

# ĐẠI HỌC BÁCH KHOA HÀ NỘI

## HỆ TRỢ GIÚP QUYẾT ĐỊNH

Xây dựng hệ thống trợ giúp khách hàng tìm thuê chỗ ở

**Học viên:** Hoàng Nhật Minh - 20251225M

Trần Ngọc Bảo - 20251227M

**Giảng viên:** PGS. TS. Trần Đình Khang

**Khoa:** Khoa học máy tính

**Trường:** Công nghệ thông tin và Truyền thông

HÀ NỘI, 11/2025

## TÓM TẮT NỘI DUNG BÁO CÁO

Trong bối cảnh đô thị hóa nhanh và nhu cầu di chuyển tăng cao, người thuê nhà phải đối mặt với khối lượng thông tin lớn, chất lượng thông tin không đồng đều và nhiều tiêu chí lựa chọn đa dạng dẫn đến quyết định mang tính cảm tính và thiếu căn cứ. Báo cáo này đề xuất và triển khai một hệ thống trợ giúp ra quyết định nhằm hỗ trợ người dùng tìm kiếm và lựa chọn chỗ ở phù hợp theo các tiêu chí cá nhân.

Mục tiêu chính của nghiên cứu là chuẩn hóa và tổ chức dữ liệu nhà thuê, cho phép người dùng nhập các tiêu chí và trọng số ưu tiên, áp dụng phương pháp ra quyết định đa tiêu chí để đánh giá và xếp hạng các phương án, đồng thời trình bày kết quả dưới dạng giao diện trực quan.

Giải pháp được thiết kế gồm ba thành phần chính: Thu thập và chuẩn hóa dữ liệu lưu trữ trong cơ sở dữ liệu quan hệ; Mô-đun tính toán sử dụng topsis để xác định mức độ phù hợp giữa từng căn nhà và vector trọng số do người dùng cung cấp; Giao diện cơ bản cho phép nhập tiêu chí, điều chỉnh trọng số và hiển thị danh sách gợi ý theo thứ tự tối ưu.

Hệ thống này đã chứng minh tính khả thi của cách tiếp cận trong việc giảm thiểu chủ quan khi chọn nhà và cung cấp một công cụ đánh giá khách quan dựa trên dữ liệu có cấu trúc.

## MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI .....	1
1.1 Đặt vấn đề .....	1
1.2 Mục tiêu và phạm vi đề tài .....	1
1.3 Định hướng giải pháp .....	1
1.4 Bố cục báo cáo .....	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU .....	3
2.1 Khảo sát hiện trạng .....	3
2.2 Yêu cầu chức năng .....	3
2.2.1 Biểu đồ use case .....	3
2.2.2 Lưu đồ ảnh hưởng .....	9
2.3 Yêu cầu phi chức năng .....	12
CHƯƠNG 3. CƠ SỞ LÝ THUYẾT .....	13
3.1 Hệ trợ giúp quyết định .....	13
3.2 Ra quyết định đa tiêu chí .....	13
3.3 Thuật toán TOPSIS .....	13
CHƯƠNG 4. THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG .....	16
4.1 Thiết kế hệ thống .....	16
4.1.1 Kiến trúc tổng quan hệ thống .....	16
4.1.2 Thiết kế cơ sở dữ liệu .....	16
4.1.3 Thiết kế chương trình .....	23
4.2 Triển khai hệ thống .....	24
4.2.1 Thu thập và tiền xử lý dữ liệu .....	24
4.2.2 Phương pháp tính điểm dựa trên topsis .....	24
4.2.3 Cài đặt hệ thống .....	26
CHƯƠNG 5. KẾT QUẢ THỰC NGHIỆM .....	27
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	30
6.1 Kết luận .....	30
6.2 Hướng phát triển .....	30
TÀI LIỆU THAM KHẢO .....	31

## **DANH MỤC HÌNH VẼ**

Hình 1 . Biểu đồ use case của hệ thống .....	4
Hình 2 . Lưu đồ ảnh hưởng của hệ thống .....	12
Hình 3 . Sơ đồ thiết kế cơ sở dữ liệu .....	17
Hình 4 . Cấu trúc mã nguồn của chương trình .....	23
Hình 5 . Bộ lọc của hệ thống trợ giúp quyết định .....	27
Hình 6 . Lựa chọn các tiêu chí để tính điểm topsis .....	28
Hình 7 . Bảng tiêu chí lý tưởng .....	28
Hình 8 . Bảng xếp hạng chi tiết .....	29
Hình 9 . Kết quả trả về dựa trên điểm topsis .....	29

## **DANH MỤC BẢNG BIỂU**

Bảng 1 . Đặc tả use case Nhập/Nạp dữ liệu .....	5
Bảng 2 . Đặc tả use case Quản lý dữ liệu .....	5
Bảng 3 . Đặc tả use case Tiền xử lý dữ liệu .....	6
Bảng 4 . Đặc tả use case Phân tích / Chạy mô hình .....	6
Bảng 5 . Đặc tả use case Chạy kịch bản & So sánh .....	7
Bảng 6 . Đặc tả use case Xem khuyến nghị / Gợi ý .....	8
Bảng 7 . Đặc tả use case Cấu hình hệ thống .....	8
Bảng 8 . Đặc tả use case Xem nhật ký / Kiểm toán .....	9
Bảng 9 . Các yếu tố ảnh hưởng đến quyết định chọn nhà .....	10
Bảng 10 . Các yếu tố hệ thống không thể dự đoán được ảnh hưởng tới quyết định của khách hàng .....	10
Bảng 11 . Các biến trung gian được sử dụng trong trợ giúp quyết định ..	12
Bảng 12 . Bảng mô tả dữ liệu của tỉnh / thành .....	17
Bảng 13 . Bảng mô tả dữ liệu của quận / huyện .....	17
Bảng 14 . Bảng mô tả dữ liệu của phường / xã .....	18
Bảng 15 . Bảng mô tả thông tin của nhà thuê .....	19
Bảng 16 . Bảng mô tả dữ liệu tham chiếu theo điều kiện sống .....	20
Bảng 17 . Bảng mô tả dữ liệu các điều kiện sống của nhà cho thuê .....	20
Bảng 18 . Bảng mô tả dữ liệu nhật ký người dùng .....	22
Bảng 19 . Bảng mô tả dữ liệu kết quả tìm kiếm .....	22

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong bối cảnh tốc độ đô thị hóa và nhu cầu di chuyển của người dân ngày càng tăng, việc tìm kiếm chỗ ở phù hợp trở thành một bài toán phức tạp đối với nhiều người. Người thuê nhà thường phải đối mặt với nhiều khó khăn như lượng thông tin lớn, chất lượng thông tin không đồng đều, sự đa dạng của các tiêu chí lựa chọn (giá cả, vị trí, tiện nghi, môi trường sống,...), cũng như chi phí và thời gian cần bỏ ra để so sánh giữa các lựa chọn. Trong khi đó, các nền tảng hiện có chủ yếu cung cấp thông tin mà chưa có khả năng phân tích, đánh giá và đưa ra gợi ý tối ưu dựa trên nhu cầu cụ thể của từng người dùng. Điều này dẫn đến việc lựa chọn chỗ ở mang tính cảm tính, thiếu căn cứ và có thể không tối ưu.

Trong bối cảnh đó, việc ứng dụng các phương pháp ra quyết định đa tiêu chí (Multi-Criteria Decision Making - MCDM) vào quá trình hỗ trợ người dùng lựa chọn nhà thuê là cần thiết. Hệ trợ giúp quyết định có thể giúp người thuê đánh giá các lựa chọn nhà dựa trên nhiều tiêu chí định lượng và định tính, từ đó đưa ra kết quả phù hợp và tối ưu hơn.

## 1.2 Mục tiêu và phạm vi đề tài

Mục tiêu của đề tài là xây dựng một hệ thống trợ giúp quyết định (Decision Support System – DSS) nhằm hỗ trợ người dùng tìm kiếm và lựa chọn chỗ ở phù hợp dựa trên các tiêu chí cá nhân. Hệ thống hướng đến việc giải quyết các vấn đề chính sau: (i) Chuẩn hóa và tổ chức dữ liệu nhà thuê theo cấu trúc rõ ràng, dễ phân tích. (ii) Cho phép người dùng nhập vào các tiêu chí ưu tiên như giá thuê, diện tích, vị trí mong muốn, loại nhà, tiện nghi,... (iii) Ứng dụng các phương pháp MCDM, đặc biệt là TOPSIS, để đánh giá mức độ phù hợp của từng lựa chọn. (iv) Trực quan hóa danh sách nhà gợi ý theo thứ tự tối ưu.

Phạm vi đề tài tập trung vào việc xây dựng hệ thống mẫu gồm: cơ sở dữ liệu, mô hình tính toán ra quyết định, và giao diện cơ bản phục vụ nhu cầu nhập tiêu chí và trả về kết quả đánh giá.

## 1.3 Định hướng giải pháp

Giải pháp được định hướng bao gồm ba phần chính: (1) Thu thập và chuẩn hóa dữ liệu nhà thuê: dữ liệu về giá, diện tích, vị trí, tiện nghi, môi trường,... được lưu trữ trong cơ sở dữ liệu quan hệ. (2) Xây dựng mô hình hỗ trợ ra quyết định: sử dụng phương pháp TOPSIS để tính toán hệ số tương đồng của từng căn nhà dựa trên vector trọng số tiêu chí mà người dùng cung cấp. (3) Xây dựng giao diện đơn giản: cho phép người dùng nhập tiêu chí tìm kiếm và nhận được danh sách gợi ý theo mức độ phù hợp. Giải pháp nhằm giảm thiểu tính chủ quan trong quá

trình lựa chọn, đồng thời cung cấp công cụ đánh giá khách quan dựa trên dữ liệu.

