

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



THỰC TẬP 1 (CO5107)

Báo cáo cuối kỳ

Phát triển chatbot hỗ trợ tìm đường
và ước tính thời gian di chuyển (ETA)
trong hệ thống giao thông thông minh (ITS)

GVHD: PGS.TS Trần Minh Quang
SV thực hiện: Nguyễn Minh Anh - 2370644

TP. Hồ Chí Minh, Ngày 7 Tháng 1 Năm 2026



Mục lục

1 Giới thiệu	3
1.1 Bối cảnh và động cơ	3
1.2 Mục tiêu của đề tài	3
1.3 Phạm vi và giới hạn	4
2 Tổng quan hệ thống	5
2.1 Kiến trúc tổng thể hệ thống	5
2.2 Công nghệ và dịch vụ sử dụng	6
3 Thiết kế hội thoại và FSM	7
3.1 Thiết kế hội thoại theo hướng FSM	7
3.2 Các trạng thái hội thoại	8
3.3 Sơ đồ trạng thái (State Machine Diagram)	9
3.4 Bảng State Transition Table	11
3.4.1 START	12
3.4.2 FROM_TEXT	13
3.4.3 FROM_PICK	14
3.4.4 TO_TEXT	15
3.4.5 TO_PICK	16
3.4.6 MODE	17
3.5 Quản lý Context hội thoại	18
4 Xử lý địa điểm và định tuyến	19
4.1 Geocoding với Nominatim	19
4.2 Rút gọn và hiển thị tên địa điểm (Label)	20
4.3 Định tuyến và ETA với OSRM	21
4.4 Hiển thị tuyến đường với OpenStreetMap Directions	22
5 Triển khai và kết quả demo	23
5.1 Môi trường triển khai	23
5.2 Kịch bản demo và kết quả	24
5.3 Sequence Diagram	32
6 Dánh giá và hạn chế	34
6.1 Dánh giá	34
6.2 Hạn chế	34



7	Kết luận và hướng phát triển	35
7.1	Kết luận	35
7.2	Hướng phát triển	35



1 Giới thiệu

1.1 Bối cảnh và động cơ

Trong bối cảnh đô thị hóa nhanh chóng, nhu cầu di chuyển trong các thành phố lớn ngày càng tăng cao, kéo theo nhiều vấn đề như ùn tắc giao thông, thời gian di chuyển không ổn định và thiếu công cụ hỗ trợ lựa chọn lộ trình hiệu quả. Hệ thống giao thông thông minh (Intelligent Transportation Systems – ITS) được phát triển nhằm ứng dụng công nghệ thông tin và các dịch vụ bản đồ số để hỗ trợ người tham gia giao thông trong việc tra cứu lộ trình và ước tính thời gian di chuyển.

Bên cạnh đó, các nền tảng nhắn tin tức thời như Telegram ngày càng phổ biến và cho phép triển khai các chatbot tương tác trực tiếp với người dùng thông qua hội thoại. Việc tích hợp chatbot vào hệ thống ITS giúp cung cấp thông tin giao thông theo cách đơn giản, thân thiện và dễ tiếp cận hơn so với các ứng dụng bản đồ truyền thống.

Xuất phát từ thực tế trên, đề tài tập trung vào việc phát triển một chatbot hỗ trợ tìm đường và ước tính thời gian di chuyển (ETA), đóng vai trò là lớp giao diện tương tác người dùng trong một hệ thống giao thông thông minh.

1.2 Mục tiêu của đề tài

Mục tiêu của đề tài là phát triển một chatbot trên nền tảng Telegram nhằm hỗ trợ người dùng tìm đường và ước tính thời gian di chuyển thông qua hình thức hội thoại. Cụ thể, đề tài hướng đến các mục tiêu sau:

- Xây dựng chatbot hoạt động trên nền tảng Telegram.
- Thiết kế luồng hội thoại dựa trên mô hình máy trạng thái hữu hạn (Finite State Machine – FSM).
- Tích hợp dịch vụ geocoding để chuyển đổi địa điểm dạng văn bản sang tọa độ địa lý.
- Tích hợp dịch vụ định tuyến để tính toán quãng đường và thời gian di chuyển ước tính (ETA).
- Trả kết quả cho người dùng dưới dạng văn bản và liên kết mở bản đồ chỉ đường trực quan.



1.3 Phạm vi và giới hạn

Trong khuôn khổ học phần Thực tập 1, đề tài được giới hạn trong các phạm vi sau:

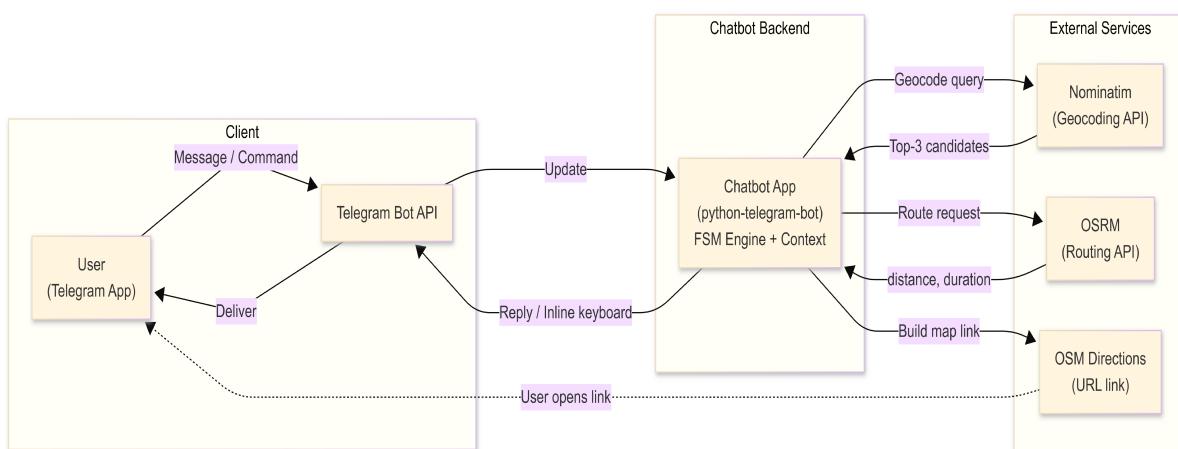
- Chatbot hỗ trợ tìm đường trong khu vực Thành phố Hồ Chí Minh.
- Việc chuyển đổi tọa độ và tính toán lộ trình sử dụng các dịch vụ công cộng (Nominatim và OSRM).
- Hệ thống hiện tại chỉ hỗ trợ tính toán lộ trình và ước tính thời gian di chuyển cho phương tiện ô tô; các chế độ phương tiện khác chưa được triển khai.
- Đề tài tập trung vào phát triển chatbot và luồng xử lý, không triển khai thuật toán định tuyến hay xử lý dữ liệu giao thông thời gian thực.

