**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**CÔNG NGHỆ PHẦN MỀM**

***Đề tài:***

**Nhóm sinh viên thực hiện: Nhóm …**

**Thành viên nhóm:** …

**Giảng viên hướng dẫn*:***

***Hà Nội, tháng 10 năm 2023***

NHIỆM VỤ 3/4: SPRINT 1/2

# **I. Tài liệu đặc tả yêu cầu phần mềm**

## 1.1 Biểu đồ Phân rã chức năng (FDD)



## 1.2 Biểu đồ Luồng dữ liệu (DFD)

### 1.2.1 DFD Đăng ký



### 1.2.2 DFD Đăng nhập

**

### 1.2.3 DFD Đổi mật khẩu

…

### 1.2.4 DFD Tạo lớp

…

### 1.2.5 DFD Tham gia lớp

…

### 1.2.6 DFD Thêm bài đăng

…

### 1.2.7 DFD Sửa bài đăng

…

### 1.2.8 DFD Xóa bài đăng

…

### 1.2.9 DFD Xem bài đăng

…

### 1.2.10 DFD Bình luận

…

### 1.2.11 DFD Xóa bình luận

…

## 1.3 Sơ đồ Use case tổng quát



## 1.4 Miêu tả chi tiết các Use cases

### 1.4.1 Use case Đăng ký

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 01 | | |
| Use Case Name | Đăng ký | | |
| Created By: | … | Last Updated By | … |
| Date Created: | … | Date Last Updated | … |

|  |  |
| --- | --- |
| Actor | Khách vãng lai |
| *Description* | Chức năng đăng ký cho phép người dùng đăng ký để tham gia hoặc tạo các lớp |
| *Trigger* | Chức năng này được sử dụng khi khách vãng lai click vào nút “Sign up” |
| *Pre-conditions* | Khách vãng lai truy cập vào website |
| *Post conditions* | Khách vãng lai đăng ký thành công |
| *Flow of Events* | 1. Khách vãng lai truy đăng nhập vào website  2. Khách vãng lai chọn “Sign up”  3. Hệ thống hiển thị giao diện nhập tên, email, mật khẩu  4. Khách vãng lai nhập tên, email, mật khẩu và chọn “Sign up”  5. Hệ thống hiển thị thông báo đăng ký thành công và yêu cầu khách vãng lai xác thực tài khoản  6. Khách vãng lai click vào link xác thực được gửi trong mail  7. Hệ thống hiển thị giao diện xác thực tài khoản thành công |

