**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**CÔNG NGHỆ PHẦN MỀM**

***Đề tài: TravelNest***

**Nhóm sinh viên thực hiện: Nhóm 3 - 65HTTT**

**Thành viên nhóm:** Trần Văn Tài – MSV: 2351160548

Thiều Quang Gia Bảo – MSV: 2351160507

Phạm Văn Tài – MSV: 2351160547

Nguyễn Tuấn Kiệt – MSV: 2351160533

Trịnh Khắc Hưng – MSV: 2351160526

Phan Văn Tâm – MSV: 2351160549

Trần Mai Ngọc Anh – MSV: 2351160504

Trần Thị Minh Thư – MSV: 2351160556

Hoàng Phương Thảo – MSV: 2351160553

**Giảng viên hướng dẫn*:***

***Hà Nội, năm 2024***

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

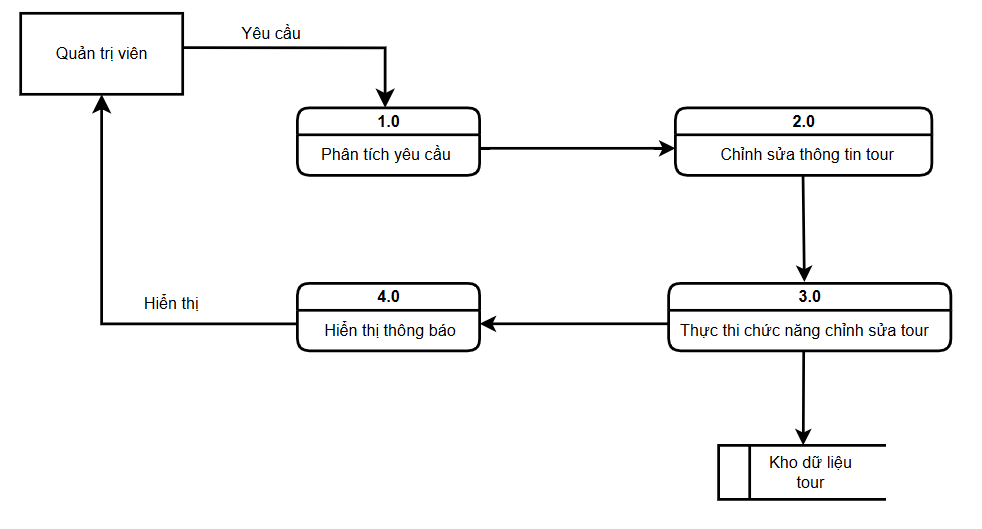
### 1.2.2 DFD Đăng nhập

### 

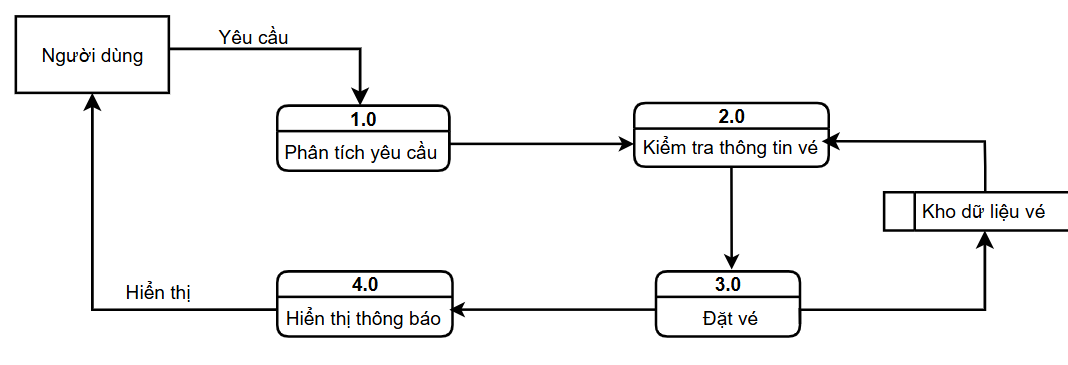
### 1.2.3 DFD Tìm kiếm vé

### 1.2.4 DFD Xem chi tiết thông tin tour

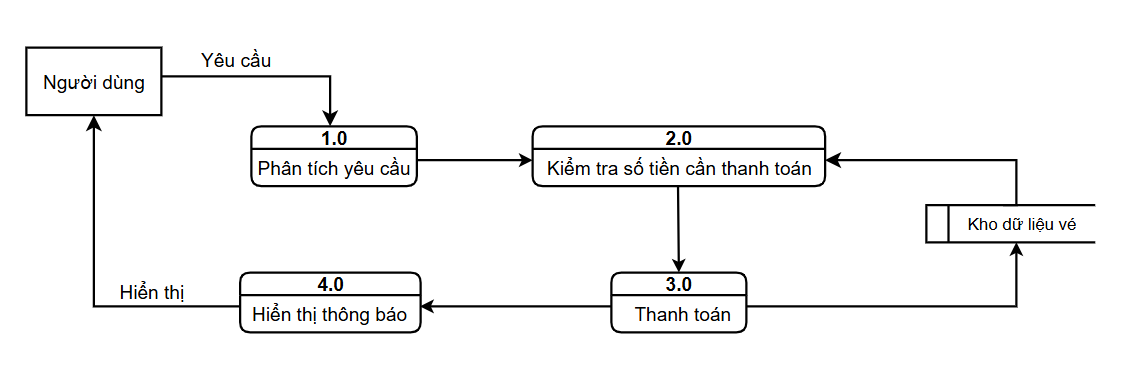
### 1.2.5 DFD Chỉnh sửa thông tin tour



### 1.2.6 DFD Đặt vé



### 1.2.7 DFD Thanh toán



### 1.2.8 DFD

…

### 1.2.9 DFD

…

### 1.2.10 DFD

…

### 1.2.11 DFD

…

## 1.3 Sơ đồ Use case tổng quát



### 1.4.5 Use case Chỉnh sửa thông tin tour

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 05 | | |
| Use Case Name | Chỉnh sửa | | |
| Created By: | Trần Mai Ngọc Anh | Last Updated By | Trần Mai Ngọc Anh |
| Date Created: | 4/12/2024 | Date Last Updated | 5/12/2024 |

