**RƯỜNG 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

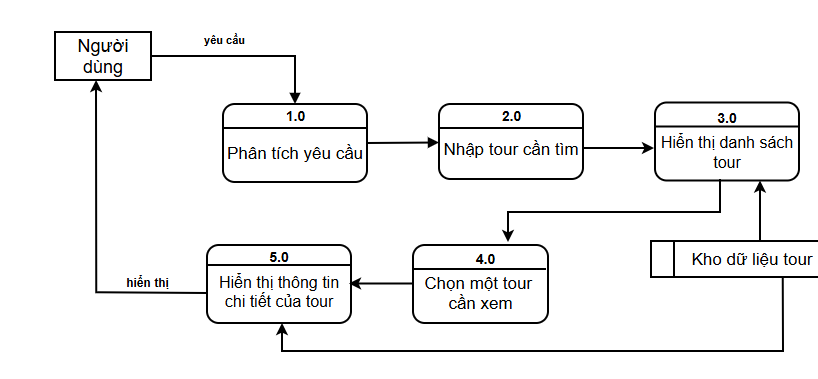


### 

### 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

A diagram of a flowchart

Description automatically generated

### 1.2.6 DFD Đặt vé

A diagram of a flowchart

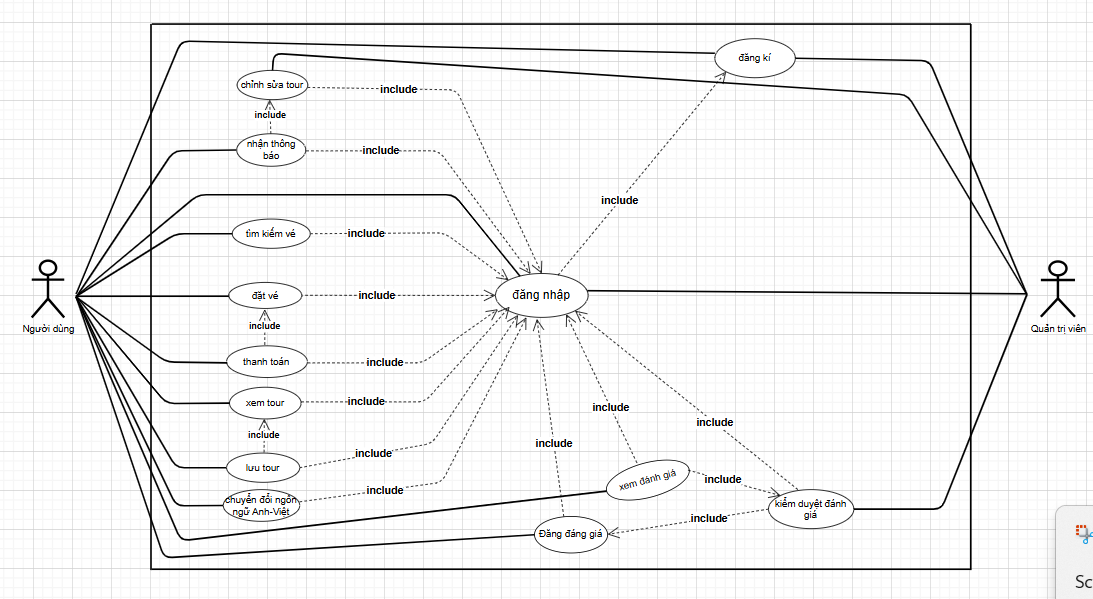
Description automatically generated

### 1.2.7 DFD Thanh toán

A diagram of a flowchart

Description automatically generated

## 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: | Trần Thị Minh Thư | Last Updated By | Trần Thị Minh Thư |
| Date Created: | 04/12/2024 | Date Last Updated | 05/12/2024 |

|  |  |
| --- | --- |
| Actor | Người dùng, quản trị viên |
| *Description* | Chức năng đăng ký cho phép người dùng hoặc quản trị viên đăng ký để tạo được tài khoản đăng nhập vào hệ thống. |
| *Trigger* | Chức năng này được sử dụng khi người dùng hoặc quản trị viên click vào nút “Sign up”. |
| *Pre-conditions* | Người dùng hoặc quản trị viên truy cập vào website |
| *Post conditions* | Người dùng hoặc quản trị viên đăng ký thành công |
| *Flow of Events* | 1. Người dùng hoặc quản trị viên truy cập vào website  2. Người dùng hoặc quản trị viên chọn “Sign up”  3. Hệ thống hiển thị giao diện nhập tên, email, mật khẩu  4. Người dùng hoặc quản trị nhập tên, email, mật khẩu và chọn “Sign up”  5. Hệ thống hiển thị thông báo đăng ký thành công và yêu cầu người dùng hoặc quản trị viên xác thực tài khoản  6. Người dùng hoặc quản trị viên 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