Mặc dù có những giới hạn nhất định, hệ thống vẫn đáp ứng tốt mục tiêu minh họa vai trò của chatbot trong việc hỗ trợ tra cứu lộ trình và ước tính thời gian di chuyển trong hệ thống giao thông thông minh.

2 Tổng quan hệ thống

2.1 Kiến trúc tổng thể hệ thống

Hệ thống chatbot được thiết kế theo mô hình kiến trúc thành phần, bao gồm ba thành phần chính: phía người dùng (Client), hệ thống chatbot backend và các dịch vụ bên ngoài. Kiến trúc tổng thể của hệ thống được minh họa trong Hình 1.



Hình 1: Kiến trúc tổng thể hệ thống chatbot hỗ trợ tìm đường

Các thành phần chính của hệ thống bao gồm:

- Client

- **User (Telegram App):** Người dùng tương tác với chatbot thông qua tin nhắn và lệnh (ví dụ: /route).
- **Telegram Bot API:** Dóng vai trò trung gian, tiếp nhận thông điệp từ người dùng và chuyển các cập nhật (update) đến chatbot backend, đồng thời gửi phản hồi ngược lại cho client.

- Chatbot Backend

- **Chatbot App:** Là thành phần trung tâm của hệ thống, được xây dựng bằng thư viện `python-telegram-bot`. Bao gồm:
 - * **FSM Engine:** Điều khiển luồng hội thoại theo mô hình máy trạng thái hữu hạn (FSM), đảm bảo người dùng cung cấp thông tin theo đúng trình tự.



- * **Context Manager:** Lưu trữ trạng thái và dữ liệu tạm thời của phiên hội thoại (điểm đi, điểm đến, tọa độ, lựa chọn phương tiện).

• External Services

- **Nominatim - Geocoding API:** Chuyển đổi địa điểm dạng văn bản sang tọa độ địa lý và trả về danh sách gợi ý (top-3 candidates).
- **OSRM - Routing API:** Tính toán quãng đường và thời gian di chuyển ước tính (ETA) dựa trên tọa độ điểm đi và điểm đến.
- **OpenStreetMap Directions:** Cung cấp liên kết bản đồ chỉ đường trực quan để người dùng mở và xem lộ trình.

Luồng xử lý tổng quát của hệ thống bao gồm việc tiếp nhận yêu cầu từ người dùng, điều phối xử lý theo FSM, lần lượt gọi các dịch vụ geocoding và routing, sau đó tổng hợp và trả kết quả cho người dùng dưới dạng văn bản kèm liên kết bản đồ chỉ đường.

2.2 Công nghệ và dịch vụ sử dụng

Hệ thống chatbot được xây dựng dựa trên các công nghệ và dịch vụ sau:

- Telegram Bot API: Được sử dụng làm kênh giao tiếp với người dùng, cho phép chatbot nhận tin nhắn, lệnh và gửi phản hồi.
- `python-telegram-bot`: Thư viện Python dùng để xây dựng chatbot, hỗ trợ quản lý hội thoại nhiều bước thông qua `ConversationHandler`, cho phép xây dựng luồng hội thoại nhiều bước theo FSM.
- Nominatim (Geocoding API): Dùng để chuyển đổi địa điểm dạng văn bản sang tọa độ địa lý (vĩ độ, kinh độ), phục vụ cho việc xác định điểm đi và điểm đến.
- OSRM (Routing API): Dùng để tính toán quãng đường và thời gian di chuyển ước tính ETA dựa trên tọa độ điểm đi và điểm đến.
- OpenStreetMap Directions: Dùng để tạo liên kết bản đồ chỉ đường, cho phép người dùng xem trực quan lộ trình di chuyển.



3 Thiết kế hội thoại và FSM

3.1 Thiết kế hội thoại theo hướng FSM

Chatbot hỗ trợ tìm đường là một ứng dụng hội thoại đa lượt, trong đó người dùng cần cung cấp thông tin theo từng bước như điểm đi, điểm đến và lựa chọn phương tiện. Trong đề tài này, luồng hội thoại được thiết kế dựa trên mô hình máy trạng thái hữu hạn (Finite State Machine - FSM).

Việc sử dụng FSM cho phép:

- Chia hội thoại thành các bước rõ ràng, mỗi trạng thái xử lý một loại thông tin cụ thể.
- Đảm bảo người dùng nhập thông tin theo đúng trình tự, tránh sai ngữ cảnh.
- Đơn giản hóa việc xử lý hội thoại và kiểm soát luồng tương tác.

Bên cạnh FSM, hệ thống sử dụng *context* để lưu trữ dữ liệu tạm thời của phiên hội thoại, bao gồm điểm đi, điểm đến, tọa độ và lựa chọn phương tiện. Context được khởi tạo khi người dùng bắt đầu lệnh `/route` và được xóa khi hội thoại kết thúc hoặc bị hủy bằng lệnh `/cancel`.

Cách tiếp cận này phù hợp với phạm vi đề tài Thực tập 1, giúp hệ thống hoạt động ổn định, dễ triển khai và vẫn đảm bảo hiệu quả tương tác với người dùng, đồng thời tạo nền tảng thuận lợi cho việc mở rộng trong các phiên bản tiếp theo.



3.2 Các trạng thái hội thoại

Luồng hội thoại của chatbot được thiết kế theo mô hình máy trạng thái hữu hạn (FSM), trong đó mỗi trạng thái tương ứng với một bước xử lý cụ thể trong quá trình tìm đường.

