

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	6
1.1 Đặt vấn đề	6
1.2 Mục tiêu và phạm vi đề tài	6
1.3 Định hướng giải pháp	6
1.4 Bố cục báo cáo	7
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU	8
2.1 Khảo sát hiện trạng	8
2.2 Yêu cầu chức năng	8
2.2.1 Biểu đồ use case	8
2.2.2 Lưu đồ ảnh hưởng	14
2.3 Yêu cầu phi chức năng	17
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG	18
CHƯƠNG 4. THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG	19
4.1 Thiết kế hệ thống	19
4.1.1 Kiến trúc tổng quan hệ thống	19
4.1.2 Thiết kế cơ sở dữ liệu	19
4.1.3 Thiết kế chương trình	26
4.2 Triển khai hệ thống	27
4.2.1 Thu thập và tiền xử lý dữ liệu	27
4.2.2 Cài đặt hệ thống	27
CHƯƠNG 5. KẾT QUẢ THỰC NGHIỆM	29
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	9
Hình 2 . Lưu đồ ảnh hưởng của hệ thống	17
Hình 3 . Sơ đồ thiết kế cơ sở dữ liệu	20
Hình 4 . Cấu trúc mã nguồn của chương trình	26

DANH MỤC BẢNG BIỂU

Bảng 1 . Đặc tả use case Nhập/Nạp dữ liệu	10
Bảng 2 . Đặc tả use case Quản lý dữ liệu	10
Bảng 3 . Đặc tả use case Tiền xử lý dữ liệu	11
Bảng 4 . Đặc tả use case Phân tích / Chạy mô hình	11
Bảng 5 . Đặc tả use case Chạy kịch bản & So sánh	12
Bảng 6 . Đặc tả use case Xem khuyến nghị / Gợi ý	13
Bảng 7 . Đặc tả use case Cấu hình hệ thống	13
Bảng 8 . Đặc tả use case Xem nhật ký / Kiểm toán	14
Bảng 9 . Các yếu tố ảnh hưởng đến quyết định chọn nhà	15
Bảng 10 . Các yếu tố hệ thống không thể dự đoán được ảnh hưởng tới quyết định của khách hàng	15
Bảng 11 . Các biến trung gian được sử dụng trong trợ giúp quyết định ..	16
Bảng 12 . Bảng mô tả dữ liệu của tỉnh / thành	20
Bảng 13 . Bảng mô tả dữ liệu của quận / huyện	20
Bảng 14 . Bảng mô tả dữ liệu của phường / xã	21
Bảng 15 . Bảng mô tả thông tin của nhà thuê	22
Bảng 16 . Bảng mô tả dữ liệu tham chiếu theo điều kiện sống	23
Bảng 17 . Bảng mô tả dữ liệu các điều kiện sông của nhà cho thuê	23
Bảng 18 . Bảng mô tả dữ liệu nhật ký người dùng	25
Bảng 19 . Bảng mô tả dữ liệu kết quả tìm kiếm	25

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 thành 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 Bố 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.

