

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÀNH PHỐ HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

MÔN: CÔNG NGHỆ PHẦN MỀM



**Đề tài: Xây dựng hệ thống giám sát thị trường chứng
khoán cộng đồng sử dụng Node.js**

- GVHD: *ThS. Bùi Mạnh Toàn*

- Sinh viên:

+ *Trần Phúc Thành* *MSSV: 2280602964*

+ *Huỳnh Ngọc Tiên* *MSSV: 2280603202*

+ *Nguyễn Xuân Vinh* *MSSV: 2280603702*

+ *Đinh Nhật Quỳnh Giang* *MSSV: 2280600754*

Tp Hồ Chí Minh, ngày 22 tháng 06 năm 2025

LỜI CẢM ƠN

Nhóm sinh viên chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy Bùi Mạnh Toàn - giảng viên hướng dẫn môn Công nghệ Phần mềm - đã tận tình chỉ dẫn, định hướng và hỗ trợ nhóm trong suốt quá trình thực hiện đồ án “Xây dựng hệ thống giám sát thị trường chứng khoán cộng đồng sử dụng Node.js”.

Nhóm cũng xin bày tỏ lòng biết ơn đến Khoa Công nghệ Thông tin – Trường Đại học Công nghệ Thành phố Hồ Chí Minh (HUTECH) đã tạo điều kiện học tập thuận lợi, cung cấp tài liệu, môi trường học tập năng động để nhóm có cơ hội rèn luyện kỹ năng, phối hợp nhóm và áp dụng kiến thức vào thực tiễn.

Nhóm xin ghi nhận và trân trọng cảm ơn sự đoàn kết, tinh thần trách nhiệm và sự phối hợp chặt chẽ của từng thành viên. Dù còn những hạn chế nhất định, đây là cơ hội quý báu để nhóm học hỏi, tích lũy kinh nghiệm và nâng cao năng lực phát triển phần mềm trong thực tế.

Một lần nữa, nhóm xin chân thành cảm ơn sự đồng hành và hỗ trợ của Thầy và Nhà trường.

Đại diện nhóm thực hiện

Nhóm F4JS

LỜI CAM ĐOAN

Nhóm xin cam kết rằng toàn bộ báo cáo của đồ án “Xây dựng hệ thống giám sát thị trường chứng khoán cộng đồng sử dụng Node.js” là kết quả của quá trình làm việc nghiêm túc, nghiên cứu và thực hiện miệt mài của tất cả các thành viên trong nhóm. Tất cả nội dung trong báo cáo đều do nhóm tự tìm hiểu, phân tích và phát triển, không sao chép từ bất kỳ nguồn nào mà không ghi rõ trích dẫn. Các số liệu, hình ảnh minh họa và kết quả trong báo cáo đều được thực hiện thực tế, đảm bảo tính chính xác, phản ánh trung thực toàn bộ quá trình thực hiện đồ án.

Nhóm xin hoàn toàn chịu trách nhiệm trước Thầy hướng dẫn và Nhà trường về tính trung thực, khách quan cũng như sự chính xác của toàn bộ nội dung báo cáo này, đồng thời cam đoan sẽ nghiêm túc tiếp thu mọi ý kiến đóng góp để hoàn thiện và nâng cao chất lượng sản phẩm hơn nữa.

Đại diện nhóm thực hiện

Nhóm F4JS

LỜI MỞ ĐẦU

Trong bối cảnh công nghệ phát triển không ngừng, các giải pháp phần mềm hiện đại đang đóng vai trò quan trọng trong nhiều lĩnh vực, đặc biệt là lĩnh vực tài chính – chứng khoán. Việc giám sát và phân tích dữ liệu thị trường chứng khoán một cách nhanh chóng, chính xác sẽ giúp nhà đầu tư đưa ra quyết định kịp thời và nâng cao hiệu quả đầu tư.

Nhận thức được tầm quan trọng đó, nhóm sinh viên chúng em đã lựa chọn đề tài “Xây dựng hệ thống giám sát thị trường chứng khoán cộng đồng sử dụng Node.js” để vận dụng kiến thức đã học, kết hợp với công nghệ web hiện đại, nhằm xây dựng một hệ thống giám sát, tra cứu, phân tích giá cổ phiếu và thảo luận cộng đồng theo thời gian thực.

Thông qua đồ án này, nhóm không chỉ củng cố lý thuyết mà còn rèn luyện kỹ năng làm việc nhóm, giải quyết vấn đề và nâng cao năng lực triển khai giải pháp công nghệ thực tế. Nhóm xin trân trọng trình bày báo cáo và kính mong nhận được những ý kiến đóng góp quý báu từ Thầy và Nhà trường để hoàn thiện sản phẩm tốt hơn.

MỤC LỤC

Chương 1: TỔNG QUAN VỀ CÔNG NGHỆ PHẦN MỀM.....	1
1.1. Giới thiệu chung về công nghệ phần mềm.....	1
1.1.1. Khái niệm công nghệ phần mềm.....	1
1.1.2. Mục tiêu và vai trò của công nghệ phần mềm.....	1
1.1.3. Lịch sử phát triển ngành công nghệ phần mềm.....	2
1.2. Quy trình và kỹ thuật phát triển phần mềm.....	2
1.2.1. Quy trình xây dựng phần mềm.....	2
1.2.2. Xác định yêu cầu phần mềm.....	5
1.2.3. Quy trình xác định yêu cầu phần mềm.....	5
1.2.4. Mô hình hóa hệ thống phần mềm.....	6
1.2.5. Thiết kế kiến trúc phần mềm.....	6
1.2.6. Thiết kế giao diện người dùng.....	7
1.2.7. Kiểm thử phần mềm.....	7
Chương 2: TỔNG QUAN VỀ ĐỀ TÀI.....	9
2.1. Bối cảnh và lý do chọn đề tài.....	9
2.2. Mục tiêu của đề tài.....	10
2.3. Phạm vi và giới hạn của đề tài.....	11
2.4. Công nghệ sử dụng trong đề tài.....	12
Chương 3: PHÂN TÍCH YÊU CẦU.....	13
3.1. Đặc tả yêu cầu chức năng.....	13
3.2. Yêu cầu chức năng.....	15
3.2.1. Theo dõi thị trường chứng khoán thời gian thực.....	15
3.2.2. Biểu đồ chứng khoán nâng cao.....	15
3.2.3. Quản lý danh mục đầu tư cá nhân.....	15
3.2.4. Giao dịch cổ phiếu.....	15
3.2.5. Tin tức và doanh nghiệp.....	16
3.2.6. Cộng đồng và tương tác.....	16
3.2.7. Tìm kiếm và lọc cổ phiếu.....	16
3.2.8. Hệ thống người dùng và quản trị.....	16
3.2.9. Hệ thống kỹ thuật.....	16
3.3. Yêu cầu phi chức năng.....	16

3.3.2. Cập nhật dữ liệu thời gian thực.....	17
3.3.3. Khả năng mở rộng và chịu tải cao.....	17
3.3.4. Bảo mật và an toàn thông tin.....	18
3.4. Phân tích vai trò Node.js trong đáp ứng yêu cầu đề tài.....	18
Chương 4: MÔ HÌNH HÓA HỆ THỐNG.....	21
4.1. User Stories.....	21
4.1.1. Quản lý tài khoản.....	21
4.1.2. Quản lý danh mục đầu tư cá nhân.....	22
4.1.3. Thực hiện giao dịch.....	22
4.1.4. Tương tác cộng đồng.....	23
4.1.5. Tìm kiếm cổ phiếu.....	24
4.1.6. Truy cập dữ liệu thị trường.....	25
4.1.7. Đồng bộ dữ liệu thị trường.....	26
4.2. Biểu đồ Use Case.....	26
4.2.1. Tổng quan chức năng.....	26
4.2.2. Quản lý danh mục đầu tư cá nhân.....	27
4.2.3. Đồng bộ dữ liệu thị trường.....	27
4.2.4. Thực hiện giao dịch.....	28
4.2.5. Quản lý tài khoản.....	28
4.2.6. Tìm kiếm cổ phiếu.....	29
4.2.7. Truy cập dữ liệu thị trường.....	30
4.2.8. Tương tác giữa các nhà đầu tư.....	30
4.3. Sơ đồ Class Diagram.....	31
4.4. Biểu đồ hoạt động (Activity Diagram).....	33
4.4.2. Quản lý danh mục đầu tư.....	34
4.4.3. Đồng bộ dữ liệu thị trường.....	36
4.4.4. Giao dịch cổ phiếu.....	37
4.4.5. Tìm kiếm cổ phiếu.....	38
4.4.6. Truy cập dữ liệu thị trường.....	39
4.4.7. Tương tác cộng đồng.....	40
4.5. Biểu đồ trình tự (Sequence Diagram).....	41
4.5.1. Đăng ký và Đăng nhập.....	41
4.5.2. Quản lý danh mục.....	42
4.5.3. Đồng bộ dữ liệu thị trường.....	44

4.5.4. Giao dịch mua/bán cổ phiếu.....	44
4.5.5. Tìm kiếm và lọc cổ phiếu.....	45
4.5.6. Truy cập dữ liệu thị trường thời gian thực.....	46
4.5.7. Tương tác cộng đồng.....	47
Chương 5: THIẾT KẾ HỆ THỐNG.....	48
5.1. Kiến trúc hệ thống tổng quát.....	48
5.1.1. Các thành phần chính trong kiến trúc.....	48
5.1.2. Luồng hoạt động của hệ thống.....	51
5.1.3. Ưu điểm của kiến trúc.....	53
5.2. Thiết kế cơ sở dữ liệu (PortgreSQL).....	53
5.2.1. Mục tiêu thiết kế.....	54
5.2.2. Mô hình khái niệm.....	55
5.2.3. Mô hình quan hệ dữ liệu (ERD).....	56
5.2.4. Chuẩn hóa dữ liệu.....	60
5.2.5. Định hướng triển khai vật lý.....	61
5.2.6. Kết luận.....	63
5.3. Thiết kế giao diện người dùng.....	64
5.3.1. Giao diện Trang chủ.....	64
5.3.2. Giao diện Đăng nhập người dùng.....	66
5.3.3. Giao diện Thông tin người dùng.....	67
5.3.4. Giao diện Bảng giá cổ phiếu.....	69
5.3.5. Giao diện đặt lệnh mua bán cổ phiếu.....	70
5.3.6. Giao diện biểu đồ kỹ thuật cổ phiếu.....	71
5.3.7. Giao diện Tin tức và doanh nghiệp.....	73
5.3.8. Giao diện Quản lý danh mục đầu tư.....	75
5.3.9. Giao diện Cộng đồng thảo luận cổ phiếu.....	77
Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	79
6.1. Đánh giá chung về hệ thống đã triển khai.....	79
6.3. Hướng phát triển tương lai của hệ thống.....	82

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Ký hiệu viết tắt	Chữ viết đầy đủ
1	2FA	Two-Factor Authentication
2	ACB	Asia Commercial Bank
3	AI	Artificial Intelligence
4	API	Application Programming Interface
5	ATO/ATC	At The Open / At The Close
6	BI	Business Intelligence
7	CCV	Card Code Verification
8	CI/CD	Continuous Integration / Continuous Deployment
9	CSV	Comma-Separated Values
10	CSRF	Cross-Site Request Forgery
11	DDS	Design Document Specification
12	DDoS	Distributed Denial of Service
13	EMA	Exponential Moving Average
14	EPS	Earnings Per Share

15	ERD	Entity Relationship Diagram
16	FPT	FPT Corporation
17	HNX	Hanoi Stock Exchange
18	HOSE	Ho Chi Minh Stock Exchange
19	HPG	Hòa Phát Group
20	IDE	Integrated Development Environment
21	IoT	Internet of Things
22	I/O	Input/Output
23	JWT	JSON Web Token
24	LO	Limit Order
25	MA/EMA	Moving Average / Exponential Moving Average
26	MACD	Moving Average Convergence Divergence
27	MTL	Market To Limit
28	NASDAQ	National Association of Securities Dealers Automated Quotations

29	NLP	Natural Language Processing
30	NYSE	New York Stock Exchange
31	OHLC	Open High Low Close
32	OTP	One-Time Password
33	P/E	Price to Earnings
34	PaaS	Platform as a Service
35	PM2	Process Manager 2
36	RESTful API	Representational State Transfer Application Programming Interface
37	ROE	Return On Equity
38	RSI	Relative Strength Index
39	SaaS	Software as a Service
40	SDLC	Software Development Life Cycle
41	SRS	Software Requirement Specification
42	SQL	Structured Query Language
43	SSL/TLS	Secure Sockets Layer / Transport Layer Security

44	UI	User Interface
45	UML	Unified Modeling Language
46	UPCOM	Unlisted Public Company Market
47	UX	User Experience
48	VNM	Vinamilk
49	WebSocket	WebSocket Protocol
50	XSS	Cross-Site Scripting

DANH MỤC HÌNH ẢNH

Hình 1.1: Quy trình phát triển phần mềm.....	3
Hình 4.1: Use case tổng quát chức năng.....	27
Hình 4.2: Use case chức năng quản lý danh mục đầu tư cá nhân.....	27
Hình 4.3: Use case chức năng đồng bộ dữ liệu thị trường.....	28
Hình 4.4: Use case chức năng thực hiện giao dịch.....	28
Hình 4.5: Use case chức năng quản lý tài khoản.....	29
Hình 4.6: Use case chức năng tìm kiếm cổ phiếu.....	29
Hình 4.7: Use case chức năng truy cập dữ liệu thị trường.....	30
Hình 4.8: Use case chức năng tương tác giữa các nhà đầu tư.....	30
Hình 4.10: Activity diagram đăng nhập/đăng kí.....	34
Hình 4.11: Activity diagram quản lý cổ phiếu yêu thích.....	35
Hình 4.12: Activity diagram thống kê danh mục đầu tư.....	36
Hình 4.13: Activity diagram đồng bộ dữ liệu thị trường.....	37
Hình 4.14: Activity diagram giao dịch cổ phiếu.....	38
Hình 4.15: Activity diagram tìm kiếm cổ phiếu.....	39
Hình 4.16: Activity diagram chức năng truy cập dữ liệu thị trường.....	40
Hình 4.17: Activity diagram chức năng tương tác cộng đồng.....	41
Hình 4.18: Sequence diagram đăng kí/đăng nhập.....	42
Hình 4.19: Sequence diagram quản lý cổ phiếu yêu thích.....	43
Hình 4.20: Sequence diagram thống kê danh mục đầu tư.....	43
Hình 4.21: Sequence diagram đồng bộ dữ liệu thị trường.....	44
Hình 4.22: Sequence diagram giao dịch cổ phiếu.....	45
Hình 4.23: Sequence diagram tìm kiếm cổ phiếu.....	46
Hình 4.24: Sequence diagram chức năng truy cập dữ liệu thị trường.....	47
Hình 4.25: Sequence diagram chức năng tương tác cộng đồng.....	47
Hình 5.1: Mô hình ERD.....	57
Hình 5.2: Giao diện trang chủ.....	65
Hình 5.3: Giao diện đăng nhập.....	67
Hình 5.4: Giao diện thông tin người dùng.....	68
Hình 5.5: Giao diện bảng giá cổ phiếu.....	69
Hình 5.6: Giao diện mua bán cổ phiếu.....	71
Hình 5.7: Giao diện biểu đồ cổ phiếu 1.....	72
Hình 5.8: Giao diện biểu đồ cổ phiếu 2.....	73

Hình 5.9: Giao diện tin tức doanh nghiệp 1.....	74
Hình 5.10: Giao diện tin tức doanh nghiệp 2.....	75
Hình 5.11: Giao diện quản lý danh mục đầu tư 1.....	76
Hình 5.12: Giao diện quản lý danh mục đầu tư 2.....	76
Hình 5.13: Giao diện cộng đồng tương tác.....	78

DANH MỤC BẢNG

Bảng 4.1: User story quản lý tài khoản.....	20
Bảng 4.2: User story quản lý danh mục đầu tư cá nhân.....	21
Bảng 4.3: User story thực hiện giao dịch.....	22
Bảng 4.4: User story chức năng tương tác cộng đồng.....	23
Bảng 4.5: User story chức năng tìm kiếm cổ phiếu.....	24
Bảng 4.6: User story chức năng truy cập thị trường.....	24
Bảng 4.7: User story chức năng đồng bộ dữ liệu thị trường.....	25

Chương 1: TỔNG QUAN VỀ CÔNG NGHỆ PHẦN MỀM

1.1. Giới thiệu chung về công nghệ phần mềm

1.1.1. Khái niệm công nghệ phần mềm

Công nghệ phần mềm (Software Technology) là tập hợp các công cụ, kỹ thuật, nền tảng và mô hình được sử dụng để hỗ trợ quy trình phát triển phần mềm từ giai đoạn phân tích, thiết kế, triển khai cho đến bảo trì. Đây là phần cốt lõi trong việc đảm bảo phần mềm có thể hoạt động một cách ổn định, hiệu quả, dễ bảo trì và dễ dàng mở rộng trong môi trường sản xuất thực tế.

Khác với kỹ thuật phần mềm (Software Engineering) vốn tập trung vào quy trình và phương pháp luận phát triển, công nghệ phần mềm nhấn mạnh vào việc áp dụng các công nghệ hiện đại như:

- Ngôn ngữ lập trình hiện đại (Python, JavaScript, Go)
- Framework và thư viện hỗ trợ (React, Django, Spring Boot)
- Công nghệ triển khai (Docker, Kubernetes, Serverless)
- Công cụ tự động hóa (CI/CD pipelines)
- Nền tảng AI, học máy, xử lý dữ liệu lớn (TensorFlow, Apache Spark)

Công nghệ phần mềm đóng vai trò quan trọng trong việc thu hẹp khoảng cách giữa ý tưởng và sản phẩm thực tế giúp hiện thực hóa các yêu cầu người dùng một cách linh hoạt và hiệu quả hơn bao giờ hết.

1.1.2. Mục tiêu và vai trò của công nghệ phần mềm

Công nghệ phần mềm hiện đại có 4 mục tiêu cốt lõi:

- **Nâng cao hiệu suất phát triển:** Thông qua các công cụ phát triển tích hợp (IDE), framework hỗ trợ mã nguồn mở và nền tảng phát triển nhanh (Rapid Application Development).
- **Đảm bảo chất lượng phần mềm:** Bằng cách áp dụng kiểm thử tự động (unit test, integration test, load test) và công cụ CI/CD giúp triển khai nhanh, ổn định.
- **Tối ưu hóa khả năng mở rộng và bảo trì:** Sử dụng các kiến trúc như Microservices, Clean Architecture hoặc Domain-Driven Design (DDD) giúp phần mềm dễ bảo trì và phát triển lâu dài.

- **Thích ứng nhanh với thay đổi:** Với các công cụ như Docker, Kubernetes, Cloud-native, phần mềm có thể dễ dàng thay đổi môi trường, mở rộng quy mô hoặc cập nhật tính năng.

Vai trò của công nghệ phần mềm:

- Là cầu nối giữa ý tưởng, giải pháp và hiện thực hóa sản phẩm phần mềm.
- Tạo điều kiện để các nhóm phát triển phần mềm hoạt động hiệu quả, linh hoạt và thích nghi với thay đổi liên tục.
- Mang đến giải pháp chuyển đổi số toàn diện cho doanh nghiệp, góp phần giảm thiểu rủi ro phát sinh và tối ưu hoá chi phí hoạt động.

1.1.3. Lịch sử phát triển ngành công nghệ phần mềm

Sự phát triển của công nghệ phần mềm gắn liền với sự tiến hóa của phần cứng, internet và nhu cầu ngày càng cao về phần mềm chất lượng:

- **1960s – 1970s:** Xuất hiện cuộc khủng hoảng phần mềm “Software Crisis” khi các hệ thống phần mềm trở nên phức tạp hơn, đòi hỏi những cách tiếp cận mới. Các ngôn ngữ như COBOL, FORTRAN ra đời và lần đầu tiên mô hình Waterfall được đề xuất.
- **1980s – 1990s:** Phát triển mạnh mẽ các công cụ CASE, khái niệm lập trình hướng đối tượng, UML và kỹ thuật thiết kế phần mềm có cấu trúc. Các hệ thống phần mềm quy mô lớn bắt đầu hình thành.
- **2000s:** Những năm 2000 đánh dấu sự xuất hiện của các công cụ IDE mạnh mẽ như Eclipse, Visual Studio cùng với sự phổ biến của phương pháp Agile, đã làm thay đổi cách phát triển phần mềm từ quy trình cứng nhắc sang mô hình linh hoạt, thích ứng nhanh với thay đổi.
- **2010 đến nay:** Sự bùng nổ của công nghệ đám mây, Microservices, DevOps, học máy, IoT và các mô hình phần mềm như SaaS, PaaS, Serverless. Công nghệ phần mềm hiện đại không chỉ phục vụ phát triển

mà còn tích hợp mạnh mẽ vào vận hành, giám sát và nâng cao trải nghiệm người dùng.

1.2. Quy trình và kỹ thuật phát triển phần mềm

1.2.1. Quy trình xây dựng phần mềm

Quy trình xây dựng phần mềm – hay còn gọi là vòng đời phát triển phần mềm (Software Development Life Cycle – SDLC) – là một chuỗi các giai đoạn được tổ

chức và thực hiện tuân tự nhằm đảm bảo việc phát triển một hệ thống phần mềm hiệu quả, đúng yêu cầu và dễ bảo trì. Đây là một khung chuẩn giúp các nhóm phát triển phần mềm có thể lập kế hoạch, tổ chức, kiểm soát và thực thi toàn bộ quá trình sản xuất phần mềm.



Hình 1.1: Quy trình phát triển phần mềm

SDLC giúp giảm thiểu rủi ro kỹ thuật, đảm bảo chất lượng sản phẩm và tối ưu chi phí, thời gian phát triển. Một quy trình SDLC tiêu chuẩn thường bao gồm 6 đến 7 giai đoạn chính được mô tả như sau:

- **Giai đoạn 1: Lập kế hoạch và phân tích yêu cầu (Planning and Requirement Analysis)**

Lập kế hoạch là bước khởi đầu quan trọng trong bất kỳ dự án phần mềm nào. Trong giai đoạn này, các nhà phát triển tiến hành thu thập thông tin và phân tích yêu cầu từ nhiều nguồn như: khách hàng, phòng kinh doanh hoặc các khảo sát thị trường. Đây chính là cơ sở quan trọng để hình thành cấu trúc tổng quan và định hướng phát triển cho toàn bộ dự án.

Chất lượng của phần mềm trong tương lai phụ thuộc nhiều vào chất lượng của công tác lập kế hoạch. Do đó, trong giai đoạn này, nhóm phát triển sẽ phác thảo dự án sơ bộ dựa trên các thông tin thu thập được.

- **Giai đoạn 2: Xác định yêu cầu (Defining Requirements)**

Tại giai đoạn này, tất cả yêu cầu chức năng và phi chức năng của phần mềm sẽ được xác định một cách cụ thể và chi tiết. Các yêu cầu được xác minh và phê duyệt bởi khách hàng, chuyên gia thị trường và các bên liên quan.

Toàn bộ yêu cầu sẽ được tổng hợp trong tài liệu Đặc tả yêu cầu phần mềm (SRS – Software Requirement Specification). Đây là cơ sở tham chiếu xuyên suốt vòng đời dự án, định hướng cho thiết kế, lập trình và kiểm thử phần mềm.

- Giai đoạn 3: Thiết kế kiến trúc (Designing Architecture)

Dựa trên tài liệu SRS, các kiến trúc sư phần mềm sẽ đề xuất các mô hình kiến trúc phù hợp để hiện thực hóa hệ thống. Các phương án thiết kế này được mô tả chi tiết trong Tài liệu đặc tả thiết kế (DDS – Design Document Specification).

DDS sẽ được đánh giá bởi các nhà phân tích và bên liên quan. Sau quá trình đánh giá toàn diện, phương án thiết kế tối ưu nhất – xét trên các yếu tố kỹ thuật, kinh tế và vận hành – sẽ được chọn làm cơ sở để phát triển phần mềm.

- Giai đoạn 4: Phát triển sản phẩm (Developing Product)

Đây là giai đoạn hiện thực hóa phần mềm. Các lập trình viên tiến hành viết mã nguồn dựa trên thiết kế đã được thông qua trong DDS. Các công cụ như trình biên dịch, trình thông dịch và gỡ lỗi sẽ được sử dụng trong giai đoạn này.

Lập trình viên có thể sử dụng nhiều ngôn ngữ khác nhau tùy vào đặc điểm dự án như: C/C++, Java, Python. Điều quan trọng là phải tuân thủ đúng quy chuẩn và các hướng dẫn kỹ thuật đã được xác định trước đó.

- Giai đoạn 5: Kiểm thử và tích hợp (Product Testing and Integration)

Sau khi phát triển xong sản phẩm, phần mềm sẽ được kiểm thử kỹ lưỡng để đảm bảo hoạt động đúng như mong đợi. Dù ở mỗi giai đoạn đều có các bước kiểm thử tối thiểu nhưng ở giai đoạn này, toàn bộ hệ thống sẽ được kiểm tra toàn diện để phát hiện lỗi, sửa chữa và kiểm thử lại.

Ngoài ra, giai đoạn này còn bao gồm:

- Tài liệu hóa: giúp người dùng hiểu cách sử dụng và bảo trì phần mềm.
- Đào tạo và hỗ trợ kỹ thuật: đảm bảo người dùng cuối có thể sử dụng phần mềm hiệu quả.

- Giai đoạn 6: Triển khai và bảo trì (Deployment and Maintenance)

Khi phần mềm đã vượt qua kiểm thử, nó sẽ được triển khai chính thức theo kế hoạch của tổ chức. Sản phẩm thường được phát hành theo từng giai đoạn và được đánh giá trong môi trường thực tế.

Sau khi thu thập phản hồi từ người dùng, phần mềm có thể được cập nhật hoặc cải tiến. Tuy nhiên, triển khai không phải là điểm kết thúc – phần mềm cần được duy trì,

giám sát và cập nhật định kỳ để đảm bảo hoạt động ổn định, bảo mật và phù hợp với các nhu cầu thay đổi trong tương lai.

1.2.2. Xác định yêu cầu phần mềm

Xác định yêu cầu phần mềm là bước đầu tiên và đóng vai trò then chốt trong quy trình phát triển một hệ thống phần mềm. Mục tiêu của giai đoạn này là làm rõ những chức năng và đặc điểm mà người dùng mong đợi, từ đó làm cơ sở để lập kế hoạch về thời gian, chi phí cũng như lựa chọn công nghệ phù hợp nhằm đáp ứng các yêu cầu đó một cách hiệu quả.

Quá trình xác định yêu cầu bao gồm nhiều hoạt động: thu thập, lọc lại, phân tích, đặc tả và xác minh yêu cầu. Việc xác định yêu cầu sai sót hoặc không đủ dẫn đến việc thiếu tính năng, lãng phí nguồn lực hoặc thất bại dự án.

Có hai loại yêu cầu cơ bản:

- **Yêu cầu chức năng (Functional Requirements):** Mô tả các hành vi cụ thể mà phần mềm phải thực hiện ví dụ như chức năng đăng ký người dùng, xử lý thanh toán, tìm kiếm sản phẩm.
- **Yêu cầu phi chức năng (Non-functional Requirements):** Mô tả các ràng buộc về hiệu suất, bảo mật, khả năng tương thích, khả năng mở rộng, độ tin cậy.

Kết quả của bước này thường được tài liệu hóa dưới dạng SRS (Software Requirement Specification) sẽ là cơ sở quan trọng cho toàn bộ quy trình phát triển.

1.2.3. Quy trình xác định yêu cầu phần mềm

Quy trình xác định yêu cầu bao gồm nhiều bước liên quan mật thiết và có tính lặp lại như sau:

a) Thu thập yêu cầu (Elicitation):

- Sử dụng kỹ thuật như phỏng vấn, khảo sát, quan sát, phân tích tài liệu hiện tại, workshop tương tác.
- Phải làm việc với nhiều đối tượng: người dùng cuối, khách hàng, quản lý dự án.

b) Phân tích và làm rõ yêu cầu (Analysis):

- Kiểm tra mâu thuẫn giữa các yêu cầu.
- Phân loại yêu cầu theo mức độ ưu tiên.
- Xác định yêu cầu nào khả thi về mặt kỹ thuật.

c) Đặc tả yêu cầu (Specification):

- Mô tả chính xác yêu cầu dưới dạng tài liệu.
- Sử dụng biểu đồ (Use Case, DFD), bảng mô tả hoặc mô hình hóa UML.

d) Xác minh và xác nhận (Validation & Verification):

- Kiểm tra yêu cầu với người dùng, bảo đảm đúng và đủ.
- Có thể dùng prototype để trình diễn yêu cầu.

1.2.4. Mô hình hóa hệ thống phần mềm

Mô hình hóa hệ thống phần mềm là quá trình trừu tượng hóa hệ thống thực thành các biểu đồ trực quan, giúp các nhà phát triển, người phân tích và các bên liên quan hiểu được cách phần mềm hoạt động. Đây là bước cầu nối giữa yêu cầu người dùng và thiết kế kỹ thuật.

Mục đích của mô hình hóa:

- Làm rõ các thành phần chính và luồng xử lý trong hệ thống.
- Hỗ trợ giao tiếp giữa nhóm phát triển và khách hàng.
- Phát hiện sớm thiếu sót hoặc mâu thuẫn trong yêu cầu.

Các loại mô hình phổ biến:

- **Use Case Diagram:** Mô tả các chức năng chính của hệ thống và tương tác với tác nhân.
- **Class Diagram:** Mô tả cấu trúc lớp và mối quan hệ giữa các đối tượng.
- **Activity Diagram:** Diễn tả luồng công việc (workflow) trong hệ thống.
- **Data Flow Diagram (DFD):** Biểu diễn dòng dữ liệu giữa các tiến trình.
- **State Diagram:** Mô tả trạng thái và chuyển tiếp trạng thái của đối tượng.

Ngôn ngữ mô hình hóa chuẩn hiện nay là UML (Unified Modeling Language), được sử dụng phổ biến trong thiết kế và phân tích hệ thống.

1.2.5. Thiết kế kiến trúc phần mềm

Thiết kế kiến trúc phần mềm đề cập đến việc xác định cấu trúc tổng thể và tổ chức các thành phần phần mềm trong hệ thống. Đây là bước thiết kế cấp cao, nơi xác định cách các module giao tiếp với nhau, chia lớp hệ thống và định nghĩa các chuẩn kết nối nội bộ.

Một kiến trúc phần mềm tốt cần đạt được:

- Tính mô đun hóa và phân chia trách nhiệm rõ ràng.

- Tính mở rộng và khả năng bảo trì cao.
- Tính tương thích với nền tảng triển khai (Web, Desktop, Cloud).
- Tính bảo mật và hiệu năng.

Một số mẫu kiến trúc tiêu biểu:

- **Monolithic Architecture:** Tất cả các thành phần tích hợp trong một khối duy nhất.
- **Layered Architecture (3-tier):** Gồm giao diện người dùng, lớp xử lý nghiệp vụ và lớp dữ liệu.
- **Microservices:** Chia thành nhiều dịch vụ nhỏ độc lập, dễ mở rộng, triển khai nhanh.
- **Event-driven Architecture:** Hệ thống xử lý dựa trên sự kiện phát sinh.

Thiết kế kiến trúc tốt đóng vai trò quan trọng trong việc đảm bảo tính bền vững, khả năng bảo trì và phát triển lâu dài của hệ thống.

1.2.6. Thiết kế giao diện người dùng

Giao diện người dùng (UI – User Interface) là yếu tố trực tiếp ảnh hưởng đến trải nghiệm của người dùng với phần mềm. Một giao diện tốt cần dễ sử dụng, trực quan, phản hồi nhanh và phù hợp với hành vi người dùng.

Nguyên tắc thiết kế UI hiệu quả:

- **Khả năng sử dụng (Usability):** Giao diện đơn giản, dễ hiểu, dễ thao tác.
- **Tính nhất quán (Consistency):** Các thành phần giống nhau phải có cùng hành vi.
- **Phản hồi rõ ràng (Feedback):** Cung cấp phản hồi cho người dùng khi họ thực hiện thao tác.
- **Tối ưu trên nhiều thiết bị (Responsive Design):** Hiển thị tốt trên desktop, tablet và smartphone.

