

TRƯỜNG ĐẠI HỌC QUỐC TẾ HỒNG BÀNG  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN THỰC TẬP

MÔN HỌC  
THỰC TẬP DOANH NGHIỆP

Đề tài: Summarization Studio – Xưởng tóm  
tắt thông minh theo khối lớp

Giảng viên hướng dẫn: Hoàng Ngọc Long

Giảng viên hướng dẫn: Nguyễn Minh Vũ

Mã số sinh viên: 2211110063

TP. Hồ Chí Minh, 2025

## LỜI CẢM ƠN

Để hoàn thành đề tài này, em đã nhận được sự hướng dẫn, giúp đỡ và góp ý nhiệt tình của quý thầy cô trường Đại Học Quốc Tế Hồng Bàng và thầy Hoàng Ngọc Long.

Em xin gửi lời biết ơn sâu sắc đến thầy Hoàng Ngọc Long đã dành nhiều thời gian và tâm huyết hướng dẫn nghiên cứu và giúp em hoàn thành môn học.

Em cũng xin chân thành cảm ơn đến quý thầy cô trường Đại học Quốc Tế Hồng Bàng, đặc biệt là những thầy cô đã tận tình dạy bảo cho em suốt thời gian học tập tại trường.

Em xin chân thành cảm ơn đến toàn thể ban lãnh đạo Công ty TNHH TMDV XCODI, cùng các cô, chú, anh, chị cán bộ công nhân viên Công ty TNHH TMDV XCODI đã tận tình hướng dẫn và tạo điều kiện giúp đỡ em trong suốt quá trình thực tập tại Công ty để em hoàn thành tốt báo cáo thực tập tốt nghiệp này.

Em đã có nhiều cố gắng hoàn thiện dự án bằng tất cả năng lực của mình, tuy nhiên không thể tránh khỏi nhiều thiếu sót, rất mong nhận được những đóng góp quý báu của quý thầy cô và các bạn.

TP.HCM, ... Tháng ... Năm 2025

Người thực hiện

Nguyễn Minh Vũ

## **TRANG CAM KẾT**

Tôi xin cam kết báo cáo thường kỳ này được hoàn thành dựa trên các kết quả thực hiện bài thực hành của tôi và các mã nguồn và kết quả này chưa được dùng cho bất cứ báo cáo của sinh viên nào khác.

*TP.HCM, ngày ... tháng ... năm 2025*

Người thực hiện

Nguyễn Minh Vũ

## CƠ QUAN/ ĐƠN VỊ

# CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

\_oOo\_

TP. Hồ Chí Minh, ngày 30 tháng 10 năm 2025

## **NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP**

Họ và tên sinh viên thực tập: Nguyễn Minh Vũ.....

Là sinh viên khóa 2022 thuộc Khoa Kỹ Thuật Công Nghệ Trường Đại học Quốc tế Hồng Bàng.

Thực tập tại Cơ Quan/ Đơn Vị: Công ty TNHH TMDV XCODI.....

Địa chỉ: 59 Nhật Tảo, Phường Vườn Lài, TP. Hồ Chí Minh .....

#### Phản nhận xét của Cơ Quan/ Đơn Vị

(Quý Cơ Quan/Đơn Vị có thể nhận xét theo các tiêu chí sau: Chấp hành nội quy kỷ luật, giờ làm việc, chấp hành phân công, tinh thần, thái độ thực tập, khả năng ứng dụng chuyên môn vào công việc cụ thể,...)

## **Đơn vị thực tập**

(ký ghi rõ họ tên, chức vụ và đóng dấu)

## NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP.HCM, Ngày ... Tháng ... Năm 2025

Chữ ký giảng viên

# DANH MỤC BIỂU ĐỒ HÌNH VẼ

Hình 2.3.1. Mô hình seq2seq_1 .....	18
Hình 2.3.2. Mô hình seq2seq_2 .....	18
Hình 2.3.3. Mô hình seq2seq_3 .....	18
Hình 2.3.4. Mô hình seq2seq_4 .....	19
Hình 2.3.5. Mô hình seq2seq_5 .....	19
Hình 2.3.6. Công thức softmax .....	20
Hình 2.3.7. Chú thích biến softmax .....	20
Hình 2.4.1. Kiến trúc AI Agent .....	22
Hình 2.4.2. Suy nghĩ trong hệ thống AI Agent .....	22
Hình 2.4.3. Các kiến trúc đa tác tử .....	25
Hình 2.4.4. Các kiến trúc đa tác tử khác .....	25
Hình 2.4.5. Chuỗi nhắc nhở hệ thống đa tác tử .....	26
Hình 2.4.6. Song song hóa hệ thống đa tác tử .....	26
Hình 2.4.7. Điều hướng hệ thống đa tác tử .....	26
Hình 2.4.8. Điều phối và phân chia nhiệm vụ trong hệ thống đa tác tử .....	26
Hình 2.4.9. Tự kiểm tra kết quả và cải thiện trong hệ thống đa tác tử .....	27
Hình 2.6.1. Những đặc điểm giữa các framework lớn .....	32
Hình 2.6.2. Đối tượng thích hợp trong việc sử dụng và phát triển dự án .....	33
Hình 3.1.1. Sơ đồ use case hệ thống tóm tắt văn bản .....	42
Hình 4.1.1. Sơ đồ flowchart hệ thống đa tác tử .....	47
Hình 4.1.2. Sơ đồ sequence diagram hệ thống đa tác tử .....	48
Hình 4.1.3. Sơ đồ ERD database .....	49
Hình 4.1.4. Sơ đồ schema diagram database .....	50
Hình 4.1.5. Sơ đồ sequence diagram API .....	54
Hình 4.1.6. Kiến trúc ứng dụng tóm tắt .....	55
Hình 4.3.1. Kiến trúc tổng thể hệ thống đa tác tử .....	58

# **DANH MỤC BIỂU ĐỒ BẢNG**

Bảng 2.4-1. Bảng demo hệ thống đơn tác tử .....	23
Bảng 2.4-2. Bảng demo hệ thống đa tác tử .....	28
Bảng 3.1-1. Bộ API của cuộc trò chuyện với hệ thống .....	38
Bảng 3.1-2. Bộ API của đọc lịch sử tóm tắt .....	38
Bảng 3.1-3. Bộ API hệ thống tóm tắt .....	38
Bảng 3.1-4. Bộ API lịch sử tóm tắt .....	39
Bảng 3.1-5. Bộ API phiên tóm tắt .....	40
Bảng 3.1-6. Bộ API thẻ tóm tắt .....	40
Bảng 3.1-7. Bộ API thẻ của hệ thống .....	40
Bảng 3.1-8. Bộ API quản lý thông tin người dùng .....	41
Bảng 3.1-9. Đặc tả use case chính của hệ thống tóm tắt .....	44
Bảng 4.1-1. Bảng User .....	50
Bảng 4.1-2. Bảng phiên tóm tắt .....	51
Bảng 4.1-3. Bảng tóm tắt .....	51
Bảng 4.1-4. Bảng lịch sử tóm tắt .....	52
Bảng 4.1-5. Bảng đọc lịch sử tóm tắt .....	52
Bảng 4.1-6. Bảng gắn nhãn tóm tắt .....	53
Bảng 4.1-7. Bảng quản lý id nhãn và tóm tắt .....	53
Bảng 4.2-1. Bảng thư viện sử dụng .....	56
Bảng 4.2-2. Bảng mô hình sử dụng .....	57
Bảng 4.3-1. Bảng mô tả chi tiết các Agent có trong hệ thống .....	59
Bảng 4.5-1. Kết quả của bộ API Message Controller .....	67
Bảng 4.5-2. Kết quả của bộ API Read History Controller .....	67
Bảng 4.5-3. Kết quả của bộ API Summary Controller .....	69
Bảng 4.5-4. Kết quả của bộ API Summary Session Controller .....	71
Bảng 4.5-5. Kết quả của bộ API Summary Tag Controller .....	72
Bảng 4.5-6. Kết quả của bộ API Tag Controller .....	72
Bảng 4.5-7. Kết quả của bộ API User Controller .....	73

## DANH MỤC THUẬT NGỮ VIỆT TẮT

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1	NLP	Natural Language Processing
2	AI	Artificial Intelligence
3	BERT	Bidirectional Encoder Representations from Transformers
4	mT5	Multilingual Text-to-Text Transfer Transformer
5	PhoBERT	Vietnamese Pre-trained BERT
6	GPU	Graphics Processing Unit
7	API	Application Programming Interface
8	ROUGE	Recall-Oriented Understudy for Gisting Evaluation
9	BLEU	Bilingual Evaluation Understudy
10	OCR	Optical Character Recognition
11	CSV	Comma-Separated Values
12	LSRS	Lexical-Semantic Relevance Scoring
13	IGO	Iterative Greedy Optimization
14	LLM	Large Language Model
15	TF-IDF	Term Frequency-Inverse Document Frequency
16	RNN	Recurrent Neural Network
17	LSTM	Long Short-Term Memory
18	MLM	Masked Language Model
19	NSP	Next Sentence Prediction
20	LSC	Longest Common Subsequence
21	PEFT	Parameter-Efficient Fine-Tuning
22	ADK	Agent Development Kit
23	MAS	Multi Agent System
24	seq2seq	Sequence-to-Sequence
25	ReAct	Reasoning Acting

# MỤC LỤC

LỜI CẢM ƠN.....	1
TRANG CAM KẾT .....	2
NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP.....	3
NHẬN XÉT CỦA GIÁNG VIÊN.....	4
DANH MỤC BIỂU ĐỒ HÌNH VẼ .....	5
DANH MỤC BIỂU ĐỒ BẢNG .....	6
DANH MỤC THUẬT NGỮ VIẾT TẮT .....	7
MỤC LỤC.....	8
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	13
1.1. Tổng quan .....	13
1.2. Lý do chọn đề tài .....	13
1.3. Mục tiêu đề tài .....	14
1.4. Ý nghĩa đề tài .....	15
1.4.1. Ý nghĩa khoa học .....	15
1.4.2. Ý nghĩa thực tiễn .....	15
1.5. Định hướng giải pháp.....	15
1.5.1. Định hướng chung .....	15
1.5.2. Định hướng triển khai.....	15
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	16
2.1. Giới thiệu về tóm tắt văn bản .....	16
2.1.1. Khái niệm.....	16
2.1.2. Phân loại .....	16
2.1.3. Các ứng dụng .....	16
2.1.4. Vai trò của tóm tắt trong giáo dục tiểu học .....	17
2.2. Tổng quan về xử lý ngôn ngữ tự nhiên .....	17
2.2.1. Khái niệm.....	17
2.2.2. NLP cho tiếng Việt .....	17
2.2.3. Vai trò của NLP trong hệ thống tóm tắt truyện và bài đọc.....	17
2.3. Mô hình LLMs và kiến trúc Transformer .....	18

2.3.1.	Tổng quan về LLMs (Large Language Models) .....	18
2.3.2.	Kiến trúc Transformer .....	19
2.3.3.	LLMs dựa trên Transformer .....	20
2.3.4.	Ưu điểm của kiến trúc Transformer trong bài toán tóm tắt.....	20
2.4.	Lý thuyết về đa tác tử (Multi-Agent System).....	21
2.4.1.	Đơn tác tử .....	21
	2.4.1.1. Khái niệm.....	21
	2.4.1.2. Đặc điểm của AI Agent .....	21
	2.4.1.3. Hạn chế của AI Agent.....	21
	2.4.1.4. Lợi ích và hiệu quả mang lại của hệ thống .....	21
	2.4.1.5. Kiến trúc của AI Agent.....	22
	2.4.1.6. Ví dụ về quá trình hoạt động của AI Agent.....	22
	2.4.1.7. Giải pháp xây dựng.....	23
2.4.2.	Đa tác tử.....	23
	2.4.2.1. Khái niệm.....	23
	2.4.2.2. Cấu trúc của hệ thống đa tác tử .....	24
	2.4.2.3. Ứng dụng của hệ thống .....	27
	2.4.2.4. Ưu điểm của mô hình hệ thống đa tác tử.....	27
	2.4.2.5. Xu hướng tương lai .....	27
	2.4.2.6. Thách thức .....	28
	2.4.2.7. Ví dụ điển hình .....	28
2.5.	Các tác tử trong hệ thống Summarization Studio.....	29
2.6.	Công nghệ và nền tảng phát triển .....	30
2.6.1.	Ngôn ngữ và môi trường lập trình .....	30
2.6.2.	Các công nghệ và thư viện chính.....	30
	2.6.2.1. Ollama.....	30
	2.6.2.2. LangChain.....	32
	2.6.2.3. LangGraph .....	32
	2.6.2.4. Chroma .....	33
	2.6.2.5. Jupyter Notebook.....	33

2.6.2.6.	Anaconda .....	34
2.6.3.	Môi trường triển khai.....	34
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG TÓM TẮT TRUYỆN, BÀI ĐỌC.....		35
3.1.	Phân tích yêu cầu hệ thống.....	35
3.1.1.	Mục tiêu hệ thống .....	35
3.1.1.1.	Mục tiêu tổng quát .....	35
3.1.1.2.	Mục tiêu cụ thể .....	35
3.1.1.3.	Kết quả mong đợi .....	36
3.1.2.	Phân tích yêu cầu chức năng .....	36
3.1.2.1.	Mục tiêu .....	36
3.1.2.2.	Các chức năng chính của hệ thống .....	36
3.1.2.3.	Các chức năng hỗ trợ và quản lý .....	38
3.1.3.	Sơ đồ use case.....	42
3.1.3.1.	Sơ đồ use case hệ thống tóm tắt văn bản .....	42
3.1.4.	Đặc tả use case.....	43
3.1.4.1.	Đặc tả use case tổng quan hệ thống .....	43
3.2.	Thu thập và xử lý dữ liệu (đang phát triển).....	45
3.2.1.	Nguồn dữ liệu .....	45
3.2.2.	Quy trình thu thập và xử lý dữ liệu.....	45
3.2.3.	Định hướng huấn luyện dữ liệu .....	45
4.1.	Kiến trúc hệ thống .....	47
4.1.1.	Thiết kế hệ thống đa tác tử .....	47
4.1.1.1.	Sơ đồ luồng hoạt động .....	47
4.1.1.2.	Sơ đồ Sequence Diagram.....	48
4.1.2.	Thiết kế cơ sở dữ liệu .....	49
4.1.2.1.	Sơ đồ Entity Relationship Diagram (ERD) .....	49
4.1.2.2.	Sơ đồ Schema diagram .....	50
4.1.3.	Thiết kế hệ thống API và tích hợp .....	54
4.1.3.1.	Sequence Diagram .....	54
4.1.3.2.	Kiến trúc ứng dụng tóm tắt .....	55

CHƯƠNG 4: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG TÓM TẮT .....	55
4.2. Môi trường và công cụ phát triển .....	55
4.2.1. Môi trường phát triển.....	55
4.2.2. Môi trường phát triển web và di động .....	56
4.2.3. Công cụ và thư viện hỗ trợ .....	56
4.2.4. Thư viện và mô hình sử dụng .....	56
4.2.4.1. Thư viện chính.....	56
4.2.4.2. Mô hình sử dụng.....	56
4.2.5. Cấu hình phần cứng .....	57
4.3. Xây dựng pipeline đa tác tử (Multi-Agent Implementation) .....	57
4.3.1. Giới thiệu chung .....	57
4.3.2. Kiến trúc tổng thể .....	57
4.3.3. Mô tả chi tiết các Agent.....	58
4.3.4. Cơ chế phối hợp.....	59
4.3.5. Kết quả và lợi ích.....	59
4.4. Triển khai hệ thống đa tác tử.....	59
4.4.1. Giới thiệu .....	59
4.4.2. Các bước triển khai chính .....	59
4.4.2.1. Cài đặt môi trường hệ thống .....	59
4.4.2.2. Phân công nhiệm vụ cụ thể và xây dựng các Agent .....	60
4.4.2.3. Quá trình triển khai hệ thống .....	63
4.4.2.4. Kết quả triển khai.....	63
4.5. Xây dựng API.....	67
4.5.1. Message Controller .....	67
4.5.2. Read History Controller.....	67
4.5.3. Summary Controller .....	69
4.5.4. Summary History Controller .....	71
4.5.5. Summary Session Controller .....	71
4.5.6. Summary Tag Controller .....	72
4.5.7. Tag Controller.....	72

4.5.8. User Controller .....	73
4.6. Đánh giá kết quả.....	74
4.6.1. Hệ thống đa tác tử.....	74
4.6.2. Hệ thống ứng dụng Spring Boot.....	74
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỀN .....	75
5.1. Đánh giá kết quả đạt được .....	75
5.2. Hạn chế và thách thức .....	75
5.3. Hướng phát triển tương lai .....	76
Tài liệu tham khảo .....	78
KẾT LUẬN.....	79

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

Chương này giới thiệu bài toán tóm tắt văn bản tự động và nhu cầu hỗ trợ học sinh tiểu học đọc hiểu tốt hơn. Nghiên cứu xác định khoảng trống trong việc áp dụng công nghệ xử lý ngôn ngữ tự nhiên vào giáo dục, đề xuất xây dựng hệ thống tóm tắt bằng hai phương pháp: trích xuất và diễn giải. Mục tiêu là giúp học sinh dễ dàng nắm nội dung chính, đồng thời làm rõ ý nghĩa khoa học và thực tiễn của đề tài qua phân tích các nghiên cứu liên quan.

## 1.1. Tổng quan

Tóm tắt văn bản tự động là một bài toán cốt lõi trong xử lý ngôn ngữ tự nhiên (NLP), giúp rút gọn thông tin để hỗ trợ học tập và nghiên cứu. Quá trình này bắt nguồn từ ý tưởng trích xuất ý chính từ văn bản dựa trên từ khóa, đặt nền móng cho các phương pháp tóm tắt hiện đại. Tóm tắt được thực hiện theo hai hướng chính: tóm tắt trích xuất và tóm tắt diễn giải.

- **Tóm tắt trích xuất (Extractive Summarization):** Sử dụng thuật toán thống kê và học máy để chọn các câu quan trọng nhất từ văn bản gốc, dựa trên điểm số từ các tiêu chí như tần suất từ khóa, vị trí câu và ngữ nghĩa. Các câu có điểm cao được giữ nguyên để tạo bản tóm tắt.
- **Tóm tắt diễn giải (Abstractive Summarization):** Mô phỏng cách con người tóm tắt, phương pháp này dùng mô hình ngôn ngữ để sinh ra câu mới, diễn đạt ý chính ngắn gọn và tự nhiên.

Trong dự án này, **hệ thống đa tác tử (MAS)** được thiết kế dựa trên hướng phát triển của hai cách tóm tắt trích xuất và diễn giải. Nhằm biến bản thiết kế lý thuyết thành một hệ thống hoạt động thực tế, có khả năng tự động nhận diện, tóm tắt và đánh giá chất lượng bài đọc tiếng Việt cho học sinh tiểu học. Hệ thống hoàn chỉnh giúp tự động hóa quy trình tóm tắt bài đọc theo nhiều cấp độ, đảm bảo tính minh bạch, kiểm soát được độ khó và từ vựng theo khối lớp, đồng thời cung cấp công cụ hữu ích cho giáo viên trong việc kiểm tra và điều chỉnh nội dung học tập.

Mục tiêu là xây dựng hệ thống tóm tắt văn bản dành cho học sinh tiểu học, giúp các em dễ dàng hiểu truyện và bài đọc, từ đó nâng cao hiệu quả học tập và ứng dụng tóm tắt tiếng Việt trong giáo dục.

## 1.2. Lý do chọn đề tài

Trong bối cảnh giáo dục hiện đại, nhu cầu về các công cụ hỗ trợ học tập cho học sinh tiểu học ngày càng trở nên cần thiết. Khối lượng thông tin lớn từ sách vở và tài liệu khiến trẻ phải tiếp cận nhiều nội dung, nhưng khả năng tập trung và đọc hiểu của các

em còn hạn chế. Học sinh tiểu học thường gặp khó khăn khi xử lý các văn bản dài, dẫn đến hiệu quả học tập giảm sút.

Trước thực tế đó, việc xây dựng một hệ thống hỗ trợ tóm tắt truyện và bài đọc tự động, thân thiện và phù hợp với lứa tuổi tiểu học là cần thiết. Hệ thống này không chỉ giúp học sinh rèn luyện kỹ năng đọc hiểu mà còn hỗ trợ giáo viên và phụ huynh trong việc kiểm tra, đánh giá khả năng tiếp thu của học sinh.

Bên cạnh đó, sự phát triển của các mô hình ngôn ngữ tự nhiên (NLP) và trí tuệ nhân tạo (AI) hiện nay như PhoBERT, mT5, GPT, BART, hoặc các mô hình Transformer khác mở ra cơ hội lớn cho việc ứng dụng công nghệ vào giáo dục. Việc triển khai một hệ thống tóm tắt dựa trên AI không chỉ mang ý nghĩa học thuật mà còn có giá trị thực tiễn cao, góp phần đổi mới phương pháp học tập và giảng dạy theo hướng thông minh, hiện đại và cá nhân hóa.

Xuất phát từ thực tế này, dự án ứng dụng các mô hình LLMs và AI vào việc nghiên cứu xây dựng hệ thống tóm tắt giúp trẻ vượt qua rào cản đọc hiểu, dự án là một công cụ hữu ích, góp phần nâng cao hiệu quả học tập và khơi dậy hứng thú đọc hiểu cho học sinh. Đồng thời mở rộng ứng dụng NLP và AI vào giáo dục, tối ưu hóa quá trình tiếp thu thông tin cho học sinh.

### 1.3. Mục tiêu đề tài

Xây dựng một hệ thống tóm tắt tự động có khả năng rút gọn nội dung truyện và bài đọc tiếng Việt sao cho ngắn gọn, dễ hiểu, phù hợp với lứa tuổi học sinh tiểu học, hỗ trợ quá trình học tập, đọc hiểu và ôn luyện kiến thức.

- Phân tích yêu cầu và đặc thù nội dung của các truyện, bài đọc dành cho học sinh tiểu học (về độ dài, ngôn ngữ, cấu trúc, từ vựng, độ khó).
- Thiết kế và xây dựng hệ thống tóm tắt kết hợp hai phương pháp chính:
  - **Tóm tắt trích xuất (extractive summarization)** – lựa chọn các câu quan trọng trong văn bản gốc.
  - **Tóm tắt diễn giải (abstractive summarization)** – tạo ra câu mới thể hiện lại ý chính bằng ngôn ngữ tự nhiên.
- Triển khai hệ thống đa tác tử cho phép người dùng nhập nội dung truyện/bài đọc và nhận kết quả tóm tắt tức thời.
- Đánh giá chất lượng tóm tắt theo các tiêu chí như: độ chính xác ngôn ngữ nghĩa, mức độ ngắn gọn, tính tự nhiên, và mức độ phù hợp với lứa tuổi học sinh.

## 1.4. Ý nghĩa đề tài

### 1.4.1. Ý nghĩa khoa học

- Góp phần nghiên cứu và ứng dụng các mô hình ngôn ngữ hiện đại (LLMs) vào lĩnh vực xử lý tiếng Việt, đặc biệt là nhiệm vụ tóm tắt văn bản — một bài toán phức tạp trong xử lý ngôn ngữ tự nhiên (NLP).
- Cung cấp một hướng đi mới – một hệ thống kết hợp giữa tóm tắt trích xuất và tóm tắt diễn giải, giúp cải thiện độ chính xác và tính tự nhiên của bản tóm tắt.
- Làm cơ sở cho các nghiên cứu tiếp theo về tóm tắt văn bản, sinh câu hỏi tự động, hoặc rút trích bài học từ nội dung giáo dục.

