

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

**BÁO CÁO TỔNG KẾT
ĐỀ TÀI NCKH SINH VIÊN**

**XÂY DỰNG HỆ THỐNG HỖ TRỢ TRẺ EM LẬP TRÌNH VẼ ĐỐI
TƯỢNG VỚI HÀM THƯ VIỆN TURTLE CỦA PYTHON**

Hà Nội, 5/2026

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

**BÁO CÁO TỔNG KẾT
ĐỀ TÀI NCKH SINH VIÊN**

**XÂY DỰNG HỆ THỐNG HỖ TRỢ TRẺ EM LẬP TRÌNH VẼ ĐỐI TƯỢNG
VỚI HÀM THƯ VIỆN TURTLE CỦA PYTHON**

Trưởng nhóm nghiên cứu: Phạm Đức Khôi
Thành viên tham gia thực hiện: Nguyễn Thanh Sơn
Nguyễn Huyền My
Nguyễn Thị Ngân
Đinh Thị Thảo Thư

Người hướng dẫn: TS. Nguyễn Thị Hữu Phương

Hà Nội, 5/2026

MỤC LỤC

LỜI CẢM ƠN.....	iv
TÓM TẮT NỘI DUNG NGHIÊN CỨU KHOA HỌC.....	v
DANH MỤC CÁC HÌNH VẼ.....	vi
DANH MỤC CÁC BẢNG BIỂU	viii
DANH MỤC NHỮNG TỪ VIẾT TẮT	x
MỞ ĐẦU	1
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN	7
1.1. Tổng quan về giáo dục lập trình cho trẻ em.....	7
1.1.1. Bối cảnh và xu hướng toàn cầu.....	7
1.1.2. Các mô hình giáo dục lập trình phổ biến.....	7
1.1.3. Nguyên tắc sư phạm trong lập trình trực quan	8
1.1.4. Đặc điểm tâm lý trẻ em (Tiểu học - THCS)	9
1.2. Ngôn ngữ lập trình Python và thư viện đồ họa Turtle	10
1.2.1. Ngôn ngữ lập trình Python.....	10
1.2.2. Thư viện đồ họa Turtle	11
1.3. Kiến trúc hệ thống Web-based Client-Server	13
1.4. Các công nghệ nền tảng liên quan	15
1.4.1. Công nghệ giao diện người dùng.....	15
1.4.2. Công nghệ xử lý logic.....	15
1.4.3. Cơ sở dữ liệu và hạ tầng	16
1.5. Nguyên tắc thiết kế giao diện cho trẻ em.....	17
1.5.1. Lý thuyết về tải nhận thức (Cognitive Load Theory)	17
1.5.2. Nguyên lý giàn giáo sư phạm (Instructional Scaffolding).....	17
1.5.3. Thiết kế cảm xúc và tác nhân sư phạm.....	18
1.5.4. Nguyên tắc khả năng tiếp cận (Accessibility)	19
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	21
2.1. Khảo sát hệ thống.....	21
2.1.1. Mục tiêu khảo sát.....	21
2.1.2. Đối tượng khảo sát.....	22
2.1.3. Phương pháp khảo sát.....	22
2.1.4. Kết quả khảo sát.....	23
2.1.5. Đánh giá kết quả khảo sát.....	24

2.2. Phân tích yêu cầu hệ thống.....	24
2.2.1. Phạm vi và mục tiêu.....	24
2.2.2. Đối tượng sử dụng hệ thống	25
2.2.3. Mục tiêu thiết kế hệ thống	25
2.3. Yêu cầu chức năng và phi chức năng.....	26
2.3.1. Yêu cầu chức năng.....	26
2.3.2. Yêu cầu phi chức năng.....	27
2.4. Sơ đồ Ca sử dụng (Use Case Diagram).....	28
2.4.1. Xác định tác nhân (Actor).....	28
2.4.2. Use Case tổng quát	29
2.4.3. Use Case Đăng ký.....	30
2.4.4. Use Case Đăng nhập	32
2.4.5. Use Case Quản lý bài vẽ	34
2.4.6. Use Case Thực thi chương trình	36
2.4.7. Use Case Quản lý tài khoản cá nhân	38
2.4.8. Use Case Quản lý người dùng	40
2.4.9. Use Case Quản lý bài học	43
2.5. Sơ đồ hoạt động (Activity).....	46
2.5.1. Sơ đồ hoạt động Đăng ký	46
2.5.2. Sơ đồ hoạt động Đăng nhập.....	47
2.5.3. Sơ đồ hoạt động Quản lý bài vẽ.....	48
2.5.4. Sơ đồ hoạt động Quản lý tài khoản cá nhân	56
2.5.5. Sơ đồ hoạt động Quản lý người dùng	58
2.5.6. Sơ đồ hoạt động Quản lý bài học.....	59
2.5.7. Sơ đồ hoạt động Thực thi chương trình	61
2.6. Sơ đồ lớp (Class Diagram).....	63
2.7. Sơ đồ tuần tự (Sequence Diagram)	64
2.7.1. Sơ đồ tuần tự Đăng ký	64
2.7.2. Sơ đồ tuần tự Đăng nhập	65
2.7.3. Sơ đồ tuần tự quản lý bài vẽ	66
2.7.4. Sơ đồ tuần tự quản lý tài khoản cá nhân.....	67
2.7.5. Sơ đồ tuần tự Quản lý người dùng.....	69
2.8. Thiết kế cơ sở dữ liệu	71
2.8.1. Sơ đồ ERD	71
2.8.2. Mô tả dữ liệu.....	71

2.9. Thiết kế giao diện người dùng (Mockup)	76
2.9.1. Giao diện đăng nhập	76
2.9.2. Giao diện màn hình chính khi đã đăng nhập	76
2.9.3. Giao diện cập nhật thông tin tài khoản	77
2.9.4. Giao diện tra cứu nội dung Wiki cập nhật thông tin tài khoản.....	78
CHƯƠNG 3. CÀI ĐẶT VÀ ĐÁNH GIÁ KẾT QUẢ.....	79
3.1. Triển khai và điều kiện sử dụng.....	79
3.1.1. Người dùng muốn học trên hệ thống (học sinh, người học thử)	79
3.1.2. Phần cứng và môi trường sử dụng phía người học	81
3.1.3. Triển khai phía máy chủ/phòng thí nghiệm (quản trị viên).....	81
3.1.4. Tổng hợp yêu cầu thực thi của các đối tượng sử dụng”	83
3.2. Đánh giá kết quả.....	84
3.2.1. Đánh giá về chức năng.....	84
3.2.2. Đánh giá về phi chức năng và chất lượng kỹ thuật	85
3.2.3. Ưu nhược điểm	85
KẾT LUẬN	87
TÀI LIỆU THAM KHẢO	90

LỜI CẢM ƠN

Đề tài “Xây dựng hệ thống hỗ trợ trẻ em lập trình vẽ đối tượng với hàm thư viện Turtle của Python” là kết quả của một hành trình học tập và nghiên cứu đầy ý nghĩa tại Trường Đại học Mở - Địa chất.

Trước tiên, chúng em xin gửi lời cảm ơn chân thành nhất đến Ban Giám hiệu Nhà trường, Khoa Công nghệ thông tin và các thầy cô trong Bộ môn Công nghệ phần mềm đã tận tâm giảng dạy, trang bị cho chúng em những kiến thức chuyên môn và kỹ năng thực tiễn quý báu trong suốt những năm tháng qua.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc tới TS. Nguyễn Thị Hữu Phương. Cô không chỉ là người trực tiếp hướng dẫn mà còn là người luôn tận tình định hướng, khích lệ và truyền cảm hứng cho chúng em vượt qua những khó khăn để hoàn thành tốt đề tài này. Sự tận tâm của cô chính là kim chỉ nam cho những kết quả mà chúng em đạt được.

Cuối cùng, chúng em xin cảm ơn gia đình và bạn bè đã luôn ủng hộ, đồng hành cùng nhóm trong suốt quá trình học tập.

Mặc dù đã nỗ lực hết mình, nhưng do hạn chế về thời gian và kinh nghiệm, đề tài khó tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ quý thầy cô để nghiên cứu được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG NGHIÊN CỨU KHOA HỌC

Đề tài “*Xây dựng hệ thống hỗ trợ trẻ em học lập trình vẽ đối tượng với thư viện Turtle của Python*” nhằm phát triển một nền tảng học tập tương tác, giúp trẻ em làm quen với tư duy lập trình thông qua hoạt động vẽ đồ họa trực quan. Hệ thống được thiết kế với mục tiêu giáo dục chính là phát triển tư duy logic, khả năng sáng tạo và kỹ năng giải quyết vấn đề ở lứa tuổi thiếu nhi, đồng thời tạo hứng thú với ngôn ngữ lập trình Python một cách tự nhiên và dễ hiểu.

Về mặt chức năng, hệ thống tích hợp các tính năng hỗ trợ học tập gồm: nhập lệnh trực tiếp, kéo thả (*drag-and-drop*), thực thi chương trình và hiển thị kết quả vẽ tức thời. Việc quan sát trực tiếp mối quan hệ giữa câu lệnh và hình ảnh giúp trẻ dễ dàng tiếp thu tư duy lập trình và ghi nhớ cú pháp một cách tự nhiên. Hệ thống còn cung cấp kho bài học mẫu và bài tập được phân cấp từ cơ bản đến nâng cao, kèm theo khả năng lưu trữ và quản lý các bài vẽ, hỗ trợ quá trình ôn tập và theo dõi tiến độ học tập.

Về kiến trúc kỹ thuật, hệ thống được xây dựng dựa trên ngôn ngữ Python và thư viện Turtle, với thiết kế mô-đun linh hoạt, thuận tiện cho việc mở rộng và nâng cấp. Giao diện người dùng được tối ưu với phong cách thân thiện, màu sắc tươi sáng, phù hợp với tâm lý lứa tuổi, góp phần duy trì sự tập trung và hứng thú trong quá trình học.

Kết quả nghiên cứu cho thấy hệ thống đáp ứng tốt các yêu cầu về tính năng, hiệu quả giáo dục và trải nghiệm người dùng. Hệ thống mở ra hướng phát triển cho các tính năng nâng cao trong tương lai như cá nhân hóa lộ trình học, tích hợp yếu tố trò chơi và mở rộng sang các nền tảng học tập trực tuyến khác.

DANH MỤC CÁC HÌNH VẼ

Hình 1. Logo chính thức của ngôn ngữ lập trình Python	10
Hình 2.1 Sơ đồ Use Case tổng quát.....	29
Hình 2.2. Sơ đồ UC đăng ký	30
Hình 2.3. Sơ đồ UC đăng nhập tài khoản.....	32
Hình 2.4. Sơ đồ UC quản lý bài vẽ.....	34
Hình 2.5. Sơ đồ UC thực thi chương trình	36
Hình 2.6. Sơ đồ UC quản lý tài khoản cá nhân	38
Hình 2.7. Sơ đồ UC quản lý người dùng.....	40
Hình 2.8. Sơ đồ UC quản lý bài học.....	43
Hình 2.9. Sơ đồ hoạt động đăng ký	46
Hình 2.11. Sơ đồ hoạt động quên mật khẩu	47
Hình 2.10. Sơ đồ hoạt động đăng nhập	47
Hình 2.13. Sơ đồ hoạt động tải bài vẽ	48
Hình 2.12. Sơ đồ hoạt động lưu bài vẽ.....	48
Hình 2.15. Sơ đồ hoạt động cập nhật thông tin tài khoản	56
Hình 2.14. Sơ đồ hoạt động Xem thông tin tài khoản.....	56
Hình 2.16. Sơ đồ hoạt động đổi mật khẩu.....	57
Hình 2.17. Sơ đồ hoạt động tìm kiếm người dùng	58
Hình 2.18. Sơ đồ hoạt động xóa tài khoản người dùng.....	58
Hình 2.20. Sơ đồ hoạt động thêm bài học	59
Hình 2.19. Sơ đồ hoạt động xem chi tiết tài khoản người dùng.....	59
Hình 2.21. Sơ đồ hoạt động cập nhật bài học.....	60
Hình 2.23. Sơ đồ hoạt động nhập lệnh	61
Hình 2.22. Sơ đồ hoạt động xóa bài học	61
Hình 2.24. Sơ đồ hoạt động thực thi lệnh.....	62
Hình 2.25. Sơ đồ lớp.....	63
Hình 2.26. Sơ đồ tuần tự đăng ký.....	64
Hình 2.27. Sơ đồ tuần tự đăng nhập	65
Hình 2.29. Sơ đồ tuần tự lưu bài vẽ.....	66
Hình 2.28. Sơ đồ tuần tự quên mật khẩu	66
Hình 2.31. Sơ đồ tuần tự cập nhật tài khoản cá nhân	67

Hình 2.30. Sơ đồ tuần tự tải bài vẽ.....	67
Hình 2.32. Sơ đồ tuần tự đổi mật khẩu.....	68
Hình 2.33. Sơ đồ tuần tự tìm kiếm người dùng.....	69
Hình 2.34. Sơ đồ tuần tự xóa tài khoản người dùng.....	70
Hình 2.35. Sơ đồ tuần tự xem chi tiết tài khoản.....	70
Hình 2.36. Sơ đồ ERD.....	71
Hình 2.37. Giao diện đăng ký, đăng nhập.....	76
Hình 2.38. Giao diện màn hình chính khi đã đăng nhập.....	76
Hình 2.39. Giao diện cập nhật thông tin tài khoản.....	77
Hình 2.40. Giao diện tra cứu nội dung Wiki.....	78

DANH MỤC CÁC BẢNG BIỂU

Bảng 1. Một số lệnh cơ bản trong thư viện Turtle	12
Bảng 2.1. Bảng mô tả Use Case tổng quát	29
Bảng 2.2. Đặc tả ca sử dụng nhập thông tin đăng ký	30
Bảng 2.3. Kịch bản ca sử dụng gửi yêu cầu đăng ký	31
Bảng 2.4. Kịch bản ca sử dụng chia sẻ bài vẽ	32
Bảng 2.5. Đặc tả ca sử dụng quên mật khẩu	33
Bảng 2.6. Đặc tả ca sử dụng lưu bài vẽ	34
Bảng 2.7. Đặc tả ca sử dụng tải bài vẽ	35
Bảng 2.8. Kịch bản ca sử dụng nhập lệnh	36
Bảng 2.9. Kịch bản ca sử dụng thực thi lệnh.....	37
Bảng 2.10. Kịch bản ca sử dụng xem thông tin tài khoản.....	38
Bảng 2.11. Kịch bản ca sử dụng cập nhật thông tin tài khoản	39
Bảng 2.12. Kịch bản ca sử dụng thay đổi mật khẩu	39
Bảng 2.13. Kịch bản ca sử dụng xem danh sách người dùng.....	41
Bảng 2.14. Kịch bản ca sử dụng xem chi tiết người dùng	41
Bảng 2.15. Kịch bản ca sử dụng xoá tài khoản người dùng.....	42
Bảng 2.16. Kịch bản ca sử dụng thêm bài học	43
Bảng 2.17. Kịch bản ca sử dụng cập nhật bài học.....	44
Bảng 2.18. Kịch bản ca sử dụng xoá bài học	45
Bảng 2.19. Mô tả dữ liệu bảng auth_user.....	71
Bảng 2.20. Mô tả dữ liệu bảng auth_group	72
Bảng 2.21. Mô tả dữ liệu bảng auth_permission.....	72
Bảng 2.22. Mô tả dữ liệu bảng auth_group_permission	72
Bảng 2.23. Mô tả dữ liệu bảng auth_user_user_permission	72
Bảng 2.24. Mô tả dữ liệu bảng auth_user_groups.....	73
Bảng 2.25. Mô tả dữ liệu bảng django_content_type.....	73
Bảng 2.26. Mô tả dữ liệu bảng django_session.....	73
Bảng 2.27. Mô tả dữ liệu bảng django_admin_log	73
Bảng 2.28. Mô tả dữ liệu bảng user.....	74
Bảng 2.29. Mô tả dữ liệu bảng user_groups.....	74
Bảng 2.30. Mô tả dữ liệu bảng user_user_permission	75

Bảng 2.31. Mô tả dữ liệu bảng user_session	75
Bảng 2.32. Mô tả dữ liệu bảng lesson	75
Bảng 2.33. Mô tả dữ liệu bảng source_code	76
Bảng 3.1. Một số yêu cầu phần cứng ở máy người học	81
Bảng 3.2. Một số yêu cầu phần cứng ở máy người chủ	82
Bảng 3.3. So sánh trách nhiệm và yêu cầu cài đặt phần mềm theo vai trò	83

DANH MỤC NHỮNG TỪ VIẾT TẮT

STT	Từ viết tắt	Từ tiếng Anh	Từ và nghĩa tiếng Việt
1	ACID	Atomicity, Consistency, Isolation, Durability	Tính nguyên tử, nhất quán, cô lập, bền vững
2	API	Application Programming Interface	Giao diện lập trình ứng dụng
3	ASGI	Asynchronous Server Gateway Interface	Giao diện cổng máy chủ bất đồng bộ
4	CI/CD	Continuous Integration / Continuous Delivery	Tích hợp liên tục / Triển khai liên tục
5	CORS	Cross-Origin Resource Sharing	Chia sẻ tài nguyên đa nguồn gốc
6	CSRF	Cross-Site Request Forgery	Tấn công giả mạo yêu cầu liên trang
7	CSS	Cascading Style Sheets	Bảng định kiểu tầng
8	CSDL	—	Cơ sở dữ liệu
9	DB	Database	Cơ sở dữ liệu
10	DOM	Document Object Model	Mô hình đối tượng tài liệu
11	DRF	Django REST Framework	Khung làm việc REST của Django
12	ERD	Entity Relationship Diagram	Sơ đồ quan hệ thực thể
13	FK	Foreign Key	Khóa ngoại
14	HTTP/HTTPS	HyperText Transfer Protocol (Secure)	Giao thức truyền tải siêu văn bản (Bảo mật)
15	IDE	Integrated Development Environment	Môi trường phát triển tích hợp
16	JSON	JavaScript Object Notation	Định dạng dữ liệu JavaScript
17	JWT	JSON Web Token	Token web dạng JSON
18	MIT	Massachusetts Institute of Technology	Học viện Công nghệ Massachusetts
19	NCKH	—	Nghiên cứu khoa học

STT	Từ viết tắt	Từ tiếng Anh	Từ và nghĩa tiếng Việt
20	ORM	Object-Relational Mapping	Ánh xạ đối tượng – quan hệ
21	OTP	One-Time Password	Mật khẩu một lần
22	PK	Primary Key	Khóa chính
23	PWA	Progressive Web Application	Ứng dụng web tiên bộ
24	REST	Representational State Transfer	Kiểu kiến trúc truyền trạng thái đại diện
25	S3	Simple Storage Service	Dịch vụ lưu trữ đơn giản
26	SDLC	Software Development Life Cycle	Vòng đời phát triển phần mềm
27	STEM	Science, Technology, Engineering, Mathematics	Khoa học, Công nghệ, Kỹ thuật, Toán học
28	THCS	—	Trung học cơ sở
29	UC	Use Case	Cas sử dụng
30	UI	User Interface	Giao diện người dùng
31	UML	Unified Modeling Language	Ngôn ngữ mô hình hóa thống nhất
32	URL	Uniform Resource Locator	Định vị tài nguyên thống nhất
33	UUID	Universally Unique Identifier	Định danh duy nhất toàn cầu
34	UX	User Experience	Trải nghiệm người dùng
35	VPN	Virtual Private Network	Mạng riêng ảo
36	XSS	Cross-Site Scripting	Tấn công kịch bản liên trang

MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực đề tài

Trong bối cảnh cách mạng công nghiệp 4.0 và xu hướng giáo dục STEM trên toàn cầu, việc trang bị kỹ năng lập trình cho trẻ em từ sớm đang trở thành một nhu cầu cấp thiết. Nhiều nghiên cứu và mô hình giáo dục quốc tế như Code.org, Scratch hay Tynker đã chứng minh rằng việc tiếp cận lập trình thông qua hình ảnh và tương tác trực quan giúp trẻ phát triển tư duy logic, sáng tạo một cách tự nhiên. Tại Việt Nam, lĩnh vực giáo dục lập trình cho trẻ em đang phát triển mạnh mẽ nhưng vẫn còn những hạn chế nhất định. Các nền tảng hiện nay chủ yếu tập trung vào lập trình khối (*block-based*), trong khi việc chuyển giao sang lập trình văn bản (*text-based*) như Python vẫn còn là một thách thức lớn đối với trẻ em do tính trừu tượng của câu lệnh.

Thư viện Turtle của Python được đánh giá là một "cây cầu" phù hợp để giảm khoảng cách vấn đề này. Nhờ khả năng hiển thị kết quả đồ họa hiển thị gần như tức thời, giúp trẻ dễ liên hệ giữa câu lệnh và chuyển động trên màn hình. Tuy nhiên, hầu hết hướng dẫn hiện nay vẫn gắn với môi trường cài đặt cục bộ (Python, IDE, thư viện...) hoặc tài liệu rời rạc, thiếu một hệ thống học tập tập trung, có lộ trình và giao diện tối ưu cho tâm lý lứa tuổi tại Việt Nam. Điều đó dễ tạo rào cản kỹ thuật cho chính trẻ, phụ huynh và giáo viên cần cấu hình máy, phiên bản phần mềm, quyền cài đặt trên thiết bị trường lớp hay máy gia đình.

Xu hướng triển khai ứng dụng web mở ra hướng tiếp cận thân thiện hơn cho đối tượng nhỏ tuổi. Với hình thức này, trẻ không cần cài đặt hay cập nhật phần mềm; chỉ cần nhận một đường link (URL) và mở bằng trình duyệt là có thể vào học, gõ lệnh và xem rùa vẽ ngay trên màn hình. Những lợi ích trực tiếp đối với học sinh bao gồm: giảm thiểu thao tác phức tạp, có thể học tập tại nhà, ở lớp hoặc trên thiết bị di động (chỉ cần kết nối Internet và trình duyệt), đảm bảo trải nghiệm đồng nhất giữa các em, đồng thời rút ngắn thời gian bắt đầu bài học – phù hợp với tiết học ngắn và hoạt động ngoại khóa. Từ góc độ giáo dục, đây là giải pháp giúp hạ thấp “ngưỡng cửa” công nghệ, tạo điều kiện để học sinh tập trung vào nội dung lập trình và quan sát phản hồi trực quan, thay vì bị phân tán bởi các bước cài đặt ban đầu.

Chính vì vậy, việc nghiên cứu và xây dựng một hệ thống hỗ trợ học lập trình với Turtle Python không chỉ mang tính kế thừa các thành tựu giáo dục quốc tế mà còn đáp

ứng nhu cầu thực tiễn về một công cụ học tập lập trình văn bản trực quan, sinh động cho trẻ em.

2. Lý do lựa chọn đề tài

Thứ nhất, về nhu cầu thực tiễn: Trong bối cảnh chuyển đổi số, nhu cầu tiếp cận lập trình dành cho trẻ em ngày càng gia tăng. Tuy nhiên, nhiều công cụ hiện nay vẫn thiên về lý thuyết hoặc lập trình khối, tạo ra khoảng trống khi trẻ cần làm quen với cú pháp văn bản thực thụ như Python. Do đó, một nền tảng vừa trực quan, vừa hỗ trợ trẻ viết lệnh theo “đúng chuẩn lập trình văn bản” là hết sức cần thiết, nhằm giúp các em rèn luyện kỹ năng lập trình cơ bản và phát triển tư duy logic một cách tự nhiên.

Thứ hai, về tính phù hợp của công cụ và hình thức triển khai: Turtle có cú pháp đơn giản và phản hồi đồ họa tức thời, rất phù hợp nhập môn. Tuy nhiên, Turtle thường đi kèm môi trường lập trình truyền thống (IDE) và yêu cầu cài đặt, khó đồng bộ trong lớp đông trẻ hoặc khi phụ huynh không quen công nghệ. Vì thế, việc tích hợp Turtle vào một hệ thống học tập web hoàn chỉnh – có lộ trình, bài học, khu vực thực hành và giao diện dành cho trẻ – là hướng đi hợp lý: trẻ chỉ cần bấm vào đường link là bắt đầu học, không phải qua các bước cài Python hay cấu hình máy, từ đó tăng khả năng tiếp cận và công bằng trong học lập trình.

Thứ ba, về giá trị học thuật và ứng dụng: Đề tài là cơ hội để nhóm vận dụng tổng hợp các kiến thức từ phân tích – thiết kế hệ thống, lập trình hướng đối tượng đến thiết kế trải nghiệm người dùng (UX/UI). Sản phẩm hoàn thiện không chỉ mang ý nghĩa giáo dục mà còn có khả năng ứng dụng thực tiễn tại các câu lạc bộ, trung tâm kỹ năng hoặc trong các tiết học trải nghiệm, nhờ ưu thế triển khai trực tiếp qua nền tảng web.

Thứ tư, về động lực cá nhân: Xuất phát từ niềm đam mê với công nghệ giáo dục và mong muốn mang lại một công cụ học tập hữu ích cho trẻ em Việt Nam, nhóm mong muốn vận dụng những kiến thức đã học tại Trường Đại học Mở - Địa chất để giải quyết một vấn đề thực tế, góp phần nhỏ bé vào việc phổ cập lập trình cho thế hệ trẻ.

Xuất phát từ những lý do trên, nhóm đã quyết định lựa chọn đề tài: “Xây dựng hệ thống hỗ trợ trẻ em học lập trình vẽ đối tượng với thư viện Turtle của Python” làm nội dung nghiên cứu và phát triển.

3. Mục tiêu của đề tài

Mục tiêu tổng quát của đề tài là xây dựng một hệ thống phần mềm học lập trình trực quan dành cho trẻ em, giúp các em hình thành tư duy lập trình thông qua việc vẽ

hình bằng thư viện Turtle trong Python, từ đó phát triển tư duy logic, khả năng sáng tạo và kỹ năng giải quyết vấn đề. Hệ thống hướng tới triển khai dạng ứng dụng web: trẻ chỉ cần truy cập bằng đường link trên trình duyệt là có thể học và thực hành, không bắt buộc cài đặt Python, IDE hay các gói phần mềm phụ trên máy cục bộ, nhằm giảm rào cản kỹ thuật, thuận tiện cho giờ học tại lớp, học nhóm và học tại nhà.

Để đạt được mục tiêu này, đề tài tập trung vào một số định hướng cụ thể. Trước hết, về nghiên cứu, hệ thống được xây dựng trên cơ sở tìm hiểu lý thuyết về giáo dục lập trình cho trẻ em và các phương pháp giảng dạy trực quan, đồng thời nghiên cứu chuyên sâu thư viện Turtle để chọn lọc các tập lệnh phù hợp với lứa tuổi. Về thiết kế, hệ thống được phân tích theo yêu cầu người dùng và triển khai kiến trúc mô-đun linh hoạt, với giao diện thân thiện, màu sắc sinh động và thao tác tối ưu cho trẻ em. Về phát triển, hệ thống cung cấp các tính năng cốt lõi như nhập lệnh trực tiếp, hỗ trợ kéo thả, thực thi chương trình, hiển thị kết quả đồ họa tức thời, đồng thời tích hợp các bài học và ví dụ minh họa; phần triển khai kỹ thuật ưu tiên mô hình client-server trên nền web để dễ chia sẻ đường dẫn truy cập, đồng bộ trải nghiệm giữa các thiết bị có trình duyệt và thuận lợi khi mở rộng quy mô sử dụng. Về nội dung, hệ thống xây dựng kho bài học mẫu và bài tập thực hành được phân cấp từ cơ bản đến nâng cao, hỗ trợ quá trình học tập theo từng bước. Về quản lý, hệ thống cho phép lưu trữ và quản lý sản phẩm vẽ cá nhân, đồng thời theo dõi lộ trình tiến bộ của người học. Về đánh giá, hệ thống được kiểm thử và đánh giá hiệu quả dựa trên các tiêu chí về tính năng kỹ thuật, trải nghiệm người dùng và giá trị hỗ trợ giáo dục. Cuối cùng, đề tài cũng đề xuất các hướng phát triển và mở rộng hệ thống trong tương lai nhằm nâng cao chất lượng và phạm vi ứng dụng.

4. Cách tiếp cận và phương pháp nghiên cứu

Đề tài được triển khai dựa trên nhiều cách tiếp cận khác nhau nhằm đảm bảo tính toàn diện và phù hợp với đối tượng người dùng. Trước hết, nhóm nghiên cứu áp dụng cách tiếp cận theo vòng đời phát triển hệ thống (SDLC), thực hiện đầy đủ các giai đoạn từ khảo sát, phân tích yêu cầu, thiết kế kiến trúc, cài đặt chương trình cho đến kiểm thử và vận hành. Đồng thời, đề tài lấy người dùng làm trung tâm (*User-Centered Design*), đặt trẻ em vào vị trí trọng tâm để nghiên cứu đặc điểm tâm lý, khả năng nhận thức và hành vi tương tác, từ đó tối ưu hóa giao diện và trải nghiệm người dùng. Bên cạnh đó, cách tiếp cận kết hợp lý thuyết và thực tiễn cũng được áp dụng, vận dụng các học thuyết

về giáo dục lập trình trực quan để xây dựng sản phẩm, đồng thời tiến hành thực nghiệm nhằm điều chỉnh và hoàn thiện các chức năng.

Về phương pháp nghiên cứu, đề tài sử dụng phương pháp nghiên cứu tài liệu để thu thập, tổng hợp và phân tích các tài liệu khoa học liên quan đến giáo dục STEM, lập trình Turtle và các nền tảng học tập tương tự trên thế giới. Tiếp đó, phương pháp phân tích và thiết kế hệ thống được áp dụng với ngôn ngữ mô hình hóa thống nhất UML (Use Case Diagram, Activity Diagram, Sequence Diagram) nhằm mô tả cấu trúc và hành vi của hệ thống. Ngoài ra, nhóm nghiên cứu tham khảo ý kiến chuyên gia, đặc biệt là sự hướng dẫn của giảng viên, đồng thời khảo sát các mô hình học tập hiện có để xác định tập chức năng tối ưu. Trong quá trình phát triển, phương pháp lập trình thực nghiệm được sử dụng với ngôn ngữ Python kết hợp các thư viện hỗ trợ như Tkinter cho desktop hoặc Flask/Django/Brython nếu triển khai trên web, nhằm hiện thực hóa các mô-đun chức năng. Cuối cùng, phương pháp thử nghiệm và đánh giá được tiến hành thông qua kiểm thử hộp đen (*Black-box testing*) để đảm bảo tính đúng đắn của chức năng, đồng thời khảo sát mức độ hài lòng của trẻ em khi trực tiếp tương tác với hệ thống.

5. Đối tượng và phạm vi nghiên cứu

Đề tài được triển khai dựa trên nhiều cách tiếp cận nhằm đảm bảo tính toàn diện và phù hợp với đối tượng người dùng. Trước hết, nhóm nghiên cứu áp dụng cách tiếp cận theo vòng đời phát triển phần mềm (SDLC), thực hiện đầy đủ các giai đoạn từ khảo sát, phân tích yêu cầu, thiết kế kiến trúc, cài đặt chương trình đến kiểm thử và vận hành. Đồng thời, đề tài lấy người dùng làm trung tâm (User-Centered Design), đặt trẻ em vào vị trí trọng tâm để phân tích đặc điểm tâm lý, khả năng nhận thức và hành vi tương tác – trong đó kênh sử dụng chính là trình duyệt web qua đường dẫn truy cập, không yêu cầu cài đặt môi trường lập trình hay thư viện trên máy người học, nhằm giảm rào cản kỹ thuật và tập trung vào hoạt động học tập. Bên cạnh đó, cách tiếp cận kết hợp lý thuyết và thực tiễn được áp dụng: vận dụng các học thuyết về giáo dục lập trình trực quan để xây dựng sản phẩm, đồng thời tổ chức thử nghiệm và điều chỉnh chức năng theo phản hồi thực tế.

Về phương pháp nghiên cứu, đề tài sử dụng nghiên cứu tài liệu để thu thập, tổng hợp và phân tích các công trình khoa học liên quan đến giáo dục STEM, lập trình trực quan, thư viện Turtle của Python và các nền tảng học tập tương tự. Tiếp theo, phương pháp phân tích và thiết kế hệ thống được áp dụng với ngôn ngữ mô hình hóa thống nhất

UML (sơ đồ Use Case, Activity, Sequence, ...) nhằm mô tả cấu trúc và hành vi của hệ thống. Ngoài ra, nhóm tham khảo ý kiến chuyên gia, đặc biệt là giảng viên hướng dẫn, đồng thời khảo sát các mô hình học tập hiện có để xác định tập chức năng phù hợp.

Trong giai đoạn hiện thực hóa, đề tài lựa chọn kiến trúc ứng dụng web tách lớp: tầng backend xây dựng bằng Python với framework Django, tích hợp Django REST Framework phục vụ API. Cơ sở dữ liệu sử dụng PostgreSQL; Redis phục vụ bộ nhớ đệm và hạ tầng hàng đợi; Celery (worker/beat) hỗ trợ các tác vụ nền theo thiết kế hệ thống. Tầng frontend xây dựng bằng React, đóng gói và phát triển với Vite, tương tác với backend qua HTTP/HTTPS (và WebSocket khi có kênh real-time), để người học chỉ cần truy cập bằng đường link trên trình duyệt là có thể đăng nhập, học bài và thực hành. Toàn bộ phần xử lý nặng và lưu trữ dữ liệu được đặt phía máy chủ, phù hợp với định hướng không cài đặt phần mềm lập trình trên máy trẻ.

Cuối cùng, phương pháp thử nghiệm và đánh giá được tiến hành thông qua kiểm thử hộp đen (Black-box testing) nhằm kiểm chứng tính đúng đắn của các chức năng theo đặc tả; đồng thời quan sát và khảo sát trải nghiệm người dùng khi thao tác trên giao diện web (độ dễ dùng, mức độ hứng thú, khả năng hoàn thành nhiệm vụ học tập), làm cơ sở đánh giá hiệu quả giáo dục và hiệu quả kỹ thuật của hệ thống.

6. Đóng góp của đề tài

Đề tài mang ý nghĩa ở nhiều khía cạnh: Từ học thuật, sư phạm cho đến thực tiễn giáo dục và xã hội. Trong bối cảnh giáo dục Việt Nam đang chuyển mình mạnh mẽ theo hướng đổi mới phương pháp dạy và học, đặc biệt là với trẻ em, đề tài góp phần làm rõ hướng tiếp cận mới – học thông qua trải nghiệm và tương tác trực tuyến.

Về mặt sư phạm và tư duy: Đề xuất và hiện thực hóa phương pháp học lập trình trực quan, giúp giảm tính trừu tượng của ngôn ngữ văn bản. Hệ thống giúp trẻ em tiếp cận các khái niệm cốt lõi (vòng lặp, điều kiện, hàm) một cách tự nhiên thông qua hình ảnh, từ đó khơi dậy niềm đam mê công nghệ theo hướng "học mà chơi, chơi mà học".

Về mặt kỹ thuật và giải pháp: Xây dựng một hệ thống phần mềm hoàn chỉnh với kiến trúc mô-đun linh hoạt. Điểm khác biệt lớn nhất là việc tích hợp trình soạn thảo, khung hiển thị kết quả và kho bài học vào một nền tảng duy nhất, thay vì sử dụng các công cụ rời rạc, giúp tối ưu hóa quy trình học tập cho trẻ.

Về mặt thực tiễn và xã hội: Sản phẩm có khả năng ứng dụng cao tại các trường học, câu lạc bộ STEM hoặc trung tâm đào tạo kỹ năng. Kết quả nghiên cứu còn đóng

vai trò là tài liệu tham khảo giá trị cho các dự án phát triển phần mềm giáo dục tương lai tại Việt Nam.

