ĐẠI HỌC PHENIKAA

**A logo for a university

Description automatically generatedTRƯỜNG CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO BÀI TẬP KẾT THÚC HỌC PHẦN**

**MÔN HỌC**: **Phân tích và thiết kế phần mềm**  
**TÊN ĐỀ TÀI: Hệ thống Multi-Agent Viet Nam Stock**

**Lớp: PTTKPM\_N05**

**Nhóm**: **011**

**Giảng viên hướng dẫn: TS. Vũ Quang Dũng & TS. Nguyễn Xuân Quế**

| Họ và tên | Mã sinh viên | Gmail |
| --- | --- | --- |
| Nguyễn Minh Dương | 23010441 | [23010441@st.phenikaa-uni.edu.vn](mailto:23010441@st.phenikaa-uni.edu.vn)  https://github.com/nminduo2k5 |

**Hà Nội, 10-2025**

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **Mã**  **sinh viên** | **Công việc được giao** | **Khối lượng công việc** |
| Nguyễn Minh Dương | 23010441 | Tất cả | Tất cả |

# MỤC LỤC

**Chương I: Giới thiệu**  
  1. Lý do chọn đề tài  
  2. Mục tiêu nghiên cứu/phát triển hệ thống  
  3. Phạm vi và đối tượng nghiên cứu  
  4. Tổng quan về thị trường chứng khoán Việt Nam và nhu cầu tự động hóa phân tích

**Chương II. Cơ sở lý thuyết**  
  1. Tổng quan về hệ thống Multi-Agent (Định nghĩa, kiến trúc, ưu điểm)  
  2. Các mô hình Multi-Agent phổ biến  
  3. Công nghệ và ngôn ngữ lập trình sử dụng (Python, framework MAS, …)  
  4. Kiến thức cơ bản về chứng khoán Việt Nam

**Chương III. Phân tích hệ thống**  
  1. Yêu cầu chức năng  
    1.1 Thu thập dữ liệu chứng khoán tự động  
    1.2 Phân tích dữ liệu và đưa ra khuyến nghị  
    1.3 Giao tiếp giữa các agent  
    1.4 Lưu trữ và truy xuất dữ liệu  
    1.5 Giao diện người dùng (nếu có)

  2. Yêu cầu phi chức năng  
    1. Tính mở rộng  
    2. Bảo mật  
    3. Hiệu năng  
    4. Khả năng tích hợp

  3. Các đối tượng và tác nhân (Actor)  
    1. Người dùng cuối  
    2. Agent thu thập dữ liệu  
    3. Agent phân tích dữ liệu  
    4. Agent giao tiếp/hợp tác  
    5. Hệ thống lưu trữ

  4. Biểu đồ ca sử dụng (Use Case Diagram)

**Chương IV. Thiết kế hệ thống**  
  1. Kiến trúc tổng thể hệ thống  
    1. Mô hình kiến trúc (client-server, multi-agent, microservices, …)  
    2. Sơ đồ kiến trúc tổng thể

  2. Thiết kế chi tiết các agent  
    1. Agent thu thập dữ liệu (chức năng, quy trình hoạt động, sơ đồ lớp/sơ đồ trình tự)  
    2. Agent phân tích dữ liệu (chức năng, giải thuật sử dụng, sơ đồ lớp/sơ đồ trình tự)  
    3. Agent hợp tác/giao tiếp (cơ chế trao đổi thông tin, phối hợp tác vụ)

  3. Thiết kế cơ sở dữ liệu  
    1. Mô hình dữ liệu  
    2. Sơ đồ quan hệ thực thể (ERD)  
    3. Các bảng dữ liệu chính

  4. Thiết kế giao diện (nếu có)  
    1. Mô tả giao diện người dùng  
    2. Sơ đồ luồng giao diện

**Chương V. Triển khai và thử nghiệm**  
  1. **Nền tảng và công nghệ sử dụng**  
  2. Quy trình cài đặt và chạy thử  
  3. Kịch bản kiểm thử các chức năng chính  
  4. Đánh giá kết quả

**Chương VI. Đánh giá và hướng phát triển**  
  1. Những điểm mạnh, điểm yếu của hệ thống  
  2. Khó khăn gặp phải và cách khắc phục  
  3. Hướng phát triển trong tương lai (nâng cấp, tích hợp AI, mở rộng phân tích, …)

**Chương VII. Kết luận**

**Chương VIII. Tài liệu tham khảo**

**Chương I: Giới thiệu**

**1. Lý do chọn đề tài**

Thị trường chứng khoán Việt Nam hiện đang phát triển mạnh mẽ, trở thành kênh huy động vốn quan trọng và thu hút sự tham gia ngày càng đông đảo của cả nhà đầu tư cá nhân lẫn tổ chức. Tuy nhiên, khối lượng dữ liệu khổng lồ cùng tốc độ biến động liên tục của thị trường đặt ra nhiều thách thức trong việc phân tích, xử lý thông tin và đưa ra quyết định đầu tư kịp thời. Trong bối cảnh toàn cầu, xu hướng ứng dụng tự động hóa và trí tuệ nhân tạo, đặc biệt là các hệ thống Multi-Agent (đa tác tử), đang chứng minh được tính hiệu quả trong lĩnh vực tài chính. Vì vậy, việc nghiên cứu và xây dựng một hệ thống Multi-Agent hỗ trợ phân tích thị trường chứng khoán Việt Nam là cần thiết, góp phần nâng cao độ chính xác, tốc độ phản ứng và hiệu quả trong việc hỗ trợ ra quyết định đầu tư.

**2. Mục tiêu nghiên cứu/phát triển hệ thống**

Mục tiêu đầu tiên của đề tài là ứng dụng mô hình Multi-Agent vào bài toán phân tích dữ liệu ,dự đoán giá và xu hướng chứng khoán trên thị trường Việt Nam. Việc áp dụng mô hình này hứa hẹn mang lại cách tiếp cận mới mẻ, giúp xử lý thông tin nhanh chóng, đa chiều và có khả năng thích ứng linh hoạt với sự biến động phức tạp của thị trường.

Bên cạnh đó, hệ thống được xây dựng nhằm đảm bảo khả năng tự động thu thập, xử lý dữ liệu, phân tích và đưa ra các khuyến nghị đầu tư cho người dùng. Nhờ vậy, nhà đầu tư có thể tiết kiệm thời gian, tiếp cận thông tin chính xác và kịp thời hơn trong quá trình ra quyết định.

Một mục tiêu quan trọng khác là nâng cao mức độ tự động hóa, qua đó giảm thiểu sự chi phối của yếu tố chủ quan trong phân tích và đầu tư. Điều này góp phần tạo nên sự khách quan, minh bạch và nhất quán trong các khuyến nghị mà hệ thống đưa ra.

Cuối cùng, đề tài hướng tới việc đề xuất một kiến trúc hệ thống cùng các giải pháp công nghệ phù hợp, đảm bảo khả năng mở rộng và dễ dàng tích hợp thêm các chức năng mới trong tương lai. Đây sẽ là nền tảng để phát triển hệ thống thành một công cụ phân tích toàn diện và thông minh hơn, đáp ứng nhu cầu ngày càng đa dạng của thị trường chứng khoán Việt Nam.

**3. Phạm vi và đối tượng nghiên cứu**

Phạm vi của đề tài tập trung vào việc xây dựng một hệ thống Multi-Agent phục vụ cho phân tích dữ liệu chứng khoán Việt Nam. Cụ thể, hệ thống được thiết kế để thu thập và xử lý dữ liệu từ các mã cổ phiếu niêm yết trên ba sàn giao dịch chính bao gồm: Sở Giao dịch Chứng khoán TP. Hồ Chí Minh (HOSE), Sở Giao dịch Chứng khoán Hà Nội (HNX) và thị trường UPCOM. Việc giới hạn phạm vi nghiên cứu trong khuôn khổ các sàn giao dịch này giúp đảm bảo tính khả thi, đồng thời tập trung vào các nguồn dữ liệu có tính chính thống, đa dạng và có ảnh hưởng trực tiếp đến nhà đầu tư trong nước.