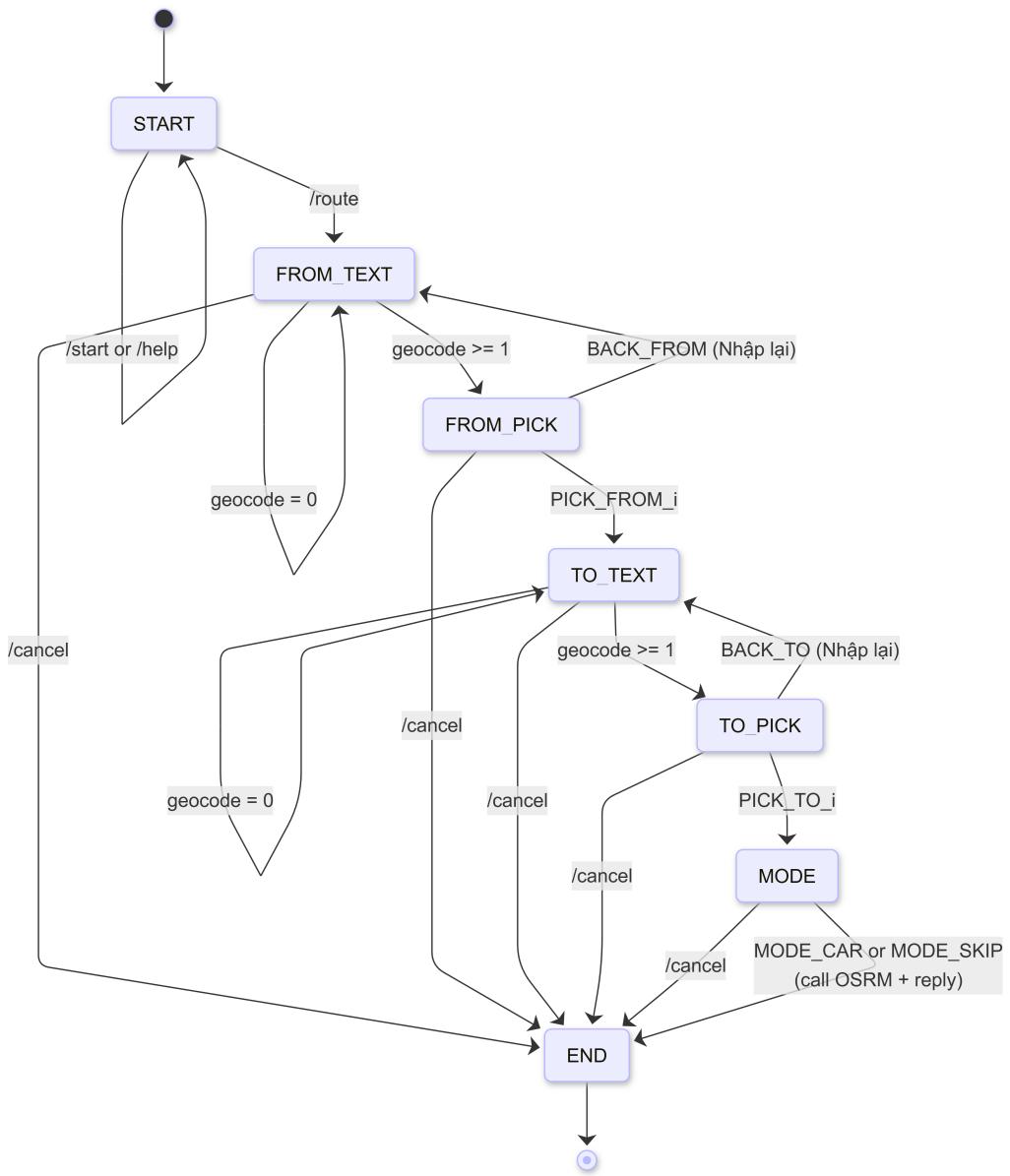
Các trạng thái chính của hệ thống bao gồm:

- **START**: Trạng thái khởi đầu, người dùng bắt đầu luồng tìm đường bằng lệnh /route.
- **FROM_TEXT**: Nhận địa điểm xuất phát dưới dạng văn bản.
- **FROM_PICK**: Người dùng chọn điểm xuất phát từ danh sách gợi ý.
- **TO_TEXT**: Nhận địa điểm đích dưới dạng văn bản.
- **TO_PICK**: Người dùng chọn điểm đến từ danh sách gợi ý.
- **MODE**: Lựa chọn phương tiện di chuyển (phiên bản demo chỉ hỗ trợ ô tô).
- **END**: Trả kết quả định tuyến và kết thúc hội thoại.

Việc phân chia hội thoại thành các trạng thái rõ ràng giúp kiểm soát luồng tương tác, hạn chế sai sót khi nhập liệu và đơn giản hóa quá trình xử lý hội thoại.

3.3 Sơ đồ trạng thái (State Machine Diagram)

Sơ đồ máy trạng thái hữu hạn (FSM) của chatbot được minh họa trong Hình 2. Sơ đồ mô tả toàn bộ luồng hội thoại từ khi người dùng bắt đầu lệnh tìm đường cho đến khi hệ thống trả kết quả và kết thúc phiên làm việc.



Hình 2: Sơ đồ máy trạng thái hội thoại của chatbot



Luồng hội thoại chính bắt đầu từ trạng thái START. Khi người dùng nhập lệnh /route, hệ thống chuyển sang trạng thái FROM_TEXT để thu thập địa điểm xuất phát. Sau khi thực hiện geocoding, nếu có kết quả hợp lệ, chatbot hiển thị danh sách gợi ý và chuyển sang trạng thái FROM_PICK để người dùng lựa chọn.

Quy trình tương tự được áp dụng cho địa điểm đích thông qua các trạng thái TO_TEXT và TO_PICK. Khi cả điểm đi và điểm đến đã được xác định, hệ thống chuyển sang trạng thái MODE để người dùng xác nhận phương tiện di chuyển (phiên bản demo mặc định sử dụng ô tô). Sau đó, chatbot gọi dịch vụ định tuyến, trả kết quả quãng đường và thời gian ước tính, và chuyển sang trạng thái END để kết thúc hội thoại.

Ngoài luồng chính, sơ đồ cũng thể hiện các nhánh xử lý phụ như nhập lại địa điểm khi không tìm thấy kết quả geocoding và cơ chế hủy hội thoại bằng lệnh /cancel tại bất kỳ trạng thái nào. Thiết kế này giúp luồng hội thoại rõ ràng, dễ kiểm soát và hạn chế các tình huống nhập sai hoặc sai ngữ cảnh.



3.4 Bảng State Transition Table

Để mô tả chi tiết hành vi của hệ thống tại từng trạng thái hội thoại, đề tài sử dụng các bảng State Transition Table. Mỗi bảng tương ứng với một trạng thái trong FSM và mô tả đầy đủ các tình huống có thể xảy ra trong quá trình tương tác với người dùng.

