TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

****

**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**TÊN ĐỀ TÀI**

**ỨNG DỤNG CÔNG CỤ SELENIUM WEBDRIVER THỰC HIỆN KIỂM THỬ TỰ ĐỘNG CHO WEBSITE “ COVID-19 VIỆT NAM ”**

**Đơn vị thực tập: TMA SOLUTIONS BÌNH ĐỊNH**

**Sinh viên thực hiện:**

Lê Thị Cẩm Tú Lớp : 44k21.2

**Giảng viên hướng dẫn:** Ths. Cao Thị Nhâm

**Đà Nẵng, 7/2021**

**LỜI CẢM ƠN**

Chuyên đề thực tập nghề nghiệp chuyên ngành Quản trị hệ thống thông tin với đề tài “**ỨNG DỤNG CÔNG CỤ SELENIUM WEBDRIVER THỰC HIỆN KIỂM THỬ TỰ ĐỘNG CHO WEBSITE “ COVID-19 VIỆT NAM ”** là kết quả của quá trình cố gắng của bản thân và được sự giúp đỡ tận tình, động viên khích lệ của thầy cô. Qua đây, chúng em xin gửi lời cảm ơn chân thành đến những người đã giúp đỡ chúng tôi trong thời gian học tập - Nghiên cứu chuyên đề vừa qua.

Chúng em xin trân trọng gửi đến cô Cao Thị Nhâm - Người đã trực tiếp tận tình hướng dẫn cũng như cung cấp tài liệu cần thiết, chỉ bảo, góp ý để chúng em có thể hoàn thành được báo cáo một cách tốt nhất lời cảm ơn chân thành và sâu sắc nhất.

Xin cảm ơn lãnh đạo, ban giám hiện cùng toàn thể các thầy cô giáo trường Đại học Kinh Tế - Đại học Đà Nẵng, khoa Thống kê Tin học đã tạo điều kiện cho chúng em hoàn thành tốt chuyên đề tốt nghiệp của mình.

Chúng em xin gửi lời cảm ơn tới Công ty TMA Solutions Bình Định , đặc biệt là anh metor Nguyễn Văn Tường và cũng như sự hỗ trợ nhiệt tình từ nhiều phía của các anh chị trong công ty để chúng em có thể hoàn thành tốt đợt thực tập vừa qua. Với những gì đã tích lũy được, giờ đây chúng em đã xác nhận được hướng đi đúng đắn nhất cho bản thân để có thể vươn tới thành công trong tương lai.

Bước đầu đi vào thực tế của chúng em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót, rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô để kiến thức của được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao kiến thức của mình.

Chúng em xin chân thành cảm ơn!

Sinh viên

Lê Thị Cẩm Tú

**LỜI CAM ĐOAN**

Chúng em xin cam đoan đề tài “**ỨNG DỤNG CÔNG CỤ SELENIUM WEBDRIVER THỰC HIỆN KIỂM THỬ TỰ ĐỘNG CHO WEBSITE “ COVID-19 VIỆT NAM ”** là kết quả nghiên cứu độc lập dưới sự hướng dẫn của giảng viên hướng dẫn: TS. Cao Thị Nhâm và mentor: Nguyễn Văn Tường. Ngoài ra, không có bất cứ sự sao chép nào của người khác.

Đề tài, nội dung báo cáo là sản phẩm mà chúng em đã nổ lực nghiên cứu trong quá trình học tại nhà và bên thực tập. Các dữ liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực, chúng em xin chịu hoàn toàn trách nhiệm, kỷ luật bộ môn và nhà trường đề ra nếu có vấn đề nào xảy ra.

# MỤC LỤC

[MỤC LỤC ii](#_Toc74235359)

[DANH MỤC HÌNH ẢNH iv](#_Toc74235360)

[DANH MỤC BẢNG BIỂU v](#_Toc74235361)

[DANH MỤC CÁC TỪ VIẾT TẮT vi](#_Toc74235362)

[LỜI MỞ ĐẦU 1](#_Toc74235363)

[CHƯƠNG 1. TỔNG QUAN VỀ LÝ THUYẾT 2](#_Toc74235364)

[1.1. Software Testing Overview 2](#_Toc74235365)

[1.1.1. Introduction to Software Testing 2](#_Toc74235366)

[1.1.2. Software Development Process 2](#_Toc74235367)

[1.1.3. Principle of testing 2](#_Toc74235367)

[1.1.4. Error/ Fault/ Failure 2](#_Toc74235367)

[1.1.5. What is Verification & Validation 2](#_Toc74235367)

[1.1.6. Difference between QA & QC 2](#_Toc74235367)

[1.2. Software Development Life cycle (SDLC) 2](#_Toc74235368)

[1.2.1. SDLC Phases 2](#_Toc74235367)

[1.2.2. Các mô hình phổ biến của SDLC 2](#_Toc74235367)

[1.2.3. Scrum Methodology 2](#_Toc74235367)

[1.2.4. Kanban Methodology 2](#_Toc74235367)

[1.3. Software Testing Type 2](#_Toc74235368)

[1.3.1. Manual Testing 2](#_Toc74235367)

[1.3.2. Automation Testing 2](#_Toc74235367)

[1.4. Software Testing Methodologies 2](#_Toc74235368)

[1.4.1. Static Testing 2](#_Toc74235367)

[1.4.2. Dynamic Testing 2](#_Toc74235367)

[1.4.3. White Box Testing 2](#_Toc74235367)

[1.4.4. Black Box Testing 2](#_Toc74235367)

[1.4.5. Gray Box Testing 2](#_Toc74235367)

[1.5. Software Testing Levels 2](#_Toc74235368)

[1.5.1. Unit Testing 2](#_Toc74235367)

[1.5.2. Integration Testing 2](#_Toc74235367)

[1.5.3. System Testing 2](#_Toc74235367)

[1.5.4. Acceptance Testing 2](#_Toc74235367)

[1.6. Test case Design Technique 2](#_Toc74235368)

[1.6.1. Khái niệm 2](#_Toc74235367)

[1.6.2. Các loại kỹ thuật Testcase 2](#_Toc74235367)

[1.7. WebDriver Commands 2](#_Toc74235368)

[1.8. Cơ hội nghề nghiệp 2](#_Toc74235368)

[CHƯƠNG 2. TỔNG QUAN VỀ HỆ THỐNG CẦN TEST 3](#_Toc74235369)

[2.1. Tổng quan về hệ thống 3](#_Toc74235370)

[2.2. Phân tích yêu cầu 3](#_Toc74235370)

[2.2.1. Selenium IDE 3](#_Toc74235371)

[2.2.2. Selenium Remote Control 3](#_Toc74235372)

[2.2.3. Selenium Web Driver 3](#_Toc74235372)

[2.2.4. Selenium Grid 3](#_Toc74235372)

[2.3.2. Tại sao sử dụng Selenium Web Driver? 3](#_Toc74235372)

[2.3.3. Sử dụng Selenium Web Driver ở đâu 3](#_Toc74235372)s

[CHƯƠNG 3. THIẾT KẾ TESTCASE VÀ THỰC HIỆN TEST… 4](#_Toc74235374)

[3.1. Mục 3.1 4](#_Toc74235375)

[3.1.1. Mục 3.1.1 4](#_Toc74235376)

[3.1.2. Mục 3.1.2 4](#_Toc74235377)

[3.2. Mục 3.2 4](#_Toc74235378)

DANH MỤC HÌNH ẢNH

[Hình 1.2.2 SDLC Phases 2](#_Toc74235469)

[Hình 1.2.2.1 Mô hình Water Fall 5](#_Toc74235470)

[Hình 1.2.2.2 Mô hình V Model 2](#_Toc74235469)

[Hình 1.2.2.3 Mô hình Agile Model 5](#_Toc74235470)

[Hình 1.2.3 Scrum Methodology 2](#_Toc74235469)

[Hình 1.2.4 Kanban Methodology 5](#_Toc74235470)

[Hình 1.4.3 White Box Testing 2](#_Toc74235469)

[Hình 1.4.4 Black Box Testing 5](#_Toc74235470)

[Hình 1.4.5 Gray Box Testing 2](#_Toc74235469)

DANH MỤC BẢNG BIỂU

[Bảng 1.1.5 What is Verification & Validation 3](#_Toc74235471)

[Bảng 1.1.6 Difference between QA &QC 4](#_Toc74235471)

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

* Đề tài này nghiên cứu …

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ...
* ...

1. **Phương pháp nghiên cứu**

• Tìm hiểu về quy trình phát triển phần mềm

• Tìm hiểu về quy trình kiểm thử

• Tìm hiểu về các phương pháp kiểm thử, cấp độ kiểm thử, phương pháp thiết kế trường hợp kiểm thử, cách thức báo cáo lỗi

• Tìm hiểu cách sử dụng công cụ Selenium WebDriver và testNG framework để xây dựng test Automation test

• Phân tích hệ thống website “COVID-19 VIỆT NAM”

• Thực hiện kiểm thử thủ công

• Sử dụng công cụ Selenium WebDriver để kiểm thử tự động

1. **Đối tượng và phạm vi nghiên cứu**

* ...

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, 3 chương

* Mở đầu
* **Chương 1**: Tổng quan về lý thuyết.
* **Chương 2**: Tổng quan về hệ thống test.
* **Chương 3**: Thiết kế test case và thực hiện test.

# TỔNG QUAN về lý thuyết

## Software Testing Overview

### Introduction to Software Testing.

1. ***Software Testing là gì?***

* Kiểm thử phần mềm là một quá trình thực thi một chương trình hoặc ứng dụng với mục đích tìm ra lỗi phần mềm. Và được sử dụng để xác định tính đúng đắn, đầy đủ và chất lượng của phần mềm máy tính đã phát triển.

1. ***Objectives of Testing***

* Để tìm và ngăn ngừa lỗi trong sản phẩm phần mềm.
* Giảm rủi ro về chất lượng phầm mềm.
* Để có được niềm tin của khách hàng bằng cách cung cấp cho họ sản phẩm chất lượng.
* Để đảm bảo rằng kết quả cuối cùng đáp ứng các yêu cầu của doanh nghiệp và người dùng.

### Sofware Development Process.

* Quy trình phát triển phần mềm là một tập hợp các hoạt động tổ chức mà mục đích của chúng là xây dựng và phát triển phần mềm.
* Có 4 thao tác là nền tảng của hầu hết các quy trình phát triển phần mềm:
* Đặc tả phần mềm.
* Quy trình phát triển phần mềm.
* Đánh giá phần mềm.
* Tiến hóa phần mềm.

### Principle of testing.

1. nguyên tắc của Testing:

* Testing shows presence of defects
* Exhaustive testing is impossible
* Early testing
* Defect clustering
* Pesticide paradox
* Testing is context dependent
* Absence – of – errors fallacy

### Error/Fault/Failure.

* Error:
* Đề cập đến sự khác biệt giữa sản phẩm thực tế và sản phẩm mong đợi.
* Các nhà phát triển và kỹ sư kiểm tra tự động hóa nêu ra lỗi.
* Sự cố trong mã dẫn đến lỗi.
* Fault:
* Là một tình trạng khiến phần mềm không thực hiện được chức năng cần thiết của nó.
* Sai lầm của con người gây ra lỗi.
* Failure:
* Là hệ thống hoặc thành phần không có khả năng thực hiện chức năng cần thiết theo đặc điểm kỹ thuật của nó.
* Người dùng cuối phát hiện ra một vấn đề trong sản phẩm.