Đối tượng nghiên cứu của đề tài bao gồm các kỹ thuật và phương pháp liên quan đến việc xây dựng hệ thống Multi-Agent, từ cơ chế tổ chức, phối hợp đến khả năng giao tiếp giữa các tác tử. Song song với đó, đề tài cũng chú trọng vào các thuật toán phân tích tài chính cơ bản như phân tích xu hướng, chỉ báo kỹ thuật hay so sánh định giá, nhằm làm cơ sở cho việc đưa ra khuyến nghị đầu tư.

Ngoài ra, các công nghệ lập trình hiện đại như Python cùng với những framework hỗ trợ xây dựng Multi-Agent System (MAS) sẽ được khai thác để triển khai hệ thống. Bên cạnh đó, framework **Streamlit** cũng được sử dụng nhằm phát triển giao diện trực quan, cho phép người dùng dễ dàng theo dõi dữ liệu, biểu đồ phân tích và các khuyến nghị đầu tư do hệ thống đưa ra. Streamlit giúp đơn giản hóa quá trình xây dựng ứng dụng web phân tích, hỗ trợ hiển thị kết quả một cách sinh động, thân thiện và dễ sử dụng.

Cuối cùng, dữ liệu chứng khoán Việt Nam được lựa chọn làm nền tảng thực nghiệm, giúp kiểm chứng tính ứng dụng và hiệu quả của hệ thống trong bối cảnh thực tế.

**4. Tổng quan về thị trường Việt Nam**

Thị trường chứng khoán Việt Nam đang có tốc độ phát triển nhanh chóng, với số lượng mã cổ phiếu niêm yết và khối lượng giao dịch ngày càng gia tăng. Nhà đầu tư hiện nay phải đối mặt với khối lượng thông tin khổng lồ, không chỉ bao gồm dữ liệu về giá và khối lượng giao dịch, mà còn cả các yếu tố cơ bản, kỹ thuật và dòng tin tức tác động trực tiếp đến thị trường. Khối lượng dữ liệu đa dạng và biến động liên tục này đặt ra thách thức lớn trong việc xử lý và khai thác thông tin kịp thời để đưa ra quyết định đầu tư chính xác.

Trong bối cảnh đó, nhu cầu về các công cụ phân tích tự động ngày càng trở nên cấp thiết. Những công cụ này không chỉ giúp nhà đầu tư xử lý dữ liệu một cách hiệu quả, mà còn hỗ trợ phát hiện sớm các tín hiệu giao dịch tiềm năng, đồng thời góp phần giảm thiểu rủi ro trong đầu tư. Đặc biệt, các hệ thống dựa trên mô hình Multi-Agent có ưu thế vượt trội nhờ khả năng thu thập dữ liệu từ nhiều nguồn, phân tích theo nhiều chiều cạnh khác nhau và phối hợp đưa ra khuyến nghị một cách linh hoạt. Việc ứng dụng Multi-Agent vào phân tích chứng khoán vì thế hứa hẹn mang lại giải pháp tối ưu, đáp ứng yêu cầu về tốc độ, độ chính xác và tính thích ứng của thị trường hiện đại.

**Chương II: Cơ sở lý thuyết**

1. **Tổng quan về hệ thống Multi-Agent**

**Định nghĩa:** Hệ thống Multi-Agent (MAS – Multi-Agent System) là một hệ thống bao gồm nhiều tác tử (agent) phần mềm có khả năng hoạt động độc lập hoặc phối hợp với nhau nhằm thực hiện những nhiệm vụ cụ thể. Mỗi agent trong hệ thống thường được trang bị khả năng tự động thu thập thông tin, xử lý dữ liệu và ra quyết định dựa trên tri thức cũng như mục tiêu đã được thiết lập.

**Kiến trúc:** Về mặt kiến trúc, một MAS thường được cấu thành từ nhiều agent chuyên biệt, chẳng hạn như agent thu thập dữ liệu, agent phân tích, agent tương tác với người dùng,… Các agent này liên kết với nhau thông qua những giao thức giao tiếp chuẩn hóa, đồng thời được hỗ trợ bởi cơ chế phối hợp và điều phối tác vụ nhằm đảm bảo sự hợp tác hiệu quả trong quá trình xử lý.

**Ưu điểm:** MAS mang lại nhiều lợi thế nổi bật, bao gồm tính linh hoạt trong thiết kế và vận hành, khả năng mở rộng khi hệ thống cần xử lý khối lượng công việc lớn, cũng như mức độ tự động hóa cao trong việc giải quyết các tác vụ phức tạp. Nhờ sự phân chia nhiệm vụ rõ ràng và cơ chế hợp tác hiệu quả giữa các agent, hệ thống MAS có thể nâng cao năng suất, rút ngắn thời gian xử lý và cải thiện chất lượng trong việc đưa ra quyết định.

**2. Các mô hình Multi-Agent phổ biến**  
Hệ thống đa tác tử (Multi-Agent System – MAS) là một hướng nghiên cứu quan trọng trong lĩnh vực trí tuệ nhân tạo, được ứng dụng rộng rãi trong nhiều kịch bản thực tế đòi hỏi sự tự động hóa, phân tán và phối hợp hành động. Các mô hình MAS phổ biến có thể phân loại dựa trên cách thức tổ chức, khả năng ra quyết định và mức độ phức tạp trong hành vi của các tác tử.

* **Tác tử phản xạ (Reactive Agents):** Đây là mô hình đơn giản nhất, trong đó agent phản ứng trực tiếp với môi trường dựa trên các quy tắc định sẵn mà không cần lưu trữ trạng thái hay lập kế hoạch dài hạn. Mô hình này phù hợp với các hệ thống yêu cầu tốc độ phản ứng nhanh, chẳng hạn như robot di động hoặc tác vụ giám sát tự động.
* **Tác tử định hướng mục tiêu (Goal-Based Agents):** Ở mức độ cao hơn, các agent không chỉ phản ứng tức thời mà còn hoạt động dựa trên mục tiêu cụ thể. Chúng có khả năng lập kế hoạch, lựa chọn hành động tối ưu để đạt được trạng thái mong muốn. Loại mô hình này đặc biệt hiệu quả trong các ứng dụng phân tích dữ liệu và ra quyết định, ví dụ như giao dịch chứng khoán tự động hoặc hệ thống khuyến nghị.
* **Tác tử học tập (Learning Agents):** Đây là mô hình tiên tiến, cho phép agent tự cải thiện năng lực của mình thông qua kinh nghiệm hoặc dữ liệu huấn luyện. Bằng cách áp dụng các thuật toán học máy (Machine Learning, Reinforcement Learning), tác tử có thể tối ưu hóa hành vi theo thời gian, thích ứng với sự thay đổi của môi trường. Trong lĩnh vực tài chính, tác tử học tập được sử dụng để dự đoán xu hướng thị trường và điều chỉnh chiến lược đầu tư.
* **Tác tử xã hội (Social Agents):** Ngoài việc hoạt động độc lập, nhiều hệ thống MAS còn tập trung vào khả năng **tương tác, hợp tác hoặc cạnh tranh** giữa các agent. Các agent có thể phối hợp để cùng giải quyết một nhiệm vụ phức tạp (cooperative agents) hoặc đối kháng để tối ưu hóa lợi ích cá nhân (competitive agents). Đây là nền tảng cho các hệ thống mô phỏng kinh tế, đấu giá điện tử, và đặc biệt là thị trường tài chính, nơi các quyết định mua – bán luôn chịu ảnh hưởng từ hành vi tập thể.

Trên thực tế, các mô hình trên không tồn tại hoàn toàn tách biệt mà thường được **kết hợp linh hoạt** tùy theo bài toán. Chẳng hạn, trong hệ thống phân tích chứng khoán, có thể triển khai các tác tử phản xạ để giám sát biến động giá theo thời gian thực, tác tử định hướng mục tiêu để đưa ra chiến lược đầu tư, và tác tử học tập để liên tục cải tiến độ chính xác của mô hình dự báo. Nhờ vậy, MAS trở thành một công cụ mạnh mẽ, hỗ trợ xử lý các bài toán phức tạp trong nhiều lĩnh vực như giao dịch tài chính, quản lý chuỗi cung ứng, điều phối giao thông thông minh, robot tự hành hay các hệ thống phân tán quy mô lớn.