Mỗi bảng State Transition Table bao gồm các thành phần chính:

- **Current State:** Trạng thái hiện tại của hội thoại.
- **User Action:** Hành động hoặc dữ liệu đầu vào từ người dùng.
- **System Action:** Các xử lý nội bộ của hệ thống.
- **Bot Reply:** Phản hồi của chatbot gửi đến người dùng.
- **Next State:** Trạng thái tiếp theo của FSM.

Trong phạm vi đề tài, FSM được mô tả thông qua 6 bảng State Transition Table.



3.4.1 START

Current State	User Action	System Action	Bot Reply	Next State
START	/start	—	Giới thiệu + hướng dẫn nhanh: dùng /route, /help, /cancel	START
START	/help	—	Hướng dẫn chi tiết	START
START	/route	Clear context	"Bắt đầu tìm đường. Bạn đi từ đâu? (Nhập địa điểm xuất phát bằng chữ)"	FROM_TEXT
START	Text bất kỳ	—	"Gõ /route để bắt đầu, hoặc /help để xem hướng dẫn."	START

Bảng 1: State Transition Table - START



3.4.2 FROM_TEXT

Current State	User Action	System Action	Bot Reply	Next State
FROM_TEXT	/cancel	Clear context + end flow	"Đã hủy. Gõ /route để bắt đầu lại."	END
FROM_TEXT	Non-text (sticker/photo/..)	—	"Mình chỉ nhận địa điểm dạng chữ. Bạn nhập điểm xuất phát bằng text nhé."	FROM_TEXT
FROM_TEXT	Text = điểm đi	Lưu from-text, gọi geocode (top-3)	(Phụ thuộc geocode)	(Phụ thuộc)
FROM_TEXT	Geocode = 0	Set from-candidates = []	"Không tìm thấy. Bạn nhập rõ hơn (số nhà/đường...)"	FROM_TEXT
FROM_TEXT	Geocode ≥ 1	Set from-candidates = top-3. Show inline keyboard.	"Mình tìm thấy các địa điểm sau. Bạn chọn đúng điểm xuất phát:"	FROM_PICK

Bảng 2: State Transition Table - FROM_TEXT



3.4.3 FROM_PICK

Current State	User Action	System Action	Bot Reply	Next State
FROM_PICK	/cancel	Clear context + end flow	"Đã hủy. Gõ /route để bắt đầu lại."	END
FROM_PICK	User text	—	"Vui lòng bấm chọn một địa điểm bên dưới hoặc bấm 'Nhập lại'."	FROM_PICK
FROM_PICK	Callback "Nhập lại"	Clear from-text, from-candidates, from-coord	"Ok, bạn nhập lại điểm xuất phát nhé."	FROM_TEXT
FROM_PICK	Callback candidate i	Lưu from-coord, from-label	"Đã chọn điểm đi. Bạn muốn đến đâu? (nhập địa điểm đích)"	TO_TEXT
FROM_PICK	Callback không hợp lệ	—	"Lựa chọn không hợp lệ. Bạn chọn lại giúp mình nhé."	FROM_PICK

Bảng 3: State Transition Table - FROM_PICK



3.4.4 TO_TEXT

Current State	User Action	System Action	Bot Reply	Next State
TO_TEXT	/cancel	Clear context + end flow	"Đã hủy. Gõ /route để bắt đầu lại."	END
TO_TEXT	Non-text	—	"Mình chỉ nhận địa điểm dạng chữ. Bạn nhập điểm đến bằng text nhé."	TO_TEXT
TO_TEXT	Text = điểm đến	Lưu text, gọi geocode (top-3)	(Phụ thuộc geocode)	(Phụ thuộc)
TO_TEXT	Geocode = 0	Set to_candidates = []	"Không tìm thấy. Bạn nhập rõ hơn (số nhà..)"	TO_TEXT
TO_TEXT	Geocode ≥ 1	Set to-candidates = top-3. Show inline keyboard.	"Mình tìm thấy các địa điểm sau. Bạn chọn đúng điểm đến:"	TO_PICK

Bảng 4: State Transition Table - TO_TEXT



3.4.5 TO_PICK

Current State	User Action	System Action	Bot Reply	Next State
TO_PICK	/cancel	Clear context + end flow	"Đã hủy. Go /route để bắt đầu lại."	END
TO_PICK	User text	—	"Vui lòng bấm chọn một địa điểm bên dưới hoặc bấm 'Nhập lại'."	TO_PICK
TO_PICK	Callback "Nhập lại"	Clear to-text, to-candidates, to-coord	"Ok, bạn nhập lại điểm đến nhé."	TO_TEXT
TO_PICK	Callback candidate i	Lưu to-coord, to-label	"Đã chọn điểm đến. Bạn muốn đi bằng phương tiện nào?"	MODE
TO_PICK	Callback không hợp lệ	—	"Lựa chọn không hợp lệ. Bạn chọn lại giúp mình nhé."	TO_PICK

Bảng 5: State Transition Table - TO_PICK



3.4.6 MODE

Current State	User Action	System Action	Bot Reply	Next State
MODE	/cancel	Clear context + end flow	"Đã hủy. Gõ /route để bắt đầu lại."	END
MODE	User text	—	"Vui lòng bấm chọn 'Ô tô' hoặc 'Bỏ qua'."	MODE
MODE	Callback "Ô tô"	Set mode = "car". Gọi OSRM route. Format kết quả + tạo link.	Trả kết quả tuyến + link	END
MODE	Callback "Bỏ qua"	Set mode = "car". Gọi OSRM route. Format kết quả + tạo link.	Trả kết quả tuyến + link	END

Bảng 6: State Transition Table - MODE



3.5 Quản lý Context hội thoại

Context được sử dụng để lưu trữ dữ liệu tạm thời của phiên hội thoại, phục vụ cho việc duy trì trạng thái và truyền dữ liệu giữa các bước trong FSM.

Key	Type	Giải thích
from_text	str	Địa điểm xuất phát do người dùng nhập
from_candidates	list[Candidate]	Danh sách địa điểm gợi ý cho điểm đi
from_coord	tuple(float, float)	Tọa độ điểm đi (lat, lon)
from_label	str	Tên rút gọn của điểm đi để hiển thị
to_text	str	Địa điểm đích do người dùng nhập
to_candidates	list[Candidate]	Danh sách địa điểm gợi ý cho điểm đến
to_coord	tuple(float, float)	Tọa độ điểm đến (lat, lon)
to_label	str	Tên rút gọn của điểm đến để hiển thị
mode	str	Phương tiện di chuyển (demo: car)
route_result	dict	Kết quả định tuyến (quãng đường, thời gian, map link)

Bảng 7: Các khóa dữ liệu trong context hội thoại

Context được khởi tạo khi bắt đầu lệnh /route và được xóa khi hội thoại kết thúc hoặc bị hủy.