7. Cấu trúc đồ án

Đồ án “Xây dựng hệ thống hỗ trợ trẻ em học lập trình vẽ đối tượng với thư viện Turtle của Python” được bố cục thành ba chương chính cùng phần kết luận và hướng phát triển. Mỗi chương đảm nhận một vai trò cụ thể, phản ánh tiến trình nghiên cứu, phân tích, thiết kế và triển khai hệ thống.

- Chương 1 – Cơ sở lý thuyết và tổng quan về đề tài: Giáo dục STEM, tư duy lập trình cho trẻ, ngôn ngữ Python và đặc điểm thư viện Turtle.
- Chương 2 – Phân tích và thiết kế hệ thống: Phân tích yêu cầu chức năng/phi chức năng, sơ đồ UML, thiết kế kiến trúc mô-đun và giao diện UI/UX.
- Chương 3 – Cài đặt, thử nghiệm và đánh giá kết quả: Môi trường triển khai, demo các chức năng hệ thống, kết quả thử nghiệm và đánh giá hiệu quả.
- Kết luận – Tóm tắt kết quả, hạn chế, hướng phát triển.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN

1.1. Tổng quan về giáo dục lập trình cho trẻ em

1.1.1. Bối cảnh và xu hướng toàn cầu

Trong bối cảnh cách mạng công nghiệp 4.0, giáo dục STEM đã trở thành trọng tâm của các chính sách giáo dục trên phạm vi toàn cầu. Việc trang bị kỹ năng lập trình cho trẻ em từ sớm được xem là yếu tố then chốt nhằm phát triển tư duy thuật toán, khả năng giải quyết vấn đề và năng lực sáng tạo [1]. Các nghiên cứu giáo dục đã chứng minh rằng việc tiếp cận lập trình thông qua giao diện trực quan giúp trẻ em hình thành các khái niệm trừu tượng một cách hiệu quả hơn thông qua tương tác trực tiếp với kết quả đồ họa [2].

Tại Việt Nam, lĩnh vực giáo dục lập trình cho trẻ em đang phát triển mạnh mẽ trong những năm gần đây. Tuy nhiên, phần lớn các nền tảng hiện có vẫn tập trung vào mô hình lập trình khối, trong khi việc chuyển giao sang lập trình văn bản như Python vẫn còn là thách thức đáng kể đối với trẻ em do tính trừu tượng của cú pháp ngôn ngữ [3].

1.1.2. Các mô hình giáo dục lập trình phổ biến

Hiện nay, các mô hình giáo dục lập trình cho trẻ em thường có xu hướng chuyển đổi từ các dòng lệnh văn bản phức tạp sang dạng khối hoặc đồ họa trực quan để phù hợp với khả năng nhận thức của lứa tuổi nhỏ. Dưới đây là các mô hình tiêu biểu:

- **Mô hình Logo (Seymour Papert, 1967):** Được coi là nền tảng đầu tiên áp dụng phương pháp "tư duy bằng hình ảnh" (*visual thinking*) trong giáo dục lập trình cho trẻ em. Mô hình này dựa trên lý thuyết kiến tạo (*Constructionism*), cho phép người học điều khiển một nhân vật đồ họa (*turtle*) trên màn hình để vẽ hình học, từ đó hình thành các khái niệm cơ bản về lập trình [4]. Logo được xem là tiền đề trực tiếp của thư viện Turtle trong Python hiện đại.
- **Mô hình Scratch (MIT Media Lab, 2007):** Là ngôn ngữ lập trình trực quan phổ biến nhất thế giới dành cho trẻ em từ 8-16 tuổi. Scratch sử dụng phương pháp lập trình khối, cho phép trẻ kéo thả các khối lệnh màu sắc để tạo ra câu chuyện, trò chơi và hoạt hình [5]. Ưu điểm nổi bật của Scratch là loại bỏ lỗi cú pháp, giúp trẻ tập trung vào logic lập trình. Tính đến năm 2024, Scratch đã có hơn 100 triệu tài khoản người dùng trên toàn cầu.

- **Mô hình Blockly (Google, 2012):** Là thư viện mã nguồn mở cho phép xây dựng các trình biên tập lập trình trực quan dựa trên khối. Khác với Scratch, Blockly có khả năng chuyển đổi từ khối lệnh sang các ngôn ngữ lập trình văn bản thực tế như Python, JavaScript hay PHP, tạo điều kiện thuận lợi cho quá trình chuyển tiếp từ tư duy khối sang viết mã thuần [6].
- **Nền tảng Code.org (2013):** Là nền tảng giáo dục phi lợi nhuận cung cấp lộ trình học tập từ mầm non đến phổ thông. Nền tảng này nổi bật nhờ tích hợp các nhân vật nổi tiếng từ trò chơi và phim ảnh (Minecraft, Frozen) vào bài học, duy trì hứng thú thông qua hệ thống bài tập có hướng dẫn chi tiết theo từng cấp độ [7].
- **Scratch (MIT Media Lab):** Đây là ngôn ngữ lập trình trực quan phổ biến nhất thế giới dành cho trẻ từ 8-16 tuổi. Trẻ thực hiện lập trình bằng cách kéo thả các khối lệnh màu sắc để tạo ra các câu chuyện, trò chơi và hoạt hình. Ưu điểm lớn nhất là loại bỏ được lỗi cú pháp và giúp trẻ phát triển tư duy logic, sáng tạo thông qua một cộng đồng chia sẻ dự án rộng lớn.
- **Các nền tảng khác:** Tynker (học qua dự án thực tế), Alice (lập trình không gian 3D), App Inventor (phát triển ứng dụng di động) cũng góp phần làm phong phú hệ sinh thái giáo dục lập trình cho trẻ em [8].

1.1.3. Nguyên tắc sư phạm trong lập trình trực quan

Để việc giảng dạy và học tập đạt hiệu quả tối ưu, các hệ thống lập trình trực quan dành cho trẻ em cần tuân thủ bốn nguyên tắc cốt lõi. Thứ nhất là tính trực quan (*Visualisation*), bởi trẻ em có khả năng tư duy cụ thể tốt hơn tư duy trừu tượng. Do đó, mã nguồn cần được thiết kế dưới dạng các khối màu sắc phân biệt theo chức năng, đồng thời kết quả đồ họa được hiển thị ngay lập tức để trẻ dễ dàng quan sát luồng thực thi của chương trình. Thứ hai là tính tương tác (*Interactivity*), nghĩa là hệ thống phải phản hồi nhanh chóng với mọi hành động của trẻ, chẳng hạn việc điều chỉnh tham số làm thay đổi hình dáng đối tượng ngay tức thì hoặc các khối lệnh chỉ có thể ghép nối khi đúng logic, qua đó tạo ra quá trình học tập hai chiều liên tục. Thứ ba là nguyên tắc vui học (*Gamification*), áp dụng tư tưởng “Hard fun” – vừa vui vừa đầy thách thức. Việc tích hợp hệ thống cấp độ, huy hiệu và các nhiệm vụ dưới dạng trò chơi giúp duy trì động lực học tập lâu dài, biến các bài học khô khan thành trải nghiệm thú vị. Thứ tư là khám phá và thử – sai (*Discovery & Trial-and-Error*), khuyến khích trẻ coi lỗi là một phần tất yếu

của quá trình học tập. Môi trường lập trình cần được thiết kế như một “hộp cát” (*sandbox*) an toàn, nơi trẻ có thể tự do thử nghiệm, sáng tạo và mày mò các chức năng mới mà không sợ thất bại.

Bên cạnh các nguyên tắc trên, quá trình giảng dạy còn được hỗ trợ bởi những phương pháp sư phạm hiện đại như học từ đơn giản đến phức tạp (*Scaffolding*), học qua dự án (*Project-Based Learning*), học cộng tác (*Collaborative Learning*) và học gắn với thực tiễn (*Contextual Learning*). Các nguyên tắc này sẽ là kim chỉ nam cho việc thiết kế giao diện và luồng trải nghiệm người dùng trong hệ thống đề tài, đảm bảo trẻ em luôn cảm thấy an toàn, vui vẻ và được khuyến khích sáng tạo.

1.1.4. Đặc điểm tâm lý trẻ em (Tiểu học - THCS)

Việc hiểu rõ đặc điểm tâm lý và năng lực nhận thức theo từng lứa tuổi đóng vai trò quan trọng trong thiết kế nội dung và phương pháp giảng dạy lập trình. Dựa trên thuyết phát triển nhận thức của Jean Piaget, khả năng tư duy của trẻ thay đổi theo từng giai đoạn phát triển, từ tư duy cụ thể ở bậc Tiểu học đến tư duy trừu tượng ở bậc THCS.

Ở giai đoạn Tiểu học (6–11 tuổi), trẻ em thường dựa vào hình ảnh và đồ vật trực quan để tiếp nhận kiến thức. Khả năng tập trung ngắn, chỉ khoảng 15–20 phút, khiến việc tiếp cận các khái niệm trừu tượng như biến số phức tạp hay hàm trở nên khó khăn. Về mặt tâm lý, trẻ hiếu động, ham chơi và rất nhạy cảm với sự khích lệ, vì vậy cần một môi trường học tập an toàn, nơi lỗi sai không bị coi là thất bại mà được xem như cơ hội thử nghiệm. Phương pháp tiếp cận phù hợp trong giai đoạn này là sử dụng các nền tảng lập trình khối, kết hợp với bài học ngắn gọn dưới dạng trò chơi hoặc hoạt hình có phản hồi tức thì, giúp duy trì hứng thú và khuyến khích trẻ tham gia tích cực.

Khi bước sang giai đoạn THCS (12–14 tuổi), trẻ bắt đầu phát triển tư duy trừu tượng và khả năng phân tích logic rõ rệt hơn. Thời gian tập trung có thể kéo dài từ 30–45 phút, cho phép tiếp cận các khái niệm phức tạp như hàm, mảng và thuật toán phân nhánh. Trẻ cũng có khả năng tự gỡ lỗi tốt hơn, đồng thời tâm lý hướng đến sự độc lập, tự quyết và mong muốn tham gia vào các dự án mang tính thực tế hoặc thể hiện bản sắc cá nhân. Đây là thời điểm thích hợp để chuyển tiếp từ lập trình khối sang lập trình văn bản, với các dự án dài hạn, có tính ứng dụng cao và hoạt động học tập cộng tác, nhằm phát huy tối đa năng lực tư duy và sự sáng tạo của trẻ.

Mặc dù các mô hình lập trình khối đã thành công trong việc dạy tư duy lập trình cơ bản, việc chuyển tiếp sang lập trình văn bản vẫn là thách thức lớn đối với trẻ em.

Nghiên cứu của Wilson và Lowe (2020) chỉ ra rằng khoảng 65% học sinh trong độ tuổi 10 – 14 gặp khó khăn khi chuyển từ Scratch sang Python do sự khác biệt về cú pháp và tư duy trừu tượng [9]. Trong bối cảnh giáo dục Việt Nam, các nền tảng lập trình cho trẻ em chủ yếu tập trung vào mô hình khối, trong khi nhu cầu tiếp cận lập trình văn bản ngày càng tăng cao. Phần lớn các hướng dẫn hiện nay vẫn mang tính chất tài liệu tham khảo rời rạc, thiếu một hệ thống học tập tập trung, có lộ trình và giao diện được tối ưu hóa riêng cho tâm lý lứa tuổi trẻ em tại Việt Nam.

Chính vì vậy, việc nghiên cứu và xây dựng một hệ thống hỗ trợ học lập trình với Turtle Python không chỉ mang tính kế thừa các thành tựu giáo dục quốc tế mà còn đáp ứng nhu cầu thực tiễn về một công cụ học tập lập trình văn bản trực quan, sinh động cho trẻ em.

1.2. Ngôn ngữ lập trình Python và thư viện đồ họa Turtle

Trong thiết kế các nền tảng giáo dục lập trình cho trẻ em, việc lựa chọn ngôn ngữ giảng dạy nền tảng đóng vai trò quyết định đến mức độ hứng thú và khả năng tiếp thu của người học. Thay vì phải đối mặt với những ngôn ngữ có cú pháp phức tạp dễ gây nản chí, trẻ em cần một công cụ đủ đơn giản để làm quen với tư duy logic, nhưng cũng phải mang lại các phản hồi đồ họa trực quan để duy trì động lực học tập. Xuất phát từ yêu cầu khắt khe về mặt sư phạm đó, đề tài lựa chọn ngôn ngữ lập trình Python kết hợp thư viện đồ họa Turtle làm công cụ cốt lõi để xây dựng lộ trình tương tác cho hệ thống.

1.2.1. Ngôn ngữ lập trình Python



Hình 1. Logo chính thức của ngôn ngữ lập trình Python

[Nguồn:python.org]

Python là ngôn ngữ lập trình bậc cao, đa năng và hướng đối tượng, được Guido van Rossum phát hành lần đầu tiên vào năm 1991. Với triết lý thiết kế ưu tiên sự rõ ràng, Python hỗ trợ nhiều mô hình lập trình như lập trình cấu trúc, hướng đối tượng và chức năng. Nhờ thư viện tiêu chuẩn toàn diện – thường được mô tả là ngôn ngữ "bao gồm

pin" (*batteries included*). Python được ứng dụng rộng rãi trong các lĩnh vực như trí tuệ nhân tạo, phân tích dữ liệu, phát triển web và đặc biệt là giáo dục.

Python được lựa chọn là ngôn ngữ lập trình chủ đạo cho hệ thống nhờ sự cân bằng giữa tính đơn giản và sức mạnh kỹ thuật, cụ thể qua các đặc điểm sau:

- **Tối ưu cho tư duy (Logic-First):** Cú pháp của Python rất gần với ngôn ngữ tự nhiên (tiếng Anh), loại bỏ các ký tự rườm rà như dấu chấm phẩy (;) hay ngoặc nhọn ({}). Đặc điểm bắt buộc thực lẽ không chỉ giúp mã nguồn sạch hơn mà còn hình thành thói quen lập trình tốt cho trẻ ngay từ đầu, giúp người học tập trung vào giải thuật thay vì lỗi cú pháp.
- **Giảm thiểu độ phức tạp:** So với các ngôn ngữ truyền thống như Java hay C++, Python có thể giảm tới 80% độ phức tạp về mặt cú pháp. Điều này giúp trẻ em dễ dàng ghi nhớ và thực hiện các đoạn mã phức tạp với số lượng dòng lệnh tối giản.
- **Tính tương tác và phản hồi tức thì:** Là một ngôn ngữ thông dịch, Python cho phép chạy từng dòng lệnh mà không cần qua các bước biên dịch phức tạp. Đặc tính này khuyến khích trẻ tự do thử nghiệm, giúp việc gỡ lỗi trở thành một phần trải nghiệm học tập tự nhiên và thú vị.

1.2.2. Thư viện đồ họa Turtle

Thư viện Turtle Graphics là một thành phần được tích hợp sẵn trong thư viện chuẩn của Python, có nguồn gốc từ ngôn ngữ Logo được phát triển bởi Seymour Papert và nhóm tại MIT trong những năm 1960 [4]. Nguyên lý hoạt động dựa trên mô hình "con rùa vẽ tranh": người lập trình điều khiển một con rùa đồ họa di chuyển trên màn hình theo các lệnh, để lại dấu vết tạo thành hình học.

Trong Python, Turtle được triển khai như một module đối tượng với các phương thức cơ bản bao gồm:

- Di chuyển: `forward()`, `backward()`, `goto()`
- Xoay: `left()`, `right()`
- Điều khiển bút vẽ: `pendown()`, `penup()`
- Thuộc tính đồ họa: `color()`, `pensize()`, `speed()`
- Vẽ hình học: `circle()`, `dot()`

Bảng 1. Một số lệnh cơ bản trong thư viện Turtle

Tên lệnh	Chức năng	Ví dụ
<code>forward(x)</code>	Tiến x bước	<code>t.forward(100)</code>
<code>backward(x)</code>	Lùi x bước	<code>t.backward(50)</code>
<code>right(a)</code>	Quay phải a độ	<code>t.right(90)</code>
<code>left(a)</code>	Quay trái a độ	<code>t.left(45)</code>
<code>penup()</code>	Nhấc bút (ngừng vẽ)	<code>t.penup()</code>
<code>pendown()</code>	Hạ bút (tiếp tục vẽ)	<code>t.pendown()</code>
<code>color("mau")</code>	Đặt màu nét vẽ	<code>t.color("red")</code>
<code>pensize(n)</code>	Độ dày nét vẽ	<code>t.pensize(3)</code>
<code>speed(n)</code>	Tốc độ (0-10)	<code>t.speed(5)</code>
<code>circle(r)</code>	Vẽ hình tròn bán kính r	<code>t.circle(50)</code>
<code>goto(x, y)</code>	Di chuyển đến toạ độ x,y	<code>t.goto(100, 100)</code>
<code>write("text")</code>	Viết chữ trên màn hình	<code>t.write("Xin chào")</code>

Trong giáo dục lập trình cho trẻ em, Turtle mang lại nhiều ưu điểm nổi bật giúp quá trình học tập trở nên dễ tiếp cận và hiệu quả hơn. Trước hết, tính trực quan tức thời của Turtle cho phép mỗi dòng lệnh đều tạo ra phản hồi đồ họa ngay lập tức, giúp trẻ dễ dàng liên hệ giữa cú pháp và kết quả hành động; theo nghiên cứu của Liu và cộng sự (2021), việc nhìn thấy kết quả trực quan có thể giảm tới 40% thời gian để học sinh tiểu học hiểu các khái niệm lập trình cơ bản [10]. Bên cạnh đó, cú pháp đơn giản, gần gũi với ngôn ngữ tự nhiên, chẳng hạn như lệnh `forward(100)`, giúp trẻ tập trung vào tư duy logic thay vì phải ghi nhớ cú pháp phức tạp, từ đó giảm bớt rào cản ban đầu khi tiếp cận lập trình. Không chỉ dừng lại ở mức cơ bản, Turtle còn có khả năng mở rộng khi hỗ trợ đầy đủ các cấu trúc lập trình như vòng lặp (`for`, `while`), rẽ nhánh (`if-else`) và hàm (`function`), tạo nền tảng vững chắc để học sinh chuyển tiếp sang các nội dung nâng cao của Python. Một ưu điểm khác là tính tương tác cao: Python là ngôn ngữ thông dịch, cho phép thực thi từng dòng lệnh mà không cần biên dịch trước, khuyến khích trẻ thử nghiệm, sáng tạo và phát triển tư duy khám phá. Nhờ những đặc điểm này, Turtle không chỉ là công cụ khởi đầu thân thiện mà còn đóng vai trò cầu nối quan trọng, giúp trẻ em hình thành nền tảng lập trình vững chắc và nuôi dưỡng hứng thú lâu dài với khoa học máy tính.

Trong bối cảnh giáo dục hiện nay, việc sử dụng Turtle thuần túy trong các IDE truyền thống như IDLE hay PyCharm tuy có nhiều ưu điểm về tính trực quan và khả năng giúp học sinh tiếp cận lập trình một cách sinh động, nhưng vẫn tồn tại những hạn chế đáng kể. Trước hết, giao diện của các IDE này vốn được thiết kế cho người dùng trưởng thành, chứa nhiều tính năng phức tạp và không cần thiết, khiến trẻ em dễ bị rối khi thao tác và khó tập trung vào nội dung học tập chính. Thêm vào đó, việc thiếu một hệ thống bài học có lộ trình rõ ràng khiến người học phải tự tìm kiếm tài liệu, dẫn đến tình trạng học tập rời rạc, thiếu sự hướng dẫn từ cơ bản đến nâng cao. Một hạn chế khác là không có cơ chế theo dõi tiến độ, khiến giáo viên và phụ huynh khó đánh giá sự tiến bộ của học sinh một cách hệ thống và khoa học. Bên cạnh đó, khả năng chia sẻ và cộng tác cũng bị giới hạn, do các IDE không tích hợp tính năng lưu trữ đám mây, chia sẻ dự án hay học tập nhóm, làm giảm cơ hội học sinh được trao đổi, thảo luận và phát triển kỹ năng làm việc nhóm. Những yếu tố này cho thấy khoảng trống lớn trong việc ứng dụng Turtle vào giáo dục trẻ em, đồng thời đặt ra nhu cầu cấp thiết về một hệ thống học tập tích hợp, được thiết kế chuyên biệt cho lứa tuổi này, nhằm vừa đảm bảo tính trực quan, vừa có lộ trình học tập rõ ràng, có khả năng theo dõi tiến độ và hỗ trợ chia sẻ, cộng tác hiệu quả.

1.3. Kiến trúc hệ thống Web-based Client-Server

Trong bối cảnh giáo dục hiện đại, việc lựa chọn kiến trúc ứng dụng nền web (*Web-based Application*) để phát triển hệ thống học tập cho trẻ em không chỉ xuất phát từ những ưu điểm kỹ thuật mà còn gắn liền với nhu cầu thực tiễn của người dùng. Trước hết, tính khả tiếp cận cao là yếu tố then chốt: trẻ em và giáo viên có thể sử dụng hệ thống từ bất kỳ thiết bị nào có trình duyệt web, bao gồm máy tính, máy tính bảng hay điện thoại thông minh, mà không cần cài đặt môi trường lập trình phức tạp. Điều này đặc biệt hữu ích trong các trường học hoặc câu lạc bộ nơi cơ sở hạ tầng công nghệ còn hạn chế. Bên cạnh đó, việc đồng bộ hóa dữ liệu tập trung trên máy chủ cho phép lưu trữ toàn bộ tiến độ học tập, mã nguồn và kết quả, giúp học sinh có thể tiếp tục học ở bất kỳ đâu và trên bất kỳ thiết bị nào, đồng thời tạo điều kiện cho giáo viên dễ dàng theo dõi và quản lý. Kiến trúc Client-Server cũng mang lại khả năng mở rộng và bảo trì hiệu quả, nhờ sự phân tách rõ ràng giữa giao diện người dùng (*Frontend*) và xử lý logic (*Backend*), giúp việc nâng cấp hay bổ sung tính năng mới không ảnh hưởng đến toàn bộ hệ thống. Một điểm quan trọng khác là bảo mật dữ liệu: mã nguồn của người học được thực thi trong

môi trường cách ly trên máy chủ, giảm thiểu rủi ro bảo mật và đảm bảo an toàn cho hệ thống. Như vậy, kiến trúc Web-based không chỉ đáp ứng yêu cầu về mặt kỹ thuật mà còn phù hợp với định hướng giáo dục hiện đại, nơi việc học tập cần được cá nhân hóa, dễ tiếp cận, an toàn và có khả năng mở rộng lâu dài.

Hệ thống được thiết kế theo mô hình Client-Server phân tầng với hai thành phần chính là Frontend và Backend. Ở phía Frontend, hệ thống đóng vai trò là giao diện tương tác trực tiếp với người dùng, đặc biệt là trẻ em, cung cấp môi trường soạn thảo mã nguồn với hỗ trợ cú pháp, hiển thị kết quả đồ họa Turtle theo thời gian thực thông qua canvas hoặc iframe, đồng thời quản lý trải nghiệm người dùng với giao diện thân thiện và màu sắc sinh động. Trong khi đó, Backend đảm nhiệm xử lý logic nghiệp vụ cốt lõi như xác thực người dùng, phân quyền, quản lý bài học, đồng thời cung cấp môi trường thực thi mã Python an toàn trong sandbox với giới hạn tài nguyên, lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu, cũng như cung cấp RESTful API để giao tiếp với Frontend.

Luồng dữ liệu của hệ thống diễn ra tuần tự: người dùng nhập mã lệnh vào trình soạn thảo → Frontend thực hiện xác thực cú pháp sơ bộ → Gửi mã nguồn đến Backend qua API → Backend tiến hành thực thi trong môi trường sandbox, thu thập kết quả đồ họa → Kết quả được trả về cho Frontend để hiển thị tức thời → Hệ thống ghi nhận tiến độ học tập vào cơ sở dữ liệu.

Kiến trúc Client-Server mang lại nhiều ưu điểm quan trọng trong việc xây dựng hệ thống học tập trực tuyến cho trẻ em. Trước hết, khả năng mở rộng ngang (*Horizontal Scalability*) cho phép hệ thống dễ dàng đáp ứng khi số lượng người dùng tăng lên bằng cách bổ sung thêm máy chủ Backend và sử dụng load balancer để phân phối tải, đảm bảo hiệu năng ổn định. Bên cạnh đó, yếu tố bảo mật được tăng cường nhờ việc mã nguồn của người dùng chỉ được thực thi trên máy chủ trong môi trường cách ly, hạn chế tối đa nguy cơ truy cập trái phép vào thiết bị máy khách. Hệ thống cũng hỗ trợ quản lý tập trung, khi toàn bộ dữ liệu học tập được đồng bộ hóa, giúp việc sao lưu, phân tích và đánh giá trở nên thuận tiện, đồng thời tạo điều kiện cho giáo viên theo dõi tiến độ học tập một cách hệ thống. Cuối cùng, kiến trúc này cho phép nâng cấp độc lập giữa Frontend và Backend, mỗi tầng có thể phát triển bằng công nghệ phù hợp mà không ảnh hưởng đến phần còn lại, từ đó tăng tính linh hoạt và khả năng duy trì lâu dài của hệ thống.

1.4. Các công nghệ nền tảng liên quan

1.4.1. Công nghệ giao diện người dùng

Trong hệ thống học tập trực tuyến cho trẻ em, ba công nghệ chính được lựa chọn cho tầng Frontend mang lại những lợi thế khác nhau và bổ trợ lẫn nhau. Trước hết, **React.js** – thư viện JavaScript mã nguồn mở do Meta phát triển – được sử dụng nhờ kiến trúc component-based và cơ chế Virtual DOM. Nhờ đó, giao diện người dùng có thể được xây dựng một cách động, hiệu suất cao, chỉ cập nhật những phần cần thiết khi dữ liệu thay đổi. Đây là yếu tố đặc biệt phù hợp với yêu cầu hiển thị kết quả đồ họa Turtle tức thời, giúp trẻ em quan sát trực tiếp sự thay đổi và duy trì hứng thú học tập [11].

Tiếp theo, **Tailwind CSS** đóng vai trò là framework CSS utility-first, hỗ trợ xây dựng giao diện nhanh chóng thông qua hệ thống class có sẵn. Việc chuẩn hóa bảng màu, spacing và typography giúp đảm bảo tính nhất quán, dễ bảo trì và giảm thiểu sai sót trong thiết kế. Tailwind cũng đặc biệt thích hợp cho việc phát triển giao diện đáp ứng (*responsive*), cho phép hệ thống hiển thị tốt trên nhiều kích thước màn hình khác nhau, từ máy tính đến thiết bị di động. Điều này không chỉ nâng cao trải nghiệm người dùng mà còn mở rộng khả năng tiếp cận của hệ thống trong môi trường giáo dục đa dạng.

Cuối cùng, **CodeMirror 6** hoặc **Monaco Editor** được lựa chọn làm thư viện trình soạn thảo mã nguồn nhúng cao cấp, mang lại trải nghiệm lập trình chuyên nghiệp ngay trong môi trường học tập trực tuyến. Các tính năng nổi bật bao gồm syntax highlighting cho Python, autocompletion và IntelliSense, real-time linting để phát hiện lỗi cú pháp tức thì, cùng với khả năng tìm kiếm – thay thế nâng cao và hỗ trợ multi-cursor editing. Những tính năng này không chỉ giúp trẻ em giảm thiểu lỗi khi viết mã mà còn khuyến khích tư duy thử nghiệm, sáng tạo và phát triển kỹ năng lập trình một cách hệ thống.

1.4.2. Công nghệ xử lý logic

Trong hệ thống Backend, ba công nghệ chính được lựa chọn nhằm đảm bảo tính ổn định, an toàn và khả năng mở rộng của kiến trúc. Trước hết, **Django và Django REST Framework (DRF)** đóng vai trò là nền tảng cốt lõi. Django, với triết lý “batteries included”, cung cấp sẵn nhiều thành phần cần thiết như ORM, hệ thống xác thực người dùng và bảng điều khiển quản trị, giúp giảm thiểu thời gian phát triển và đảm bảo tính nhất quán. Việc sử dụng Python cho Backend cũng tạo sự đồng bộ tuyệt đối với ngôn ngữ giảng dạy của hệ thống, giúp học sinh dễ dàng hình dung mối liên hệ giữa lý thuyết

và thực tiễn. DRF bổ sung khả năng xây dựng RESTful API chuẩn, đảm bảo giao tiếp giữa Frontend và Backend diễn ra độc lập, an toàn và có thể mở rộng [12].

Tiếp theo, **Celery và Redis** được tích hợp để xử lý các tác vụ bất đồng bộ, vốn thường tốn nhiều tài nguyên và cần được thực hiện ngoài luồng chính. Celery là distributed task queue cho phép phân phối và quản lý các tác vụ phức tạp, trong khi Redis đóng vai trò là in-memory data structure store, hỗ trợ tốc độ truy xuất cực nhanh. Sự kết hợp này giúp hệ thống thực hiện hiệu quả các chức năng như chạy mã nguồn trong sandbox, chấm điểm tự động, gửi email thông báo hay tạo báo cáo tiến độ học tập. Nhờ đó, trải nghiệm của người dùng không bị gián đoạn, đồng thời hệ thống vẫn duy trì hiệu năng ổn định ngay cả khi khối lượng tác vụ tăng cao.

Cuối cùng, **Docker** được lựa chọn như một nền tảng containerization nhằm đóng gói ứng dụng cùng toàn bộ môi trường phụ thuộc. Điều này đảm bảo tính nhất quán từ môi trường phát triển đến sản xuất, giúp giảm thiểu lỗi phát sinh do khác biệt cấu hình. Docker cũng hỗ trợ triển khai nhanh chóng, dễ dàng mở rộng quy mô hệ thống khi số lượng người dùng tăng, đồng thời tạo điều kiện thuận lợi cho việc bảo trì và nâng cấp. Nhờ sự kết hợp của Django/DRF, Celery/Redis và Docker, hệ thống Backend không chỉ đảm bảo tính ổn định và bảo mật mà còn có khả năng mở rộng linh hoạt, đáp ứng tốt yêu cầu của một nền tảng học tập trực tuyến dành cho trẻ em.

1.4.3. Cơ sở dữ liệu và hạ tầng

Trong hệ thống lưu trữ dữ liệu và quản lý phiên bản, hai công nghệ chủ đạo được lựa chọn là PostgreSQL và Git, mỗi công nghệ đảm nhận một vai trò khác nhau nhưng bổ trợ lẫn nhau để đảm bảo tính toàn vẹn và hiệu quả của hệ thống. PostgreSQL là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở nổi tiếng với khả năng tuân thủ ACID, hỗ trợ dữ liệu phức tạp và hiệu năng cao. Nhờ đó, hệ thống có thể lưu trữ và quản lý một cách an toàn các thông tin quan trọng như dữ liệu người dùng và phân quyền, cấu trúc bài học và bài tập, tiến độ học tập cùng lịch sử mã nguồn, cũng như kết quả đánh giá và phản hồi. Việc sử dụng PostgreSQL không chỉ đảm bảo tính nhất quán và độ tin cậy của dữ liệu mà còn tạo điều kiện thuận lợi cho việc phân tích, đánh giá và mở rộng quy mô hệ thống trong tương lai.

Song song với đó, Git được triển khai như một hệ thống quản lý phiên bản phân tán, đóng vai trò thiết yếu trong việc hỗ trợ làm việc nhóm, theo dõi thay đổi mã nguồn và quản lý các phiên bản phát triển. Nhờ Git, các nhóm phát triển có thể cộng tác hiệu

quả, dễ dàng quay lại các phiên bản trước khi cần, đồng thời kiểm soát toàn bộ lịch sử thay đổi của dự án. Điều này không chỉ giúp duy trì tính minh bạch trong quá trình phát triển mà còn giảm thiểu rủi ro khi triển khai các tính năng mới. Sự kết hợp giữa PostgreSQL và Git tạo nên một nền tảng vững chắc, vừa đảm bảo quản lý dữ liệu học tập tập trung, vừa hỗ trợ quy trình phát triển phần mềm linh hoạt và an toàn, đáp ứng đầy đủ yêu cầu của một hệ thống học tập trực tuyến hiện đại dành cho trẻ em.

1.5. Nguyên tắc thiết kế giao diện cho trẻ em

1.5.1. Lý thuyết về tải nhận thức (Cognitive Load Theory)

Theo lý thuyết tải nhận thức của Sweller (1988), bộ nhớ làm việc của trẻ em có giới hạn rất hạn chế [9], do đó việc thiết kế giao diện học tập cần chú trọng đến việc giảm thiểu các yếu tố gây phân tán và tối ưu hóa khả năng xử lý thông tin. Một trong những giải pháp quan trọng là phân khu không gian làm việc (*Workspace Compartmentalization*), tức chia màn hình thành các khu vực chức năng rõ ràng như khu vực soạn thảo mã, khu vực hiển thị kết quả đồ họa và khu vực hướng dẫn bài học. Cách bố trí này giúp loại bỏ hiệu ứng “chia sẻ sự chú ý” (*split-attention effect*), cho phép trẻ dễ dàng đối chiếu giữa dòng lệnh và hành động của đối tượng Turtle, từ đó tăng khả năng ghi nhớ và hiểu khái niệm lập trình.

Bên cạnh đó, việc trực quan hóa không gian toán học cũng đóng vai trò quan trọng trong việc giảm tải nhận thức. Bằng cách sử dụng lưới tọa độ mờ phía sau canvas vẽ, trẻ em có thể dễ dàng chuyển đổi tư duy từ không gian trừu tượng sang các đại lượng vật lý cụ thể như khoảng cách và góc độ. Điều này không chỉ giúp trẻ hình dung rõ hơn mối quan hệ giữa cú pháp lập trình và hình học không gian, mà còn tạo nền tảng cho việc kết nối kiến thức toán học với lập trình, góp phần phát triển tư duy logic và khả năng giải quyết vấn đề.

Như vậy, việc áp dụng các nguyên tắc của lý thuyết tải nhận thức vào thiết kế giao diện không chỉ giúp giảm thiểu gánh nặng cho bộ nhớ làm việc của trẻ mà còn tạo ra môi trường học tập trực quan, thân thiện và hiệu quả hơn. Đây là yếu tố then chốt để đảm bảo trẻ em có thể tiếp thu kiến thức lập trình một cách tự nhiên, hứng thú và bền vững.

1.5.2. Nguyên lý giàn giáo sư phạm (Instructional Scaffolding)

Dựa trên lý thuyết vùng phát triển tiếp cận (*Zone of Proximal Development*) của Vygotsky, thiết kế hệ thống học tập cần cung cấp các công cụ hỗ trợ trung gian để giúp

trẻ em vượt qua khoảng cách giữa năng lực hiện tại và tiềm năng phát triển. Một trong những công cụ quan trọng là Quick-Type Palette, nơi giao diện không bắt buộc trẻ phải gõ toàn bộ câu lệnh mà cung cấp các nút bấm đồ họa đại diện cho lệnh khối. Khi trẻ chọn một nút, hệ thống sẽ tự động chèn mã văn bản tương ứng vào trình soạn thảo. Đây là bước đệm hoàn hảo cho quá trình chuyển tiếp từ lập trình khối sang lập trình thuần túy, giúp trẻ vừa duy trì sự quen thuộc với thao tác trực quan vừa dần làm quen với cú pháp văn bản.