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 02 | | |
| Use Case Name | Đăng nhập | | |
| Created By: | Trần Thị Minh Thư | Last Updated By | Trần Thị Minh Thư |
| Date Created: | 04/12/2024 | Date Last Updated | 05/12/2024 |

|  |  |
| --- | --- |
| Actor | Người dùng, quản trị viên |
| *Description* | Chức năng đăng nhập cho phép người dùng hoặc quản trị viên đăng nhập để trải nghiệm đầy đủ các dịch vụ của website travelnest |
| *Trigger* | Chức năng này được sử dụng khi người dùng hoặc quản trị viên click vào nút “Log in” |
| *Pre-conditions* | Người dùng hoặc quản trị viên đã đăng kí tài khoản thành công |
| *Post conditions* | Người dùng hoặc quản trị viên đăng nhập thành công |
| *Flow of Events* | 1. Người dùng hoặc quản trị viên truy cập vào website  2. Người dùng hoặc quản trị viên chọn “Log in”  3. Hệ thống hiển thị giao diện nhập tên đăng nhập và mật khẩu  4. Người dùng hoặc quản trị viên nhập tên đăng nhập và mật khẩu và chọn “Log in”  5. Hệ thống xác minh tên đăng nhập và mật khẩu với cơ sở dữ liệu  6. Nếu người dùng hoặc quản trị viên đăng nhập thành công thì hệ thống sẽ chuyển đến trang chủ của hệ thống. |
| *Alternative Flows* | Nếu người dùng hoặc quản trị viên quên mật khẩu:  - Người dùng hoặc quản trị viên truy cập vào trang đăng nhập.  - Người dùng hoặc quản trị viên nhấn vào nút "Forgot password".  -Trang web chuyển hướng người dùng hoặc quản trị viên đến trang khôi phục mật khẩu.  - Người dùng hoặc quản trị viên nhập địa chỉ email đã đăng ký và yêu cầu khôi phục mật khẩu. Hệ thống gửi một liên kết khôi phục mật khẩu đến địa chỉ email.  - Người dùng hoặc truy cập vào địa chỉ email và làm theo hướng dẫn để khôi phục mật khẩu. |

### 1.4.3 Use case Tìm kiếm vé

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 03 | | |
| Use Case Name | Tìm kiếm vé | | |
| Created By: | Trần Thị Minh Thư | Last Updated By | Trần Thị Minh Thư |
| Date Created: | 04/12/2024 | Date Last Updated | 05/12/2024 |

|  |  |
| --- | --- |
| Actor | Người dùng |
| *Description* | Chức năng tìm kiếm vé cho phép người dùng tìm kiếm theo địa điểm, ngày giờ, loại vé để chọn lựa và đặt vé phù hợp. |
| *Trigger* | Chức năng này được sử dụng khi người dùng nhấp vào ô “Tìm kiếm” |
| *Pre-conditions* | Người dùng đã đăng nhập vào hệ thống |
| *Post conditions* | Tìm kiếm được vé mong muốn |
| *Flow of Events* | 1. Người dùng truy cập vào trang web 2. Trên trang web người dùng nhập tiêu chí tìm kiếm vé trên thanh “Tìm kiếm” (có thể nhập địa điểm hoặc ngày hoặc ngân sách) 3. Người dùng ấn nút “Tìm kiếm” 4. Hệ thống sẽ hiển thị kết quả tìm kiếm |

### 1.4.4 Use case Xem chi tiết tour

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 04 | | |
| Use Case Name | Thông tin chi tiết | | |
| Created By: | Trần Thị Minh Thư | Last Updated By | Trần Thị Minh Thư |
| Date Created: | 04/12/2024 | Date Last Updated | 05/12/2024 |

|  |  |
| --- | --- |
| Actor | Người dùng |
| *Description* | Chức năng xem thông tin chi tiết cho phép tác nhân xem chi tiết 1 tour du lịch bao gồm lịch trình, giá cả, điều kiện đặt vé. |
| *Trigger* | Chức năng này được sử dụng khi tác nhân click vào nút “Xem chi tiết”. |
| *Pre-conditions* | Người dùng đã đăng nhập vào hệ thống |
| *Post conditions* | Người dùng xem được thông tin chi tiết về tour |
| *Flow of Events* | 1.Người dùng truy cập vào website  2. Trên trang web người dùng chọn “tour du lịch”  3. Hệ thống sẽ hiện thị danh sách các tour du lịch  4. Người dùng sẽ nhấp chọn một tour du lịch mà họ quan tâm và click vào “Xem chi tiết”  5. Hệ thống sẽ hiển thị các thông tin chi tiết về tour |

