống nước. Sau đó AI sẽ sử dụng thông tin độ dài và giá của đường ống nước đó mà đưa ra được hệ thống tối ưu hơn.

Những ưu điểm và nhược điểm của đồ án

+ Giao diện dễ sử dụng đối với người dùng

+ Tính toán chính xác được đường dẫn nước có giá thấp nhất thông quâ thuật toán Dijkstra

+ Giúp người dùng có thể tìm ra được đường dẫn nước tối ưu 1 cách nhanh nhất có thể

Nhược điểm

+ Chưa click được vào bản đồ để chọn tọa độ khi người dùng không biết rõ chính xác được tọa độ điểm cần chọn.

+ Chưa sử dụng được GPS để đo được khoảng cách của các đường ống nước

Mặc dù đã gặp phải rất nhiều khó khăn trong quá trình thực hiện dự án này nhưng nhờ những khó khăn ấy đã giúp chúng em có thể tìm hiểu và học hỏi được nhiều kiến thức mới hơn góp phần lớn cho việc học tập và công việc sau này.

**D. TÀI LIỆU THAM KHẢO**

[1]. <https://tapchinuoc.vn/phat-trien-ung-dung-quan-ly-mang-luoi-cap-nuoc-dua-tren-bo-cong-cu-epanet-toolkit-va-moi-truong-lap-trinh-matlab-175240511135020082.htm>

[2]. <https://viblo.asia/p/thuat-toan-dijkstra-va-ung-dung-aWj53zgQl6m>

[3]. <https://hackernoon.com/lang/vi/c%C3%A1ch-t%C3%B4i-s%E1%BB%AD-d%E1%BB%A5ng-tr%C4%83n-v%C3%A0-folium-%C4%91%E1%BB%83-h%C3%ACnh-dung-c%C3%A1c-ho%E1%BA%A1t-%C4%91%E1%BB%99ng-ngo%C3%A0i-tr%E1%BB%9Di-c%E1%BB%A7a-m%C3%ACnh>