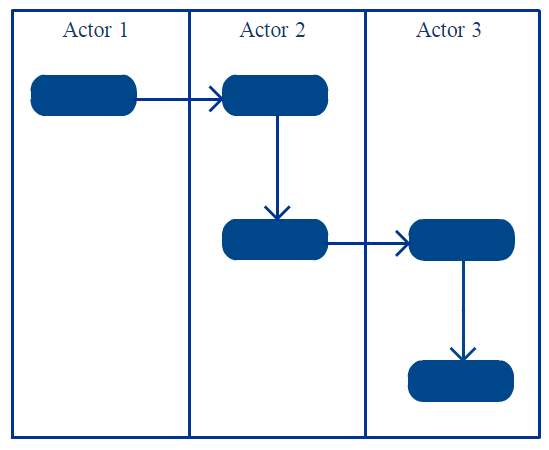
**Activity diagram**

**-Các thành phần chính:**

|  |  |  |
| --- | --- | --- |
| **Kí hiệu** | **Tên gọi** | **Đặc trưng** |
| IMG_256 | Start/Initial node | Thể hiện điểm bắt đầu hoặc trạng thái ban đầu của hoạt động |
| IMG_256 | Activity | Mô tả hành vi của đối tượng trong quy trình |
| IMG_256 | Control flow | Mô tả sự chuyển đổi trạng thái của các hoạt động, chiều của mũi tên biểu thị chiều của luồng xử lý. Hoặc biểu thị luồng của một đối tượng từ hoạt động này chuyển sang hoạt động khác. |
| IMG_256 | Object flow | Mô tả sự tạo ra hoặc sửa đổi các đối tượng theo các hoạt động |
|  | Decision node | -Tập các điều kiện kích hoạt việc chuyển trạng thái. Gồm có một dòng điều khiển đi vào và hai hoặc nhiều dòng điều khiển đi ra, mỗi dòng phải chứa một điều kiện(guard) và các điều kiện này phải loại trừ nhau. |
|  | Merge node | Có hai hoặc nhiều dòng điều khiển đi vào và chỉ có một dòng điều khiển đi ra |
|  | Synchronization bar | Mô tả các dòng điều khiển được thực hiện song song |
| IMG_256 | Fork | Một dòng điều khiển được tách ra thực hiện song song  Một dòng điều khiển đi vào, hai hoặc nhiều dòng điều khiển đi ra  Dùng fork khi các hoạt động thực hiện không quan tâm thứ tự |
| IMG_256 | Join | Kết hợp các dòng điều khiển song song lại thành một luồng chính.  Dòng điều khiển ra được tạo khi tất các các dòng cần thiết đã vào |
|  | Time event | Gián đoạn luồng hoạt động lại một khoảng thời gian |
| IMG_256 | Comment | Thêm ghi chú để mô tả, làm rõ các thành phần |
| IMG_256 | Final node | Mô tả trạng thái kết thúc quy trình  Một activity diagram có thể có nhiều trạng thái kết thúc |
| IMG_256 | Flow final node | Mô tả trạng thái kết thúc của một luồng đơn. |

**Swimlane**



Swimlane là cách để nhóm các hoạt động được thực hiện bởi cùng một actor trên activity diagram.

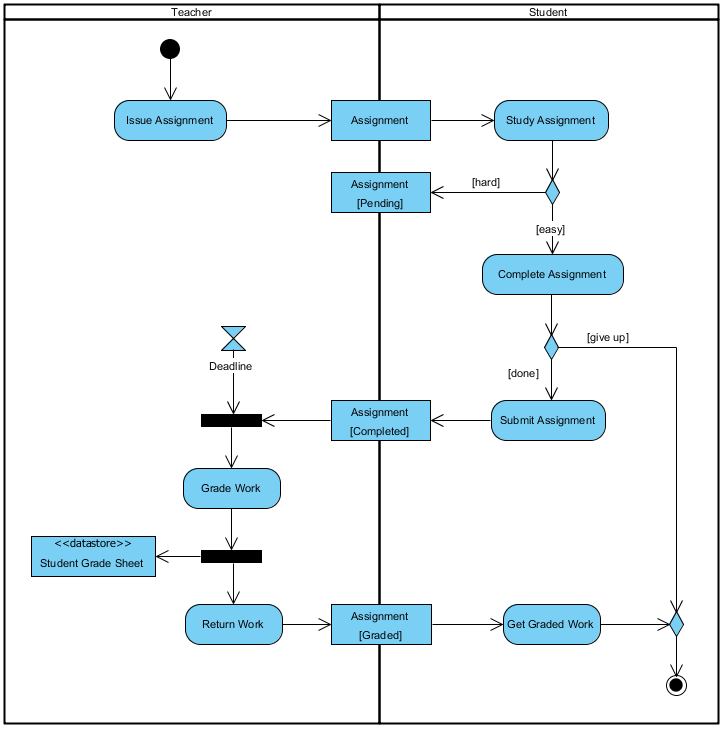
Một vài mẹo để sử dụng swimlane hiệu quả:

+Thêm swimlane vào các quy trình tuyến tính, giúp diagram dễ đọc hơn.