### 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 |

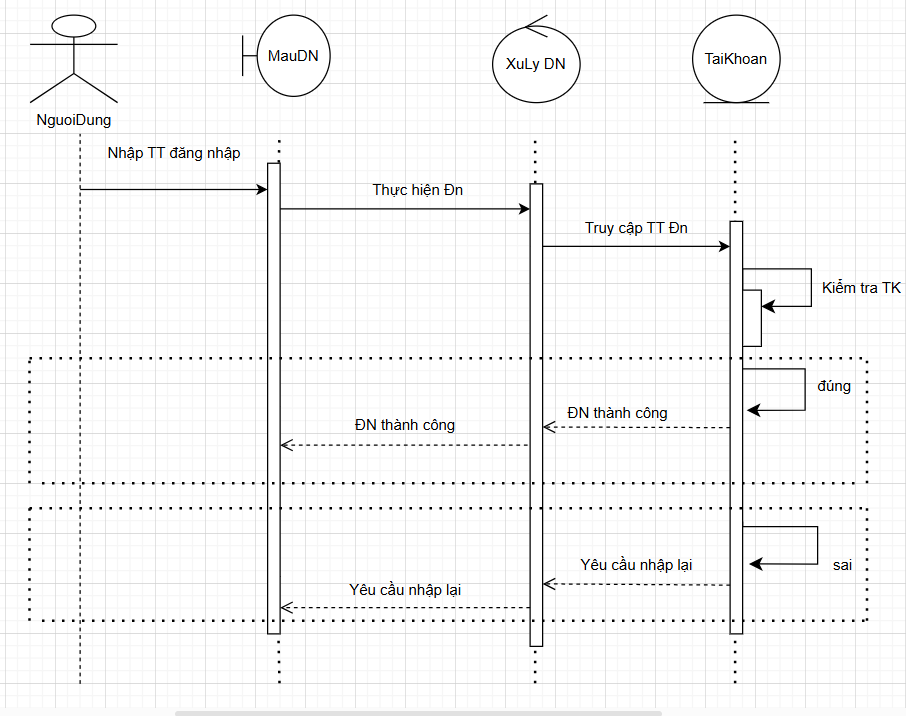
### 1.4.7 Use case Thanh toán

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

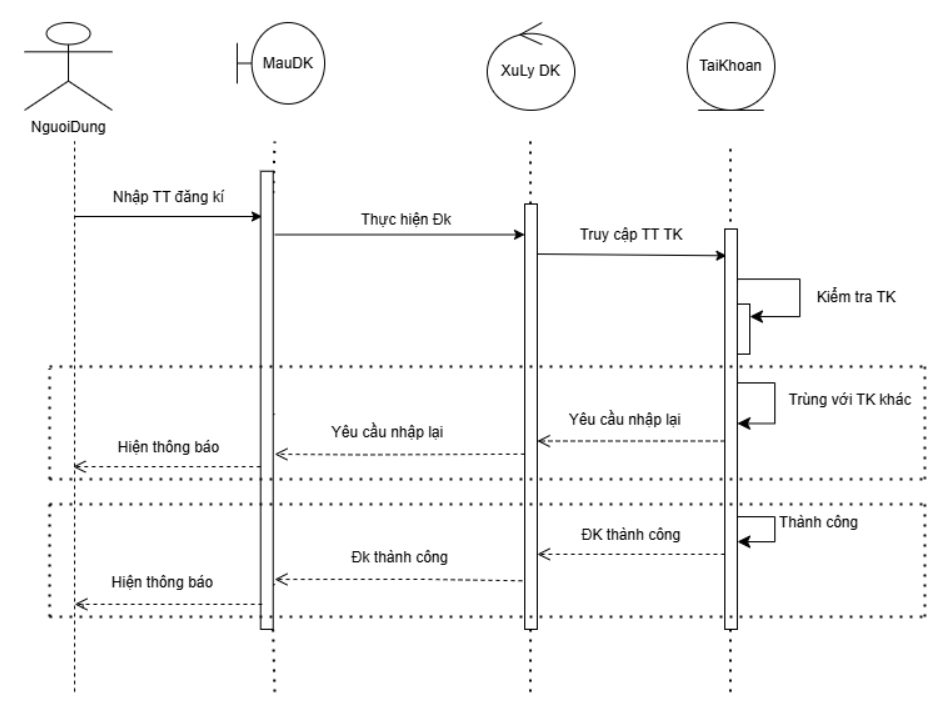
|  |  |
| --- | --- |
| Actor | Người dùng |
| *Description* | Chức năng thanh toán cho phép người dùng thanh toán chi phí của vé đã đặt để giao dịch được thực hiện một cách nhanh chóng và thuận tiện |
| *Trigger* | Chức năng này được sử dụng khi người dùng click vào nút “Thanh toán”. |
| *Pre-conditions* | - 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 |

