**Tổng hợp UML Diagrams**

[**Class Diagram**](HOW/Class-Diagram.pdf)

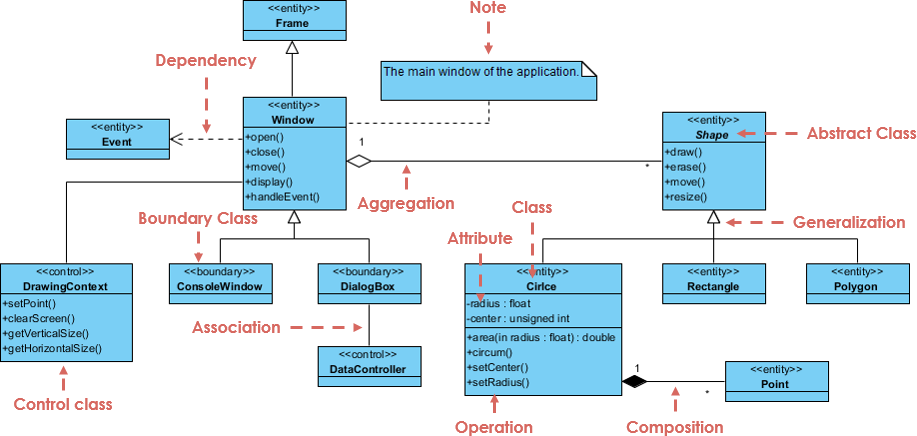
• ➤ Định nghĩa: Mô tả các lớp, thuộc tính, phương thức và mối quan hệ giữa các lớp.

• ➤ Khi nào dùng: Dùng để mô hình hóa cấu trúc tĩnh của hệ thống.

• ➤ Cách vẽ: Vẽ các lớp với tên, thuộc tính, và phương thức. Liên kết bằng Association, Aggregation (◇), Composition (◆), Generalization (△). Sử dụng multiplicity (1, 0..\*, v.v.).

• ➤ Lưu ý: Aggregation là quan hệ 'có' nhưng phần tử có thể tồn tại độc lập. Composition là quan hệ 'có' nhưng phần tử không thể tồn tại độc lập.

Ảnh minh hoạ:



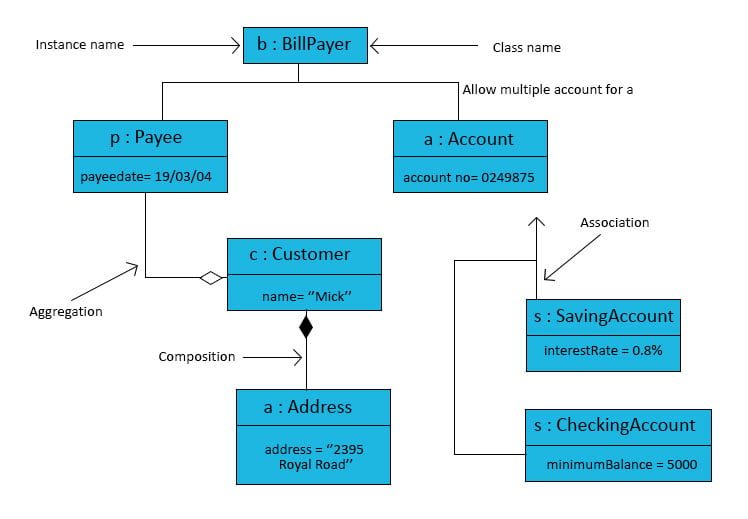
[**Object Diagram**](HOW/Object-Diagram.pdf)

• ➤ Định nghĩa: Mô tả trạng thái cụ thể của hệ thống tại một thời điểm với các đối tượng cụ thể.

• ➤ Khi nào dùng: Dùng để minh họa các đối tượng và mối quan hệ tại runtime.

• ➤ Cách vẽ: Giống class diagram nhưng sử dụng instance cụ thể, có thể ghi giá trị thuộc tính.