+Không thêm nhiều hơn 5 swimlane

+Sắp xếp các swimlane theo một trật tự logic.

**Ví dụ**

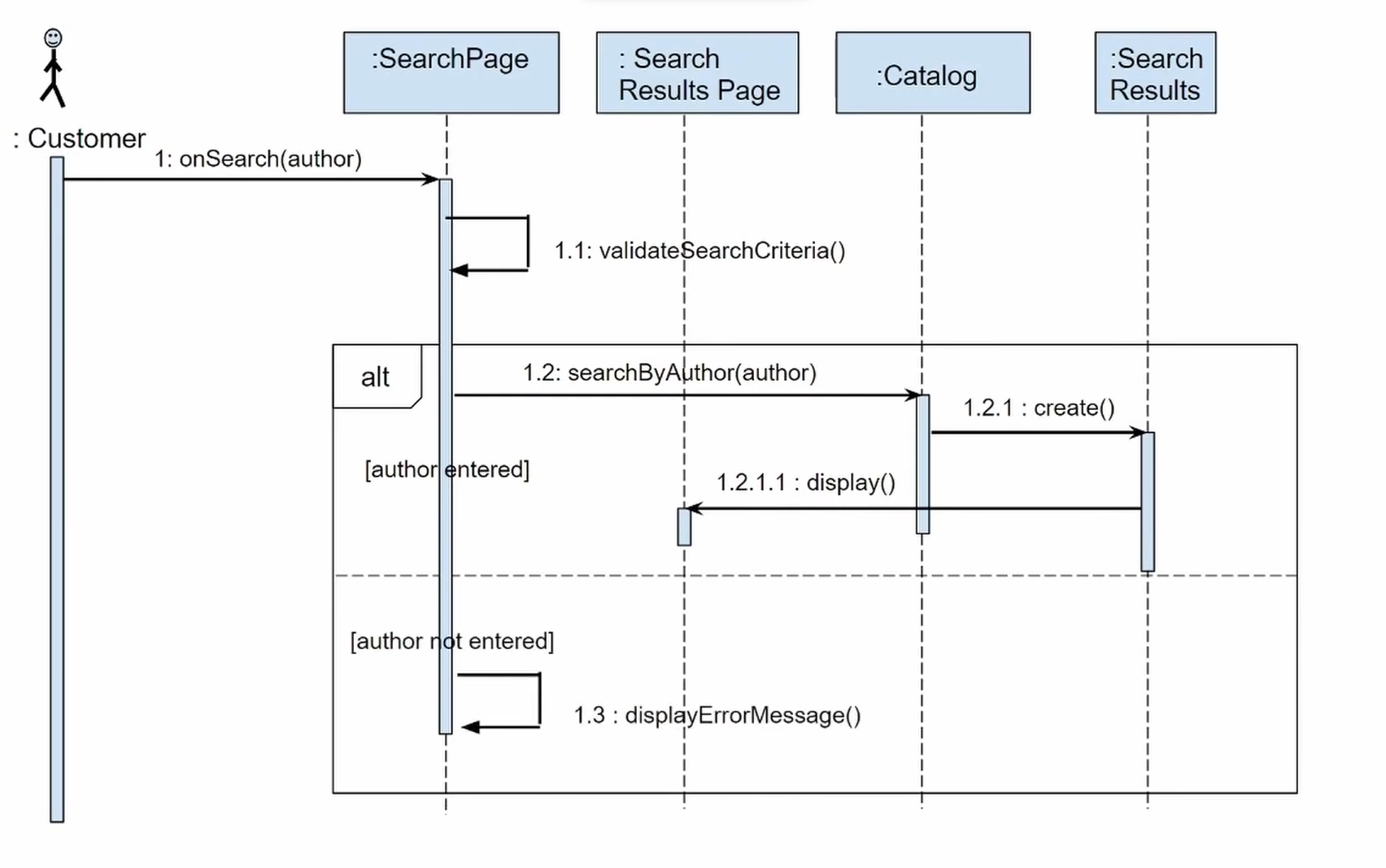


Sequence diagram

**-Các thành phần chính**

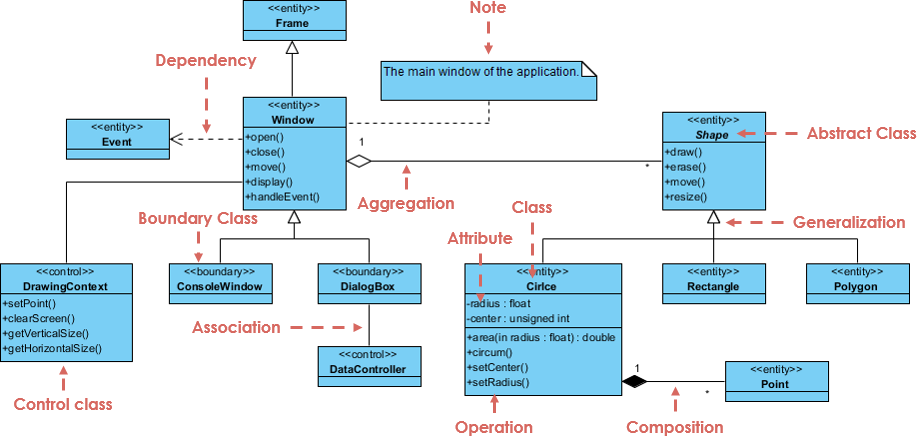
|  |  |
| --- | --- |
| **Kí hiệu** | **Tên gọi - Đặc trưng** |
|  | Đóng vai trò người tương tác với các thành phần trong hệ thống |
|  | Các object trong hệ thống |
|  | Lifeline: biểu thị cho thời gian tồn tại của đối tượng trong hệ thống |
| IMG_256 | Activation:  Nằm dọc trên lifeline của phần tử, thể hiện thời gian phần tử đó đang thực hiện một hoạt động. |
| IMG_256 | Call message:  Được thể hiện bằng mũi tên nét liền, nối giữa các lifeline. |
|  | Synchronous message:  Được thể hiện bằng mũi tên, tô đầy ở phần mũi.  Vd:Khi một máy khách gửi thông báo đến máy chủ và phản hồi trước khi làm bất kỳ điều gì . |
| IMG_256 | Return message:  Được thể hiện bằng mũi tên nét đứt |
| IMG_256 | Self message:  Message được gọi trên cùng một lifeline |
|  | Recursive message:  Là trường hợp đặc biệt của self-message: dùng để mô tả các lời gọi đệ quy |
|  | Found message:  Là tin nhắn biết điểm nhận, nhưng không biết từ đâu gửi đến |
|  | Lost message:  Là tin nhắn biết nơi gửi, nhưng không biết rõ nơi nhận |
|  | Duration constrait |
| IMG_256 | Create message:  Thể hiện sự khởi tạo một lifeline mới |
| IMG_256 | Destroy message:  Thể hiện sự chấm dứt dòng đời của một đối tượng trong hệ thống. |
| IMG_256 | Duration message:  Thể hiện sự chênh lệch thời gian khi truyền tín hiệu từ lifeline gốc đến lifeline mục tiêu |
| IMG_256 | Comment |
