3.2.2 Thiết kế kiến trúc phần mềm

3.2.2.1 Mô hình kiến trúc

3.2.2.1.1 Kiến trúc hướng đối tượng (Objects based Architecture)

*Khái niệm và Vai trò:*

Khái niệm: Kiến trúc hướng đối tượng (Objects based Architecture) hay còn được biết tới là Mô hình kiến trúc hướng đối tượng (Object-Oriented Design - OOD) là một phương pháp thiết kế phần mềm dựa trên việc xác định các đối tượng trong hệ thống và các mối quan hệ giữa chúng. Một đối tượng bao gồm dữ liệu và các thao tác trên dữ liệu đó.

Vai trò:

Tăng tính mô đun (modularity) và khả năng tái sử dụng của phần mềm.

Dễ dàng bảo trì và mở rộng.

Tăng cường tính nhất quán và khả năng trực quan hóa hệ thống.

*Các khái niệm cơ bản:*

Lớp (Class): Định nghĩa các thuộc tính và phương thức chung cho các đối tượng.

Đối tượng (Object): Thể hiện cụ thể của một lớp, với các giá trị thuộc tính riêng biệt.

Kế thừa (Inheritance): Tạo ra các lớp con thừa hưởng thuộc tính và phương thức từ lớp cha.

Đa hình (Polymorphism): Cho phép các đối tượng khác nhau trả lời cùng một thông điệp theo các cách khác nhau.

Đóng gói (Encapsulation): Giấu thông tin chi tiết của đối tượng và chỉ cung cấp giao diện bên ngoài.

*Quy trình thiết kế hướng đối tượng:*

Xác định các lớp và đối tượng: Từ các yêu cầu hệ thống, xác định các đối tượng cần thiết và phân loại chúng vào các lớp.

Xác định các mối quan hệ: Xác định các mối quan hệ kế thừa, kết hợp giữa các lớp.

Thiết kế giao diện lớp: Định nghĩa các phương thức và thuộc tính cho mỗi lớp.

Tạo biểu đồ tương tác: Mô tả cách các đối tượng tương tác với nhau trong các tình huống sử dụng cụ thể.

Thiết kế chi tiết: Xây dựng chi tiết các phương thức và thuộc tính cho mỗi lớp dựa trên các mẫu thiết kế.

*Ưu và nhược điểm của OOD:*

Ưu điểm:

Dễ bảo trì và mở rộng do tính mô đun cao.

Tăng khả năng tái sử dụng mã nguồn.

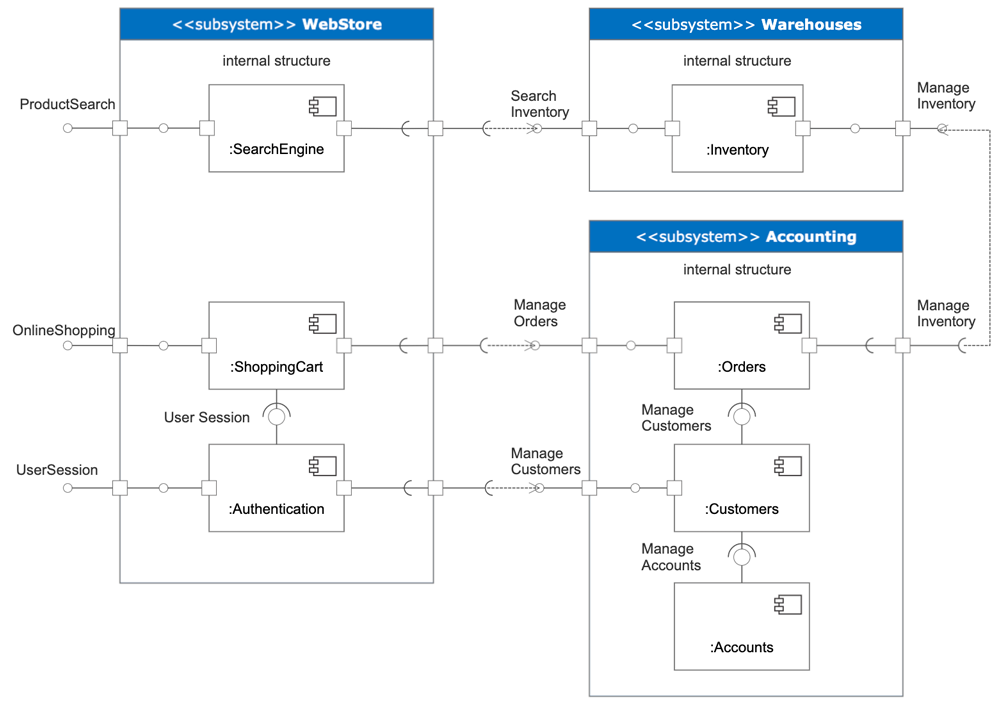
Phản ánh chính xác thế giới thực.