### 1.4.2 Use case Đăng nhập

…

### 1.4.3 Use case Tạo lớp

…

### 1.4.4 Use case Tham gia lớp

…

### 1.4.5 Use case Bình luận

…

### 1.4.6 Use case Xóa bình luận

…

### 1.4.7 Use case Xem bài đăng

…

### 1.4.8 Use case Tạo bài đăng

…

### 1.4.9 Use case Sửa bài đăng

…

### 1.4.10 Use case Xóa bài đăng

…

## 1.5 Sơ đồ Trình tự (SD)

### 1.5.1 SD Đăng Nhập

…

1.5.2 SD Đăng ký

…

### 1.5.3 SD Đổi mật khẩu



1.5.4 SD Tạo bài đăng

…

1.5.5 SD Xóa bài đăng

…

### 1.5.6 SD Sửa bài đăng

…

### 1.5.7 SD Xem bài đăng

…

### 1.5.8 SD Bình luận

…

### 1.5.9 SD Xóa bình luận

…

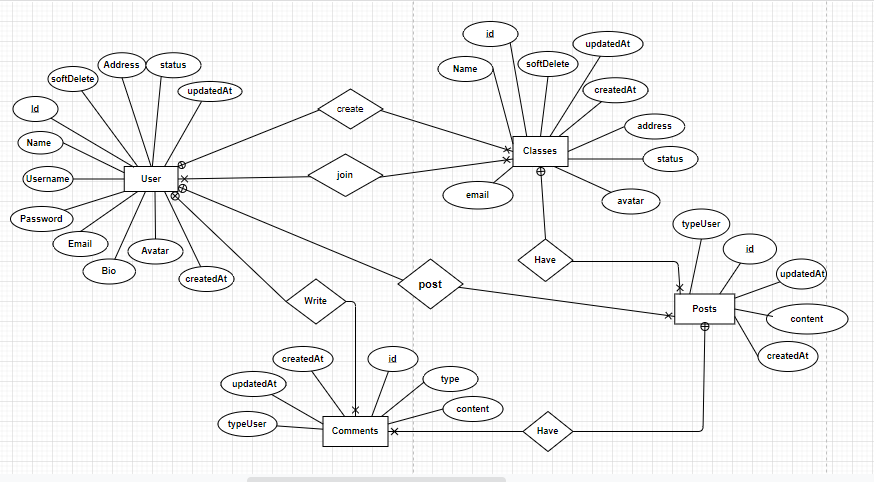
### 1.5.10 SD Tạo lớp tham gia

…

### 1.5.11 SD Tham gia lớp

…

## 1.6 Biểu đồ quan hệ thực thể (ERD)

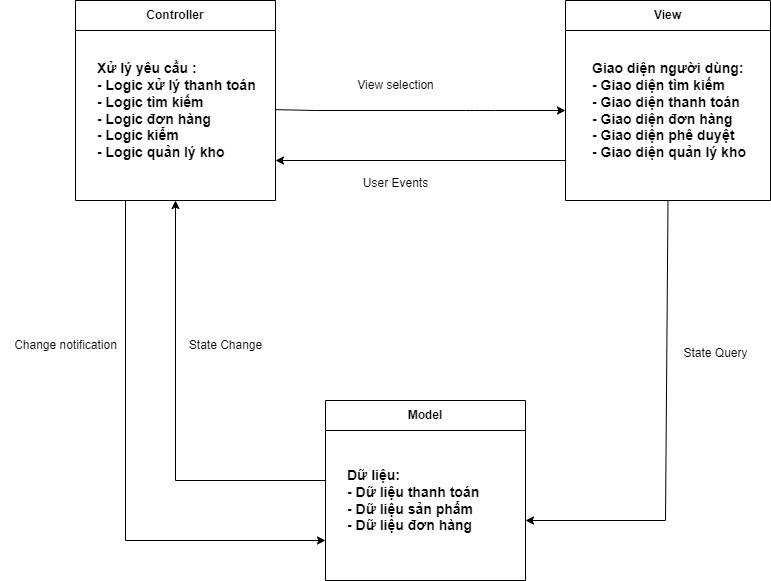


# **II. Thiết kế phần mềm**

## 1. **Thiết kế Kiến trúc**

<< HÌNH VẼ MINH HOẠ VÀ GIẢI THÍCH CÁC THÀNH PHẦN TRONG KIẾN TRÚC CỦA PHẦN MỀM >>

**MÔ HÌNH MVC**



Trong mô hình này:

* Model chứa dữ liệu liên quan đến thanh toán (Payment Data), sản phẩm (Product Data), và đơn hàng (Order Data).
* View đảm nhận vai trò hiển thị giao diện người dùng (Search UI, Payment UI, Order UI, Approval UI, Inventory UI) để người dùng tương tác với hệ thống và xem thông tin.
* Controller là thành phần điều khiển luồng của ứng dụng. Nó chứa logic xử lý cho thanh toán (Payment Logic), tìm kiếm (Search Logic), đơn hàng (Order Logic), kiểm duyệt (Approval Logic) và quản lý kho (Inventory Logic).

**CÁCH THỨC HOẠT ĐỘNG MÔ HÌNH MVC TRONG CÁC CHỨC NĂNG DỰ ÁN NHÓM:**

**1.Chức năng đa dạng hình thức thanh toán trực tuyến:**

Model: Chứa dữ liệu liên quan đến các hình thức thanh toán, ví dụ: thông tin tài khoản, API kết nối với các cổng thanh toán (ZaloPay, Momo, v.v.).

View: Hiển thị thông tin và giao diện cho người dùng để lựa chọn hình thức   
 thanh toán.

Controller: Xử lý các sự kiện từ người dùng, gửi yêu cầu thanh toán đến Model  
 và cập nhật View sau khi thanh toán thành công hoặc thất bại.

