TRƯỜNG ĐẠI HỌC TIỀN GIANG

**KHOA KỸ THUẬT CÔNG NGHỆ**

-----🙞🙜🕮🙞🙜-----

****

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI**

**XÂY DỰNG ỨNG DỤNG ÔN LUYỆN**

**TRẮC NGHIỆM STUDY-SPACE**

Người hướng dẫn: Ks. Nguyễn Thái Duy

Sinh viên thực hiện:

Đặng Hữu Lộc 018101027

Nguyễn Ngọc Minh Yến 018101108

Tiền Giang, tháng 01 năm 2021

**LỜI NÓI ĐẦU**

Hiện nay, các bạn học sinh phải chuẩn bị rất nhiều tài liệu, sách vở khi đến trường, đặc biệt đối với các bạn thi chuyển cấp phải đem lượng lớn đề kiểm tra, đề thi thử ở lớp và cả lớp học thêm để thuận tiện cho việc ôn tập. Việc giữ nhiều tài liệu cho nhiều môn học khiến các bạn gặp nhiều khó khăn khi bảo quản, sắp xếp cũng như tìm kiếm chúng.

Bên cạnh đó, việc phát tài liệu học tập, đề kiểm tra tại các trường hiện nay vẫn hoạt động truyền thống thông qua việc in ấn. Hình ảnh một bạn đại diện lớp xách hai tay hai thùng giấy lớn tài liệu không còn là điều xa lạ. Điều này gây mất thời gian khi giáo viên tìm giao cho một bạn, cả lớp phải liên hệ bạn để nhận bài. Đồng thời gây bất tiện khi phải vận chuyển rất nhiều giấy tờ cùng lúc, lãng phí giấy mực và tốn nhiều sức lực.

Đồng thời giáo viên ở trường vừa dạy theo chương trình vừa tổ chức ôn tập nên rất ít thời gian để sửa hết các đề. Khi học thêm, một lớp ôn luyện có nhiều học viên, điều này dẫn đến giáo viên khó có thể chấm điểm từng bài, hay hướng dẫn cụ thể cho từng bạn. Vì vậy tự ôn luyện tại nhà luôn được các thầy cô khuyến khích.

Mặt khác, nhiều giáo viên có phần e dè khi sử dụng công nghệ, đặc biệt là tham gia giảng dạy trực tuyến như hiện nay vì khó sử dụng, khó thao tác. Dẫn đến việc hạn chế khả năng hỗ trợ các bạn học.

Thế nhưng việc ôn luyện tại nhà mang lại nhiều mặt tốt nhưng lại khó có thể tìm người hướng dẫn khi không tìm ra cách giải bài tập. Điều này khiến nhiều bạn gặp nản chí trong việc tự học.

Vì vậy, chúng tôi muốn xây dựng “ỨNG DỤNG ÔN LUYỆN TRẮC NGHIỆM STUDY-SPACE” giúp giải quyết những vấn đề trên và phát triển thêm nhiều tiện ích, mang lại không gian học tập tốt nhất cho người dùng.

**LỜI CẢM ƠN**

🙞🙞🕮🙜🙜

Chúng tôi xin gửi lời cảm ơn chân thành đến cô.. thạc sĩ Nguyễn Thái Duy VNPT IT là người hướng dẫn đồ án cho chúng tôi. Anh đã luôn theo sát quá trình thực hiện đồ án, nhiệt tình hướng dẫn, chỉ bảo, động viên để chúng tôi hoàn thành đồ án này.

Trong quá trình làm đồ án, mặc dù đã rất cố gắng trong quá trình làm đồ án, nhưng không thể tránh khỏi những thiếu sót. Rất mong nhận được sự thông cảm và góp ý tận tình của quý Thầy Cô và các bạn.

Cuối cùng chúng tôi kính chúc quý Thầy/Cô dồi dào sức khỏe và thành công trong công việc.

Xin chân thành cảm ơn!

Tiền Giang, ngày … tháng 01 năm 2021

Sinh viên thực hiện

Nhóm sinh viên thực hiện

**ĐỀ CƯƠNG CHI TIẾT KHÓA LUẬN TỐT NGHIỆP**

**Tên đề tài**: Xây dựng ứng dụng ôn luyện trắc nghiệm Study-Space

**Người hướng dẫn**: Ks. Nguyễn Thái Duy

**Sinh viên thực hiện:**

Đặng Hữu Lộc MSSV: 018101027

Nguyễn Ngọc Minh Yến MSSV: 0181101108

**Loại đề tài:**

**Nội dung đề tài:**

**Nhiệm vụ đề tài:**

**Bối cảnh:**

Ngày nay, việc làm bài, ôn bài và kiểm tra theo hình thức trắc nghiệm dần trở nên phổ biến nhất là đối tượng thi chuyển cấp cũng như sinh viên các trường. Việc giữ nhiều tài liệu cho nhiều môn học khiến các bạn gặp nhiều khó khăn khi bảo quản, sắp xếp cũng như tìm kiếm chúng. Đặc biệt, trong tình hình dịch bệnh như hiện nay, việc kiểm tra ở hầu hết trường lớp đều sử dụng phương án làm bài trắc nghiệm. Hình thức làm bài trắc nghiệm mang lại nhiều lợi ích giúp kiểm tra tổng quát kiến thức đã học, đồng thời cũng tiết kiệm rất nhiều thời gian và công sức của giáo viên trong việc chấm điểm.

Hiện có nhiều phần mềm hỗ trợ tạo bài trắc nghiệm và thống kê kết quả kiểm tra nhưng tính năng còn hạn chế, khó sử dụng, chưa thật sự phù hợp với nhiều đối tượng, tổ chức sử dụng.

Vì vậy, việc xây dựng “Phần mềm quản lý và hổ trợ tự học trắc nghiệm StudySapce” là vấn đề cấp thiết, nhằm giải quyết những vấn đề trên và phát triển thêm nhiều tiện ích, mang lại không gian học tập tốt nhất cho người dùng.

**Phân tích, thiết kế, cài đặt, thử nghiệm:**

* Thiết kế Usecase cho từng nhóm người dùng.
* Xây dựng kịch bản cho các chức năng.
* Thiết kế cơ sở dữ liệu cho hệ thống, chia thành 3 nhóm riêng “quản lý nhóm”, “quản lý đề tài”, “quản lý làm bài”.
* Xây dựng hệ thống theo kiến trúc Microservices.
* Thiết kế giao diện trực quan với các chức năng cần thiết.
* Cài đặt ứng dụng trên Docker Container để chạy thử và kiểm lỗi thường xuyên.
* Báo cáo và chỉnh sửa theo yêu cầu.

**Yêu cầu sinh viên**:

* Có kiến thức phân tích thiết kế hệ thống thông tin.
* Có kiến thức về thiết kế hướng lĩnh vực.
* Hiểu biết cơ bản về kiến trúc Microservice.
* Hiểu biết ứng dụng web lũy tiến (progressive web app).
* Có kinh nghiệm về spring framework và lập trình front end với React.
* Biết sử dụng công cụ test Cypress.
* Có kiến thức về các ngôn ngữ được sử dụng trong dự án: Java, Typescript, HTML, CSS.

**Nghiên cứu tài liệu chuyên môn:**

* Nghiên cứu thực tế: nghiên cứu mô hình Microservices thiết kế hệ thống, đồng thời kết hợp sử dụng React, Spring Boot, Spring Data R2DBC, hỗ trợ triển khai hệ thống bằng Kubernetes và hệ quản trị cơ sở dữ liệu MySql.
* Xác định phạm vi đề tài: Đề tài xây dựng nhằm mục đích phục vụ cho việc quản lý và hỗ trợ tạo bài trắc nghiệm và tự ôn tập trắc nghiệm cho các trường học, tổ chức.
* Nghiên cứu tài liệu chuyên môn: Nghiên cứu tài liệu tổng quan về thiết kế hướng lĩnh vực của Eric Evans`, Microservice từ thiết kế đến triển khai, tài liệu học tập môn cơ sở dữ liệu của trường Đại học Tiền Giang, Progressive Web App with Workbox.

