

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**Khoa Công Nghệ Thông tin**

----- o0o -----



**BÁO CÁO BÀI TẬP LỚN**  
**MÔN: KIỂM THỬ PHẦN MỀM**

**Đề tài: Nghiên cứu kỹ thuật kiểm thử Frontend, Backend, GUI  
và tính khả dụng.**

**Hà Nội, 2021**

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**Khoa Công Nghệ Thông tin**

----- o0o -----



**BÁO CÁO BÀI TẬP LỚN**  
**MÔN: KIỂM THỬ PHẦN MỀM**

**Đề tài: Nghiên cứu kỹ thuật kiểm thử Frontend, Backend, GUI  
và tính khả dụng.**

**Giáo viên hướng dẫn:**

**Nhóm sinh viên thực hiện:**

**Thành viên nhóm:**

**Lớp:**

**Hoàng Quang Huy**

**Nhóm 13**

**Bùi Thị Hải – CNTT5- K14**

**Nguyễn Thị Tú Anh – CNTT5 – K14**

**Hà Thị Phụng – CNTT5 – K14**

**Hoàng Thị Kim – CNTT5 –K14**

**202110503191002**

**Hà Nội, 2021**

## **Lời mở đầu**

Kiểm thử phần mềm được biết đến là một tiến trình hay một tập hợp tiến trình được thiết kế để thực thi chương trình với mục tiêu cụ thể là tìm ra lỗi trước khi bàn giao tới khách hàng. Kiểm thử phần mềm chiếm một phần rất quan trọng trong quá trình phát triển phần mềm.

Kiểm thử chiếm rất nhiều thời gian và công sức trong quá trình phát triển phần mềm, với mục tiêu hướng đến sự thân thiện với người dùng và tăng độ tin cậy của sản phẩm phần mềm. Kiểm thử phần mềm có phạm vi rất lớn nên chủ đề nghiên cứu của nhóm chỉ xoay quanh những loại kiểm thử thường gặp và cần được thực hiện để đảm bảo tính thân thiện cũng như độ tin cậy và chất lượng của một sản phẩm phần mềm:

### **Nội dung bài báo cáo gồm 4 chương:**

Chương 1: Kiểm thử Backend (Backend testing).

Chương 2: Kiểm thử Frontend (Frontend testing).

Chương 3: Kiểm thử giao diện đồ họa người dùng (GUI testing).

Chương 4: Kiểm thử tính khả dụng (Usability testing).

Bản báo cáo này được hoàn thành dưới sự chỉ dẫn của thầy giáo Hoàng Quang Huy – giảng viên bộ môn Kiểm thử phần mềm. Nhưng do còn thiếu kinh nghiệm thực tế và gặp khó khăn trong việc tìm tài liệu và dịch thuật tài liệu tham khảo nên có thể bài báo cáo vẫn còn một vài thiếu sót. Vậy nên, chúng em hi vọng nhận được sự nhận xét, đóng góp ý kiến từ thầy cô và các bạn để bản báo cáo được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

***Nhóm sinh viên thực hiện báo cáo.***

## Mục lục

Lời mở đầu .....	3
Phân chia công việc trong nhóm .....	8
Chương 1: Kiểm thử backend ( Backend Testing ). .....	9
1. Kiểm thử Back-end là gì?.....	9
1.1. Khái niệm .....	9
1.2. Mục đích.....	9
2. Vị trí và cách thực hiện .....	9
2.1. Kiểm thử Back-end chủ yếu bao gồm xác thực: .....	9
2.2. Vị trí thực hiện .....	10
2.3. Cách thực hiện.....	11
3. Các loại kiểm thử cơ sở dữ liệu(Backend Testing hay Database Testing) .....	11
3.1. Kiểm thử cấu trúc(Structural Testing) .....	11
3.1.1.Kiểm tra lược đồ ( Schema Testing ) .....	13
3.1.2. Kiểm tra cột và bảng cơ sở dữ liệu(Database Table,Column Testing)..	13
3.1.3. Kiểm tra các khóa và chỉ số(Keys and Indexes Testing) .....	14
3.1.4. Kiểm tra thủ tục được lưu trữ(Stored Procedures Testing).....	15
3.1.5. Kiểm tra trình kích hoạt(Triigger testing).....	16
3.1.6. Xác thực máy chủ cơ sở dữ liệu(Database server validations).....	17
3.2. Kiểm thử chức năng (Function Testing) .....	18
3.2.1. Tính nguyên tử(Atomicity): .....	18
3.2.2. Tính nhất quán(consistency) : .....	18
3.2.3. Tính cô lập (Isolation):.....	18
3.2.4. Tính bền vững .....	19
3.2.1. Kiểm tra tính toàn vẹn và nhất quán của dữ liệu ( Checking data integrity and consistency) .....	19
3.2.2. Đăng nhập và bảo mật người dùng (Login and user security).....	20
3.3. Kiểm thử phi chức năng (Non-Function Testing).....	20
3.3.1. Kiểm thử việc tải (Load Testing) .....	20

3.3.2. Kiểm thử áp lực (Database stress testing hay torturous testing or fatigue testing) .....	21
4. Các công cụ kiểm thử back end tốt nhất .....	21
4.1. Data Factory: .....	21
4.2. Data Generator: .....	21
4.3. TurboData.....	21
4.4. TOAD .....	21
4.5. php MyAdmin .....	21
5. Lợi ích ,vấn đề và giải pháp của kiểm thử Backend .....	21
5.1. Lợi ích .....	22
5.2. Vấn đề và giải pháp:.....	22
6. Những quan niệm sai lầm về kiểm thử cơ sở dữ liệu ( Backend testing or database testing) .....	23
7. Kết luận .....	23
7.1. Nội dung.....	23
7.2. Tài liệu tham khảo.....	24
Chương 2: Kiểm thử Frontend (Frontend Testing).....	25
1. Kiểm thử giao diện người dùng là gì?.....	25
1.1. Khái niệm .....	25
1.2. Mục tiêu.....	25
1.3. Ví dụ :.....	25
2. Làm thế nào để tạo một kế hoạch kiểm tra giao diện người dùng cho trang web? .....	26
3. Tại sao phải tạo kế hoạch kiểm thử giao diện người dùng ?.....	27
3.1. Kế hoạch Kiểm tra giao diện người dùng giúp bạn xác định .....	27
3.2. Bằng cách tạo kiểm thử giao diện người dùng, kế hoạch của bạn sẽ nhận được những lợi thế sau: .....	29
4. Những mẹo cho việc kiểm thử giao diện người dùng tốt hơn .....	29
5. Các công cụ kiểm thử giao diện người dùng .....	29
5.1. Công cụ kiểm thử đa trình duyệt (Cross-Browser Testing Tool): .....	29

5.1.1. LambdaTest ( <a href="https://app.lambdatest.com/">https://app.lambdatest.com/</a> ) .....	29
5.2. Công cụ kiểm thử JavaScript ( JS testing tool):.....	32
5.2.1. Jasmine .....	32
5.3. Công cụ kiểm thử chức năng (Functional testing tool):.....	32
5.3.1. Selenium.....	32
5.4. Công cụ kiểm thử CSS (CSS tool):.....	33
5.4.1. Needle.....	33
6. Tối ưu hóa hiệu suất giao diện ( Front-End Performance Optimization) .....	34
7. Tại sao tối ưu hóa hiệu suất Front-End lại quan trọng? .....	34
8. Lợi ích của tối ưu hóa hiệu suất Front-End là gì? .....	34
9. Công cụ kiểm tra hiệu suất giao diện người dùng .....	35
9.1. Tốc độ tải trang ( Page Speed) .....	35
9.2. Yslow.....	36
10. Kết luận .....	36
10.1. Nội dung .....	36
10.2. Tài liệu tham khảo.....	36
Chương 3: Kiểm thử giao diện đồ họa người dùng (GUI Testing).....	38
1. GUI là gì? .....	38
2. Kiểm thử giao diện đồ họa người dùng là gì? (GUI Testing).....	38
3. Sự cần thiết của kiểm thử giao diện đồ họa người dùng? .....	39
4. Cần làm gì trong kiểm thử giao diện đồ họa người dùng (GUI Testing)?.....	39
5. Các kỹ thuật kiểm thử giao diện đồ họa người dùng: .....	40
5.1. Kiểm tra bằng tay (Manual Based Testing) .....	40
5.2. Ghi lại và phát triển (Record and Replay) .....	40
5.3. Kiểm thử dựa trên mô hình (Model Based Testing) .....	41
6. Các mẫu ca kiểm thử trong kiểm thử giao diện đồ họa người dùng (GUI testing). .....	42
7. Những thách thức trong kiểm thử giao diện đồ họa người dùng. ....	43
8. Demo: Cách thực hiện kiểm thử giao diện đồ họa người dùng. ....	43

9. Các công cụ kiểm thử giao diện đồ họa người dùng.....	44
10. Kết luận. ....	44
10.1. Nội dung.....	44
10.2. Tài liệu tham khảo.....	44
Chương 4: Kiểm thử tính khả dụng (Usability Testing) .....	46
1. Khái niệm .....	46
2. Lý do tại sao nên kiểm thử tính khả dụng.....	46
3. Mục tiêu kiểm thử tính khả dụng .....	46
3.1. Hiệu quả của hệ thống.....	46
3.2. Hiệu quả .....	47
3.3. Độ chính xác.....	47
3.4. Người dùng thân thiện.....	47
4. Quy trình kiểm thử tính khả dụng .....	47
4.1. Lập kế hoạch .....	47
4.2. Tuyển dụng.....	47
4.3. Kiểm thử tính khả dụng.....	48
4.4. Phân tích dữ liệu.....	48
4.5. Báo cáo .....	48
5. Các phương pháp kiểm thử tính khả dụng .....	48
5.1. Kiểm thử tính khả dụng trong phòng thí nghiệm.....	48
5.2. Kiểm thử tính khả dụng từ xa.....	48
6. Danh sách các mục cần kiểm tra cho kiểm thử trải nghiệm người dùng.....	48
7. Ưu điểm của kiểm thử tính khả dụng.....	49
8. Nhược điểm của kiểm thử tính khả dụng .....	49
9. Kết luận .....	49
9.1. Nội dung.....	49
9.2. Tài liệu tham khảo.....	49
Tài liệu tham khảo.....	50

## Phân chia công việc trong nhóm

STT	Người thực hiện	Nội dung công việc	Phương pháp thực hiện	Kết quả đạt được
1	Cả Nhóm	Họp thành viên: Tổng hợp những điều cần phải làm và lên kế hoạch thực hiện	Trao đổi, thảo luận giữa các thành viên trong nhóm	Đã hoàn thành
2	Nguyễn Thị Tú Anh	Thực hiện tìm hiểu <b>Chương 1: Kiểm thử backend ( Backend Testing )</b> .	Tìm hiểu, trình bày theo form đã thống nhất	Đã hoàn thành
3	Bùi Thị Hải	Thực hiện tìm hiểu: <b>Chương 2: Kiểm thử Frontend (Frontend Testing)</b> .	Tìm hiểu, trình bày theo form đã thống nhất	Đã hoàn thành
4	Hoàng Thị Kim	Thực hiện tìm hiểu : <b>Chương 3: Kiểm thử giao diện đồ họa người dùng (GUI Testing)</b>	Tìm hiểu, trình bày theo form đã thống nhất	Đã hoàn thành
5	Hà Thị Phụng	Thực hiện tìm hiểu <b>Chương 4: Kiểm thử tính khả dụng (Usability Testing)</b> .	Tìm hiểu, trình bày theo form đã thống nhất	Đã hoàn thành
6	Bùi Thị Hải	Tổng hợp nội dung của các thành viên, Viết báo cáo bài tập lớn	Tổng hợp nội dung thành báo cáo, chỉnh sửa( nếu có)	Đã hoàn thành
7	Cả nhóm	Tổng kết lại những điều đã làm được và chưa làm được, đưa ra bài học kinh nghiệm	Họp thành viên, thảo luận bài làm, chỉnh sửa nếu có	Đã hoàn thành



## Chương 1: Kiểm thử backend ( Backend Testing ).

### 1. Kiểm thử Back-end là gì?



#### 1.1. Khái niệm

Kiểm thử Backend(Backend Testing hay Database Testing) có nghĩa là kiểm thử phía sau giao diện hay kiểm thử cơ sở dữ liệu, là một phương pháp kiểm thử nhằm kiểm tra phía máy chủ hoặc cơ sở dữ liệu của các ứng dụng web hoặc một phần mềm.

Dữ liệu được nhập vào front end sẽ được lưu trữ trong cơ sở dữ liệu(backend ). Cơ sở dữ liệu có thể là SQL Server, MySQL, Oracle, DB2, v.v. Dữ liệu sẽ được tổ chức trong các bảng dưới dạng bản ghi và nó được sử dụng để hỗ trợ nội dung của trang.

#### 1.2. Mục đích

Mục đích của kiểm thử Back-end là kiểm tra lớp ứng dụng hoặc lớp cơ sở dữ liệu để đảm bảo rằng ứng dụng web hoặc phần mềm không có lỗi cơ sở dữ liệu như bế tắc, hỏng dữ liệu hoặc mất dữ liệu.

Kiểm thử Back-end rất quan trọng vì nếu nó không được thực hiện đúng cách, nó có thể gây ra một số biến chứng nghiêm trọng như bế tắc, hỏng dữ liệu, mất dữ liệu, ...

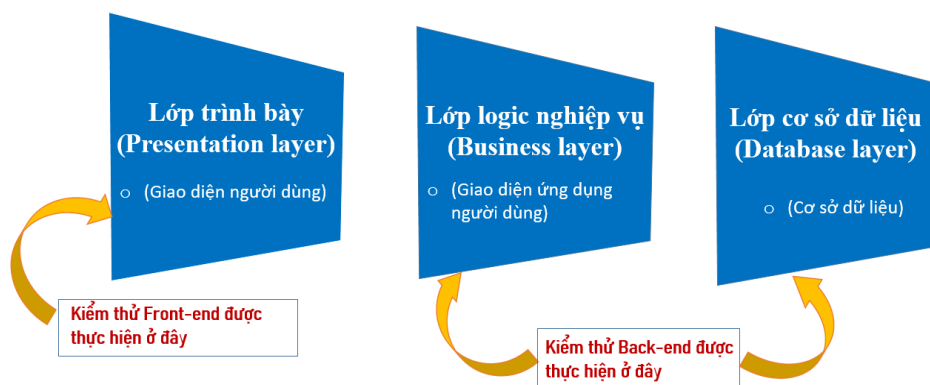
## 2. Vị trí và cách thực hiện

### 2.1. Kiểm thử Back-end chủ yếu bao gồm xác thực:

- Lược đồ (schema)
- Bảng cơ sở dữ liệu (Data tables)
- Cột (Columns)

- Các khóa và chỉ mục (Keys and Indexes)
- Các thủ tục được lưu trữ (stored procedures)
- Các kích hoạt(triggers)
- Xác thực máy chủ cơ sở dữ liệu (Database server validations)
- Xác thực việc sao chép dữ liệu (Validating data duplication)

## 2.2. Vị trí thực hiện



Hình ảnh: Vị trí thực hiện kiểm thử Back-end

- Lớp trình bày(Presentation layer): có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện ( win form, web form,...) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business.  
VD: Người dùng thực hiện đăng kí tài khoản. Hệ thống hiển thị màn hình form nhập yêu cầu người dùng nhập dữ liệu. Sau đó hệ thống kiểm tra tính đúng đắn của dữ liệu rồi chuyển lên lớp Business.
- Lớp logic nghiệp vụ (BLL-Business logic layer hay business layer), lớp này thuộc lớp ứng dụng và phân ra thành 2 nhiệm vụ:
  - + Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation layer trước khi truyền xuống Database layer và lưu xuống hệ quản trị CSDL.
  - + Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation layer.  
VD: Sau khi người dùng nhập dữ liệu hợp lệ,lớp logic nghiệp vụ tiến hành mã hóa dữ liệu rồi truyền xuống lớp cơ sở dữ liệu.
- Lớp cơ sở dữ liệu (DAL-data access layer hay database layer): Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu ( tìm kiếm, thêm, xóa, sửa,...).