#### **1.4 Bộ cục báo cáo**

Phần còn lại của báo cáo được tổ chức như sau:

Chương 2 khảo sát hiện trạng và phân tích yêu cầu của hệ thống.

Chương 3 trình bày các công nghệ được sử dụng trong quá trình xây dựng hệ thống.

Chương 4 mô tả chi tiết thiết kế và triển khai hệ thống, bao gồm thiết kế cơ sở dữ liệu và mô hình chương trình.

Chương 5 trình bày kết quả thực nghiệm và đánh giá hệ thống.

Chương 6 kết luận và đề xuất hướng phát triển trong tương lai.

## **CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU**

### **2.1 Khảo sát hiện trạng**

Khảo sát thực tế về quá trình tìm kiếm chỗ ở dài hạn cho thấy người dùng thường phải tiếp cận thông tin từ nhiều nguồn như các trang web bất động sản, mạng xã hội hoặc môi giới. Tuy nhiên, chất lượng dữ liệu không đồng đều; nhiều tin đăng thiếu thông tin quan trọng, hình ảnh không phản ánh đúng thực tế hoặc nhà đã “ngưng cho thuê” nhưng vẫn còn xuất hiện trên các nền tảng. Điều này khiến người dùng mất nhiều thời gian để sàng lọc và xác minh.

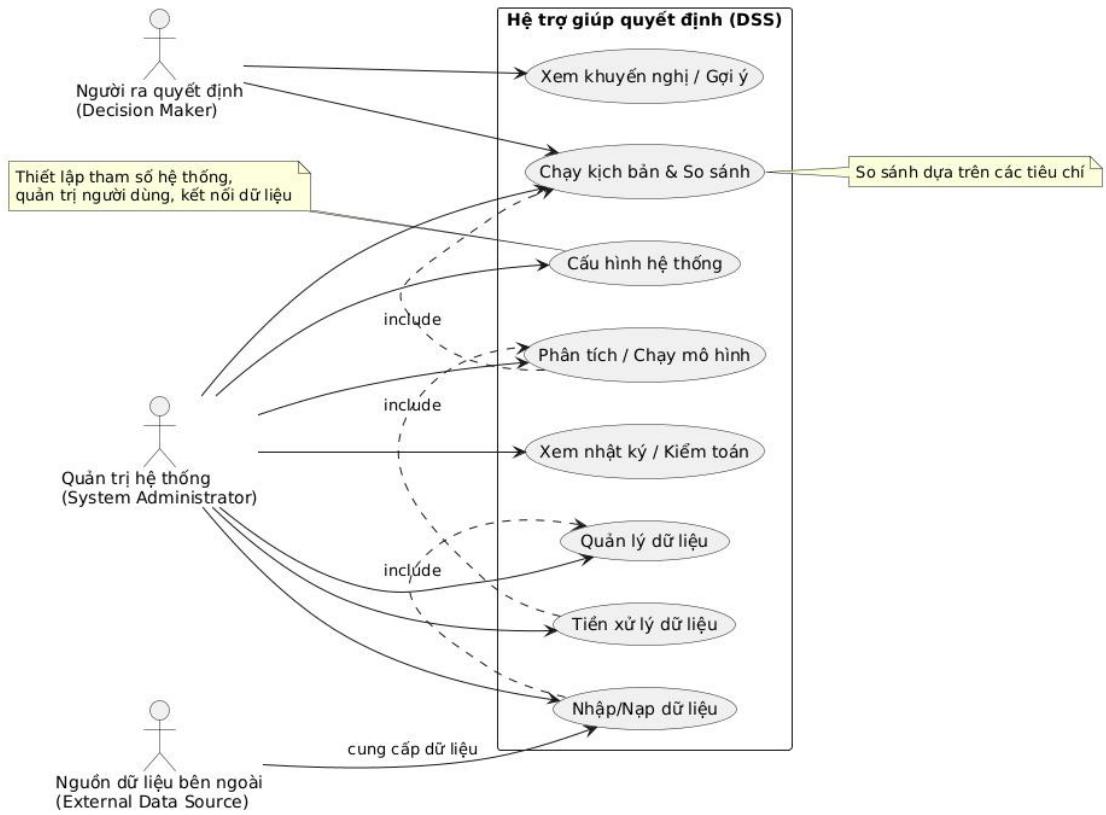
Trong giai đoạn so sánh và đánh giá, phần lớn người dùng phải tự ghi chép hoặc lưu thủ công các lựa chọn để đối chiếu các tiêu chí như giá thuê, diện tích, vị trí và tiện nghi. Việc không có công cụ hỗ trợ phân tích làm cho quá trình ra quyết định dễ mang tính cảm tính, đặc biệt là với những người chưa có kinh nghiệm thuê nhà. Sau khi lựa chọn được một số căn phù hợp, người dùng thường phải liên hệ trực tiếp với chủ nhà để hỏi thêm về chi phí phát sinh, điều kiện hợp đồng hay các tiện ích thực tế, dẫn đến việc tiêu tốn thêm thời gian và công sức.

Bên cạnh đó, người thuê gần như luôn phải đến xem nhà trực tiếp trước khi đưa ra quyết định cuối cùng. Khảo sát cho thấy một người có thể phải xem từ ba đến năm căn trước khi chọn được nơi phù hợp, và không ít trường hợp căn nhà thực tế khác xa mô tả ban đầu. Tổng thể, quá trình tìm thuê nhà hiện nay tốn thời gian, thiếu sự hỗ trợ đánh giá khách quan và phụ thuộc nhiều vào kinh nghiệm cá nhân. Điều này cho thấy nhu cầu về một hệ thống trợ giúp quyết định là cần thiết nhằm giúp người thuê có được thông tin chuẩn hóa, đánh giá rõ ràng theo tiêu chí và rút ngắn thời gian lựa chọn.

### **2.2 Yêu cầu chức năng**

#### **2.2.1 Biểu đồ use case**

Biểu đồ use case của hệ thống gồm có 2 tác nhân chính: Người dùng và Hệ thống DSS. Người dùng có thể nhập tiêu chí tìm kiếm, xem danh sách gợi ý, xem chi tiết từng nhà; hệ thống xử lý lọc, đối sánh, xếp hạng và trả về kết quả. Chi tiết như sau:



Hình 1. Biểu đồ use case của hệ thống

Đặc tả chi tiết của từng use case được mô tả qua các bảng sau:

Use case “Nhập/Nạp dữ liệu”:

<b>Tên</b>	Nhập/Nạp dữ liệu
<b>Diễn giải ngắn</b>	Hệ thống nhập dữ liệu từ nguồn file/DB/External API
<b>Actors</b>	Quản trị hệ thống, Nguồn dữ liệu bên ngoài
<b>Preconditions</b>	
<b>Postconditions</b>	Dữ liệu mới được lưu vào kho dữ liệu DSS (raw), ghi nhận meta-data và lịch sử import
<b>Luồng chính</b>	<ol style="list-style-type: none"> <li>Chọn nguồn: file CSV/Excel, DB, API bên ngoài</li> <li>Hệ thống kết nối/nạp dữ liệu, hiển thị preview dữ liệu</li> <li>Hệ thống lưu dữ liệu thô và ghi log import</li> </ol>
<b>Luồng thay thế</b>	Nếu kết nối thất bại: báo lỗi kết nối

<b>Yêu cầu phi chức năng</b>	Xử lý file lớn, tái chạy import, ghi log đầy đủ
------------------------------	---

Bảng 1. Đặc tả use case Nhập/Nạp dữ liệu

Use case “Quản lý dữ liệu”:

<b>Tên</b>	Quản lý dữ liệu
<b>Diễn giải ngắn</b>	Các tác vụ quản lý dữ liệu (xem, chỉnh sửa meta, xóa, phiên bản)
<b>Actors</b>	Quản trị hệ thống
<b>Preconditions</b>	Dữ liệu đã có trong hệ thống
<b>Postconditions</b>	Thay đổi quản lý dữ liệu được lưu; phiên bản dữ liệu cập nhật
<b>Luồng chính</b>	<ol style="list-style-type: none"> <li>Admin truy cập danh sách dataset.</li> <li>Chọn dataset để xem/sửa/meta.</li> <li>Thực hiện thao tác (đổi tên, mô tả, xóa, phục hồi).</li> <li>Hệ thống lưu thay đổi và ghi lịch sử phiên bản.</li> </ol>
<b>Luồng thay thế</b>	
<b>Yêu cầu phi chức năng</b>	

Bảng 2. Đặc tả use case Quản lý dữ liệu

Use case “Tiền xử lý dữ liệu”:

<b>Tên</b>	Tiền xử lý dữ liệu
<b>Diễn giải ngắn</b>	Chuỗi thao tác tiền xử lý (cleaning, normalize, feature engineering)
<b>Actors</b>	Quản trị hệ thống
<b>Preconditions</b>	Dữ liệu thô đã nhập
<b>Postconditions</b>	Tạo dataset tiền xử lý sẵn sàng cho phân tích; lưu

	pipeline tiền xử lý
<b>Luồng chính</b>	1. Hệ thống thực hiện từng bước hoặc preview transformation. 2. Admin lưu pipeline để tái sử dụng.
<b>Luồng thay thế</b>	Preview hiển thị distribution; admin điều chỉnh tham số
<b>Yêu cầu phi chức năng</b>	hỗ trợ replay pipeline, hiệu năng khi xử lý bộ dữ liệu lớn

Bảng 3. Đặc tả use case Tiền xử lý dữ liệu

Use case “Phân tích / Chạy mô hình”:

<b>Tên</b>	Phân tích / Chạy mô hình
<b>Diễn giải ngắn</b>	Tính điểm topsis để tạo khuyến nghị
<b>Actors</b>	Quản trị hệ thống
<b>Preconditions</b>	Dataset đã tiền xử lý; mô hình và tham số đã sẵn sàng
<b>Postconditions</b>	
<b>Luồng chính</b>	1. Tính điểm topsis 2. Hệ thống trả về kết quả 3. Admin lưu kết quả hoặc chạy lại với tham số khác
<b>Luồng thay thế</b>	Báo lỗi để chỉnh tiền xử lý
<b>Yêu cầu phi chức năng</b>	

Bảng 4. Đặc tả use case Phân tích / Chạy mô hình

Use case “Chạy kịch bản & So sánh”:

<b>Tên</b>	Chạy kịch bản & So sánh
<b>Diễn giải ngắn</b>	Người ra quyết định hoặc admin chạy các kịch bản

	và so sánh kết quả dựa trên các tiêu chí
<b>Actors</b>	Người ra quyết định, Quản trị hệ thống
<b>Preconditions</b>	Hệ thống tính điểm topsis sẵn sàng
<b>Postconditions</b>	
<b>Luồng chính</b>	<ol style="list-style-type: none"> <li>1. Chọn hoặc tạo kịch bản (thay biến tham số đầu vào).</li> <li>2. Tính điểm topsis trên từng kịch bản.</li> <li>3. Hệ thống so sánh và hiển thị bảng so sánh.</li> <li>4. Người dùng chọn kịch bản tốt nhất / lưu kịch bản.</li> </ol>
<b>Luồng thay thế</b>	
<b>Yêu cầu phi chức năng</b>	

Bảng 5. Đặc tả use case Chạy kịch bản & So sánh

Use case “Xem khuyến nghị / Gợi ý”:

<b>Tên</b>	Xem khuyến nghị / Gợi ý
<b>Diễn giải ngắn</b>	Người ra quyết định xem kết luận, khuyến nghị do hệ thống đưa ra
<b>Actors</b>	Người ra quyết định, Quản trị hệ thống
<b>Preconditions</b>	Kết quả phân tích đã sẵn sàng
<b>Postconditions</b>	Người ra quyết định nhận thông tin để đưa quyết định
<b>Luồng chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng mở màn hình kết quả.</li> <li>2. Hệ thống hiển thị khuyến nghị, điểm tin cậy, các chỉ số giải thích.</li> </ol>
<b>Luồng thay thế</b>	
<b>Yêu cầu phi chức năng</b>	

Bảng 6. Đặc tả use case Xem khuyến nghị / Gợi ý

Use case “Cấu hình hệ thống”:

<b>Tên</b>	Cấu hình hệ thống
<b>Diễn giải ngắn</b>	Thiết lập kết nối dữ liệu, quản trị người dùng, tham số hệ thống
<b>Actors</b>	Quản trị hệ thống
<b>Preconditions</b>	Người dùng là admin
<b>Postconditions</b>	Cấu hình được lưu; có thể ảnh hưởng tới các lần phân tích sau
<b>Luồng chính</b>	2. Thực hiện cấu hình: hiệu suất (resource pools), kết nối DB, quản lý roles, backup. 3. Lưu và kiểm tra cấu hình.
<b>Luồng thay thế</b>	
<b>Yêu cầu phi chức năng</b>	

Bảng 7. Đặc tả use case Cấu hình hệ thống

Use case “Xem nhật ký / Kiểm toán”:

<b>Tên</b>	Xem nhật ký / Kiểm toán
<b>Diễn giải ngắn</b>	Admin xem log hệ thống, lịch sử chạy, lịch sử import, thay đổi cấu hình
<b>Actors</b>	Quản trị hệ thống
<b>Preconditions</b>	Người dùng là admin
<b>Postconditions</b>	Cấu hình được lưu; có thể ảnh hưởng tới các lần phân tích sau
<b>Luồng chính</b>	1. Admin truy vấn log theo bộ lọc (time/user/action). 2. Hệ thống trả về tập log; cho phép export.

<b>Luồng thay thế</b>	
<b>Yêu cầu phi chức năng</b>	

Bảng 8. Đặc tả use case Xem nhật ký / Kiểm toán

### 2.2.2 Lưu đồ ảnh hưởng

Mục đích của lưu đồ: (V) Chọn được căn hộ phù hợp với tiêu chí hoặc tối ưu nhất trong khả năng. Trong giai đoạn tìm kiếm thông tin và so sánh đánh giá, các yếu tố ảnh hưởng đến quyết định chọn nhà của người dùng là:

Ký hiệu	Tên biến	Mô tả	Kiểu dữ liệu	Nguồn dữ liệu
(P)	Giá thuê	Chi phí thuê nhà mỗi tháng	Số nguyên	Dữ liệu nhà ở
(L)	(Lat, Lon) vị trí ưu tiên, vị trí mà người dùng muốn tìm tro xung quanh	Chỗ ở như địa chỉ trường học, nơi làm việc,...	Cặp số thực	
(S)	Diện tích	Diện tích phòng hoặc căn hộ	Số thực	Dữ liệu nhà ở
(A)	Tiện nghi (amenities)	Mức độ đầy đủ của các tiện nghi, ít tiện nghi nếu điểm thấp.	Thang điểm 1-10	Dữ liệu nhà ở hoặc đánh giá từ người dùng trước
(T)	Loại căn hộ	Nhà trọ, căn hộ dịch vụ, nhà nguyên căn,...	Enum	Dữ liệu nhà ở
(CP)	Thời gian hiệu lực của hợp	Thời gian người thuê thỏa thuận với người	Số	Dữ liệu nhà ở hoặc có thể linh

	đồng (Contract period)	cho thuê về việc sử dụng căn hộ	số nguyên	đóng theo thỏa thuận
(U <sub>w</sub> )	Trọng số tiêu chí của người dùng	Mức độ ưu tiên các tiêu chí của người dùng	Vector số điểm	Input của người dùng

Bảng 9. Các yếu tố ảnh hưởng đến quyết định chọn nhà

Các yếu tố hệ thống không thể dự đoán được ảnh hưởng trực tiếp tới quyết định của khách hàng:

Ký hiệu	Tên biến	Mô tả	Kiểu dữ liệu	Nguồn dữ liệu
(IQ)	Chất lượng thông tin (information quality)	Độ chính xác của thông tin nhà trọ từ hệ thống	Số nguyên	Chỉ có thể thu thập một phần dựa vào đánh giá của khách hàng
(E)	Môi trường sống (environment)	Môi trường xung quanh của nhà trọ, ( ô nhiễm tiếng ồn, ô nhiễm không khí, an ninh kém,... ) mang ý nghĩa tiêu cực với số nhỏ.	Thang điểm 1-10	Mang tính thời điểm, chỉ có thể xác định sau khi khách hàng tới xem nhà
(EP)	Chi phí phát sinh (extra price)	Chi phí phát sinh trong quá trình lưu trú, có thể là giá điện, nước, phí dịch vụ,...	Số nguyên	Thông thường dựa vào chủ quan chi tiêu của người dùng

Bảng 10. Các yếu tố hệ thống không thể dự đoán được ảnh hưởng tới quyết định của khách hàng

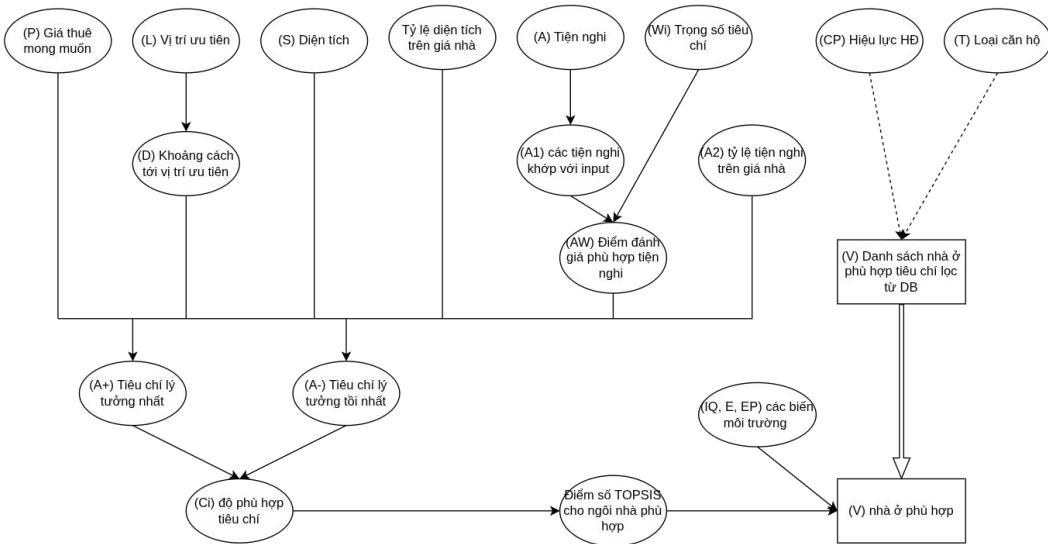
Các biến trung gian sẽ được sử dụng trong quá trình trợ giúp quyết định:

Ký hiệu	Tên biến	Mô tả	Kiểu dữ liệu	Nguồn dữ liệu
(Ci)	Hệ số tương đồng (topsis score)	Xác định mức tối ưu tương đối của mỗi lựa chọn	số thực	
(D)	Khoảng cách tới vị trí ưu tiên	Khoảng cách euclid từ vị trí ưu tiên tới ví trí cản trở	Số thực	
(S1)	Tỷ lệ diện tích trên giá	Tỷ lệ này thể hiện nếu bỏ cùng một số tiền có thể thuê được diện tích bao nhiêu. Tỷ lệ càng cao thì càng lời cho việc thuê nhà.	Số thực	
(A1, AW)	Điểm đánh giá phù hợp tiện nghi	Tính dựa trên những tiện nghi mong muốn và tiện nghi của căn hộ nhân với trọng số.	Số thực	
(A2)	Tỷ lệ tiện nghi trên giá	Tính dựa trên điểm đánh giá tiện nghi chia cho giá tiền. Điểm này càng cao chứng tỏ với cùng 1 số tiền bỏ ra, căn nhà càng mang lại nhiều tiện nghi.	Số thực	
(A+, A-)	Trọng số đánh giá topsis	của tiêu chí lý tưởng và tiêu chí lý tưởng xấu nhất	Số thực	

(C11)	Danh sách nhà trợ phù hợp		Số thực	
-------	------------------------------	--	---------	--

Bảng 11. Các biến trung gian được sử dụng trong trợ giúp quyết định

Lưu đồ ảnh hưởng mô phỏng quá trình ra quyết định của khách hàng:



Hình 2. Lưu đồ ảnh hưởng của hệ thống

### 2.3 Yêu cầu phi chức năng

Hệ thống cần đáp ứng các yêu cầu phi chức năng quan trọng để đảm bảo hiệu quả, độ tin cậy và khả năng mở rộng.

Thứ nhất, về hiệu năng: hệ thống phải phản hồi nhanh khi thực hiện tìm kiếm và xếp hạng.

Thứ hai, về bảo mật: bảo vệ thông tin người dùng và nhật ký tìm kiếm.

Thứ ba, về khả dụng: giao diện rõ ràng, dễ sử dụng với nhiều đối tượng.

Thứ tư, về khả mở: có thể tích hợp thêm tiêu chí và nguồn dữ liệu sau này.

## CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

### 3.1 Hệ trợ giúp quyết định

Hệ trợ giúp quyết định (Decision Support System - DSS) là hệ thống thông tin được thiết kế nhằm hỗ trợ người dùng đưa ra các quyết định trong những tình huống phức tạp, đòi hỏi phân tích nhiều tiêu chí khác nhau. DSS kết hợp dữ liệu, mô hình và giao diện tương tác để cung cấp các gợi ý hoặc đánh giá khách quan. Một hệ DSS điển hình thường bao gồm ba thành phần chính:

(i) Hệ quản trị dữ liệu (Data Management Subsystem): lưu trữ và quản lý dữ liệu đầu vào, ví dụ như giá thuê, diện tích, tiện nghi và vị trí của các chỗ ở trong đề tài này.

(ii) Hệ quản trị mô hình (Model Management Subsystem): thực thi các thuật toán phân tích, trong đó có các mô hình ra quyết định đa tiêu chí (MCDM).

(iii) Giao diện người dùng (User Interface Subsystem): cho phép người dùng nhập tiêu chí, xem kết quả, và tương tác với hệ thống.

DSS đặc biệt phù hợp trong các bài toán lựa chọn tối ưu, nơi người dùng cần đánh giá nhiều phương án dựa trên nhiều tiêu chí, giống như bài toán tìm thuê chỗ ở trong đề tài này.

### 3.2 Ra quyết định đa tiêu chí

Ra quyết định đa tiêu chí (Multi-Criteria Decision Making - MCDM) là lĩnh vực nghiên cứu các phương pháp giúp lựa chọn hoặc xếp hạng các phương án khi có nhiều tiêu chí cùng tham gia đánh giá. Trong thực tế, rất hiếm bài toán chỉ phụ thuộc vào một tiêu chí duy nhất; việc thuê nhà phụ thuộc đồng thời vào giá thuê, diện tích, mức độ tiện nghi, vị trí và nhiều yếu tố khác.

Các phương pháp MCDM phổ biến có thể kể đến như AHP (Analytic Hierarchy Process), ELECTRE, VIKOR và TOPSIS. Trong đó, TOPSIS được lựa chọn cho đề tài này vì tính đơn giản, trực quan và khả năng xử lý hiệu quả các tiêu chí dạng benefit/cost.

### 3.3 Thuật toán TOPSIS

TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) là một trong những phương pháp MCDM phổ biến nhất, được xây dựng dựa trên ý tưởng rằng phương án tối ưu phải gần nhất với phương án lý tưởng tốt nhất và xa nhất với phương án lý tưởng xấu nhất.

Trong đề tài, mỗi phòng trọ được xem như một phương án (alternative), còn các đặc trưng như giá, diện tích, tiện ích, khoảng cách... là các tiêu chí đánh giá (criteria). TOPSIS chuyển các tiêu chí này

về dạng số, chuẩn hóa chúng và tính ra một điểm số cuối cùng (topsis\_score). Quy trình TOPSIS gồm các bước sau:

(1) Xây dựng ma trận quyết định: Giả sử có  $n$  phòng trọ và  $m$  tiêu chí, ta xây dựng ma trận:

$$X = [x_{ij}]_{n \times m}$$

Trong đó:

$x_{ij}$  là giá trị tiêu chí thứ  $j$  của phòng trọ thứ  $i$

(2) Chuẩn hóa ma trận: Mục đích để đưa các tiêu chí về cùng thang đo:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^n x_{kj}^2}}$$

(3) Áp dụng trọng số: Với vector trọng số chuẩn hóa  $w = \{w_1, w_2, \dots, w_m\}$ :

$$v_{ij} = w_j \cdot r_{ij}$$

(4) Xác định phương án lý tưởng tốt nhất và xấu nhất

Phương án lý tưởng tốt nhất (Ideal Best):

$$A^+ = \{v_1^+, v_2^+, \dots, v_m^+\}$$

$$v_j^+ = \begin{cases} \max(v_{ij}) & j = benefit \\ \min(v_{ij}) & j = cost \end{cases}$$

Phương án lý tưởng xấu nhất (Ideal Worst):

$$A^- = \{v_1^-, v_2^-, \dots, v_m^-\}$$

$$v_j^- = \begin{cases} \min(v_{ij}) & j = benefit \\ \max(v_{ij}) & j = cost \end{cases}$$

(5) Tính khoảng cách đến giải pháp lý tưởng

Khoảng cách tới phương án tốt nhất:

$$S_i^+ = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^+)^2}$$

Khoảng cách tới phương án xâu nhất:

$$S_i^- = \sqrt{\sum_{j=1}^m (v_{ij} - v_j^-)^2}$$

(6) Tính điểm xếp hạng TOPSIS

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-}$$

Trong đó:

$$0 \leq C_i \leq 1$$

$C_i$  càng lớn  $\Rightarrow$  phòng trọ càng gần với phương án lý tưởng  $\Rightarrow$  xếp hạng càng cao

# CHƯƠNG 4. THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG

## 4.1 Thiết kế hệ thống

### 4.1.1 Kiến trúc tổng quan hệ thống

Kiến trúc tổng quan hệ thống này được thiết kế theo kiến trúc nhiều lớp, tương ứng với chuỗi xử lý từ thu thập dữ liệu đến cung cấp khuyến nghị cho người dùng.

(1) Lớp dữ liệu (Data layer) lưu trữ các tập dữ liệu gốc và các phiên bản đã xử lý; các file dữ liệu có thể được mount vào container hoặc lưu dưới dạng CSV/JSON phục vụ cho phân tích và huấn luyện.

(2) Lớp xử lý và mô hình (Processing & Model layer) phần lớn được phát triển bằng Jupyter Notebook và Python, nơi tiến hành khám phá dữ liệu, tiền xử lý, huấn luyện mô hình và đánh giá kết quả.

