|  |  |  |
| --- | --- | --- |
| 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 PTWEB2** | HKI – [2019 – 2020]

**BÁO CÁO TÌM HIỂU VỀ UML/Use Case**

**Nhóm K gồm các thành viên:**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 8 | Lê Minh Đức | 17211TT2526 |
| 29 | Lê Minh Tiến | 17211TT2494 |
| 20 | Phan Đỗ Hoàng Nhật | 17211TT2669 |
| 19 | Phạm Thành Năng | 17211TT3867 |
| 37 | Nguyễn Minh Cảnh | 1451TT0989 |

**Bảng phân công công việc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ và Tên | Phân công công việc | Thời gian phân việc | Thời gian hoàn thành |
| 8 | Lê Minh Đức | Trình bày về UML/Use case:   * UML là gì? Giới thiệu UML | 10h 03/10/2019 | 20h  03/10/2019 |
| 29 | Lê Minh Tiến | Trình bày về UML/Use case:   * Các mô hình UML | 10h 03/10/2019 | 20h  03/10/2019 |
| 20 | Phan Đỗ Hoàng Nhật | Trình bày về UML/Use case:   * Các công cụ vẽ UML phổ biến | 10h 03/10/2019 | 20h  03/10/2019 |
| 19 | Phạm Thành Năng | Trình bày về UML/Use case:   * Use case   + Khái niệm  + Các thành phần chính (mối quan hệ giữa các Use case) | 10h 03/10/2019 | 20h  03/10/2019 |
| 37 | Nguyễn Minh Cảnh | Trình bày về UML/Use case:   * Use case   + Các thành phần chính(Tác nhân là gì?, các Use case) | 10h 03/10/2019 | 20h  03/10/2019 |

**MỤC LỤC**

Trình bày về UML/Use case

[**I.** **UML** 4](#_Toc21101429)

[**1.** **Giới thiệu về UML** 4](#_Toc21101430)

[**2.** **Các mô hình UML** 4](#_Toc21101431)

[**3.** **Các công cụ vẽ UML phổ biến** 4](#_Toc21101432)

[**II.** **Use Case** 4](#_Toc21101433)

[**1.** **Khái niệm** 4](#_Toc21101434)

[**2.** **Các thành phần chính** 4](#_Toc21101435)

[**2.1** **Use case, tác nhân là gì?** 4](#_Toc21101436)

[*Hình 1. Ký hiệu* 5](#_Toc21101437)

[*Hình 2. Nhiều Actor sử dụng một use case* 5](#_Toc21101438)

[**2.2** **Các Use case** 5](#_Toc21101439)

[**3.** **Mối quan hệ giữa các Use case** 6](#_Toc21101440)

[*Hình 3. Ví dụ về Include* 6](#_Toc21101441)

[*Hình 4. Ví dụ* 6](#_Toc21101442)

[*Hình 5. Ví dụ Extend* 7](#_Toc21101443)

[*Hình 6. Ví dụ Extend* 7](#_Toc21101444)

1. **UML**
2. **Giới thiệu về UML**

* UML( Unified Modeling Language ) là ngôn ngữ mô hình gồm các ký hiệu đồ họa mà các phương pháp hướng đối tượng sử dụng để thiết kế các hệ thống thông tin một cách nhanh chóng
* UML là ngôn ngữ trực quan
* Giúp công việc phát triển được xử lý nhất quán,giảm thiểu lỗi xảy ra
  + Giúp dễ hình dung hơn cấu trúc của hệ thống
  + Hiệu quả hơn trong việc liên lạc, trao đổi
  + Trong tổ chức
  + Bên ngoài tổ chức

1. **Các mô hình UML**

⮚ Các mô hình UML có thể kết nối trực tiếp với rất nhiều ngôn ngữ lập trình.

❑ Ánh xạ sang Java, C++, Visual Basic…

❑ Cho phép các kỹ nghệ xuôi (chuyển UML thành mã nguồn)

❑ Cho phép kỹ nghệ ngược (xây dựng mô hình hệ thống từ mã nguồn)

1. **Các công cụ vẽ UML phổ biến**

* Có rất nhiều công cụ được sử dụng để vẽ các bản vẽ UML rất chuyên nghiệp như:
  + Microsoft Visio
  + Enterprise Architect
  + Rational Rose,…
* Các công cụ có cách sử dụng khá giống nhau và ký hiệu của các bạn vẽ trên UML cũng đã thống nhất nên việc nắm bắt một công cụ khi chuyển sang làm việc với một công cụ khá không quá khó khăn.