Song song với đó, hệ thống còn tích hợp công cụ gợi ý và tự động hoàn thành trong trình soạn thảo mã nguồn. Các tính năng như gợi ý cú pháp, tự động hoàn thành tên lệnh, phát hiện lỗi cú pháp theo thời gian thực không chỉ giảm thiểu lỗi đánh máy mà còn tăng tốc độ học tập. Điều này đặc biệt quan trọng trong giai đoạn trẻ mới tiếp cận lập trình, khi sai sót nhỏ có thể gây ra sự bối rối hoặc mất hứng thú. Nhờ có sự hỗ trợ này, trẻ em có thể tập trung vào việc phát triển tư duy logic và giải quyết vấn đề thay vì bị cản trở bởi những chi tiết kỹ thuật vụn vặt.

Như vậy, việc áp dụng lý thuyết của Vygotsky vào thiết kế giao diện không chỉ tạo ra môi trường học tập thân thiện mà còn đảm bảo trẻ em được hỗ trợ đúng lúc, đúng mức. Các công cụ trung gian như Quick-Type Palette và hệ thống gợi ý cú pháp đóng vai trò như “người hướng dẫn ảo”, giúp trẻ vượt qua giới hạn hiện tại, tiến gần hơn đến năng lực tiềm năng, đồng thời duy trì sự hứng thú và động lực trong quá trình học lập trình.

1.5.3. Thiết kế cảm xúc và tác nhân sư phạm

Trong thiết kế hệ thống học tập dành cho trẻ em, việc bổ sung tác nhân sư phạm (*Pedagogical Agent*) và thiết kế cảm xúc (*Emotional Design*) đóng vai trò quan trọng trong việc tạo ra trải nghiệm học tập thân thiện, hấp dẫn và hiệu quả. Tác nhân sư phạm được xây dựng dưới dạng nhân vật hoạt hình gần gũi, có khả năng hướng dẫn, động viên và cung cấp phản hồi trực tiếp cho trẻ. Thay vì hiển thị các thông báo lỗi khô khan, hệ thống chuyển đổi chúng thành hội thoại tự nhiên từ nhân vật ảo, áp dụng phương pháp Socratic nhằm khuyến khích trẻ đặt câu hỏi, suy nghĩ phản biện và tự tìm ra lời giải. Nhờ đó, quá trình học tập trở nên sinh động, giảm bớt cảm giác thất bại khi gặp lỗi và tăng cường sự gắn kết giữa trẻ với môi trường học tập.

Song song với đó, thiết kế cảm xúc được áp dụng để xây dựng giao diện mang tính thân thiện và an toàn. Việc sử dụng màu sắc ấm áp, hình dạng bo tròn và các hiệu ứng

animation nhẹ nhàng giúp tạo ra không gian học tập dễ chịu, giảm căng thẳng và khuyến khích trẻ tiếp tục thử nghiệm. Khi trẻ mắc lỗi, hệ thống không đưa ra phản hồi tiêu cực mà thay vào đó sử dụng thông điệp động viên, khuyến khích thử lại, qua đó nuôi dưỡng sự kiên trì và tinh thần học tập tích cực.

Sự kết hợp giữa tác nhân sư phạm và thiết kế cảm xúc không chỉ hỗ trợ về mặt kỹ thuật mà còn tác động mạnh mẽ đến tâm lý học tập của trẻ. Đây là minh chứng cho việc áp dụng các lý thuyết giáo dục hiện đại vào thiết kế công nghệ, nhằm tạo ra một môi trường học tập toàn diện, nơi trẻ em vừa được hướng dẫn, vừa được khích lệ, đồng thời phát triển cả kỹ năng lập trình lẫn sự tự tin và hứng thú trong quá trình học.

1.5.4. Nguyên tắc khả năng tiếp cận (Accessibility)

Trong thiết kế giao diện học tập dành cho trẻ em, việc chú trọng đến yếu tố trực quan và khả năng thao tác là vô cùng quan trọng để đảm bảo trải nghiệm thân thiện và hiệu quả. Một trong những nguyên tắc cơ bản là mục tiêu chạm lớn (*Large Touch Targets*), với kích thước nút bấm tối thiểu 44x44px và khoảng cách giữa các phần tử đủ rộng. Điều này đặc biệt cần thiết vì trẻ em thường chưa hoàn thiện kỹ năng vận động tinh, nên việc thao tác chính xác trên màn hình nhỏ có thể gặp khó khăn. Nhờ thiết kế này, trẻ có thể dễ dàng bấm chọn mà không bị nhầm lẫn, từ đó tăng sự tự tin và giảm cảm giác thất vọng khi thao tác sai.

Song song với đó, hệ thống áp dụng mã hóa màu sắc (*Color Coding*) để định danh các nhóm chức năng. Ví dụ, nhóm di chuyển được gắn với màu xanh, nhóm vẽ với màu đỏ, và nhóm điều khiển với màu xanh lá. Việc sử dụng bảng màu nhất quán không chỉ giúp trẻ dễ dàng phân biệt các loại lệnh mà còn hình thành phản xạ có điều kiện, hỗ trợ quá trình ghi nhớ và nhận diện nhanh chóng. Đây là một phương pháp trực quan hóa thông tin hiệu quả, giúp trẻ em giảm tải nhận thức và tập trung hơn vào việc phát triển tư duy logic thay vì phải ghi nhớ quá nhiều chi tiết.

Ngoài ra, hệ thống còn tích hợp hỗ trợ đa ngôn ngữ, trong đó tiếng Việt được sử dụng làm ngôn ngữ chính để phù hợp với đối tượng học sinh trong nước. Bên cạnh văn bản, giao diện còn đi kèm tài liệu hướng dẫn bằng hình ảnh và video minh họa, giúp trẻ dễ dàng tiếp cận nội dung ngay cả khi vốn từ vựng còn hạn chế. Điều này không chỉ tăng tính thân thiện mà còn mở rộng khả năng tiếp cận cho nhiều đối tượng khác nhau, đảm bảo rằng trẻ em ở nhiều cấp độ ngôn ngữ vẫn có thể tham gia học tập một cách hiệu quả.

TÓM TẮT CHƯƠNG 1

Chương 1 đã trình bày cơ sở lý thuyết và tổng quan về đề tài nghiên cứu, tạo nền tảng khoa học cho việc phát triển hệ thống học tập trực tuyến hỗ trợ trẻ em học lập trình với Turtle Python. Nội dung chương bao gồm phân tích bối cảnh giáo dục lập trình cho trẻ em trong xu thế STEM toàn cầu, giới thiệu các mô hình lập trình trực quan phổ biến như Logo, Scratch, Blockly và Code.org, đồng thời làm rõ đặc điểm tâm lý của lứa tuổi tiểu học và THCS. Tiếp đó, chương trình bày chi tiết về thư viện Turtle trong Python, từ lịch sử phát triển, nguyên lý hoạt động đến những ưu điểm và hạn chế khi sử dụng trong môi trường IDE truyền thống.

Phần tiếp theo tập trung vào kiến trúc hệ thống Web-based Client-Server, giải thích lý do lựa chọn, mô hình phân tầng, luồng dữ liệu và các ưu điểm nổi bật trong bối cảnh giáo dục trực tuyến. Song song, chương cũng giới thiệu các công nghệ nền tảng được sử dụng, bao gồm Frontend (React.js, Tailwind CSS, CodeMirror), Backend (Django, Celery/Redis), Database (PostgreSQL) và Infrastructure (Docker, Git), cho thấy sự phối hợp đồng bộ giữa các thành phần để đảm bảo hiệu năng và tính mở rộng.

Cuối cùng, chương làm rõ các nguyên tắc thiết kế giao diện dựa trên lý thuyết giáo dục hiện đại như Cognitive Load Theory và Instructional Scaffolding, kết hợp với thiết kế cảm xúc và khả năng tiếp cận cho trẻ em. Những nội dung này không chỉ cung cấp cơ sở lý luận mà còn định hướng cho việc phân tích yêu cầu, thiết kế kiến trúc và triển khai hệ thống trong các chương tiếp theo.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Dựa trên cơ sở lý thuyết và tổng quan đã trình bày ở Chương 1, chương này tập trung vào việc khảo sát thực tế, phân tích yêu cầu và thiết kế hệ thống hỗ trợ trẻ em học lập trình với thư viện Turtle của Python. Nội dung bao gồm: khảo sát hành vi và thói quen học tập của trẻ em trong độ tuổi 8–14, phân tích các yếu tố ảnh hưởng đến quá trình học, từ đó xác định yêu cầu chức năng và phi chức năng của hệ thống. Tiếp theo, chương trình bày kiến trúc tổng thể, các sơ đồ UML, thiết kế cơ sở dữ liệu và giao diện người dùng, nhằm đảm bảo hệ thống vừa đáp ứng mục tiêu giáo dục vừa phù hợp với đặc điểm tâm lý của trẻ.

2.1. Khảo sát hệ thống

2.1.1. Mục tiêu khảo sát

Khảo sát được thực hiện nhằm thu thập thông tin thực tế về hành vi, khả năng nhận thức và thói quen học tập của trẻ trong độ tuổi 8–14 (tiểu học và đầu trung học cơ sở), đồng thời xác định các yếu tố ảnh hưởng trực tiếp đến việc học. Mục tiêu chính là định hình hướng xây dựng hệ thống học lập trình với thư viện Turtle trong Python sao cho hấp dẫn và phù hợp với đặc điểm tâm lý của trẻ.

Cụ thể, khảo sát tập trung vào bốn khía cạnh:

- **Quan sát phản ứng của trẻ với các hình thức học khác nhau:** So sánh sự khác biệt giữa việc học thông qua hình ảnh, trò chơi và lời giảng truyền thống để nhận diện phương thức giúp trẻ tiếp thu hiệu quả nhất.
- **Phân tích tác động của yếu tố vui chơi và tương tác:** Đánh giá mức độ ảnh hưởng của các hoạt động có yếu tố trò chơi (kéo thả, chọn số, ghép hình) đến khả năng nhận biết và ghi nhớ.
- **Khảo sát vai trò của yếu tố phần thưởng trong học tập:** Nghiên cứu sự thay đổi về thái độ và động lực học khi trẻ được khích lệ bằng lời khen, huy hiệu hoặc điểm thưởng so với khi học không có phản hồi tích cực.
- **Nghiên cứu phối màu trong tương tác người máy:** Xem xét tác động của màu sắc đến tâm lý và khả năng tập trung, trong đó màu đỏ giúp tăng sự chú ý, màu vàng tạo cảm giác năng động và sáng tạo, màu cam tăng năng lượng và sự hứng thú. Việc lựa chọn màu sắc phù hợp nhằm tránh gây chói, mỏi mắt hay mất tập trung cho trẻ.

2.1.2. Đối tượng khảo sát

Khảo sát được tiến hành trên ba nhóm đối tượng chính có liên quan trực tiếp đến quá trình học lập trình và việc áp dụng công nghệ trong giáo dục tiểu học – THCS. Nhóm trọng tâm là trẻ em từ 8 đến 14 tuổi, đây là giai đoạn phát triển nhận thức quan trọng, khi trẻ bắt đầu hình thành tư duy logic và khả năng tiếp cận các khái niệm lập trình cơ bản. Trẻ ở độ tuổi này có khả năng quan sát tốt, ghi nhớ nhanh thông qua hình ảnh và thao tác thực hành, nhưng cũng dễ bị mất tập trung hoặc giảm hứng thú nếu bài học kéo dài hoặc thiếu yếu tố tương tác.

Ngoài ra, khảo sát còn hướng đến giáo viên – những người trực tiếp giảng dạy, có vai trò định hướng phương pháp tiếp cận phù hợp, và phụ huynh – những người theo dõi, hỗ trợ và đánh giá tiến trình học tập của trẻ. Việc kết hợp cả ba nhóm đối tượng giúp kết quả khảo sát phản ánh toàn diện hơn, vừa từ góc nhìn người học, vừa từ góc nhìn người hướng dẫn và người giám sát, qua đó làm cơ sở để thiết kế hệ thống học tập phù hợp, hiệu quả và có tính thực tiễn cao.

2.1.3. Phương pháp khảo sát

Để đảm bảo dữ liệu thu thập có độ tin cậy và giá trị thực tiễn, quá trình khảo sát được triển khai bằng ba phương pháp bổ trợ lẫn nhau:

- **Quan sát trực tiếp:** Nhóm nghiên cứu theo dõi hoạt động học tập của 15 trẻ tại lớp học và gia đình trong khoảng thời gian 2 tuần, tập trung vào thời gian chú ý, phản ứng khi tiếp nhận kiến thức, khả năng ghi nhớ và mức độ hứng thú. Phương pháp này cho phép ghi nhận hành vi tự nhiên của trẻ trong môi trường học tập thực tế. Kết quả sơ bộ cho thấy trẻ hào hứng hơn khi học qua hình ảnh, trò chơi hoặc có phần thưởng.
- **Phỏng vấn giáo viên và phụ huynh:** Tiến hành trao đổi ngắn với 5 giáo viên mầm non và 8 phụ huynh có con trong độ tuổi 8–14. Mục tiêu là thu thập nhận xét từ những người trực tiếp giảng dạy và giám sát quá trình học tập của trẻ, qua đó bổ sung góc nhìn thực tiễn về nhu cầu và kỳ vọng đối với một hệ thống học lập trình trực tuyến. Phản hồi cho thấy phần lớn trẻ thích học qua trò chơi, có phần thưởng hoặc lời khen, trong khi phụ huynh mong muốn có một nền tảng web dễ sử dụng và có khả năng theo dõi kết quả học tập.
- **Tham khảo tài liệu giáo dục:** Nhóm nghiên cứu đối chiếu kết quả quan sát và phỏng vấn với các nghiên cứu, tài liệu chuyên ngành về giáo dục lập trình cho

trẻ em. Việc này giúp củng cố cơ sở lý luận và đảm bảo tính nhất quán giữa dữ liệu thực tế và nền tảng khoa học.

Kết hợp ba phương pháp trên cho phép khảo sát vừa phản ánh hành vi học tập của trẻ, vừa ghi nhận quan điểm của giáo viên và phụ huynh, đồng thời đặt trong bối cảnh lý thuyết giáo dục hiện đại. Những dữ liệu thu thập được từ ba phương pháp trên đã cung cấp cái nhìn toàn diện về hành vi và nhu cầu học tập của trẻ, làm nền tảng cho việc tổng hợp và phân tích kết quả khảo sát ở mục tiếp theo.

2.1.4. Kết quả khảo sát

Sau khi tiến hành quan sát, tham khảo tài liệu giáo dục và trao đổi với giáo viên cùng phụ huynh, nhóm nghiên cứu đã tổng hợp được một số kết quả nổi bật phản ánh đặc điểm học tập của trẻ trong độ tuổi 8–14 khi tiếp cận lập trình.

Thứ nhất, về đặc điểm nhận thức và hành vi học tập, trẻ có khả năng ghi nhớ nhanh thông qua hình ảnh, âm thanh và thao tác thực hành, nhưng thời gian tập trung thường chỉ kéo dài từ 5–10 phút cho mỗi hoạt động. Trẻ tiếp thu tốt hơn khi được tham gia trực tiếp vào các hoạt động có yếu tố vui chơi, khám phá và phản hồi tích cực, thay vì chỉ nghe giảng hoặc lặp lại thao tác theo yêu cầu. Quan sát cũng cho thấy trẻ thích tự thao tác và thử nghiệm, chẳng hạn như kéo thả, vì những hình thức này giúp trẻ ghi nhớ câu lệnh dễ dàng hơn.

Thứ hai, về phản ứng đối với hình thức học truyền thống và học trực tuyến, kết quả khảo sát cho thấy trẻ dễ cảm thấy nhàm chán khi chỉ nghe giảng, do thiếu yếu tố hình ảnh động, màu sắc và âm thanh, đặc biệt trong các buổi học kéo dài. Ngược lại, với hình thức học trực tuyến có yếu tố trò chơi, trẻ thể hiện sự phấn khích, tập trung và phản hồi nhanh hơn. Trong các bài học có tương tác như ghép hình hoặc kéo thả, phần lớn trẻ hoàn thành nhiệm vụ nhanh và mong muốn tiếp tục tham gia. Giáo viên và phụ huynh cũng ghi nhận rằng khi trẻ được học qua giao diện sinh động, có hình ảnh ngộ nghĩnh hoặc âm thanh khen thưởng, mức độ chú ý và kiên nhẫn của trẻ tăng lên rõ rệt.

Cuối cùng, về ảnh hưởng của phần thưởng, khảo sát khẳng định đây là động lực học tập mạnh mẽ nhất. Khi được khen ngợi hoặc nhận phần thưởng nhỏ như huy hiệu, điểm sao hoặc lời khen từ nhân vật hoạt hình, trẻ biểu hiện cảm xúc tích cực và chủ động lặp lại bài học để đạt thêm phần thưởng. Ngược lại, ở nhóm trẻ không có phần thưởng hoặc phản hồi tích cực, mức độ hứng thú giảm rõ rệt chỉ sau 5–7 phút đầu tiên.

Những kết quả này cho thấy yếu tố trò chơi hóa, phản hồi tích cực và thiết kế giao diện sinh động đóng vai trò quan trọng trong việc duy trì động lực và nâng cao hiệu quả học tập của trẻ, đồng thời là cơ sở định hướng cho quá trình phân tích và thiết kế hệ thống ở các mục tiếp theo.

2.1.5. Đánh giá kết quả khảo sát

Qua quá trình khảo sát, có thể nhận thấy rằng phương pháp học truyền thống dựa trên lời giảng tuy giúp trẻ nắm được những khái niệm cơ bản về lập trình, nhưng chưa tạo được sự hứng thú lâu dài. Nhiều trẻ nhanh chóng cảm thấy nhàm chán, khó duy trì sự tập trung và dễ quên kiến thức sau một thời gian ngắn. Điều này cho thấy hình thức học thụ động không phù hợp với đặc điểm tâm lý của lứa tuổi 8–14.

Ngược lại, việc kết hợp yếu tố trò chơi trong quá trình học mang lại hiệu quả cao hơn rõ rệt. Khi bài học được thiết kế với hình ảnh sinh động, âm thanh vui nhộn hoặc hoạt động tương tác, trẻ thể hiện sự hứng thú, phản ứng nhanh và ghi nhớ tốt hơn. Học qua trò chơi giúp trẻ tiếp thu kiến thức một cách tự nhiên, không bị cảm giác ép buộc, đồng thời khuyến khích sự chủ động và sáng tạo.

Bên cạnh đó, phần thưởng được xác định là yếu tố có tác động mạnh mẽ đến động lực học tập. Khi được khen ngợi hoặc nhận phần thưởng nhỏ như huy hiệu, điểm sao hay lời khen từ nhân vật hoạt hình, trẻ trở nên tích cực và chủ động hơn trong việc hoàn thành bài học. Ngược lại, ở nhóm trẻ không nhận được phản hồi tích cực, mức độ hứng thú giảm rõ rệt chỉ sau vài phút đầu tiên.

Từ những phân tích trên, có thể kết luận rằng hệ thống học lập trình dành cho trẻ em cần được thiết kế theo hướng trò chơi hóa, có cơ chế phản hồi tích cực và phần thưởng, đồng thời tận dụng hình ảnh, âm thanh và màu sắc sinh động để duy trì sự chú ý. Đây sẽ là nền tảng quan trọng cho quá trình xây dựng hệ thống ở các phần tiếp theo.

2.2. Phân tích yêu cầu hệ thống

Phân tích yêu cầu hệ thống được tiến hành dựa trên kết quả khảo sát ở mục trước, nhằm xác định cụ thể phạm vi, mục tiêu, đặc điểm người dùng và các yếu tố kỹ thuật cần thiết cho việc xây dựng hệ thống dạy trẻ em lập trình với thư viện Turtle Python.

2.2.1. Phạm vi và mục tiêu

Hệ thống được xây dựng với phạm vi tập trung vào môi trường thực hành lập trình Python sử dụng thư viện Turtle trực tiếp trên trình duyệt. Đối tượng hướng đến là học sinh đang làm quen với lập trình. Người dùng có thể soạn thảo mã nguồn, chạy và quan

sát kết quả đồ họa ngay lập tức, đồng thời quản lý các bài tập cá nhân và tra cứu tài liệu lệnh hỗ trợ quá trình học tập.

Mục tiêu cụ thể của hệ thống là xây dựng một môi trường trực tuyến cung cấp công cụ thực hành lập trình trực quan, cho phép người học trải nghiệm mà không cần cài đặt phần mềm. Hệ thống được thiết kế nhằm hỗ trợ người học tiếp cận ngôn ngữ lập trình Python thông qua việc viết lệnh và quan sát phản hồi đồ họa tức thì, từ đó tạo ra sự gắn kết giữa lý thuyết và thực hành. Đồng thời, hệ thống còn tích hợp chức năng quản lý bài tập cá nhân và tra cứu tài liệu lệnh Turtle, góp phần nâng cao hiệu quả tự học, giúp người học chủ động hơn trong quá trình rèn luyện kỹ năng lập trình.

2.2.2. Đối tượng sử dụng hệ thống

Hệ thống được thiết kế hướng đến hai nhóm người dùng chính, với nhu cầu và vai trò khác nhau. Nhóm thứ nhất là người dùng (học sinh/người học), đây là đối tượng trung tâm sử dụng nền tảng để thực hành viết và chạy code Python Turtle. Họ cần một giao diện đơn giản, trực quan, thao tác nhất quán, trong đó các luồng chính như đăng nhập, mở workspace, viết code và lưu bài phải dễ thực hiện mà không đòi hỏi hướng dẫn phức tạp. Trải nghiệm học tập được thiết kế nhằm giúp người học tập trung vào việc thực hành và quan sát kết quả đồ họa tức thì, qua đó tiếp cận lập trình một cách tự nhiên và hiệu quả.

Nhóm thứ hai là quản trị viên (*administrator*), những người chịu trách nhiệm vận hành và duy trì hệ thống thông qua giao diện Django Admin. Quản trị viên có nhiệm vụ quản lý tài khoản người dùng như xem, khóa hoặc xóa, đồng thời kiểm soát trạng thái dữ liệu hệ thống để đảm bảo tính toàn vẹn và ổn định. Ngoài ra, họ còn theo dõi hoạt động hạ tầng backend bao gồm Celery, Redis và các tác vụ nền, nhằm duy trì sự liên tục và hiệu quả của hệ thống.

2.2.3. Mục tiêu thiết kế hệ thống

Mục tiêu thiết kế hệ thống được xác định dựa trên đặc điểm người dùng và kết quả khảo sát, nhằm đảm bảo hệ thống học lập trình với thư viện Turtle trong Python vừa phù hợp với trẻ em, vừa đáp ứng yêu cầu quản lý và mở rộng. Các mục tiêu cụ thể gồm:

- **Đảm bảo tính trực quan và phản hồi tức thì:** Người dùng có thể viết lệnh Python trong editor và quan sát kết quả đồ họa Turtle ngay trên trình duyệt, không cần cài đặt thêm môi trường ngoài.

- **Thiết kế giao diện đơn giản, nhất quán:** Giao diện tập trung vào các thao tác cốt lõi như đăng nhập, mở workspace, chạy code và lưu bài, giúp người học có thể sử dụng độc lập mà không mất nhiều thời gian làm quen.
- **Quản lý bài tập cá nhân:** Hệ thống cho phép người dùng tạo, lưu trữ, chỉnh sửa và xóa các bài tập của mình, hỗ trợ quá trình học tập liên tục theo tiến độ cá nhân.
- **Khả năng mở rộng và bảo trì:** Kiến trúc backend được tổ chức theo module (user, lesson) và tách thành các service (web, redis, worker, beat), tạo nền tảng để dễ dàng bổ sung tính năng trong tương lai và đảm bảo khả năng bảo trì lâu dài.

2.3. Yêu cầu chức năng và phi chức năng

2.3.1. Yêu cầu chức năng

Các chức năng của hệ thống được chia thành hai nhóm chính: chức năng dành cho người dùng (trẻ em) và chức năng dành cho quản trị viên.

F1. Nhóm chức năng dành cho người dùng

F1.1. Quản lý tài khoản: Đăng ký, đăng nhập và đăng xuất. Xem thông tin cá nhân, cập nhật hồ sơ (avatar, thông tin cơ bản) và đổi mật khẩu.

F1.2. Soạn thảo mã nguồn: Viết code Python Turtle trực tiếp trong soạn thảo tích hợp trên trình duyệt.

F1.3. Thực thi và Trực quan hóa: Chạy mã Python và quan sát kết quả khung nhìn trực tiếp trên trình duyệt.

F1.4. Quản lý bài tập / dự án cá nhân: Tạo mới, xem danh sách, chỉnh sửa và xóa các bài tập (lesson/project) của bản thân.

F1.5. Thư viện tra cứu (Wiki): Tra cứu danh mục các lệnh Turtle kèm ví dụ minh họa.

F1.6. Tương tác cùng Gia sư Rùa: Nhận hỗ trợ, gợi ý giải bài thông qua cửa sổ chat tích hợp AI.

F2. Nhóm chức năng dành cho quản trị viên

F2.1. Đăng nhập và quản trị hệ thống: Quản trị viên đăng nhập bằng tài khoản riêng và quản lý qua giao diện Django Admin tại /admin. Thông tin đăng nhập được mã hóa đảm bảo an toàn.

F2.2. Quản lý người dùng: Xem danh sách người dùng, tìm kiếm, khóa hoặc xóa tài khoản. Xem thông tin cơ bản như email và ngày tạo tài khoản.

F2.3. Quản lý dữ liệu hệ thống: Quản lý các model đã được đăng ký trong admin (đặc biệt là user và các dữ liệu liên quan); theo dõi và kiểm soát trạng thái tài khoản ở mức hệ thống.

2.3.2. Yêu cầu phi chức năng

Các yêu cầu phi chức năng của hệ thống được xác định nhằm đảm bảo trải nghiệm học tập vui vẻ, ổn định và an toàn cho trẻ nhỏ. Cụ thể:

NF1. Hiệu năng (*Performance*)

- Thời gian phản hồi của các API phổ biến phải dưới 2 giây.
- Trang chính tải lần đầu dưới 3 giây trong điều kiện mạng nội bộ bình thường.

NF2. Bảo mật (*Security*)

- Xác thực người dùng bằng JWT + refresh token; mật khẩu được mã hóa trước khi lưu trữ.
- Phân quyền rõ giữa user và admin; các API chưa xác thực bị chặn truy cập.
- Bảo vệ hệ thống khỏi các tấn công phổ biến thông qua cấu hình CORS và CSRF đúng chuẩn.

NF3. Khả dụng (*Availability*)

- Hệ thống hoạt động ổn định; các service tự động khởi động lại khi gặp lỗi thông qua Docker.
- Mục tiêu uptime đạt $\geq 99\%$ trong môi trường triển khai.

NF4. Toàn vẹn dữ liệu (*Data Integrity*)

- Dữ liệu người dùng và bài học không bị mất khi khởi động lại service.
- Migration cơ sở dữ liệu được kiểm soát theo phiên bản schema, tránh mất dữ liệu khi cập nhật hệ thống.

NF5. Khả năng mở rộng (*Scalability*)

- Backend tách biệt thành các service độc lập: web, redis, worker, beat.
- Hỗ trợ scale số lượng Celery worker khi khối lượng tác vụ nền tăng lên.

NF6. Bảo trì (*Maintainability*)

- Mã nguồn chia module rõ ràng: backend theo các app Django (user, lesson); frontend theo cấu trúc pages / components / contexts.
- Dễ dàng bổ sung chức năng mới không ảnh hưởng đến các phần hiện có.

NF7. Tương thích (*Compatibility*)

- Môi trường phát triển chạy trên Windows + Docker.
- Frontend hỗ trợ các trình duyệt hiện đại: Chrome, Edge, Firefox.

NF8. Quan sát & Vận hành (*Observability*)

- Hệ thống có log đầy đủ cho các service: web, worker, beat.
- Dễ dàng kiểm tra và chẩn đoán lỗi thông qua lệnh docker compose logs.

NF9. Khôi phục sự cố (*Disaster Recovery*)

- Cơ chế backup cơ sở dữ liệu định kỳ và quy trình restore đã được kiểm thử.

NF10. Trải nghiệm người dùng (*Usability*)

- Giao diện trực quan, dễ sử dụng phù hợp với đối tượng học sinh.
- Các thao tác chính (đăng nhập, mở workspace, chạy code, lưu bài) được thiết kế đơn giản và nhất quán.

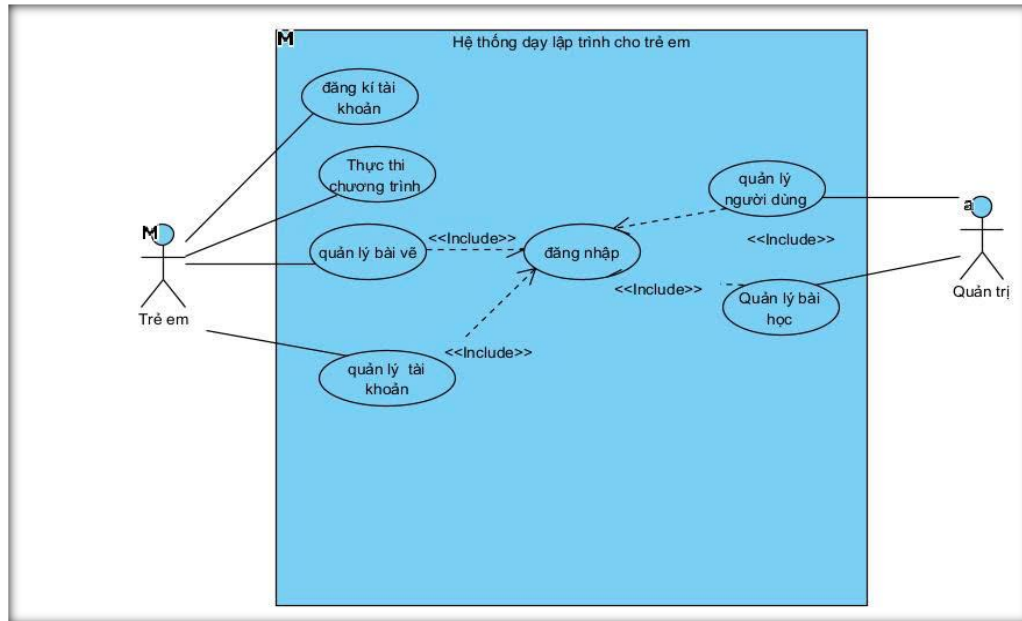
2.4. Sơ đồ Ca sử dụng (Use Case Diagram)

2.4.1. Xác định tác nhân (*Actor*)

Hệ thống bao gồm hai tác nhân chính tham gia tương tác:

- **Trẻ em (Người dùng):** Là tác nhân trung tâm của hệ thống, trực tiếp tham gia quá trình học lập trình thông qua nền tảng web. Các hành vi chính bao gồm: đăng ký, đăng nhập và đăng xuất tài khoản; cập nhật hồ sơ cá nhân (avatar, thông tin cơ bản, đổi mật khẩu); soạn thảo và thực thi mã Python Turtle với kết quả preview trực tiếp trên trình duyệt; quản lý bài tập cá nhân (tạo, xem, sửa, xóa); và tra cứu tài liệu lệnh tại trang Wiki. Mục tiêu của tác nhân này là học lập trình thông qua thực hành trực quan và phản hồi tức thì.
- **Quản trị viên (Admin):** Là tác nhân có quyền hạn cao nhất, chịu trách nhiệm vận hành và duy trì sự ổn định của hệ thống. Quản trị viên làm việc thông qua giao diện Django Admin tại /admin. Các hành vi chính bao gồm: quản lý tài khoản người dùng (xem danh sách, tìm kiếm, khóa hoặc xóa tài khoản); kiểm soát trạng thái dữ liệu hệ thống; và hỗ trợ vận hành hạ tầng backend (Celery, Redis, các tác vụ nền. Nhằm đảm bảo hệ thống hoạt động ổn định, an toàn và dữ liệu người dùng được kiểm soát đúng cách.

2.4.2. Use Case tổng quát



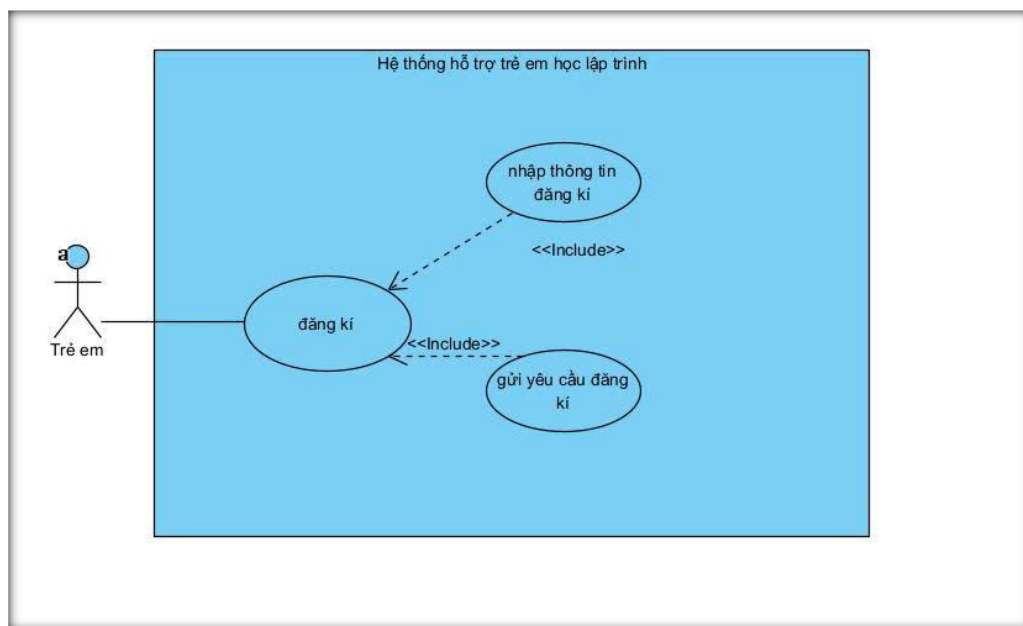
Hình 2.1 Sơ đồ Use Case tổng quát

Bảng 2.1. Bảng mô tả Use Case tổng quát

STT	Tên UC	Tác nhân	Mô tả
1	Đăng ký tài khoản	Trẻ em	Trẻ em tạo tài khoản bằng số điện thoại và mật khẩu chính. Sau khi đăng ký, tài khoản được kích hoạt cho phép trẻ em sử dụng hệ thống học lập trình
2	Đăng nhập	Trẻ em, Quản trị	Cho phép người dùng đăng nhập vào hệ thống bằng số điện thoại/mật khẩu Trẻ em: sử dụng thông tin đăng nhập của tài khoản. Quản trị viên: đăng nhập bằng tài khoản quản trị riêng có quyền cao nhất.
3	Thực thi chương trình	Trẻ em	Người dùng nhập lệnh turtle và chạy, nếu chạy đúng màn hình sẽ hiện thị con rùa vẽ hình theo đúng câu lệnh
4	Quản lý bài vẽ	Trẻ em	Cho phép người dùng lưu, tải, chia sẻ bài vẽ
5	Hướng dẫn học	Trẻ em	
6	Quản lý tài khoản cá nhân	Trẻ em, Quản trị	Cho phép người dùng quản lý và cập nhật thông tin tài khoản của mình.