Quá trình thiết kế UI thường bao gồm:

- Phác thảo khung dây (Wireframe)
- Tạo nguyên mẫu tương tác (Prototype)
- Kiểm thử giao diện với người dùng thật (Usability Testing)

1.2.7. Kiểm thử phần mềm

Kiểm thử phần mềm là quá trình nhằm phát hiện lỗi, xác minh rằng phần mềm hoạt động đúng như yêu cầu và đảm bảo rằng nó sẵn sàng đưa vào môi trường thực tế. Đây là giai đoạn quan trọng để đánh giá chất lượng sản phẩm trước khi triển khai.

Các mức độ kiểm thử:

- **Unit Testing:** Kiểm tra từng đơn vị mã nhỏ nhất (thường là hàm).
- **Integration Testing:** Kiểm tra tương tác giữa các module.
- **System Testing:** Kiểm tra toàn bộ hệ thống như một khối.
- **Acceptance Testing:** Kiểm tra theo góc nhìn người dùng cuối.

Phương pháp kiểm thử:

- **Kiểm thử hộp trắng:** Dựa trên cấu trúc mã nguồn.
- **Kiểm thử hộp đen:** Dựa trên đầu vào – đầu ra, không quan tâm mã nguồn.
- **Kiểm thử tự động:** Dùng tool (Selenium, JUnit) để kiểm thử nhanh và lặp lại.

Vai trò của kiểm thử:

- Phát hiện lỗi sớm và giảm chi phí sửa chữa.
- Cải thiện độ tin cậy và độ ổn định.
- Tăng niềm tin của người dùng vào hệ thống.

Chương 2: TỔNG QUAN VỀ ĐỀ TÀI

2.1. Bối cảnh và lý do chọn đề tài

Trong những năm gần đây, thị trường chứng khoán Việt Nam đang chứng kiến sự phát triển vượt bậc cả về quy mô, độ sâu thị trường lẫn mức độ tham gia của các nhà đầu tư. Giờ đây thị trường chứng khoán không chỉ giới hạn trong phạm vi các tổ chức tài chính lớn mà ngày càng có nhiều nhà đầu tư cá nhân và đặc biệt là thế hệ trẻ đã và đang tham gia mạnh mẽ vào thị trường này với kỳ vọng tìm kiếm cơ hội sinh lời và gia tăng tài sản. Song song với đó là thị trường này đang cũng thu hút sự quan tâm đặc biệt từ các cơ quan quản lý, doanh nghiệp và giới truyền thông nhờ vai trò ngày càng rõ nét trong việc huy động vốn, điều tiết dòng tiền và thúc đẩy sự minh bạch, hiệu quả của nền kinh tế.

Sự bùng nổ về số lượng tài khoản giao dịch, sự tăng trưởng nhanh chóng của các chỉ số thị trường cùng với sự xuất hiện của nhiều sản phẩm tài chính mới đã cho thấy thị trường chứng khoán không còn là “sân chơi” dành riêng cho một bộ phận nhỏ mà đã và đang trở thành kênh đầu tư phổ biến, đóng vai trò then chốt trong hệ sinh thái tài chính quốc gia. Đặc biệt, trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ trên toàn cầu và lan rộng sang lĩnh vực tài chính, các nền tảng công nghệ hiện đại đã và đang góp phần tái định hình cách thức nhà đầu tư tiếp cận thông tin, phân tích dữ liệu và đưa ra quyết định giao dịch.

Chính trong bối cảnh đó, nhu cầu về các công cụ hỗ trợ đầu tư chứng khoán toàn diện đang trở nên ngày càng cần thiết, đặt nền móng cho việc phát triển những hệ thống thông minh và hiệu quả hơn. Hiện nay, trên thị trường đã xuất hiện nhiều hệ thống và nền tảng hỗ trợ nhà đầu tư trong việc theo dõi thị trường, quản lý danh mục. Dù có nhiều tính năng vượt trội nhưng phần lớn các hệ thống này thường chỉ tập trung vào một số tính năng riêng lẻ thiếu sự kết hợp toàn diện. Có hệ thống cung cấp biểu đồ kỹ thuật chi tiết nhưng lại không hỗ trợ quản lý danh mục đầu tư. Ngược lại, một số nền tảng như FireAnt, TCInvest, SSI iBoard, VnDirect TradingView, CafeF có chức năng gợi ý cổ phiếu hoặc cung cấp tin tức nhưng lại thiếu khả năng hiển thị biểu đồ nến, phân tích kỹ thuật hoặc tương tác giữa người dùng. Sự phân mảnh về chức năng này khiến nhà đầu tư buộc phải sử dụng nhiều công cụ song song, gây bất tiện và thiếu sự đồng bộ.

Xuất phát từ thực tế đó, nhóm đã quyết định lựa chọn và phát triển đề tài “Xây dựng hệ thống giám sát thị trường chứng khoán cộng đồng sử dụng Node.js” với mục tiêu kết hợp nhiều chức năng hữu ích trong cùng một nền tảng. Hệ thống này không chỉ đáp ứng nhu cầu theo dõi giá cổ phiếu, quản lý danh mục, theo dõi tin tức doanh nghiệp hay lọc cổ phiếu mà còn tích hợp điểm nổi bật mang tính khác biệt – chức năng cộng đồng và tương tác người dùng theo thời gian thực.

Cụ thể, hệ thống sẽ bao gồm một khu vực cộng đồng, nơi các nhà đầu tư có thể:

- **Bỏ phiếu đánh giá chiều hướng cổ phiếu:** qua hai tùy chọn Vote Tăng / Vote Giảm cho từng mã cổ phiếu
- **Theo dõi người dùng khác:** quan sát các ý kiến, chiến lược đầu tư của họ
- **Tham gia vào giao diện thảo luận cộng đồng:** như một “group” tương tác, chia sẻ kiến thức và nhận định đầu tư theo thời gian thực.

Tính năng cộng đồng không chỉ mang đến trải nghiệm tương tác đa chiều mà còn mở rộng khả năng học hỏi, trao đổi và tạo ra một hệ sinh thái đầu tư thân thiện, minh bạch, mang tính cộng tác cao – điều còn thiếu ở nhiều nền tảng hiện có.

Chính sự kết hợp giữa các chức năng kỹ thuật chuyên sâu và tính năng tương tác cộng đồng là điểm nhấn độc đáo và là lý do quan trọng thôi thúc nhóm triển khai đề tài này. Hệ thống hướng đến không chỉ là một công cụ đầu tư mà còn là một không gian mở, năng động và hiện đại, phục vụ một cách toàn diện cho nhu cầu đầu tư chứng khoán trong kỷ nguyên số.

2.2. Mục tiêu của đề tài

Đề tài tập trung vào việc thiết kế và phát triển một hệ thống giám sát thị trường chứng khoán cộng đồng theo thời gian thực được tích hợp nhiều chức năng hỗ trợ đầu tư trong cùng một nền tảng thống nhất. Thay vì phân mảnh như các hệ thống hiện nay, đề tài hướng đến việc kết hợp các công cụ kỹ thuật, quản lý danh mục, theo dõi mã cổ phiếu. và tính năng cộng đồng nhằm giúp nhà đầu tư dễ dàng sử dụng công cụ để sử dụng các tính năng cần thiết mà tránh được sự bất tiện khi phải dùng song song nhiều công cụ khác nhau. Từ đó, giảm thiểu sự bất tiện để nhà đầu tư có thể tập trung quan sát để đưa ra các quyết định chính xác, kịp thời và khách quan. Cụ thể, các mục tiêu chính của hệ thống bao gồm:

- **Cung cấp nền tảng theo dõi giá cổ phiếu, chỉ số và khối lượng giao dịch thời gian thực, kèm biểu đồ kỹ thuật và thông tin giá chi tiết:** Hiển thị giá cổ

phiếu, khối lượng giao dịch cùng với các thông tin chi tiết khác và biểu đồ kỹ thuật với các khung thời gian linh hoạt, sử dụng biểu đồ nến để tương tác.

- **Hỗ trợ quản lý danh mục đầu tư cá nhân:** bao gồm chức năng thêm/xoá cổ phiếu yêu thích, tính toán tổng giá trị danh mục và lãi/lỗ hiện tại.
- **Tìm kiếm và lọc cổ phiếu thông minh:** dựa trên các tiêu chí như mã, tên, giá hiện tại, khối lượng, vốn hóa và phân ngành, giúp nhà đầu tư nhanh chóng tiếp cận các mã cổ phiếu phù hợp chiến lược đầu tư.
- **Tích hợp tin tức thị trường:** Hiện thị tin tức theo từng mã cổ phiếu từ các nguồn dữ liệu tài chính uy tín.
- **Phát triển các chức năng cộng đồng và tương tác người dùng:** bao gồm thảo luận về cổ phiếu, đánh giá xu hướng thị trường qua hình thức bỏ phiếu (Vote Tăng/Giảm), theo dõi người dùng khác để từ đó tạo ra một không gian trao đổi kiến thức và nhận định đầu tư theo thời gian thực.
- **Tận dụng sức mạnh của Node.js:** trong việc xử lý bất đồng bộ, kết nối WebSocket và streaming dữ liệu real-time, đảm bảo hiệu năng cao và khả năng mở rộng linh hoạt trong môi trường dữ liệu tài chính biến động liên tục.

2.3. Phạm vi và giới hạn của đề tài

Đề tài được triển khai trong phạm vi mô hình nguyên mẫu, tập trung vào việc xây dựng hệ thống giám sát và hỗ trợ đầu tư với các tính năng cốt lõi sau:

- Dữ liệu thị trường chứng khoán Việt Nam: lựa chọn một số mã cổ phiếu tiêu biểu như VNM, FPT, HPG, ACB và lấy dữ liệu thông qua API công khai (ví dụ: Vnstock, FireAnt, TCBS) hoặc giả lập dữ liệu để phục vụ mô phỏng.
- Hiện thị biểu đồ kỹ thuật: biểu đồ nến (candlestick), khối lượng giao dịch, tích hợp các chỉ báo kỹ thuật như RSI, MACD, MA, EMA, Bollinger Bands theo nhiều khung thời gian.
- Quản lý danh mục đầu tư: cho phép người dùng thêm mã yêu thích, thống kê lãi/lỗ theo thời gian thực.
- Tìm kiếm và lọc cổ phiếu: lọc theo các tiêu chí như tên/mã, giá hiện tại, vốn hóa, khối lượng và ngành nghề.
- Tính năng cộng đồng: hỗ trợ đăng ký, đăng nhập, phân quyền người dùng, vote chiều hướng tăng giảm của cổ phiếu, theo dõi người dùng khác, tham gia giao diện thảo luận cộng đồng.

Song song với đó thì đề tài vẫn có một số giới hạn nhất định do yếu tố thời gian, kỹ thuật và nguồn lực:

- Không thực hiện giao dịch thực tế: Hệ thống không cho phép đặt lệnh mua/bán cổ phiếu vì hệ thống vẫn còn là đề tài môn học đang được phát triển chưa công bố ra thị trường nên không thể liên kết với sàn giao dịch thực tế.
- Chưa tích hợp AI nâng cao: Hệ thống hiện chưa tích hợp các công nghệ AI nâng cao để hỗ trợ phân tích và ra quyết định.
- Phân tích tin tức ở mức sơ bộ: Chỉ sử dụng dữ liệu có sẵn từ API, chưa có cơ chế crawling dữ liệu ngoài phạm vi hoặc xử lý nội dung ngôn ngữ tự nhiên (NLP) ở mức sâu.

2.4. Công nghệ sử dụng trong đề tài

Đề tài chọn **Node.js** làm nền tảng chính cho Backend do những ưu điểm vượt trội như:

- Hiệu năng cao: Node.js sử dụng cơ chế event-driven và non-blocking I/O, rất phù hợp để xử lý các luồng dữ liệu chứng khoán theo thời gian thực mà không gây nghẽn luồng.
- Cộng đồng phát triển mạnh mẽ và hệ sinh thái thư viện phong phú: Với hệ sinh thái rộng lớn, Node.js hỗ trợ đầy đủ các công cụ cần thiết như thư viện vẽ biểu đồ, gửi email (Nodemailer), gọi API (Axios), phân tích dữ liệu.
- Khả năng tích hợp và mở rộng tốt: Node.js dễ dàng kết nối với PostgreSQL, WebSocket và RESTful API – những thành phần quan trọng để xây dựng hệ thống theo mô hình microservices.
- Ứng dụng thực tiễn cao: Nhiều nền tảng FinTech hiện đại cũng đang sử dụng Node.js để xây dựng dashboard giám sát, xử lý cảnh báo và quản lý dòng dữ liệu tài chính quy mô lớn (Savani, 2025).

Ngoài ra, hệ thống còn sử dụng:

- **PostgreSQL:** Lưu trữ lịch sử giá, tín hiệu giao dịch và dữ liệu người dùng.
- **Express.js:** Xây dựng API RESTful phục vụ frontend.
- **Plotly.js hoặc TradingView Charting Library:** Hiển thị biểu đồ nến tương tác và trực quan hóa tín hiệu kỹ thuật.
- **React.js và Socket.IO Client:** Xây dựng giao diện người dùng động, đồng bộ dữ liệu realtime với server.
- **Axios và Nodemailer:** Gọi API tài chính và gửi cảnh báo qua email.

Chương 3: PHÂN TÍCH YÊU CẦU

3.1. Đặc tả yêu cầu chức năng

Hệ thống giám sát thị trường chứng khoán theo thời gian thực được xây dựng nhằm đáp ứng nhu cầu ngày càng cao của các nhà đầu tư cá nhân và tổ chức trong việc theo dõi biến động giá, phân tích kỹ thuật, thực hiện giao dịch và quản lý danh mục đầu tư một cách trực quan, chính xác và hiệu quả. Hệ thống này đóng vai trò như một nền tảng trung tâm tích hợp nhiều công cụ hỗ trợ đầu tư, cho phép người dùng truy cập và tương tác với các dữ liệu chứng khoán một cách chủ động và tức thời.

Trước hết, hệ thống phải cung cấp chức năng hiển thị thông tin giá cổ phiếu theo thời gian thực. Điều này bao gồm các chỉ số thiết yếu như giá khớp lệnh gần nhất, giá mở cửa, giá cao nhất, giá thấp nhất trong phiên và tỷ lệ phần trăm thay đổi so với giá tham chiếu. Các thông tin này cần được thể hiện theo định dạng bảng dữ liệu có sử dụng màu sắc trực quan để phản ánh xu hướng tăng, giảm hoặc không đổi của từng mã cổ phiếu. Dữ liệu giá phải được cập nhật liên tục thông qua kết nối WebSocket với máy chủ dữ liệu và cần có cơ chế fallback polling (đồng bộ định kỳ) để đảm bảo tính liên tục khi kết nối thời gian thực bị gián đoạn. Hệ thống cũng cần ghi rõ thời điểm cuối cùng dữ liệu được cập nhật để đảm bảo minh bạch và độ tin cậy thông tin.

Tiếp theo, hệ thống cần cung cấp biểu đồ kỹ thuật nâng cao nhằm hỗ trợ nhà đầu tư phân tích xu hướng và hành vi giá của cổ phiếu. Các biểu đồ bao gồm: biểu đồ nến thể hiện dữ liệu OHLC (Open – High – Low – Close) theo khung thời gian linh hoạt (1 phút, 5 phút, 1 giờ, 1 ngày), biểu đồ khối lượng giao dịch, và các biểu đồ tích hợp chỉ báo phân tích kỹ thuật như RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), MA/EMA (Moving Average/Exponential Moving Average) và Bollinger Bands. Các biểu đồ này phải tương tác được (zoom, pan, chọn thời điểm, bật/tắt chỉ báo) và phản hồi ngay khi người dùng thay đổi tham số phân tích.

Một chức năng cốt lõi khác là hỗ trợ giao dịch cổ phiếu trực tuyến. Hệ thống cần cho phép người dùng thực hiện các loại lệnh phổ biến trong thị trường chứng khoán bao gồm: lệnh thị trường (market order), lệnh giới hạn (limit order) và lệnh điều kiện (stop-market, stop-limit). Khi người dùng khởi tạo lệnh, hệ thống phải tiến hành kiểm tra dữ liệu đầu vào, xác thực tài khoản, kiểm tra điều kiện giao dịch, gửi lệnh đến hệ

thông khớp lệnh và theo dõi trạng thái lệnh theo thời gian thực. Ngoài ra, người dùng có thể huỷ lệnh chưa khớp, xem trạng thái lệnh (đã khớp, đang chờ, đã huỷ) và truy xuất toàn bộ lịch sử giao dịch trong một khoảng thời gian tùy chọn.

Đối với chức năng tìm kiếm và lọc, hệ thống cần cung cấp công cụ hỗ trợ người dùng nhanh chóng tìm ra cổ phiếu mục tiêu thông qua việc nhập mã cổ phiếu hoặc tên doanh nghiệp. Đồng thời, tính năng lọc nâng cao phải cho phép lựa chọn theo nhiều tiêu chí như giá, khối lượng giao dịch, vốn hóa thị trường và ngành nghề. Các truy vấn tìm kiếm hoặc lọc dữ liệu cần được xử lý với độ trễ thấp, đảm bảo kết quả trả về gần như ngay lập tức để phục vụ các quyết định đầu tư ngắn hạn và thời gian thực.

Hệ thống phải có khả năng so sánh doanh nghiệp với các công ty cùng ngành dựa trên các chỉ số tài chính như EPS, ROE, P/E để đề xuất các mã cổ phiếu có tiềm năng.

Một điểm nổi bật của hệ thống là khả năng tích hợp cộng đồng đầu tư trực tuyến. Người dùng có thể thảo luận, phản hồi hoặc tương tác về từng mã cổ phiếu thông qua các nhóm cộng đồng. Chức năng đánh giá xu hướng thị trường (tăng hoặc giảm) theo bình chọn cộng đồng cũng là một yếu tố tham khảo hữu ích. Người dùng có thể theo dõi các tài khoản khác, xây dựng mạng lưới đầu tư và tiếp cận các quan điểm chiến lược từ cộng đồng.

Hệ thống cũng cần hỗ trợ người dùng quản lý danh mục đầu tư cá nhân. Người dùng có thể thêm mã cổ phiếu vào danh mục, đồng thời hệ thống tự động tính toán tổng giá trị hiện tại, tổng lãi/lỗ. Ngoài ra người dùng cũng có thể thêm, xóa các mã cổ phiếu muốn theo dõi.

Về mặt quản trị hệ thống, người dùng cần có khả năng đăng ký, đăng nhập và quản lý thông tin cá nhân. Hệ thống phải phân quyền rõ ràng giữa tài khoản người dùng và quản trị viên (admin), đảm bảo mỗi vai trò có quyền truy cập và chức năng phù hợp. Các thông tin như theme hiển thị, ngôn ngữ, cấu hình giao diện phải được lưu vào hồ sơ cá nhân và tự động áp dụng cho các phiên đăng nhập sau.

Cuối cùng, hệ thống kỹ thuật cần đảm bảo vận hành ổn định, an toàn và hiệu suất cao. Điều này bao gồm việc quản lý kết nối WebSocket một cách hiệu quả, caching dữ liệu từ các API để giảm tải truy vấn, ghi log toàn bộ hoạt động và lỗi phát sinh cũng như áp dụng các biện pháp bảo mật hệ thống như giới hạn số lượng truy cập theo IP, ngăn chặn tấn công từ chối dịch vụ (DDoS) và chống spam. Những yếu tố này đóng

vai trò nền tảng để hệ thống có thể mở rộng, bảo trì dễ dàng và vận hành tin cậy trong môi trường thực tế với số lượng người dùng lớn.

3.2. Yêu cầu chức năng

3.2.1. Theo dõi thị trường chứng khoán thời gian thực

Hệ thống cung cấp khả năng theo dõi dữ liệu giá cổ phiếu và chỉ số thị trường theo thời gian thực, đảm bảo người dùng luôn tiếp cận thông tin mới nhất.

- Hiện thị chi tiết giá hiện tại (giá khớp lệnh gần nhất), giá mở cửa, giá cao nhất, giá thấp nhất trong phiên và phần trăm thay đổi so với giá tham chiếu.
- Các thông tin được trực quan hóa bằng màu sắc: xanh (tăng), đỏ (giảm), vàng (không đổi), trình bày theo dạng bảng hoặc thẻ (card).
- Dữ liệu được cập nhật liên tục thông qua kết nối WebSocket, đồng thời có cơ chế đồng bộ dự phòng định kỳ (fallback polling). Hệ thống cũng hiển thị thời gian cập nhật cuối cùng để người dùng dễ theo dõi.

3.2.2. Biểu đồ chứng khoán nâng cao

Cung cấp các loại biểu đồ trực quan hỗ trợ phân tích kỹ thuật:

- Biểu đồ nến với tùy chọn khung thời gian (1 phút, 5 phút, 1 giờ, 1 ngày) và dữ liệu OHLC (Open, High, Low, Close).
- Biểu đồ khối lượng giao dịch.
- Biểu đồ phân tích kỹ thuật với các chỉ báo phổ biến như RSI, MACD, MA/EMA, Bollinger Bands.

3.2.3. Quản lý danh mục đầu tư cá nhân

Hỗ trợ người dùng quản lý danh mục cổ phiếu của riêng mình:

- Thêm hoặc xóa các mã cổ phiếu yêu thích.
- Tính toán tổng giá trị hiện tại và tổng lãi/lỗ của danh mục.

3.2.4. Giao dịch cổ phiếu

Hệ thống cho phép người dùng thực hiện giao dịch cổ phiếu với nhiều loại lệnh linh hoạt, phù hợp với quy định của thị trường chứng khoán Việt Nam:

- Mua/bán theo các lệnh MTL (lệnh thị trường khớp nhanh, phần dư trở thành lệnh giới hạn), ATO (lệnh tại giá mở cửa), ATC (lệnh tại giá đóng cửa) và lệnh giới hạn (LO) cho phép người dùng tự nhập mức giá cụ thể.

- Hệ thống hỗ trợ các thao tác hủy lệnh, theo dõi trạng thái lệnh (đã khớp, chưa khớp) và tra cứu lịch sử giao dịch chi tiết để đảm bảo người dùng dễ dàng kiểm soát toàn bộ quy trình giao dịch mô phỏng.

3.2.5. Tin tức và doanh nghiệp

Cung cấp tin tức liên quan đến từng mã cổ phiếu và gợi ý các cổ phiếu doanh nghiệp cùng ngành để người dùng tham khảo.

3.2.6. Cộng đồng và tương tác

Cho phép người dùng tương tác và chia sẻ thông tin trong cộng đồng nhà đầu tư:

- Đăng bài chia sẻ (bài viết sẽ được admin duyệt trước khi hiển thị).
- Bình luận về cổ phiếu.
- Vote chiều hướng cổ phiếu (Tăng hoặc Giảm).
- Theo dõi tài khoản người dùng khác.
- Giao diện thảo luận cộng đồng theo chủ đề.

3.2.7. Tìm kiếm và lọc cổ phiếu

Hỗ trợ người dùng tìm kiếm và lọc danh sách cổ phiếu theo các tiêu chí:

- Tìm kiếm theo mã hoặc tên công ty.
- Lọc theo các thông số: giá, khối lượng, vốn hoá, ngành.

3.2.8. Hệ thống người dùng và quản trị

Đảm bảo quản lý người dùng đầy đủ và linh hoạt:

- Chức năng đăng ký, đăng nhập.
- Quản lý và chỉnh sửa hồ sơ cá nhân.
- Phân quyền người dùng theo vai trò (user/admin).
- Lưu cấu hình cá nhân như theme, ngôn ngữ và các tùy chỉnh giao diện.

3.2.9. Hệ thống kỹ thuật

Hệ thống backend được thiết kế để duy trì hiệu suất và độ tin cậy cao:

- Quản lý kết nối WebSocket.
- Triển khai cơ chế caching dữ liệu API.
- Ghi log và giám sát lỗi hệ thống.
- Tối ưu hiệu suất backend bằng các kỹ thuật throttle và limit.

3.3. Yêu cầu phi chức năng

3.3.1. Giao diện người dùng thân thiện và dễ sử dụng

Một yêu cầu thiết yếu trong phát triển hệ thống công nghệ phần mềm hiện đại là đảm bảo giao diện người dùng trực quan, dễ tiếp cận và tối ưu trải nghiệm người dùng. Đối với hệ thống giám sát thị trường chứng khoán – vốn hướng đến cả nhà đầu tư chuyên nghiệp lẫn người mới, việc thiết kế giao diện cần tuân thủ nguyên lý “lấy người dùng làm trung tâm” (User-Centered Design).

Giao diện được tổ chức khoa học, dễ quan sát với màu sắc cân đối, biểu tượng trực quan và bố cục hợp lý. Người dùng có thể nhanh chóng tìm kiếm mã cổ phiếu, theo dõi danh mục đầu tư, hoặc nhận cảnh báo từ hệ thống mà không gặp rào cản thao tác. Ngoài ra, yếu tố cá nhân hóa như chế độ sáng/tối, tùy chọn cỡ chữ và bảng màu giúp đáp ứng đa dạng nhu cầu thị giác, hỗ trợ sử dụng trong thời gian dài. Việc tập trung vào trải nghiệm giao diện không chỉ nâng cao hiệu quả sử dụng mà còn tăng mức độ gắn bó của người dùng với nền tảng.

3.3.2. Cập nhật dữ liệu thời gian thực

Với tính chất đặc thù của thị trường chứng khoán – nơi thông tin giá cả biến động từng giây – việc cập nhật dữ liệu theo thời gian thực không chỉ là ưu tiên mà còn là điều kiện tiên quyết. Hệ thống cần đảm bảo mọi thông tin giá, khối lượng, chỉ số thị trường và tín hiệu mua/bán được cập nhật gần như tức thời, với độ trễ không vượt quá 1 giây.

Để đáp ứng yêu cầu này, công nghệ được lựa chọn phải hỗ trợ truyền dữ liệu realtime hiệu quả, đảm bảo phản hồi nhanh từ phía máy chủ đến giao diện người dùng. Điều này đặc biệt quan trọng trong các chức năng như: giám sát giá cổ phiếu, hiển thị biểu đồ nền realtime, đặt lệnh mô phỏng hay xử lý cảnh báo biến động. Tính năng realtime tạo ra trải nghiệm tương tác liền mạch, hỗ trợ người dùng hành động kịp thời trước các biến động thị trường nhanh chóng.

3.3.3. Khả năng mở rộng và chịu tải cao

Là một nền tảng hướng đến cộng đồng nhà đầu tư, hệ thống cần đáp ứng khả năng phục vụ đồng thời cho hàng trăm, thậm chí hàng nghìn người dùng trong cùng thời điểm. Do đó, hiệu năng và khả năng mở rộng là hai tiêu chí quan trọng trong lựa chọn công nghệ phần mềm.

Hệ thống cần vận hành ổn định khi có lượng kết nối lớn, đồng thời có thể mở rộng linh hoạt về tài nguyên khi nhu cầu tăng. Điều này yêu cầu kiến trúc phần mềm cần hỗ trợ cơ chế cân bằng tải, caching hợp lý và khả năng triển khai phân tán (distributed). Các yếu tố như thời gian phản hồi, xử lý đồng bộ và kiểm soát lỗi phải được tối ưu để không làm gián đoạn trải nghiệm người dùng – đặc biệt trong các thời điểm thị trường biến động mạnh.

3.3.4. Bảo mật và an toàn thông tin

Vấn đề bảo mật thông tin người dùng là một trong những yêu cầu quan trọng hàng đầu trong phát triển công nghệ phần mềm, đặc biệt trong các hệ thống tài chính. Các thông tin cá nhân như tài khoản, lịch sử giao dịch, danh mục đầu tư và email cần được bảo vệ nghiêm ngặt thông qua các cơ chế mã hóa khi lưu trữ và truyền tải. Việc áp dụng các tiêu chuẩn như HTTPS, SSL/TLS, mã hóa AES, cùng các lớp bảo vệ như xác thực hai yếu tố (2FA) hay giới hạn quyền truy cập là điều bắt buộc.

Ngoài ra, hệ thống cần chống lại các hình thức tấn công phổ biến như SQL Injection, XSS, CSRF thông qua việc sử dụng các thư viện và phương pháp bảo vệ tiêu chuẩn. Cuối cùng, cơ chế sao lưu và phục hồi dữ liệu định kỳ cũng đóng vai trò đảm bảo tính ổn định và tin cậy của nền tảng trước các sự cố bất khả kháng.

3.4. Phân tích vai trò Node.js trong đáp ứng yêu cầu đề tài

Node.js không chỉ đơn thuần là một nền tảng lập trình phía máy chủ, mà còn là một công nghệ phần mềm phù hợp, phù hợp với việc phát triển các hệ thống hiện đại đòi hỏi tính tương tác cao, cập nhật thời gian thực, khả năng xử lý đồng thời lớn và dễ mở rộng. Trong bối cảnh xây dựng một hệ thống giám sát thị trường chứng khoán thời gian thực – nơi dữ liệu biến động liên tục, lượng người dùng lớn, và yêu cầu phản hồi tức thì – Node.js tỏ ra đặc biệt phù hợp nhờ những đặc điểm kỹ thuật và công nghệ nổi bật sau:

Thứ nhất, Node.js vận hành theo mô hình bất đồng bộ (asynchronous) và hướng sự kiện (event-driven). Điều này giúp hệ thống xử lý nhiều tác vụ đồng thời mà không bị nghẽn luồng hoặc gián đoạn. Đây là yếu tố cực kỳ quan trọng trong các ứng dụng realtime như: theo dõi giá cổ phiếu, cập nhật biểu đồ nến hoặc gửi tín hiệu mua/bán từ AI. Kết hợp với thư viện Socket.IO, Node.js dễ dàng thiết lập kênh truyền dữ liệu hai chiều giữa client-server, cho phép cập nhật dữ liệu liên tục mà không cần tải lại trang – giúp nâng cao trải nghiệm người dùng.

Thứ hai, kiến trúc single-threaded non-blocking I/O của Node.js giúp tối ưu hiệu năng khi có nhiều kết nối đồng thời. Không giống như mô hình thread-based truyền thống (như Java hay PHP), Node.js sử dụng Event Loop để xử lý các sự kiện theo hàng đợi. Từ đó tiết kiệm tài nguyên hệ thống, giảm chi phí phần cứng mà vẫn đảm bảo tốc độ phản hồi nhanh – điều rất quan trọng trong lĩnh vực tài chính với số lượng người dùng đông đảo.

Thứ ba, Node.js hỗ trợ tổ chức hệ thống theo kiến trúc Microservices, cho phép phân tách các chức năng như: phân tích tín hiệu AI, cảnh báo rủi ro, quản lý danh mục, xử lý người dùng, thành các dịch vụ độc lập. Việc chia nhỏ này giúp dễ dàng mở rộng quy mô, cập nhật từng phần hệ thống mà không ảnh hưởng đến toàn bộ nền tảng, đồng thời hỗ trợ tích hợp CI/CD và kiểm thử tự động hiệu quả.

Thứ tư, Node.js sử dụng JavaScript – ngôn ngữ phổ biến cả ở frontend lẫn backend. Điều này giúp đội ngũ phát triển dễ dàng chia sẻ code, đồng bộ dữ liệu và giảm thời gian học công nghệ mới. Trong các mô hình Agile hiện đại, lợi thế "one language" (một ngôn ngữ cho toàn bộ hệ thống) giúp tăng tốc độ phát triển, linh hoạt trong triển khai, và tối ưu nguồn lực.

Thứ năm, Node.js có hệ sinh thái npm khổng lồ với hàng triệu gói thư viện phục vụ đa dạng nhu cầu phát triển: từ bảo mật (Helmet, CORS, bcrypt), xác thực người dùng (JWT), xử lý hiệu năng (PM2, Cluster), đến AI/ML (TensorFlow.js, Brain.js). Việc tận dụng hệ sinh thái giúp tiết kiệm thời gian phát triển, tăng tính mở rộng và đảm bảo chuẩn hóa theo các quy trình DevOps hiện đại.

