|  |  |  |
| --- | --- | --- |
| 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 web 2**

**BÁO CÁO UML VÀ USECASE**

**Nhóm B:**

* **Bùi Hửu Thắng –** 17211TT3157
* **Nguyễn Hải Phúc Khang –** 17211TT3103
* **Trần Minh Thuận –** 17211TT3143
* **Nguyễn Minh Trinh –** 17211TT3608
* **Đoàn Vũ Quốc Khiêm** **–** 17211TT3640

**Mục Lục**

[**CHƯƠNG 1 : TÌM HIỂU UML** 3](#_Toc21155573)

[**I.** **LÝ THUYẾT** 3](#_Toc21155574)

[1. UML là gì? 3](#_Toc21155575)

[2. Mục Tiêu Của Uml 4](#_Toc21155576)

[3. Một số dạng biểu đồ UML phổ biến 5](#_Toc21155577)

[**II.** **Chương trình vẽ** 10](#_Toc21155578)

[1. Giới thiệu công cụ StarUML: 10](#_Toc21155579)

[2. Cách tạo các Diagram 10](#_Toc21155580)

[3. Cách tạo ra 1 bản vẽ: 11](#_Toc21155581)

[**CHƯƠNG 2 : TÌM HIỂU USECASE** 13](#_Toc21155582)

[**1.** **Use Case là gì ?** 13](#_Toc21155583)

[2.1 Actor 14](#_Toc21155584)

[2.2 Use Case 14](#_Toc21155585)

[2.3 Các quan hệ trong user case 14](#_Toc21155586)

[2.3.1 Quan hệ gộp (Includes) 15](#_Toc21155587)

[2.3.2 Quan hệ mở rộng (Extends): 15](#_Toc21155588)

[2.3.3 Quan hệ kết hợp (Association): 16](#_Toc21155589)

**Mục lục hình ảnh**

[HÌNH 1 : UML 5](#_Toc21155177)

[HÌNH 2 : StartUML 8](#_Toc21155178)

[HÌNH 3 : EclipseUML 8](#_Toc21155179)

[HÌNH 4 : UmlDesigner 8](#_Toc21155180)

[HÌNH 5 : Argo UML 9](#_Toc21155181)

[HÌNH 6 : Enterprise Architect 9](#_Toc21155182)

[HÌNH 7 : SmartDraw 9](#_Toc21155183)

[HÌNH 8 : Visual Paradigm for UML 10](#_Toc21155184)

[HÌNH 9 : Microsoft Visio 10](#_Toc21155185)

[HÌNH 10 : IBM Rational Software Architect 10](#_Toc21155186)

[HÌNH 11 : Giao diện chính của StarUML 11](#_Toc21155187)

[HÌNH 12 : Cách tạo Diagram 12](#_Toc21155188)

[HÌNH 13 : Cách tạo bản vẻ 13](#_Toc21155189)

[HÌNH 14 : Usecase 14](#_Toc21155190)

[HÌNH 15 : Actor 15](#_Toc21155191)

[HÌNH 16 : Includes 16](#_Toc21155192)

[HÌNH 17 : Extends 16](#_Toc21155193)

[HÌNH 18 : Association 17](#_Toc21155194)

[HÌNH 19 : Generalization 17](#_Toc21155195)

## 

## CHƯƠNG 1 : TÌM HIỂU UML

1. **LÝ THUYẾT**
2. UML là gì?

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.

UML tạo cơ hội để viết thiết kế  hệ thống, bao gồm những khái niệm như tiến trình nghiệp vụ và các chức năng của hệ thống.

Cụ thể, nó hữu dụng cho những ngôn ngữ khai báo, giản đồ cơ sở dữ liệu, thành phần phần mềm có khả năng tái sử dụng.

UML được phát triển bởi Rational Rose và một số nhóm cộng tác, nó nhanh chóng trở thành một trong những ngôn ngữ chuẩn để xây dựng hệ thống phần mềm hướng đối tượng (Object-Oriented).



HÌNH 1 : UML

1. Mục Tiêu Của Uml

UML cung cấp cho người dùng một ngôn ngữ mô hình hoá trực quan sẵn sàng để dùng và có ý nghĩa:

* 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.
* Tích hợp một cách tốt nhất với thực tiễn.

1. Một số dạng biểu đồ UML phổ biến

#### 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.

#### Biểu Đồ Gói (Package Diagram):

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:

Uses case diagram 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 này hiển thị những chức năng của hệ thống hoặc các lớp và tương tác của hệ thống với thế giới bên ngoài như thế nào. Uses case diagram được 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.