4 Xử lý địa điểm và định tuyến

4.1 Geocoding với Nominatim

Trong hệ thống, dịch vụ Nominatim được sử dụng để chuyển đổi địa điểm do người dùng nhập dưới dạng văn bản sang tọa độ địa lý (vĩ độ, kinh độ). Kết quả geocoding là cơ sở để xác định điểm đi, điểm đến và thực hiện bước định tuyến tiếp theo.

Để giảm nhiễu kết quả và phù hợp với phạm vi đề tài, việc gọi Nominatim được cấu hình với bộ tham số cố định như sau:

- **q:** Chuỗi truy vấn địa điểm do người dùng nhập.
- **format=jsonv2:** Định dạng dữ liệu trả về.
- **limit=3:** Giới hạn tối đa 3 kết quả gợi ý.
- **addressdetails=1:** Trả về thông tin địa chỉ chi tiết.
- **countrycodes=vn:** Giới hạn trong phạm vi Việt Nam.
- **accept-language=vi:** Ưu tiên hiển thị tiếng Việt.
- **viewbox:** Giới hạn khu vực Thành phố Hồ Chí Minh.
- **bounded=1:** Chỉ trả kết quả nằm trong vùng giới hạn.

Ví dụ minh họa: với truy vấn địa điểm “Đại học Bách Khoa”, hệ thống thực hiện yêu cầu geocoding như sau:

```
https://nominatim.openstreetmap.org/search?  
q=Đại+học+Bách+Khoa  
&format=jsonv2  
&limit=3  
&addressdetails=1  
&countrycodes=vn  
&accept-language=vi  
&viewbox=106.3567007,10.1399458,107.0276712,11.1603083  
&bounded=1
```

Kết quả trả về là danh sách tối đa 3 địa điểm phù hợp, mỗi địa điểm bao gồm tên hiển thị, thông tin địa chỉ và tọa độ địa lý. Các kết quả này được sử dụng để hiển thị danh sách lựa chọn cho người dùng trong luồng hội thoại.



4.2 Rút gọn và hiển thị tên địa điểm (Label)

Kết quả geocoding từ Nominatim thường trả về tên địa điểm dài và chi tiết, không phù hợp để hiển thị trực tiếp trên giao diện hội thoại và các nút chọn (inline keyboard). Vì vậy, hệ thống thực hiện rút gọn và chuẩn hóa tên địa điểm trước khi hiển thị cho người dùng.

Nguyên tắc xây dựng label cho mỗi địa điểm bao gồm:

- Sử dụng tên chính của địa điểm làm phần trung tâm (base name).
- Bổ sung một số thông tin địa chỉ ngắn gọn (ví dụ: số nhà, khu phố, phường) để phân biệt giữa các địa điểm trùng tên.

Ví dụ minh họa:

- Kết quả API trả về: “*Ký túc xá DH Bách Khoa, 497, Đường Hòa Hảo, Khu phố 27, Phường Diên Hồng, Thành phố Thủ Đức, Thành phố Hồ Chí Minh, 72415, Việt Nam*,”
- Label hiển thị: “*Ký túc xá DH Bách Khoa - 497 Hòa Hảo, KP 27, P. Diên Hồng*”



4.3 Định tuyến và ETA với OSRM

Sau khi xác định được tọa độ điểm đi và điểm đến, hệ thống sử dụng dịch vụ OSRM (Open Source Routing Machine) để tính toán lộ trình và ước tính thời gian di chuyển (ETA). OSRM là một công cụ định tuyến mã nguồn mở, cho phép trả về quãng đường và thời gian di chuyển dựa trên dữ liệu bản đồ OpenStreetMap.

Trong phạm vi đề tài, chatbot sử dụng OSRM public server với chế độ định tuyến cho phương tiện ô tô (**driving**). Các tham số chính khi gọi API bao gồm:

- Tọa độ điểm đi và điểm đến theo định dạng lon, lat.
- **overview=false**: Không yêu cầu trả về hình dạng tuyến chi tiết, chỉ lấy kết quả tổng quát.

Ví dụ minh họa: với điểm đi và điểm đến lần lượt là (10.7720, 106.6578) và (10.8188, 106.6518), hệ thống gửi yêu cầu định tuyến như sau:

```
https://router.project-osrm.org/route/v1/driving/  
106.6578,10.7720;106.6518,10.8188?overview=false
```

Kết quả trả về từ OSRM bao gồm thông tin quãng đường (đơn vị mét) và thời gian di chuyển ước tính (đơn vị giây). Các giá trị này được hệ thống chuyển đổi sang đơn vị km và phút trước khi hiển thị cho người dùng.



4.4 Hiển thị tuyến đường với OpenStreetMap Directions

Sau khi tính toán lộ trình và thời gian di chuyển bằng OSRM, hệ thống tạo liên kết chỉ đường trực quan bằng dịch vụ OpenStreetMap Directions để người dùng có thể xem tuyến đường trên bản đồ.

Liên kết được xây dựng từ tọa độ điểm đi và điểm đến theo định dạng `lat,lon`, sử dụng engine định tuyến của OSRM cho phương tiện ô tô.

Mẫu liên kết chỉ đường:

`https://www.openstreetmap.org/directions?engine=fossgis_osrm_car&route=lat1,lon1;lat2,lon2`

Ví dụ minh họa:

`https://www.openstreetmap.org/directions?engine=fossgis_osrm_car&route=10.7720,106.6578;10.8188,106.6518`

Liên kết này được gửi kèm trong kết quả phản hồi của chatbot, cho phép người dùng mở bản đồ và quan sát trực quan lộ trình di chuyển.



5 Triển khai và kết quả demo

5.1 Môi trường triển khai

Chatbot được triển khai dưới dạng một ứng dụng Python, hoạt động trên nền tảng Telegram thông qua Telegram Bot API. Hệ thống sử dụng cơ chế polling để nhận và xử lý các yêu cầu từ người dùng theo thời gian thực.

Môi trường và công nghệ triển khai bao gồm:

- Ngôn ngữ lập trình: Python.
- Thư viện chatbot: `python-telegram-bot`.
- Cơ chế giao tiếp: Telegram Bot API (polling).
- Dịch vụ bên ngoài: Nominatim (geocoding), OSRM (routing), OpenStreetMap Directions.
- Nền tảng triển khai: máy tính cá nhân phục vụ mục đích demo.