Ảnh minh hoạ:



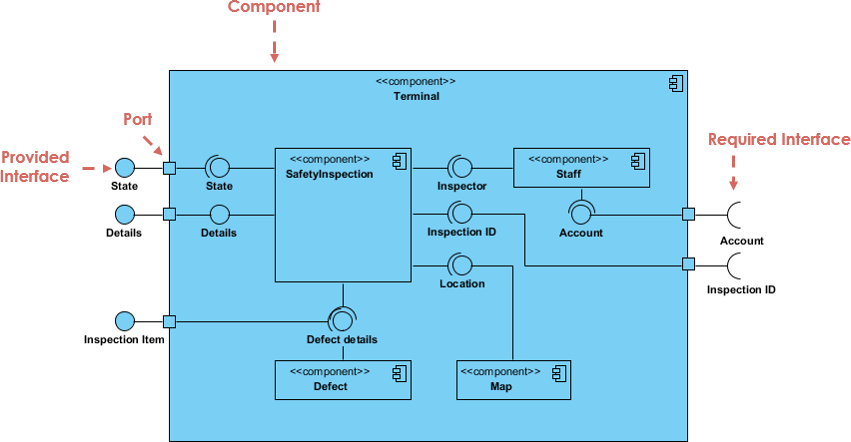
[**Component Diagram**](HOW/Component-Diagram.pdf)

• ➤ Định nghĩa: Mô tả các thành phần phần mềm và mối liên hệ giữa chúng.

• ➤ Khi nào dùng: Dùng để mô hình hóa kiến trúc cấp cao, module, dependency.

• ➤ Cách vẽ: Vẽ component dưới dạng hộp chữ nhật có biểu tượng hai hình chữ nhật nhỏ. Kết nối bằng interface (required/provided).

Ảnh minh hoạ:



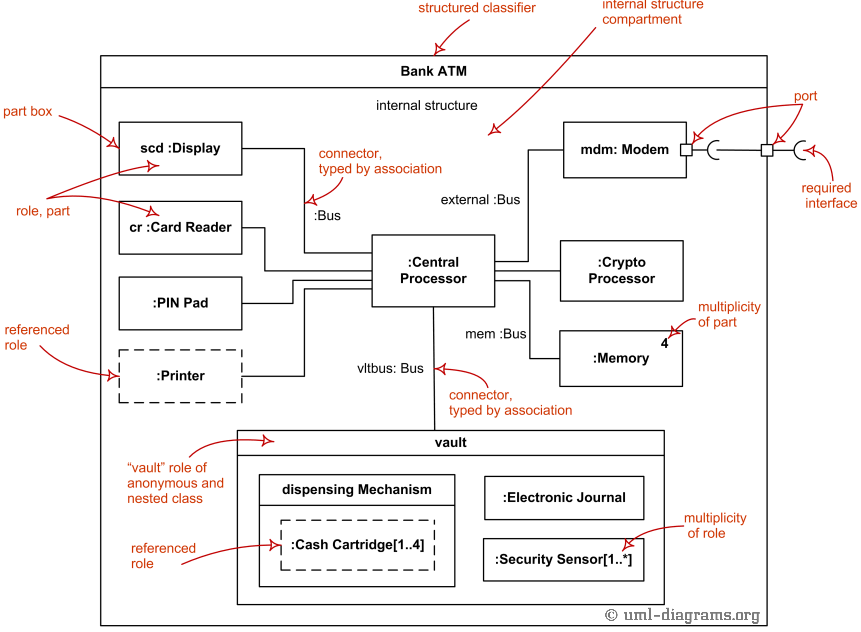
[**Composite Structure Diagram**](HOW/Composite-Structure-Diagram.pdf)

• ➤ Định nghĩa: Mô tả cấu trúc bên trong của một lớp và sự tương tác giữa các phần tử nội bộ.

• ➤ Khi nào dùng: Khi cần mô hình hóa chi tiết nội bộ lớp.

• ➤ Cách vẽ: Gồm các part, port, connector.