#### Biểu Đồ Tương Tác:

Collaboration Diagram cung cấp về cách nhìn sự tương tác hoặc mối quan hệ có cấu trúc giữa các đối tượng trong mô hình hiện thời. Collaboration Diagram bao gồm các đối tượng, liên kết và thông báo. Sử dụng mô hình như là một phương tiện chính để mô tả những tương tác và cách giải quyết của các hành vi trong hệ thống.

#### 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 case (tình huống). Sequence diagram hiển thị cho bạn từng bước những sự kiện xảy ra trong phương pháp case. Biểu đồ này là cách tốt nhất để phân tích và thiết kế hệ thống bởi vì nó khá đơn giản và dễ lĩnh hội. Đây là một ví dụ nhỏ mô tả một quá trình phục vụ việc bấm nút đi thang máy.

#### Biểu Đồ Trạng Thái (Statechart Diagram):

Bạn sử dụng 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.

#### Biểu Đồ Hoạt Động (Activity Diagram):

Activity Diagram 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. Khi xây dựng activity diagram nó giúp bạn có thể hiểu được toàn bộ tiến trình hoạt động. Nó rất hữu dụng khi bạn mô tả những tiến trình song song hoặc mô tả  một vài tương tác trong use case.

#### 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 (binary code) và những thành phần có khả năng thực thi.

#### 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.

1. Các công cụ của UML

* Công cụ mã nguồn mở:



HÌNH 2 : StartUML



HÌNH 3 : EclipseUML



HÌNH 4 : UmlDesigner



HÌNH 5 : Argo UML

* Công cụ thương mại:



HÌNH 6 : Enterprise Architect



HÌNH 7 : SmartDraw



HÌNH 8 : Visual Paradigm for UML



HÌNH 9 : Microsoft Visio

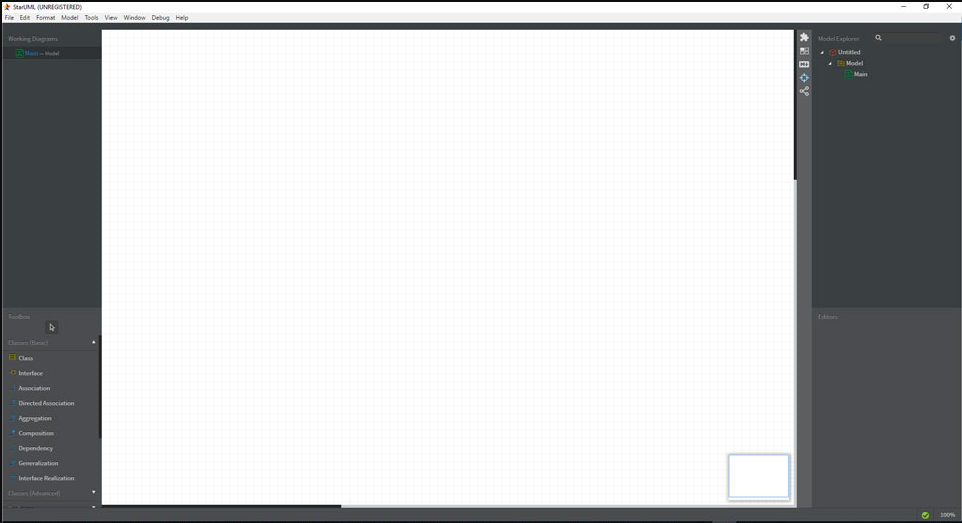
Hình 9.Microsoft Visio



HÌNH 10 : IBM Rational Software Architect

1. **Chương trình vẽ**
2. Giới thiệu công cụ StarUML:

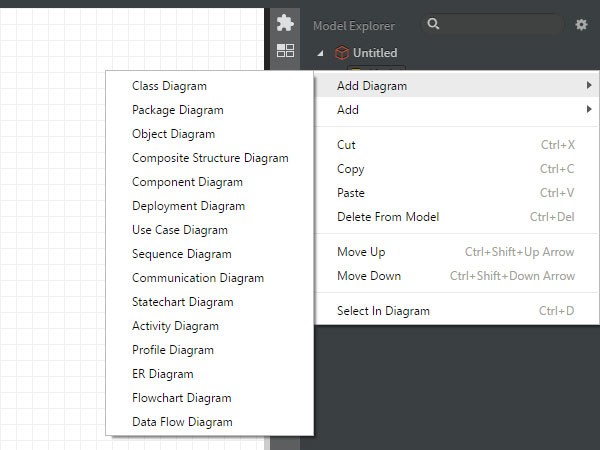
* Hiện nay có rất nhiều công cụ được sử dụng để vẽ các bản vẽ UML rất chuyên nghiệp như Gliffy, Microsoft Visio, ArgoUML, MagicDraw, StarUML …Các công cụ có cách sử dụng khá giống nhau và ký hiệu của các bạn vẽ trên UML cũng đã thống nhất nên việc nắm bắt một công cụ khi chuyển sang làm việc với một công cụ khá không quá khó khăn.Và công cụ phổ biến hiện nay là StarUML.