Lý do lựa chọn Node.js thay vì các công nghệ khác

Trong giai đoạn lựa chọn công nghệ nền tảng cho hệ thống, nhóm phát triển đã tiến hành so sánh các giải pháp phổ biến như:

- PHP (Laravel): dễ học, nhiều tài liệu, nhưng kém hiệu quả trong xử lý realtime và thiếu khả năng mở rộng đồng thời.
- Python (Django): mạnh trong xử lý dữ liệu, AI, nhưng không tối ưu cho các ứng dụng thời gian thực hoặc tải lớn.
- Java (Spring Boot): mạnh mẽ, bảo mật cao, nhưng phức tạp trong cấu hình và triển khai, chi phí vận hành cao.

Cuối cùng, Node.js được lựa chọn vì đáp ứng đầy đủ cả về yêu cầu kỹ thuật lẫn chiến lược công nghệ phần mềm, cụ thể:

- Hiệu năng thời gian thực vượt trội
- Dễ mở rộng theo chiều ngang
- Ngôn ngữ JavaScript đồng bộ toàn hệ thống
- Hệ sinh thái phong phú, cập nhật liên tục
- Phù hợp với các mô hình Agile, DevOps và CI/CD

Có thể thấy Node.js không chỉ là một công nghệ phát triển ứng dụng phía server, mà còn là một nền tảng phần mềm phù hợp với chiến lược phát triển phần mềm hiện đại. Nhờ vào sự kết hợp giữa hiệu năng, khả năng realtime, tính mở rộng, bảo mật và dễ triển khai. Node.js là lựa chọn lý tưởng cho hệ thống giám sát thị trường chứng khoán thời gian thực, đáp ứng cả các yêu cầu phi chức năng (UI, bảo mật, realtime, hiệu năng) lẫn mục tiêu công nghệ dài hạn của doanh nghiệp.

Chương 4: MÔ HÌNH HÓA HỆ THỐNG

4.1. User Stories

4.1.1. Quản lý tài khoản

Title:	Quản lý tài khoản
Value Statement:	Là người dùng, tôi muốn đăng ký, đăng nhập, chỉnh sửa thông tin hồ sơ và đổi mật khẩu để quản lý tài khoản cá nhân an toàn và thuận tiện.
Acceptance Criteria:	<p>Criterion 1: Người dùng có thể đăng ký tài khoản mới với email hợp lệ, mật khẩu đủ mạnh (8 ký tự trở lên, bao gồm chữ hoa, chữ thường, số, ký tự đặc biệt và được mã hóa trước khi lưu vào CSDL).</p> <p>Criterion 2: Hệ thống kiểm tra tài khoản đã tồn tại và thông báo nếu email đã đăng ký trước đó.</p> <p>Criterion 3: Người dùng có thể đăng nhập bằng email/mật khẩu đúng, sai thì hiện thông báo lỗi.</p> <p>Criterion 4: Cho phép cập nhật tên, số điện thoại, địa chỉ, đổi mật khẩu; mật khẩu cũ phải khớp thì mới đổi được.</p>
Definition of Done:	<ul style="list-style-type: none">- Unit Test đạt.- Đạt tiêu chí chấp nhận.- Code đã review.- Test chức năng đạt.- Yêu cầu phi chức năng đạt.- Product Owner phê duyệt.
Owner:	Team phát triển
Iteration:	Chưa lên lịch
Estimate:	5 Points

Bảng 4.1: User story quản lý tài khoản

4.1.2. Quản lý danh mục đầu tư cá nhân

Title:	Quản lý danh mục đầu tư cá nhân
Value Statement:	Là người dùng, tôi muốn thêm/xóa mã cổ phiếu vào danh mục đầu tư để quản lý các mã đang sở hữu và theo dõi lãi/lỗ dễ dàng.
Acceptance Criteria:	<p>Criterion 1: Người dùng xem được danh sách cổ phiếu đã thêm trong danh mục.</p> <p>Criterion 2: Cho phép xóa mã cổ phiếu bất kỳ khỏi danh mục nếu không còn cần theo dõi.</p> <p>Criterion 3: Cập nhật ngay danh mục sau khi thêm/xóa, tính toán giá trị danh mục hiện tại.</p>
Definition of Done:	<ul style="list-style-type: none"> - Unit Test đạt - Đạt tiêu chí chấp nhận. - Code đã review. - Test chức năng đạt. - Yêu cầu phi chức năng đạt. - Product Owner phê duyệt.
Owner:	Team phát triển
Iteration:	Chưa lên lịch
Estimate:	5 Points

Bảng 4.2: User story quản lý danh mục đầu tư cá nhân

4.1.3. Thực hiện giao dịch

Title:	Thực hiện giao dịch
Value Statement:	Là người dùng, tôi muốn thực hiện các lệnh mua/bán cổ phiếu để mô phỏng quá trình đầu tư thực tế.

Acceptance Criteria:	<p>Criterion 1: Người dùng nhập mã cổ phiếu, loại lệnh (mua/bán), số lượng và giá đặt.</p> <p>Criterion 2: Hệ thống kiểm tra số dư tài khoản (khi mua) hoặc số lượng cổ phiếu (khi bán).</p> <p>Criterion 3: Nếu điều kiện không đủ thì thông báo lỗi; nếu đủ thì ghi lệnh vào hệ thống.</p> <p>Criterion 4: Người dùng có thể xem trạng thái lệnh (đã khớp, chờ khớp, hủy).</p> <p>Criterion 5: Cho phép hủy lệnh nếu lệnh chưa khớp.</p> <p>Criterion 6: Nếu không khớp lệnh trong vòng 48h thì lệnh đặt sẽ tự hủy.</p>
Definition of Done:	<ul style="list-style-type: none"> - Unit Test đạt. - Đạt tiêu chí chấp nhận. - Code đã review - Test chức năng đạt. - Yêu cầu phi chức năng đạt. - Product Owner phê duyệt.
Owner:	Team phát triển
Iteration:	Chưa lên lịch
Estimate:	5 Points

Bảng 4.3: User story thực hiện giao dịch

4.1.4. Tương tác cộng đồng

Title:	Tương tác cộng đồng
Value Statement:	Là người dùng, tôi muốn đăng bài viết, bình luận, theo dõi nhà đầu tư khác để chia sẻ kiến thức và kinh nghiệm.
Acceptance Criteria:	Criterion 1: Người dùng có thể tạo bài viết mới với tiêu đề, nội dung, tag liên quan.

	<p>Criterion 2: Các bài viết sẽ được quản trị viên duyệt trước khi hiển thị công khai.</p> <p>Criterion 3: Người dùng có thể bình luận dưới bài viết hoặc xóa bình luận của mình.</p> <p>Criterion 4: Cho phép báo cáo bài viết vi phạm, admin nhận được báo cáo để kiểm duyệt.</p>
Definition of Done:	<ul style="list-style-type: none"> - Unit Test đạt. - Đạt tiêu chí chấp nhận. - Code đã review. - Test chức năng đạt. - Yêu cầu phi chức năng đạt. - Product Owner phê duyệt.
Owner:	Team phát triển
Iteration:	Chưa lên lịch
Estimate:	5 Points

Bảng 4.4: User story chức năng tương tác cộng đồng

4.1.5. Tìm kiếm cổ phiếu

Title:	Tìm kiếm cổ phiếu
Value Statement:	Là người dùng, tôi muốn tìm kiếm cổ phiếu theo mã hoặc tên công ty và lọc theo ngành, giá, vốn hóa để dễ dàng lựa chọn mã phù hợp.
Acceptance Criteria:	<p>Criterion 1: Người dùng nhập từ khóa, hệ thống trả về danh sách cổ phiếu phù hợp.</p> <p>Criterion 2: Có bộ lọc nâng cao theo ngành, giá, vốn hóa, sắp xếp tăng/giảm.</p> <p>Criterion 3: Kết quả hiển thị nhanh, chính xác.</p>
Definition of	- Unit Test đạt.

Done:	<ul style="list-style-type: none"> - Đạt tiêu chí chấp nhận. - Code đã review - Test chức năng đạt. - Yêu cầu phi chức năng đạt. - Product Owner phê duyệt.
Owner:	Team phát triển
Iteration:	Chưa lên lịch
Estimate:	5 Points

Bảng 4.5: User story chức năng tìm kiếm cổ phiếu

4.1.6. Truy cập dữ liệu thị trường

Title:	Truy cập dữ liệu thị trường
Value Statement:	Là người dùng, tôi muốn xem dữ liệu giá cổ phiếu theo thời gian thực để theo dõi biến động và ra quyết định kịp thời.
Acceptance Criteria:	<p>Criterion 1: Hệ thống hiển thị giá realtime qua WebSocket, fallback sang polling nếu WebSocket lỗi.</p> <p>Criterion 2: Hiển thị các thông số giá mở cửa, cao nhất, thấp nhất, giá khớp gần nhất, khối lượng.</p> <p>Criterion 3: Dữ liệu tự động làm mới và đồng bộ chính xác.</p>
Definition of Done:	<ul style="list-style-type: none"> - Unit Test đạt. - Đạt tiêu chí chấp nhận. - Code đã review - Test chức năng đạt.

	<ul style="list-style-type: none"> - Yêu cầu phi chức năng đạt. - Product Owner phê duyệt.
Owner:	Team phát triển
Iteration:	Chưa lên lịch
Estimate:	5 Points

Bảng 4.6: User story chức năng truy cập thị trường

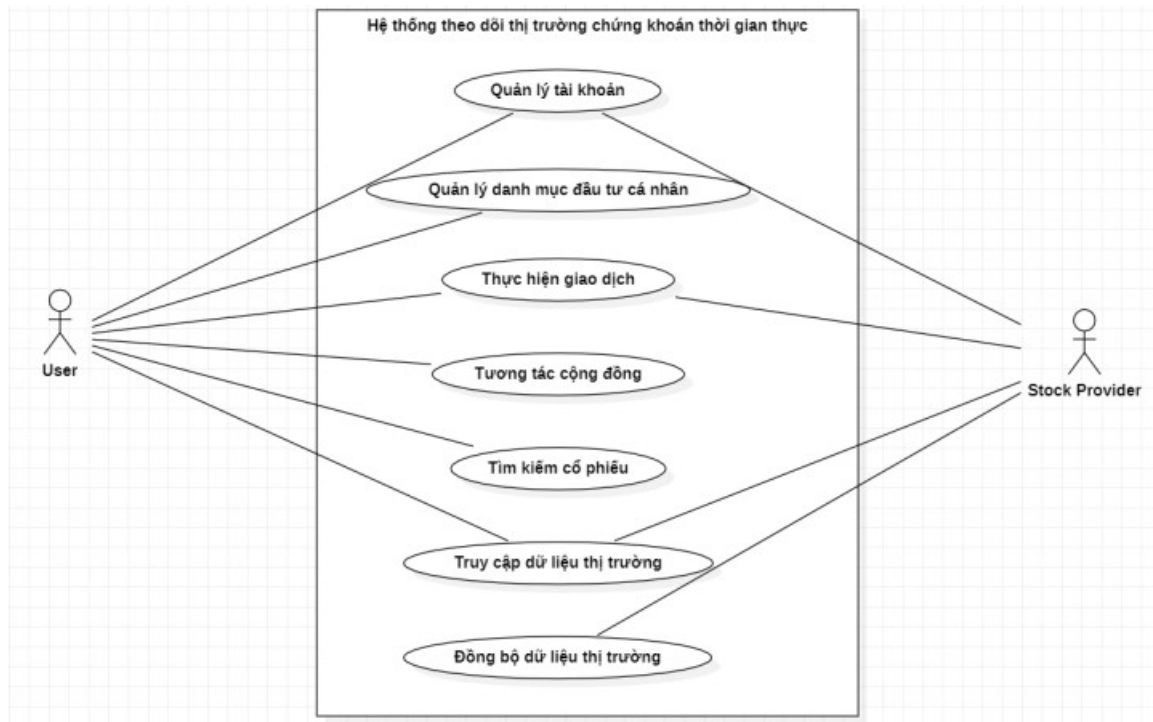
4.1.7. Đồng bộ dữ liệu thị trường

Title:	Đồng bộ dữ liệu thị trường
Value Statement:	Là hệ thống, tôi muốn tự động đồng bộ dữ liệu thị trường từ Stock Provider định kỳ để đảm bảo dữ liệu luôn chính xác và mới nhất.
Acceptance Criteria:	<p>Criterion 1: Thiết lập lịch chạy đồng bộ định kỳ (1 phút, 5 phút, theo cấu hình).</p> <p>Criterion 2: Tự động lưu cache dữ liệu mới, thay thế dữ liệu cũ.</p> <p>Criterion 3: Ghi log trạng thái đồng bộ, cảnh báo khi lỗi xảy ra.</p>
Definition of Done:	<ul style="list-style-type: none"> - Unit Test đạt. - Đạt tiêu chí chấp nhận. - Code đã review - Test chức năng đạt. - Yêu cầu phi chức năng đạt. - Product Owner phê duyệt.
Owner:	Team phát triển
Iteration:	Chưa lên lịch
Estimate:	5 Points

Bảng 4.7: User story chức năng đồng bộ dữ liệu thị trường

4.2. Biểu đồ Use Case

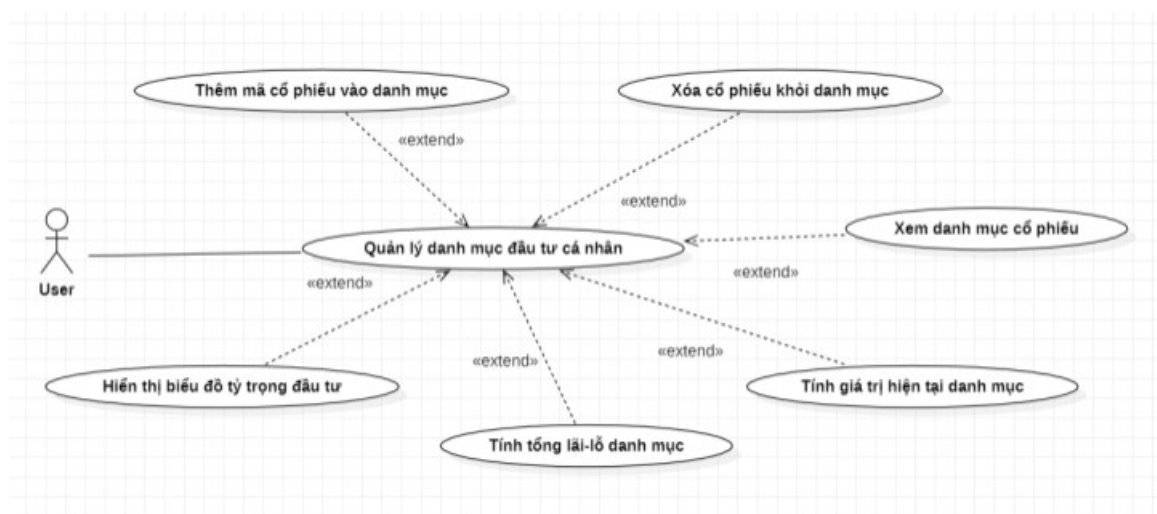
4.2.1. Tổng quan chức năng



Hình 4.1: Use case tổng quát chức năng

4.2.2. Quản lý danh mục đầu tư cá nhân

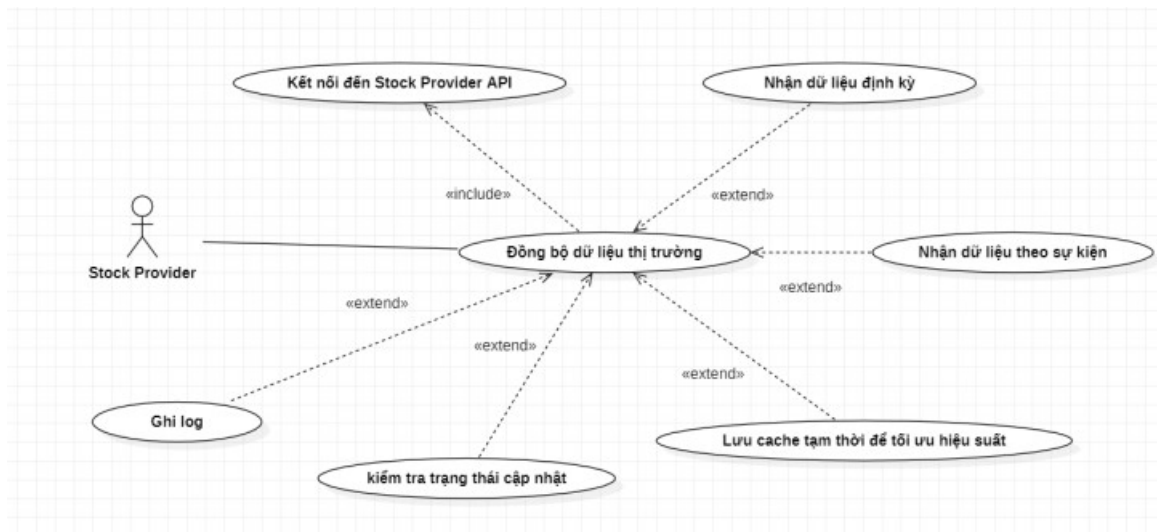
Người dùng có thể quản lý danh mục đầu tư của mình bằng cách thêm mã cổ phiếu, xóa cổ phiếu không cần thiết, xem danh mục hiện tại, tính giá trị và lãi/lỗ, cũng như hiển thị biểu đồ tỷ trọng đầu tư để đưa ra quyết định tối ưu.



Hình 4.2: Use case chức năng quản lý danh mục đầu tư cá nhân

4.2.3. Đồng bộ dữ liệu thị trường

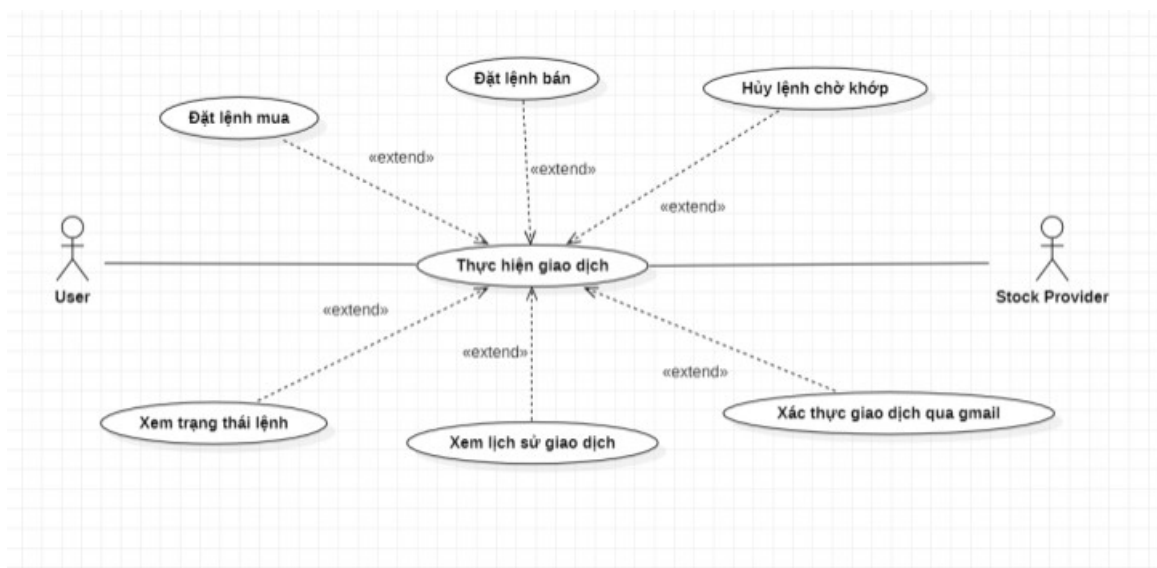
Hệ thống tự động kết nối với Stock Provider API để nhận dữ liệu giá cổ phiếu định kỳ hoặc theo sự kiện. Dữ liệu được lưu cache tạm thời, kiểm tra trạng thái cập nhật và ghi log để đảm bảo tính toàn vẹn và tối ưu hiệu suất.



Hình 4.3: Use case chức năng đồng bộ dữ liệu thị trường

4.2.4. Thực hiện giao dịch

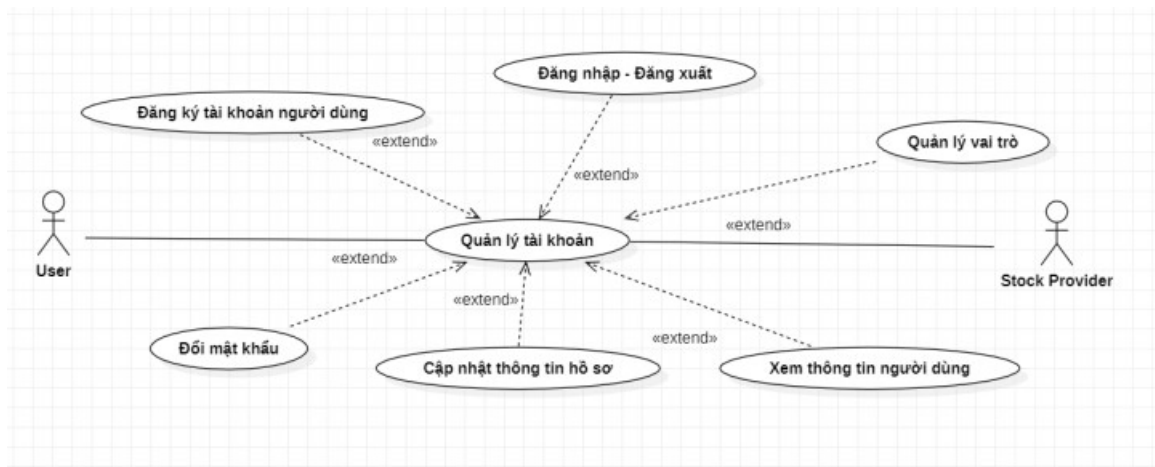
Người dùng có thể thực hiện các lệnh mua, bán cổ phiếu. Hệ thống hỗ trợ hủy lệnh đang chờ khớp, xác thực giao dịch qua email, đồng thời cho phép người dùng theo dõi trạng thái và lịch sử giao dịch.



Hình 4.4: Use case chức năng thực hiện giao dịch

4.2.5. Quản lý tài khoản

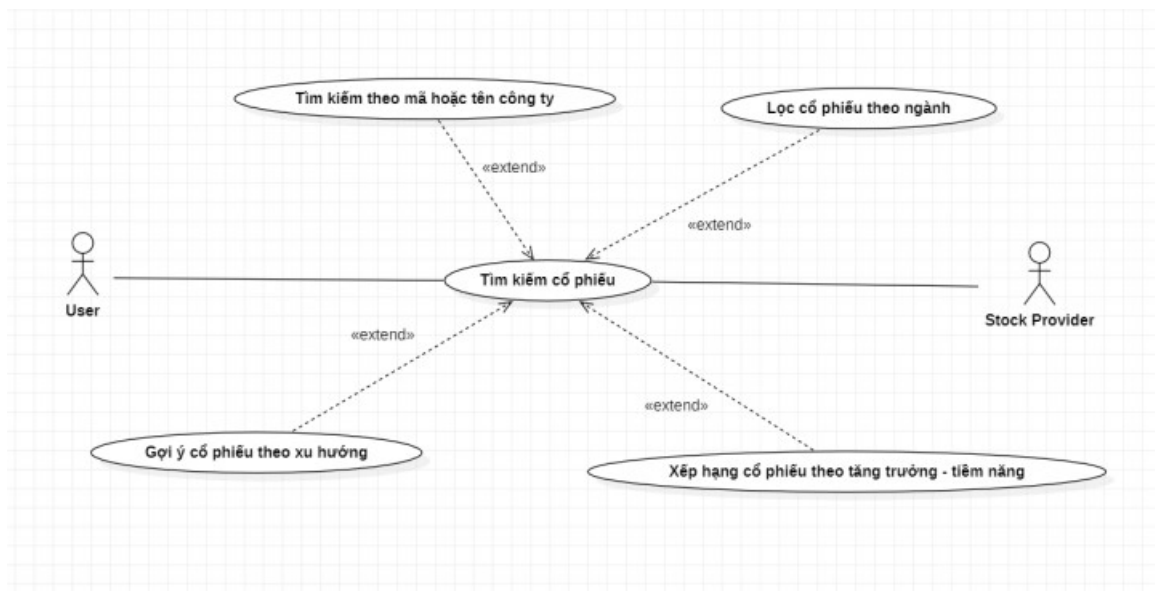
Người dùng thực hiện đăng ký, đăng nhập, đăng xuất tài khoản, quản lý vai trò, cập nhật thông tin hồ sơ, đổi mật khẩu và xem thông tin người dùng. Các chức năng này đảm bảo quản lý an toàn và cá nhân hóa trải nghiệm.



Hình 4.5: Use case chức năng quản lý tài khoản

4.2.6. Tìm kiếm cổ phiếu

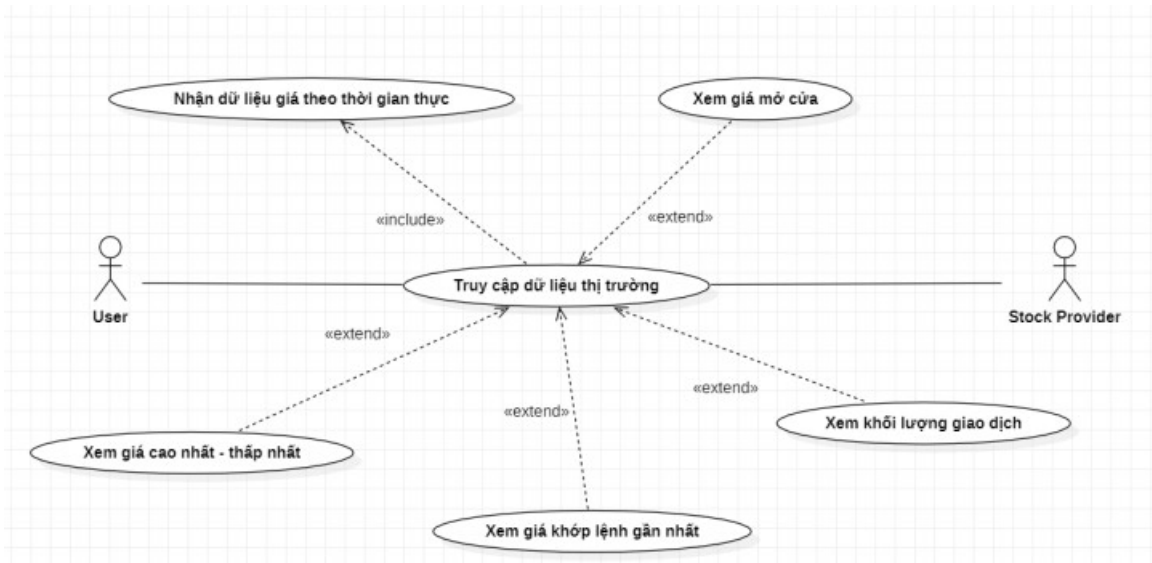
Người dùng có thể tìm kiếm cổ phiếu bằng mã hoặc tên công ty, lọc cổ phiếu theo ngành, đồng thời hệ thống cung cấp gợi ý cổ phiếu theo xu hướng và xếp hạng cổ phiếu dựa trên tiềm năng tăng trưởng.



Hình 4.6: Use case chức năng tìm kiếm cổ phiếu

4.2.7. Truy cập dữ liệu thị trường

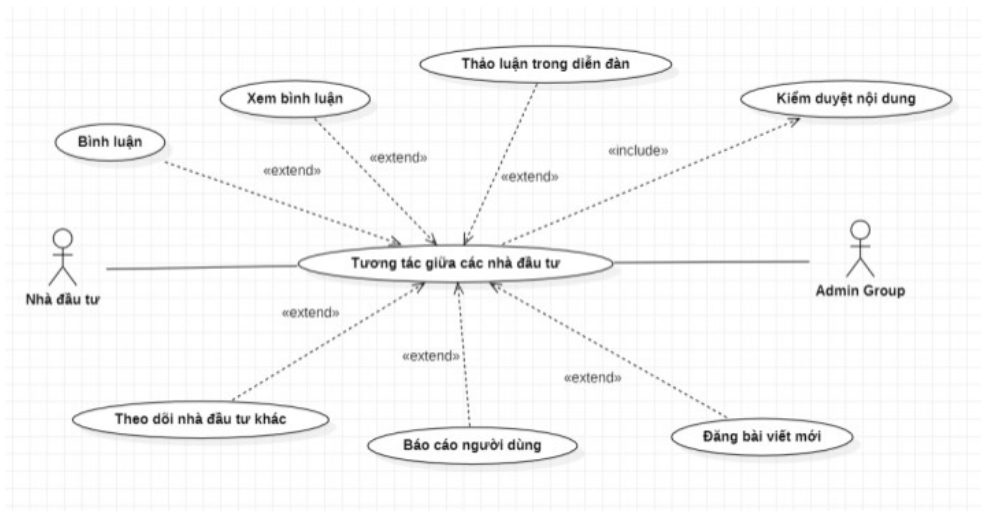
Người dùng có thể nhận dữ liệu giá cổ phiếu theo thời gian thực và truy xuất các thông tin như giá mở cửa, giá cao nhất, giá thấp nhất, giá khớp lệnh gần nhất và khối lượng giao dịch để đưa ra quyết định đầu tư.



Hình 4.7: Use case chức năng truy cập dữ liệu thị trường

4.2.8. Tương tác giữa các nhà đầu tư

Nhà đầu tư có thể tham gia diễn đàn thảo luận, đăng bài viết mới, bình luận, xem bình luận, theo dõi nhà đầu tư khác, báo cáo người dùng và Admin Group sẽ kiểm duyệt nội dung để đảm bảo môi trường lành mạnh.



Hình 4.8: Use case chức năng tương tác giữa các nhà đầu tư

4.3. Sơ đồ Class Diagram

Sơ đồ lớp mô tả cấu trúc các thành phần chính của hệ thống giám sát và giao dịch cổ phiếu. Mỗi lớp đại diện cho một thực thể trong hệ thống, bao gồm thông tin, chức năng xử lý và quan hệ với các lớp khác.

- User

- Mục đích: Quản lý thông tin người dùng như tên, email, số điện thoại, vai trò và giấy tờ tùy thân.
- Phương thức: Register(), Login(), UpdateProfile() - cho phép đăng ký, đăng nhập và cập nhật hồ sơ cá nhân.
- Quan hệ:
 - Kết nối Role để phân quyền.
 - Kết nối IdentityCard để quản lý giấy tờ.
 - Quan hệ với Wallet để quản lý ví thanh toán.
 - Quan hệ với Order, Favorite, Post, Comment.

- Role

- Mục đích: Lưu thông tin quyền hạn của người dùng (Admin, User).
- Phương thức: AddRole(), UpdateRole(), DeleteRole().
- Quan hệ: Liên kết User.

- IdentityCard

- Mục đích: Quản lý thông tin giấy tờ cá nhân của User như CMND, CCCD.
- Phương thức: CreateCard(), UpdateCard(), DeleteCard().
- Quan hệ: Liên kết User, Wallet.

- **Wallet**

- Mục đích: Lưu thông tin ví thanh toán của User: số thẻ, ngân hàng, hạn thẻ, CCV.
- Phương thức: CreateWallet(), UpdateWallet(), DeleteWallet(), CheckBalance().
- Quan hệ: Liên kết User, IdentityCard.

- **Post & Comment**

- Post: Người dùng có thể tạo bài viết để chia sẻ, phân tích.
 - Phương thức: CreatePost(), DeletePost(), EditPost().
- Comment: Người dùng bình luận trên các bài viết.
 - Phương thức: AddComment(), EditComment(), DeleteComment().
- Quan hệ: Post và Comment liên kết chặt chẽ với User.

- **Order**

- Mục đích: Đại diện lệnh mua/bán cổ phiếu.
- Phương thức: PlaceOrder(), CancelOrder(), CheckOrderStatus().
- Quan hệ: Liên kết User, Stock, IdentityCard.

- **Favorite**

- Mục đích: Danh sách các công ty/cổ phiếu mà người dùng quan tâm.
- Phương thức: AddToFavorite(), ListFavorites(), RemoveFromFavorite().
- Quan hệ: Liên kết User, Stock, Company.