**Kết quả dự kiến của đề tài:**

* Công dụng phần mềm: quản lý và hỗ trợ việc tạo đề trắc nghiệm và làm bài cho ngành giáo dục.
* Chức năng:
* Đăng nhập hệ thống.
* Quản lý tài khoản:
* Phân quyền cho tài khoản người dùng.
* Tạo tài khoản người dùng.
* Chỉnh sửa thông tin tài khoản.
* Quản lý bài làm, tự động tính điểm, thống kê kết quả.
* Quản lý nhóm học theo các môn, hạn chế quyền truy cập tài nguyên theo nhóm.
* Duyệt yêu cầu vào nhóm.
* Thêm xóa thành viên nhóm.
* Đặc điểm chính:
* Quản lý đề bài, tạo đề ngẫu nhiên từ kho câu hỏi.
* Xuất đề bài thành file word hoặc excel.
* Ứng dụng hoạt động được trên nhiều loại thiết bị.
* Hoạt động cả khi không kết nối Internet.
* Hệ thống xây dựng theo mô hình Microservices, kết nối với nhau qua Kafka.
* Yêu cầu cần đạt:
* Hệ thống xây dựng thành công bằng mô hình Microservices, đảm bảo tính phân quyền và bảo mật, hoạt động ổn định.
* Phần mềm đầy đủ chức năng theo yêu cầu.
* Giao diện hệ thống đơn giản, dễ sử dụng cho mọi đối tượng sử dụng, bố cục hợp lí, các menu chức năng rõ ràng.
* Ứng dụng có thể sử dụng trực tuyến cũng như không cần kết nối Internet.
* Ứng dụng chạy tốt trên cả thiết bị di động và máy tính cá nhân.

**Yêu cầu khoa học công nghệ:**

Cần cài đặt Java OpenJDK phiên bản 11, Nodejs và NPM phiên bản LTS 14., Intellij IDEA Community, Visual Studio Code. Docker và docker-compose(cài đặt Docker Desktop khi dùng Win10), Kafka Desktop client Conduktor.

**Kế hoạch thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Các nội dung, công việc cần thực hiện** | **Từ ngày** | **Đến ngày** |
| **1** | Phân tích đề tài | 01/04/2021 | 15/04/2021 |
| **2** | Thiết kế CSDL | 12/04/2021 | 28/04/2021 |
| **3** | Lập trình Back end | 27/04/2021 | 15/06/2021 |
| **4** | Lập trình Front end | 17/05/2021 | 15/06/2021 |
| **5** | Thiết kế các báo cáo | 02/06/2021 | 08/06/2021 |
| **6** | Chạy thử và sửa lỗi | 08/06/2021 | 15/06/2021 |
| **7** | Viết báo cáo và sửa lỗi đề tài | 14/06/2021 | 21/06/20 |

**Xác nhận của người hướng dẫn Ngày tháng 01 năm 2021**

**Nhóm sinh viên thực hiện**

Nguyễn Thái DuyĐặng Hữu Lộc

Nguyễn Ngọc Minh Yến

**NHẬN XÉT CỦA CÁN BỘ HƯỚNG DẨN**

**NHẬN XÉT CỦA HỘI ĐỒNG ĐÁNH GIÁ**

MỤC LỤC

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 12](#_Toc79006920)

[1.1. Lý do chọn đề tài 12](#_Toc79006921)

[1.2. Mục tiêu đề tài 12](#_Toc79006922)

[1.3. Nhiệm vụ chính đề tài 12](#_Toc79006923)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THỰC TIỄN ĐỀ TÀI 13](#_Toc79006924)

[2.1. Cơ sở lý thuyết 13](#_Toc79006925)

[2.2. Cơ sở thực tiễn 13](#_Toc79006926)

[2.2.1. Mục tiêu, yêu cầu của hệ thống: 13](#_Toc79006927)

[2.2.2. Hệ thống hiện tại: 13](#_Toc79006928)

[CHƯƠNG 3: PHÂN TÍCH, THIẾT KẾ CÀI ĐẶT GIẢI PHÁP 14](#_Toc79006929)

[3.1. Mô tả công việc nghiên cứu: 14](#_Toc79006930)

[3.1.1. Quy trình nghiệp vụ: 14](#_Toc79006931)

[3.1.2. Kiến trúc hệ thống: 14](#_Toc79006932)

[3.1.3. Sơ đồ chức năng: 14](#_Toc79006933)

[3.1.4. Sơ đồ UseCase: 14](#_Toc79006934)

[3.1.5. Sơ đồ mô hình quan niệm ở mức dữ liệu (Conceptual Data Model) 14](#_Toc79006935)

[3.1.6. Sơ đồ mô hình quan niệm ở mức vật lý (Physical Data Model) 14](#_Toc79006936)

[3.2. Quy trình thực hiện: 14](#_Toc79006937)

[3.3. Kết quả thực hiện: 14](#_Toc79006938)

[CHƯƠNG 4: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN 15](#_Toc79006939)

[TÀI LIỆU THAM KHẢO 16](#_Toc79006940)

**DANH MỤC HÌNH**

[Hình 1 Sơ đồ usecase 12](#_Toc74596115)

[Hình 2 Sơ đồ lớp của ứng dụng 13](#_Toc74596116)

[Hình 3 Giao diện đăng nhập 18](#_Toc74596117)

[Hình 4 Giao diện chính 19](#_Toc74596118)

[Hình 5 Giao diện quản lý sản phẩm 19](#_Toc74596119)

[Hình 6 Giao diện tin nhắn 20](#_Toc74596120)

[Hình 7 Giao diện lấy tọa độ khách hàng 20](#_Toc74596121)

[Hình 8 Xử lý khi nhấn button login và button register 21](#_Toc74596122)

[Hình 9 Code xử lý hiển thị một danh sách ra RecycleView 21](#_Toc74596123)

[Hình 10 Code xử lý chức năng đặt hàng 22](#_Toc74596124)

[Hình 11 Code xử lý chức năng gửi message 22](#_Toc74596125)

[Hình 12 Code xử lý chức năng đọc danh sách tin nhắn từ database ra giao diện 23](#_Toc74596126)

**DANH MỤC BẢNG**

[Bảng 1 Usecase đăng ký 13](#_Toc74522104)

[Bảng 2 Usecase đăng nhập 13](#_Toc74522105)

[Bảng 3 Usecase cập nhật thông tin cá nhân 14](#_Toc74522106)

[Bảng 4 Usecase quản lý sản phẩm 14](#_Toc74522107)

[Bảng 5 Usecase thêm sản phẩm 15](#_Toc74522108)

[Bảng 6 Usecase quản lý đơn đặt hàng 15](#_Toc74522109)

[Bảng 7 Usecase đăng xuất 16](#_Toc74522110)

[Bảng 8 Usecase tìm kiếm và thêm sản phẩm vào giỏ hàng 16](#_Toc74522111)

[Bảng 9 Usecase thanh toán 17](#_Toc74522112)

[Bảng 10 Usecase nhắn tin 17](#_Toc74522113)

# TÓM TẮT

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

### **Lý do chọn đề tài**

Việc tự học luôn mang lại hiệu quả cao cho quá trình trau dồi, rèn luyện và ôn tập kiến thức. Bên cạnh đó người dùng Internet ngày càng nhiều, việc sử dụng các ứng dụng hỗ trợ trong học tập cũng được mọi người quan tâm hơn, giúp người dùng dễ dàng tiếp cận, tổng hợp kiến thức và tra cứu thông tin một cách nhanh chóng.