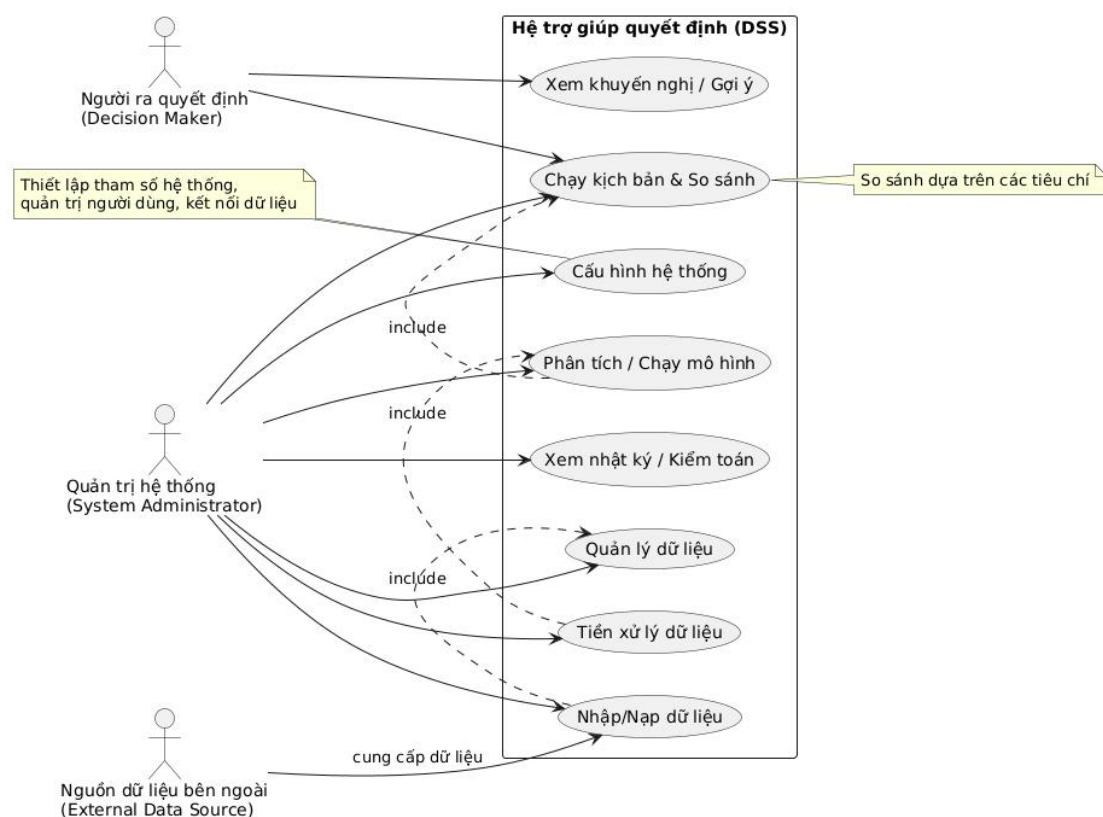
Trong giai đoạn so sánh và đánh giá, phần lớn người dùng phải tự ghi chép hoặc lưu thủ công các lựa chọn để đối chiếu các tiêu chí như giá thuê, diện tích, vị trí và tiện nghi. Việc không có công cụ hỗ trợ phân tích làm cho quá trình ra quyết định dễ mang tính cảm tính, đặc biệt là với những người chưa 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”:

Tên	Nhập/Nạp dữ liệu
Diễn giải ngắn	Hệ thống nhập dữ liệu từ nguồn file/DB/External API
Actors	Quản trị hệ thống, Nguồn dữ liệu bên ngoài
Preconditions	
Postconditions	Dữ liệu mới được lưu vào kho dữ liệu DSS (raw), ghi nhận meta-data và lịch sử import
Luồng chính	<ol style="list-style-type: none"> 1. Chọn nguồn: file CSV/Excel, DB, API bên ngoài 2. Hệ thống kết nối/nạp dữ liệu, hiển thị preview dữ liệu 3. Hệ thống lưu dữ liệu thô và ghi log import
Luồng thay thế	Nếu kết nối thất bại: báo lỗi kết nối

Yêu cầu phi chức năng	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”:

Tên	Quản lý dữ liệu
Diễn giải ngắn	Các tác vụ quản lý dữ liệu (xem, chỉnh sửa meta, xóa, phiên bản)
Actors	Quản trị hệ thống
Preconditions	Dữ liệu đã có trong hệ thống
Postconditions	Thay đổi quản lý dữ liệu được lưu; phiên bản dữ liệu cập nhật
Luồng chính	<ol style="list-style-type: none"> 1. Admin truy cập danh sách dataset. 2. Chọn dataset để xem/sửa/meta. 3. Thực hiện thao tác (đổi tên, mô tả, xóa, phục hồi). 4. Hệ thống lưu thay đổi và ghi lịch sử phiên bản.
Luồng thay thế	
Yêu cầu phi chức năng	

Bảng 2. Đặc tả use case Quản lý dữ liệu

Use case “Tiền xử lý dữ liệu”:

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

	pipeline tiền xử lý
Luồng chính	1. Hệ thống thực hiện từng bước hoặc preview transformation. 2. Admin lưu pipeline để tái sử dụng.
Luồng thay thế	Preview hiển thị distributon; admin điều chỉnh tham số
Yêu cầu phi chức năng	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”:

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

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

Use case “Chạy kịch bản & So sánh”:

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

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

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

Use case “Xem khuyến nghị / Gợi ý”:

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

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

Use case “Cấu hình hệ thống”:

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

Bảng 7. Đặc tả use case Cấu hình hệ thống

Use case “Xem nhật ký / Kiểm toán”:

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

Luồng thay thế	
Yêu cầu phi chức năng	

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)	(L_at, L_on) vị trí ưu tiên, vị trí mà người dùng muốn tìm trọ xung quanh	Chẳng hạn 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	nguyên	động theo thỏa thuận
(U_w)	Trọng số tiêu chí của người dùng	Mức độ ưu tiên các tiêu chí của người dùng	Vector trọng số	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	Thường dựa vào chủ quan chi tiêu của người dùng

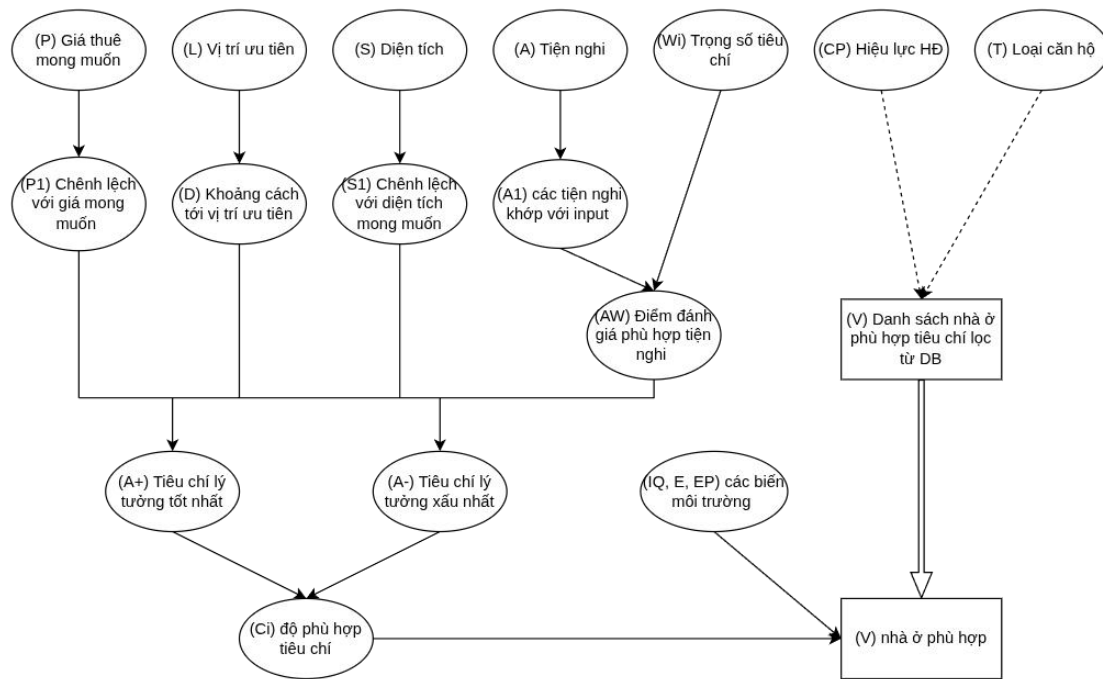
Bảng 10. Các yếu tố hệ thống không thể dự đoán được ảnh hưởng tới quyết định của khách hàng

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	
(P1)	Chênh lệch với giá mong muốn		Số thực	
(D)	Khoảng cách tới vị trí ưu tiên	Khoảng cách euclid từ vị trí ưu tiên tới vị trí căn trợ	Số thực	
(S1)	Chênh lệch với diện tích khách hàng mong muốn		Số thực	
(A1, AW)	Điểm đánh giá phù hợp tiện nghi	Tính dựa trên những tiện nghi mong muốn và tiện nghi của căn trợ nhân với trọng số.	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 trọ 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. CÔNG NGHỆ SỬ DỤNG

Trong quá trình xây dựng hệ thống trợ giúp quyết định tìm thuê chỗ ở, nhóm đã sử dụng kết hợp nhiều công nghệ từ xử lý dữ liệu, lưu trữ, triển khai backend đến xây dựng giao diện frontend. Việc lựa chọn các công nghệ này nhằm đảm bảo hệ thống vận hành hiệu quả, dễ mở rộng và phù hợp với yêu cầu phân tích – gợi ý nhà thuê theo nhiều tiêu chí của người dùng.

(1) **Jupyter Notebook** – Tiền xử lý dữ liệu Jupyter Notebook được sử dụng trong giai đoạn thu thập, làm sạch và chuẩn hóa dữ liệu nhà thuê. Công cụ này hỗ trợ thao tác dữ liệu trực quan, dễ thử nghiệm và thuận tiện cho việc thực hiện các bước như loại bỏ dữ liệu trùng, chuẩn hóa thông tin vị trí, chuyển đổi kiểu dữ liệu và tính toán các biến trung gian cần thiết cho mô hình hỗ trợ quyết định. Ngoài ra, Jupyter Notebook cho phép ghi lại toàn bộ quy trình xử lý dữ liệu, giúp dễ dàng kiểm tra và tái sử dụng trong các lần chạy tiếp theo.

