|  |
| --- |
|  |
| |  |  | | --- | --- | | logo truong cao dang cong nghe thu duc |  | | TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC  **KHOA CÔNG NGHỆ THÔNG TIN** | |
| |  |  | | --- | --- | |  |  | |

**BÁO CÁO UML VÀ USE CASE**

**CHUYÊN ĐỀ WEB 2**

**Giảng Viên:** **Phan Thanh Nhuần**

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** |
| 1 | Phan Trung Hiếu | 17211TT2487 |
| 2 | Cáp Tống Xuân Trung | 17211TT2506 |
| 3 | Lê Việt Hoàng | 17211TT2356 |
| 4 | Nguyễn Thanh Phong | 17211TT2726 |
| 5 | Nguyễn Hoàng Minh Hiếu | 17211TT1584 |

***Mục tiêu họp của nhóm:***

1. **Thảo luận, bàn bạc chia câu về thuyết trình.**
2. **Sắp xếp thời gian thực hiện từng công việc để hoàn thành.**
3. **Phân công nhiệm vụ rõ ràng cho từng thành viên trong nhóm.**

Địa điểm: Tại lớp nhóm bàn bạc trao đổi và phân chia công việc cho từng thành viên tìm hiểu và làm báo cáo kịp ngày 5/10/2019.

Thời gian bắt đầu và kết thúc:

Thời gian bắt đầu ngày (kể từ ngày nhận được thầy giao) vào buổi tối (tạo nhóm qua Facebook ,tổ chức làm bài và trao đổi thông tin trực tuyến).

1. **Kết quả buổi họp:**

Các thành viên hiểu được nội dung của từng phần mà mình được giao.

Các thành viên hiểu nhau, đoàn kết hơn.

Phân chia công việc hợp lý nên các thành viên hoàn thành công việc được giao.

1. **Nội dung:**
2. **Trình bày UML**
3. **Trình bày Use case**
4. **Tổ chức công việc:**

Kiểm tra và chỉnh lỗi trong suốt quá trình làm báo cáo.

Cùng nhau thảo luận khi gặp một vấn đề khó khăn.

Viết file báo cáo phân công công việc.

1. **Vấn Đề gặp phải trong quá trình làm báo cáo**

* Một số phần sẽ không thể tìm được ở trên mạng, gây khó khăn cho nhóm.

|  |
| --- |
|  |

1. **Cách giải quyết vấn đề**

* Cùng nhau giải đáp những vấn đề khó khăn.
* Tìm hiểu thêm trên mạng.

1. **Bảng phân công công việc :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | | **Người phụ trách** | **Mô tả nội dung công việc** | **Bắt đầu** | **Kết thúc** | **Kết quả** |
| 1. | Phan Trung Hiếu | | Giới thiệu, các mô hình, tại sao chúng ta cần UML. | 4/9/2019 | 4/9/2019 | Hoàn thành tốt |
| 2. | Nguyễn Hoàng Minh Hiếu | | Các công cụ phổ biến, các dạng biểu đồ của UML. | 4/9/2019 | 4/9/2019 | Hoàn thành tốt |
| 3. | Lê Việt Hoàng | | Khái niệm, các thành phần chính use case. | 4/9/2019 | 4/9/2019 | Hoàn thành tốt |
| 4. | Cáp Tống Xuân Trung | | Mối quan hệ và các bước xây dựng use case. | 4/9/2019 | 4/9/2019 | Hoàn thành tốt |
| 5 | Nguyễn Thanh Phong | | Mục đích và ví dụ sơ đồ use case. | 4/9/2019 | 4/9/2019 | Hoàn thành tốt |

TP. Hồ Chí Minh, ngày 4 tháng 10 năm 2019

Thư Kí Trưởng Nhóm

Trung Hiếu

Cáp Tống Xuân Trung Phan Trung Hiếu

Mục Lục

[**I.** **UML** 6](#_Toc21104360)

[**1.Giới thiệu về UML** 6](#_Toc21104361)

[**2. Các mô hình UML** 6](#_Toc21104362)

