**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**KHOA CÔNG NGHỆ THÔNG TIN**

---🙘🙚---



**BÁO CÁO TIỂU LUẬN: KIỂM THỬ PHẦN MỀM**

**ĐỀ TÀI: XÂY DỰNG WEB QUẢN LÝ NHÀ TRỌ**

**Giảng viên hướng dẫn:** ThS Võ Phương Bình

**Sinh viên thực hiện:** 2113005 – Lê Hà Hiếu Nghĩa

2115277 – Nguyễn Trung Tín

2116977 – Trần Hữu Bằng

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

Đà Lạt, ngày ...... tháng ...... năm ......

Giảng viên hướng dẫn

[Ký và ghi rõ họ tên]

# MỤC LỤC

[MỤC LỤC 3](#_Toc166488695)

[LỜI CẢM ƠN 4](#_Toc166488696)

[BẢNG GIẢI THÍCH THUẬT NGỮ 5](#_Toc166488697)

[1. Lý thuyết 5](#_Toc166488698)

[1.1. Một số khái niệm 5](#_Toc166488699)

[1.2. Kiểm thử thủ công và tự động 6](#_Toc166488700)

[1.3. Các công cụ hỗ trợ KTPM 6](#_Toc166488701)

[2. Hệ thống 7](#_Toc166488702)

[2.1. Tóm tắt hệ thống 7](#_Toc166488703)

[2.2. Sơ đồ ngữ cảnh của hệ thống 7](#_Toc166488704)

[2.3. Tóm tắt tính năng của hệ thống 8](#_Toc166488705)

[2.4. Sơ đồ Use Case 8](#_Toc166488706)

[2.5. Danh sách Use Cases 9](#_Toc166488707)

[**2.5.1. Đăng nhập 9**](#_Toc166488708)

[**2.5.2. Đăng thông tin 9**](#_Toc166488709)

[**2.5.3. Tìm kiếm thông tin 10**](#_Toc166488710)

[**2.5.4. Cập nhật 10**](#_Toc166488711)

[2.6. Các yêu cầu và ràng buộc phi chức năng 11](#_Toc166488712)

[2.7. Công nghệ và công cụ sử dụng 11](#_Toc166488713)

[3. Domain Name class 11](#_Toc166488714)

[3.1. Liên kết giữa các thuộc tính 11](#_Toc166488715)

[3.2. Hợp đồng hoạt động 12](#_Toc166488716)

[3.3. Sơ đồ hoạt động (Activity Diagram) 12](#_Toc166488717)

[4. Kết quả 12](#_Toc166488718)

[**4.1. Giao diện 12**](#_Toc166488719)

[**4.2. Kết quả các test scenario 13**](#_Toc166488720)

[**4.3. Mã nguồn 13**](#_Toc166488721)

[TÀI LIỆU THAM KHẢO 13](#_Toc166488722)

# LỜI CẢM ƠN

Lời đầu tiên chúng em xin chân thành cảm ơn giảng viên hướng dẫn, ThS.Võ Phương Bình – người hướng dẫn chúng em tận tình trong thời gian học tập và làm bài. Những sự chỉ dẫn là hướng đi chính xác để đề tài của nhóm em triển khai và hoàn thiện đáp ứng những tiêu chí đã được đưa ra.

Chúng em cũng xin chân thành cảm ơn đến quý thầy/cô Khoa Công nghệ Thông tin – Trường Đại Học Đà Lạt đã dành rất nhiều tâm huyết tận tình giảng dạy, truyền đạt những kiến thức cần thiết và tạo điều kiện tốt nhất cho em thực hiện đề tài này. Trong quá trình thực hiện bài tiểu luận, chúng em đã có cơ hội được tiếp cận với những công nghệ mới, đồng thời cũng giúp cho chúng em có thêm kinh nghiệm, nền tảng quý báu cho bản thân để tiếp tục phát triển trên con đường học tập của mình.

Dưới đây là kết quả của quá trình tìm hiểu và nghiên cứu mà nhóm em đã đạt được trong thời gian vừa qua. Trong quá trình thực hiện, do khả năng còn hạn chế và chưa có được nhiều kinh nghiệm thực tế nên chúng em không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và những góp ý chỉ bảo tận tình của quý thầy, cô.