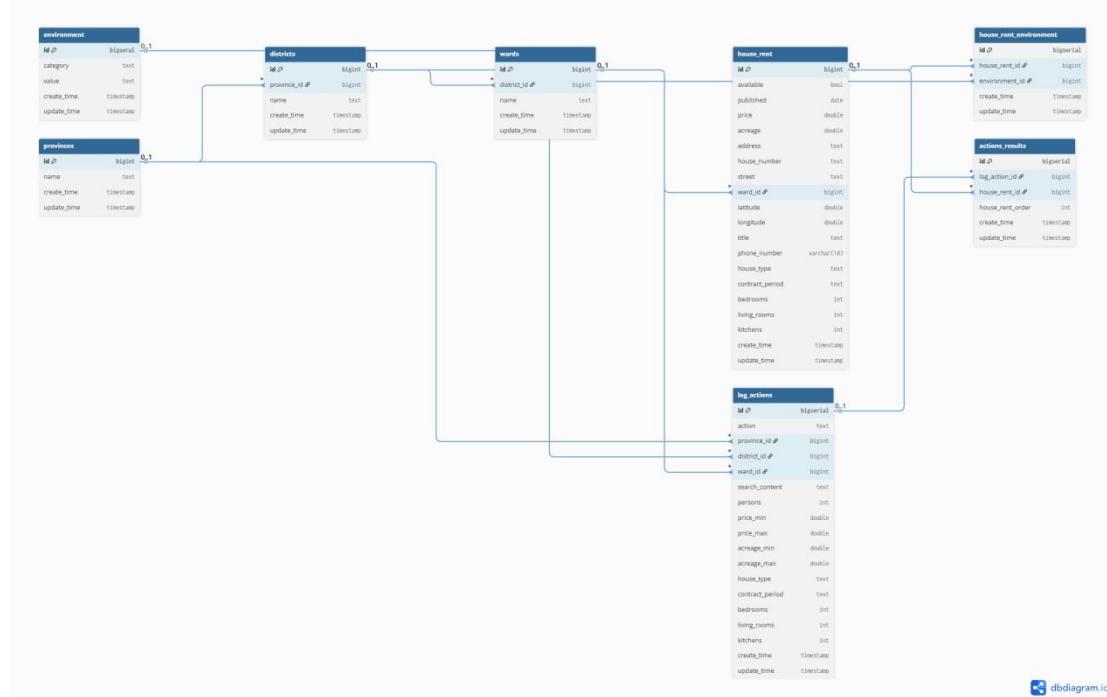
(3) Lớp dịch vụ (API layer) triển khai các endpoint để phục vụ mô hình và dữ liệu cho giao diện và thực hiện dưới dạng một service backend.

(4) Lớp giao diện người dùng (Frontend) sử dụng JavaScript, HTML, CSS để hiển thị giao diện tìm kiếm, lọc, danh sách và so sánh chỗ ở.

Toàn bộ các thành phần được đóng gói và phối hợp bằng Docker Compose, cho phép triển khai đồng bộ backend, frontend và các dịch vụ phụ trợ.

### 4.1.2 Thiết kế cơ sở dữ liệu

Hệ thống sử dụng PostgreSQL làm hệ quản trị cơ sở dữ liệu, có cơ sở dữ liệu chính: system. Chi tiết thiết kế cơ sở dữ liệu “system” như sau:



### Hình 3. Sơ đồ thiết kế cơ sở dữ liệu

Database “system” có các bảng sau: provinces, districts, wards, house\_rent, environment, house\_rent\_environment, log\_actions, action\_results. Các bảng được mô tả chi tiết như sau:

Bảng “provinces”: Thông tin về tỉnh / thành. Chi tiết như sau:

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Bigint	Mã tỉnh / thành	Khóa chính
name	Text	Tên tỉnh / thành	
create_time	Timestamp	Thời gian tạo bản ghi	
update_time	Timestamp	Thời gian cập nhật bản ghi	

Bảng 12. Bảng mô tả dữ liệu của tỉnh / thành

Bảng “districts”: Thông tin về quận / huyện. Chi tiết như sau:

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Bigint	Mã quận / huyện	Khóa chính
province_id	Bigint	Mã tỉnh / thành	Khóa ngoại với id trong bảng “provinces”
name	Text	Tên quận / huyện	
create_time	Timestamp	Thời gian tạo bản ghi	
update_time	Timestamp	Thời gian cập nhật bản ghi	

Bảng 13. Bảng mô tả dữ liệu của quận / huyện

Bảng “wards”: Thông tin về phường / xã. Chi tiết như sau:

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Bigint	Mã phường / xã	Khóa chính
district_id	Bigint	Mã quận / huyện	Khóa ngoại với id trong bảng “districts”
name	Text	Tên phường / xã	
create_time	Timestamp	Thời gian tạo bản ghi	
update_time	Timestamp	Thời gian cập nhật bản ghi	

Bảng 14. Bảng mô tả dữ liệu của phường / xã

Bảng “house\_rent”: Thông tin nhà thuê về giá, diện tích, địa chỉ, mô tả. Chi tiết như sau:

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Bigint		Khóa chính
available	Boolean	Trạng thái	True: còn. False: Không còn
published	Date	Ngày đăng tin	
price	Double	Giá nhà	Tính theo tháng. Đơn vị: triệu đồng
acreage	Double	Diện tích	Đơn vị: m <sup>2</sup>
address	Text	Địa chỉ chi tiết	
house_number	Text	Số nhà	
street	Text	Đường / Phố	
ward_id	Bigint	Mã phường / xã	Khóa ngoại với id trong bảng “ward”

latitude	Double	Vĩ độ	
longitude	Double	Kinh độ	
title	Text	Thông tin mô tả	
phone_number	Varchar(10)	Số điện thoại liên hệ	
house_type	Text	Kiểu nhà	Có 4 giá trị: CĂN HỘ DỊCH VỤ, CHUNG CƯ MINI, PHÒNG TRỌ, NHÀ NGUYÊN CĂN
contract_period	Text	Thời hạn hợp đồng	Có 2 giá trị: 6 THÁNG, 12 THÁNG
bedrooms	Int	Số phòng ngủ	
living_room	Int	Số phòng khác	
kitchens	Int	Số nhà bếp	
create_time	Timestamp	Thời gian tạo bản ghi	
update_time	Timestamp	Thời gian cập nhật bản ghi	

Bảng 15. Bảng mô tả thông tin của nhà thuê

Bảng “environment”: Thông tin tham chiếu của các điều kiện sống.  
Chi tiết như sau:

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Bigserial		Khóa chính
category	Text	Kiểu điều kiện sống	Có 3 loại giá trị: TIỆN NGHI; AN NINH; CHU

			KỲ ĐÓNG TIỀN
value	Text	Giá trị của điều kiện sống	Có danh sách các giá trị như: KHÉP KÍN, ĐỦ ĐỒ, CÓ CAMERA ...
create_time	Timestamp	Thời gian tạo bản ghi	
update_time	Timestamp	Thời gian cập nhật bản ghi	

Bảng 16. Bảng mô tả dữ liệu tham chiếu theo điều kiện sống

Bảng “house\_rent\_environment”: Thông tin về danh sách các điều kiện sống của mỗi nhà cho thuê. Chi tiết như sau:

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Bigserial		Khóa chính
house_rent_id	Bigint	id của house_rent	Khóa ngoại với id trong bảng “house_rent”
environment_id	Bigint	id của environment	Khóa ngoại với id trong bảng “environment”
create_time	Timestamp	Thời gian tạo bản ghi	
update_time	Timestamp	Thời gian cập nhật bản ghi	

Bảng 17. Bảng mô tả dữ liệu các điều kiện sống của nhà cho thuê

Bảng “log\_actions”: Thông tin về hành động (tìm kiếm, lựa chọn) của người dùng. Chi tiết như sau:

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
---------	--------------	-------	---------

<b>id</b>	Bigserial		Khóa chính
<b>action</b>	Text	Hành động	Có 2 giá trị: SEARCH, SELECT
<b>province_id</b>	Bigint	Mã tỉnh / thành	Khóa ngoại với id trong bảng “provinces”
<b>district_id</b>	Bigint	Mã quận / huyện	Khóa ngoại với id trong bảng “districts”
<b>ward_id</b>	Bigint	Mã phường / xã	Khóa ngoại với id trong bảng “wards”
<b>search_content</b>	Text	Nội dung tìm kiếm	
<b>persons</b>	Int	Số người ở	
<b>price_min</b>	Double	Lọc giá thấp nhất	Tính theo tháng. Đơn vị: triệu đồng
<b>price_max</b>	Double	Lọc giá cao nhất	Tính theo tháng. Đơn vị: triệu đồng
<b>acreage_min</b>	Double	Lọc diện tích thấp nhất	Đơn vị: m <sup>2</sup>
<b>acreage_max</b>	Double	Lọc diện tích cao nhất	Đơn vị: m <sup>2</sup>
<b>house_type</b>	Text	Kiểu nhà	Có 4 giá trị: CĂN HỘ DỊCH VỤ, CHUNG CƯ MINI, PHÒNG TRỌ, NHÀ NGUYÊN CĂN

contract_period	Text	Thời hạn hợp đồng	Có 2 giá trị: 6 THÁNG, 12 THÁNG
bedrooms	Int	Số phòng ngủ	
living_roomg	Int	Số phòng khác	
kitchens	Int	Số nhà bếp	
create_time	Timestamp	Thời gian tạo bản ghi	
update_time	Timestamp	Thời gian cập nhật bản ghi	