[**3. Tại sao chúng ta cần UML?** 6](#_Toc21104363)

[**4.Các công cụ vẽ UML phổ biến** 7](#_Toc21104364)

[**5. Các dạng biểu đồ của UML** 7](#_Toc21104365)

[**II.Use case** 9](#_Toc21104366)

[**1. Khái niệm** 9](#_Toc21104367)

[**2.Các thành phần chính** 9](#_Toc21104368)

[***2.1 Use case tác nhân là gì?*** 9](#_Toc21104369)

[***2.2 Các Use case*** 9](#_Toc21104370)

[**3. Mối quan hệ giữa các Use case** 10](#_Toc21104371)

[**4. Các bước xây dựng use case** 11](#_Toc21104372)

[**5. Mục đích của use case** 12](#_Toc21104373)

[**6. Ví dụ sơ đồ use case** 12](#_Toc21104374)

1. **UML**

* **1.Giới thiệu về UML**
* Ngôn ngữ mô hình hóa thống nhất (tiếng Anh: Unified Modeling Language, viết tắt thành UML.) UML là một ngôn ngữ để:
  + Trực quan hóa (visualizing)
  + Đặc tả (specifying)
  + Xây dựng (constructing)
  + Tài liệu hóa (documenting)
* 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

## **2. 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…
  + Các bảng trong RDBMS hoặc kho lưu trữ trong OODBMS
  + 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)

## **3. Tại sao chúng ta cần UML?**

* + Một ngôn ngữ mô hình hóa chuẩn, trực quan bằng hình ảnh
  + Phát triển hệ thống hiệu quả, thiết kế chính xác và hiệu quả
  + Dễ dàng giao tiếp giữa các nhóm trong cùng project
  + Giao tiếp dễ dàng giữa các người liên quan đến project (khách hàng, nhà phát triển,…)
  + Cho “cái nhìn tổng thể” về project

## **4.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.

## **5. Các dạng biểu đồ của UML**

1. **Biểu Đồ Lớp (Class Diagram)**

- 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.

1. **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

1. **Biểu Đồ Chức Năng(Uses case diagram)**

* Đưa ra cái nhìn bao quát cách sử dụng hệ thống, cũng như cách nhìn hệ thống từ bên ngoài.
* Hiển thị những chức năng của hệ thống.
* Dùng trong quá trình phân tích hệ thống để nắm bắt được yêu cầu của hệ thống và hiểu được sự hoạt động của hệ thống.

1. **Biểu Đồ Tương Tác(Collaboration Diagram )**

- Được sử dụng để mô tả các hoạt động và các hành động được thực hiện trong một use case.

- Đặc tả cho hành vi được diễn tả như một luồng thực thi

thông qua sự sắp xếp thứ tự của các đơn vị nhỏ hơn.

- Các đơn vị nhỏ hơn bao gồm các hoạt động lồng nhau

và các hành động riêng lẻ cơ bản

1. **Biểu Đồ Tiến Trình(Sequence diagram)**

- Mô tả sự tương tác của các lớp trong trình tự về thời gian. Những mô hình này được liên kết với phương pháp biểu đồ bao gồm các đối tượng, liên kết và thông báo.

- Phân tích và thiết kế hệ thống bởi vì nó khá đơn giản.

1. **Biểu Đồ Trạng Thái (Statechart Diagram)**

* Mô tả những hành động của các lớp và đối tượng riêng lẻ
* Mô tả trình tự những trạng thái mà các đối tượng sẽ đi qua.

1. **Biểu Đồ Hoạt Động (Activity Diagram)**

* Mô tả tiến trình xử lý và trình tự những hành động trong tiến trình xử lý
* Activity diagram nó giúp bạn có thể hiểu được toàn bộ tiến trình hoạt động.

1. **Biểu Đồ Thành Phần (Component Diagram)**

* Component Diagram 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 và những thành phần có khả năng thực thi.

1. **Biểu Đồ Triển Khai (Deployment Diagram)**

* Deployment Diagrams 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.