**3. Công nghệ và ngôn ngữ lập trình sử dụng**  
Trong quá trình phát triển hệ thống Multi-Agent, việc lựa chọn ngôn ngữ lập trình và công nghệ đi kèm đóng vai trò then chốt, quyết định đến mức độ hiệu quả, khả năng mở rộng cũng như tính linh hoạt của hệ thống. Trong đó, **Python** được xem là ngôn ngữ phổ biến và được ưa chuộng nhất nhờ cú pháp đơn giản, dễ tiếp cận, đồng thời sở hữu hệ sinh thái thư viện phong phú hỗ trợ nhiều lĩnh vực khác nhau như phân tích dữ liệu, trí tuệ nhân tạo, trực quan hóa và mô phỏng.

Song song với Python, nhiều **framework chuyên dụng cho phát triển MAS** cũng đã được xây dựng nhằm giảm bớt độ phức tạp trong thiết kế và triển khai. Chẳng hạn, **Crew** (dựa trên Python) tập trung vào việc tổ chức và quản lý tác tử theo nhóm, hỗ trợ khả năng cộng tác hiệu quả. **Mesa (Python)** chủ yếu phục vụ cho các mô hình mô phỏng dựa trên agent (Agent-Based Modeling), thích hợp cho các bài toán nghiên cứu và phân tích hành vi. Trong khi đó, **SPADE** nổi bật nhờ khả năng triển khai hệ thống MAS trên môi trường mạng phân tán, sử dụng các chuẩn giao tiếp như XMPP để đảm bảo khả năng trao đổi thông tin giữa các agent.

Ngoài các framework cốt lõi cho MAS, đề tài cũng ứng dụng **Streamlit** – một công cụ mạnh mẽ để xây dựng giao diện web trực quan, đơn giản mà không cần nhiều kiến thức về phát triển front-end. Với Streamlit, hệ thống Multi-Agent có thể dễ dàng cung cấp cho người dùng những bảng biểu, biểu đồ, dashboard trực quan hóa dữ liệu chứng khoán, giúp nhà đầu tư tiếp cận kết quả phân tích một cách nhanh chóng và thuận tiện. Đây cũng là cầu nối quan trọng giữa hệ thống backend (MAS) và người dùng cuối, góp phần nâng cao tính ứng dụng thực tiễn.

Bên cạnh đó, một số **công nghệ bổ trợ** khác cũng đóng vai trò quan trọng trong việc hiện thực hóa hệ thống MAS. Các kỹ thuật **mô hình hóa môi trường** giúp xây dựng bối cảnh hoạt động cho agent, đảm bảo tính chân thực và khả năng phản ứng trước những thay đổi từ bên ngoài. **Xử lý song song (multi-threading, multiprocessing)** nâng cao hiệu suất, cho phép nhiều agent vận hành đồng thời mà không gây tắc nghẽn tài nguyên. Ngoài ra, **giao thức giao tiếp giữa các agent (Agent Communication Protocols – ACPs)** như FIPA-ACL cung cấp nền tảng chuẩn hóa để các agent trao đổi dữ liệu, phối hợp hành động và đưa ra quyết định một cách nhất quán.

Tổng thể, sự kết hợp giữa Python, các framework chuyên dụng (Crew, JADE, Mesa, SPADE), công cụ giao diện như Streamlit cùng những công nghệ hỗ trợ khác đã tạo nên một hệ sinh thái toàn diện, giúp việc phát triển và triển khai hệ thống Multi-Agent trở nên khả thi, hiệu quả và có khả năng mở rộng trong thực tiễn.

**4. Kiến thức cơ bản về thị trường chứng khoán Việt Nam**  
Thị trường chứng khoán Việt Nam gồm ba sàn giao dịch chính: HOSE (TP.HCM), HNX (Hà Nội) và UPCoM (phi tập trung). Các công cụ đầu tư chủ yếu bao gồm cổ phiếu, trái phiếu và chứng chỉ quỹ. Giao dịch được quản lý bởi Ủy ban Chứng khoán Nhà nước (SSC) và tuân theo các quy định chặt chẽ về công bố thông tin và kiểm soát thị trường. Nhà đầu tư cần nắm vững các chỉ số quan trọng như VN-Index, HNX-Index, cũng như hiểu rõ về phân tích cơ bản và kỹ thuật để đưa ra quyết định đầu tư hiệu quả.

**Chương III: Phân tích hệ thống**

**1. Yêu cầu chức năng**

**1. 1 Tự động thu thập dữ liệu thị trường chứng khoán**  
Trong bất kỳ hệ thống phân tích chứng khoán nào, dữ liệu chính là yếu tố nền tảng, quyết định đến chất lượng của toàn bộ quá trình phân tích và dự báo. Do đó, hệ thống Multi-Agent được thiết kế với một nhóm tác tử (data collection agents) chuyên trách việc **tự động thu thập dữ liệu thị trường chứng khoán** từ nhiều nguồn khác nhau. Các agent này hoạt động theo cơ chế phân tán và song song, giúp tối ưu hóa tốc độ thu thập cũng như đảm bảo dữ liệu luôn được cập nhật kịp thời.

Nguồn dữ liệu mà hệ thống khai thác bao gồm: **giá cổ phiếu theo từng phiên giao dịch, khối lượng khớp lệnh, chỉ số thị trường (VN-Index, HNX-Index, UPCoM-Index), dữ liệu lịch sử giá, cũng như các thông tin bổ trợ khác** như tin tức kinh tế – tài chính, báo cáo doanh nghiệp, hay biến động dòng tiền. Các thông tin này có thể được lấy từ **API chính thức của Sở Giao dịch Chứng khoán TP.HCM (HOSE), Sở Giao dịch Chứng khoán Hà Nội (HNX), hệ thống UPCoM**, hoặc từ các cổng dữ liệu và trang web tài chính uy tín trong nước và quốc tế. Trong trường hợp không có API trực tiếp, các agent crawler sẽ được triển khai để tự động trích xuất dữ liệu từ website theo chu kỳ định trước.

Một đặc điểm quan trọng của hệ thống là khả năng **cập nhật dữ liệu theo thời gian thực hoặc theo từng phiên giao dịch**. Điều này giúp hệ thống duy trì độ chính xác cao, cung cấp thông tin kịp thời cho các agent phân tích và nhà đầu tư. Đồng thời, cơ chế xử lý lỗi và kiểm chứng dữ liệu (data validation) cũng được tích hợp, đảm bảo rằng dữ liệu bị thiếu, sai lệch hoặc trùng lặp sẽ được phát hiện và loại bỏ trước khi đưa vào quá trình phân tích.

Nhờ sự hoạt động tự động, liên tục và có tổ chức của các tác tử thu thập dữ liệu, hệ thống không chỉ tiết kiệm đáng kể công sức thủ công mà còn nâng cao độ tin cậy của thông tin. Đây chính là bước khởi đầu quan trọng, đặt nền móng cho toàn bộ quá trình phân tích, dự báo và ra quyết định trong hệ thống Multi-Agent phân tích chứng khoán Việt Nam.

**1.2 Phân tích dữ liệu và khuyến nghị đầu tư**  
Sau khi dữ liệu thị trường được các tác tử thu thập và lưu trữ, nhiệm vụ tiếp theo thuộc về **các tác tử phân tích (analysis agents)**. Nhóm agent này đóng vai trò trung tâm trong hệ thống, có nhiệm vụ xử lý và khai thác dữ liệu để tạo ra những thông tin có giá trị, phục vụ cho quá trình ra quyết định đầu tư.

Cụ thể, các agent phân tích thực hiện việc **tính toán và áp dụng các chỉ báo kỹ thuật (technical indicators)** phổ biến như:

* **Moving Average (MA):** đường trung bình động giúp xác định xu hướng giá trong ngắn hạn và dài hạn.
* **Relative Strength Index (RSI):** chỉ số sức mạnh tương đối để đánh giá tình trạng quá mua hoặc quá bán của cổ phiếu.
* **Moving Average Convergence Divergence (MACD):** công cụ phân tích động lượng, xác định tín hiệu mua/bán dựa trên sự giao cắt của các đường trung bình.
* **Bollinger Bands:** dải biến động cho thấy mức độ dao động giá và khả năng đảo chiều.

Bên cạnh phân tích kỹ thuật, hệ thống có thể mở rộng để tích hợp **các yếu tố phân tích cơ bản (fundamental analysis)** như chỉ số tài chính của doanh nghiệp, kết quả kinh doanh, hoặc tin tức vĩ mô ảnh hưởng đến thị trường. Điều này giúp cho kết quả khuyến nghị đầu tư trở nên toàn diện và bám sát thực tế hơn.