## 1.5 Sơ đồ Trình tự (SD)

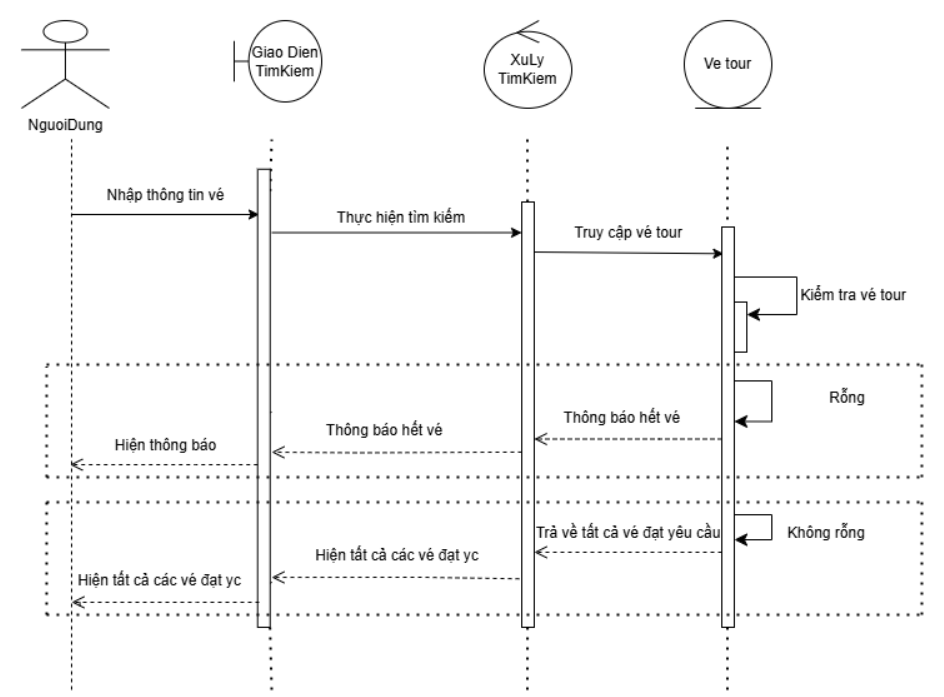
### 1.5.1 SD Đăng Nhập



1.5.2 SD Đăng ký



### 1.5.3 SD Tìm kiếm vé

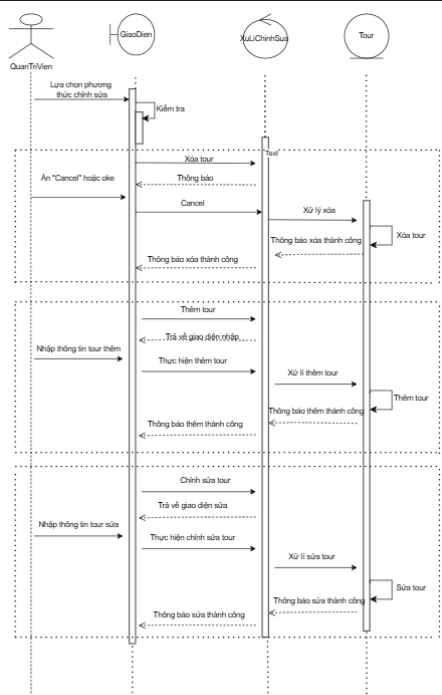


1.5.4 SD Xem chi tiết thông tin tour

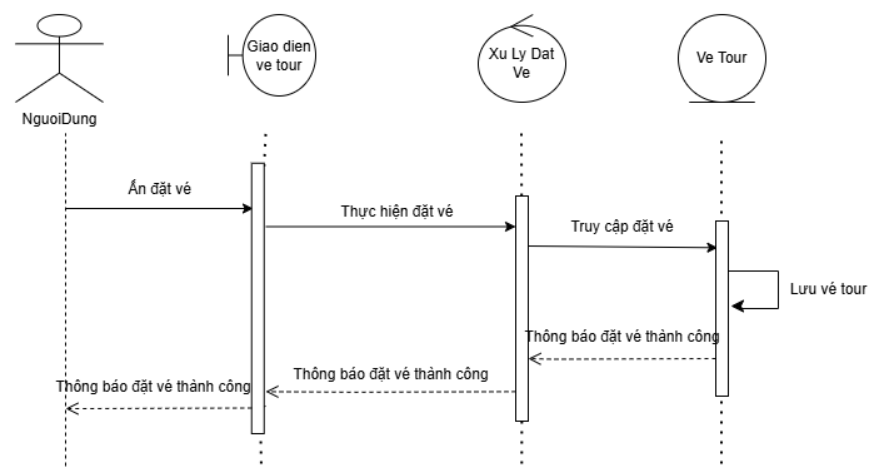
A diagram of a tour

Description automatically generated

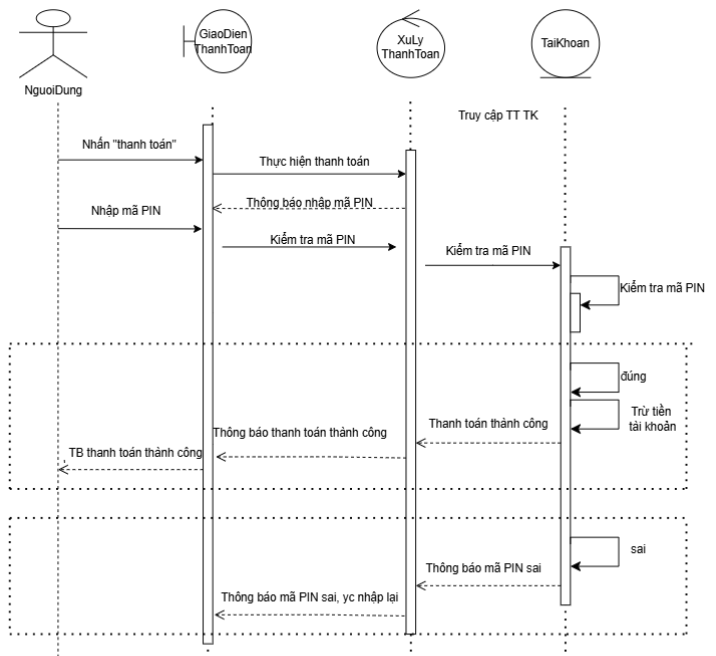
1.5.5 SD Chỉnh sửa tour



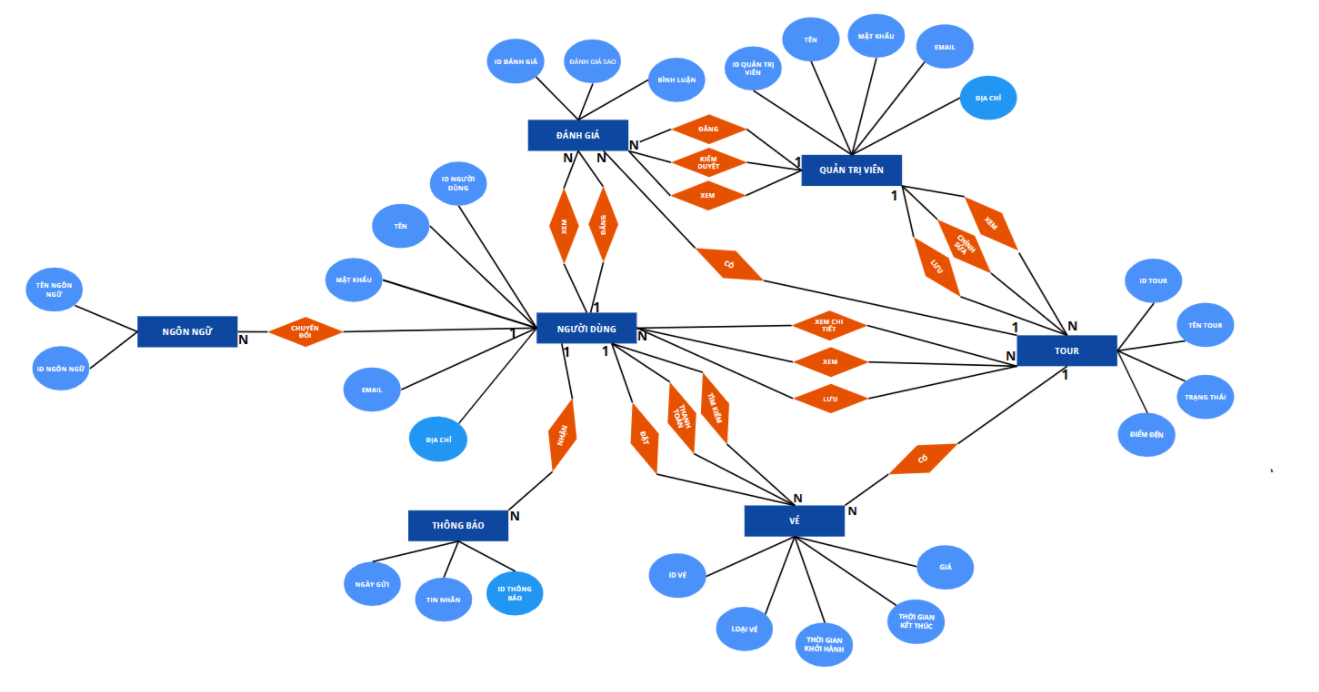
### 1.5.6 SD Đặt vé



### 1.5.7 SD Thanh Toán

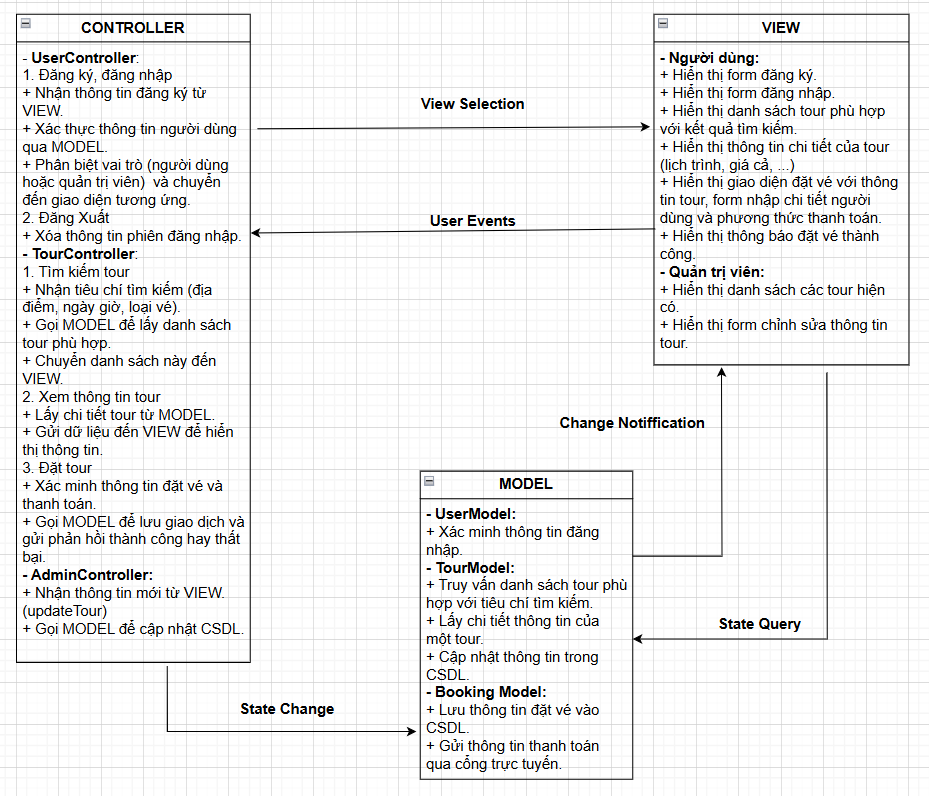


## 1.6 Biểu đồ quan hệ thực thể (ERD)

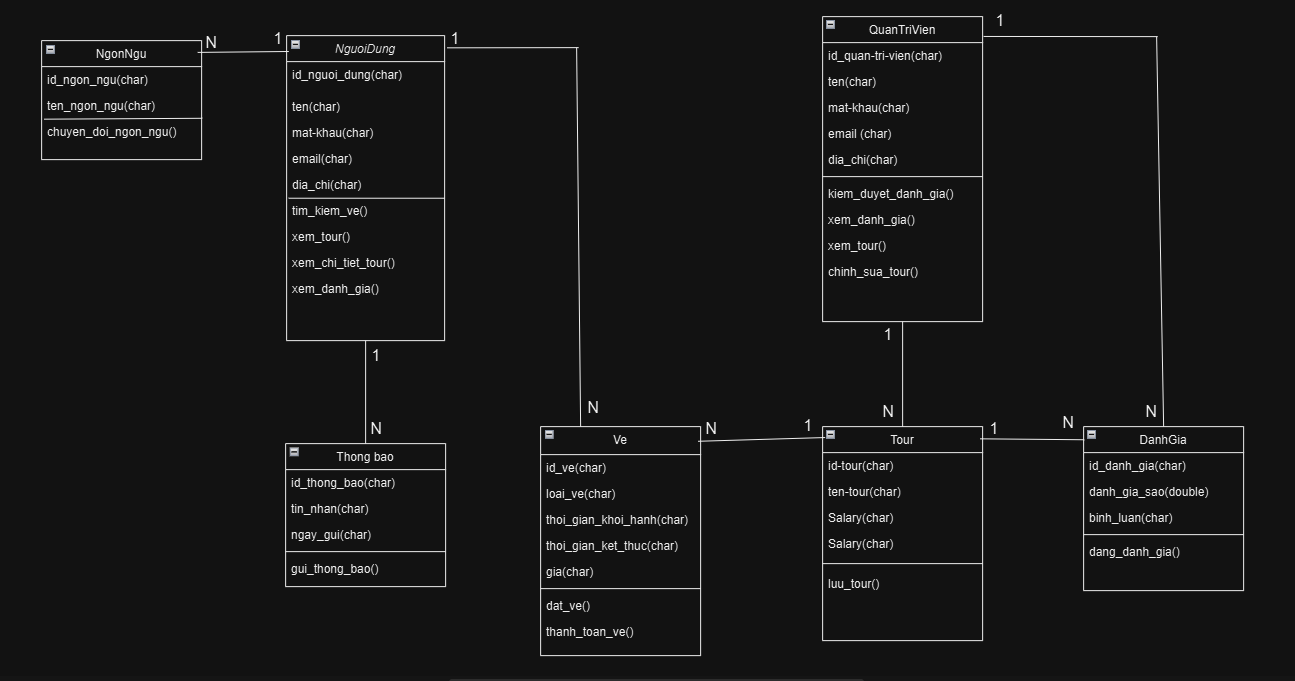