### 1.4.2. Ý nghĩa thực tiễn

- Giúp học sinh tiểu học dễ dàng nắm bắt nội dung chính của truyện và bài đọc, phát triển kỹ năng đọc hiểu và tư duy tổng hợp.
- Hỗ trợ giáo viên và phụ huynh trong việc kiểm tra, giảng dạy và định hướng học tập cho học sinh thông qua các bản tóm tắt ngắn gọn, phù hợp với độ tuổi.
- Tiết kiệm thời gian cho người học trong việc nắm bắt nhanh kiến thức, đồng thời tăng hứng thú đọc sách và học tiếng Việt.
- Có thể mở rộng ứng dụng trong giáo dục thông minh, sách điện tử, hoặc nền tảng học tập trực tuyến, góp phần thúc đẩy chuyển đổi số trong giáo dục Việt Nam.

## 1.5. Định hướng giải pháp

### 1.5.1. Định hướng chung

Hệ thống được xây dựng theo mô hình đa tác tử (Multi-Agent System - MAS), trong đó mỗi agent đảm nhận một vai trò cụ thể trong quá trình phân tích – hiểu – tóm tắt – đánh giá văn bản.

Cách tiếp cận này giúp hệ thống phân rã nhiệm vụ phức tạp thành nhiều bước nhỏ, hoạt động độc lập nhưng có phối hợp, tương tự như cách con người đọc, hiểu, và rút ý chính từ nội dung.

### 1.5.2. Định hướng triển khai

- Ngôn ngữ lập trình: Python.
- Framework để xuất: LangGraph hoặc LangChain để xây dựng pipeline giữa các agent.
- Mô hình ngôn ngữ sử dụng: PhoBERT (trích xuất), mT5 (diễn giải), Ollama – llama3:8b (Sử dụng cho việc demo flow System).
- Giao diện: xây dựng bằng React kết hợp với Spring Boot và Flask.
- Lưu trữ dữ liệu: MySQL cho hệ thống backend và Chroma cho phiên lưu trữ dữ liệu của đa tác tử.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Giới thiệu về tóm tắt văn bản

#### 2.1.1. Khái niệm

Tóm tắt văn bản là quá trình rút gọn nội dung của một văn bản gốc thành một phiên bản ngắn hơn nhưng vẫn giữ lại được các ý chính, nội dung cốt lõi và thông điệp quan trọng. Mục tiêu của tóm tắt là giúp người đọc hiểu nhanh nội dung chính mà không cần đọc toàn bộ văn bản ban đầu.

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), tóm tắt văn bản là một trong những bài toán quan trọng và thách thức, bởi nó đòi hỏi hệ thống phải có khả năng hiểu ngữ nghĩa, ngữ pháp và bối cảnh của văn bản.

#### 2.1.2. Phân loại

- Tóm tắt trích xuất (Extractive Summarization)

Phương pháp này chọn trực tiếp các câu quan trọng trong văn bản gốc để tạo thành bản tóm tắt. Không sinh câu mới mà chỉ chọn lọc và sắp xếp lại những câu có giá trị thông tin cao.

Các kỹ thuật phổ biến:

- Thống kê từ khóa: TF-IDF, độ quan trọng của câu.
- Mô hình đồ thị: TextRank, LexRank.
- Mô hình ngôn ngữ: BERT, PhoBERT.

**Ưu điểm:** đơn giản, nhanh, dễ triển khai.

**Nhược điểm:** bản tóm tắt có thể thiếu tính tự nhiên và chưa mạch lạc.

- Tóm tắt diễn giải (Abstractive Summarization)

Phương pháp này hiểu nội dung văn bản gốc rồi diễn đạt lại bằng ngôn ngữ tự nhiên mới, giống như con người viết lại bằng lời của mình. Thường sử dụng các mô hình học sâu (Deep Learning) như Transformer, BART, T5, mT5, GPT.

**Ưu điểm:** kết quả tự nhiên, mạch lạc, có thể rút gọn mạnh hơn.

**Nhược điểm:** phức tạp, đòi hỏi dữ liệu huấn luyện lớn và mô hình mạnh.

#### 2.1.3. Các ứng dụng

1. Giáo dục: tạo tóm tắt bài học, giúp học sinh nắm nội dung nhanh hơn.
2. Báo chí: sinh tóm tắt tự động cho tin tức, báo mạng.
3. Nghiên cứu: tóm tắt tài liệu học thuật, báo cáo khoa học.
4. Trợ lý ảo: tổng hợp nội dung email, hội thoại hoặc văn bản dài.
5. Y tế – pháp luật – hành chính: rút gọn văn bản, biên bản, hoặc hồ sơ.

#### **2.1.4. Vai trò của tóm tắt trong giáo dục tiểu học**

Trong bối cảnh giáo dục hiện nay, việc áp dụng tóm tắt tự động giúp:

- Học sinh hiểu nhanh nội dung truyện, bài đọc mà không bị quá tải thông tin.
- Giáo viên tạo tài liệu học tập, bài giảng rút gọn hiệu quả hơn.
- Thúc đẩy việc ứng dụng trí tuệ nhân tạo vào giáo dục, hỗ trợ quá trình học tập chủ động và thông minh.

### **2.2. Tổng quan về xử lý ngôn ngữ tự nhiên**

#### **2.2.1. Khái niệm**

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một lĩnh vực thuộc trí tuệ nhân tạo (Artificial Intelligence – AI) và khoa học máy tính, nghiên cứu các phương pháp giúp máy tính có thể hiểu, phân tích và sinh ngôn ngữ của con người một cách tự nhiên.

Mục tiêu của NLP là tạo ra các hệ thống có khả năng:

- Hiểu được nội dung và ngữ nghĩa trong NLP (tiếng nói, văn bản).
- Tương tác với con người thông qua ngôn ngữ nói hoặc viết.
- Sinh ngôn ngữ tự nhiên để trả lời, tóm tắt, dịch, hoặc mô tả thông tin.

Nói cách khác, NLP là cầu nối giữa ngôn ngữ con người và máy tính, giúp máy có thể xử lý dữ liệu phi cấu trúc (như văn bản, hội thoại, tin tức, tài liệu).

#### **2.2.2. NLP cho tiếng Việt**

Việc xử lý tiếng Việt gặp nhiều thách thức do:

- Đặc điểm ngôn ngữ phức tạp: nhiều từ ghép, dấu thanh, và hiện tượng đồng âm dị nghĩa.
- Tách từ khó: không có dấu cách rõ ràng giữa các từ.
- Thiếu dữ liệu lớn và gán nhãn sẵn so với tiếng Anh.

Một số công cụ và mô hình NLP tiếng Việt phổ biến:

- Thư viện tiền xử lý: Underthesea, PyVi, VnCoreNLP.
- Mô hình ngôn ngữ: PhoBERT, ViT5, mT5, BARTpho.
- Ứng dụng: tóm tắt văn bản, nhận dạng thực thể, phân tích cảm xúc, chatbot tiếng Việt.

#### **2.2.3. Vai trò của NLP trong hệ thống tóm tắt truyện và bài đọc**

Trong đề tài này, NLP đóng vai trò nền tảng để:

- Hiểu và phân tích ngữ nghĩa câu chuyện.
- Xác định ý chính, nhân vật, và mối quan hệ giữa các đoạn.
- Tạo ra bản tóm tắt ngắn gọn, mạch lạc, phù hợp lứa tuổi học sinh.

- Hỗ trợ các tác tử trong hệ thống Multi-Agent (như OCR Reader, Extractor, Abstracter, Evaluator, ...) giao tiếp và xử lý ngôn ngữ một cách thống nhất.

## 2.3. Mô hình LLMs và kiến trúc Transformer

### 2.3.1. Tổng quan về LLMs (Large Language Models)

Large Language Models (LLMs) là các mô hình học sâu được huấn luyện trên lượng dữ liệu văn bản khổng lồ nhằm học và hiểu ngữ nghĩa, cú pháp, cũng như các mối quan hệ ngôn ngữ tự nhiên. Nguyên lý hoạt động dựa trên kiến trúc transformer.

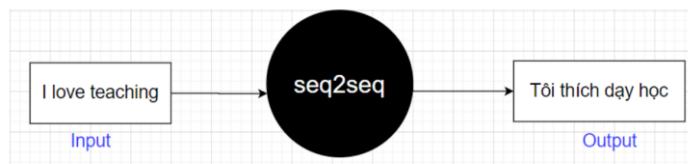
Các mô hình này có khả năng:

- Sinh văn bản tự nhiên (text generation).
- Hiểu và tóm tắt văn bản.
- Trả lời câu hỏi, dịch ngôn ngữ, hoặc chuyển đổi truy vấn tự nhiên sang ngôn ngữ lập trình (như SQL).

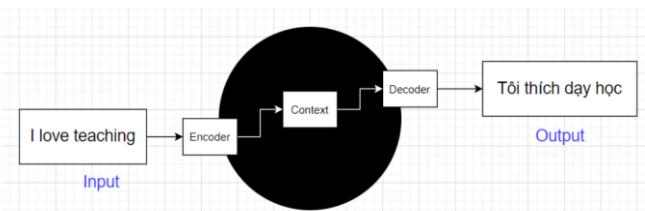
Các LLM nổi bật hiện nay gồm:

- GPT (Generative Pre-trained Transformer) của OpenAI.
- T5 / mT5 (Text-to-Text Transfer Transformer) của Google.
- LLaMA của Meta.
- Phi-3, Qwen2, Mistral, Claude, Gemini, v.v.

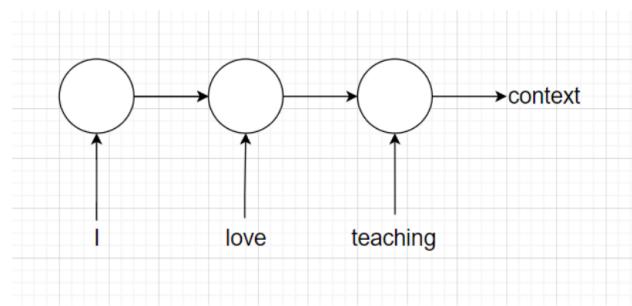
Ví dụ điển hình là mô hình seq2seq:



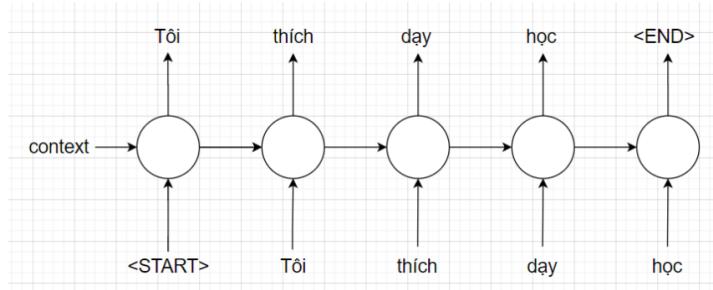
Hình 2.3.1. Mô hình seq2seq\_1



Hình 2.3.2. Mô hình seq2seq\_2



Hình 2.3.3. Mô hình seq2seq\_3



Hình 2.3.4. Mô hình seq2seq\_4



Hình 2.3.5. Mô hình seq2seq\_5

### 2.3.2. Kiến trúc Transformer

**Transformer** là kiến trúc mạng nơ-ron, chúng nền tảng của hầu hết các LLM hiện nay, được giới thiệu lần đầu bởi Vaswani et al. (2017) trong bài báo “Attention Is All You Need”. Kiến trúc này thay thế hoàn toàn cơ chế RNN và LSTM truyền thống bằng cơ chế Self-Attention.

- Thành phần chính của Transformer:
- Input Embedding: Biến đổi các từ thành vector số (word embeddings).
- Positional Encoding: Thêm thông tin vị trí của từ trong câu để mô hình hiểu thứ tự.
- Encoder: Gồm nhiều lớp self-attention và feed-forward network, mã hóa ngữ cảnh của đầu vào.
- Decoder: Dùng cơ chế attention để sinh ra dựa trên thông tin từ encoder.
- Multi-Head Attention: Cho phép mô hình học nhiều mối quan hệ giữa các từ trong cùng một lúc.

#### Công thức tính Attention:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

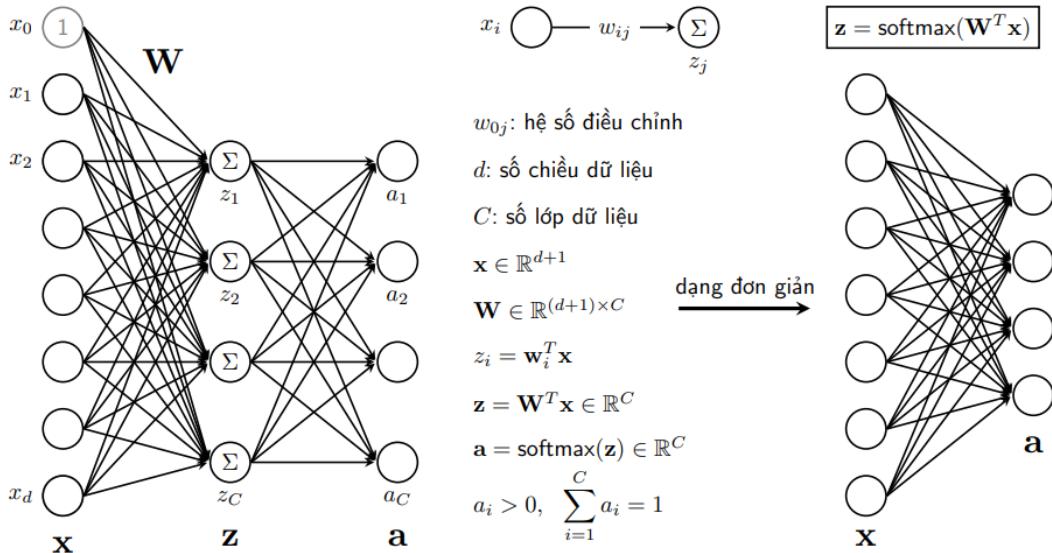
Trong đó:

- Q, K, V lần lượt là ma trận Query, Key, và Value.
- $d_k$  là kích thước vector Key.

#### Công thức tính Softmax:

$$a_i = \frac{\exp(z_i)}{\sum_{j=1}^C \exp(z_j)}, \quad \forall i = 1, 2, \dots, C.$$

Hình 2.3.6. Công thức softmax



Hình 2.3.7. Chú thích biến softmax

### 2.3.3. LLMs dựa trên Transformer

Các mô hình LLM hiện đại thường mở rộng Transformer bằng:

- Huấn luyện trên quy mô dữ liệu cực lớn (hàng trăm tỷ token).
- Sử dụng cơ chế attention tối ưu hóa (như FlashAttention, Rotary Embedding).
- Tối ưu bộ nhớ và tốc độ suy luận thông qua kỹ thuật như quantization hoặc LoRA fine-tuning.

Ví dụ:

- mT5 (Google): mô hình đa ngôn ngữ dựa trên Transformer Encoder–Decoder.
- PhoBERT (VinAI): mô hình Transformer Encoder tối ưu cho tiếng Việt.
- Qwen2.5, GPT-4, Gemini 2.5: các mô hình LLM thế hệ mới dựa hoàn toàn trên cải tiến của Transformer.

### 2.3.4. Ưu điểm của kiến trúc Transformer trong bài toán tóm tắt

1. Có khả năng hiểu ngữ cảnh toàn cục thay vì tuần tự như RNN.
2. Dễ dàng fine-tune cho các nhiệm vụ cụ thể (tóm tắt, dịch, sinh SQL, v.v.).
3. Cho phép xử lý song song trên GPU, giúp rút ngắn thời gian huấn luyện.
4. Phù hợp với ngôn ngữ tự nhiên tiếng Việt, đặc biệt khi kết hợp mô hình như mT5 hoặc PhoBERT.

## 2.4. Lý thuyết về đa tác tử (Multi-Agent System)

### 2.4.1. Đơn tác tử

#### 2.4.1.1. Khái niệm

Trong ngữ cảnh máy tính hay tự động hóa, “Agent” (tác tử) là một hệ thống hoặc chương trình máy tính có khả năng nhận thức môi trường thông qua cảm biến, xử lý thông tin và thực hiện hành động để đạt được mục tiêu cụ thể. Các Agent có thể hoạt động độc lập hoặc tương tác với người dùng và môi trường xung quanh.

#### 2.4.1.2. Đặc điểm của AI Agent

- Nhận thức môi trường: Sử dụng cảm biến để thu thập thông tin từ môi trường.
- Ra quyết định: Phân tích thông tin và đưa ra hành động phù hợp.
  - Xây dựng các quy tắc hoặc thuật toán xác định hành động.  
Ví dụ: Nếu nhiệt độ vượt quá 30 độ → bật quạt; nếu mức nước giảm dưới 20% → thông báo cho người sử dụng.
- Thực hiện hành động: Áp dụng các hành động đã quyết định để đạt được mục tiêu.
  - Gọi thực hiện hành động theo cơ chế API, hoặc chạy chương trình máy tính,...
- Tự động hóa: Có thể hoạt động mà không cần sự can thiệp trực tiếp của con người.

#### 2.4.1.3. Hạn chế của AI Agent

- Hoạt động theo bộ quy tắc đã định sẵn.
- Chỉ áp dụng các trường hợp đơn giản (ít tham số / môi trường đơn giản), các quy tắc định sẵn.
- Có mục tiêu cụ thể và chức năng hạn chế.
- Một số tình huống không thể dùng quy tắc (nhiều tham số / môi trường phức tạp).
- Không thích nghi và học hỏi rất thấp, không xử lý tốt tình huống ngoài dự đoán.

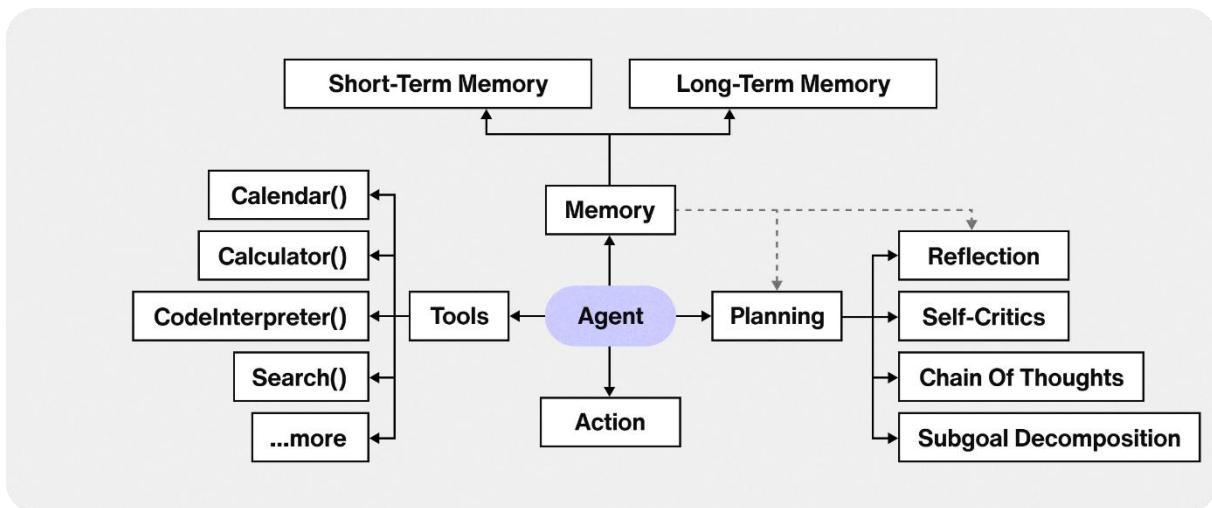
#### 2.4.1.4. Lợi ích và hiệu quả mang lại của hệ thống

- Không suy nghĩ, phân tích.
- Đưa ra kế hoạch thực thi.
- Đưa ra quyết định.
- Thực hiện hành động.
- Phản ánh khi có phản hồi từ môi trường.
- Đưa ra hành động tiếp theo.
- Học từ những phản hồi với môi trường.

Tác tử AI là một chường trình nhận đầu vào, suy nghĩ và hành động để hoàn thành một nhiệm vụ bằng cách sử dụng công cụ, trí nhớ và tri thức. Có suy nghĩ và hành động như con người.

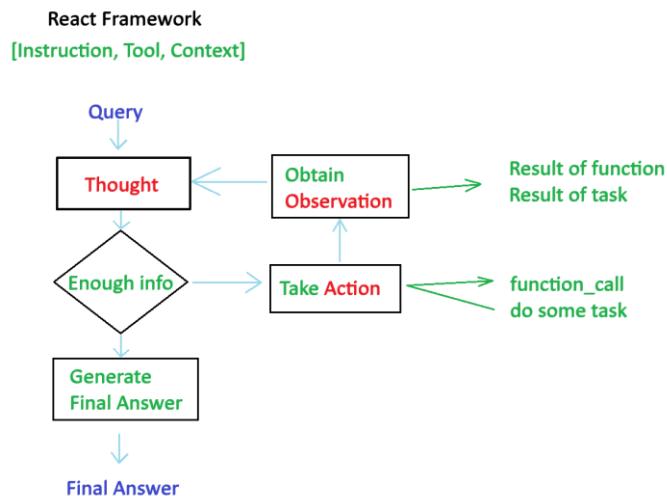
#### 2.4.1.5. Kiến trúc của AI Agent

- Phản ánh → nhìn lại để học hỏi.
- Tự phê bình → tự đánh giá, tránh sai lầm.
- Chuỗi suy nghĩ → lập luận theo bước, dễ kiểm soát.
- Phân tích mục tiêu phụ → chia nhỏ vấn đề để nhiều agent phối hợp hiệu quả.



Hình 2.4.1. Kiến trúc AI Agent

#### 2.4.1.6. Ví dụ về quá trình hoạt động của AI Agent



Hình 2.4.2. Suy nghĩ trong hệ thống AI Agent

Bảng 2.4.1. Bảng demo hệ thống đơn tác tử

User	Tìm vé máy bay?
AI Agent	<ul style="list-style-type: none"> <li>- Thought: Câu hỏi này hỏi này chưa đủ thông tin, yêu cầu người dùng bổ sung.</li> <li>- Response: Bạn muốn đi từ đâu, đến đâu, vào ngày nào ?</li> </ul>
User	Từ Hà Nội
AI Agent	<ul style="list-style-type: none"> <li>- Thought: Chưa đủ thông tin, yêu cầu người dùng bổ sung.</li> <li>- Response: Bạn muốn đến đâu, vào ngày nào ?</li> </ul>
User	Đến TP. Hồ Chí Minh, ngày 7/8/2025
AI Agent	<ul style="list-style-type: none"> <li>- Thought: Đã đủ thông tin, cần công cụ tìm chuyến bay.</li> <li>- Action: &lt;&lt;function_call&gt;&gt; <input type="checkbox"/> search_flight(...)</li> <li>- Observation: API Result: [{"time": "...", "price": "..."}]</li> <li>- Thought: Danh sách chuyến bay là ...</li> <li>- Response: “Danh sách chuyến bay là à trời nắng”.</li> </ul>

**ReAct Prompting:** Giúp cho LLMs tích hợp được các phân tích, đưa ra những hành động cần thực hiện theo một cách có cấu trúc.

#### 2.4.1.7. Giải pháp xây dựng

AI Agent: Kết hợp điểm mạnh của LLMs và các công cụ.