VD: Dữ liệu lúc này sẽ được kiểm tra xem có trùng lặp hay không?

Nếu không trùng lặp thì tiến hành thêm vào cơ sở dữ liệu.

Nếu trùng thì trả kết quả về lớp business sau đó lớp lại trả kết quả về lớp trình bày tại đây hiển thị yêu cầu nhập lại dữ liệu cho hợp lệ.

### 2.3. Cách thực hiện

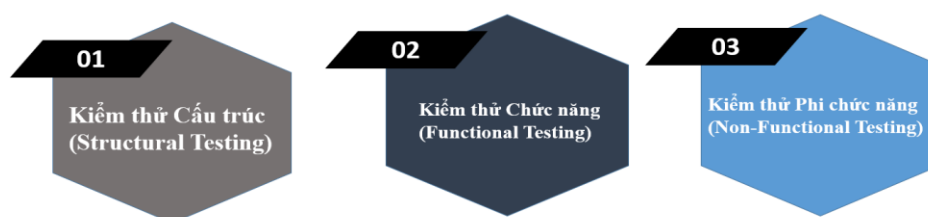
- Bước 1.Chuẩn bị môi trường.
- Bước 2.Chạy kiểm thử.
- Bước 3.Kiểm tra kết quả kiểm thử.
- Bước 4:Xác thực theo kết quả mong đợi.
- Bước 5:Báo cáo kết quả cho các bên liên quan tương ứng.

#### Chú ý:

- Bạn không bắt buộc phải sử dụng giao diện đồ họa người dùng(GUI)
- Bạn có thể trực tiếp chuyển tiếp yêu cầu thông qua một số trình duyệt và các tham số cần thiết cho chức năng và nhận được phản hồi ở một số định dạng mặc định.Ví dụ: XML(Extensible Markup Language) hoặc JSON(JavaScript Object Notation).
- Hay bạn có thể kết nối trực tiếp với cơ sở dữ liệu và xác minh dữ liệu bằng cách sử dụng truy vấn SQL.

### 3. Các loại kiểm thử cơ sở dữ liệu(Backend Testing hay Database Testing)

- Gồm 3 loại:



*Hình ảnh: Các loại kiểm thử cơ sở dữ liệu*

#### 3.1. Kiểm thử cấu trúc(Structural Testing)

- Kiểm thử cấu trúc xử lý cấu trúc bên trong cơ sở dữ liệu (tức là siêu dữ liệu). Nó liên quan đến việc xác thực Bảng và tất cả các đối tượng cơ sở dữ liệu khác mà người dùng không thể truy cập trực tiếp, như Cột, Khóa, Chỉ mục, Lược đồ, Trình kích hoạt, Hàm, .... Nó cũng có thể liên quan đến việc kiểm tra các máy chủ DB của bạn.
- Kiểm thử cấu trúc bao gồm :

**Ví dụ:** Cho CSDL Quản lý sinh viên:

Khoa(MaKhoa,TenKhoa,SoDT)

Lop(MaLop,TenLop,SiSo,MaKhoa)

SinhVien(MaSV,HoTen,GioiTinh,NgaySinh,MaLop)

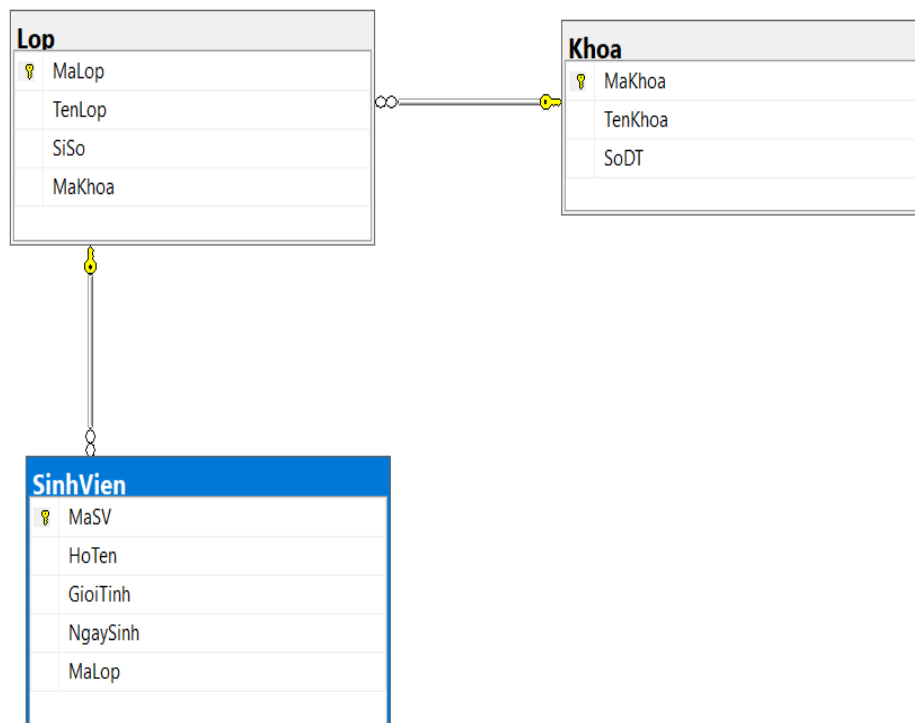
Code:

```

create database QLSV
use QLSV
create table Khoa
(
    MaKhoa char(10) not null primary key,
    TenKhoa nvarchar(30),
    SoDT char(10)
)
create table Lop
(
    MaLop char(10) not null primary key,
    TenLop nvarchar(30),
    SiSo int,
    MaKhoa char(10) not null,
    constraint fk1 foreign key (MaKhoa) references Khoa(MaKhoa) on update cascade on delete cascade
)
create table SinhVien
(
    MaSV char(10) not null primary key,
    HoTen nvarchar(30),
    GioiTinh bit,
    NgaySinh date,
    MaLop char(10) not null,
    constraint fk2 foreign key (MaLop) references Lop(MaLop) on update cascade on delete cascade
)

```

Lược đồ cơ sở dữ liệu :



### 3.1.1. Kiểm tra lược đồ ( Schema Testing )

**VD:** Thực hiện kiểm tra View để cập nhập lại sĩ số Lớp thì sĩ số Lớp sẽ được cập nhập lại đúng theo số lượng sinh viên đã được nhập.

a) Code:

```
create view CapNhat
as
select Lop.MaLop, TenLop, Count(*) as 'SiSo', MaKhoa
from Lop inner join SinhVien on Lop.MaLop=SinhVien.MaLop
group by Lop.MaLop, TenLop, MaKhoa
```

b) Dữ liệu ban đầu: với sĩ số nhập vào không trùng khớp với số lượng sinh viên

	MaLop	TenLop	SiSo	MaKhoa
1	01	CNTT1	70	1
2	02	Dien Tu 4	70	3
3	03	Co khi 2	70	2

	MaSV	HoTen	GioiTinh	NgaySinh	MaLop
1	001	Nguyen Van A	0	1999-01-12	03
2	002	Nguyen Van B	1	2000-03-02	01
3	003	Nguyen Van C	0	2001-12-09	03
4	004	Nguyen Van g	1	1999-10-12	03
5	005	Nguyen Van m	0	1999-01-03	02

c) Kết quả sau khi chạy view: Kết quả thu được chính xác

	MaLop	TenLop	SiSo	MaKhoa
1	01	CNTT1	1	1
2	02	Dien Tu 4	1	3
3	03	Co khi 2	3	2

### 3.1.2. Kiểm tra cột và bảng cơ sở dữ liệu(Database Table, Column Testing)

**Ví dụ:** Khi thực hiện insert vào bảng Khoa phù hợp với kiểu dữ liệu được định dạng bên trên với kiểu dữ liệu của số điện thoại là kiểu char(10).

❖ Xét các trường hợp

a) TH - không hợp lệ: Nếu người dùng nhập quá 10 kí tự thì sẽ không thực hiện insert và báo lỗi.

- Code:

```
insert into Khoa values('4','Du lịch','09029929292929')
```

- Kết quả:

Messages

Msg 2628, Level 16, State 1, Line 29

String or binary data would be truncated in table 'QLSV.dbo.Khoa', column 'SoDT'. Truncated value: '0902992929'.

The statement has been terminated.

b) TH - hợp lệ: Nếu người dùng nhập đúng chính xác các trường thuộc bảng Khoa thì thực hiện insert vào bảng khoa

- Code:

```
insert into Khoa values('1','CNTT','0123456789')
insert into Khoa values('2','Co Khi','0925898955')
insert into Khoa values('3','Dien Tu','0385545545')
```

- Kết quả:

select \* from Khoa

93 %

	MaKhoa	TenKhoa	SoDT
1	1	CNTT	0123456789
2	2	Co Khi	0925898955
3	3	Dien Tu	0385545545

❖ Khi thực hiện kiểm tra bảng và cột:

- Kiểm tra toàn bộ tính hợp lệ của dữ liệu nhập vào.
- Yêu cầu tất cả dữ liệu phải nhập đúng thì mới thực hiện.

### 3.1.3. Kiểm tra các khóa và chỉ số(Keys and Indexes Testing)

**Ví dụ:** Trong bảng Lớp có khóa chính là MaLop, Khóa ngoài là MaKhoa

- Xét các trường hợp :

a) TH - không hợp lệ: Nếu mã khoa không tồn tại trong bảng Khoa thì không thêm và hiển thị lỗi

+ Code:

```
insert into Lop values('04','CNTT5',70,'4')
```

+ Kết quả:

## Messages

Msg 547, Level 16, State 0, Line 34

The INSERT statement conflicted with the FOREIGN KEY constraint "fk1". The conflict occurred in database "QLSV", table "dbo.Khoa", column 'MaKhoa'.

The statement has been terminated.

b) TH - hợp lệ: Nếu mã khoa đã tồn tại trong bảng Khoa thì thực hiện thêm vào bảng Lóp

+ Code:

```
insert into Lop values('01','CNTT1',70,'1')
insert into Lop values('03','Co khi 2',70,'2')
insert into Lop values('02','Dien Tu 4',70,'3')
```

+ Kết quả:

select \*from Lop

93 %

Results Messages

	MaLop	TenLop	SiSo	MaKhoa
1	01	CNTT1	70	1
2	02	Dien Tu 4	70	3
3	03	Co khi 2	70	2

### 3.1.4. Kiểm tra thủ tục được lưu trữ(Stored Procedures Testing)

**Ví dụ:**Thực hiện thủ tục thêm 1 sinh viên vào bảng SinhVien nếu MaLop của sinh viên đó tồn tại.

a) Code:

```
create proc Them_SV(@MaSV char(10),@HoTen nvarchar(30),@GioiTinh bit,@NgaySinh date,@MaLop nvarchar(10))
as
begin
if(exists(select MaLop from Lop where MaLop=@MaLop))
insert into SinhVien values(@MaSV,@HoTen,@GioiTinh,@NgaySinh,@MaLop)
else
raiserror('không thể thêm',16,1)
end
```

b) Xét các trường hợp:

- TH 1: MaLop không hợp lệ (không thuộc bảng Lóp ) thì không thực hiện thêm sinh viên và báo lỗi

+ Code:

```
exec Them_SV '006','Nguyen Van k',0,'9/9/1999','04'
```

+ Kết quả:

## Messages

Msg 50000, Level 16, State 1, Procedure Them\_SV, Line 7 [Batch Start Line 73]  
không thể thêm

- TH 2: MaLop đã tồn tại thì thực hiện thêm vào bảng

+ Code, Kết quả:

```
exec Them_SV '006','Nguyen Van k',0,'9/9/1999','01'
select *from SinhVien
```

93 %

	MaSV	HoTen	GioiTinh	NgaySinh	MaLop
1	001	Nguyen Van A	0	1999-01-12	03
2	002	Nguyen Van B	1	2000-03-02	01
3	003	Nguyen Van C	0	2001-12-09	03
4	004	Nguyen Van g	1	1999-10-12	03
5	005	Nguyen Van m	0	1999-01-03	01
6	006	Nguyen Van k	0	1999-09-09	01

c) Một số công cụ Kiểm tra Cơ sở dữ liệu hữu ích để kiểm tra các thủ tục được lưu trữ là LINQ, công cụ SP Test, ...

### 3.1.5. Kiểm tra trình kích hoạt(Trigger testing)

Ví dụ: Hãy tạo Trigger để tự động tăng **sĩ số** sinh viên trong bảng lớp, mỗi khi thêm mới dữ liệu cho bảng Sinh viên. Nếu tuổi của sinh viên dưới 18 tuổi thì không cho thêm ngược lại thì thêm vào và update lại sĩ số lớp.

a) Code:

```
create trigger Them_moi
on SinhVien
for insert
as
begin
declare @Tuoi int
declare @SS int
set @SS=(select count(*)from SinhVien inner join inserted on SinhVien.MaSV=inserted.MaSV)
set @Tuoi=(select year(getdate())-year(NgaySinh) from inserted)
if(@Tuoi<18)
begin
raiserror('khong the them',16,1)
rollback tran
end
else
update Lop set SiSo=@SS from Lop inner join inserted on Lop.MaLop=inserted.MaLop
end
```

b) Xét các trường hợp

- TH1:Dưới 18 tuổi không thể thêm và hiển thị thông báo lỗi.

Code và kết quả

```
insert into SinhVien values('007','Nguyen Van m',0,'1/3/2006','02')
```

93 %

Msg	Severity	Level	State	Procedure	Line	Batch Start Line
Msg 50000, Level 16, State 1, Procedure Them_moi, Line 12 [Batch Start Line 156]						
khong the them						
Msg 2609, Level 16, State 1, Line 157						
The transaction ended in the trigger. The batch has been aborted.						