### What is Verification & Validation.

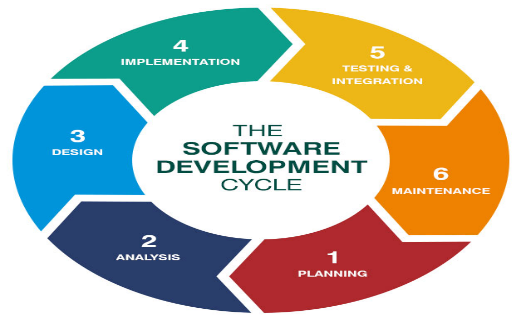
|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Xác minh giải quyết mối lo ngại:  “Bạn có đang xây dựng nó đúng không?” | Xác thực giải quyết mối quan tâm:  “ Bạn có đang xây dựng đúng không?” |
| Đảm bảo rằng hệ thống phần mềm đáp ứng tất cả các chức năng. | Đảm bảo rằng các chức năng đáp ứng hành vi dự định. |
| Việc xác minh diễn ra trước tiên và bao gồm việc kiểm tra tài liệu, mã,… | Việc xác nhận xảy ra sau khi xác minh, chủ yếu liên quan đến việc kiểm tra tổng thể sản phẩm. |
| Do nhà phát triển thực hiện. | Thực hiện bởi người thử nghiệm. |

### Difference between QA &QC.

|  |  |  |
| --- | --- | --- |
|  | **QA (Quality Assurance)** | **QC ( Quality Control)** |
| ***Khái niệm*** | Người chịu trách nhiệm đảm bảo chất lượng sản phẩm thông qua việc đưa ra quy trình giữa các bên liên quan. | Người chịu trách nhiệm thực hiện kiểm tra chất lượng phần mềm.  Vị trí: manual QC (ko cần kỹ năng lập trình), auto QC (cần lập trình) |
| ***Nhiệm vụ*** | - Đưa ra quy trình phát triển sản phẩm.  - K.tra, audit việcc thực thi quy trình của bộ phận trong nhóm làm sản phẩm có đúng quy trình QA đề ra không.  - Nhắc nhở đội ngũ phát triển sản phẩm việc tuân thủ theo quy trình đưa ra.  - Điều chỉnh, thay đổi quy trình phù hợp. | -Tìm hiểu, phân tích tài liệu mô tả về hệ thống và thiết kế test case,và thực hiện việc test phần mềm trước khi giao cho khách hàng.  - Lên kế hoạch kiểm thử  - Viết Script cho automation test (nếu có áp dụng kiểm thử tự động)  - Sử dụng các test tool để tạo và thực hiện các test case/script chi tiết.  - Phối hợp với nhóm lập trình trong việc fix bug và báo cáo chi tiết cho Project Manager hoặc các bên liên quan khác tuỳ dự án. |
| ***Kỹ năng*** | -Hiểu sâu về kiến trúc hệ thống phần mềm.  - Kỹ năng tổ chức, tư duy logic, phân tích, làm trên số liệu tốt  - Kỹ năng giao tiếp trong team  - Hiểu rõ về chứng chỉ CMMI, ISO.. trong phần mềm để xây dựng quy trình chuẩn. | - Code  - Giám sát (nhah chóng phát hiện các lỗi)  - Giao tiếp (phản hồi với dev, pro manager)  - Quản lý nhân viên (đôn đốc nhân viên) |

## Sofware Development Life Cycle (SDLC).

### SDLC Phases.

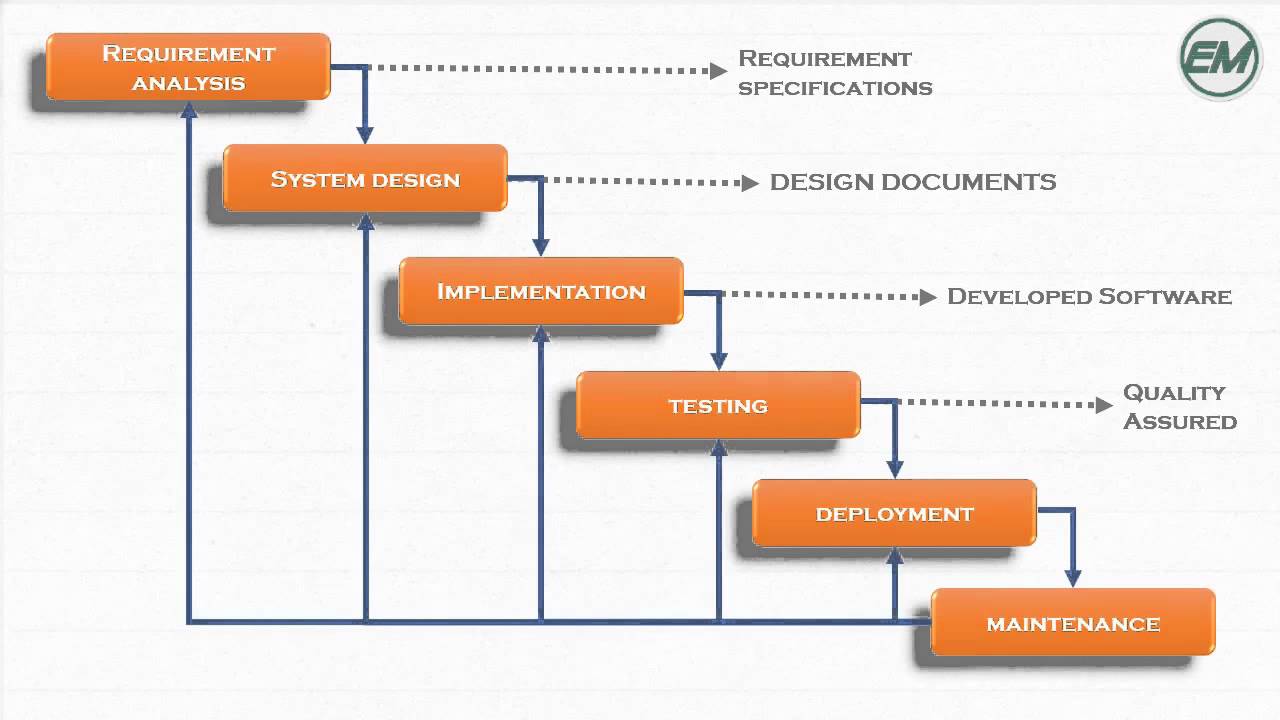


*Hình 1.2.1 SDLC Phases*

* SDLC, Vòng đời phát triển phần mềm là một quá trình được ngành công nghiệp phần mềm sử dụng để thiết kế, phát triển và kiểm tra phần mềm chất lượng cao.
* Mục tiêu của SDLC là tạo ra một phần mềm chất lượng cao đáp ứng hoặc vượt quá mong đợi của khách hàng, hoàn thành trong thời gian và chi phí ước tính.
* SDLC có 6 giai đoạn:
  + Requirement gathering and analysis
  + Design
  + Implementation or coding
  + Testing
  + Deployment
  + Maintenance

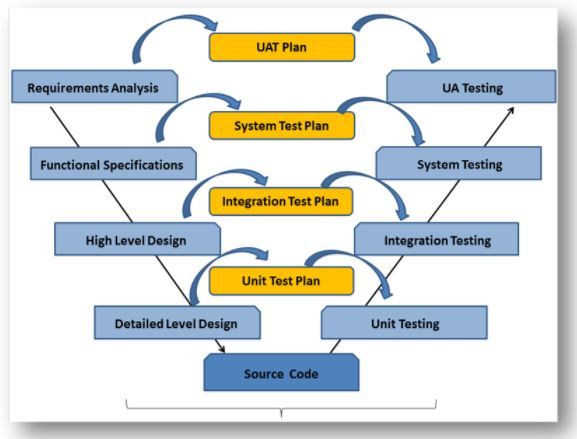
### Các mô hình phổ biến của SDLC.

* Water Fall.



*Hình 1.2.2.1 Mô hình Water Fall*

* V Model.



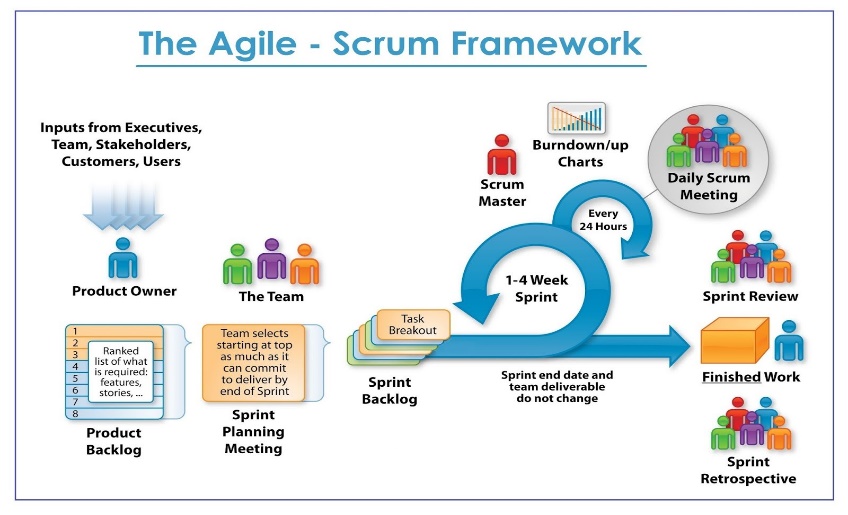
*Hình 1.2.2.2 Mô hình V Model*

* Agile Model.



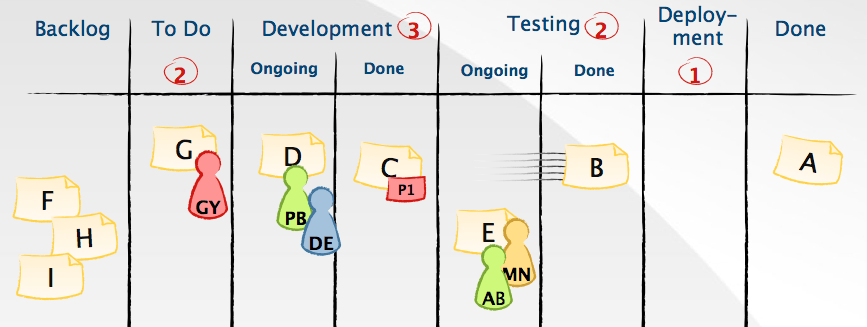
*Hình 1.2.2.3 Mô hình Agile Model*

### Scrum Methodology.



*Hình 1.2.3 Scrum Methodology*

### Kanban Methodology.



*Hình 1.2.4 Kanban Methodology*

## Sofware Testing Type.

### Manual Testing.

* Kiểm thử thủ công bao gồm kiểm tra phần mềm theo cách thủ công, tức là không sử dụng bất kỳ công cụ tự động nào hoặc bất kỳ tập lệnh nào. Trong loại này, người kiểm tra đảm nhận vai trò của người dùng cuối và kiểm tra phần mềm để xác định bất kỳ hành vi hoặc lỗi không mong muốn nào.
* Có các giai đoạn khác nhau để kiểm tra thủ công như kiểm thử đơn vị, kiểm thử tích hợp, kiểm tra hệ thống và kiểm thử chấp nhận người dùng.

### Automation Testing.

* Kiểm thử tự động, còn được gọi là Test Automation, là khi người kiểm thử viết các kịch bản và sử dụng một phần mềm khác để kiểm tra sản phẩm. Quá trình này liên quan đến việc tự động hóa quy trình thủ công. Kiểm thử tự động được sử dụng để chạy lại các kịch bản kiểm tra đã được thực hiện theo cách thủ công, nhanh chóng và lặp đi lặp lại.
* Ngoài kiểm thử hồi quy, kiểm thử tự động hóa cũng được sử dụng để kiểm tra ứng dụng từ quan điểm tải, hiệu suất và căng thẳng. Nó làm tăng phạm vi kiểm tra, cải thiện độ chính xác và tiết kiệm thời gian và tiền bạc so với kiểm tra thủ công.

