[BÁO CÁO CHI TIẾT HỆ THỐNG CHATBOTYTE - TƯ VẤN SỨC KHỎE 1](#_Toc201180872)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc201180873)

[1.1. Cơ sở lý thuyết 2](#_Toc201180874)

[1.2. Phát biểu bài toán 3](#_Toc201180875)

[1.3. Kết chương 3](#_Toc201180876)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 3](#_Toc201180877)

[2.1. Phát biểu bài toán 4](#_Toc201180878)

[2.2. Phân tích tình trạng 4](#_Toc201180879)

[2.3. Phân tích chức năng 4](#_Toc201180880)

[2.4. Biểu đồ use case 5](#_Toc201180881)

[2.5. Đặc tả ca sử dụng 5](#_Toc201180882)

[2.6. Thiết kế cơ sở dữ liệu 5](#_Toc201180883)

[2.7. Thiết kế hệ thống 6](#_Toc201180884)

[2.8. Triển khai hệ thống 6](#_Toc201180885)

[2.9. Kết chương 6](#_Toc201180886)

[CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 6](#_Toc201180887)

[3.1. Mô hình triển khai 6](#_Toc201180888)

[3.2. Môi trường để chạy chương trình 6](#_Toc201180889)

[3.3. Kết quả thực nghiệm 6](#_Toc201180890)

[3.4. Đánh giá hệ thống 7](#_Toc201180891)

[3.5. Kết chương 7](#_Toc201180892)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 7](#_Toc201180893)

[4.1. Kết quả đạt được 7](#_Toc201180894)

[4.2. Hạn chế và giải pháp 7](#_Toc201180895)

[4.3. Hướng phát triển 8](#_Toc201180896)

[PHỤ LỤC 8](#_Toc201180897)

[A. Tính năng nâng cao Google Gemini 8](#_Toc201180898)

[B. Kỹ thuật Prompt Engineering y tế 9](#_Toc201180899)

# BÁO CÁO CHI TIẾT HỆ THỐNG CHATBOTYTE - TƯ VẤN SỨC KHỎE

## CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

### 1.1. Cơ sở lý thuyết

#### 1.1.1. FastAPI

FastAPI là một framework hiện đại, hiệu suất cao để xây dựng API với Python. Được sử dụng làm nền tảng backend chính cho ChatBotYte với các ưu điểm: tốc độ cao, tự động tạo tài liệu API, validation dữ liệu tích hợp và hỗ trợ async.

#### 1.1.2. LlamaIndex

LlamaIndex là framework xử lý dữ liệu và truy xuất thông tin cho các ứng dụng AI. Trong dự án, LlamaIndex được sử dụng để quản lý, lập chỉ mục và truy xuất thông tin từ nguồn dữ liệu y tế lớn.

#### 1.1.3. React.js & Shaden

React là thư viện JavaScript phổ biến dùng để xây dựng giao diện người dùng, và Shaden là bộ thư viện UI components được sử dụng để tạo giao diện hiện đại cho ChatBotYte.

#### 1.1.4. PostgreSQL & SQLAlchemy & Alembic

PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở. SQLAlchemy là thư viện ORM (Object-Relational Mapping) cho Python. Alembic là công cụ quản lý phiên bản cơ sở dữ liệu, giúp quản lý migrations.

#### 1.1.5. Qdrant

Qdrant là vector database mã nguồn mở phục vụ tìm kiếm tương tự và lọc theo metadata. Hỗ trợ các thuật toán tìm kiếm vector hiệu suất cao cho các bài toán tìm kiếm ngữ nghĩa.

#### 1.1.6. Supabase

Supabase là một nền tảng backend-as-a-service mã nguồn mở, cung cấp các dịch vụ như authentication, database, storage, và serverless functions.

#### 1.1.7. Elasticsearch

Elasticsearch là công cụ tìm kiếm và phân tích phân tán, được sử dụng để tìm kiếm toàn văn và triển khai thuật toán BM25 cho hybrid search.