HÌNH 11 : Giao diện chính của StarUML

1. Cách tạo các Diagram

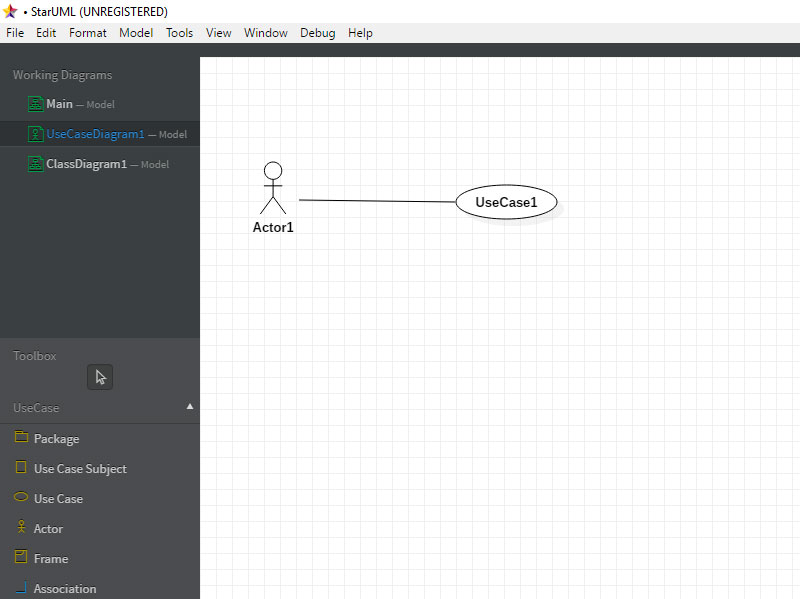
* Để tạo bản vẽ mới, chúng ta chuột phải vào model muốn tạo bản vẽ rồi chọn Add Diagram, trong menu đổ ra chọn tiếp bản vẽ mong muốn để tạo.

[](https://trinhtuantai.com/wp-content/uploads/2019/03/cach-tao-diagram-trong-staruml.jpg)

HÌNH 12 : Cách tạo Diagram

1. Cách tạo ra 1 bản vẽ:

* Sau khi chọn bản vẽ, cửa sổ bên trái sẽ hiển thị thanh công cụ chứa các ký hiệu tương ứng của bản vẽ để bạn có thể vẽ được các bản vẽ một các dễ dàng

[](https://trinhtuantai.com/wp-content/uploads/2019/03/cach-tao-ban-ve-trong-staruml.jpg)

HÌNH 13 : Cách tạo bản vẻ

.