Ảnh minh hoạ:



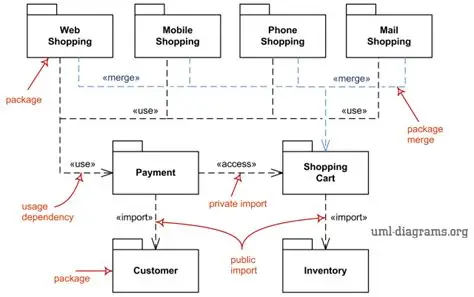
[**Package Diagram**](HOW/Package-Diagram.pdf)

• ➤ Định nghĩa: Mô tả cách tổ chức hệ thống thành các package/module.

• ➤ Khi nào dùng: Để nhóm các class/component liên quan vào package.

• ➤ Cách vẽ: Vẽ các package (hình chữ nhật có tab), kết nối bằng dependency.

Ảnh minh hoạ:



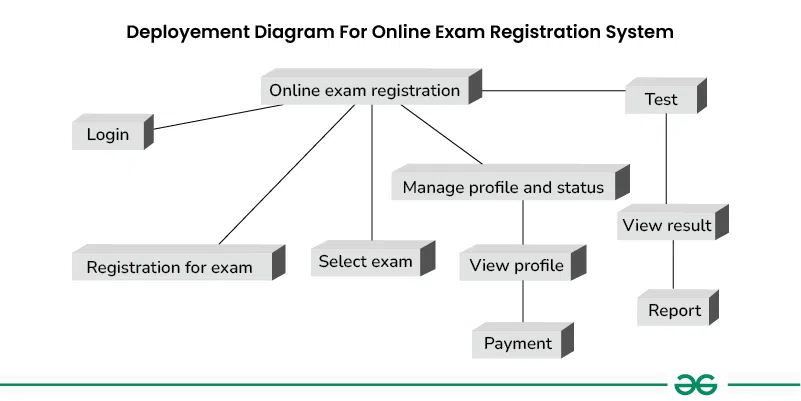
[**Deployment Diagram**](HOW/Deployment-Diagram.pdf)

• ➤ Định nghĩa: Mô tả cách các thành phần phần mềm được triển khai trên phần cứng.

• ➤ Khi nào dùng: Dùng trong giai đoạn triển khai hệ thống.

• ➤ Cách vẽ: Node (thiết bị) là hộp 3D, artifact là phần mềm trong node.

Ảnh minh hoa:



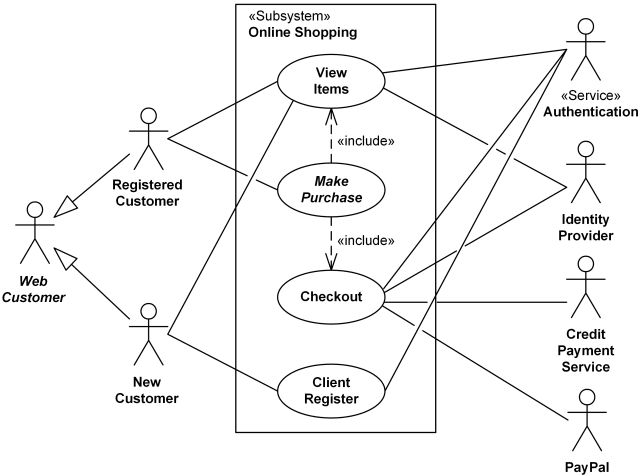
[**Use Case Diagram**](HOW/Use-Case-Diagram.pdf)

• ➤ Định nghĩa: Mô tả các chức năng hệ thống cung cấp và ai sử dụng.

• ➤ Khi nào dùng: Dùng ở giai đoạn phân tích yêu cầu.

• ➤ Cách vẽ: Actor (hình người), use case (oval), mối quan hệ include, extend, generalization.

Ảnh minh hoạ:



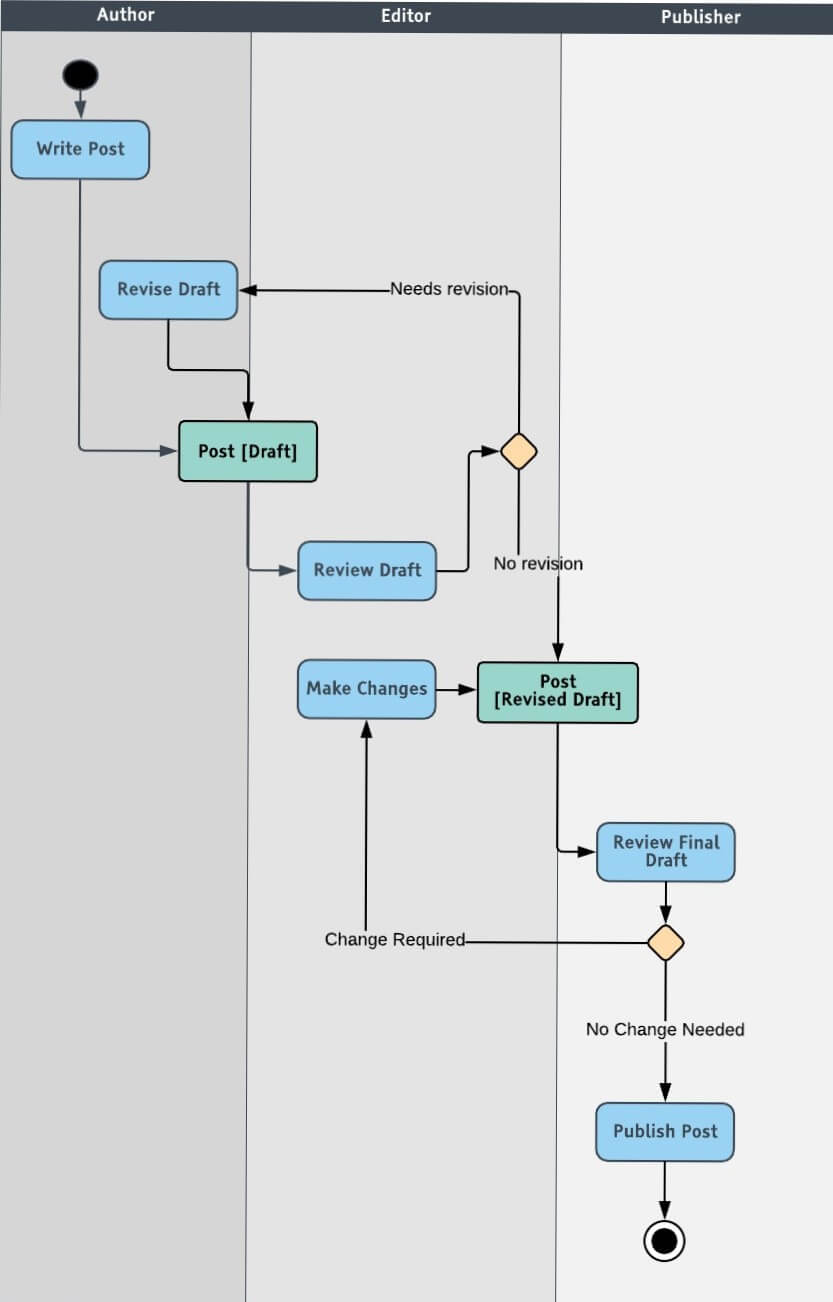
[**Activity Diagram**](HOW/Activity-Diagram.pdf)

• ➤ Định nghĩa: Mô tả luồng công việc, hành động và điều kiện rẽ nhánh.

• ➤ Khi nào dùng: Khi mô tả quy trình nghiệp vụ hoặc xử lý.

• ➤ Cách vẽ: Start node, activity node, decision/merge node, end node. Dùng mũi tên chỉ hướng.

Ảnh minh hoạ:



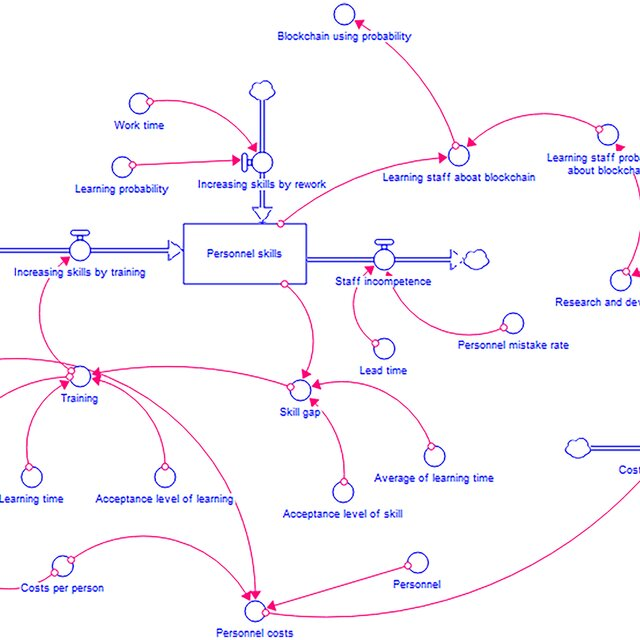
[**State Machine Diagram**](HOW/State-Machine-Diagram.pdf)