Một ứng dụng tích hợp nhiều tiện ích tạo không gian học tập bổ ích, tiết kiệm tối đa thời gian, mang lại hiệu quả cao cho người dùng. Hệ thống luôn được cập nhật, có độ chính xác cao là yếu tố quan trọng trong việc xây dựng ứng dụng ôn tập.

Hiện nay, xu hướng học tập online trên các ứng dụng ngày càng phổ biến đặc biệt là trong tình hình dịch bệnh Covid 19, việc giảng dạy, học tập tại lớp trở nên khó khăn hơn. Chính vì thế, xây dựng ứng dụng nhằm hỗ trợ dạy và học trực tuyến cũng như các ứng dụng giúp các bạn ôn luyện là vấn đề cấp thiết và đang được quan tâm hiện nay.

### **Mục tiêu đề tài**

Đề tài được thực hiện nhằm xây dựng thành công ứng dụng hoạt động tốt, giao diện thân thiện, tích hợp đầy đủ tính năng phù hợp nhu cầu người dùng, mọi đối tượng sử dụng.

### **Nhiệm vụ chính của đề tài**

Nghiên cứu thực tế: nghiên cứu mô hình Microservices thiết kế hệ thống, đồng thời kết hợp sử dụng React, Spring Boot, Spring Data R2DBC, hỗ trợ triển khai hệ thống bằng Kubernetes và hệ quản trị cơ sở dữ liệu MySql.

Xác định phạm vi đề tài: xây dựng ứng dụng đầy đủ môn học, tính năng cơ bản cho giáo viên, các bạn học sinh trong việc tạo đề và ôn luyện trắc nghiệm.

Nghiên cứu tài liệu chuyên môn:

* Tổng quan về Microservices.
* Cách thức hoạt động Kubernates.
* Kiến thức cơ bản về React, Spring Boot.
* Giới thiệu về Spring Data Reactive Relational Database Connectivity.
  1. **Đối tượng và phạm vi nghiên cứu**

1. **Đối tượng nghiên cứu:**

Về nghiệp vụ:

Nghiên cứu tài liệu, cấu trúc các bài giảng, dạng kiểm tra của các trường.

Về công nghệ:

Tìm hiểu kiến trúc Microservices và cách xây dựng trang web microservices với Jhipster.

Rest API, xây dựng API bằng Spring Boot;

Ứng dụng Single page Applo]ication và xây dựng ứng dụng bằng ReactJS;

Cách lưu trữ dữ liệu với MySQL và cách Mapping các table trong cơ sở dữ lie65y sang các Object trog java với Hibernate để đơn giản hơn trong việc truy vấn dữ liệu.

Về kỹ thuật:

Phân tích yêu cầu, phân tích hệ thống.

Xây dựng một hệ thống hoàn chỉnh từ cả 2 khía cạnh cilent và server.

Sử dụng các công cụ hỗ trợ lập rình.

1. **Phạm vi nghiên cứu:**

Nghiên cứu các hệ thống học trực tuyến hiện nay, tham khảo các thức dạy và học từ các trường từ đó đúc kết thiếu sót để xây dựng hệ thống hoàn chỉnh, đáp ứng đầy đủ các tính năng, giao diện dễ sử dụng phù hợp mọi đối tượng sử dụng.

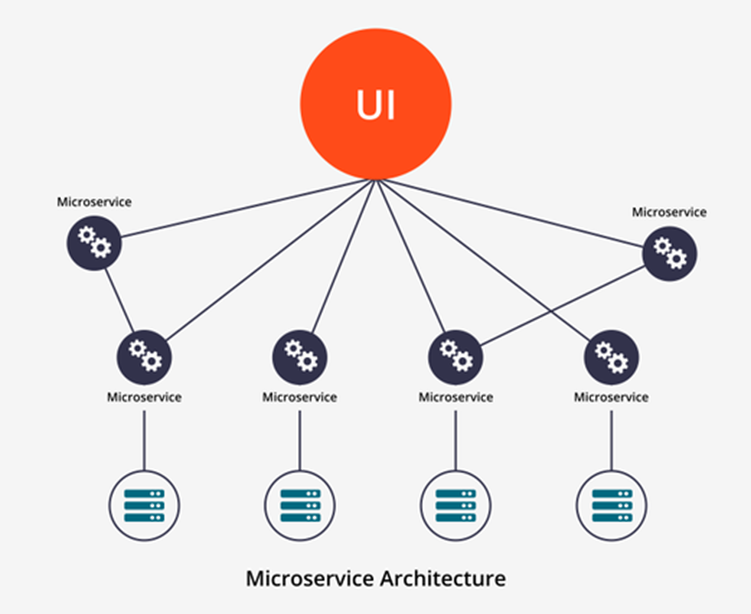
* 1. **Yêu cầu cần đạt:**
* Hệ thống xây dựng thành công bằng mô hình Microservices, đảm bảo tính phân quyền và bảo mật, hoạt động ổn định.
* Giao diện hệ thống đơn giản, dễ sử dụng cho mọi đối tượng sử dụng, bố cục hợp lí, các menu chức năng rõ ràng.
* Ứng dụng có thể sử dụng trực tuyến cũng như không cần kết nối Internet.
* Ứng dụng chạy tốt trên mọi trình duyệt web (Google Chorme, Edge, Firefox, …).

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THỰC TIỄN ĐỀ TÀI

## **Cơ sở lý thuyết**

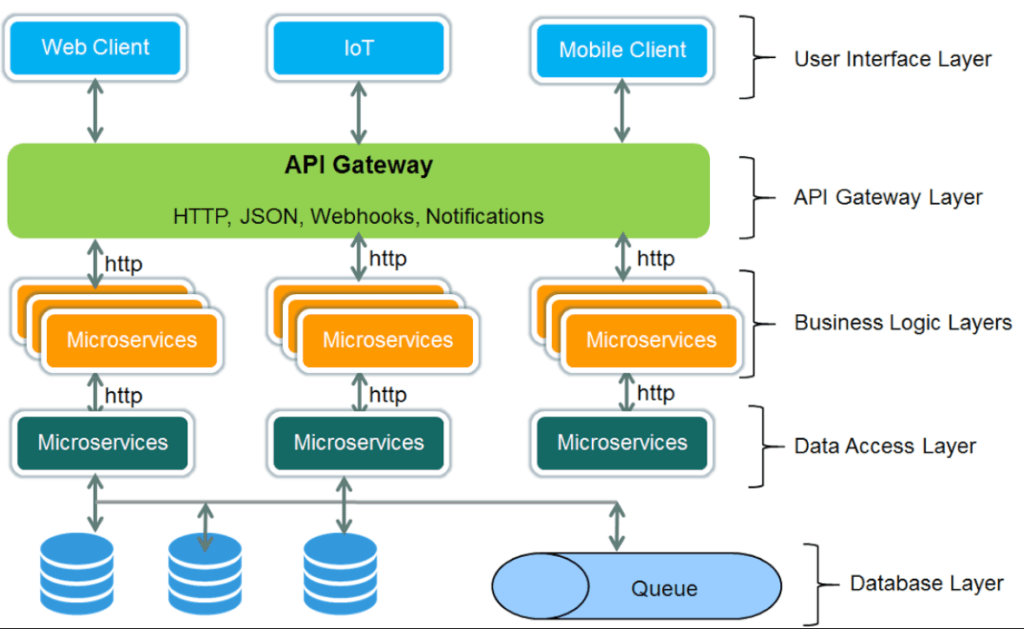
### **Kiến trúc Microservice**

Microservice là một kiến trúc phần mềm. Các module trong phần mềm này được chia thành các service rất nhỏ (microservice). Mỗi service sẽ được đặt trên một server riêng -> dễ dàng để nâng cấp và scale ứng dụng.



**Kiến trúc:**

Khác biệt với kiến trúc của Monolith, thay vì phải gom tất cả module thành một khối (monolith). Ở Microservices ta tách các module thành những service siêu nhỏ. Mỗi service sẽ được đặt trên một server riêng và thực hiện giao tiếp với nhau thông qua mạng.