Mã nguồn của hệ thống được công bố tại:

<https://github.com/redevil24/tt1-route-eta-chatbot>



5.2 Kịch bản demo và kết quả

Phần này minh họa hoạt động của chatbot thông qua một kịch bản hội thoại hoàn chỉnh trên Telegram: tìm đường từ *Dai hoc Bach Khoa* đến *Benh vien Nhi Dong*. Các hình dưới đây được chụp trực tiếp trong quá trình chạy demo.



Hình 3: Demo bước 1: Người dùng gõ /start và chatbot hiển thị giới thiệu, hướng dẫn nhanh



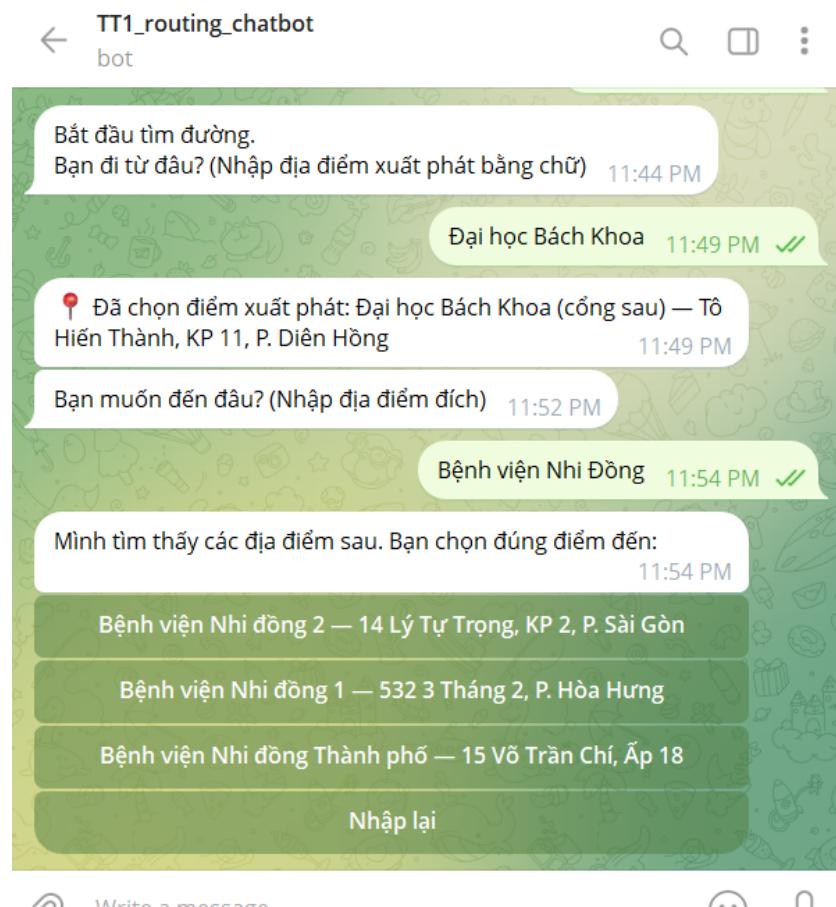
Hình 4: Demo bước 2: Người dùng gõ /route, chatbot khởi tạo luồng và yêu cầu nhập điểm xuất phát



Hình 5: Demo bước 3: Nhập điểm đi (*Đại học Bách Khoa*) và chatbot hiển thị danh sách gợi ý (inline keyboard)



Hình 6: Demo bước 4: Người dùng chọn điểm xuất phát, chatbot xác nhận và chuyển sang yêu cầu nhập điểm đến



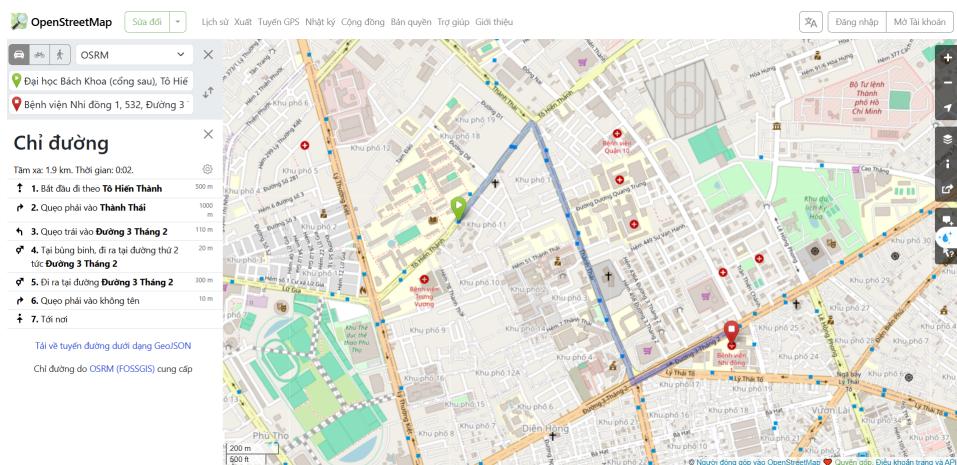
Hình 7: Demo bước 5: Nhập điểm đến (*Bệnh viện Nhi Đồng*) và chatbot hiển thị danh sách gợi ý (inline keyboard)



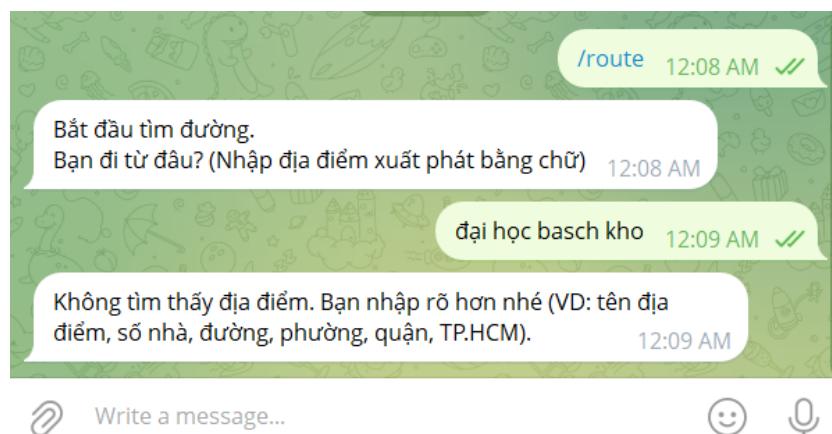
Hình 8: Demo bước 6: Người dùng chọn điểm đến, chatbot xác nhận và chuyển sang bước chọn phương tiện