(2) **PostgreSQL** – Hệ quản trị cơ sở dữ liệu PostgreSQL được lựa chọn làm hệ quản trị cơ sở dữ liệu của hệ thống vì khả năng xử lý các tập dữ liệu lớn, hỗ trợ giao dịch mạnh mẽ và tính ổn định cao. Cơ sở dữ liệu lưu trữ các thông tin về tỉnh/thành, quận/huyện, phường/xã, dữ liệu nhà cho thuê, điều kiện sống và nhật ký tương tác của người dùng. PostgreSQL cũng hỗ trợ mở rộng cấu trúc dữ liệu và tối ưu truy vấn, đảm bảo hiệu năng khi hệ thống cần lọc và truy xuất nhiều bản ghi theo tiêu chí của người dùng.

(3) **FastAPI** – Backend xử lý logic và thuật toán DSS FastAPI được sử dụng để xây dựng backend cho hệ thống nhờ tốc độ xử lý nhanh, cú pháp gọn nhẹ và khả năng xây dựng API hiện đại dựa trên chuẩn OpenAPI. Backend chịu trách nhiệm tiếp nhận yêu cầu từ người dùng, truy vấn cơ sở dữ liệu, xử lý các tiêu chí tìm kiếm và thực hiện mô hình hỗ trợ quyết định (như TOPSIS) để xếp hạng các lựa chọn. FastAPI đồng thời hỗ trợ cơ chế bất đồng bộ (async) giúp cải thiện hiệu năng khi có nhiều yêu cầu truy vấn đồng thời, phù hợp với hệ thống hướng dữ liệu như DSS.

(4) **HTML, CSS và JavaScript** – Giao diện người dùng (Frontend) Frontend của hệ thống được xây dựng bằng HTML, CSS và JavaScript, nhằm tạo ra giao diện đơn giản, trực quan và dễ sử dụng cho người dùng cuối. HTML giúp cấu trúc nội dung, CSS giúp tạo bố cục và định dạng hiển thị, trong khi JavaScript đảm nhiệm việc xử lý các tương tác động như gửi yêu cầu tìm kiếm, cập nhật danh sách gợi ý theo thời gian thực, cũng như hỗ trợ trực quan hóa thông tin kết quả trả về từ backend. Sự kết hợp này đảm bảo hệ thống có giao diện nhẹ, linh hoạt và dễ mở rộng trong tương lai.

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 Hệ thống Hỗ trợ Quyết định 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.

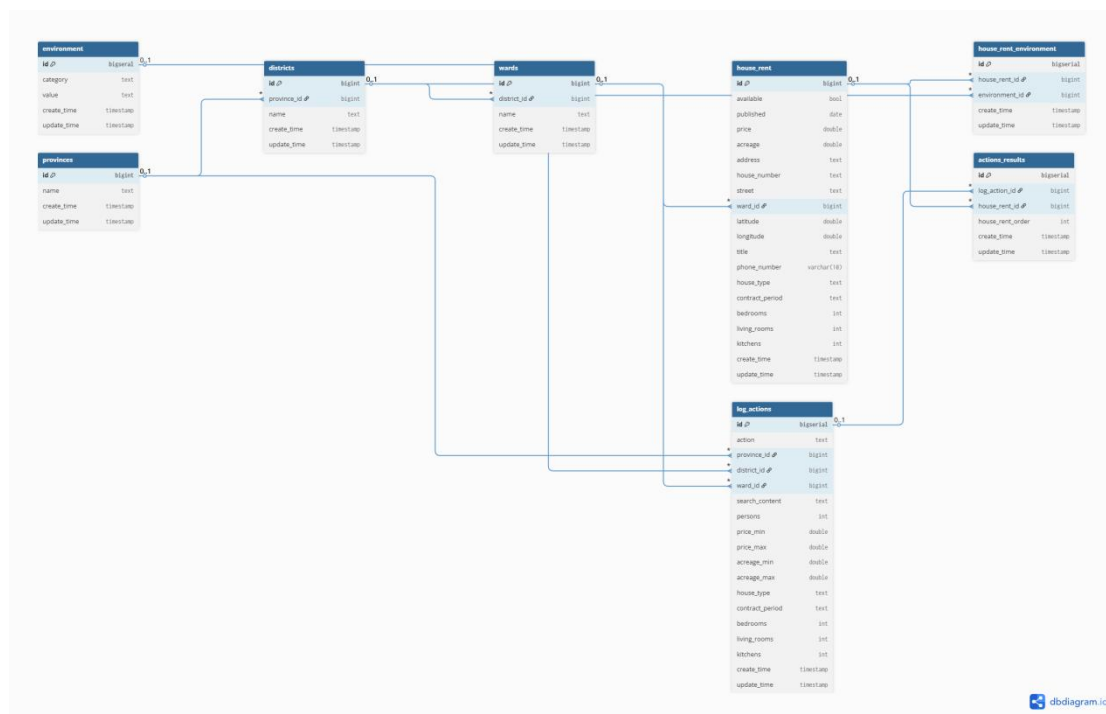
(3) Lớp dịch vụ (API layer) triển khai các endpoint để phục vụ mô hình và dữ liệu cho giao diện và thực hiện dưới dạng một service backend.