STT	Tên UC	Tác nhân	Mô tả
			Quản trị viên: có thể thay đổi thông tin cá nhân của tài khoản quản trị, đổi mật khẩu.
7	Báo cáo thống kê	Quản trị	Cho phép xem thống kê về người dùng
8	Quản lý người dùng	Quản trị	Cho phép xem, tìm kiếm, khóa hoặc xóa tài khoản.
9	Quản lý bài học	Quản trị	Cho phép thêm, sửa, xóa các bài học, bổ sung câu lệnh nếu có

2.4.3. Use Case Đăng ký



Hình 2.2. Sơ đồ UC đăng ký

Kịch bản sử dụng các Sub_UC của đăng ký tài khoản

- Nhập thông tin đăng ký:

Bảng 2.2. Đặc tả ca sử dụng nhập thông tin đăng ký

Tên Use Case	Nhập thông tin đăng ký
Tác nhân	Trẻ em
Mô tả	Tác nhân nhập các thông tin cần thiết để tạo tài khoản mới, bao gồm họ tên, số điện thoại, mật khẩu và xác nhận mật khẩu.
Sự kiện kích hoạt	Tác nhân (chưa có tài khoản) nhấn vào nút "Đăng ký".
Tiền điều kiện	Tác nhân chưa có tài khoản trong hệ thống.

Tên Use Case	Nhập thông tin đăng ký
Đảm bảo tối thiểu	Nếu có lỗi (trường trống / định dạng sai / mật khẩu không khớp), giao diện hiển thị thông báo lỗi và trạng thái hệ thống không thay đổi.
Đảm bảo thành công	Toàn bộ thông tin được nhập đầy đủ và hợp lệ; giao diện hiển thị trạng thái "sẵn sàng gửi".
Chuỗi sự kiện chính	1) Tác nhân vào trang đăng ký 2) Hệ thống hiển thị form đăng ký (Họ và tên, Số điện thoại, Mật khẩu, Xác nhận mật khẩu) 3) Tác nhân nhập thông tin vào các trường 4) Hệ thống thực hiện kiểm tra cục bộ: trường bắt buộc, định dạng số điện thoại, mật khẩu và xác nhận mật khẩu có khớp không. 5) Nếu hợp lệ giao diện hiển thị trạng thái sẵn sàng gửi.
Luồng ngoại lệ	4.a) Một hoặc nhiều trường bị bỏ trống, số điện thoại sai định dạng, Xác nhận mật khẩu không khớp 5.a Nếu không hợp lệ thì thông báo lỗi đến tác nhân

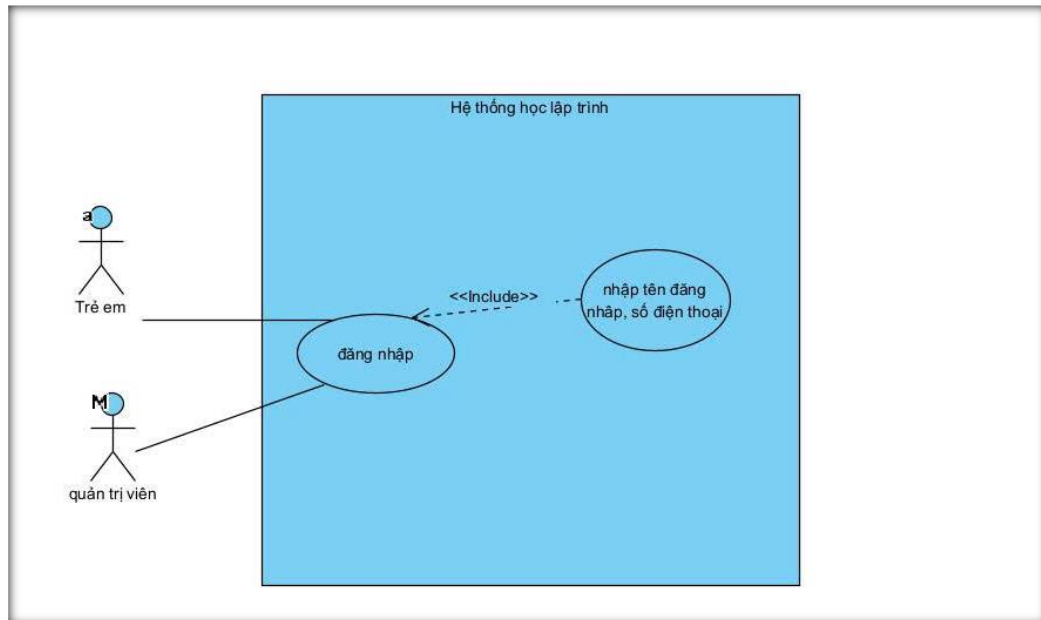
- Gửi yêu cầu đăng ký:

Bảng 2.3. Kịch bản ca sử dụng gửi yêu cầu đăng ký

Tên Use Case	Gửi yêu cầu đăng ký
Tác nhân	Trẻ em
Mô tả	Sau khi nhập đầy đủ thông tin, tác nhân gửi yêu cầu đăng ký đến hệ thống để tạo tài khoản mới.
Sự kiện kích hoạt	Tác nhân nhấn nút “Đăng ký tài khoản”.
Tiền điều kiện	Thông tin đăng ký đã được nhập đầy đủ và hợp lệ.
Đảm bảo tối thiểu	Hệ thống không lưu tài khoản nếu thông tin sai hoặc trùng lặp.
Đảm bảo thành công	Hệ thống mã hóa mật khẩu và lưu tài khoản mới vào CSDL, thông báo “Đăng ký thành công”.
Chuỗi sự kiện chính:	
1) Tác nhân nhấn “Đăng ký tài khoản”. 2) Hệ thống xác minh mật khẩu và xác nhận mật khẩu khớp nhau. 3) Hệ thống mã hóa mật khẩu và lưu thông tin tài khoản mới.	

Tên Use Case	Gửi yêu cầu đăng ký
4) Hệ thống hiển thị thông báo “Đăng ký thành công”.	
Ngoại lệ: 2.a Xác nhận mật khẩu không khớp (hoặc bỏ trống).	

2.4.4. Use Case Đăng nhập



Hình 2.3. Sơ đồ UC đăng nhập tài khoản

Kịch bản sử dụng UC của đăng nhập.

Bảng 2.4. Kịch bản ca sử dụng chia sẻ bài vẽ

Thuộc tính	Mô tả
Tên Use Case	Đăng nhập
Tác nhân chính	Trẻ em, Quản trị viên
Mức	Mức 3 – Use Case nghiệp vụ chi tiết
Tiền điều kiện	Người dùng đã có tài khoản hợp lệ; Trang đăng nhập đã sẵn sàng
Đảm bảo tối thiểu	Nếu đăng nhập thất bại (sai tài khoản hoặc mật khẩu), hệ thống hiển thị thông báo lỗi
Đảm bảo thành công	Người dùng được xác thực thành công và được chuyển đến trang chủ phù hợp (trang quản trị hoặc trang nhân viên)
Kích hoạt	Người dùng chọn chức năng “Đăng nhập” trên trang web
Chuỗi sự kiện chính	1. Người dùng chọn chức năng đăng nhập 2. Hệ thống hiển thị form đăng nhập 3. Người dùng nhập tên đăng nhập và mật khẩu

Thuộc tính	Mô tả
	4. Hệ thống kiểm tra tính hợp lệ dữ liệu (không để trống, đúng định dạng) 5. Hệ thống gửi thông tin đăng nhập đến CSDL để xác nhận 6. CSDL trả kết quả xác thực 7. Nếu thông tin đúng: • Tạo phiên đăng nhập • Chuyển đến trang chủ phù hợp Hiển thị thông báo “Đăng nhập thành công” 8. Người dùng bắt đầu sử dụng hệ thống
Ngoại lệ	4.a. Thông tin không hợp lệ (bỏ trống, sai định dạng) 4.a.1. Hệ thống hiển thị thông báo lỗi 4.a.2. Người dùng nhập lại thông tin 6.a. Sai tài khoản hoặc mật khẩu 6.a.1. Hệ thống thông báo “Tên tài khoản hoặc mật khẩu không đúng” 6.a.2. Người dùng nhập lại thông tin

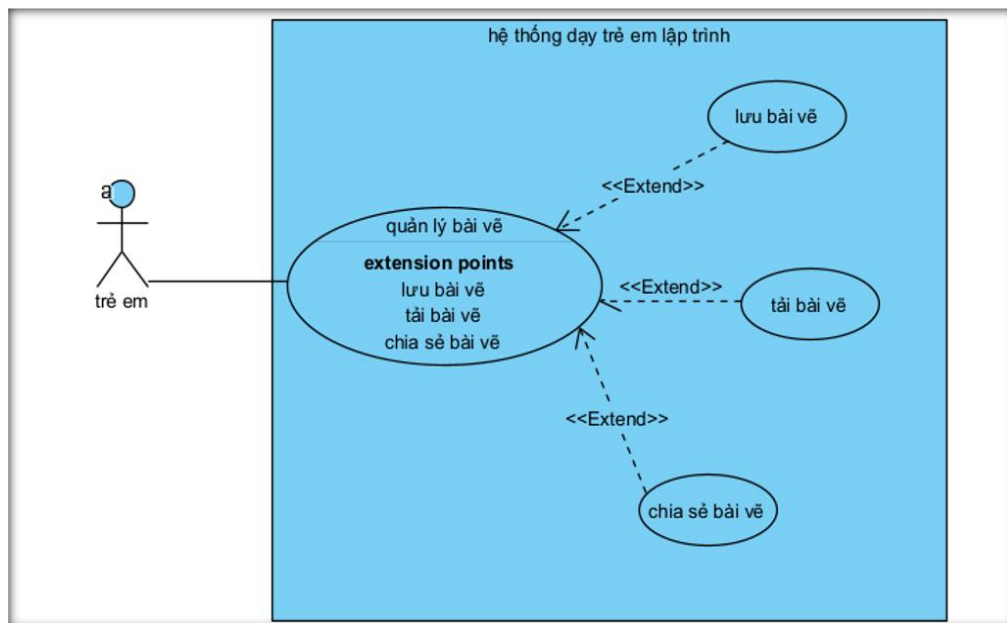
Kịch bản ca sử dụng UC “Quên mật khẩu”:

Bảng 2.5. Đặc tả ca sử dụng quên mật khẩu

Tên Use Case	Quên mật khẩu
Tác nhân	Trẻ em, Quản trị viên
Mô tả	Cho phép tác nhân lấy lại tài khoản và mật khẩu đăng nhập
Sự kiện kích hoạt	Tác nhân nhấn vào liên kết "Quên mật khẩu"
Tiền điều kiện	Tác nhân có một tài khoản (với số điện thoại hợp lệ) nhưng không nhớ mật khẩu
Đảm bảo tối thiểu	Hệ thống không thay đổi mật khẩu.
Đảm bảo thành công	Mật khẩu mới của Tác nhân được cập nhật trong CSDL.
Chuỗi sự kiện chính	1) Tác nhân vào trang quên mật khẩu 2) Hệ thống hiển thị form yêu cầu nhập số điện thoại đã đăng ký 3) Tác nhân nhập số điện thoại và nhấn "Gửi yêu cầu" 4) Hệ thống kiểm tra Số điện thoại tồn tại trong CSDL

Tên Use Case	Quên mật khẩu
	5) Hệ thống xác thực và hiển thị form "Đặt lại mật khẩu" 6) Tác nhân nhập mật khẩu mới (khớp nhau) và nhấn "Lưu" 7) Hệ thống cập nhật mật khẩu mới 8) Hệ thống báo thành công và chuyển về trang Đăng nhập.
Ngoại lệ	4.a Số điện thoại không tồn tại 4.a.1 hệ thống thông báo” số điện thoại không tồn tại” 4.a.2 hệ thống yêu cầu tác nhân nhập lại

2.4.5. Use Case Quản lý bài vẽ



Hình 2.4. Sơ đồ UC quản lý bài vẽ

Đặc tả sử dụng các Sub_UC của Quản lý bài vẽ.

- Lưu bài vẽ:

Bảng 2.6. Đặc tả ca sử dụng lưu bài vẽ

Tên Use Case	Lưu bài vẽ
Tác nhân	Trẻ em
Mô tả	Cho phép tác nhân lưu lại bài vẽ đã vẽ trước đó
Sự kiện kích hoạt	Tác nhân nhấn vào “Lưu bài vẽ”
Tiền điều kiện	Tác nhân có 1 bài vẽ muốn lưu
Đảm bảo tối thiểu	Hệ thống lưu bài vẽ

Tên Use Case	Lưu bài vẽ
Đảm bảo thành công	Bài vẽ được lưu thành công
Chuỗi sự kiện chính	1) Người dùng nhấn nút "Lưu bài vẽ". 2) Hệ thống hiển thị form nhập thông tin bài vẽ (tên bài vẽ, mô tả – tùy chọn). 3) Người dùng nhập thông tin và xác nhận lưu. 4) Hệ thống thu thập dữ liệu: • Thông tin người dùng • Danh sách lệnh Turtle • Trạng thái (màu, hướng) 5) Hệ thống lưu dữ liệu vào cơ sở dữ liệu. 6) Hệ thống thông báo Lưu bài vẽ thành công.
Ngoại lệ	AF1: Người dùng chưa đăng nhập Hệ thống yêu cầu người dùng đăng nhập trước khi lưu. 5.a: Lỗi lưu dữ liệu 5.a.1 Hệ thống hiển thị thông báo lỗi và yêu cầu thử lại.

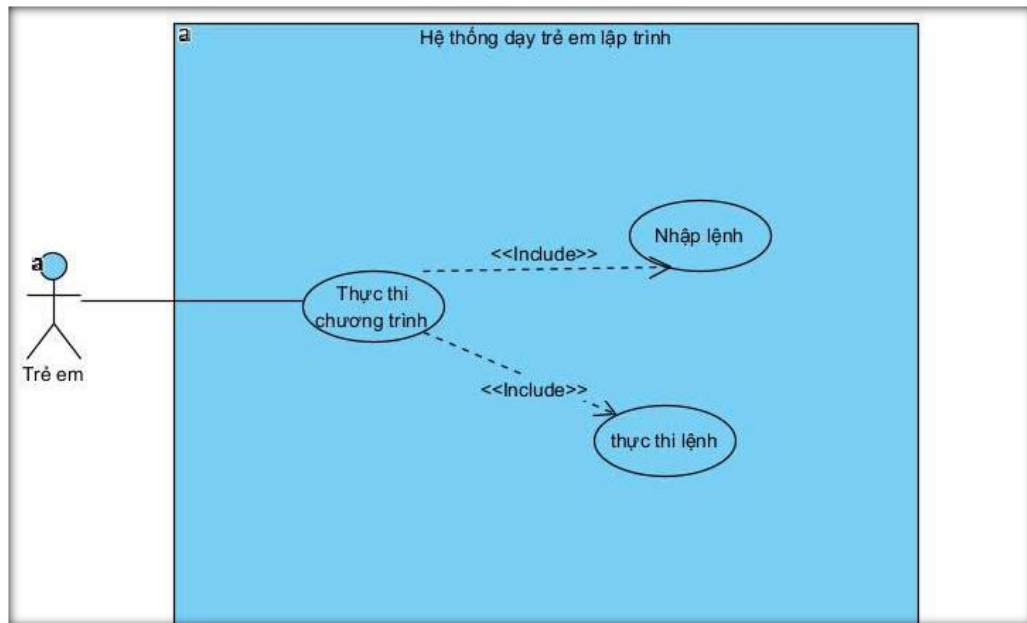
- Tải bài vẽ:

Bảng 2.7. Đặc tả ca sử dụng tải bài vẽ

Tên Use Case	Tải bài vẽ
Tác nhân	Trẻ em
Mô tả	Cho phép tác nhân tải lại bài vẽ đã vẽ trước đó
Sự kiện kích hoạt	Tác nhân nhấn vào “Lưu bài vẽ”
Tiền điều kiện	Tác nhân có 1 bài vẽ
Đảm bảo tối thiểu	Hệ thống tải bài vẽ
Đảm bảo thành công	file bài vẽ được tải thành công về thiết bị người dùng. Dữ liệu tải về đúng với hình vẽ hoặc bài vẽ đã chọn.
Chuỗi sự kiện chính	1) Người dùng đăng nhập vào hệ thống. 2) Người dùng truy cập danh sách Bài vẽ đã lưu. 3) Người dùng chọn một bài vẽ cần tải. 4) Người dùng nhấn nút "Tải bài vẽ". 5) Hệ thống hiển thị tùy chọn định dạng tải. 6) Hệ thống tạo file tương ứng và gửi về trình duyệt.

Tên Use Case	Tải bài vẽ
	7) Người dùng tải file thành công.
Ngoại lệ	1.a: Người dùng chưa đăng nhập khi tải bài vẽ đã lưu 3.a: Bài vẽ đã lưu không tồn tại Hệ thống thông báo lỗi và quay lại danh sách bài vẽ.

2.4.6. Use Case Thực thi chương trình



Hình 2.5. Sơ đồ UC thực thi chương trình

Kịch bản sử dụng các Sub_UC của thực thi chương trình.

- Nhập lệnh:

Bảng 2.8. Kịch bản ca sử dụng nhập lệnh

Tên Use Case	Nhập lệnh
Tác nhân	Trẻ em
Mô tả	Use Case này cho phép trẻ em nhập các lệnh Turtle vào ô nhập lệnh để chuẩn bị cho quá trình vẽ hình. Các lệnh sau khi nhập sẽ được gửi sang chức năng kiểm tra cú pháp và thực thi.
Sự kiện kích hoạt	Tác nhân nhấn ô nhập lệnh
Tiền điều kiện	Người dùng đang ở giao diện vẽ Turtle
Đảm bảo tối thiểu	Hệ thống ghi nhận dữ liệu lệnh do người dùng nhập.
Đảm bảo thành công	Các lệnh Turtle được nhập đầy đủ và sẵn sàng để kiểm tra cú pháp và thực thi.
Chuỗi sự kiện chính	1) Người dùng nhập các lệnh Turtle vào ô nhập lệnh.

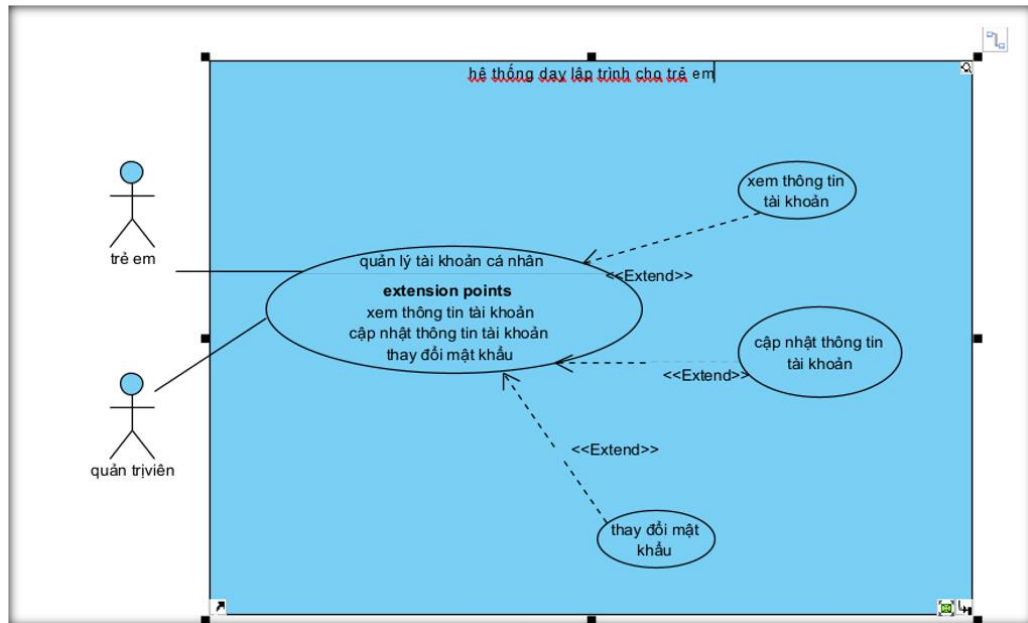
Tên Use Case	Nhập lệnh
	2) Người dùng nhập lệnh. 3) Người dùng nhấn nút Thực thi.
Ngoại lệ	2.a Người dùng để trống ô nhập lệnh, hệ thống yêu cầu nhập lệnh.

- Thực thi lệnh

Bảng 2.9. Kịch bản ca sử dụng thực thi lệnh

Tên Use Case	Thực thi lệnh vẽ turtle
Tác nhân	Hệ thống
Mô tả	Use Case này cho phép hệ thống thực thi tuần tự các lệnh Turtle hợp lệ để tạo hình vẽ tương ứng trên vùng vẽ
Sự kiện kích hoạt	Hệ thống nhận được kết quả kiểm tra lệnh hợp lệ từ Use Case Kiểm tra lệnh Turtle
Tiền điều kiện	Các lệnh Turtle đã được kiểm tra và hợp lệ.
Đảm bảo tối thiểu	Hệ thống thực thi lệnh mà không gây lỗi hoặc gián đoạn hệ thống.
Đảm bảo thành công	Hình vẽ được hiển thị chính xác trên màn hình theo các lệnh Turtle đã nhập.
Chuỗi sự kiện chính	1) Hệ thống khởi tạo trạng thái ban đầu của Turtle (vị trí, hướng, trạng thái bút). 2) Hệ thống thực thi từng lệnh Turtle theo đúng thứ tự. 3) Turtle di chuyển và vẽ tương ứng với mỗi lệnh. 4) Vùng vẽ được cập nhật theo thời gian thực. 5) Hệ thống hoàn tất quá trình vẽ.
Ngoại lệ	2.a Xảy ra lỗi trong quá trình thực thi. Hệ thống dừng thực thi và hiển thị thông báo lỗi

2.4.7. Use Case Quản lý tài khoản cá nhân



Hình 2.6. Sơ đồ UC quản lý tài khoản cá nhân

Kịch bản sử dụng các Sub_UC của quản lý tài khoản cá nhân.

- Xem thông tin tài khoản

Bảng 2.10. Kịch bản ca sử dụng xem thông tin tài khoản

Tên Use Case	Xem thông tin tài khoản
Tác nhân	Trẻ em, Quản trị viên
Mô tả	Cho phép tác nhân xem các thông tin cá nhân của mình như họ tên, số điện thoại, ngày tạo tài khoản.
Sự kiện kích hoạt	Tác nhân truy cập mục “Tài khoản của tôi” trong hệ thống.
Tiền điều kiện	Tác nhân đã đăng nhập vào hệ thống.
Đảm bảo tối thiểu	Nếu không tải được dữ liệu, hệ thống hiển thị thông báo lỗi.
Đảm bảo thành công	Toàn bộ thông tin cá nhân của tác nhân được hiển thị đầy đủ.
Chuỗi sự kiện chính	1) Actor chọn chức năng Quản lý tài khoản cá nhân. 2) Hệ thống xác thực trạng thái đăng nhập. 3) Hệ thống truy xuất thông tin tài khoản của actor. 4) Hệ thống hiển thị thông tin tài khoản (tên hiển thị, ảnh đại diện, ...).
Luồng ngoại lệ	2.a Người dùng chưa đăng nhập. Hệ thống yêu cầu đăng nhập

- Cập nhật thông tin tài khoản:

Bảng 2.11. Kịch bản ca sử dụng cập nhật thông tin tài khoản

Tên Use Case	Cập nhật thông tin cá nhân
Tác nhân	Phụ huynh, Quản trị viên
Mô tả	Cho phép tác nhân chỉnh sửa thông tin như họ tên, ngày sinh hoặc giới tính, đổi mật khẩu.
Sự kiện kích hoạt	Tác nhân chọn “Chỉnh sửa thông tin cá nhân”.
Tiền điều kiện	Tác nhân đã mua hoặc sở hữu ít nhất một vật phẩm.
Đảm bảo tối thiểu	Hệ thống không thay đổi dữ liệu nếu có lỗi khi lưu.
Đảm bảo thành công	Thông tin mới được lưu vào cơ sở dữ liệu và hiển thị lại.
Chuỗi sự kiện chính	1) trẻ em chọn chức năng Quản lý tài khoản cá nhân. 2) Hệ thống hiển thị thông tin tài khoản hiện tại. 3) Trẻ em chọn chức năng Thay đổi thông tin cá nhân. 4) Hệ thống hiển thị form chỉnh sửa thông tin. 5) Trẻ em nhập thông tin mới. 6) Trẻ em nhấn nút Lưu thay đổi. 7) Hệ thống kiểm tra tính hợp lệ của thông tin. 8) Hệ thống cập nhật thông tin vào cơ sở dữ liệu. 9) Hệ thống thông báo cập nhật thành công.
Luồng ngoại lệ	7.a Thông tin không hợp lệ. Hệ thống hiển thị thông báo lỗi. 8.a Lỗi cập nhật dữ liệu. Hệ thống yêu cầu thử lại.

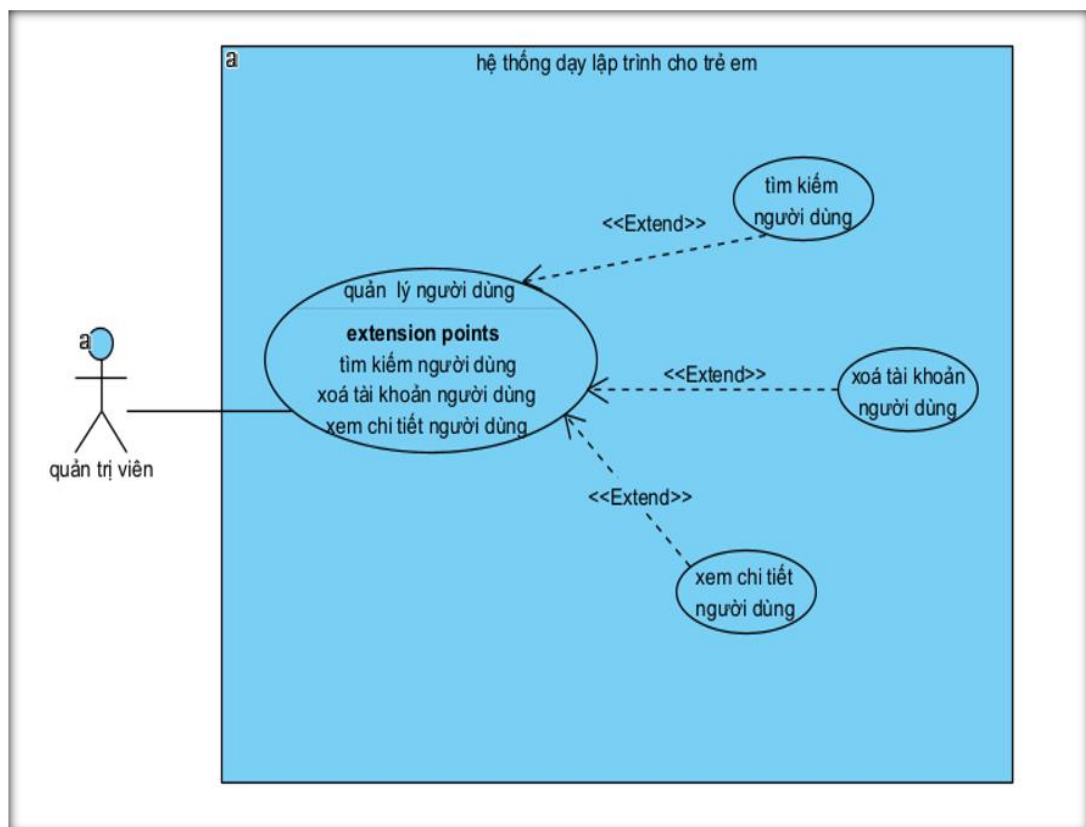
- Thay đổi mật khẩu:

Bảng 2.12. Kịch bản ca sử dụng thay đổi mật khẩu

Tên Use Case	Thay đổi mật khẩu
Tác nhân	Trẻ em, Quản trị viên
Mô tả	Use Case này cho phép tác nhân thay đổi mật khẩu đăng nhập để đảm bảo an toàn cho tài khoản cá nhân.
Sự kiện kích hoạt	Tác nhân chọn chức năng Thay đổi mật khẩu trong mục Quản lý tài khoản.
Tiền điều kiện	Tác nhân đã đăng nhập vào hệ thống.
Đảm bảo tối thiểu	Mật khẩu cũ vẫn được giữ nguyên nếu thay đổi thất bại.
Đảm bảo thành công	Mật khẩu mới được cập nhật và có hiệu lực.

Tên Use Case	Thay đổi mật khẩu
Chuỗi sự kiện chính	1) Trẻ em chọn chức năng Quản lý tài khoản cá nhân. 2) Trẻ em chọn chức năng Thay đổi mật khẩu. 3) Hệ thống hiển thị form thay đổi mật khẩu. 4) Trẻ em nhập mật khẩu hiện tại. 5) Trẻ em nhập mật khẩu mới và xác nhận mật khẩu. 6) Trẻ em nhấn nút Xác nhận. 7) Hệ thống kiểm tra mật khẩu hiện tại. 8) Hệ thống kiểm tra tính hợp lệ của mật khẩu mới. 9) Hệ thống cập nhật mật khẩu mới. 10) Hệ thống thông báo thay đổi mật khẩu thành công.
Luồng ngoại lệ	7.a Mật khẩu hiện tại không đúng .Hệ thống hiển thị lỗi. 8.a Mật khẩu mới không hợp lệ .Hệ thống yêu cầu nhập lại.

2.4.8. Use Case Quản lý người dùng



Hình 2.7. Sơ đồ UC quản lý người dùng

Kịch bản sử dụng các Sub_UC của quản lý người dùng

- Tìm kiếm người dùng

Bảng 2.13. Kịch bản ca sử dụng xem danh sách người dùng

Tên Use Case	Tìm kiếm người dùng
Tác nhân	Quản trị viên
Mô tả	Cho phép quản trị viên tìm kiếm người dùng theo các tiêu chí như tên, email hoặc trạng thái tài khoản.
Sự kiện kích hoạt	Tác nhân truy cập phần “Quản lý người dùng”.
Tiền điều kiện	Tác nhân đã đăng nhập vào hệ thống và có quyền truy cập trang quản trị.
Đảm bảo tối thiểu	Nếu không tìm thấy kết quả, hiển thị thông báo “Không có kết quả phù hợp”.
Đảm bảo thành công	Hệ thống hiển thị danh sách người dùng phù hợp với điều kiện tìm kiếm.
Chuỗi sự kiện chính	1) Quản trị viên nhập tiêu chí tìm kiếm. 2) Quản trị viên nhấn nút Tìm kiếm. 3) Hệ thống lọc dữ liệu người dùng theo tiêu chí. 4) Hệ thống hiển thị kết quả tìm kiếm.
Ngoại lệ	3.a Không có kết quả phù hợp. Hệ thống thông báo không tìm thấy người dùng.

- Xem chi tiết người dùng:

Bảng 2.14. Kịch bản ca sử dụng xem chi tiết người dùng

Tên Use Case	Xem chi tiết người dùng
Tác nhân	Quản trị viên
Mô tả	Cho phép quản trị viên xem thông tin chi tiết của một tài khoản người dùng.
Sự kiện kích hoạt	Tác nhân truy cập phần “Quản lý người dùng”.
Tiền điều kiện	Tác nhân đã đăng nhập vào hệ thống và có quyền truy cập trang quản trị.
Đảm bảo tối thiểu	Nếu không tìm thấy kết quả, hiển thị thông báo “Không có kết quả phù hợp”.
Đảm bảo thành công	Hệ thống hiển thị danh sách thông tin đầy đủ của người dùng gồm tên số điện thoại
Chuỗi sự kiện chính	1) Quản trị viên chọn một người dùng trong danh sách.

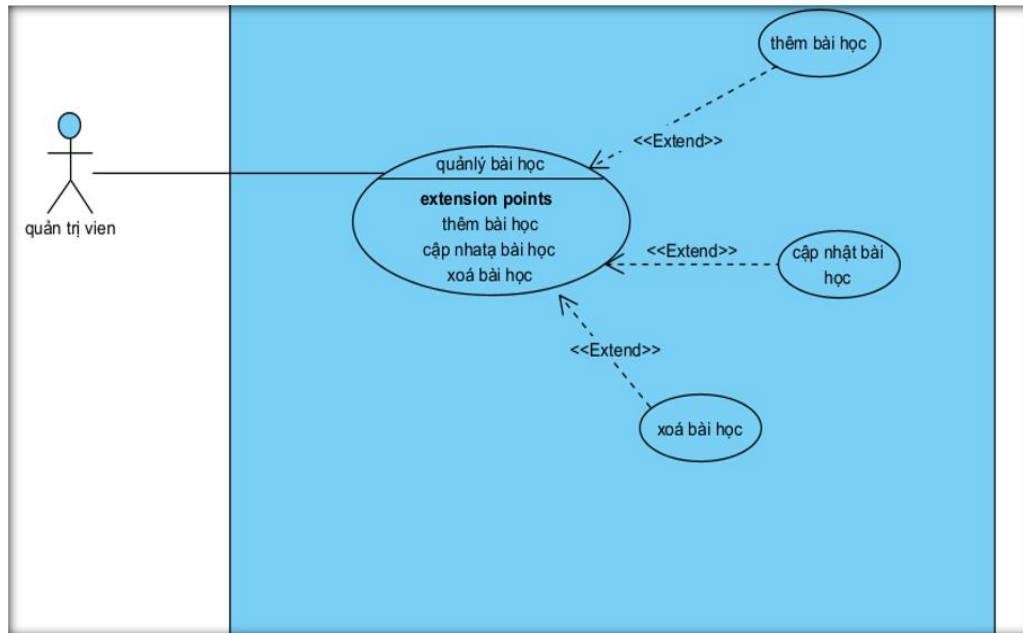
Tên Use Case	Xem chi tiết người dùng
	2) Quản trị viên chọn chức năng Xem chi tiết tài khoản. 3) Hệ thống truy xuất thông tin chi tiết người dùng. 4) Hệ thống hiển thị thông tin chi tiết.
Ngoại lệ	3.a Không truy xuất được dữ liệu. Hệ thống hiển thị thông báo lỗi

- Xóa tài khoản người dùng:

Bảng 2.15. Kịch bản ca sử dụng xóa tài khoản người dùng

Tên Use Case	Xóa tài khoản người dùng
Tác nhân	Quản trị viên
Mô tả	Cho phép tác nhân xóa hoàn toàn tài khoản người dùng ra khỏi hệ thống.
Sự kiện kích hoạt	Tác nhân chọn người dùng và nhấn “Xóa tài khoản”.
Tiền điều kiện	Tác nhân đã đăng nhập vào hệ thống và có quyền truy cập trang quản trị, tài khoản người dùng tồn tại và không đang bị xóa.
Đảm bảo tối thiểu	Nếu lỗi xảy ra, hệ thống không thay đổi dữ liệu.
Đảm bảo thành công	Tài khoản người dùng bị xóa khỏi CSDL, không thể đăng nhập lại.
Chuỗi sự kiện chính	1) Quản trị viên chọn người dùng cần xóa. 2) Quản trị viên chọn chức năng Xóa tài khoản. 3) Hệ thống yêu cầu xác nhận xóa. 4) Quản trị viên xác nhận. 5) Hệ thống xóa tài khoản người dùng. 6) Hệ thống thông báo xóa thành công
Luồng ngoại lệ	4.a Quản trị viên hủy thao tác. Hệ thống không xóa dữ liệu.

2.4.9. Use Case Quản lý bài học



Hình 2.8. Sơ đồ UC quản lý bài học

Kịch bản sử dụng các Sub_UC của quản lý bài học

- Thêm bài học

Bảng 2.16. Kịch bản ca sử dụng thêm bài học

Tên Use Case	Thêm bài học
Tác nhân	Quản trị viên
Mô tả	Cho phép tác nhân thêm mới một bài học vào hệ thống nhằm phục vụ cho quá trình giảng dạy và học tập.
Sự kiện kích hoạt	Tác nhân nhấn nút Thêm bài học trong chức năng Quản lý bài học.
Tiền điều kiện	Tác nhân đã đăng nhập hệ thống Tác nhân có quyền quản lý bài học
Đảm bảo tối thiểu	Không thêm bài học nếu thông tin không hợp lệ
Đảm bảo thành công	Bài học mới được lưu thành công vào hệ thống
Chuỗi sự kiện chính	1) Tác nhân truy cập chức năng Quản lý bài học. 2) Hệ thống hiển thị danh sách các bài học hiện có. 3) Tác nhân nhấn nút Thêm bài học. 4) Hệ thống hiển thị form nhập thông tin bài học. 5) Tác nhân nhập các thông tin bài học (tên bài học, mô tả, nội dung bài học, ví dụ minh họa...).

