

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN



## Báo cáo về unit test

Teachers:

Nguyễn Lê Hoàng Dũng

Members:

Names	MSSV
Lê Minh Nhật	22127308

1. Testing Framework .....	3
2. Các thành phần được test, hiệu quả và cải tiến chương trình .....	3
2.1 Test chức năng thêm khoa mới .....	3
2.2 Test chức năng đổi tên khoa .....	3
2.3 Test chức năng thêm chương trình mới .....	3
2.4 Test chức năng đổi tên chương trính.....	4
2.5 Test chức năng thêm trạng thái mới.....	4
2.6 Test chức năng đổi tên chương trính.....	4
2.7 Test các chức năng student .....	4

## 1. Testing Framework

Sử dụng Jest làm Testing Framework chính cho project.

## 2. Các thành phần được test, hiệu quả và cải tiến chương trình

### 2.1 Test chức năng thêm khoa mới

- Test 3 trường hợp:
  - TH1: Thêm một khoa mới hoàn toàn. Kết quả: passed.
  - TH2: Thêm một khoa có tên rỗng. Kết quả: fail. Cải tiến thêm

```
if (!department) {  
  console.log("🚨 Không thêm department do tên trống");  
  return;  
}
```

- TH3: Thêm một khoa trùng tên với khoa có sẵn. Kết quả fail. Cải tiến thêm

```
const isDuplicate = departments.some(d => d.Department.toLowerCase() === department.toLowerCase());  
if (isDuplicate) {  
  console.log("🚨 Khoa đã tồn tại, không thể thêm!");  
  return;  
}
```

### 2.2 Test chức năng đổi tên khoa

- Test 3 trường hợp:
  - TH1: Đổi tên khoa tồn tại và lưu. Kết quả: passed.
  - TH2: Đổi tên khoa không tồn tại. Kết quả: passed.
  - TH3: Đổi tên khoa tồn tại thành tên rỗng. Kết quả: fail. Cải tiến thêm

```
if (!s.Department) {  
  console.log("🚨 Không thêm department do tên trống");  
  return;  
}
```

### 2.3 Test chức năng thêm chương trình mới

- Test 3 trường hợp:
  - TH1: Thêm một chương trình mới hoàn toàn. Kết quả: passed.
  - TH2: Thêm một chương trình có tên rỗng. Kết quả: fail. Cải tiến thêm

```
if (!program) {  
  console.log("🚨 Không thêm chương trình do tên trống");  
  return;  
}
```

- TH3: Thêm một chương trình trùng tên với khoa có sẵn. Kết quả fail. Cải tiến thêm

```
const isDuplicate = programs.some(d => d.Program.toLowerCase() === program.toLowerCase());  
if (isDuplicate) {  
  console.log("🚨 Chương trình đã tồn tại, không thể thêm!");  
  return;  
}
```

## 2.4 Test chức năng đổi tên chương trình

- Test 3 trường hợp:
  - TH1: Đổi tên chương trình tồn tại và lưu. Kết quả: passed.
  - TH2: Đổi tên chương trình không tồn tại. Kết quả: passed.
  - TH3: Đổi tên chương trình tồn tại thành tên rỗng. Kết quả: fail. Cải tiến thêm

```
if (!s.Program) {  
    console.log("🚫 Không đổi tên chương trình do tên trống");  
    return;  
}
```

## 2.5 Test chức năng thêm trạng thái mới

- Test 3 trường hợp:
  - TH1: Thêm một trạng thái mới hoàn toàn. Kết quả: passed.
  - TH2: Thêm một trạng thái có tên rỗng. Kết quả: fail. Cải tiến thêm

```
if (!status) {  
    console.log("🚫 Không thêm trạng thái do tên trống");  
    return;  
}
```

- TH3: Thêm một trạng thái trùng tên với khoa có sẵn. Kết quả fail. Cải tiến thêm

```
const isDuplicate = statuses.some(d => d.Status.toLowerCase() === status.toLowerCase());  
if (isDuplicate) {  
    console.log("🚫 trạng thái đã tồn tại, không thể thêm!");  
    return;  
}
```

## 2.6 Test chức năng đổi tên trạng thái

- Test 3 trường hợp:
  - TH1: Đổi tên trạng thái tồn tại và lưu. Kết quả: passed.
  - TH2: Đổi tên trạng thái không tồn tại. Kết quả: passed.
  - TH3: Đổi tên trạng thái tồn tại thành tên rỗng. Kết quả: fail. Cải tiến thêm

```
if (!s.Status) {  
    console.log("🚫 Không đổi tên trạng thái do tên trống");  
    return;  
}
```

## 2.7 Test các chức năng student

- Test 7 tính năng
  - Tính năng thêm học sinh. Kết quả: passed.
  - Tính năng xóa học sinh. Kết quả: passed.
  - Tính năng cập nhật thông tin học sinh. Kết quả passed.
  - Tính năng tìm kiếm học sinh theo tên. Kết quả passed.
  - Tính năng tìm kiếm học sinh theo MSSV. Kết quả passed.

- Tính năng tìm kiếm các học sinh trong khoa. Kết quả: passed.
- Tính năng tìm kiếm các học sinh trong khoa và tên. Kết quả passed.