- **Company**

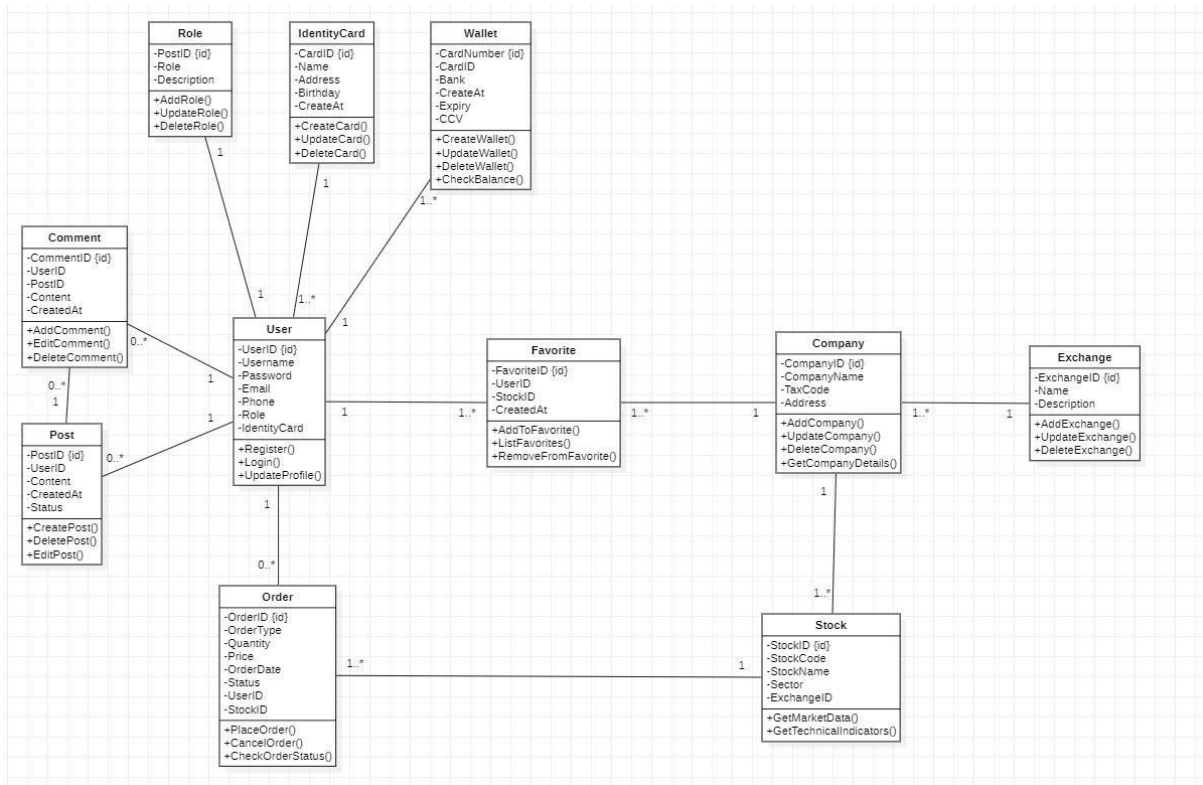
- Mục đích: Thông tin công ty phát hành cổ phiếu (Tên, mã số thuế, địa chỉ).
- Phương thức: AddCompany(), UpdateCompany(), DeleteCompany(), GetCompanyDetails().
- Quan hệ: Liên kết Stock, Exchange, Favorite.

- **Stock**

- Mục đích: Quản lý mã cổ phiếu, tên, lĩnh vực, sản phẩm yết.
- Phương thức: GetMarketData(), GetTechnicalIndicators().
- Quan hệ: Liên kết Company, Order, Favorite.

- **Exchange**

- Mục đích: Quản lý thông tin các sàn giao dịch chứng khoán.
- Phương thức: AddExchange(), UpdateExchange(), DeleteExchange().
- Quan hệ: Liên kết Company.



Hình 4.9: Sơ đồ class diagram

4.4. Biểu đồ hoạt động (Activity Diagram)

4.4.1. Đăng nhập / Đăng ký

Sơ đồ này mô tả luồng hoạt động khi người dùng lựa chọn hành động đăng nhập hoặc đăng ký. Người dùng nhập thông tin cần thiết, hệ thống kiểm tra tính hợp lệ, nếu có lỗi (sai cú pháp hoặc sai thông tin), người dùng có thể nhập lại hoặc kết thúc. Khi thông tin hợp lệ, hệ thống sẽ cho phép đăng nhập hoặc lưu tài khoản mới.



Hình 4.10: Activity diagram đăng nhập/đăng kí

4.4.2. Quản lý danh mục đầu tư

- Quản lý danh mục cổ phiếu yêu thích

Mô tả quy trình người dùng đăng nhập vào hệ thống, mở danh mục yêu thích, truy vấn và hiển thị danh sách các mã cổ phiếu đã lưu. Người dùng có thể thực hiện thêm hoặc xóa mã cổ phiếu khỏi danh mục, sau đó hệ thống cập nhật và hiển thị danh sách mới nhất để người dùng tiếp tục quản lý.



Hình 4.11: Activity diagram quản lý cổ phiếu yêu thích

- Tính lãi/lỗ danh mục

Mô tả quá trình người dùng đăng nhập, mở danh mục đầu tư và hệ thống tự động truy vấn giá mua cùng giá hiện tại của từng mã cổ phiếu. Dựa trên dữ liệu đó, hệ thống tính

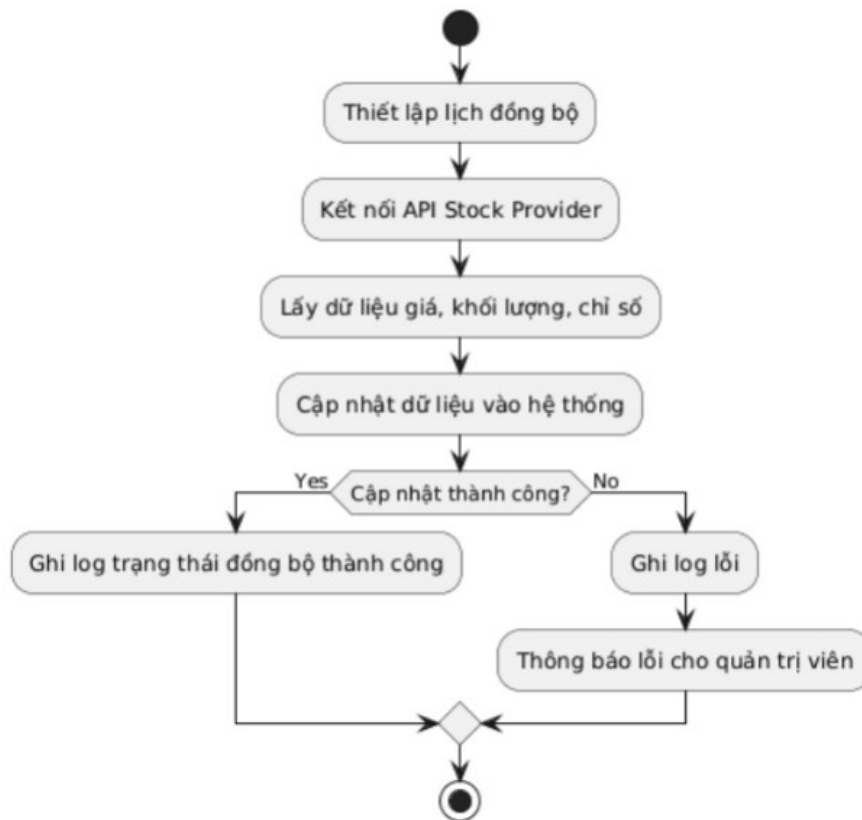
toán lãi/lỗ cho từng mã, sau đó tổng hợp lãi/lỗ của toàn danh mục và hiển thị kết quả để người dùng đánh giá hiệu quả đầu tư của mình.



Hình 4.12: Activity diagram thống kê danh mục đầu tư

4.4.3. Đồng bộ dữ liệu thị trường

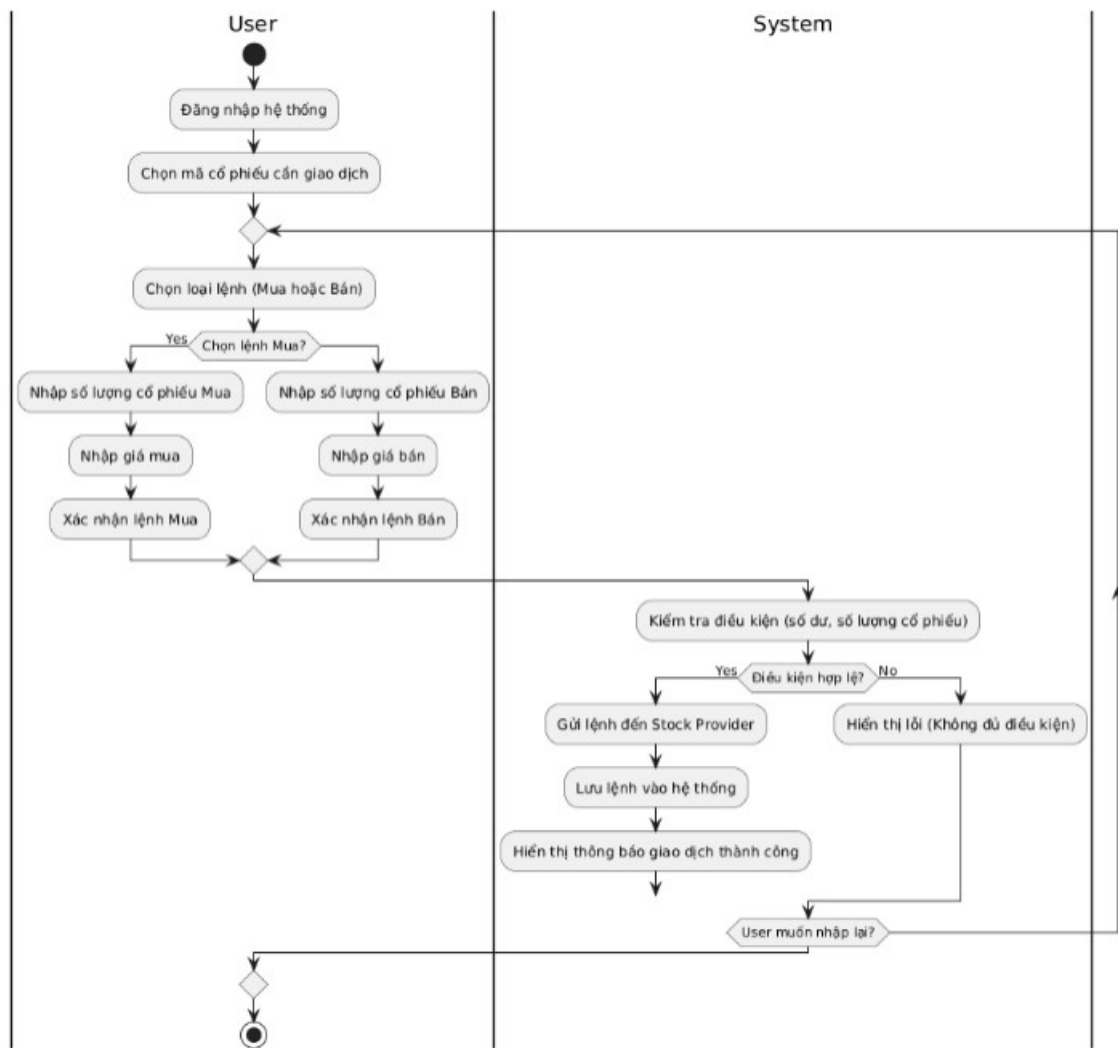
Quy trình tự động thiết lập lịch đồng bộ dữ liệu. Hệ thống kết nối API Stock Provider, lấy dữ liệu giá, khối lượng, chỉ số, cập nhật vào hệ thống. Nếu cập nhật thành công thì ghi log trạng thái, nếu thất bại thì ghi log lỗi và thông báo quản trị viên.



Hình 4.13: Activity diagram đồng bộ dữ liệu thị trường

4.4.4. Giao dịch cổ phiếu

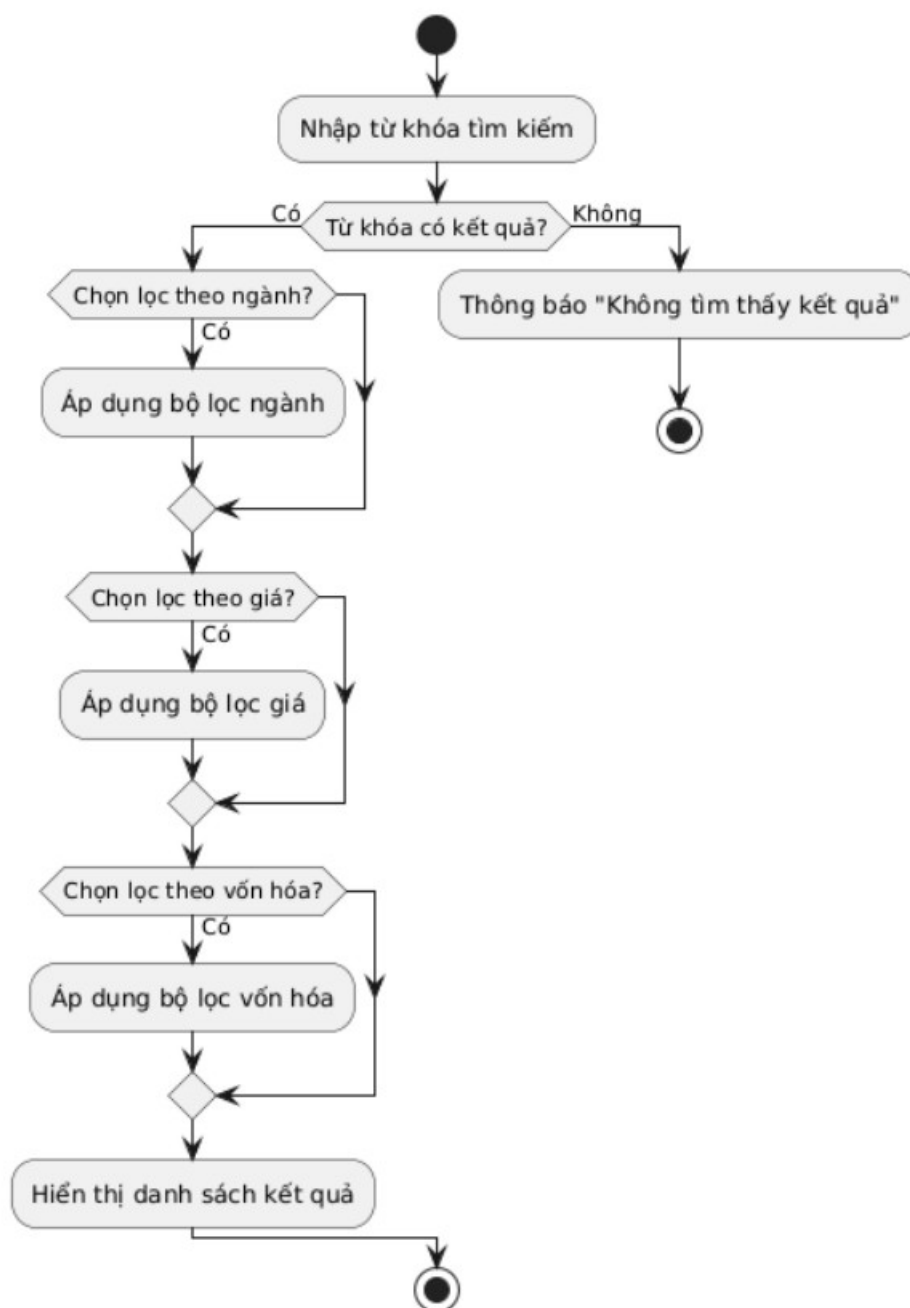
Biểu đồ thể hiện các bước người dùng thực hiện giao dịch cổ phiếu. Người dùng chọn mã cổ phiếu, loại lệnh (mua/bán), nhập thông tin lệnh. Hệ thống kiểm tra điều kiện, nếu đủ thì gửi lệnh đến nhà cung cấp và lưu lệnh. Nếu không đủ điều kiện, người dùng có thể nhập lại hoặc dừng.



Hình 4.14: Activity diagram giao dịch cổ phiếu

4.4.5. Tìm kiếm cổ phiếu

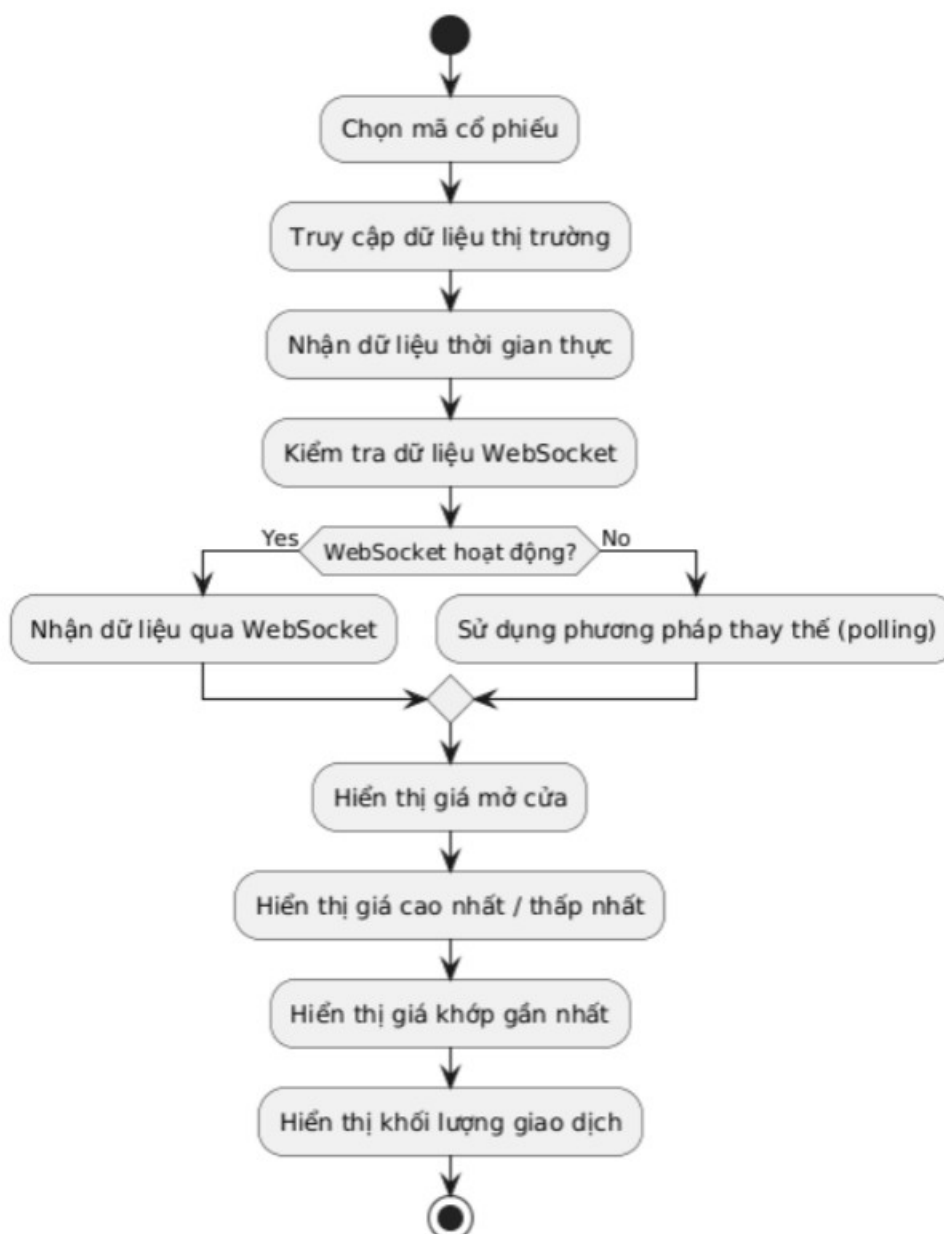
Quy trình người dùng nhập từ khóa tìm kiếm cổ phiếu. Hệ thống hiển thị kết quả. Người dùng có thể áp dụng bộ lọc theo ngành, giá, vốn hóa để thu hẹp kết quả.



Hình 4.15: Activity diagram tìm kiếm cổ phiếu

4.4.6. Truy cập dữ liệu thị trường

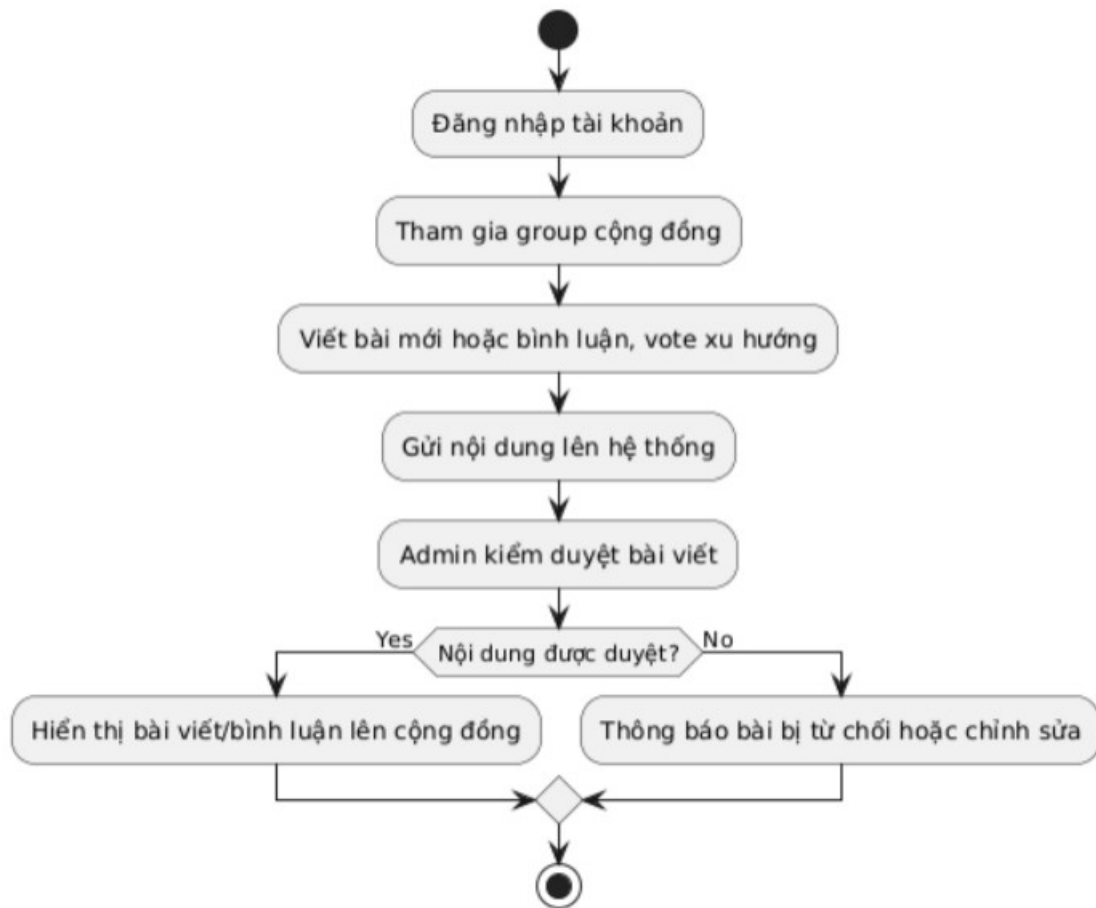
Biểu đồ mô tả quy trình truy cập và hiển thị dữ liệu thị trường của một mã cổ phiếu. Hệ thống nhận dữ liệu realtime qua WebSocket hoặc Polling nếu WebSocket lỗi. Hệ thống hiển thị giá mở cửa, giá cao nhất/thấp nhất, giá khớp gần nhất và khối lượng giao dịch.



Hình 4.16: Activity diagram chức năng truy cập dữ liệu thị trường

4.4.7. Tương tác cộng đồng

Mô tả quy trình người dùng tham gia group cộng đồng, đăng bài viết mới, bình luận, vote xu hướng. Nội dung được gửi lên hệ thống, admin sẽ kiểm duyệt. Nội dung hợp lệ sẽ được hiển thị, nếu không sẽ bị từ chối hoặc yêu cầu chỉnh sửa.

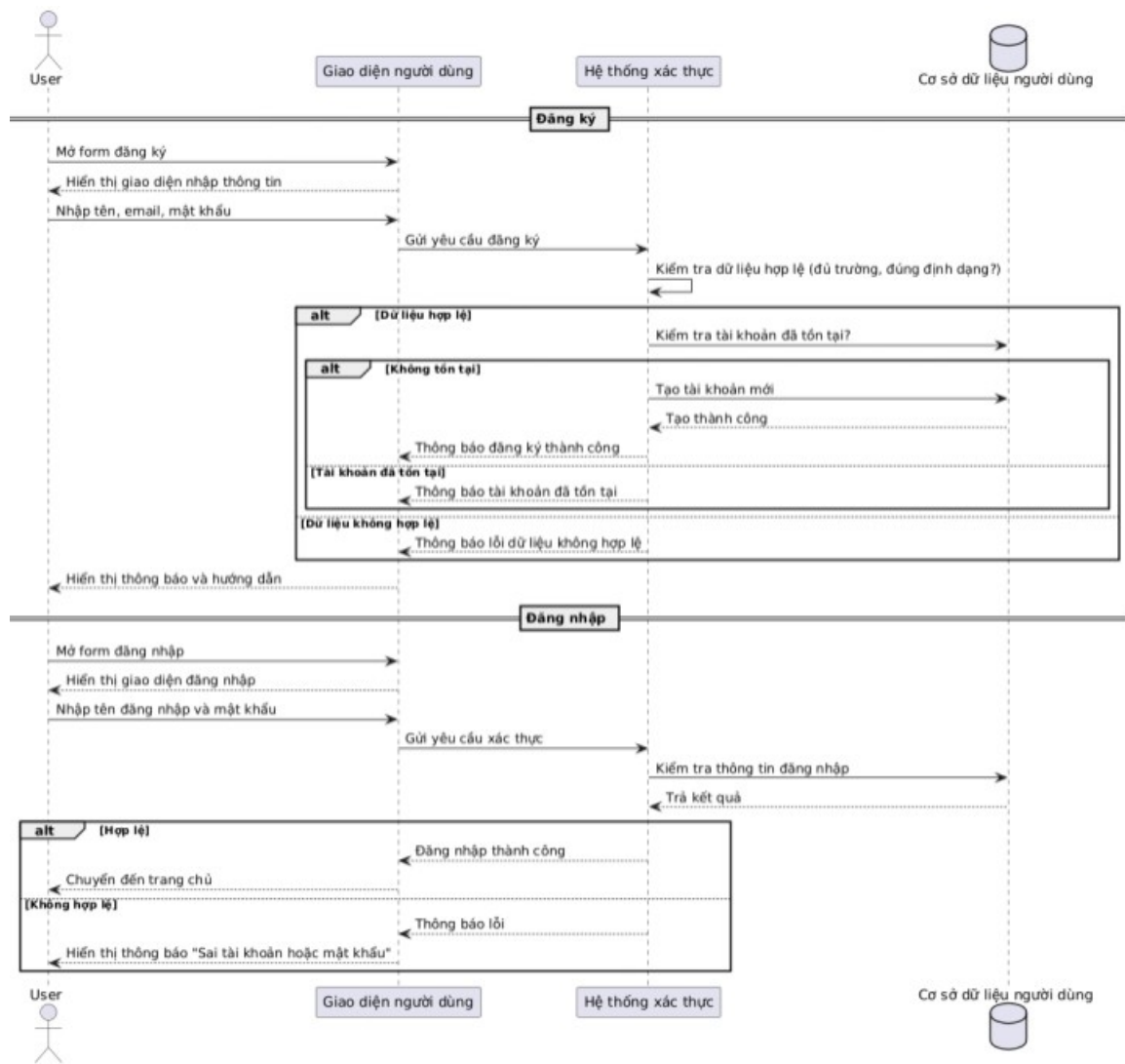


Hình 4.17: Activity diagram chức năng tương tác cộng đồng

4.5. Biểu đồ trình tự (Sequence Diagram)

4.5.1. Đăng ký và Đăng nhập

Người dùng thực hiện đăng ký bằng cách nhập thông tin cá nhân (tên, email, mật khẩu). Hệ thống xác thực dữ liệu, kiểm tra tài khoản đã tồn tại chưa, nếu hợp lệ thì lưu vào cơ sở dữ liệu và thông báo thành công. Với đăng nhập, người dùng nhập tài khoản, mật khẩu, hệ thống kiểm tra và phản hồi kết quả.

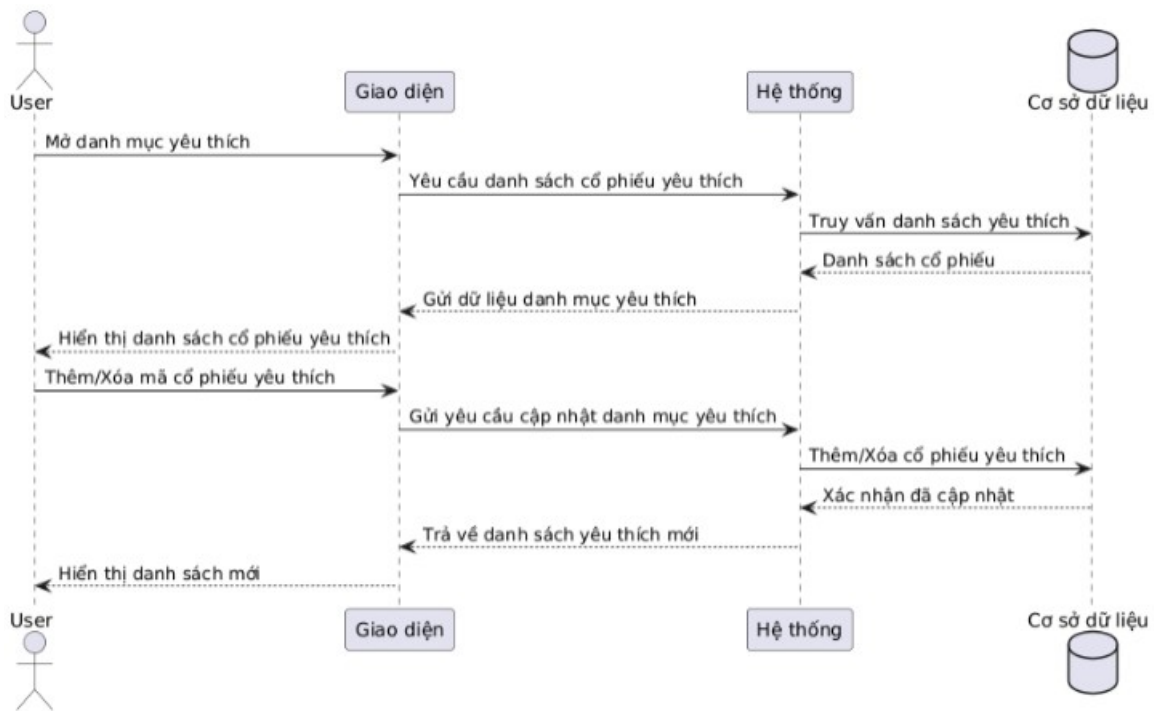


Hình 4.18: Sequence diagram đăng kí/đăng nhập

4.5.2. Quản lý danh mục

- Quản lý danh mục cổ phiếu yêu thích

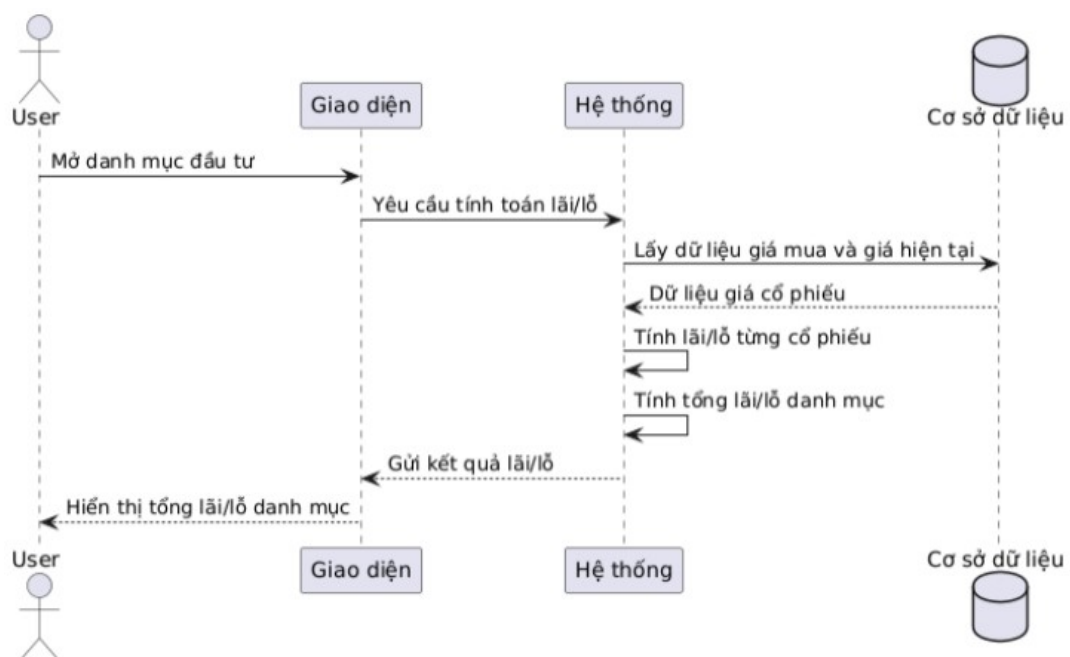
Người dùng mở danh mục cổ phiếu yêu thích, hệ thống truy vấn và trả về danh sách hiện tại. Người dùng có thể thêm hoặc xóa mã cổ phiếu yêu thích, hệ thống sẽ cập nhật dữ liệu và trả về danh sách mới nhất.