## Sofware Testing Methodologies.

### Static Testing.

* Thử nghiệm tĩnh là loại kiểm tra trong đó code không được thực hiện. Nó có thể được thực hiện bằng tay hoặc bằng một bộ công cụ. Loại kiểm tra này thực hiện kiểm tra code, tìm ra lỗi, các lỗi code và mã độc tiềm ẩn trong ứng dụng phần mềm.
* Static test bắt đầu sớm hơn trong vòng đời phát triển, và do đó nó còn được gọi là thử nghiệm xác minh (verification testing).

#### *Các kỹ thuật Thử nghiệm tĩnh bao gồm:*

* Inspection
* Walk-through
* Technical reviews
* Infomal reviews

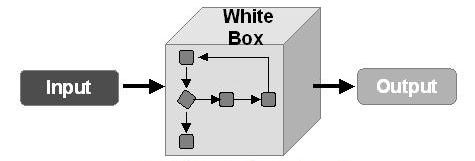
### Dynamic Testing.

* Thử nghiệm động được thực hiện khi code đang ở chế độ thực thi. thực thi chạy chương trình ứng dụng. Khi code được thực thi, thì đầu vào được truyền một giá trị, kết quả hoặc đầu ra của việc thực hiện được so sánh với kết quả dự kiến ban đầu đã đưa ra.
* Thử nghiệm dynamic còn được gọi là thử nghiệm xác nhận (Validation testing), đánh giá sản phẩm.

#### *Các loại kỹ thuật Thử nghiệm động*

* Unit Testing
* Integration Testing
* System Testing
* Acceptance Testing

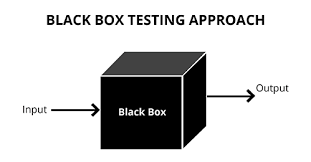
### White Box Testing.



*Hình 1.4.3 White Box Testing*

* Kiểm thử hộp trắng là việc điều tra chi tiết về logic và cấu trúc bên trong của mã. Thử nghiệm hộp trắng còn được gọi là thử nghiệm kính hoặc thử nghiệm hộp mở.
* Để thực hiện kiểm tra hộp trắng trên một ứng dụng, người kiểm tra cần biết các hoạt động bên trong của mã.

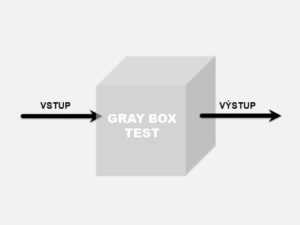
### Black Box Testing.



*Hình 1.4.4 Black Box Testing*

* Kỹ thuật kiểm tra mà không có bất kỳ kiến thức nào về hoạt động bên trong của ứng dụng được gọi là kiểm thử hộp đen
* Người kiểm tra không biết gì về kiến trúc hệ thống và không có quyền truy cập vào mã nguồn

### Gray Box Testing.



*Hình 1.4.5 Gray Box Testing*

* Sự kết hợp giữa black box test và white box test.

## Sofware Testing Levels.

### Unit Testing.

* Đơn vị là phần nhỏ nhất có thể kiểm tra được của một ứng dụng như hàm, lớp, thủ tục, giao diện. Kiểm thử đơn vị là một phương pháp mà các đơn vị mã nguồn riêng lẻ được kiểm tra để xác định xem chúng có phù hợp để sử dụng hay không.
* Các bài kiểm tra đơn vị về cơ bản được viết và thực hiện bởi các nhà phát triển phần mềm để đảm bảo rằng mã đáp ứng thiết kế và yêu cầu của nó và hoạt động như mong đợi
* Unit Test được thực hiện bằng cách sử dụng phương pháp Kiểm thử Hộp Trắng

### Intergration Testing.

* Kiểm thử tích hợp kiểm tra sự tích hợp hoặc giao diện giữa các thành phần, tương tác với các phần khác nhau của hệ thống như hệ điều hành, hệ thống tệp và phần cứng hoặc giao diện giữa các hệ thống
* Các hình thức test:
  + Bottom up:
  + Top down
  + Hybrid

### System Testing.

* Bao gồm các thử nghiệm dựa trên rủi ro và / hoặc đặc tả yêu cầu, quy trình nghiệp vụ, trường hợp sử dụng hoặc các mô tả cấp cao khác về hành vi của hệ thống, tương tác với hệ điều hành và tài nguyên hệ thống.
* Kiểm tra hệ thống thường là kiểm tra cuối cùng để xác minh rằng hệ thống được cung cấp đáp ứng các đặc điểm kỹ thuật và mục đích của nó..
* Kiểm thử hệ thống theo kiểm thử chức năng và phi chức năng:
  + Kiểm thử chức năng:
    - Requirement-based testing: Thử nghiệm dựa trên yêu cầu
    - Business-process-based testing: Thử nghiệm dựa trên quy trình kinh doanh
  + Kiểm thử phi chức năng:
    - Sanity testing
    - Re-test
    - Regression Test
    - Security Test
    - Recovery Test
    - Usability Test
    - Performance Test
    - Load Test
* Stress test

### Acceptance Testing.

* Sau khi kiểm tra hệ thống đã sửa chữa tất cả hoặc hầu hết các khiếm khuyết, hệ thống sẽ được giao cho người dùng hoặc khách hàng để kiểm tra chấp nhận.
* Kiểm tra chấp nhận về cơ bản được thực hiện bởi người dùng hoặc khách hàng mặc dù các bên liên quan khác cũng có thể tham gia.

Có 2 loại:

* + Alpha test
  + Beta test

## Test case Design Technique.

### Khái niệm.

* Trường hợp thử nghiệm là một tài liệu, trong đó có một tập hợp dữ liệu thử nghiệm, điều kiện tiên quyết, kết quả mong đợi và điều kiện hậu kỳ, được phát triển cho một kịch bản thử nghiệm cụ thể xác minh sự tuân thủ theo một yêu cầu cụ thể.
* Test Case đóng vai trò là điểm bắt đầu cho quá trình thực thi thử nghiệm và sau khi áp dụng một tập hợp các giá trị đầu vào, ứng dụng sẽ có kết quả cuối cùng và rời khỏi hệ thống tại một số điểm kết thúc hoặc còn được gọi là điều kiện hậu thực thi.

### Các loại kỹ thuật Testcase.

* Specification-based techniques.
  + Boundary Value Analysis (BVA) Phân tích giá trị biên
  + Equivalence Partitioning (EP) Phân vùng tương đương (EP)
  + Decision Table Testing Kiểm tra Bảng Quyết định
  + State Transition Diagrams Sơ đồ chuyển đổi trạng thái
  + Use Case Testing Kiểm tra trường hợp sử dụng
* Experience-based techniques.
* Error Guessing (Đoán lỗi)
* Exploratory Testing (Thử nghiệm thăm dò)

## WebDriver Commands.

* Browser Commands
* Browser Navigation Commands
* WebElement Commands
* FindElements Commands
* Check Box & Radio buttons
* Drop Down & Multiple Selects
* WebTable Handling

## Cơ hội nghề nghiệp.

* ***Có nhu cầu tuyển dụng cao:***

Nghề nghiệp ổn định, khi theo đuổi nghề tester sẽ thường xuyên được cập nhật những công nghệ mới và được tiếp xúc với các dự án khác nhau, được học được nhiều thứ mới lạ vì thế sẽ không nhàm chán.

Đặc biệt nghề này có quy định về nghề nghiệp và cơ hội thăng tiến rõ ràng: Tester càng nhiều kinh nghiệm thì lại càng được quan tâm và chú trọng.

* ***Nghề Tester không có tuổi:***

Có thể thấy rằng đối với nghề lập trình sự nhạy bén của tuổi trẻ rất quan trọng, tuy nhiên đối với nghề Tester thì kinh nghiệm đã được tích lũy nhiều năm mới là điều quan trọng nhất.

Nếu như một người Tester giỏi tiếng anh thì lại càng có nhiều cơ hội để làm ở các công ty phần mềm lớn với các dự án outsourcing của nước ngoài với mức lương rất cao.

Nhu cầu tuyển dụng cao.

Nghề nghiệp ổn định, thường xuyên được cập nhật những công nghệ mới và tiếp xúc với những dự án khác nhau, học được nhiều thứ mới lạ nên không nhàm chán.

Có quy định về nghề nghiệp và cơ hội thăng tiến rõ ràng. Tester càng nhiều kinh nghiệm càng được quan tâm và chú trọng.

* **Mô tả vị trí việc làm về testing:**
* Nghiên cứu yêu cầu và thiết kế của dự án.
* Lập kế hoạch test và kịch bản test, chuẩn bị dữ liệu test.
* Viết test case, Test các dự án phần mềm.
* Quản lý, phân tích và theo dõi kết quả test, báo cáo kết quả test.
* Thực hiện các công việc khác theo sự phân công của Quản lý trực tiếp.
* **Các kỹ năng cần có:**
* Rèn luyện khả năng sắp xếp công việc.
* Viết báo cáo lỗi chi tiết.
* Viết test case phù hợp.
* Giao tiếp với đồng nghiệp.
* Tự đặt câu hỏi.
* **Mức lương khởi điểm dành cho Fresher.**
* Với sinh viên mới ra trường chưa có kinh nghiệm làm việc, mức lương sẽ tương đối thấp. Trung bình tiền lương sẽ dao động từ **4 – 8 triệu đồng**.
* Tuy nhiên, đó chỉ là mức khởi điểm ban đầu cho vị trí nhân viên thử việc hoặc thực tập sinh. Khi trở thành nhân viên chính thức mức lương sẽ được điều chỉnh theo quyết định của Ban giám đốc.

# TỔNG QUAN HỆ THỐNG CẦN TEST

## Tổng quan về hệ thống

Kể từ tháng 3/2020, việc bùng phát dịch bệnh viêm đường hô hấp cấp do chủng mới của vi-rút Corona (COVID-19) đã được tuyên bố là tình trạng y tế công cộng khẩn cấp gây quan ngại quốc tế và loại vi-rút này đã lan rộng ra nhiều quốc gia và vùng lãnh thổ, trong đó có Việt Nam. Chính vì vậy, nhằm mục đích kiểm soát bệnh dịch Covid lây lan ở các tỉnh khắp Việt Nam, VnEpress đã thiết lập trang website “Số liệu Covid-19 tại Việt Nam” cập nhật thông tin, số liệu tình hinh dịch tại các thành phố ở Việt Nam.

Các tác nhân chính của hệ thống bao gồm: Quản trị viên và người dùng.

