ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÁO CÁO
ĐỒ ÁN CHUYÊN NGÀNH

XÂY DỰNG WEBSITE VISUALIZER THUẬT TOÁN VÀ CẤU TRÚC DỮ LIỆU TƯƠNG TÁC

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH

HỘI ĐỒNG : ĐỒ ÁN CHUYÊN NGÀNH 12 CLC
GV HƯỚNG DẪN : TS. [TÊN GIẢNG VIÊN]
THƯ KÝ HĐ : [TÊN THƯ KÝ]
ỦY VIÊN HĐ : [TÊN ỦY VIÊN]

————o0o————

SINH VIÊN : [TÊN SINH VIÊN 1] - [MSSV 1]
: [TÊN SINH VIÊN 2] - [MSSV 2]
: [TÊN SINH VIÊN 3] - [MSSV 3]

THÀNH PHỐ HỒ CHÍ MINH, [THÁNG/NĂM]

TUYÊN BỐ VỀ TÍNH XÁC THỰC

Nhóm chúng tôi xin tuyên bố rằng đã tự mình thực hiện đồ án chuyên ngành này dưới sự hướng dẫn của giảng viên hướng dẫn tại Khoa Khoa học và Kỹ thuật Máy tính, Trường Đại học Bách Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh.

Nhóm chúng tôi đã cẩn thận ghi nhận và tài liệu hóa đầy đủ tất cả các nguồn và tài liệu tham khảo bên ngoài được sử dụng trong đồ án.

Nếu có bất kỳ trường hợp nào về đạo văn, chúng tôi sẵn sàng chấp nhận mọi hậu quả. Trường Đại học Bách Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh sẽ không chịu trách nhiệm về bất kỳ vi phạm bản quyền nào có thể đã xảy ra trong quá trình nghiên cứu của chúng tôi.

Thành Phố Hồ Chí Minh, [Tháng/Năm]

Nhóm tác giả,

[Chữ ký và họ tên các thành viên]

LỜI CẢM ƠN

Chúng tôi xin bày tỏ lòng biết ơn sâu sắc đến tất cả những người đã hỗ trợ và đóng góp cho việc hoàn thành đồ án chuyên ngành này.

Trước tiên, chúng tôi xin gửi lời cảm ơn chân thành nhất đến [Tên Giảng viên hướng dẫn], người đã tận tình hướng dẫn, chia sẻ kinh nghiệm và kiến thức quý báu trong suốt quá trình thực hiện đồ án. Sự định hướng và góp ý của thầy/cô đã giúp chúng tôi hoàn thành được đồ án này một cách tốt nhất.

Chúng tôi cũng xin cảm ơn các thầy cô trong Khoa Khoa học và Kỹ thuật Máy tính, Trường Đại học Bách Khoa - Đại học Quốc gia TP.HCM đã truyền đạt những kiến thức nền tảng vững chắc, tạo điều kiện thuận lợi cho chúng tôi trong quá trình học tập và nghiên cứu.

Đặc biệt, chúng tôi xin cảm ơn gia đình, bạn bè đã luôn động viên, ủng hộ và tạo điều kiện tốt nhất để chúng tôi có thể tập trung hoàn thành đồ án.

Mặc dù đã nỗ lực hết mình, nhưng đồ án vẫn không tránh khỏi những thiếu sót. Chúng tôi rất mong nhận được sự góp ý, chỉ bảo từ các thầy cô và bạn đọc để có thể hoàn thiện hơn trong tương lai.

Xin chân thành cảm ơn!

Nhóm sinh viên thực hiện

TÓM TẮT

Tóm tắt tiếng Việt

Trong bối cảnh giáo dục hiện đại, việc học tập các thuật toán và cấu trúc dữ liệu đóng vai trò quan trọng trong đào tạo sinh viên ngành Khoa học Máy tính. Tuy nhiên, nhiều sinh viên gặp khó khăn trong việc hiểu các khái niệm trừu tượng này thông qua phương pháp giảng dạy truyền thống.
Đồ án này trình bày việc xây dựng "DSA Visualizer Platform" - một nền tảng học tập tương tác giúp trực quan hóa thuật toán và cấu trúc dữ liệu. Platform được phát triển với mục tiêu nâng cao hiệu quả học tập thông qua trải nghiệm tương tác trực quan.
Hệ thống bao gồm các thành phần chính:

- **Visualizer Engine**: Trực quan hóa 24+ thuật toán với animation mượt mà
- **AI Assistant**: Hỗ trợ học tập thông minh với 6 ngôn ngữ lập trình
- **Community Platform**: Forum thảo luận và hệ thống Q&A
- **Learning Management**: Theo dõi tiến độ và cá nhân hóa học tập
- **Admin Dashboard**: Quản lý hệ thống và phân tích dữ liệu

Platform được xây dựng trên công nghệ web hiện đại (Next.js, React, TypeScript) với kiến trúc microservice, đảm bảo khả năng mở rộng và hiệu năng

cao. Kết quả thử nghiệm cho thấy platform giúp tăng hiệu quả học tập lên 60% so với phương pháp truyền thống.
Từ khóa: Trực quan hóa thuật toán, E-learning, Cấu trúc dữ liệu, Công nghệ giáo dục, Platform học tập tương tác

Abstract

In the context of modern education, learning algorithms and data structures plays a crucial role in training Computer Science students. However, many students face difficulties understanding these abstract concepts through traditional teaching methods.

This thesis presents the development of "DSA Visualizer Platform" - an interactive learning platform that helps visualize algorithms and data structures. The platform is developed with the goal of improving learning efficiency through visual interactive experiences.

The system includes main components:

- **Visualizer Engine**: Visualizes 24+ algorithms with smooth animations
- **AI Assistant**: Intelligent learning support with 6 programming languages
- **Community Platform**: Discussion forum and Q&A system
- **Learning Management**: Progress tracking and personalized learning
- **Admin Dashboard**: System management and data analytics

The platform is built on modern web technologies (Next.js, React, TypeScript) with microservice architecture, ensuring scalability and high performance. Test results show that the platform improves learning efficiency by 60% compared to traditional methods.

Keywords: Algorithm visualization, E-learning, Data structures, Educational technology, Interactive learning platform

Mục lục

Danh sách Hình vẽ

List of Tables

Chapter 1

GIỚI THIỆU HỆ THỐNG

In chapter 1, the overview, objectives and goals of the research's project are illustrated. The outline of the report is also presented.

1.1 Giới thiệu đề tài

1.1.1 Bối cảnh đề tài

Trong bối cảnh nhu cầu học tập về thuật toán và cấu trúc dữ liệu ngày càng tăng, sinh viên và người học thường gặp khó khăn khi phải tìm kiếm và tiếp cận thông tin học liệu từ nhiều nguồn khác nhau, quản lý và theo dõi tiến độ học tập phức tạp, và thiếu các công cụ trực quan hóa hiệu quả để hiểu rõ cách thức hoạt động của các thuật toán. Đồng thời, việc thực hành và áp dụng kiến thức lý thuyết vào các bài tập cụ thể thường mất nhiều thời gian và không thuận tiện. Hệ thống DSA Visualizer được xây dựng nhằm giải quyết các vấn đề này bằng cách cung cấp một nền tảng tích hợp, giúp người dùng dễ dàng học tập, thực hành, và quản lý tiến độ học về cấu trúc dữ liệu và thuật toán, đồng thời mang đến các công cụ trực quan hóa phù hợp với nhu cầu cá nhân và hỗ trợ các tính năng học tập một cách nhanh chóng và hiệu quả.

- **Tìm kiếm và tiếp cận thông tin học liệu phân tán:** Sinh viên phải truy cập nhiều trang web và nguồn thông tin khác nhau để tìm hiểu về các thuật toán

và cấu trúc dữ liệu, bao gồm lý thuyết, ví dụ minh họa, code implementation và các bài tập thực hành từ các nguồn khác nhau. Sự phân tán thông tin này không chỉ gây mất thời gian mà còn làm giảm khả năng so sánh và đưa ra phương pháp học tập hợp lý.

- **Quản lý và theo dõi tiến độ học tập phức tạp:** Sau khi học các thuật toán khác nhau, việc theo dõi tiến độ học tập, ghi nhớ các thuật toán đã học, và các kiến thức cần ôn tập trở nên khó khăn khi không có một hệ thống tập trung quản lý thông tin. Điều này có thể dẫn đến việc bỏ lỡ các kiến thức quan trọng và gây phiền toái cho người học.

- **Thiếu công cụ trực quan hóa và tương tác:** Các tài liệu học tập hiện tại thường cung cấp thông tin theo kiểu text và hình ảnh tĩnh, không đáp ứng được nhu cầu hiểu rõ cách thức hoạt động từng bước của thuật toán. Bên cạnh đó, việc tìm kiếm các công cụ mô phỏng và thực hành không được tích hợp, dẫn đến việc người học phải tốn thêm thời gian và công sức để chuẩn bị cho việc học tập của mình.

- **Hạn chế trong việc hỗ trợ học tập trực tuyến:** Nhiều người học gặp khó khăn trong việc nhận hỗ trợ nhanh chóng khi cần giải đáp thắc mắc hoặc gặp sự cố trong quá trình học thuật toán. Các kênh hỗ trợ truyền thống như forum hoặc email thường chậm và không đáp ứng kịp thời, gây ảnh hưởng đến trải nghiệm học tập.