Dựa trên các kết quả phân tích và **tiêu chí định sẵn (rule-based decision)**, hệ thống sẽ tự động đưa ra **khuyến nghị đầu tư** cho từng mã cổ phiếu với các trạng thái như: **mua (buy), bán (sell) hoặc nắm giữ (hold)**. Ngoài ra, một điểm mạnh quan trọng là hệ thống có khả năng tích hợp **các mô hình học máy và trí tuệ nhân tạo** (như học có giám sát, học tăng cường – reinforcement learning) để dự đoán xu hướng giá trong tương lai. Nhờ vậy, độ chính xác của khuyến nghị được cải thiện đáng kể và có khả năng thích ứng linh hoạt với sự thay đổi liên tục của thị trường.

Một ưu điểm khác của hệ thống Multi-Agent là khả năng **tổng hợp, so sánh và đánh giá nhiều mã cổ phiếu cùng lúc**. Các agent có thể phối hợp để phân loại danh mục cổ phiếu theo ngành nghề, vốn hóa hoặc mức độ rủi ro, từ đó đề xuất chiến lược đầu tư tối ưu cho từng nhóm nhà đầu tư (ngắn hạn, trung hạn hoặc dài hạn). Người dùng cuối sẽ nhận được báo cáo phân tích trực quan, bao gồm bảng biểu, biểu đồ xu hướng và khuyến nghị cụ thể, hỗ trợ họ đưa ra quyết định nhanh chóng và chính xác hơn.

Như vậy, thông qua sự kết hợp giữa **các kỹ thuật phân tích tài chính truyền thống** và **các phương pháp hiện đại dựa trên trí tuệ nhân tạo**, hệ thống không chỉ giúp giảm thiểu yếu tố chủ quan trong quá trình phân tích đầu tư mà còn nâng cao hiệu quả, mang lại giá trị thiết thực cho nhà đầu tư trong bối cảnh thị trường chứng khoán Việt Nam ngày càng phức tạp và biến động.

**1.3 Giao tiếp linh hoạt giữa các agent**

Một trong những đặc trưng cốt lõi của hệ thống Multi-Agent là khả năng **trao đổi thông tin và phối hợp hành động** giữa các tác tử. Trong hệ thống phân tích chứng khoán, các agent không hoạt động đơn lẻ mà tương tác liên tục thông qua các cơ chế truyền thông đa dạng như **message queue (RabbitMQ, Kafka), giao tiếp socket (TCP/UDP), hoặc các phương thức gọi thủ tục từ xa (Remote Procedure Call – RPC)**. Nhờ vậy, việc truyền tải dữ liệu diễn ra nhanh chóng, ổn định và có thể mở rộng theo nhu cầu thực tế.

Ví dụ, sau khi **agent thu thập dữ liệu** hoàn tất nhiệm vụ của mình, dữ liệu thô sẽ ngay lập tức được gửi sang **agent phân tích** để xử lý và trích xuất thông tin hữu ích. Tiếp đó, kết quả phân tích được chuyển đến **agent giao diện người dùng** để hiển thị trực quan hoặc sang **agent lưu trữ** để quản lý dữ liệu lâu dài. Với cơ chế giao tiếp phân tán này, hệ thống có thể vận hành theo kiểu **pipeline** hoặc **song song**, giúp tăng hiệu năng và giảm độ trễ trong xử lý.

Một ưu điểm quan trọng khác là **tính linh hoạt và khả năng mở rộng**. Khi có nhu cầu, hệ thống có thể dễ dàng tích hợp thêm các agent mới mà không ảnh hưởng đến cấu trúc tổng thể. Chẳng hạn, có thể bổ sung **agent cảnh báo rủi ro** để gửi thông báo khi thị trường biến động mạnh, hoặc **agent giám sát theo thời gian thực** nhằm theo dõi những mã cổ phiếu trọng điểm. Tất cả được thực hiện thông qua các giao thức giao tiếp chuẩn hóa, đảm bảo tính tương thích và khả năng cộng tác giữa các thành phần.

**1.4 Lưu trữ và truy xuất dữ liệu hiệu quả**

Dữ liệu trong hệ thống chứng khoán bao gồm hai nhóm chính: **dữ liệu thô** (giá, khối lượng, tin tức, chỉ số) và **dữ liệu phân tích** (kết quả tính toán chỉ báo, khuyến nghị đầu tư, mô hình dự đoán). Để quản lý hiệu quả khối lượng lớn thông tin này, hệ thống cần một **giải pháp lưu trữ linh hoạt, an toàn và có khả năng mở rộng**.

Tùy thuộc vào kiến trúc triển khai, có thể lựa chọn các hệ quản trị cơ sở dữ liệu như **SQLite** (nhẹ, phù hợp môi trường thử nghiệm), **MySQL/PostgreSQL** (quan hệ, mạnh về truy vấn phức tạp), hoặc **MongoDB** (phi quan hệ, tối ưu cho dữ liệu phi cấu trúc và tốc độ truy cập cao). Dữ liệu được lưu trữ theo các cấu trúc được thiết kế sẵn, cho phép **truy vấn nhanh** theo các tiêu chí như mã cổ phiếu, khoảng thời gian, hoặc loại chỉ báo. Nhờ vậy, cả người dùng lẫn các agent khác trong hệ thống đều có thể khai thác dữ liệu một cách thuận tiện.

Ngoài khả năng truy xuất nhanh, hệ thống lưu trữ còn phải đáp ứng các yêu cầu quan trọng khác như:

* **Bảo mật dữ liệu**: áp dụng cơ chế phân quyền truy cập, mã hóa dữ liệu nhạy cảm.
* **Đảm bảo toàn vẹn**: ngăn chặn lỗi ghi trùng lặp hoặc sai lệch dữ liệu trong quá trình cập nhật.
* **Khả năng mở rộng**: hỗ trợ tăng dung lượng lưu trữ hoặc phân tán dữ liệu trên nhiều máy chủ khi hệ thống phát triển.

Nhờ vậy, cơ sở dữ liệu không chỉ đóng vai trò là nơi lưu giữ thông tin mà còn trở thành **nền tảng cho việc phân tích và ra quyết định chính xác**, góp phần nâng cao hiệu quả tổng thể của hệ thống

**1.5 Giao diện người dùng trực quan**

Bên cạnh lớp backend với các agent thu thập và phân tích dữ liệu, hệ thống cần một **giao diện người dùng trực quan (User Interface – UI)** để truyền tải thông tin đến nhà đầu tư một cách sinh động và dễ hiểu. Giao diện web được xem là lựa chọn tối ưu vì khả năng tiếp cận cao, không yêu cầu cài đặt phức tạp và có thể sử dụng trực tiếp qua trình duyệt.

Trong quá trình phát triển, có thể áp dụng các framework mạnh mẽ như **Flask**, **Dash** hoặc đặc biệt là **Streamlit** – một công cụ đơn giản nhưng hiệu quả để xây dựng các dashboard phân tích tài chính. Nhờ Streamlit, hệ thống có thể hiển thị **bảng giá, biểu đồ kỹ thuật theo thời gian thực, kết quả phân tích chỉ báo, và các khuyến nghị đầu tư** một cách trực quan, hỗ trợ nhà đầu tư nắm bắt thông tin nhanh chóng.

Người dùng còn có khả năng **tương tác linh hoạt** với hệ thống, chẳng hạn: lựa chọn mã cổ phiếu muốn theo dõi, tùy chỉnh loại chỉ báo kỹ thuật cần phân tích (MA, RSI, MACD,…), thiết lập khoảng thời gian quan sát, hoặc tra cứu lại lịch sử khuyến nghị trước đó. Với những tính năng này, giao diện không chỉ đóng vai trò hiển thị mà còn là **cầu nối quan trọng giữa con người và hệ thống Multi-Agent**, đảm bảo trải nghiệm sử dụng mượt mà, thân thiện và hỗ trợ quá trình ra quyết định đầu tư một cách hiệu quả.

