|  |  |  |
| --- | --- | --- |
| Macintosh HD:Users:hoangnguyen:Documents:TDC:fitlog_blue.png | TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC  **Khoa Công Nghệ Thông Tin** | Macintosh HD:Users:hoangnguyen:Documents:TDC:logoTDC_blue.png |
|  |  |  |

**Chuyên đề PTW2** | HKI – [2019 – 2020]

PRJ – REPORT

**BÁO CÁO TỔNG QUAN VỀ UML VÀ USE CASE**

**Nhóm A:**

* **Bùi Thị Nhựt Anh –** 17211TT1884
* **Huỳnh Thị Mỹ nhiên –** 17211TT0988

**BẢNG PHÂN CÔNG TRONG NHÓM**

|  |  |  |
| --- | --- | --- |
| MSSV | Họ & Tên | Công việc |
| 17211TT1884 | Bùi Thị Nhựt Anh | USE CASE |
| 17211TT0988 | Huỳnh Thị Mỹ Nhiên | UML |

**MỤC LỤC**

[I. UML 4](#_Toc20943680)

[1. Giới thiệu 4](#_Toc20943681)

[2. Mục tiêu của UML 4](#_Toc20943682)

[3. Biểu đồ cơ bản của UML 4](#_Toc20943683)

[II. USE CASE 5](#_Toc20943684)

[1. Giới thiệu 5](#_Toc20943685)

[2. Các thành phần của user case 5](#_Toc20943686)

[3. Các quan hệ trong Use Case 5](#_Toc20943687)

[4. Các giai đoạn xây dựng một Use Case Diagram 6](#_Toc20943688)

**MỤC LỤC HÌNH ẢNH**

[Hình 1 UML 4](file:///C:\Users\COMPUTER\Downloads\UMLvaUSECASE.docx#_Toc23942261)

[Hình 2 Use Case 5](file:///C:\Users\COMPUTER\Downloads\UMLvaUSECASE.docx#_Toc23942262)

[Hình 3 Actor 5](file:///C:\Users\COMPUTER\Downloads\UMLvaUSECASE.docx#_Toc23942263)

[Hình 4 Use Case 5](file:///C:\Users\COMPUTER\Downloads\UMLvaUSECASE.docx#_Toc23942264)

[Hình 5 Quan hệ bao gồm (Include) 6](file:///C:\Users\COMPUTER\Downloads\UMLvaUSECASE.docx#_Toc23942265)

[Hình 6 Quan hệ mở rộng (Extend) 6](file:///C:\Users\COMPUTER\Downloads\UMLvaUSECASE.docx#_Toc23942266)

[Hình 7 Quan hệ tổng quát - Generalization 7](#_Toc23942267)

UML

Hình UML

Giới thiệu

UML (Unified Modeling Language) là ngôn ngữ dành cho việc đặc tả, hình dung, xây dựng và làm tài liệu của các hệ thống phần mềm.

Mục tiêu của UML

* Cho phép phát triển và trao đổi những mô hình mang nhiều ý nghĩa.
* Cung cấp khả năng mở rộng và chuyên môn hoá để mở rộng những khái niệm cốt lõi.
* Độc lập với ngôn ngữ lập trình chuyên biệt và các tiến trình phát triển.
* Cung cấp nền tảng về sự hiểu biết ngôn ngữ mô hình hoá.
* Khuyến khích và hỗ trợ sự phát triển của các công cụ hướng đối tượng.
* Hỗ trợ những khái niệm phát triển cấp độ cao như collaboration, framework, pattern and component.

Biểu đồ cơ bản của UML

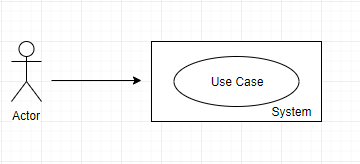
* **Biểu đồ lớp (Class Diagram)**: là xương sống của hầu như tất cả các phương pháp hướng đối tượng, bao gồm cả UML. Chúng mô tả các cấu trúc tĩnh của hệ thống.
* **Biểu đồ gói ( Package Diagram):** là tập hợp các class diagram. Các package diagram thiết lập mối quan hệ giữa các pakage, trong đó pakage là những nhóm phần tử của hệ thống có mối quan hệ liên quan đến nhau.
* **Biểu đồ chức năng**: đưa ra cách nhìn bao quát (từ trên xuống) cách sử dụng của hệ thống cũng như cách nhìn hệ thống từ bên ngoài.
* **Biểu đồ trạng thái**: mô tả những hành động của các lớp và dối tượng riêng lẽ, mô tả trình tự những trạng thái mà các đối tượng sẽ đi qua. Tích hợp một cách tốt nhất với thực tiễn
* **Biểu đồ hoạt động**: mô tả tiến trình xử lý và trình tự những hành động trong tiến trình xử lý. Trông nó giống như biểu đồ tiến trình (flowchart) bởi vì nó mô tả dòng làm việc từ hoạt động sang hoạt động và từ hoạt động sang trạng thái.
* **Biểu đồ thành phần**: cho chúng ta cách nhìn vật lý của mô hình thực tế. Nó thể hiện rõ cho chúng ta thấy sự cấu tạo và sự phụ thuộc giữa các thành phần của phần mềm bao gồm mã nguồn, mã nhị phân (binary code) và những thành phần có khả năng thực thi.
* **Biểu đồ triển khai**: mô tả các tài nguyên vật lý trong hệ thống, bao gồm các nút (node), thành phần và kết nối. Mỗi mô hình chỉ bao gồm một deployment diagram hiển thị ánh xạ giữa những tiến trình xử lý tới thiết bị phần cứng.