#### 1.1.8. Retrieval - Augmented Generation (RAG)

RAG là phương pháp kết hợp truy xuất thông tin (retrieval) và sinh văn bản (generation). Mô hình RAG đầu tiên truy xuất thông tin liên quan từ kho dữ liệu, sau đó sử dụng thông tin này làm ngữ cảnh để sinh ra câu trả lời chính xác.

#### 1.1.9. AI Agent

AI Agent là hệ thống AI có khả năng chủ động đưa ra quyết định và thực hiện hành động dựa trên mục tiêu, ngữ cảnh và tri thức. Trong ChatBotYte, AI Agent được sử dụng để quản lý luồng hội thoại và thực hiện các tác vụ phức tạp.

#### 1.1.10. AI Workflow

AI Workflow là quy trình thiết kế, triển khai và giám sát các hệ thống AI. Bao gồm các bước từ thu thập dữ liệu, tiền xử lý, huấn luyện mô hình đến triển khai và giám sát.

### 1.2. Phát biểu bài toán

#### 1.2.1. Yêu cầu chức năng

Phát triển hệ thống chatbot y tế với khả năng:

* Cung cấp thông tin y tế chính xác từ nguồn đáng tin cậy
* Phân tích các triệu chứng và đưa ra khuyến nghị sơ bộ
* Duy trì ngữ cảnh cuộc trò chuyện liên tục
* Hỗ trợ đa ngôn ngữ với trọng tâm là tiếng Việt
* Bảo mật thông tin y tế người dùng

#### 1.2.2. Yêu cầu phi chức năng

* Độ chính xác cao trong thông tin y tế
* Thời gian phản hồi nhanh (<2 giây)
* Khả năng mở rộng để xử lý nhiều người dùng cùng lúc
* Bảo mật dữ liệu theo các tiêu chuẩn y tế
* Khả năng hoạt động 24/7 với độ tin cậy cao

### 1.3. Kết chương

Chương này đã trình bày tổng quan các công nghệ nền tảng và kiến trúc hệ thống của ChatBotYte. Các công nghệ được lựa chọn kỹ lưỡng để xây dựng một hệ thống chatbot y tế hiệu quả, chính xác và bảo mật.

## CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 2.1. Phát biểu bài toán

ChatBotYte cần trả lời các câu hỏi về sức khỏe từ người dùng bằng cách truy xuất thông tin từ nguồn y tế đáng tin cậy, phân tích nội dung và tạo ra câu trả lời chính xác, dựa trên bằng chứng.

### 2.2. Phân tích tình trạng

Hiện nay tại Việt Nam, các chatbot tư vấn y tế còn hạn chế về độ chính xác, khả năng cá nhân hóa, và tính khả dụng của thông tin y tế tiếng Việt. ChatBotYte nhằm giải quyết các hạn chế này.

### 2.3. Phân tích chức năng

#### 2.3.1. Yêu cầu chức năng

* Xác thực và quản lý người dùng
* Tư vấn sức khỏe thông qua trò chuyện
* Quản lý hồ sơ sức khỏe cá nhân
* Tìm kiếm thông tin y tế từ nguồn đáng tin cậy
* Theo dõi lịch sử trò chuyện và sử dụng

#### 2.3.2. Yêu cầu phi chức năng

* Độ chính xác cao trong thông tin y tế
* Thời gian phản hồi nhanh
* Bảo mật dữ liệu người dùng
* Khả năng mở rộng hệ thống

#### 2.3.3. Quy trình chính của hệ thống

1. Người dùng đăng nhập/đăng ký
2. Gửi câu hỏi về sức khỏe
3. Hệ thống xử lý, truy xuất thông tin liên quan
4. Tạo câu trả lời dựa trên thông tin truy xuất được
5. Lưu lịch sử trò chuyện để tham khảo sau

### 2.4. Biểu đồ use case

#### 2.4.1. Biểu đồ use case tổng quan

Mô tả các tác nhân (người dùng, quản trị viên) và các use case chính của hệ thống ChatBotYte.

#### 2.4.2. Biểu đồ use case chức năng tìm kiếm bệnh

Mô tả chi tiết quy trình người dùng tìm kiếm thông tin về bệnh lý.