Bảng 18. Bảng mô tả dữ liệu nhật ký người dùng

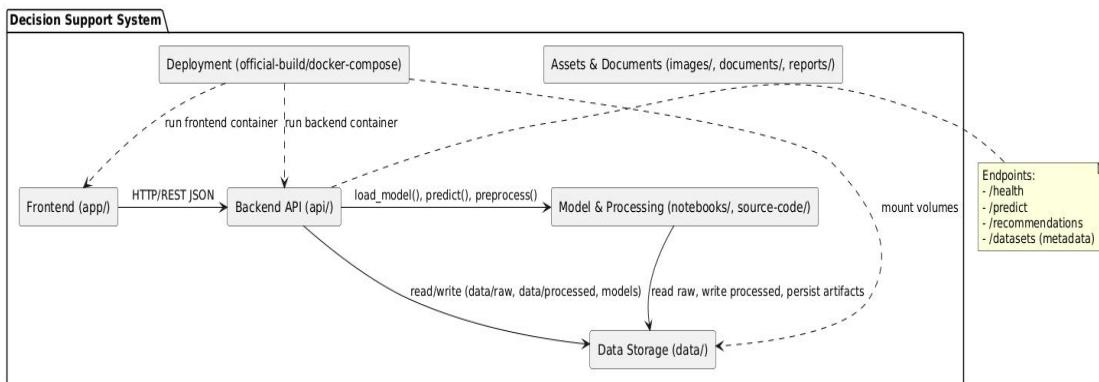
Bảng “actions\_results”: Thông tin về danh sách nhà cho thuê trả về theo hành động (tìm kiếm, lựa chọn) của người dùng. Chi tiết như sau:

Tên cột	Kiểu dữ liệu	Mô tả	Ghi chú
id	Bigserial		Khóa chính
log_action_id	Bigint	id của log_action	Khóa ngoại với id trong bảng “log_action”
house_rent_id	Bigint	id của house_rent	Khóa ngoại với id trong bảng “house_rent”
house_rent_order	Int	Thứ tự trả về của nhà cho thuê	
create_time	Timestamp	Thời gian tạo bản ghi	
update_time	Timestamp	Thời gian cập nhật bản ghi	

Bảng 19. Bảng mô tả dữ liệu kết quả tìm kiếm

### 4.1.3 Thiết kế chương trình

Phần này mô tả thiết kế chương trình của hệ hỗ trợ quyết định này dựa trên mã nguồn và cấu trúc thư mục trong repository<sup>1</sup>. Mục tiêu của thiết kế là cung cấp một nền tảng dễ bảo trì, dễ mở rộng và có khả năng chuyển đổi mượt từ môi trường nghiên cứu (Jupyter Notebook) sang môi trường dịch vụ (API/production). Thiết kế nhấn mạnh vào việc phân tách rõ trách nhiệm giữa các lớp chức năng, lưu trữ để đảm bảo khả năng tái tạo kết quả, và cấu hình hóa hệ thống qua biến môi trường hoặc file cấu hình nhằm thuận tiện cho việc triển khai.



Hình 4. Cấu trúc mã nguồn của chương trình

Mục tiêu thiết kế và các nguyên tắc chủ đạo được áp dụng bao gồm:

- (1) Module hoá logic, tách rõ các trách nhiệm như Data I/O, Preprocessing, Feature Engineering, Model Inference, API adapter và Frontend rendering để mỗi phần có thể phát triển độc lập;
- (2) Idempotence & Reproducibility: mọi bước tiền xử lý và định dạng đầu vào phải có thể tái tạo, lưu cùng với model;
- (3) Configuration-driven: tham số môi trường, đường dẫn dữ liệu và tham số mô hình được tách ra khỏi mã nguồn, sử dụng env vars hoặc file config;
- (4) Logging & Audit: mọi dự đoán và thao tác quan trọng được ghi log kèm metadata để phục vụ giám sát và truy vết.

Cấu trúc module để xuất phản ánh trực tiếp thư mục trong mã nguồn và được viết dưới dạng các thành phần rõ ràng. Phần frontend (app/) chịu trách nhiệm giao diện người dùng: cung cấp form tìm kiếm/ lọc, bảng kết quả, trang chi tiết và chế độ so sánh. Frontend giao tiếp với backend qua HTTP/REST, nhận JSON kết quả, hiển thị danh sách. Phần backend (api/) triển khai service REST (FastAPI) và expose các endpoint chính như /health, /predict, /recommendations, /datasets; backend đóng vai trò adapter giữa request HTTP và các module xử lý/model, chịu trách nhiệm validate đầu vào, tải artifacts, chạy pipeline inference, tạo output và ghi log. Thư mục source-code/ chứa các module lõi đã được tách từ notebook:

data.loader, data.persistence, preprocessing.pipeline, features, modeling và explainability; đây là điểm nối giữa công việc nghiên cứu (notebooks/) và dịch vụ (api/). Thư mục notebooks/ giữ các tập notebook EDA và prototype; khi logic ổn định, cần refactor thành module trong source-code/. Thư mục data/ giữ cấu trúc lưu trữ rõ ràng (data/raw, data/processed, data/features, models) để dễ mount volume trong Docker. Cuối cùng, official-build/ chứa Dockerfile và docker-compose.yml để định nghĩa stack triển khai backend, frontend và phụ trợ.

## 4.2 Triển khai hệ thống

### 4.2.1 Thu thập và tiền xử lý dữ liệu

Dữ liệu đầu vào cho hệ thống được thu thập từ một bộ dataset trên Kaggle<sup>ii</sup> và được lưu trữ nguyên bản trong thư mục data/raw/ của mã nguồn; mỗi lần nạp dữ liệu đều được ghi lại với metadata cơ bản (nguồn, thời điểm import, số bản ghi) để đảm bảo truy vết.

Công tác khám phá dữ liệu (EDA) được thực hiện bằng Jupyter Notebook trong thư mục notebooks/, nơi tiến hành kiểm tra schema, thống kê mô tả, phân bố biến, phát hiện giá trị thiếu và kiểm tra nhất quán định dạng (thông tin nhà cho thuê, định dạng địa chỉ). Từ kết quả EDA, các bước tiền xử lý được xác định và thực thi: loại bỏ bản ghi trùng lặp, làm sạch các trường ký tự (loại bỏ ký tự không mong muốn, chuẩn hóa địa chỉ), xử lý missing values bằng phương pháp thông thường và xử lý outlier cơ bản.

### 4.2.2 Phương pháp tính điểm dựa trên topsis

Gồm có 4 bước chính như sau:

(1) **Thu Thập và Chuẩn Bị Dữ Liệu:** Đầu tiên, hàm *compare* nhận vào một yêu cầu (*CompareRequest*) chứa các thông tin quan trọng: *house\_rent\_ids*: Danh sách ID của các phòng trọ bạn đã chọn để so sánh. *amenities* và *weights*: Các tiện ích bạn quan tâm và trọng số (mức độ quan trọng) của từng tiện ích đó. *topsis\_weight*: Trọng số cho các tiêu chí chính của TOPSIS (Giá, Diện tích, Tiện ích). *prefer\_location*: Tọa độ vị trí bạn ưu tiên (ví dụ: trường học, công ty). Nếu không cung cấp, hệ thống sẽ lấy một vị trí mặc định ở trung tâm Hà Nội.

Sau đó, hệ thống: Lấy thông tin chi tiết (giá, diện tích, địa chỉ, danh sách tiện ích có sẵn, tọa độ,...) của các phòng trọ đã chọn từ cơ sở dữ liệu. Chuyển đổi dữ liệu này thành một cấu trúc dạng bảng (DataFrame) để dễ dàng tính toán.

(2) **Vector Hóa Dữ Liệu** (Hàm *\_data\_vectorizer*): Đây là bước quan trọng để biến các thông tin thô thành những con số có ý nghĩa cho việc so

sánh. Hàm `_data_vectorizer` sẽ tạo ra các "thuộc tính" mới cho mỗi phòng trọ:

*price* (Giá): Giữ nguyên giá thuê. *acreage* (Diện tích): Giữ nguyên diện tích. *acreage\_ratio* (Tỷ lệ Diện tích/Giá): Một chỉ số thể hiện "độ đáng tiền" về mặt không gian. Tỷ lệ này càng cao, bạn càng có nhiều diện tích trên mỗi đồng chi trả. *amenities\_w* (Điểm tiện ích có trọng số): Hệ thống duyệt qua tất cả tiện ích có sẵn của một phòng trọ. Nếu một tiện ích có trong danh sách bạn quan tâm, điểm của tiện ích đó (do bạn đặt ra trong weights) sẽ được cộng vào tổng điểm. Kết quả là một con số duy nhất đại diện cho mức độ phù hợp về tiện ích của phòng trọ đó với yêu cầu của bạn. *amenities\_ratio* (Tỷ lệ Điểm tiện ích/Giá): Tương tự như *acreage\_ratio*, chỉ số này cho biết "độ đáng tiền" về mặt tiện ích. *distance\_to\_prefer\_location* (Khoảng cách đến vị trí ưu tiên): Tính khoảng cách "đường chim bay" từ phòng trọ đến vị trí bạn mong muốn.

Kết quả của bước này là một "ma trận quyết định" (*decision\_matrix*), trong đó mỗi hàng là một phòng trọ và mỗi cột là một trong 6 tiêu chí số hóa ở trên.

(3) **Áp Dụng Thuật Toán TOPSIS:** Đây là trái tim của quá trình so sánh.

Xác định Trọng số và Loại Tiêu chí: Trọng số (*topsis\_weights*): Hệ thống lấy các trọng số bạn đã thiết lập cho 3 nhóm tiêu chí chính (Giá, Diện tích, Tiện ích) và chuẩn hóa chúng. Nếu bạn không đặt, nó sẽ chia đều trọng số cho cả 6 tiêu chí. Loại tiêu chí (*criteria\_types*): Hệ thống xác định tiêu chí nào là "càng cao càng tốt" (benefit) và "càng thấp càng tốt" (cost). *cost* (càng thấp càng tốt): *price*, *distance\_to\_prefer\_location*. *benefit* (càng cao càng tốt): *acreage*, *acreage\_ratio*, *amenities\_w*, *amenities\_ratio*.