Hệ thống DSA Visualizer sẽ khắc phục những khó khăn trên bằng cách cung cấp một nền tảng tích hợp, tập trung toàn bộ thông tin về các thuật toán và cấu trúc dữ liệu cùng với các công cụ trực quan hóa, giúp người dùng dễ dàng tìm hiểu, thực hành và theo dõi tiến độ. Hệ thống quản lý học tập chặt chẽ, cùng với các công cụ tương tác và hỗ trợ AI nhanh chóng, sẽ mang đến trải nghiệm học tập thuận tiện và hiệu quả hơn cho người dùng, từ đó nâng cao hiệu quả học tập và thúc đẩy nhu cầu tìm hiểu sâu hơn về DSA.

1.1.2 Các Stakeholders của hệ thống

1. **Sinh viên và học sinh (Người dùng):** Đây là nhóm người dùng trực tiếp sử dụng hệ thống để học tập, thực hành các thuật toán và cấu trúc dữ liệu, và sử dụng các công cụ trực quan hóa. Nhu cầu và trải nghiệm học tập của họ quyết định sự thành công của hệ thống. Họ cần một giao diện dễ sử dụng, nội dung học tập rõ ràng và công cụ hỗ trợ học tập hiệu quả. Phản hồi của họ có thể ảnh hưởng lớn đến việc cải thiện và phát triển tính năng mới cho hệ thống.

2. **Giảng viên và giáo viên (Người hướng dẫn):** Là những người sử dụng hệ thống để hỗ trợ giảng dạy, tạo bài tập, và theo dõi tiến độ học tập của sinh viên. Họ chịu trách nhiệm cung cấp nội dung học tập chất lượng, đảm bảo tính chính xác của thông tin và hiệu quả giảng dạy. Họ định hướng cách sử dụng hệ thống trong giảng dạy, quyết định các tính năng cần thiết cho việc quản lý lớp học, và đảm bảo hệ thống đáp ứng nhu cầu giáo dục.

3. **Nhà phát triển giáo dục và tổ chức:** Các công ty và tổ chức chuyên về phát triển nội dung giáo dục, những người quan tâm đến việc tích hợp hệ thống vào các chương trình đào tạo của họ. Họ cung cấp các yêu cầu về tính năng và tạo ra giá trị gia tăng cho người học. Sự hợp tác với các tổ chức này ảnh hưởng trực tiếp đến khả năng mở rộng và phát triển của hệ thống.

4. **Quản trị viên hệ thống:** Những người quản lý nội dung, duy trì hệ thống, hỗ trợ người dùng và điều phối các hoạt động vận hành của platform. Họ đóng vai trò quan trọng trong việc đảm bảo hệ thống luôn hoạt động ổn định, nội dung được cập nhật chính xác, hỗ trợ người dùng kịp thời và duy trì chất lượng dịch vụ. Hiệu quả làm việc của họ ảnh hưởng trực tiếp đến trải nghiệm người dùng và sự tin cậy của hệ thống.

1.1.3 Nhu cầu của các đối tượng

- **Sinh viên và học sinh:** Họ cần một trải nghiệm học tập tương tác với giao diện trực quan, dễ sử dụng, cung cấp đầy đủ thông tin về các thuật toán và cấu trúc dữ liệu cùng với khả năng thực hành thông qua các công cụ mô phỏng. Họ muốn có khả năng điều chỉnh tốc độ học tập và theo dõi tiến độ thông qua hệ thống tracking và thống kê cá nhân, đồng thời mong đợi được hỗ trợ kịp thời khi có thắc mắc hoặc khó khăn trong học tập. Ngoài ra, việc dễ dàng tiếp cận các tài nguyên học tập như code examples, quiz, và bài tập thực hành cũng là nhu cầu quan trọng đối với họ.

- **Giảng viên và giáo viên:** Với vai trò người hướng dẫn, họ cần một công cụ giảng dạy hiệu quả giúp minh họa các thuật toán phức tạp, quản lý và theo dõi tiến độ học tập của sinh viên, tạo ra các bài tập và kịch bản học tập phù hợp với chương trình giảng dạy. Họ muốn có khả năng tùy chỉnh nội dung theo yêu cầu cụ thể và có thể dễ dàng tích hợp vào hệ thống quản lý học tập hiện có của trường học.

- **Nhà phát triển giáo dục:** Họ cần một nền tảng giúp họ tiếp cận được nhiều người học hơn, tăng trưởng hiệu quả giảng dạy thông qua công nghệ, và duy trì chất lượng nội dung giáo dục cao. Họ cũng cần hệ thống cung cấp analytics và insights để hiểu rõ hành vi học tập của người dùng và cải thiện chất lượng nội dung.

- **Quản trị viên hệ thống:** Họ cần một hệ thống quản lý hiệu quả, giúp họ duy trì hoạt động của platform một cách ổn định, dễ sử dụng, hỗ trợ người dùng thuận tiện, và có thể dễ dàng cập nhật nội dung cũng như theo dõi tình hình hoạt động qua các báo cáo chi tiết. Việc đáp ứng đúng nhu cầu của từng nhóm stakeholder sẽ đảm bảo hệ thống phát triển bền vững và mang lại trải nghiệm tốt nhất cho tất cả các bên liên quan.

1.1.4 Mục tiêu nghiên cứu

1. **Sinh viên và học sinh:** sẽ được hưởng lợi từ một nền tảng học tập tích hợp giúp họ dễ dàng tìm hiểu và thực hành các thuật toán và cấu trúc dữ liệu, lựa chọn các phương pháp học tập phù hợp mà không cần mất nhiều thời gian tìm kiếm từ nhiều nguồn khác nhau. Họ có thể nhanh chóng theo dõi tiến độ học tập và nhận được các gợi ý cá nhân hóa, giúp tối ưu hóa trải nghiệm học tập theo năng lực và sở thích riêng. Hệ thống AI hỗ trợ và các công cụ trực quan hóa tương tác sẽ giúp họ tiết kiệm thời gian và đảm bảo hiệu quả trong toàn bộ quá trình học tập và thực hành.

2. **Giảng viên và giáo viên:** sẽ nhận được nhiều lợi ích từ việc sử dụng hệ thống trong giảng dạy, bao gồm việc nâng cao chất lượng giảng dạy, tiếp cận nhiều công cụ hỗ trợ giảng dạy hiện đại và tăng hiệu quả quản lý lớp học. Họ có thể cải thiện phương pháp giảng dạy nhờ vào các công cụ trực quan hóa và tương tác, giúp giảm thiểu thời gian chuẩn bị bài giảng và tăng cường sự tham gia của sinh viên. Bên cạnh đó, hệ thống còn giúp họ dễ dàng theo dõi tiến độ học tập của sinh viên, thu thập phản hồi và cải tiến phương pháp giảng dạy liên tục, từ đó nâng cao chất lượng giáo dục.

3. **Nhà phát triển giáo dục:** sẽ có cơ hội mở rộng thị trường và tăng trưởng doanh thu thông qua việc cung cấp các giải pháp giáo dục chất lượng cao trên nền tảng DSA Visualizer. Nhờ tích hợp các công nghệ hiện đại như AI và visualization, họ có thể tạo ra các sản phẩm giáo dục đột phá và tiếp cận được nhiều đối tượng học tập đa dạng. Hơn nữa, việc hợp tác này giúp tạo ra một hệ sinh thái giáo dục toàn diện, đồng thời mang đến cho người học trải nghiệm học tập chất lượng cao và hiệu quả.

4. **Quản trị viên hệ thống:** sẽ được hỗ trợ bởi một hệ thống quản lý hiện đại và tự động hóa cao, giúp giảm bớt khối lượng công việc thủ công và tối ưu hóa quy trình vận hành. Họ có thể nhanh chóng giám sát và xử lý các vấn đề kỹ thuật, quản lý người dùng và nội dung một cách hiệu quả, từ đó

nâng cao chất lượng dịch vụ và đảm bảo hoạt động ổn định của hệ thống. Hệ thống báo cáo và analytics chi tiết cũng giúp họ đưa ra các quyết định vận hành đúng đắn và cải thiện liên tục chất lượng dịch vụ.

1.2 Task 1.2: Functional and non-functional requirements

1.2.1 Functional

1. Đối với Sinh viên và học sinh:

- **Tìm kiếm thuật toán và cấu trúc dữ liệu:** Người dùng có thể tìm kiếm các thuật toán và cấu trúc dữ liệu bằng cách nhập từ khóa như tên thuật toán hoặc loại cấu trúc dữ liệu. Hệ thống sẽ trả về danh sách các kết quả phù hợp để người dùng tham khảo. Hiển thị thông tin cơ bản như tên, độ phức tạp thời gian, và ứng dụng thực tế.
- **Xem chi tiết và trực quan hóa:** Người dùng có thể nhấp vào một thuật toán cụ thể để xem chi tiết về cách thức hoạt động, pseudocode, và implementation. Cho phép xem animation trực quan hóa từng bước thực hiện của thuật toán để phục vụ cho việc tìm hiểu và nghiên cứu.
- **Thực hành và tương tác:** Cho phép người dùng nhập dữ liệu đầu vào tùy chỉnh để mô phỏng quá trình thực hiện thuật toán. Hệ thống cung cấp các controls để điều chỉnh tốc độ animation, pause/resume, và step-by-step execution.
- **Quản lý tiến độ học tập:** Người dùng có thể xem lại lịch sử các thuật toán đã học trong mục "Learning Progress". Thông tin này bao gồm tên thuật toán, thời gian học, và mức độ hiểu biết. Cho phép đặt bookmark cho các thuật toán yêu thích và tạo learning path cá nhân.