Tên Use Case	Thêm bài học
	6) Tác nhân nhấn nút Lưu. 7) Hệ thống kiểm tra tính hợp lệ của dữ liệu đã nhập. 8) Hệ thống lưu bài học mới vào cơ sở dữ liệu. 9) Hệ thống thông báo thêm bài học thành công và cập nhật danh sách bài học
Ngoại lệ	7.a Dữ liệu không hợp lệ Hệ thống hiển thị thông báo các trường không được bỏ trống.

- Cập nhật bài học

Bảng 2.17. Kịch bản ca sử dụng cập nhật bài học

Tên Use Case	Cập nhật bài học
Tác nhân	Quản trị viên
Mô tả	Cho phép tác nhân chỉnh sửa và cập nhật thông tin của bài học đã tồn tại trong hệ thống.
Sự kiện kích hoạt	Chọn chức năng Cập nhật bài học tại một bài học cụ thể.
Tiền điều kiện	Bài học cần cập nhật đã tồn tại trong hệ thống Tác nhân có quyền chỉnh sửa bài học
Đảm bảo tối thiểu	Không thay đổi dữ liệu nếu cập nhật không hợp lệ
Đảm bảo thành công	Bài học mới được chỉnh sửa thành công vào hệ thống
Chuỗi sự kiện chính	1) Tác nhân truy cập chức năng Quản lý bài học. 2) Hệ thống hiển thị danh sách các bài học. 3) Tác nhân chọn một bài học cần cập nhật. 4) Tác nhân nhấn nút Cập nhật bài học. 5) Hệ thống hiển thị thông tin hiện tại của bài học. 6) Tác nhân chỉnh sửa nội dung bài học. 7) Tác nhân nhấn nút Lưu. 8) Hệ thống kiểm tra tính hợp lệ của dữ liệu cập nhật. 9) Hệ thống cập nhật thông tin bài học trong cơ sở dữ liệu. 10) Hệ thống thông báo cập nhật thành công.
Ngoại lệ	6.a Dữ liệu chỉnh sửa không hợp lệ. Hệ thống yêu cầu chỉnh sửa lại. 9.a Bài học không tồn tại. Hệ thống thông báo lỗi.

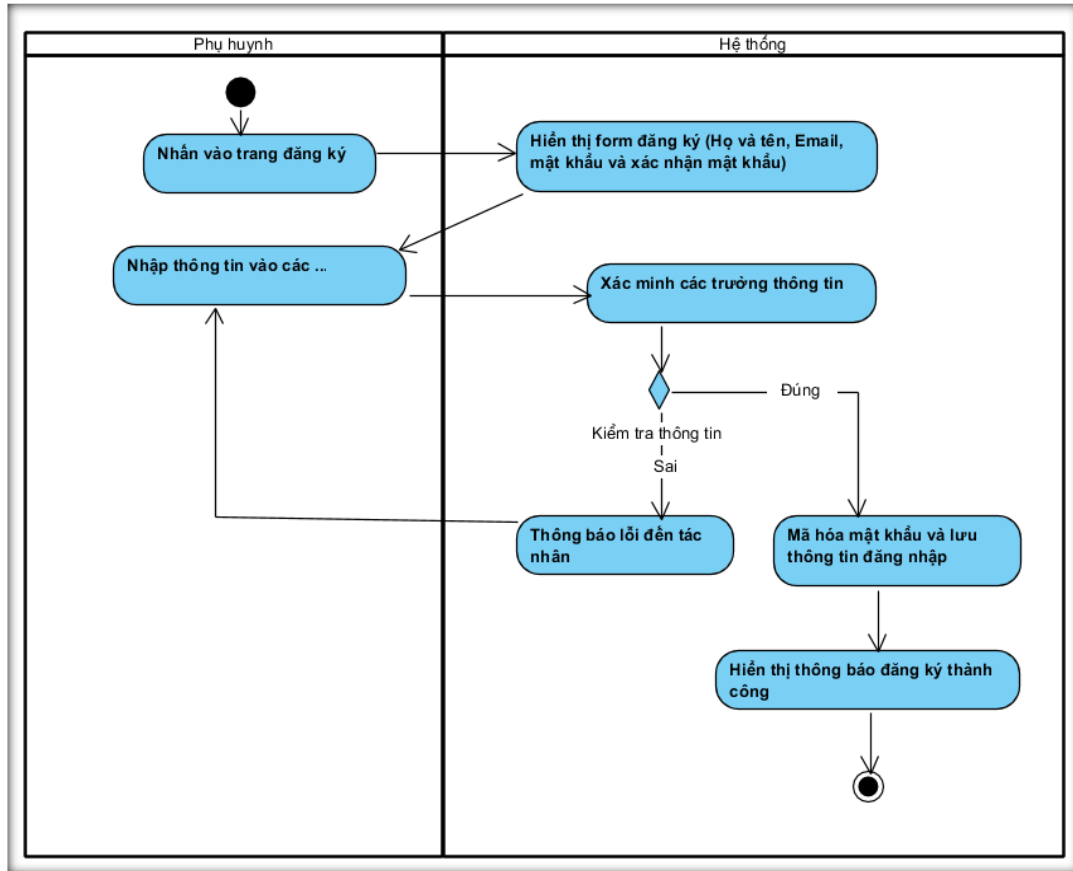
- Xoá bài học

Bảng 2.18. Kịch bản sử dụng xoá bài học

Tên Use Case	Xoá bài học
Tác nhân	Quản trị viên
Mô tả	Cho phép tác nhân xoá một bài học không còn sử dụng khỏi hệ thống.
Sự kiện kích hoạt	Tác nhân nhấn nút xoá ở bài học cần xoá.
Tiền điều kiện	Tác nhân đã đăng nhập hệ thống Tác nhân có quyền quản lý bài học
Đảm bảo tối thiểu	Không xoá bài học khi chưa xác nhận
Đảm bảo thành công	Bài học được xoá khỏi hệ thống thành công
Chuỗi sự kiện chính	1) tác nhân truy cập chức năng Quản lý bài học. 2) Hệ thống hiển thị danh sách các bài học. 3) Tác nhân chọn bài học cần xoá. 4) Tác nhân nhấn nút Xoá bài học. 5) Hệ thống hiển thị hộp thoại xác nhận xoá. 6) Tác nhân xác nhận xoá bài học. 7) Hệ thống xoá bài học khỏi cơ sở dữ liệu. 8) Hệ thống thông báo xoá thành công và cập nhật danh sách bài học.
Ngoại lệ	6.a Tác nhân hủy thao tác xoá. Hệ thống đóng hộp thoại, không xoá dữ liệu.

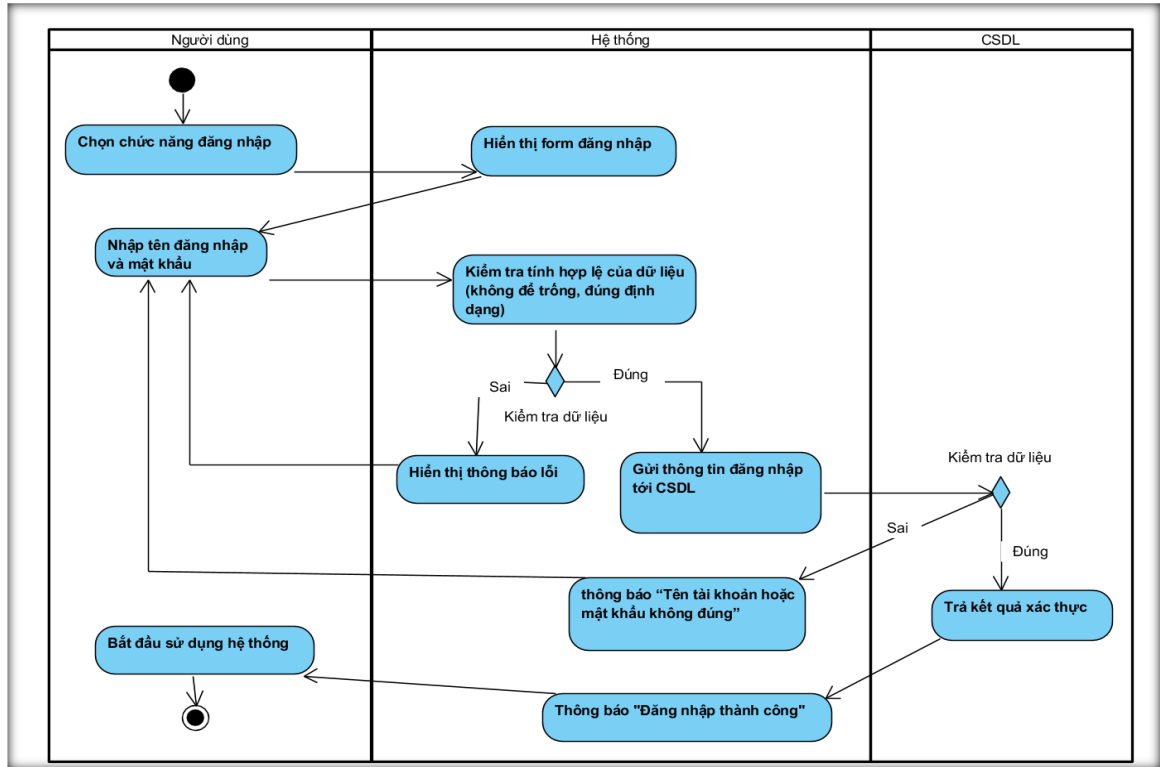
2.5. Sơ đồ hoạt động (Activity)

2.5.1. Sơ đồ hoạt động Đăng ký

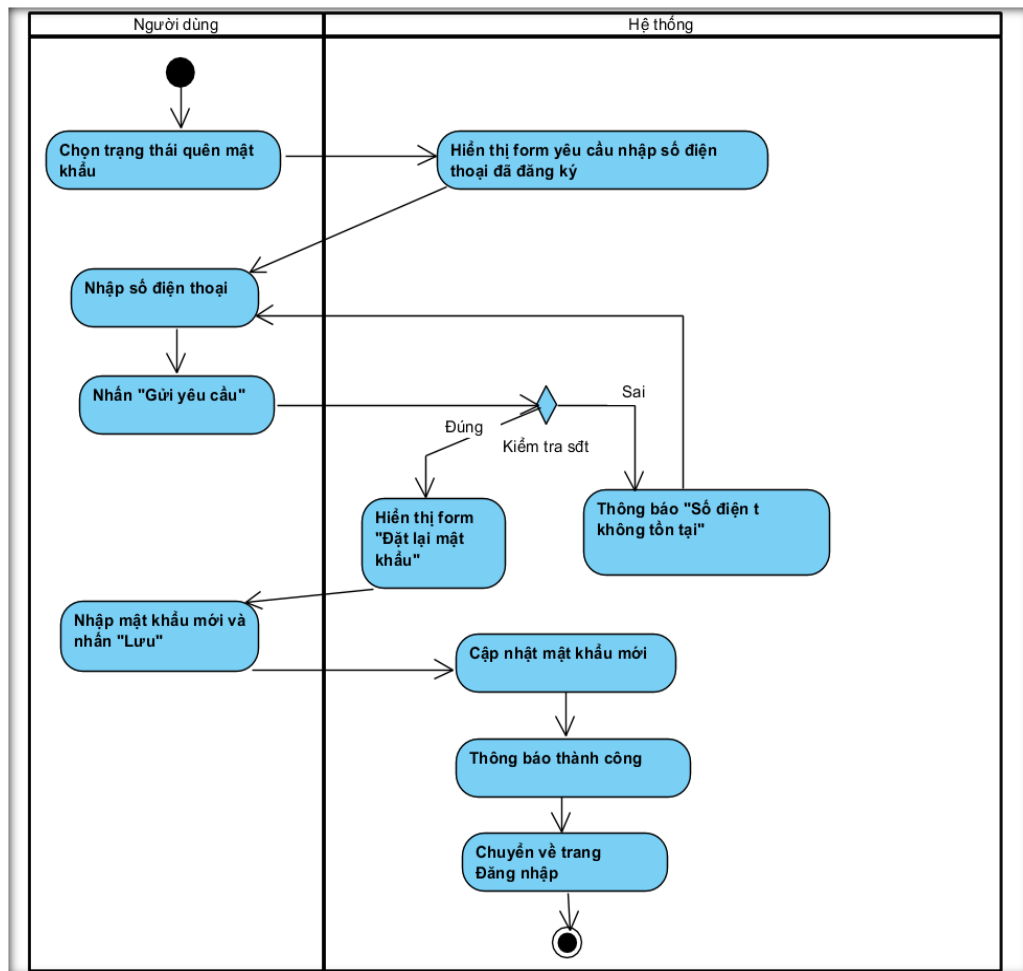


Hình 2.9. Sơ đồ hoạt động đăng ký

2.5.2. Sơ đồ hoạt động Đăng nhập

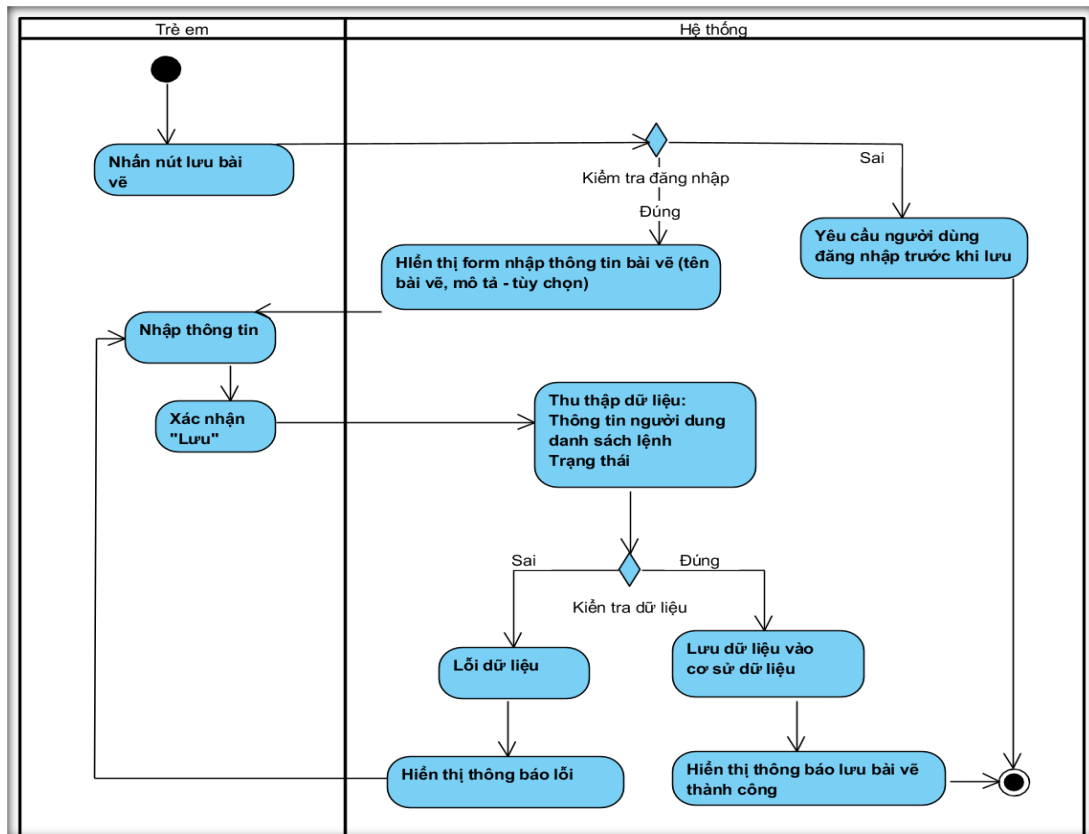


Hình 2.10. Sơ đồ hoạt động đăng nhập

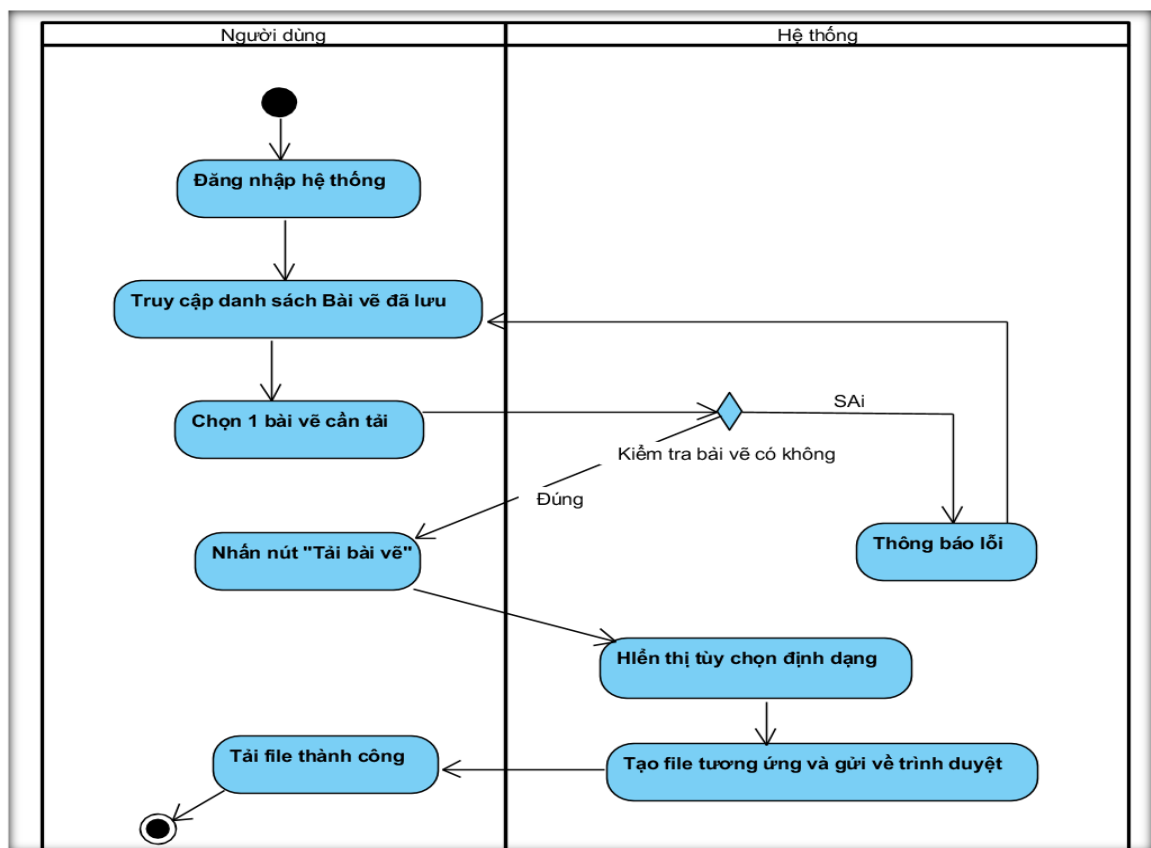


Hình 2.11. Sơ đồ hoạt động quên mật khẩu

2.5.3. Sơ đồ hoạt động Quản lý bài vẽ

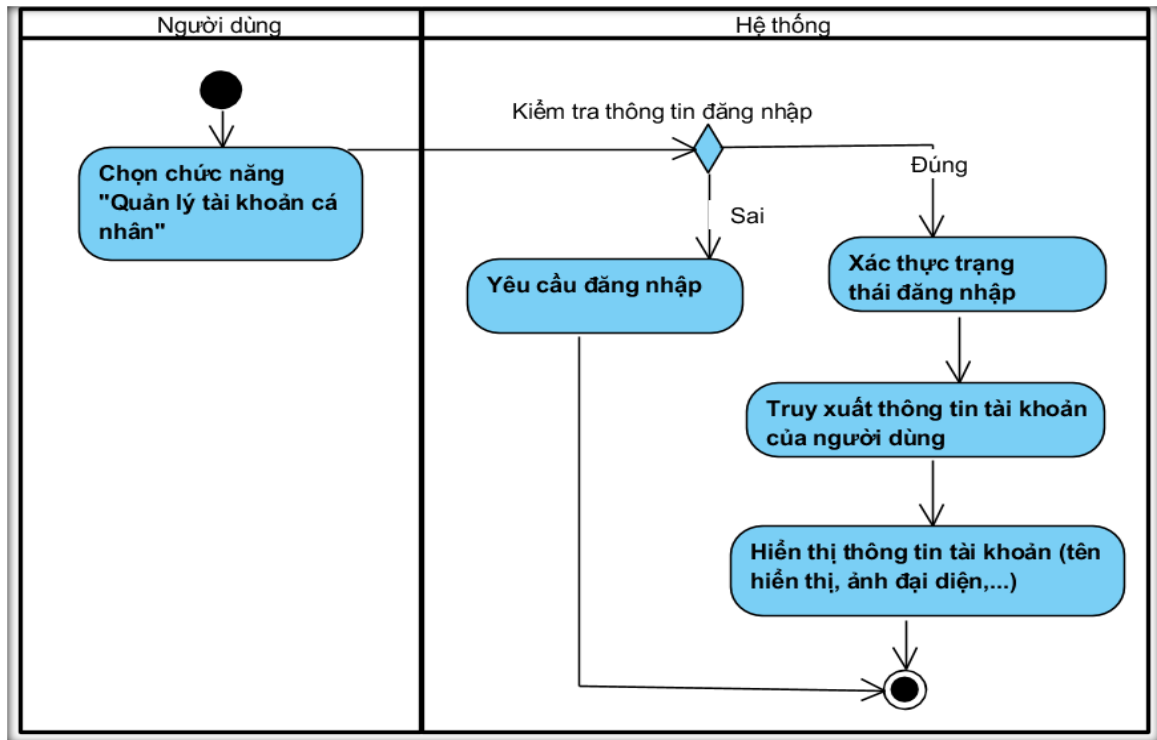


Hình 2.12. Sơ đồ hoạt động lưu bài vẽ

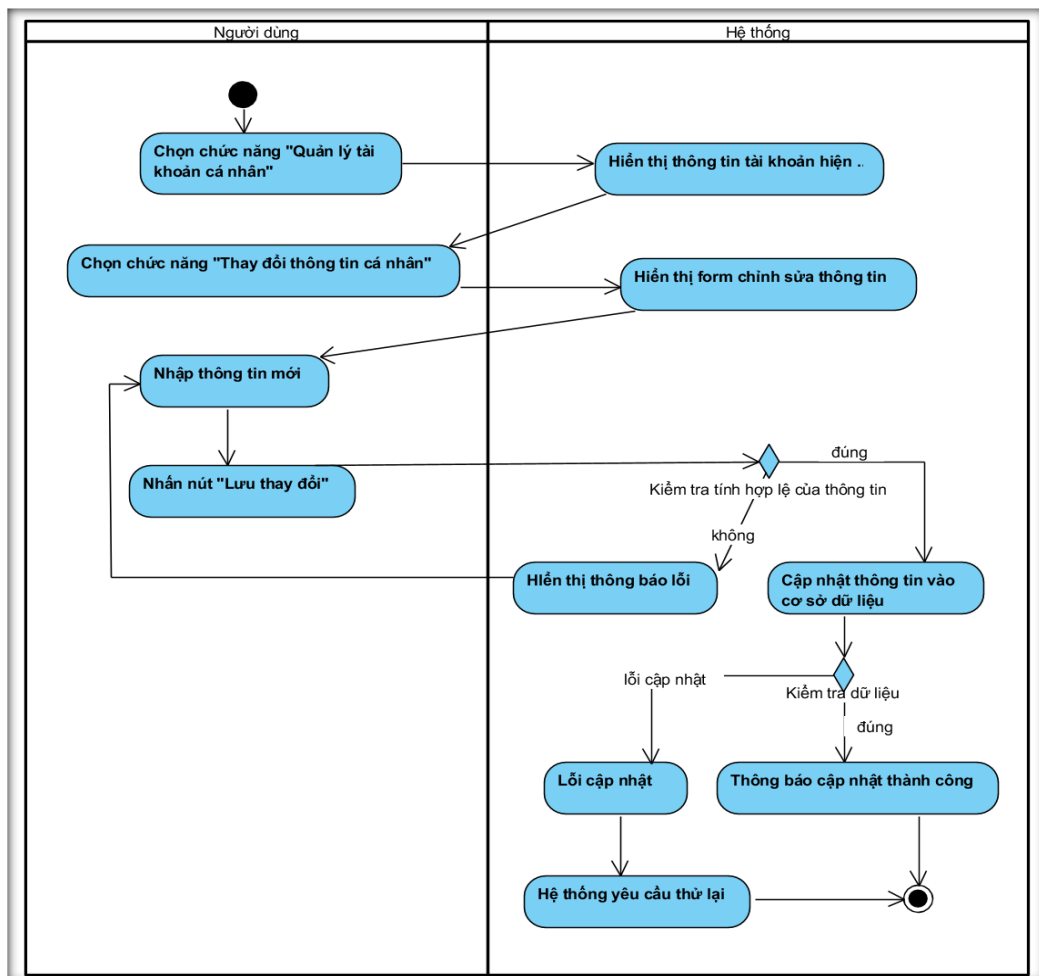


Hình 2.13. Sơ đồ hoạt động tải bài vẽ

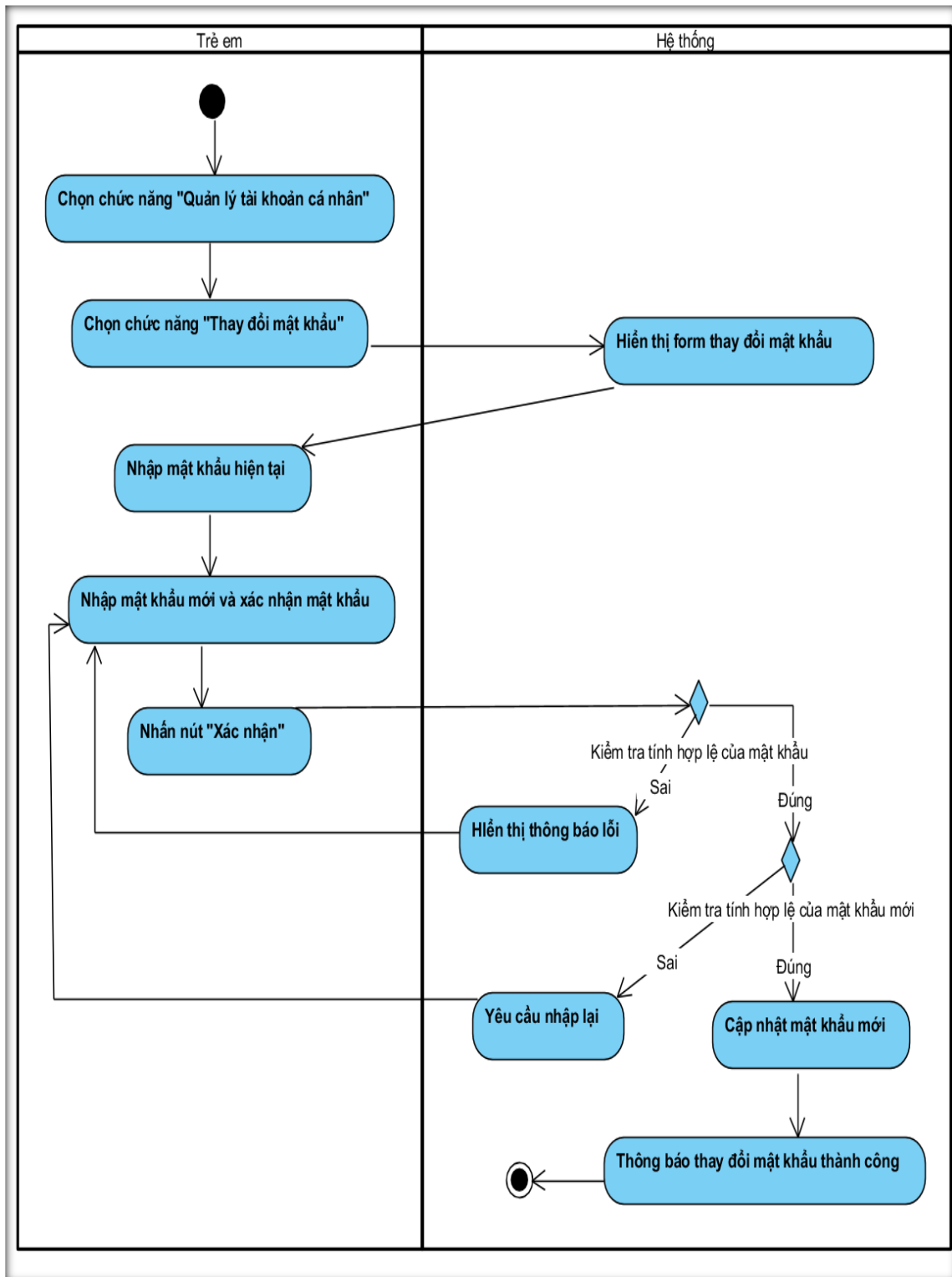
2.5.4. Sơ đồ hoạt động Quản lý tài khoản cá nhân



Hình 2.14. Sơ đồ hoạt động Xem thông tin tài khoản

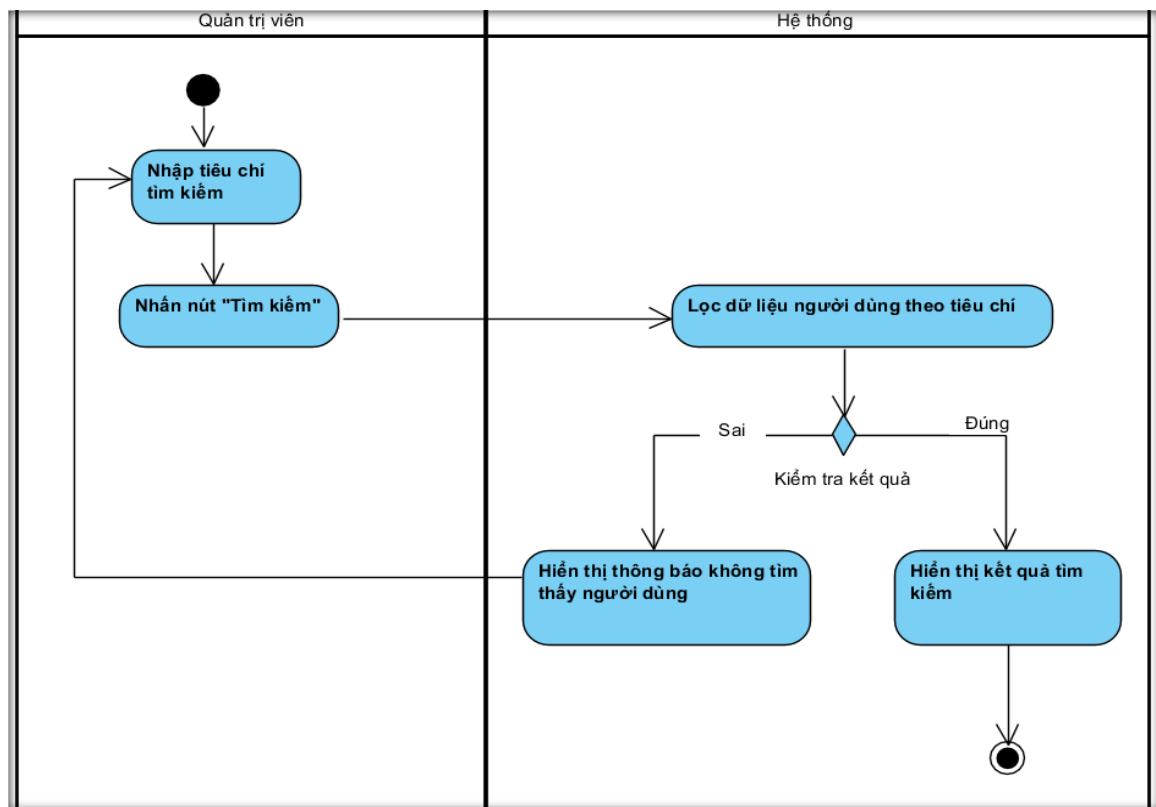


Hình 2.15. Sơ đồ hoạt động cập nhật thông tin tài khoản

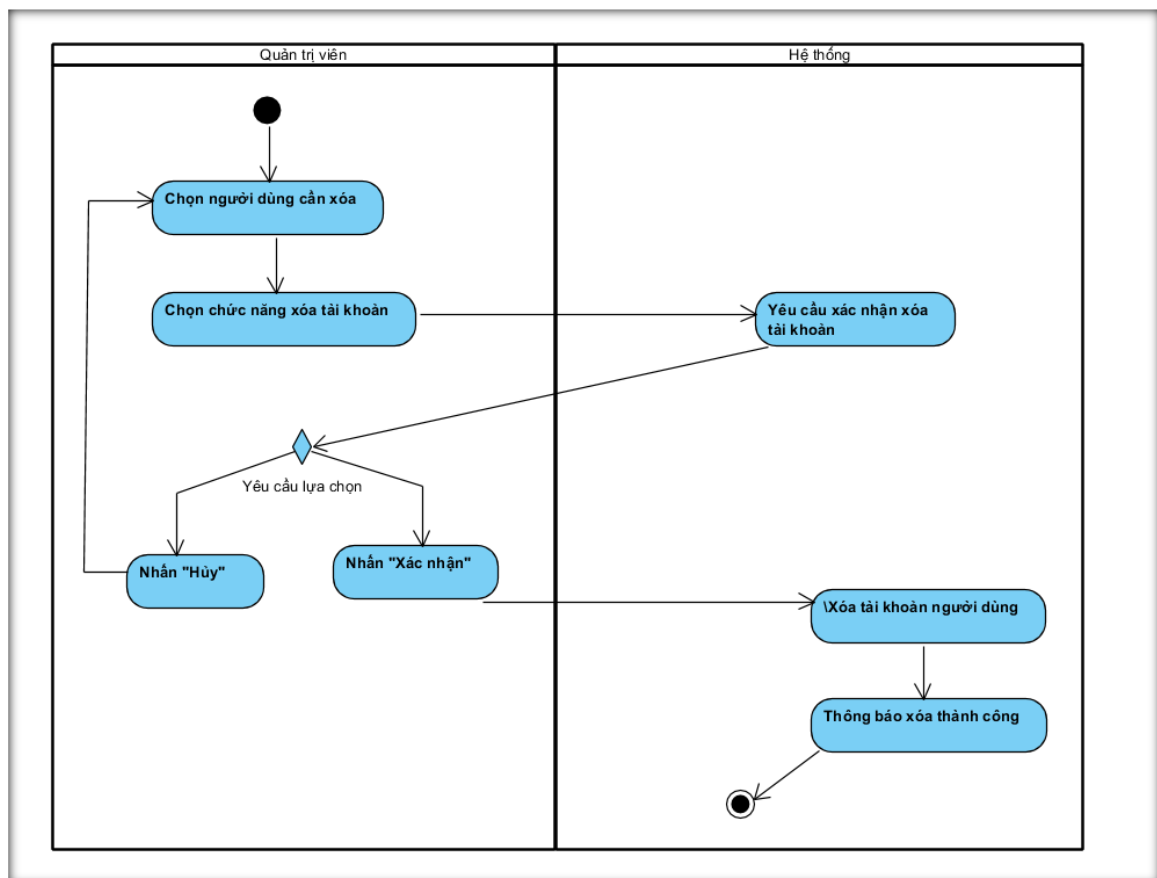


Hình 2.16. Sơ đồ hoạt động đổi mật khẩu

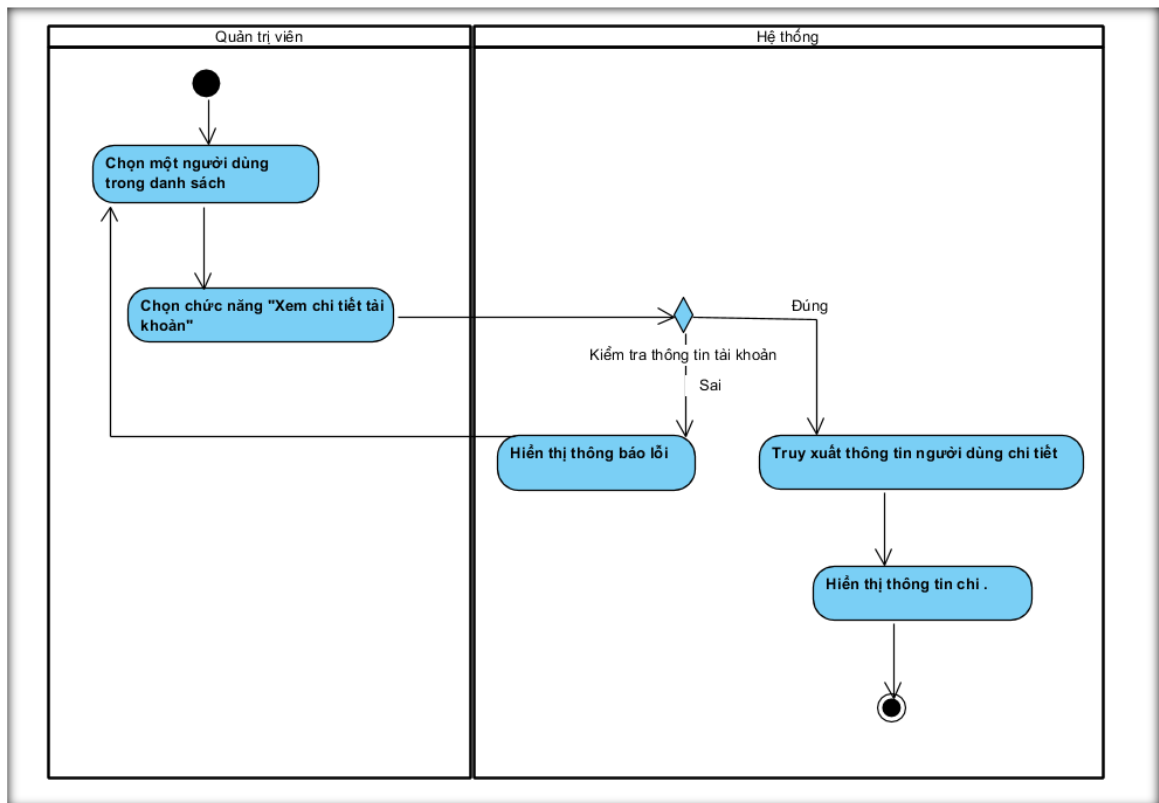
2.5.5. Sơ đồ hoạt động Quản lý người dùng



