

ĐỀ CƯƠNG Ý TƯỞNG ĐỀ ÁN TỐT NGHIỆP

TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ LỚP HỌC TRỰC TUYẾN KITECLASS PLATFORM

SINH VIÊN THỰC HIỆN: NGUYỄN VĂN KIỆT

PHẦN 1: Ý TƯỞNG ĐỀ TÀI

1.1. Bối cảnh và lý do chọn đề tài

Trong bối cảnh chuyển đổi số giáo dục đang diễn ra mạnh mẽ, nhu cầu về các nền tảng học trực tuyến linh hoạt, có khả năng mở rộng và tùy chỉnh cao ngày càng tăng. Các giải pháp hiện tại thường gặp hạn chế về khả năng tùy biến theo đặc thù từng tổ chức giáo dục, đặc biệt là với các giảng viên độc lập hoặc tổ chức giáo dục nhỏ và vừa. Ví dụ: các giáo viên các cấp THCS, THPT dạy thêm, các gia sư chứng chỉ IELTS, JLPT, AWS, các trung tâm giảng dạy du học, ...

KiteClass Platform được phát triển nhằm giải quyết vấn đề này bằng cách cung cấp một nền tảng đa dạng, cho phép khách hàng nhanh chóng khởi tạo và vận hành hệ thống lớp học trực tuyến riêng của mình với giao diện và thương hiệu cá nhân hóa, được hỗ trợ bởi trí tuệ nhân tạo.

1.2. Mục tiêu đề tài

- Xây dựng nền tảng quản lý lớp học trực tuyến theo kiến trúc microservices với khả năng mở rộng cao. Phát triển hệ thống có khả năng tùy chỉnh linh hoạt theo nhu cầu khách hàng thông qua mô hình multi-tenancy.

- Tích hợp AI Agent để tự động hóa quy trình tạo thương hiệu và giao diện cá nhân hóa cho mỗi instance. Xây dựng hệ thống KiteHub để quản lý toàn bộ vòng đời sản phẩm từ bán hàng đến vận hành.

1.3. Phạm vi nghiên cứu

Đối tượng: Các tổ chức giáo dục, doanh nghiệp đào tạo, giảng viên độc lập.

Phạm vi chức năng: Quản lý lớp học, người dùng, nội dung học tập, streaming video, diễn đàn và vận hành nền tảng.

Phạm vi kỹ thuật: Áp dụng kiến trúc microservices cho KiteClass instances, modular monolith cho KiteHub, triển khai trên AWS EKS.

PHẦN 2: KIẾN TRÚC HỆ THỐNG VÀ LÝ DO LỰA CHỌN

2.1. Tổng quan kiến trúc

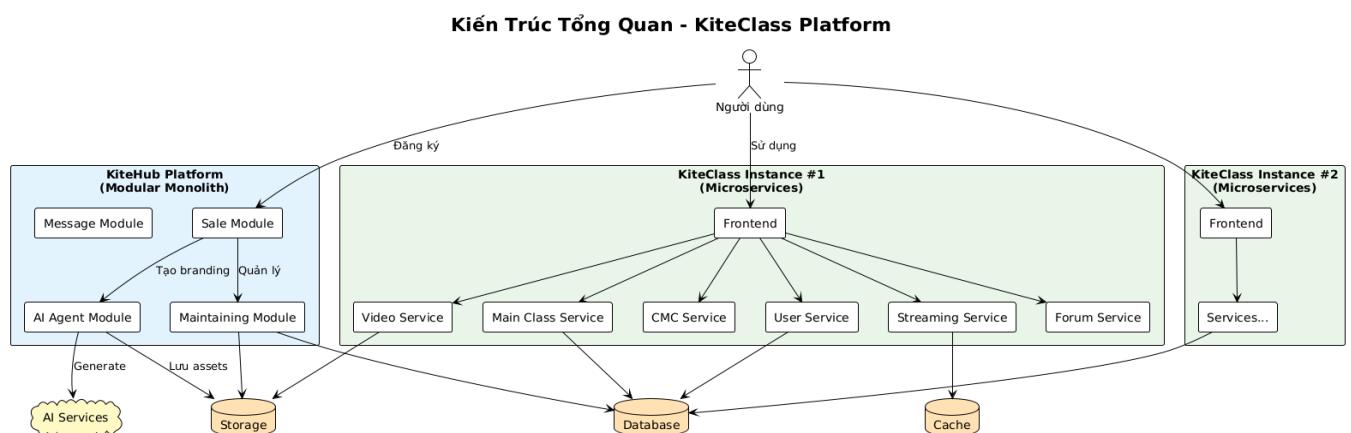
Hệ thống KiteClass Platform được thiết kế theo mô hình phân tán gồm 2 thành phần chính:

KiteHub Platform (Modular Monolith):

- Nhận vai trò trung tâm quản lý, điều phối toàn bộ hệ thống
- Được thiết kế theo kiến trúc Modular Monolith để tối ưu hiệu năng và giảm chi phí vận hành
- Bao gồm các module: Sale (bán hàng), Message (tư vấn realtime), Maintaining (quản lý instance), AI Agent (tạo nội dung)

KiteClass Instances (Microservices):

- Mỗi instance là một hệ thống quản lý lớp học độc lập
- Áp dụng kiến trúc Microservices với 7 services riêng biệt
- Services: Main Class, User, CMC, Video, Streaming, Forum, Frontend



2.2. Lý do sử dụng Microservices cho KiteClass Instance

2.2.1. Khả năng scale độc lập theo nhu cầu

Mỗi service trong KiteClass instance có thể được scale riêng biệt tùy theo tải. Ví dụ:

- Streaming Service cần scale nhiều hơn trong giờ học peak
- Video Service cần tài nguyên lớn hơn khi người dùng xem video đồng thời
- Forum Service có thể scale nhẹ hơn khi sử dụng ít

2.2.2. Linh hoạt trong lựa chọn công nghệ

Các services có thể sử dụng công nghệ phù hợp nhất:

- Streaming Service dùng Node.js để xử lý WebSocket hiệu quả
- Các services quản lý chính dùng Java Spring Boot để đảm bảo ổn định
- Frontend dùng Next.js SSR để tối ưu SEO và tốc độ tải trang

2.2.3. Dễ dàng bảo trì và nâng cấp

- Mỗi service có thể được cập nhật, sửa lỗi độc lập mà không ảnh hưởng đến các service khác
- Giảm thiểu rủi ro khi triển khai thay đổi
- Dễ dàng test và debug từng thành phần riêng biệt