**2.Yêu cầu phi chức năng**

**2.1**

Hệ thống Multi-Agent Viet Nam Stock cần đáp ứng đầy đủ các yêu cầu phi chức năng nhằm đảm bảo khả năng vận hành ổn định, hiệu quả và an toàn trong môi trường thực tế. Trước hết, **tính mở rộng** là yếu tố quan trọng, giúp hệ thống thích ứng linh hoạt với sự gia tăng về số lượng người dùng, dữ liệu giao dịch và tác vụ xử lý. Hệ thống cần hỗ trợ mở rộng theo chiều ngang, cho phép phân tán các tác tử (agent) trên nhiều máy chủ hoặc nút xử lý độc lập, đồng thời cần có cơ chế tự động giám sát và điều chỉnh quy mô phù hợp với tải hệ thống. Việc sử dụng các công nghệ hiện đại như container (Docker) và hệ thống điều phối (Kubernetes) sẽ giúp tối ưu hóa khả năng mở rộng của hệ thống.

2.2

Về **bảo mật**, hệ thống phải đảm bảo an toàn thông tin ở mức cao, đặc biệt là trong môi trường tài chính có yêu cầu nghiêm ngặt về dữ liệu. Tất cả dữ liệu truyền tải giữa các tác tử và hệ thống bên ngoài phải được mã hóa bằng các giao thức an toàn như HTTPS hoặc TLS. Ngoài ra, cần áp dụng cơ chế xác thực và phân quyền chặt chẽ (Role-Based Access Control) cho người dùng và các tác tử. Các tác tử chỉ được phép giao tiếp với các thành phần đã được xác thực, nhằm ngăn chặn các hành vi tấn công từ bên ngoài hoặc từ nội bộ hệ thống. Đồng thời, toàn bộ hoạt động của hệ thống cần được ghi log đầy đủ để phục vụ công tác kiểm tra, giám sát và xử lý sự cố khi cần thiết.

2.3

Tiếp theo, **hiệu năng** là một yêu cầu then chốt, đặc biệt trong bối cảnh thị trường chứng khoán biến động liên tục và yêu cầu phản hồi nhanh. Hệ thống phải đảm bảo độ trễ thấp trong quá trình xử lý và truyền thông giữa các tác tử, hỗ trợ xử lý song song và tối ưu hóa việc sử dụng tài nguyên như CPU, bộ nhớ và băng thông. Việc áp dụng các cơ chế như cache dữ liệu, xử lý bất đồng bộ và đa luồng sẽ giúp nâng cao hiệu năng tổng thể, đảm bảo các tác tử có thể đưa ra phân tích và khuyến nghị đầu tư trong thời gian gần như thực.

2.4

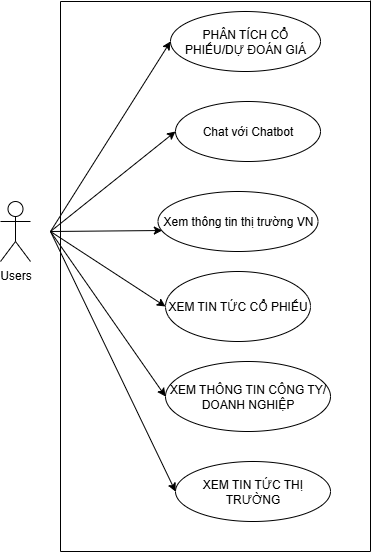
Cuối cùng, **khả năng tích hợp** cũng là một tiêu chí quan trọng đối với hệ thống. Multi-Agent Viet Nam Stock cần dễ dàng kết nối với các hệ thống dữ liệu chứng khoán như HOSE, HNX, và các nền tảng giao dịch như SSI, VNDIRECT thông qua các API chuẩn. Các tác tử phải được thiết kế theo kiến trúc mở, hỗ trợ giao tiếp thông qua các chuẩn công nghiệp như RESTful API, WebSocket hoặc gRPC. Đồng thời, hệ thống cần hỗ trợ trao đổi dữ liệu với các hệ thống bên ngoài thông qua các định dạng phổ biến như JSON, XML hoặc CSV, cũng như có khả năng tích hợp với các mô-đun trung gian hoặc bus tích hợp dữ liệu để đảm bảo luồng thông tin giữa các thành phần được quản lý hiệu quả và nhất quán.

**3. Các đối tượng và tác nhân (Actors)**

Trong hệ thống MultiAgentVnStock, các đối tượng và tác nhân (actors) được xác định dựa trên vai trò của chúng trong việc thu thập, phân tích, lưu trữ và tương tác với dữ liệu chứng khoán. Hệ thống được thiết kế theo kiến trúc multi-agent, nơi các agent hoạt động độc lập nhưng phối hợp chặt chẽ thông qua main\_agent.py để cung cấp phân tích toàn diện. Dưới đây là danh sách chi tiết các actors chính, bao gồm người dùng cuối cùng, hệ thống lưu trữ, và đầy đủ 6 AI agents chuyên biệt được triển khai trong repository (dựa trên các file như price\_predictor.py, investment\_expert.py, v.v.). Mỗi agent được mô tả với chức năng chính, cách thức hoạt động, và vai trò trong hệ thống tổng thể để đảm bảo tính toàn diện và khả năng mở rộng.

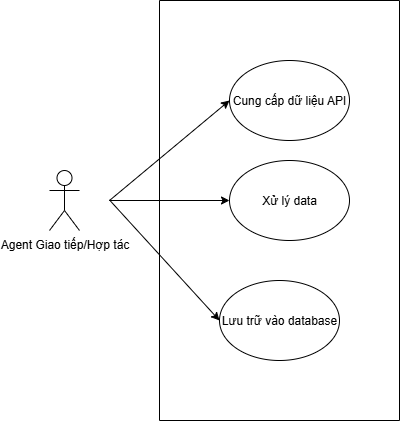
**3.1 Người dùng cuối**

* **Người dùng cuối (End user)**: Đây là đối tượng chính sử dụng hệ thống, bao gồm các nhà đầu tư cá nhân, chuyên gia tài chính hoặc người dùng quan tâm đến thị trường chứng khoán Việt Nam (với hơn 37 mã cổ phiếu được hỗ trợ như VCB, VIC, MSN). Người dùng tương tác qua giao diện Streamlit frontend (với 6 tabs chuyên nghiệp) hoặc FastAPI backend (qua các endpoints như /analyze hoặc /query). Họ cung cấp đầu vào như mã cổ phiếu, thời gian đầu tư (ngắn hạn, trung hạn, dài hạn), mức độ rủi ro (0-100), và số tiền đầu tư (từ 1 triệu đến 10 tỷ VND). Hệ thống trả về phân tích cá nhân hóa, khuyến nghị BUY/SELL/HOLD, dự đoán giá, và đánh giá rủi ro. Người dùng có thể sử dụng chế độ offline fallback khi hết quota API Gemini, đảm bảo tính liên tục. Vai trò của người dùng là khởi xướng các yêu cầu và nhận kết quả, với các tính năng như gợi ý câu hỏi mẫu trong tab AI Chatbot để nâng cao trải nghiệm.



Hình 1: Actor end users sử dụng hệ thống

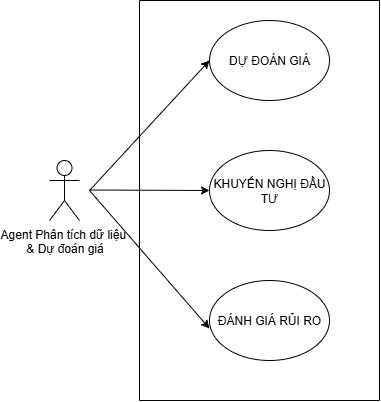
* 1. **Agent thu thập dữ liệu**
* **Agent thu thập dữ liệu (Data collection agent)**: Bao gồm hai agent chuyên biệt từ 6 agent chính, tập trung vào việc thu thập dữ liệu thời gian thực từ các nguồn đáng tin cậy.
  + **TickerNews Agent**: Chịu trách nhiệm crawl tin tức liên quan đến cổ phiếu cụ thể từ các nguồn như CafeF, VietStock, và VCI. Agent này sử dụng CrewAI (Serper.dev) để tích hợp multi-source crawling, thực hiện phân tích sentiment (tâm lý thị trường) và ưu tiên highlighting tin quan trọng. Nó hoạt động bằng cách nhận mã cổ phiếu từ người dùng, thu thập dữ liệu qua vn\_stock\_api.py và crewai\_collector.py, sau đó lọc và phân loại tin tức để tránh thông tin nhiễu. Tính năng đặc biệt bao gồm real-time updates và integration với Gemini AI để nâng cao độ chính xác.
  + **MarketNews Agent**: Tập trung vào tin tức thị trường tổng thể, với lọc dựa trên rủi ro (risk-based filtering). Agent này crawl từ các nguồn underground như F319, F247, FB Groups, cũng như nguồn chính thức như CafeF, VnEconomy, và DanTri. Nó sử dụng risk\_based\_news.py để điều chỉnh nội dung theo hồ sơ rủi ro của người dùng (thận trọng, cân bằng, mạo hiểm), và tự động phân loại tin tức theo chủ đề. Vai trò chính là cung cấp dữ liệu thị trường rộng lớn, hỗ trợ tab Thị trường VN và Tin tức thị trường trong giao diện.