****

# **II. Thiết kế phần mềm**

## 1. **Thiết kế Kiến trúc**



## 2. **Thiết kế Hướng đối tượng**



### 3. **Thiết kế Dữ liệu**

### 3.1 Các thực thể và thuộc tính

* **NGUOIDUNG**(IDNguoiDung, Ten, MatKhau, Email, DiaChi)
* **QUANTRIVIEN**(IDQuanTriVien, Ten, MatKhau, Email, DiaChi)
* **NGONNGU**(IDNgonNgu, TenNgonNgu)
* **DANHG**IA(IDDanhGia, DanhGiaSao, BinhLuan)
* **TOUR**(IDTour, TenTour, TrangThai, DiemDen)
* **THONGBAO**(IDThongBao, TinNhan, NgayGui)
* **VE**(IDVe, LoaiVe, ThoiGianKhoiHanh, ThoiGianKetThuc, Gia)

### 3.2 Mối liên kết

* **NGUOIDUNG** và **NGONNGU** là liên kết 1 – N.
* **QUANTRIVIEN** và **DANHGIA** là liên kết 1 – N.
* **QUANTRIVIEN** và **TOUR** là liên kết 1 – N.
* **NGUOIDUNG** và **TOUR là** liên kết M – N.
* **NGUOIDUNG** và **VE** là liên kết 1 – N.
