Bài 3: TẠO ACTIVITY - DIAGRAM TRONG PTTKPM

Xem bài học trên website để ủng hộ Kteam: Tạo Activity - Diagram trong PTTKPM

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Qua bài trước bạn đã nắm được LƯỢC ĐỒ USE – CASE là gì? **Actor**, **Use – Case** là gì? Xác định Actor như thế nào? Đặc tả Use – Case ra sao?

Ở bài này chúng ta sẽ cùng nhau thiết kế luồng đi của hệ thống. Luồng đi của dữ liệu. Thông qua **Activity – Diagram**.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- Đã từng sử dụng qua vài phần mềm
- Đã từng suy nghĩ đến việc cấu thành của một phần mềm ra sao
- Biết sử dụng máy tính cũng như các công cụ thành thạo.
- Đã đọc hiểu rõ bài GIỚI THIỆU VỀ PHÂN TÍCH THIẾT KẾ PHẦN MỀM
- Đã nắm rõ bài TẠO LƯỢC ĐỒ USE – CASE TRONG PHÂN TÍCH THIẾT KẾ PHẦN MỀM

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Activity – Diagram là gì?

- Các thành phần của một Activity - Diagram
- Cách ánh xạ sơ đồ Use – Case qua Activity - Diagram
- Ví dụ minh họa
- Bài tập

Activity – Diagram là gì?

Activity Diagram là một mô hình logic dùng để mô hình hoá các hoạt động trong một quy trình nghiệp vụ. Hay có thể hiểu. **Activity – Diagram** là sơ đồ luồng xử lý của hệ thống. Bao gồm luồng đi của dòng dữ liệu, dòng sự kiện.

Dùng để mô tả các hoạt động trong một chức năng của hệ thống. Hay có thể hiểu là mô tả luồng xử lý của một **Use – Case**.

Mô tả hoạt động chính và mối quan hệ giữa các hoạt động này trong quy trình. Hay có thể hiểu là mô tả cả luồng xử lý chính của hệ thống bao gồm các luồng con, luồng xử lý của các **Use – Case** gom lại mà thành.

Các thành phần của Activity - Diagram

Cũng như **Use – Case**. **Activity – Diagram** cũng có các thành phần cấu thành sơ đồ như hình.

- Start
- Activity
- Transition
- Decision
  - Merge
  - Branch
- Synchronization bar
  - Fork
  - Join
- End

Start

- Kí hiệu :
- Đặc trưng
  - Khởi tạo một hoạt động.
  - Một activity diagram có thể có nhiều trạng thái start.

Hay có thể hiểu là điểm bắt đầu của luồng xử lý.

Activity

- Kí hiệu : 

- Đặc trưng
  - Mô tả hành vi của đối tượng trong quy trình
  - Tên hoạt động phải ngắn gọn – đủ nghĩa

Nên đặt tên là động từ. Và mô tả đủ ý nghĩa tổng thể của hoạt động nhất có thể.

Ví dụ:

- Nhấn button Đăng nhập
- Gửi dữ liệu xuống server
- Nhận mã xác nhận

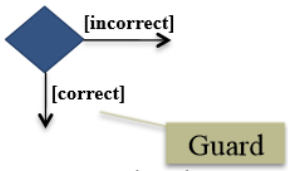
Transition

- Kí hiệu : 
- Đặc trưng
  - Mô tả sự chuyển đổi trạng thái của các hoạt động.

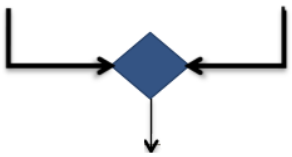
Từ hoạt động này tới hoạt động khác cần có Transition biểu thị đường đi. Lưu ý Transition có mũi tên biểu thị chiều của luồng xử lý.

Decisition

- Kí hiệu : 
- Đặc trưng
  - Tập các điều kiện kích hoạt việc chuyển trạng thái.
  - Branch
    - Mô tả điều kiện rẻ nhánh
    - Chỉ một dòng điều khiển đi vào
    - Hai hoặc nhiều dòng điều khiển ra
    - Chỉ một dòng điều khiển ra dẫn đến kết quả
    - Mỗi dòng chứa một điều kiện (guard), guard phải liên quan đến điều kiện và loại trừ nhau



  - Merge
    - Có hai hoặc nhiều dòng điều khiển đi vào
    - Chỉ một dòng điều khiển đi ra



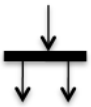
Có thể hiểu đây là ký hiệu biểu thị nút điều kiện chuyển hướng. Tùy theo trường hợp đúng hay sai của kết quả biểu thức logic bên trong ký hiệu mà có hướng di chuyển tiếp theo tương ứng.

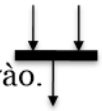
Ví dụ: 1 > 2

- (true) in ra màn hình "Tầm bậy"
- (false) in ra màn hình "Trên đời này chuyện quái gì cũng có thể xảy ra".