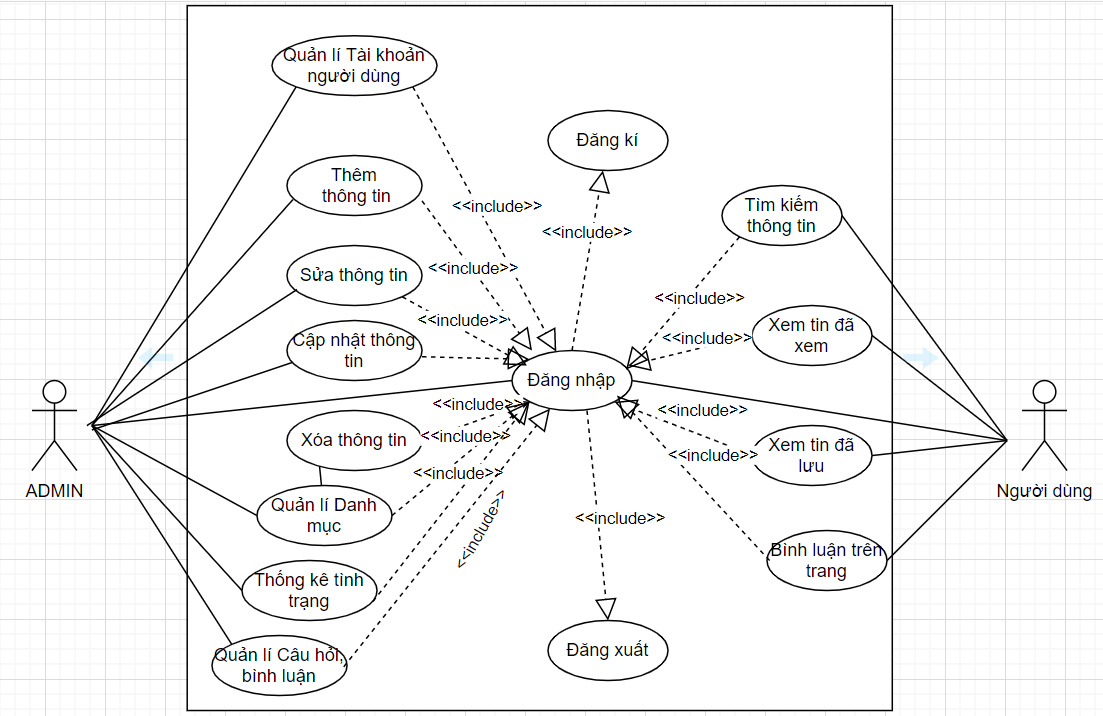
Quản trị viên (Admin) có chức năng quản lý:

* Các tài khoản người dùng
* Cập nhật, thêm, sửa, xóa thông tin, số liệu tình hình dịch bệnh
* Thống kê tình trạng: Nhiễm, Khỏi, Tử vong, Đang điều trị trong cả nước
* Các danh mục
* Các câu hỏi, bình luận

Người dùng có quyền:

* Đăng kí, đăng nhập tài khoản
* Xem tin đã lưu, tin đã xem
* Tìm kiếm thông tin
* Bình luận trên trang

## Sơ đồ hệ thống- USE CASE.



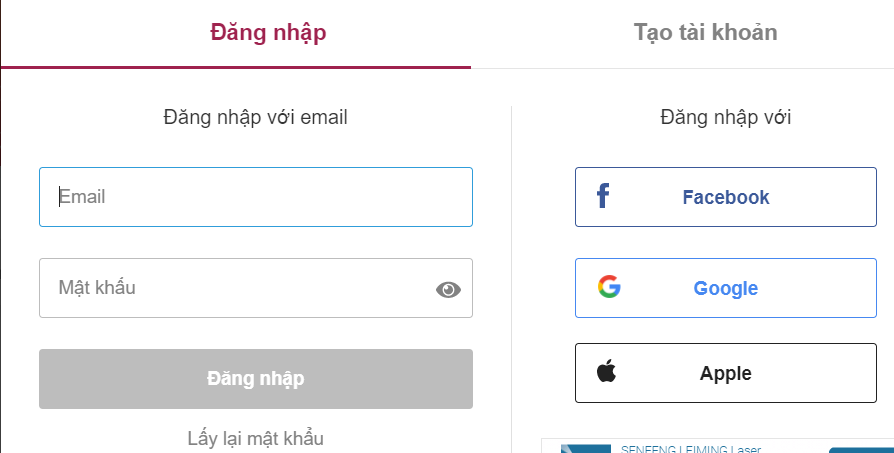
## Đặc tả hệ thống.

### Giao diện người dùng.

2.3.1.1 Đăng nhập.

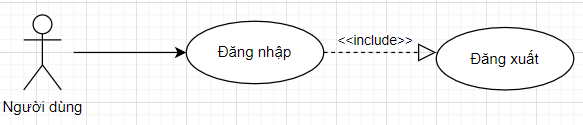
Đặc tả màn hình

* Khi người dùng truy cập vào trang web, hệ thống sẽ hiển thị như hình 2.3.1.1, từ đây người dùng sẽ có thể đăng nhập với nhiều tài khoản như Email, FaceBook, Google, Apple.



Hình 2.3.1.1 Đăng nhập trang web

* Usecase Đăng nhập

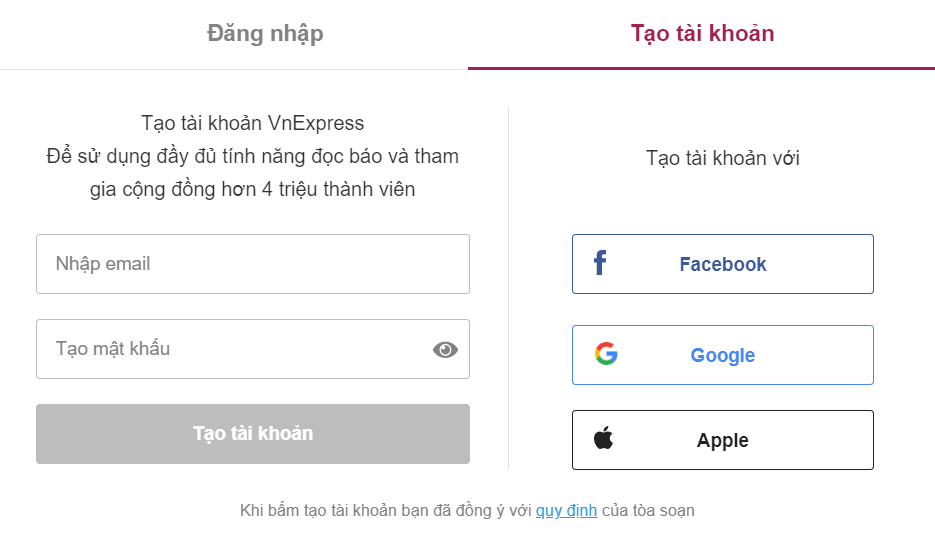


* Đặc tả usecase đăng nhập.

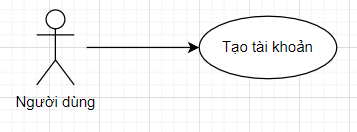
|  |  |
| --- | --- |
| Usecasse | Nội dung |
| Tên | Đăng nhập |
| Mô tả | Usecase cho phép người dùng đăng nhập vào hệ thống để thực hiện chức năng của mình. |
| Actor | Admin và người dùng |
| Điều kiện kích hoạt | Khi người dùng mở ứng dụng |
| Tiền điều kiện | Người dùng phải có tài khoản trên hệ thống |
| Hậu điều kiện | Người dùng đăng nhập thành công |
| Luồng sự kiện chính | 1.Hệ thống hiển thị màn hình đăng nhập.  2. Người dùng đăng nhập bằng email có sẳn hoặc dùng tài khoản Facebook, Google, Apple  3. Nếu đăng nhập thành công màn hình hiển thị trang Số liệu Covid-19 tại Việt nam.  4. Kết thúc Usecase. |

2.3.1.2 Tạo tài khoản.

Đặc tả giao diện Tạo tài khoản.



* Usecase Tạo tài khoản.