**Đặc điểm của Microservices:**

Decoupling – Các service trong một hệ thống phần lớn được tách rời. Vì vậy, toàn bộ ứng dụng có thể dễ dàng được xây dựng, thay đổi và thu nhỏ.

Componentization – Microservices được coi là các thành phần độc lập có thể dễ dàng thay thế và nâng cấp.

Business Capabilities – mỗi một thành phần trong kiến trúc microservice rất đơn giản và tập trung vào một nhiệm vụ duy nhất.

Autonomy – các lập trình viên hay các nhóm có thể làm việc độc lập với nhau trong quá trình phát triển.

Continous Delivery – Cho phép phát hành phần mềm thường xuyên, liên tục.

Responsibility .

Decentralized Governance – không có mẫu chuẩn hóa hoặc bất kỳ mẫu công nghệ nào. Được tự do lựa chọn các công cụ hữu ích tốt nhất để có thể giải quyết vấn đề.

Agility – microservice hỗ trợ phát triển theo mô hình Agile.

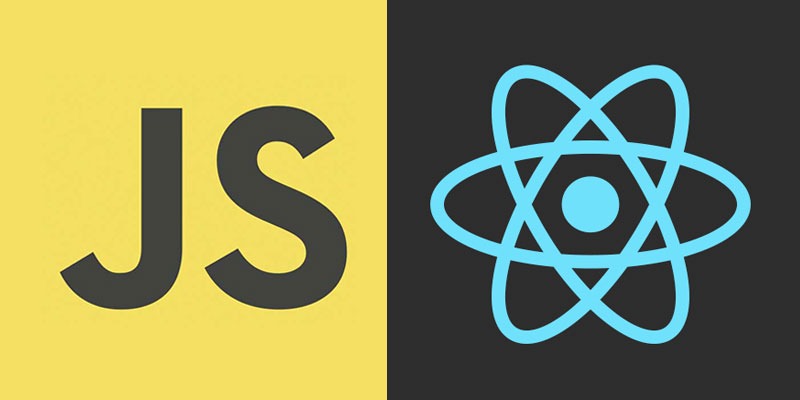
Dựa trên việc chia nhỏ hệ thống thành nhiều service riêng biệt, kiến trúc microservice có những **ưu điểm** sau:

* **Các service có thể được bảo trì độc lập**: Các service là riêng biệt, vì vậy khi có một service được bảo trì nó không làm ảnh hưởng tới các service khác.
* **Dễ dàng nắm bắt các business logic**: Mỗi service xử lý một nghiệp vụ riêng, vì vậy mà ta có thể dễ dàng nắm bắt được các business logic.
* **Áp dụng được nhiều công nghệ**: Mỗi service được coi như một dự án riêng, vì vậy mà chúng có thể sử dụng các công nghệ (như ngôn ngữ lập trình, database) khác nhau.
* **Dễ dàng mở rộng khi hệ thống trở nên phức tạp**: Microservice có khả năng mở rộng theo chiều ngang, chỉ việc bổ sung thêm service.
* Giảm thiểu được rủi ro: lỗi trong một service thì chỉ có service đó bị ảnh hưởng. Các services khác sẽ vẫn tiếp tục xử lý các yêu cầu.
* Dễ dàng thay đổi để sử dụng các công nghệ mới: Khi thực hiện triển khai các services bạn có thể lựa chọn nhiều công nghệ mới phù hợp.

**Hạn chế**:

* **Microservice khuyến khích làm nhỏ gọn các service, nhưng việc chia nhỏ các service dẫn đến khó kiểm soát**.
* **Phải đảm bảo giao dịch phân tán (distributed transaction)** .
* **Testing một dịch vụ trong kiến trúc microservices đôi khi yêu cầu phải chạy cả các dịch vụ nhỏ khác mà nó phụ thuộc**
* **Việc triển khai microservices phức tạp hơn rất nhiều nếu làm thủ công theo cách đã làm với ứng dụng monolithic**.
  + 1. **ReactJS**

**ReactJS** là một thư viện JavaScript được phát triển bởi Facebook vào năm 2013. Nó được dùng để xây dựng các tương tác với các thành phần trên website, mang lại hiệu quả và linh hoạt trong xây dựng các thành phần giao diện người dùng (UI) có thể sử dụng lại. **ReactJS** giúp phân chia các UI phức tạp thành các thành phần nhỏ.



Ưu điểm:

Ngoài việc hỗ trợ xây dựng giao diện nhanh, hạn chế lỗi trong quá trình code, cải thiện performance website thì những ReactJS còn mang lại nhiều tính năng đặc biệt dưới đây:

**Phù hợp với đa dạng thể loại website:** ReactJS khiến cho việc khởi tạo website dễ dàng hơn và nó đã cung cấp nhiều tính năng để có thể dùng cho nhiều trường hợp.

**Biến các thành phần UI phức tạp trở thành những component độc lập:** Có thể thay đổi trạng thái cho nhiều component con và không gây ảnh hưởng đến component gốc dù đang ở trong đang ở trạng thái Stateful.

**Tái sử dụng các Component**: Các component được tạo ra tương ứng với giao diện có thể kết hợp hoặc tái sử dụng một cách thuận tiện.

**Có thể sử dụng cho cả Mobile application:** sử dụng thêm framework React Native sẽ dễ dàng chia sẻ các Component khi phát triển thêm ứng dụng Mobile

**Debug dễ dàng:** Chrome extension dùng trong việc debug trong quá trình phát triển ứng dụng. Điều đó giúp tăng tốc quá trình release sản phẩm cung như quá trình code.

**Hiệu suất tốt hơn với Virtual DOM:** Khả năng xác định chính xác khi nào cần render lại cũng như khi nào bỏ đi phần tử DOM điều này giúp ứng dụng tránh tình trạng bị gián đoạn. Người dùng sử dụng liên tục, website ổn định giúp doanh thu duy trì và tăng trưởng vượt bậc.

* + 1. **Spring Boot**
* Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh.
* Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring.
* Spring Boot không yêu cầu cấu hình XML
* Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho developer.
* **Ưu điểm của SpringBoot:**

+ Có các tính năng của Spring Framework.

+ Tạo ứng dụng độc lập, có thể chạy bằng java -jar (cho cả java web)

+ Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) do đó không cần phải triển khai file WAR

+ Cấu hình ít, tự động cậu hình bất kì khi nào có thể (Giảm thời gian viết code, tăng năng suất)

+ Không yêu cầu XML config…

+ Cung cấp nhiều plugin

+ Chuẩn cho Microservices (Cloud support; giảm việc setup, config; các thư viện hỗ trợ…)

**Nhược điểm:**

Thiếu kiểm soát. Do style cố định, Spring Boot tạo ra nhiều phụ thuộc không được sử dụng dẫn đến kích thước tệp triển khai lớn.

Quá trình chuyển đổi dự án Spring cũ hoặc hiện có thành các ứng dụng Spring Boot nhiều khó khăn và tốn thời gian.

Không thích hợp cho các dự án quy mô lớn. Hoạt động liên tục với các microservices, theo nhiều nhà phát triển, Spring Boot không phù hợp để xây dựng các ứng dụng nguyên khối.

* + 1. **Spring WebFlux**

Spring Webflux là một framework cho việc phát triển các ứng dụng Web có từ Spring 5. So với Spring MVC thì Spring Webflux cải tiến hơn và mang lại nhiều lợi ích cùng với phong cách lập trình hiện đại.

* + 1. **Spring Data** **Reactive Relational Database Connectivity (R2DBC)**

Spring Data R2DBC là một phần trong Spring Data, giúp dễ dàng triển khai các kho lưu trữ.

