|  |  |
| --- | --- |
| **Course** | CS504070 - Service-Oriented Architecture  (HK2/2023-2024 |
| **Fullname** | Lê Hữu Lợi |
| **StudentID** | 52100819 |
| **Homework** | HW1 |

**Câu 1:**

**a. Domain-Driven Design (DDD)**

Domain-Driven Design (DDD) là một phương pháp phát triển phần mềm tập trung vào việc tạo ra các hệ thống phần mềm phản ánh chặt chẽ miền kinh doanh mà chúng hỗ trợ. DDD giúp các nhà phát triển hiểu rõ hơn về miền kinh doanh và tạo ra các hệ thống dễ bảo trì và mở rộng hơn.

**b. Các thành phần và nguyên tắc chính của DDD**

Các thành phần và nguyên tắc chính của DDD bao gồm:

Miền (Domain): Miền là phạm vi kinh doanh mà hệ thống phần mềm sẽ hỗ trợ.

Mô hình miền (Domain Model): Mô hình miền là một biểu diễn trừu tượng của miền kinh doanh. Mô hình miền được tạo ra bằng cách xác định các thực thể, thuộc tính và hành vi chính trong miền.

Ngôn ngữ miền cụ thể (DSL):\*\* DSL là một ngôn ngữ được thiết kế riêng để mô tả miền kinh doanh. DSL giúp các nhà phát triển giao tiếp về miền kinh doanh một cách dễ dàng hơn.

Thiết kế theo miền (Domain-Driven Design):\*\* Thiết kế theo miền là quá trình tạo ra các hệ thống phần mềm phản ánh chặt chẽ mô hình miền. Thiết kế theo miền giúp các hệ thống dễ bảo trì và mở rộng hơn.

**c. Ví dụ về DDD**

Một ví dụ về DDD là hệ thống quản lý đơn hàng. Mô hình miền của hệ thống quản lý đơn hàng sẽ bao gồm các thực thể như đơn hàng, khách hàng, sản phẩm, v.v. Các thuộc tính của các thực thể này sẽ bao gồm tên, địa chỉ, số lượng, v.v. Các hành vi của các thực thể này sẽ bao gồm tạo đơn hàng, hủy đơn hàng, giao hàng, v.v.

DSL của hệ thống quản lý đơn hàng có thể bao gồm các thuật ngữ như "đơn hàng", "khách hàng", "sản phẩm", "tạo đơn hàng", "hủy đơn hàng", "giao hàng", v.v.

Thiết kế theo miền của hệ thống quản lý đơn hàng sẽ bao gồm các lớp và thành phần phản ánh chặt chẽ mô hình miền. Ví dụ, hệ thống có thể có một lớp Đơn hàng, một lớp Khách hàng và một lớp Sản phẩm. Lớp Đơn hàng có thể có các phương thức để tạo đơn hàng, hủy đơn hàng và giao hàng. Lớp Khách hàng có thể có các phương thức để tạo khách hàng và cập nhật thông tin khách hàng. Lớp Sản phẩm có thể có các phương thức để tạo sản phẩm và cập nhật thông tin sản phẩm.

DDD là một phương pháp phát triển phần mềm mạnh mẽ và linh hoạt. Nó có thể được sử dụng để xây dựng các hệ thống phần mềm phức tạp, bao gồm cả các hệ thống doanh nghiệp.

**Câu 2:**

**Sự khác nhau giữa Coupling và Cohesion**

**Coupling**

* Coupling là mức độ phụ thuộc giữa các thành phần hoặc mô-đun trong một hệ thống phần mềm.
* Coupling cao có nghĩa là các thành phần hoặc mô-đun phụ thuộc nhiều vào nhau.
* Coupling thấp có nghĩa là các thành phần hoặc mô-đun phụ thuộc ít vào nhau.

**Cohesion**

* Cohesion là mức độ liên kết giữa các phần tử bên trong một thành phần hoặc mô-đun.
* Cohesion cao có nghĩa là các phần tử bên trong một thành phần hoặc mô-đun liên kết chặt chẽ với nhau.
* Cohesion thấp có nghĩa là các phần tử bên trong một thành phần hoặc mô-đun liên kết lỏng lẻo với nhau.

**Sự khác nhau chính giữa coupling và cohesion**

* Coupling là mức độ phụ thuộc giữa các thành phần hoặc mô-đun, trong khi cohesion là mức độ liên kết giữa các phần tử bên trong một thành phần hoặc mô-đun.
* Coupling cao có thể dẫn đến các hệ thống khó bảo trì và mở rộng, trong khi cohesion cao có thể dẫn đến các hệ thống dễ bảo trì và mở rộng.

**Câu 3**

* Dịch vụ (Service): Mỗi microservice đại diện cho một dịch vụ cụ thể, như quản lý đăng nhập, xử lý thanh toán hoặc gửi email. Mỗi dịch vụ được triển khai và hoạt động độc lập với các dịch vụ khác.
* Cơ sở dữ liệu (Database): Mỗi microservice có thể có cơ sở dữ liệu riêng để lưu trữ dữ liệu liên quan đến chức năng của nó. Sử dụng cơ sở dữ liệu riêng biệt giúp tách biệt dữ liệu và đảm bảo tính riêng tư cho mỗi dịch vụ.
* Giao tiếp và cân nhắc (Communication and Coordination): Các microservices cần giao tiếlịnh và tương tác với nhau để hoạt động như một hệ thống. Điều này thường được thực hiện thông qua các giao thức chuẩn như RESTful API, giao thức message queue hoặc giao thức gRPC.