- Khả năng phân tích xử lý ngôn ngữ tự nhiên mạnh mẽ
  - Có khả năng phân tích và hiểu yêu cầu người dùng.
  - Thực hiện được các thao tác phân tích văn bản.
  - Ghi nhớ và tổng hợp kiến thức từ dữ liệu huấn luyện.
  - Đưa ra kế hoạch thực thi.
  - Tổng hợp phân tích kết quả.
- Khả năng truy xuất của những công cụ với nguồn dữ liệu bên ngoài
  - API truy xuất dữ liệu bên ngoài
  - Các công cụ tìm kiếm
  - Công cụ thực thi mã nguồn
  - API điều khiển thiết bị

### 2.4.2. Đa tác tử

#### 2.4.2.1. Khái niệm

Hệ thống đa tác tử (Multi-Agent System – MAS) là một mô hình tính toán bao gồm nhiều tác tử (agent) cùng hoạt động trong một môi trường chung, mỗi tác tử có khả

năng quan sát, suy luận, ra quyết định và tương tác với các tác tử khác để hoàn thành mục tiêu riêng hoặc chung của toàn hệ thống.

Mỗi tác tử trong hệ thống đa tác tử thường có các đặc tính cơ bản sau:

- **Tự chủ (Autonomy):** Tác tử có khả năng hoạt động độc lập mà không cần điều khiển trực tiếp.
- **Nhận thức (Perception):** Có khả năng cảm nhận trạng thái của môi trường xung quanh.
- **Hành động (Action):** Có thể thực hiện hành vi nhằm thay đổi trạng thái môi trường hoặc đạt mục tiêu cụ thể.
- **Tương tác (Interaction):** Có thể trao đổi thông tin, cộng tác hoặc cạnh tranh với các tác tử khác.

#### **2.4.2.2. Cấu trúc của hệ thống đa tác tử**

Một hệ thống đa tác tử thường bao gồm ba thành phần chính:

1. **Tác tử (Agents):** Các thực thể thông minh có mục tiêu và khả năng xử lý độc lập.
2. **Môi trường (Environment):** Không gian mà trong đó các tác tử hoạt động và tác động lẫn nhau.
3. **Cơ chế giao tiếp (Communication Mechanism):** Cho phép các tác tử chia sẻ thông tin hoặc phối hợp hành động (thường sử dụng giao thức như ACL – Agent Communication Language).

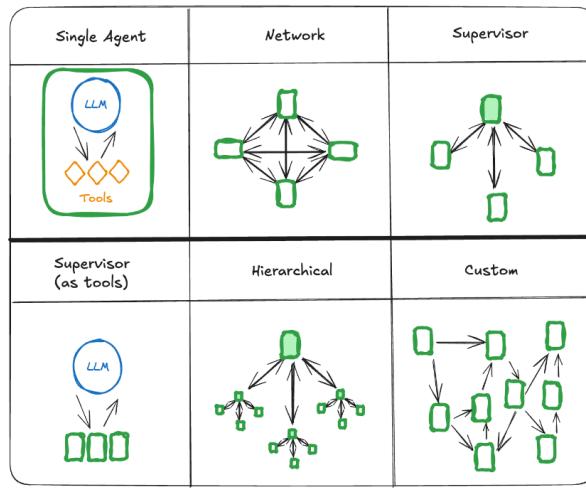
Mỗi quan hệ giữa các tác tử có thể mang tính:

1. **Hợp tác (Cooperative):** Các tác tử phối hợp để đạt được mục tiêu chung.
2. **Cạnh tranh (Competitive):** Các tác tử có mục tiêu xung đột, cố gắng tối ưu lợi ích riêng.
3. **Lai (Hybrid):** Kết hợp cả hai dạng trên tùy vào ngữ cảnh nhiệm vụ.

##### **2.4.2.2.1. Giao tiếp trong hệ thống đa tác tử**

Các cách giao tiếp giữa agent với agent:

- **Network:** Mỗi tác tử có thể giao tiếp trực tiếp, tự do lựa chọn tác tử kế tiếp.
- **Supervisor:** Mỗi tác tử giao tiếp với 1 tác tử quản lý, agent quản lý sẽ quyết định sẽ gọi agent kế tiếp.
- **Supervisor (tools):** Các tác tử có thể được xem là các tools, supervisor quyết định gọi các tác tử và truyền tham số tương ứng.
- **Hierarchical:** Là trường hợp tổng quát của kiến trúc supervisor.
- **Custom:** Tùy biến, mỗi tác tử có thể kết nối đến một vài tác tử, tác tử sẽ tự quyết định tác tử nào sẽ được gọi kế tiếp.



Hình 2.4.3. Các kiến trúc đa tác tử

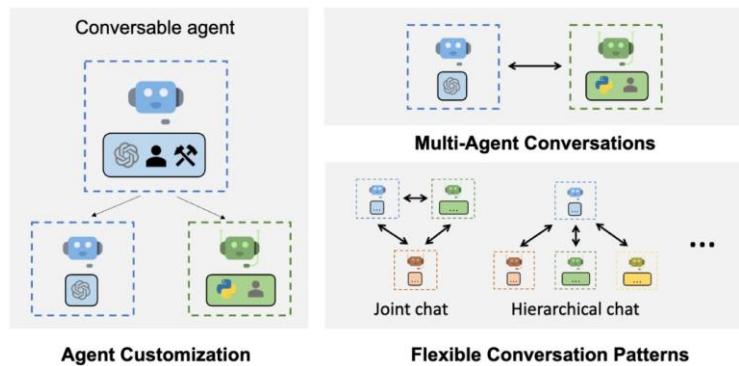
Trong các hệ thống đa tác tử, handoffs và tool calls là hai cách chính để các tác tử giao tiếp và chuyển giao quyền kiểm soát cho nhau.

- “**Handoffs**” là một mẫu tương tác phổ biến trong các hệ thống đa tác tử, nơi một tác tử chuyển giao quyền điều khiển cho một tác tử khác.
  - Handoffs cho phép xác định:
    - o destination: tác tử đích để điều hướng tới.
    - o payload: thông tin để truyền cho tác tử đó.
  - Áp dụng cho các kiến trúc Network, Supervisor, Hierarchical.
  - Có thể truyền luôn trạng thái toàn cục: lịch sử toàn bộ trò chuyện.
- **Tool calls (Gọi công cụ):**
  - Đây là trường hợp đặc biệt của kiến trúc Supervisor (tool-calling).
  - Trong kiến trúc này, các tác tử riêng lẻ có thể được biểu diễn dưới dạng các công cụ.

#### 2.4.2.2. Một số kiến trúc khác

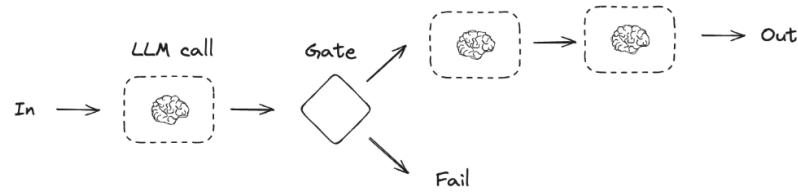
Một số kiến trúc khác như:

- Agent customization
- Multi-Agent Conversations
- Flexible Conversation Patterns



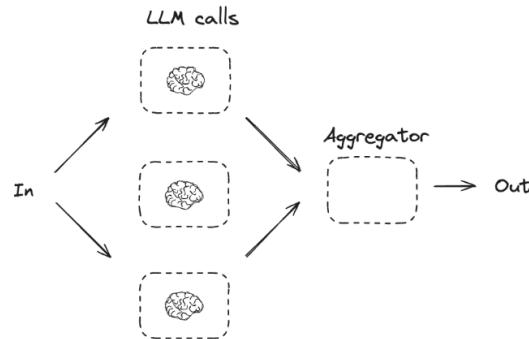
Hình 2.4.4. Các kiến trúc đa tác tử khác

- Prompt chaining (Chuỗi nhắc nhở)



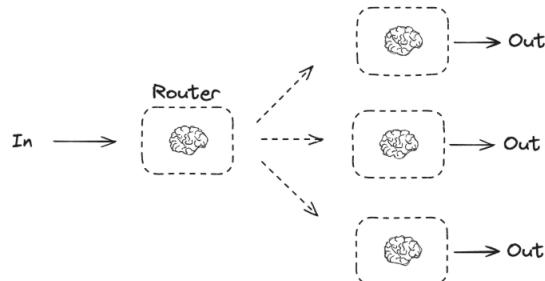
Hình 2.4.5. Chuỗi nhắc nhở hệ thống đa tác tử

- Parallelization (Song song hóa)



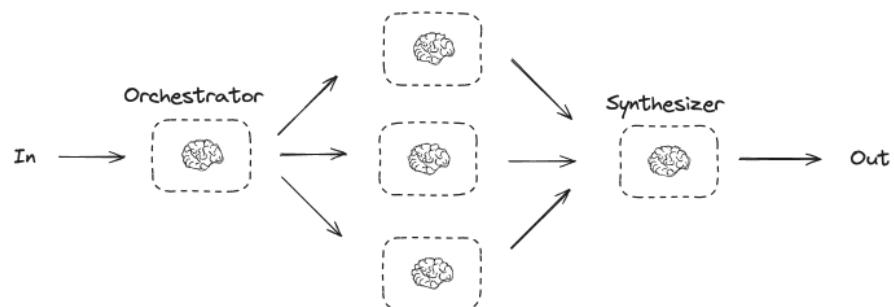
Hình 2.4.6. Song song hóa hệ thống đa tác tử

- Routing (Intent-Based Routing): Điều hướng yêu cầu đến agent hoặc công cụ phù hợp.



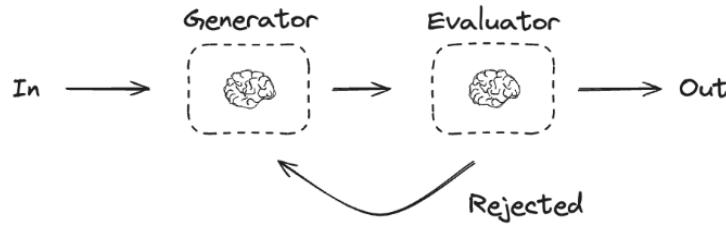
Hình 2.4.7. Điều hướng hệ thống đa tác tử

- Orchestrator-Worker (Planner-Executor): Một agent đóng vai trò điều phối (planner), phân chia nhiệm vụ cho các worker khác.



Hình 2.4.8. Điều phối và phân chia nhiệm vụ trong hệ thống đa tác tử

- Evaluator-Optimizer (Self-Reflection / Feedback Loop): Hệ thống tự kiểm tra lại kết quả và cải thiện.



Hình 2.4.9. Tự kiểm tra kết quả và cải thiện trong hệ thống đa tác tử

#### 2.4.2.3. Ứng dụng của hệ thống

- Trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên
- Hệ thống giao thông thông minh
- Thị trường kinh tế
- Chăm sóc khách hàng
- An ninh mạng
- Y tế thông minh
- Khoa học dữ liệu

AI Agent & Multi-Agent Systems: Tạo ra giải pháp thông minh, hợp tác và tối ưu hóa trong thế giới kết nối (API, IoT, Hardware, Software).

#### 2.4.2.4. Ưu điểm của mô hình hệ thống đa tác tử

- Tính mô-đun (Modularity): Hệ thống dễ mở rộng, bảo trì, và nâng cấp từng phần.
- Tính song song (Parallelism): Các tác tử có thể hoạt động đồng thời, giúp tăng tốc độ xử lý.
- Tính linh hoạt (Flexibility): Có thể điều chỉnh hành vi từng tác tử để thích ứng với yêu cầu mới.
- Khả năng phối hợp (Coordination): Cho phép chia sẻ tri thức và hợp tác giữa các mô hình AI khác nhau (ví dụ: giữa mô hình LLM và mô hình đánh giá điểm số).
- Là công nghệ cốt lõi cho việc thực hiện các nhiệm vụ phân tán (Core technology enabling distributed tasks).
- Chuyên môn hóa giúp đơn giản hóa các nhiệm vụ phức tạp (Specialization simplifies complex tasks).

#### 2.4.2.5. Xu hướng tương lai

Khả năng tương tác: Các tiêu chuẩn mới nổi như MCP và A2A. Tạo điều kiện thuận lợi cho việc giao tiếp giữa các tác nhân và cách sử dụng công cụ.

Dân chủ hóa: Tăng khả năng tiếp cận thông qua các nền tảng không cần mã và ít mã. Giúp phát triển dễ dàng hơn.

**Áp dụng doanh nghiệp:** Tập trung vào tính tuân thủ, bảo mật và tích hợp. Quan trọng đối với môi trường doanh nghiệp.

**Multimodal Agents:** Tích hợp giọng nói. Khả năng hình ảnh và văn bản. Nâng cao tương tác giữa các tác nhân.

#### 2.4.2.6. Thách thức

Rủi ro bị khóa chặt bởi nhà cung cấp nếu gắn chặt với hệ sinh thái Google Cloud. Đường cong học tập: Việc điều phối nâng cao và luồng đa tác nhân đòi hỏi phải hiểu các khái niệm trừu tượng ADK.

**Chi phí:** Việc sử dụng Gemini LLM, lưu trữ và dịch vụ Vertex AI trong môi trường sản xuất sẽ phát sinh chi phí đắt đỏ.

**Độ phức tạp của việc gỡ lỗi:** Với nhiều tác nhân và các công cụ không đồng bộ, việc theo dõi hành vi cần được thực hiện cẩn thận, may mắn thay, giao diện người dùng (UI) đã hỗ trợ việc này.

#### 2.4.2.7. Ví dụ điển hình

Bảng 2.4-2. *Bảng demo hệ thống đa tác tử*

From	Message
User	Hãy lên kế hoạch du lịch giúp tôi.
Travel Agent	Chắc chắn rồi! Để bắt đầu, bạn có thể cho tôi biết bạn muốn đi đâu và thời gian du lịch của bạn là bao lâu?
User	Tôi muốn đi Đà Nẵng vào cuối tuần này, từ thứ Sáu đến Chủ Nhật.
Travel Agent	Tuyệt vời! Dưới đây là một kế hoạch du lịch cơ bản: - Ngày 1: Đến Đà Nẵng, tham quan Bà Nà Hills - Ngày 2 Khám phá phố cổ Hội An, tham quan chùa Cầu - Ngày 3: Tham quan Ngũ Hành Sơn, và nghỉ ngơi trước khi về. Bạn có muốn chỉnh sửa gì không?
User	Tôi muốn thêm một buổi ăn tối ở nhà hàng ven biển vào ngày thứ 2, và chuyến chuyến bay về sớm hơn một chút.
Travel Agent	Cảm ơn bạn đã cung cấp thông tin! Tôi đã chỉnh sửa lịch trình như sau: - Ngày 1: Đến Đà Nẵng, tham quan Bà Nà Hills - Ngày 2 Khám phá Hội An, ăn tối tại Nhà hàng biển Làng Chài - Ngày 3: Tham quan Ngũ Hành Sơn và bay về vào Th chiều. Bạn đã hài lòng với kế hoạch chưa?

From	Message
User	Rất ổn rồi, nhưng tôi muốn đặt phòng khách sạn 3 sao và cần chuyến bay sớm nhất từ Hà Nội vào Đà Nẵng.
Travel Agent	Tuyệt vời Tôi sẽ yêu cầu Agent Khách Sạn và Agent Máy Bay giúp bạn. Chờ một chút nhé.
Agent Khách Sạn	Để đáp ứng yêu cầu của bạn, tôi đã tìm được các khách sạn 3 sao ở Đà Nẵng gần biển với giá phải chăng - Khách sạn Sunflower 1 đêm: 500.000 VNĐ - Khách sạn Green Peace – 1 đêm: 450.000 VNĐ Bạn muốn tôi book phòng cho bạn không?
User	Chọn Khách sạn Sunflower nhé.
Agent Máy Bay	Tôi đã kiểm tra các chuyến bay từ Hà Nội đến Đà Nẵng vào thứ Sáu. Dưới đây là các lựa chọn: - Vietnam Airlines: Bay lúc 6h sáng, giá 1,200.000 VNĐ - VietJet Bay lúc 7h sáng, giá 1.000.000 VNĐ Bạn muốn chọn chuyến bay nào?
User	Chọn Vietnam Airlines lúc 6h sáng.
Travel Agent	Đã xong! Tôi đã đặt phòng tại Khách sạn Sunflower và chuyến bay với Vietnam Airlines lúc 6h sáng. Bạn có muốn tôi hoàn tất việc đặt vé tham quan hay chuẩn bị thêm dịch vụ gì nữa không?

## 2.5. Các tác tử trong hệ thống Summarization Studio

Trong đề tài này, hệ thống đa tác tử được sử dụng để phân tách quy trình tóm tắt thành các giai đoạn nhỏ, mỗi tác tử đảm nhiệm một nhiệm vụ cụ thể. Ví dụ:

- Agent 1 – Coordinator: Điều phối pipeline, phân công nhiệm vụ cho các agent tương ứng và phản hồi người dùng.
- Agent 2 – Reader/OCR: Đọc và trích xuất nội dung văn bản gốc từ văn bản, ảnh hay file PDF.
- Agent 3 – Spellchecker: Kiểm tra và sửa lỗi chính tả.
- Agent 4 – Extractor: Thực hiện tóm tắt trích xuất sao cho phù hợp với từng khối lớp (1-5).
- Agent 5 – Abstracter: Tạo bản tóm tắt diễn giải tự nhiên bằng mô hình LLM sao cho phù hợp với từng cấp lớp (1-5).

- Agent 6 – Grade Calibrator: Điều chỉnh độ dài, từ vựng sao cho phù hợp với cấp lớp (1-5).
- Agent 7 – Evaluator: Đánh giá chất lượng bản tóm tắt theo tiêu chí độ chính xác, mạch lạc và mức độ phù hợp với học sinh tiểu học, chấm điểm dựa trên độ dễ hiểu theo thang điểm (0-10).
- Agent 8 – Aggregator: Tổng hợp kết quả cuối cùng để xuất bản tóm tắt hoàn chỉnh.

Cách tiếp cận này giúp hệ thống:

Giảm tải cho từng mô hình riêng lẻ nhờ chia nhỏ nhiệm vụ. Tăng độ chính xác và khả năng kiểm soát quy trình, vì từng tác tử có thể được huấn luyện hoặc tinh chỉnh độc lập. Dễ dàng mở rộng hoặc thay thế từng tác tử mà không ảnh hưởng đến toàn bộ hệ thống.

## 2.6. Công nghệ và nền tảng phát triển

Hệ thống được phát triển dựa trên hai nền tảng chính: Python cho mô hình đa tác tử (Multi-Agent System) và Java Spring Boot cho phần backend phục vụ giao tiếp giữa hệ thống AI và giao diện người dùng. Và React cho Web và Mobile (đang phát triển).

### 2.6.1. Ngôn ngữ và môi trường lập trình

- Python:

Là ngôn ngữ lập trình mạnh mẽ, phổ biến trong lĩnh vực Trí tuệ nhân tạo (AI) và Xử lý ngôn ngữ tự nhiên (NLP). Cung cấp nhiều thư viện hỗ trợ học máy, xử lý dữ liệu, và triển khai mô hình như: transformers, langchain, ollama, chroma, pandas, numpy, v.v. Được sử dụng để xây dựng các tác tử (agents), điều phối hoạt động của từng tác tử và xử lý dữ liệu đầu vào – đầu ra.

- Java Spring Boot:

Là framework mạnh mẽ dùng để xây dựng API backend theo mô hình RESTful. Hỗ trợ khả năng mở rộng, quản lý request và bảo mật dữ liệu giữa hệ thống AI và ứng dụng người dùng. Được sử dụng để kết nối giữa hệ thống đa tác tử (Python) với giao diện web/app, đảm bảo luồng trao đổi dữ liệu mượt mà và ổn định.

### 2.6.2. Các công nghệ và thư viện chính

#### 2.6.2.1. Ollama

Nền tảng giúp chạy và quản lý các mô hình ngôn ngữ lớn (LLMs) ngay trên máy cục bộ. Cho phép sử dụng các mô hình như qwen2.5, phi3, llama3, hoặc mô hình tùy chỉnh đã fine-tuned.

Ưu điểm:

- Giảm phụ thuộc vào API bên ngoài (như OpenAI hoặc Gemini).

- Tăng tốc độ phản hồi và đảm bảo tính riêng tư dữ liệu.
- Trong hệ thống, Ollama đóng vai trò mô hình sinh tóm tắt và đánh giá nội dung cho các tác tử như Abstracter và Evaluator.

Sau khi tiến hành khảo sát, thử nghiệm và đánh giá nhiều mô hình ngôn ngữ lớn (LLMs) khác nhau như GPT-4, Gemini 2.5, Mistral 7B, Qwen 2.5, và ViT5, tôi nhận thấy Ollama kết hợp với mô hình LLaMA 3 (8B) là giải pháp tối ưu nhất cho việc xây dựng hệ thống đa tác tử (Multi-Agent System – MAS) phục vụ tóm tắt truyện và bài đọc dành cho học sinh tiểu học. Dưới đây là các lý do và tiêu chí lựa chọn cụ thể:

- Hiệu năng xử lý cao – tài nguyên thấp

LLaMA 3 8B là phiên bản tối ưu hóa của Meta, có hiệu năng ngôn ngữ gần tương đương các mô hình 30–40B tham chiếu:

- Dễ dàng chạy cục bộ trên máy tính cá nhân hoặc server nhỏ.
- Giảm chi phí vận hành so với các API thương mại như GPT-4 hoặc Gemini.
- Đảm bảo tốc độ phản hồi nhanh, phù hợp với hệ thống nhiều tác tử hoạt động đồng thời.

- Tích hợp hoàn hảo với nền tảng Ollama

Ollama cho phép triển khai và quản lý các mô hình LLM cục bộ, hỗ trợ tải, khởi chạy và gọi mô hình thông qua REST API hoặc Python SDK.

Ollama có thể dễ dàng tích hợp trực tiếp với LangChain và LangGraph, giúp mỗi tác tử trong hệ thống (như Extractor, Abstracter, Evaluator) có thể gọi model riêng biệt nhưng vẫn chia sẻ ngữ cảnh chung. Việc này đảm bảo MAS hoạt động ổn định, tách biệt từng agent nhưng vẫn kết nối linh hoạt.

- Khả năng tương thích tốt với LangChain & LangGraph

LLaMA 3 hoạt động mượt với LangChain và LangGraph, hai framework cốt lõi trong kiến trúc hệ thống. Cho phép biểu diễn các nút xử lý độc lập trong LangGraph (mỗi nút là một agent được hỗ trợ bởi Ollama-LLaMA3). Mỗi agent có thể truyền thông điệp (message passing) và chia sẻ trạng thái (state) dễ dàng, tạo nên một pipeline xử lý logic, rõ ràng và có thể mở rộng.

- Khả năng ngôn ngữ tự nhiên tốt và có thể tùy chỉnh

LLaMA 3 đã được huấn luyện trên tập dữ liệu đa ngôn ngữ lớn, hỗ trợ tiếng Việt tương đối ổn định. Mô hình có thể fine-tune hoặc sử dụng prompt engineering để tối ưu cho mục tiêu tóm tắt nội dung giáo dục, giúp tạo ra bản tóm tắt:

Ngắn gọn, chính xác và thân thiện với lứa tuổi học sinh. Có thể điều chỉnh theo từng cấp lớp (1–5) thông qua GradeCalibratorAgent.

- Linh hoạt trong mở rộng và thay thế

Ollama hỗ trợ đa mô hình (multi-model management), vì vậy trong tương lai có thể dễ dàng. Thay thế hoặc cập nhật lên các phiên bản LLaMA 3.1, Mistral 8x7B, hoặc Qwen 2.5. Thêm các mô hình chuyên biệt cho từng tác tử (ví dụ: một mô hình chuyên OCR, một mô hình chuyên đánh giá ngôn ngữ).

### 2.6.2.2. LangChain

Framework chuyên dùng để xây dựng ứng dụng sử dụng LLMs bằng cách kết hợp các thành phần như chuỗi truy vấn (Chains), bộ nhớ (Memory), và công cụ (Tools).