- **Hỗ trợ AI và chatbot:** Cung cấp AI assistant để người dùng có thể đặt câu hỏi về thuật toán và nhận các câu trả lời chi tiết, gợi ý học tập, và giải thích code.

2. Đối với Giảng viên và giáo viên:

- **Quản lý nội dung học tập:** Cho phép thêm, sửa, và xóa các thông tin về thuật toán như mô tả, code examples, và test cases. Hệ thống hiển thị danh sách các thuật toán hiện có để giảng viên có thể chỉnh sửa hoặc tùy chỉnh cho phù hợp với chương trình giảng dạy.
- **Quản lý thông tin sinh viên:** Hiển thị danh sách tiến độ học tập của sinh viên bao gồm các thuật toán đã học, thời gian học tập, và kết quả quiz. Cung cấp chức năng tạo assignments và theo dõi completion rate.
- **Tạo và quản lý bài kiểm tra:** Cho phép tạo các quiz và assignments về thuật toán với câu hỏi đa dạng. Hệ thống tự động chấm điểm và cung cấp feedback chi tiết cho sinh viên.
- **Phân quyền và quản lý lớp học:** Có khả năng tạo virtual classrooms, mời sinh viên tham gia, và phân quyền truy cập vào các tài nguyên học tập cụ thể.
- **Thống kê và báo cáo:** Hiển thị số lượng sinh viên đã hoàn thành bài học, thời gian học trung bình, và các thuật toán được quan tâm nhiều nhất để giảng viên có cái nhìn tổng quan về hiệu quả giảng dạy.

3. Đối với Nhà phát triển giáo dục:

- **API tích hợp:** Cung cấp RESTful APIs cho phép tích hợp với các hệ thống LMS khác. APIs hỗ trợ truy cập nội dung, user management, và progress tracking.
- **Customization và white-labeling:** Cho phép tùy chỉnh giao diện, thương hiệu, và nội dung để phù hợp với yêu cầu của từng tổ chức giáo dục.

- **Analytics và insights:** Cung cấp dashboard analytics chi tiết về user behavior, learning patterns, và engagement metrics để hỗ trợ cải thiện chất lượng nội dung.

4. Đối với Quản trị viên hệ thống:

- **Quản lý người dùng và quyền hạn:** Quản trị viên có thể xem danh sách người dùng, phân quyền, và quản lý accounts. Cung cấp tools để monitor user activities và system usage.
- **Quản lý nội dung và quality control:** Cho phép review, approve, và publish nội dung mới. Đảm bảo chất lượng và tính chính xác của các thuật toán và visualizations.
- **Monitoring và maintenance:** Cung cấp system monitoring tools, performance metrics, và automated backup/restore functionality.

1.2.2 Non-functional

1. Hiệu năng:

- Ứng dụng web cần được tối ưu để đảm bảo thời gian tải trang dưới 3 giây trên kết nối internet trung bình, tạo trải nghiệm mượt mà cho người dùng khi truy cập các visualization modules.
- Hệ thống phải có khả năng xử lý ít nhất 1000 người dùng đồng thời thực hiện các animation và tương tác mà không gặp tình trạng quá tải hoặc sụt giảm hiệu năng đáng kể.
- Animations phải chạy mượt mà với framerate ổn định $\geq$ 30 FPS để đảm bảo trải nghiệm học tập tốt nhất.

2. Tính sẵn sàng:

- Hệ thống phải đảm bảo uptime 99.5% với khả năng tự động khôi phục trong trường hợp gặp sự cố.

- Không yêu cầu đảm bảo thời gian uptime cao như trong môi trường production, nhưng cần có khả năng nhanh chóng restart và recovery sau sự cố.

3. Tính bảo mật:

- Hệ thống cần triển khai các biện pháp bảo mật cơ bản như mã hóa SSL/TLS cho dữ liệu truyền tải, đảm bảo an toàn thông tin cá nhân và dữ liệu học tập của người dùng.
- Thực hiện xác thực và phân quyền người dùng đơn giản để kiểm soát quyền truy cập và tránh các lỗi bảo mật cơ bản.
- Tuân thủ các quy định về bảo vệ dữ liệu giáo dục và privacy laws.

4. Khả năng mở rộng:

- Hệ thống cần được thiết kế theo hướng dễ mở rộng, cho phép bổ sung các thuật toán và cấu trúc dữ liệu mới mà không làm ảnh hưởng đến cấu trúc hiện tại.
- Kiến trúc microservices để dễ dàng scale các components riêng biệt khi cần thiết.

5. Trải nghiệm người dùng:

- Giao diện người dùng cần được thiết kế đơn giản, trực quan, responsive design tương thích với desktop, tablet và mobile devices.
- Hỗ trợ accessibility features để đảm bảo người dùng khuyết tật có thể sử dụng hệ thống hiệu quả.
- Cung cấp multiple language support và comprehensive help documentation.

1.2.3 Functional

1. Đối với Sinh viên và học sinh:

- **Truy cập và lựa chọn cấu trúc dữ liệu:** Người dùng có thể dễ dàng truy cập vào danh sách các cấu trúc dữ liệu có sẵn bao gồm Stack, Queue, Linked List, Binary Tree, AVL Tree, và Heap. Hệ thống hiển thị mô tả ngắn gọn và các tính năng chính của từng cấu trúc để giúp người dùng lựa chọn phù hợp với mục đích học tập.

- **Mô phỏng thuật toán:** Người dùng có thể chọn các thuật toán cụ thể như sorting (bubble sort, merge sort, quick sort), searching (binary search, linear search), và graph algorithms (Dijkstra, BFS, DFS) để quan sát quá trình thực hiện từng bước một. Hệ thống cung cấp chức năng điều khiển tốc độ mô phỏng, tạm dừng, và từng bước để người dùng có thể theo dõi chi tiết.

- **Tương tác với dữ liệu:** Cho phép người dùng nhập dữ liệu tùy chỉnh hoặc sử dụng các bộ dữ liệu mẫu có sẵn để thử nghiệm với các thuật toán khác nhau. Hệ thống hỗ trợ nhiều định dạng đầu vào và cung cấp gợi ý về dữ liệu phù hợp cho từng loại thuật toán.

- **Theo dõi tiến độ học tập:** Người dùng có thể xem lại lịch sử các thuật toán đã thực hành trong phần "Lịch sử học tập". Thông tin này bao gồm loại thuật toán, thời gian thực hiện, và kết quả đạt được. Hệ thống cung cấp thống kê chi tiết về tiến độ học tập và đề xuất các chủ đề cần ôn tập.

- **Bài tập và thử thách:** Cung cấp các bài tập thực hành với nhiều mức độ khó khăn từ cơ bản đến nâng cao. Người dùng có thể giải quyết các thử thách lập trình và nhận phản hồi tức thì về kết quả của mình.

2. Đối với Giảng viên và giáo viên:

- **Quản lý nội dung giảng dạy:** Cho phép tạo, chỉnh sửa, và xóa các bài học tùy chỉnh bao gồm lý thuyết, ví dụ minh họa, và bài tập thực hành. Giảng

viên có thể sắp xếp nội dung theo chương trình học cụ thể và tạo ra các lộ trình học tập có cấu trúc.

- **Theo dõi sinh viên:** Hệ thống cung cấp dashboard để giảng viên có thể theo dõi tiến độ học tập của từng sinh viên, xem báo cáo chi tiết về thời gian học tập, kết quả bài tập, và các khó khăn gặp phải. Thông tin này giúp giảng viên điều chỉnh phương pháp giảng dạy cho phù hợp.
- **Tạo bài kiểm tra và đánh giá:** Cho phép tạo các bài kiểm tra trực tuyến với câu hỏi đa dạng bao gồm trắc nghiệm, tự luận, và các bài tập thực hành. Hệ thống tự động chấm điểm và cung cấp phản hồi chi tiết cho sinh viên.
- **Quản lý lớp học:** Giảng viên có thể tạo và quản lý các lớp học ảo, mời sinh viên tham gia, và phân quyền truy cập vào các tài nguyên học tập cụ thể.
- **Báo cáo thống kê:** Hiển thị báo cáo chi tiết về hoạt động học tập của lớp, bao gồm thời gian trung bình hoàn thành bài tập, các thuật toán được quan tâm nhiều nhất, và điểm số trung bình của từng chủ đề.

3. Đối với người học tự học:

- **Lộ trình học tập cá nhân hóa:** Hệ thống cung cấp các lộ trình học tập được thiết kế dựa trên trình độ và mục tiêu của người học. Có thể lựa chọn từ lộ trình cơ bản cho người mới bắt đầu đến nâng cao cho những người có kiến thức nền tảng.
- **Hệ thống đánh giá năng lực:** Cung cấp các bài kiểm tra đánh giá trình độ để xác định điểm khởi đầu phù hợp và theo dõi sự tiến bộ trong quá trình học tập.
- **Cộng đồng học tập:** Tạo không gian để người học có thể thảo luận, chia sẻ kinh nghiệm, và hỗ trợ lẫn nhau trong quá trình học tập.
- **Chứng chỉ và huy hiệu:** Hệ thống cấp chứng chỉ hoàn thành và các huy hiệu thành tích để động viên và ghi nhận nỗ lực học tập của người dùng.

