**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

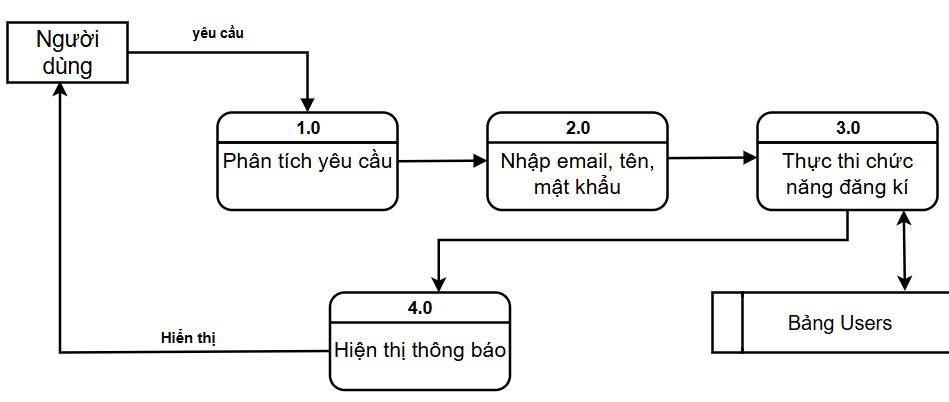
# **I. Tài liệu đặc tả yêu cầu phần mềm**