|  |  |

**Ví dụ về sequence diagram:**



**Class diagram**

**- Ví dụ về một Class Diagram:**



**- Các thành phần chính:**

1) Classes (Các lớp) là thành phần chính của Class Diagram, chúng mô tả về nhóm đối tượng cùng tính chất, hành động trong hệ thống. Classes sẽ bao gồm:

Tên Class;

Thuộc tính (Attributes);

Phương thức (Method).

2) Relationship (Mối quan hệ) là yếu tố thể hiện mối quan hệ giữa các Class với nhau. Trong Class Diagram, một số quan hệ thường được dùng như: Association; Aggregation; Composition; Inheritance (or Generalization).

Cụ thể:

|  |  |  |
| --- | --- | --- |
| Tên | Kí hiệu | Chức năng |
| Association |  | Mô tả quan hệ giữa hai lớp với nhau, nó thể hiện rằng giữa chúng có liên quan với nhau. |
| Aggregation |  | Là một loại của quan hệ Association nhưng mạnh hơn Association. Aggregation **có thể** cùng chung thời gian sống (cùng sinh ra hoặc chết đi). |
| Composition |  | Là một loại quan hệ còn mạnh hơn cả quan hệ Aggregation. Nó thể hiện mối quan hệ class này là một phần của class kia nên dẫn đến **chúng sẽ cùng tạo ra hoặc cùng chết đi.** |
| Inheritance  (or Generalization) | Hay    *(Các lớp ở cuối cùng như Short Term, Long Term, Curent a/c, Savings a/c gọi là các lớp cụ thể (concrete Class). Chúng có thể tạo ra đối tượng và các đối tượng này thừa kế toàn bộ các thuộc tính, phương thức của các lớp trên. Các lớp trên như Account, Term Based, Transaction Based là những lớp trừu tượng (Abstract Class), những lớp này không tạo ra đối tượng.)* | Là quan hệ thừa kế |
| Multiplicity | IMG_256 |  |