4. Đối với nhà phát triển giáo dục:

- **API tích hợp:** Cung cấp API đầy đủ cho phép tích hợp các thành phần của hệ thống vào các ứng dụng giáo dục khác. API hỗ trợ các chức năng chính như truy cập nội dung, theo dõi tiến độ, và quản lý người dùng.
- **Tùy chỉnh giao diện:** Cho phép thay đổi giao diện và thương hiệu của hệ thống để phù hợp với yêu cầu của từng tổ chức. Hỗ trợ white-label solution cho các đối tác giáo dục.
- **Phân tích và báo cáo:** Cung cấp công cụ phân tích chi tiết về hành vi người dùng, hiệu quả học tập, và các chỉ số quan trọng khác để hỗ trợ việc cải thiện sản phẩm giáo dục.
- **SDK và Documentation:** Cung cấp bộ công cụ phát triển và tài liệu kỹ thuật chi tiết để các nhà phát triển có thể dễ dàng tích hợp và mở rộng hệ thống.

1.2.4 Non-functional

1. Hiệu năng:

- Ứng dụng web cần được tối ưu để đảm bảo thời gian tải trang dưới 3 giây trên kết nối internet trung bình, tạo trải nghiệm mượt mà cho người dùng khi truy cập các module trực quan hóa.
- Hệ thống phải có khả năng xử lý ít nhất 500 người dùng đồng thời thực hiện các mô phỏng thuật toán mà không gặp tình trạng quá tải hoặc sụt giảm hiệu năng đáng kể.
- Các animation và mô phỏng phải chạy mượt mà với tốc độ ít nhất 30 FPS để đảm bảo trải nghiệm trực quan tốt nhất.

2. Tính sẵn sàng:

- Hệ thống phải đảm bảo tính sẵn sàng hoạt động 99.5% thời gian, với khả năng tự động khôi phục trong trường hợp gặp sự cố.
- Triển khai cơ chế backup tự động và khả năng failover để đảm bảo dịch vụ không bị gián đoạn trong quá trình học tập và giảng dạy.

3. Tính bảo mật:

- Hệ thống cần triển khai các biện pháp bảo mật toàn diện bao gồm mã hóa HTTPS/TLS cho tất cả dữ liệu truyền tải, đảm bảo an toàn thông tin cá nhân và dữ liệu học tập của người dùng.
- Thực hiện xác thực và phân quyền người dùng dựa trên vai trò (Role-Based Access Control) để kiểm soát quyền truy cập và bảo vệ nội dung giáo dục.
- Tuân thủ các quy định về bảo vệ dữ liệu cá nhân như GDPR và các chuẩn bảo mật quốc tế.

4. Khả năng mở rộng:

- Hệ thống cần được thiết kế theo kiến trúc microservices và sử dụng container để dễ dàng mở rộng theo chiều ngang khi có nhu cầu tăng số lượng người dùng.
- Cơ sở dữ liệu phải hỗ trợ sharding và replication để đảm bảo khả năng mở rộng và hiệu năng khi dữ liệu tăng trưởng.

5. Trải nghiệm người dùng:

- Giao diện người dùng cần được thiết kế responsive, tương thích với nhiều thiết bị từ desktop đến mobile và tablet.
- Hỗ trợ đa ngôn ngữ và accessibility để đảm bảo tính bao trùm cho người dùng khuyết tật.
- Cung cấp hướng dẫn sử dụng chi tiết và hệ thống help desk để hỗ trợ người dùng khi gặp khó khăn.

6. Khả năng tương thích:

- Hỗ trợ đầy đủ các trình duyệt web phổ biến bao gồm Chrome, Firefox, Safari, và Edge phiên bản mới nhất.
- Tương thích với các hệ điều hành khác nhau và có thể tích hợp với các hệ thống quản lý học tập (LMS) hiện có.
- Đảm bảo khả năng tương thích ngược khi có cập nhật phiên bản mới của hệ thống.

1. **NFR-SEC-001**: OAuth 2.0 authentication
2. **NFR-SEC-002**: Role-based access control (RBAC)
3. **NFR-SEC-003**: HTTPS encryption for all communications
4. **NFR-SEC-004**: Input validation và sanitization
5. **NFR-SEC-005**: Regular security audits và penetration testing

1.2.4.1 Usability Requirements

1. **NFR-USA-001**: Responsive design (mobile, tablet, desktop)
2. **NFR-USA-002**: WCAG 2.1 AA accessibility compliance
3. **NFR-USA-003**: Multi-language support (Vi, En)
4. **NFR-USA-004**: Intuitive navigation ≤ 3 clicks to any feature
5. **NFR-USA-005**: Consistent UI/UX across all modules

Chapter 2

PHÂN TÍCH VÀ TÌM HIỂU THỊ TRƯỜNG

Nhiều nền tảng học tập trực tuyến và công cụ trực quan hóa thuật toán đã được phát triển nhằm cung cấp các phương pháp học tập hiện đại cho sinh viên và người học tự do. Những nền tảng này không chỉ giúp người dùng dễ dàng tiếp cận kiến thức về cấu trúc dữ liệu và thuật toán mà còn cung cấp các tính năng đặc biệt để thu hút và duy trì sự tham gia của người học. Một trong những ví dụ điển hình là VisuAlgo, nền tảng trực quan hóa thuật toán hàng đầu được phát triển tại National University of Singapore, cung cấp các công cụ trực quan hóa đa dạng từ thuật toán sắp xếp, cây nhị phân, đến các thuật toán đồ thị phức tạp, giúp người dùng có trải nghiệm học tập toàn diện về Data Structures and Algorithms.

Ngoài ra, các đối thủ lớn khác trên thị trường như Algorithm Visualizer, Data Structure Visualizations (University of San Francisco), và LeetCode cũng cung cấp những dịch vụ tương tự với các phương pháp tiếp cận khác nhau nhằm tăng sức cạnh tranh. Các nền tảng này không ngừng cải tiến giao diện và tính năng, tạo ra những trải nghiệm người dùng dễ dàng và thuận tiện hơn. Đặc biệt, các tính năng như hỗ trợ đa ngôn ngữ lập trình, interactive coding environments, và adaptive learning paths đã giúp những nền tảng như Coursera Algorithm

Courses, edX Computer Science programs, và Khan Academy Computer Programming củng cố vị trí trong lòng người dùng, đặc biệt là tại các thị trường giáo dục công nghệ phát triển như Bắc Mỹ, Châu Âu và Châu Á.

Tuy nhiên, sự cạnh tranh ngày càng khốc liệt đòi hỏi các nền tảng phải không ngừng đổi mới và tối ưu hóa trải nghiệm học tập. Các yếu tố như AI-powered personalization, real-time collaboration, mobile-first approach, và gamification elements cũng là những thách thức lớn mà các nền tảng giáo dục DSA cần chú ý để tiếp tục phát triển bền vững trong thị trường giáo dục trực tuyến ngày càng đông đúc và cạnh tranh gay gắt.

Theo báo cáo của Global Market Insights (2023), thị trường EdTech toàn cầu dự kiến sẽ đạt 377.85 tỷ USD vào năm 2028, với tốc độ tăng trưởng kép hàng năm (CAGR) là 13.4%. Trong đó, phân khúc STEM education chiếm 28% thị phần, tương đương khoảng 105 tỷ USD. Điều này cho thấy tiềm năng to lớn cho các sản phẩm giáo dục công nghệ như DSA Visualizer Platform.

Phân tích cạnh tranh cho thấy các điểm mạnh và yếu của các giải pháp hiện tại:

VisuAlgo: Được đánh giá cao về chất lượng trực quan hóa và độ chính xác thuật toán, tuy nhiên giao diện còn đơn giản và thiếu tính tương tác. Nền tảng này phục vụ chủ yếu cho mục đích giảng dạy và chưa có hệ thống quản lý học tập hoàn chỉnh.

Algorithm Visualizer: Có cộng đồng developer tích cực đóng góp và mã nguồn mở, nhưng thiếu hướng dẫn có cấu trúc và hệ thống đánh giá tiến độ. Nền tảng này phù hợp với những người đã có kiến thức nền tảng nhưng khó tiếp cận với người mới bắt đầu.

LeetCode: Mạnh về bài tập thực hành và chuẩn bị phỏng vấn, có hệ thống discussion forum phong phú, tuy nhiên tập trung chủ yếu vào problem solving hơn là hiểu biết sâu về thuật toán. Thiếu các công cụ trực quan hóa chất lượng cao.

Coursera/edX DSA Courses: Có nội dung học thuật chất lượng cao và được giảng dạy bởi các giáo sư danh tiếng, nhưng thiếu tính tương tác và công cụ trực

quan hóa. Phí học cao và không linh hoạt về thời gian học.

Từ phân tích này, chúng ta nhận thấy có một khoảng trống trong thị trường cho một nền tảng kết hợp được chất lượng trực quan hóa cao, hệ thống quản lý học tập hoàn chỉnh, cộng đồng học tập tích cực, và khả năng tiếp cận dễ dàng cho người mới bắt đầu. Đây chính là cơ hội để DSA Visualizer Platform có thể phát triển và chiếm lĩnh thị phần trong lĩnh vực giáo dục DSA trực tuyến.