Spring Data R2DBC là một spec định nghĩa cách thức sẽ làm việc với các relational database như MySQL, PostgreSQL, … theo cơ chế Reactive. Nó cung cấp bộ thư viện [Service Provider Interface](https://github.com/r2dbc/r2dbc-spi) (SPI) giúp có thể implement driver cho từng loại database system tương ứng.

Spring Data R2DBC không cung cấp bộ nhớ đệm, ghi đè hoặc nhiều tính năng khác của ORM framework vì thế R2DBC trở thành ánh xạ đơn giản

* + 1. **Progressive Web App (PWA)**

PWA được giới thiệu lần đầu bởi **Google** vào năm 2015, với mục đích mang lại thật nhiều lợi ích cho cả người dùng và các nhà phát triển.

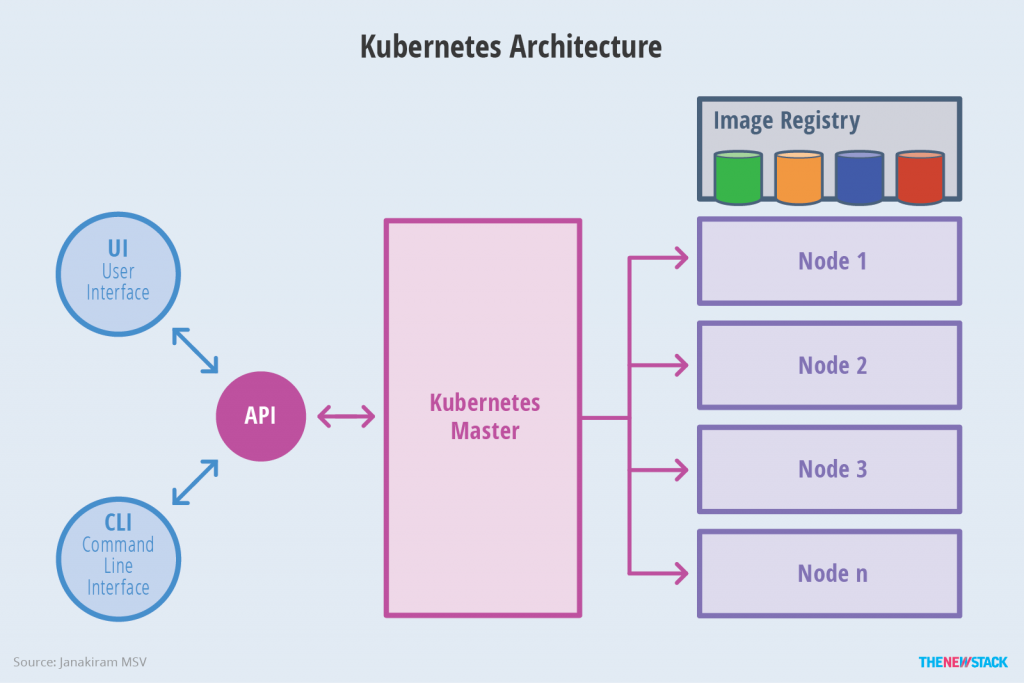
Một ứng dụng **PWA** có thể cung cấp các tính năng bổ sung dựa trên thiết bị hỗ trợ, cung cấp khả năng ngoại tuyến (**offline**), đẩy thông báo (**push notification**), giao diện và tốc độ tương đương ứng dụng **Native** và lưu trữ cục bộ các nguồn tài nguyên (**local caching**).

Mục đích sử dụng Progressive Web App là để mang lại cho người dùng sự trải nghiệm mới có cách thức hoạt động giống như là một ứng dụng native trên điện thoại cài từ AppStore hay PlayStore.

**Ưu điểm:**

* Be reliable: Lần đầu tải trang nhanh hơn, hỗ trợ chạy app ngay cả khi offline
* Fast: Phản hồi lại các hành động của người dùng trên app trở lên nhanh chóng
* Engaging: Cung cấp trải nghiệm tốt hơn khi hoạt động của app trở nên giống native app
* Đồng bộ dữ liệu khi app ở trạng thái background.
  + 1. **Kubernates**

Kubernetes là một nền tảng nguồn mở, khả chuyển, có thể mở rộng để quản lý các ứng dụng được đóng gói và các service, giúp thuận lợi trong việc cấu hình và tự động hoá việc triển khai ứng dụng. Kubernetes là một hệ sinh thái lớn và phát triển nhanh chóng. Các dịch vụ, sự hỗ trợ và công cụ có sẵn rộng rãi.



Kubernetes cung cấp:

* **Service discovery và cân bằng tải**: Nếu lượng traffic truy cập đến một container cao, Kubernetes có thể cân bằng tải và phân phối lưu lượng mạng (network traffic) để việc triển khai được ổn định.
* **Điều phối bộ nhớ**: Kubernetes cho phép tự động mount một hệ thống lưu trữ mà bạn chọn, như local storages, public cloud providers, v.v.
* **Tự động rollouts và rollbacks**: Có thể tự động hoá Kubernetes để tạo mới các container cho việc triển khai, xoá các container hiện có và áp dụng tất cả các resource của chúng vào container mới.
* **Đóng gói tự động**  
  Kubernetes có thể điều phối các container đến các node để tận dụng tốt nhất các resource.
* **Tự phục hồi**  
  Kubernetes khởi động lại các containers bị lỗi, thay thế các container, xoá các container không phản hồi lại cấu hình health check do người dùng xác định và không cho các client biết đến chúng cho đến khi chúng sẵn sàng hoạt động.
* **Quản lý cấu hình và bảo mật**  
  Kubernetes cho phép lưu trữ và quản lý các thông tin nhạy cảm như: password, OAuth token và SSH key.

## **Cơ sở thực tiễn**

### **Mục tiêu, yêu cầu của hệ thống:**

Hiện nay với tình hình dịch bệnh diễn biến phức tạp cùng với xu hướng tự học bằng các ứng dụng ngày càng phổ biến. Thế nên, đề tài “Xây dựng ứng dựng ôn luyện trắc nghiệm Study-Space” được xây dựng và triển khai nhằm tạo ra ứng dụng tốt, phù hợp nhu cầu người dùng. Ứng dụng là cầu nối giữa giáo viên và học viên, đảm bảo tính chính xác, thiết thực.

* Về mặt tổng quan, hệ thống hướng tới mục tiêu:
* Giao diện trực quan, dễ sử dụng, phù hợp mọi đối tượng.
* Đáp ứng đầy đủ chức năng.
* Đảm bảo tốt khả năng bảo mật và phân quyền.
* Dễ dàng bảo trì, nâng cấp, mở rộng hệ thống.
* Đáp ứng nhu cầu hiệu năng, hoạt động hiệu quả, nhanh chóng.
* Hoạt động tốt khi online, offline.
* Về phạm vi chức năng, hệ thống xây dựng các chức năng như sau:
* Đăng kí.
* Đăng nhập tài khoản người dùng.
* Chỉnh sửa tài khoản cá nhân (tên và mật khẩu người dùng).
* Tạo đề kiểm tra trắc nghiệm.
* Tạo cuộc thi nhóm.
* Làm bài trắc nghiệm.
* Tính điểm tự động.
* Quản lý nhóm người dùng.
* Trao đổi với các thành viên.
* Tạo ghi chú.
* Tạo nhắc nhở làm bài.

# CHƯƠNG 3: PHÂN TÍCH, THIẾT KẾ CÀI ĐẶT GIẢI PHÁP

# Mô tả công việc nghiên cứu:

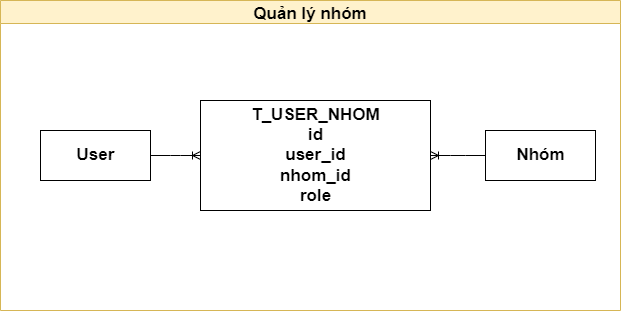
Các thành phần sử dụng hệ thống được phân bổ chức năng như sau:

* Người quản trị hệ thống: quản lý người dùng.
* Admin nhóm: phê duyệt người dùng vào nhóm, xóa học viên khỏi nhóm, tạo đề kiểm tra, tạo cuộc thi thử, trao đổi, xem danh sách điểm thi nhóm.
* Người dùng: làm bài kiểm thuộc phạm vi, tự tạo đề trắc nghiệm ôn tập, ghi chú, tạo nhắc nhở, trao đổi.