2.2.4. Tăng cường độ tin cậy (Reliability)

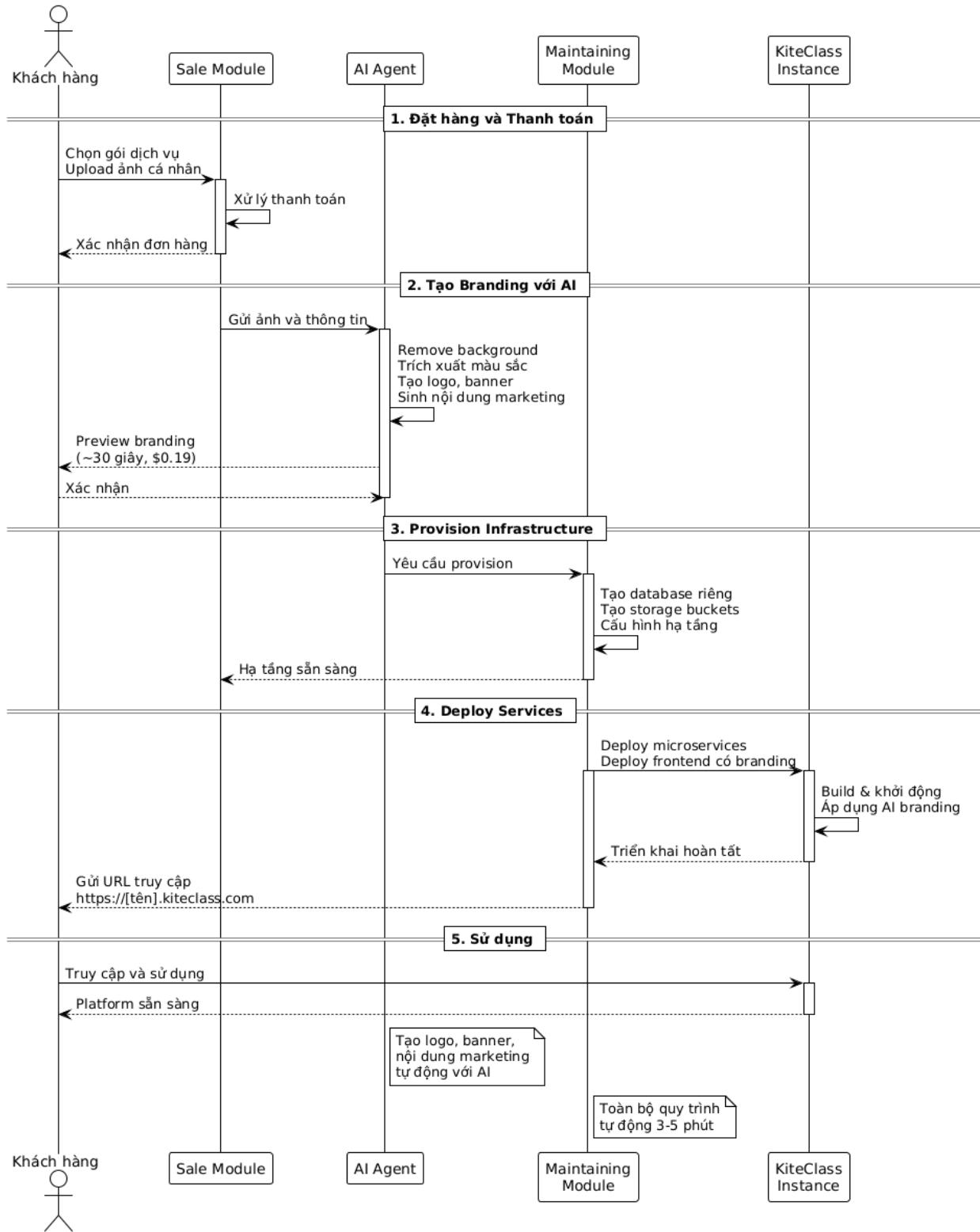
- Khi một service gặp sự cố, các service khác vẫn hoạt động bình thường
- Circuit breaker và retry mechanism giúp hệ thống chịu lỗi tốt
- Isolation giúp hạn chế tác động khi có lỗi xảy ra

2.2.5. Phù hợp với mô hình multi-tenancy

- Mỗi KiteClass instance chạy độc lập với database riêng
- Có thể tùy chỉnh giao diện, chức năng theo nhu cầu từng khách hàng
- Isolation tốt giúp đảm bảo bảo mật và hiệu năng

PHẦN 3: QUY TRÌNH MỞ NODE KITECLASS VỚI AI AGENT

Quy Trình Mở Node KiteClass



3.1. Bước 1: Khách hàng đặt hàng và cấu hình

- Khách hàng truy cập KiteHub Platform
- Chọn gói dịch vụ phù hợp (Basic, Professional, Enterprise)
- Cấu hình các thông số kỹ thuật: domain, số lượng người dùng, các service cần thiết
- Upload ảnh cá nhân/logo của tổ chức
- Thanh toán qua cổng thanh toán tích hợp