Hình 4.19: Sequence diagram quản lý cổ phiếu yêu thích

- Tính lãi/lỗ danh mục

Người dùng yêu cầu tính toán tổng lãi/lỗ danh mục đầu tư. Hệ thống lấy dữ liệu giá mua và giá hiện tại của các cổ phiếu, tính lãi/lỗ cho từng mã và tổng hợp kết quả trả về cho người dùng.



Hình 4.20: Sequence diagram thống kê danh mục đầu tư

4.5.3. Đồng bộ dữ liệu thị trường

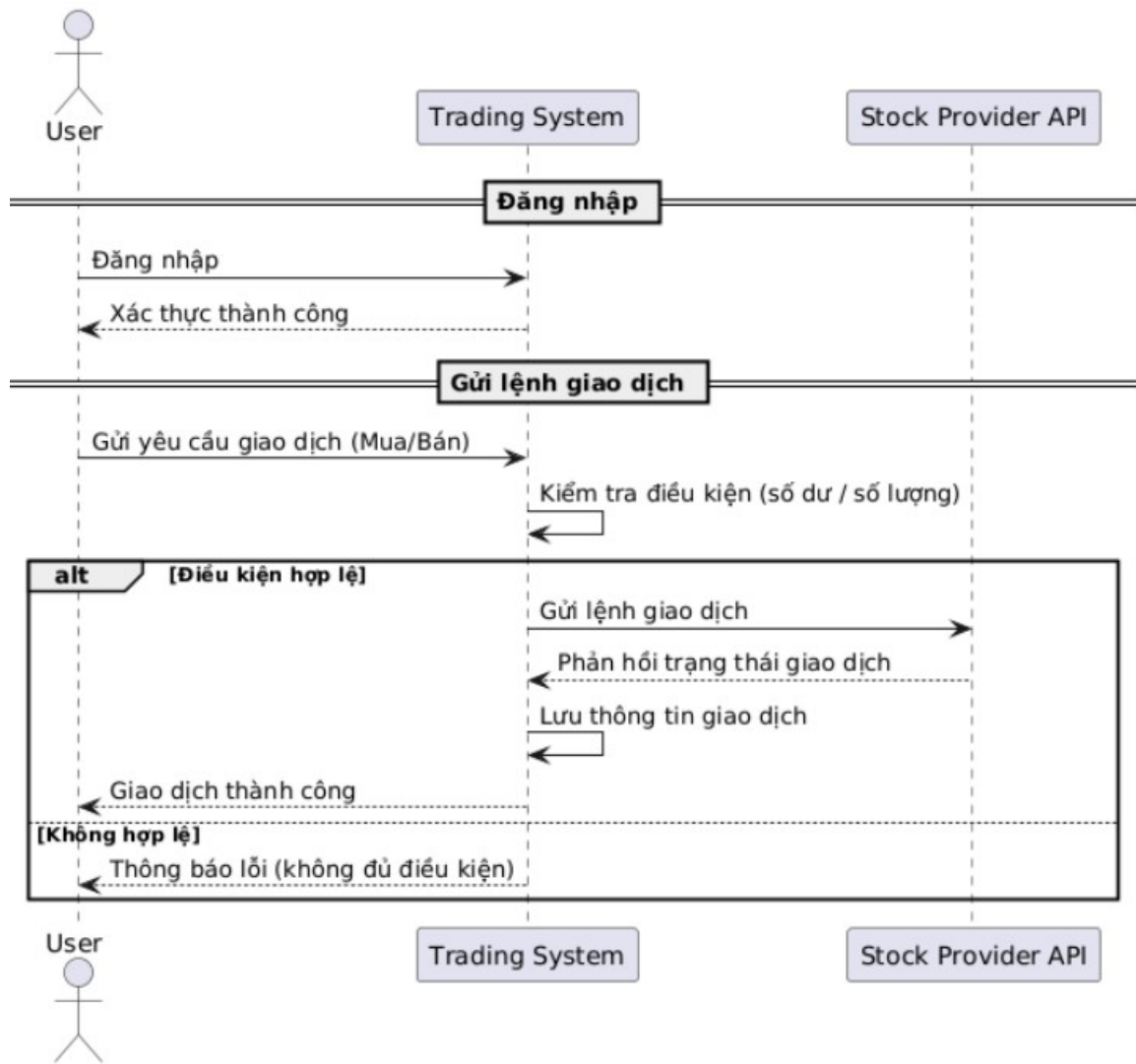
Hệ thống tự động kích hoạt tiến trình đồng bộ dữ liệu định kỳ. Dữ liệu giá cổ phiếu được lấy từ Stock Provider, lưu trữ và cập nhật vào cache để tối ưu hiệu năng. Kết quả đồng bộ được ghi log trạng thái.



Hình 4.21: Sequence diagram đồng bộ dữ liệu thị trường

4.5.4. Giao dịch mua/bán cổ phiếu

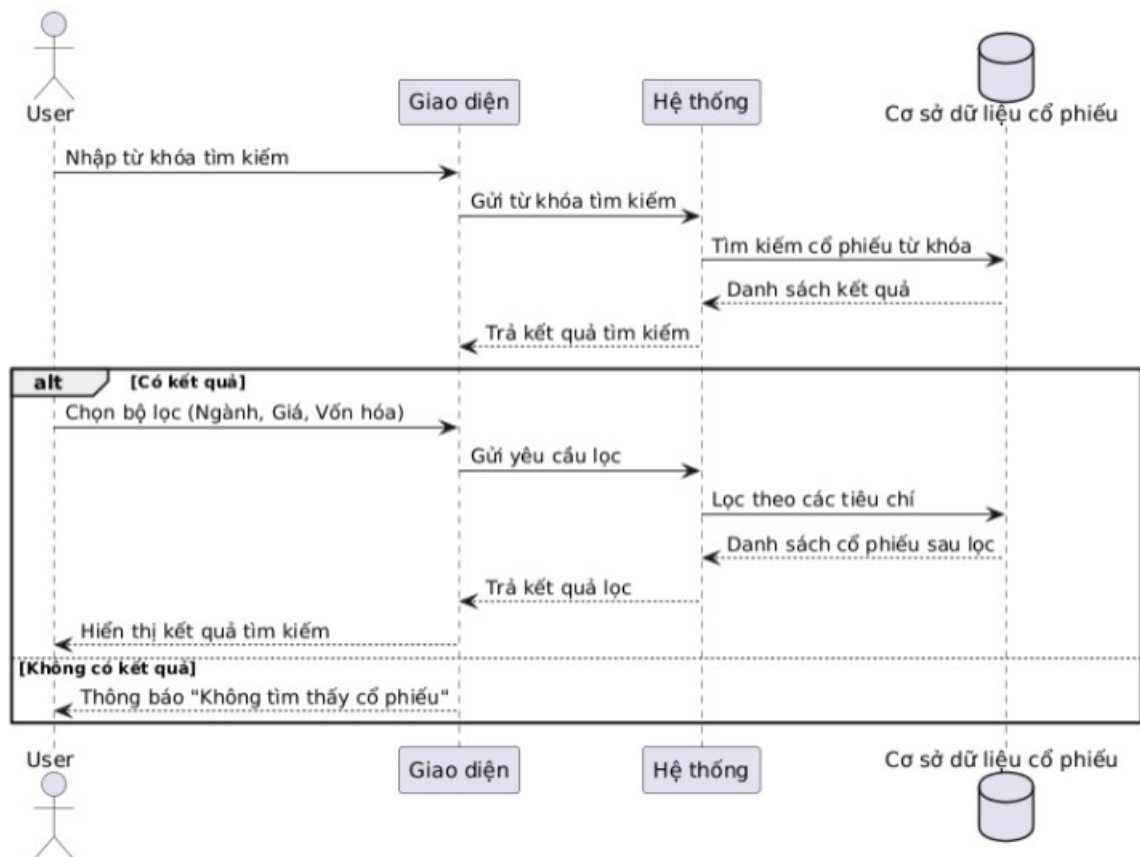
Người dùng gửi yêu cầu giao dịch (mua/bán), hệ thống kiểm tra điều kiện (số dư, số lượng cổ phiếu), nếu đủ điều kiện thì gửi lệnh đến Stock Provider và lưu lại thông tin. Kết quả được trả về để thông báo cho người dùng.



Hình 4.22: Sequence diagram giao dịch cổ phiếu

4.5.5. Tìm kiếm và lọc cổ phiếu

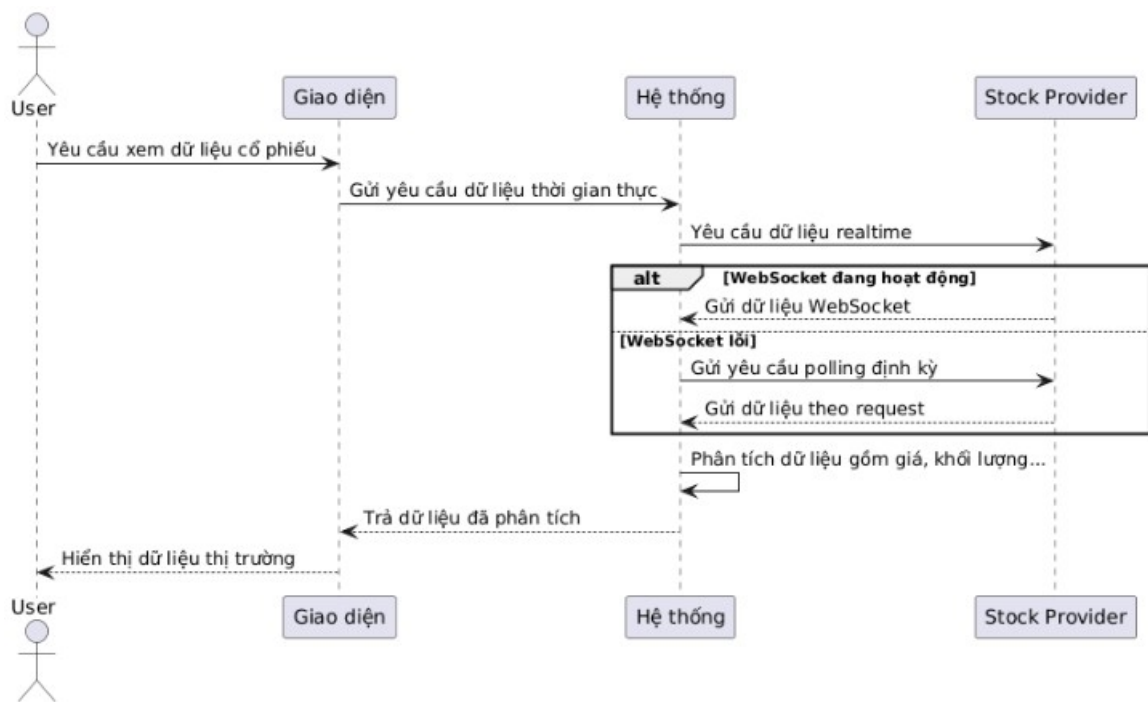
Người dùng nhập từ khóa tìm kiếm, hệ thống trả kết quả các cổ phiếu khớp từ cơ sở dữ liệu. Ngoài ra, người dùng có thể áp dụng các bộ lọc như ngành, giá, vốn hóa để thu hẹp danh sách.



Hình 4.23: Sequence diagram tìm kiếm cổ phiếu

4.5.6. Truy cập dữ liệu thị trường thời gian thực

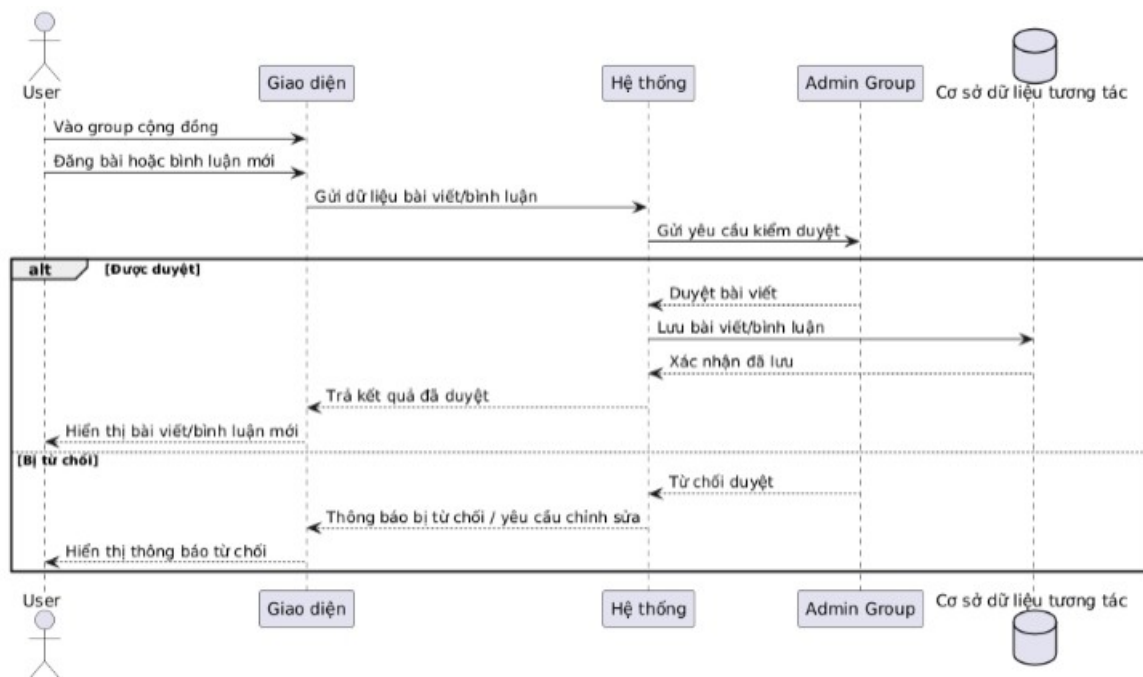
Người dùng yêu cầu xem dữ liệu giá cổ phiếu thời gian thực. Hệ thống yêu cầu Stock Provider gửi dữ liệu qua WebSocket. Nếu WebSocket lỗi, sẽ tự động dùng phương pháp polling thay thế. Dữ liệu gồm giá mở cửa, giá cao/thấp nhất, giá khớp gần nhất, khối lượng giao dịch được phân tích và hiển thị.



Hình 4.24: Sequence diagram chức năng truy cập dữ liệu thị trường

4.5.7. Tương tác cộng đồng

Người dùng có thể tham gia cộng đồng bằng cách viết bài, bình luận. Nội dung sẽ được gửi lên hệ thống để Admin duyệt. Nếu bài viết/bình luận được duyệt sẽ được hiển thị công khai, nếu bị từ chối sẽ thông báo và yêu cầu chỉnh sửa.



Hình 4.25: Sequence diagram chức năng tương tác cộng đồng

Chương 5: THIẾT KẾ HỆ THỐNG

5.1. Kiến trúc hệ thống tổng quát

Hệ thống giám sát thị trường chứng khoán thời gian thực được xây dựng theo kiến trúc client-server đa tầng, đảm bảo phân tách rõ ràng giữa giao diện, xử lý nghiệp vụ và lưu trữ dữ liệu, đồng thời hỗ trợ cập nhật thông tin tức thời với hiệu năng cao.

Cấu trúc tổng thể bao gồm bốn thành phần chính:

- Frontend (giao diện người dùng): Là lớp tiếp xúc trực tiếp với người dùng cuối, nơi hiển thị dữ liệu thị trường và tiếp nhận tương tác. Frontend đảm nhận vai trò trình bày và truyền đạt thông tin một cách trực quan, thân thiện và phản hồi nhanh, đặc biệt đối với các dữ liệu biến động theo thời gian thực.
- Backend (xử lý nghiệp vụ): Là trung tâm điều phối hoạt động, chịu trách nhiệm xử lý các yêu cầu từ người dùng, thực hiện logic nghiệp vụ, kết nối cơ sở dữ liệu và tương tác với các dịch vụ bên ngoài. Backend đồng thời duy trì kết nối realtime với frontend để đảm bảo việc truyền tải thông tin diễn ra liên tục, không gián đoạn.
- Database (PostgreSQL): Đóng vai trò là nơi lưu trữ dữ liệu tập trung của toàn hệ thống, được thiết kế theo mô hình quan hệ. Hệ quản trị cơ sở dữ liệu này đảm bảo tính toàn vẹn, hỗ trợ truy vấn nhanh, mở rộng tốt và bảo mật dữ liệu hiệu quả.
- Kênh giao tiếp thời gian thực & Dịch vụ bên ngoài: Để đáp ứng yêu cầu realtime đặc thù trong lĩnh vực chứng khoán, hệ thống sử dụng WebSocket làm kênh truyền thông hai chiều giữa server và client. Đồng thời, backend tích hợp với các API bên ngoài như nguồn dữ liệu cổ phiếu, tin tức tài chính và dịch vụ AI phân tích sentiment, nhằm bổ sung thêm chiều sâu cho dữ liệu hiển thị và phân tích.

Tổng thể kiến trúc được thiết kế theo hướng module hóa và mở rộng linh hoạt, giúp hệ thống thích nghi tốt với môi trường dữ liệu biến động liên tục, đáp ứng hiệu quả cả về mặt kỹ thuật lẫn trải nghiệm người dùng.

5.1.1. Các thành phần chính trong kiến trúc

- a) Frontend (Client-side UI)

Frontend là thành phần giao diện người dùng – nơi người dùng cuối trực tiếp tương tác với hệ thống. Đây là lớp đóng vai trò trung gian giữa người dùng và toàn bộ logic xử lý phía backend. Giao diện người dùng được xây dựng bằng các công nghệ web phổ biến như ReactJS, VueJS hoặc HTML/CSS/JavaScript thuần, đảm bảo tính linh hoạt, hiệu suất cao và khả năng tương thích trên nhiều thiết bị khác nhau như desktop, tablet, smartphone.

Frontend chịu trách nhiệm:

- Hiển thị các thông tin thị trường như: giá cổ phiếu theo thời gian thực, biểu đồ kỹ thuật, tin tức tài chính, lịch sử giao dịch, danh mục đầu tư.
- Cung cấp các chức năng tương tác như: tìm kiếm cổ phiếu, gửi bình luận cộng đồng, bỏ phiếu sentiment, đặt lệnh giao dịch, theo dõi người dùng khác.
- Tạo trải nghiệm người dùng thân thiện và trực quan thông qua thiết kế UX/UI hiện đại.

Giao tiếp giữa frontend và backend được thực hiện thông qua hai phương thức:

- RESTful API: sử dụng cho các tác vụ như lấy danh mục đầu tư, thông tin người dùng, lịch sử giao dịch.
- WebSocket (qua Socket.IO): sử dụng để nhận dữ liệu thời gian thực như giá cổ phiếu, cập nhật trạng thái lệnh hoặc bình luận cộng đồng mới mà không cần tải lại trang.

Nhờ cơ chế cập nhật real-time kết hợp API linh hoạt, frontend đảm bảo cung cấp trải nghiệm đầu tư mượt mà và phản hồi nhanh cho người dùng.

b) Backend (Node.js)

Backend là lớp xử lý nghiệp vụ trung tâm của toàn hệ thống. Được xây dựng trên nền tảng Node.js, backend sử dụng kiến trúc bất đồng bộ và hướng sự kiện, rất phù hợp với các ứng dụng thời gian thực có lượng lớn kết nối đồng thời như hệ thống theo dõi thị trường chứng khoán.

Các chức năng chính của backend bao gồm:

- Tiếp nhận và xử lý các yêu cầu từ frontend (qua REST API).
- Kết nối với cơ sở dữ liệu PostgreSQL để truy vấn, lưu trữ và cập nhật dữ liệu.
- Phát sự kiện dữ liệu thời gian thực đến frontend thông qua WebSocket.
- Giao tiếp với các dịch vụ bên ngoài như API giá cổ phiếu, tin tức thị trường.

- Xác thực và phân quyền người dùng thông qua cơ chế bảo mật bằng JWT (JSON Web Token).
- Ghi log hệ thống, kiểm soát lỗi và đảm bảo hiệu suất xử lý ổn định.

Backend được tổ chức thành các module xử lý độc lập, mỗi module phụ trách một nghiệp vụ riêng, bao gồm:

- StockMarketModule: xử lý và cập nhật dữ liệu giá cổ phiếu thời gian thực.
- ChartModule: xây dựng biểu đồ nến và các chỉ báo phân tích kỹ thuật (RSI, MACD, Bollinger Bands).
- PortfolioModule: quản lý danh mục đầu tư, tính toán lãi/lỗ.
- OrderModule: xử lý các thao tác giao dịch như đặt lệnh mua/bán, hủy lệnh.
- CommunityModule: hỗ trợ tương tác người dùng như bình luận, vote xu hướng thị trường.
- UserModule: quản lý thông tin người dùng và phân quyền truy cập.

Việc thiết kế backend theo hướng module hóa giúp hệ thống dễ bảo trì, dễ mở rộng và tăng khả năng tái sử dụng.

c) Cơ sở dữ liệu (PostgreSQL)

Hệ thống sử dụng PostgreSQL – một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, mã nguồn mở – làm nơi lưu trữ toàn bộ dữ liệu có cấu trúc của hệ thống. PostgreSQL được lựa chọn nhờ khả năng xử lý tốt các truy vấn phức tạp, đảm bảo tính toàn vẹn dữ liệu và hỗ trợ mạnh mẽ cho các mô hình dữ liệu lớn.

Các loại dữ liệu được lưu trữ bao gồm:

- Thông tin người dùng: tài khoản, mật khẩu đã mã hóa, vai trò, hồ sơ cá nhân.
- Lịch sử và trạng thái giao dịch: các lệnh mua/bán, thời gian, trạng thái xử lý.
- Dữ liệu cổ phiếu: mã cổ phiếu, giá hiện tại, khối lượng, lịch sử giá.
- Danh mục đầu tư: cổ phiếu yêu thích, số lượng nắm giữ, giá mua, tổng giá trị, lãi/lỗ.
- Tin tức: nội dung bài viết, bài báo của các cổ phiếu.
- Tương tác cộng đồng: bình luận, lượt vote, người theo dõi, nội dung thảo luận.

PostgreSQL cũng hỗ trợ các tính năng như backup dữ liệu định kỳ, phân quyền truy cập, tạo chỉ mục để tăng hiệu suất truy vấn và mở rộng dung lượng linh hoạt theo nhu cầu.

d) Dịch vụ bên ngoài (External APIs)

Hệ thống cần khai thác dữ liệu từ các nguồn bên ngoài để phục vụ việc hiển thị và phân tích. Các dịch vụ bên ngoài bao gồm:

- API dữ liệu giá cổ phiếu thời gian thực: cung cấp thông tin giá mở cửa, giá hiện tại, cao nhất, thấp nhất, khối lượng giao dịch.
- API tin tức thị trường: truy xuất các bài viết, bản tin tài chính liên quan đến từng mã cổ phiếu hoặc toàn ngành.

Các API này có thể được truy vấn định kỳ (polling) hoặc nhận dữ liệu thông qua WebSocket, tùy thuộc vào khả năng kết nối và định dạng dữ liệu của bên thứ ba. Sau khi nhận dữ liệu, backend sẽ xử lý, lọc và lưu vào PostgreSQL và phân phối đến frontend.

e) WebSocket Server

Để đáp ứng yêu cầu cập nhật dữ liệu real-time (thời gian thực) – một đặc trưng quan trọng trong lĩnh vực chứng khoán, hệ thống tích hợp WebSocket server dựa trên nền tảng Socket.IO (tích hợp trong backend Node.js).

Chức năng chính của WebSocket server là:

- Duy trì kết nối hai chiều liên tục giữa client và server, giúp dữ liệu luôn được cập nhật ngay khi có thay đổi mà không cần reload trang.
- Gửi các sự kiện realtime đến người dùng, bao gồm:
 - Giá cổ phiếu thay đổi theo từng giây
 - Bình luận mới trong cộng đồng
 - Trạng thái lệnh giao dịch
 - Biểu quyết vote chiều hướng thị trường
- Giảm tải cho server bằng cách thay thế cơ chế polling liên tục từ client.

Trong trường hợp WebSocket bị ngắt kết nối hoặc không khả dụng, hệ thống được thiết kế để fallback sang cơ chế polling định kỳ, đảm bảo tính liên tục và ổn định trong truyền tải dữ liệu.

5.1.2. Luồng hoạt động của hệ thống

Quy trình vận hành tổng thể của hệ thống giám sát thị trường chứng khoán thời gian thực được triển khai theo mô hình client-server kết hợp cơ chế cập nhật realtime. Cụ thể, các bước hoạt động diễn ra như sau:

(1) Tương tác người dùng và gửi yêu cầu:

Khi người dùng truy cập vào hệ thống, họ sẽ thực hiện các thao tác như tìm kiếm mã cổ phiếu, xem biểu đồ kỹ thuật, theo dõi danh mục đầu tư, gửi bình luận cộng đồng, vote chiều hướng cổ phiếu hoặc đặt lệnh mua/bán cổ phiếu. Những thao tác này được thực hiện thông qua giao diện người dùng (frontend) và được gửi tới backend dưới dạng các yêu cầu HTTP thông qua RESTful API.

(2) Xử lý logic nghiệp vụ tại backend:

Ngay khi nhận được yêu cầu, backend sẽ phân tích và thực hiện các logic nghiệp vụ tương ứng. Tùy vào tính chất của yêu cầu, backend có thể cần xác thực người dùng (kiểm tra token JWT), kiểm tra quyền truy cập, truy vấn dữ liệu từ hệ quản trị cơ sở dữ liệu PostgreSQL hoặc bộ nhớ cache nếu có. Ví dụ, khi người dùng muốn xem thông tin cổ phiếu, backend sẽ lấy giá hiện tại, lịch sử giao dịch và các chỉ báo kỹ thuật từ dữ liệu đã được thu thập và xử lý trước đó.

(3) Trả dữ liệu phản hồi về frontend:

Sau khi xử lý xong, backend sẽ trả kết quả về cho frontend dưới dạng JSON. Dữ liệu này sẽ được frontend tiếp nhận và hiển thị tương ứng trên giao diện người dùng, chẳng hạn như cập nhật lại biểu đồ giá, hiển thị thông tin danh mục hoặc thông báo trạng thái lệnh giao dịch.

(4) Giao tiếp dữ liệu thời gian thực qua WebSocket:

Đối với các dữ liệu yêu cầu cập nhật liên tục – chẳng hạn như giá cổ phiếu theo thời gian thực, phản hồi về lệnh mua/bán, bình luận mới trong cộng đồng hay kết quả vote chiều hướng cổ phiếu – backend sẽ sử dụng cơ chế WebSocket (Socket.IO) để truyền tải dữ liệu trực tiếp đến frontend. Việc sử dụng WebSocket giúp loại bỏ độ trễ do client không cần phải liên tục gửi request để kiểm tra dữ liệu mới. Thay vào đó, server sẽ chủ động đẩy dữ liệu đến client ngay khi có sự thay đổi.

(5) Cơ chế dự phòng bằng polling định kỳ:

Trong một số trường hợp (ví dụ: kết nối WebSocket bị ngắt, trình duyệt không hỗ trợ, hoặc mạng không ổn định), hệ thống sẽ tự động chuyển sang chế độ dự phòng – gọi là polling fallback. Khi đó, frontend sẽ gửi các request API định kỳ (ví dụ mỗi 5 giây/lần) để cập nhật dữ liệu mới, đảm bảo rằng người dùng vẫn được cung cấp thông tin liên tục mà không bị gián đoạn.

(6) Cập nhật đồng bộ đến nhiều người dùng:

Các sự kiện mang tính cộng đồng như gửi bình luận, bỏ phiếu sentiment hoặc theo dõi người dùng khác sẽ không chỉ được phản hồi cho người gửi mà còn được hệ thống tự động phát tán đến tất cả những người dùng khác đang kết nối tới cùng chủ đề (topic). Điều này giúp duy trì một môi trường tương tác liên tục, góp phần xây dựng không gian thảo luận sôi động và phản ánh đúng tính chất thời gian thực của thị trường chứng khoán.

5.1.3. Ưu điểm của kiến trúc

Kiến trúc được lựa chọn và triển khai trong hệ thống giám sát thị trường chứng khoán thời gian thực mang lại nhiều ưu điểm nổi bật, thể hiện rõ tính hiện đại, linh hoạt và sẵn sàng mở rộng trong tương lai.

- Trước hết, việc phân tách rõ ràng giữa ba lớp chính gồm frontend (giao diện người dùng), backend (xử lý nghiệp vụ) và database (lưu trữ dữ liệu) giúp hệ thống dễ dàng quản lý, phát triển và bảo trì. Nhờ cấu trúc phân tầng, mỗi lớp có thể được phát triển độc lập bởi các nhóm khác nhau mà không ảnh hưởng đến nhau, đồng thời hỗ trợ kiểm thử và triển khai hiệu quả hơn. Điều này đặc biệt quan trọng đối với các hệ thống quy mô lớn, có yêu cầu bảo trì thường xuyên hoặc thay đổi tính năng linh hoạt.
- Tiếp theo, việc tích hợp kết nối WebSocket mang lại khả năng cập nhật dữ liệu realtime (thời gian thực), một yêu cầu tối quan trọng đối với lĩnh vực chứng khoán. Người dùng có thể nhận được các thay đổi về giá cổ phiếu, trạng thái lệnh giao dịch hay bình luận cộng đồng gần như ngay lập tức mà không cần tải lại trang. Điều này không chỉ nâng cao trải nghiệm người dùng, mà còn giúp họ phản ứng nhanh hơn với biến động thị trường - yếu tố then chốt trong các quyết định đầu tư.
- Hệ thống cũng được thiết kế với khả năng mở rộng cao, nhờ tổ chức backend theo hướng module hóa. Mỗi module phụ trách một nhóm chức năng riêng như phân tích biểu đồ, xử lý lệnh, quản lý danh mục, tin tức, cộng đồng. Với cách tổ chức này, khi cần mở rộng hệ thống (ví dụ: bổ sung chức năng mới như chatbot tài chính, phân tích AI, hoặc thêm báo cáo phân tích kỹ thuật), có thể triển khai độc lập mà không ảnh hưởng đến hoạt động hiện tại.
- Thêm vào đó, hệ thống có khả năng tích hợp dễ dàng với các dịch vụ bên ngoài, đặc biệt là các API cung cấp dữ liệu chứng khoán, tin tức thị

trường. Việc giao tiếp giữa backend và dịch vụ bên ngoài được chuẩn hóa, đảm bảo rằng khi có thay đổi về nguồn dữ liệu hoặc cập nhật từ nhà cung cấp, hệ thống vẫn hoạt động ổn định mà không cần điều chỉnh nhiều ở tầng frontend hay database.