USE CASE

Giới thiệu

Use Case là một kỹ thuật được dùng trong kỹ thuật phần mềm và hệ thống để nắm bắt yêu cầu chức năng của hệ thống.

Use case mô tả sự tương tác đặc trưng giữa người dùng bên ngoài (actor) và hệ thống.

Nó được đặt tên giống Động từ hoặc Động từ + cụm danh từ.

Hình Use Case



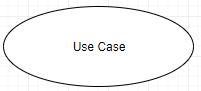
Hình Actor

Các thành phần của user case

* **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

* **Use Case**



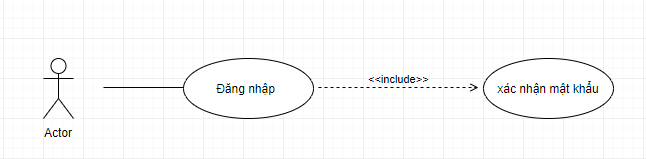
Hình Use Case

Use Case là chức năng mà các Actor sẽ sử dụng.

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

Các quan hệ trong Use Case

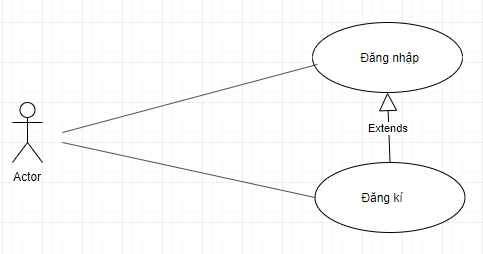
* **Quan hệ bao gồm (Include)**

Use Case A bao gồm Use Case B để thực hiện hay hoàn thành Use Case A thì cần phải thực hiện hay hoàn thành Use Case B.

Hình Quan hệ bao gồm (Include)

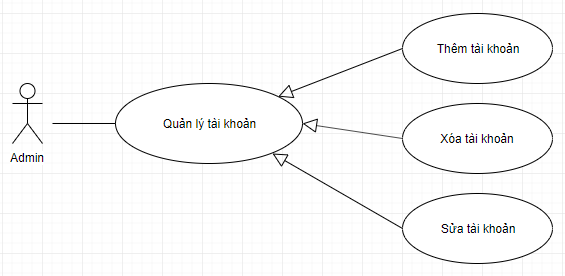
* **Quan hệ mở rộng (Extend)**

Use Case A mở rộng Use Case B nếu để thực hiện hay hoàn thành Use Case A có thể phải thực hiện hay hoàn thành Use Case B phụ thuộc vào điều kiện cụ thể.



Hình Quan hệ mở rộng (Extend)

* **Quan hệ tổng quát (Generalization)**

Use Case A có quan hệ tổng quát với Use Case B nếu Use Case B là một trường hợp cụ thể/ chi tiết của Use Case B.

Hình Quan hệ tổng quát - Generalization

Các giai đoạn xây dựng một Use Case Diagram

* Giai đoạn mô hình hóa

Bước 1: Thiết lập ngữ cảnh của hệ thống đích.

Bước 2: Chỉ định các Actor.

Bước 3: Chỉ định các Use Case.

Bước 4: Định nghĩa các quan hệ giữa các Actor và các Use Case.

Bước 5: Đánh giá các Actor và các Use Case để tìm cách chi tiết hóa.

* Giai đoạn cấu trúc

Bước 6: Đánh giá các Use Case cho quan hệ phụ thuộc «include».

Bước 7: Đánh giá các Use Case cho quan hệ phụ thuộc «extend».

Bước 8: Đánh giá các Use Case cho quan hệ generalizations.