Chúng em xin cảm ơn giảng viên ThS. Võ Phương Bình đã giảng dạy và chỉ bảo tận tình trong quá trình học tập, nghiên cứu và hoàn thành bài tiểu luận này.

Trong khuôn khổ của bài tiểu luận, do thời gian và kinh nghiệm thực tế còn hạn chế nên có những phần thực hiện chưa được tốt, chúng em rất mong nhận được sự góp ý của thầy và các bạn.

Chúng em xin chân thành cảm ơn !

# BẢNG GIẢI THÍCH THUẬT NGỮ

|  |  |
| --- | --- |
| **Từ viết tắt** | **Tên đầy đủ** |
| **KTPM** | Kiểm thử phần mềm |
| **API** | Giao diện lập trình ứng dụng |
| **CSDL** | Cơ sở dữ liệu |
| **SinhVien** | Đối tượng sinh viên |
| **ChuTro** | Những người cho thuê trọ |
| **QuanTriVien** | Quản trị viên |
| **Tro** | Phòng trọ |
| **DSNguoiThue** | Danh sách người đến thuê trọ |

# Lý thuyết

## Một số khái niệm

**Khái niệm phần mềm**

Phần mềm thường được sử dụng để hướng dẫn, cho phép người dùng có thể tương tác với các phần cứng của máy tính hoặc thực hiện nhiều tác vụ có liên quan tới công việc, hoạt động của máy tính. Nó bao gồm một tập hợp những câu lệnh được lập trình từ một hay nhiều ngôn ngữ lập trình khác nhau theo một trật tự nhất định. Điều này nhằm giúp máy tính hiểu được và thực hiện các chức năng hay một vấn đề nào đó.

Chỉ cần một sự cố nhỏ trong phần mềm đều dẫn đến những tác động nghiêm trọng như thiệt hại về tiền bạc, thời gian và danh tiếng đối với các công ty, doanh nghiệp. Năm 1980, lỗi phần mềm trong mã điều khiển máy xạ trị Therac-25 là nguyên nhân trực tiếp gây một số ca tử vong và thiệt hại 370 triệu USD [1] [2]. Một điểm khác cần xem xét đến là chất lượng của phần mềm ở Mỹ từ giai đoạn 2020 trở về sau thì tồi tệ hơn [3].

**Khái niệm KTPM**

Glenford J.Myers (nhà khoa học máy tính và doanh nhân người Mỹ) định nghĩa kiểm thử phần mềm là “ *quá trình thực thi một chương trình hoặc một phần của chương trình với mục đích tìm ra lỗi* [4]”.

Nó đảm bảo sản phẩm phải đáp ứng chính xác theo các yêu cầu của khách hàng đề ra. Ngoài ra, nó còn tạo điều kiện cho người dùng tận dụng tối đa tư duy sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa nhìn thấy.

Phần mềm có nhiều khả năng xảy ra lỗi vì chúng được thiết kế thủ công. Những lỗi này có thể xuất hiện trong giai đoạn mã hóa phần mềm và trong suốt vòng đời phát triển. Trong vòng đời phát triển, lỗi có thể xảy ra ở giai đoạn đầu khi các yêu cầu được hiểu, viết hoặc thiết kế. Ngoài ra, các tài liệu được xác định rõ ràng, khi được giao cho các nhóm khác nhau, thường có thể dẫn đến những cách hiểu khác nhau nếu các nhóm không được đào tạo để hiểu các tài liệu yêu cầu.

## Kiểm thử thủ công và tự động

Kiểm thử phần mềm thường được tiến hành trong ba giai đoạn: tạo dựng, thực hiện và đánh giá các trường hợp kiểm thử [5]. Các kỹ thuật chính được sử dụng là kiểm thử thủ công và tự động.

**Kiểm thử thủ công (Manual testing)**

Đây là việc kiểm thử phần mềm được thực hiện thủ công bởi QA/tester. Nó được thực hiện để tìm ra bug trong quá trình phát triển phần mềm. Trong manual testing, Tester phải kiểm tra tất cả các tính năng cần thiết của ứng dụng. Trong quá trình này, Tester phải execute test, lập báo cáo mà không cần sự trợ giúp của bất kì công cụ tự động nào.