- TH 2:Trên 18 tuổi thì thực hiện thêm vào bảng SinhVien và cập nhật lại sĩ số Lớp.

Code và kết quả



```

insert into SinhVien values('007','Nguyen Van m',0,'1/3/2001','02')
select *from SinhVien
select *from Lop
create trigger cau7
on SinhVien

```

	MaSV	HoTen	GioiTinh	NgaySinh	MaLop
1	001	Nguyen Van A	0	1999-01-12	03
2	002	Nguyen Van B	1	2000-03-02	01
3	003	Nguyen Van C	0	2001-12-09	03
4	004	Nguyen Van g	1	1999-10-12	03
5	005	Nguyen Van m	0	1999-01-03	01
6	006	Nguyen Van k	0	1999-09-09	01
7	007	Nguyen Van m	0	2001-01-03	02

	MaLop	TenLop	SiSo	MaKhoa
1	01	CNTT1	70	1
2	02	Dien Tu 4	1	3
3	03	Co khi 2	70	2

- TH3.Trên 18 tuổi và MaLop không tồn tại thì hiện thông báo lỗi

+ Code và kết quả

```

insert into SinhVien values('008','Nguyen Van m',0,'1/3/2001','04')

```

93 %

Messages

Msg 547, Level 16, State 0, Line 159  
The INSERT statement conflicted with the FOREIGN KEY constraint "fk2". The conflict occurred in database "QLSV", table "dbo.Lop", column 'MaLop'.  
The statement has been terminated.

c) Khi thực hiện kiểm thử trình kích hoạt:

- Kiểm tra các trường hợp có thể xảy ra.
- Kiểm tra tính hợp lệ của dữ liệu nhập,xóa,update.

### 3.1.6. Xác thực máy chủ cơ sở dữ liệu(Database server validations)



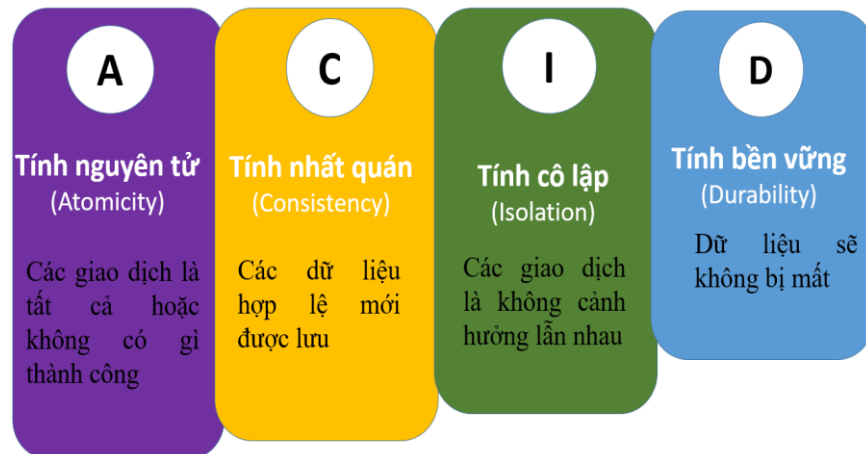
Hình ảnh: Máy chủ cơ sở dữ liệu

- Mục tiêu:
- + Kiểm tra cấu hình máy chủ cơ sở dữ liệu theo yêu cầu nghiệp vụ được chỉ định.
- + Người dùng được ủy quyền chỉ có thể thực hiện các hành động mà họ được ủy quyền.

- + Kiểm tra để đảm bảo rằng máy chủ cơ sở dữ liệu có thể đáp ứng nhu cầu của số lượng giao dịch người dùng tối đa được phép như được chỉ định bởi các đặc tả yêu cầu nghiệp vụ

### 3.2. Kiểm thử chức năng (Function Testing)

- Là một loại kiểm thử cơ sở dữ liệu được sử dụng để xác nhận các yêu cầu chức năng của cơ sở dữ liệu từ quan điểm của người dùng .
- Mục tiêu chính của kiểm thử chức năng là để kiểm tra xem các yêu cầu và hoạt động được thực hiện bởi người dùng có liên quan đến cơ sở dữ liệu có hoạt động như mong đợi hay không.
- Kiểm thử chức năng tập trung vào cách dữ liệu được ánh xạ từ giao diện người dùng đến cơ sở dữ liệu. (Ví dụ: Xác thực việc nhấp vào một nút có ảnh hưởng chính xác đến các bảng / cột liên quan hay không).
- Tất cả các trao đổi dữ liệu phải tuân thủ theo 4 thuộc tính:



Hình ảnh: Mọi trao đổi dữ liệu phải tuân theo 4 thuộc tính này

#### 3.2.1. Tính nguyên tử(Atomicity):

Toàn bộ các giao dịch DB sẽ thành công. Nếu một phần nào đó của giao dịch không thành công, thì toàn bộ giao dịch sẽ được coi là thất bại.

VD: Để mua hàng online, bạn phải điền tất cả các trường cần thiết như dữ liệu thẻ tín dụng và thông tin giao hàng nếu không sẽ nhận được thông báo lỗi.

#### 3.2.2. Tính nhất quán(consistency) :

Cơ sở dữ liệu chỉ nên lưu các giao dịch hợp lệ không vi phạm các quy tắc đối với dữ liệu hợp lệ.

VD: Thông tin vận chuyển của bạn phải là thành phố thực và số thẻ tín dụng của bạn phải có 16 chữ số.

#### 3.2.3. Tính cô lập (Isolation):

Các giao dịch không được ảnh hưởng lẫn nhau. Nếu một số giao dịch được thực hiện đồng thời, kết quả sẽ giống như khi chúng được chạy do đó.

VD:Nếu chỉ còn 1 sản phẩm trong kho. Nếu bạn và bạn của bạn bắt đầu quá trình mua cùng một lúc, người nào hoàn thành đơn hàng trước sẽ nhận được sản phẩm . Người còn lại sẽ không thể hoàn tất giao dịch mua.

### **3.2.4. Tính bền vững**

Sau khi giao dịch được cam kết, những thay đổi mà nó mang lại cho cơ sở dữ liệu sẽ vẫn tồn tại trong hệ thống, không bị ảnh hưởng bởi sự cố hoặc mất điện.

VD:Sau khi bạn mua thành công,nếu bạn bị gặp sự cố mất mạng hoặc mất điện thì dữ liệu của bạn vẫn sẽ tồn tại trong hệ thống mà không bị ảnh hưởng gì cả.

### **3.2.1. Kiểm tra tính toàn vẹn và nhất quán của dữ liệu ( Checking data integrity and consistency)**

- Sau khi bạn Tạo, Đọc, Cập nhật hoặc Xóa dữ liệu trong cơ sở dữ liệu, giao diện người dùng sẽ hiển thị dữ liệu cập nhật trên tất cả các màn hình. Sẽ không có trường hợp bạn có thể xem dữ liệu cập nhật trên một số màn hình trong khi những màn hình khác hiển thị dữ liệu cũ.

- Các trường hợp thử nghiệm nên được viết theo cách để kiểm tra tất cả các vị trí mà dữ liệu có thể được hiển thị và đảm bảo rằng dữ liệu đó giống nhau ở mọi nơi .

- Mục tiêu:

- + Dữ liệu có được tổ chức hợp lý hay không?
- + Dữ liệu được lưu trữ trong các bảng có chính xác và theo yêu cầu nghiệp vụ hay không?
- + Cho dù có bất kỳ dữ liệu không cần thiết nào hiện diện trong ứng dụng đang thử nghiệm không?
- + Dữ liệu đã được lưu trữ theo yêu cầu đối với dữ liệu đã được cập nhật từ giao diện người dùng chưa?
- + Các hoạt động TRIM có được thực hiện trên dữ liệu trước khi chèn dữ liệu vào Cơ sở dữ liệu đang được kiểm tra hay không?
- + Các giao dịch đã được thực hiện theo các yêu cầu kỹ thuật của nghiệp vụ hay chưa và kết quả có đúng hay không?
- + Dữ liệu đã được cam kết đúng hay chưa nếu giao dịch đã được thực hiện thành công?
- + Liệu dữ liệu đã được khôi phục thành công hay chưa nếu giao dịch không được thực hiện thành công bởi người dùng cuối?
- + Liệu dữ liệu đã được sao lưu chưa nếu giao dịch không được thực hiện thành công và nhiều cơ sở dữ liệu không đồng nhất đã tham gia vào giao dịch được đề cập?

- + Liệu tất cả các giao dịch đã được thực hiện bằng cách sử dụng các thủ tục thiết kế bắt buộc như được chỉ định bởi các yêu cầu nghiệp vụ của hệ thống hay chưa?

### **3.2.2. Đăng nhập và bảo mật người dùng (Login and user security)**

Mục tiêu:

- Ứng dụng có cho phép người dùng tiếp tục trong ứng dụng trong trường hợp tên người dùng không hợp lệ nhưng mật khẩu hợp lệ, tên người dùng hợp lệ nhưng mật khẩu không hợp lệ, tên người dùng không hợp lệ và mật khẩu không hợp lệ hay không?
- Liệu người dùng có được phép chỉ thực hiện các hoạt động cụ thể được chỉ định bởi các yêu cầu nghiệp vụ hay không?
- Dữ liệu có được bảo mật khỏi truy cập trái phép hay không?
- Cho dù có các vai trò người dùng khác nhau được tạo với các quyền khác nhau?
- Liệu tất cả người dùng có các cấp độ truy cập cần thiết trên Cơ sở dữ liệu được chỉ định theo yêu cầu của các đặc tả nghiệp vụ hay không?
- Kiểm tra xem dữ liệu quan trọng như mật khẩu, số thẻ tín dụng có được mã hóa và không được lưu trữ dưới dạng văn bản thuần túy trong Cơ sở dữ liệu hay không. Một thực tiễn tốt là đảm bảo tất cả các tài khoản phải có mật khẩu phức tạp và không dễ đoán.

### **3.3. Kiểm thử phi chức năng (Non-Function Testing)**

- Kiểm thử phi chức năng kiểm tra hiệu suất, độ tin cậy, khả năng mở rộng và các khía cạnh khác của hệ thống phần mềm.
- Được thực hiện sau kiểm thử chức năng
- Kiểm thử phi chức năng bao gồm:

#### **3.3.1. Kiểm thử việc tải (Load Testing)**

- Mục đích của kiểm thử việc tải cần được hiểu rõ ràng và được lập thành văn bản. Các loại cấu hình sau đây là bắt buộc phải kiểm thử việc tải.
  - + Các giao dịch của người dùng được sử dụng thường xuyên nhất có khả năng ảnh hưởng đến hiệu suất của tất cả các giao dịch khác nếu chúng không hiệu quả.
  - + Ít nhất một giao dịch người dùng không chỉnh sửa nên được bao gồm trong bộ thử nghiệm cuối cùng, để hiệu suất của các giao dịch đó có thể được phân biệt với các giao dịch khác phức tạp hơn.
  - + Cần đưa vào các giao dịch quan trọng hơn tạo điều kiện cho các mục tiêu cốt lõi của hệ thống, vì theo định nghĩa, sự thất bại của các giao dịch này có tác động lớn nhất.
  - + Ít nhất một giao dịch có thể chỉnh sửa nên được bao gồm để hiệu suất của các giao dịch đó có thể được phân biệt với các giao dịch khác.

- + Thời gian đáp ứng tối ưu với số lượng người dùng ảo khổng lồ cho tất cả các yêu cầu tương lai.
- + Thời gian hiệu quả để tìm nạp các bản ghi khác nhau.
- Các công cụ kiểm thử việc tải quan trọng là load runner, win runner và JMeter.

### **3.3.2. Kiểm thử áp lực (Database stress testing hay torturous testing or fatigue testing)**

- Là một phương pháp kiểm thử được sử dụng để kiểm tra hệ thống cơ sở dữ liệu với tải nặng đến mức nó bị lỗi tại một số điểm. Điều này giúp xác định điểm sự cố của hệ thống cơ sở dữ liệu. Nó đòi hỏi phải có kế hoạch và nỗ lực phù hợp để tránh sử dụng tài nguyên quá mức.
- Các công cụ kiểm tra ứng suất quan trọng là LoadRunner và JMeter.

## **4. Các công cụ kiểm thử back end tốt nhất**

### **4.1. Data Factory:**

Data factory là công cụ kiểm thử cơ sở dữ liệu. Nó hoạt động như trình tạo dữ liệu và quản lý dữ liệu để kiểm thử cơ sở dữ liệu. Nó có giao diện rất dễ sử dụng và có khả năng quản lý mối quan hệ dữ liệu phức tạp.

### **4.2. Data Generator:**

DTM Data Generator là một công cụ kiểm thử Back-end khác. Nó được sử dụng để tạo các hàng dữ liệu và các lược đồ(schema objects) để kiểm thử cơ sở dữ liệu. Công cụ này hỗ trợ khả năng sử dụng tải và kiểm tra hiệu suất trên cơ sở dữ liệu.

### **4.3. TurboData**

Công cụ phần mềm Turbodata có thể được sử dụng để tạo dữ liệu kiểm thử với khóa ngoại. Nó cho phép sử dụng các lệnh Chọn, Cập nhật và Xóa SQL. Nó cũng hỗ trợ nhiều tệp tuần tự và cơ sở dữ liệu quan hệ.

### **4.4. TOAD**

Một trong những công cụ kiểm tra và quản lý DB phổ biến nhất. Nó cho phép bạn kết nối và gửi các truy vấn tới DB của bạn. Công cụ này có phiên bản miễn phí và trả phí, cả hai đều hỗ trợ nhiều loại cơ sở dữ liệu và nền tảng.

### **4.5. php MyAdmin**

Một ứng dụng mã nguồn mở miễn phí để quản lý và thử nghiệm MySQL / MariaDB. Nó đi kèm với một GUI thân thiện với người dùng và có nhiều tài liệu.

## **5. Lợi ích ,vấn đề và giải pháp của kiểm thử Backend**

**5.1. Lợi ích**

- Kiểm soát hoàn toàn độ bao phủ và độ sâu của kiểm thử
  - Trong giai đoạn phát triển ban đầu, nhiều lỗi có thể được tìm thấy một cách hiệu quả.
  - Đảm bảo các giá trị dữ liệu và thông tin nhận được và lưu trữ vào CSDL hợp lệ.
  - Giúp tránh mất mát dữ liệu, lưu dữ liệu giao dịch bị hủy bỏ hoặc không có truy cập trái phép vào thông tin.
  - Phát hiện lỗi sớm hơn khi chi phí sửa chúng ở mức thấp nhất.
  - Có toàn quyền kiểm soát phạm vi kiểm tra và kiểm tra ứng dụng của bạn kỹ lưỡng hơn.
  - Tránh các vấn đề như bế tắc và mất dữ liệu.
  - Cải thiện chức năng ổn định và mạnh mẽ của hệ thống.
  - Tăng chất lượng dữ liệu và tiết lộ các vấn đề về hiệu suất sớm hơn.
- ➔ Để thực hiện kiểm thử backend, người kiểm thử phải có kiến thức nền tảng vững chắc về máy chủ cơ sở dữ liệu và kiến thức về các ngôn ngữ truy vấn có cấu trúc.