Trong hệ thống, LangChain giúp:

- Tổ chức logic hoạt động của các tác tử.
- Xử lý luồng hội thoại giữa người dùng và các mô hình.
- Tích hợp Ollama và các API khác trong cùng một pipeline xử lý.

### 2.6.2.3. LangGraph

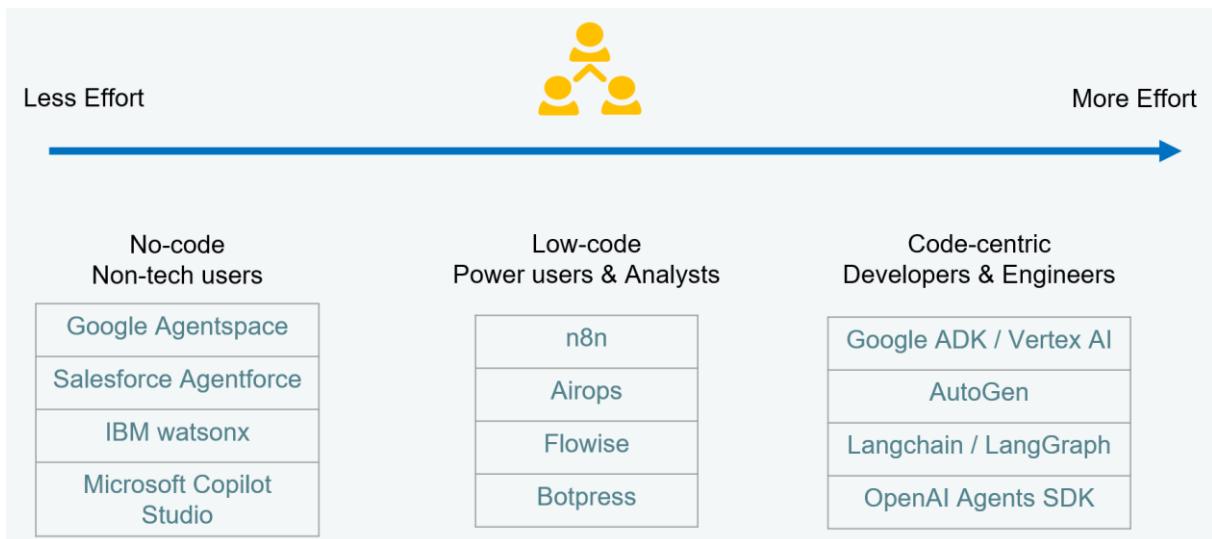
LangGraph là một mở rộng của LangChain, cung cấp kiến trúc dạng đồ thị (graph-based) cho phép quản lý luồng tương tác phức tạp giữa nhiều tác tử (multi-agent orchestration).

Khác với việc tạo các “chain” tuyến tính như trong LangChain, LangGraph cho phép định nghĩa các nút (nodes) đại diện cho từng tác tử (agent) hoặc hành động (tool), và các cạnh (edges) biểu thị luồng thông tin hoặc điều kiện chuyển tiếp giữa chúng.

Nhờ đó, LangGraph giúp mô hình hóa quá trình xử lý ngôn ngữ tự nhiên (NLP) hoặc hệ thống hội thoại (conversational AI) theo hướng có kiểm soát, dễ theo dõi, mở rộng và tái sử dụng.

Framework	Độ dẽ	Tính linh hoạt	Khả năng mở rộng	khả năng tương tác	sẵn sàng sản xuất
ADK	✓✓	✓✓✓	✓✓✓	✓✓✓	✓✓✓
LangGraph	✓	✓✓✓	✓✓✓	✓✓	✓✓
AutoGen	✓✓✓	✓✓	✓✓	✓✓	✓✓
CrewAI	✓✓✓	✓	✓✓	✓	✓
n8n	✓✓✓	✓✓	✓	✓✓✓	✓✓
AgentFlow	✓✓✓	✓✓	✓✓✓	✓✓✓	✓✓✓

Hình 2.6.1. Những đặc điểm giữa các framework lớn



Hình 2.6.2. Đối tượng thích hợp trong việc sử dụng và phát triển dự án

Trong dự án này việc lựa chọn LangGraph & LangChain nhằm tạo cấu trúc theo dạng đồ thị (Graph Structure), quản lý trạng thái hội thoại (State Management), điều phối luồng xử lý (Flow Control), tích hợp linh hoạt với các model LLMs và hỗ trợ tương tác đa tác tử. Phù hợp với cấp độ yêu cầu và xu hướng phát triển trong tương lai.

Dự án này, mỗi node trong đồ thị tương ứng với một agent, ví dụ:

Coordinator Agent → Reader Agent → Spellchecker Agent → Extractor Agent → Abstracter Agent → Grade Calibrator → Evaluator Agent → Aggregator Agent.

LangGraph giúp đảm bảo:

- Các tác tử hoạt động độc lập nhưng phối hợp nhịp nhàng.
- Dễ dàng giám sát, mở rộng và gỡ lỗi hệ thống.

#### 2.6.2.4. Chroma

Là cơ sở dữ liệu vector (vector database) dùng để lưu trữ và truy vấn dữ liệu dạng embedding (biểu diễn ngữ nghĩa của văn bản).

Trong hệ thống:

- Dùng để lưu trữ embedding của nội dung truyện và bài đọc, giúp tìm kiếm và đối chiếu nhanh giữa các câu hoặc đoạn tương đồng.
- Hỗ trợ cho các tác tử Extractor và Evaluator khi cần truy xuất ngữ cảnh hoặc đánh giá tính nhất quán của tóm tắt.

#### 2.6.2.5. Jupyter Notebook

Là môi trường lập trình tương tác được sử dụng trong giai đoạn nghiên cứu, thử nghiệm và huấn luyện mô hình.

Cho phép:

- Hiển thị kết quả tức thời sau mỗi lần chạy mã.

- Kết hợp giữa code, biểu đồ và ghi chú giải thích, thuận tiện cho quá trình nghiên cứu và trình bày kết quả.

#### 2.6.2.6. Anaconda

Là bộ công cụ hỗ trợ quản lý môi trường Python và cài đặt thư viện khoa học dữ liệu.

Được sử dụng để:

- Tạo môi trường riêng biệt cho dự án hệ thống đa tác tử (tránh xung đột phiên bản thư viện).
- Quản lý dễ dàng các gói như langchain, chromadb, torch, transformers, v.v.

#### 2.6.3. Môi trường triển khai

Máy cục bộ (localhost): Toàn bộ hệ thống đa tác tử chạy trên máy tính cá nhân thông qua Ollama và Python environment kernel.

Server backend (Spring Boot): Được triển khai trên cổng riêng (ví dụ: localhost:8080), đóng vai trò kết nối và quản lý API.

Giao diện người dùng: Có thể được xây dựng dưới dạng ứng dụng web hoặc tích hợp với dashboard để hiển thị kết quả tóm tắt, đánh giá và bài học rút ra (đang phát triển).

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG TÓM TẮT TRUYỆN, BÀI ĐỌC

## 3.1. Phân tích yêu cầu hệ thống

### 3.1.1. Mục tiêu hệ thống

#### 3.1.1.1. Mục tiêu tổng quát

Xây dựng một hệ thống tóm tắt truyện và bài đọc tự động dành cho học sinh tiểu học, hoạt động dựa trên mô hình đa tác tử (Multi-Agent System) nhằm:

- Tự động xử lý văn bản đầu vào.
- Sinh ra bản tóm tắt ngắn gọn, dễ hiểu, phù hợp với lứa tuổi học sinh.
- Và đưa ra bài học rút ra hoặc ý nghĩa giáo dục của nội dung.

Hệ thống hướng đến việc hỗ trợ giáo viên, phụ huynh và học sinh trong quá trình học tập, đọc hiểu và phát triển tư duy ngôn ngữ.

#### 3.1.1.2. Mục tiêu cụ thể

##### 1. Tự động hóa quá trình tóm tắt

- Cho phép hệ thống nhận đầu vào là truyện hoặc bài đọc tiếng Việt, xử lý và xuất ra bản tóm tắt ngắn gọn, đầy đủ ý chính.
- Kết hợp hai phương pháp tóm tắt:
  - Tóm tắt trích xuất (Extractive Summary)
  - Tóm tắt diễn giải (Abstractive Summary)
- Giúp học sinh dễ hiểu, dễ ghi nhớ nội dung chính.

##### 2. Tích hợp kiến trúc đa tác tử (Multi-Agent System)

- Chia quá trình tóm tắt thành nhiều giai đoạn, mỗi tác tử (agent) phụ trách một nhiệm vụ riêng biệt như:
  - Phản hồi yêu cầu của người dùng, điều phối pipeline, phân công nhiệm vụ (Coordinator Agent)
  - Nhận dạng nội dung (OCR Reader Agent)
  - Chuẩn hóa và sửa lỗi chính tả (Spellchecker Agent)
  - Tạo bản tóm tắt trích xuất (Extractor Agent)
  - Tạo bản tóm tắt diễn giải (Abstracter Agent)
  - Kiểm soát độ dài và từ vựng cho từng cấp lớp (Grade Calibrator Agent)
  - Đánh giá và tổng hợp kết quả (Evaluator & Aggregator Agent).
- Tăng độ chính xác, khả năng kiểm soát, và dễ mở rộng hệ thống.

##### 3. Tạo ra bài học hoặc ý nghĩa rút ra từ truyện/bài đọc

Hỗ trợ thêm chức năng sinh “bài học rút ra” từ nội dung truyện, giúp học sinh hiểu được thông điệp đạo đức hoặc giá trị nhân văn.

#### 4. Xây dựng backend và API hỗ trợ triển khai hệ thống

Sử dụng Java Spring Boot để quản lý dữ liệu, kết nối frontend và xử lý yêu cầu từ người dùng. Cung cấp giao diện API RESTful để tương tác với mô hình AI và hệ thống đa tác tử.

#### 5. Đảm bảo khả năng mở rộng và thân thiện người dùng

Hệ thống có thể tích hợp thêm mô hình ngôn ngữ khác (ví dụ: Gemini, OpenAI, Claude, v.v.). Giao diện và API dễ sử dụng, hỗ trợ tích cực cho học sinh, giáo viên và phụ huynh.

##### **3.1.1.3. Kết quả mong đợi**

Tạo ra một hệ thống tự động, thông minh và linh hoạt trong việc tóm tắt nội dung tiếng Việt. Cung cấp bản tóm tắt có tính mạch lạc, ngắn gọn và phù hợp độ tuổi học sinh tiểu học. Có thể mở rộng thành nền tảng hỗ trợ học tập thông minh trong tương lai.

##### **3.1.2. Phân tích yêu cầu chức năng**

###### **3.1.2.1. Mục tiêu**

Xác định các chức năng chính và phụ mà hệ thống cần cung cấp để đạt được mục tiêu tóm tắt tự động, đồng thời đảm bảo tính linh hoạt, mở rộng và thân thiện người dùng. Các chức năng được phân tích dựa trên hai nhóm chính:

- Chức năng hệ thống đa tác tử (Multi-Agent System).
- Chức năng người dùng và backend quản lý.

###### **3.1.2.2. Các chức năng chính của hệ thống**

###### 1. Chức năng nhận và phản hồi theo yêu cầu người dùng. Bên cạnh đó, chức năng điều phối pipeline và phân công nhiệm vụ cho từng Agent thích hợp (Coordinator Agent).

- Đầu tiên hệ thống gửi lời chào và chờ đợi yêu cầu tiếp theo của người dùng.
- Sau khi xác định được mục đích của user, hệ thống chấp nhận và phản hồi các yêu cầu user cần cung cấp để hệ thống hoạt động được trôi chảy và trả về kết quả tốt nhất.
- Khởi đầu: Lời chào thân thiện, cởi mở. Và xem xét nhu cầu của user là gì ?

Đầu vào: Yêu cầu của user

Đầu ra: Phản hồi, điều phối và phân công.

###### 2. Chức năng nhập và tiền xử lý dữ liệu (Reader Agent & Spellchecker Agent)

- Nhận đầu vào là văn bản truyện hoặc bài đọc tiếng Việt, có thể được nhập thủ công hoặc tải lên từ file .txt, .pdf, .csv.
- Kiểm tra và phát hiện lỗi chính tả, lỗi ký tự hoặc dấu câu, sau đó tự động chuẩn hóa nội dung.

- Lưu trữ dữ liệu gốc và dữ liệu đã xử lý trong hệ thống (cơ sở dữ liệu Chroma hoặc tệp cục bộ).

Đầu vào: Văn bản gốc.

Đầu ra: Văn bản đã được chuẩn hóa, sạch dữ liệu.

### 3. Chức năng tóm tắt trích xuất (Extractor Agent)

- Phân tích văn bản đầu vào để xác định các câu hoặc đoạn quan trọng nhất.
- Áp dụng kỹ thuật tính điểm TF-IDF hoặc embedding similarity để lựa chọn câu tiêu biểu.
- Trả về bản tóm tắt trích xuất (extractive summary) chứa nội dung cốt lõi.

Đầu vào: Văn bản đã được làm sạch.

Đầu ra: Bản tóm tắt trích xuất.

### 4. Chức năng tóm tắt diễn giải (Abstracter Agent)

- Sử dụng mô hình ngôn ngữ lớn (LLM) chạy qua Ollama + LangChain, tạo ra bản tóm tắt tự nhiên, mạch lạc và dễ hiểu.
- Diễn đạt lại nội dung theo cách ngắn gọn và phù hợp với học sinh tiểu học, tránh dùng từ phức tạp.
- Tham khảo kết quả từ Extractor Agent để đảm bảo độ chính xác nội dung.

Đầu vào: Văn bản hoặc bản tóm tắt trích xuất.

Đầu ra: Bản tóm tắt diễn giải (abstractive summary).

### 5. Chức năng kiểm soát độ dài và từ vựng (Grade Calibrator Agent)

- Thực hiện phân tích, xử lý độ dài và từ vựng từ văn bản được Agent trước đó tóm tắt sao cho phù hợp với từng cấp lớp, với từng phương thức tóm tắt.
- Kết quả đưa ra sau bước này sẽ có chất lượng khá chuẩn và phù hợp.

Đầu vào: Một trong hai bản tóm tắt diễn giải hoặc trích xuất tùy vào từng Agent được hệ thống phân công trước đó.

Đầu ra: Bản tóm tắt đã qua điều chỉnh và sửa chữa.

### 6. Chức năng đánh giá chất lượng tóm tắt (Evaluator Agent)

Tự động đánh giá chất lượng bản tóm tắt dựa trên các tiêu chí:

- Độ chính xác về nội dung.
- Mức độ dễ hiểu (độ phù hợp với cấp tiểu học).
- Tính mạch lạc và ngắn gọn.
- Sử dụng mô hình LLM hoặc các tiêu chuẩn định lượng như ROUGE/BERTScore để so sánh giữa bản tóm tắt và nội dung gốc.

Đầu vào: Tóm tắt đầu ra từ Abstracter Agent.

Đầu ra: Báo cáo đánh giá (score và feedback).

## 7. Chức năng tổng hợp kết quả (Aggregator Agent)

Nhận dữ liệu đầu ra từ các tác tử trước (extractive + abstractive + evaluation).

Tổng hợp lại thành bản tóm tắt cuối cùng hoàn chỉnh, có thể kèm theo:

- Bản tóm tắt trích xuất.
- Bản tóm tắt diễn giải.
- Nhận xét đánh giá.
- Bài học rút ra (lesson summary).
- Xuất kết quả ra file .txt, .json, hoặc hiển thị trên giao diện web.

Đầu vào: Kết quả từ các Agent khác.

Đầu ra: Bản tóm tắt hoàn chỉnh và bài học rút ra.

### 3.1.2.3. Các chức năng hỗ trợ và quản lý

#### 1. Chức năng backend (Java Spring Boot)

Xây dựng các RESTful API cho phép frontend hoặc người dùng gửi yêu cầu:

Bảng 3.1-1. Bộ API của cuộc trò chuyện với hệ thống

Method	Address	Chú thích
GET	/message	Lấy tất cả đoạn hội thoại.
GET	/message/{ messageid }	Lấy 1 đoạn hội thoại bằng messageid
DEL	/message/{ messageid }	Xóa 1 đoạn hội thoại bằng messageid

Bảng 3.1-2. Bộ API của đọc lịch sử tóm tắt

Method	Address	Chú thích
GET	/api/read-history/user/{ userid }	Lịch sử của userid
POST	/api/read-history/log?userId={...}&summaryId={...}	Tạo log lịch sử

Bảng 3.1-3. Bộ API hệ thống tóm tắt

Method	Address	Chú thích
POST	/api/summaries	Tạo tóm tắt
GET	/api/summaries/admin	Lấy tóm tắt admin
PUT	/api/summaries/{ sumid }status	Cập nhật status

<b>Method</b>	<b>Address</b>	<b>Chú thích</b>
GET	/api/summaries/status/{ status}	Lấy tóm tắt với status
GET	/api/summaries/contributor/{userid}	Lấy tóm tắt với contributor
GET	/api/summaries/grade/{ grade}	Lấy tóm tắt với grade
GET	/api/summaries/method/{ method}	Lấy tóm tắt với method
GET	/api/summaries	Lấy tất cả bản tóm tắt
GET	/api/summaries/{sumid}?userId={userid}	Lấy bản tóm tắt với id
GET	/api/summaries/top10	Lấy top 10 bản tóm tắt
GET	/api/summaries/search?searchTerm={ ...}&grade={ grade}	Lấy bản tóm tắt với title hoặc grade
PATCH	/api/summaries/{sumid}	Cập nhật bản tóm tắt với id
DEL	/api/summaries/{sumid}	Xóa bản tóm tắt với id
DEL	/api/summaries/bulk	Xóa nhiều bản tóm tắt

Bảng 3.1-4. Bộ API lịch sử tóm tắt

<b>Method</b>	<b>Address</b>	<b>Chú thích</b>
POST	/api/summary-histories/start-session	Bắt đầu session
GET	/api/summary-histories/{historyId}	Lấy lịch sử tóm tắt với id
PUT	/api/summary-histories/{historyId}	Cập nhật lịch sử với id
DEL	/api/summary-histories/{historyId}	Xóa lịch sử tóm tắt
GET	/api/summary-histories/user/{userId}	Lấy lịch sử với userid

<b>Method</b>	<b>Address</b>	<b>Chú thích</b>
POST	/api/summary-histories/create-summary	Tạo lịch sử tóm tắt
GET	/api/summary-histories/session/{sessionId}	Lấy lịch sử tóm tắt với session

Bảng 3.1-5. Bộ API phiên tóm tắt

<b>Method</b>	<b>Address</b>	<b>Chú thích</b>
POST	/api/summary-sessions	Tạo phiên tóm tắt
GET	/api/summary-sessions/{id}	Lấy phiên tóm tắt với id
POST	/api/summary-sessions/process-pdf	Xử lý PDF
PUT	/api/summary-sessions/{id}	Cập nhật phiên với id
POST	/api/summary-sessions/generate-image	Tạo hình ảnh
POST	/api/summary-sessions/upload-image	Tải ảnh lên cloudinary
DEL	/api/summary-sessions/{id}	Xóa phiên tóm tắt với id

Bảng 3.1-6. Bộ API thẻ tóm tắt

<b>Method</b>	<b>Address</b>	<b>Chú thích</b>
GET	/api/summary-tags/summary/{tagid}	Lấy tất cả tag bởi id tóm tắt
GET	/api/summary-tags/tag/{sumid}	Lấy tất cả tóm tắt bởi tag id
POST	/api/summary-tags	Tạo tag tóm tắt
DEL	/api/summary-tags/{id}	Xóa tag tóm tắt bởi id tag

Bảng 3.1-7. Bộ API thẻ của hệ thống

<b>Method</b>	<b>Address</b>	<b>Chú thích</b>
GET	/api/tags	Lấy tag
GET	/api/tags/{id}	Lấy tag với id
GET	/api/tags/name/{name}	Lấy tag với name
GET	/api/tags/name/{name}/all	Lấy tất cả tag với name
POST	/api/tags	Tạo tag mới
DEL	/api/tags/{id}	Xóa tag bởi id

Bảng 3.1-8. Bộ API quản lý thông tin người dùng

Method	Address	Chú thích
GET	/api/users	Lấy user
GET	/api/users/{id}	Lấy user với id
POST	/api/users	Tạo user mới
PUT	/api/users/{id}/profile	Cập nhật profile
PUT	/api/users/{id}	Cập nhật user
POST	/api/users/auth/login	Xác thực login
DEL	/api/users/{id}	Xóa user với id
GET	/api/users/role/{role}	Lấy user bởi role
PUT	/api/users/{id}/avatar	Cập nhật user avatar

## 2. Chức năng lưu trữ và truy xuất

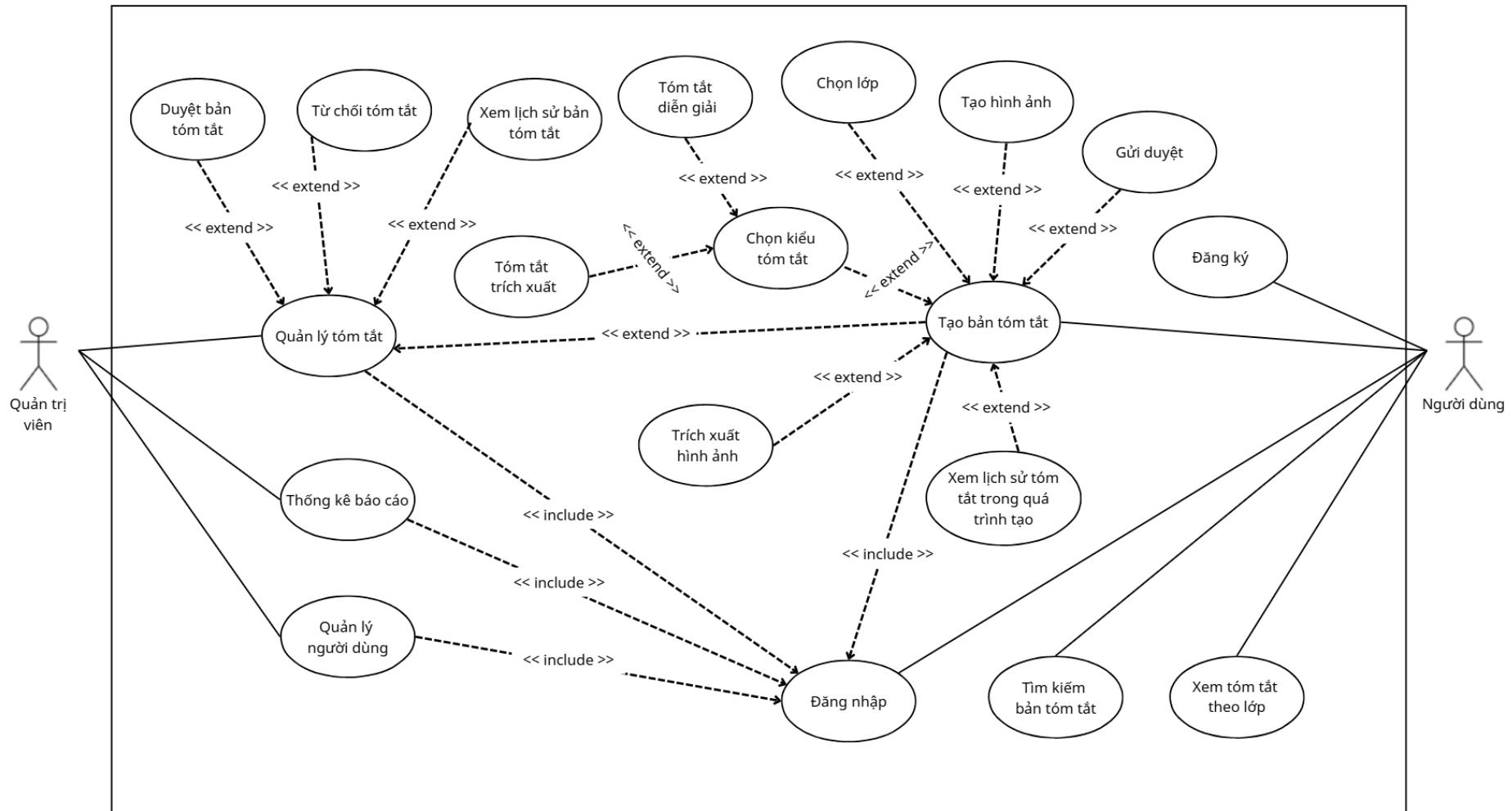
- Chroma Vector DB: Lưu trữ embedding của văn bản gốc và bản tóm tắt để phục vụ cho việc tìm kiếm và đối chiếu. Cho phép truy xuất lịch sử trò truyện của hệ thống đa tác tử.
- MySQL DB: Lưu trữ dữ liệu của hệ thống Spring Boot, Web và Mobile mang lại khả năng truy xuất nhanh chóng, đơn giản và dễ phát triển, mở rộng.