#### 2.4.3. Biểu đồ use case chức năng hỏi đáp chatbot

Mô tả chi tiết luồng tương tác giữa người dùng và chatbot.

#### 2.4.4. Biểu đồ use case chức năng quản trị viên

Mô tả các chức năng quản trị hệ thống dành cho admin.

### 2.5. Đặc tả ca sử dụng

#### 2.5.1. Đặc tả usecase chức năng đăng ký người dùng

Mô tả chi tiết quy trình đăng ký tài khoản mới.

#### 2.5.2. Đặc tả usecase chức năng đăng nhập người dùng

Mô tả chi tiết quy trình đăng nhập vào hệ thống.

#### 2.5.3. Đặc tả usecase chức năng tìm kiếm bệnh

Mô tả chi tiết quy trình tìm kiếm thông tin bệnh lý.

#### 2.5.4. Đặc tả usecase chức năng hỏi đáp bệnh với chatbot

Mô tả chi tiết quy trình tương tác với chatbot y tế.

#### 2.5.5. Đặc tả usecase chức năng tùy chỉnh prompt và rule của quản trị viên

Mô tả quy trình quản trị viên cấu hình prompt và rule hệ thống.

#### 2.5.6. Đặc tả usecase chức năng quản lý dữ liệu chatbot của quản trị viên

Mô tả quy trình quản lý dữ liệu y tế và lịch sử trò chuyện.

### 2.6. Thiết kế cơ sở dữ liệu

#### 2.6.1. Cơ sở dữ liệu quan hệ PostgreSQL

Thiết kế schema cho dữ liệu người dùng, hội thoại, phản hồi và cấu hình hệ thống.

#### 2.6.2. Cơ sở dữ liệu vector Qdrant

Thiết kế cấu trúc lưu trữ vector embeddings cho RAG pipeline.

### 2.7. Thiết kế hệ thống

#### 2.7.1. Hệ thống crawl data bệnh tật

Thiết kế và triển khai hệ thống thu thập dữ liệu y tế từ các nguồn đáng tin cậy.

#### 2.7.2. Hệ thống tìm kiếm bệnh

Thiết kế hệ thống tìm kiếm thông tin bệnh lý bằng kết hợp tìm kiếm ngữ nghĩa và từ khóa.

#### 2.7.3. Hệ thống quản lý dữ liệu chatbot

Thiết kế quy trình quản lý dữ liệu training, embeddings và cải thiện hệ thống.

#### 2.7.4. Hệ thống hỏi đáp chatbot

Thiết kế luồng xử lý câu hỏi, truy xuất thông tin và tạo câu trả lời.

#### 2.7.5. Thiết kế hệ thống microservice của website

Thiết kế kiến trúc microservice cho khả năng mở rộng và bảo trì.

### 2.8. Triển khai hệ thống

Kế hoạch triển khai từng thành phần hệ thống, bao gồm tiền xử lý dữ liệu, xây dựng embeddings, và triển khai API.

### 2.9. Kết chương

Tóm tắt thiết kế và kiến trúc hệ thống đã được đề xuất, nhấn mạnh các đặc điểm chính.

## CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

### 3.1. Mô hình triển khai

Mô hình triển khai phần mềm với Docker và kiến trúc microservice.

### 3.2. Môi trường để chạy chương trình

Yêu cầu phần cứng và phần mềm cho việc triển khai và vận hành hệ thống.

### 3.3. Kết quả thực nghiệm

#### 3.3.1. Landing page

Giới thiệu giao diện trang chủ và các tính năng chính.

#### 3.3.2. Giao diện đối với người dùng

Trình bày chi tiết giao diện người dùng, bao gồm đăng ký, đăng nhập, trang cá nhân và giao diện chat.

#### 3.3.3. Giao diện đối với quản trị viên

Trình bày chi tiết dashboard quản trị và các công cụ giám sát hệ thống.

### 3.4. Đánh giá hệ thống

#### 3.4.1. Phương pháp LLM-as-a-Judge

Sử dụng LLM để đánh giá chất lượng câu trả lời của chatbot.

#### 3.4.2. Faithfulness Metric

Đánh giá mức độ trung thực của thông tin so với nguồn gốc.