Hình 2.17. Sơ đồ hoạt động tìm kiếm người dùng

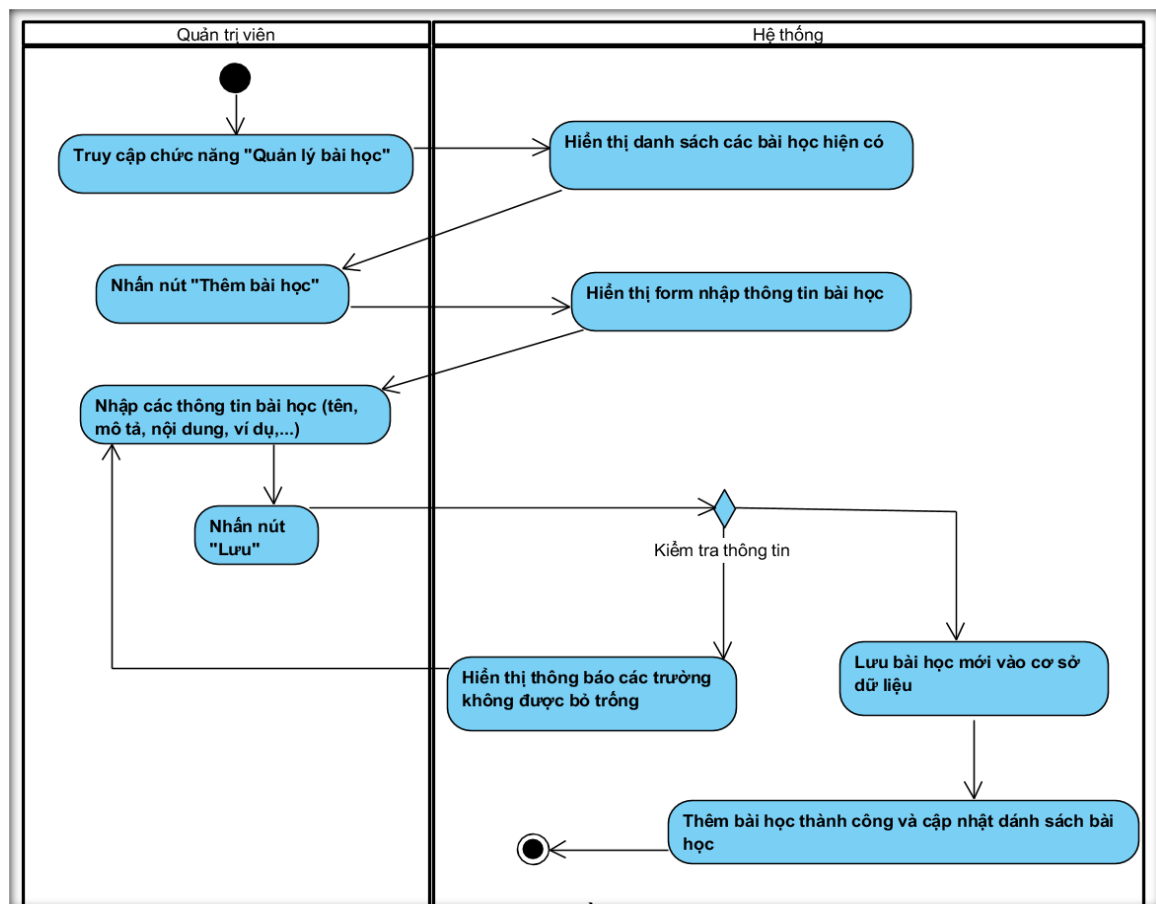


Hình 2.18. Sơ đồ hoạt động xóa tài khoản người dùng

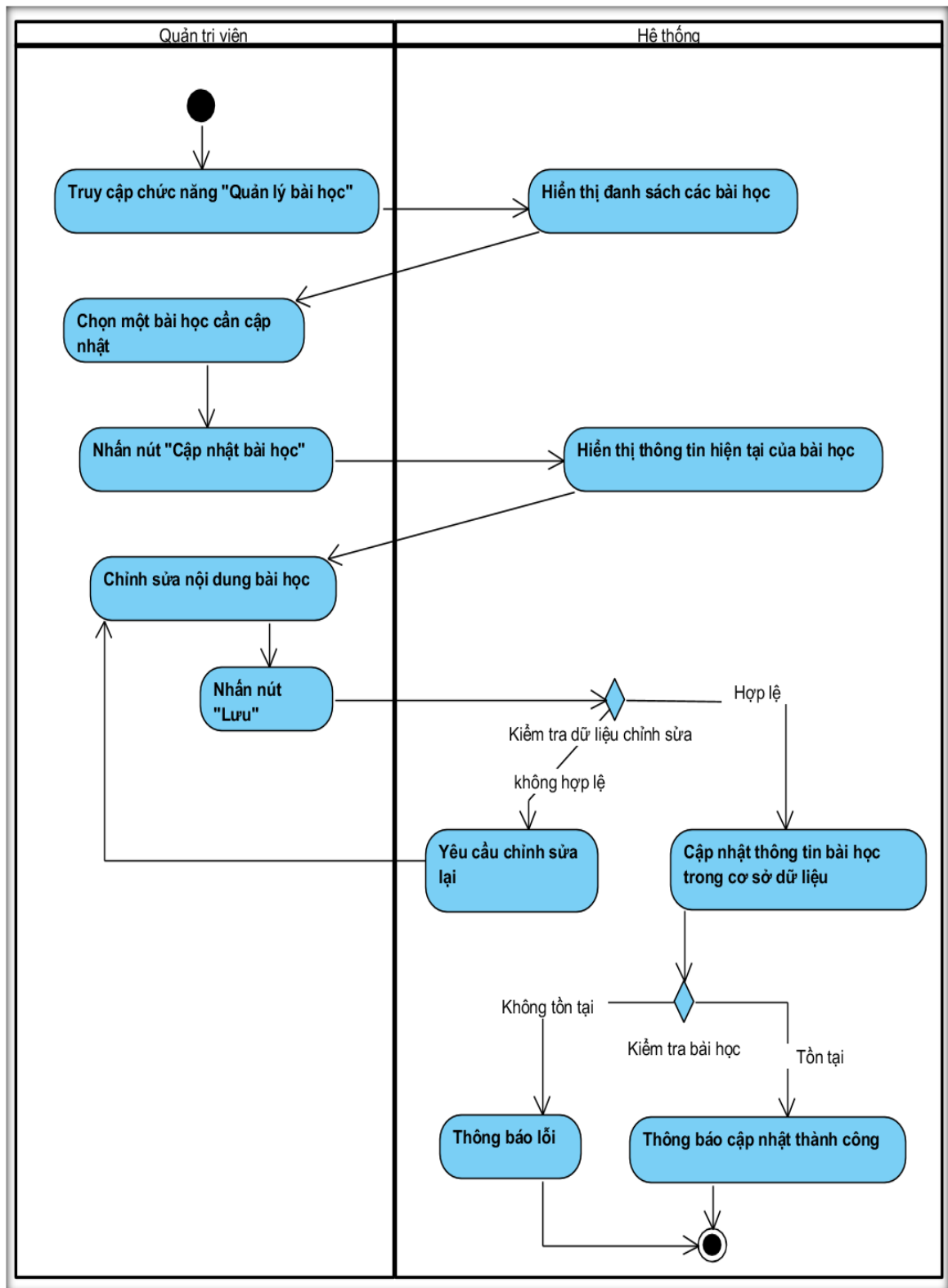


Hình 2.19. Sơ đồ hoạt động xem chi tiết tài khoản người dùng

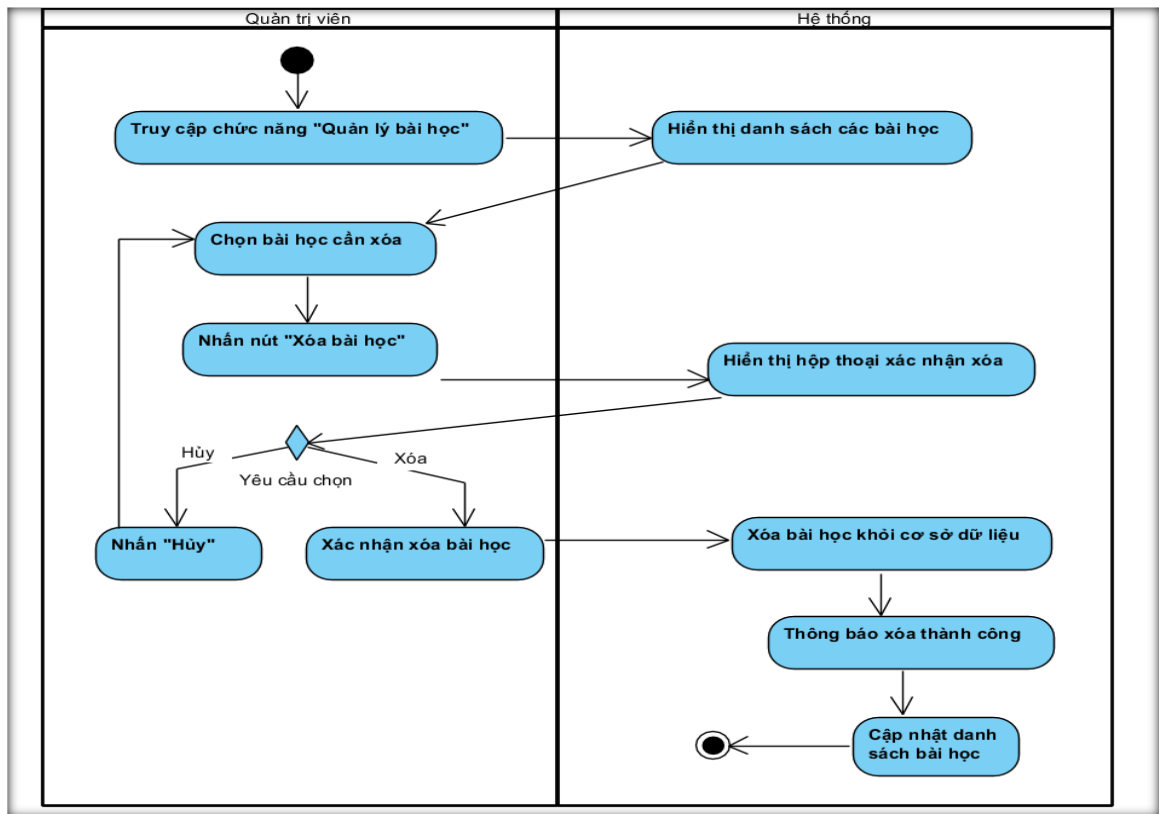
2.5.6. Sơ đồ hoạt động Quản lý bài học



Hình 2.20. Sơ đồ hoạt động thêm bài học

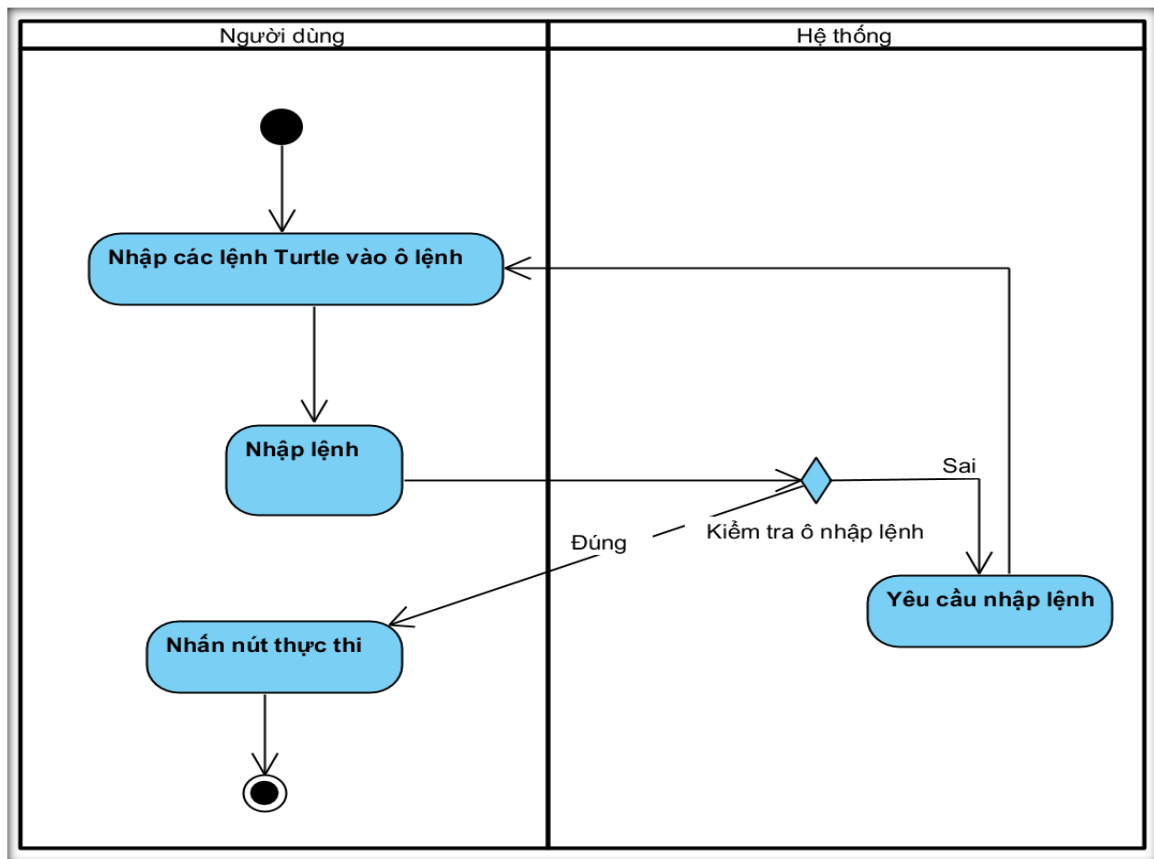


Hình 2.21. Sơ đồ hoạt động cập nhật bài học

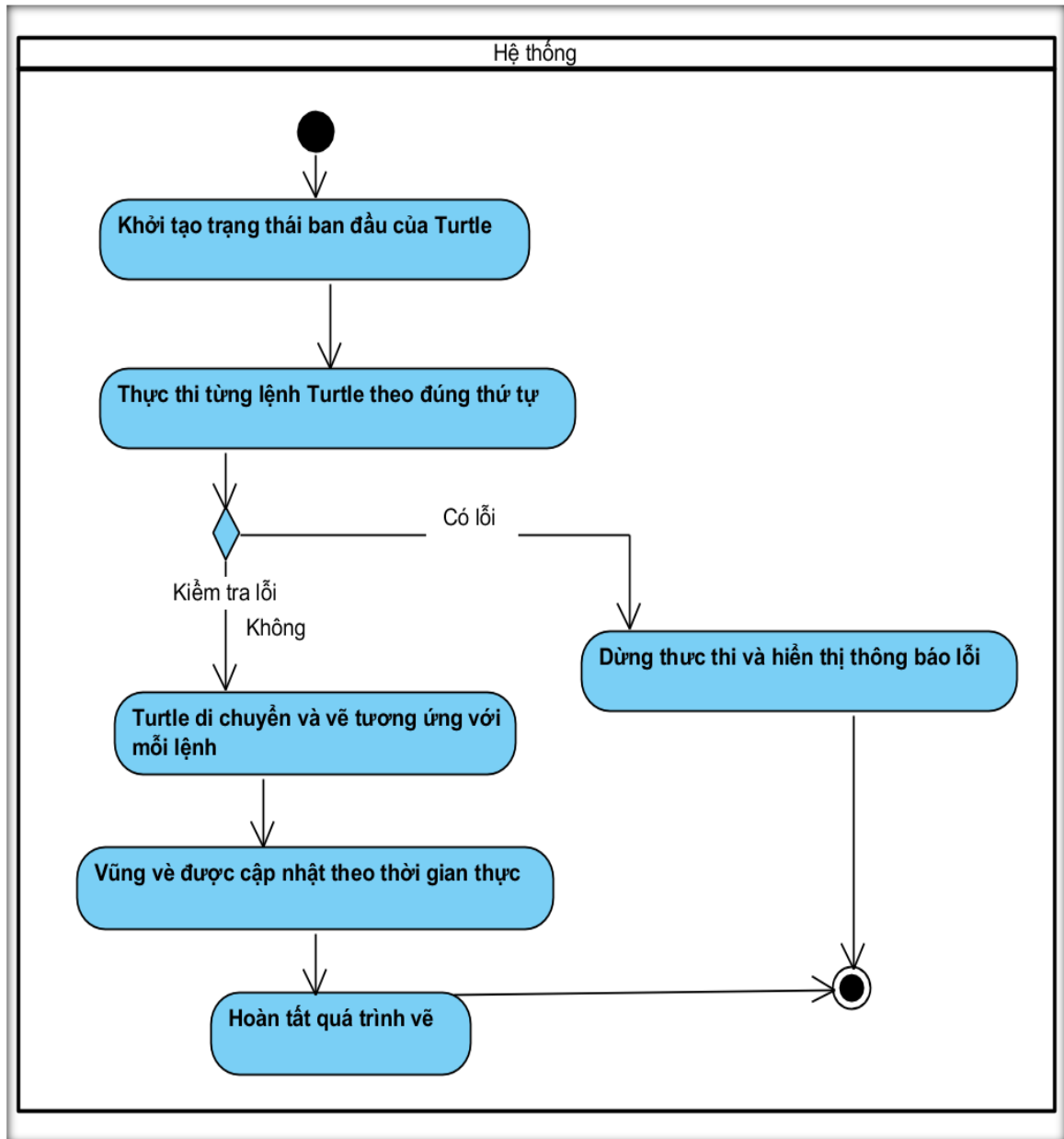


Hình 2.22. Sơ đồ hoạt động xóa bài học

2.5.7. Sơ đồ hoạt động Thực thi chương trình

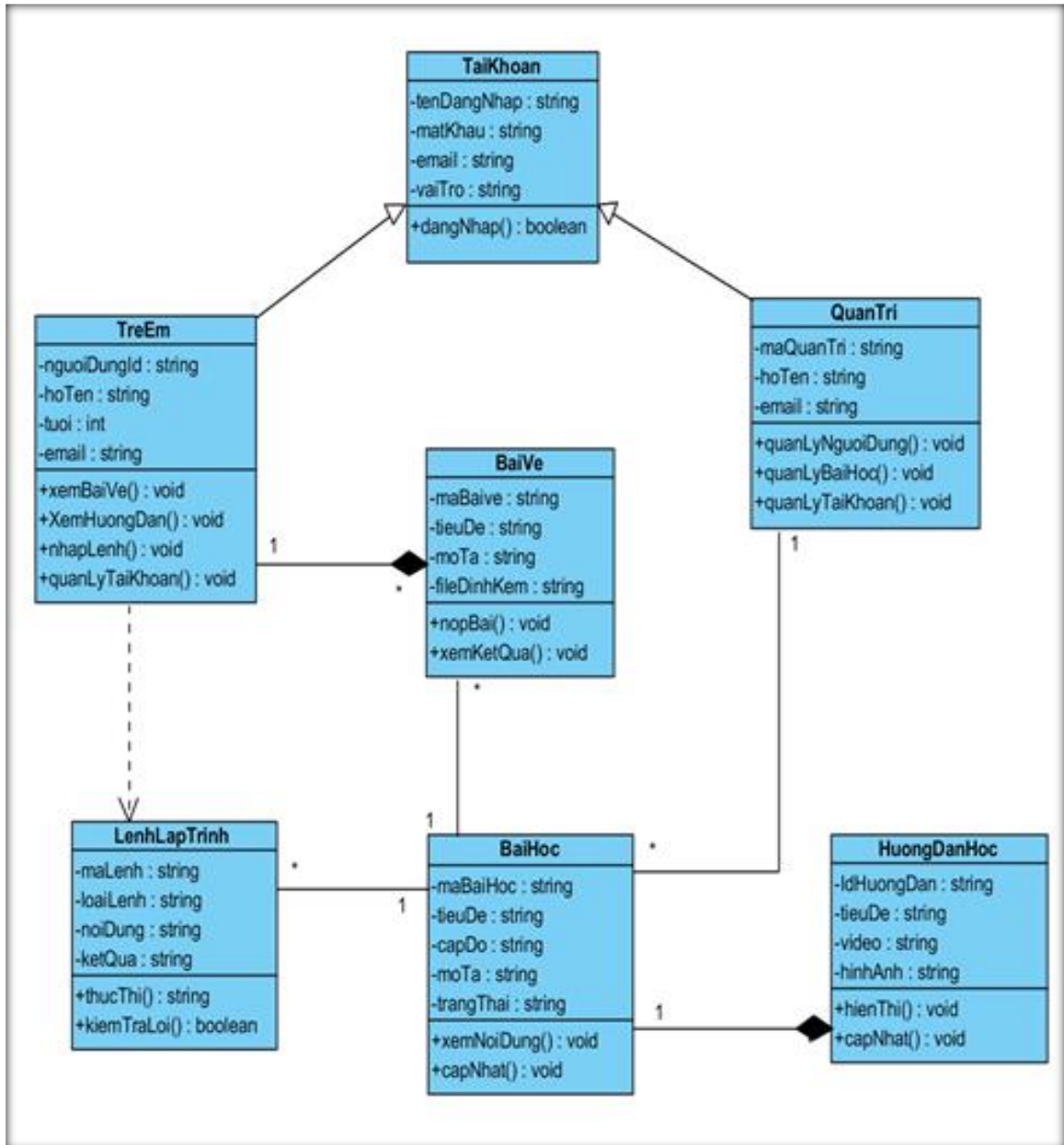


Hình 2.23. Sơ đồ hoạt động nhập lệnh



Hình 2.24. Sơ đồ hoạt động thực thi lệnh

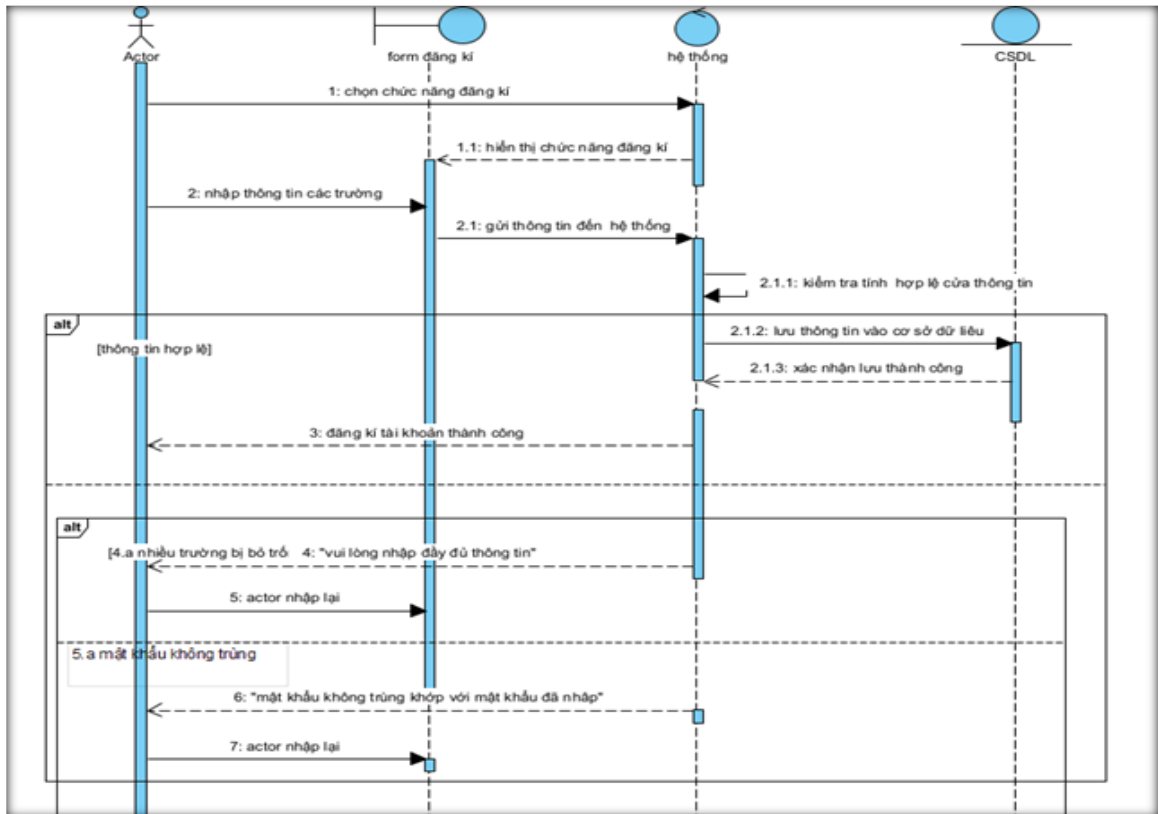
2.6. Sơ đồ lớp (Class Diagram)



Hình 2.25. Sơ đồ lớp

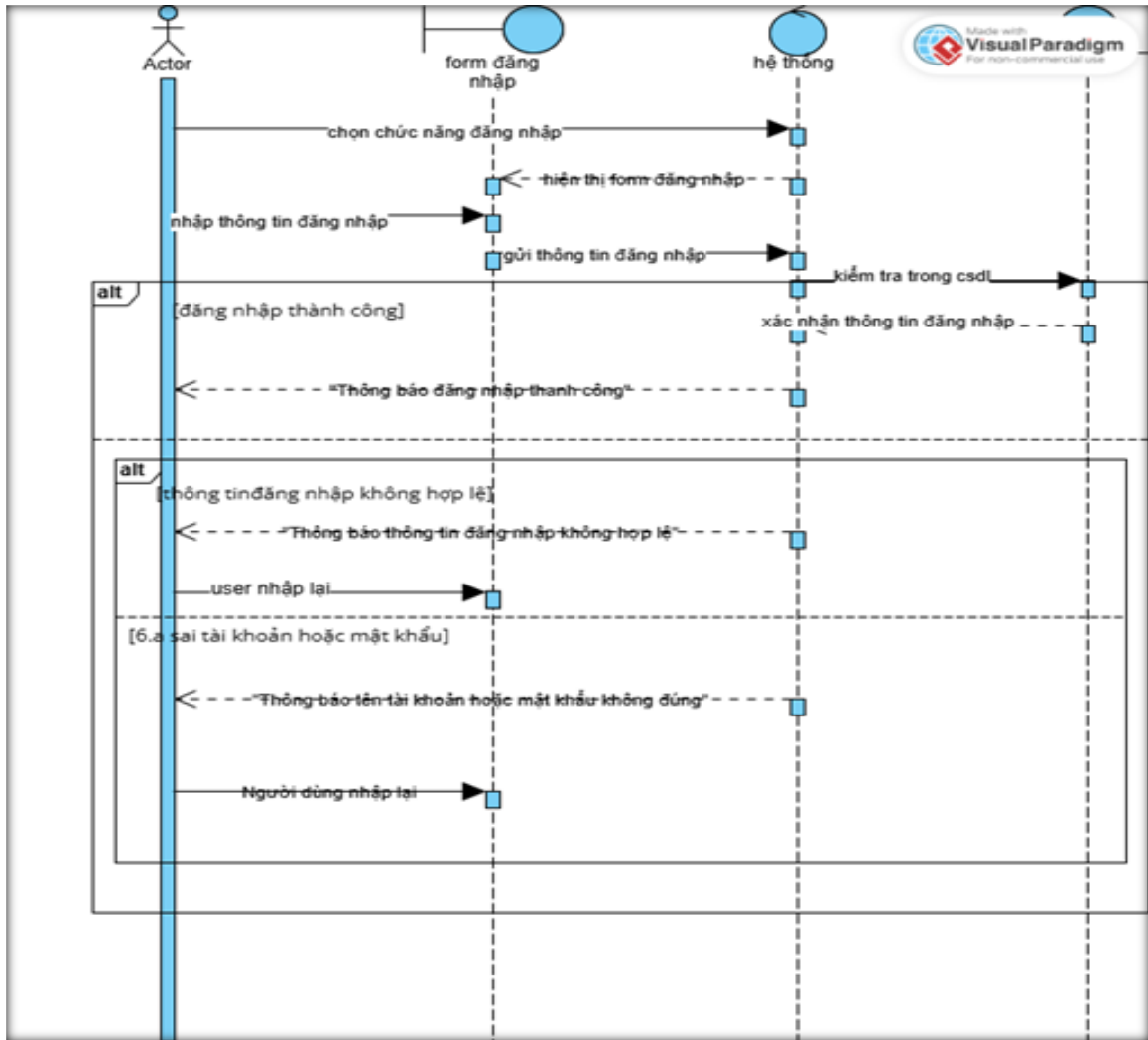
2.7. Sơ đồ tuần tự (Sequence Diagram)