Quy trình Testint manual:

1. Hiểu rõ các yêu cầu
2. Viết các test case

Công cụ kiểm thử:

* + Quản lý kiểm tra: HP-ALM hoặc testlink: có thể tự viết ra các kịch bản kiểm thử, các test case và test scripting, có thêt liên kết yêu cầu của dự án. Nó còn đảm bảo các báo cáo thử nghiệm được dễ dàng và hiệu quả.
  + Quản lý lỗi: Bugzilla, Mantis: đây là cần thiết để nâng cao và quản lý các lỗi. Đôi khi bạn phải tạo các báo cáo lỗi như đã nêu ở bản phát hành(từ trước)

**Kiểm thử tự động**

Kiểm thử phần mềm tự động là thực hiện kiểm thử phần mềm bằng một chương trình đặc biệt với rất ít hoặc không có sự tương tác của con người, giúp cho người thực hiện việc kiểm thử phần mềm (tester) không phải lặp đi lặp lại các bước nhàm chán.

## Các công cụ hỗ trợ KTPM

Đó là công cụ dùng để kiểm tra xem sản phẩm phần mềm trước khi đưa vào sử dụng có phù hợp với các yêu cầu mong đợi hay không và nó có khiếm khuyết hay lỗi nào không. Việc lựa chọn công cụ kiểm thử phần mềm phù hợp với tính chất của dự án sẽ giúp bạn tiết kiệm thời gian, công sức, đảm bảo kết quả nhất quán và năng suất kiểm nghiệm cao. Bên cạnh đó, thông tin về kiểm tra cũng được trình bày một cách rõ ràng, dễ hiểu.



*Hình 1.1: Một số công cụ phổ biến hỗ trợ việc KTPM*

**Selenium** là một trong những công cụ kiểm thử phần mềm phổ biến nhất. Nó được thiết kế đặc biệt để hỗ trợ Kiểm tra tự động hóa các khía cạnh chức năng của các ứng dụng trên web, đa dạng các nền tảng và trình duyệt.

**Applitools** là một công cụ kiểm tra tự động, tự động xác nhận giao diện và trải nghiệm người dùng của các ứng dụng và trang web. Nó được thiết kế để dễ dàng tích hợp với các bài kiểm tra hiện có thay vì yêu cầu tạo một bài kiểm tra mới.

**Postman** hiện là một trong những công cụ phổ biến nhất được sử dụng trong kiểm thử API. Nó bắt đầu vào năm 2012 như một dự án phụ của Abhinav Asthana để đơn giản hóa quy trình làm việc API trong kiểm thử và phát triển.

Ưu và nhược điểm của kiểm thử Hộp đen theo tác giả Isha Sunita Sangwan [6]:

|  |  |
| --- | --- |
| **Ưu điểm** | **Nhược điểm** |
| Số lượng trường hợp kiểm thử được giảm để đạt được kiểm thử hợp lý | Các trường hợp kiểm thử khó thiết kế nếu không có thông số kỹ thuật rõ ràng. |
| Các trường hợp kiểm thử có thể hiển thị sự hiện diện hoặc vắng mặt của các lớp lỗi | Chỉ một số lượng nhỏ đầu vào có thể thực sự có thể được kiểm tra. |
| Hiệu quả hơn trên các đơn vị mã lớn hơn so với kiểm tra hộp rõ ràng. | Một số phần của backend hoàn toàn không được thử nghiệm. |
| Lập trình viên và tester đều độc lập với nhau. | Cơ hội có các đường dẫn không xác định trong quá trình thử nghiệm này |

*Bảng 1.1: So sánh ưu / nhược điểm của phương pháp Black-box*

# Hệ thống

## Tóm tắt hệ thống

Hệ thống nhà trọ Đà Lạt là một trang web tìm nhà trọ, các căn hộ hoặc nhà nguyên căn nhằm đáp ứng cho các sinh viên cũng như những người lao động từ những tỉnh thành khác trên cả nước khi đến để làm việc và học tập. Trang web giúp người dùng tiết kiệm được thời gian cũng như kinh phí sau khi sử dụng trang web.