**Thành phần quan trọng nhất cho một dự án cuối cùng:**

Thành phần quan trọng nhất cho một dự án cuối cùng sẽ phụ thuộc vào dự án cụ thể và mục tiêu của dự án. Tuy nhiên, một số yếu tố cần xem xét bao gồm:

* **Độ phức tạp của dự án:** Nếu dự án phức tạp và có nhiều thành phần khác nhau, thì việc điều phối và khám phá dịch vụ có thể quan trọng hơn.
* **Khả năng mở rộng của dự án:** Nếu dự án dự kiến sẽ mở rộng quy mô cho một số lượng lớn người dùng, thì API gateway và khám phá dịch vụ có thể quan trọng hơn.
* **An ninh của dự án:** Nếu dự án xử lý dữ liệu nhạy cảm, thì cổng API và xác thực có thể quan trọng hơn.

Nói chung, tôi sẽ nói rằng **sự phối hợp** là thành phần quan trọng nhất cho một dự án cuối cùng. Điều này là do điều phối chịu trách nhiệm quản lý vòng đời và giao tiếp của microservices. Nếu không có sự phối hợp, sẽ rất khó để xây dựng một hệ thống microservices có thể mở rộng và đáng tin cậy.

**Đối với dự án cuối cùng của em, em đang xây dựng một hệ thống thương mại điện tử dựa trên microservices. Em tin rằng điều phối là thành phần quan trọng nhất cho dự án này vì nó sẽ cho phép em quản lý vòng đời và giao tiếp của các microservice khác nhau trong hệ thống. Điều này sẽ giúp em xây dựng một hệ thống có thể mở rộng và đáng tin cậy có thể xử lý một số lượng lớn người dùng.**

**Câu 4**

**Link- git:** [**https://github.com/lejame/kientrucdichvu\_52100819.git**](https://github.com/lejame/kientrucdichvu_52100819.git)

**Giấy phép Apache 2.0**

* **Giấy phép cho phép:** Cho phép người dùng sử dụng, sửa đổi và phân phối phần mềm cho bất kỳ mục đích nào, có hoặc không có sửa đổi, mà không phải trả tiền bản quyền hoặc xin phép.
* **Yêu cầu thông báo bản quyền và giấy phép phải được bao gồm trong tất cả các bản sao của phần mềm.**
* **Cấp bằng sáng chế:** Nếu một người đóng góp thêm bằng sáng chế vào dự án, họ sẽ cấp giấy phép không độc quyền, trên toàn thế giới, miễn phí bản quyền để sử dụng bằng sáng chế đó liên quan đến phần mềm.

**Giấy phép Công cộng GNU v3.0 (GPL-3.0)**

* **Giấy phép copyleft:** Yêu cầu người dùng phân phối phần mềm theo cùng điều khoản cấp phép, bao gồm mọi sửa đổi mà họ thực hiện.
* **Yêu cầu thông báo bản quyền và giấy phép phải được bao gồm trong tất cả các bản sao của phần mềm.**
* **Cấp bằng sáng chế:** Nếu một người đóng góp thêm bằng sáng chế vào dự án, họ sẽ cấp giấy phép không độc quyền, trên toàn thế giới, miễn phí bản quyền để sử dụng bằng sáng chế đó liên quan đến phần mềm.

**Giấy phép MIT**

* **Giấy phép cho phép:** Cho phép người dùng sử dụng, sửa đổi và phân phối phần mềm cho bất kỳ mục đích nào, có hoặc không có sửa đổi, mà không phải trả tiền bản quyền hoặc xin phép.
* **Yêu cầu thông báo bản quyền và giấy phép phải được bao gồm trong tất cả các bản sao của phần mềm.**
* **Không cấp bằng sáng chế.**

**Giấy phép Công cộng Mozilla 2.0 (MPL-2.0)**

* **Giấy phép copyleft:** Yêu cầu người dùng phân phối phần mềm theo cùng điều khoản cấp phép, bao gồm mọi sửa đổi mà họ thực hiện.
* **Yêu cầu thông báo bản quyền và giấy phép phải được bao gồm trong tất cả các bản sao của phần mềm.**
* **Cấp bằng sáng chế:** Nếu một người đóng góp thêm bằng sáng chế vào dự án, họ sẽ cấp giấy phép không độc quyền, trên toàn thế giới, miễn phí bản quyền để sử dụng bằng sáng chế đó liên quan đến phần mềm.

**Giấy phép Phần mềm Mở 3.0 (OSL-3.0)**

* **Giấy phép cho phép:** Cho phép người dùng sử dụng, sửa đổi và phân phối phần mềm cho bất kỳ mục đích nào, có hoặc không có sửa đổi, mà không phải trả tiền bản quyền hoặc xin phép.
* **Yêu cầu thông báo bản quyền và giấy phép phải được bao gồm trong tất cả các bản sao của phần mềm.**
* **Không cấp bằng sáng chế.**

**Sự so sánh**

| **Tính năng** | **Giấy phép Apache 2.0** | **Giấy phép Công cộng GNU v3.0** | **Giấy phép MIT** | **Giấy phép Công cộng Mozilla 2.0** | **Mở Giấy phép Phần mềm 3.0** |
| --- | --- | --- | --- | --- | --- |
| Loại giấy phép | Permissive | Copyleft | Permissive | Copyleft | Permissive |
| Thông báo bản quyền và giấy phép | Bắt buộc | Bắt buộc | Bắt buộc | Bắt buộc | Bắt buộc |
| Cấp bằng sáng chế | Không độc quyền, trên toàn thế giới, miễn phí bản quyền | Không độc quyền, trên toàn thế giới, miễn phí bản quyền | Không ai | Không độc quyền, trên toàn thế giới, miễn phí bản quyền | Không ai |
| Các phiên bản sửa đổi có thể được phân phối theo một giấy phép khác không? | Có | Không | Có | Không | Có |