Ngoài ra, admin nhóm và người dùng đều có chức năng đăng kí, đăng nhập, chỉnh sửa thông tin tài khoản.

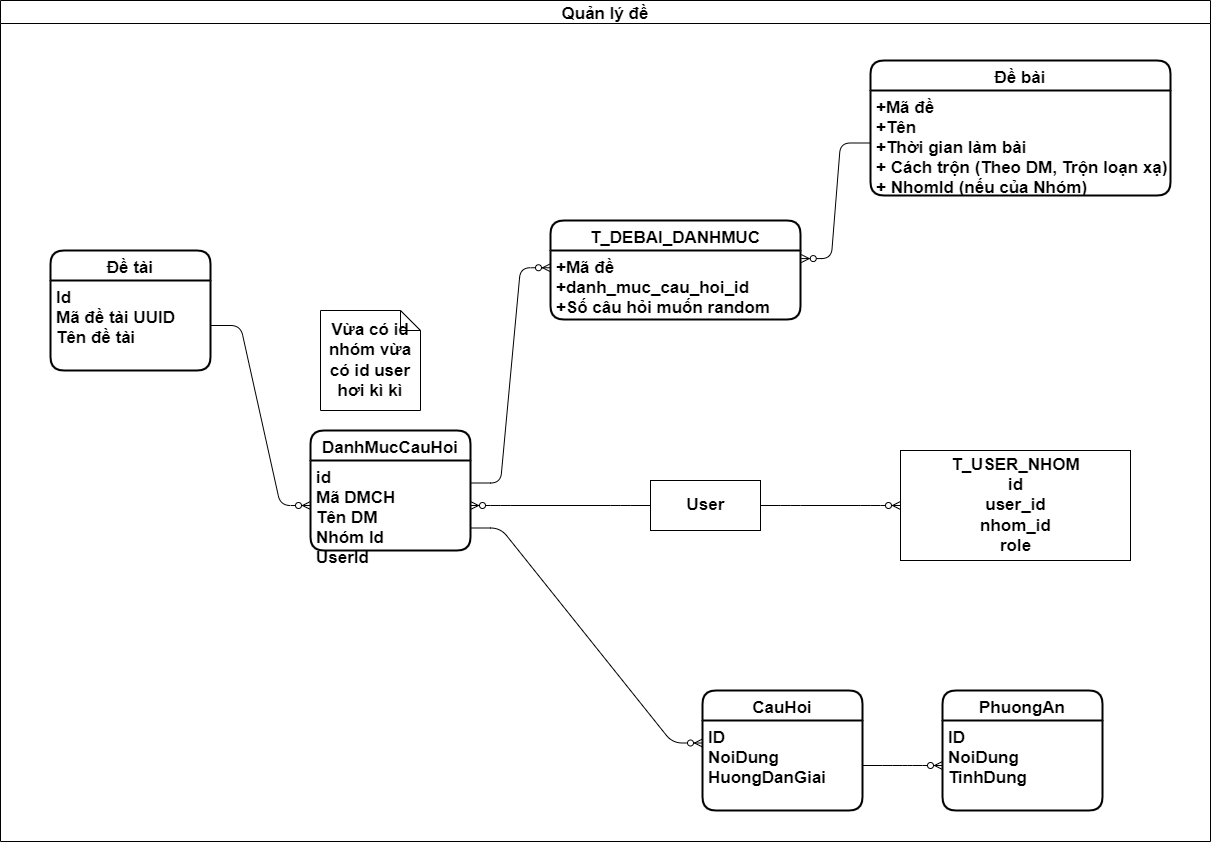
### **Kiến trúc hệ thống: …**

### **Sơ đồ chức năng:**

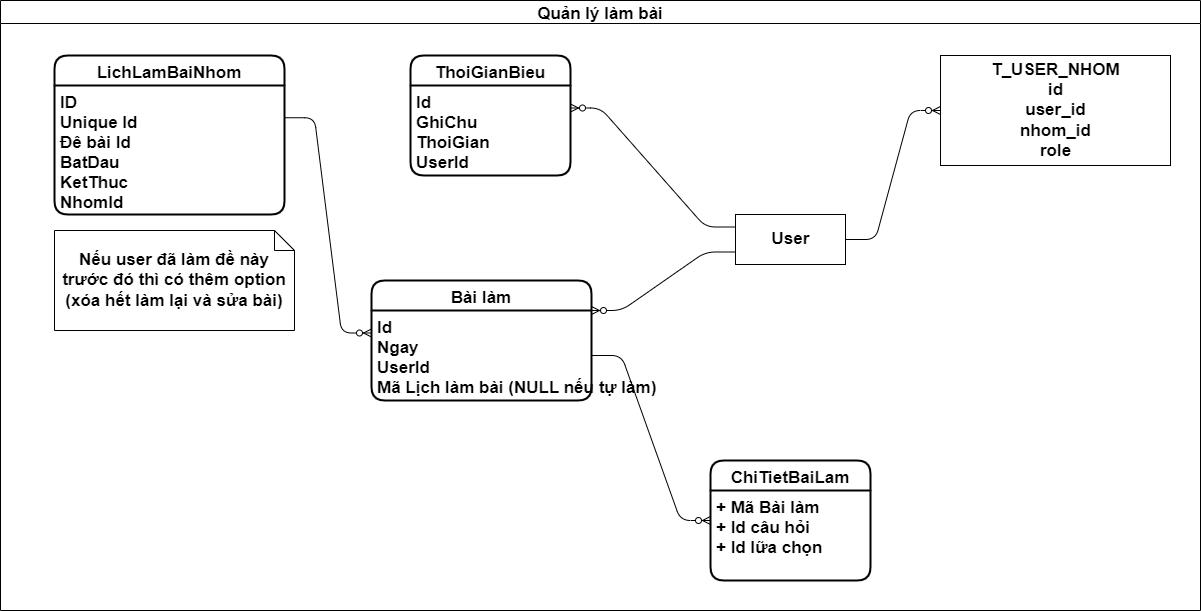


Quản lý nhóm có chức năng sau:

* Tạo nhóm.
* Phân quyền admin cho trưởng nhóm.
* Thêm, xóa thành viên nhóm.
* Cập nhật thông tin nhóm.
* Lấy danh sách thành viên của nhóm.
* Tìm kiếm thành viên có trong nhóm.
* Xóa nhóm.