• ➤ Định nghĩa: Mô tả các trạng thái của đối tượng và cách chuyển đổi.

• ➤ Khi nào dùng: Khi cần mô tả vòng đời của đối tượng.

• ➤ Cách vẽ: State (hình chữ nhật bo góc), transition (mũi tên), initial/final state (hình tròn).

Ảnh minh hoạ:



[**Sequence Diagram**](HOW/Sequence-Diagram.pdf)

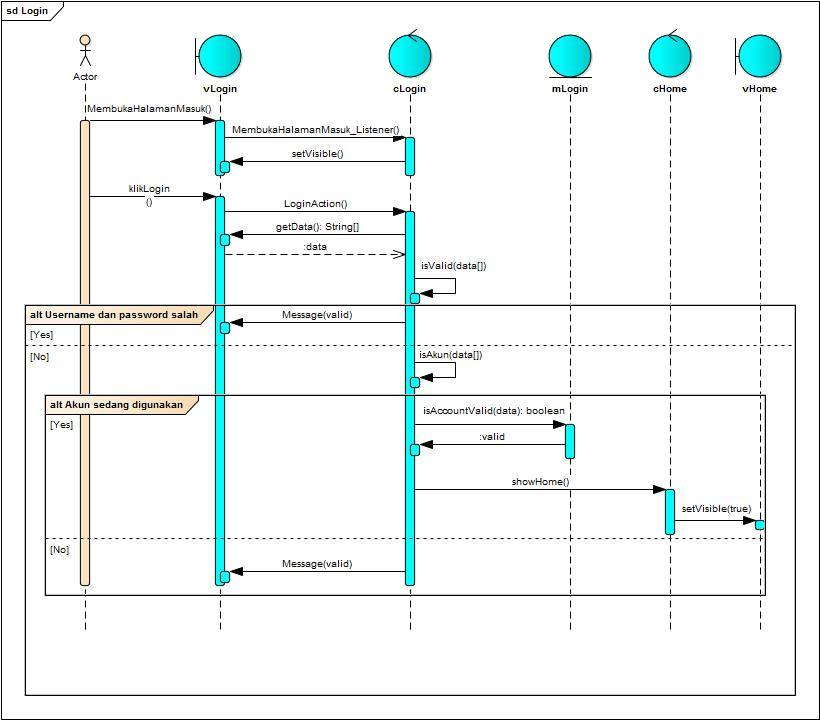
• ➤ Định nghĩa: Mô tả trình tự các thông điệp giữa đối tượng theo thời gian.

• ➤ Khi nào dùng: Dùng để mô tả luồng tương tác chi tiết.

• ➤ Cách vẽ: Lifeline, message (mũi tên), activation bar.

• ➤ Lưu ý: Các type: Boundary (UI), Control (Logic), Entity (Dữ liệu).

Ảnh minh hoạ:

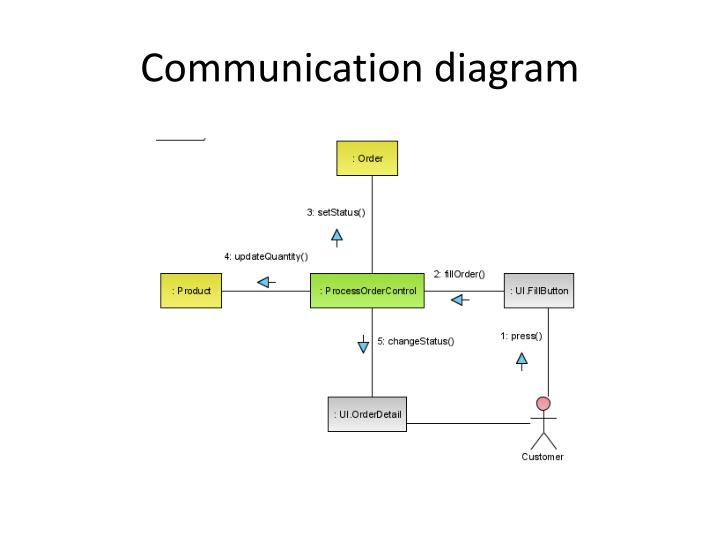


[**Communication Diagram**](HOW/Communication-Diagram.pdf)

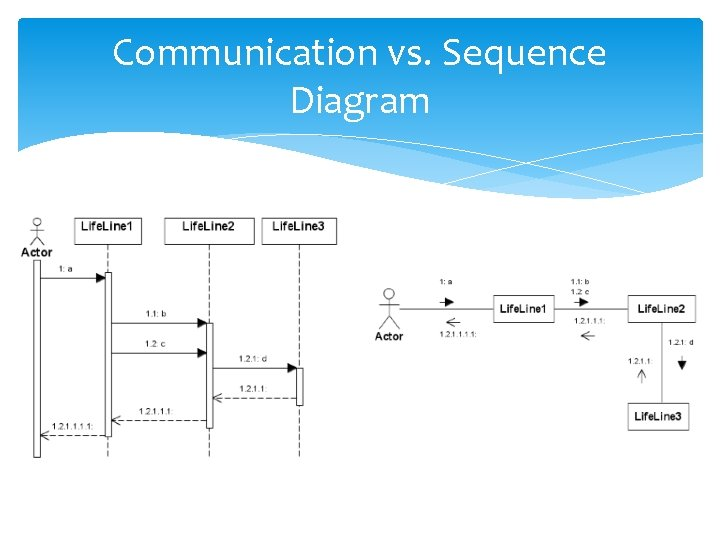
• ➤ Định nghĩa: Mô tả tương tác giữa các đối tượng, tập trung vào quan hệ hơn là thời gian.

• ➤ Khi nào dùng: Khi muốn thấy mối quan hệ giữa các đối tượng trong tương tác.

• ➤ Cách vẽ: Các đối tượng kết nối bằng đường kẻ, tin nhắn được đánh số.

Ảnh minh hoạ:  


So sánh 2 loại Sequence và Communication:



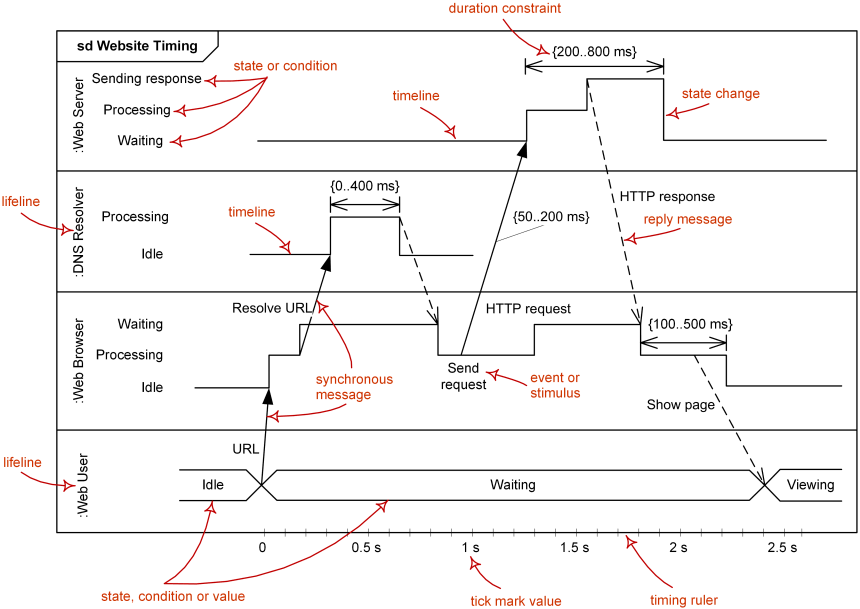
[**Timing Diagram**](HOW/Timing-Diagram.pdf)

• ➤ Định nghĩa: Mô tả thay đổi trạng thái theo thời gian.