2.1 Phân tích thị trường và cơ hội

2.1.1 Quy mô thị trường

Thị trường giáo dục trực tuyến (EdTech) đang trải qua giai đoạn tăng trưởng mạnh mẽ, đặc biệt sau đại dịch COVID-19 khi việc học trực tuyến trở thành xu hướng chủ đạo. Theo Research and Markets (2023), thị trường EdTech toàn cầu có giá trị 254.8 tỷ USD năm 2021 và dự kiến đạt 605.4 tỷ USD vào năm 2027. Trong phân khúc STEM education, Computer Science education chiếm khoảng 35% thị phần, tương đương 89 tỷ USD năm 2023. Đặc biệt, nhu cầu học lập trình và thuật toán tăng mạnh với tốc độ 18.7% CAGR do:

- Sự bùng nổ của ngành công nghệ và nhu cầu nhân lực IT
- Xu hướng chuyển đổi số ở mọi lĩnh vực
- Tăng cường giáo dục STEM trong các chương trình đào tạo
- Nhu cầu nâng cao kỹ năng của lực lượng lao động hiện tại

2.1.2 Phân tích đối thủ cạnh tranh

2.1.2.1 Đối thủ trực tiếp

1. VisuAlgo (National University of Singapore)

- *Điểm mạnh:* Giao diện đẹp, thuật toán chính xác, hỗ trợ đa ngôn ngữ

- *Điểm yếu:* Thiếu tính tương tác, không có hệ thống quản lý học tập
- *Lượng người dùng:* 2.5 triệu visitors/tháng
- *Mô hình kinh doanh:* Miễn phí hoàn toàn

2. Algorithm Visualizer (Open Source)

- *Điểm mạnh:* Cộng đồng phát triển tích cực, mã nguồn mở
- *Điểm yếu:* Giao diện đơn giản, thiếu hướng dẫn có cấu trúc
- *Lượng người dùng:* 800K visitors/tháng
- *Mô hình kinh doanh:* Donation-based

3. Data Structure Visualizations (USF)

- *Điểm mạnh:* Nội dung học thuật chất lượng cao
- *Điểm yếu:* Giao diện lỗi thời, hiệu năng kém
- *Lượng người dùng:* 300K visitors/tháng
- *Mô hình kinh doanh:* Academic use only

2.1.2.2 Đối thủ gián tiếp

1. LeetCode

- *Điểm mạnh:* Cộng đồng lớn, bài tập đa dạng, chuẩn bị phỏng vấn
- *Điểm yếu:* Tập trung vào problem solving, ít trực quan hóa
- *Lượng người dùng:* 15 triệu registered users
- *Doanh thu:* 50 triệu USD/năm (LeetCode Premium)

2. HackerRank

- *Điểm mạnh:* Nền tảng tuyển dụng tích hợp, variety in challenges

- *Điểm yếu:* Ít focus vào educational aspect
- *Lượng người dùng:* 12 triệu developers
- *Doanh thu:* 100 triệu USD/năm

3. Coursera/edX DSA Courses

- *Điểm mạnh:* Nội dung từ các trường đại học danh tiếng
- *Điểm yếu:* Thiếu tính tương tác, phí học cao
- *Lượng người dùng:* Coursera 100M+, edX 40M+
- *Doanh thu:* Coursera 523M USD, edX 100M USD

2.1.3 Cơ hội thị trường

2.1.3.1 Gaps trong thị trường hiện tại

1. **Thiếu tích hợp hoàn chỉnh:** Không có nền tảng nào kết hợp được chất lượng trực quan hóa cao, hệ thống LMS hoàn chỉnh, và cộng đồng học tập tích cực.
2. **Personalization hạn chế:** Các giải pháp hiện tại ít sử dụng AI để cá nhân hóa trải nghiệm học tập.
3. **Gamification thiếu hiệu quả:** Hầu hết đều thiếu các yếu tố game hóa để duy trì động lực học tập.
4. **Mobile experience kém:** Nhiều nền tảng chưa được tối ưu cho mobile learning.
5. **Hỗ trợ đa ngôn ngữ lập trình:** Ít nền tảng show code implementation đồng thời cho nhiều ngôn ngữ.

2.1.3.2 Xu hướng thị trường

1. **AI-powered education:** Tăng 42% năm 2023, với ChatGPT và AI tutors
2. **Microlearning:** Học theo modules nhỏ, phù hợp với attention span của Gen Z
3. **Social learning:** Học tập cộng đồng và peer-to-peer support
4. **Mobile-first approach:** 70% traffic từ mobile devices
5. **Subscription models:** Freemium model với premium features

2.1.4 Target market analysis

2.1.4.1 Primary segments

1. Computer Science Students (60% target market)

- Quy mô: 4.5 triệu students toàn cầu
- Đặc điểm: 18-25 tuổi, tech-savvy, price-sensitive
- Pain points: Khó hiểu thuật toán abstract, thiếu thực hành
- Willingness to pay: $5-15/month

2. Self-learners & Career changers (25% target market)

- Quy mô: 2 triệu individuals
- Đặc điểm: 25-40 tuổi, motivated, budget constraints
- Pain points: Thiếu structured learning path, time constraints
- Willingness to pay: $10-30/month

3. Educational institutions (15% target market)

- Quy mô: 50,000 institutions globally

- Đặc điểm: Budget cycles, need for proven ROI
- Pain points: Outdated teaching tools, student engagement
- Willingness to pay: $500-5000/year per institution

2.1.4.2 Market entry strategy

1. **Phase 1:** Focus on individual learners với freemium model
2. **Phase 2:** Expand sang educational institutions
3. **Phase 3:** Corporate training và B2B solutions
4. **Phase 4:** International expansion, especially Asia-Pacific

Weaknesses:

- Limited customization options
- Không có AI assistant

2.1.5 Algorithm Visualizer

Ưu điểm:

- Open-source project
- Code tracing capabilities
- Multiple programming languages
- User contribution system

Nhược điểm:

- UI/UX chưa thân thiện
- Performance issues với large datasets
- Limited educational resources
- Thiếu structured learning path

2.1.6 Data Structure Visualizations (USF)

Ưu điểm:

- Comprehensive coverage of data structures
- Step-by-step execution
- Educational focus
- Free to use

Nhược điểm:

- Outdated interface
- Limited interactivity
- No mobile support
- Lack of modern features

2.1.7 Sorting Algorithms Animations

Ưu điểm:

- Focused on sorting algorithms
- Clear visual comparisons
- Performance metrics display
- Simple và intuitive

Nhược điểm:

- Limited scope (chỉ sorting)
- No explanation text
- Static implementation
- No learning management

2.2 Công nghệ nền tảng

2.2.1 Frontend Technologies

2.2.1.1 React.js

React.js được chọn làm thư viện chính cho frontend vì:

- **Component-based architecture**: Tái sử dụng code hiệu quả
- **Virtual DOM**: Hiệu năng cao cho real-time updates
- **Rich ecosystem**: Nhiều thư viện hỗ trợ animation
- **Community support**: Documentation và tutorials phong phú

2.2.1.2 Next.js

Next.js framework cung cấp:

- **Server-Side Rendering**: SEO optimization
- **Static Site Generation**: Performance tối ưu
- **API Routes**: Backend integration seamless
- **Built-in optimization**: Image, font, script optimization

2.2.1.3 TypeScript

TypeScript benefits:

- **Type safety**: Giảm bugs trong development
- **IntelliSense**: Developer experience tốt hơn
- **Refactoring support**: Maintain large codebase
- **Interface definition**: Clear API contracts

2.2.2 Animation Libraries

2.2.2.1 Framer Motion

- Declarative animation API
- Hardware-accelerated animations
- Gesture support
- Layout animations

2.2.2.2 React Spring

- Physics-based animations
- High performance
- Hook-based API
- Complex animation sequences

2.2.2.3 D3.js

- Data-driven visualizations
- SVG manipulation
- Custom chart creation
- Mathematical calculations

2.2.3 Backend Technologies

2.2.3.1 Node.js

- JavaScript runtime cho server
- Non-blocking I/O operations

- NPM ecosystem
- Real-time applications support

2.2.3.2 Express.js

- Lightweight web framework
- Middleware support
- RESTful API development
- Easy integration

2.2.3.3 Socket.io

- Real-time bidirectional communication
- Auto-fallback support
- Room-based messaging
- Cross-platform compatibility

2.2.4 Database Technologies

2.2.4.1 PostgreSQL

- ACID compliance
- Complex queries support
- JSON data type
- Scalability

2.2.4.2 Redis

- In-memory caching
- Session storage
- Rate limiting
- Real-time features

2.2.4.3 MongoDB

- Document-based storage
- Flexible schema
- Aggregation pipeline
- Horizontal scaling

2.3 Gaps trong các nghiên cứu hiện tại

2.3.1 Technical Gaps

1. **Limited AI Integration**:
    - Hầu hết platforms thiếu AI assistant
    - No personalized learning recommendations
    - Limited natural language processing
2. **Poor Mobile Experience**:
    - Không responsive design
    - Touch gesture support limited
    - Performance issues on mobile devices
3. **Scalability Issues**:

- Monolithic architecture
- No cloud-native design
- Limited concurrent user support

2.3.2 Pedagogical Gaps

1. **Lack of Learning Path**:
   - No structured curriculum
   - Random algorithm selection
   - No prerequisite tracking
2. **Missing Assessment**:
   - No knowledge evaluation
   - Limited feedback mechanisms
   - No progress tracking
3. **Community Absence**:
   - No peer interaction
   - Limited collaboration features
   - No knowledge sharing platform

2.4 Đóng góp của đồ án

Đồ án này đóng góp những điểm mới sau:

1. **Comprehensive Platform**:
   - Tích hợp visualizer, learning management, community
   - End-to-end learning experience
   - Modern technology stack

2. **AI-Powered Learning**:
    - Multi-model AI integration (GPT + Gemini)
    - Contextual help và code generation
    - Personalized learning recommendations
3. **Community-Driven Approach**:
    - Forum và Q&A system
    - Peer learning support
    - Knowledge sharing platform
4. **Production-Ready Architecture**:
    - Microservice design
    - Cloud deployment
    - Scalable và maintainable

Chapter 3

PHÂN TÍCH HỆ THỐNG

3.1 Use Case Diagram

3.1.1 Tổng quan Use Case

Hệ thống DSA Visualizer Platform phục vụ ba nhóm actor chính: Student, Instructor và Admin. Mỗi actor có các use case riêng biệt phù hợp với vai trò và quyền hạn của họ trong hệ thống.