* Tạo danh mục câu hỏi
* Tạo câu hỏi
* Tạo đề tài mới
* Tạo đề kiểm tra
* Thêm, sửa, xóa thông tin câu hỏi.
* Thêm, xóa, sửa đáp án câu hỏi.
* Thêm, sửa, xóa bài kiểm tra.
* Kiểm tra kết quả làm bài



Thêm, xóa, sửa lịch làm bài cho nhóm

Thêm, xóa, sửa nhắc nhở cho user

Tạo bài làm

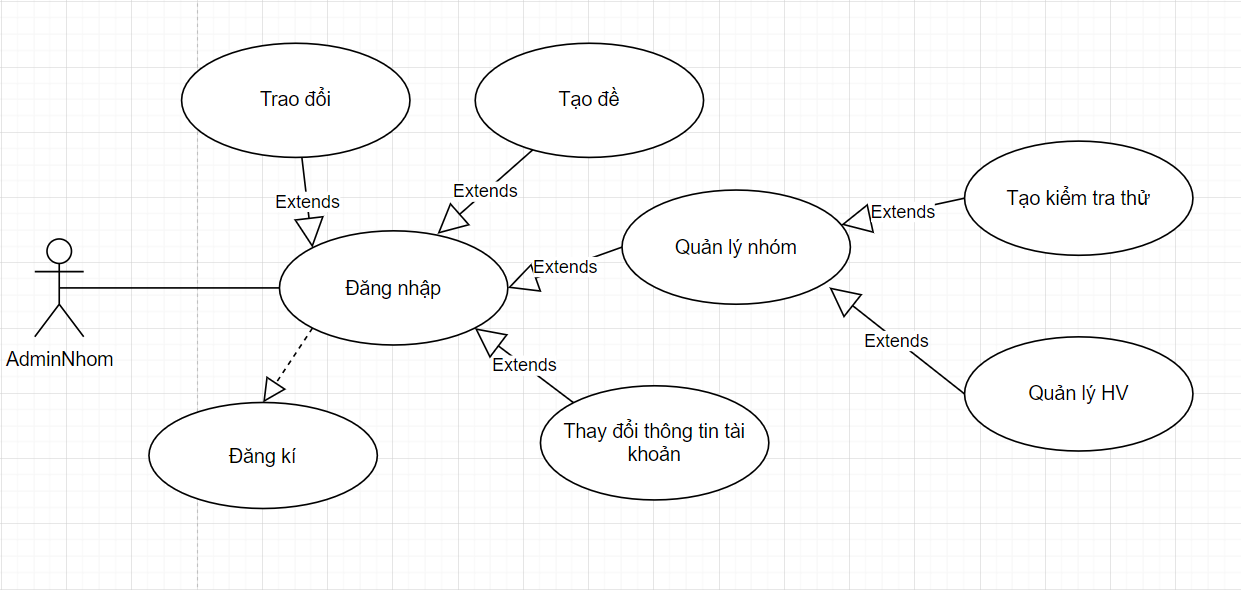
Tạo 1 lựa chọn trong bài làm

Xem chi tiết bài làm

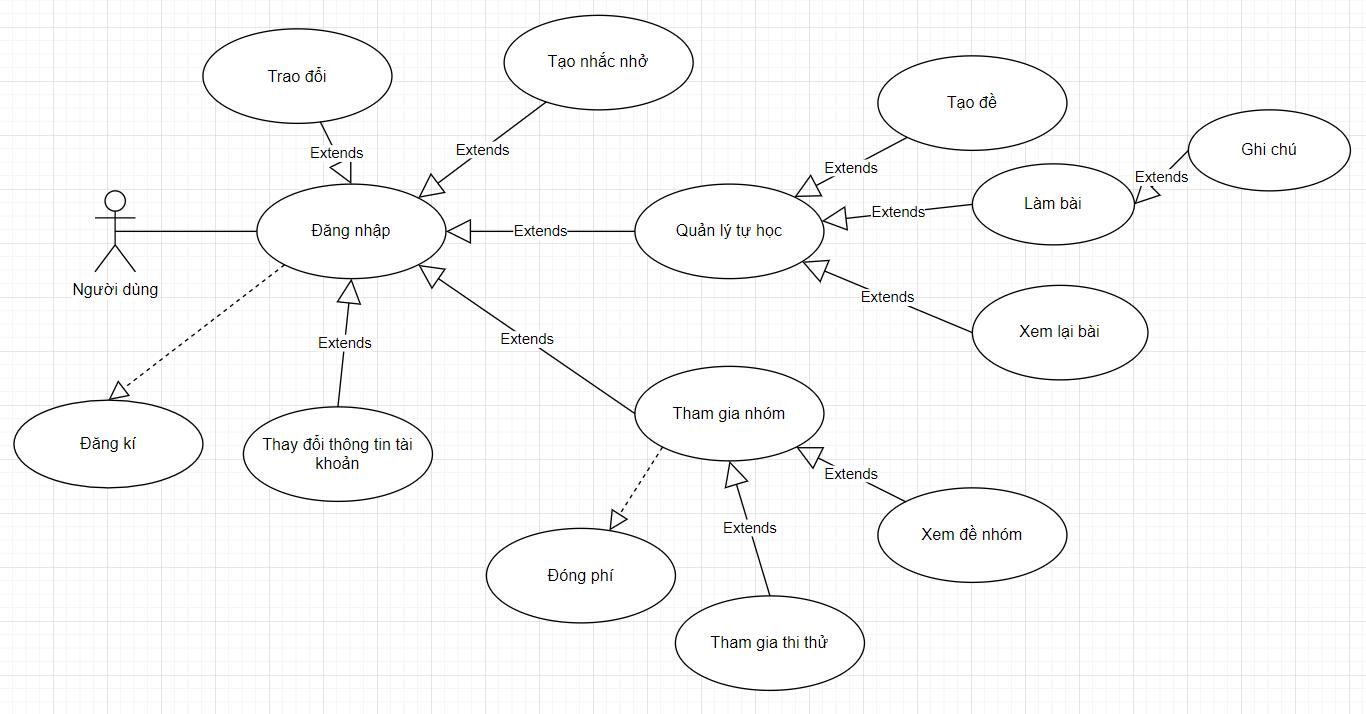
Xem tất cả bài làm trong lịch làm nhóm

### **Sơ đồ UseCase:**

1. **Usecase từng thành phần**



Hình Sơ đồ Usecase quản trị viên nhóm



Hình Sơ đồ Usecse người dùng

1. **Mô tả Usecase**

Usecase Đăng kí

|  |  |
| --- | --- |
| Actor | Tất cả actor trong hệ thống |
| Điều kiện | Không có |
| Các bước thực hiện | 1. Chọn nút “Đăng kí” ở trang chủ. 2. Hệ thống hiển thị giao diện đăng kí. 3. Người dùng nhập thông tin. 4. Chọn “Đăng kí”. 5. Hệ thống kiểm tra thông tin   Nếu hợp lệ, hệ thống thông báo tạo tài khoản thành công.  Nếu không hợp lệ, hệ thống báo đỏ nơi lỗi, yêu cầu nhập lại. |

Usecase Đăng nhập

|  |  |
| --- | --- |
| Actor | Tất cả actor trong hệ thống |
| Điều kiện | Đã tạo tài khoản thành công |
| Các bước thực hiện | 1. Vào giao diện đăng nhập của hệ thống. 2. Nhập tên tài khoản và mật khẩu. 3. Hệ thống kiểm tra tên giao diện và mật khẩu   Nếu nhập sai tên tài khoản hoặc mật khẩu, hệ thống yêu cầu nhập lại.  Nếu nhập đúng, hệ thống hiển thị giao diện trang chủ. |

Usecase Tạo đề trắc nghiệm

|  |  |
| --- | --- |
| Actor | Quản trị viên nhóm |
| Điều kiện | Đã truy cập vào nhóm |
| Các bước thực hiện | 1. Chọn “Tạo đề”. 2. Hệ thống hiển thị giao diện tạo đề. 3. Người dùng chọn nút “Thêm câu hỏi”. 4. Nhập nội dung câu hỏi. 5. Tạo đáp án cho câu hỏi và đánh dấu đáp án đúng.   Nếu chưa đánh dấu đáp án đúng, hệ thống yêu cầu chọn.   1. Nhập lời giải thích nếu có. 2. Chọn “Lưu”. 3. Hệ thống kiểm tra các mục   Nhập đủ thông tin, hệ thống hiển thị giao diện câu hỏi lưu thành công.  Nhập thiếu. hệ thống báo đỏ nơi nhập thiếu. |