3.2. Bước 2: AI Agent tự động tạo nội dung thương hiệu

Sau khi nhận được ảnh upload, AI Agent Module tự động xử lý:

a) Remove background (sử dụng Remove.bg API):

- Loại bỏ nền ảnh để có ảnh trong suốt
- Tạo các phiên bản ảnh với các kích thước khác nhau

b) Extract primary colors:

- Phân tích ảnh để trích xuất bảng màu chủ đạo
- Tạo color palette cho giao diện

c) Generate marketing copy (sử dụng OpenAI GPT-4):

- Tạo các khẩu hiệu, slogan phù hợp
- Sinh nội dung marketing đa dạng
- Tạo mô tả cho các trang chính

d) Create visual assets (sử dụng Stable Diffusion XL):

- Tạo logo với 3 phiên bản khác nhau
- Tạo banner cho các vị trí trên website (5 kích thước)
- Tạo thumbnail và avatar

Toàn bộ quy trình AI mong muốn dự kiến sẽ mất khoảng 30 giây và chi phí 0.19 USD/instance.

3.3. Bước 3: Preview và xác nhận

- Khách hàng xem trước các tài nguyên AI đã tạo

- Có thể yêu cầu tạo lại hoặc điều chỉnh nếu cần
- Xác nhận và bắt đầu quy trình provision

3.4. Bước 4: Tự động provision infrastructure

Maintaining Module nhận request và thực hiện:

- a) Tạo database riêng (Aurora PostgreSQL Serverless v2): Khởi tạo schema database, Insert dữ liệu mặc định (roles, permissions, settings), Cấu hình auto-scaling: 0.5-4 ACU
- b) Tạo S3 buckets riêng: Bucket cho assets (logo, banner, images), Bucket cho videos bài giảng, Bucket cho uploads của người dùng
- c) Deploy các microservices lên EKS: Main Class Service (quản lý lớp học, khóa học), User Service (quản lý người dùng, xác thực), CMC Service (diễn đàn, bài tập, đánh giá), Video Service (quản lý video bài giảng), Streaming Service (phát trực tiếp), Forum Service (diễn đàn trao đổi)
- d) Deploy và build Frontend: Inject các thông tin branding từ AI-generated assets, Áp dụng color palette cho giao diện, Cấu hình routing và domain, Build và deploy Next.js application

Toàn bộ quy trình provision mong muốn dự kiến mất 3-5 phút và được tự động hóa hoàn toàn qua Terraform và Helm charts.

3.5. Bước 5: Bàn giao và kích hoạt

- Hệ thống gửi email thông báo hoàn thành
- Cung cấp URL truy cập: [https://\[custom\].kiteclass.com](https://[custom].kiteclass.com)
- Tài khoản admin mặc định cho khách hàng
- Tài liệu hướng dẫn sử dụng và quản trị

3.6. Lợi ích của quy trình tự động với AI Agent

- Tiết kiệm thời gian: Từ vài ngày xuống còn 3-5 phút
- Tiết kiệm chi phí: Không cần thuê designer (chỉ \$0.19 cho AI)
- Nhấn quan thương hiệu: AI đảm bảo các asset phù hợp với nhau
- Tùy biến cao: Mỗi instance có giao diện và thương hiệu riêng
- Chuyên nghiệp: Chất lượng tài nguyên AI tương đương designer

PHẦN 4: ĐIỂM MẠNH VÀ THỬ THÁCH CỦA ĐỀ TÀI

4.1. Áp dụng kiến trúc hiện đại và phù hợp, tích hợp AI Agent sáng tạo

- Microservices cho KiteClass instances đảm bảo linh hoạt, scale tốt
- Sử dụng AI để tự động hóa quy trình tạo nội dung, giao diện
- Kết hợp nhiều AI services: GPT-4, Stable Diffusion, Remove.bg
- Tạo giá trị thực tế: tiết kiệm thời gian và chi phí cho khách hàng
- Thể hiện khả năng tích hợp công nghệ AI vào hệ thống thực tế