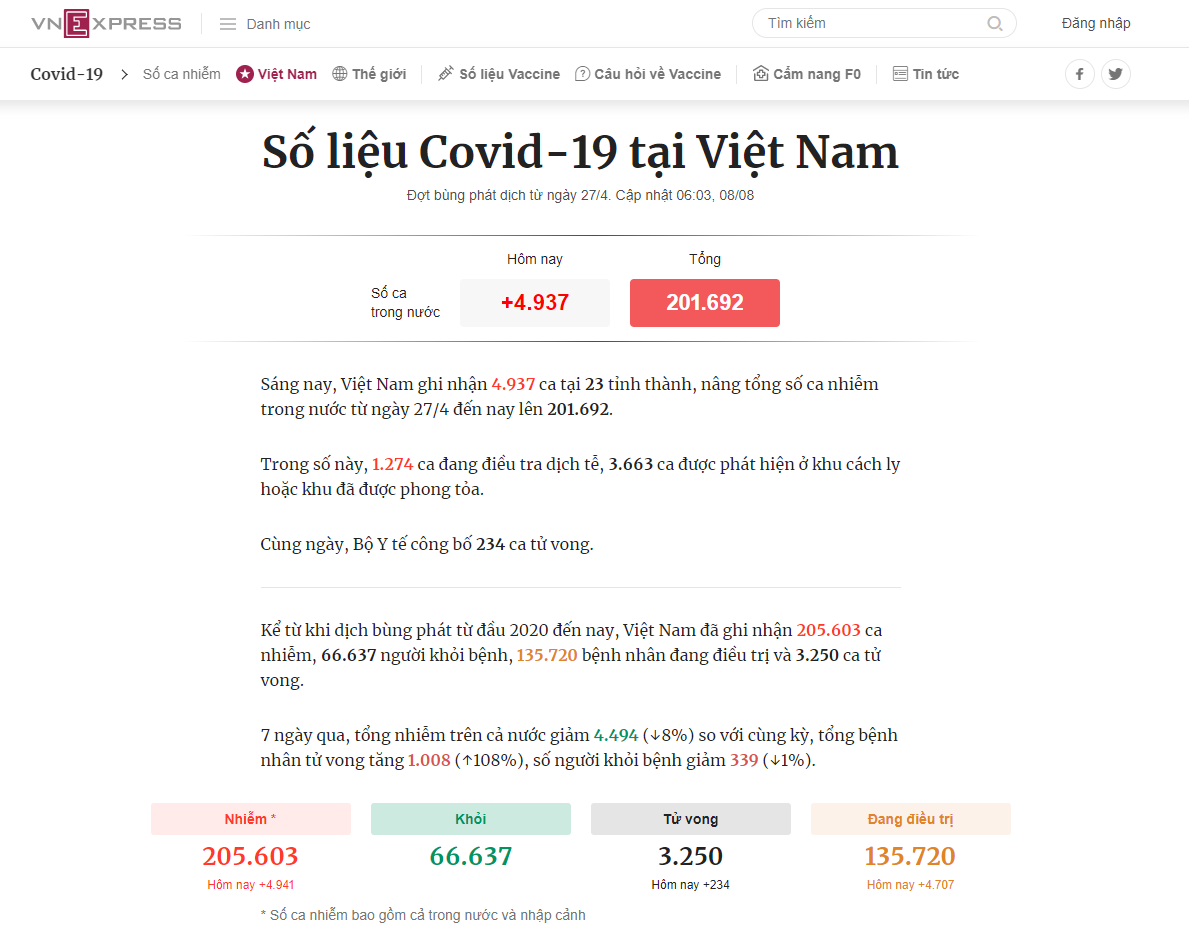


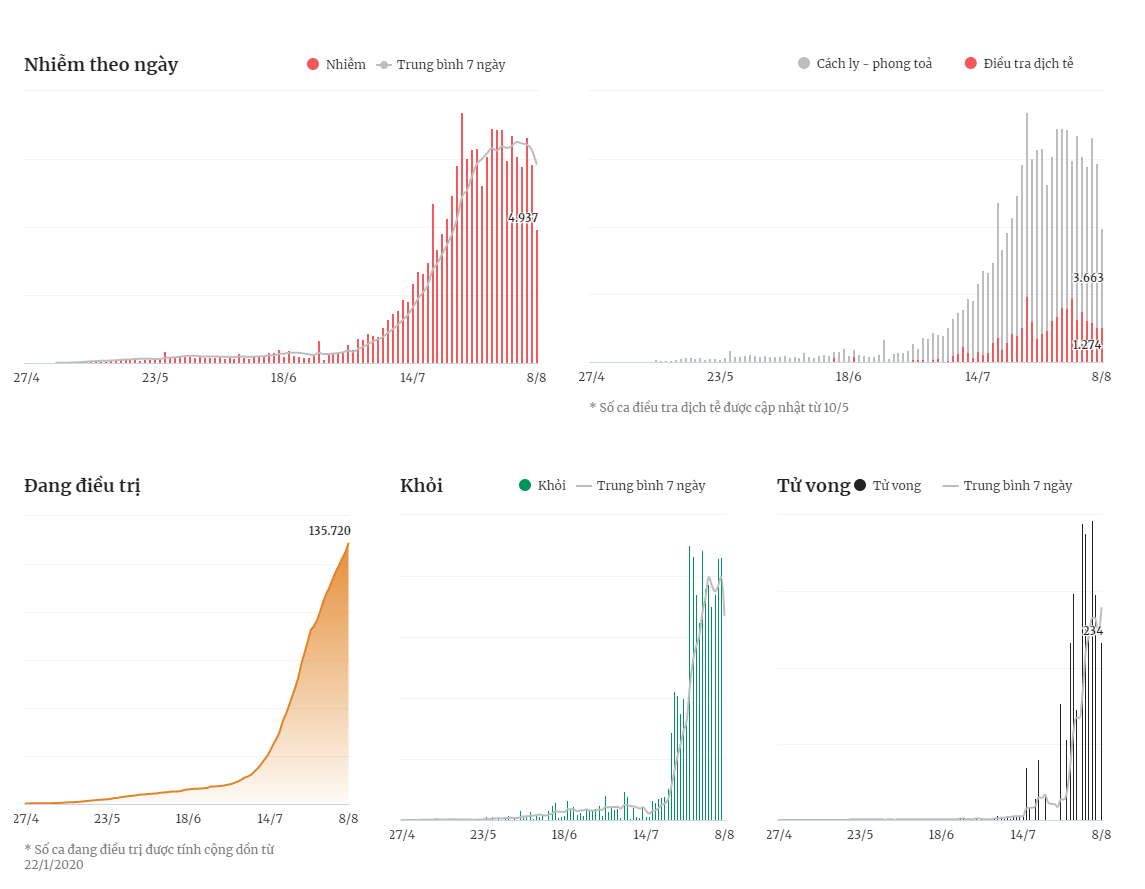
* Đặc tả usecase Tạo tài khoản.

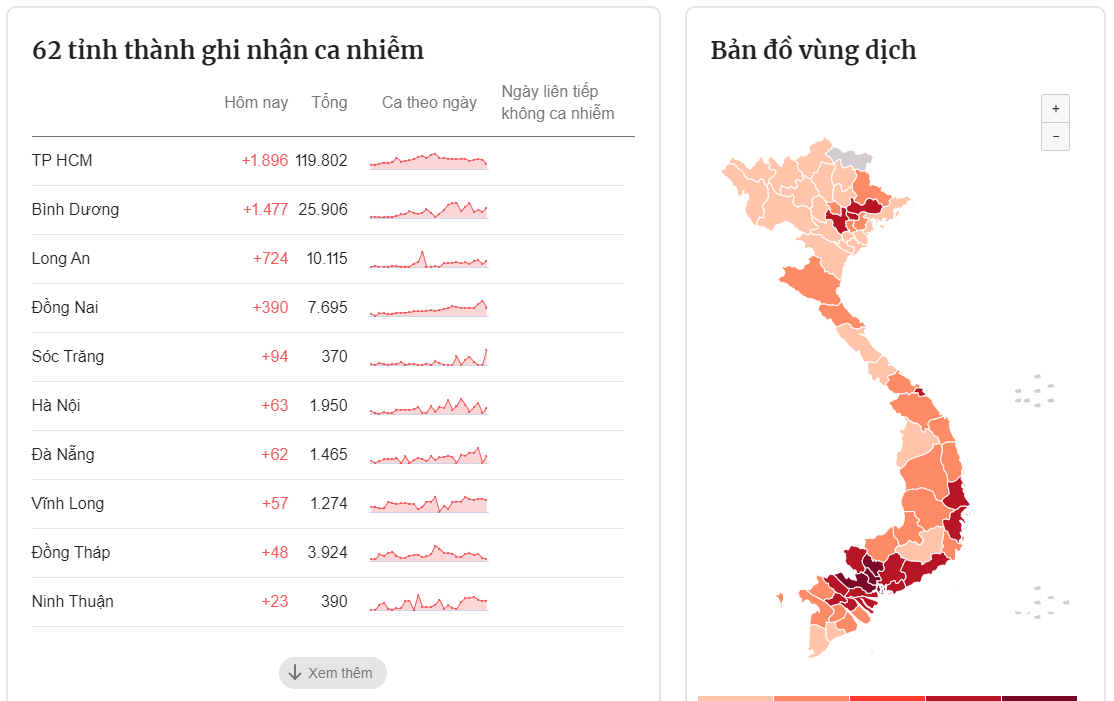
|  |  |
| --- | --- |
| Usecase | Nội dung |
| Tên Usecase | Tạo tài khoản |
| Mô tả | Usecase cho phép người dùng tạo tài khoản hệ thống để thực hiện chức năng của mình. |
| Actor | Người dùng |
| Điều kiện kích hoạt | Khi người dùng mở hệ thống |
| Tiền điều kiện | Người dùng truy cập vào hệ thống |
| Hậu điều kiện | Người dùng tạo tài khoản thành công |
| Luồng sự kiện chính | 1.Hệ thống hiển thị màn hình tạo tài khoản.  2. Người dùng nhập email và tạo mật khẩu nhấn nút tạo tài khoản.  3. Tạo tài khoản thành công.  4. Kết thúc usecase. |

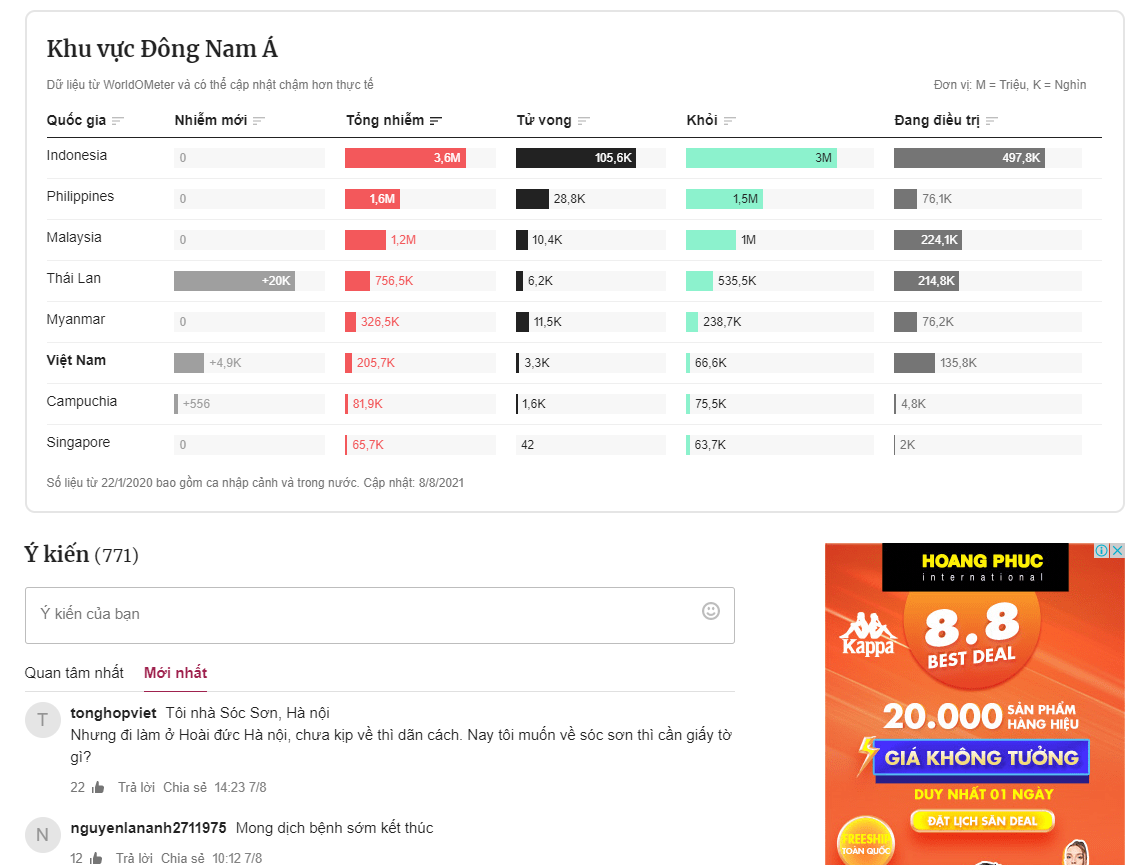
2.3.2 Trang Co-vid 19 Việt Nam.

- Đặc tả màn hình trang.