Hình 9: Demo bước 7: Người dùng chọn phương tiện và Chatbot trả kết quả định tuyến gồm quãng đường, ETA và liên kết mở bản đồ chỉ đường



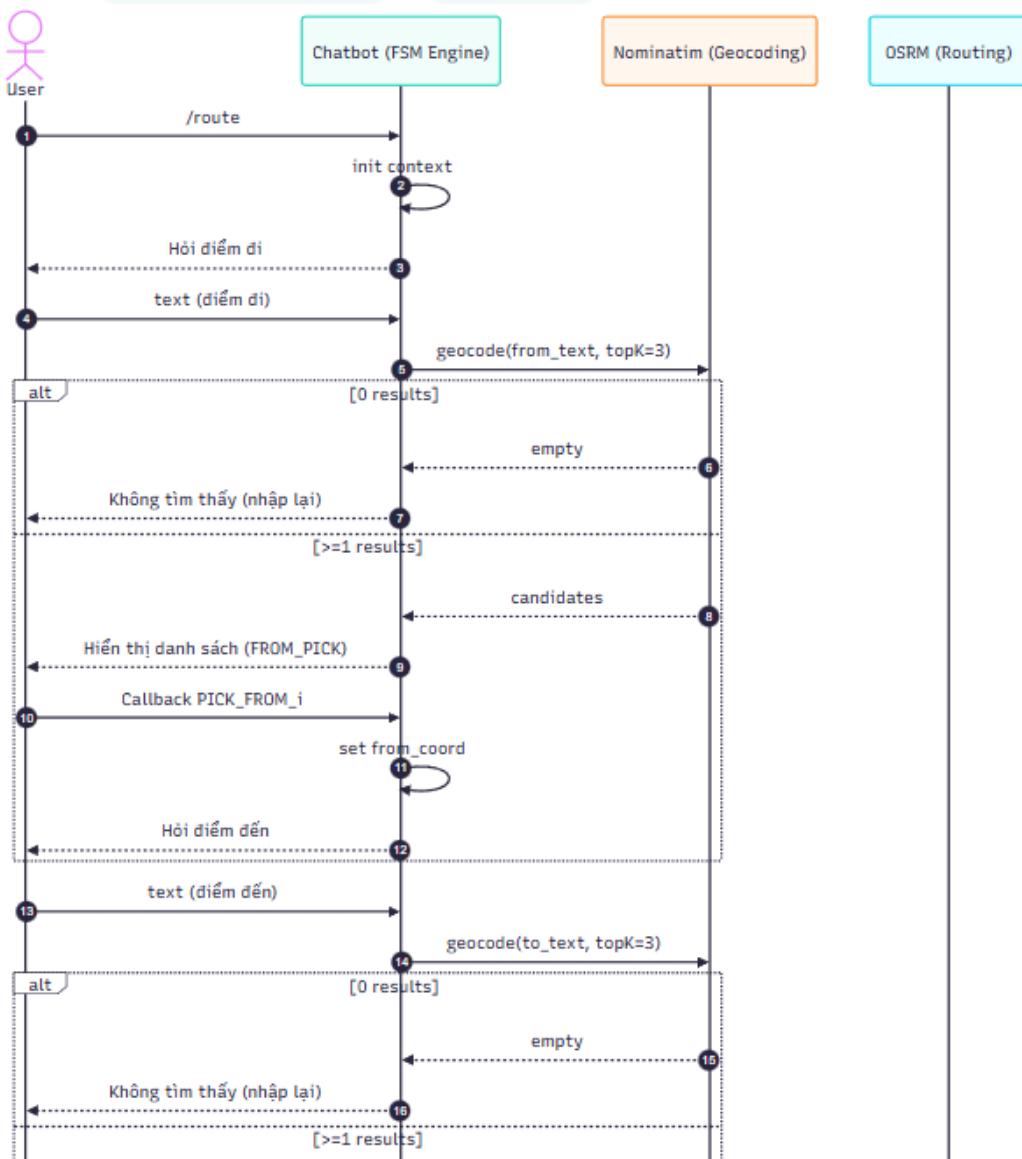
Hình 10: Demo bước 8: Mở liên kết OpenStreetMap Directions để xem tuyến đường trực quan trên bản đồ