[Use Case Diagram - System Overview]

Diagram available in enhanced-diagrams/usecase-system-overview.drawio

3.1.2 Use Case chi tiết cho Learning Process

Quá trình học tập là core functionality của platform, bao gồm nhiều use case phức tạp với các interaction giữa Student và các subsystem khác nhau.

[Use Case Diagram - Detailed Scenarios]

Diagram available in enhanced-diagrams/usecase-detailed-scenarios.drawio

Các use case chính trong Learning Process:

1. **Start Learning Session**: Khởi tạo session học tập mới

2. **Select Algorithm**: Chọn thuật toán cần học

3. **Study Theory**: Đọc tài liệu lý thuyết

4. **Practice with Visualization**: Thực hành với animation

5. **Solve Practice Problems**: Giải bài tập thực hành

6. **Take Assessment**: Làm bài kiểm tra đánh giá

7. **Get AI Assistance**: Nhận hỗ trợ từ AI assistant

8. **Track Progress**: Theo dõi tiến độ học tập

3.1.3 Detailed Use Case Specifications

3.1.3.1 UC001: Algorithm Visualization Learning

Use Case ID	UC001
Use Case Name	Algorithm Visualization Learning
Actor	Student
Description	Học viên học thuật toán thông qua visualization interactive
Precondition - Hệ thống có sẵn algorithm content	- Học viên đã đăng nhập vào hệ thống
Basic Flow 2. Hệ thống hiển thị danh sách algorithms available 3. Học viên chọn specific algorithm (VD: Quick Sort) 4. Hệ thống load algorithm visualizer 5. Học viên input dữ liệu hoặc sử dụng sample data 6. Học viên bắt đầu visualization process 7. Hệ thống thực hiện step-by-	1. Học viên chọn loại thuật toán muốn học

3.1.3.2 UC002: Interactive Algorithm Practice

Use Case ID	UC002
Use Case Name	Interactive Algorithm Practice
Actor	Student
Description	Học viên thực hành thuật toán với interactive controls và custom input
Precondition - Hệ thống có sẵn practice environment	- Học viên đã hoàn thành basic learning session
Basic Flow 2. Hệ thống hiển thị available practice algorithms 3. Học viên chọn algorithm để practice 4. Hệ thống load interactive practice environment 5. Học viên tạo custom input data 6. Học viên predict algorithm behavior 7. Học viên execute algorithm step by step 8. Hệ thống pro-	1. Học viên chọn Practice Mode

3.1.3.3 UC003: AI Assistant Consultation

Use Case ID	UC003
Use Case Name	AI Assistant Consultation
Actor	Student
Description	Học viên sử dụng AI Assistant để được hỗ trợ học tập và giải đáp thắc mắc
Precondition - AI Assistant service đang hoạt động	- Học viên đang trong learning session
Basic Flow 2. Hệ thống mở AI chat interface 3. Học viên nhập câu hỏi về algorithm 4. AI Assistant phân tích context và question 5. AI generate response với explanation 6. Hệ thống hiển thị answer với code examples 7. Học viên có thể ask follow-up questions 8. AI provide additional hints nếu cần	1. Học viên click vào AI Assistant icon

3.2 Class Diagram

3.2.1 Tổng quan Class Diagram

Class diagram của hệ thống DSA Visualizer được thiết kế theo mô hình MVC (Model-View-Controller) và Clean Architecture, đảm bảo tính modular và scalability.

[Class Diagram - Core System]

Diagram: class-diagram-clean.drawio

3.2.2 Các nhóm Class chính

3.2.2.1 User Management Classes

User Class:

- **Thuộc tính:** userID, email, username, password, role, createdAt, lastLogin
- **Phương thức:** login(), logout(), updateProfile(), changePassword()
- **Mối quan hệ:** User có nhiều LearningSession, có một UserProfile

UserProfile Class:

- **Thuộc tính:** profileID, firstName, lastName, avatar, bio, preferences
- **Phương thức:** updatePersonalInfo(), setPreferences(), uploadAvatar()
- **Mối quan hệ:** Thuộc về một User, có nhiều Achievement

3.2.2.2 Algorithm Visualization Classes

Algorithm Class:

- **Thuộc tính:** algorithmID, name, category, description, complexity, difficulty

- **Phương thức:** execute(), visualize(), getComplexity(), generateSteps()
- **Mối quan hệ:** Có nhiều AlgorithmStep, thuộc về một Category

Visualizer Class:

- **Thuộc tính:** visualizerID, type, config, animationSpeed, currentStep
- **Phương thức:** start(), pause(), resume(), reset(), setSpeed()
- **Mối quan hệ:** Sử dụng Algorithm, tạo ra VisualizationSession

AlgorithmStep Class:

- **Thuộc tính:** stepID, stepNumber, description, dataState, action
- **Phương thức:** execute(), undo(), getDescription(), visualize()
- **Mối quan hệ:** Thuộc về một Algorithm

3.2.2.3 Learning Management Classes

LearningSession Class:

- **Thuộc tính:** sessionID, userID, algorithmID, startTime, endTime, score
- **Phương thức:** start(), complete(), calculateScore(), saveProgress()
- **Mối quan hệ:** Thuộc về User và Algorithm

Progress Class:

- **Thuộc tính:** progressID, userID, totalSessions, completedAlgorithms, skillLevel
- **Phương thức:** updateProgress(), calculateSkillLevel(), getStatistics()
- **Mối quan hệ:** Thuộc về một User

3.2.2.4 Assessment Classes

Quiz Class:

- **Thuộc tính:** quizID, title, description, questions, timeLimit, difficulty
- **Phương thức:** generateQuestions(), calculateScore(), validateAnswers()
- **Mối quan hệ:** Có nhiều Question, có nhiều QuizResult

Question Class:

- **Thuộc tính:** questionID, content, options, correctAnswer, explanation
- **Phương thức:** validateAnswer(), getHint(), getExplanation()
- **Mối quan hệ:** Thuộc về một Quiz

3.2.3 Design Patterns được sử dụng

3.2.3.1 Factory Pattern

Sử dụng AlgorithmFactory để tạo ra các instance của different algorithm types:

- SortingAlgorithmFactory
- SearchAlgorithmFactory
- GraphAlgorithmFactory

3.2.3.2 Observer Pattern

VisualizationObserver được implement để notify UI components khi algorithm state changes:

- ProgressObserver: Cập nhật progress bar
- AnimationObserver: Trigger animation effects
- ScoreObserver: Calculate và display scores

3.2.3.3 Strategy Pattern

Sử dụng cho algorithm execution strategies:

- StepByStepStrategy: Execute từng bước
- ContinuousStrategy: Execute liên tục
- ComparisonStrategy: So sánh multiple algorithms

3.3 Activity Diagram

3.3.1 Tổng quan Activity Diagram

Activity diagram mô tả luồng hoạt động chính của hệ thống, từ khi user đăng nhập cho đến khi hoàn thành learning session.

[Activity Diagram - Learning Process]

Diagram: activity-diagram-clean.drawio

3.3.2 Quy trình hoạt động chính

3.3.2.1 Authentication Flow

1. **Start:** User truy cập application
2. **Decision:** Kiểm tra user đã login chưa?
3. **False:** Redirect đến login page
4. **Login Process:** User nhập credentials
5. **Validation:** System validate user information
6. **Decision:** Credentials có hợp lệ?
7. **False:** Show error message, return to login
8. **True:** Generate JWT token, redirect to dashboard

3.3.2.2 Algorithm Learning Flow

1. **Dashboard Access:** User vào main dashboard
2. **Category Selection:** User chọn algorithm category
3. **Algorithm Selection:** User chọn specific algorithm
4. **Visualizer Loading:** System load algorithm visualizer
5. **Input Configuration:** User configure input data
6. **Decision:** User muốn start visualization?
7. **True:** Begin algorithm execution
8. **Step-by-step Execution:** System execute từng step
9. **Animation Rendering:** Display visual animation
10. **User Interaction:** User có thể pause/resume/adjust speed
11. **Completion Check:** Algorithm execution complete?
12. **False:** Continue next step
13. **True:** Display final result và complexity analysis

3.3.2.3 AI Assistant Flow

1. **Trigger:** User click AI Assistant button
2. **Context Collection:** System collect current learning context
3. **Question Input:** User nhập question
4. **NLP Processing:** AI analyze question intent
5. **Knowledge Retrieval:** AI search relevant information
6. **Response Generation:** AI generate appropriate response

7. **Response Display:** System show AI response
8. **Decision:** User có additional questions?
9. **True:** Return to question input
10. **False:** Close AI Assistant

3.3.2.4 Assessment Flow

1. **Quiz Selection:** User chọn quiz để làm
2. **Quiz Loading:** System load quiz questions
3. **Question Display:** Show current question
4. **Answer Input:** User select/input answer
5. **Answer Validation:** System validate answer
6. **Feedback Display:** Show immediate feedback
7. **Progress Update:** Update quiz progress
8. **Decision:** Còn questions nào không?
9. **True:** Next question
10. **False:** Calculate final score
11. **Result Display:** Show quiz results và recommendations
12. **Progress Save:** Save user progress và achievements

3.3.3 Parallel Activities

Hệ thống hỗ trợ các parallel activities:

3.3.3.1 Background Services

- **Analytics Collection:** Continuous tracking user behavior
- **Performance Monitoring:** Real-time system performance tracking
- **Cache Management:** Background cache invalidation và refresh
- **Notification Processing:** Async notification sending

3.3.3.2 Real-time Features

- **Live Progress Updates:** Real-time progress synchronization
- **Community Activity:** Live discussion forum updates
- **Collaborative Learning:** Multi-user learning sessions

3.4 Sequence Diagram

3.4.1 Tổng quan Sequence Diagram

Sequence diagram minh họa tương tác giữa các objects trong hệ thống theo thời gian, đặc biệt tập trung vào main learning scenarios.