2. **Chức năng tìm kiếm sản phẩm:**

Model: Chứa dữ liệu sản phẩm và logic tìm kiếm, bao gồm các tiêu chí tìm   
 kiếm như tên, danh mục, từ khóa, v.v.

View: Hiển thị giao diện cho người dùng nhập các tiêu chí tìm kiếm và hiển thị   
 kết quả tìm kiếm.

Controller: Xử lý yêu cầu tìm kiếm từ View, truy xuất dữ liệu từ Model và cập   
 nhật View với kết quả tìm kiếm.

3. **Chức năng xem lại thông tin đơn hàng:**

Model: Lưu trữ dữ liệu đơn hàng và trạng thái giao hàng, cũng như tích hợp với bên giao hàng để cập nhật thông tin.

View: Hiển thị thông tin đơn hàng và trạng thái giao hàng cho người dùng.

Controller: Xử lý yêu cầu xem lại đơn hàng từ View, truy xuất dữ liệu từ Model   
 và cập nhật View với thông tin đơn hàng và trạng thái giao hàng.

4.**Chức năng kiểm duyệt và phê duyệt bài viết:**

Model: Lưu trữ dữ liệu bài viết và trạng thái kiểm duyệt.

View: Hiển thị danh sách bài viết chờ kiểm duyệt và giao diện cho quản trị viên   
 để xem và phê duyệt bài viết.

Controller: Xử lý yêu cầu kiểm duyệt và phê duyệt bài viết từ View, cập nhật   
 trạng thái bài viết trong Model và cập nhật View sau khi phê duyệt.

5. **Chức năng kiểm tra số lượng và chất lượng sản phẩm và dự đoán tồn kho:**

Model: Lưu trữ dữ liệu về sản phẩm, số lượng hiện có và chất lượng. Áp dụng các thuật toán dự đoán tồn kho.

View: Hiển thị thông tin số lượng và chất lượng sản phẩm, cũng như kết quả dự đoán tồn kho cho quản lý kho.

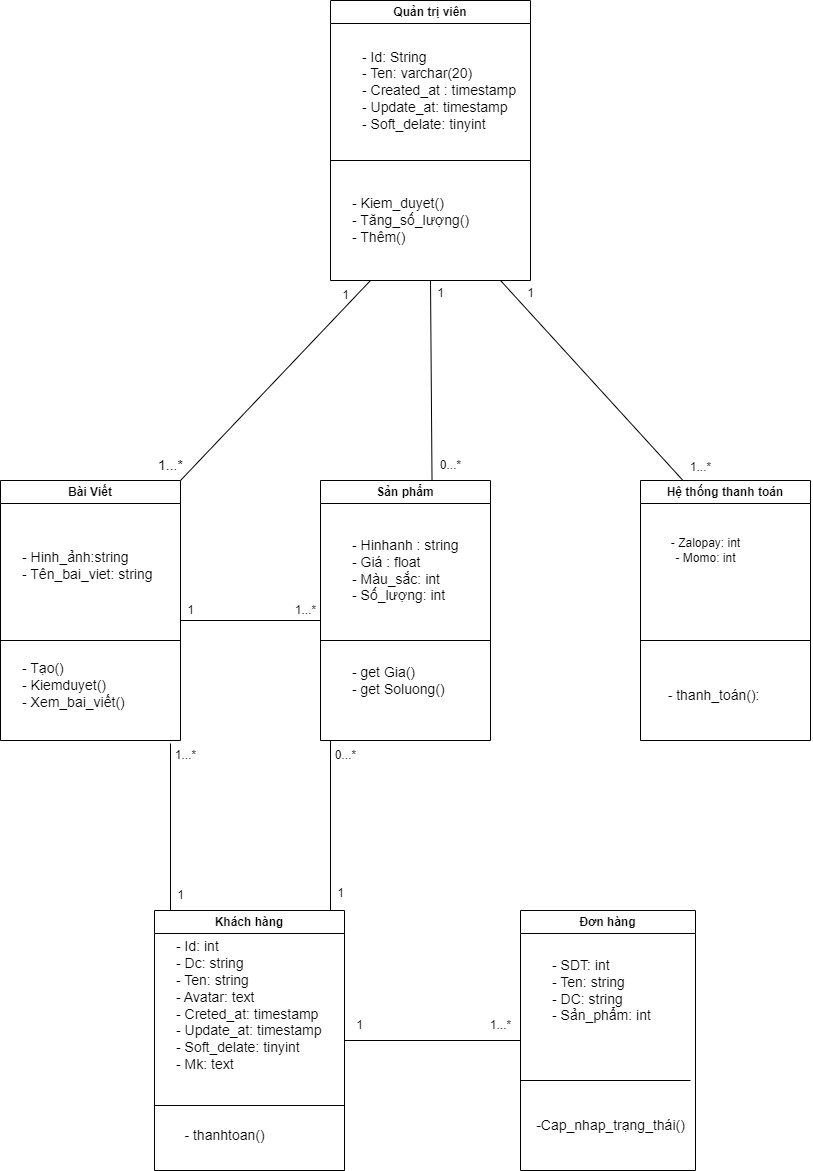
Controller: Xử lý yêu cầu kiểm tra số lượng và chất lượng từ View, truy xuất dữ liệu từ Model và cập nhật View với thông tin cần thiết.