Hình 2: Actor Collect Data

**3.3 Agent phân tích dữ liệu, dự đoán giá (Data analysis agent and Price Predict Agent)**:

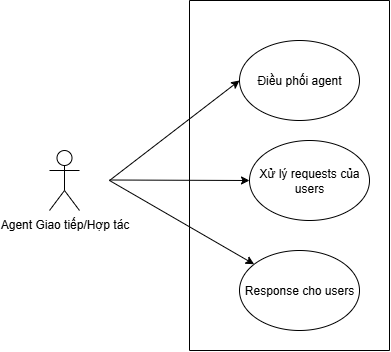
* **Agent phân tích dữ liệu (Data analysis agent)**: Bao gồm ba agent chuyên sâu từ 6 agent, sử dụng các thuật toán nâng cao như LSTM Neural Network và phân tích tài chính để xử lý dữ liệu thu thập được.
  + **PricePredictor Agent**: Sử dụng LSTM Neural Network (từ lstm\_price\_predictor.py) kết hợp với technical analysis để dự đoán giá cổ phiếu trên multi-timeframe (1 ngày đến 1 năm). Agent này đào tạo mô hình thời gian thực với 18 mô hình ML khác nhau (từ basic LSTM đến Transformer), tính toán confidence scoring cho độ tin cậy dự đoán, và tích hợp với Gemini AI để nâng cao kết quả. Vai trò là cung cấp dự báo giá chính xác, hỗ trợ tab Phân tích cổ phiếu và endpoints
  + **InvestmentExpert Agent**: Thực hiện phân tích cơ bản (fundamental analysis) với real financial ratios (P/E, P/B, EPS, Dividend yield) và đưa ra khuyến nghị BUY/SELL/HOLD dựa trên thời gian đầu tư và mức độ rủi ro của người dùng. Agent này sử dụng investment\_expert.py để cân bằng giữa technical và fundamental analysis, tính toán position sizing và risk management (stop-loss, take-profit). Nó đóng góp vào phân tích toàn diện, giúp người dùng ra quyết định đầu tư thông minh.
  + **RiskExpert Agent**: Đánh giá rủi ro với các metrics nâng cao như VaR (Value at Risk), Beta, Sharpe ratio, và Max Drawdown, sử dụng risk\_expert.py. Agent này cung cấp lời khuyên AI-based dựa trên dữ liệu thực tế, điều chỉnh theo rủi ro cá nhân (0-30: thận trọng với blue-chip stocks; 71-100: mạo hiểm với growth stocks). Vai trò là giảm thiểu rủi ro, hỗ trợ tab Đánh giá rủi ro và endpoints.



Hình 3: Actor Phân tích dữ liệu & Dự đoán giá

**3.4** **Agent giao tiếp/hợp tác (Communication/collaboration agent)**:

* **Agent giao tiếp/hợp tác (Communication/collaboration agent)**: Đây là agent phối hợp tổng thể, đại diện bởi StockInfo Agent từ 6 agent chính, kết hợp với **main\_agent.py** làm orchestrator.
  + **StockInfo Agent**: Cung cấp thông tin chi tiết về cổ phiếu và công ty, hiển thị metrics thời gian thực và interactive charts (sử dụng Plotly từ stock\_info.py). Agent này thu thập dữ liệu từ CrewAI và vn\_stock\_api.py, sau đó phối hợp với các agent khác để tích hợp thông tin (ví dụ: kết hợp tin tức từ TickerNews với dự đoán từ PricePredictor). Vai trò là làm cầu nối giao tiếp giữa các agent, đảm bảo dữ liệu được trao đổi mượt mà qua cơ chế CrewAI, và hỗ trợ tab Thông tin công ty với enhanced display (professional styling, color-coded metrics). Ngoài ra, nó xử lý giao tiếp với Gemini AI chatbot cho phản hồi ngôn ngữ tự nhiên, và fallback offline khi cần.



Hình 4: Actor Giao tiếp

**3.5** **Hệ thống lưu trữ (Storage system)**

* **Mô tả tổng quan**: Hệ thống lưu trữ trong MultiAgentVnStock đóng vai trò là backend hỗ trợ lưu trữ, truy xuất và bảo vệ dữ liệu cho toàn bộ hệ thống. Đây không phải là một agent riêng biệt mà là một hệ thống tích hợp, kết nối với các agent và module chức năng thông qua các file trong thư mục `src/data/` (bao gồm `vn\_stock\_api.py`, `crewai\_collector.py`, và `company\_search\_api.py`). Hệ thống này lưu trữ dữ liệu thời gian thực từ các nguồn như VNStock API, yfinance, bao gồm lịch sử giá cổ phiếu, các chỉ số tài chính (financial ratios), tin tức thị trường, và các mô hình ML đã đào tạo.
* **Thành phần và công nghệ sử dụng:**

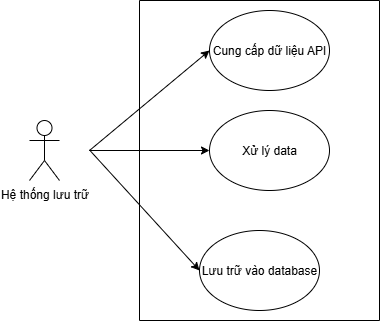
- Cơ sở dữ liệu: Hiện tại hệ thống đã sử dụng cơ sở dữ liệu quan hệ SQLite với file `duong\_trading.db` để lưu trữ và quản lý dữ liệu một cách có cấu trúc, thay cho phương pháp lưu file rời hoặc lưu trữ tạm trong bộ nhớ như trước đây.

- Xử lý dữ liệu: Sử dụng các thư viện Python như `pandas` và `numpy` để xử lý, chuẩn hóa, phân tích dữ liệu chứng khoán. Dữ liệu được trích xuất, xử lý và lưu vào các bảng trong SQLite.

- Tích hợp với các module khác: Hệ thống lưu trữ phối hợp với các file tiện ích như `error\_handler.py` và `performance\_monitor.py` để giám sát, ghi nhận lỗi và theo dõi hiệu suất truy xuất dữ liệu.

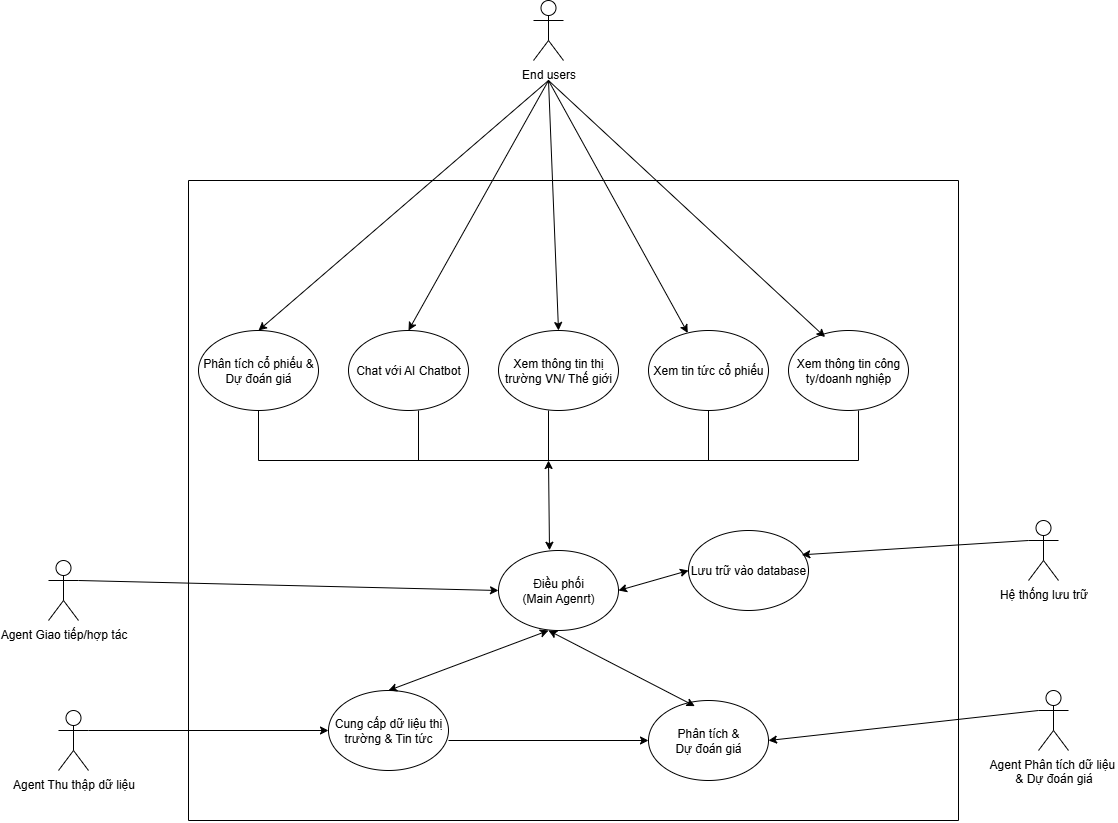
- Bảo mật: Tích hợp với `security\_manager.py` nhằm bảo vệ thông tin nhạy cảm, đặc biệt là hạn chế rò rỉ API key khi tương tác với nguồn dữ liệu bên ngoài.