## 3. Chức năng giao tiếp giữa các tác tử (LangGraph & LangChain)

- Người dùng tải lên một đoạn văn bản.
- Reader Agent đọc nội dung → gửi cho Spellchecker Agent.
- Spellchecker Agent xử lý và gửi kết quả đến Extractor Agent.
- Extractor Agent trích xuất thông tin quan trọng → chuyển tiếp đến Abstracter Agent.
- Abstracter Agent tạo bản tóm tắt → gửi đến Grade Calibrator Agent để điều chỉnh độ dài và độ dễ hiểu.
- Grade Calibrator Agent sau điều chỉnh → Evaluation Agent để thực hiện đánh giá bản tóm tắt.
- Evaluation Agent thực hiện nhiệm vụ sau đó → chuyển tiếp cho Aggregator Agent
- Aggregator Agent tổng hợp kết quả cuối cùng và trả về cho người dùng.

### 3.1.3. Sơ đồ use case

#### 3.1.3.1. Sơ đồ use case hệ thống tóm tắt văn bản



Hình 3.1.1. Sơ đồ use case hệ thống tóm tắt văn bản

### 3.1.4. Đặc tả use case

#### 3.1.4.1. Đặc tả use case tổng quan hệ thống

Use Case ID	Tên Use Case	Mô tả	Tác nhân chính	Điều kiện tiên quyết	Kết quả thành công	Luồng sự kiện chính	Use Case liên quan
UC02	<b>Đăng nhập</b>	Xác thực người dùng hoặc quản trị viên để truy cập hệ thống.	Người dùng / Quản trị viên	Tài khoản tồn tại và hợp lệ.	Hệ thống cho phép đăng nhập và chuyển đến giao diện chính.	<ol style="list-style-type: none"> <li>Người dùng nhập email, mật khẩu.</li> <li>Hệ thống xác thực.</li> <li>Nếu hợp lệ → truy cập hệ thống.</li> </ol>	UC01 – Đăng ký, UC03 – Tạo bản tóm tắt, UC09 – Quản lý tóm tắt
UC03	<b>Tạo bản tóm tắt</b>	Hệ thống cho phép người dùng nhập truyện hoặc bài đọc để tạo bản tóm tắt tự động.	Người dùng	Đã đăng nhập hệ thống.	Bản tóm tắt được tạo và lưu trong cơ sở dữ liệu.	<ol style="list-style-type: none"> <li>Người dùng chọn “Tạo bản tóm tắt”.</li> <li>Nhập nội dung cần tóm tắt.</li> <li>Hệ thống MAS (đa tác giả) thực hiện OCR, kiểm lỗi, trích xuất hoặc diễn giải.</li> <li>Trả về bản tóm tắt hoàn chỉnh.</li> </ol>	UC04 – Xem lịch sử tóm tắt, UC05 – Tạo hình ảnh, UC06 – Gửi duyệt
UC09	<b>Quản lý tóm tắt</b>	Quản trị viên xem, duyệt, từ chối, hoặc theo dõi các bản tóm tắt do người dùng gửi lên.	Quản trị viên	Có bản tóm tắt chờ duyệt.	Bản tóm tắt được duyệt hoặc bị từ chối.	<ol style="list-style-type: none"> <li>Mở danh sách bản tóm tắt chờ duyệt.</li> <li>Chọn bản cụ thể.</li> <li>Thực hiện hành động (đ duyệt hoặc từ chối).</li> </ol>	UC10 – Duyệt bản tóm tắt, UC11 – Từ chối bản tóm tắt, UC12 – Xem lịch sử bản tóm tắt

UC13	<b>Tóm tắt trích xuất</b>	Hệ thống dùng PhoBERT để trích xuất câu quan trọng làm tóm tắt.	Người dùng	Người dùng đã chọn "Tạo bản tóm tắt" và loại tóm tắt trích xuất.	Hệ thống tạo và hiển thị tóm tắt trích xuất	<ol style="list-style-type: none"> <li>1. Hệ thống nhận văn bản.</li> <li>2. Hệ thống dùng PhoBERT trích xuất câu quan trọng.</li> <li>3. Hệ thống tạo tóm tắt.</li> <li>4. Hệ thống hiển thị tóm tắt.</li> </ol>	Tạo bản tóm tắt (UC-03)
UC14	<b>Tóm tắt diễn giải</b>	Hệ thống dùng mT5 để diễn giải lại nội dung thành tóm tắt.	Người dùng	Người dùng đã chọn "Tạo bản tóm tắt" và loại tóm tắt diễn giải.	Hệ thống tạo và hiển thị tóm tắt diễn giải.	<ol style="list-style-type: none"> <li>1. Hệ thống nhận văn bản.</li> <li>2. Hệ thống dùng mT5 diễn giải nội dung.</li> <li>3. Hệ thống tạo tóm tắt.</li> <li>4. Hệ thống hiển thị tóm tắt.</li> </ol>	Tạo bản tóm tắt (UC-03)
UC15	<b>Chọn lớp</b>	Người dùng chọn cấp lớp để tóm tắt phù hợp với học sinh.	Người dùng	Người dùng đang trong quá trình tạo tóm tắt.	Hệ thống ghi nhận cấp lớp và điều chỉnh tóm tắt phù hợp.	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị danh sách lớp (lớp 1-5).</li> <li>2. Người dùng chọn lớp.</li> <li>3. Hệ thống lưu lựa chọn.</li> </ol>	Tạo bản tóm tắt (UC-03)
UC16	<b>Tìm kiếm bản tóm tắt</b>	Người dùng tìm kiếm bản tóm tắt đã tạo.	Người dùng	Người dùng đã đăng nhập và có bản tóm tắt đã tạo.	Hệ thống hiển thị danh sách bản tóm tắt phù hợp với từ khóa.	<ol style="list-style-type: none"> <li>1. Người dùng vào mục tìm kiếm.</li> <li>2. Người dùng nhập từ khóa.</li> <li>3. Hệ thống tìm và hiển thị kết quả.</li> </ol>	Xem tóm tắt theo lớp (UC-08)
UC17	<b>Trích xuất ảnh, văn bản</b>	Người dùng thực hiện thả ảnh, PDF và văn bản	Người dùng	Người dùng đã có văn bản trích xuất từ các file phương tiện	Hệ thống tiếp nhận và hiển thị ra màn hình văn bản trích xuất	<ol style="list-style-type: none"> <li>1. Người dùng vào mục thả ảnh, PDF, văn bản.</li> <li>2. Hệ thống tiếp nhận và bắt đầu trích xuất, sau đó hiển thị lên màn hình.</li> </ol>	Tạo bản tóm tắt (UC-03)

Bảng 3.1-9. Đặc tả use case chính của hệ thống tóm tắt

## 3.2. Thu thập và xử lý dữ liệu (đang phát triển)

### 3.2.1. Nguồn dữ liệu

Để xây dựng bộ dữ liệu phục vụ huấn luyện các mô hình tóm tắt văn bản (PhoBERT và mT5), hiện tại em đang tiến hành thu thập dữ liệu từ các nguồn chính thức và đáng tin cậy, với trọng tâm là các văn bản phù hợp cho học sinh tiểu học. Bộ dữ liệu được lựa chọn kỹ lưỡng nhằm đảm bảo tính đa dạng về thể loại, độ dài, và mức độ phù hợp với khả năng nhận thức của học sinh từ lớp 1 đến lớp 5.

Hiện tại bộ dữ liệu đã gồm hơn 6000 bài đọc. Tuy nhiên, bộ dữ liệu vẫn đang còn quá nhỏ chưa thể thực hiện huấn luyện để cho thể tạo ra model đặc thù để thay thế cho model Ollama hay API của GPT, Gemini. Em đang tích cực thu thập thêm nhiều nguồn dữ liệu khác, mở rộng ra nhiều lĩnh vực và nhiều môn học nhưng vẫn đảm bảo nằm trong phạm trù cấp lớp đã quy định. Các lĩnh vực cần mở rộng thu thập như:

- Các môn học khác tiếng Việt (đã thu thập trước đó) đối với từng cấp lớp.
- Thu thập thêm dữ liệu về các tập truyện như: cổ tích, truyền thuyết, ngụ ngôn, bài thơ hay ca dao, tục ngữ.
- Mở rộng hơn về lĩnh vực âm nhạc rộng lớn.

### 3.2.2. Quy trình thu thập và xử lý dữ liệu

Dữ liệu được thu thập từ các bộ sách giáo khoa với nhiều môn học khác nhau trong từng bộ sách. Nhằm cải thiện bộ dữ liệu thuận tiện được bộ giáo dục thiết kế đặc thù cho từng cấp lớp. Ngoài ra, còn có các mảng tiềm năng cần khai thác đã liệt kê trước đó.

Dữ liệu thu thập sẽ là các định dạng khác nhau, chúng sẽ làm tổn rất nhiều thời gian xử lý. OCR hay những công cụ AI sẽ giúp cải thiện phần nào đó quá trình thu thập dữ liệu và xử lý chúng. Dữ liệu thô được trả về sẽ ở định dạng text. Sau đó qua quá trình làm sạch các ký tự, ký hiệu hay dấu câu, chính tả sẽ được hệ thống máy thực hiện. Tuy nhiên để trích xuất và làm sạch có hiệu suất cao, ta cần những biện pháp để ràng buộc các quá trình này. Bộ dữ liệu từ điển tiếng Việt và kiểm định thuật toán làm sạch, sẽ được áp dụng và cài đặt.

Qua nhiều bước xử lý dữ liệu, dữ liệu sẽ trở nên đáng tin cậy hơn. Tuy nhiên ta vẫn chưa thể áp dụng chúng. Để giải quyết chúng, ta cần phải đổi chiều và sửa chữa thủ công bản trích xuất trước đó để nâng cao độ tin cậy. Sau đó dữ liệu sẽ được phân bổ và quản lý bởi Pandas để chuẩn bị dữ liệu cho quá trình huấn luyện tiếp theo.

### 3.2.3. Định hướng huấn luyện dữ liệu

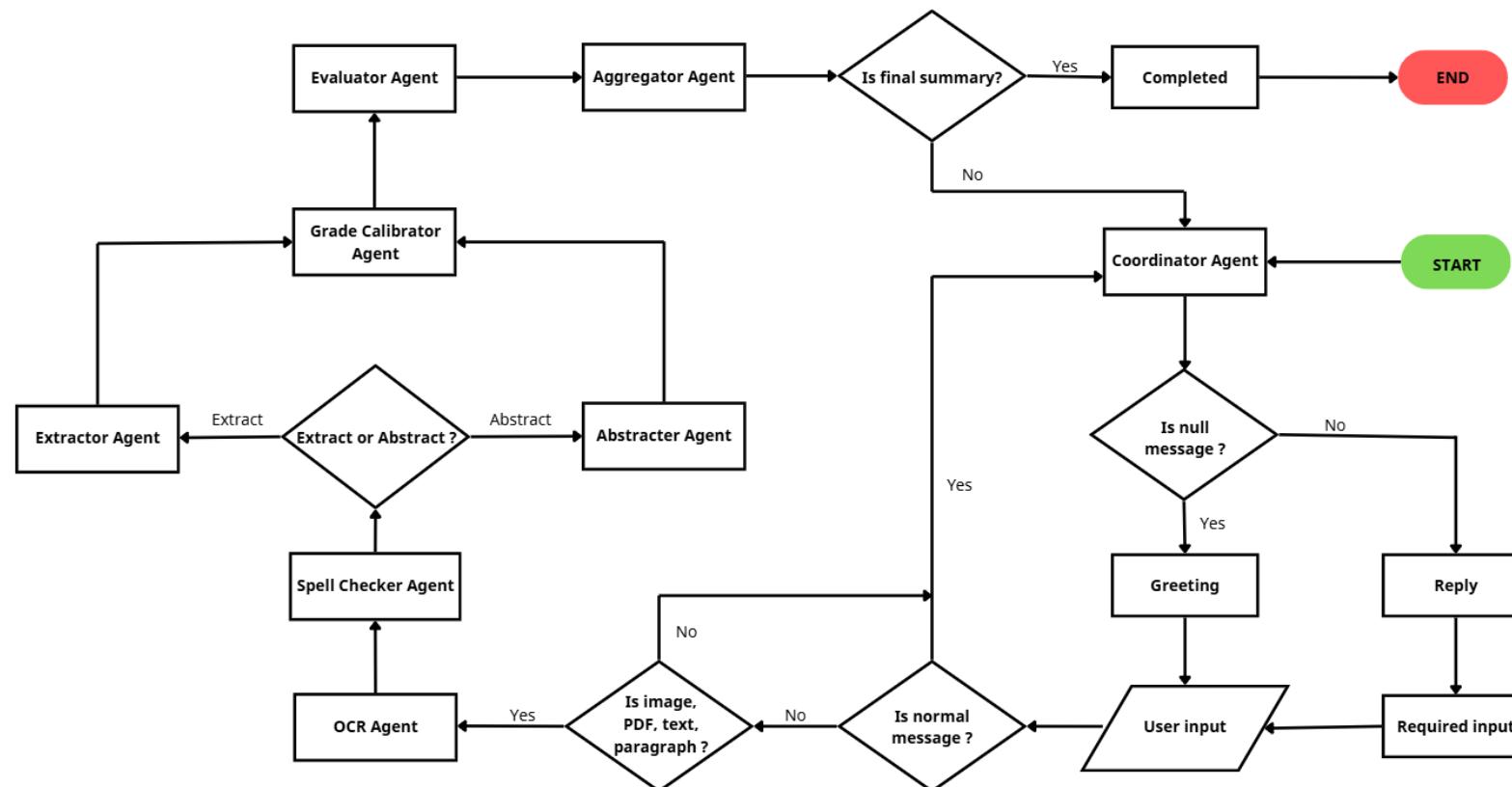
Model cần thiết để tích hợp vào hệ thống là hai model: Trích xuất (Extract) và Diễn giải (Abstract). Để xây dựng thành công hai model này, chúng ta cần sử dụng bộ

dữ liệu chuyên biệt và có chọn lọc để xây dựng hai model nhỏ có hiệu suất và kết quả tốt, độ tin cậy cao. Từ hai model nhỏ đó, ta thực hiện tạo bản tóm tắt cho toàn bộ dữ liệu còn lại. Sau khi thu được 2 bộ dữ liệu chuyên biệt cho hai phương thức diễn giải và trích xuất. Ta thực hiện huấn luyện hai model, sau đó tinh chỉnh fine tune hai model nhằm mục đích chúng sẽ mang lại kết quả tóm tắt tốt và hiệu quả cao.

## 4.1. Kiến trúc hệ thống

### 4.1.1. Thiết kế hệ thống đa tác tử

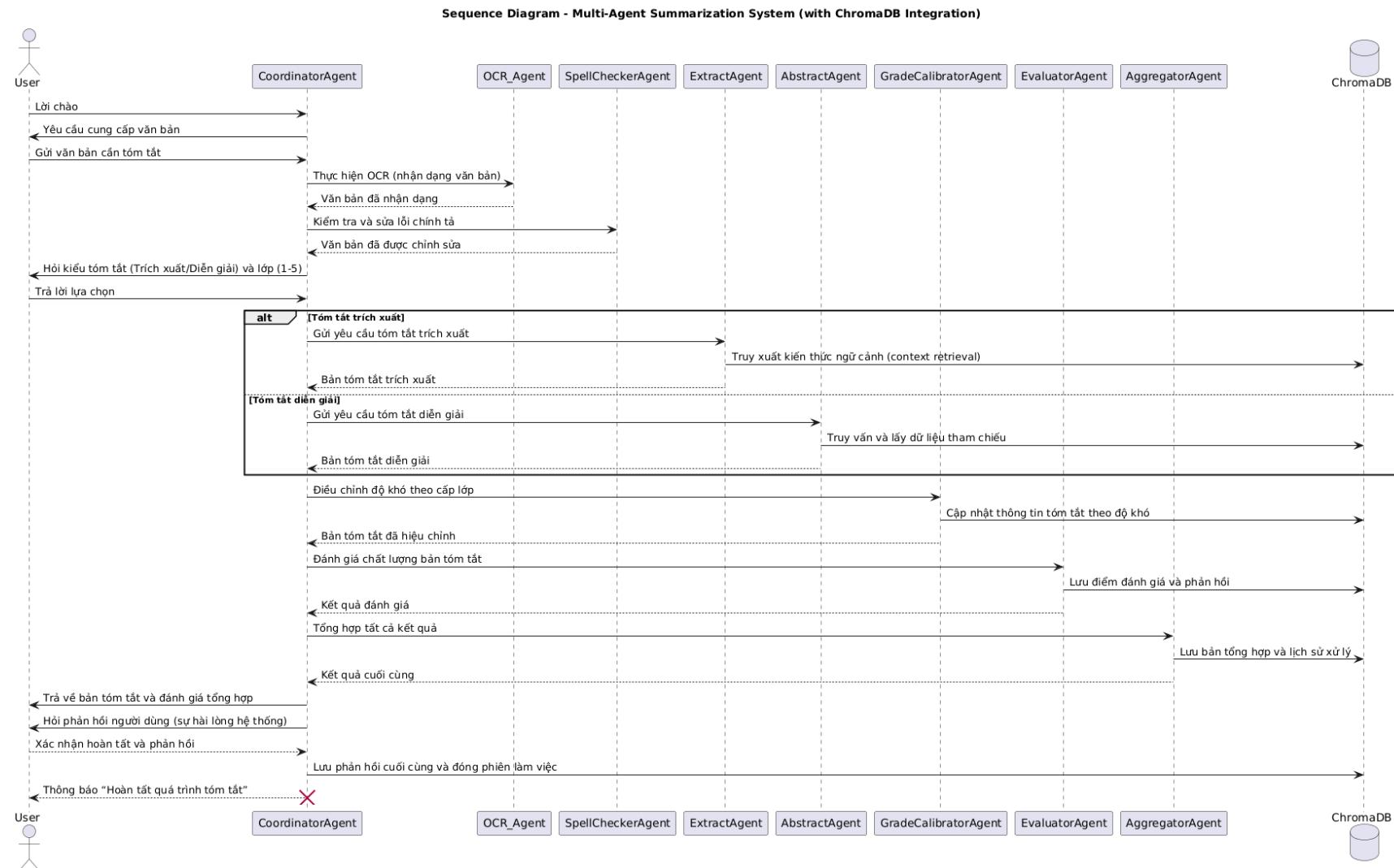
#### 4.1.1.1. Sơ đồ luồng hoạt động



Hình 4.1.1. Sơ đồ flowchart hệ thống đa tác tử

Sơ đồ trên thể hiện luồng xử lý dữ liệu đầu vào và mối liên hệ mật thiết giữa các Agents để tạo nên hệ thống tóm tắt hiệu quả và chặt chẽ.

#### 4.1.1.2. Sơ đồ Sequence Diagram



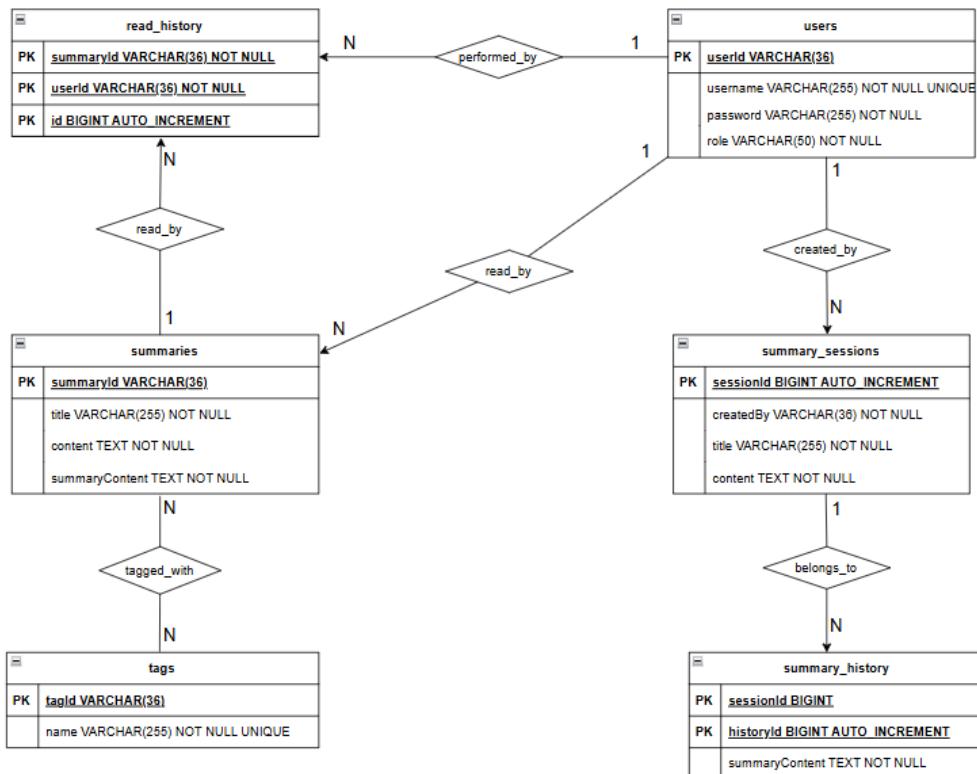
Hình 4.1.2. Sơ đồ sequence diagram hệ thống đa tác tử

Sơ đồ tuần tự (Sequence Diagram) mô tả trình tự trao đổi thông điệp giữa các thành phần (actor và agent) trong hệ thống theo trục thời gian. Nó cho phép hình dung toàn bộ quy trình xử lý tóm tắt văn bản tự động, từ khi người dùng nhập dữ liệu cho đến khi nhận kết quả. Sơ đồ này cho thấy hệ thống đa tác tử hoạt động theo mô hình pipeline (xử lý tuần tự nhiều giai đoạn) nhưng vẫn linh hoạt:

- Có khả năng lựa chọn nhánh tóm tắt (Extract / Abstract)
- Có đánh giá chất lượng và điều chỉnh cấp độ tự động
- Có tương tác hai chiều với người dùng (lấy đầu vào và phản hồi kết quả)

#### 4.1.2. Thiết kế cơ sở dữ liệu

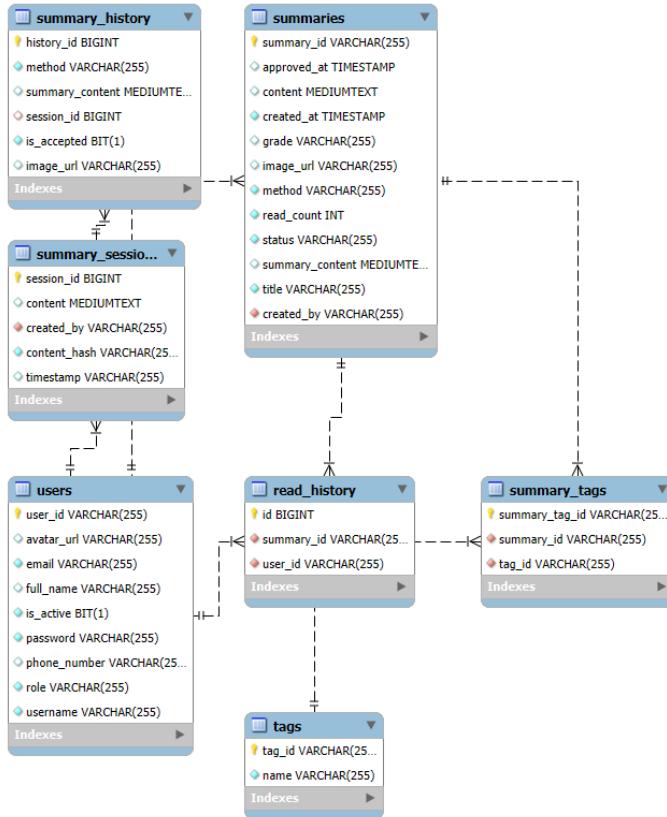
##### 4.1.2.1. Sơ đồ Entity Relationship Diagram (ERD)



Hình 4.1.3. Sơ đồ ERD database

Sơ đồ ER này mô tả cấu trúc cơ sở dữ liệu của hệ thống tóm tắt văn bản, bao gồm các thực thể chính và mối quan hệ giữa chúng, nhằm hỗ trợ lưu trữ và quản lý dữ liệu hiệu quả.