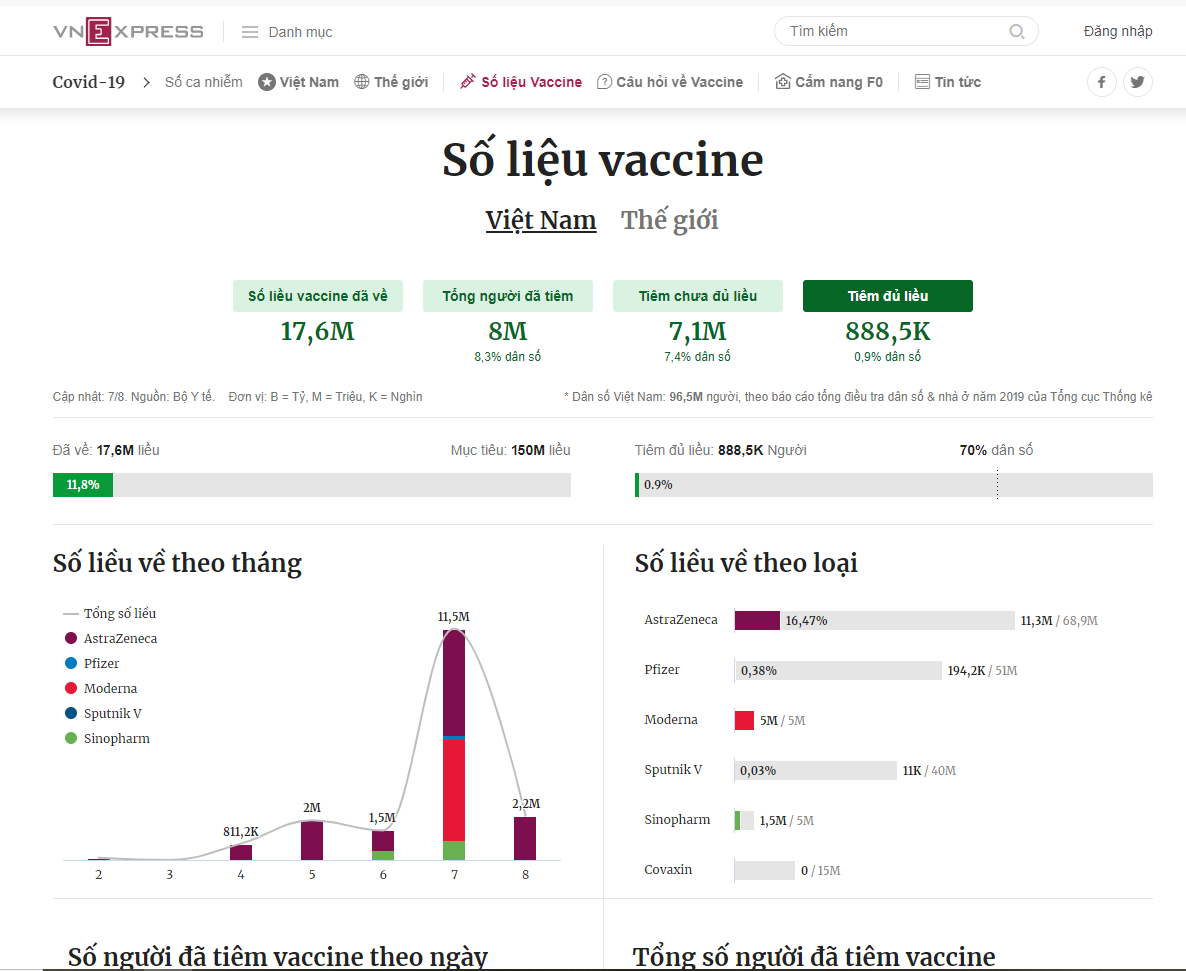


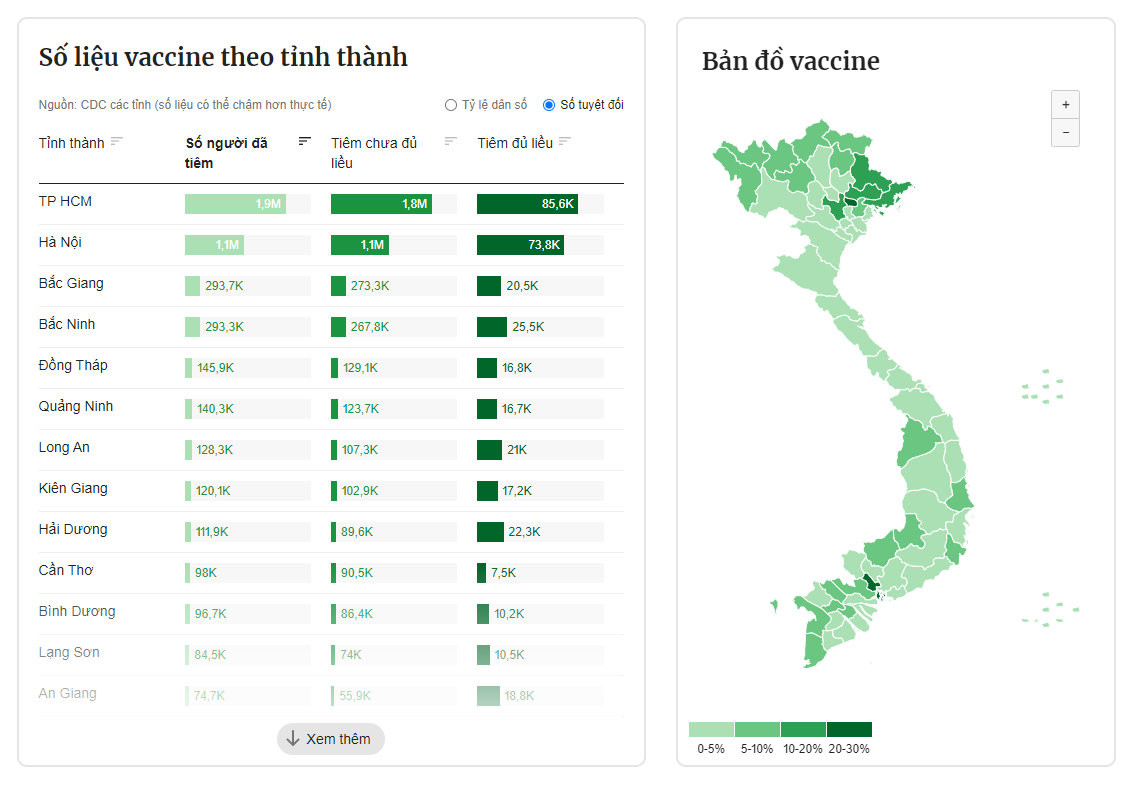






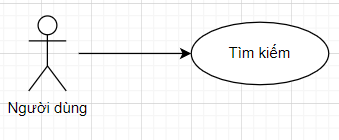
* + 1. Trang Số liệu Vaccine.
* Đặc tả màn hình trang:





* + 1. Đặc tả chức năng Tìm kiếm.

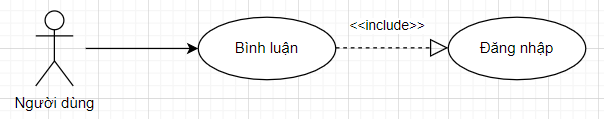
Usecase Tìm kiếm.



Đặc tả usecase Tìm kiếm.

|  |  |
| --- | --- |
| Usecase | Nội dung |
| Tên | Tìm kiếm |
| Mô tả | Usecase cho phép người dùng thực hiện chức năng tìm kiếm thông tin |
| Actor | Người dùng |
| Điều kiện kích hoạt | Người dùng có thể đăng nhập hoặc không đăng nhập vào hệ thống. |
| Hậu điều kiện | Người dùng tìm kiếm được thông tin cần xem trên hệ thống. |
| Luồng sự kiện chính | 1.Hệ thống hiển thị mục Tìm kiếm bên phải màn hình.  2. Người dùng muốn tìm kiếm nhấp chuột và gõ nội dung tìm kiếm.  3. Kết thúc usecase. |

* + 1. Đặc tả chức năng Bình luận (Ý kiến).
* Usecase Bình luận.



Đặc tả usecase Bình luận.

|  |  |
| --- | --- |
| Usecase | Nội dung |
| Tên | Bình luận |
| Mô tả | Usecase cho phép người dùng bình luận trên trang. |
| Actor | Người dùng |
| Điều kiện kích hoạt | Người dùng đăng nhập hệ thống |
| Tiền điều kiện | Người dùng phải có tài khoản trên hệ thống |
| Hậu điều kiện | Người dùng có thê đăng và xem nội dung bình luận trên trang |
| Luồng sự kiện chính | 1.Hệ thống hiển thị màn hình Bình luận cuối trang.  2. Người dùng gõ nội dung Bình luận trên trang sau khi đăng nhập thành công.  3. Kết thúc Usecase |

# THIẾT KẾ TESTCASE VÀ THỰC HIỆN TEST.

## Kế hoạch kiểm thử.

### Yêu cầu kiểm thử.

Tiến hành kiểm thử sẽ phải đáp ứng những yêu cầu tuy cập sau

• Thiết bị: Laptop, Desktop

• Browser: Google Chrome Version 92.0.4515.131

• Automation test tool: Selenium (Java language)

* Truy cập vào Eclipse: Viết test case và thực thi kiểm thử.

• Access to the website: <https://vnexpress.net/covid-19/covid-19-viet-nam>

### Công cụ kiểm thử.

|  |  |
| --- | --- |
| Giai đoạn | Công cụ |
| Tạo test case | Selenium Web driver |
| Thực thi kiểm thử | Tự động: Selenium Web driver |
| Quản lý test case | Selenium Web driver |
| Quản lý và theo dõi lỗi | * Selenium Web driver * Eclipse |

### Trạng thái của testcase.

• Successfull: Nhập nội dung này nếu testcase đã được thực hiện và kết quả đã kiểm thử đúng như kết quả mong đợi

• Failed: Nhập nội dung này nếu testcase đã được thực hiện và kết quả đã kiểm thử không đúng như kết quả mong đợi

• Pending: Nhập nội dung này nếu test case đã được thực hiện, tuy nhiên còn chưa kết luận được là Passed hay Failed do đang chờ xác nhận với các bên liên quan.

• N/A: Nhập nội dung này nếu test case thuộc đối tượng cần phải thực hiện trong đợt kiểm thử, tuy nhiên vì lý do thiếu môi trường kiểm thử, thiếu thiết bị ... nên đã không thể thực hiện được

### Độ ưu tiên của lỗi.

• Low: Bug không cần sửa ngay, có thể sửa sau khi các bug High và Nomal đã được sửa hết

• Nomal: Bug có thể được sửa trong lần cập nhật phiên bản sau.

• High: Sửa ngay khi sửa các lỗi có độ ưu tiên Immediate và Urgent.

• Urgent: Sửa gấp. Sửa ngay sau khi sửa các lỗi có độ nghiêm trọng Immediate.

• Immediate: Sửa ngay lập tức. Sửa ngay trong vòng 24 giờ kể từ khi phát hiện lỗi.

### Giả định.

* Kiểm thử nên được hoàn thành đúng thời hạn.

• Tài liệu, tài nguyên và quyền hạn của nhóm kiểm thử luôn có sẵn và được cung cấp đầy đủ.

• Tất cả các yêu cầu bao gồm yêu cầu kinh doanh, yêu cầu hệ thống và yêu cầu dữ liệu sẽ được hoàn thành, cập nhật và quản lý khi có sự thay đổi.

• Kế hoạch kiểm thử sẽ bao gồm một mức độ chi tiết cần thiết để cho phép chúng được sử dụng lại cho các giai đoạn thử nghiệm khác nhau.

• Các nguồn lực phù hợp hỗ trợ trong việc tạo ra các điều kiện thử nghiệm, script và thực hiện thử nghiệm.

• Nhóm kiểm thử sẽ cung cấp test case và kết quả kiểm thử đáp ứng đúng kế hoạch. • Nhóm kiểm thử có thể yêu cầu các sản phẩm thử nghiệm hệ thống sau đây: lập kế hoạch, chiến lược kiểm thử để có thể xem xét chất lượng và kết quả kiểm tra hệ thống nếu leader kiểm tra cảm thấy cần thiết và hữu ích

### Phụ thuộc

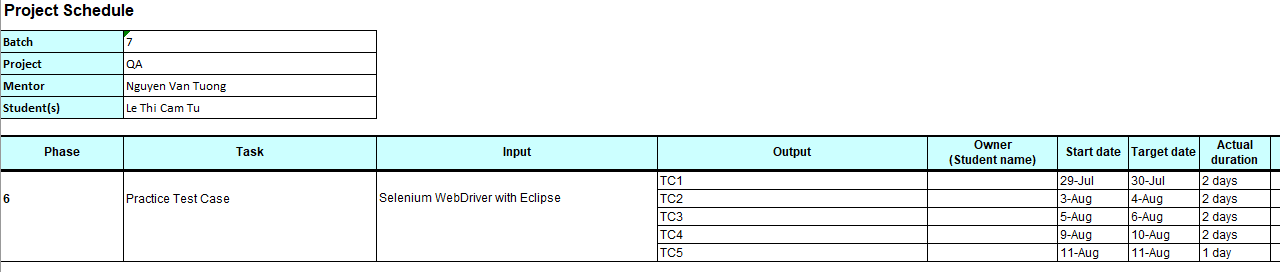
Không thể thực hiện kiểm thử nếu

- Test case chưa hoàn thành.