- Fallback: Trong trường hợp các agent như CrewAI thất bại khi thu thập dữ liệu, hệ thống có thể sử dụng dữ liệu static backup hoặc dữ liệu từ các file notebook trong thư mục `deep-learning/`



Hình 5: Actor Hệ thống lưu trữ

**4. Biểu đồ ca sử dụng (Use Case Diagram)**

****

**Hình 6**

**Chương IV. Thiết kế hệ thống**

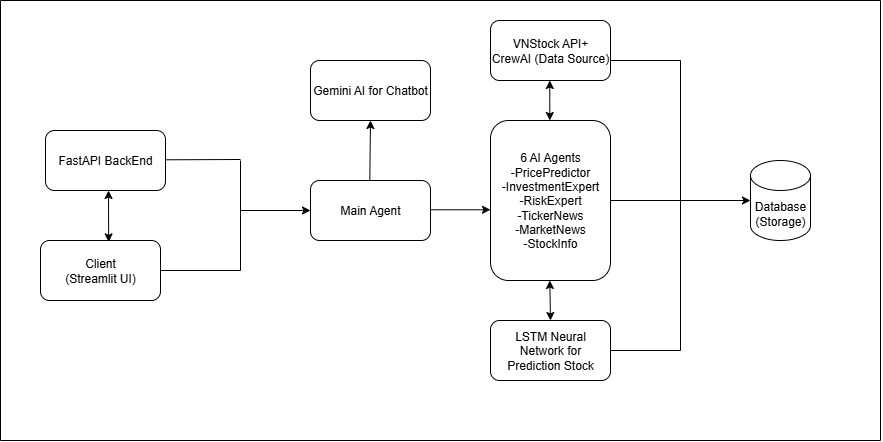
1.Kiến trúc tổng thể hệ thống

1.1 Mô hình kiến trúc

Hệ thống được thiết kế dựa trên mô hình **client-server** kết hợp **multi-agent** và **microservices**.

* **Client-server**: Giao diện người dùng (Streamlit Frontend) giao tiếp với backend FastAPI để xử lý yêu cầu và trả về kết quả.
* **Multi-agent**: 6 AI Agents (PricePredictor, InvestmentExpert, RiskExpert, TickerNews, MarketNews, StockInfo) hoạt động độc lập nhưng phối hợp thông qua MainAgent để phân tích và đưa ra quyết định.
* **Microservices**: Các thành phần như VNStock API, CrewAI, và Gemini AI được triển khai dưới dạng dịch vụ độc lập, đảm bảo tính mở rộng và bảo trì.

**1.2 Sơ đồ kiến trúc tổng thể**

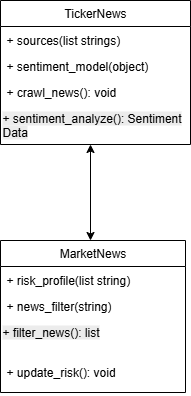


Hình 7

**2. Thiết kế chi tiết các agent**

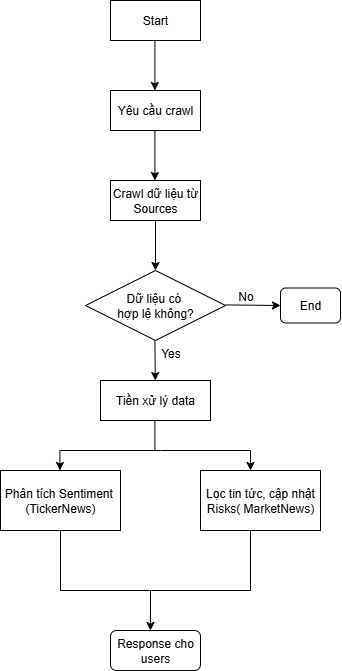
**2.1. Agent thu thập dữ liệu (Stock\_info, Ticker\_news)**

* **Chức năng**: Thu thập dữ liệu tin tức từ các nguồn như CafeF, VietStock, VCI, và dữ liệu thị trường từ VNStock API/CrewAI.
* **Quy trình hoạt động**:
  1. Kết nối đến nguồn dữ liệu (multi-source crawling).
  2. Lọc và xử lý dữ liệu (sentiment analysis, risk-based filtering).
  3. Lưu trữ dữ liệu tạm thời và chuyển đến các agent khác.
* **Sơ đồ lớp**:



Hình 8

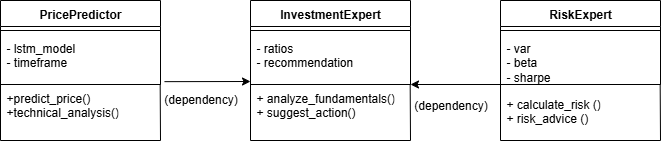
* 1. Class TickerNews: Thuộc tính (sources, sentiment\_model), Phương thức (crawl\_news(), analyze\_sentiment()).
  2. Class MarketNews: Thuộc tính (risk\_profile, news\_filter), Phương thức (filter\_news(), update\_risk()).
* **Sơ đồ trình tự**: Yêu cầu crawl → Xử lý dữ liệu → Gửi kết quả đến MainAgent.