Synchronization bar

- Kí hiệu :
- Đặc trưng
  - Mô tả các dòng điều khiển thực hiện song song.
  - Fork
    - Mô tả một dòng điều khiển được tách ra thực hiện song song.
    - Chỉ một dòng điều khiển đi vào
    - Có hai hoặc nhiều dòng điều khiển ra.
    - Dùng fork khi các hoạt động thực hiện không quan tâm thứ tự.
  - Join
    - Kết hợp các dòng điều khiển song song (fork).
    - Có hai hoặc nhiều dòng điều khiển vào
    - Chỉ một dòng điều khiển ra
    - Dòng điều khiển ra được tạo khi tất cả các dòng cần thiết đã vào.





Ghi chú: fork và join không cần nhãn

Có thể hiểu đơn giản. Có các trường hợp cần hội tụ đủ nhiều luông điều khiển một lúc để gộp thành một luồng xử lý thì cần dùng Join.

Và đôi khi cần phải tách một luồng điều khiển ra hai hoặc nhiều luồng khác biệt nhau thì cần Fork. Và mỗi luồng của Fork hoàn toàn không lệ thuộc nhau.

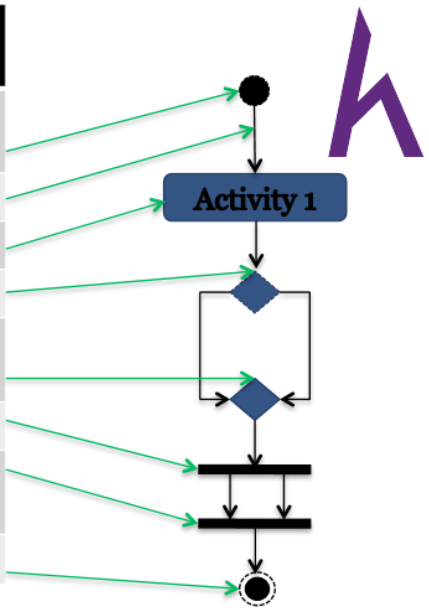
End

- Kí hiệu : 
- Đặc trưng
  - Mô tả trạng thái kết thúc quy trình.
  - Một activity diagram có một hoặc nhiều trạng thái kết thúc.

Điểm kết thúc của luồng xử lý.

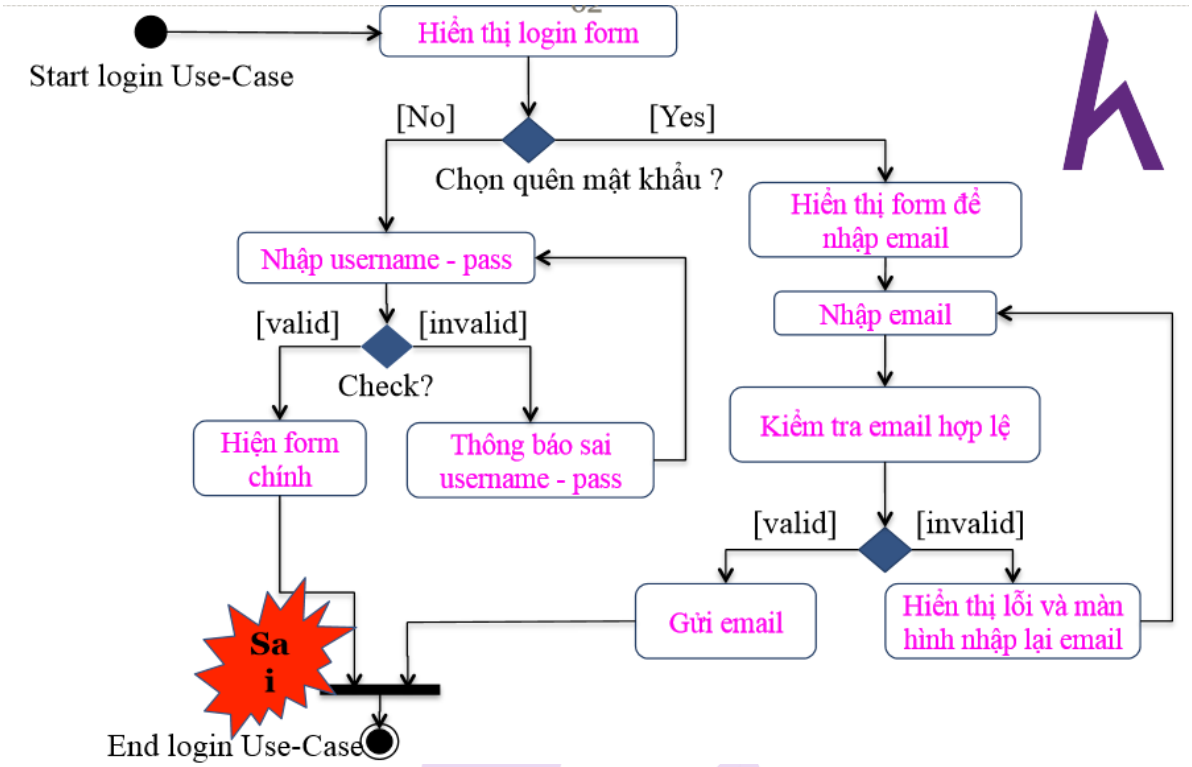
Cách ánh xạ từ sơ đồ Use – Case qua Activity - Diagram