# **II. Use case**

## **1. Khái niệm**

* Use case là đối tượng người dùng muốn nhận được từ hệ thống.
* Nó được đặt tên giống Động từ hoặc Động từ + cụm danh từ.
* Tên Use case thường ngắn gọn, rõ ràng, cụ thể và miêu tả đủ nghĩa của đối tượng người dùng.

## **2.Các thành phần chính**

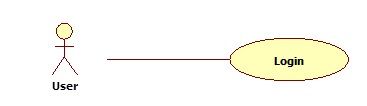
### ***2.1 Use case tác nhân là gì?***

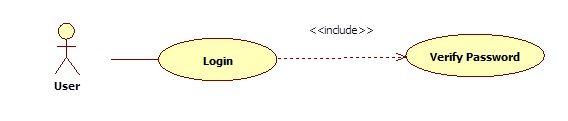
* Là đối tượng bên ngoài tương tác với hệ thống theo 3 hình thức:
  + Tương tác trao đổi thông tin với hệ thống hoặc sử dụng chức năng.
  + Cung cấp đầu vào hoặc nhận thông tin đầu ra từ hệ thống.
  + Không điều khiển hoạt động của hệ thống.
* Đặt tên: theo vai trò, không theo tên cụ thể vì nó là lớp.
* Kí hiệu:
* 
* Hình 1. Kí hiệu use case tác nhân

### ***2.2 Các Use case***

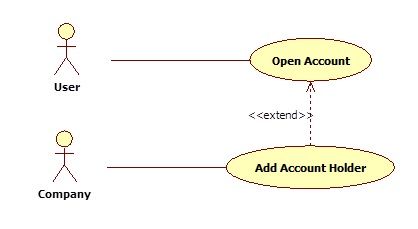
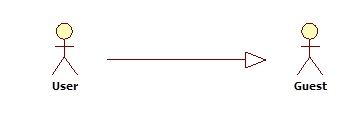
* Đây là thành phần cơ bản của biểu đồ use case. Các use case được biểu diễn bởi các hình elip.Tên các use case thể hiện một chức năng xác định của hệ thống.
* Các Use case được kí hiệu bằng hình elips.
* Kí hiệu:
* 
* Hình 2. Kí hiệu use case

## **3. Mối quan hệ giữa các Use case**

* Association: thường được dùng để mô tả mối quan hệ giữa Actor và Use Case và giữa các Use Case với nhau.
* VD:
* 
* Hình 3. Sơ đồ quan hệ actor và use case
* 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.
* VD:



Hình 4. Sơ đồ quan hệ include

* Extent: 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.
* VD:
* 
* Hình 5. Sơ đồ quan hệ Extend
* Generalization: được sử dụng để thể hiện quan hệ thừa kế giữa các Actor hoặc giữa các Use Case với nhau.
* VD:
* 
* Hình 6. Sơ đồ Generalization

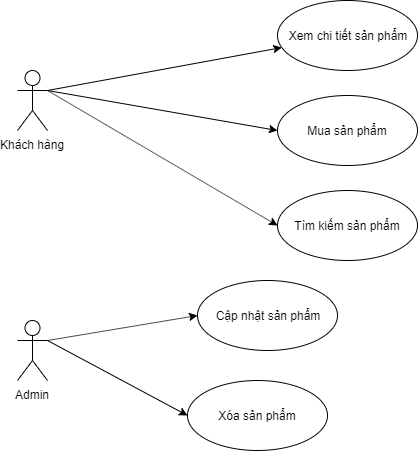
## **4. Các bước xây dựng use case**

* 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.

## **5. Mục đích của use case**

* Đưa chính xác những hạn chế của hệ thống.
* Hiều rõ những yều cầu mà khách hàng, người sử dụng mong đợi.
* Để đơn giản hóa việc thay đổi và mở rộng hệ thống.
* Hệ thống là một hệ thống phần mềm, doanh nghiệp,…

## **6. Ví dụ sơ đồ use case**

* 
* Hình 7. Sơ đồ bán hàng