## Sơ đồ ngữ cảnh của hệ thống

A diagram of a diagram

Description automatically generated

*Hình 2.1 – Mô tả ngữ cảnh của hệ thống*

## Tóm tắt tính năng của hệ thống

|  |  |
| --- | --- |
| **ID** | **Mô tả tính năng** |
| **FEA-01** | **Đăng nhập** |
| **FEA-02** | **Đăng ký** |
| **FEA-03** | **Đăng thông tin liên quan** |
| **FEA-04** | **Tìm kiếm thông tin** |
| **FEA-05** | **Cập nhật thông tin của SinhVien, do ChuTro thực hiện** |

*Bảng 2.1: Tóm tắt tính năng hệ thống*

## Sơ đồ Use Case

**A diagram of a network

Description automatically generated**

*Hình 2.2 – Sơ đồ Use Case*

## Danh sách Use Cases

### Đăng nhập

|  |  |
| --- | --- |
| **Use Case UC1: Đăng nhập** | |
| **Phạm vi truy cập** | Tất cả các actor |
| **Luồng sự kiện chính** | 1. Hệ thống hiển thị màn hình đăng nhập 2. Người dùng nhập username, password 3. Nếu thành công, hệ thống thể hiện việc đăng nhập đã hoàn tất 4. Kết thúc use case |
| **Luồng sự kiện phụ** | **1a** Mật khẩu không hợp lệ (khi nhập sai username hoặc password   * Bắt nhập lại thông tin, đi kèm đó là thông báo nhập sai và yêu cầu nhập lại * Quay lại bước 2 trong luồng sự kiện chính   **1b.** Quên mật khẩu: khi người dùng chọn chức năng quên mật khẩu trên màn đăng nhập   * Hiển thị màn hình |

*Bảng 2.2: Use case đăng nhập*



*Hình: Sơ đồ tuần tự cho UC1*

### Đăng thông tin

|  |  |
| --- | --- |
| **Use Case UC2: Đăng thông tin** | |
| **Phạm vi truy cập** | ChuTro |
| **Luồng sự kiện chính** | 1a: Tạo mới   1. Thực hiện việc đăng nhập 2. Đưa ra form chứa các textbox liên quan, người chủ chỉ cần nhập thông tin liên quan vào 3. Nhấn xác nhận 4. Gửi thông báo thành công, thông tin mới đó được hiển thị lên form 5. Kết thúc thao tác (có thể load lại trang) |
| **Luồng sự kiện phụ** | * Nếu chưa đăng nhập thì phải thực hiện lại bước **1** trong luồng sự kiện chính |

*Bảng 2.2: Use case đăng thông tin*

### Tìm kiếm thông tin

|  |  |
| --- | --- |
| **Use Case UC3: Tìm kiếm thông tin** | |
| **Phạm vi truy cập** | SinhVien |
| **Luồng sự kiện chính** | 1a:   1. Truy cập trang web |
| **Luồng sự kiện phụ** | * Có thể không cần đăng nhập |

*Bảng 2.2: Use case đăng nhập*

### Cập nhật

|  |  |
| --- | --- |
| **Use Case UC4: Cập nhật( Thêm xóa, sửa)** | |
| **Phạm vi truy cập** | ChuTro |
| **Điều kiện kiên quyết** | * Phải hoàn thành việc đăng nhập vào web |
| **Luồng sự kiện chính** | 1a: Chức năng thêm   1. Hệ thống hiển thị màn hình quản lý (điền thông tin của người mới đến thuê) 2. ChuTro nhập thông tin liên quan 3. Nếu thành công, hiển thị thêm thông tin thành công, cập nhật trong db 4. Kết thúc thao tác   1b: Chức năng chỉnh sửa hoặc xóa   1. Hệ thống hiển thị màn hình quản lý 2. Nhấn vào nút edit hoặc delete ở góc bên phải 3. Hiển thị thông tin để user có thể thực hiện thao tác 4. Hệ thống gửi form xác nhận có muốn thay đổi hay không? 5. Nếu thành công, hiển thị thông tin đã chỉnh sửa hoặc xóa thành công 6. Kết thúc thao tác |