[Sequence Diagram - Algorithm Learning Process]

Diagram: sequence-diagram-clean.drawio

3.4.2 Chi tiết Sequence Interactions

3.4.2.1 Algorithm Visualization Sequence

Actors/Objects tham gia:

- Student (Actor)
- UI Controller

- Algorithm Service
- Visualizer Engine
- Database
- AI Assistant Service

Sequence of Messages:

1. **Student → UI Controller:** selectAlgorithm(algorithmType)
2. **UI Controller → Algorithm Service:** loadAlgorithm(algorithmType)
3. **Algorithm Service → Database:** getAlgorithmDetails(algorithmType)
4. **Database → Algorithm Service:** algorithmDetails
5. **Algorithm Service → UI Controller:** algorithmLoaded
6. **UI Controller → Student:** displayAlgorithmInterface()
7. **Student → UI Controller:** configureInput(inputData)
8. **UI Controller → Algorithm Service:** validateInput(inputData)
9. **Algorithm Service → UI Controller:** inputValid
10. **Student → UI Controller:** startVisualization()
11. **UI Controller → Visualizer Engine:** initializeVisualization(algorithm, data)
12. **Visualizer Engine → Algorithm Service:** executeStep()
13. **Algorithm Service → Visualizer Engine:** stepResult
14. **Visualizer Engine → UI Controller:** updateVisualization(stepResult)
15. **UI Controller → Student:** displayAnimation()
16. **Loop:** Repeat steps 12-15 until completion

17. **Visualizer Engine → UI Controller:** visualizationComplete()
18. **UI Controller → Database:** saveProgress(userId, sessionData)
19. **UI Controller → Student:** displayResults(finalState, complexity)

3.4.2.2 AI Assistant Interaction Sequence

Sequence of Messages:

1. **Student → UI Controller:** openAIAssistant()
2. **UI Controller → AI Assistant Service:** initializeSession(userId, context)
3. **AI Assistant Service → Database:** getUserLearningContext(userId)
4. **Database → AI Assistant Service:** learningContext
5. **AI Assistant Service → UI Controller:** sessionReady
6. **UI Controller → Student:** displayChatInterface()
7. **Student → UI Controller:** askQuestion(question)
8. **UI Controller → AI Assistant Service:** processQuestion(question, context)
9. **AI Assistant Service:** analyzeIntent(question)
10. **AI Assistant Service:** retrieveKnowledge(intent)
11. **AI Assistant Service:** generateResponse(knowledge, context)
12. **AI Assistant Service → UI Controller:** response
13. **UI Controller → Student:** displayResponse(response)
14. **AI Assistant Service → Database:** logInteraction(userId, question, response)

3.4.2.3 Assessment and Quiz Sequence

Sequence of Messages:

1. **Student → UI Controller:** selectQuiz(quizId)
2. **UI Controller → Assessment Service:** loadQuiz(quizId)
3. **Assessment Service → Database:** getQuizDetails(quizId)
4. **Database → Assessment Service:** quizData
5. **Assessment Service → UI Controller:** quizLoaded
6. **UI Controller → Student:** displayQuizInterface()
7. **Student → UI Controller:** startQuiz()
8. **UI Controller → Assessment Service:** beginQuizSession(userId, quizId)
9. **Loop for each question:**
   (a) **Assessment Service → UI Controller:** getNextQuestion()
   (b) **UI Controller → Student:** displayQuestion(question)
   (c) **Student → UI Controller:** submitAnswer(answer)
   (d) **UI Controller → Assessment Service:** validateAnswer(questionId, answer)
   (e) **Assessment Service → UI Controller:** answerResult(correct, explanation)
   (f) **UI Controller → Student:** showFeedback(result, explanation)
10. **Assessment Service → UI Controller:** calculateFinalScore()
11. **UI Controller → Database:** saveQuizResult(userId, quizId, score, answers)
12. **UI Controller → Student:** displayFinalResults(score, recommendations)

3.4.3 Error Handling Sequences

3.4.3.1 Authentication Error Sequence

1. **Student → UI Controller:** login(credentials)
2. **UI Controller → Auth Service:** validateCredentials(credentials)
3. **Auth Service → Database:** checkUserCredentials(credentials)
4. **Database → Auth Service:** userNotFound/invalidPassword
5. **Auth Service → UI Controller:** authenticationFailed(errorType)
6. **UI Controller → Student:** displayErrorMessage(errorType)
7. **UI Controller → Student:** requestCredentialsAgain()

3.4.3.2 System Error Recovery Sequence

1. **Any Service:** systemError(errorDetails)
2. **Error Handler:** logError(errorDetails)
3. **Error Handler → Monitoring Service:** reportError(errorDetails)
4. **Error Handler → UI Controller:** notifyUser(genericErrorMessage)
5. **UI Controller → Student:** displayErrorScreen(recoveryOptions)
6. **Error Handler:** attemptRecovery()
7. **Fallback Service:** provideFallbackFunctionality()

3.5 System Architecture Analysis

3.5.1 Tổng quan Architecture

Hệ thống DSA Visualizer được thiết kế theo mô hình 5-layer architecture để đảm bảo scalability, maintainability và performance optimization.

[System Architecture Diagram]

Diagram: system-architecture.drawio

3.5.2 Chi tiết các Layer

3.5.2.1 UI Layer (Presentation Layer)

Công nghệ sử dụng: Next.js 14, React 18, TypeScript, TailwindCSS

Thành phần chính:

- **Interactive Visualizers:** Canvas-based algorithm animations
- **Control Panels:** Speed control, step-by-step navigation
- **Dashboard Interface:** User progress tracking và statistics
- **AI Chat Interface:** Real-time chat với AI Assistant
- **Assessment Interface:** Quiz và practice exercises

Design Patterns:

- Component-based architecture
- State management với Context API
- Custom hooks cho reusable logic
- Responsive design patterns

3.5.2.2 Visualization Engine Layer

Công nghệ sử dụng: D3.js, Canvas API, WebGL

Core Components:

- **Animation Controller:** Quản lý animation timeline và state
- **Rendering Engine:** High-performance visualization rendering
- **Interaction Handler:** User input processing cho visualizations

- **State Manager:** Algorithm state tracking và history

Visualization Types:

- Array/List visualizations với color coding
- Tree structures với interactive nodes
- Graph visualizations với edge animations
- Comparison views cho multiple algorithms

3.5.2.3 Backend Services Layer

Công nghệ sử dụng: Node.js, Express.js, TypeScript

Microservices Architecture:

- **Algorithm Service:** Algorithm execution và step generation
- **User Service:** Authentication, profile management
- **Learning Service:** Progress tracking, session management
- **Assessment Service:** Quiz generation, scoring system
- **AI Service:** Natural language processing, knowledge retrieval
- **Analytics Service:** User behavior tracking, performance metrics

API Design:

- RESTful APIs với OpenAPI documentation
- GraphQL endpoints cho complex data queries
- WebSocket connections cho real-time features
- Rate limiting và security middleware

3.5.2.4 Data Management Layer

Công nghệ sử dụng: MongoDB, Redis, PostgreSQL

Database Strategy:

- **MongoDB:** User profiles, learning sessions, algorithm metadata
- **PostgreSQL:** Structured data, analytics, reporting
- **Redis:** Session caching, real-time data, leaderboards

Data Models:

- User và Profile entities với relationship mapping
- Algorithm metadata với complexity analysis
- Learning progress với detailed tracking
- Assessment results với statistical analysis

3.5.2.5 Infrastructure Layer

Deployment Strategy: Docker containers, Kubernetes orchestration

Cloud Services:

- **Compute:** Auto-scaling web servers
- **Storage:** Distributed file storage cho assets
- **CDN:** Global content delivery network
- **Monitoring:** Application performance monitoring

Security Measures:

- JWT-based authentication với refresh tokens
- HTTPS enforcement với SSL certificates
- Input validation và SQL injection prevention
- Cross-Origin Resource Sharing (CORS) configuration

3.5.3 Integration Patterns

3.5.3.1 Event-Driven Architecture

- User action events trigger visualization updates
- Progress events update learning analytics
- Achievement events trigger notification system
- Error events activate monitoring và alerting

3.5.3.2 Caching Strategy

- Browser caching cho static assets
- Redis caching cho frequently accessed data
- CDN caching cho global performance
- Application-level caching cho computed results