(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: m2
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	
longtitude	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 NGUYỄN 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 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 NGHỈ; AN NINH; CHU

			KỶ 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
---------	--------------	-------	---------

id	Bigserial		Khóa chính
action	Text	Hành động	Có 2 giá trị: SEARCH, SELECT
province_id	Bigint	Mã tỉnh / thành	Khóa ngoại với id trong bảng “provinces”
district_id	Bigint	Mã quận / huyện	Khóa ngoại với id trong bảng “districts”
ward_id	Bigint	Mã phường / xã	Khóa ngoại với id trong bảng “wards”
search_content	Text	Nội dung tìm kiếm	
persons	Int	Số người ở	
price_min	Double	Lọc giá thấp nhất	Tính theo tháng. Đơn vị: triệu đồng
price_max	Double	Lọc giá cao nhất	Tính theo tháng. Đơn vị: triệu đồng
acreage_min	Double	Lọc diện tích thấp nhất	Đơn vị: m2
acreage_max	Double	Lọc diện tích cao nhất	Đơn vị: m2
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_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

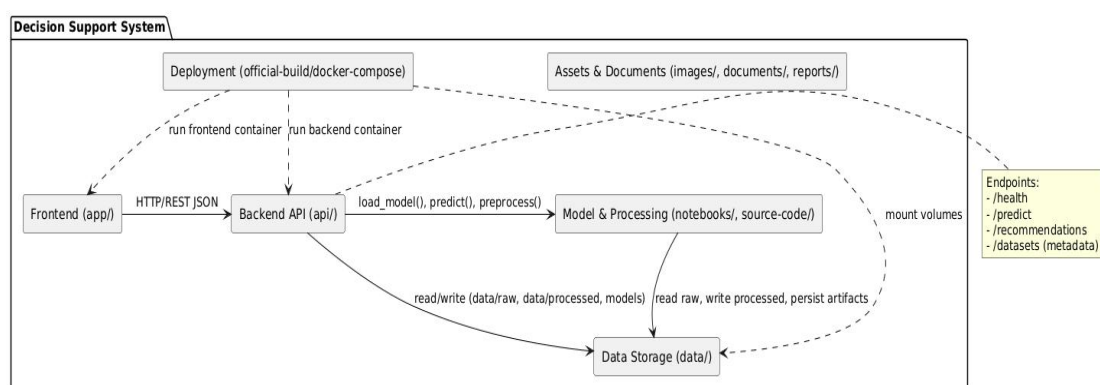
Bảng “actions_results”: Thông tin về danh sách nhà cho thuê trả về theo hành động (tìm kiếm, lựa chọn) của người dùng. Chi tiết như sau:

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 ⁱ. 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 artifacts để đảm bảo khả năng tái tạo kết quả, và cấu hình hoá 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 (user, timestamp, model_version) để phục vụ giám sát và truy vết.

Cấu trúc module đề 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 Kaggleⁱⁱ 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 thống nhất và xử lý outlier cơ bản.

4.2.2 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

(npm/yarn) 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:

Danh sách các kết quả trả về dựa sắp xếp theo thứ tự ưu tiên dựa trên điểm topsis:

Thông tin chi tiết của một căn nhà cần cho thuê với các tiêu chí ... :

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

ⁱ Mã nguồn của báo cáo [online]: https://github.com/Tran-Ngoc-Bao/Decision_Support_System

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