*Bảng 2.2: Use case đăng nhập*

## Các yêu cầu và ràng buộc phi chức năng

|  |  |  |
| --- | --- | --- |
| **ID** | **Loại** | **Mô tả** |
| NFA-01 | Hiệu suất | Truy cập trang web nhanh |
| NFA-02 | Khả năng sử dụng | Các đối tượng dùng web |
| NFA-03 | Độ tin cậy | Thông tin chính xác |

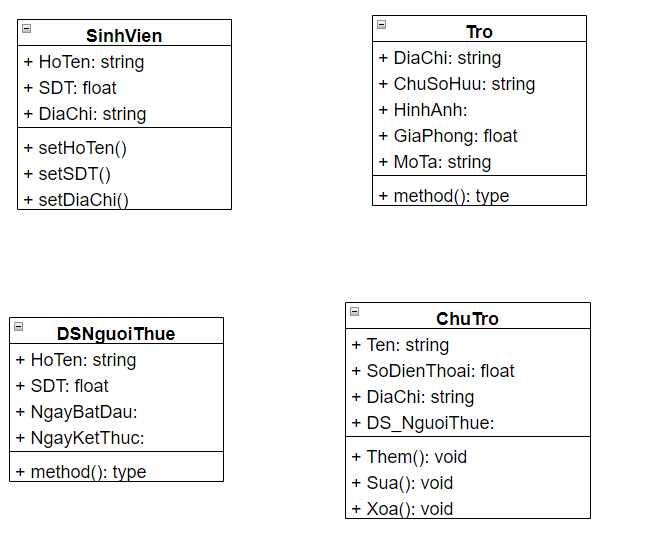
## Công nghệ và công cụ sử dụng

* Dự án được viết bằng ngôn ngữ ReactJS, sử dụng CSDL MongoDB
* Môi trường phát triển: Visual Studio Code

# Domain Name class

## Liên kết giữa các thuộc tính

|  |
| --- |
| **Mô tả các quan hệ** |
| Một **SinhVien** có thể đến thuê tại một phòng trong một khoảng thời gian nhất định. |
| Một **chủ trọ** có thể cho nhiều **SinhVien** thuê tại một khu trọ |
| Một **chủ trọ** có thể sở hữu nhiều khu trọ (nếu có thể) |
| Một **ChuTro** chỉ có thể có được duy nhất một **DSNguoiThue** |
| Một **Tro** có thể được nhiều người sử dụng trong nhiều lần, và chỉ được sở hữu bởi duy nhất một **ChuTro** |