Usecase Tạo cuộc thi nhóm

|  |  |
| --- | --- |
| Actor | Quản trị viên nhóm |
| Điều kiện | Đã truy cập vào nhóm |
| Các bước thực hiện | 1. Chọn “ Tạo cuộc thi”. 2. Hệ thống hiển thị giao diện tạo cuộc thi. 3. Người dùng chọn thời gian.   Chọn thời gian không hợp lệ, hệ thống yêu cầu chọn lại.   1. Chọn đề cho cuộc thi từ các đề đã tạo trong nhóm. 2. Chọn “Lưu”. 3. Hệ thống kiểm tra   Nhập đầy đủ thông tin, hệ thống lưu thành công, gửi thông báo lên nhóm.  Nhập thiếu, hệ thống báo đỏ nơi lỗi. |

Usecase Duyệt thành viên vào nhóm

|  |  |
| --- | --- |
| Actor | Quản trị viên nhóm |
| Điều kiện | Đã truy cập vào nhóm |
| Các bước thực hiện | 1. Chọn “Quản lý nhóm”. 2. Hệ thống hiển thị giao diện 2 tab danh sách. 3. Chọn tab “Chưa duyệt” để xem danh sách thành viên yêu cầu vào nhóm. 4. Hệ thống hiển thị danh sách thành viên chưa duyệt.   Chọn nút “Chấp nhận” để thêm thành viên.  Chọn nút “Xóa thành viên” để xóa khỏi nhóm. |

Usecase Xem, xóa thành viên trong nhóm

|  |  |
| --- | --- |
| Actor | Quản trị viên nhóm |
| Điều kiện | Đã truy cập vào nhóm |
| Các bước thực hiện | 1. Chọn “Quản lý nhóm”. 2. Hệ thống hiển thị giao diện 2 tab danh sách. 3. Chọn tab “Danh sách nhóm” để xem danh sách thành viên. 4. Hệ thống hiển thị giao diện danh sách thành viên trong nhóm. 5. Chọn nút “Xem” để xem thông tin thành viên .   Chọn nút “Xóa thành viên” để xóa khỏi nhóm. |

Usecase Làm đề trắc nghiệm

|  |  |
| --- | --- |
| Actor | Học viên |
| Điều kiện | Làm các đề thuộc phạm vi người dùng (đề chung, đề nhóm) |
| Các bước thực hiện | 1. Người dùng chọn môn học tại giao diện chính. 2. Hệ thống hiển thị các đề trắc nghiệm trong phạm vi xem của người dùng. 3. Người dùng chọn đề muốn làm. 4. Hệ thống hiển thị giao diện đề trắc nghiệm. 5. Người dùng chọn nút “Làm bài”. 6. Hệ thống bắt đầu tính giờ làm. 7. Người dùng chọn nút “Nộp bài” khi muốn nộp. 8. Hệ thống lưu bài và tự động trả về số câu đúng sai. 9. Người dùng chọn “Xem đáp án” để xem bài đã làm. 10. Hệ thống hiển thị giao diện đáp án người dùng, đáp án đúng và lời giải thích. |

Usecase Ghi chú

|  |  |
| --- | --- |
| Actor | Học viên |
| Điều kiện | Đã làm đề trắc nghiệm đó |
| Các bước thực hiện | 1. Người dùng chọn biểu tượng cây viết trước mỗi câu hỏi. 2. Hệ thống hiển thịc giao diện ghi chú. 3. Người dùng nhập nội dung ghi chú. 4. Chọn “Lưu”. 5. Hệ thống lưu ghi chú thành công và chuyển thành màu khác cho biểu tượng cây bút. |

Usecase Tương tác trao đổi

|  |  |
| --- | --- |
| Actor | Tất cả actor trong hệ thống |
| Điều kiện | Thuộc phạm vi người dùng. |
| Các bước thực hiện | 1. Người dùng vào các đề. 2. Chọn nút “Bình luận”. 3. Hệ thống hiển thị giao diện bình luận.   Nhập bình luận vào khung nhập. Bấm nút “Đăng” để đóng góp ý kiến, giúp đỡ. |

### **Sơ đồ mô hình quan niệm ở mức dữ liệu (Conceptual Data Model)**

… Sơ đồ CDM

1. **Mô tả dữ liệu cho sơ đồ Conceptual Data Model**

Bảng mô tả các thực thể trong sơ đồ CDM

….

### **Sơ đồ mô hình quan niệm ở mức vật lý (Physical Data Model)**

… SƠ đồ PDM

1. **Mô tả dữ liệu từ bảng dữ liệu**
2. **Sơ đồ Sequence diagram**

## **Quy trình thực hiện: …**

## **Kết quả thực hiện: (hình ảnh) …**

# CHƯƠNG 4: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

* 1. **Kết quả đạt được:**
     1. **Về kiến thức:**

Sau thời gian nghiên cứu công nghệ, kiến thức chuyên môn để thực hiện đồ án, em đã biết thêm về nhiều kiến thức mới cũng như nâng cao kĩ năng của bản thân về:

Phân tích và thiết kế hệ thống thông tin.

Lập kịch bản cho từng chức năng.

Thiết kế sơ đồ Usecase, sơ đồ chức năng, sơ đồ nghiệp vụ.

Hiểu được quy trình phát triển và vòng đời phát triển phần mềm.

Nắm được kiến trúc Microservices.

Sử dụng được các Framework, thư viện tốt: Jhipster, Spring Boot, Hibernate,

ReactJS, Ant Design, ..

Sử dụng thành thạo các công cụ hỗ trợ tốt cho việc lập trình: IntelliJ IDEA,

MySQL workbench 8.0, Power Designer, Star UML,…

* + 1. **Về chương trình:**

Xây dựng thành công hệ thống với các mục tiêu như:

Hệ thống đầy đủ các chức năng cơ bản.

Xây dựng được một ứng dụng web đạt được các mục tiêu sau:

Đáp ứng tương đối đầy đủ cho quy trình nghiệp vụ.

Bảo mật và phân quyền đầy đủ.

Thiết kế giao diện trực quan, dễ sử dụng.

Hệ thống hoạt động tốt đáp ứng các yêu cầu cần thiết của một website ứng dụng công nghệ vào công việc thực tế.

* 1. **Khả năng ứng dụng**

Hệ thống khả năng ứng dụng cao, phù hợp áp dụng thực tế cho các trường giảng dạy trong bối cảnh dịch bệnh hiện nay.

Ứng dụng được xây dựng dựa trên công nghệ Microservices nên có thể mở rộng,

nâng cấp tính năng dễ dàng.

* 1. **Hạn chế …**
  2. **Hướng phát triển**

Để tiếp tục phát triển đề tài này, em nhận thấy cần phải tiếp tục thực hiện một số việc sau:

* Xử lý các lỗi chặt chẽ hơn trước khi đưa vào sử dụng.
* Mở rộng bài toán cho nhiều trường riêng biệt.
* Bổ sung nhiều hơn các báo cáo, thống kê.
* Thêm tính năng ghi log thông báo chi tiết khi truy cập chương trình và  
  thực hiện thao tác thay đổi trong hệ thống để có thể giám sát hệ thống.
* Phát triển các tính năng hỗ trợ thanh toán (ATM, PayNet, …) khi tổ chức thành các lớp online.

# TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

[1]: Cách tạo một ứng dụng chat trong Android với Firebase <https://code.tutsplus.com/vi/tutorials/how-to-create-an-android-chat-app-using-firebase--cms-27397>

[2]: Firebase là gì? Giải thích những chức năng cơ bản của Firebase <https://viblo.asia/p/firebase-la-gi-giai-thich-nhung-chuc-nang-co-ban-cua-firebase-bWrZn0jQ5xw>

[3]: Xây dựng Car Location Tracking cho android <https://vntalking.com/car-location-tracking-voi-firebase.html>

Tài liệu tiếng Anh

[4]: Android training course <https://developer.android.com/courses>

[5]: Firebase Doc <https://firebase.google.com/docs>