Thực thi TOPSIS: Lớp *TOPSIS* được khởi tạo với ma trận quyết định, trọng số, và loại tiêu chí. Thuật toán sẽ thực hiện các bước kinh điển của TOPSIS: a. Chuẩn hóa ma trận: Đưa tất cả các giá trị về cùng một thang đo. b. Nhân trọng số: Áp dụng trọng số vào ma trận đã chuẩn hóa. c. Tìm giải pháp lý tưởng tốt nhất (Ideal Best) và xấu nhất (Ideal Worst): Tìm ra một "phòng trọ ảo" tốt nhất (giá thấp nhất, diện tích lớn nhất,...) và một "phòng trọ ảo" xấu nhất từ dữ liệu. d. Tính khoảng cách: Tính khoảng cách từ mỗi phòng trọ thực tế đến "phòng trọ ảo" tốt nhất và xấu nhất. e. Tính điểm tương đối (*topsis\_score*): Dựa trên hai khoảng cách trên, tính ra một điểm số cuối cùng cho mỗi phòng trọ. Điểm này càng gần 1, phòng trọ đó càng gần với lựa chọn lý tưởng.

(4) **Trả Về Kết Quả:** Cuối cùng, hệ thống: Gán điểm *topsis\_score* vừa tính được cho mỗi phòng trọ. Sắp xếp danh sách các phòng trọ theo thứ tự *topsis\_score* từ cao đến thấp. Gán thứ hạng (*rank*) cho chúng. Bỏ

sung thêm thông tin về các tiện ích bạn yêu cầu mà phòng trọ đó đáp ứng (*matched\_amenities*).

Trả về một đối tượng *TopsisCompareResponse* bao gồm:  
*ranked\_houses*: Danh sách các phòng trọ đã được xếp hạng cùng với  
điểm số và thông tin chi tiết. *ideal\_best* và *ideal\_worst*: Các giá trị của  
"phòng trọ ảo" tốt nhất và xấu nhất, giúp bạn hiểu được đâu là ngưỡng tốt  
nhất/tệ nhất trong các lựa chọn của mình.

#### 4.2.3 Cài đặt hệ thống

Hệ thống được đóng gói và triển khai bằng Docker Compose; cấu  
hình triển khai nằm trong thư mục `official-build/` và lệnh khởi chạy được  
ghi trong `README` là: `cd official-build && docker-compose up -d --build`. Thực thi lệnh này đã dựng các container cần thiết cho dự án, bao  
gồm service `backend (api)`, `frontend (app)` và các volume mount cho dữ  
liệu và `models` — giúp tái tạo môi trường chạy đồng nhất giữa máy phát  
triển và môi trường thử nghiệm. Các thành phần dịch vụ giao tiếp qua  
`HTTP/REST`, `backend` cung cấp các endpoint chính như `/health` và  
`/predict` để kiểm tra trạng thái và thực hiện inference, trong khi `frontend`  
gọi các endpoint này để hiển thị kết quả cho người dùng.

Cấu hình môi trường được quản lý bằng biến môi trường (`env vars`)  
và file cấu hình nằm ngoài mã nguồn để dễ thay đổi khi triển khai; khi  
chạy bằng Docker Compose, các biến này được truyền vào qua file `env`  
hoặc trực tiếp trong `docker-compose.yml`, và các thư mục `data/` được  
mount vào container để giữ persistent storage. Backend được đóng gói  
với môi trường Python và được cấu hình để cài đặt các phụ thuộc từ  
`requirements`; frontend được build bằng công cụ JavaScript tương ứng và  
phục vụ các tài nguyên tĩnh. Jupyter Notebook để thực hiện EDA và huấn  
luyện vẫn được chạy tại môi trường phát triển (hoặc trong container khi  
cần) bằng lệnh `jupyter lab`; để tái chạy notebook một cách tự động, các  
notebook có thể được thực thi qua `nbconvert` trong pipeline.

Sau khi hệ thống được khởi chạy, các bước kiểm chứng chức năng đã  
được thực hiện: gọi endpoint `/health` để xác nhận backend sẵn sàng, thực  
hiện vài request mẫu tới `/predict` để kiểm tra luồng tiền xử lý → inference  
→ trả kết quả, và kiểm tra logs container để rà lỗi khởi tạo hoặc lỗi  
runtime. Dữ liệu được lưu ở volume gắn ngoài nhằm bảo toàn qua các lần  
khởi động lại container. Các tập tin cấu hình, lệnh chạy và script hỗ trợ  
được ghi rõ trong tài liệu trong mã nguồn để phục vụ việc lặp lại quy  
trình triển khai và chạy lại các bước huấn luyện/tiền xử lý khi cần.

## CHƯƠNG 5. KẾT QUẢ THỰC NGHIỆM

Chương sau trình bày về một số hình ảnh kết quả của hệ thống như sau:

Bộ lọc của hệ thống để người dùng chọn, nhập một số các thông tin về căn nhà cần cho thuê dựa trên các tiêu chí, để từ đó hệ thống trả lại kết quả tìm kiếm:

**Hệ Thống Hồ Trợ Tìm Kiếm Nhà Trọ**

Tỉnh/Thành phố	Quận/Huyện	Phường/Xã	Giá tối thiểu (triệu)	Giá tối đa (triệu)
HÀ NỘI	CẦU GIÁY	NGHĨA ĐÔ	3	7
Diện tích tối thiểu (m <sup>2</sup> )	Diện tích tối đa (m <sup>2</sup> )	Loại hình	Hợp đồng	
20	50	CHUNG CƯ MINI	6 THÁNG	
Tim Kiếm				

<p><b>CHÍNH CHỦ CHO THUÊ CC MINI CAO CẤP MỚI TOANH, FULL ĐỒ, GIÁ: 3TR</b></p> <p><b>Địa chỉ:</b> Đường Nguyễn Đình Hoàn, Phường Nghĩa Đô, Quận Cầu Giấy, Hà Nội</p> <p><b>Loại hình:</b> CHUNG CƯ MINI</p> <p><b>Hợp đồng:</b> 6 THÁNG</p> <p><b>Liên hệ:</b> 0956370939</p> <p><b>TIỆN NGHỊ:</b> ĐÚ ĐỒ TIỆN NGHỊ: MỚI</p> <p><b>4 triệu/tháng - 20 m<sup>2</sup></b></p>	<p><b>Cho thuê phòng trọ dạng chung cư mini gần chợ bưởi (gần cây đa hoàng quốc việt)</b></p> <p><b>Địa chỉ:</b> 34 ngõ 96 Võ Chí Công, Phường Nghĩa Đô, Quận Cầu Giấy, Hà Nội</p> <p><b>Loại hình:</b> CHUNG CƯ MINI</p> <p><b>Hợp đồng:</b> 6 THÁNG</p> <p><b>Liên hệ:</b> 0380722091</p> <p><b>4.2 triệu/tháng - 22 m<sup>2</sup></b></p>	<p><b>Cho thuê phòng trọ dạng chung cư mini gần chợ bưởi (gần cây đa hoàng quốc việt),</b></p> <p><b>Địa chỉ:</b> 34 ngõ 96 Đường Võ Chí Công, Phường Nghĩa Đô, Quận Cầu Giấy, Hà Nội</p> <p><b>Loại hình:</b> CHUNG CƯ MINI</p> <p><b>Hợp đồng:</b> 6 THÁNG</p> <p><b>Liên hệ:</b> 0807996644</p> <p><b>4.2 triệu/tháng - 22 m<sup>2</sup></b></p>
---	--	---

Hình 5. Bộ lọc của hệ thống trợ giúp quyết định

Khách hàng lựa chọn căn nhà muốn so sánh, sau đó sẽ tiếp tục sang màn hình so sánh các căn nhà. Màn hình so sánh hiện thị bảng so sánh, phương án tốt nhất, phương án tệ nhất. Điểm topsis được tính toán chính là khoảng cách của phương án tệ nhất đến phương án tốt nhất. Chi tiết như sau:

[« Quay lại trang tìm kiếm](#)

### So Sánh các lựa chọn

**Chọn tiêu chí & trọng số**

<input checked="" type="checkbox"/> TIỆN NGHI: ĐỦ ĐÓ	50
<input checked="" type="checkbox"/> AN NINH: CÓ CAMERA	50
<input checked="" type="checkbox"/> TIỆN NGHI: NÓNG LẠNH	50
<input checked="" type="checkbox"/> TIỆN NGHI: GIƯỜNG TÙ	50
<input type="checkbox"/> TIỆN NGHI: ĐẸP	50
<input type="checkbox"/> TIỆN NGHI: ĐIỀU HÒA	50

[Chọn vị trí ưu tiên](#)

Hình 6. Lựa chọn các tiêu chí để tính điểm topsis

**So sánh**

**Bảng Tiêu Chí Lý Tưởng**

Tiêu chí	Lý tưởng (Tốt nhất)	Tốt nhất
Giá (triệu)	4.00	7.00
Diện tích (m <sup>2</sup> )	40.00	20.00
Tỉ lệ diện tích/giá	6.491	5.000
Điểm tiện ích	150.000	0.000
Tỉ lệ tiện ích/giá	31.915	0.000
Điểm vị trí (km)	0.000	0.127