**5.2. Vấn đề và giải pháp:**

Vấn đề	Giải pháp
- Chi phí có thể phát sinh để xác định trạng thái của các giao dịch cơ sở dữ liệu.	- Cần có kế hoạch và thời gian cho tiến trình tổng thể để không có vấn đề nào về thời gian và chi phí xuất hiện.
- Dữ liệu kiểm tra mới phải được thiết kế sau khi dọn dẹp dữ liệu kiểm tra cũ.	- Có kế hoạch và phương pháp cho việc tạo dữ liệu kiểm tra.
- Điều kiện tiên quyết được đề cập ở trên đảm bảo rằng việc thiết lập quy trình kiểm tra cơ sở dữ liệu có thể tốn kém cũng như tốn thời gian.	- Cần có sự cân bằng giữa chất lượng và thời gian biểu dự án tổng thể.
- Cần có trình tạo SQL để chuyển đổi trình xác thực SQL nhằm đảm bảo các truy vấn SQL phù hợp để xử lý các trường hợp kiểm tra cơ sở dữ liệu bắt buộc.	- Việc duy trì các truy vấn SQL và cập nhật liên tục của chúng là một phần quan trọng của quá trình kiểm tra tổng thể, nên là một phần của chiến lược kiểm tra tổng thể .

**6. Những quan niệm sai lầm về kiểm thử cơ sở dữ liệu ( Backend testing or database testing)**

Quan niệm sai lầm	Thực tế
- Kiểm thử backend đòi hỏi nhiều kiến thức chuyên môn và đó là một công việc rất tẻ nhạt	- Kiểm thử back-end ngày càng hiệu quả trong Kiểm thử phần mềm cung cấp sự ổn định chức năng lâu dài cho ứng dụng tổng thể, do đó cần phải nỗ lực hết mình.
- Kiểm thử backend bổ sung thêm công việc tắc nghẽn	- Ngược lại, kiểm thử backend sẽ tăng thêm giá trị cho công việc tổng thể bằng cách tìm ra các vấn đề tiềm ẩn và do đó tích cực giúp cải thiện ứng dụng tổng thể.
- Kiểm thử backend làm chậm quá trình phát triển tổng thể	- Lượng kiểm thử back-end đáng kể giúp cải thiện chất lượng tổng thể cho ứng dụng cơ sở dữ liệu.
- Kiểm thử Back-end có thể quá tốn kém	- Bất kỳ chi phí nào cho việc kiểm thử backend đều là một khoản đầu tư dài hạn dẫn đến sự ổn định lâu dài và mạnh mẽ của ứng dụng. Vì vậy, chi phí cho kiểm thử back-end hoặc Kiểm tra SQL là cần thiết.

**7. Kết luận****7.1. Nội dung**

- Tất cả dữ liệu bao gồm siêu dữ liệu cũng như dữ liệu chức năng cần phải được xác thực theo ánh xạ của chúng bằng các tài liệu thông số kỹ thuật được yêu cầu.
- Việc xác minh dữ liệu kiểm thử đã được tạo ra bởi / tham vấn với nhóm phát triển cần phải được xác nhận.
- Xác nhận dữ liệu đầu ra bằng cách sử dụng cả thủ tục tự động cũng như thủ công.
- Triển khai các kỹ thuật khác nhau như kỹ thuật vẽ đồ thị hiệu ứng nguyên nhân, kỹ thuật phân vùng tương đương và kỹ thuật phân tích giá trị biên để tạo ra các điều kiện dữ liệu thử nghiệm bắt buộc.
- Các quy tắc xác nhận tính toàn vẹn tham chiếu cho các bảng cơ sở dữ liệu cũng cần phải được xác nhận hợp lệ.

- Việc lựa chọn các giá trị bằng mặc định để xác nhận tính nhất quán của cơ sở dữ liệu là rất quan trọng. Đó là các event log đã được thêm thành công vào cơ sở dữ liệu cho tất cả các event đăng nhập được yêu cầu
- Các công việc theo lịch trình có thực thi đúng lúc không?
- Thực hiện sao lưu kịp thời Cơ sở dữ liệu.

## **7.2. Tài liệu tham khảo**

Bài làm được tham khảo từ các tài liệu:

- <https://www.guru99.com/data-testing.html>
- <https://www.guru99.com/what-is-backend-testing.html>
- [https://www.mindk.com/blog/backend-testing/?fbclid=IwAR2fzaX\\_CT0eG1VV4TZNg4Nzq2r5-CgzmLlsTlC28PyBF8IwXb19esTYPQ0](https://www.mindk.com/blog/backend-testing/?fbclid=IwAR2fzaX_CT0eG1VV4TZNg4Nzq2r5-CgzmLlsTlC28PyBF8IwXb19esTYPQ0)



## Chương 2: Kiểm thử Frontend (Frontend Testing).

### 1. Kiểm thử giao diện người dùng là gì?

#### 1.1. Khái niệm

- **Kiểm thử giao diện người dùng** là một kỹ thuật kiểm thử trong đó giao diện người dùng đồ họa (GUI), chức năng và khả năng sử dụng của các ứng dụng web hoặc một phần mềm được kiểm tra.

- Kiểm thử GUI là những gì người dùng nhìn thấy. Nếu bạn truy cập vào một trang web, những gì bạn thấy trên trang chủ được gọi là giao diện đồ họa người dùng của trang web. Người dùng sẽ không nhìn thấy mã nguồn, giao diện người dùng chỉ tập trung vào cấu trúc thiết kế, hình ảnh hiển thị ra ngoài có đúng như lập trình mong đợi hay không.

#### 1.2. Mục tiêu

Mục tiêu của kiểm thử giao diện ( Front-end Testing) là kiểm thử các chức năng tổng thể để đảm bảo lớp trình bày của ứng dụng web hoặc phần mềm không có lỗi với các bản cập nhật liên tiếp.

#### 1.3. Ví dụ :

- Nếu bạn nhập tên của mình vào phần mở đầu của đơn đăng ký thành viên, các số sẽ không được chấp nhận.

### ĐĂNG KÝ THÀNH VIÊN

Họ và tên

Bùi Thị Hải 1

\*Họ và tên không được dùng chữ số

Hình

*ảnh: Ví dụ không dùng số trong việc nhập họ tên*

- Một ví dụ khác là kiểm tra hiển thị trên các trình duyệt và các thiết bị khác nhau, nếu người dùng thay đổi kích thước màn hình thì hình ảnh và nội dung cần được co lại hoặc không cắt hoặc không chồng chéo nhau.

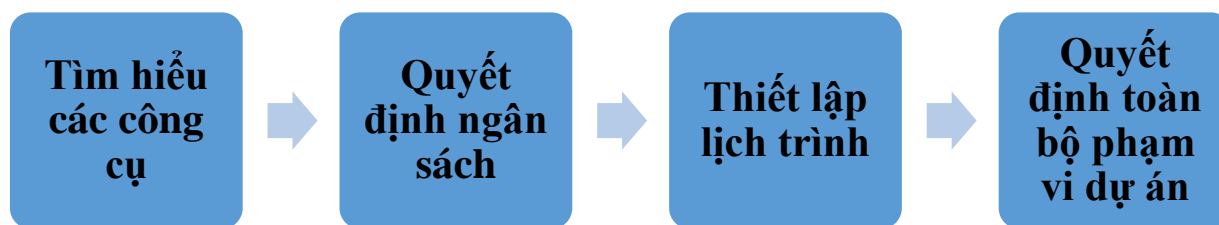


*Hình ảnh: Bố cục trang web trên các thiết bị khác nhau*

- Ngoài ra thử nghiệm Frontend này được tiến hành cho:
  - CSS Regression Testing (Kiểm tra hồi quy CSS): Những thay đổi nhỏ trong CSS làm phá vỡ bố cục giao diện người dùng
  - Các thay đổi đối với tệp JS khiến giao diện người dùng không hoạt động
  - Kiểm thử hiệu suất ( xác định hoặc xác nhận các đặc tính tốc độ, khả năng mở rộng, hoặc tính ổn định của sản phẩm được kiểm thử.)

## 2. Làm thế nào để tạo một kế hoạch kiểm tra giao diện người dùng cho trang web?

- Kế hoạch kiểm tra giao diện người dùng là một tài liệu chính thức liên quan đến việc kiểm tra ứng dụng web của bạn. Điều này bao gồm phạm vi của một ứng dụng web, cách tiếp cận thử nghiệm, các hoạt động, v.v. Đây là bước rất quan trọng và là bước đầu tiên trong quá trình thử nghiệm ứng dụng web của bạn.
- Quy trình: Gồm 4 bước



- *Cụ thể:*

**Bước 1:** Tìm hiểu các công cụ để quản lý kế hoạch kiểm thử của bạn.

( Một vài công cụ phổ quản lý kế hoạch kiểm thử phổ biến hiện nay: Selenium, LambdaTest , Jasmine, Needle)

**Bước 2:** Quyết định ngân sách cho kiểm thử giao diện người dùng.

**Bước 3:** Thiết lập lịch trình cho toàn bộ quá trình.

**Bước 4:** Quyết định toàn bộ phạm vi của dự án.

+ Phạm vi bao gồm các mục sau:

- Hệ điều hành và trình duyệt được sử dụng bởi người dùng ISP gói đối tượng của bạn
- Các thiết bị phổ biến được người dùng sử dụng
- Khả năng của người sử dụng của bạn
- Tốc độ điều chỉnh Internet của người sử dụng

### 3. Tại sao phải tạo kế hoạch kiểm thử giao diện người dùng ?

#### 3.1. Kế hoạch Kiểm tra giao diện người dùng giúp bạn xác định

- Các trình duyệt

+ Trình duyệt web (web browser) là một chương trình phần mềm để trình bày và khám phá nội dung trên World Wide Web. Những phần nội dung này, bao gồm hình ảnh, video và trang web, được kết nối bằng cách sử dụng các siêu liên kết và được phân loại bằng URI (Số nhận dạng tài nguyên đồng nhất).

+ Trình duyệt web phổ biến hiện nay: Google Chrome, Microsoft Edge, Microsoft Internet Explorer, Mozilla Firefox, Cốc Cốc Trình, Opera , Apple Safari,...

+ Nguồn tài liệu: <https://wikimaytinhtinh.com/trinh-duyet-web-la-gi-cac-loai-trinh-duyet-web.html> )



*Hình ảnh: Các trình duyệt phổ biến hiện nay*

- Các hệ điều hành.

+ Hệ điều hành là phần mềm hệ thống giúp tạo ra môi trường làm việc thân thiện giữa người sử dụng và thiết bị công nghệ. Hiện nay có rất nhiều hệ điều hành khác nhau được sử dụng.

+ Hệ điều hành phổ biến hiện nay: Android , iOS, Windows Phone, Blackberry OS, Firefox OS, Windows, Mac OS, Linux, Ubuntu, Chrome OS...



*Hình ảnh: Các hệ điều hành phổ biến hiện nay*

+ Nguồn tài liệu: <https://meohaycuocsong.com/mot-so-he-dieu-hanh-va-lich-su-cac-he-dieu-hanh>

➔ Dự án của bạn cần phải bao gồm. Có vô số sự kết hợp giữa Trình duyệt và Hệ điều hành mà bạn có thể thử nghiệm giao diện người dùng của mình. Có một kế hoạch sẽ giúp bạn giảm bớt công sức kiểm tra và tiền bạc.

### **3.2. Bằng cách tạo kiểm thử giao diện người dùng, kế hoạch của bạn sẽ nhận được những lợi thế sau:**

- Nó giúp bạn có được sự rõ ràng một cách đầy đủ về phạm vi của dự án
- Thực hiện kiểm thử giao diện người dùng cũng mang lại niềm tin trong việc triển khai dự án.

### **4. Những mẹo cho việc kiểm thử giao diện người dùng tốt hơn**

- Chuẩn bị ngân sách, nguồn lực và thời gian của bạn một cách thận trọng.
- Sử dụng trình duyệt không có giao diện người dùng đồ họa để kiểm thử được thực hiện nhanh hơn.
- Cắt giảm số lượng hiển thị (mô hình) DOM trong các ca kiểm thử để thực thi nhanh hơn.
- Cô lập các trường hợp kiểm thử, do đó, nguyên nhân từ gốc của lỗi được xác định nhanh chóng để có chu kỳ sửa chữa lỗi nhanh hơn.
- Sử dụng lại các tập lệnh kiểm thử của bạn để có chu kỳ hồi quy nhanh hơn.
- Bạn nên sử dụng quy ước đặt tên nhất quán cho các tập lệnh kiểm thử của mình.

### **5. Các công cụ kiểm thử giao diện người dùng**

#### **5.1. Công cụ kiểm thử đa trình duyệt (Cross-Browser Testing Tool):**

- Kiểm thử trình duyệt chéo là kiểm tra trang web hoặc ứng dụng của bạn trong nhiều trình duyệt, nhiều thiết bị khác nhau và đảm bảo rằng nó hoạt động nhất quán và chính xác mà không có bất kỳ vấn đề nào xảy ra.
- Bây giờ, có hai phương pháp thực hiện kiểm thử chéo.
  - + Manual testing: phương pháp kiểm tra theo cách thủ công trên mọi trình duyệt.
  - + Kiểm thử thông qua các công cụ trực tuyến ( VD: LambdaTest )

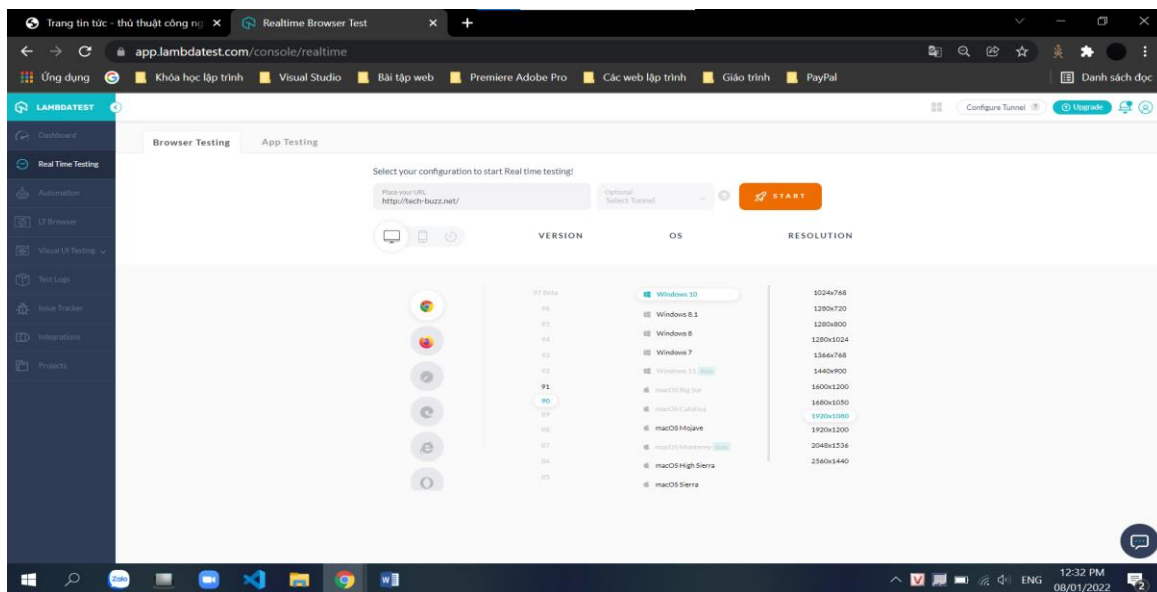
##### **5.1.1. LambdaTest ( <https://app.lambdatest.com/> )**

- **LambdaTest** là một công cụ trực tuyến được sử dụng để thực hiện kiểm thử trình duyệt tương tác trực tiếp trên các trang web và ứng dụng web được publish hoặc tại local của bạn trên 2000 trình duyệt di động và máy tính để bàn thực sự chạy trên hệ điều hành thực. Gỡ lỗi trong thời gian thực trên cả trình duyệt máy tính để bàn và thiết bị di động với các công cụ gỡ lỗi tích hợp.