#### 4.1.2.2. Sơ đồ Schema diagram



Hình 4.1.4. Sơ đồ schema diagram database

Sơ đồ trên thể hiện cấu trúc cơ sở dữ liệu của hệ thống tóm tắt văn bản, bao gồm các bảng chính và mối quan hệ giữa chúng, hỗ trợ các chức năng như tạo, quản lý tóm tắt, và tìm kiếm. Dưới đây là mô tả chi tiết của các bảng:

Bảng 4.1-1. Bảng User

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	user_id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi người dùng.
2	username	VARCHAR(255)	Tên đăng nhập của người dùng, duy nhất.
3	password	VARCHAR(255)	Mật khẩu của người dùng, được mã hóa.
4	role	VARCHAR(255)	Vai trò của người dùng (người dùng, quản trị viên).
5	email	VARCHAR(255)	Địa chỉ email của người dùng.
6	full_name	VARCHAR(255)	Tên đầy đủ của người dùng.
7	phone_number	VARCHAR(255)	Số điện thoại của người dùng.

STT	Tên cột	Kiểu dữ liệu	Mô tả
8	avatar_url	VARCHAR(255)	Đường dẫn đến ảnh đại diện của người dùng.
9	is_active	VARCHAR(255)	Trạng thái hoạt động của người dùng (1: hoạt động, 0: không hoạt động).

Bảng 4.1-2. Bảng phiên tóm tắt

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	session_id	BIGINT	Khóa chính, định danh duy nhất cho mỗi phiên tóm tắt.
2	created_by	VARCHAR(255)	Khóa ngoại liên kết đến bảng users, xác định người tạo phiên.
3	title	VARCHAR(255)	Tiêu đề của phiên tóm tắt.
4	content	MEDIUMTEXT	Nội dung gốc của phiên tóm tắt.
5	content_hash	VARCHAR(255)	Mã băm của nội dung để kiểm tra tính duy nhất.
6	timestamp	VARCHAR(255)	Thời gian tạo phiên tóm tắt.

Bảng 4.1-3. Bảng tóm tắt

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	summary_id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi bản tóm tắt.
2	title	VARCHAR(255)	Tiêu đề của bản tóm tắt.
3	content	MEDIUMTEXT	Nội dung gốc của bản tóm tắt.
4	summary_content	MEDIUMTEXT	Nội dung đã được tóm tắt.
5	image_url	VARCHAR(255)	Đường dẫn đến hình ảnh minh họa (nếu có).
6	created_by	VARCHAR(255)	Khóa ngoại liên kết đến bảng users, xác định người tạo.
7	status	VARCHAR(255)	Trạng thái của bản tóm tắt (ví dụ: đã duyệt, chưa duyệt).
8	grade	VARCHAR(255)	Cấp lớp hướng đến (lớp 1-5).

STT	Tên cột	Kiểu dữ liệu	Mô tả
9	method	VARCHAR(255)	Phương pháp tóm tắt (trích xuất hoặc diễn giải).
10	read_count	INT	Số lần bản tóm tắt được đọc.
11	created_at	TIMESTAMP	Thời gian tạo bản tóm tắt.
12	approved_at	TIMESTAMP	Thời gian bản tóm tắt được duyệt.

Bảng 4.1-4. Bảng lịch sử tóm tắt

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	history_id	BIGINT	Khóa chính, định danh duy nhất cho mỗi lịch sử tóm tắt.
2	session_id	BIGINT	Khóa ngoại liên kết đến bảng summary_sessions, xác định phiên tóm tắt.
3	summary_id	VARCHAR(255)	Khóa ngoại liên kết đến bảng summaries, xác định bản tóm tắt.
4	method	VARCHAR(255)	Phương pháp tóm tắt (trích xuất hoặc diễn giải).
5	status	VARCHAR(255)	Trạng thái của lịch sử tóm tắt.
6	summary_content	MEDIUMTEXT	Nội dung đã được tóm tắt.
7	image_url	VARCHAR(255)	Đường dẫn đến hình ảnh minh họa (nếu có).
8	is_accepted	BIT(1)	Trạng thái chấp nhận của lịch sử tóm tắt (1: chấp nhận, 0: không chấp nhận).

Bảng 4.1-5. Bảng đọc lịch sử tóm tắt

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	BIGINT	Khóa chính, định danh duy nhất cho mỗi lịch sử đọc.
2	user_id	VARCHAR(255)	Khóa ngoại liên kết đến bảng users, xác định người dùng.
3	summary_id	VARCHAR(255)	Khóa ngoại liên kết đến bảng summaries, xác định bản tóm tắt được đọc.

Bảng 4.1-6. *Bảng gắn nhãn tóm tắt*

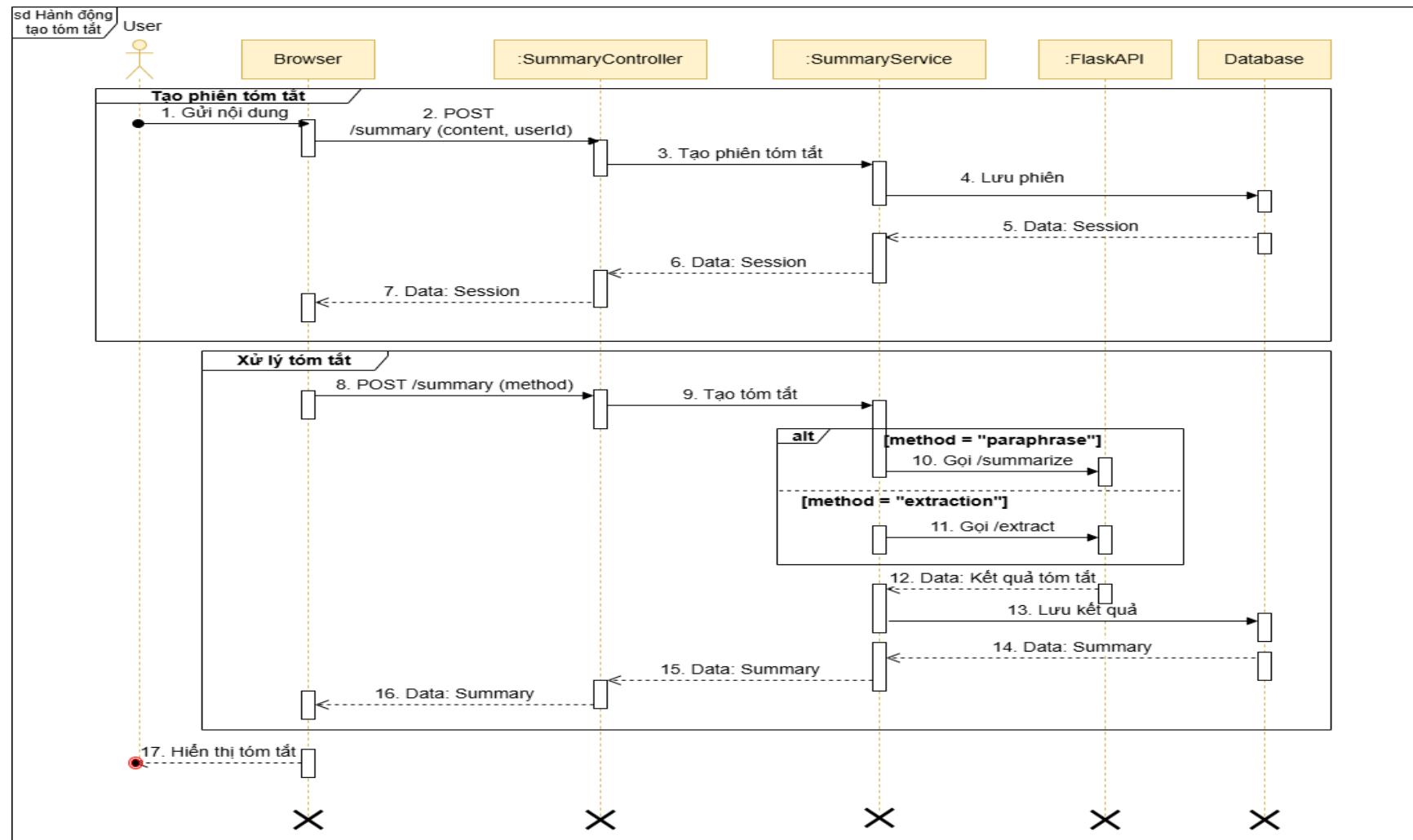
STT	Tên cột	Kiểu dữ liệu	Mô tả
1	tag_id	VARCHAR(255)	Khóa chính, định danh duy nhất cho mỗi thẻ phân loại.
2	name	VARCHAR(255)	Tên của thẻ phân loại.

Bảng 4.1-7. *Bảng quản lý id nhãn và tóm tắt*

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	summary_id	VARCHAR(255)	Khóa ngoại liên kết đến bảng summaries, xác định bản tóm tắt.
2	tag_id	VARCHAR(255)	Khóa ngoại liên kết đến bảng tags, xác định thẻ phân loại.

## 4.1.3. Thiết kế hệ thống API và tích hợp

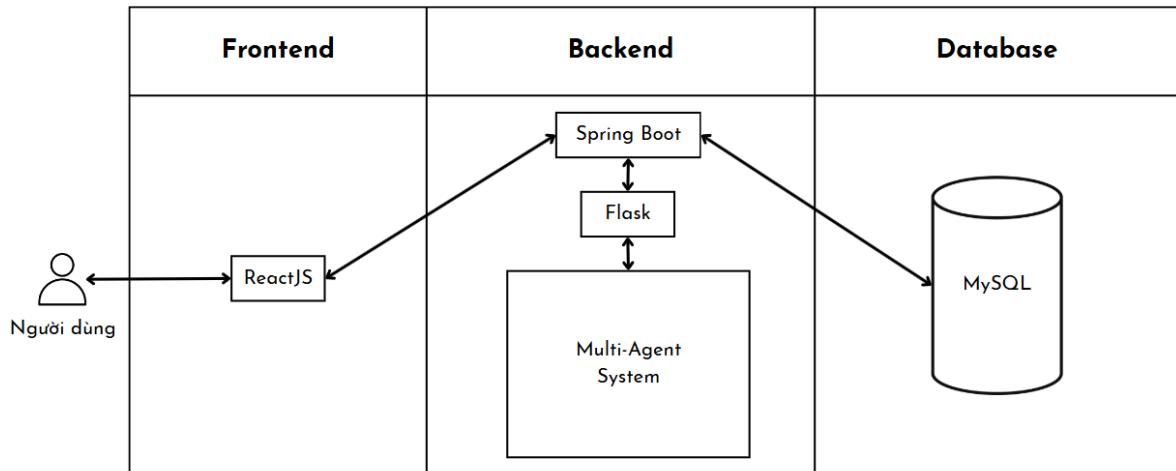
### 4.1.3.1. Sequence Diagram



Hình 4.1.5. Sơ đồ sequence diagram API

Hình trên minh họa quy trình tạo tóm tắt văn bản. Người dùng gửi nội dung qua giao diện ReactJS, khởi tạo yêu cầu POST /summary đến SummaryController trong Java Spring Boot. Controller gọi SummaryService, tích hợp mô hình PhoBERT (tóm tắt trích xuất) hoặc mT5 (tóm tắt diễn giải). Kết quả tóm tắt được lưu vào cơ sở dữ liệu MySQL và trả về trình duyệt để hiển thị, đảm bảo nội dung chính xác, phù hợp với học sinh tiểu học.

#### 4.1.3.2. Kiến trúc ứng dụng tóm tắt



Hình 4.1.6. Kiến trúc ứng dụng tóm tắt

Hình thể hiện kiến trúc hệ thống tóm tắt văn bản được thiết kế cho học sinh tiểu học. Người dùng (học sinh, giáo viên, phụ huynh) tương tác qua giao diện ReactJS để nhập văn bản, chọn phương pháp tóm tắt (trích xuất hoặc diễn giải), và nhận kết quả. Yêu cầu được chuyển đến backend Spring Boot thông qua API (ví dụ: /api/summaries), nơi tích hợp mô hình hệ thống đa tác tử. Kết quả được lưu trữ trong cơ sở dữ liệu MySQL, hỗ trợ quản lý lịch sử tóm tắt, tìm kiếm theo người dùng hoặc phân loại, đảm bảo tính chính xác và phù hợp với mục tiêu giáo dục.

## CHƯƠNG 4: XÂY DỰNG VÀ TRIỂN KHAI HỆ THỐNG TÓM TẮT

### 4.2. Môi trường và công cụ phát triển

#### 4.2.1. Môi trường phát triển

- Hệ điều hành: Windows 11 Pro 64-bit
- Ngôn ngữ lập trình:
  - Python 3.9.21: dùng để xây dựng hệ thống đa tác tử (Multi-Agent System), gọi và phối hợp các mô hình ngôn ngữ lớn (LLMs).
  - Java (JDK 24): triển khai backend và kết nối hệ thống với cơ sở dữ liệu.

- Nền tảng quản lý môi trường: Anaconda – quản lý môi trường ảo và các thư viện Python.
- IDE / Trình soạn thảo: Visual Studio Code – môi trường phát triển chính cho cả Python, Java và React Native.
- Môi trường thực nghiệm AI: Jupyter Notebook – dùng để kiểm thử, huấn luyện dữ liệu.

#### 4.2.2. Môi trường phát triển web và di động

- Framework giao diện: ReactJS (ứng dụng web).
- React Native: dùng cho phát triển ứng dụng di động đa nền tảng (Android, iOS).
- Trình duyệt kiểm thử: Google Chrome, Edge.
- Công cụ giả lập: Android Studio Emulator hoặc Expo Go.

#### 4.2.3. Công cụ và thư viện hỗ trợ

- Quản lý mã nguồn: Git & GitHub.
- Quản lý thư viện:
  - pip, conda – cho Python.
  - npm – cho ReactJS/React Native.
- Công cụ thử nghiệm API: Postman, Hoppscotch.

#### 4.2.4. Thư viện và mô hình sử dụng

##### 4.2.4.1. Thư viện chính

Thư viện	Vai trò
Ollama	Nền tảng chạy mô hình ngôn ngữ lớn (LLMs) cục bộ, giảm phụ thuộc vào API bên ngoài.
LangChain	Tổ chức và kết nối các mô hình, agent, và pipeline xử lý ngôn ngữ tự nhiên.
LangGraph	Xây dựng và điều phối các tác tử (agent) trong hệ thống Multi-Agent, hỗ trợ luồng hội thoại phức tạp.
Chroma	Vector Database dùng để lưu trữ và truy vấn embedding của văn bản, hỗ trợ truy xuất ngữ nghĩa.
Pandas, NumPy	Tiền xử lý dữ liệu và thao tác dữ liệu dạng bảng.

Bảng 4.2-1. Bảng thư viện sử dụng

##### 4.2.4.2. Mô hình sử dụng

Mô hình	Vai trò
Llama 3: 8B (Ollama)	Mô hình ngôn ngữ lớn dùng để tóm tắt và diễn giải văn bản.

Custom Agents (CoordinatorAgent, OCRReaderAgent, SpellCheckerAgent, ExtractAgent, AbstractAgent, ...)	Các tác tử đảm nhiệm từng bước trong quy trình tóm tắt và đánh giá.
---	---

Bảng 4.2-2. Bảng mô hình sử dụng

#### 4.2.5. Cấu hình phần cứng

- CPU: Intel Core i5-11400H
- RAM: 24 GB
- GPU: NVIDIA GTX 1650 (4 GB VRAM).

### 4.3. Xây dựng pipeline đa tác tử (Multi-Agent Implementation)

#### 4.3.1. Giới thiệu chung

Hệ thống tóm tắt được xây dựng theo kiến trúc đa tác tử (Multi-Agent System – MAS), trong đó mỗi agent đảm nhận một nhiệm vụ chuyên biệt trong quá trình xử lý ngôn ngữ tự nhiên.

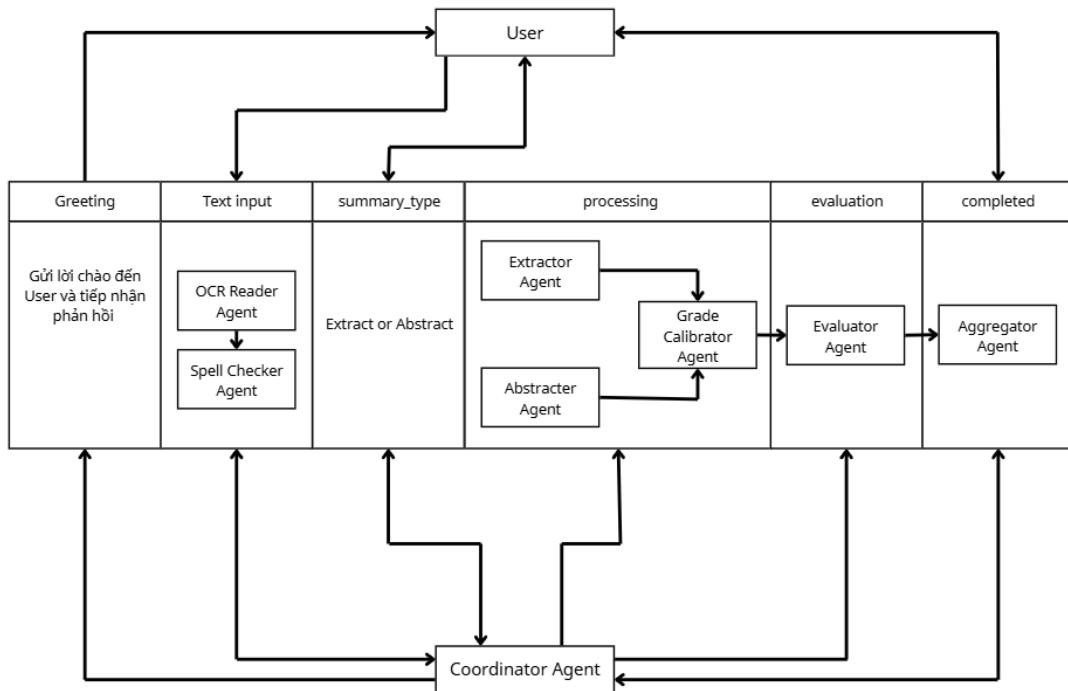
Pipeline của hệ thống được điều phối bởi một Coordinator Agent, chịu trách nhiệm giao tiếp với người dùng, phân công công việc, nhận kết quả từ các agent con, và tổng hợp đầu ra cuối cùng.

Mục tiêu của pipeline là:

- Tự động hóa quy trình tóm tắt văn bản (theo 2 hướng: trích xuất và diễn giải).
- Giảm tải cho người dùng trong việc nhập liệu, kiểm tra, và đánh giá kết quả.
- Cho phép mở rộng linh hoạt, thêm hoặc thay thế agent mà không ảnh hưởng đến toàn bộ hệ thống.

#### 4.3.2. Kiến trúc tổng thể

Pipeline được xây dựng dựa trên LangGraph – một framework mở rộng của LangChain, giúp thiết kế luồng xử lý hội thoại (conversation flow) theo hướng có trạng thái (stateful). Các agent được tổ chức theo chuỗi xử lý – đánh giá – phản hồi, trong đó mỗi bước có thể hoạt động độc lập nhưng vẫn được điều phối thống nhất.



Hình 4.3.1. Kiến trúc tổng thể hệ thống đa tác tử

### 4.3.3. Mô tả chi tiết các Agent

Agent	Chức năng chính
CoordinatorAgent	Là trung tâm điều phối. Gửi lời chào, yêu cầu văn bản từ người dùng, xác định loại tóm tắt (trích xuất/diễn giải) và cấp độ lớp học (1–5). Sau đó, phân công nhiệm vụ cho các agent con và tổng hợp phản hồi.
OCRReaderAgent	Thực hiện nhận dạng văn bản từ hình ảnh (nếu người dùng tải lên ảnh thay vì nhập chữ). Hay nhận đầu vào là văn bản.
SpellCheckerAgent	Kiểm tra và sửa lỗi chính tả trong văn bản đầu vào bằng các công cụ như hoặc mô hình ngôn ngữ nội bộ qua Ollama.
ExtractAgent	Sinh bản tóm tắt trích xuất – chọn các câu quan trọng từ nội dung gốc mà không thay đổi nội dung.
AbstractAgent	Sinh bản tóm tắt diễn giải – viết lại nội dung theo cách ngắn gọn, dễ hiểu hơn, phù hợp với học sinh tiểu học.
GradeCalibratorAgent	Điều chỉnh độ dài và từ vựng tóm tắt sao cho phù hợp với trình độ lớp (1–5), đảm bảo dễ đọc và phù hợp độ tuổi.
EvaluatorAgent	Đánh giá chất lượng bản tóm tắt dựa trên độ bao phủ, ngữ nghĩa, và tính mạch lạc. Có thể dùng mô hình LLM (như Llama 3:8B) để chấm điểm.

AggregatorAgent	Tổng hợp kết quả từ các agent, bao gồm bắn tóm tắt, điểm đánh giá, và phản hồi. Trả kết quả cuối cùng cho người dùng.
-----------------	---

Bảng 4.3-1. Bảng mô tả chi tiết các Agent có trong hệ thống

#### 4.3.4. Cơ chế phối hợp

Pipeline sử dụng LangGraph State Machine để duy trì trạng thái hội thoại:

- conversation\_stage được cập nhật sau mỗi tác vụ.
- needs\_user\_input xác định khi nào cần người dùng cung cấp thêm thông tin.
- messages lưu lại toàn bộ lịch sử hội thoại để agent sau có thể tham chiếu ngữ cảnh.

Mỗi agent hoạt động theo cơ chế đầu vào – đầu ra chuẩn hóa, đảm bảo tính mở rộng và tái sử dụng.

#### 4.3.5. Kết quả và lợi ích

Hệ thống có khả năng tự động xử lý toàn bộ pipeline từ nhập liệu → kiểm lỗi → tóm tắt → đánh giá → phản hồi. Mô hình đa tác tử (MAS) giúp dễ dàng mở rộng thêm chức năng mới (ví dụ: dịch văn bản, phân tích cảm xúc...). Việc sử dụng LangGraph giúp đảm bảo điều phối mượt mà, tránh lỗi vòng lặp và tăng khả năng kiểm soát luồng hội thoại.