2.7.1. Sơ đồ tuần tự Đăng ký

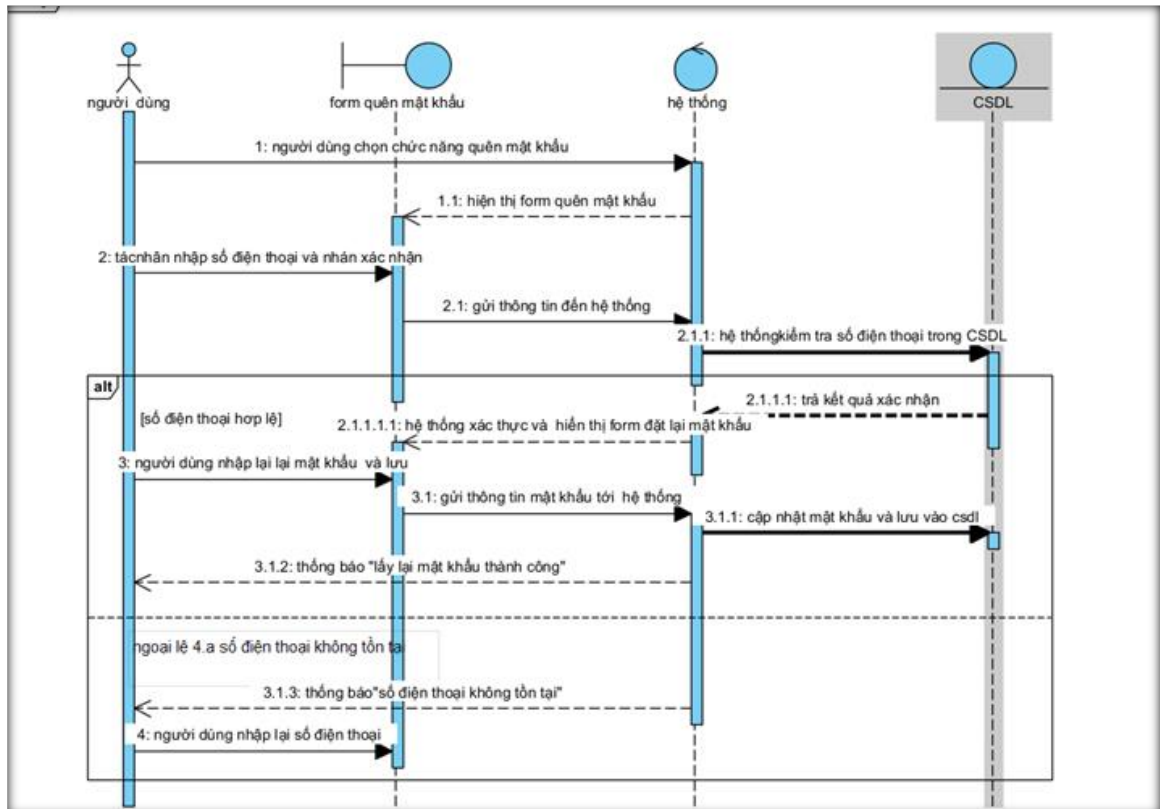


Hình 2.26. Sơ đồ tuần tự đăng ký

2.7.2. Sơ đồ tuần tự Đăng nhập

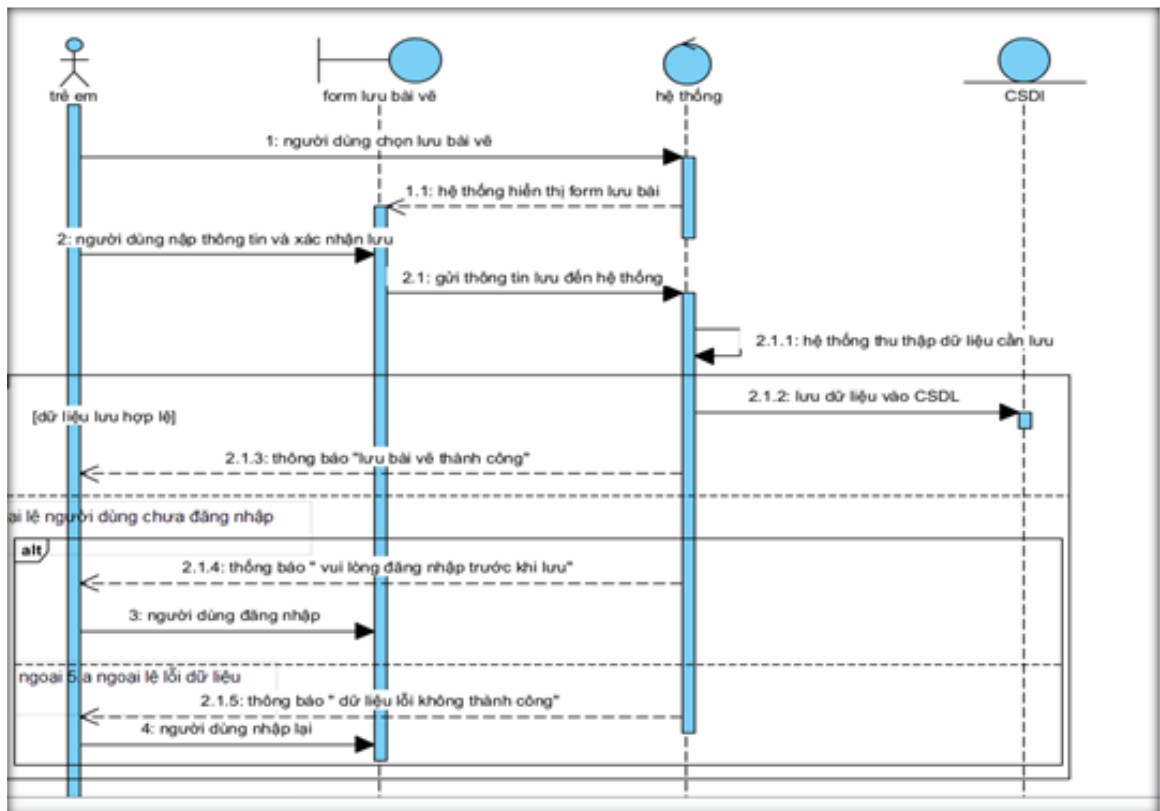


Hình 2.27. Sơ đồ tuần tự đăng nhập

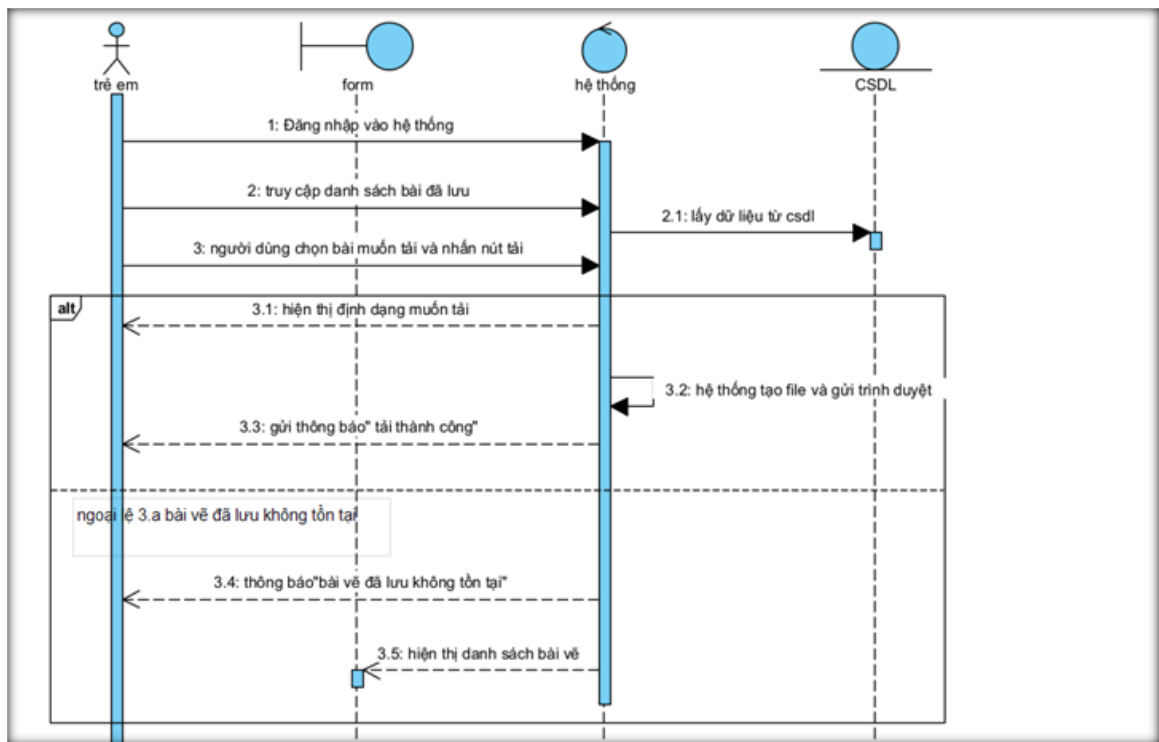


Hình 2.28. Sơ đồ tuần tự quên mật khẩu

2.7.3. Sơ đồ tuần tự quản lý bài vẽ

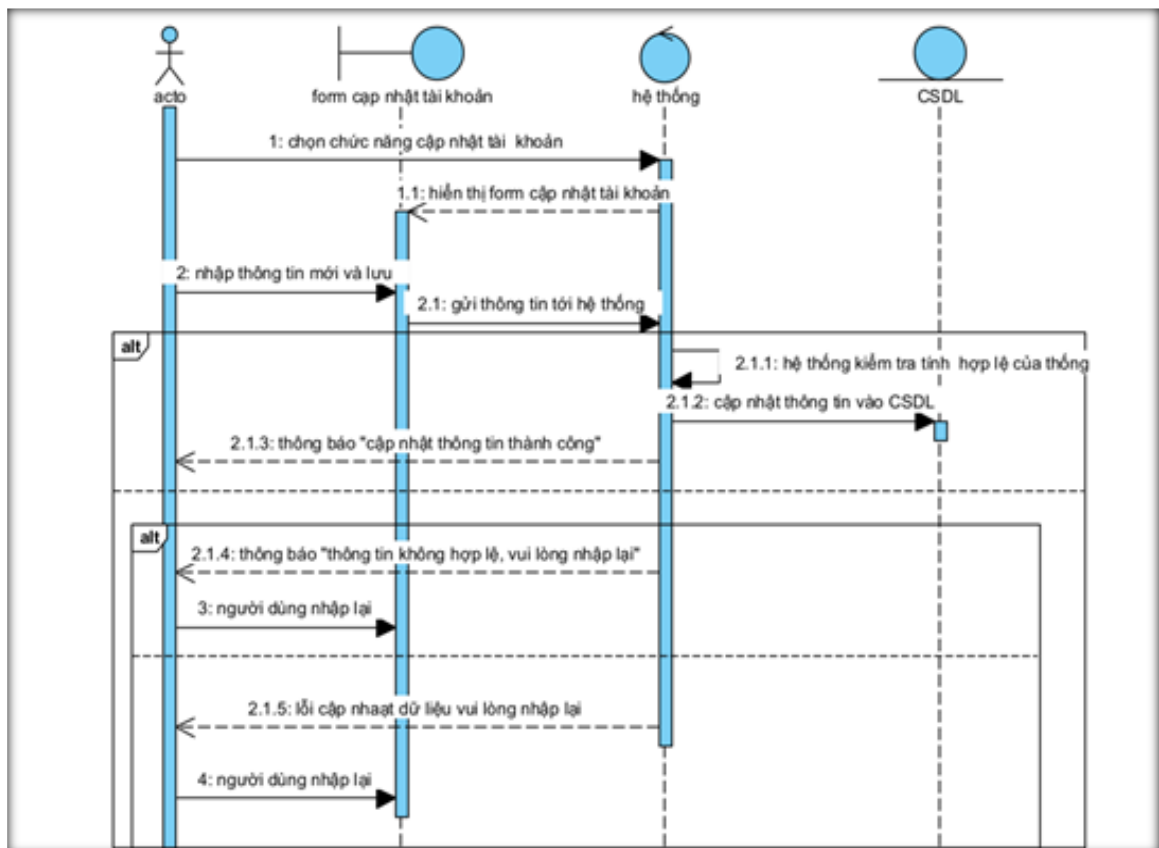


Hình 2.29. Sơ đồ tuần tự lưu bài vẽ

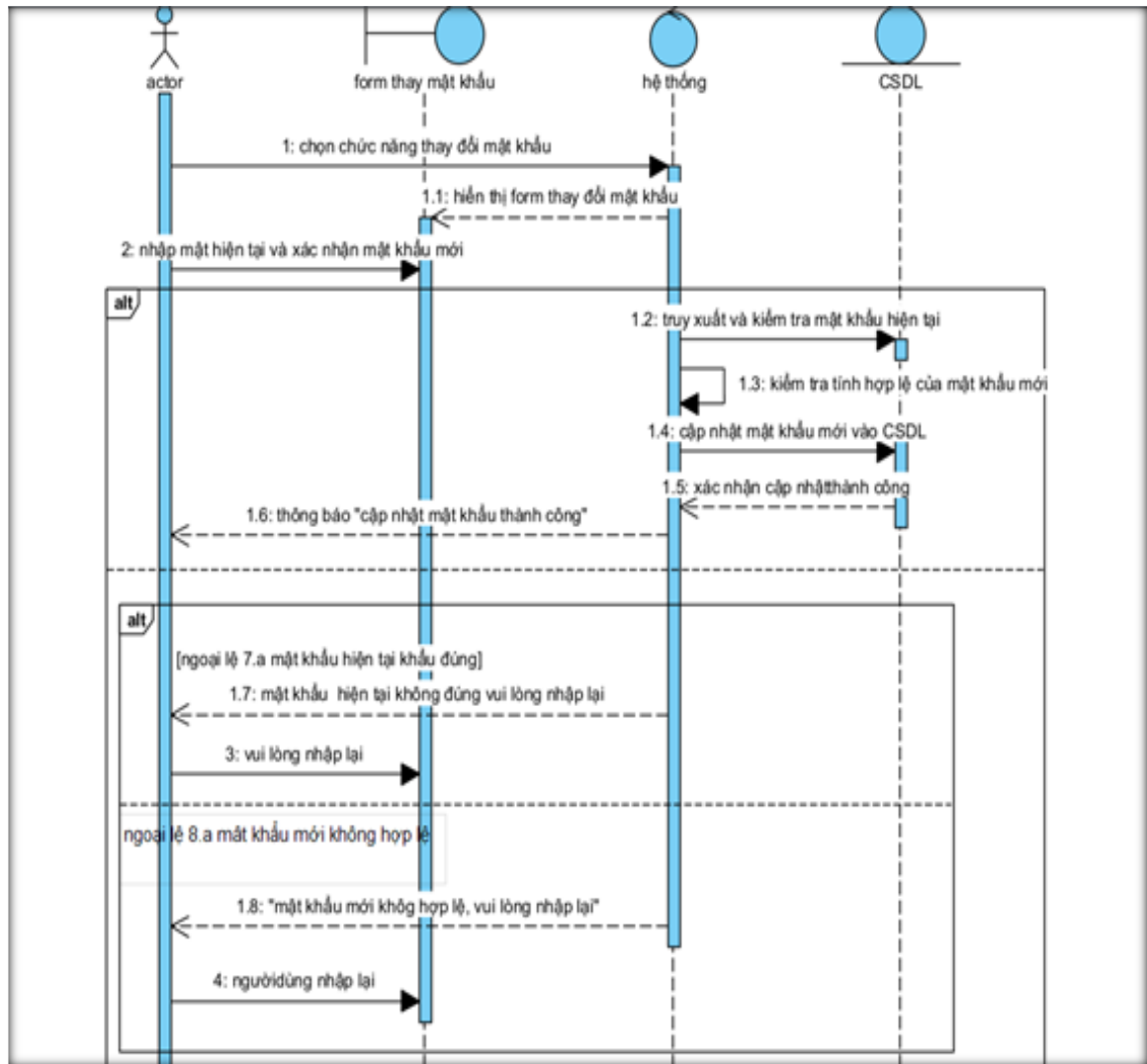


Hình 2.30. Sơ đồ tuần tự tải bài vẽ

2.7.4. Sơ đồ tuần tự quản lý tài khoản cá nhân

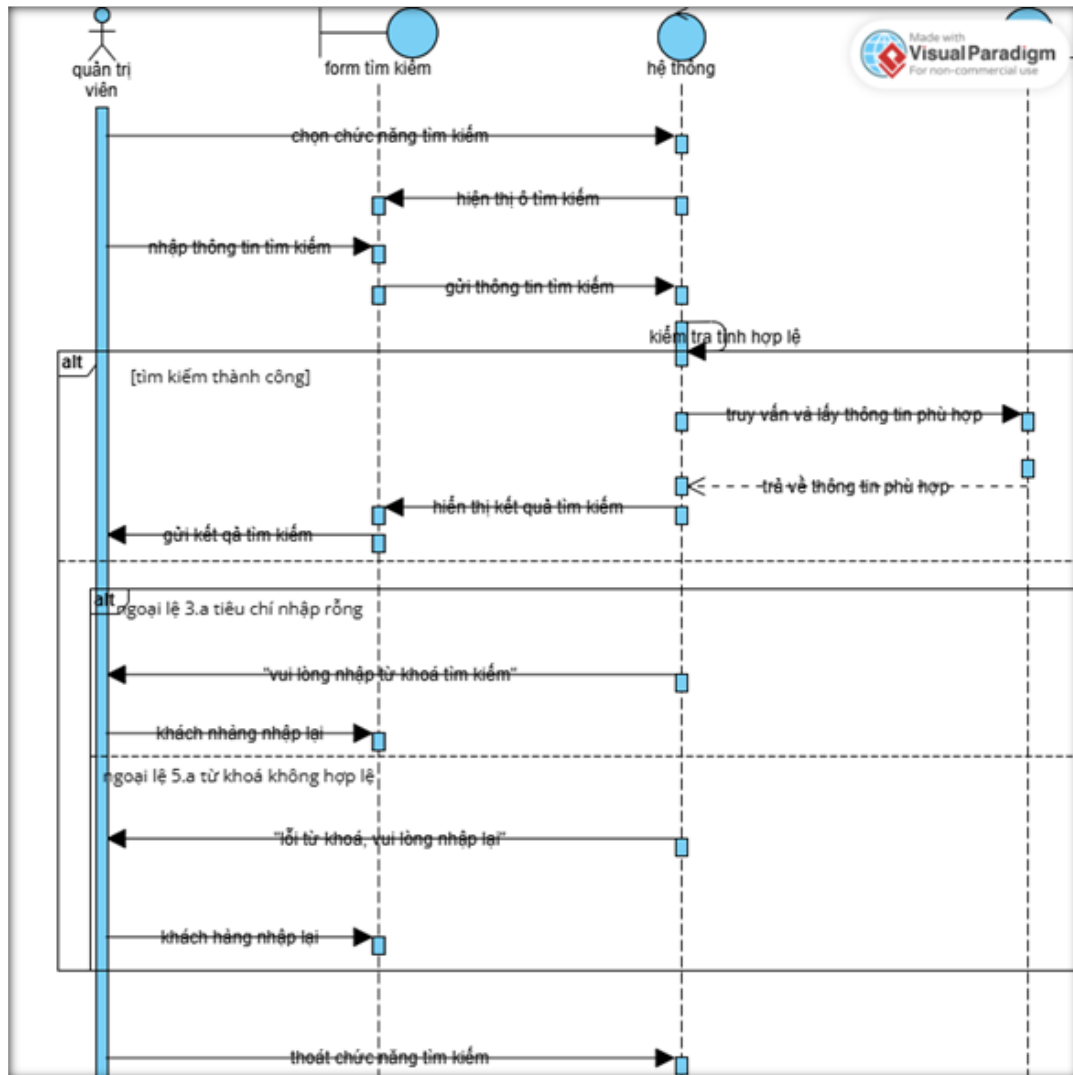


Hình 2.31. Sơ đồ tuần tự cập nhật tài khoản cá nhân

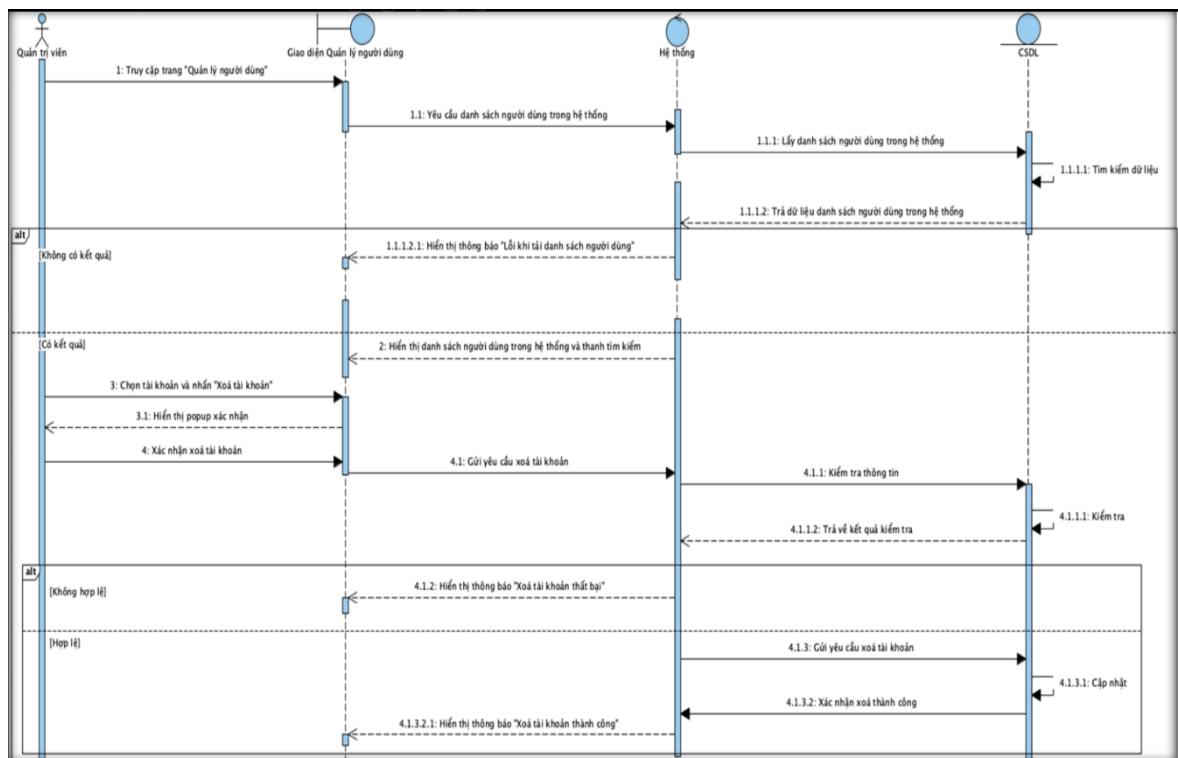


Hình 2.32. Sơ đồ tuần tự đổi mật khẩu

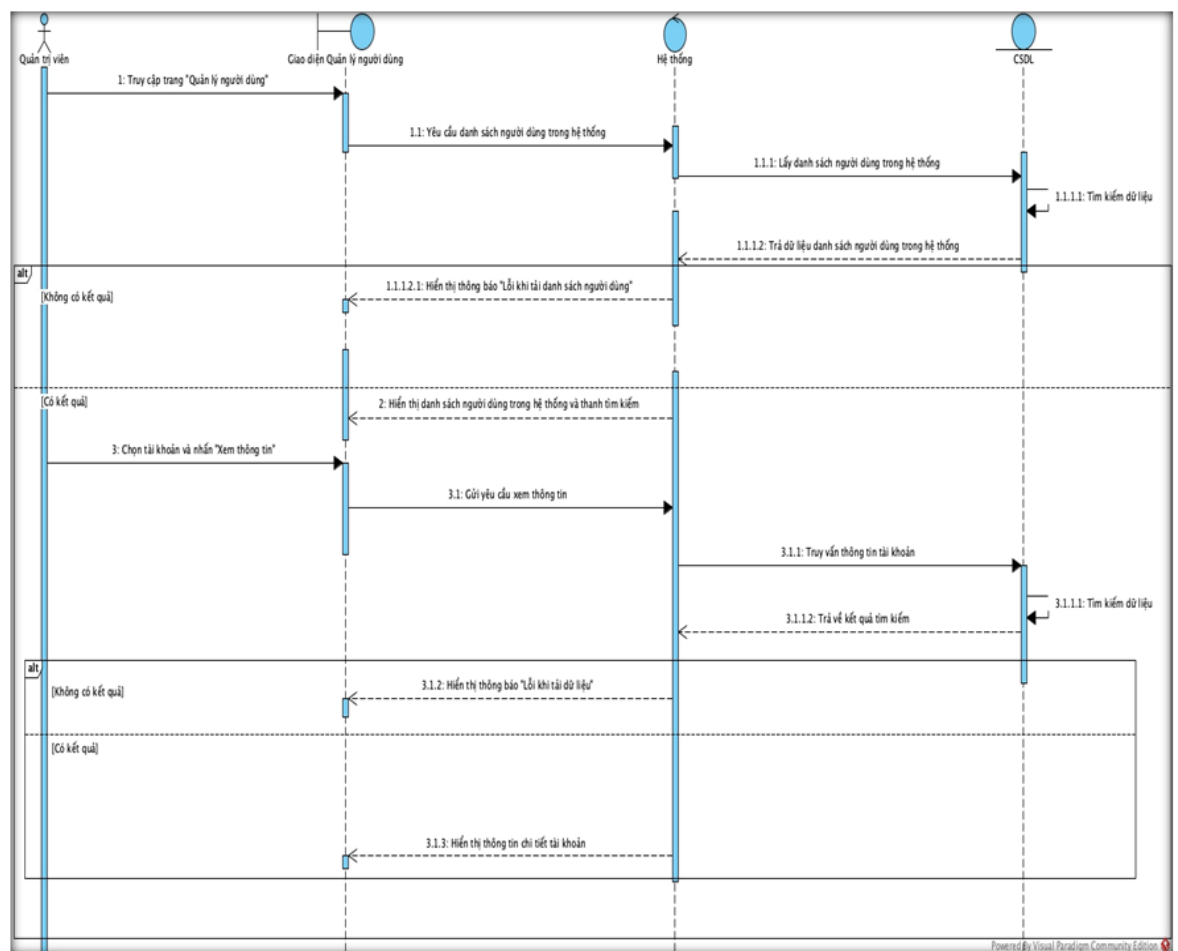
2.7.5. Sơ đồ tuần tự Quản lý người dùng



Hình 2.33. Sơ đồ tuần tự tìm kiếm người dùng



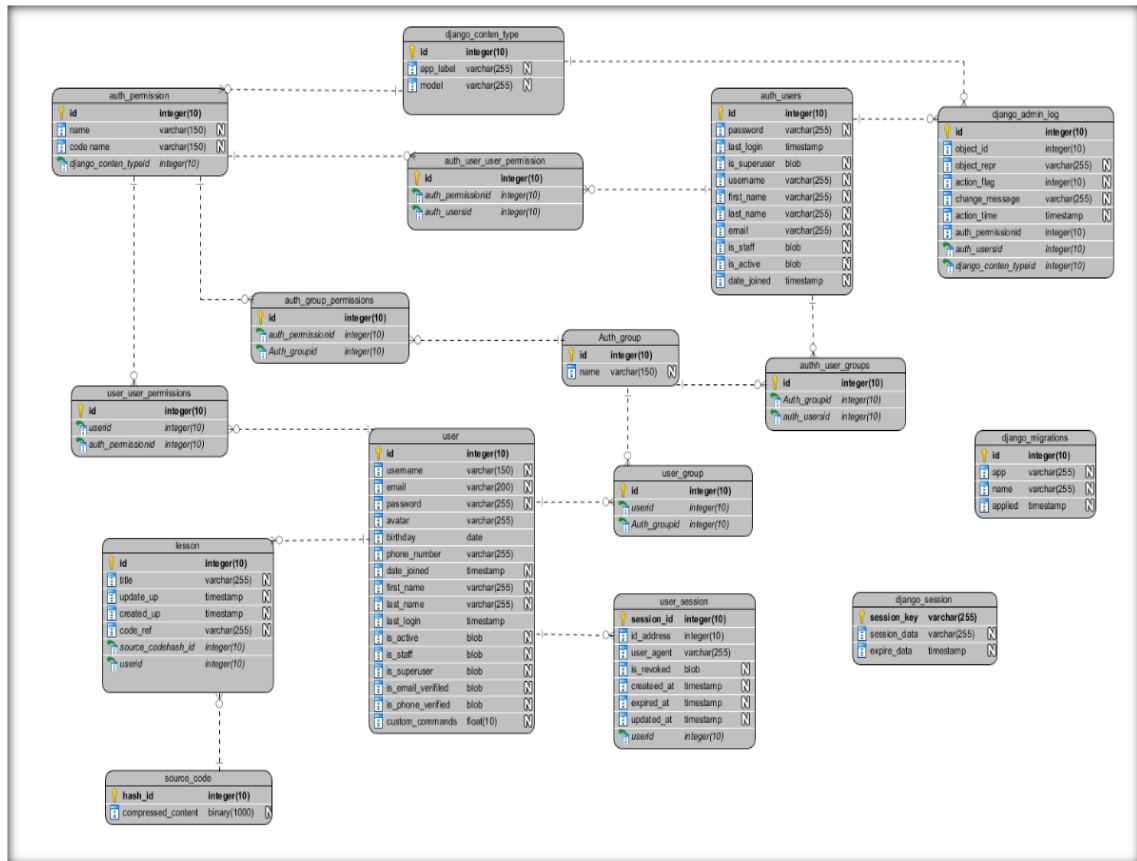
Hình 2.34. Sơ đồ tuần tự xóa tài khoản người dùng



Hình 2.35. Sơ đồ tuần tự xem chi tiết tài khoản

2.8. Thiết kế cơ sở dữ liệu

2.8.1. Sơ đồ ERD



Hình 2.36. Sơ đồ ERD

2.8.2. Mô tả dữ liệu

Bảng **auth_user** là bảng xác thực người dùng được Django tự động sinh ra, chịu trách nhiệm lưu trữ thông tin đăng nhập và phân quyền cơ bản

Bảng 2.19. Mô tả dữ liệu bảng *auth user*

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh người dùng
username	Varchar (150)	Tên đăng nhập (duy nhất)
password	Varchar (128)	Mật khẩu đã mã hóa
email	Varchar (254)	Địa chỉ email
first_name	Varchar (150)	Tên
last_name	Varchar (150)	Họ
is_active	Boolean	Trạng thái hoạt động của tài khoản
is_staff	Boolean	Quyền truy cập trang quản trị (admin)
is_superuser	Boolean	Quyền quản trị cao nhất

Tên trường	Kiểu dữ liệu	Mô tả
last_login	Timestamp	Thời điểm đăng nhập gần nhất
date_joined	Timestamp	Ngày tạo tài khoản

Bảng **auth_group** lưu danh sách các nhóm quyền dùng để phân quyền hàng loạt cho người dùng.

Bảng 2.20. Mô tả dữ liệu bảng auth_group

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh nhóm
name	Varchar (150)	Tên nhóm quyền (duy nhất)

Bảng **auth_permission** định nghĩa các quyền hạn cụ thể trong hệ thống, được liên kết với nhóm và người dùng thông qua các bảng trung gian

Bảng 2.21. Mô tả dữ liệu bảng auth_permission

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh quyền
name	Varchar (150)	Tên mô tả quyền
codename	Varchar	Mã quyền dạng rút gọn
content_type_id	Int (FK)	Liên kết tới bảng django_content_type

Bảng **auth_group_permissions** là bảng trung gian thể hiện quan hệ N–N giữa nhóm và quyền hạn

Bảng 2.22. Mô tả dữ liệu bảng auth_group_permission

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh bản ghi
group_id	Int (FK)	Liên kết tới bảng auth_group
permission_id	Int (FK)	Liên kết tới bảng auth_permission

Bảng **auth_user_user_permissions** lưu quan hệ N–N giữa người dùng và quyền hạn được cấp trực tiếp

Bảng 2.23. Mô tả dữ liệu bảng auth_user_user_permission

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh bản ghi
user_id	Int (FK)	Liên kết tới bảng auth_user
permission_id	Int (FK)	Liên kết tới bảng auth_permission

Bảng **auth_user_groups** lưu quan hệ N–N giữa người dùng và nhóm quyền

Bảng 2.24. Mô tả dữ liệu bảng *auth_user_groups*

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh bản ghi
user_id	Int (FK)	Liên kết tới bảng <i>auth_user</i>
group_id	Int (FK)	Liên kết tới bảng <i>auth_group</i>

Bảng **django_content_type** theo dõi toàn bộ các model trong hệ thống, phục vụ cơ chế phân quyền theo nội dung của Django

Bảng 2.25. Mô tả dữ liệu bảng *django_content_type*

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh loại nội dung
app_label	Varchar	Tên ứng dụng chứa model
model	Varchar	Tên model

Bảng **django_session** lưu dữ liệu phiên làm việc phía server theo cơ chế session mặc định của Django

Bảng 2.26. Mô tả dữ liệu bảng *django_session*

Tên trường	Kiểu dữ liệu	Mô tả
session_key	Varchar (PK)	Khóa định danh phiên làm việc
session_data	Text	Dữ liệu phiên đã mã hóa
expire_date	Timestamp	Thời điểm hết hạn phiên

Bảng **django_admin_log** ghi lại toàn bộ lịch sử thao tác của quản trị viên trên trang Django Admin, phục vụ mục đích kiểm tra và giám sát hệ thống

Bảng 2.27. Mô tả dữ liệu bảng *django_admin_log*

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Khóa định danh phiên làm việc
action_time	Timestamp	Thời điểm thực hiện thao tác
object_id	Text	Mã đối tượng bị tác động
object_repr	Varchar	Chuỗi mô tả đối tượng
action_flag	Int	Loại hành động (thêm/sửa/xóa)
change_message	Text	Nội dung thay đổi chi tiết
content_type_id	Int (FK)	Liên kết tới <i>django_content_type</i>
user_id	Int (FK)	Quản trị viên thực hiện thao tác

Bảng **user** là model người dùng tùy chỉnh, thay thế hoàn toàn auth_user mặc định của Django. Ngoài các trường xác thực cơ bản, bảng bổ sung thông tin cá nhân, trạng thái xác minh và các trường đặc thù phục vụ nghiệp vụ của hệ thống

Bảng 2.28. Mô tả dữ liệu bảng user

Tên bảng	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh người dùng
username	Varchar (150)	Tên đăng nhập (duy nhất)
password	Varchar (128)	Mật khẩu đã mã hóa
email	Varchar (254)	Địa chỉ email
first_name	Varchar (150)	Tên
last_name	Varchar (150)	Họ
avatar	Varchar (255)	Đường dẫn ảnh đại diện
birthday	Date	Ngày sinh
phone_number	Varchar (10)	Số điện thoại
is_active	Boolean	Trạng thái hoạt động
is_staff	Boolean	Quyền truy cập trang quản trị
is_superuser	Boolean	Quyền quản trị cao nhất (admin)
is_email_verified	Boolean	Trạng thái xác minh email
is_phone_verified	Boolean	Trạng thái xác minh số điện thoại
telegram_id	Varchar	Định danh tài khoản Telegram
custom_commands	JSON	Lưu trữ các lệnh tùy chỉnh người dùng
last_login	Timestamp	Thời gian đăng nhập gần nhất
date_joined	Timestamp	Ngày tạo tài khoản

Bảng **user_groups** là bảng trung gian liên kết người dùng tùy chỉnh với nhóm quyền, tương tự auth_user_groups nhưng tham chiếu tới model user của hệ thống

Bảng 2.29. Mô tả dữ liệu bảng user_groups

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh bản ghi
user_id	Int (FK)	Liên kết tới bảng user
group_id	Int (FK)	Liên kết tới bảng auth_group

Bảng **user_user_permissions** lưu quan hệ cấp quyền trực tiếp giữa người dùng tùy chỉnh và các quyền hạn cụ thể trong hệ thống

Bảng 2.30. Mô tả dữ liệu bảng *user_permission*

Tên trường	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh bản ghi
user_id	Int (FK)	Liên kết tới bảng user
permission_id	Int (FK)	Liên kết tới bảng auth_permission

Bảng **user_session** quản lý phiên đăng nhập theo cơ chế JWT refresh token, cho phép theo dõi thiết bị, địa chỉ IP và thu hồi phiên khi cần thiết, đảm bảo an toàn cho hệ thống.

Bảng 2.31. Mô tả dữ liệu bảng *user_session*

Tên bảng	Kiểu dữ liệu	Mô tả
session_id	UUID (PK)	Mã định danh phiên làm việc
user_id	Int (FK)	Liên kết tới bảng user
refresh_token	Varchar	Token dùng để làm mới phiên đăng nhập
ip_address	Int	Địa chỉ IP đăng nhập
user_agent	Text	Thông tin trình duyệt và thiết bị
is_revoked	Boolean	Trạng thái thu hồi phiên
created_at	Timestamp	Thời gian tạo phiên
expired_at	Timestamp	Thời điểm hết hạn phiên
update_at	timestamp	Thời gian cập nhật gần nhất

Bảng **lesson** lưu trữ thông tin các bài học trong hệ thống, liên kết với người học và tham chiếu tới mã nguồn minh họa tương ứng.

Bảng 2.32. Mô tả dữ liệu bảng *lesson*

Tên bảng	Kiểu dữ liệu	Mô tả
id	Int (PK)	Mã định danh bài học
title	Varchar	Tiêu đề bài học
student_at	Int (FK)	Liên kết tới người học (bảng user)
code_ref	Varchar (FK)	Tham chiếu tới mã nguồn (bảng source_code)
created_at	Timestamp	Thời gian tạo bài học
update_at	Timestamp	Thời gian cập nhật gần nhất

Bảng **source_code** lưu trữ nội dung mã nguồn Python đã được nén, sử dụng giá trị hash làm khóa chính nhằm tránh lưu trùng lặp khi nhiều bài học có cùng đoạn mã.