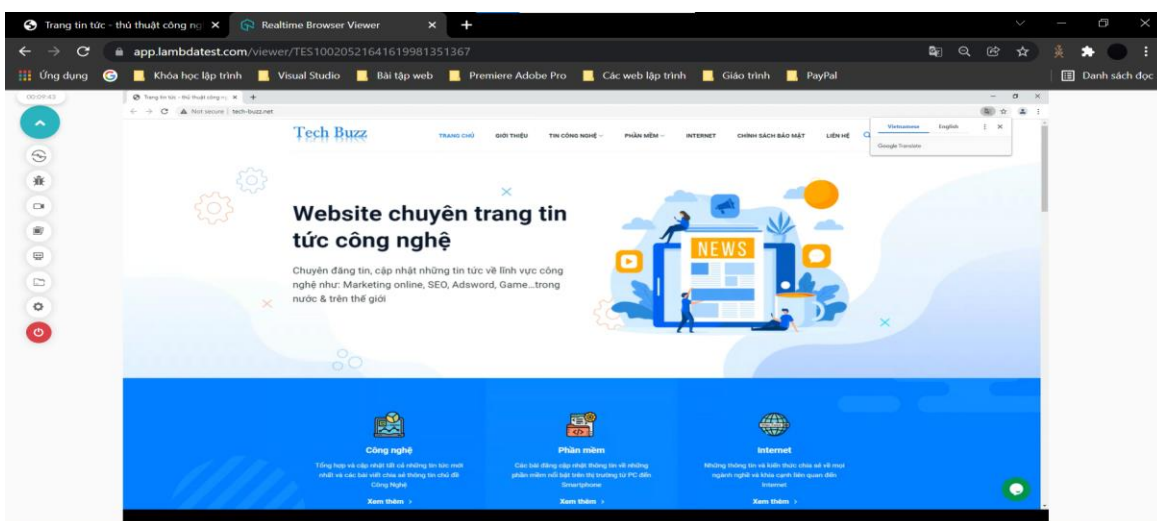
- LambdaTest cũng cung cấp một vài các tính năng bao gồm:

- + Kiểm thử thời gian thực ( có ví dụ)
- + Chụp ảnh màn hình tự động của các bài kiểm tra. ( có ví dụ)
- + Kiểm thử khả năng đáp ứng (Responsiveness test) ( có ví dụ)
- + Sử dụng Tunnel để test các trang local

- + Kiểm thử tự động hóa Selenium.
- + Kiểm thử hồi quy (Regression testing)
- + Test độ phân giải màn hình
- + Kiểm thử theo vị trí (GeoLocation test)
- Ví dụ kiểm thử cho trang web của nhóm: <http://tech-buzz.net/>
- + Ví dụ 1: Kiểm thử thời gian thực ( Real Time Testing)
- Màn hình chọn các thông số để thực hiện kiểm thử:

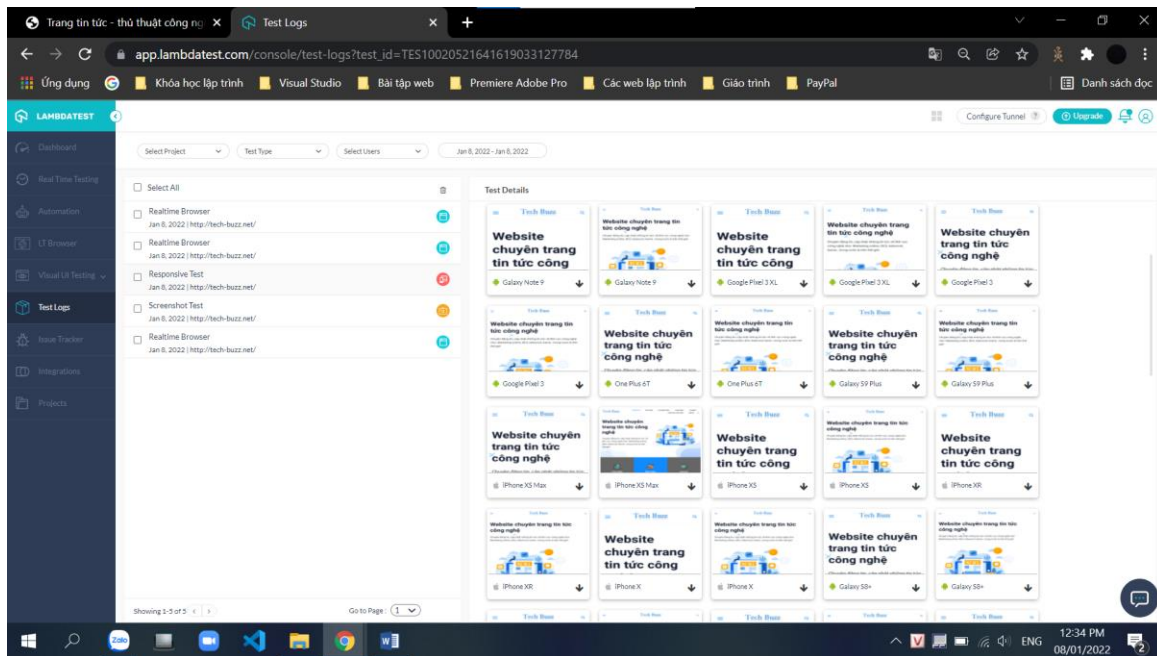


Kết quả màn hình sau khi thực hiện chạy Real Time Testing:

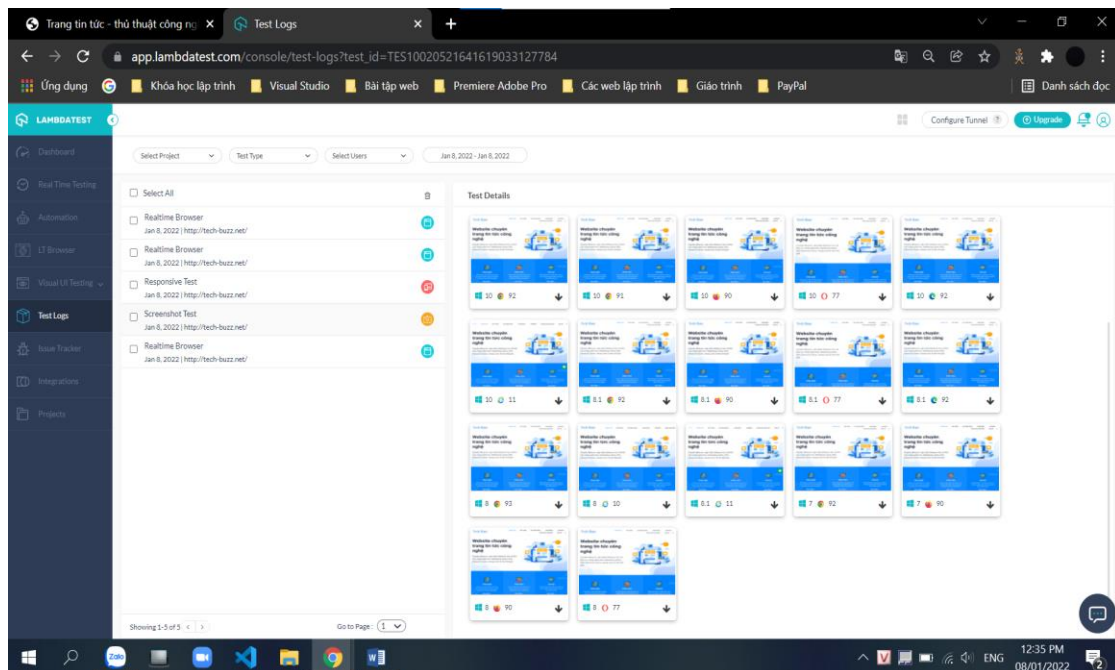




## VD2: Responsive Test



## VD3: Screen Test



- Nguồn tài liệu: <https://viblo.asia/p/cross-browser-testing-voi-lambdatest-bJzKmzbXZ9N>

**5.2. Công cụ kiểm thử JavaScript ( JS testing tool):**

- Kiểm thử JS là một phương pháp kiểm thử trong đó mã nguồn JavaScript viết cho một trang web hoặc mô-đun ứng dụng web được kết hợp với HTML sẽ được kiểm thử như một trình xử lý sự kiện và được thực thi trong trình duyệt để kiểm tra xem tất cả các chức năng có hoạt động tốt hay không.

**5.2.1. Jasmine**

- Nó là một khung phát triển theo hướng hành vi để kiểm tra mã JavaScript. Công cụ này tập trung nhiều hơn vào giá trị kinh doanh hơn là các chi tiết kỹ thuật. Nó có một cú pháp rõ ràng giúp bạn viết các bài kiểm tra một cách dễ dàng. Nó không phụ thuộc vào bất kỳ khung JavaScript nào khác, không yêu cầu DOM. Nó bị ảnh hưởng nhiều bởi các khuôn khổ kiểm thử đơn vị, chẳng hạn như JSpec, ScrewUnit, JSpec và RSpec.

- Jasmine là một công cụ mã nguồn mở rất tiện dụng & dễ dùng để xây dựng kịch bản kiểm thử đơn vị (Unit test) với Mã nguồn JavaScript:

+ Jasmine dựa trên nền BDD (behavior driven development) framework, xây dựng kịch bản kiểm thử đơn vị theo mô phỏng hướng hành vi nên trực quan trong việc xây dựng kịch bản kiểm thử.

+ Jasmine cung cấp môi trường độc lập để kiểm thử và gọi chạy JavaScript.

- Nguồn tài liệu: <https://truonganhhoang.gitbooks.io/van-de-hien-dai-cai-hoc-2017/content/js-testing.html>

**5.3. Công cụ kiểm thử chức năng (Functional testing tool):**

- Kiểm thử chức năng là một loại kiểm thử hộp đen (black box) và ca kiểm thử của nó được dựa trên đặc tả của ứng dụng phần mềm/thành phần đang kiểm thử. Các chức năng được kiểm tra bằng cách nhập vào các giá trị nhập và kiểm tra kết quả đầu ra, và ít quan tâm đến cấu trúc bên trong của ứng dụng.
- Nó là một quy trình cố gắng tìm ra các khác biệt giữa đặc tả bên ngoài của phần mềm và thực tế mà phần mềm cung cấp. Với các đặc tả bên ngoài của phần mềm là đặc tả chính xác về hành vi của phần mềm theo góc nhìn của người dùng.
- Nguồn tài liệu: <https://viblo.asia/p/tim-hieu-ve-kiem-thu-chuc-nang-functionality-testing-bJzKmLVB59N>

**5.3.1. Selenium**

- Selenium là một công cụ kiểm tra giao diện người dùng. Nó bao gồm một loạt các công cụ và thư viện cho phép và hỗ trợ hóa các trình duyệt web. Cung cấp các tiện ích mở rộng để mô phỏng tương tác (nhấp – click, cuộn – scroll, đóng tab, click check box, ...) của người dùng với trình duyệt.



- Nó thực hiện thử nghiệm từ đầu đến cuối trên nhiều trình duyệt và nền tảng khác nhau như Windows, Mac và Linux. Nó cho phép bạn viết các bài kiểm thử bằng nhiều ngôn ngữ lập trình khác nhau như Java, JavaScript, Python, PHP, C #, ...
- Là công cụ hoàn toàn miễn phí, dễ hiểu, dễ sử dụng.
- Công cụ này cung cấp các tính năng ghi và phát lại để viết các bài kiểm tra mà không cần phải học Selenium IDE.
- Nguồn tài liệu: <https://topdev.vn/blog/selenium-la-gi-mot-so-kinh-nghiem-lam-viec-voi-selenium/>

#### 5.4. Công cụ kiểm thử CSS (CSS tool):

- Kiểm thử CSS là 1 phương pháp kiểm thử các hình ảnh của trang web hiển thị, nội dung hiển thị giống với màn hình tốt nhất.

##### 5.4.1. Needle

- Needle là một công cụ để kiểm tra CSS và hình ảnh của trang web với Selenium và nose.
- Nó kiểm tra xem các yếu tố hình ảnh (phông chữ / CSS / hình ảnh / VSG...) có hiển thị chính xác hay không bằng cách chụp ảnh màn hình các phần nhất định trên trang web của bạn. Sau đó, công cụ sẽ so sánh với một số ảnh chụp màn hình tốt đã biết.
- Nó cũng cung cấp các công cụ để kiểm tra các giá trị CSS được tính toán và vị trí của các phần tử HTML.
- Một trang web kiểm tra mã nguồn CSS khá hữu ích: <https://jigsaw.w3.org/css-validator/>

➔ Kết quả sau kiểm tra:

**Xin chúc mừng! Không tìm thấy lỗi nào.**

Tài liệu này hợp lệ theo khuyến nghị của [CSS cấp 3 + SVG](#) !

- Nguồn tài liệu: <https://needle.readthedocs.io/en/latest/>

## 6. Tối ưu hóa hiệu suất giao diện ( Front-End Performance Optimization)

- Kiểm tra hiệu suất giao diện người dùng sẽ kiểm tra “Trang tải nhanh như thế nào”.