Hình 7. Bảng tiêu chí lý tưởng

Bảng Xếp Hạng Chi Tiết

Hạng	Tiêu đề	Giá (triệu)	Diện tích (m <sup>2</sup> )	Điểm Điện Tích/Giá	Điểm Tiện Ích	Điểm Tiện Ích/Giá	Điểm Vị Trí (km)	Điểm TOPSIS	Tiện ích phù hợp
1	Phòng trọ khép kín đủ đồ, điều hòa, không chung chủ 119 Âu Cơ, Tây Hồ	4.70	27.00	5.745	150.000	31.915	0.045	0.8040	TIỀN NGHỊ: KHÉP KÍN TIỀN NGHỊ: ĐỦ ĐỒ TIỀN NGHỊ: KHÔNG CHUNG CHỦ
2	Phòng trọ đầy đủ đồ, điện nước giá dân cực rẻ	4.00	20.00	5.000	100.000	25.000	0.045	0.6665	TIỀN NGHỊ: ĐỦ ĐỒ TIỀN NGHỊ: ĐIỆN NƯỚC GIÁ DÂN
3	Cho nữ thuê phòng khép kín, đủ nội thất, nóng lạnh, điều hòa	4.20	22.00	5.238	100.000	23.810	0.045	0.6627	TIỀN NGHỊ: KHÉP KÍN TIỀN NGHỊ: NÓNG LẠNH
4	Phòng cho thuê mới sửa đầy đủ tiện nghi Âu Cơ- Nghĩ Tâm	5.00	30.00	6.000	100.000	20.000	0.045	0.6421	TIỀN NGHỊ: ĐỦ ĐỒ TIỀN NGHỊ: MỚI
5	LẠC LONG QUÂN - KHÉP KÍN FULL ĐỒ- CHÍNH CHỦ	5.70	37.00	6.491	100.000	17.544	0.045	0.6211	TIỀN NGHỊ: KHÉP KÍN TIỀN NGHỊ: ĐỦ ĐỒ

Hình 8. Bảng xếp hạng chi tiết

6	nghị hệ thống phòng rộng rãi từ 12m2, 15m2.	5.00	30.00	6.000	100.000	20.000	0.127	0.4823	TIỀN NGHỊ: ĐỦ ĐỒ
7	Cho thuê phòng trọ 20m2 đầy đủ tiện nghi	4.00	20.00	5.000	50.000	12.500	0.045	0.4437	TIỀN NGHỊ: ĐỦ ĐỒ
<b>Phòng trọ khép kín đủ đồ, điều hòa, không chung chủ 119 Âu Cơ, Tây Hồ</b>		<b>Phòng trọ đầy đủ đồ, điện nước giá dân cực rẻ</b>				<b>Cho nữ thuê phòng khép kín, đủ nội thất, nóng lạnh, điều hòa</b>			
Địa chỉ: 12A ngõ 119 Đường Âu Cơ, Phường Yên Phụ, Quận Tây Hồ, Hà Nội		Địa chỉ: đường Thụy Khuê, Phường Thụy Khuê, Quận Tây Hồ, Hà Nội				Địa chỉ: 26 ngách 23 ngõ 38 xuân la Đường Xuân La, Phường Xuân La, Quận Tây Hồ, Hà Nội			
Loại hình: PHÒNG TRỌ		Loại hình: PHÒNG TRỌ				Loại hình: PHÒNG TRỌ			
Hợp đồng: 6 THÁNG		Hợp đồng: 6 THÁNG				Hợp đồng: 6 THÁNG			
Liên hệ: 0270918984		Liên hệ: 0535132259				Liên hệ: 0367957292			
TIỀN NGHỊ: KHÉP KÍN TIỀN NGHỊ: ĐỦ ĐỒ TIỀN NGHỊ: KHÔNG CHUNG CHỦ		TIỀN NGHỊ: ĐIỆN NƯỚC GIÁ DÂN				TIỀN NGHỊ: KHÉP KÍN TIỀN NGHỊ: ĐIỆU HÒA TIỀN NGHỊ: NÓNG LẠNH			
4.7 triệu/tháng - 27 m <sup>2</sup>		4 triệu/tháng - 20 m <sup>2</sup>				4.2 triệu/tháng - 22 m <sup>2</sup>			
<b>Phòng cho thuê mới sửa đầy đủ tiện nghi Âu Cơ- Nghĩ Tâm</b>		<b>LẠC LONG QUÂN - KHÉP KÍN FULL ĐỒ- CHÍNH CHỦ</b>				<b>Phòng trọ khép kín đầy đủ tiện nghi hệ thống phòng rộng rãi từ 12m2, 15m2.</b>			
Địa chỉ: Đường Âu Cơ, Phường Quảng An, Quận Tây Hồ, Hà Nội		Địa chỉ: ngõ 603 Đường Lạc Long Quân, Phường Xuân La, Quận Tây Hồ, Hà Nội				Địa chỉ: khu đô thị hut Đường số 70, Xã Văn Canh, Huyện Hoài Đức, Hà Nội			
Loại hình: PHÒNG TRỌ		Loại hình: PHÒNG TRỌ				Loại hình: PHÒNG TRỌ			
Hợp đồng: 6 THÁNG		Hợp đồng: 6 THÁNG				Hợp đồng: 6 THÁNG			
Liên hệ: 0747756483		Liên hệ: 0937142876				Liên hệ: 0205186592			

Hình 9. Kết quả trả về dựa trên điểm topsis

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Trên cơ sở triển khai hệ thống mẫu, nghiên cứu khẳng định rằng việc ứng dụng phương pháp TOPSIS trong một DSS cho người thuê nhà là khả thi và đem lại lợi ích thực tiễn: TOPSIS cho phép tổng hợp nhiều tiêu chí định lượng và định tính thành một chỉ số tương đồng duy nhất, hỗ trợ việc xếp hạng nhanh chóng và minh bạch các lựa chọn. Việc chuẩn hóa dữ liệu và tổ chức theo cấu trúc rõ ràng là yếu tố nền tảng quyết định đến độ tin cậy của kết quả; khi dữ liệu đầy đủ và nhất quán, mô-đun MCDM cho kết quả có tính diễn giải cao và hữu ích cho người dùng. Giao diện đơn giản với khả năng nhập trọng số cá nhân hóa làm tăng tính phù hợp của đề xuất với nhu cầu thực tế của từng người thuê.

Tuy nhiên, nghiên cứu cũng nêu rõ những hạn chế hiện tại: hệ thống mẫu có phạm vi dữ liệu còn hạn chế và chưa xử lý triệt để các tiêu chí phi cấu trúc (mô tả văn bản, cảm nhận chủ quan), chưa có cơ chế cập nhật theo thời gian thực và chưa đánh giá sâu về độ nhạy giữa trọng số và thứ tự xếp hạng. Ngoài ra, TOPSIS mặc dù nhanh và dễ triển khai nhưng không phải luôn là phương pháp tối ưu cho mọi cấu hình tiêu chí; do đó cần so sánh và đánh giá với các phương pháp MCDM khác trước khi áp dụng rộng rãi.

### 6.2 Hướng phát triển

Ở ngắn hạn, cần mở rộng và làm phong phú nguồn dữ liệu bằng cách thu thập thêm mẫu từ nhiều khu vực và loại nhà khác nhau, tự động hóa quy trình làm sạch và cập nhật dữ liệu (qua API hoặc web-scraping), đồng thời hoàn thiện giao diện người dùng: cung cấp công cụ nhập trọng số trực quan (slider, so sánh cặp) và tài liệu hướng dẫn giúp người dùng xác định ưu tiên hợp lý. Song song đó cần tổ chức thử nghiệm người dùng (user testing) để thu thập phản hồi về tính hữu dụng, khả năng hiểu kết quả và trải nghiệm giao diện.

Ở dài hạn, định hướng ứng dụng thực tế bao gồm tích hợp chức năng bản đồ và phân tích không gian (geocoding, đánh giá khoảng cách đến tiện ích, phân tích thời gian di chuyển), hỗ trợ cập nhật thời gian thực từ các nền tảng đăng tin và thêm chức năng tương tác giữa người thuê và chủ nhà. Cần xây dựng ứng dụng di động, hệ thống cảnh báo thay đổi giá và tính toán chi phí tổng sở hữu, đồng thời đảm bảo các yếu tố bảo mật và tuân thủ pháp lý về dữ liệu cá nhân. Cuối cùng, triển khai thí điểm với bộ người dùng thực để thu thập các chỉ số KPI (độ hài lòng, thời gian tìm nhà, tỷ lệ chuyển đổi thuê, độ ổn định của xếp hạng) sẽ là bước cần thiết để đánh giá hiệu quả và chuẩn bị cho các phương án mở rộng hoặc kinh doanh hóa sản phẩm.

## TÀI LIỆU THAM KHẢO

---

<sup>i</sup> Mã nguồn của báo cáo [online]: [https://github.com/Tran-Ngoc-Bao/Decision\\_Support\\_System](https://github.com/Tran-Ngoc-Bao/Decision_Support_System)

<sup>ii</sup> Bộ dữ liệu gốc [online]:  
<https://www.kaggle.com/datasets/vanviethieuanh/vietnam-house-rent-dataset?select=hn.csv>