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

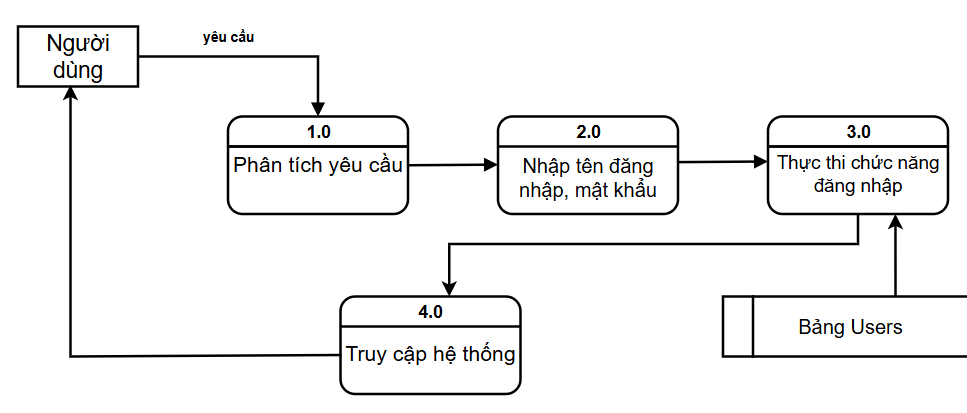


## 1.2 Biểu đồ Luồng dữ liệu (DFD)

### 1.2.1 DFD Đăng ký

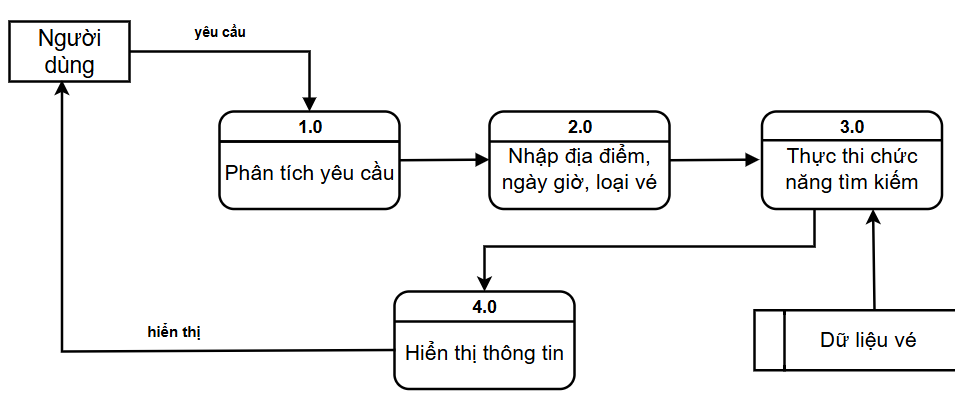


### 1.2.2 DFD Đăng nhập

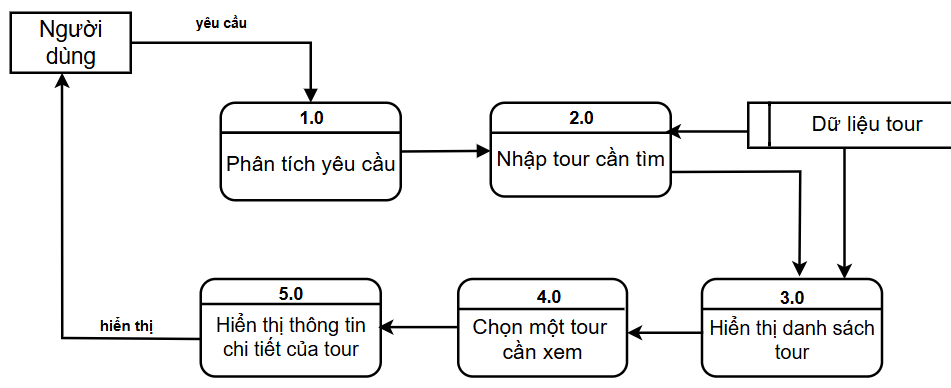


### 

### 1.2.3 DFD Tìm kiếm vé



### 1.2.4 DFD Xem chi tiết thông tin tour



### 1.2.5 DFD

…

### 1.2.6 DFD

…

### 1.2.7 DFD

…

### 1.2.8 DFD

…

### 1.2.9 DFD

…

### 1.2.10 DFD

…

### 1.2.11 DFD

…

## 1.3 Sơ đồ Use case tổng quát



|  |  |  |
| --- | --- | --- |
| stt | Use case | Tác nhân |
| 1 | Đăng ký | Người dùng, quản trị viên |

## 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 | Người dùng |
| *Description* | Chức năng đăng ký cho phép tác nhân đăng ký để tạo được tài khoản đăng nhập vào hệ thống. |
| *Trigger* | Chức năng này được sử dụng khi tác nhân click vào nút “Sign up”. |
| *Pre-conditions* | Tác nhân truy cập vào website |
| *Post conditions* | Tác nhân đăng ký thành công |
| *Flow of Events* | 1. Tác nhân truy cập vào website  2. Tác nhân chọn “Sign up”  3. Hệ thống hiển thị giao diện nhập tên, email, mật khẩu  4. Tác nhân nhập tên, email, mật khẩu và chọn “Sign up”  5. Hệ thống hiển thị thông báo đăng ký thành công và yêu cầu tác nhân xác thực tài khoản  6. Tác nhân click vào link xác thực được gửi trong mail  7. Hệ thống hiển thị giao diện xác thực tài khoản thành công |
| *Alternative Flows* | Nếu tác nhân nhập thông tin không hợp lệ (ví dụ: email sai định dạng hoặc mật khẩu yếu), hệ thống sẽ thông báo và yêu cầu tác nhân chỉnh sửa lại thông tin |
| *Exceptions* | Nếu email tác nhân đã được đăng ký trước đó trong hệ thống, hệ thống sẽ thông báo lỗi và yêu cầu tác nhân nhập email khác. |
| *Special Requirements* | Địa chỉ email đăng ký phải là duy nhất. Mật khẩu phải tuân thủ các yêu cầu bảo mật tối thiểu (độ dài bằng 8 và phải chứa ký tự đặc biệt). |

### 1.4.2 Use case Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | 02 | | |
| Use Case Name | Đăng nhập | | |
| Created By: | … | Last Updated By | … |
| Date Created: | … | Date Last Updated | … |

|  |  |
| --- | --- |
| Actor | Tác nhân |
| *Description* | Chức năng đăng nhập cho phép tác nhân đăng nhập để trải nghiệm đầy đủ các tính năng |
| *Trigger* | Chức năng này được sử dụng khi tác nhân click vào nút “Log in”. |
| *Pre-conditions* | Tác nhân đã có tài khoản |
| *Post conditions* | Tác nhân đăng nhập thành công |
| *Flow of Events* | 1. Tác nhân truy cập vào website  2. Tác nhân chọn “Log in”  3. Hệ thống hiển thị giao diện nhập tên đăng nhập và mật khẩu  4. Tác nhân 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 tác nhân đăng nhập thành công thì hệ thống sẽ chuyển đến trang chủ của hệ thống. |
| *Alternative Flows* | Nếu tác nhân quên mật khẩu:  - Tác nhân truy cập vào trang đăng nhập.  - Tác nhân nhấn vào nút "Forgot password".  -Trang web chuyển hướng tác nhân đến trang khôi phục mật khẩu.  - Tác nhân 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 của tác nhân.  - Tác nhân truy cập vào địa chỉ email và làm theo hướng dẫn để khôi phục mật khẩu. |
| *Exception* |  |
| *Special Requirements* |  |

### 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 |  |
| Date Created: |  | Date Last Updated |  |

|  |  |
| --- | --- |
| 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” (gồm: địa điểm, ngày, ngân sách) 3. Người dùng ấn nút “Tìm kiếm” 4. Hệ thống sẽ hiển thị kết quả tìm kiếm |
| *Alternative Frow* | Nếu không có vé phù hợp với tiêu chí tìm kiếm, hệ thống sẽ thông báo rằng không có kết quả và có thể gợi ý cho người dùng thay đổi tiêu chí tìm kiếm. |
| *Exception* | Nếu người dùng chưa đăng nhập, hệ thống sẽ yêu cầu đăng nhập trước khi thực hiện tìm kiếm vé. |