- Hệ thống chưa sẵn sàng

### Test Plan của từng testcase.

Testcase 1 (TC1), Testcase 2 (TC2),…



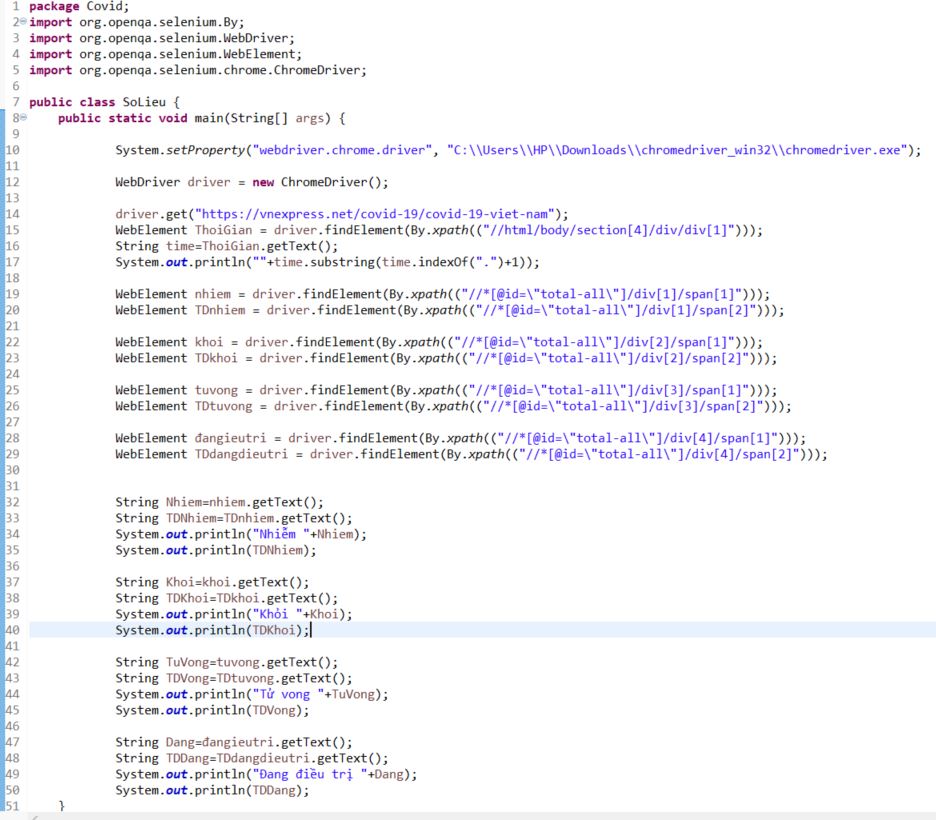
## Thực hiện kiểm thử.

### Testcase 1:

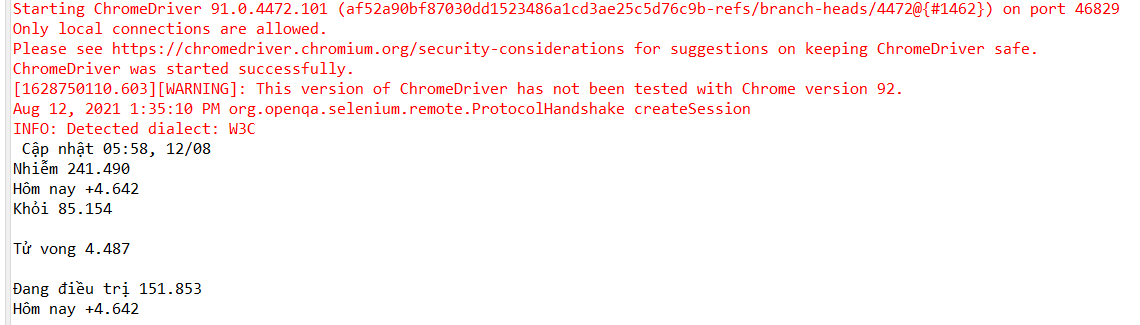
* Tại trang “Việt Nam” lấy số liệu ngày cập nhật và số ca Nhiễm, Khỏi, Tử Vong, Đang điều trị ( số ca nhiễm cả nước).

### 



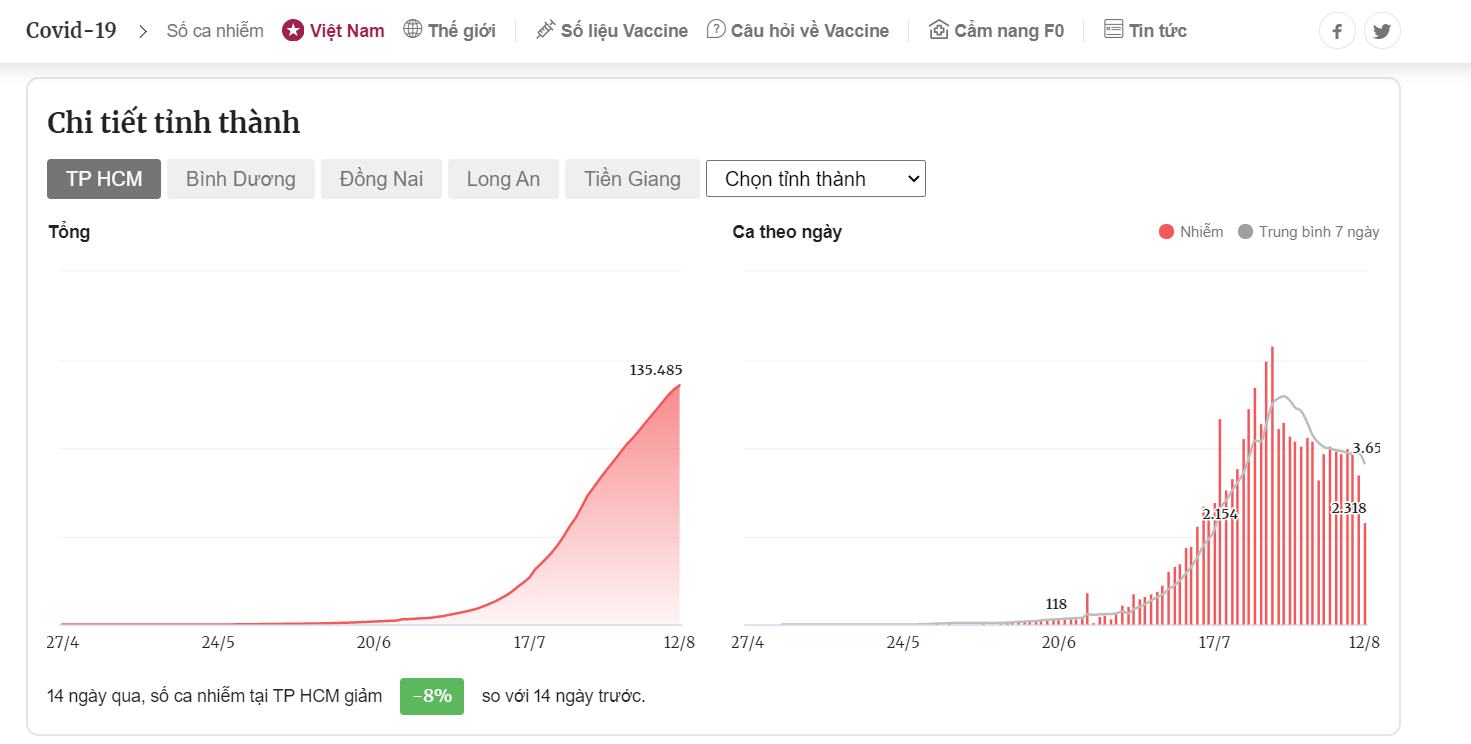
***Case:***

***Kết quả hiển thị khi chạy xong script case:***



### Testcase 2:

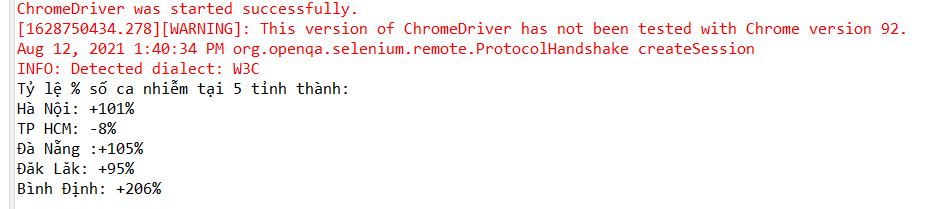
* Chọn 5 tỉnh ( trong đó 3 tỉnh bắt buộc là TPHCM, Đà Nẵng, Hà Nội, và 2 tỉnh khác chọn bất kì). In ra số liệu số ca nhiễm 14 ngày gần nhất ( lấy tỉ lệ %).



***Case:***

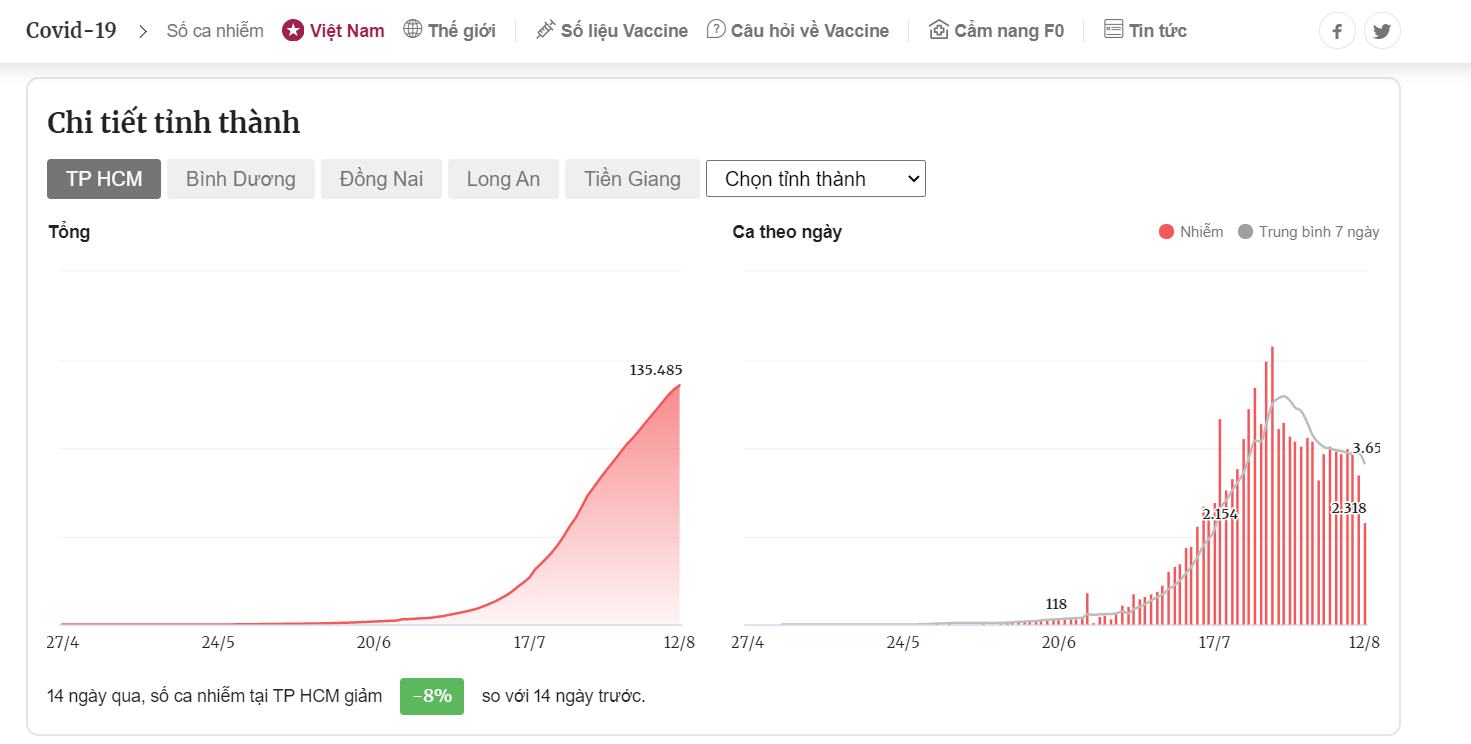
**

***Kết quả hiển thị khi chạy xong script case:***



### Testcase 3:

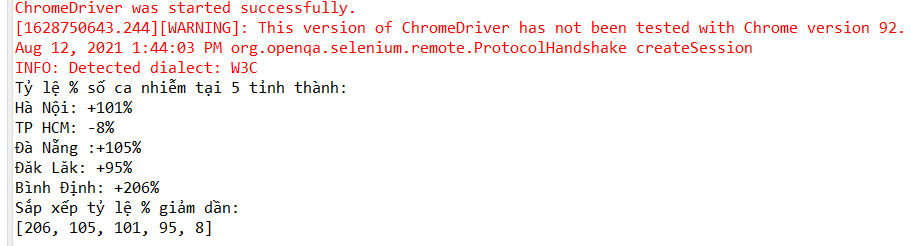
* Từ số liệu % số ca nhiễm ở Testcase 2. So sánh tỷ lệ % của 5 tỉnh đó và sắp xếp theo thứ tự giảm dần.