***Hình: Mô hình miền của các lớp với liên kết và thuộc tính***

## Hợp đồng hoạt động

Mấy cái bảng Contract Operation

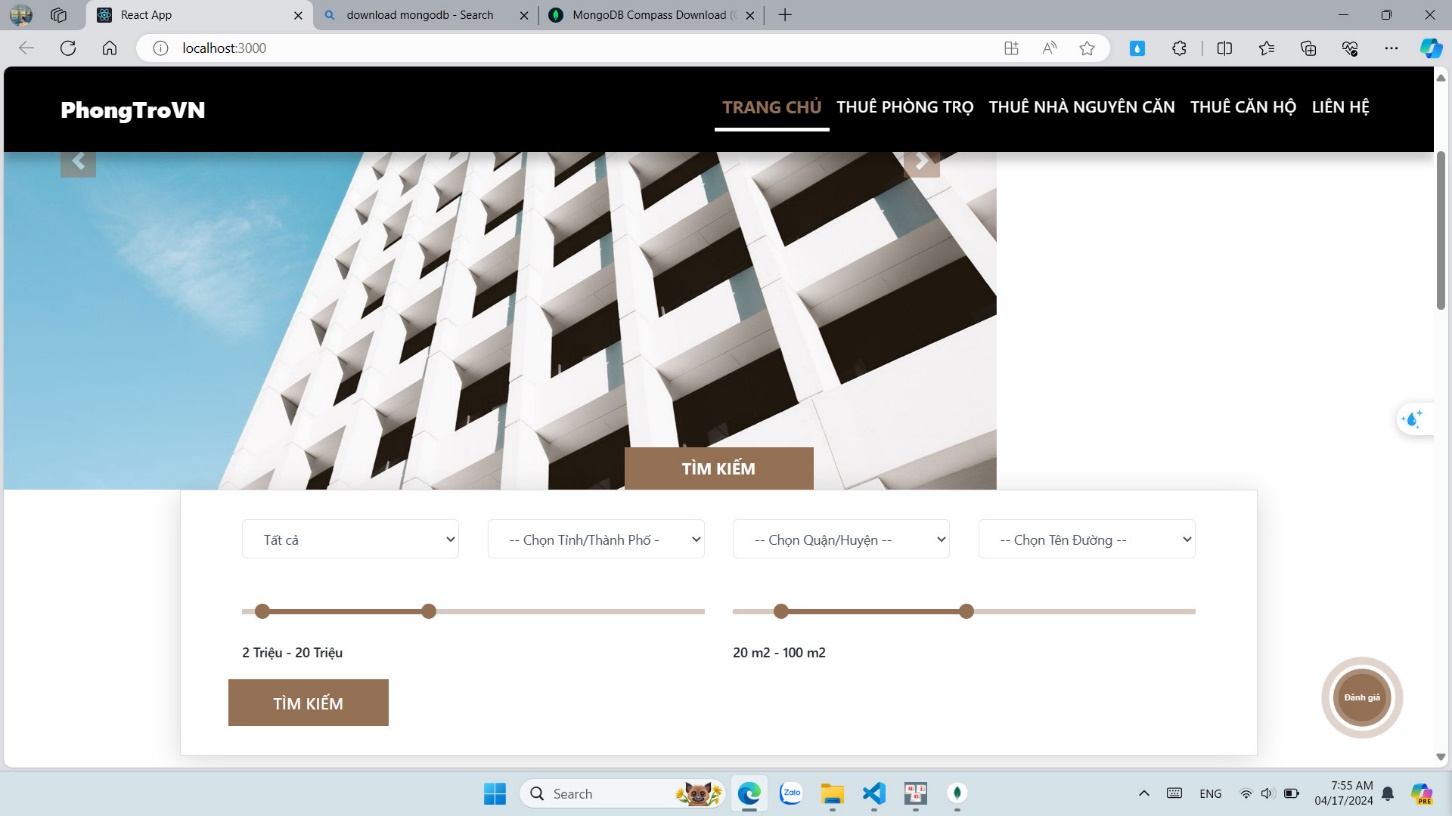
|  |  |
| --- | --- |
| **Contract Operation ID** | **Mô tả** |
| CO-01 | Đăng nhập |
| CO-02 | Đăng ký |
| CO-03 | Hiển thị thông tin |
| CO-04 | Cập nhật( thêm, xóa, sửa) |
| CO-05 | Đăng xuất |

## Sơ đồ hoạt động (Activity Diagram)

# Kết quả

### Giao diện

* 1. **Giao diện chính**



*Hình 4.1: Giao diện trang chủ*

Mô tả:

* 1. **Giao diện**

### Kết quả các test scenario

### Mã nguồn

Chi tiết mã nguồn để tại link:

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Simplilearn, "Fundamentals of Software Testing: Concepts and Process," 21 7 2023. [Online]. Available: https://www.simplilearn.com/tutorials/devops-tutorial/fundamentals-of-software-testing. |
| [2] | P. H. Hoàng, "NHỮNG DÒNG CODE GIẾT NGƯỜI – ĐẠO ĐỨC VÀ CÁI TÂM CỦA DEVELOPER," 30 7 2016. [Online]. Available: https://toidicodedao.com/2016/12/15/dao-duc-developer/. |
| [3] | H. Krasner, "The cost of Poor Software Quality in the US: A 2022 Report," Spicewood, 2022. |
| [4] | Glenford J. Myers, Corey Sandler, Tom Badgett, The Art of Software Testing 3rd Edition, Wiley, 2023. |
| [5] | FATIH GURCAN, GONCA GOKCE MENEKSE DALVEREN, NERGIZ ERCIL CAGILTAY, "Evolution of Software Testing Strategies and Trends: Semantic Content Analysis of Software Research Corpus of the Last 40 Years," *NTNU Open,* vol. 10, 2022. |
| [6] | S. S. Isha, “Software Testing Techniques and Strategies,” tập 4, số 4, pp. 99-102, 4 2014. |