*Hình ảnh: Các mức của tốc độ tải trang*

- Tối ưu hóa hiệu suất front-end cho một người dùng là một phương pháp hay trước khi thử nghiệm một ứng dụng có tần suất và lượt người dùng tải cao.
  - + Kiểm tra thời gian phản hồi của website với tốc độ kết nối khác nhau.
  - + Kiểm tra website có xử lý được nhiều yêu cầu người dùng vào cùng một thời điểm.
  - + Kiểm tra website có hoạt động tốt trong thời điểm lượt tải cao.
  - + Kiểm tra dữ liệu đầu vào lớn từ người dùng.
  - + Kiểm tra hành vi của website khi kết nối với cơ sở dữ liệu.
  - + Kiểm tra các phương pháp tối ưu hóa như giảm thời gian tải bằng cách bật bộ nhớ cache trên trình duyệt và phía máy chủ, nén gzip...
- Nguồn tài liệu: <https://mona.media/tim-hieu-ve-quy-trinh-kiem-thu-website/>

## 7. Tại sao tối ưu hóa hiệu suất Front-End lại quan trọng?

- Tối ưu hóa hiệu suất trước đó có nghĩa là tối ưu hóa phía máy chủ. Đó là bởi vì hầu hết các trang web chủ yếu là tĩnh và hầu hết quá trình xử lý được thực hiện ở phía máy chủ.
- Tuy nhiên, với sự ra đời của công nghệ Web 2.0, các ứng dụng web trở nên năng động hơn. Do đó, mã phía máy khách đã trở thành một yếu tố hiệu suất.

## 8. Lợi ích của tối ưu hóa hiệu suất Front-End là gì?

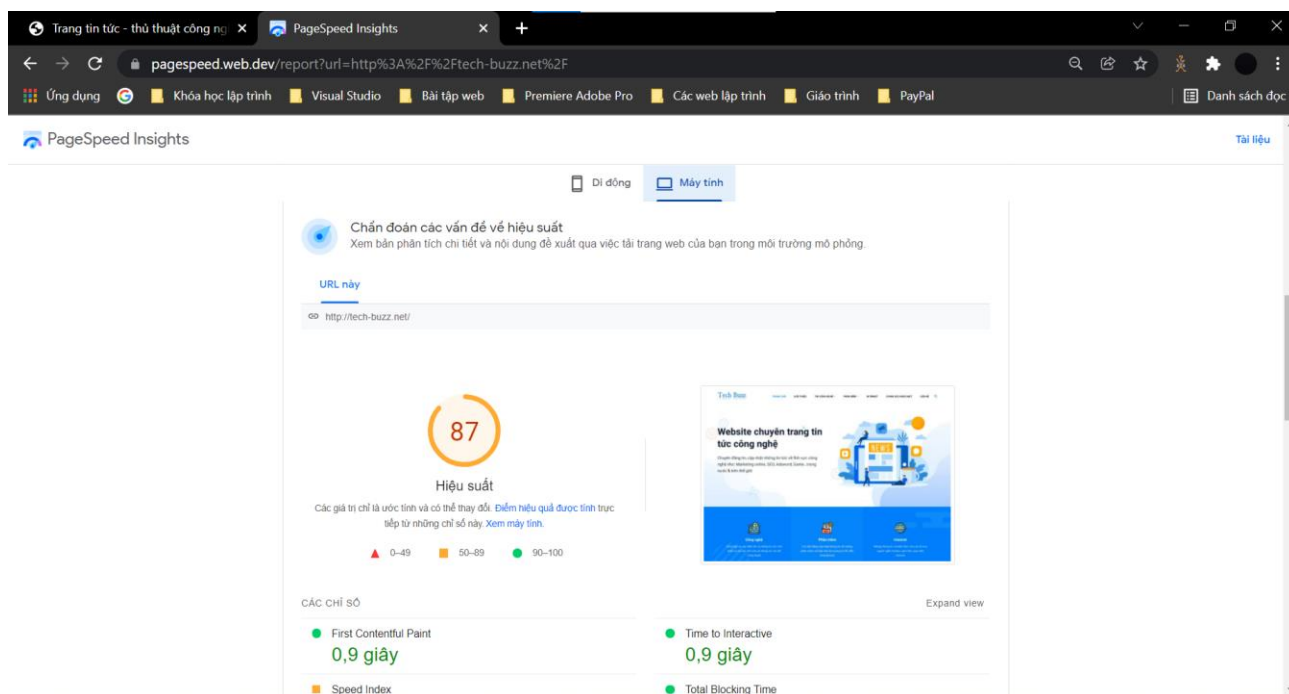
- Trong thử nghiệm trang web, ngoài sự tắc nghẽn của máy chủ, việc tìm kiếm các vấn đề về hiệu suất phía máy khách cũng quan trọng không kém vì chúng dễ dàng ảnh hưởng đến trải nghiệm của người dùng.

- Cải thiện 50% hiệu suất back-end sẽ tăng 10% hiệu suất tổng thể của ứng dụng.
- Tuy nhiên, việc cải thiện 50% hiệu suất front-end sẽ tăng 40% hiệu suất tổng thể của ứng dụng.
- Hơn nữa, việc tối ưu hóa hiệu suất của front-end dễ dàng và tiết kiệm chi phí hơn so với back-end.

## 9. Công cụ kiểm tra hiệu suất giao diện người dùng

### 9.1. Tốc độ tải trang ( Page Speed)

- Tốc độ tải trang (Pagespeed) là khoảng thời gian các nội dung trên trang được hiển thị hoàn toàn từ khi người dùng truy cập. Bạn có thể đo lường tốc độ tải trang qua công cụ **Pagespeed Insights** của Google.
- Trang web: <https://pagespeed.web.dev/>  
Tốc độ tải trang không chỉ ảnh hưởng trực tiếp tới trải nghiệm của người dùng mà còn tác động tới quá trình thu thập dữ liệu của Google.
- Page Speed là một tiện ích bổ sung kiểm tra hiệu suất nguồn mở do Google đưa ra. Công cụ đánh giá trang web và cung cấp các đề xuất để giảm thiểu thời gian tải. Nó giúp cho việc truy xuất trang web nhanh hơn khi người dùng truy cập các trang web bằng công cụ tìm kiếm của Google.
- VD Kiểm thử trang web: <http://tech-buzz.net/>



- Nguồn tài liệu: <https://www.toponseek.com/blogs/pagespeed-toc-do-tai-trang/>

## **9.2. Yslow**

- **YSlow** là một công cụ nguồn mở kiểm tra hiệu suất giao diện người dùng của trang web, phân tích trang web của bạn và cung cấp cho bạn các đề xuất để cải thiện hiệu suất của nó.
- Nó phân tích hiệu suất trang web bằng cách thu thập dữ liệu DOM để tìm tất cả các thành phần (hình ảnh, tập lệnh, bảng định kiểu, v.v.) trong trang, kiểm tra tất cả các thành phần trên trang, bao gồm cả các thành phần được tạo bằng cách sử dụng JavaScript, lấy tất cả dữ liệu này về và tạo điểm cho từng quy tắc, tạo ra điểm tổng thể.
- Bạn vào trang web mà bạn muốn kiểm tra, sau đó YSlow sẽ phân tích nó theo một bộ quy tắc và cung cấp cho bạn cả điểm số và danh sách những gì trang web của bạn hoạt động tốt khi hoạt động hiệu quả và hoạt động kém.
- Nó cũng đo lường hiệu suất của trang và đưa ra các đề xuất cho người dùng.
- Nguồn tài liệu: <http://yslow.org/> và <https://qnet88.com/yslow-la-gi-va-lam-the-nao-de-toi-uu-no/>

## **10. Kết luận**

### **10.1. Nội dung**

- Kiểm thử Front-end là kiểm thử hoặc xác minh các chức năng giao diện người dùng, GUI và khả năng sử dụng.
- Mục đích chính của kiểm thử Front-end để đảm bảo rằng mọi người dùng đều được bảo vệ tốt khỏi các lỗi.
- Tạo một kế hoạch kiểm thử đầu vào giúp bạn biết các thiết bị, trình duyệt và hệ thống mà dự án của bạn cần.
- Nó cũng giúp bạn có được sự rõ ràng hoàn chỉnh về phạm vi của dự án.
- Jasmine, Selenium, Browser, TestComplete, Needle là một số ví dụ về công cụ kiểm thử Front-end.

### **10.2. Tài liệu tham khảo**

Bài làm được tham khảo từ các tài liệu:

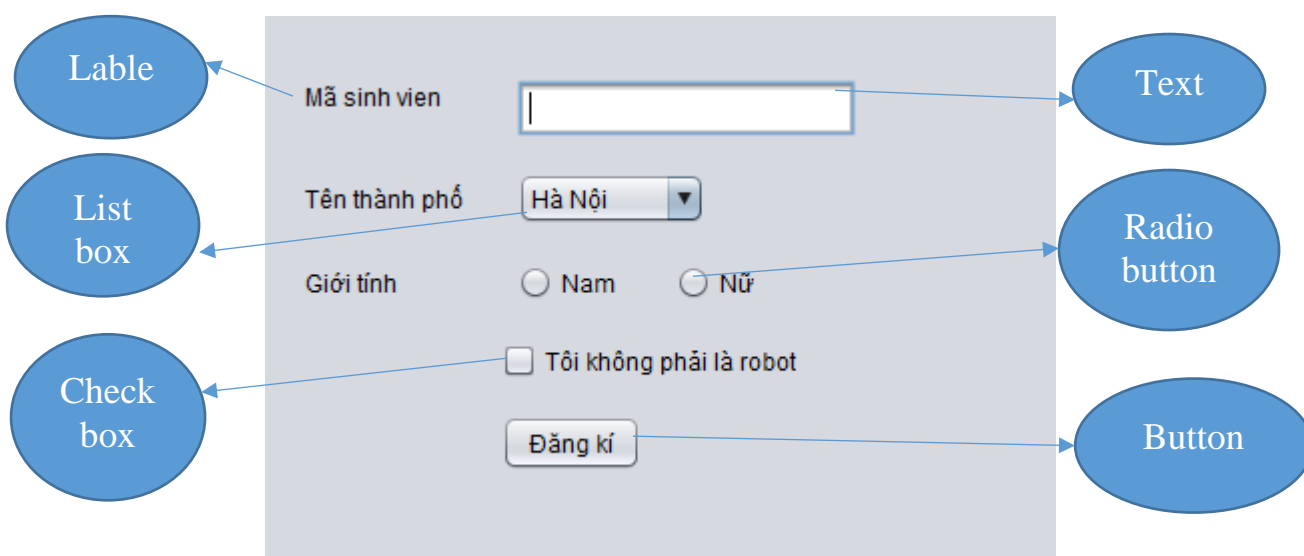
- <https://www.guru99.com/frontend-testing.html>
- <https://wikimaytinh.com/trinh-duyet-web-la-gi-cac-loai-trinh-duyet-web.html>
- <https://meohaycuocsong.com/mot-so-he-dieu-hanh-va-lich-su-cac-he-dieu-hanh.html>
- <https://viblo.asia/p/cross-browser-testing-voi-lambda-test-bJzKmzbXZ9N>
- <https://topdev.vn/blog/selenium-la-gi-mot-so-kinh-nghiem-lam-viec-voi-selenium/>
- <https://mona.media/tim-hieu-ve-quy-trinh-kiem-thu-website/>

- <https://viblo.asia/p/tim-hieu-ve-kiem-thu-chuc-nang-functionality-testing-bJzKmLVB59N>
- <https://www.toonseek.com/blogs/pagespeed-toc-do-tai-trang/>
- <http://yslow.org/> và <https://qnet88.com/yslow-la-gi-va-lam-the-nao-de-toi-uu-no/>

## Chương 3: Kiểm thử giao diện đồ họa người dùng (GUI Testing)

### 1. GUI là gì?

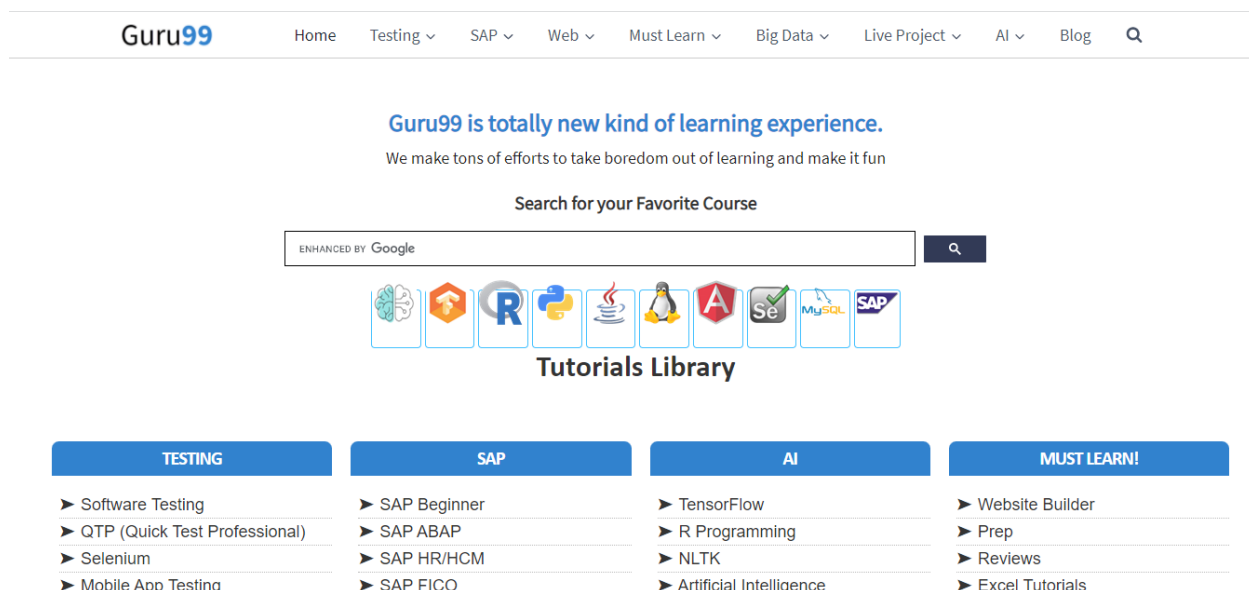
- GUI là viết tắt của từ Graphical User Interface (Giao diện đồ họa người dùng)
- Có hai loại giao diện người dùng cho các ứng dụng trên máy tính
  - + **Giao diện dòng lệnh** (Command line interface) là người dùng có thể ra mệnh lệnh cho máy tính thông qua các tập câu lệnh.
  - + **Giao diện người dùng đồ họa** (Graphical User Interface) là người dùng tương tác với máy tính thông qua bằng hình ảnh và chữ viết thay. Một số thành phần trong tương tác giữa người dùng với các ứng dụng.