* **NGUOIDUNG** và **DANHGIA** là liên kết 1 – N.
* **NGUOIDUNG** và **THONGBAO** là liên kết 1 – N.
* **TOUR** và **DANHGIA** là liên kết 1 – N.
* **TOUR** và **VE** 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:

* **NGUOIDUNG**(IDNguoiDung, Ten, MatKhau, Email, DiaChi)
* **QUANTRIVIEN**(IDQuanTriVien, Ten, MatKhau, Email, DiaChi)
* **NGONNGU**(IDNgonNgu, TenNgonNgu)
* **DANHGIA**(IDDanhGia, DanhGiaSao, BinhLuan)
* **TOUR**(IDTour, TenTour, TrangThai, DiemDen)
* **THONGBAO**(IDThongBao, TinNhan, NgayGui)
* **VE**(IDVe, LoaiVe, ThoiGianKhoiHanh, ThoiGianKetThuc, Gia)
* **NGUOIDUNG\_TOUR**(ID\_NguoiDung\_Tour, IDNguoiDung, IDTour)

### 3.4 Chuẩn hóa dữ liệu

### 3.4.1 Xác định phụ thuộc hàm

1. IDNguoiDung xác định một người dùng duy nhất bao gồm thông tin người dùng: Tên, Mật khẩu, Email, Địa chỉ.
2. IDQuanTriVien xác định một quản trị viên duy nhất bao gồm thông tin: Tên, Mật khẩu, Email, Địa chỉ.
3. IDNgonNgu xác định một ngôn ngữ duy nhất gồm thông tin ngôn ngữ: Tên ngôn ngữ.
4. IDDanhGia xác định một đánh giá duy nhất gồm các thông tin liên quan đến đánh giá đó: Đánh giá sao, Bình luận.
5. IDTour xác định một tour duy nhất bao gồm các thông tin về tour: Tên tour, Trạng thái, Điểm đến.
6. IDThongBao xác định một thông báo duy nhất bao gồm thông tin thông báo: Tin nhắn, Ngày gửi.
7. IDVe xác định một vé duy nhất bao gồm thông tin về vé: Loại vé, Thời gian khởi hành, Thời gian kết thúc, Giá.