***Case:***

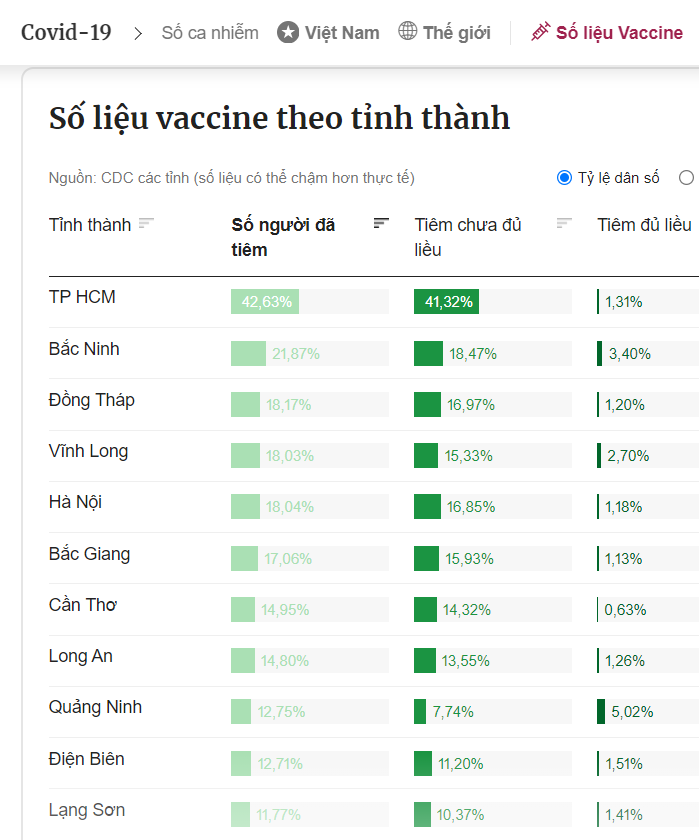
**

***Kết quả hiển thị khi chạy xong script case:***

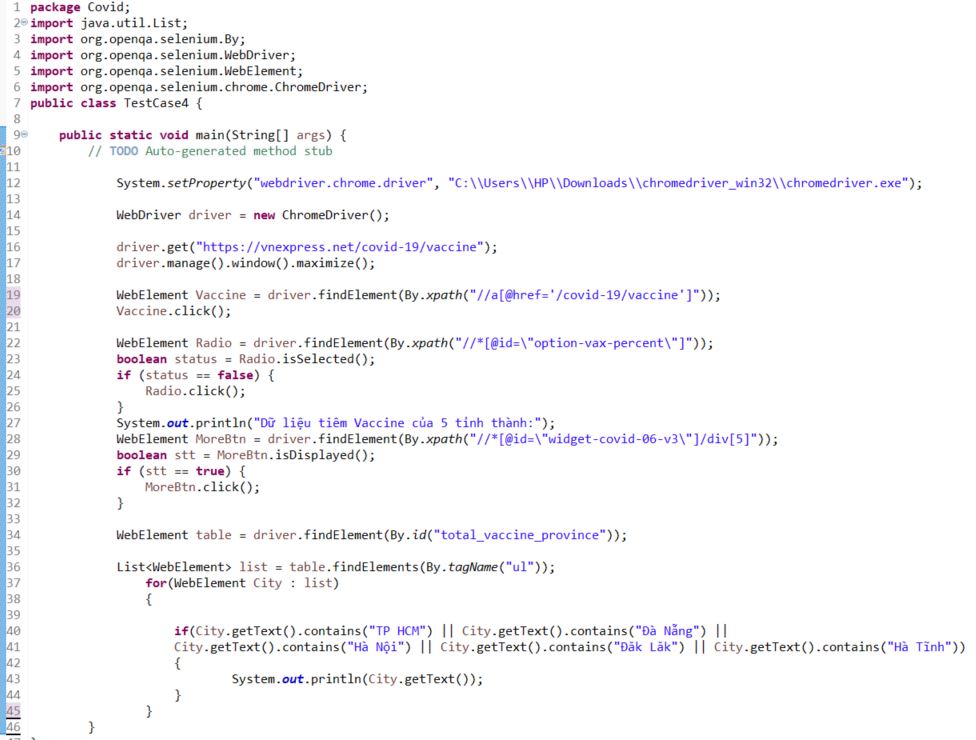


### Testcase 4:

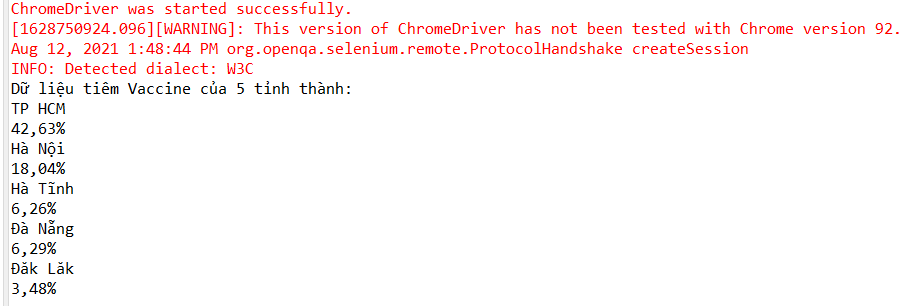
* Tại trang “ Số liệu vaccine” – Tỷ lệ dân số. Lấy dữ liệu “số người đã tiêm” của 5 tỉnh đã chọn ở Testcase 2.



***Case:***

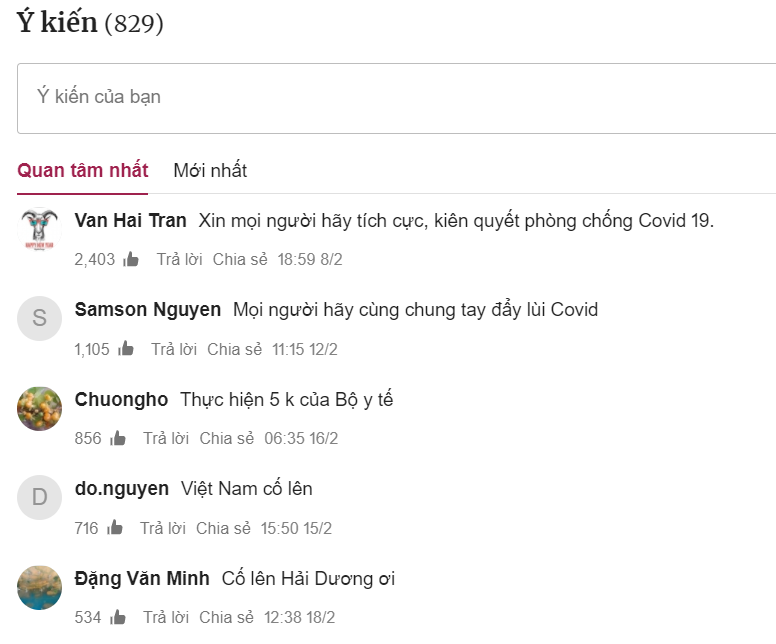
**

***Kết quả hiển thị khi chạy xong script case:***

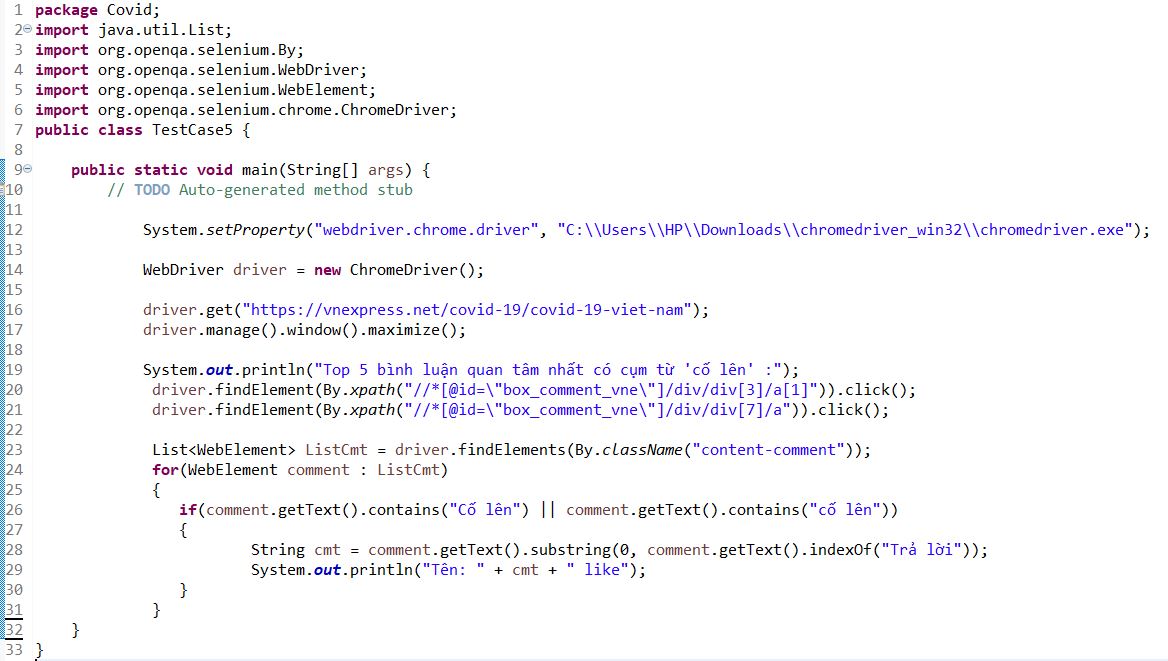


### Testcase 5:

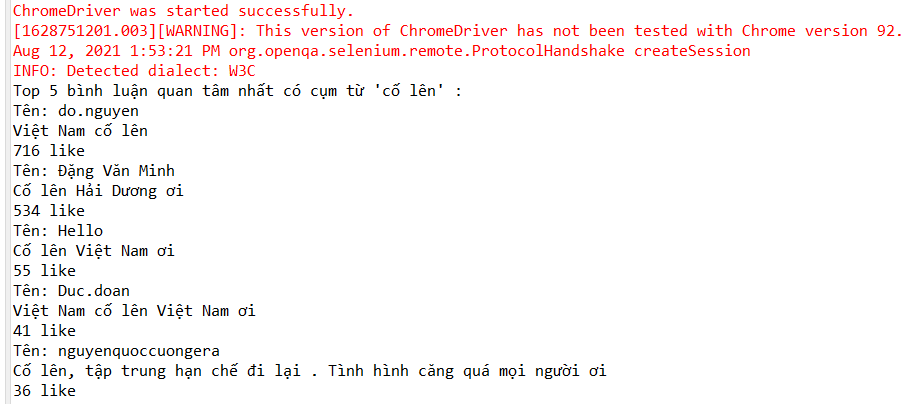
* Tại mục “Ý kiến” – “Quan tâm nhất”. Kiểm tra xem comment có chữ “cố lên” và in ra kết quả sắp xếp từ trên xuống dưới theo số lượt “like” nhiều nhất. In ra Tên người comment + số lượng like + nội dung comment.



***Case:***

**

***Kết quả hiển thị khi chạy xong script case:***



# TÀI LIỆU THAM KHẢO

<https://www.tutorialspoint.com/software_testing/index.htm>

<http://tryqa.com/>

<https://www.guru99.com/>

<https://www.toolsqa.com/selenium-webdriver/>