

Trong bài trước chúng ta đã biết vai trò của bản vẽ Use Case là rất quan trọng, nó giúp chúng ta hiểu yêu cầu, kiến trúc chức năng của hệ thống và chi phối tất cả các bản vẽ còn lại. Trong bài này chúng ta sẽ tìm hiểu về các thành phần cấu tạo nên bản vẽ này, cách xây dựng và sử dụng nó.

Bạn đang xem: Use case diagram là gì

1. Các thành phần trong bản vẽ Use Case

Đầu tiên, chúng ta xem một ví dụ về Use Case Diagram.

Hình 1. Bản vẽ Use Case về ứng dụng ATM

Nhìn bản vẽ này chúng ta thấy có hai người dùng là Customer và ATM Technician và một đối tượng sử dụng hệ thống là Bank. Bên cạnh đó nó mô tả các chức năng của hệ thống và người dùng nào dùng chức năng gì. Điều này giúp chúng ta hình dung được là chúng ta sẽ xây dựng hệ thống với những chức năng gì? Cho ai dùng.

Bây giờ chúng ta sẽ tìm hiểu kỹ hơn về các thành phần của bản vẽ.

1.1 Actor

Actor được dùng để chỉ người sử dụng hoặc một đối tượng nào đó bên ngoài tương tác với hệ thống chúng ta đang xem xét. Lưu ý, chúng ta hay bỏ quên đối tượng tương tác với hệ thống, ví dụ như Bank ở trên.

Actor được biểu diễn như sau:

Hoặc

Hình 2. Các ký hiệu của Actor

1.2 Use Case

Use Case là chức năng mà các Actor sẽ sử dụng. Nó được ký hiệu như sau:

Hình 3. Ký hiệu về Use Case

Với việc xác định các chức năng mà Actor sử dụng bạn sẽ xác định được các Use Case cần có trong hệ thống.

1.3 Relationship(Quan hệ)

Relationship hay còn gọi là connector được sử dụng để kết nối giữa các đối tượng với nhau tạo nên bản vẽ Use Case. Có các kiểu quan hệ cơ bản sau:

- Association
- Generalization
- Include
- Extend

Chúng ta sẽ lần lượt tìm hiểu về các kiểu quan hệ dưới đây.

+ Quan hệ Association

Association thường được dùng để mô tả mối quan hệ giữa Actor và Use Case và giữa các Use Case với nhau.

Hình 4. Ví dụ thể hiện Actor User sử dụng Use Case Login

+Quan hệ Generalization

Generalization được sử dụng để thể hiện quan hệ thừa kế giữa các Actor hoặc giữa các Use Case với nhau.

Hình 5. Ví dụ Actor User thừa kế toàn bộ quyền của Actor Guest

+ Quan hệ Include

Include là quan hệ giữa các Use Case với nhau, nó mô tả việc một Use Case lớn được chia ra thành các Use Case nhỏ để dễ cài đặt (module hóa) hoặc thể hiện sự dùng lại.

Hình 6. Ví dụ về quan hệ Include giữa các Use Case

Chúng ta thấy Use Case “Verify Password” có thể gộp chung vào Use Case Login nhưng ở đây chúng ta tách ra để cho các Use Case khác sử dụng hoặc để module hóa cho dễ hiểu, dễ cài

đặt.

+ Quan hệ Extend

Extend dùng để mô tả quan hệ giữa 2 Use Case. Quan hệ Extend được sử dụng khi có một Use Case được tạo ra để bổ sung chức năng cho một Use Case có sẵn và được sử dụng trong một điều kiện nhất định nào đó.

Hình 7. Ví dụ về quan hệ Extend giữa các Use Case

Trong ví dụ trên “Open Account” là Use Case cơ sở để cho khách hàng mở tài khoản. Tuy nhiên, có thêm một điều kiện là nếu khách hàng là công ty thì có thể thêm người sở hữu lên tài khoản này. Add Account Holder là chức năng mở rộng của Use Case “Open Account” cho trường hợp cụ thể nếu Actor là Công ty nên quan hệ của nó là quan hệ Extend.

1.4 System Boundary

System Boundary được sử dụng để xác định phạm vi của hệ thống mà chúng ta đang thiết kế. Các đối tượng nằm ngoài hệ thống này có tương tác với hệ thống được xem là các Actor.

System Boundary sẽ giúp chúng ta dễ hiểu hơn khi chia hệ thống lớn thành các hệ thống con để phân tích, thiết kế.

Xem thêm: Máy Tính Không Nhận Ổ Cứng Ngoài Với 4 Cách Sửa Lỗi Hiệu Quả

2. Các bước xây dựng Use Case Diagram

Chúng ta đã nắm được các ký hiệu của bản vẽ Use Case, bây giờ là lúc chúng ta tìm cách lắp chúng lại để tạo nên bản vẽ hoàn chỉnh. Thực hiện các bước sau để xây dựng một bản vẽ Use Case:

+ Bước 1: Tìm các Actor

Trả lời các câu hỏi sau để xác định Actor cho hệ thống:

- Ai sử dụng hệ thống này?
- Hệ thống nào tương tác với hệ thống này?