Use-Case	Activity
Tiền điều kiện	Initial state
Yêu cầu của tác nhân	Transition
Hoạt động của hệ thống	Activity
Điều kiện rẻ nhánh (luồng sự kiện phụ)	Decision
Từ các nhánh trả về luồng sự kiện chính	Merge
Các hoạt động thực hiện song song	Fork
Kết hợp các nhánh hoạt động song song tra	Join
Hậu điều kiện	Final node

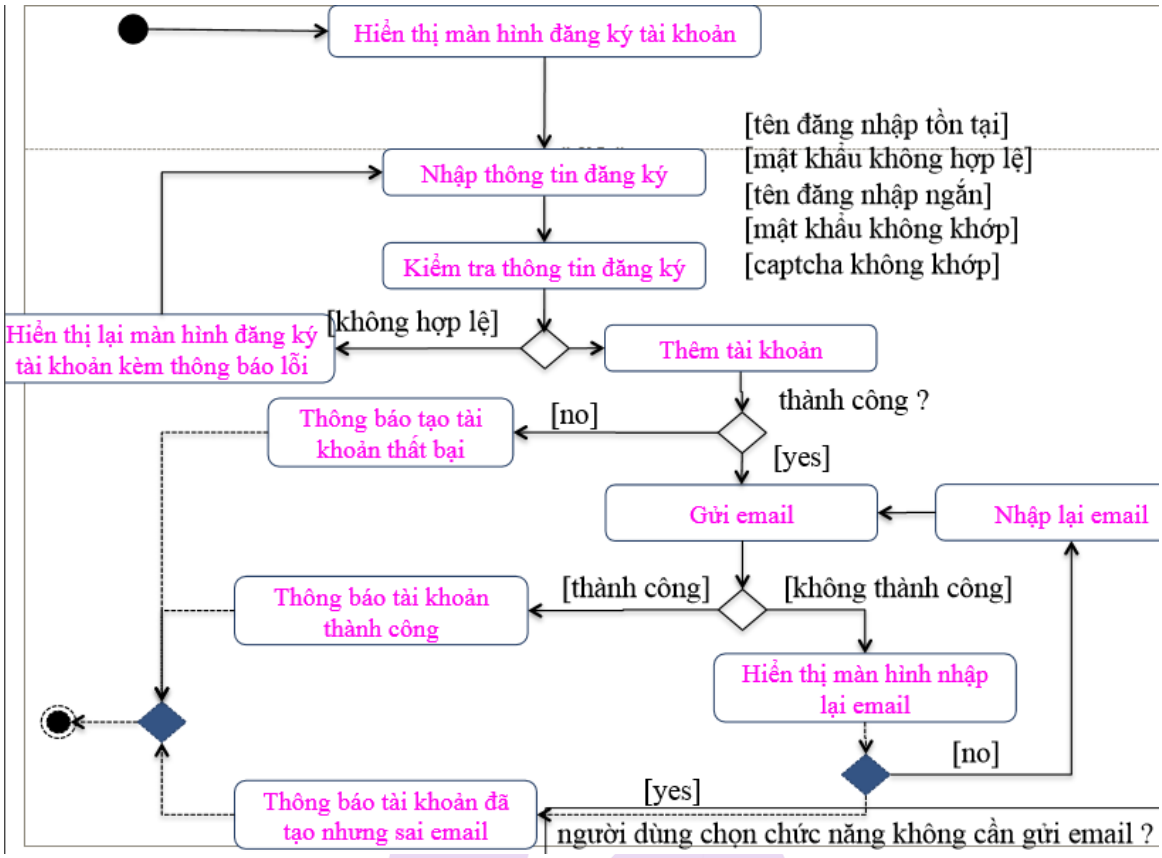


Bài tập minh họa

Activity đăng nhập



Activity đăng ký tài khoản



Kết luận

> Bài viết có sử dụng nhiều hình ảnh của tài liệu PTTKHT của trường ĐH KHTN

Qua bài này các bạn đã nắm được Activity – Diagram là gì. Cách để ánh xạ từ sơ đồ Use – Case thành Activity – Diagram.

Bài sau chúng ta sẽ cùng tìm hiểu về GIỚI THIỆU VỀ ER - DIAGRAM trong phân tích thiết kế phần mềm.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".