*Hình ảnh: Một số thành phần tử của GUI*

### 2. Kiểm thử giao diện đồ họa người dùng là gì? (GUI Testing)

- Kiểm thử giao diện đồ họa người dùng là một loại kiểm thử phần mềm nhằm kiểm tra giao diện đồ họa người dùng (GUI) của phần mềm. Mục đích của kiểm thử giao diện đồ họa người dùng là để đảm bảo các chức năng của ứng dụng phần mềm hoạt động theo thông số kỹ thuật bằng cách kiểm tra màn hình và các điều khiển như menu, button, icons, text, v.v
- Giao diện đồ họa người dùng (GUI) là những gì người dùng nhìn thấy. Giả sử nếu bạn truy cập [guru99.com](http://guru99.com) những gì bạn nhìn thấy chính là GUI của trang web. Người dùng không nhìn thấy mã nguồn.



Hình ảnh: Giao diện đồ họa của trang web guru99.com

- Trong hình ảnh trên, nếu chúng ta phải thực hiện kiểm thử giao diện đồ họa người dùng, đầu tiên chúng ta cần kiểm tra rằng các hình ảnh có hiển thị đúng trên các trình duyệt khác nhau. Ngoài ra, các button hay hyperlink trên màn hình nên hoạt động đúng khi được người dùng click chuột vào. Hay người dùng thay đổi kích thước màn hình thì các hình ảnh và nội dung không nên bị cắt mất, co lại hay hiển thị chồng chéo lên nhau. Kiểm thử giao diện đồ họa người dùng được hiểu một cách đơn giản là kiểm tra tính hợp lệ của các đối tượng/thành phần trên màn hình.

### 3. Sự cần thiết của kiểm thử giao diện đồ họa người dùng?

- Tại sao phải kiểm thử giao diện đồ họa người dùng?
- Nó có thực sự cần thiết không?
- Kiểm tra chức năng và logic của ứng dụng là chưa đủ à? Vậy thì tại sao phải lãng phí thời gian vào kiểm tra giao diện người dùng.
- Để trả lời câu hỏi, hãy suy nghĩ như một người dùng chứ không phải người kiểm thử. Người dùng không có kiến thức chuyên sâu về phần mềm / ứng dụng. Cái mà họ quan tâm đầu tiên sẽ là giao diện của ứng dụng hay sản phẩm. Nếu người dùng không thoải mái với giao diện hoặc nhận thấy ứng dụng thật khó để sử dụng tất nhiên anh ta sẽ không có lý do để tiếp tục sử dụng ứng dụng đó nữa.
- Đó là lý do tại sao giao diện đồ họa người dùng là một vấn đề đáng lo ngại và cần phải thực hiện kiểm tra thích hợp để đảm bảo rằng giao diện không có lỗi.

### 4. Cần làm gì trong kiểm thử giao diện đồ họa người dùng (GUI Testing)?

Dưới đây là những công việc cụ thể cần thực hiện khi kiểm thử giao diện người dùng.

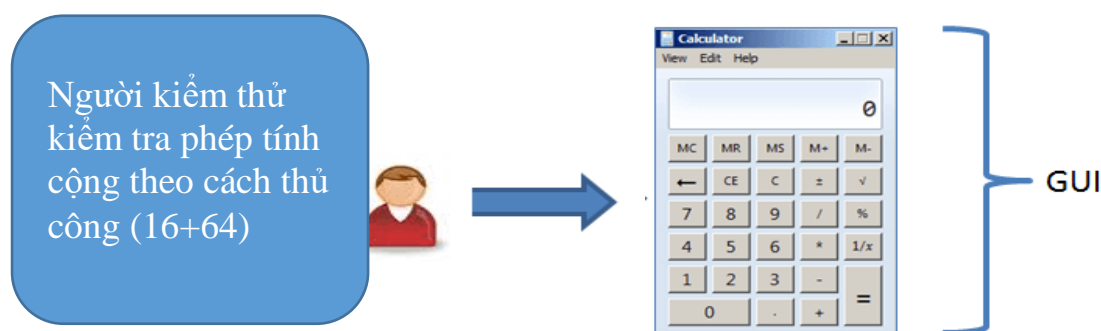
- + Kiểm tra tất cả các thành phần của GUI về kích thước, vị trí, chiều rộng, chiều dài và việc chấp nhận các ký tự hoặc số.
- + Kiểm tra xem bạn có thể thực thi chức năng dự kiến của ứng dụng bằng GUI không.
- + Kiểm tra thông báo lỗi có hiển thị chính xác không.
- + Kiểm tra ranh giới rõ ràng của các phần khác nhau trên màn hình.
- + Kiểm tra phông chữ được sử dụng trong ứng dụng có thể đọc được không.
- + Kiểm tra căn chỉnh của văn bản có phù hợp không.
- + Kiểm tra màu sắc của phông chữ, các thông báo cảnh báo có đẹp mắt không.
- + Kiểm tra để đảm bảo rằng hình ảnh có độ rõ nét tốt.
- + Kiểm tra xem hình ảnh có được căn chỉnh đúng cách không.
- + Kiểm tra vị trí của các phần tử GUI cho độ phân giải màn hình khác nhau.

## 5. Các kỹ thuật kiểm thử giao diện đồ họa người dùng:

Có 3 kỹ thuật trong kiểm thử giao diện đồ họa người dùng.

### 5.1. Kiểm tra bằng tay (Manual Based Testing)

Theo cách tiếp cận này, các đối tượng GUI (giao diện đồ họa người dùng) trên màn hình được kiểm tra bằng tay bởi tester đảm bảo đúng với các yêu cầu nêu trong tài liệu yêu cầu dự án hoặc file thiết kế giao diện đính kèm.



Hình ảnh: Kỹ thuật kiểm tra bằng tay (Manual Based Testing)

### 5.2. Ghi lại và phát triển (Record and Replay)

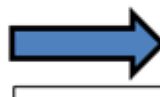
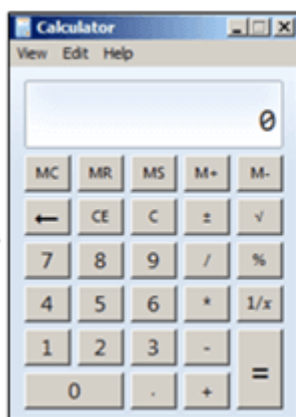
Kiểm tra các đối tượng GUI có thể được thực hiện bằng các công cụ tự động hóa. Cách này gồm 2 phần là quá trình ghi lại (Record) các bước bởi một tool tự động và phát lại chúng (Replay) để kiểm tra cho những lần test sau.



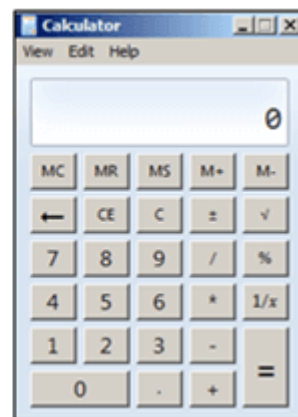
Người kiểm thử  
kiểm tra phép tính  
cộng theo cách thủ  
công (16+64)



Ghi lại

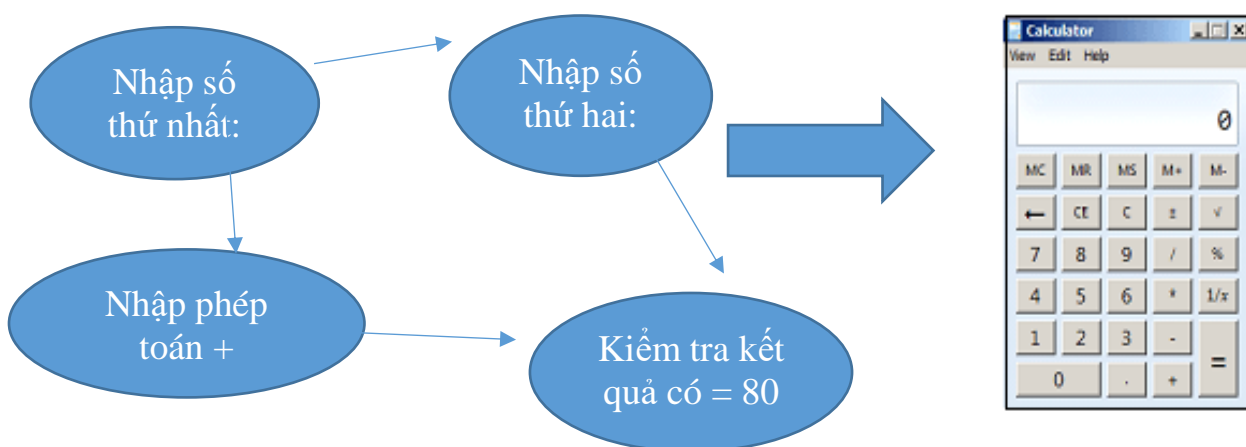


Phát



Hình ảnh: Kỹ thuật ghi lại và phát triển (Record and Replay)

### 5.3. Kiểm thử dựa trên mô hình (Model Based Testing)



Hình ảnh: Kiểm thử dựa trên mô hình hóa (Model Based Testing)

- Mô hình là một mô tả đồ họa về hành vi của hệ thống. Nó giúp chúng ta hiểu và dự đoán các hành vi của hệ thống. Các mô hình giúp tạo ra các ca kiểm thử hiệu quả bằng cách sử dụng các yêu cầu hệ thống.
- Những vấn đề sau đây cần được xem xét đối với kiểm thử dựa trên mô hình:
  - + Xây dựng mô hình
  - + Xác định đầu vào cho mô hình
  - + Tính kết quả dự kiến cho mô hình
  - + Chạy các bài kiểm tra

- + So sánh kết quả thực với đầu ra dự kiến
- + Quyết định về hành động tiếp theo trên mô hình
- Một số kỹ thuật mô hình hóa từ đó các ca kiểm thử có thể được rút ra:
  - + Biểu đồ - Mô tả trạng thái của hệ thống và kiểm tra trạng thái sau một số đầu vào.
  - + Bảng Quyết định - Bảng được sử dụng để xác định kết quả cho từng đầu vào được áp dụng.
- Kiểm thử dựa trên mô hình là một kỹ thuật phát triển để tạo ra các ca kiểm thử từ các yêu cầu. So với hai phương pháp trên, lợi thế chính của nó là: có thể xác định các trạng thái không mong muốn mà bạn có thể mắc phải.
- Bảng dưới đây là một số công cụ mã nguồn mở dùng để kiểm tra tự động giao diện người dùng.

Sản phẩm	URL
AutoHotkey	<a href="https://www.autohotkey.com/">https://www.autohotkey.com/</a>
Selenium	<a href="https://www.selenium.dev/">https://www.selenium.dev/</a>
Sikuli	<a href="http://sikuli.org">http://sikuli.org</a>
Watir	<a href="http://www.watir.com/">http://www.watir.com/</a>
Dojo Toolkit	<a href="http://dojotoolkit.org/">http://dojotoolkit.org/</a>

## 6. Các mẫu ca kiểm thử trong kiểm thử giao diện đồ họa người dùng (GUI testing).

- Kiểm thử giao diện đồ họa người dùng về cơ bản bao gồm:
  - + Kiểm tra kích thước, vị trí, chiều rộng, chiều cao của các phần tử
  - + Kiểm tra các thông báo lỗi đang được hiển thị.
  - + Kiểm tra các phần khác nhau của màn hình.
  - + Kiểm tra phông chữ xem nó có thể đọc được hay không.
  - + Thử nghiệm màn hình ở các độ phân giải khác nhau với sự trợ giúp của phóng to và thu nhỏ như 640 x 480, 600 x 800, v.v.
  - + Kiểm tra sự căn chỉnh của văn bản và các yếu tố khác như biểu tượng, nút, v.v. có ở đúng vị trí hay không.
  - + Kiểm tra màu sắc của phông chữ.
  - + Kiểm tra màu sắc của các thông báo lỗi, thông báo cảnh báo.
  - + Kiểm tra xem hình ảnh có độ rõ nét tốt hay không.

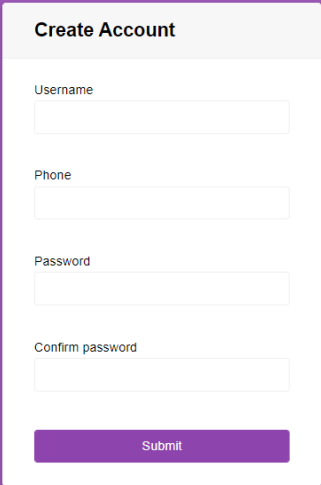
- + Kiểm tra sự liên kết của các hình ảnh.
- + Kiểm tra chính tả.
- + Người dùng có bức tức khi sử dụng giao diện hệ thống.
- + Kiểm tra xem giao diện có hấp dẫn hay không.
- + Kiểm tra các thanh cuộn theo kích thước của trang nếu có.
- + Kiểm tra các trường bị vô hiệu hóa nếu có.
- + Kiểm tra kích thước của màn ảnh.
- + Kiểm tra các tiêu đề xem của nó có được căn chỉnh đúng không.
- + Kiểm tra màu sắc của các siêu liên kết.

### 7. Những thách thức trong kiểm thử giao diện đồ họa người dùng.

- Trong kỹ thuật phần mềm, vấn đề phổ biến nhất khi thực hiện **kiểm thử hồi quy** chính là giao diện đồ họa người dùng của ứng dụng thường xuyên thay đổi.
- Rất khó để kiểm tra và xác định xem đó là một vấn đề hay một nâng cấp. Sự cố xuất hiện khi bạn không có bất kỳ tài liệu nào liên quan đến các thay đổi của GUI (giao diện đồ họa người dùng).

### 8. Demo: Cách thực hiện kiểm thử giao diện đồ họa người dùng.

Chúng ta sẽ có một số ca kiểm thử cho màn hình sau đây:

A screenshot of a 'Create Account' form. The form is white with a light gray header containing the title 'Create Account'. It has four input fields: 'Username', 'Phone', 'Password', and 'Confirm password'. Each field has a small label above it. At the bottom of the form is a purple 'Submit' button. The form is centered on a solid purple background.

*Hình ảnh: Giao diện đồ họa người dùng cho chức năng tạo tài khoản*

- + TC 01- Xác minh hộp văn bản có nhãn “**User name**” được căn chỉnh đúng cách.
- + TC 02 - Xác minh hộp văn bản có nhãn “**Phone**” được căn chỉnh đúng cách.
- + TC 03 - Xác minh hộp văn bản có nhãn “**Password**” được căn chỉnh đúng cách.
- + TC 04 - Xác minh hộp văn bản có nhãn “**Confirm password**” được căn chỉnh đúng cách.

- + TC 05 - Xác minh nút “**Submit**” được căn chỉnh đúng cách.
- + TC 06 - Xác minh khi click nút “**Submit**”, con trỏ chuột mặc định phải được thay đổi thành con trỏ chuột tay.
- + TC 07 - Xác minh rằng màu chữ của thông báo lỗi là màu ĐỎ
- + TC 08 - Xác minh phải có một lỗi thích hợp được tạo ra nếu người dùng nhập sai.
- + TC 09 - Xác minh các nhãn thích hợp phải được sử dụng trong các thông báo lỗi.
- + TC 10 - Xác minh rằng tất cả các trang phải có tiêu đề phù hợp.
- + TC 11 - Xác minh văn bản có font chữ phù hợp và đẹp mắt.
- + TC 12 - Xác minh rằng sau khi người dùng nhập đúng dữ liệu đầu vào, một thông báo phải được hiển thị.