• ➤ Khi nào dùng: Khi cần kiểm soát thời gian chuyển trạng thái.

• ➤ Cách vẽ: Trục X là thời gian, trục Y là trạng thái. Các dòng trạng thái thay đổi theo thời gian.

Ảnh minh hoạ:

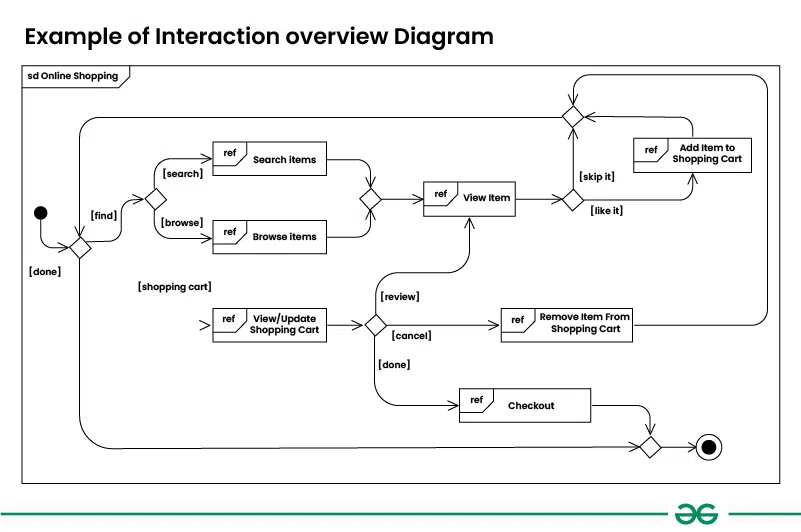


[**Interaction Overview Diagram**](HOW/Interaction-Overview-Diagram.pdf)

• ➤ Định nghĩa: Tổng quan luồng tương tác, kết hợp giữa Activity và Sequence.

• ➤ Khi nào dùng: Mô hình hóa luồng lớn chia thành nhiều interaction nhỏ.

• ➤ Cách vẽ: Dùng các node giống activity diagram, mỗi node có thể là sequence diagram.

Ảnh minh hoạ: 

Bảng Overview so sánh chi tiết các loại Interaction Diagram, để giải quyết bài toán của câu Q2:

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại Diagram** | **Mô Tả** | **Khi Nào Dùng** | **Đặc Điểm Nổi Bật** |
| **1. Sequence Diagram** | Mô tả trình tự các thông điệp giữa các đối tượng theo thời gian (trục dọc). | Khi cần mô tả **chi tiết luồng xử lý theo thời gian**. | Thể hiện rõ **ai gọi ai, theo thứ tự nào**, dễ thấy được **thứ tự và luồng gọi**. |
| **2. Communication Diagram** | Mô tả sự tương tác giữa các đối tượng, tập trung vào **mối quan hệ hơn là thời gian**. | Khi cần thể hiện **cấu trúc kết nối giữa các đối tượng** và **số lượng tương tác**. | Message được **đánh số**, giúp thấy **các đối tượng nào tương tác và kết nối với nhau**. |
| **3. Timing Diagram** | Mô tả sự **thay đổi trạng thái của đối tượng theo thời gian**. | Khi thiết kế hệ thống **thời gian thực** hoặc cần **kiểm soát thời gian cụ thể**. | Trục thời gian là **ngang**, biểu diễn **thời điểm thay đổi trạng thái**. |
| **4. Interaction Overview Diagram** | Tổng hợp các luồng tương tác lớn, phân rã thành **các đoạn Sequence hoặc Activity**. | Khi hệ thống **phức tạp**, cần **mô hình hóa tổng thể luồng xử lý** trước khi đi vào chi tiết. | Kết hợp giữa **Activity + Interaction**, **có thể nhúng Sequence Diagram bên trong**. |