Bảng 2.33. Mô tả dữ liệu bảng source code

Tên trường	Kiểu dữ liệu	Mô tả
hash_id	Varchar (PK)	Mã hash định danh nội dung mã nguồn
compressed_content	Bytea	Nội dung mã nguồn đã được nén

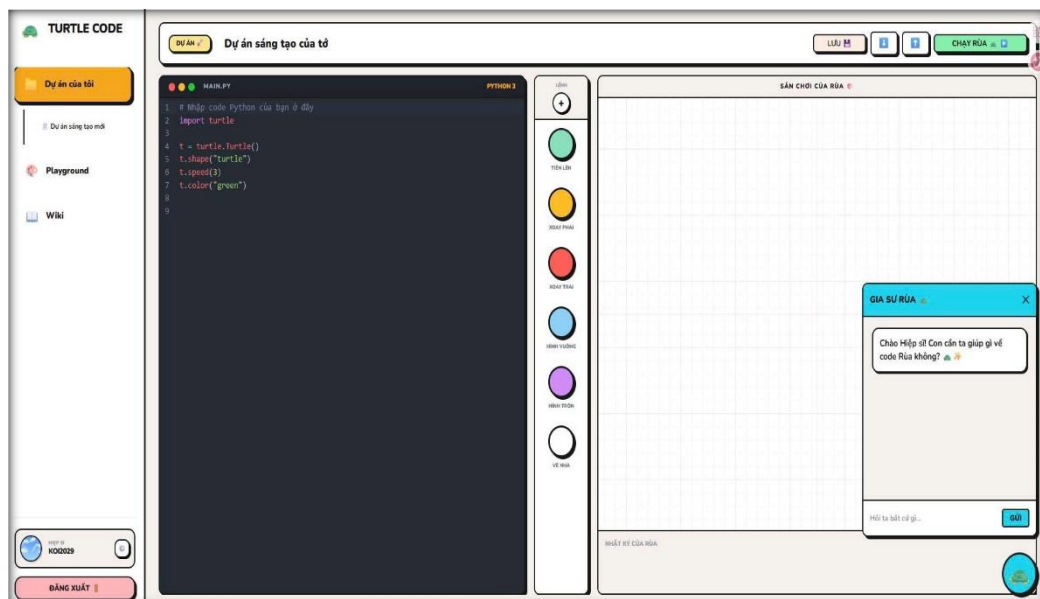
2.9. Thiết kế giao diện người dùng (Mockup)

2.9.1. Giao diện đăng nhập

The image shows two side-by-side login interface mockups. Both feature a 'QUAY LẠI' (Go Back) button at the top left. The left mockup, titled 'BẮT ĐẦU HÀNH TRÌNH', has a 'TÊN ĐĂNG NHẬP' (Login Name) field with 'thu@gmail.com', a 'MẬT KHẨU' (Password) field with masked characters, a 'XÁC NHẬN MẬT KHẨU' (Confirm Password) field, and a large orange 'XÁC NHẬN' (Confirm) button. Below is a 'HOẶC' (Or) separator and two buttons for 'G Google' and 'f Facebook'. At the bottom, it says 'Đã có tài khoản? Đăng nhập' (Already have an account? Log in). The right mockup, titled 'MỪNG CON TRỞ LẠI!', has a 'TÊN ĐĂNG NHẬP' field with 'koi2029@', a 'MẬT KHẨU' field with a 'QUÊN MẬT KHẨU?' link, and a large orange 'XÁC NHẬN' button. It also features the same social login buttons and a bottom link 'Chưa có tài khoản? Đăng ký ngay' (Don't have an account? Sign up now).

Hình 2.37. Giao diện đăng ký, đăng nhập

2.9.2. Giao diện màn hình chính khi đã đăng nhập



Hình 2.38. Giao diện màn hình chính khi đã đăng nhập

2.9.3. Giao diện cập nhật thông tin tài khoản

The image displays two screenshots of a mobile application interface for updating account information. Both screenshots feature a header with a 'QUAY LẠI' (Back) button and the title 'HỒ SƠ HIỆP SĨ' (Warrior Profile).

Top Screenshot: PERSONAL INFORMATION (THÔNG TIN CÁ NHÂN)

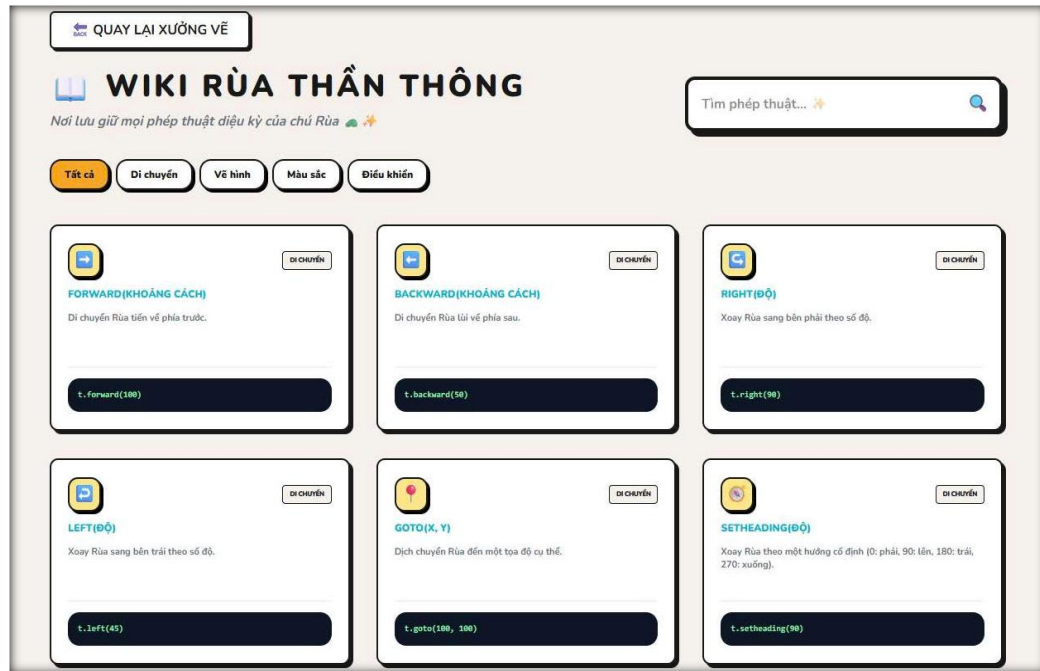
- Profile Card:** Displays a circular profile picture, the username 'KOI2029', the email '@KOI2029@', and the login date 'NGÀY GIA NHẬP 29/4/2026'.
- Form Fields:**
 - TÊN HIỂN THỊ (Display Name): Koi2029
 - NGÀY SINH (Date of Birth): 01/04/2026
 - SỐ ĐIỆN THOẠI (Phone Number): 09xx xxx xxx
 - EMAIL: koi2029@
- Buttons:** A yellow 'LƯU THAY ĐỔI' (Save Changes) button and a white 'BẢO MẬT TÀI KHOẢN' (Account Security) button with a plus icon.

Bottom Screenshot: ACCOUNT SECURITY (BẢO MẬT TÀI KHOẢN)

- Form Fields:**
 - TÊN ĐĂNG NHẬP (Login Name): koi2029@
 - THAY ĐỔI MẬT KHẨU (Change Password):
 - Current password field (masked with dots).
 - New password field: Mật khẩu mới (ít nhất 8 ký tự) (New password (at least 8 characters)).
 - Confirm password field: Xác nhận mật khẩu mới (Confirm new password).
- Buttons:** A pink 'CẬP NHẬT BẢO MẬT' (Update Security) button.

Hình 2.39. Giao diện cập nhật thông tin tài khoản

2.9.4. Giao diện tra cứu nội dung Wiki cập nhật thông tin tài khoản



Hình 2.40. Giao diện tra cứu nội dung Wiki

CHƯƠNG 3. CÀI ĐẶT VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. Triển khai và điều kiện sử dụng

3.1.1. Người dùng muốn học trên hệ thống (học sinh, người học thử)

Do hệ thống được xây dựng theo mô hình ứng dụng web, người học không cần cài đặt bất kỳ phần mềm lập trình nào trên máy cá nhân. Tuy nhiên, để đảm bảo trải nghiệm học tập ổn định và mượt mà, người dùng cần đáp ứng một số điều kiện tối thiểu về phần cứng và phần mềm như sau.

a) Yêu cầu phần cứng (phía người học – client)

Bộ xử lý (CPU): Tối thiểu bộ xử lý 64-bit, tốc độ ≥ 1.6 GHz (tương đương Intel Core i3 thế hệ 6 trở lên hoặc chip ARM trên tablet đời mới). Yêu cầu này đủ để trình duyệt xử lý giao diện React và thực thi mã Skulpt (Python chạy trên trình duyệt) một cách trơn tru.

Bộ nhớ RAM: Tối thiểu 4 GB; khuyến nghị 8 GB nếu người học mở nhiều tab hoặc chia màn hình song song giữa khung soạn mã và khung xem kết quả vẽ (preview).

Màn hình: Độ phân giải tối thiểu 1366×768 pixel; khuyến nghị 1920×1080 pixel để hiển thị đồng thời vùng soạn thảo mã và vùng xem hình vẽ mà không cần cuộn ngang, cải thiện trải nghiệm kéo-thả lệnh.

Thiết bị nhập liệu: Bàn phím và chuột (đối với laptop/PC). Trường hợp dùng máy tính bảng, nên có bàn phím ngoài để thuận tiện khi nhập lệnh lập trình dạng văn bản.

Kết nối mạng: Đường truyền Internet ổn định, băng thông tối thiểu 5 Mbps. Độ trễ thấp giúp đảm bảo việc gọi API thực thi mã, tải bài học và đăng nhập diễn ra nhanh chóng, không bị gián đoạn trong quá trình học.

b) Yêu cầu phần mềm (phía người học)

Hệ điều hành: Hệ thống tương thích với mọi hệ điều hành phổ biến hiện nay, bao gồm Windows 10/11, macOS 12 (Monterey) trở lên, các bản phân phối Linux desktop (Ubuntu, Fedora...), Android 10 trở lên và iPadOS 15 trở lên, miễn là thiết bị hỗ trợ trình duyệt web hiện đại.

Trình duyệt web: Đây là phần mềm duy nhất bắt buộc phía người học. Khuyến nghị sử dụng Google Chrome, Microsoft Edge hoặc Mozilla Firefox phiên bản phát hành trong vòng 12–24 tháng gần nhất. Trình duyệt cần bật JavaScript và cho phép cookie từ tên miền của hệ thống để chức năng đăng nhập và lưu phiên hoạt động đúng. Không sử dụng chế độ “Duyệt web riêng tư” (Incognito) nếu cần lưu tiến độ học.

Không yêu cầu cài đặt thêm: Người học không cần cài Python, Visual Studio Code, PyCharm, thư viện Turtle hay bất kỳ công cụ lập trình cục bộ nào. Toàn bộ quá trình biên dịch và thực thi mã Python (thông qua Skulpt) diễn ra trực tiếp trên trình duyệt hoặc phía máy chủ, giúp loại bỏ rào cản kỹ thuật cho người học và giáo viên.

c) Yêu cầu về môi trường thực thi chương trình

Hệ thống sử dụng Skulpt — một thư viện JavaScript cho phép thực thi mã Python trực tiếp trên trình duyệt mà không cần kết nối lại máy chủ mỗi lần chạy. Khi người học nhấn nút “Chạy” (Run), mã Python được Skulpt phân tích cú pháp và thực thi ngay trong ngữ cảnh trình duyệt, kết quả vẽ Turtle hiển thị tức thời trong vùng preview trên giao diện. Cơ chế này giúp giảm tải cho máy chủ ở các thao tác chạy thử thông thường, đồng thời đảm bảo phản hồi nhanh cho người học.

Đối với các chức năng yêu cầu tương tác với dữ liệu tài khoản (đăng nhập, lưu bài vẽ, tải bài học...), trình duyệt giao tiếp với máy chủ Backend thông qua giao thức HTTP/HTTPS. Vì vậy, điều kiện kết nối mạng ổn định là yêu cầu bắt buộc để các tính năng này hoạt động đúng. Trong môi trường triển khai nội bộ (intranet của trường), người học cần kết nối đúng mạng nội bộ hoặc VPN theo hướng dẫn của đơn vị quản lý.

Khi hệ thống đã được nhà trường / nhóm phát triển triển khai lên máy chủ và cung cấp địa chỉ truy cập (URL), người học không cần cài Python, IDE hay thư viện Turtle trên máy cá nhân. Toàn bộ xử lý logic và lưu trữ dữ liệu thực hiện trên máy chủ, trình duyệt chỉ đóng vai trò hiển thị giao diện và gửi yêu cầu đến API.

Các bước thực hiện (luồng sử dụng tối thiểu):

1. Chuẩn bị: Máy tính bảng, laptop hoặc máy tính trong phòng máy; kết nối Internet ổn định; trình duyệt web cập nhật (khuyến nghị Chrome, Edge hoặc Firefox bản mới).
2. Mở đường link do giáo viên hoặc quản trị cấp (ví dụ dạng <https://...> hoặc trong môi trường thử nghiệm là <http://địa-chi-máy-chủ:...>).
3. Đăng ký tài khoản (nếu lần đầu) hoặc đăng nhập bằng tài khoản đã có.
4. Vào khu vực học tập / bài học: đọc hướng dẫn, mở không gian lập trình (soạn mã Python dùng thư viện Turtle theo quy ước của hệ thống), chạy thử và quan sát khung xem kết quả vẽ (preview).
5. Lưu bài / xem lại bài vẽ (nếu hệ thống bật chức năng quản lý bài vẽ theo tài khoản).

Lưu ý: Một số chức năng phụ (ví dụ gửi mã OTP qua email, xử lý nền) phụ thuộc cấu hình máy chủ; nếu giáo viên thông báo “chỉ dùng trong mạng trường”, người học cần kết nối đúng mạng nội bộ hoặc VPN theo hướng dẫn của đơn vị.

3.1.2. Phần cứng và môi trường sử dụng phía người học

Phần cứng (máy người học - client):

Bảng 3.1. Một số yêu cầu phần cứng ở máy người học

Thành phần	Khuyến nghị tối thiểu	Ghi chú
Bộ xử lý (CPU)	Bộ xử lý 64-bit, đời tương đương Intel Core i3 / AMD tương đương trở lên (hoặc chip ARM trên tablet đời mới)	Đủ cho trình duyệt và tab học tập
Bộ nhớ RAM	≥ 4 GB (nên 8 GB nếu mở nhiều tab hoặc chia màn hình)	Giảm giật lag khi vẽ preview
Ổ đĩa	Không bắt buộc dung lượng lớn cho phần mềm Turtle local	Chủ yếu dùng bộ nhớ cho trình duyệt
Màn hình	Độ phân giải $\geq 1366 \times 768$; nên $\geq 1920 \times 1080$ để vừa khung soạn mã vừa khung xem hình	Cải thiện trải nghiệm kéo – thả / đọc bài
Thiết bị nhập	Bàn phím + chuột (laptop/PC); tablet nên có bàn phím ngoài nếu gõ nhiều	Học lập trình văn bản thuận hơn với bàn phím
Mạng	Băng thông ổn định, độ trễ thấp	Ảnh hưởng tải bài, đăng nhập, gọi API thực thi mã

Phần mềm phía người học:

- Hệ điều hành: Windows 10/11, macOS, Linux desktop, hoặc Android/iPadOS (trình duyệt hỗ trợ đầy đủ).
- Trình duyệt: phiên bản mới trong vòng ~12–24 tháng; bật JavaScript; nên tắt chặn cookie của site nếu đăng nhập gặp lỗi (theo chính sách trường).
- Không bắt buộc: Python, Visual Studio Code, PyCharm – chỉ cần khi tự chạy mã ngoài hệ thống, không thuộc luồng học chuẩn trên web.

3.1.3. Triển khai phía máy chủ/phòng thí nghiệm (quản trị viên)

Phần này mô tả cài đặt để hệ thống chạy được, phục vụ thử nghiệm hoặc triển khai nội bộ. Kiến trúc tham chiếu đề tài: tách Frontend (React + Vite) và Backend (Django REST + Daphne/ASGI), cơ sở dữ liệu PostgreSQL, bộ nhớ đệm/hàng đợi Redis, tác vụ

nền Celery (worker và beat), lưu file đối tượng qua S3 tương thích (ví dụ Supabase Storage) theo cấu hình dự án.

a) Yêu cầu phần cứng máy chủ phát triển

Bảng 3.2. Một số yêu cầu phần cứng ở máy người chủ

Thành phần	Mức tối thiểu	Mức khuyến nghị
CPU	4 lõi vật lý	8 lõi trở lên
RAM	8 GB	16 GB trở lên (PostgreSQL + Redis + Celery + Django)
Ổ cứng	≥ 20 GB trống (mã nguồn, DB, log, Docker image)	SSD ≥ 100 GB
Mạng	Cổng mở nội bộ theo cấu hình (ví dụ HTTP 8000, frontend dev 5173)	HTTPS qua reverse proxy (Nginx/Caddy), tường lửa cấu hình đúng

b) Yêu cầu phần mềm máy chủ / máy phát triển

- Python phiên bản 3.12 (theo pyproject.toml: $\geq 3.12, < 4.0$).
- Poetry (quản lý gói Python cho thư mục Back-end).
- Node.js (khuyến nghị LTS, ví dụ 20.x trở lên) và npm cho thư mục Front-end.
- PostgreSQL (phiên bản tương thích với psycopg2-binary, có thể dùng image PostgreSQL 17 trong kho docker/docker_db nếu triển khai bằng container).
- Redis (ví dụ 7.x, cổng mặc định 6379 — trùng với docker-compose mẫu).
- Docker / Docker Compose (tùy chọn nhưng được dự án hỗ trợ qua Back-end/docker-compose.yml).

c) Biến môi trường (file .env tại thư mục Back-end)

Các tham số điển hình cần khai báo (tên biến minh họa, không ghi giá trị bí mật vào báo cáo): DJANGO_SECRET_KEY, DJANGO_DEBUG, POSTGRES_DB, POSTGRES_USERNAME, POSTGRES_PASSWORD, POSTGRES_HOST, POSTGRES_PORT, các biến Supabase S3 (SUPABASE_S3_ACCESS_KEY, SUPABASE_S3_SECRET_KEY, ...) theo settings.py. Khi chạy Celery, cần đảm bảo worker và beat dùng chung cấu hình broker/backend với Redis.

d) Cài đặt Backend (không Docker – mô tả từng bước)

Bước 1. Clone mã nguồn, vào thư mục Back-end.

Bước 2. Cài dependency: poetry install (hoặc môi trường tương đương theo hướng dẫn nhóm).

Bước 3. Tạo và chỉnh file .env đầy đủ biến kết nối PostgreSQL và các dịch vụ ngoài.

- Bước 4. Khởi tạo CSDL: trong thư mục chứa `manage.py` (`turtle_project`), chạy `migrate`:
- Bước 5. `python manage.py migrate`
- Bước 6. Khởi chạy Redis (local hoặc container). Trong `settings.py`, cache Redis mặc định trở `redis://127.0.0.1:6379/1` khi chạy local — cần khớp với môi trường thực tế.
- Bước 7. Khởi chạy Celery worker và Celery beat (nếu dùng OTP email, tác vụ định kỳ...).
- Bước 8. Khởi chạy ASGI server: ví dụ `daphne -b 0.0.0.0 -p 8000 core.asgi:application` (thư mục làm việc là `turtle_project`), tương ứng với Dockerfile mẫu.

e) Cài đặt Frontend

- Vào thư mục Front-end, chạy `npm install`.
- Chạy môi trường phát triển: `npm run dev` (Vite, cổng mặc định thường 5173).
- Trong mã nguồn hiện tại, địa chỉ API mặc định khai báo tại `src/utlis/constants.js` là `http://localhost:5173/api` (và WebSocket `ws://localhost:5173/ws`). Khi triển khai thực tế, cần đổi sang URL máy chủ và dùng HTTPS/WSS phù hợp, đồng thời cấu hình CORS trên Django cho đúng miền frontend.

f) Triển khai bằng Docker Compose (thư mục Back-end)

File `docker-compose.yml` mẫu định nghĩa dịch vụ Redis, web (build từ `docker/docker_env/dockerfile`, mở cổng 8000, chạy Daphne), `celery_worker`, `celery_beat`, nạp biến từ `.env`. Người triển khai cần bổ sung/đồng bộ dịch vụ PostgreSQL (container riêng hoặc CSDL managed) cho khớp biến `POSTGRES_*` trong `.env`, rồi chạy `migrate` trước khi phục vụ người dùng.

3.1.4. Tổng hợp yêu cầu thực thi của các đối tượng sử dụng”

Dưới đây là bảng tổng hợp chi tiết về trách nhiệm và yêu cầu kỹ thuật đối với từng nhóm đối tượng khi tiếp cận và vận hành hệ thống

Bảng 3.3. So sánh trách nhiệm và yêu cầu cài đặt phần mềm theo vai trò

Đối tượng	Việc phải làm	Cài đặt phần mềm nặng
Học sinh / người học	Mở link, đăng ký/đăng nhập, học trên trình duyệt	Không (khi hệ thống đã online)
Giáo viên	Cung cấp URL, hướng dẫn tài khoản, tổ chức buổi học	Không (trừ khi tự host)

Đối tượng	Việc phải làm	Cài đặt phần mềm nặng
Quản trị viên	Cài stack Backend + DB + Redis + Celery + build Frontend, cấu hình .env, bảo mật, HTTPS	Có

3.2. Đánh giá kết quả

3.2.1. Đánh giá về chức năng

Đánh giá chức năng được tiến hành theo hướng kiểm thử hộp đen: lập danh sách tình huống theo từng nhóm nghiệp vụ, thực hiện trên trình duyệt (môi trường người dùng cuối), đối chiếu với đặc tả use case đã phân tích ở Chương 2. Kết quả tổng hợp như sau (mô tả theo phạm vi mã nguồn hiện có):

(1) Xác thực và tài khoản: Trang /auth hỗ trợ luồng đăng ký / đăng nhập tương tác với API backend; token JWT (SimpleJWT) dùng cho các thao tác cần xác thực. Trang /profile được bảo vệ bởi ProtectedRoute, cho phép cập nhật thông tin cá nhân (trong đó có đồng bộ nút lệnh tùy chỉnh custom_commands xuống server khi người dùng đã đăng nhập).

(2) Không gian làm việc (Workspace) và thực hành Turtle: Trang / đóng vai trò IDE web: soạn mã Python với CodeMirror, bảng lệnh gợi ý / kéo-thả mẫu (palette), kích hoạt chạy thử cập nhật vùng Preview. Việc thực thi mã và vẽ Turtle được thực hiện trên trình duyệt nhờ Skulpt (Python tập con chạy bằng JavaScript), không đòi hỏi cài Python trên máy học sinh và giảm tải cho máy chủ ở thao tác “Run” thuần preview. Người dùng chưa đăng nhập vẫn có thể dùng Playground (lưu nháp cục bộ localStorage); người đã đăng nhập có luồng dự án với cơ chế nháp / cảnh báo khi thoát và tích hợp gọi API bài học / dự án (lessonApi, authApi) theo thiết kế trang.

(3) Tra cứu nội dung (Wiki): Đường dẫn /wiki cho phép tra cứu tài liệu tham khảo, hỗ trợ người học tra cứu cú pháp và khái niệm trong quá trình thực hành.

(4) API bài học (Backend): Ứng dụng lesson cung cấp ViewSet qua DefaultRouter (chuẩn REST: danh sách/chi tiết bài học theo endpoint /api/.../lessons/), phục vụ frontend tải nội dung học tập và quản lý dữ liệu bài học phía server.

(5) Tác vụ nền và real-time (theo cấu hình dự án): Redis + Celery phục vụ các tác vụ bất đồng bộ (như gửi email/OTP, dọn dữ liệu theo lịch trong settings.py). Channels + WebSocket được cấu hình trong ASGI phục vụ các kịch bản tương tác thời gian thực (ví dụ điều khiển phiên đăng nhập) trong phạm vi triển khai.

Nhận xét tổng thể: Các nhóm chức năng cốt lõi phục vụ luồng “mở link → học → chạy thử → (đăng nhập) lưu / quản lý” hoạt động thống nhất với định hướng không cài đặt môi trường lập trình trên máy trẻ. Các chức năng phụ thuộc dịch vụ ngoài (email, Supabase S3 cho avatar/lưu trữ đối tượng...) cần cấu hình .env đầy đủ mới đạt trạng thái “đủ tính năng” trên môi trường thật.

3.2.2. *Đánh giá về phi chức năng và chất lượng kỹ thuật*

Hiệu năng và khả năng mở rộng: Thực thi Skulpt tận dụng CPU phía client, phù hợp lớp đông người học đồng thời ở mức “chạy thử cá nhân”; tuy nhiên thiết bị yếu hoặc trình duyệt cũ có thể gây trễ khi vòng lặp lớn / vẽ phức tạp. Backend tập trung cho API, xác thực, CSDL nên dễ nhân rộng theo chiều ngang khi có reverse proxy, connection pool, cache Redis hợp lý.

Bảo mật: JWT, CORS, CSRF (theo cấu hình Django) tạo nền tảng bảo vệ API; với mô hình chạy mã Skulpt trên trình duyệt, rủi ro “thi hành mã tùy ý trên server” được hạn chế so với mô hình exec() Python phía backend, nhưng vẫn cần giáo dục người dùng và kiểm soát nội dung nhúng (XSS, tải script) theo chuẩn triển khai web.

Khả năng sử dụng (Usability): Giao diện workspace tách vùng soạn thảo và vùng xem hình, có thông báo lỗi thân thiện khi Skulpt báo lỗi cú pháp/lỗi chạy; phù hợp lứa tuổi nhỏ khi có hướng dẫn của giáo viên lần đầu làm quen.

Khả năng bảo trì: Tách Front-end (React + Vite) và Back-end (Django) giúp nhóm phát triển độc lập từng phần, cập nhật API và giao diện theo phiên bản.

3.2.3. *Ưu nhược điểm*

• Dự án sở hữu nhiều ưu điểm nổi bật, trước hết là:

Mức độ tiếp cận học tập thấp khi người học chỉ cần trình duyệt và đường link mà không phải cài đặt Python hay IDE, qua đó giảm thiểu rào cản công nghệ ban đầu.

Khả năng phản hồi trực quan nhanh nhờ Turtle chạy trực tiếp trên client với Skulpt giúp học sinh dễ dàng quan sát mối liên hệ giữa lệnh và hình ảnh, tạo trải nghiệm học tập sinh động.

Kiến trúc hệ thống hiện đại, sử dụng Django, DRF, PostgreSQL, Redis/Celery và ASGI, kết hợp với frontend React được module hóa theo trang, đảm bảo khả năng mở rộng và duy trì lâu dài.

Tính linh hoạt cho người dùng cũng được chú trọng, với Playground dành cho khách cho phép “thử ngay” trước khi đăng ký, cùng cơ chế hỗ trợ cá nhân hóa nhẹ thông qua nút lệnh tùy chỉnh đồng bộ theo tài khoản.

- Tuy nhiên, dự án cũng tồn tại những hạn chế nhất định.

Skulpt chỉ là tập con của Python nên một số thư viện hoặc hành vi chưa tương đương hoàn toàn với CPython + Turtle trên máy trạm, khiến giáo viên cần lựa chọn bài tập trong phạm vi tương thích.

Hiệu năng còn phụ thuộc vào chất lượng thiết bị client, đặc biệt khi chạy các đoạn mã nặng.

Việc triển khai đầy đủ các chức năng như email OTP, lưu avatar trên S3 hay WebSocket phụ thuộc nhiều vào cấu hình môi trường và vận hành, dễ phát sinh lỗi nếu thiếu tài liệu nội bộ. Ngoài ra, đánh giá định lượng về số lượng học sinh và thang đo chuẩn trong báo cáo mới chỉ ở mức đề xuất, cần được bổ sung bằng số liệu khảo sát thực tế để tăng độ tin cậy và tính đại diện.

1. Tóm tắt kết quả đạt được

Đề tài “Xây dựng hệ thống hỗ trợ trẻ em học lập trình vẽ đối tượng với thư viện Turtle của Python” đã hoàn thành việc nghiên cứu lý thuyết nền tảng về giáo dục lập trình trực quan, đặc điểm người dùng nhỏ tuổi và vai trò của Turtle trong việc giảm độ trừu tượng của lập trình văn bản. Đồng thời phân tích, thiết kế và hiện thực hóa một ứng dụng web với kiến trúc tách lớp phù hợp thực tiễn triển khai.

Về mặt kỹ thuật, nhóm đã xây dựng backend trên nền Python – Django, kết hợp Django REST Framework cho các API phục vụ đăng ký, đăng nhập, quản lý người dùng, bài học, mã nguồn và sản phẩm thực hành; sử dụng PostgreSQL làm cơ sở dữ liệu, Redis cho bộ nhớ đệm và hạ tầng hàng đợi, Celery cho tác vụ nền (theo thiết kế hệ thống), và Daphne/ASGI để phục vụ ứng dụng đồng thời hướng tới khả năng mở rộng. Frontend được xây dựng bằng React và Vite, cung cấp giao diện thân thiện, hỗ trợ soạn thảo mã, thực thi và quan sát kết quả vẽ trong môi trường trình duyệt. Nhờ hình thức web, trẻ em và người học chỉ cần truy cập bằng đường link, không phải cài đặt Python hay IDE trên máy cá nhân, giúp hạ thấp rào cản triển khai trong lớp học và tại nhà.

Về mặt chức năng và giáo dục, hệ thống tích hợp các hoạt động học – thực hành xoay quanh Turtle, có kho bài học/bài tập và cơ chế lưu – quản lý sản phẩm theo tài khoản (trong phạm vi đã thiết kế), hướng tới mục tiêu phát triển tư duy logic, sáng tạo và kỹ năng giải quyết vấn đề. Công trình còn thể hiện quy trình phân tích yêu cầu, mô hình hóa UML, triển khai và kiểm thử, đáp ứng yêu cầu một đồ án NCKH có tính ứng dụng và tái lập cao.

2. Hạn chế của đề tài

Mặc dù đã đạt được những kết quả đáng khích lệ, đề tài vẫn còn tồn tại một số hạn chế nhất định do giới hạn về thời gian và nguồn lực thực hiện:

Thứ nhất, phạm vi kiến thức lập trình trên nền Turtle được giới hạn ở mức nhập môn – trung bình (điều khiển rùa, hình học cơ bản, lặp, rẽ nhánh, hàm đơn giản...), chưa bao quát toàn bộ nội dung Python hay lập trình hướng đối tượng nâng cao.

Thứ hai, chất lượng trải nghiệm phụ thuộc nhiều vào hạ tầng mạng, cấu hình máy chủ và số người truy cập đồng thời; khi tải cao có thể ảnh hưởng đến độ trễ thực thi và phản hồi giao diện, cần tối ưu và chiến lược triển khai riêng khi mở rộng quy mô.

Thứ ba, đánh giá hiệu quả giáo dục mới dừng ở mức thử nghiệm trong phạm vi nhóm (kiểm thử chức năng, khảo sát/đánh giá trải nghiệm với cỡ mẫu hạn chế), chưa có thiết kế thực nghiệm sư phạm quy mô lớn, kéo dài nhiều tháng trên nhiều cơ sở giáo dục để suy ra kết luận mang tính đại diện toàn quốc.

Thứ tư, hệ thống phụ thuộc dịch vụ ngoài (như lưu trữ đối tượng trên kho S3 tương thích, email OTP...) nên khi thiếu cấu hình hoặc gián đoạn dịch vụ, một số chức năng phụ có thể bị ảnh hưởng; đồng thời yêu cầu an toàn khi cho phép người dùng chạy mã (kiểm soát tài nguyên, sandbox, giới hạn thư viện...) cần được củng cố liên tục trong các phiên bản sau.

Thứ năm, giao diện và nội dung vẫn có thể tinh chỉnh thêm cho khả năng tiếp cận (Accessibility), thiết bị màn nhỏ và đa ngôn ngữ (nếu mở rộng đối tượng), chưa khai thác hết các tiêu chuẩn thiết kế dành riêng cho trẻ khuyết tật hoặc lứa tuổi rất nhỏ.

3. Hướng phát triển trong tương lai

Dựa trên những kết quả đạt được và những hạn chế còn tồn tại, nhóm đề xuất các hướng phát triển và mở rộng hệ thống trong giai đoạn tiếp theo như sau: Về sản phẩm, cần tiếp tục hoàn thiện tối ưu hiệu năng (cache, hàng đợi, giới hạn tài nguyên khi thực thi mã), đồng thời tăng cường bảo mật và giám sát (rate limit, logging, phân quyền chi tiết), cũng như bổ sung bảng điều khiển dành cho giáo viên với các chức năng quản lý lớp học, bài tập và thống kê tiến độ, kèm theo cơ chế chấm điểm và phản hồi lỗi cú pháp thân thiện với trẻ. Về sư phạm, định hướng xây dựng lộ trình học cá nhân hóa (adaptive learning), áp dụng gamification với điểm số, huy hiệu và thử thách theo tuần, đồng thời phát triển ngân hàng bài tập phân tầng rõ ràng hơn kèm gợi ý từng bước (scaffolding). Về trải nghiệm người dùng, hệ thống sẽ được phát triển theo hướng PWA/tối ưu cho thiết bị di động, hỗ trợ chế độ lớp học (chia sẻ màn hình kết quả), cải thiện trợ năng và tích hợp hướng dẫn ngay trong không gian lập trình. Về nghiên cứu, nhóm dự kiến tổ chức thực nghiệm sư phạm có đối chứng với thang đo chuẩn hóa (mức độ hứng thú, khả năng hiểu khái niệm, thời gian hoàn thành bài), đồng thời công bố bộ dữ liệu và quy trình tái lập để cộng đồng có thể mở rộng nghiên cứu. Cuối cùng, về triển khai, hệ thống sẽ được hoàn thiện quy trình CI/CD, bảo mật HTTPS, cơ chế sao lưu – phục hồi cơ sở dữ liệu, và mô hình đa tenant cho phép nhiều trường/lớp cùng sử dụng trên một nền tảng, phục vụ mục tiêu phổ cập thực tế.

Nhìn lại toàn bộ quá trình thực hiện, đề tài “Xây dựng hệ thống hỗ trợ trẻ em học lập trình vẽ đối tượng với thư viện Turtle của Python” không chỉ là một sản phẩm phần mềm hoàn chỉnh mà còn là một hành trình học tập và trưởng thành thực sự của cả nhóm nghiên cứu. Thông qua quá trình này, nhóm đã vận dụng tổng hợp kiến thức từ phân tích thiết kế hệ thống, lập trình web hiện đại đến thiết kế trải nghiệm người dùng cho đối tượng đặc thù là trẻ em. Dù còn những hạn chế cần khắc phục, nhóm tin tưởng rằng sản phẩm đã đặt nền móng vững chắc cho một hướng đi có giá trị trong lĩnh vực giáo dục công nghệ cho thế hệ trẻ tại Việt Nam. Với những hướng phát triển đã được đề xuất, hệ thống có tiềm năng trở thành một công cụ giáo dục STEM thiết thực, góp phần thu hẹp khoảng cách giữa lập trình khối và lập trình văn bản, đồng thời khơi dậy niềm đam mê công nghệ cho trẻ em theo phương châm “học mà chơi, chơi mà học”.

TÀI LIỆU THAM KHẢO

- [1] S. Papert, *Indstorms: Children, Computers, and Powerful Ideas*, Basic Books, 1980.
- [2] M. M. Lab, "Scratch: Programming for All," *Communications of the ACM*, vol. 52, no. 11, 2009.
- [3] A. Wilson & C. Lowe, "Transitioning from Block-Based to Text-Based Programming," *Journal of Educational Computing Research*, vol. 58, no. 5, 2020.
- [4] J. M. Wing, "Computational Thinking," *Communications of the ACM*, vol. 39, no. 3, pp. 33-35, 2006.
- [5] Google, "Blockly: A Library for Building Visual Programming Editors," 2012. [Online]. Available: <https://developers.google.com/blockly?hl=vi>.
- [6] Code.org, "Annual Report 2023," Code.org Foundation, 2023. [Online].
- [7] Tynker, "Project-Based Learning in K-12 Computer Science," Tynker Education Research, 2021.
- [8] e. a. H. Liu, "Immediate visual feedback in introductory programming education," *IEEE Transactions on Education*, vol. 64, no. 2, 2021.
- [9] Meta, "React Documentation: A JavaScript library for building user interfaces," Reactjs.org, 2024. [Online]. Available: <https://react.dev/>.
- [10] D. S. Foundation, "Django documentation," Djangoproject.com, 2024. [Online]. Available: <https://docs.djangoproject.com/en/5.0/>.
- [11] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, vol. 12, no. 2, 1988.
- [12] e. a. H. Liu, "Visual Programming Languages for Children: A Systematic Review," *Computers & Education*, vol. 157, 2020.