5.2. Thiết kế cơ sở dữ liệu (PortgreSQL)

5.2.1. Mục tiêu thiết kế

Thiết kế cơ sở dữ liệu là một thành phần cốt lõi trong quá trình phát triển hệ thống phần mềm, đóng vai trò nền tảng trong việc tổ chức, lưu trữ, truy xuất và xử lý dữ liệu một cách có hệ thống, chính xác, linh hoạt và hiệu quả. Đây không chỉ là bước kỹ thuật đơn thuần mà còn là yếu tố quyết định trực tiếp đến khả năng hoạt động ổn định, mở rộng và hiệu năng tổng thể của toàn bộ hệ thống.

Đối với hệ thống giám sát và giao dịch chứng khoán thời gian thực, nơi mà dữ liệu thị trường luôn biến động theo từng giây, yêu cầu đặt ra cho cơ sở dữ liệu là rất cao. Hệ thống phải có khả năng xử lý và lưu trữ khối lượng lớn dữ liệu trong thời gian ngắn, đảm bảo độ tin cậy và chính xác cao trong quá trình đồng bộ. Cơ sở dữ liệu cần hỗ trợ lưu trữ liên tục thông tin giá cổ phiếu, lịch sử giao dịch, đồng thời đáp ứng truy vấn nhanh chóng để phục vụ việc hiển thị biểu đồ, cập nhật giá và phản hồi các hành vi thao tác của người dùng theo thời gian thực.

Không chỉ đóng vai trò là nơi lưu trữ dữ liệu đơn thuần, cơ sở dữ liệu còn là điểm kết nối giữa các chức năng quan trọng trong hệ thống như: quản lý người dùng, xử lý các lệnh mô phỏng, quản lý danh sách theo dõi cổ phiếu, gửi thông báo hệ thống và ghi nhận phản hồi người dùng. Mọi thao tác, tương tác và dữ liệu phát sinh trong quá trình vận hành đều phải được ghi nhận và tổ chức hợp lý thông qua mô hình dữ liệu được thiết kế chặt chẽ.

Chính vì vậy, việc thiết kế cơ sở dữ liệu cần được thực hiện một cách khoa học, có định hướng mở rộng trong dài hạn và đảm bảo tuân thủ các nguyên tắc chuẩn hóa dữ liệu, đồng thời vẫn đáp ứng được tốc độ truy xuất nhanh và khả năng xử lý đồng thời trong môi trường nhiều người dùng.

Mục tiêu chính của quá trình thiết kế cơ sở dữ liệu bao gồm:

- Xây dựng mô hình dữ liệu rõ ràng, mạch lạc phản ánh chính xác các thực thể và mối quan hệ trong hệ thống, đồng thời dễ dàng mở rộng khi bổ sung chức năng mới hoặc tăng quy mô người dùng.

- Đảm bảo tính toàn vẹn, nhất quán và tin cậy của dữ liệu, hạn chế tối đa tình trạng dư thừa, trùng lặp hoặc sai lệch thông tin giữa các bảng.
- Tối ưu hiệu năng truy vấn và lưu trữ dữ liệu realtime, đặc biệt đối với các bảng có tốc độ cập nhật cao như dữ liệu giá cổ phiếu và lịch sử mua bán lệnh.
- Tạo nền tảng ổn định và tương thích cao với các thành phần backend được xây dựng bằng Node.js, đồng thời hỗ trợ tốt cho việc tích hợp API và các dịch vụ mở rộng khác như thống kê, phân tích.

Nhìn chung, thiết kế cơ sở dữ liệu trong hệ thống không chỉ là bước trung gian giữa phân tích và lập trình, mà còn là một phần kiến trúc quan trọng định hình khả năng phát triển lâu dài, khả năng mở rộng và hiệu quả vận hành của toàn bộ hệ thống giám sát chứng khoán thời gian thực.

5.2.2. Mô hình khái niệm

Dựa trên kết quả phân tích nghiệp vụ và các yêu cầu chức năng, mô hình khái niệm của hệ thống được xây dựng nhằm phản ánh đầy đủ các thực thể dữ liệu cốt lõi, các thuộc tính quan trọng và mối quan hệ giữa chúng. Mô hình này đóng vai trò nền tảng để phát triển chi tiết mô hình quan hệ dữ liệu (ERD) và triển khai cơ sở dữ liệu thực tế.

Các thực thể chính được xác định bao gồm:

- **Người dùng (User)**

Là thực thể trung tâm quản lý mọi thông tin định danh, liên hệ, xác thực đăng nhập và quyền truy cập của các nhóm người dùng trong hệ thống (nhà đầu tư, quản trị viên, khách).

→ Người dùng có thể thực hiện các lệnh giao dịch, lưu trữ thông tin ví điện tử, đăng bài phân tích, bình luận và quản lý danh sách công ty yêu thích.

- **Thẻ căn cước (Identity Card)**

Lưu trữ thông tin cá nhân chi tiết (họ tên, địa chỉ, ngày sinh) nhằm xác minh danh tính người dùng, đồng thời liên kết đến các nghiệp vụ khác như giao dịch, bài viết, ví điện tử để đảm bảo tính xác thực.

→ Tăng mức độ bảo mật và tính hợp lệ của dữ liệu.

- **Vai trò (Role)**

Quản lý quyền hạn và phân loại người dùng theo các nhóm vai trò cụ thể (ví dụ: admin, trader, guest).

→ Kiểm soát quyền truy cập hệ thống và chức năng tương ứng với từng loại người dùng.

- **Ví điện tử (Wallet)**

Ghi nhận thông tin thẻ ngân hàng hoặc ví điện tử liên kết với người dùng để phục vụ các giao dịch tài chính như nạp rút vốn.

→ Kết nối chặt chẽ với thông tin người dùng và thẻ căn cước để đảm bảo an toàn giao dịch.

- **Bài viết (Post)**

Cho phép người dùng đăng tải các bài viết phân tích, chia sẻ quan điểm và thông tin thị trường. Mỗi bài viết có thể được xác minh thông qua thẻ căn cước.

→ Tạo môi trường trao đổi và chia sẻ thông tin giữa các nhà đầu tư.

- **Bình luận (Comment)**

Ghi nhận các phản hồi, thảo luận của người dùng đối với các bài viết.

→ Tăng tính tương tác cộng đồng và hỗ trợ trao đổi kiến thức.

- **Lệnh giao dịch (Order)**

Phản ánh các lệnh mua/bán cổ phiếu được người dùng thực hiện. Mỗi lệnh lưu thông tin chi tiết như loại lệnh, số lượng, giá, ngày thực hiện và trạng thái.

→ Là cơ sở để phân tích lịch sử giao dịch và tính toán hiệu quả đầu tư.

- **Danh sách yêu thích (Favorite)**

Lưu thông tin về các công ty mà người dùng đã đánh dấu để theo dõi.

→ Hỗ trợ cá nhân hóa giao diện và đưa ra cảnh báo, cập nhật phù hợp với nhu cầu người dùng.

- **Công ty (Company)**

Lưu thông tin chi tiết của các công ty phát hành cổ phiếu, bao gồm tên công ty, mã số thuế, địa chỉ, và tham chiếu đến sàn giao dịch.

→ Là thực thể nền tảng để quản lý cổ phiếu phát hành.

- **Mã cổ phiếu (Stock)**

Quản lý dữ liệu giá trị thị trường của các mã cổ phiếu theo từng công ty, bao gồm giá mở cửa, giá đóng cửa, giá cao nhất, giá thấp nhất.

→ Phục vụ hiển thị biểu đồ giá, theo dõi biến động và làm cơ sở cho các lệnh giao dịch.

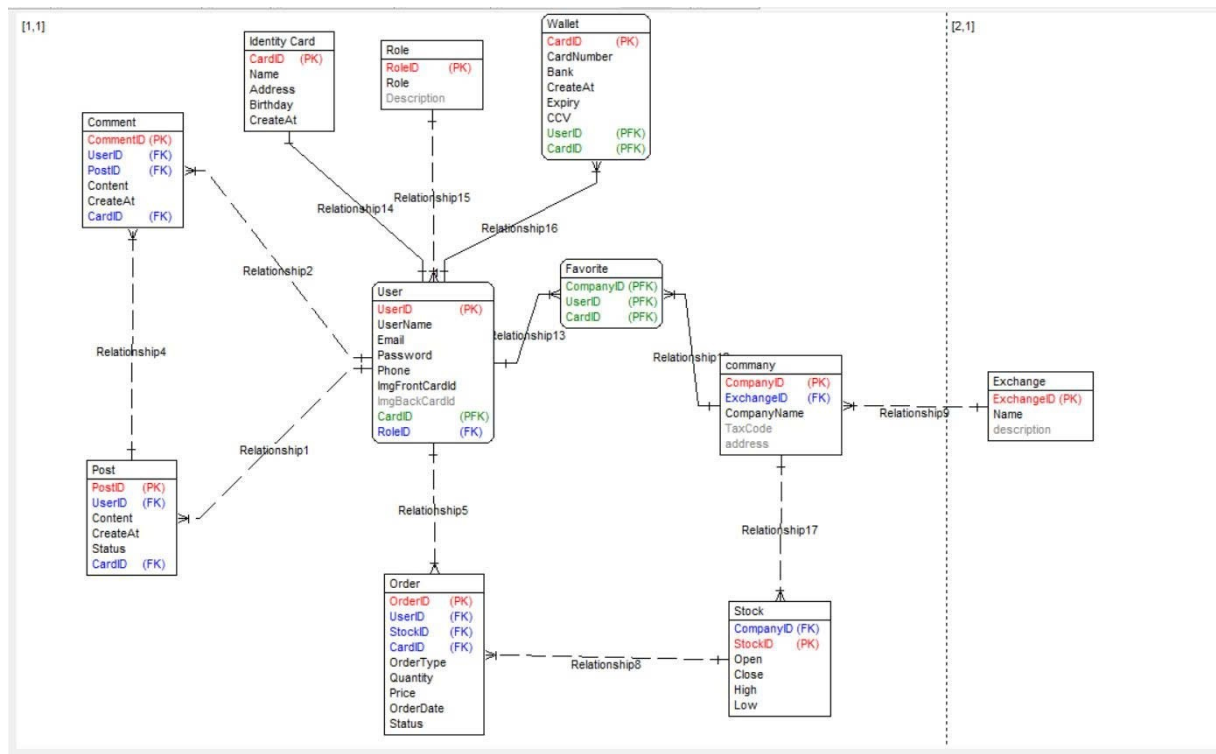
- **Sàn giao dịch (Exchange)**

Quản lý thông tin về các sàn chứng khoán nơi công ty niêm yết cổ phiếu, bao gồm tên và mô tả sàn.

→ Hỗ trợ tổ chức dữ liệu cổ phiếu theo từng sàn để thuận tiện cho việc tra cứu và quản lý.

5.2.3. Mô hình quan hệ dữ liệu (ERD)

Dựa trên kết quả phân tích nghiệp vụ và thiết kế logic hệ thống, cơ sở dữ liệu được tổ chức theo mô hình quan hệ, đảm bảo tính toàn vẹn và phục vụ các chức năng chính của hệ thống giám sát và giao dịch cổ phiếu. Mỗi bảng dữ liệu phản ánh trực tiếp một thực thể quan trọng, với các ràng buộc khóa chính – khóa ngoại rõ ràng để tối ưu lưu trữ và truy vấn.



Hình 5.1: Mô hình ERD

Hệ thống gồm các bảng dữ liệu chính, mô tả cụ thể như sau:

1. User – Người dùng

- **Thuộc tính:**

- UserID (PK): Định danh duy nhất.
- UserName, Email, Password, Phone, ImgFrontCardId, ImgBackCardId, CardID (FK), RoleID (FK).

- **Chức năng:** Quản lý thông tin người dùng, xác thực đăng nhập, gán quyền và liên kết thông tin cá nhân.
- **Quan hệ:**
 - 1–n với Post, Comment, Order, Wallet.
 - n–n với Company thông qua Favorite.
 - n–1 với Role, Identity Card.

2. Identity Card – Thẻ căn cước

- **Thuộc tính:**
 - CardID (PK): Định danh thẻ.
 - Name, Address, Birthday, CreateAt.
- **Chức năng:** Lưu trữ thông tin cá nhân để xác minh danh tính người dùng.
- **Quan hệ:**
 - 1–n với User, Comment, Post, Order, Wallet.

3. Role – Vai trò

- **Thuộc tính:**
 - RoleID (PK): Định danh vai trò.
 - Role, Description.
- **Chức năng:** Xác định phân quyền truy cập của người dùng (ví dụ: admin, trader, guest).
- **Quan hệ:**
 - 1–n với User.

4. Wallet – Ví điện tử

- **Thuộc tính:**
 - CardID (PK): Định danh ví (cũng là thẻ).
 - CardNumber, Bank, CreateAt, Expiry, CCV, UserID (FK), CardID (FK) (liên kết thông tin cá nhân).
- **Chức năng:** Quản lý thông tin thẻ ngân hàng/ví điện tử gắn với người dùng để phục vụ các giao dịch tài chính.
- **Quan hệ:**
 - n–1 với User, Identity Card.

5. Post – Bài viết

- **Thuộc tính:**

- PostID (PK): Định danh bài viết.
- UserID (FK), CardID (FK), Content, CreateAt, Status.
- **Chức năng:** Cho phép người dùng đăng bài phân tích thị trường, chia sẻ ý kiến.
- **Quan hệ:**
 - 1–n với Comment.
 - n–1 với User, Identity Card.

6. Comment – Bình luận

- **Thuộc tính:**
 - CommentID (PK): Định danh bình luận.
 - UserID (FK), PostID (FK), CardID (FK), Content, CreateAt.
- **Chức năng:** Cho phép người dùng trao đổi, phản hồi các bài viết.
- **Quan hệ:**
 - n–1 với Post, User, Identity Card.

7. Order – Lệnh giao dịch

- **Thuộc tính:**
 - OrderID (PK): Định danh lệnh.
 - UserID (FK), StockID (FK), CardID (FK), OrderType, Quantity, Price, OrderDate, Status.
- **Chức năng:** Lưu thông tin lệnh mua/bán cổ phiếu của người dùng.
- **Quan hệ:**
 - n–1 với User, Stock, Identity Card.

8. Favorite – Danh sách công ty yêu thích

- **Thuộc tính:**
 - CompanyID (PFK), UserID (PFK), CardID (PFK) (khóa phức hợp).
- **Chức năng:** Cho phép người dùng đánh dấu các công ty yêu thích để theo dõi.
- **Quan hệ:**
 - n–n giữa User và Company (thông qua Favorite).
 - Liên kết Identity Card bổ sung.

9. Company – Công ty

- **Thuộc tính:**
 - CompanyID (PK): Định danh công ty.
 - ExchangeID (FK), CompanyName, TaxCode, Address.

- **Chức năng:** Lưu thông tin công ty phát hành cổ phiếu.
- **Quan hệ:**
 - 1–n với Stock.
 - n–1 với Exchange.

10. Stock – Mã cổ phiếu

- **Thuộc tính:**
 - StockID (PK): Định danh cổ phiếu.
 - CompanyID (FK), Open, Close, High, Low.
- **Chức năng:** Quản lý dữ liệu giá cổ phiếu theo công ty.
- **Quan hệ:**
 - n–1 với Company.
 - 1–n với Order.

11. Exchange – Sàn giao dịch

- **Thuộc tính:**
 - ExchangeID (PK): Định danh sàn.
 - Name, Description.
- **Chức năng:** Lưu thông tin sàn chứng khoán niêm yết cổ phiếu.
- **Quan hệ:**
 - 1–n với Company.

5.2.4. Chuẩn hóa dữ liệu

Trong quá trình thiết kế cơ sở dữ liệu, việc chuẩn hóa dữ liệu là một bước quan trọng nhằm đảm bảo dữ liệu được lưu trữ một cách hợp lý, không dư thừa, nhất quán và dễ bảo trì. Hệ thống được chuẩn hóa đến mức Third Normal Form (3NF) – một cấp độ chuẩn hóa phổ biến trong thiết kế cơ sở dữ liệu quan hệ, đặc biệt phù hợp với hệ thống có khối lượng dữ liệu lớn và yêu cầu truy vấn realtime cao như trong bối cảnh thị trường chứng khoán.

Việc chuẩn hóa dữ liệu đến 3NF trong hệ thống này đảm bảo các nguyên tắc sau:

- **Loại bỏ dư thừa dữ liệu**
Các thuộc tính trong từng bảng được phân tách rõ ràng, không lặp lại giữa các bảng khác. Ví dụ, thông tin về người dùng (FullName, Email, Role) chỉ lưu trong bảng Users và được liên kết đến các bảng như Orders, Favorites, Feedbacks thông qua khóa ngoại (UserID). Tương tự, thông tin về cổ phiếu chỉ

lưu một lần trong bảng Stocks và được tham chiếu ở StockPrices, Orders, Favorites.

- **Đảm bảo toàn vẹn dữ liệu và mối quan hệ logic**

Mọi mối quan hệ giữa các bảng đều được xác định rõ ràng thông qua khóa chính và khóa ngoại.

- Orders liên kết UserID → Users, Symbol → Stocks.
- StockPrices liên kết Symbol → Stocks.
- Favorites là bảng trung gian giữa Users và Stocks với khóa chính tổ hợp (UserID, Symbol).
- Các bảng như Notifications và Feedbacks đều tham chiếu tới Users thông qua khóa ngoại.

Cách thiết kế này đảm bảo rằng dữ liệu luôn nhất quán giữa các bảng, không phát sinh sai lệch trong quá trình cập nhật hoặc xóa dữ liệu.

- **Hỗ trợ khả năng mở rộng và bảo trì hệ thống**

Với mô hình đã chuẩn hóa, việc bổ sung thuộc tính mới cho các bảng như Users (ví dụ: số điện thoại, địa chỉ) hoặc mở rộng bảng Orders để lưu trạng thái đơn hàng, ngày khớp lệnh. Có thể thực hiện dễ dàng mà không ảnh hưởng đến cấu trúc dữ liệu tổng thể.

Đồng thời, mô hình chuẩn hóa cũng giúp việc bảo trì, sửa đổi hoặc tối ưu truy vấn trở nên đơn giản hơn trong dài hạn.

- **Tối ưu hiệu suất truy vấn với dữ liệu lớn**

Các bảng có dữ liệu thay đổi thường xuyên và quy mô lớn như StockPrices và Orders được tách biệt rõ ràng với bảng tham chiếu như Stocks, Users. Nhờ đó, hệ thống chỉ lưu trữ những trường thông tin cần thiết trong mỗi bảng, giúp giảm tải bộ nhớ và cải thiện tốc độ truy vấn – đặc biệt quan trọng trong các thao tác realtime như:

- Hiển thị biểu đồ giá (lấy dữ liệu từ StockPrices)
- Lọc lịch sử giao dịch theo người dùng (từ Orders)
- Hiển thị các mã cổ phiếu theo dõi (từ Favorites)

→ Việc chuẩn hóa cơ sở dữ liệu đến chuẩn 3NF trong hệ thống không chỉ đảm bảo tính nhất quán và dễ mở rộng mà còn hỗ trợ tối ưu hiệu năng hệ thống trong môi trường hoạt động liên tục, nhiều người dùng và dữ liệu biến động cao như thị trường

chứng khoán. Đây cũng là tiền đề quan trọng để triển khai các chiến lược tối ưu hóa truy vấn, phân vùng dữ liệu và tích hợp hệ thống backend trong các giai đoạn tiếp theo.

5.2.5. Định hướng triển khai vật lý

Trong giai đoạn triển khai thực tế, hệ thống cơ sở dữ liệu được định hướng xây dựng trên nền tảng PostgreSQL – một hệ quản trị cơ sở dữ liệu mã nguồn mở mạnh mẽ, đáng tin cậy và phù hợp với các ứng dụng có yêu cầu xử lý dữ liệu quy mô lớn, độ chính xác cao và khả năng mở rộng tốt. Lựa chọn này không chỉ dựa trên tính phổ biến và tính năng kỹ thuật của PostgreSQL, mà còn phù hợp với kiến trúc tổng thể của hệ thống sử dụng backend là Node.js, kết nối thông qua các RESTful API, WebSocket và tương thích với các dịch vụ realtime khác.

Cụ thể, các định hướng triển khai vật lý như sau:

a) Hiệu năng xử lý dữ liệu khối lượng lớn

Hệ thống được thiết kế để giám sát giá cổ phiếu theo thời gian thực, với khối lượng dữ liệu tăng nhanh theo từng phút, từng giây. Bảng StockPrices có thể lưu trữ hàng triệu bản ghi mỗi ngày với dữ liệu OHLC và khối lượng giao dịch cho hàng trăm mã cổ phiếu.

PostgreSQL được lựa chọn vì khả năng xử lý hiệu quả dữ liệu quy mô lớn, hỗ trợ index hóa nâng cao, bộ nhớ đệm hiệu suất cao và các cơ chế tối ưu hóa truy vấn phức tạp mà vẫn đảm bảo tính toàn vẹn dữ liệu. Đây là yếu tố then chốt trong các thao tác như hiển thị biểu đồ, lọc lịch sử giao dịch và truy vấn danh sách cổ phiếu yêu thích.

b) Hỗ trợ phân vùng bảng (Table Partitioning)

PostgreSQL cho phép phân vùng bảng dựa trên giá trị cột (partition by range, list, hash). Trong hệ thống, bảng StockPrices sẽ được phân vùng theo ngày (DATE) hoặc theo mã cổ phiếu (Symbol) nhằm:

- Giảm kích thước từng phân vùng → tăng tốc độ truy vấn và cập nhật.
- Tối ưu truy vấn theo thời gian (ví dụ: hiển thị biểu đồ 1 ngày, 1 tuần, 1 tháng).
- Dễ dàng xóa, lưu trữ hoặc sao lưu dữ liệu cũ mà không ảnh hưởng toàn bộ bảng.

Ví dụ: mỗi ngày hoặc mỗi mã cổ phiếu sẽ là một phân vùng riêng (stock_prices_2025_06_10, stock_prices_VNM)

c) Khả năng tích hợp với hệ sinh thái Node.js

Cơ sở dữ liệu PostgreSQL có thể tích hợp mượt mà với backend sử dụng Node.js thông qua nhiều thư viện và driver hỗ trợ như pg, knex.js hoặc sequelize, hỗ trợ:

- Giao tiếp qua API REST/GraphQL
- Xử lý các truy vấn bất đồng bộ
- Quản lý transaction, pool kết nối và tối ưu truy cập đồng thời

Ngoài ra, PostgreSQL có thể dễ dàng kết nối với các công cụ xử lý frontend realtime như Socket.IO để phục vụ các yêu cầu hiển thị giá chứng khoán, thông báo và phản hồi người dùng theo thời gian thực.

d) Đảm bảo an toàn và bảo mật dữ liệu

Với đặc thù dữ liệu người dùng nhạy cảm (email, mật khẩu, quyền truy cập), hệ thống cần áp dụng các cơ chế bảo mật chặt chẽ:

- Mã hóa mật khẩu sử dụng hàm băm (bcrypt) phía backend, không lưu mật khẩu thô trong cơ sở dữ liệu.
- Xác thực truy cập thông qua hệ thống phân quyền (role-based access control), được định nghĩa trong bảng Users.
- Giao thức SSL/TLS và cơ chế phân quyền truy cập cấp bảng (GRANT/REVOKE) của PostgreSQL đảm bảo chỉ những thành phần hệ thống được ủy quyền mới có quyền thao tác trên dữ liệu.

e) Hỗ trợ mở rộng và tích hợp hệ thống phụ trợ

Để phục vụ cho việc mở rộng hệ thống trong tương lai, cơ sở dữ liệu cần đảm bảo khả năng kết nối với các thành phần lưu trữ hoặc phân tích khác như:

- Redis: dùng làm bộ nhớ tạm để cache dữ liệu phổ biến (ví dụ: top 10 mã cổ phiếu theo dõi nhiều nhất).
- Kafka hoặc RabbitMQ: dùng để đồng bộ các sự kiện realtime như giá cổ phiếu thay đổi, đơn hàng được khớp.
- Hệ thống lưu trữ thứ cấp (Data Lake hoặc Backup Server): dùng để lưu trữ dữ liệu lâu dài, phục vụ mục tiêu phân tích nâng cao, thống kê, hoặc khôi phục khi cần thiết.

Định hướng triển khai cơ sở dữ liệu trên nền PostgreSQL giúp hệ thống đạt được sự cân bằng giữa hiệu năng, độ ổn định, tính bảo mật và khả năng mở rộng. Việc tổ chức dữ liệu hợp lý kết hợp với các chiến lược tối ưu vật lý như phân vùng bảng, chỉ mục nâng cao và cơ chế bảo mật tích hợp sẽ đảm bảo hệ thống hoạt động hiệu

quả trong môi trường thực tế, đáp ứng tốt nhu cầu người dùng và khả năng tăng trưởng trong tương lai.

5.2.6. Kết luận

Thiết kế cơ sở dữ liệu là giai đoạn quan trọng giúp hiện thực hóa toàn bộ các phân tích nghiệp vụ và cấu trúc logic của hệ thống đã được xây dựng trước đó. Trong hệ thống giám sát và giao dịch chứng khoán cộng đồng, cơ sở dữ liệu đóng vai trò trung tâm trong việc tổ chức, lưu trữ và đảm bảo luồng dữ liệu vận hành ổn định giữa các thành phần như quản lý người dùng, danh mục cổ phiếu, giá chứng khoán thời gian thực, đơn hàng mô phỏng, danh sách theo dõi, thông báo và phản hồi.

Cơ sở dữ liệu của hệ thống đã được thiết kế theo hướng modul hóa, mỗi bảng thể hiện một thực thể riêng biệt với chức năng rõ ràng, kết nối với nhau thông qua các mối quan hệ khóa chính – khóa ngoại. Đồng thời, mô hình dữ liệu tuân thủ chuẩn hóa đến mức Third Normal Form (3NF) nhằm loại bỏ dư thừa, tăng tính toàn vẹn và tối ưu hóa hiệu suất truy vấn, đặc biệt với các bảng có khối lượng dữ liệu lớn như StockPrices và Orders.

Sự lựa chọn PostgreSQL làm hệ quản trị cơ sở dữ liệu giúp hệ thống đáp ứng tốt các yêu cầu về:

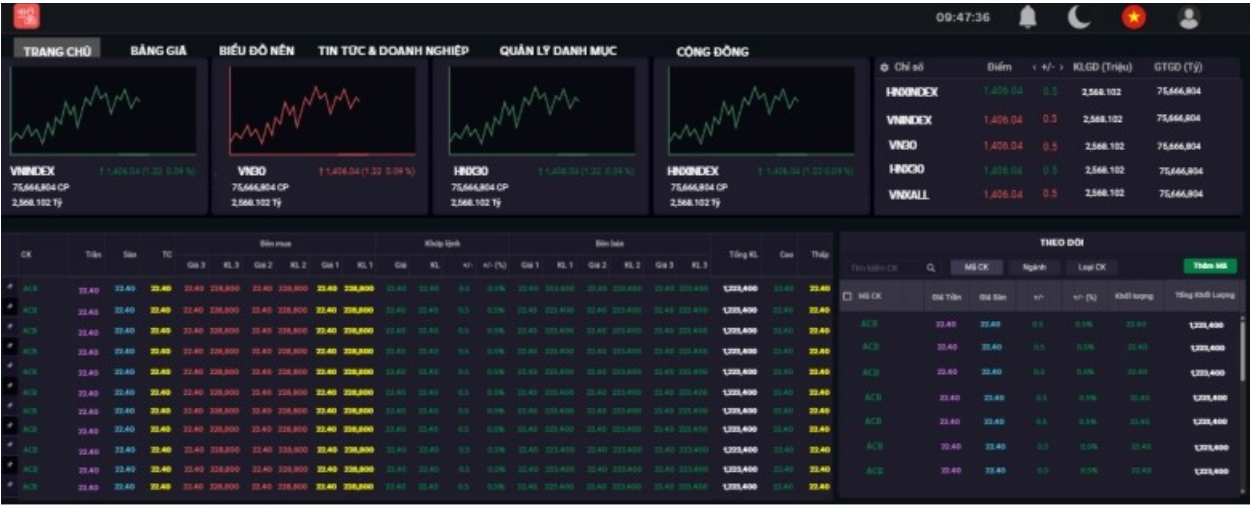
- Hiệu năng truy vấn dữ liệu realtime, phục vụ các thao tác hiển thị biểu đồ giá, thống kê giao dịch và phản hồi người dùng theo thời gian thực.
- Khả năng phân vùng dữ liệu linh hoạt, hỗ trợ xử lý hiệu quả dữ liệu tăng trưởng nhanh theo thời gian.
- Tích hợp mượt mà với backend Node.js, đảm bảo luồng xử lý ổn định, kết nối dễ dàng với các RESTful API và dịch vụ WebSocket.
- Bảo mật và kiểm soát truy cập chặt chẽ, phân quyền rõ ràng theo vai trò người dùng (admin/user) và mã hóa thông tin nhạy cảm.

Thiết kế cơ sở dữ liệu của hệ thống không chỉ đáp ứng đầy đủ các yêu cầu chức năng hiện tại mà còn sẵn sàng hỗ trợ mở rộng quy mô người dùng, bổ sung tính năng mới và tích hợp với các hệ thống phụ trợ trong tương lai. Đây chính là nền tảng vững chắc để chuyển sang các giai đoạn triển khai tiếp theo trong vòng đời phát triển phần mềm, đảm bảo tính bền vững và hiệu quả cho toàn hệ thống.

5.3. Thiết kế giao diện người dùng

5.3.1. Giao diện Trang chủ

Giao diện “Trang chủ” đóng vai trò là trung tâm tổng quan của toàn bộ hệ thống, cung cấp cho người dùng cái nhìn nhanh, trực quan và toàn diện về thị trường chứng khoán Việt Nam trong thời gian thực. Đây là giao diện mặc định được hiển thị ngay khi người dùng vào trang web. Mục tiêu chính của trang này là hỗ trợ người dùng theo dõi biến động của các chỉ số thị trường quan trọng như VNINDEX, VN30, HNXINDEX và VNXALL, đồng thời giúp nắm bắt xu hướng mua bán thông qua màu sắc và biểu đồ giá thay đổi liên tục. Ngoài ra, trang chủ còn tích hợp bảng giá cổ phiếu theo thời gian thực cùng các lỗi tắt giúp truy cập nhanh vào những chức năng chuyên sâu như phân tích kỹ thuật, tin tức tài chính, cộng đồng và danh mục đầu tư cá nhân.



Hình 5.2: Giao diện trang chủ

Trang sẽ được bố trí thành bốn khu vực chức năng chính, mỗi khu vực đảm nhiệm một vai trò riêng biệt nhưng được kết nối chặt chẽ để tạo nên một trải nghiệm thống nhất và hiệu quả. Ở phía trên cùng là thanh điều hướng cố định, trong đó bên trái là logo và tiêu đề của hệ thống, còn bên phải là các nút chức năng như đồng hồ thời gian thực, chuyển đổi chế độ giao diện (sáng/tối), chọn ngôn ngữ hiển thị, thông báo hệ thống, cùng biểu tượng avatar dẫn đến khu vực tài khoản người dùng. Thanh menu trung tâm cũng bao gồm các liên kết đến những trang chức năng chính như bảng giá, biểu đồ nền, tin tức và doanh nghiệp, quản lý danh mục và cộng đồng.

Ngay bên dưới thanh điều hướng là khu vực biểu đồ chỉ số thị trường, nơi hiển thị các biểu đồ đường nhỏ tương ứng với các chỉ số lớn của thị trường. Mỗi biểu đồ hiển thị theo thời gian thực, thể hiện rõ biến động trong phiên, điểm số hiện tại, phần trăm thay đổi cũng như khối lượng và giá trị giao dịch. Người dùng có thể di chuột để xem giá tại từng thời điểm cụ thể và nhấn vào từng biểu đồ để chuyển đến phần phân tích

chi tiết hơn. Ở phía bên phải khu vực biểu đồ là bảng tổng hợp chỉ số, trình bày dữ liệu cô đọng của các chỉ số chính như điểm hiện tại, thay đổi điểm số, phần trăm tăng/giảm, khối lượng và giá trị giao dịch, cũng như số lượng cổ phiếu tăng hoặc giảm trong từng rõ chỉ số. Các thông tin này được mã hóa màu sắc trực quan: xanh lá biểu thị tăng, đỏ biểu thị giảm và vàng hoặc xám thể hiện trạng thái không đổi.