#### 3.4.2 Xét các dạng chuẩn

* **NGUOIDUNG**(IDNguoiDung, Ten, MatKhau, Email, DiaChi)

Đặt IDNguoiDung = A, Ten = B, MatKhau = C, Email = D,

DiaChi = E

Phụ thuộc hàm:

F = {A → B, A → C, A → D, A → E}

=> 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.

* **QUANTRIVIEN**(IDQuanTriVien, Ten, MatKhau, Email, DiaChi)

Đặt IDQuanTriVien = A, Ten = B, MatKhau = C, Email = D,

DiaChi = E

Phụ thuộc hàm:

F = {A → B, A → C, A → D, A → E}

=> 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.

* **NGONNGU**(IDNgonNgu, TenNgonNgu)

Đặt IDNgonNgu = A, TenNgonNgu = B

Phụ thuộc hàm:

F = {A → B}

=> 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.

* **DANHG**IA(IDDanhGia, DanhGiaSao, BinhLuan)

Đặt IDDanhGia = A, DanhGiaSao = B, BinhLuan = C.

Phụ thuộc hàm:

F = {A → B, A → C}

=> 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.

* **TOUR**(IDTour, TenTour, TrangThai, DiemDen)

Đặt IDTour = A, TenTour = B, TrangThai = C, DiemDen = D

Phụ thuộc hàm:

F = {A → B, A → C, A → D}

=> 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.

* **THONGBAO**(IDThongBao, TinNhan, NgayGui)

Đặt IDThongBao = A, TinNhan = B, NgayGui = C

Phụ thuộc hàm:

F = {A → B, A → C}

=> 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.

* **VE**(IDVe, LoaiVe, ThoiGianKhoiHanh, ThoiGianKetThuc, Gia)

Đặt IDVe = A, LoaiVe = B, ThoiGianKhoiHanh = C, ThoiGianKetThuc = D, Gia = E

Phụ thuộc hàm:

F = {A → B, A → C, A → D, A → E}

=> 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.

* **NGUOIDUNG\_TOUR**(IDNguoiDung\_Tour, IDNguoiDung, IDTour)

Đặt IDNguoiDung\_Tour = A, IDNguoiDung = B, IDTour = C

Phụ thuộc hàm:

F = {A → B, A → C}

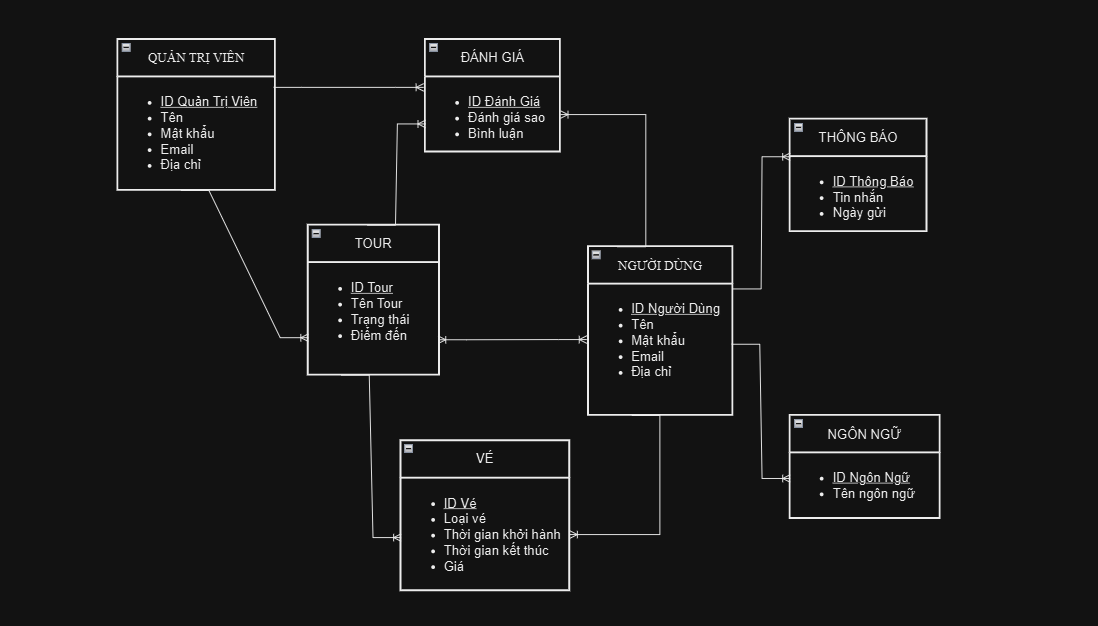
=> Đạt 1NF vì không có thuộc tính đa trị.

=> 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.

### 3.5 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)