1. **Use Case**
2. **Khái niệm**

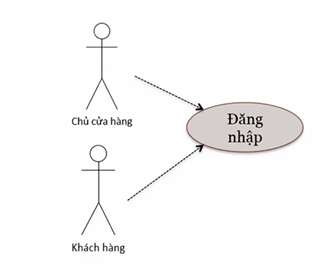
* Use case là đối tượng người dùng muốn nhận được từ hệ thống.
* Tên Use case thường ngắn gọn, rõ ràng, cụ thể và miêu tả đủ nghĩa của đối tượng người dùng.

1. **Các thành phần chính** 
   1. **Use case, tác nhân là gì?**

⮚ Là tác nhân bên ngoài hệ thống và phải có tương tác với hệ thống (actor có thể là người , phần cứng , phần mềm khác ) , 3 hình thức:

* + Tương tác trao đổi thông tin với hệ thống hoặc sử dụng chức năng.
  + Cung cấp đầu vào hoặc nhận thông tin đầu ra từ hệ thống.
  + Không điều khiển hoạt động của hệ thống.

*Hình 1. Ký hiệu*

* Mỗi Actor được phép sử dụng một hay nhiều chức năng trong hệ thống như :
  + Nhân viên được phép them xóa sửa hóa đơn.
  + Admin được phép toàn quyền với hệ thống.
* Một chức năng có thể cho phép nhiều Actor sử dụng.
  + Nhân viên và Quản lý đều được phép thông báo giờ làm việc của mình với hệ thống.
* Nhiều Actor có thê có các quyền hạn giống nhau như :
  + Admin và Quản lý đều có thể xem thống kê cua hệ thống.
* Một hay nhiều Actor còn tùy vào ngữ cảnh, ví dụ : nhân viên có thể sử dụng phần mềm thì có thể có nhiều nhân viên.
* Ví dụ : Nhiều Actor sử dụng một use case

*Hình 2. Nhiều Actor sử dụng một use case*

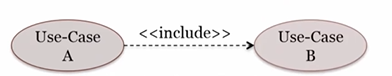
* 1. **Các Use case**

⮚ Các Use case được biểu diễn bởi các hình elip. Một use case là một chuỗi các hành động mà hệ thống thực hiện mang lại kết quả quan sát được đối với Actor.

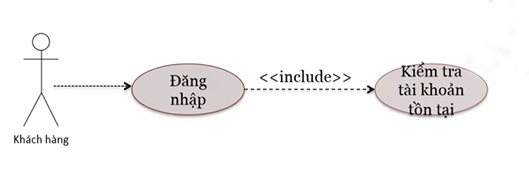
⮚ Có thể hiểu use case là một chức năng của hệ thống , mang một ý nghĩa nhất định đối với người dùng.

* Ví dụ :
  + Khách hàng đăng ký tài khoản và mua sản phẩm ( khách hàng : Actor , đăng ký tài khoản và mua sản phẩm : use case ).
  + Chức năng xuất báo cáo in ra excel ( xuất báo cáo : use case , excel : actor ).

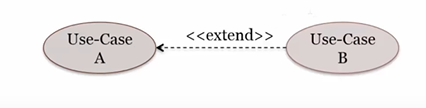
1. **Mối quan hệ giữa các Use case**

* Có 2 loại quan hệ : include & extend .
* Include: ví dụ Use case A <<include>> đến Use case B thì trong quá trình thực thi thì Use case A sẽ luôn cần thực hiện Use case B.

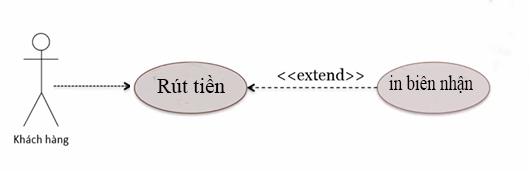
*Hình 3. Ví dụ về Include*

* Ví dụ:

*Hình 4. Ví dụ*

* Extend: ví dụ Use case A <<extend>> đến Use case B thì trong quá trình thực thi thì Use case A có thể hoặc không thực hiện Use case B.

*Hình 5. Ví dụ Extend*

* Ví dụ:

*Hình 6. Ví dụ Extend*