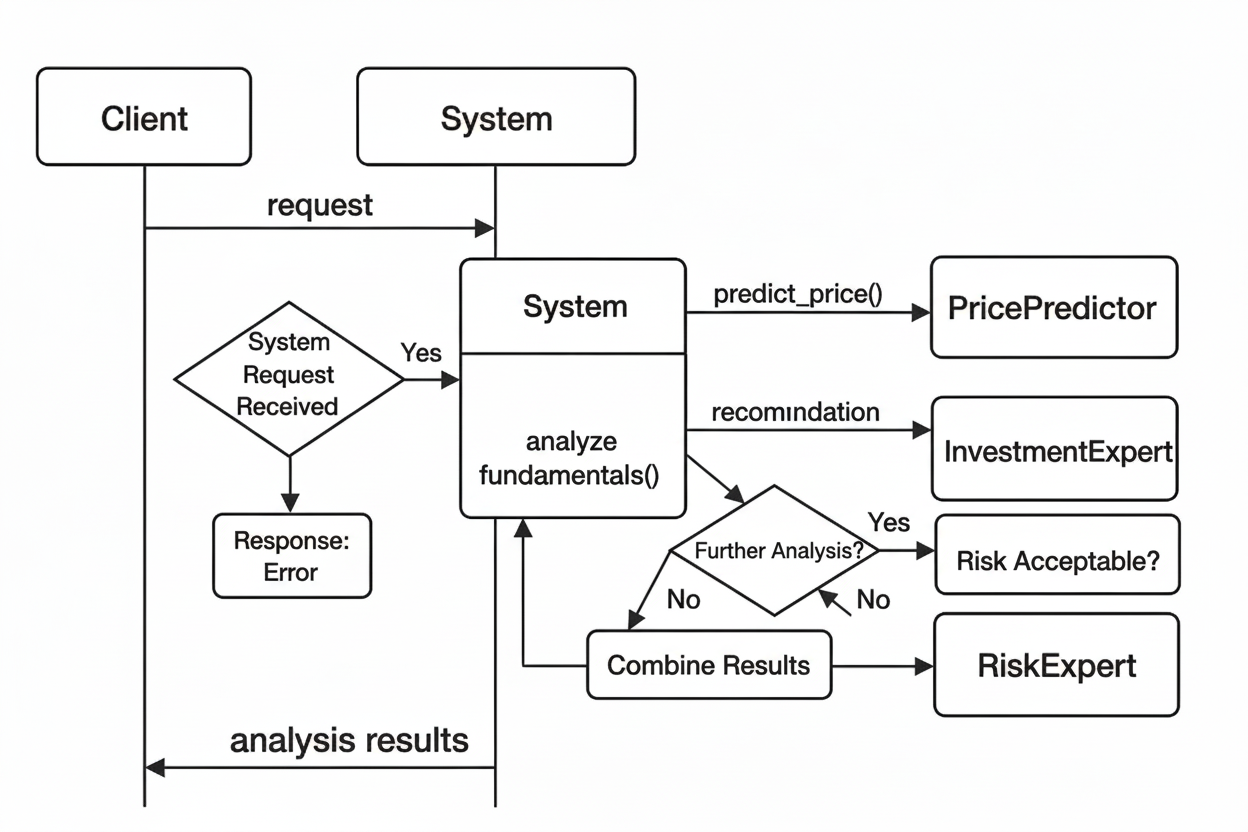
Hình 9

2.2. Agent Phân tích dữ liệu (PricePredictor, InvestmentExpert, RiskExpert)

* **Chức năng**:
  + PricePredictor: Dự đoán giá cổ phiếu bằng LSTM Neural Network và phân tích kỹ thuật.
  + InvestmentExpert: Phân tích cơ bản (P/E, P/B) và đưa ra khuyến nghị (BUY/SELL/HOLD).
  + RiskExpert: Đánh giá rủi ro (VaR, Beta, Sharpe ratio).
* **Giải thuật sử dụng**:
  + PricePredictor: LSTM (Long Short-Term Memory) với multi-timeframe.
  + InvestmentExpert: Quy tắc dựa trên financial ratios kết hợp AI-enhanced logic.
  + RiskExpert: Tính toán thống kê (VaR, Beta) với mô hình Monte Carlo.
* **Sơ đồ lớp**:
  + Class PricePredictor: Thuộc tính (lstm\_model, timeframe), Phương thức (predict\_price(), technical\_analysis()).
  + Class InvestmentExpert: Thuộc tính (ratios, recommendation), Phương thức (analyze\_fundamentals(), suggest\_action()).
  + Class RiskExpert: Thuộc tính (var, beta, sharpe), Phương thức (calculate\_risk(), risk\_advice()).



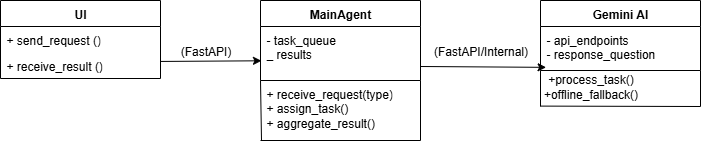
Hình 10

* **Sơ đồ trình tự**: Nhận dữ liệu → Xử lý bằng giải thuật → Trả kết quả phân tích.
* 

Hình 11

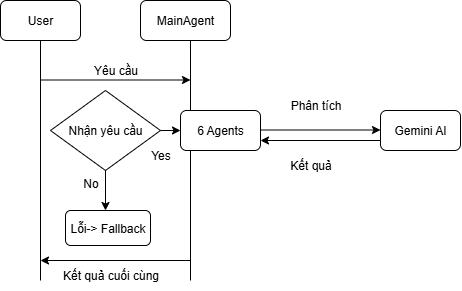
##### 2.3. Agent hợp tác/giao tiếp (MainAgent, Gemini AI)

* **Cơ chế trao đổi thông tin**: Sử dụng FastAPI endpoints để truyền dữ liệu giữa các agent và giao diện người dùng. MainAgent đóng vai trò orchestrator, điều phối tác vụ giữa các agent.
* **Phối hợp tác vụ**:
  + MainAgent nhận yêu cầu từ User(UI), phân bổ nhiệm vụ cho các agent phù hợp.
  + Gemini AI hỗ trợ chatbot với câu trả lời thông minh và khả năng offline fallback, giao tiếp qua API hoặc chế độ nội bộ.



Hình 12

* **Sơ đồ trình tự**: Yêu cầu từ User → MainAgent phân bổ → Các agent xử lý → Tổng hợp và trả về UI.



Hình 13

#### 3. Thiết kế cơ sở dữ liệu

##### 3.1. Mô hình dữ liệu

Dữ liệu được tổ chức theo **mô hình quan hệ** (Relational Model), một lựa chọn lý tưởng cho các hệ thống cần độ chính xác và khả năng truy vấn cao. Việc sử dụng cơ sở dữ liệu quan hệ mang lại nhiều lợi thế:

* **Tính nhất quán và toàn vẹn dữ liệu:** Đảm bảo rằng mọi thông tin đều chính xác và được đồng bộ.
* **Truy vấn nhanh chóng:** Hỗ trợ các câu lệnh SQL phức tạp, giúp truy xuất thông tin từ nhiều bảng một cách hiệu quả.
* **Dễ dàng tích hợp:** Tương thích tốt với các nguồn dữ liệu thời gian thực như **VNStock** và các tác vụ được thực hiện bởi các agent trong **CrewAI**.

##### 3.2. Sơ đồ quan hệ thực thể (ERD)

* **StockData**: (id, symbol, price, volume, date)
* **NewsData**: (id, source, title, content, sentiment\_score, date, stock\_id)
* **AnalysisResults**: (id, stock\_id, prediction, recommendation, risk\_score, timestamp)

##### 3.3. Các bảng dữ liệu chính

* **StockData**: Lưu trữ giá cổ phiếu và khối lượng giao dịch theo thời gian.
* **NewsData**: Lưu trữ tin tức với phân tích cảm xúc (sentiment).
* **AnalysisResults**: Lưu trữ kết quả phân tích từ các agent (dự đoán, khuyến nghị, rủi ro).

#### 4. Thiết kế giao diện

##### 4.1. Mô tả giao diện người dùng

* **Tab 1: Phân tích cổ phiếu**: Hiển thị biểu đồ giá, khuyến nghị đầu tư, và đánh giá rủi ro.
* **Tab 2: AI Chatbot**: Giao diện trò chuyện với Gemini AI, hỗ trợ offline fallback.
* **Tab 3: Thị trường VN**: Bảng xếp hạng cổ phiếu, tin tức thị trường.
* **Tab 4: Tin tức cổ phiếu**: Danh sách tin tức với phân tích cảm xúc.
* **Tab 5: Thông tin công ty**: Dữ liệu tài chính và biểu đồ tương tác.
* **Tab 6: Tin tức thị trường**: Tin tức lọc theo rủi ro.

##### 4.2. Sơ đồ luồng giao diện

* **Trang chủ (index.html)**:
  + Sidebar: Nhập API key, chọn tab.
  + Nội dung chính: Chuyển hướng đến 6 tabs dựa trên lựa chọn.
* **Tab 1 (Phân tích cổ phiếu)**:
  + Nhập symbol → Hiển thị biểu đồ giá → Hiển thị phân tích (BUY/SELL/HOLD, rủi ro).
* **Tab 2 (AI Chatbot)**:
  + Nhập câu hỏi → Trả lời từ Gemini hoặc offline → Hiển thị gợi ý câu hỏi.
* **Tab 3 (Thị trường VN)**:
  + Tải dữ liệu real-time → Hiển thị top movers, tin tức.
* **Tab 4 (Tin tức cổ phiếu)**:
  + Lọc tin tức theo symbol → Hiển thị danh sách với sentiment.
* **Tab 5 (Thông tin công ty)**:
  + Chọn công ty → Hiển thị metrics, biểu đồ Plotly.
* **Tab 6 (Tin tức thị trường)**:
  + Lọc theo rủi ro → Hiển thị tin tức phân loại.