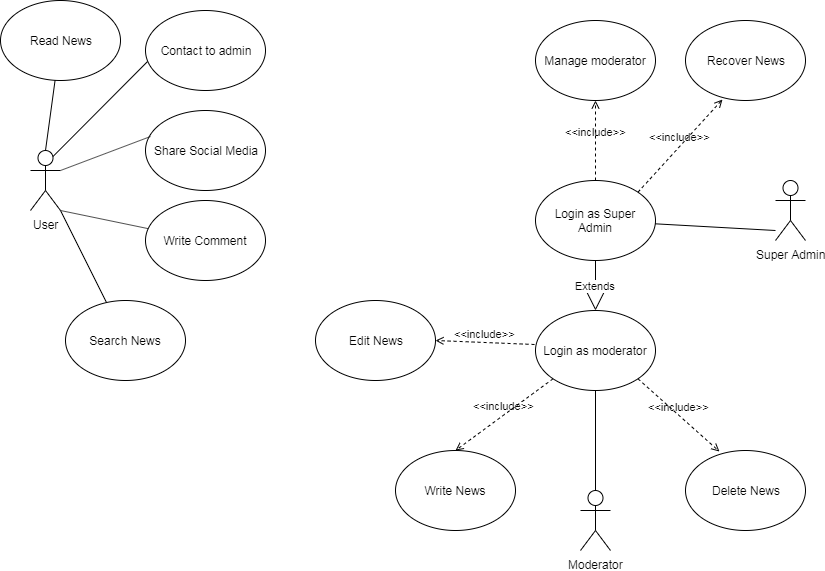
# **CHƯƠNG 2 : TÌM HIỂU USECASE**

1. **Use Case là gì ?**

Là chức năng nhỏ nhất của ứng dụng hoặc là nghiệp vụ của một hệ thống nào đó

Được sử dụng bới 1 actor hoặc một nhóm actor

Mô tả hoạt động của usecase thì người ta thường dùng Workflow hoặc UML



HÌNH 14 : Usecase

### 2.1 Actor

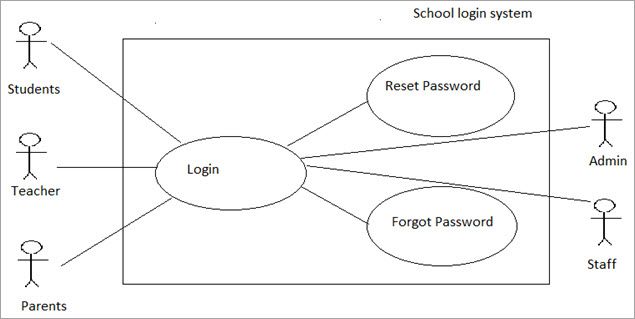
Actor được dùng để chỉ người sử dụng hoặc một đối tượng nào đó bên ngoài tương tác với hệ thống

### 2.2 Use Case

Là chức năng nhỏ nhất của ứng dụng hoặc một hệ thống nào đó và được sử dụng bởi 1 actor hoặc một nhóm actor.

Các use case được biểu diễn bởi các hình elip

Tên các use case thể hiện một chức năng xác định của hệ thống.

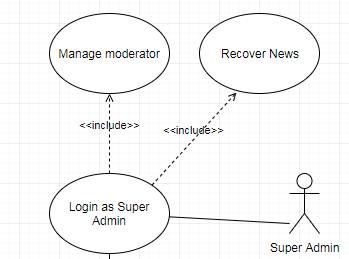


HÌNH 15 : Actor

## 2.3 Các quan hệ trong user case

### 2.3.1 Quan hệ gộp (Includes)

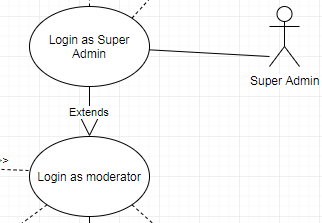
Một Use Case (gọi là base Use Case) có thể chứa («include») chức năng của một Use Case khác (gọi là inclusion Use Case) như một phần xử lý của nó. Nói chung, nó giả sử rằng mọi Use Case «include» sẽ được gọi mỗi khi tuyến Use Case chính chạy. Quan hệ này còn gọi là quan hệ sử dụng «uses».



HÌNH 16 : Includes

### 2.3.2 Quan hệ mở rộng (Extends):

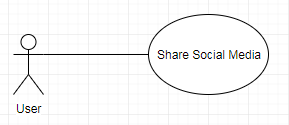
Một Use Case mở rộng (gọi là extension Use Case) có thể được mở rộng («extend») hành vi từ một Use Case khác (gọi là base Use Case); điều này thường dùng cho các trường hợp tùy chọn, ngoại lệ, chèn thêm vào …



HÌNH 17 : Extends

### 2.3.3 Quan hệ kết hợp (Association):

Tác nhân con kế thừa tính chất và hành vi của tác nhân cha

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

HÌNH 18 : Association

2.3.4 Quan hệ Generalization

Quan hệ tổng quát hóa (Generalization): Được sử dụng để thể hiện quan hệ thừa kế giữa các Actor hoặc giữa các Use Case với nhau.



HÌNH 19 : Generalization

**Bảng phân công công việc (Sprint)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Người phụ trách** | **Mô tả nội dung công việc** | **Bắt đầu** | **Kết thúc** | **Kết quả** |
| 1. | Nguyễn Hải Phúc Khang | Sơ đồ Use Case | 1/10 | 2/10 | Hoàn thành |
| 2. | Bùi Hửu Thắng | Giới thiệu công cụ StarUML  Các công cụ của UML | 1/10 | 2/10 | Hoàn thành |
| 3. | Trần Minh Thuận | UML là gì? | 1/10 | 2/10 | Hoàn thành |
| 4 | Nguyễn Minh Trinh | Cách tạo các Diagram  Cách tạo ra 1 bản vẽ | 1/10 | 2/10 | Hoàn thành |
| 5 | Đoàn Vũ Quốc Khiêm | Mục Tiêu Của Uml  Một số dạng biểu đồ UML phổ biến | 1/10 | 2/10 | Hoàn thành |