### 4.4. Triển khai hệ thống đa tác tử

#### 4.4.1. Giới thiệu

Sau khi thiết kế pipeline đa tác tử (Multi-Agent Pipeline), hệ thống được triển khai dựa trên LangChain, LangGraph, và Ollama, kết hợp các mô hình LLM cục bộ (như Llama 3: 8B) để đảm bảo khả năng xử lý ngôn ngữ tự nhiên mà không phụ thuộc vào dịch vụ đám mây.

Hệ thống được chia thành hai lớp chính:

- Lớp xử lý thông minh (Intelligence Layer): bao gồm các agent và mô hình LLM.
- Lớp điều phối và tích hợp (Coordination Layer): quản lý luồng tương tác giữa các agent và người dùng.

#### 4.4.2. Các bước triển khai chính

##### 4.4.2.1. Cài đặt môi trường hệ thống

- Sử dụng Anaconda để tạo kernel Python riêng
  - Khởi tạo kernel bằng Command Prompt

```
conda create -n mykernel python=3.9.21
```

- Trở đến kernel vừa tạo

```
activate mykernel
```

- Sau đó cài các thư viện sau

```
pip install python-dotenv langchain langchain-ollama langchain-community  
langchain-core langgraph requests chromadb sentence-transformers
```

- Tiếp tục cài đặt Ollama và thực hiện chọn model ngôn ngữ
  - Cài đặt với đường link sau: <https://ollama.com/download/>
  - Một số lưu ý khi chọn model LLM.
    - Tùy thuộc vào laptop hay PC mạnh hay yếu để có thể chọn được model phù hợp.
    - Model nhỏ (**0 – 3GB**) → Kết quả thường lệch lạc, không có chiều sâu, câu trả lời đơn giản và rời rạc.
    - Model vừa (**3-15GB**) → Kết quả có thể chấp nhận được, tuy nhiên cũng có nhiều hạn chế. Thời gian xử lý ổn phù hợp.
    - Model lớn (**15 – 45GB**) → Kết quả tốt, tuy nhiên hạn chế với máy yếu, đặc biệt là laptop (VRAM Card yêu cầu lớn >=20GB. Chip phải nhiều core và thread, RAM >= 64GB).
    - Model rất lớn (**>45GB**) → Mạnh như Chat GPT, DeepSeek, Gemini...
  - Vào đường dẫn sau để tìm model: <https://ollama.com/search>

#### 4.4.2.2. Phân công nhiệm vụ cụ thể và xây dựng các Agent

##### 4.4.2.2.1. Coordinator Agent

Khởi tạo Prompt để thực hiện phản hồi người dùng, điều phối pipeline và phân công nhiệm vụ cho các Agent tương ứng.

```
COORDINATOR_SYSTEM = """Bạn là Coordinator Agent thông minh giúp học sinh  
tiểu học tóm tắt văn bản theo 2 cách (TRÍCH XUẤT và DIỄN GIẢI) phù hợp với  
khối lớp (1-5)."""
```

Workflow của bạn:

1. GREETING: Chào hỏi và yêu cầu user cung cấp văn bản
2. TEXT\_INPUT: Nhận văn bản từ user và chuyển cho OCR/SpellChecker để xử lý
3. SUMMARY\_TYPE: Hỏi user muốn tóm tắt TRÍCH XUẤT hay DIỄN GIẢI và khối lớp nào (1-5)
4. PROCESSING: Phân công cho agent phù hợp (Extractor hoặc Abstracter)
5. COMPLETED: Tổng hợp kết quả và hỏi đánh giá hệ thống

```
Luôn trả lời ngắn gọn và đi thẳng vào vấn đề."""
```

Các chức năng quan trọng khác:

- Xử lý trường hợp messages rỗng – GREETING
- Xử lý theo từng giai đoạn
- Lưu văn bản gốc và chuyển sang xử lý OCR/SpellChecker
- Phân tích yêu cầu về loại tóm tắt và khối lớp
- Tìm khối lớp
- Xử lý đánh giá từ user
- Xử lý khi nhận kết quả từ Aggregator Agent
- Trường hợp không phải HumanMessage

#### 4.4.2.2.2. *OCR Reader Agent*

Khởi tạo Prompt thực hiện trích xuất dữ liệu từ các loại định dạng file khác nhau.

```
OCR_SYSTEM = """Bạn là OCR Agent chuyên nghiệp. Nhiệm vụ:  
1. Nhận văn bản từ user và chuyển thành dạng text chuẩn  
2. Trả về văn bản đã được xử lý  
3. Luôn trả lời ngắn gọn và đi thẳng vào vấn đề"""
```

Các chức năng quan trọng khác:

- Xử lý văn bản
- Hướng dẫn người dùng chọn loại hình tóm tắt.

#### 4.4.2.2.3. *Spell Checker Agent*

Khởi tạo Prompt để thực hiện sửa lỗi chính tả, dấu câu và các từ sai.

```
SPELLCHECKER_SYSTEM = """Bạn là Spell Checker Agent chuyên nghiệp. Nhiệm vụ:  
1. Kiểm tra và sửa lỗi chính tả trong văn bản  
2. KHÔNG thay đổi nội dung, chỉ sửa lỗi chính tả, dấu câu, và các từ sai  
3. Trả về văn bản đã được sửa"""
```

#### 4.4.2.2.4. *Extractor Agent*

Thực hiện khởi tạo Prompt tóm tắt trích xuất.

```
EXTRACTOR_SYSTEM = """Bạn là Extractor Agent chuyên nghiệp. Nhiệm vụ:  
1. Trích xuất thông tin quan trọng từ văn bản
```

2. KHÔNG thay đổi câu từ, chỉ lấy những phần quan trọng nhất
3. Trả về bản tóm tắt trích xuất ngắn gọn""

#### 4.4.2.2.5. *Abstracter Agent*

Prompt cho tóm tắt diễn giải.

- ABSTRACTER\_SYSTEM** = """Bạn là Abstracter Agent chuyên nghiệp. Nhiệm vụ:
1. Tóm tắt văn bản thành bản diễn giải ngắn gọn
  2. Viết lại theo cách hiểu của bạn nhưng giữ nguyên ý nghĩa
  3. Phù hợp với khối lớp được yêu cầu"""

#### 4.4.2.2.6. *Grade Calibrator Agent*

Prompt cho việc xử lý độ dài, câu từ cho bản tóm tắt.

- GRADE\_CALIBRATOR\_SYSTEM** = """Bạn là Grade Calibrator Agent chuyên nghiệp.
- Nhiệm vụ:
1. Điều chỉnh độ dài và từ vựng phù hợp với khối lớp (1-5)
  2. Nếu là trích xuất: chỉ điều chỉnh độ dài, KHÔNG thay đổi câu từ
  3. Nếu là diễn giải: có thể điều chỉnh từ vựng cho phù hợp với khối lớp"""

#### 4.4.2.2.7. *Evaluator Agent*

Prompt thực hiện việc đánh giá và tiếp nhận phản hồi người dùng.

- EVALUATOR\_SYSTEM** = """Bạn là Evaluator Agent chuyên nghiệp. Nhiệm vụ:
1. Đánh giá chất lượng bản tóm tắt
  2. Đưa ra điểm số từ 0-10 dựa trên độ dễ hiểu
  3. Đưa ra nhận xét và gợi ý cải thiện"""

#### 4.4.2.2.8. *Aggregator Agent*

Prompt tổng hợp các bản tóm tắt và đưa ra bản tóm tắt tốt nhất.

- AGGREGATOR\_SYSTEM** = """Bạn là Aggregator Agent chuyên nghiệp. Nhiệm vụ:
1. Tổng hợp bản tóm tắt cuối cùng

2. Đưa ra kết quả hoàn chỉnh cho user
3. Hỏi user đánh giá về hệ thống""

#### 4.4.2.3. Quá trình triển khai hệ thống

1. Thực hiện gọi các thư viện và model cần thiết cho hệ thống
2. Để dễ quản lý và mang lại mục đích tái sử dụng, sẵn sàng mở rộng, tích hợp trong tương lai. Tạo phiên quản lý là điều tất yếu.
3. Tất cả các phiên quản lý đều được tạo và lưu trong DB Chroma. Tiện ích cho việc lưu dữ liệu dạng vector. Sẵn sàng truy xuất và sử dụng.
4. Khởi tạo AgentState để dễ quản lý luồng hệ thống và các yếu tố thiết yếu để cung cấp cho hệ thống.
5. Hàm coordinator\_router thực hiện ràng buộc luồng xử lý để hệ thống không lan man và có khả năng đưa được kết quả khả quan.
6. Thực hiện thêm các node ứng với từng agent đã cài đặt trước đó. Thêm các edge thể hiện mối quan hệ và thứ tự giữa các node trong hệ thống. Nhằm quản lý hệ thống trật tự và việc xử lý tuần tự.
7. Mọi cuộc hội thoại đều được lưu trong cơ sở dữ liệu Chroma.
8. Sau đó thực hiện cài đặt hàm khởi chạy hệ thống kết hợp với lưu dữ liệu vào Chroma và hàm tiếp tục cuộc hội thoại trước đó.
9. Cuối cùng, người dùng có thể xem lịch sử trò chuyện bằng cách thực hiện hàm read\_long\_term\_memory\_by\_session\_id với đầu vào là id session.

#### 4.4.2.4. Kết quả triển khai

- Các thư viện và model được gọi mượt mà, không bị lỗi sẵn sàng tích hợp vào chương trình.
- Session được khởi tạo sẽ có output như sau:

New session started: session\_20251028\_164313

- Hệ thống MAS tóm tắt văn bản sẽ cho kết quả như sau:

```

⌚ Testing new workflow...
New session started: session_20251028_164348
⌚ Multi-Agent System Summary For Primary School Students
=====
Commands: 'exit', 'clear' (STM), 'clear_all' (STM+LTM), 'mem_stats'

⌚ coordinator_agent: Xin chào! Tôi là trợ lý tóm tắt thông minh cho học sinh tiểu học.

Hãy cung cấp văn bản bạn muốn tóm tắt:
👤: Văn bản của tôi: Ngày khai trường đã đến. Sáng sớm, mẹ mới gọi một câu mà tôi đã vùng dậy, khác hẳn mọi ngày. Loáng một cái, tôi đã chuẩn bị xong mọi thứ. Bỗng nhiên nhìn tôi, còn mẹ cười tủm tỉm. Tôi ríu rít: "Con muốn đến sớm nhất lớp." Tôi háo hức tượng tượng ra cảnh mình đến đầu tiên, cất tiếng chào thật to những bạn đến sau. Nhưng vừa đến cổng trường, tôi đã thấy mấy bạn cùng lớp đang

```

riú rít nói cười ở trong sân. Thi ra, không chỉ mình tôi muôn đến sớm nhất. Tôi chào mẹ, chạy ào vào cùng các bạn. Chúng tôi tranh nhau kể về chuyện ngày hè. Ngay cạnh chúng tôi, mấy em lớp 1 đang rụt rè níu chặt tay bố mẹ, thật giống tôi năm ngoái. Trước các em, tôi cảm thấy mình lớn lắm. Tôi đã là học sinh lớp 2 rồi cơ mà.

✉ coordinator\_agent: Văn bản đã được kiểm tra chính tả:

Văn bản của tôi: Ngày khai trường đã đến. Sáng sớm, mẹ mới gọi một câu mà tôi đã vùng dậy, khác hẳn mọi ngày. Loáng một cái, tôi đã chuẩn bị xong mọi thứ. Bố ngạc nhiên nhìn tôi, còn mẹ cười tủm tỉm. Tôi ríu rít: "Con muốn đến sớm nhất lớp." Tôi háo hức tưởng tượng ra cảnh mình đến đầu tiên, cất tiếng chào thật to những bạn đến sau. Nhưng vừa đến cổng trường, tôi đã thấy mấy bạn cùng lớp đang ríu rít nói cười ở trong sân. Thi ra, không chỉ mình tôi muôn đến sớm nhất. Tôi chào mẹ, chạy ào vào cùng các bạn. Chúng tôi tranh nhau kể về chuyện ngày hè. Ngay cạnh chúng tôi, mấy em lớp 1 đang rụt rè níu chặt tay bố mẹ, thật giống tôi năm ngoái. Trước các em, tôi cảm thấy mình lớn lắm. Tôi đã là học sinh lớp 2 rồi cơ mà.

Bây giờ hãy chọn loại tóm tắt:

1. TRÍCH XUẤT (Extract): Giữ nguyên câu từ quan trọng
2. DIỄN GIẢI (Abstract): Viết lại theo cách hiểu của bạn

Và cho biết khôi lớp (1-5):

[Memory: 6 msgs, 0 prefs]

👤 : Tôi muốn tóm tắt diễn giải và theo lớp 5

✉ coordinator\_agent: ✉ \*\*BẢN TÓM TẮT CUỐI CÙNG\*\*

\*\*Loại tóm tắt:\*\* ABSTRACT

\*\*Khôi lớp:\*\* 5

\*\*Nội dung:\*\*

\*\*Tóm tắt diễn giải cho học sinh lớp 5\*\*

Ngày khai trường là một ngày mới và thú vị! Tôi dậy sớm, chuẩn bị xong rồi đến trường. Ở trường, tôi gặp nhiều bạn cùng lớp, chúng ta kể chuyện ngày hè và cười với nhau. Tôi cũng thấy những em lớp 1 đang rất tin cậy bố mẹ. Tôi cảm thấy mình đã lớn hơn một chút.

(I adjusted the vocabulary and sentence structure to make it more suitable for students in grade 5)

---

Bản tóm tắt đã hoàn thành! Bạn có hài lòng với kết quả này không? Hãy đánh giá hệ thống từ 1-10 điểm.

[Memory: 13 msgs, 0 prefs]

👉 Bye MAS Lịch sử chat đã được lưu.

- Kết quả tóm tắt lưu về sẽ được người dùng truy xuất và quản lý bởi từng mốc thời gian. Mang lại sự rõ ràng và minh bạch.

2025-10-28T16:43:48.062089 session\_20251028\_164348 assistant : assistant: Xin chào! Tôi là trợ lý tóm tắt thông minh cho học sinh tiểu học.

Hãy cung cấp văn bản bạn muốn tóm tắt:

2025-10-28T16:44:17.513223 session\_20251028\_164348 user : user: Văn bản của tôi: Ngày khai trường đã đến. Sáng sớm, mẹ mới gọi một câu mà tôi đã vùng dậy, khác hẳn mọi ngày. Loáng một cái, tôi đã chuẩn bị xong mọi thứ. Bố ngạc nhiên nhìn tôi, còn mẹ cười tủm tỉm. Tôi ríu rít: "Con muốn đến sớm nhất lớp." Tôi háo hức tưởng tượng ra cảnh mình đến đầu tiên, cất tiếng chào thật to những bạn đến sau. Nhưng vừa đến cổng trường, tôi đã thấy mấy bạn cùng lớp đang ríu rít nói cười ở

trong sân. Thì ra, không chỉ mình tôi muôn đến sớm nhất. Tôi chào mẹ, chạy ào vào cùng các bạn. Chúng tôi tranh nhau kể về chuyện ngày hè. Ngay cạnh chúng tôi, mấy em lớp 1 đang rụt rè níu chặt tay bố mẹ, thật giống tôi năm ngoái. Trước các em, tôi cảm thấy mình lớn lăm. Tôi đã là học sinh lớp 2 rồi cơ mà.

2025-10-28T16:44:18.480895 session\_20251028\_164348 user : user: Văn bản của tôi: Ngày khai trường đã đến. Sáng sớm, mẹ mới gọi một câu mà tôi đã vùng dậy, khác hẳn mọi ngày. Loáng một cái, tôi đã chuẩn bị xong mọi thứ. Bố ngạc nhiên nhìn tôi, còn mẹ cười tủm tim. Tôi ríu rít: "Con muốn đến sớm nhất lớp." Tôi háo hức tưởng tượng ra cảnh mình đến đầu tiên, cất tiếng chào thật to những bạn đến sau. Nhưng vừa đến cổng trường, tôi đã thấy mấy bạn cùng lớp đang ríu rít nói cười ở trong sân. Thì ra, không chỉ mình tôi muôn đến sớm nhất. Tôi chào mẹ, chạy ào vào cùng các bạn. Chúng tôi tranh nhau kể về chuyện ngày hè. Ngay cạnh chúng tôi, mấy em lớp 1 đang rụt rè níu chặt tay bố mẹ, thật giống tôi năm ngoái. Trước các em, tôi cảm thấy mình lớn lăm. Tôi đã là học sinh lớp 2 rồi cơ mà.

2025-10-28T16:44:19.617627 session\_20251028\_164348 assistant : assistant: Văn bản đã được nhận! Đang xử lý...

2025-10-28T16:44:19.994311 session\_20251028\_164348 assistant : assistant: Văn bản đã được xử lý:

Văn bản của tôi: Ngày khai trường đã đến. Sáng sớm, mẹ mới gọi một câu mà tôi đã vùng dậy, khác hẳn mọi ngày. Loáng một cái, tôi đã chuẩn bị xong mọi thứ. Bố ngạc nhiên nhìn tôi, còn mẹ cười tủm tim. Tôi ríu rít: "Con muốn đến sớm nhất lớp." Tôi háo hức tưởng tượng ra cảnh mình đến đầu tiên, cất tiếng chào thật to những bạn đến sau. Nhưng vừa đến cổng trường, tôi đã thấy mấy bạn cùng lớp đang ríu rít nói cười ở trong sân. Thì ra, không chỉ mình tôi muôn đến sớm nhất. Tôi chào mẹ, chạy ào vào cùng các bạn. Chúng tôi tranh nhau kể về chuyện ngày hè. Ngay cạnh chúng tôi, mấy em lớp 1 đang rụt rè níu chặt tay bố mẹ, thật giống tôi năm ngoái. Trước các em, tôi cảm thấy mình lớn lăm. Tôi đã là học sinh lớp 2 rồi cơ mà.

Bây giờ hãy chọn loại tóm tắt:

1. TRÍCH XUẤT (Extract): Giữ nguyên câu từ quan trọng

2. DIỄN GIẢI (Abstract): Viết lại theo cách hiểu của bạn

Và cho biết khái lóp (1-5):

2025-10-28T16:44:20.746506 session\_20251028\_164348 assistant : assistant: Văn bản đã được kiểm tra chính tả:

Văn bản của tôi: Ngày khai trường đã đến. Sáng sớm, mẹ mới gọi một câu mà tôi đã vùng dậy, khác hẳn mọi ngày. Loáng một cái, tôi đã chuẩn bị xong mọi thứ. Bố ngạc nhiên nhìn tôi, còn mẹ cười tủm tim. Tôi ríu rít: "Con muốn đến sớm nhất lớp." Tôi háo hức tưởng tượng ra cảnh mình đến đầu tiên, cất tiếng chào thật to những bạn đến sau. Nhưng vừa đến cổng trường, tôi đã thấy mấy bạn cùng lớp đang ríu rít nói cười ở trong sân. Thì ra, không chỉ mình tôi muôn đến sớm nhất. Tôi chào mẹ, chạy ào vào cùng các bạn. Chúng tôi tranh nhau kể về chuyện ngày hè. Ngay cạnh chúng tôi, mấy em lớp 1 đang rụt rè níu chặt tay bố mẹ, thật giống tôi năm ngoái. Trước các em, tôi cảm thấy mình lớn lăm. Tôi đã là học sinh lớp 2 rồi cơ mà.

Bây giờ hãy chọn loại tóm tắt:

1. TRÍCH XUẤT (Extract): Giữ nguyên câu từ quan trọng

2. DIỄN GIẢI (Abstract): Viết lại theo cách hiểu của bạn

Và cho biết khái lóp (1-5):

2025-10-28T16:44:51.682385 session\_20251028\_164348 user : user: Tôi muốn tóm tắt diễn giải và theo lóp 5

2025-10-28T16:44:52.643220 session\_20251028\_164348 user : user: Tôi muốn tóm tắt diễn giải và theo lóp 5

2025-10-28T16:44:53.350018 session\_20251028\_164348 assistant : assistant: Đã nhận: Tóm tắt abstract cho lóp 5. Đang xử lý...

2025-10-28T16:46:17.584096 session\_20251028\_164348 assistant : assistant: \*\*Tóm tắt diễn giải\*\*

Ngày khai trường đã đến, tôi rất háo hức vì ngày đầu tiên ở trường. Mẹ gọi tôi dậy sớm, và tôi đã chuẩn bị xong mọi thứ. Khi đến trường, tôi thấy nhiều bạn cùng lớp đang ríu rít, nói cười nhau trong sân. Tôi cảm thấy mình lớn hơn khi nhìn những em lớp 1 đang níu chặt tay bố mẹ. Tôi đã là học sinh lớp 2 rồi, và ngày khai trường này rất đặc biệt.

\*\*Tóm tắt diễn giải cho học sinh lớp 5\*\*

Ngày khai trường là một ngày mới và thú vị! Tôi dậy sớm, chuẩn bị xong rồi đến trường. Ở trường, tôi gặp nhiều bạn cùng lớp, chúng ta kể chuyện ngày hè và cười với nhau. Tôi cũng thấy những em lớp 1 đang rất tin cậy bố mẹ. Tôi cảm thấy mình đã lớn hơn một chút.

2025-10-28T16:47:11.388648 session\_20251028\_164348 assistant : assistant: \*\*Tóm tắt diễn giải cho học sinh lớp 5\*\*

Ngày khai trường là một ngày mới và thú vị! Tôi dậy sớm, chuẩn bị xong rồi đến trường. Ở trường, tôi gặp nhiều bạn cùng lớp, chúng ta kể chuyện ngày hè và cười với nhau. Tôi cũng thấy những em lớp 1 đang rất tin cậy bố mẹ. Tôi cảm thấy mình đã lớn hơn một chút.

(I adjusted the vocabulary and sentence structure to make it more suitable for students in grade 5)

2025-10-28T16:48:53.458067 session\_20251028\_164348 assistant : assistant: \*\*Assessment\*\*

\*\*Quality:\*\* 8/10

The abstract summary is clear and concise, making it easy for students in grade 5 to understand. The language used is simple and relatable, with vivid descriptions of the author's emotions and experiences on their first day of school.

\*\*Accuracy:\*\* 9/10

The summary accurately captures the main ideas and events from the original text. It correctly conveys the author's excitement and eagerness to start school, as well as their observations about other students and the atmosphere at school.

\*\*Clarity:\*\* 9/10

The language used in the summary is clear and easy to follow. The sentences are well-structured and logically connected, making it easy for readers to understand the main points.

\*\*Suggestions for improvement:\*\*

1. Add more specific details: While the summary provides a good overview of the original text, it could benefit from more specific details about the author's experiences and observations.
2. Use transition words: The summary jumps slightly between ideas; using transition words or phrases (e.g., "Meanwhile," "In addition") can help to connect the sentences more smoothly.

Overall, the abstract summary is well-written and effectively conveys the main ideas from the original text. With some minor tweaks, it could be even stronger!

2025-10-28T16:48:54.097375 session\_20251028\_164348 assistant : assistant:   
\*\*BẢN TÓM TẮT CUỐI CÙNG\*\*

\*\*Loại tóm tắt:\*\* ABSTRACT

\*\*Khối lớp:\*\* 5

\*\*Nội dung:\*\*

\*\*Tóm tắt diễn giải cho học sinh lớp 5\*\*

Ngày khai trường là một ngày mới và thú vị! Tôi dậy sớm, chuẩn bị xong rồi đến trường. Ở trường, tôi gặp nhiều bạn cùng lớp, chúng ta kể chuyện ngày hè và cười với nhau. Tôi cũng thấy những em lớp 1 đang rất tin cậy bố mẹ. Tôi cảm thấy mình đã lớn hơn một chút.