|  |  |
| --- | --- |
| Actor | Quản trị viên |
| *Description* | Chức năng chỉnh sửa thông tin về tour du lịch cho phép quản trị viên chỉnh sửa những thông tin về tour để người dùng nắm bắt được những thông tin sát với thực tế nhất của tour |
| *Trigger* | Chức năng này được sử dụng khi quản trị viên click vào nút “Chỉnh sửa”. |
| *Pre-conditions* | Quản trị viên đã đăng nhập thành công vào hệ thống. |
| *Post conditions* | Quản trị viên chỉnh sửa thông tin thành công |
| *Flow of Events* | 1. Quản trị viên truy cập vào website  2. Quản trị viên click vào “Chỉnh sửa”  3. Hệ thống hiển thị ra các tour  3. Quản trị viên nhấp chọn tour cần chỉnh sửa  4. Quản trị viên chỉnh sửa thông tin tour  4. Hệ thống sẽ hiển thị thông báo chỉnh sửa thành công |

### 1.4.6 Use case Đặt vé

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 06 | | |
| Use Case Name | Đặt vé | | |
| Created By: | Trần Mai Ngọc Anh | Last Updated By | Trần Mai Ngọc Anh |
| Date Created: | 4/12/2024 | Date Last Updated | 5/12/2024 |

|  |  |
| --- | --- |
| Actor | Người dùng |
| *Description* | Chức năng đặt vé cho phép người dùng chọn lựa và đặt vé phù hợp để đặt chỗ trên phương tiện di chuyển |
| *Trigger* | Chức năng này được sử dụng khi người dùng click vào nút “Đặt vé”. |
| *Pre-conditions* | - Người dùng đã đăng nhập thành công vào hệ thống.  - Người dùng đã tìm kiếm được vé phù hợp |
| *Post conditions* | Người dùng đặt vé thành công |
| *Flow of Events* | 1. Người dùng truy cập vào website  2. Người dùng tìm kiếm vé  3. Người dùng click vào “Đặt vé” sau khi chọn được vé phù hợp  4. Hệ thống sẽ hiển thị thông báo đặt vé thành công |

### 1.4.7 Use case Thanh toán

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 07 | | |
| Use Case Name | Thanh toán | | |
| Created By: | Trần Mai Ngọc Anh | Last Updated By | Trần Mai Ngọc Anh |
| Date Created: | 4/12/2024 | Date Last Updated | 5/12/2024 |

|  |  |
| --- | --- |
| Actor | Người dùng |
| *Description* | Chức năng thanh toán cho phép người dùng thanh toán chi phí của vé đã đặt để giao dịch được thực hiện một cách nhanh chóng và thuận tiện |
| *Trigger* | Chức năng này được sử dụng khi người dùng click vào nút “Thanh toán”. |
| *Pre-conditions* | - Người dùng đã đăng nhập thành công vào hệ thống.  - Người dùng đã đặt vé thành công |
| *Post conditions* | Người dùng thanh toán thành công |
| *Flow of Events* | 1. Người dùng truy cập vào website  2. Người dùng tìm kiếm vé  3. Người dùng đặt được vé phù hợp  4. Người dùng click vào “Thanh toán”  5. Hệ thống hiển thị số tiền cần thanh toán  6. Người dùng chọn một trong các hình thức thanh toán online và nhập số tiền phải thanh toán sau đó chọn “Đồng ý”  7. Hệ thống thông báo thanh toán thành công |

### 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

…

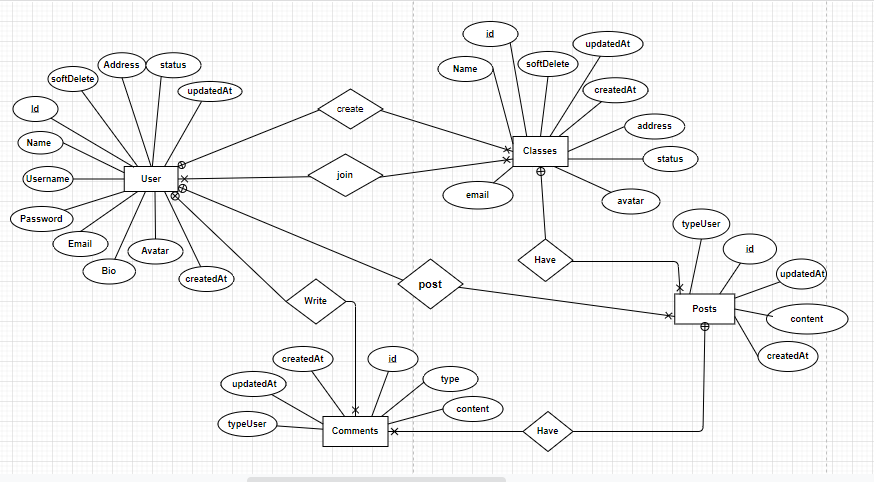
### 1.5.10 SD Tạo lớp tham gia

…

### 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 >>

## 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



## 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)