Phần lớn không gian trung tâm của giao diện được dành cho bảng giá cổ phiếu thời gian thực. Bảng này cung cấp thông tin chi tiết về từng mã cổ phiếu trên toàn thị trường hoặc có thể lọc theo từng sàn như HOSE, HNX hoặc UPCOM, ngoài ra cũng có thể tìm kiếm nâng cao với giá, khối lượng, vốn hoá và ngành. Mỗi dòng trong bảng bao gồm các thông tin như mã cổ phiếu, giá trần, giá sàn, giá tham chiếu, giá mua và bán tốt nhất, giá khớp lệnh gần nhất, khối lượng giao dịch và trạng thái tăng/giảm. Việc hiển thị trạng thái thị trường được trực quan hóa bằng màu sắc: tím cho giá trần, xanh lam cho giá sàn, xanh lá cho tăng giá, đỏ cho giảm giá và vàng cho trạng thái không thay đổi. Dữ liệu bảng được cập nhật liên tục thông qua kết nối WebSocket để đảm bảo người dùng nhận được thông tin mới nhất mà không cần tải lại trang.

Giao diện cũng cho phép người dùng tương tác linh hoạt, như nhấn vào biểu đồ để xem chi tiết, rê chuột để xem dữ liệu lịch sử hoặc chọn mã cổ phiếu để mở trang phân tích chi tiết tương ứng. Ngoài ra, hệ thống hỗ trợ cá nhân hóa bằng cách cho phép người dùng thêm cổ phiếu vào danh sách theo dõi và tùy chỉnh giao diện theo sở thích, bao gồm việc chuyển đổi giữa chế độ sáng và tối cũng như lựa chọn ngôn ngữ sử dụng. Tất cả những yếu tố này kết hợp lại tạo nên một giao diện trang chủ chuyên nghiệp, hiện đại, dễ sử dụng và phục vụ tối ưu cho nhu cầu theo dõi thị trường của người dùng.

5.3.2. Giao diện Đăng nhập người dùng

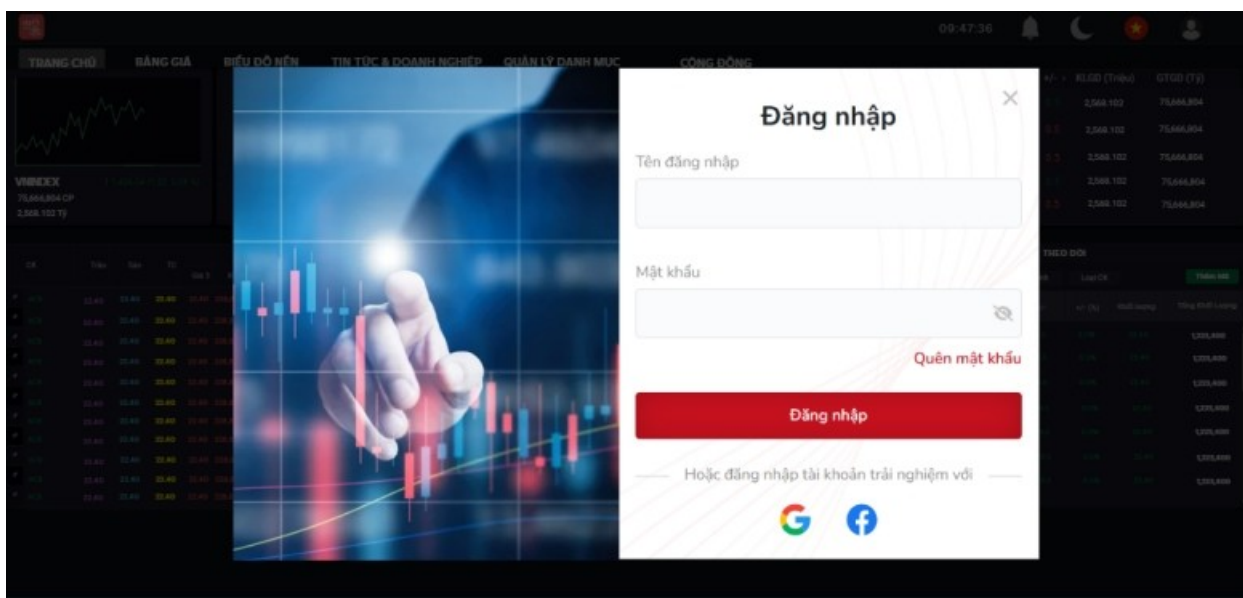
Khung đăng nhập là cổng xác thực đầu tiên cho phép người dùng truy cập vào hệ thống. Đây là bước thiết yếu nhằm đảm bảo tính riêng tư, bảo mật dữ liệu và tạo tiền đề cho quá trình cá nhân hóa trải nghiệm sử dụng. Ngoài phương thức đăng nhập truyền thống qua tên đăng nhập và mật khẩu, hệ thống còn hỗ trợ đăng nhập nhanh thông qua các nền tảng mạng xã hội như Google hoặc Facebook, giúp người dùng tiết kiệm thời gian và thao tác.

Giao diện này thường được kích hoạt khi người dùng truy cập vào các chức năng yêu cầu xác thực, chẳng hạn như quản lý danh mục hoặc tương tác trong cộng đồng hoặc

khi người dùng nhấn vào biểu tượng tài khoản trong thanh điều hướng nếu chưa đăng nhập. Giao diện được hiển thị dưới dạng cửa sổ nổi, đè trực tiếp lên trang chính, cho phép thực hiện thao tác xác thực mà không cần tải lại trang.

Thiết kế giao diện được chia theo bố cục hai cột theo chiều ngang. Bên trái là phần hình ảnh minh họa với hình bàn tay chạm vào biểu đồ chứng khoán, biểu trưng cho hoạt động đầu tư số và sự hiện diện của công nghệ trong thị trường tài chính hiện đại. Hình ảnh này không chỉ mang tính trang trí mà còn giúp tăng sự trực quan và chuyên nghiệp cho trang đăng nhập. Bên phải là khối biểu mẫu đăng nhập, nơi người dùng nhập tên đăng nhập và mật khẩu. Giao diện hỗ trợ biểu tượng hiển thị/ẩn mật khẩu nhằm tăng tính bảo mật và tiện lợi. Ngoài ra, người dùng có thể nhấn vào liên kết “Quên mật khẩu?” để thực hiện thao tác khôi phục tài khoản nếu cần thiết.

Ngay dưới biểu mẫu truyền thống là các lựa chọn đăng nhập bằng Google hoặc Facebook, được hiển thị thông qua biểu tượng thương hiệu quen thuộc, giúp việc đăng nhập trở nên nhanh chóng và thuận tiện hơn. Giao diện sử dụng phong cách thiết kế hiện đại, màu sắc tương phản rõ ràng, font chữ dễ đọc và bố cục hợp lý. Tất cả các thành phần như ô nhập liệu, nút hành động và biểu tượng đều được thiết kế nhất quán và tối ưu cho trải nghiệm người dùng.



Hình 5.3: Giao diện đăng nhập

Khi người dùng điền thông tin và nhấn “Đăng nhập”, hệ thống sẽ tiến hành kiểm tra xác thực. Nếu thông tin không chính xác, một thông báo lỗi rõ ràng sẽ được hiển thị để người dùng biết và thao tác lại. Nếu đăng nhập thành công, hệ thống sẽ tự động chuyển hướng người dùng đến trang chủ hoặc khu vực người dùng tương ứng trong hệ

thông. Nhìn chung, giao diện đăng nhập không chỉ đơn giản, dễ sử dụng mà còn được tối ưu hóa cho tốc độ, bảo mật và thân thiện với mọi loại người dùng, từ người mới đến nhà đầu tư chuyên nghiệp.

5.3.3. Giao diện Thông tin người dùng

Giao diện “Thông tin người dùng” đóng vai trò là trung tâm quản lý hồ sơ cá nhân trong hệ thống. Đây là nơi người dùng có thể xem và chỉnh sửa thông tin cơ bản của mình, thực hiện thao tác đổi mật khẩu cũng như điều chỉnh các tùy chọn cá nhân hóa như ngôn ngữ, giao diện và thông báo. Giao diện này được thiết kế dưới dạng cửa sổ nổi toàn màn hình, hiển thị khi người dùng nhấn vào biểu tượng tài khoản ở góc trên bên phải màn hình.

Toàn bộ giao diện được chia thành ba khối chức năng chính. Khối bên trái là biểu mẫu hiển thị thông tin hồ sơ cá nhân, bao gồm họ tên, ngày sinh, giới tính, quốc tịch, nghề nghiệp, số căn cước công dân, ngày cấp, nơi cấp, số điện thoại, email và địa chỉ liên hệ. Một số trường như họ tên, số CCCD hay ngày sinh được cố định và không thể chỉnh sửa nhằm đảm bảo tính nhất quán dữ liệu cá nhân. Dưới biểu mẫu là nút “Cập nhật thông tin” cho phép người dùng lưu lại các thay đổi.

Ở bên phải là khu vực đổi mật khẩu. Người dùng có thể nhập mật khẩu hiện tại, thiết lập mật khẩu mới và xác nhận lại. Việc đổi mật khẩu yêu cầu xác thực bằng mã OTP, và hệ thống cho phép lựa chọn gửi mã qua số điện thoại hoặc địa chỉ email. Khi nhấn nút “Nhận mã OTP”, mã sẽ được gửi đến kênh xác thực đã chọn, giúp tăng cường bảo mật tài khoản.

Ngay bên dưới phần đổi mật khẩu là khu vực tùy chỉnh cá nhân hoá. Tại đây, người dùng có thể thay đổi ngôn ngữ sử dụng (Tiếng Việt hoặc Tiếng Anh), lựa chọn chế độ giao diện (sáng hoặc tối), đồng thời bật hoặc tắt thông báo hệ thống theo nhu cầu. Sau khi thay đổi, người dùng có thể nhấn nút “Thay đổi tùy chỉnh” để áp dụng cấu hình mới ngay lập tức.

Hình 5.4: Giao diện thông tin người dùng

Tổng thể giao diện được bố trí khoa học, dễ thao tác và trực quan, hướng đến trải nghiệm người dùng cá nhân hoá cao và đảm bảo an toàn trong quá trình sử dụng hệ thống. Việc tập trung thông tin cá nhân và thiết lập trong cùng một không gian giúp người dùng dễ dàng kiểm soát dữ liệu của họ một cách toàn diện và hiệu quả.

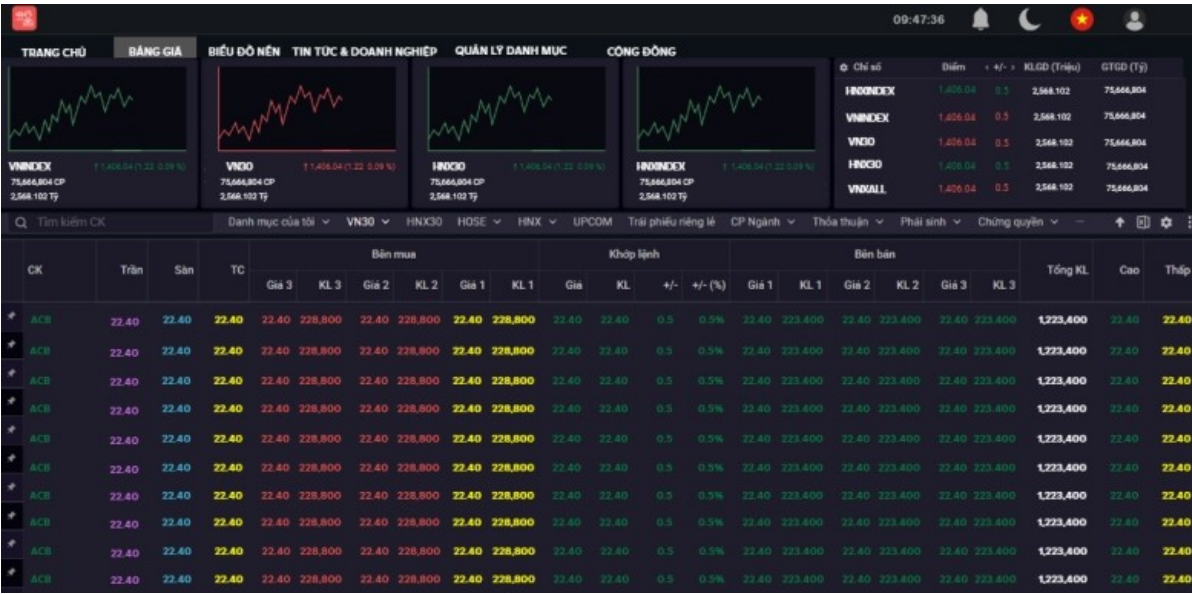
5.3.4. Giao diện Bảng giá cổ phiếu

Giao diện bảng giá là một trong những thành phần quan trọng nhất của hệ thống, được thiết kế để cung cấp cho người dùng khả năng theo dõi toàn diện diễn biến giá cổ phiếu trên thị trường chứng khoán trong thời gian thực. Trang này được truy cập thông qua nút “BẢNG GIÁ” trên thanh điều hướng chính và hiển thị dưới dạng một bảng dữ liệu lớn, cho phép người dùng quan sát và phân tích đồng thời nhiều mã cổ phiếu theo các tiêu chí khác nhau như sàn giao dịch, nhóm ngành, hoặc theo danh sách tùy chỉnh.

Bố cục của giao diện được tổ chức theo hướng trực quan, dễ quan sát. Phía trên cùng của bảng là hệ thống các thanh công cụ lọc thông minh. Người dùng có thể tìm kiếm mã cổ phiếu cụ thể bằng ô nhập liệu, hoặc lựa chọn hiển thị các nhóm như VN30, HNX30, HOSE, HNX, UPCOM, nhóm ngành tài chính, dầu khí, ngân hàng, hoặc các danh mục cổ phiếu tự tạo. Mỗi lần thay đổi bộ lọc, nội dung bảng sẽ cập nhật lại ngay lập tức mà không cần tải lại trang, tạo ra trải nghiệm mượt mà và hiệu quả.

Phần chính của giao diện là bảng dữ liệu thời gian thực, với mỗi hàng tương ứng một mã cổ phiếu và mỗi cột thể hiện một chỉ số giao dịch. Các thông tin được hiển thị bao gồm: mã cổ phiếu, giá trần, giá sàn, giá tham chiếu, các bước giá mua và bán, giá khớp lệnh hiện tại, thay đổi so với tham chiếu, phần trăm thay đổi, tổng khối lượng khớp,

giá trị giao dịch, giá cao nhất, giá thấp nhất, dư mua – dư bán và một số chỉ tiêu kỹ thuật khác.



Hình 5.5: Giao diện bảng giá cổ phiếu

Khả năng cập nhật dữ liệu hoàn toàn theo thời gian thực nhờ tích hợp WebSocket. Nhờ đó, người dùng có thể theo dõi mọi biến động của thị trường một cách liên tục và tức thời mà không cần thao tác làm mới thủ công. Ngoài ra, khi nhấn vào một mã cổ phiếu bất kỳ, hệ thống sẽ mở popup hoặc chuyển sang trang chi tiết cổ phiếu, nơi người dùng có thể xem biểu đồ kỹ thuật, tin tức liên quan và thông tin phân tích chuyên sâu. Điều này tạo nên sự kết nối mượt mà giữa bảng giá và các thành phần khác trong hệ thống.

Giao diện bảng giá không chỉ phục vụ nhu cầu theo dõi thông tin, mà còn là công cụ mạnh mẽ giúp người dùng so sánh hiệu quả giữa các cổ phiếu, phát hiện tín hiệu giao dịch, hoặc chuẩn bị cho các hành động đầu tư. Với thiết kế tập trung vào tính rõ ràng, tốc độ hiển thị và khả năng tương tác cao, bảng giá trở thành trung tâm dữ liệu quan trọng nhất trong hệ thống, hỗ trợ mạnh mẽ cho cả nhà đầu tư mới lẫn các nhà giao dịch chuyên nghiệp.

5.3.5. Giao diện đặt lệnh mua bán cổ phiếu

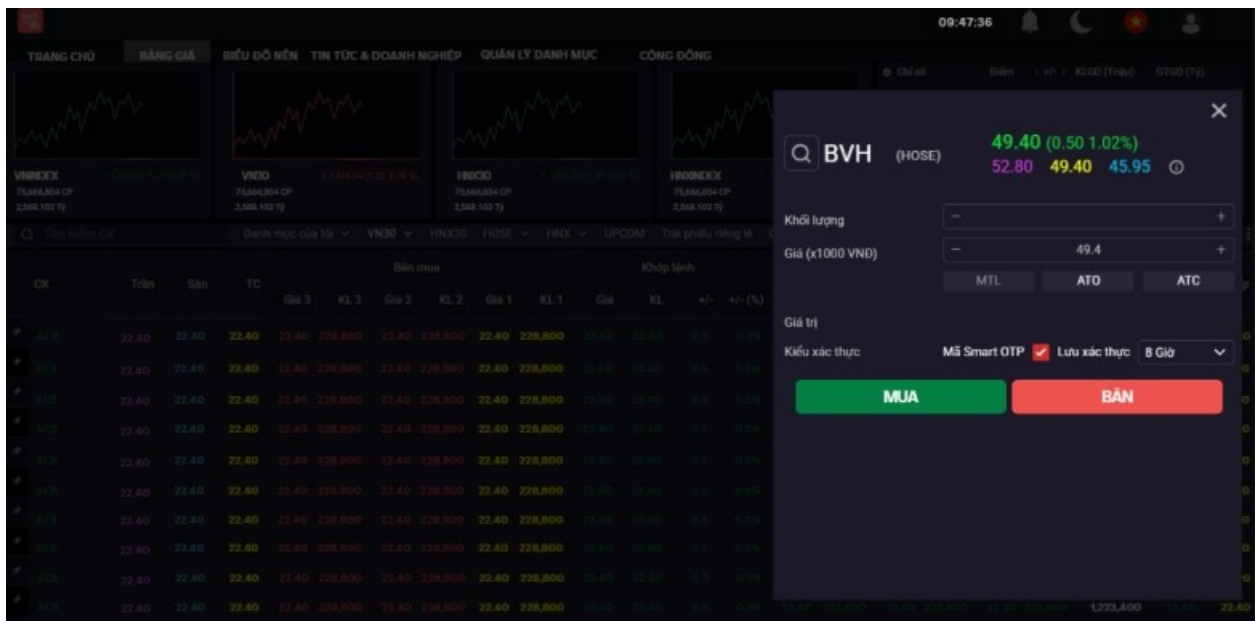
Giao diện đặt lệnh là thành phần cốt lõi trong hệ thống giao dịch chứng khoán thực tế, cho phép người dùng thực hiện các thao tác mua hoặc bán cổ phiếu trực tiếp trên thị trường. Không còn mang tính chất mô phỏng, giao diện này tương tác với tài khoản đầu tư thật của người dùng, các thao tác thực hiện trên giao diện đều sẽ ảnh hưởng trực tiếp đến danh mục tài sản và trạng thái giao dịch. Do đó, giao diện không chỉ cần đảm

bảo độ chính xác và tốc độ xử lý cao mà còn phải thân thiện, dễ sử dụng và tuyệt đối bảo mật.

Giao diện được hiển thị dưới dạng cửa sổ nổi bên phải màn hình mỗi khi người dùng nhấn vào một mã cổ phiếu bất kỳ trong bảng giá và chọn thao tác mua hoặc bán. Ngay đầu giao diện là mã cổ phiếu được chọn, hiển thị nổi bật cùng với các chỉ số quan trọng như giá hiện tại, phần trăm tăng hoặc giảm so với phiên trước, giá trần – sàn và tổng khối lượng giao dịch. Điều này giúp người dùng nắm bắt nhanh tình hình cổ phiếu trước khi quyết định giao dịch.

Phần nội dung chính của giao diện là biểu mẫu đặt lệnh, trong đó người dùng cần lựa chọn tài khoản chứng khoán thực tế muốn sử dụng để giao dịch. Tiếp theo là khối lượng cổ phiếu cần mua hoặc bán, cùng với mức giá đặt lệnh. Hệ thống hỗ trợ nhiều loại lệnh phổ biến đúng chuẩn thị trường Việt Nam gồm MTL (lệnh thị trường khớp nhanh, phần dư thành lệnh giới hạn), ATO (lệnh tại giá mở cửa) và ATC (lệnh tại giá đóng cửa). Ngoài ra, người dùng cũng có thể tự nhập mức giá cụ thể nếu chọn lệnh giới hạn, đảm bảo linh hoạt trong quản lý giá mua/bán và phù hợp với chiến lược giao dịch cá nhân. Bên cạnh đó, người dùng cũng có thể chọn đơn vị tiền giao dịch, giúp tối ưu theo nhu cầu và tài khoản hiện có.

Một điểm đặc biệt của giao diện này là cơ chế xác thực chặt chẽ trước khi đặt lệnh được xử lý. Người dùng có thể chọn hình thức xác thực qua mã Smart OTP hoặc sử dụng mã lưu sẵn theo thiết lập cá nhân. Việc xác thực hai lớp là yếu tố bắt buộc trong mọi giao dịch thực tế, giúp giảm thiểu rủi ro thao tác nhầm lệnh hoặc gian lận tài khoản. Sau khi hoàn tất nhập thông tin và xác thực, người dùng có thể nhấn vào nút “Mua” hoặc “Bán” – hai nút hành động được thiết kế nổi bật với màu sắc đặc trưng (xanh và đỏ) để đảm bảo rõ ràng trong quyết định giao dịch.



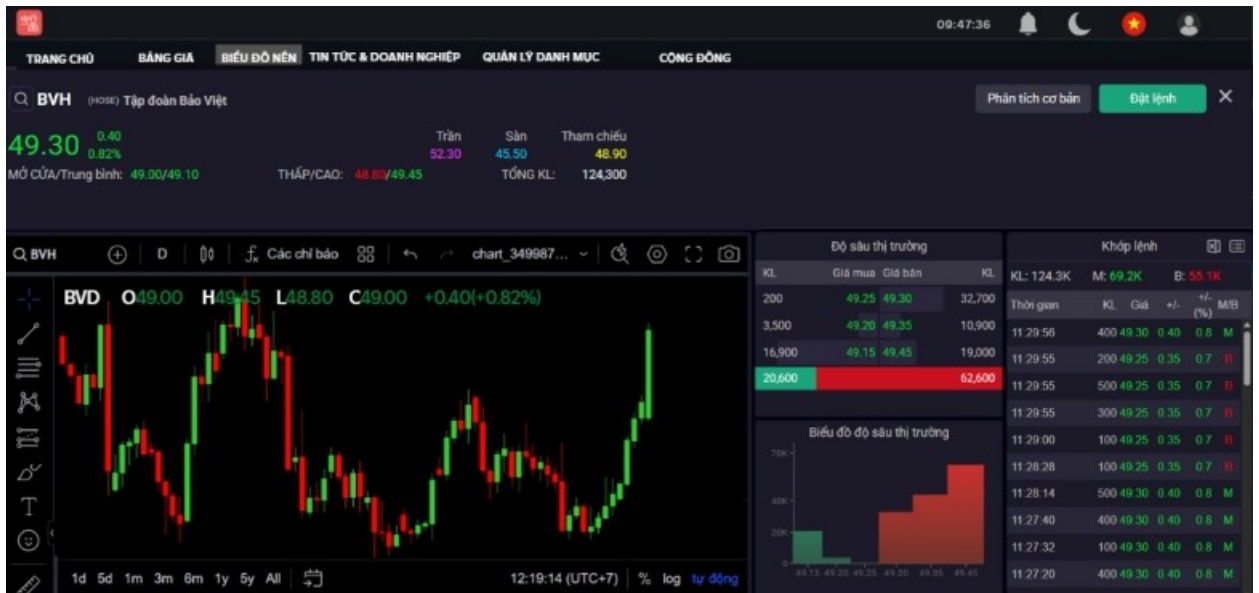
Hình 5.6: Giao diện mua bán cổ phiếu

Toàn bộ quy trình từ nhập thông tin, xác nhận, cho đến phản hồi trạng thái lệnh đều được xử lý nhanh chóng, giúp đảm bảo khả năng đặt lệnh kịp thời theo diễn biến thị trường. Giao diện này không chỉ đảm bảo các nguyên tắc khắt khe về nghiệp vụ chứng khoán, mà còn tối ưu hóa trải nghiệm người dùng nhờ bố cục trực quan, thao tác đơn giản và khả năng phản hồi tức thời. Đây là một trong những giao diện quan trọng nhất trong hệ thống, kết nối trực tiếp giữa người dùng và thị trường chứng khoán.

5.3.6. Giao diện biểu đồ kỹ thuật cổ phiếu

Giao diện biểu đồ kỹ thuật là nơi cung cấp cho người dùng công cụ trực quan để phân tích xu hướng giá cổ phiếu theo thời gian. Đây là chức năng cốt lõi phục vụ nhu cầu đầu tư trung và dài hạn, hỗ trợ các nhà đầu tư phân tích hành vi thị trường dựa trên lịch sử biến động giá và khối lượng giao dịch. Giao diện này được tích hợp chặt chẽ với các khu vực khác như bảng giá, giao dịch, danh mục và được truy cập khi người dùng nhấn vào một mã cổ phiếu bất kỳ để xem chi tiết.

Tại chế độ mặc định, biểu đồ kỹ thuật xuất hiện dưới dạng thu gọn trong phần tổng quan chi tiết của mã cổ phiếu. Ngoài thông tin biểu đồ nền cơ bản, phần này còn hiển thị các chỉ số quan trọng như giá mở cửa, giá cao nhất, thấp nhất, giá tham chiếu, khối lượng giao dịch, cùng các thông tin liên quan đến lệnh mua bán và phân tích kỹ thuật. Bố cục được sắp xếp trực quan, với biểu đồ nằm ở trung tâm và các bảng dữ liệu hỗ trợ được đặt hai bên, giúp người dùng dễ dàng quan sát tổng thể và đưa ra quyết định.



Hình 5.7: Giao diện biểu đồ cổ phiếu 1

Khi người dùng nhấn để phóng to biểu đồ, giao diện sẽ chuyển sang chế độ toàn màn hình. Biểu đồ nền candlestick sẽ chiếm toàn bộ không gian hiển thị và thể hiện đầy đủ các khung thời gian như 1 ngày, 1 tuần, 1 tháng, 3 tháng hoặc 1 năm. Biểu đồ thể hiện rõ các cây nến tương ứng với biến động giá theo thời gian, giúp người dùng dễ dàng nhận biết các vùng hỗ trợ, kháng cự, tín hiệu đảo chiều hoặc tiếp diễn xu hướng. Người dùng có thể tùy chọn loại hiển thị, kéo – zoom, chuyển đổi chế độ nền và định dạng thời gian hiển thị theo múi giờ UTC+7.



Hình 5.8: Giao diện biểu đồ cổ phiếu 2

Một tính năng nâng cao của biểu đồ là khả năng tích hợp các chỉ báo kỹ thuật thông dụng như Bollinger Bands, MACD, RSI, MA/EMA. Khi bật các chỉ báo này, biểu đồ sẽ hiển thị thêm các đường kỹ thuật hoặc biểu đồ phụ ở phía dưới, từ đó người dùng có thể kết hợp nhiều yếu tố để đánh giá trạng thái kỹ thuật của cổ phiếu. Chẳng hạn, khi bật MACD, người dùng sẽ thấy biểu đồ histogram và hai đường tín hiệu thể hiện trạng thái phân kỳ – hội tụ; còn với Bollinger Bands, ba đường giới hạn được hiển thị xung quanh biểu đồ nền giúp xác định độ biến động của giá.

Giao diện biểu đồ được tích hợp với công cụ TradingView – một trong những nền tảng biểu đồ kỹ thuật mạnh mẽ nhất hiện nay, cho phép người dùng thêm đường kẻ thủ công, vẽ vùng hỗ trợ, tạo kịch bản mô phỏng giá, lưu template chỉ báo và cá nhân hóa không gian làm việc. Toàn bộ thao tác đều được xử lý mượt mà, phản hồi nhanh và có thể lưu tạm vào local storage của trình duyệt hoặc cloud cá nhân.

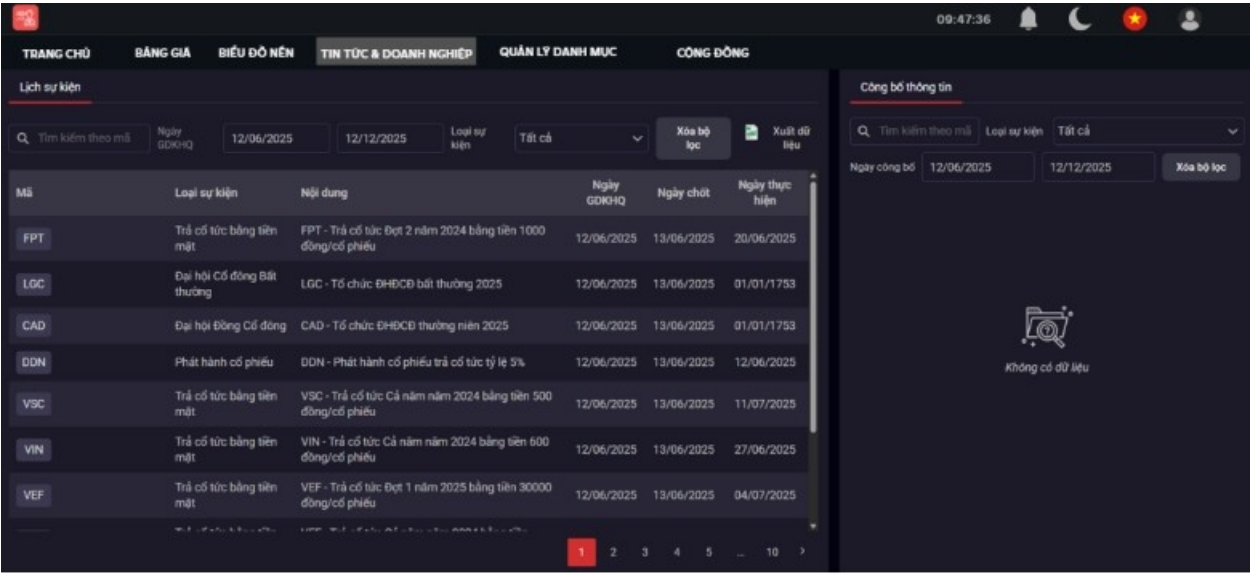
Nhờ thiết kế khoa học và tích hợp công cụ chuyên sâu, giao diện biểu đồ kỹ thuật không chỉ phục vụ mục đích theo dõi giá cổ phiếu thông thường mà còn hỗ trợ phân tích chuyên sâu, giúp nhà đầu tư nâng cao độ chính xác trong chiến lược mua – bán. Đây là giao diện không thể thiếu trong bất kỳ hệ thống giao dịch chứng khoán hiện đại nào, đóng vai trò cầu nối giữa dữ liệu giá và hành vi đầu tư chiến lược.

5.3.7. Giao diện Tin tức và doanh nghiệp

Giao diện Tin tức và doanh nghiệp là một trong những thành phần nội dung quan trọng nhất trong hệ thống, cung cấp cho người dùng khả năng tiếp cận các nguồn thông tin đầu tư chính thống, dữ liệu phân tích định lượng, cũng như công cụ trực quan để theo dõi biến động ngành nghề và nhóm cổ phiếu tiêu biểu. Giao diện này được thiết kế với ba khu vực nội dung lớn, tương ứng với các nhu cầu thông tin khác nhau của nhà đầu tư: lịch sự kiện công bố thông tin, bảng phân tích cơ bản chi tiết và bản đồ nhiệt thị trường theo từng nhóm ngành.