Xem xét ví dụ về ATM ở trên chúng ta thấy:

- Ai sử dụng hệ thống? -> Customer, ATM Technician
- Hệ thống nào tương tác với hệ thống này? -> Bank

Như vậy có 03 Actor: Customer, ATM Technician và Bank

+ Bước 2: Tìm các Actor

Trả lời câu hỏi các Actor sử dụng chức năng gì trong hệ thống? chúng ta sẽ xác định được các Use Case cần thiết cho hệ thống.

Xem xét ví dụ ở trên ta thấy:

Customer sử dụng các chức năng: Check Balance, Deposit, Withdraw và Transfer ATM technician sử dụng: Maintenance và Repair Bank tương tác với tất cả các chức năng trên.

Tóm lại, chúng ta phải xây dựng hệ thống có các chức năng: Check Balance, Deposit, Withdraw, Transfer, Maintenance và Repair để đáp ứng được cho người sử dụng và các hệ thống tương tác.

+ Bước 3: Xác định các quan hệ

Phân tích và xác định các quan hệ giữa các Actor và Use Case, giữa các Actor với nhau, giữa các Use Case với nhau sau đó nối chúng lại chúng ta sẽ được bản vẽ Use Case.

Hình 8. Bản vẽ Use Case về ATM

3. Đặc tả Use Case

Nhìn vào bản vẽ trên chúng ta nhận biết hệ thống cần những chức năng gì và ai sử dụng. Tuy nhiên, chúng ta chưa biết được chúng vận hành ra sao? Sử dụng chúng như thế nào? Để hiểu rõ hơn hệ thống chúng ta cần phải đặc tả các Use Case.

Có 2 cách để đặc tả Use Case.

Cách 1: Viết đặc tả cho các Use Case

Chúng ta có thể viết đặc tả Use Case theo mẫu sau:

Tên Use Case //Account Details**Mã số Use Case** //UCSEC35 **Mô tả tóm tắt**// Hiển thị thông tin chi tiết của Account**Các bước thực hiện** // Liệt kê các bước thực hiện**Điều kiện thoát** // Khi người dùng kích nút Close**Yêu cầu đặc biệt**// Ghi rõ nếu có**Yêu cầu trước khi thực hiện**// Phải đăng nhập**Điều kiện sau khi thực hiện** // Ghi rõ những điều kiện nếu có sau khi thực hiện Use Case này

Cách 2: Sử dụng các bản vẽ để đặc tả

Chúng ta có thể dùng các bản vẽ như Activity Diagram, Sequence Diagram để đặc tả Use case. Các bản vẽ này chúng ta sẽ bàn ở những bài tiếp theo.

4. Sử dụng UseCase Diagram

Như chúng ta đã biết Use Case Diagram có một vai trò đặc biệt quan trọng trong quá trình phân tích, thiết kế và phát triển hệ thống. Dưới đây chúng tôi liệt kê một số ứng dụng tiêu biểu của Use Case Diagram.

- Phân tích và hiểu hệ thống
- Thiết kế hệ thống.
- Làm cơ sở cho việc phát triển, kiểm tra các bản vẽ như Class Diagram, Activity Diagram, Sequence Diagram, Component Diagram.
- Làm cơ sở để giao tiếp với khách hàng, các nhà đầu tư.
- Giúp cho việc kiểm thử chức năng, kiểm thử chấp nhận.

5. Kết luận

Đến đây, chúng ta đã tìm hiểu được bản vẽ đầu tiên và rất quan trọng (use case diagram), các bạn cần tiếp tục thực hành để nắm rõ hơn về bản vẽ này cũng như cách xây dựng và sử dụng chúng trong quá trình phát triển sản phẩm phần mềm.

Xem thêm: Lip Balm Là Gì ? Top Son Môi Dưỡng Tốt Nhất Được Yêu Thích? Son Dưỡng Môi Lip Balm

Để giúp các bạn hiểu rõ hơn về bản vẽ Use Case trong bài tiếp theo chúng ta sẽ thực hiện qua từng bước bài thực hành xây dựng Use Case Diagram.

Chuyên mục: Công Nghệ 4.0

XEM THÊM CÁC THÔNG TIN KHÁC TẠI: <https://pubgonpc.com/>

Bài viết [Tại Sao Phải Có Use Case Diagram Là Gì ? Use Case Diagram](#) đã xuất hiện đầu tiên vào ngày [PUBGONPC.COM](#).

via PUBGONPC.COM

<https://pubgonpc.com/tai-sao-phai-co-use-case-diagram-la-gi-use-case-diagram/>