#### 3.4.3. Contextual Relevancy Metric

Đánh giá mức độ liên quan của thông tin với câu hỏi.

#### 3.4.4. Answer Relevancy Metric

Đánh giá mức độ phù hợp của câu trả lời với yêu cầu người dùng.

#### 3.4.5. Quy trình đánh giá hệ thống RAG

Mô tả chi tiết quy trình đánh giá toàn diện hệ thống RAG.

### 3.5. Kết chương

Tóm tắt kết quả triển khai và đánh giá, chỉ ra những thành công và thách thức.

## CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 4.1. Kết quả đạt được

* Xây dựng thành công hệ thống chatbot y tế thông minh với công nghệ RAG
* Tích hợp nhiều mô hình AI tiên tiến (Gemini, GPT, Claude)
* Hỗ trợ đa ngôn ngữ với trọng tâm là tiếng Việt
* Đảm bảo độ chính xác và độ tin cậy của thông tin y tế
* Xây dựng hệ thống bảo mật đạt tiêu chuẩn

### 4.2. Hạn chế và giải pháp

* Đa dạng hóa nguồn dữ liệu y tế tiếng Việt
* Tiếp tục cải thiện độ chính xác của phản hồi
* Tối ưu thời gian phản hồi cho trải nghiệm tốt hơn
* Mở rộng khả năng xử lý đa phương tiện

### 4.3. Hướng phát triển

* Phát triển ứng dụng di động cho ChatBotYte
* Tích hợp với các thiết bị theo dõi sức khỏe
* Triển khai giải pháp AI edge cho khu vực có kết nối internet hạn chế
* Mở rộng hỗ trợ các ngôn ngữ dân tộc thiểu số tại Việt Nam

## PHỤ LỤC

### A. Tính năng nâng cao Google Gemini

#### A.1. Trò chuyện liên tục về một chủ đề

* Duy trì focus vào chủ đề chính trong suốt cuộc trò chuyện
* Tự động phân tích độ liên quan của câu hỏi với chủ đề
* Đề xuất câu hỏi tiếp theo phù hợp

#### A.2. Quản lý ngữ cảnh thông minh

* Lưu trữ lịch sử cuộc trò chuyện
* Phân tích thông tin đã thảo luận
* Tránh lặp lại thông tin
* Xây dựng kiến thức tích lũy

#### A.3. Chế độ tư vấn chuyên nghiệp

* Medical Consultation: Tư vấn y tế chuyên sâu
* Health Education: Giáo dục sức khỏe
* Symptom Analysis: Phân tích triệu chứng
* Lifestyle Advice: Tư vấn lối sống
* General Chat: Trò chuyện tổng quát

#### A.4. API Endpoints chính

* Bắt đầu cuộc trò chuyện: POST /api/v1/chat/advanced-chat/start
* Tiếp tục cuộc trò chuyện: POST /api/v1/chat/advanced-chat/continue
* Chuyển đổi chủ đề: POST /api/v1/chat/advanced-chat/switch-topic
* Lấy tóm tắt cuộc trò chuyện: GET /api/v1/chat/advanced-chat/summary/{session\_id}

### B. Kỹ thuật Prompt Engineering y tế

#### B.1. Chain-of-thought reasoning

Kỹ thuật thiết kế prompt hướng dẫn mô hình suy luận từng bước, đặc biệt hữu ích cho các vấn đề y tế phức tạp.

#### B.2. Các loại prompt chuyên biệt

* Diagnostic Reasoning: Phân tích và suy luận chẩn đoán
* Treatment Advice: Tư vấn điều trị
* Symptom Analysis: Phân tích triệu chứng
* Medication Guidance: Hướng dẫn sử dụng thuốc
* Emergency Assessment: Đánh giá tình trạng khẩn cấp

#### B.3. Tối ưu hóa RAG pipeline

* Tỷ lệ kết hợp tối ưu: BM25 (0.3) và Semantic Search (0.7)
* Top-k retrieval: 5 kết quả phù hợp nhất
* Ngưỡng độ tương đồng: 0.7
* Caching thông minh với Redis