### 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 | … |
| Date Created: | … | Date Last Updated | … |

|  |  |
| --- | --- |
| Actor | Người dùng |
| *Description* | Chức năng xem thông tin chi tiết cho phép người dùng 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 người dùng 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 |
| *Alternative Flows* | Nếu tour mà người dùng chọn không có thông tin chi tiết hoặc có sự cố, hệ thống sẽ thông báo lỗi và yêu cầu người dùng thử lại sau. |
| *Exception* | Nếu người dùng không đăng nhập trước khi nhấn vào “Xem chi tiết”, hệ thống sẽ yêu cầu đăng nhập trước khi tiếp tục. |

### 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

…

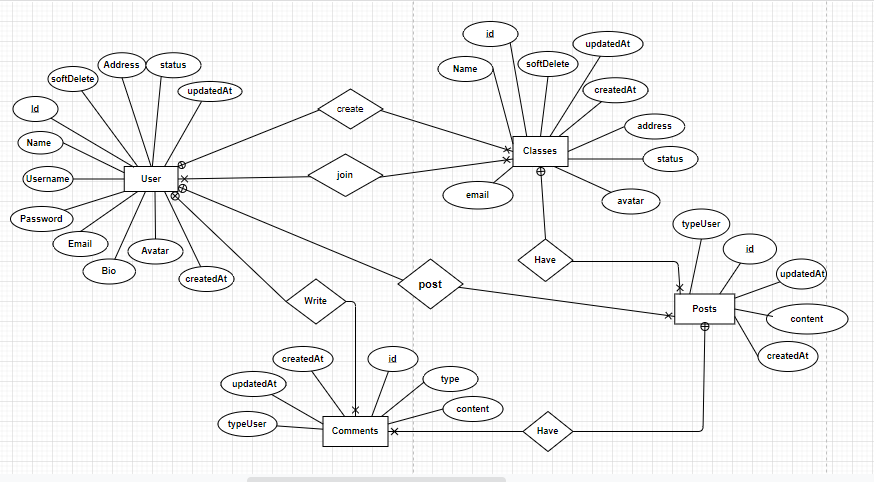
### 1.5.10 SD Tạo lớp tham gia

…

### 1.5.11 SD Tham gia lớp

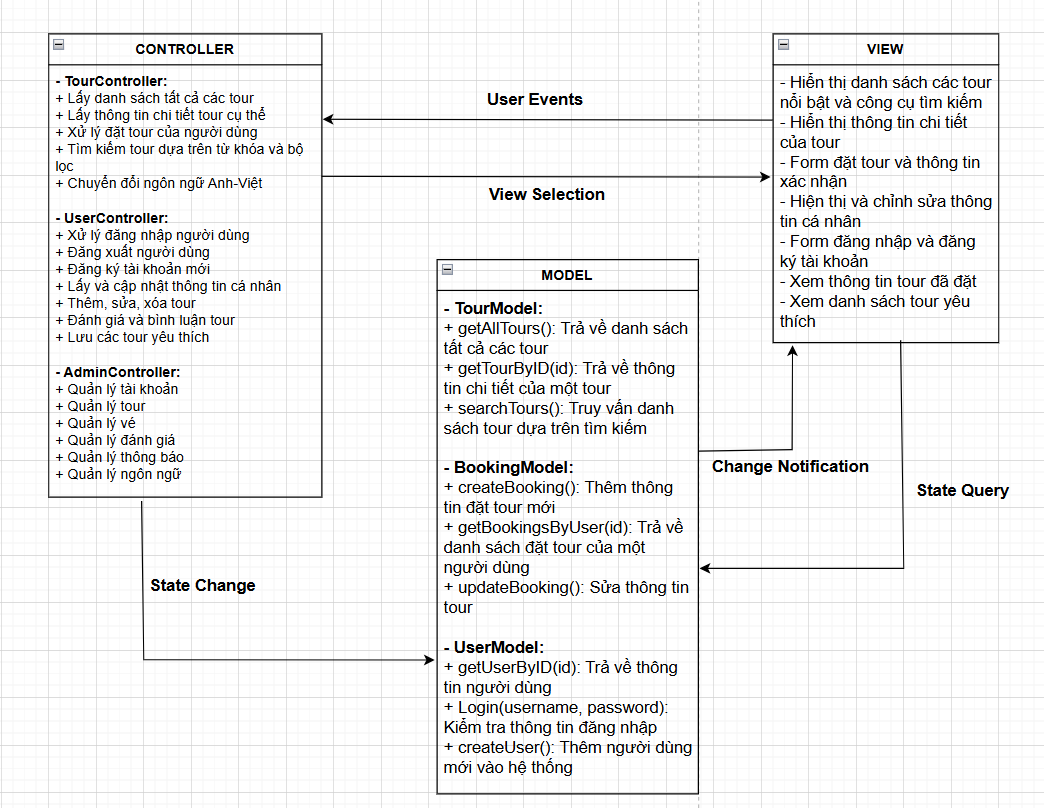
…

## 1.6 Biểu đồ quan hệ thực thể (ERD)



# **II. Thiết kế phần mềm**

## 1. **Thiết kế Kiến trúc**



## 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)