4.2. Giải quyết vấn đề thực tế

- Đáp ứng nhu cầu thị trường: nhiều giảng viên, tổ chức nhỏ muốn có nền tảng riêng
- Mô hình kinh doanh rõ ràng: SaaS platform với revenue từ subscription
- Quy trình provision nhanh chóng: 3-5 phút thay vì vài ngày/tuần
- Khách hàng có thể bắt đầu dạy học ngay sau khi đăng ký

4.3. Có khả năng mở rộng cao

- Dễ dàng thêm service mới cho KiteClass instances
- Có thể scale theo chiều ngang khi số lượng instance tăng

4.4. Yêu cầu kiến thức kỹ thuật rộng

Để hoàn thành đề tài, sinh viên cần nắm vững nhiều lĩnh vực:

- Backend Development: Java Spring Boot, Node.js, RESTful API
- Frontend Development: Next.js, React, Server-Side Rendering
- Database: PostgreSQL, Redis, database design và optimization
- DevOps: Docker, Kubernetes, CI/CD, Infrastructure as Code
- Cloud Computing: AWS services (EKS, Aurora, S3, CloudFront, ALB, Route 53)
- AI Integration: OpenAI API, Stable Diffusion, image processing
- Microservices patterns: Service discovery, API Gateway, Circuit breaker

4.5. Khối lượng code và công việc lớn

- Cần phát triển nhiều services độc lập (hơn 10 services)

- Mỗi service cần code đầy đủ các layer: Controller, Service, Repository
- Phải viết test cases cho tất cả các thành phần
- Cần setup CI/CD pipeline cho từng service
- Khối lượng code ước tính: 20,000+ lines of code

4.6. Phức tạp trong triển khai và vận hành

- Phải hiểu sâu về Kubernetes để deploy và manage containers
- Cần thiết lập monitoring, logging cho toàn bộ hệ thống
- Quản lý nhiều databases và ensure data consistency
- Xử lý các vấn đề network, security trong môi trường distributed
- Debug khó khăn hơn khi lỗi xảy ra ở nhiều services

4.7. Chi phí AWS khi phát triển và test

- Cần tài khoản AWS với credit đủ để test
- Chi phí EKS cluster: ~\$70-100/tháng
- Chi phí database, storage, networking thêm
- Cần quản lý chi phí carefully để không vượt ngân sách
- Phải tối ưu resource usage trong quá trình dev/test

KẾT LUẬN

Đề tài "Xây dựng hệ thống quản lý lớp học trực tuyến theo kiến trúc Microservices - KiteClass Platform" là một đồ án tốt nghiệp có tính thực tế cao, áp dụng nhiều công nghệ hiện đại và giải quyết vấn đề cụ thể của thị trường.

Với việc kết hợp Microservices cho KiteClass instances và Modular Monolith cho KiteHub, đề tài thể hiện khả năng phân tích và lựa chọn kiến trúc phù hợp thay vì áp đặt kiến trúc một cách giáo điều. Đặc biệt, việc tích hợp AI Agent để tự động hóa quy trình tạo thương hiệu và giao diện là một điểm sáng, tạo giá trị thực tế và thể hiện khả năng ứng dụng AI vào giải quyết vấn đề thực tế.

Tuy nhiên, đề tài cũng đưa ra nhiều thử thách: yêu cầu kiến thức rộng về nhiều lĩnh vực, khối lượng code lớn, phức tạp trong triển khai, chi phí AWS và cần khảo sát người dùng kỹ lưỡng. Nhưng chính những thử thách này sẽ giúp sinh viên phát triển toàn diện cả hard skills và soft skills, chuẩn bị tốt cho công việc sau khi tốt nghiệp.

Với lợi ích rõ ràng, kiến trúc hợp lý và khả năng ứng dụng thực tế, đề tài KiteClass Platform là một lựa chọn phù hợp cho đồ án tốt nghiệp chuyên ngành Công nghệ phần mềm.