## 2. **Thiết kế Hướng đối tượng**

<< NẾU PHẦN MỀM ĐƯỢC PHÁT TRIỂN THEO HƯỚNG ĐỐI TƯỢNG THÌ VẼ HÌNH MINH HOẠ VÀ GIẢI THÍCH CÁC LỚP, CŨNG NHƯ MỐI LIÊN HỆ GIỮA CÁC LỚP >>

VÍ DỤ





Trong mô hình trên:

* Lớp quản trị viên, có chức năng xem BàiViết() để xem các bài viết của người dùng liên quan đến cây cảnh trên diễn đàn. Trong lớp quản trị viên có QuảnLýKho đại diện cho chức năng quản lý kho.
* Lớp BàiViết chứa thông tin về các bài viết.
* Lớp SảnPhẩm đại diện cho các sản phẩm cây cảnh
* Lớp HệThốngThanhToán đại diện cho hệ thống thanh toán trực tuyến.
* Lớp KháchHàng chứa thông tin về khách hàng, bao gồm id, tên, địaChỉ và email.
* Lớp ĐơnHàng đại diện cho thông tin đơn hangf.
* Lớp QuảnLýKho đại diện cho chức năng quản lý kho, có các phương thức kiểmTraSốLượng() để kiểm tra số lượng sản phẩm trong kho, kiểmTraChấtLượng() để kiểm tra chất lượng sản phẩm và dự ĐoánTồnKho() để dự đoán tình trạng tồn kho và ra quyết định nhập kho.

## 3. **Thiết kế Dữ liệu**

<<DỰA VÀO BIỂU ĐỒ QUAN HỆ THỰC THỂ Ở TRÊN>>

### 3.1 Các thực thể và thuộc tính

* **users**(id, username, password, email, bio, avatar, status, type, age, sex, birthday, created\_at, updated\_at, soft\_delete)
* **posts**(id, status, content, created\_at, updated\_at, soft\_delete)
* **classes**(id, name, cover, status, referral\_code, created\_at, updated\_at, soft\_delete)
* **comments**(id, type\_comment, content, created\_at, updated\_at, soft\_delete)

### 3.2 Mối liên kết

* **users** và **posts** là liên kết 1 - n
* **users** và **classes** là liên kết m - n
* **users** và **comments** là liên kết 1 - n
* **posts** và **comments** là liên kết 1 - n
* **classes** và **posts** là liên kết 1 - n

### 3.3 Chuyển sang quan hệ

* Liên kết 1 - N → Khóa chính của quan hệ bên một trở thành khóa ngoại của quan hệ bên nhiều.
* Liên kết M - N → Tạo bảng quan hệ mới, chuyển khóa chính của hai quan hệ phía M và N thành khóa ngoại của quan hệ mới. Khóa chính của quan hệ mới là sự kết hợp của hai khóa ngoại.