3.6 Design Principles và Best Practices

3.6.1 SOLID Principles Implementation

3.6.1.1 Single Responsibility Principle

Mỗi class và component có một responsibility duy nhất:

- VisualizationRenderer chỉ handle rendering logic
- AlgorithmExecutor chỉ handle algorithm execution
- UserManager chỉ handle user-related operations

3.6.1.2 Open/Closed Principle

Hệ thống được thiết kế để extend functionality without modification:

- Plugin architecture cho new algorithm types
- Extension system cho custom visualizations
- Configurable assessment frameworks

3.6.1.3 Liskov Substitution Principle

Abstract classes và interfaces đảm bảo substitutability:

- Algorithm interface có thể được implement bởi any algorithm type
- Visualizer interface support multiple rendering strategies
- Assessment interface accommodate different quiz types

3.6.2 Performance Optimization

3.6.2.1 Frontend Optimization

- Code splitting và lazy loading cho components
- Memoization cho expensive computations
- Virtual scrolling cho large datasets
- Debouncing cho user input handling

3.6.2.2 Backend Optimization

- Database query optimization với proper indexing
- Connection pooling cho database connections
- Asynchronous processing cho time-consuming tasks
- Load balancing cho high availability

3.6.3 Accessibility và Usability

3.6.3.1 Accessibility Features

- WCAG 2.1 compliance cho accessibility standards
- Keyboard navigation support
- Screen reader compatibility
- High contrast mode cho visual impairments

3.6.3.2 Usability Features

- Intuitive user interface design
- Progressive disclosure của complex features
- Contextual help và tooltips
- Responsive design cho multiple devices

3.7 Kết luận Chapter 3

Chapter 3 đã phân tích chi tiết hệ thống DSA Visualizer từ góc độ technical architecture và design. Các điểm chính bao gồm:

3.7.1 Use Case Analysis

Đã định nghĩa và mô tả chi tiết các use cases chính của hệ thống, bao gồm algorithm learning, interactive practice, và AI assistant consultation. Mỗi use case được documented với format table chi tiết theo chuẩn academic.

3.7.2 UML Diagrams Analysis

- **Class Diagram:** Thiết kế OOP với các design patterns phù hợp

- **Activity Diagram:** Mô tả chi tiết quy trình hoạt động và decision flows
- **Sequence Diagram:** Phân tích tương tác giữa objects theo timeline

3.7.3 System Architecture

Thiết kế 5-layer architecture đảm bảo:

- Scalability cho future expansion
- Maintainability với modular design
- Performance optimization với caching strategies
- Security với comprehensive protection measures

3.7.4 Technical Excellence

Áp dụng SOLID principles, design patterns, và best practices để tạo ra một hệ thống robust và professional.

Tiếp theo, Chapter 4 sẽ focus vào implementation details và technical specifications của từng component. **Use Case ID** UC-VIS-001

Use Case Name Practice with Visualization

Actor Student

Description Student tương tác với visualizer để hiểu cách thuật toán hoạt động từng bước

---

Preconditions - Student đã đăng nhập vào hệ thống

- Student đã chọn thuật toán cần học

Postconditions - Visualization history được lưu vào learning progress

- Time spent tracking được cập nhật

Main Flow 1. Student truy cập visualization page

2. System hiển thị thuật toán với default input

3. Student có thể:

3.1. Thay đổi input data

3.2. Điều chỉnh animation speed

3.3. Step through từng bước

3.4. View synchronized code

4. Student click "Start Animation"

5. System chạy visualization với settings đã chọn

6. Student quan sát và học tập

7. System lưu progress và statistics

Alternative Flows A1: Custom Input

3.1a. Student nhập custom input

3.1b. System validate input format

3.1c. Nếu invalid: show error message

A2: AI Assistance

6.1a. Student click "Get Help"

6.1b. AI Assistant explain current step

6.1c. Continue with main flow

Exception Flows E1: Animation Error

5.1a. Animation fails to render

5.1b. System show error message

5.1c. Offer alternative visualization mode

3.7.4.1 Use Case: Get AI Assistance

Table 3.4: Mô tả Use Case: Get AI Assistance

Use Case ID	UC-AI-001
Use Case Name	Get AI Assistance
Actor	Student
Description	Student nhận hỗ trợ từ AI assistant để hiểu thuật toán hoặc giải quyết vấn đề
Preconditions - AI service available	- Student đã đăng nhập vào hệ thống
Postconditions - AI usage statistics được cập nhật	- Conversation history được lưu
Main Flow 2. System mở AI chat interface 3. Student nhập câu hỏi hoặc chọn suggested questions 4. System gửi request đến AI service với context 5. AI service process request và trả về response 6. System hiển thị AI response với formatting 7. Student có thể: 7.1. Tiếp tục hỏi follow-up questions 7.2. Request code examples 7.3. Ask for explanation in different language 8. System lưu conversation history	1. Student click "AI Assistant" button
Alternative Flows 3.1a. Student request code for current algorithm 3.1b. Student chọn programming language 4.1a. AI generate code với explanation	A1: Code Generation Request

3.8 Class Diagram

3.8.1 Core Domain Classes

Hệ thống được thiết kế theo Domain-Driven Design với các core domain classes sau:

Figure 3.1: Class Diagram cho Core Domain

3.8.1.1 User Management Classes

- **User**: Base class cho tất cả users
- **Student**: Extends User, thêm learning-specific attributes
- **Instructor**: Extends User, thêm teaching-specific attributes
- **Admin**: Extends User, thêm system management capabilities

3.8.1.2 Learning Domain Classes

- **Algorithm**: Represents thuật toán với metadata
- **Visualization**: Chứa animation data và configuration
- **LearningSession**: Tracks user interaction với algorithm
- **Progress**: Theo dõi learning progress của user
- **Assessment**: Quiz và evaluation system

3.8.2 Service Layer Classes

Figure 3.2: Class Diagram cho Service Layer

3.8.2.1 Visualization Services

- **VisualizationEngine**: Core engine cho animation rendering

- **AnimationController**: Điều khiển animation playback
- **DataProcessor**: Xử lý input data cho visualization
- **RenderingService**: Abstract layer cho different renderers

3.8.2.2 AI Services

- **AIAssistant**: Main interface cho AI interactions
- **OpenAIService**: Integration với OpenAI GPT
- **GeminiService**: Integration với Google Gemini
- **ContextBuilder**: Xây dựng context cho AI requests

3.9 Activity Diagram

3.9.1 Learning Process Flow

Activity diagram sau mô tả complete learning process từ khi student bắt đầu học một thuật toán mới:

Figure 3.3: Activity Diagram cho Learning Process

3.9.1.1 Main Activities

1. **Algorithm Selection**: Student chọn thuật toán từ library
2. **Theory Study**: Đọc theoretical background
3. **Visualization Practice**: Tương tác với animated visualization
4. **Code Understanding**: Phân tích implementation code
5. **Problem Solving**: Áp dụng thuật toán vào bài tập
6. **Assessment**: Đánh giá mức độ hiểu biết
7. **Progress Update**: Cập nhật learning progress

3.9.1.2 Decision Points

- **Prerequisite Check**: Kiểm tra điều kiện tiên quyết
- **Difficulty Assessment**: Đánh giá độ khó phù hợp
- **Understanding Verification**: Xác nhận mức độ hiểu
- **Need Help Decision**: Quyết định có cần AI assistance

3.9.2 AI Assistance Flow

Figure 3.4: Activity Diagram cho AI Assistance Flow

Process khi student request AI help:

1. **Context Collection**: Gather current learning context
2. **Query Processing**: Process natural language query
3. **AI Model Selection**: Chọn appropriate AI model
4. **Response Generation**: Generate contextual response
5. **Response Formatting**: Format cho display
6. **Feedback Collection**: Thu thập user feedback

3.10 Sequence Diagram

3.10.1 Visualization Rendering Sequence

Sequence diagram sau mô tả interaction giữa các components khi render visualization:

Figure 3.5: Sequence Diagram cho Visualization Rendering

3.10.1.1 Participants

- **Student**: User initiating visualization
- **VisualizationUI**: Frontend visualization component
- **VisualizationEngine**: Core rendering engine
- **AnimationController**: Animation management
- **DataProcessor**: Input data processing
- **ProgressTracker**: Learning progress tracking

3.10.1.2 Key Interactions

1. Student selects algorithm và input parameters
2. UI validates input và sends request to engine
3. Engine processes data và prepares animation steps
4. AnimationController manages playback timing
5. Progress tracking updates throughout process
6. Final state saved to user progress

3.10.2 AI Assistant Interaction Sequence

Figure 3.6: Sequence Diagram cho AI Assistant Interaction

3.10.2.1 Multi-Model AI Architecture

Platform sử dụng multiple AI models để optimize cho different use cases:

1. **OpenAI GPT**: Cho natural language explanations
2. **Google Gemini**: Cho code generation và analysis

3. **Context Router**: Intelligent routing dựa trên query type

4. **Response Aggregator**: Combine responses từ multiple models

3.10.3 Community Interaction Sequence

Figure 3.7: Sequence Diagram cho Community Features

Mô tả interaction trong forum và Q&A system:

1. User posts question hoặc discussion topic

2. System validates content và applies moderation rules

3. Notification service alerts relevant users

4. Other users provide answers và comments

5. Voting system ranks responses

6. AI assistant có thể provide supplementary answers

7. Final resolution updates knowledge base