### 9. Các công cụ kiểm thử giao diện đồ họa người dùng.

Bảng các công cụ kiểm thử giao diện đồ họa người dùng phổ biến sau đây.

Sản phẩm	URL
AutoIT	<a href="http://www.autoitscript.com/site/autoit/">http://www.autoitscript.com/site/autoit/</a>
EggPlant	<a href="http://www.testplant.com">www.testplant.com</a>
QTP	<a href="http://www8.hp.com/us/en/software-solutions/">http://www8.hp.com/us/en/software-solutions/</a>
SilkTest	<a href="https://www.microfocus.com/documentation/silk-test/205/">https://www.microfocus.com/documentation/silk-test/205/</a>
TestComplete	<a href="https://smartbear.com/product/testcomplete/free-trial/?v=1">https://smartbear.com/product/testcomplete/free-trial/?v=1</a>
Squish GUI Tester	<a href="https://www.froglogic.com/squish/download/">https://www.froglogic.com/squish/download/</a>

### 10. Kết luận.

#### 10.1. Nội dung

Thành công của một sản phẩm phần mềm phụ thuộc rất nhiều vào cách GUI (giao diện đồ họa người dùng) tương tác với người sử dụng và sự tiện lợi khi sử dụng các tính năng khác của nó. Do đó, kiểm tra giao diện là rất quan trọng. Kiểm tra bằng tay GUI đôi khi có thể được lặp đi lặp lại và nhàm chán do đó dễ bị lỗi. Tự động hóa được khuyến khích sử dụng cho kiểm thử giao diện đồ họa người dùng.

#### 10.2. Tài liệu tham khảo

Bài làm được tham khảo từ các tài liệu:

- <https://www.guru99.com/gui-testing.html>
- [https://viblo.asia/p/gui-testing-checklist-phan-1-jvElaLNoZkw?fbclid=IwAR1iLmvjshgBty1if\\_i0OGVJuYOFetnYIisHJjxNH4lNQAhENvoGyZ\\_9nhE](https://viblo.asia/p/gui-testing-checklist-phan-1-jvElaLNoZkw?fbclid=IwAR1iLmvjshgBty1if_i0OGVJuYOFetnYIisHJjxNH4lNQAhENvoGyZ_9nhE)

- <https://www.guru99.com/gui-testing.html?fbclid=IwAR1mC1IYksfSIRwPv-rIdDz3u3wye9vuhv4IT684lWhL7qFTL7Ulm9dNx6s>

## Chương 4: Kiểm thử tính khả dụng (Usability Testing)

### 1. Khái niệm

- Kiểm thử tính khả dụng là một kỹ thuật kiểm thử hộp đen để xác định rằng sản phẩm phần mềm có thân thiện với người dùng hay không.
- Mục đích của kiểm thử tính khả dụng là làm cho ứng dụng dễ sử dụng, linh hoạt, đáp ứng được yêu cầu,... khiến khách hàng/người dùng thích ngay lần đầu sử dụng.
- Được khuyến nghị thực hiện trong giai đoạn thiết kế ban đầu của Vòng đời phát triển phần mềm.

### 2. Lý do tại sao nên kiểm thử tính khả dụng

- Một sản phẩm trông tốt như thế nào thường quyết định nó hoạt động tốt như thế nào?
- Trong thực tế đã có rất nhiều sản phẩm không được chào đón vì một vài lý do như:
  - + Người dùng không biết nên bấm vào đâu để tiếp tục.
  - + Thông báo lỗi không nhất quán.
  - + Thời gian của một phiên sử dụng không đủ,...
- Kiểm thử tính khả dụng sẽ xác định các lỗi về khả năng sử dụng của sản phẩm ngay trong vòng đời phát triển phần mềm (SDLC) để giảm thiểu rủi ro trước khi đưa ra thị trường.

### 3. Mục tiêu kiểm thử tính khả dụng

*Kiểm thử tính khả dụng  
để xác định xem một ứng dụng có...*



- Mục tiêu chính là làm hài lòng người dùng, và nó chủ yếu tập trung vào các thông số hệ thống như:

#### 3.1. Hiệu quả của hệ thống

- Hệ thống có dễ học không?
- Hệ thống có hữu ích và tăng thêm giá trị cho đối tượng mục tiêu không?

- Nội dung, màu, biểu tượng, hình ảnh được sử dụng có mang tính thẩm mỹ không?

### **3.2. Hiệu quả**

- Điều hướng bắt buộc để tiếp cận màn hình / trang web mong muốn phải ít hơn. Không nên sử dụng thanh cuộn thường xuyên.
- Tính đồng nhất ở định dạng màn hình / trang trong ứng dụng / trang web của bạn.
- Cung cấp để tìm kiếm trong ứng dụng hoặc trang web phần mềm của bạn.

### **3.3. Độ chính xác**

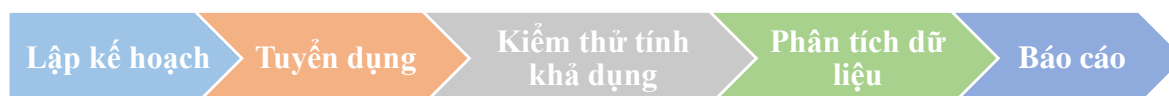
- Không có dữ liệu lỗi thời hoặc không chính xác như thông tin / địa chỉ liên hệ.
- Không có liên kết bị hỏng.

### **3.4. Người dùng thân thiện**

- Việc control được sử dụng phải tự giải thích và không được yêu cầu đào tạo để có thể hoạt động
- Cần cung cấp trợ giúp để người dùng hiểu ứng dụng / trang web
- Việc căn chỉnh với các mục tiêu trên giúp kiểm tra khả năng sử dụng hiệu quả.

## **4. Quy trình kiểm thử tính khả dụng**

Quy trình kiểm thử tính khả dụng bao gồm các giai đoạn sau:



### **4.1. Lập kế hoạch**

- Trong giai đoạn này, mục tiêu của Kiểm thử tính khả dụng được xác định. Có tình nguyện viên ngồi trước ứng dụng của bạn và ghi lại hành động của họ không phải là mục tiêu.
- Cần phải xác định các chức năng và mục tiêu quan trọng của hệ thống.
- Cần phải chỉ định nhiệm vụ cho người thử nghiệm của bạn, trong đó thực hiện các chức năng quan trọng này.
- Trong giai đoạn này, phương pháp thử khả năng sử dụng, số và nhân khẩu học của người kiểm tra khả năng sử dụng, các định dạng báo cáo thử nghiệm cũng được xác định.

### **4.2. Tuyển dụng**

Trong giai đoạn này, cần tuyển dụng số lượng người thử nghiệm mong muốn theo kế hoạch kiểm tra khả năng sử dụng của bạn. Tìm người thử nghiệm phù hợp với nhân khẩu

học của bạn (tuổi tác, giới tính, v.v.) và hồ sơ chuyên nghiệp (giáo dục, công việc, v.v.) có thể mất thời gian.

### **4.3. Kiểm thử tính khả dụng**

Trong giai đoạn này, các thử nghiệm tính khả dụng được thực thi.

### **4.4. Phân tích dữ liệu**

Dữ liệu từ các kiểm tra tính khả dụng được phân tích kỹ lưỡng để tìm ra các suy luận có ý nghĩa và đưa ra các đề xuất có thể thực hiện để cải thiện khả năng sử dụng chung của sản phẩm.

### **4.5. Báo cáo**

Các kết quả của bài kiểm tra tính dụng được chia sẻ với tất cả các bên liên quan có liên quan có thể bao gồm nhà thiết kế, nhà phát triển, khách hàng và CEO.

## **5. Các phương pháp kiểm thử tính khả dụng**

Có hai phương pháp để kiểm thử tính khả dụng:

### **5.1. Kiểm thử tính khả dụng trong phòng thí nghiệm**

- Thử nghiệm này được tiến hành trong một phòng thí nghiệm riêng biệt với sự hiện diện của các nhà quan sát.
- Những người kiểm thử được giao nhiệm vụ thực thi. Vai trò của người quan sát là theo dõi hành vi của người kiểm thử và báo cáo kết quả xét nghiệm. Người quan sát vẫn im lặng trong quá trình thử nghiệm.
- Trong thử nghiệm này, cả người quan sát và người thử nghiệm đều cùng có mặt ở cùng một vị trí.

### **5.2. Kiểm thử tính khả dụng từ xa**

- Theo các nhà quan sát thử nghiệm và kiểm thử này được định vị từ xa.
- Người thử nghiệm truy cập Hệ thống đang kiểm tra, từ xa và thực hiện các tác vụ được giao. Giọng nói, hoạt động màn hình, người thử nghiệm biểu hiện khuôn mặt của Tester được ghi lại bằng phần mềm tự động.
- Các nhà quan sát phân tích dữ liệu này và báo cáo kết quả của thử nghiệm.

## **6. Danh sách các mục cần kiểm tra cho kiểm thử trải nghiệm người dùng**

- Mục đích chính là tìm ra các vấn đề liên quan đến khả năng sử dụng của sản phẩm phần mềm trước khi đưa ra ngoài thị trường. Những mục sau đây cần phải được xem xét:
  - + Bắt đầu thực hiện kiểm thử trải nghiệm người dùng ngay trong giai đoạn đầu của thiết kế và phát triển.
  - + Tiến hành kiểm thử tính khả dụng đến những sản phẩm có sẵn trên thị trường cùng loại với sản phẩm của bạn để xác định các tiêu chuẩn về khả năng sử dụng với loại sản phẩm đó.



- + Chọn người dùng thử thích hợp để kiểm tra hệ thống.
- + Sử dụng công cụ định hình băng thông.
- + Người kiểm thử nên tập trung kiểm tra các chức năng quan trọng, hoặc các chức năng thường xuyên được người dùng sử dụng.
- + Với mỗi người thử nghiệm nên có một người quan sát để có thể ghi nhận chính xác các thao tác của họ.

### **7. Ưu điểm của kiểm thử tính khả dụng**

- Kiểm tra khả dụng tìm thấy các lỗi quan trọng của ứng dụng mà khó có thể tìm thấy trong quá trình phát triển.
- Sử dụng tài nguyên chính xác, kiểm thử tính khả dụng có thể hỗ trợ khắc phục mọi sự cố mà người dùng gặp phải trước khi phát hành ứng dụng.
- Kiểm tra khả năng sử dụng có thể được sửa đổi theo yêu cầu để hỗ trợ các loại thử nghiệm khác như kiểm tra chức năng, kiểm tra tích hợp hệ thống, kiểm tra đơn vị, v.v ...
- Kế hoạch kiểm tra khả năng sử dụng trở nên rất kinh tế, rất thành công và có lợi.
- Các vấn đề và vấn đề tiềm năng được đánh dấu trước khi sản phẩm được khởi chạy.

### **8. Nhược điểm của kiểm thử tính khả dụng**

- Quá trình lập kế hoạch và thu thập dữ liệu tốn nhiều thời gian.
- Phải mất rất nhiều nguồn lực (thời gian, tiền bạc,..) để xây dựng một phòng thí nghiệm kiểm thử tính khả dụng.
- Tuyển dụng và quản lý người thực hiện kiểm tra tính khả dụng cũng rất tốn kém.

## **9. Kết luận**

### **9.1. Nội dung**

- Kiểm thử tính khả dụng trước khi bàn giao là vô cùng quan trọng vì có thể những chi phí phải trả do lỗi,.. còn cao hơn chi phí để kiểm thử.
- Vậy nên kiểm thử tính khả dụng rất được khuyến khích thực hiện.

### **9.2. Tài liệu tham khảo**

- Usability Testing Examples: 4 Usability & Task Scenario Examples - TestLodge Blog
- Detailed Understand of Usability Testing: What? Why? & How? (softwaretestingclass.com)
- What is Usability Testing? UX(User Experience) Testing Example (guru99.com)

## **Tài liệu tham khảo**

Báo cáo được tham khảo từ các nguồn:

### **Chương 1:**

- <https://www.guru99.com/data-testing.html>
- <https://www.guru99.com/what-is-backend-testing.html>
- [https://www.mindk.com/blog/backend-testing/?fbclid=IwAR2fzaX\\_CT0eG1VV4TZNg4Nzq2r5-CgzmlsTlC28PyBF8IwXb19esTYPQ0](https://www.mindk.com/blog/backend-testing/?fbclid=IwAR2fzaX_CT0eG1VV4TZNg4Nzq2r5-CgzmlsTlC28PyBF8IwXb19esTYPQ0)

### **Chương 2:**

- <https://www.guru99.com/frontend-testing.html>
- <https://wikimaytinh.com/trinh-duyet-web-la-gi-cac-loai-trinh-duyet-web.html>
- <https://meohaycuocsong.com/mot-so-he-dieu-hanh-va-lich-su-cac-he-dieu-hanh.html>
- <https://viblo.asia/p/cross-browser-testing-voi-lambda-test-bJzKmzbXZ9N>
- <https://topdev.vn/blog/selenium-la-gi-mot-so-kinh-nghiem-lam-viec-voi-selenium/>
- <https://mona.media/tim-hieu-ve-quy-trinh-kiem-thu-website/>
- <https://viblo.asia/p/tim-hieu-ve-kiem-thu-chuc-nang-functionality-testing-bJzKmLVB59N>
- <https://www.toponseek.com/blogs/pagespeed-toc-do-tai-trang/>
- <http://yslow.org/> và <https://qnet88.com/yslow-la-gi-va-lam-the-nao-de-toi-uu-no/>

### **Chương 3:**

- <https://www.guru99.com/gui-testing.html>
- [https://viblo.asia/p/gui-testing-checklist-phan-1-jvElalNoZkw?fbclid=IwAR1iLmvjshgBty1if\\_i0OGVJuYOFetnYIisHJjxNH4lNQAHEHvoGyZ\\_9nhE](https://viblo.asia/p/gui-testing-checklist-phan-1-jvElalNoZkw?fbclid=IwAR1iLmvjshgBty1if_i0OGVJuYOFetnYIisHJjxNH4lNQAHEHvoGyZ_9nhE)
- <https://www.guru99.com/gui-testing.html?fbclid=IwAR1mC1IYksfSIRwPv-rIdDz3u3wye9vuhv4IT684lWhL7qFTL7Ulm9dNx6s>

### **Chương 4:**

- Usability Testing Examples: 4 Usability & Task Scenario Examples - TestLodge Blog
- Detailed Understand of Usability Testing: What? Why? & How? (softwaretestingclass.com)