Kết quả:

* **users**(id, username, password, email, bio, avatar, status, type, age, sex, birthday, created\_at, update\_at, soft\_delete)
* **posts**(id, status, content, created\_at, updated\_at, soft\_delete, owner\_id, class\_id)
* **classes**(id, name, cover, status, referral\_code, created\_at, updated\_at, soft\_delete)
* **comments**(id, type\_comment, content, created\_at, updated\_at, soft\_delete, owner\_id, topic\_id)
* **student\_classes**(id, created\_at, updated\_at, soft\_delete, status, user\_id, class\_id)

### 3.4 Chuẩn hóa dữ liệu

<<CÓ THỂ BỎ QUA NẾU CHƯA HỌC Ở CÁC MÔN HỌC TRƯỚC>>

#### 3.4.1 Xác định phụ thuộc hàm

* id xác định một người dùng duy nhất bao gồm tên người dùng, mật khẩu, email, lý lịch, ảnh đại diện, trạng thái, địa chỉ, ngày tạo, ngày cập nhật, xóa mềm.
* id xác định một bài đăng duy nhất bao gồm các thông tin về bài đăng : nội dung, kiểu người dùng, thời gian đăng bài, thời gian chỉnh sửa, xóa mềm.
* id xác định một lớp học duy nhất gồm thông tin lớp học: tên lớp, email, hình đại diện, trạng thái lớp học, địa chỉ, thời gian tạo, thời gian chỉnh sửa, xóa mềm.
* id xác định một bình luận hoa duy nhất gồm các thông tin liên quan đến bình luận đó: loại bình luận, nội dung, loại người dùng, thời gian bình luận, thời gian chỉnh sửa, xóa mềm, mã người dùng và mã bài đăng.

#### 3.4.2 Xét các dạng chuẩn

* **users**(id, username, password, email, bio, avatar, status, type, age, sex, birthday, created\_at, updated\_at, soft\_delete)

Đặt id = A, username = B, password = C, email = D,bio = E, avatar = F, status = G, type = H, age = I, created\_at = K, updated)at = N, soft\_delete = M, sex = O, birthday = Q

Phụ thuộc hàm:

F = {A → B, A → C, A → D, A → E, A → F, A → G, A → H, A → I, A → K, A → N, A → M, A → O, A → Q}

=> Đạt 1NF vì không có thuộc tính đa trị

=> Đạt 2NF vì các thuộc tính không khóa phụ thuộc đầy đủ vào khóa chính A

=> Đạt 3NF vì các thuộc tính không khóa phụ thuộc trực tiếp vào khóa chính A

=> Đạt BCNF vì không có thuộc tính khóa nào phụ thuộc vào thuộc tính không khóa

* **posts**(id, status, content, created\_at, updated\_at, soft\_delete, owner\_id, class\_id)

…

* **classes**(id, name, cover, status, referral\_code, created\_at, updated\_at, soft\_delete)

…

* **comments**(id, type\_comment, content, created\_at, updated\_at, soft\_delete, owner\_id, topic\_id )

…

* student\_classes(id, created\_at, updated\_at, soft\_delete, status, user\_id, class\_id )

…

### 3.5 Sơ đồ quan hệ

<<HÌNH VẼ SƠ ĐỒ QUAN HỆ >>

## 4. **Thiết kế Giao diện Người dùng**

HÌNH VẼ GIAO DIỆN CỦA PHẦN MỀM. CÓ THỂ SỬ DỤNG MỘT SỐ NỀN TẢNG CHO PHÉP THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG NHƯ: [NinjaMock](https://ninjamock.com/), [Figma](https://www.figma.com/), [MockingBot](https://mockingbot.in/), [MarvelApp](https://marvelapp.com/examples), [Sketch](https://www.sketch.com/), Adobe Photoshop, …

**NGHIÊM CẤM CHỤP MÀN HÌNH CHƯƠNG TRÌNH CÓ SẴN**

# **III. Mã nguồn**

<< LINK MÃ NGUỒN TRÊN GITHUB, NẾU CÓ>>

# **IV. Tài liệu kiểm thử**

* Tham khảo báo cáo [kiểm thử giao diện](https://docs.google.com/spreadsheets/d/1FCA8NTF_SsqQ2uHMDUXzuczcVpzEwfFQPNqQ6j1Ih9Y/edit?usp=sharing)
* Tham khảo báo cáo [kiểm thử chức năng](https://docs.google.com/spreadsheets/d/1JM-fTYkRxvUmpwPzZ_QcdknETALVys91ZCZwdBbAj6U/edit?usp=sharing)