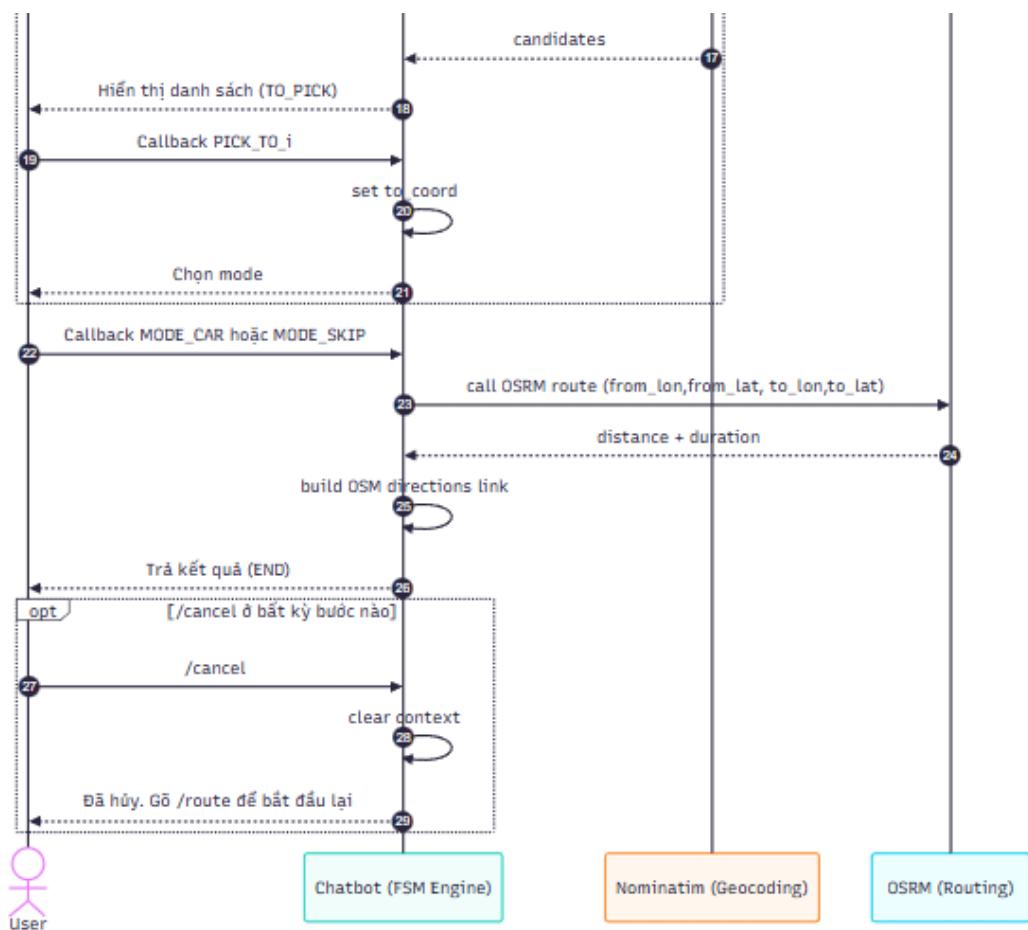


Hình 11: Trường hợp fallback: Không tìm thấy kết quả geocoding, chatbot yêu cầu người dùng nhập rõ địa điểm hơn

5.3 Sequence Diagram

Sơ đồ tuần tự (Sequence Diagram) minh họa chi tiết quá trình tương tác giữa người dùng, chatbot và các dịch vụ bên ngoài trong một phiên tìm đường hoàn chỉnh.





Hình 12: Sequence Diagram mô tả luồng tương tác giữa User, Chatbot (FSM Engine), Nominatim và OSRM



6 Đánh giá và hạn chế

6.1 Đánh giá

Hệ thống chatbot hỗ trợ tìm đường đã được xây dựng và vận hành ổn định. Các điểm đạt được chính bao gồm:

- Luồng hội thoại được thiết kế rõ ràng dựa trên mô hình máy trạng thái hữu hạn (FSM), giúp kiểm soát chặt chẽ trình tự nhập liệu và xử lý.
- Chatbot hoạt động ổn định trên nền tảng Telegram, cho phép người dùng tra cứu lộ trình thông qua hội thoại từng bước.
- Hệ thống tích hợp thành công dịch vụ geocoding (Nominatim) và định tuyến (OSRM) để xác định tọa độ, quãng đường và thời gian di chuyển ước tính.
- Kết quả được trả về dưới dạng văn bản dễ hiểu kèm liên kết bản đồ chỉ đường trực quan bằng OpenStreetMap.

Nhìn chung, hệ thống hoàn thành tốt vai trò minh họa cách xây dựng một chatbot tìm đường dựa trên FSM và các dịch vụ bản đồ công cộng.

6.2 Hạn chế

Bên cạnh các kết quả đạt được, hệ thống vẫn tồn tại một số hạn chế:

- Luồng hội thoại dựa trên FSM cố định, chưa hỗ trợ xử lý ngôn ngữ tự nhiên (NLU), do đó người dùng phải nhập thông tin theo từng bước thay vì nhập tự do trong một câu.
- Chất lượng geocoding phụ thuộc vào Nominatim, đôi khi trả về kết quả chưa chính xác, trùng tên địa điểm hoặc không tìm thấy địa điểm mong muốn so với các dịch vụ thương mại như Google Maps.
- Dịch vụ OSRM public chỉ hỗ trợ định tuyến cho phương tiện ô tô; các phương tiện khác chưa được triển khai.
- Kết quả định tuyến và ETA mang tính tĩnh, chưa xét đến tình trạng giao thông thời gian thực.
- Hệ thống phụ thuộc vào các API công cộng, có thể bị giới hạn về độ ổn định hoặc tần suất truy cập.

Các hạn chế trên là cơ sở cho các hướng phát triển trong các nghiên cứu tiếp theo.



7 Kết luận và hướng phát triển

7.1 Kết luận

Trong khuôn khổ học phần Thực tập 1, đề tài đã xây dựng thành công một chatbot hỗ trợ tìm đường và ước tính thời gian di chuyển trên nền tảng Telegram. Hệ thống được thiết kế dựa trên mô hình máy trạng thái hữu hạn (FSM), cho phép quản lý luồng hội thoại rõ ràng và kiểm soát tốt quá trình tương tác với người dùng.

Chatbot đã tích hợp hiệu quả các dịch vụ bản đồ công cộng gồm Nominatim cho geocoding và OSRM cho định tuyến, từ đó cung cấp thông tin quãng đường, ETA và liên kết bản đồ chỉ đường trực quan. Kết quả đạt được cho thấy tính khả thi của việc ứng dụng chatbot như một giao diện hội thoại đơn giản cho các bài toán tra cứu lộ trình trong hệ thống giao thông thông minh.

7.2 Hướng phát triển

Từ kết quả đạt được, hệ thống có thể được mở rộng và cải tiến trong các hướng sau:

- Tích hợp xử lý ngôn ngữ tự nhiên (NLU) để cho phép người dùng nhập tự do thông tin điểm đi và điểm đến trong một câu.
- Sử dụng các dịch vụ geocoding và routing có độ chính xác cao hơn hoặc triển khai hệ thống định tuyến riêng.
- Mở rộng hỗ trợ nhiều loại phương tiện di chuyển như xe máy, đi bộ hoặc xe đạp.
- Kết hợp dữ liệu giao thông thời gian thực để cải thiện độ chính xác của ETA.

Các hướng phát triển trên là cơ sở để tiếp tục nghiên cứu và mở rộng hệ thống trong các học phần hoặc đồ án tiếp theo.



Tài liệu

- [1] Telegram Messenger LLP, “Telegram Bot API,” Available: <https://core.telegram.org/bots/api>, accessed Jan. 8, 2026. :contentReference[oaicite:0]index=0
- [2] OpenStreetMap Foundation, “Nominatim Search UI,” Available: <https://nominatim.openstreetmap.org>, accessed Jan. 8, 2026. :contentReference[oaicite:1]index=1
- [3] python-telegram-bot Community, “python-telegram-bot — Official Site,” Available: <https://python-telegram-bot.org/>, accessed Jan. 8, 2026. :contentReference[oaicite:2]index=2
- [4] Project OSRM, “OSRM API v5.5.1 – General Options,” Available: <https://project-osrm.org/docs/v5.5.1/api>, accessed Jan. 8, 2026.
- [5] OpenStreetMap Foundation, “OpenStreetMap — Homepage,” Available: <https://www.openstreetmap.org>, accessed Jan. 8, 2026.