Ở khu vực đầu tiên, người dùng được cung cấp một bảng tổng hợp lịch sử kiện với đầy đủ thông tin liên quan đến các mã cổ phiếu chuẩn bị phát hành báo cáo, tổ chức đại hội cổ đông, chia cổ tức, chốt quyền hoặc thay đổi cơ cấu vốn. Giao diện cho phép lọc dữ liệu theo thời gian, mã cổ phiếu, loại sự kiện, tình trạng thực hiện và các tiêu chí khác. Bên trái là bảng danh sách sự kiện được hiển thị dạng bảng truyền thống gồm mã cổ phiếu, loại sự kiện, nội dung chi tiết, ngày công bố và ngày thực hiện. Bên phải là khu vực tra cứu nhanh, giúp người dùng tìm kiếm thông tin công bố riêng theo mã

hoặc theo khung thời gian cụ thể. Thiết kế đơn giản, rõ ràng, giúp người dùng dễ dàng nắm được diễn biến thông tin vĩ mô và lịch trình doanh nghiệp ảnh hưởng đến giá cổ phiếu trong tương lai gần.



Hình 5.9: Giao diện tin tức doanh nghiệp 1

Phần cuối cùng trong giao diện là hệ thống danh sách cổ phiếu được chia theo ngành nghề, cho phép người dùng nhấp chọn để phân tích chi tiết từng nhóm cụ thể. Điều này tạo ra một cái nhìn tổng thể về sức khỏe thị trường và dòng vốn theo ngành.



Hình 5.10: Giao diện tin tức doanh nghiệp 2

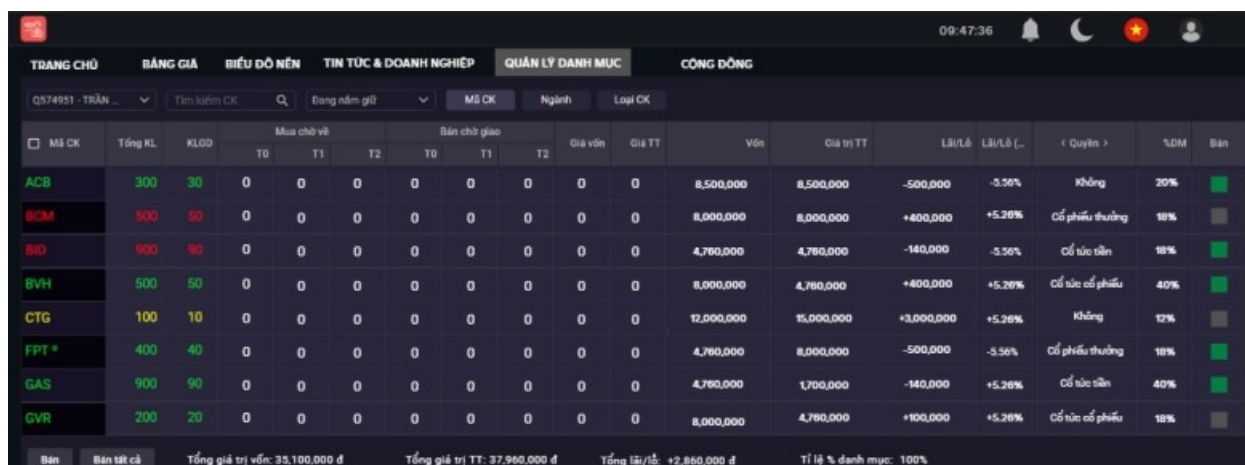
Toàn bộ giao diện được bố trí thông minh, kết hợp giữa phân tích định tính (tin tức, sự kiện) và định lượng (số liệu tài chính, chỉ báo kỹ thuật), giúp người dùng dễ dàng chuyển đổi giữa thông tin vĩ mô và từng mã cổ phiếu cụ thể. Đây là khu vực mang tính chiến lược trong hệ thống, nơi người dùng có thể bắt đầu mọi quyết định đầu tư dựa

trên dữ liệu và góc nhìn toàn diện, thay vì chỉ phụ thuộc vào trực giác hoặc thông tin rời rạc. Nhờ thiết kế chi tiết, hiện đại và khả năng truy xuất nhanh chóng, giao diện Tin tức và Phân tích không chỉ nâng cao tính chuyên nghiệp của hệ thống mà còn hỗ trợ mạnh mẽ cho nhà đầu tư trong việc ra quyết định chính xác và kịp thời.

5.3.8. Giao diện Quản lý danh mục đầu tư

Giao diện quản lý danh mục đầu tư là nơi người dùng có thể theo dõi toàn bộ danh mục cổ phiếu của mình một cách có hệ thống, chính xác và trực quan. Đây là chức năng thiết yếu trong mọi nền tảng hỗ trợ đầu tư, đặc biệt dành cho người dùng muốn phân tích hiệu quả đầu tư dài hạn, đánh giá mức sinh lời thực tế, cũng như tối ưu hóa tỷ trọng phân bổ tài sản. Giao diện được thiết kế hiện đại, chia thành ba vùng chính gồm bảng thông tin chi tiết danh mục, khu vực biểu đồ phân tích tỷ trọng, và bảng theo dõi cổ phiếu đã chọn.

Phần đầu của giao diện là bảng tổng hợp toàn bộ cổ phiếu mà người dùng đang nắm giữ. Mỗi hàng trong bảng tương ứng với một mã cổ phiếu, kèm theo các thông tin chi tiết bao gồm khối lượng nắm giữ, giá vốn, giá hiện tại, giá trị đầu tư, lợi nhuận/lỗ tạm tính, phần trăm tăng giảm, tỷ trọng hiện tại trong danh mục và các gợi ý hành động như giữ nguyên, mua thêm hoặc bán bớt. Ngoài ra, hệ thống còn hiển thị tổng giá trị đầu tư, tổng giá trị thị trường, tổng lợi nhuận hoặc lỗ, cũng như tỷ lệ tăng trưởng của toàn danh mục, được tính toán và cập nhật liên tục dựa trên giá thị trường thời gian thực.



Mã CK	Tổng KL	KLGD	Mua chủ yếu			Bán chủ yếu			Giá vốn	Giá TT	Vốn	Giá trị TT	Lãi/Lỗ	Lãi/Lỗ (%)	Quyển	%DM	Bán
			T0	T1	T2	T0	T1	T2									
ACB	300	30	0	0	0	0	0	0	0	0	8,500,000	8,500,000	-500,000	-5.88%	Không	20%	
BCM	500	50	0	0	0	0	0	0	0	0	8,000,000	8,000,000	+400,000	+5.00%	Cổ phiếu thường	18%	
BID	900	90	0	0	0	0	0	0	0	0	4,780,000	4,780,000	-140,000	-2.93%	Cổ tức tiền	18%	
BVH	500	50	0	0	0	0	0	0	0	0	8,000,000	4,780,000	+400,000	+5.00%	Cổ tức cổ phiếu	40%	
CTG	100	10	0	0	0	0	0	0	0	0	12,000,000	15,000,000	+3,000,000	+25.00%	Không	12%	
FPT*	400	40	0	0	0	0	0	0	0	0	4,780,000	8,000,000	-500,000	-10.46%	Cổ phiếu thường	18%	
GAS	900	90	0	0	0	0	0	0	0	0	4,780,000	1,700,000	-140,000	-2.93%	Cổ tức tiền	40%	
GVR	200	20	0	0	0	0	0	0	0	0	8,000,000	4,780,000	+100,000	+1.25%	Cổ tức cổ phiếu	18%	

Tổng giá trị vốn: 35,100,000 đ Tổng giá trị TT: 37,960,000 đ Tổng lãi/lỗ: +2,860,000 đ Tỷ lệ % danh mục: 100%

Hình 5.11: Giao diện quản lý danh mục đầu tư 1

Cuối giao diện là bảng “Theo dõi”, nơi người dùng có thể thêm các mã cổ phiếu chưa mua vào danh sách theo dõi để xem biến động trước khi đưa vào danh mục thật. Tại đây, mỗi mã cổ phiếu được hiển thị cùng các chỉ số cơ bản như giá trần, giá sàn, giá

hiện tại, phần trăm tăng/giảm và tổng khối lượng giao dịch. Người dùng có thể thêm mới hoặc loại bỏ các mã theo dõi, tìm kiếm theo tên mã, hoặc lọc theo ngành và trạng thái thị trường. Việc theo dõi mã không ảnh hưởng đến danh mục thực tế, nhưng lại rất hữu ích trong việc lên kế hoạch đầu tư tương lai.

THEO DÕI							
Tìm kiếm CK		Mã CK		Ngành		Loại CK	
						Thêm Mã	
<input type="checkbox"/> Mã CK		Giá Trần	Giá Sàn	+/-	+/- (%)	Khối lượng	Tổng Khối Lượng
★	ACB	22.40	19.50	0.05	0.24%	300	2,641,300
★	BCM	63.80	55.60	-0.40	-0.67%	500	177,100
★	BID	37.70	32.80	-0.05	-0.14%	900	1,191,400
★	BVH	52.00	45.20	1.00	2.06%	500	173,000
★	CTG	40.65	35.35	0.00	0.00%	100	2,739,600
★	FPT*	124.30	108.10	0.90	0.77%	400	1,658,000
★	GAS	67.70	58.90	0.20	0.32%	900	148,500

Hình 5.12: Giao diện quản lý danh mục đầu tư 2

Tổng thể giao diện quản lý danh mục được xây dựng dựa trên nguyên tắc quản trị danh mục tài chính hiện đại, giúp người dùng không chỉ theo dõi số liệu thụ động mà còn có khả năng phân tích, tối ưu và chủ động ra quyết định. Việc tích hợp biểu đồ trực quan, bảng dữ liệu động và công cụ phân tích cân bằng giúp tăng cường khả năng đánh giá rủi ro – lợi nhuận, đồng thời cải thiện hiệu suất đầu tư dài hạn của người dùng.

5.3.9. Giao diện Cộng đồng thảo luận cổ phiếu

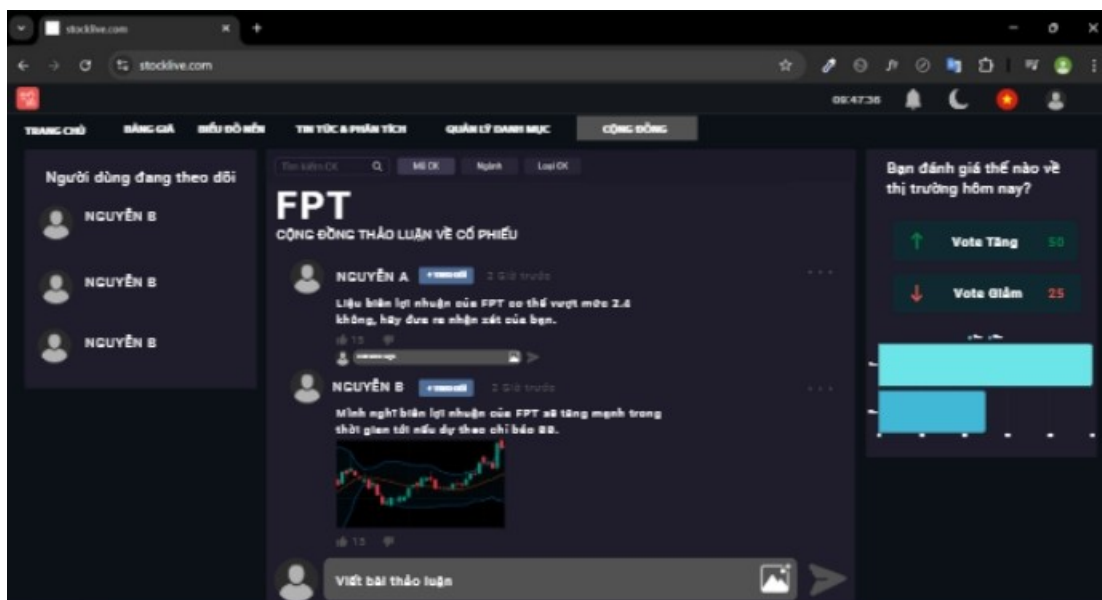
Giao diện cộng đồng thảo luận là nơi người dùng có thể trao đổi, chia sẻ quan điểm, đăng bài phân tích hoặc tương tác trực tiếp với những nhà đầu tư khác về một mã cổ phiếu cụ thể. Đây là chức năng mang tính kết nối cộng đồng, không chỉ giúp tăng sự gắn kết giữa người dùng trong hệ thống, mà còn góp phần tạo ra một hệ sinh thái tri thức, nơi mọi người có thể học hỏi từ các góc nhìn phân tích khác nhau. Giao diện này được thiết kế như một diễn đàn chuyên sâu, tích hợp các yếu tố tương tác thời gian thực, hệ thống phản hồi, cũng như cơ chế phân loại chủ đề để đảm bảo nội dung được tổ chức khoa học và dễ theo dõi.

Ngay khi người dùng truy cập vào khu vực cộng đồng, hệ thống sẽ hiển thị chủ đề thảo luận tương ứng với một mã cổ phiếu cụ thể. Trong ảnh là ví dụ về mã FPT, nơi người dùng có thể đọc các bài viết được chia sẻ công khai bởi những nhà đầu tư khác. Mỗi bài viết bao gồm tên người đăng, thời gian đăng tải, nội dung văn bản và có thể đính kèm hình ảnh minh họa như biểu đồ phân tích kỹ thuật. Dưới mỗi bài viết là hệ thống phản hồi bao gồm lượt thích, bình luận và biểu tượng chia sẻ, giúp tăng tính tương tác và lan tỏa thông tin trong cộng đồng.

Phía bên trái giao diện là danh sách những người dùng mà tài khoản hiện tại đang theo dõi. Người dùng có thể truy cập nhanh vào các trang cá nhân hoặc xem toàn bộ bài viết của người mình quan tâm. Đây là một cách tiếp cận theo mô hình mạng xã hội, cho phép cá nhân hóa dòng thông tin dựa trên kết nối của người dùng.

Bên phải giao diện là khối đánh giá tâm lý thị trường, nơi hệ thống thu thập ý kiến cộng đồng theo dạng biểu quyết trực tiếp với hai lựa chọn: “Vote Tăng” và “Vote Giảm”. Dữ liệu được thống kê theo thời gian thực và hiển thị dưới dạng biểu đồ cột nhỏ giúp người dùng dễ dàng nắm bắt xu hướng tâm lý của phần đông nhà đầu tư tại một thời điểm cụ thể. Việc biểu quyết cảm xúc thị trường không chỉ là công cụ thể hiện quan điểm mà còn có giá trị tham khảo đối với những người đang phân vân trước các quyết định đầu tư.

Dưới khu vực bài viết là ô nhập liệu cho phép người dùng viết bài thảo luận mới. Giao diện hỗ trợ nhập nhanh, cho phép đính kèm ảnh, biểu đồ hoặc liên kết ngoài. Mọi nội dung gửi lên đều được hiển thị tức thời mà không cần tải lại trang, giúp tăng tốc độ tương tác. Việc tổ chức theo chủ đề mã cổ phiếu còn giúp người dùng dễ dàng theo dõi nội dung liên quan đến từng mã cụ thể mà không bị lẫn lộn với các chủ đề khác.



Hình 5.13: Giao diện cộng đồng tương tác

Tổng thể, giao diện cộng đồng mang tính kết nối cao, đáp ứng nhu cầu chia sẻ thông tin, phân tích và thể hiện quan điểm đầu tư một cách cởi mở, văn minh. Việc tích hợp yếu tố cộng đồng vào hệ thống tài chính không chỉ nâng cao trải nghiệm người dùng mà còn mở ra tiềm năng học hỏi lẫn nhau, thúc đẩy hình thành các quyết định đầu tư dựa trên sự tham khảo đa chiều và gắn liền với cảm nhận thực tế của thị trường.

Thông qua việc thiết kế và trình bày các giao diện trọng yếu, phần này đã thể hiện rõ định hướng lấy người dùng làm trung tâm với bố cục hiện đại, dễ sử dụng và giàu tính tương tác. Mỗi giao diện được xây dựng nhất quán theo mục tiêu chức năng, hỗ trợ tối đa quá trình theo dõi thị trường và ra quyết định đầu tư. Khả năng cập nhật thời gian thực và tính cá nhân hóa cao đã góp phần tạo nên một hệ thống chuyên nghiệp, hiệu quả và phù hợp với nhu cầu thực tiễn của nhà đầu tư.

Chương 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Đánh giá chung về hệ thống đã triển khai

Hệ thống giám sát thị trường chứng khoán thời gian thực được xây dựng trong khuôn khổ đồ án môn học đã cơ bản đáp ứng đầy đủ các mục tiêu và yêu cầu kỹ thuật đặt ra từ giai đoạn đầu. Quá trình thực hiện dự án đã trải qua các bước bài bản từ khảo sát, phân tích nhu cầu, thiết kế kiến trúc tổng thể, lập trình chức năng cho đến kiểm thử và hiệu chỉnh trên môi trường giả lập. Qua đó, hệ thống không chỉ chứng minh được tính khả thi về mặt kỹ thuật, mà còn thể hiện rõ hiệu quả trong việc xử lý, cập nhật và hiển thị thông tin thị trường chứng khoán một cách nhanh chóng, ổn định và chính xác – những yếu tố đặc biệt quan trọng trong các ứng dụng tài chính có tính thời gian thực cao.

Một trong những điểm nổi bật nhất của hệ thống là khả năng cập nhật dữ liệu thị trường liên tục và tức thời thông qua giao thức WebSocket. Đây là công nghệ truyền thông hai chiều giữa server và client, cho phép hệ thống gửi dữ liệu mới đến trình duyệt người dùng ngay khi có sự thay đổi mà không cần gửi lại yêu cầu HTTP theo kiểu truyền thống. Điều này giúp giảm thiểu độ trễ và giảm tải cho server, từ đó nâng cao hiệu suất hệ thống và mang lại trải nghiệm tương tác mượt mà, trực quan cho người dùng. Với đặc điểm thị trường tài chính luôn biến động từng giây, việc áp dụng WebSocket giúp đảm bảo rằng người dùng có thể nắm bắt diễn biến giá cổ phiếu gần như tức thì, từ đó chủ động trong phân tích và ra quyết định đầu tư.

Bên cạnh năng lực xử lý dữ liệu realtime, hệ thống còn gây ấn tượng với phần giao diện người dùng được thiết kế thân thiện, dễ sử dụng và trực quan. Các thông số kỹ thuật quan trọng như giá mở cửa, giá hiện tại, mức cao nhất – thấp nhất trong ngày cùng với phần trăm thay đổi được hiển thị rõ ràng thông qua bảng giá kết hợp với biểu đồ kỹ thuật sinh động. Màu sắc được sử dụng một cách tối ưu (xanh – tăng giá, đỏ – giảm giá, vàng – không đổi), giúp người dùng dễ dàng nhận diện xu hướng biến động mà không cần đọc quá nhiều chi tiết. Ngoài ra, hệ thống còn hỗ trợ các loại biểu đồ phổ biến như biểu đồ nến, biểu đồ khối lượng giao dịch với khung thời gian đa dạng, đáp ứng nhu cầu theo dõi và phân tích thị trường từ cơ bản đến nâng cao của nhiều nhóm người dùng, từ cá nhân đến tổ chức.

Về mặt kỹ thuật, hệ thống backend được xây dựng dựa trên kiến trúc phân tầng rõ ràng, tách biệt giữa các thành phần: frontend – backend – cơ sở dữ liệu. Điều này không chỉ giúp nâng cao tính tổ chức trong codebase mà còn mang lại nhiều lợi thế trong bảo trì, kiểm thử và mở rộng về sau. Cơ sở dữ liệu được thiết kế chuẩn hóa, hỗ trợ hiệu quả cho việc lưu trữ và truy vấn thông tin cổ phiếu. Backend sử dụng Node.js với mô hình bất đồng bộ nhằm tối ưu hiệu năng và giảm độ trễ trong xử lý. Frontend giao tiếp với backend thông qua API và kênh WebSocket, đảm bảo thông tin được truyền tải nhất quán và tức thời.

Ngoài các chức năng chính, hệ thống còn thể hiện tính linh hoạt trong triển khai với khả năng tương thích nhiều thiết bị, dễ dàng mở rộng thành phiên bản đa nền tảng trong tương lai như mobile app hoặc desktop client. Nhờ ứng dụng công nghệ web hiện đại, giao diện có thể thích ứng với nhiều độ phân giải, đồng thời vẫn giữ nguyên bố cục, đảm bảo trải nghiệm người dùng ổn định trên cả máy tính cá nhân lẫn thiết bị di động.

6.2. Những hạn chế và bài học kinh nghiệm

Trong suốt quá trình thiết kế, xây dựng và thử nghiệm, hệ thống giám sát thị trường chứng khoán thời gian thực đã thể hiện được nhiều điểm mạnh nổi bật, đồng thời cũng giúp nhóm phát triển nhận diện rõ những hạn chế tồn tại cần được khắc phục trong các giai đoạn tiếp theo. Đây là quá trình không chỉ giúp hoàn thiện sản phẩm về mặt kỹ thuật mà còn là cơ hội để tích lũy nhiều kinh nghiệm thực tiễn quý báu trong phát triển phần mềm hiện đại.

Về ưu điểm, hệ thống đã đạt được nhiều thành tựu nổi bật, thể hiện qua các khía cạnh sau:

- Kiến trúc tổng thể được thiết kế bài bản và hợp lý ngay từ đầu với sự phân chia rõ ràng giữa các tầng chức năng như giao diện người dùng (frontend), xử lý logic nghiệp vụ (backend) và lưu trữ dữ liệu (database). Cách tổ chức này không những giúp tăng tính dễ hiểu và dễ kiểm soát của toàn bộ hệ thống mà còn tạo điều kiện thuận lợi cho việc bảo trì, nâng cấp và mở rộng trong tương lai.
- Cơ chế xử lý dữ liệu thời gian thực (real-time) thông qua WebSocket được triển khai hiệu quả, mang lại khả năng cập nhật dữ liệu giá cổ phiếu gần như tức thì, đảm bảo độ trễ thấp và tính ổn định cao. Đây là yếu tố then chốt trong các hệ

thông tài chính, nơi mà thông tin phải được truyền tải một cách nhanh chóng và chính xác để hỗ trợ quyết định của người dùng.

- Tính tương thích và khả năng tích hợp với các công nghệ hiện đại cũng là một điểm mạnh đáng ghi nhận. Hệ thống backend sử dụng Node.js kết hợp với PostgreSQL – một hệ quản trị cơ sở dữ liệu mạnh mẽ – cùng với khả năng kết nối linh hoạt với các API thị trường và thư viện frontend phổ biến. Nhờ vậy, quá trình phát triển được rút ngắn đáng kể mà vẫn đảm bảo chất lượng và độ mở của kiến trúc.
- Quản lý nhóm và phân công công việc trong quá trình phát triển được triển khai hiệu quả với sự phối hợp chặt chẽ giữa các thành viên trong nhóm. Việc ứng dụng các công cụ như Git để quản lý mã nguồn, theo dõi thay đổi và phân nhánh phát triển đã góp phần quan trọng vào việc duy trì tiến độ, kiểm soát chất lượng và đảm bảo tính thống nhất trong quá trình hợp tác nhóm.

Tuy nhiên, bên cạnh những kết quả tích cực, hệ thống vẫn tồn tại một số hạn chế nhất định, chủ yếu đến từ phạm vi triển khai mô phỏng và giới hạn về tài nguyên:

- Hệ thống chưa được triển khai và kiểm thử trong môi trường thực tế với số lượng người dùng lớn dẫn đến khó đánh giá chính xác khả năng chịu tải, hiệu suất hoạt động thực tế, cũng như cách hệ thống phản ứng trong các tình huống quá tải hoặc bất thường.
- Giao diện người dùng vẫn còn ở mức cơ bản, chưa thực sự tối ưu về mặt trải nghiệm, đặc biệt là trên các thiết bị di động hoặc với những người dùng có nhu cầu theo dõi phân tích thị trường chuyên sâu. Việc chưa tích hợp các công cụ phân tích kỹ thuật nâng cao có thể làm giảm trải nghiệm người dùng chuyên nghiệp.
- Chưa tích hợp hệ thống giám sát lỗi và logging thời gian thực khiến việc phát hiện, theo dõi và xử lý lỗi ở môi trường sản xuất trở nên khó khăn hơn. Việc thiếu các công cụ như dashboard theo dõi trạng thái server, thông báo lỗi tự động hoặc giám sát hiệu suất có thể gây bất lợi trong việc vận hành lâu dài.
- Quá trình xử lý dữ liệu realtime vẫn tiềm ẩn một số rủi ro kỹ thuật chẳng hạn như tràn bộ nhớ, rò rỉ kết nối WebSocket hoặc tình trạng mất đồng bộ dữ liệu

nếu không được kiểm soát chặt chẽ và không có cơ chế phục hồi kịp thời khi có sự cố xảy ra.

Từ những thuận lợi và khó khăn đã trải qua, nhóm phát triển đã rút ra được nhiều bài học kinh nghiệm quý giá. Việc lựa chọn công nghệ, thiết kế kiến trúc ngay từ đầu đóng vai trò quyết định đến sự thành công của toàn bộ hệ thống. Một nền tảng công nghệ đúng đắn không chỉ giúp tiết kiệm thời gian triển khai mà còn tạo tiền đề cho việc nâng cấp mở rộng sau này. Đồng thời, khi làm việc với dữ liệu thời gian thực – vốn có tính chất nhạy cảm và yêu cầu kỹ thuật cao, lập trình viên cần đặc biệt cẩn trọng trong việc xử lý bộ nhớ, kiểm soát luồng dữ liệu, thiết lập giới hạn kết nối và đảm bảo độ chính xác tuyệt đối trong mọi tình huống. Ngoài ra, sự phối hợp nhóm hiệu quả, việc áp dụng các công cụ quản lý dự án và kiểm soát mã nguồn khoa học đã góp phần quan trọng vào việc giữ đúng tiến độ, giảm thiểu lỗi phát sinh và đảm bảo chất lượng tổng thể của hệ thống.

Quá trình triển khai dự án không chỉ mang lại một sản phẩm công nghệ hoàn chỉnh ở mức học thuật mà còn là bước đệm quan trọng giúp nhóm làm quen với quy trình phát triển phần mềm chuyên nghiệp, tích lũy kiến thức thực tế và chuẩn bị tốt hơn cho những dự án có quy mô và tính phức tạp cao hơn trong tương lai.

6.3. Hướng phát triển tương lai của hệ thống

Trong tương lai, hệ thống giám sát thị trường chứng khoán thời gian thực hoàn toàn có tiềm năng được mở rộng và nâng cấp theo nhiều hướng chiến lược nhằm tối đa hóa giá trị sử dụng, nâng cao trải nghiệm người dùng đồng thời đáp ứng tốt hơn các yêu cầu khắt khe trong môi trường tài chính hiện đại. Việc phát triển hệ thống không nên chỉ dừng lại ở mức đảm bảo các chức năng cơ bản, mà cần hướng đến việc xây dựng một nền tảng mạnh mẽ, thông minh và thích ứng linh hoạt với nhu cầu ngày càng đa dạng và biến động của thị trường.

Trước hết, một trong những hướng đi quan trọng hàng đầu là – tích hợp thêm nhiều nguồn dữ liệu thị trường tài chính chính thống, bao gồm các chỉ số và dữ liệu giá cổ phiếu từ nhiều sàn giao dịch khác nhau cả trong nước như HOSE, HNX, UPCOM và quốc tế như NYSE, NASDAQ, Tokyo Stock Exchange hay London Stock Exchange. Việc mở rộng nguồn dữ liệu không chỉ góp phần gia tăng độ phủ và độ tin cậy của thông tin mà còn cho phép người dùng có cái nhìn toàn cảnh hơn về bối cảnh tài chính toàn cầu, từ đó đưa ra những phân tích đa chiều và chính xác hơn. Đồng thời, hệ thống

cũng cần – bổ sung các chức năng tùy chỉnh biểu đồ và bộ lọc nâng cao chẳng hạn như lọc cổ phiếu theo ngành nghề, mức tăng trưởng, vốn hóa, khối lượng giao dịch, mức biến động giá hoặc các tiêu chí phân tích riêng biệt do người dùng tự cấu hình. Điều này sẽ giúp người dùng cá nhân hóa trải nghiệm, hỗ trợ tối đa cho việc phân tích, so sánh và ra quyết định đầu tư một cách hiệu quả và chuyên nghiệp hơn.

Bên cạnh đó, một yếu tố quan trọng khác không thể thiếu là tối ưu hóa giao diện người dùng (UI/UX) theo chuẩn hiện đại, đảm bảo không chỉ thẩm mỹ mà còn thân thiện, dễ sử dụng và đặc biệt là có thiết kế responsive để tương thích tốt với nhiều thiết bị có kích thước màn hình khác nhau từ máy tính để bàn, laptop đến tablet và điện thoại thông minh. Khi thiết bị di động ngày càng trở thành công cụ chính trong đời sống công nghệ, thì việc xây dựng một trải nghiệm đồng nhất trên mọi nền tảng là điều tất yếu. Song song đó, một bước tiến đáng giá là – triển khai phiên bản mobile app độc lập, giúp người dùng không còn phụ thuộc vào trình duyệt web, từ đó có thể theo dõi thị trường, nhận thông báo biến động và thao tác nhanh chóng mọi lúc, mọi nơi. Ứng dụng di động sẽ mở rộng phạm vi tiếp cận của hệ thống, đặc biệt là với thế hệ nhà đầu tư trẻ ưa thích sự linh hoạt, tốc độ và tiện lợi.

Một hướng phát triển khác cũng vô cùng cần thiết là – tăng cường khả năng giám sát và bảo mật của hệ thống, nhằm đảm bảo tính ổn định và an toàn dữ liệu trong quá trình vận hành thực tế. Cụ thể, cần tích hợp các cơ chế log server chi tiết, giám sát hiệu suất theo thời gian thực, hệ thống cảnh báo sớm khi xảy ra lỗi và các dashboard quản lý hoạt động server phục vụ cho công tác vận hành – bảo trì. Đồng thời, việc triển khai các cơ chế xác thực người dùng nâng cao như OTP, xác thực hai bước (2FA) hoặc phân quyền truy cập chi tiết sẽ giúp bảo vệ tài khoản và ngăn ngừa rủi ro bảo mật, đặc biệt trong môi trường tài chính nhạy cảm với dữ liệu cá nhân và thông tin giao dịch.

Cuối cùng nhưng không kém phần quan trọng, hệ thống cần – hỗ trợ truy xuất và hiển thị lịch sử dữ liệu giá cổ phiếu, giúp người dùng xem lại toàn bộ quá trình biến động trong một khoảng thời gian tùy chọn (theo ngày, tuần, tháng, năm). Chức năng này không chỉ có ý nghĩa về mặt lưu trữ thông tin mà còn phục vụ rất tốt cho mục đích phân tích kỹ thuật, đánh giá xu hướng, xây dựng chiến lược đầu tư cá nhân hoặc kiểm chứng giả định phân tích. Việc cho phép người dùng tải về dữ liệu quá khứ dưới dạng file CSV hoặc Excel sẽ nâng cao tính thực dụng của hệ thống và mở ra khả năng tích

hợp với các công cụ phân tích bên ngoài như Excel, Python hay các nền tảng BI chuyên nghiệp.

TÀI LIỆU THAM KHẢO

IEEE (1990). *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Std 610.12-1990.

Node.js Foundation (2024). *Node.js Official Documentation*, Truy cập tháng 6/2025, <https://nodejs.org/en/docs>.

PostgreSQL Global Development Group (2024). *PostgreSQL Official Documentation*, Truy cập tháng 6/2025, <https://www.postgresql.org/docs/>.

Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.), McGraw-Hill Education, New York.

Schmedtmann, J. (n.d.). *Node.js, Express, MongoDB & More: The Complete Bootcamp*, Udemy, Truy cập tháng 3/2024, <https://www.udemy.com/course/nodejs-express-mongodb-bootcamp/>.

Sommerville, I. (2016). *Software Engineering* (10th ed.), Pearson Education Limited, London.

GeeksforGeeks (n.d.). *Software Development Life Cycle (SDLC)*, Truy cập tháng 3/2024, <https://www.geeksforgeeks.org/software-engineering/software-development-life-cycle-sdlc/>.

Studocu (n.d.). *Bài 1: Nhập môn Công nghệ phần mềm*, Truy cập tháng 3/2024, <https://www.studocu.vn/vn/document/truong-dai-hoc-cong-nghiep-ha-noi/nhap-mon-cong-nghe-phan-mem/bai-1-nhap-mon-cong-nghe-phan-mem/53111456>.