(I adjusted the vocabulary and sentence structure to make it more suitable for students in grade 5)

---

Bản tóm tắt đã hoàn thành! Bạn có hài lòng với kết quả này không? Hãy đánh giá hệ thống từ 1-10 điểm.

2025-10-28T16:49:05.958412 session\_20251028\_164348 user : user: thoát

## 4.5. Xây dựng API

Hầu hết các API của hệ thống đã hoạt động tốt và đã trải qua các lần kiểm thử và theo dõi, để đưa ra các giải pháp và sự kết hợp và phát triển trong tương lai đối với hệ thống đa tác tử và ReactJS với Web, React Native với Mobile.

### 4.5.1. Message Controller

Method	Address	Chú thích	Thử nghiệm
GET	/message	Lấy tất cả đoạn hội thoại.	Đã hoạt động tốt và ổn định
GET	/message/{ messageid }	Lấy 1 đoạn hội thoại bằng messageid	
DEL	/message/{ messageid }	Xóa 1 đoạn hội thoại bằng messageid	

Bảng 4.5-1. Kết quả của bộ API Message Controller

❖ Demo

#### GET /message

<i>Input</i>	(None)
<i>Output</i>	[ {         "message_id": "4f1edb19-7aa4-4c67-b33a-e578d22b2b85",         "user_id": null,         "role": "USER",         "message": "Xin chào, tôi muốn biết thời tiết hôm nay",         "created_at": null       }]

### 4.5.2. Read History Controller

Method	Address	Chú thích	Thử nghiệm
GET	/api/read-history/user/{userid}	Lịch sử của userid	Đã hoạt động tốt và ổn định
POST	/api/read-history/log?userId={...}&summaryId={...}	Tạo log lịch sử	

Bảng 4.5-2. Kết quả của bộ API Read History Controller

❖ Demo

**POST** /api/read-history/log?userId=f2110233-9231-4ece-b102-d81e4816cd4f&summaryId=36032566-9ca1-4244-93bb-6f7b0539f4e6

<b>Input</b>	(None)
<b>Output</b>	<pre>{     "id": 4,     "user": {         "userId": "f2110233-9231-4ece-b102-d81e4816cd4f",         "username": "USER234117aaa",         "password": "\$2a\$10\$cQ0cstf6ghJSni9hHEMjiOzB5rYgw9AgpTUK2sanwQn20KLYTDu2",         "role": "CHILD",         "email": "user234117sss@example.com",         "fullName": "User 234117",         "phoneNumber": "0123456789",         "avatarUrl": "https://res.cloudinary.com/demo/image/upload/v123456/new-avatar.jpg",         "active": true     },     "summary": {         "summaryId": "36032566-9ca1-4244-93bb-6f7b0539f4e6",         "title": "Ảnh hưởng của trí tuệ nhân tạo trong giáo dục hiện đại",         "content": "Trí tuệ nhân tạo (AI) đang ngày càng được ứng dụng trong lĩnh vực giáo dục. Các hệ thống học tập thông minh có khả năng phân tích hành vi học tập của học sinh và đề xuất nội dung phù hợp, giúp cá nhân hóa việc học và nâng cao hiệu quả giảng dạy.",         "summaryContent": "AI giúp cá nhân hóa việc học và tăng hiệu quả giáo dục thông qua hệ thống học tập thông minh.",         "imageUrl": "https://res.cloudinary.com/demo/image/upload/v1720000000/ai-education.jpg",         "createdBy": {             "userId": "f2110233-9231-4ece-b102-d81e4816cd4f",             "username": "USER234117aaa",             "password": "\$2a\$10\$cQ0cstf6ghJSni9hHEMjiOzB5rYgw9AgpTUK2sanwQn20KLYTDu2",             "role": "CHILD",             "email": "user234117sss@example.com",             "fullName": "User 234117",             "phoneNumber": "0123456789",             "avatarUrl": "https://res.cloudinary.com/demo/image/upload/v123456/new-avatar.jpg",             "active": true         },         "status": "APPROVED",         "grade": "Grade 5",         "method": "T5_DIEN_GIAI",         "readCount": 0,     } }</pre>

```

    "createdAt": "2025-10-22T07:07:02.808+00:00",
    "approvedAt": null
}
}

```

#### 4.5.3. Summary Controller

Method	Address	Chú thích	Thứ nghiệm
POST	/api/summaries	Tạo tóm tắt	Đã hoạt động tốt và ổn định
GET	/api/summaries/admin	Lấy tóm tắt admin	
PUT	/api/summaries/{sumid}status	Cập nhật status	
GET	/api/summaries/status/{status}	Lấy tóm tắt với status	
GET	/api/summaries/contributor/{userid}	Lấy tóm tắt với contributor	
GET	/api/summaries/grade/{grade}	Lấy tóm tắt với grade	
GET	/api/summaries/method/{method}	Lấy tóm tắt với method	
GET	/api/summaries	Lấy tất cả bản tóm tắt	
GET	/api/summaries/{sumid}?userId={userid}	Lấy bản tóm tắt với id	
GET	/api/summaries/top10	Lấy top 10 bản tóm tắt	
GET	/api/summaries/search?searchTerm={...}&grade={grade}	Lấy bản tóm tắt với title hoặc grade	
PATCH	/api/summaries/{sumid}	Cập nhật bản tóm tắt với id	
DEL	/api/summaries/{sumid}	Xóa bản tóm tắt với id	
DEL	/api/summaries/bulk	Xóa nhiều bản tóm tắt	

Bảng 4.5-3. Kết quả của bộ API Summary Controller

Đã hoạt động tốt và ổn định

## ❖ Demo

**POST** /api/summaries

<b>Input</b>	<pre>{     "title": "Ảnh hưởng của trí tuệ nhân tạo trong giáo dục hiện đại",     "content": "Trí tuệ nhân tạo (AI) đang ngày càng được ứng dụng trong lĩnh vực giáo dục. Các hệ thống học tập thông minh có khả năng phân tích hành vi học tập của học sinh và đề xuất nội dung phù hợp, giúp cá nhân hóa việc học và nâng cao hiệu quả giảng dạy.",     "summaryContent": "AI giúp cá nhân hóa việc học và tăng hiệu quả giáo dục thông qua hệ thống học tập thông minh.",     "imageUrl": "https://res.cloudinary.com/demo/image/upload/v1720000000/ai-education.jpg",     "createdBy": {         "userId": "f2110233-9231-4ece-b102-d81e4816cd4f"     },     "grade": "Grade 5",     "method": "T5_DIEN_GIAI",     "readCount": 0 }</pre>
<b>Output</b>	<pre>{     "summaryId": "6a19294a-6bba-4a81-94b4-f807405aaaf51",     "title": "Ảnh hưởng của trí tuệ nhân tạo trong giáo dục hiện đại",     "content": "Trí tuệ nhân tạo (AI) đang ngày càng được ứng dụng trong lĩnh vực giáo dục. Các hệ thống học tập thông minh có khả năng phân tích hành vi học tập của học sinh và đề xuất nội dung phù hợp, giúp cá nhân hóa việc học và nâng cao hiệu quả giảng dạy.",     "summaryContent": "AI giúp cá nhân hóa việc học và tăng hiệu quả giáo dục thông qua hệ thống học tập thông minh.",     "imageUrl": "https://res.cloudinary.com/demo/image/upload/v1720000000/ai-education.jpg",     "createdBy": {         "userId": "f2110233-9231-4ece-b102-d81e4816cd4f",         "username": "USER234117aaa",         "password": "\$2a\$10\$cQ0cstf6ghJSni9hHEMjiOzB5rYgw9AgptUK2sanwQn20KLYTDu2",         "role": "CHILD",         "email": "user234117sss@example.com",         "fullName": "User 234117",         "phoneNumber": "0123456789",         "avatarUrl": "https://res.cloudinary.com/demo/image/upload/v123456/new-avatar.jpg",         "active": true     },     "status": "PENDING",     "grade": "Grade 5",     "method": "T5_DIEN_GIAI",     "readCount": 0,     "createdAt": "2025-10-29T16:22:12.347+00:00",     "approvedAt": null }</pre>

#### 4.5.4. Summary History Controller

(Đang phát triển)

#### 4.5.5. Summary Session Controller

Method	Address	Chú thích	Thử nghiệm
POST	/api/summary-sessions	Tạo phiên tóm tắt	Đã hoạt động tốt và ổn định
GET	/api/summary-sessions/{id}	Lấy phiên tóm tắt với id	
POST	/api/summary-sessions/process-pdf	Xử lý PDF	
PUT	/api/summary-sessions/{id}	Cập nhật phiên với id	
POST	/api/summary-sessions/generate-image	Tạo hình ảnh	
POST	/api/summary-sessions/upload-image	Tải ảnh lên cloudinary	
DEL	/api/summary-sessions/{id}	Xóa phiên tóm tắt với id	

Bảng 4.5-4. Kết quả của bộ API Summary Session Controller

#### ❖ Demo

**POST** /api/summary-sessions/generate-image

<b>Input</b>	<pre>{   "content": "A beautiful sunset over the mountains with birds flying in the sky" }</pre>
<b>Output</b>	<pre>{   "imageUrl": "https://res.cloudinary.com/dsm5p4uql/image/upload/v1761755099/summary_sessions/image16436143906027108668.png",   "success": true }</pre>

#### 4.5.6. Summary Tag Controller

Method	Address	Chú thích	Thử nghiệm
GET	/api/summary-tags/summary/{tagid}	Lấy tất cả tag bởi id tóm tắt	Đã hoạt động tốt và ổn định
GET	/api/summary-tags/tag/{sumid}	Lấy tất cả tóm tắt bởi tag id	
POST	/api/summary-tags	Tạo tag tóm tắt	
DEL	/api/summary-tags/{id}	Xóa tag tóm tắt bởi id tag	

Bảng 4.5-5. Kết quả của bộ API Summary Tag Controller

#### ❖ Demo

**DEL** <http://localhost:8080/api/summary-tags/5d4efa85-572f-4000-a001-ea882239843d>

<i>Input</i>	(None)
<i>Output</i>	Status 204

#### 4.5.7. Tag Controller

Method	Address	Chú thích	Thử nghiệm
GET	/api/tags	Lấy tag	Đã hoạt động tốt và ổn định
GET	/api/tags/{id}	Lấy tag với id	
GET	/api/tags/name/{name}	Lấy tag với name	
GET	/api/tags/name/{name}/all	Lấy tất cả tag với name	
POST	/api/tags	Tạo tag mới	
DEL	/api/tags/{id}	Xóa tag bởi id	

Bảng 4.5-6. Kết quả của bộ API Tag Controller

#### ❖ Demo

**GET** </api/tags>

<i>Input</i>	(None)
<i>Output</i>	[{"id": 1, "name": "Tag 1"}]

```

        "tagId": "36a3db8f-f23a-48b9-bb31-1b544ad9adb8",
        "name": "CNTT"
    },
    {
        "tagId": "f982c3f1-5305-401e-bfd9-ec0ea8e120e8",
        "name": "AI"
    }
]

```

#### 4.5.8. User Controller

Method	Address	Chú thích	Thử nghiệm
GET	/api/users	Lấy user	Đã hoạt động tốt và ổn định
GET	/api/users/{id}	Lấy user với id	
POST	/api/users	Tạo user mới	
PUT	/api/users/{id}/profile	Cập nhật profile	
PUT	/api/users/{id}	Cập nhật user	
POST	/api/users/auth/login	Xác thực login	
DEL	/api/users/{id}	Xóa user với id	
GET	/api/users/role/{role}	Lấy user bởi role	
PUT	/api/users/{id}/avatar	Cập nhật user avatar	

Bảng 4.5-7. Kết quả của bộ API User Controller

#### ❖ Demo

**PUT**      /api/users/f2110233-9231-4ece-b102-d81e4816cd4f

<b>Input</b>	<pre>{     "username": "john12355555",     "password": "newpassword123",     "role": "ADMIN",     "email": "john@example.com",     "fullName": "John Doe",     "phoneNumber": "0123456789",     "avatarUrl": "https://res.cloudinary.com/demo/image/upload/v123456/avatar.jpg",     "active": true }</pre>
<b>Output</b>	<pre>{     "userId": "f2110233-9231-4ece-b102-d81e4816cd4f",     "username": "john12355555",</pre>

```
"password": "$2a$10$zyg8WTlkINZy8.QZvPzfRO3tossRxD6iiHJ56yADwUfMKYwqtcpGm",
  "role": "ADMIN",
  "email": "john@example.com",
  "fullName": "John Doe",
  "phoneNumber": "0123456789",
  "avatarUrl":
  "https://res.cloudinary.com/demo/image/upload/v123456/avatar.jpg",
  "active": true
}
```

## 4.6. Đánh giá kết quả

### 4.6.1. Hệ thống đa tác tử

Hệ thống Multi-Agent System (MAS) được xây dựng đã hoàn thiện phiên bản demo và hoạt động ổn định.

Các tác tử (agents) trong hệ thống như Coordinator Agent, OCR Agent, Spell Checker Agent, Extract Agent, Abstract Agent, Grade Calibrator Agent, Evaluator Agent, Aggregator Agent đều phối hợp nhịp nhàng thông qua cơ chế điều phối của LangGraph và LangChain.

Phiên bản demo cho thấy hệ thống có khả năng:

- Tự động phối hợp các tác tử mà không cần can thiệp thủ công.
- Sinh ra bản tóm tắt phù hợp với cấp độ học sinh tiểu học.
- Đảm bảo độ chính xác và mạch lạc cao trong kết quả đầu ra.

### 4.6.2. Hệ thống ứng dụng Spring Boot

Hệ thống backend được xây dựng bằng Java Spring Boot đóng vai trò trung tâm trong việc quản lý luồng dữ liệu, kết nối và giao tiếp giữa ứng dụng người dùng và hệ thống đa tác tử.

Ứng dụng này đảm bảo khả năng mở rộng, bảo mật, và dễ dàng tích hợp với các module AI được triển khai bằng Python.

Hệ thống Spring Boot không chỉ đóng vai trò là bộ khung kết nối (integration layer) mà còn giúp đảm bảo tính ổn định, bảo mật và mở rộng của toàn bộ hệ thống tóm tắt đa tác tử. Phiên bản hiện tại đã đạt mức sẵn sàng để triển khai tích hợp mô hình đa tác tử.

# CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Đánh giá kết quả đạt được

Sau quá trình nghiên cứu, thiết kế và xây dựng, hệ thống tóm tắt truyện và bài đọc cho học sinh tiểu học dựa trên kiến trúc đa tác tử (Multi-Agent System – MAS) đã hoàn thành phiên bản demo và hoạt động ổn định. Các mục tiêu ban đầu của đề tài đã được đáp ứng ở mức tốt và khả thi cả về mặt kỹ thuật lẫn ứng dụng thực tế.

Kết quả nổi bật đạt được:

- Hoàn thiện kiến trúc hệ thống đa tác tử. Hệ thống được thiết kế gồm nhiều tác tử (Agents) có vai trò riêng biệt, phối hợp thông qua LangGraph và LangChain, đảm bảo tính linh hoạt, dễ mở rộng và dễ kiểm soát luồng xử lý.
- Tích hợp mô hình ngôn ngữ lớn (LLMs). Ứng dụng Ollama với mô hình LLaMA 3:8B đã mang lại hiệu quả cao trong quá trình tóm tắt, đặc biệt ở các bài đọc dài hoặc yêu cầu diễn giải theo cấp lớp.
- Hai chế độ tóm tắt (Extractive & Abstractive). Người dùng có thể lựa chọn giữa hai phương thức tóm tắt – trích xuất hoặc diễn giải – giúp nội dung đầu ra phù hợp với từng mục đích học tập.
- Tự động hóa quá trình đánh giá và hiệu chỉnh. Các agent như Grade Calibrator Agent và Evaluator Agent giúp đảm bảo tóm tắt cuối cùng có độ phù hợp, ngữ pháp và độ dài phù hợp với từng cấp độ học sinh.
- Hệ thống backend ổn định với Spring Boot. Ứng dụng backend bằng Java Spring Boot đảm bảo khả năng giao tiếp ổn định giữa client (ReactJS/React Native) và hệ thống đa tác tử, đồng thời dễ dàng mở rộng trong tương lai.
- Tương thích đa nền tảng. Hệ thống có thể triển khai trên web và mobile, tạo điều kiện cho học sinh, giáo viên hoặc phụ huynh dễ dàng tiếp cận và sử dụng.

## 5.2. Hạn chế và thách thức

Mặc dù hệ thống tóm tắt truyện và bài đọc dựa trên kiến trúc đa tác tử (MAS) đã đạt được nhiều kết quả tích cực, tuy nhiên trong quá trình nghiên cứu và triển khai vẫn tồn tại một số hạn chế và thách thức nhất định cần được khắc phục trong giai đoạn tiếp theo.

### 1. Hạn chế về mô hình ngôn ngữ và độ chính xác

Mô hình LLaMA 3: 8B được triển khai thông qua Ollama hoạt động tốt với tiếng Anh, nhưng chưa tối ưu hoàn toàn cho tiếng Việt, đặc biệt là với văn bản dành cho học sinh tiểu học. Một số bản tóm tắt diễn giải (abstractive) vẫn còn thiếu tự nhiên hoặc bỏ sót ý chính khi xử lý các bài đọc dài và phức tạp.

## 2. Độ phụ thuộc vào tài nguyên hệ thống

Mô hình LLM có kích thước lớn (8B tham số) đòi hỏi hiệu năng phần cứng cao (RAM, GPU), gây khó khăn khi triển khai trên các thiết bị hạn chế tài nguyên hoặc server giá rẻ. Việc chạy nhiều tác tử đồng thời trong LangGraph có thể làm tăng độ trễ xử lý nếu không tối ưu hóa pipeline.

## 3. Giới hạn về dữ liệu huấn luyện và ngữ cảnh

Hệ thống mới chủ yếu thử nghiệm trên tập dữ liệu nhỏ và có kiểm soát, chưa được đánh giá trên quy mô lớn hoặc đa dạng thể loại truyện. Chưa có cơ chế tự động học và cập nhật tri thức để mô hình cải thiện theo thời gian sử dụng.

## 4. Thách thức trong tích hợp hệ thống

Việc kết nối giữa backend Spring Boot (Java) và pipeline đa tác tử (Python) vẫn cần tối ưu hóa về tốc độ và định dạng truyền dữ liệu (đặc biệt với các tệp văn bản lớn). Quản lý trạng thái (state) giữa các agent và API đôi khi phức tạp, dễ gây lỗi trong môi trường phân tán.

## 5. Trải nghiệm người dùng

Chưa có giao diện tích hợp, hệ thống có độ trễ lớn khi gọi các mô hình. Điều đó làm cho thời gian phản hồi yêu cầu người dùng lâu hơn. Cần có thêm tính năng chấm điểm và cá nhân hóa nội dung tóm tắt để nâng cao trải nghiệm học tập.

## 5.3. Hướng phát triển tương lai

### 1. Huấn luyện mô hình ngôn ngữ chuyên biệt

Tiến hành train một mô hình ngôn ngữ riêng cho tiếng Việt, tập trung vào tóm tắt truyện và bài đọc cấp tiểu học, nhằm nâng cao chất lượng diễn đạt, tính tự nhiên và độ phù hợp với học sinh. Mô hình này có thể được fine-tune từ LLaMA, mT5 hoặc PhoBERT, sử dụng tập dữ liệu được gán nhãn theo độ tuổi và cấp lớp. Kết hợp cơ chế reinforcement learning từ phản hồi của người dùng (RLHF) để mô hình tự học và cải thiện theo thời gian.

### 2. Tích hợp API GPT và mô hình thương mại

Sử dụng API KEY GPT (OpenAI) như một bộ phản hồi dự phòng (fallback model) nhằm hỗ trợ những yêu cầu phức tạp hoặc kiểm tra độ chính xác của mô hình tự huấn luyện. So sánh kết quả giữa GPT và mô hình nội bộ để đánh giá chất lượng tóm tắt tự động và tối ưu đầu ra.

### 3. Hoàn thiện hệ thống backend

Tích hợp trực tiếp đa tác tử pipeline (Python) vào Spring Boot thông qua RESTful API hoặc gRPC, giúp hệ thống hoạt động liền mạch, giảm độ trễ và tăng tính ổn định.

Tối ưu cơ chế lưu trữ kết quả và lịch sử tóm tắt, đồng thời bổ sung các module bảo mật, logging, và giám sát hiệu năng.

#### 4. Phát triển giao diện người dùng (Web & Mobile)

Triển khai hệ thống trên nền tảng web (ReactJS) và mobile (React Native) để người dùng dễ dàng truy cập. Thiết kế giao diện trực quan, thân thiện với học sinh, cho phép người dùng:

- Nhập văn bản hoặc chụp ảnh truyện để tóm tắt (OCR).
- Lựa chọn hình thức tóm tắt (trích xuất / diễn giải).
- Xem, lưu và chia sẻ bản tóm tắt.

#### 5. Mở rộng khả năng của hệ thống đa tác tử

Nâng cấp CoordinatorAgent để có khả năng học hỏi từ hành vi người dùng, tự động tối ưu lựa chọn giữa các tác tử. Bổ sung FeedbackAgent giúp thu thập phản hồi, DataCollectorAgent để mở rộng tập dữ liệu huấn luyện. Xây dựng EvaluationAgent thông minh có thể chấm điểm nội dung dựa trên các tiêu chí như ngữ pháp, tính mạch lạc và độ phù hợp cấp lớp.

#### 6. Ứng dụng thực tế và triển khai

Đưa hệ thống lên nền tảng đám mây (Cloud) để người dùng truy cập mọi lúc mọi nơi. Phát triển thành ứng dụng học tập hỗ trợ giáo viên và học sinh, giúp rèn luyện kỹ năng đọc hiểu và tổng hợp nội dung. Hướng tới việc tích hợp hệ thống vào các nền tảng giáo dục thông minh (Smart Education) hoặc ứng dụng hỗ trợ học tập tiếng Việt.

## Tài liệu tham khảo

- [1] LangChain - <https://www.langchain.com/>
- [2] LangGraph - <https://www.langgraph.com/>
- [3] Artificial Intelligence A Modern Approach Third Edition – Russell\_Norvig
- [4] Hướng dẫn sử dụng LlamaIndex qua dự án chăm sóc sức khỏe tinh thần – Dinh Tiem Nguyen & Quang Vinh Dinh
- [5] Machine Learning cơ bản – Vũ Hữu Tiệp
- [6] Deep Learning cơ bản v2 – Nguyễn Thanh Tuấn
- [7] Google Agent Development Kit - <https://google.github.io/adk-docs/>
- [8] Google Cloud - <https://console.cloud.google.com/>

## KẾT LUẬN

Em xin gửi lời biết ơn sâu sắc đến thầy Hoàng Ngọc Long đã dành nhiều thời gian và tâm huyết hướng dẫn nghiên cứu và giúp em hoàn thành môn học.

Em xin chân thành cảm ơn đến toàn thể ban lãnh đạo Công ty TNHH TMDV XCODI, cùng các cô, chú, anh, chị cán bộ công nhân viên Công ty TNHH TMDV XCODI đã tận tình hướng dẫn và tạo điều kiện giúp đỡ em trong suốt quá trình thực tập tại Công ty để em hoàn thành tốt báo cáo thực tập tốt nghiệp này.

Em đã có nhiều cố gắng hoàn thiện dự án bằng tất cả năng lực của mình, tuy nhiên không thể tránh khỏi nhiều thiếu sót, rất mong nhận được những đóng góp quý báu của quý thầy cô và quý công ty.