Nhược điểm:

Có thể phức tạp với các hệ thống lớn và cần nhiều tài liệu thiết kế.

Đòi hỏi kiến thức sâu về lập trình hướng đối tượng.

*Biểu đồ ví dụ minh họa:*



3.2.2.1.2 Kiến trúc client-server (client-server Architecture)

*Khái niệm và Vai trò:*

Khái niệm: Mô hình Client-Server là một mô hình kiến trúc phân tán trong đó các máy tính (clients) yêu cầu dịch vụ từ các máy chủ (servers). Máy chủ cung cấp dịch vụ và xử lý các yêu cầu từ các máy khách.

Vai trò:

Phân tách nhiệm vụ và tài nguyên, giúp hệ thống hoạt động hiệu quả hơn.

Dễ dàng mở rộng và nâng cấp hệ thống.

Tăng cường bảo mật và quản lý tập trung.

Thành phần chính: Máy khách (Client): Gửi yêu cầu đến máy chủ và hiển thị kết quả cho người dùng. Ví dụ: trình duyệt web, ứng dụng di động.

Máy chủ (Server): Xử lý các yêu cầu từ máy khách, truy xuất dữ liệu và gửi lại kết quả. Ví dụ: máy chủ web, máy chủ cơ sở dữ liệu.

Mạng liên kết: Cho phép giao tiếp giữa máy khách và máy chủ.

*Các bước triển khai mô hình Client-Server:*

Thiết kế giao diện người dùng: Phát triển giao diện mà người dùng tương tác trên máy khách.

Thiết kế và triển khai máy chủ: Xây dựng máy chủ với các dịch vụ cần thiết.

Giao tiếp mạng: Thiết lập kết nối và giao thức truyền thông giữa máy khách và máy chủ.

Kiểm thử và triển khai: Thực hiện kiểm thử chức năng và hiệu suất, sau đó triển khai hệ thống.

*Ưu và nhược điểm của mô hình Client-Server:*

Ưu điểm:

Khả năng mở rộng cao.

Quản lý tập trung dễ dàng hơn.

Tăng cường bảo mật nhờ quản lý tập trung.

Nhược điểm:

Phụ thuộc vào máy chủ, nếu máy chủ gặp sự cố, toàn bộ hệ thống có thể bị ảnh hưởng.

Chi phí thiết lập và bảo trì máy chủ có thể cao.

*Biểu đồ ví dụ minh họa*

A diagram of a computer

Description automatically generated

3.2.2.2. Mô tả kiến trúc

3.2.2.2.1 Biểu đồ gói (package Diagram)

A diagram of a business

Description automatically generated

3.2.2.2.2 Biểu đồ thành phần (Component Diagram)

A diagram of a computer

Description automatically generated

3.2.2.2.3 Biểu đồ triển khai (Deployment Diagram)

A diagram of a computer

Description automatically generated