

**XÂY DỰNG WEBSITE BÁN ÁO SƠ MI BEE SHIRT**

**KẾ HOẠCH KIỂM THỬ**

| **Mã dự án** | **SD-19** |
| --- | --- |
| **Mã tài liệu** | **TP\_V1.0** |
| **Ngày** | **10/01/2024** |

**Nhóm: SD-19**

**Tên thành viên Vũ Nguyên Hạ - PH29444**

**Vũ Thế Long - PH29309**

**Dương Việt Hùng - PH29421**

**Ma Thu Thủy - PH29502**

**Phạm Văn Bình - PH29510**

**Phạm Văn Hiếu - PH29564**

**Hà Nội - 01-2024**

**BẢN GHI NHẬN THAY ĐỔI TÀI LIỆU**

| **Ngày thay đổi** | **Vị trí thay đổi** | **Lý do** | **Nguồn gốc** | **Phiên bản cũ** | **Mô tả thay đổi** | **Phiên bản mới** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**TRANG KÝ**

**Người lập:** <Ngày>

<Chức vụ>

**Người xem xét:** <Ngày>

<Chức vụ>

<Ngày>

<Chức vụ>

**Người phê duyệt:** <Ngày>

<Chức vụ>

**MỤC LỤC**

[**1** **GIỚI THIỆU 5**](#_heading=h.1fob9te)

[1.1 Mục đích 5](#_heading=h.3znysh7)

[1.2 Thông tin chung 5](#_heading=h.2et92p0)

[1.3 Tài liệu liên quan 5](#_heading=h.tyjcwt)

[1.4 Phạm vi test 5](#_heading=h.3dy6vkm)

[1.5 Ràng buộc 6](#_heading=h.1t3h5sf)

[1.6 Liệt kê các mạo hiểm 6](#_heading=h.4d34og8)

[**2** **CÁC YÊU CẦU CHO Test 7**](#_heading=h.2s8eyo1)

[**3** **CHIẾN LƯỢC TEST 7**](#_heading=h.3rdcrjn)

[3.1 Các kiểu test 7](#_heading=h.26in1rg)

[3.1.1 Test chức năng (Functional Testing) 11](#_heading=h.lnxbz9)

[3.1.2 Test hiệu suất (Performance testing) 14](#_heading=h.2jxsxqh)

[3.1.3 Test Bảo mật và Kiểm soát truy cập (Security and Access Control Testing) 18](#_heading=h.4i7ojhp)

[3.1.4 Test hồi quy (Regression Testing) 19](#_heading=h.2xcytpi)

[3.2 Giai đoạn test](#_heading=h.1ci93xb) 20

[3.3 Các công cụ test 20](#_heading=h.3whwml4)

[3.4 Môi trường test 21](#_heading=h.2bn6wsx)

[**4** **TÀI NGUYÊN 21**](#_heading=h.qsh70q)

[4.1 Nhân lực 21](#_heading=h.3as4poj)

[4.2 Hệ thống 21](#_heading=h.1pxezwc)

[**5** **CÁC MỐC KIỂM SOÁT CỦA GIAI ĐOẠN TEST (TEST MILESTONES) 22**](#_heading=h.49x2ik5)

[**6** **CÁC SẢN PHẨM 22**](#_heading=h.2p2csry)

# 

# GIỚI THIỆU

## Mục đích

* Mục đích của Website bán áo sơ mi Bee Shirt:
  + Tinh giản các thủ tục phức tạp, dễ dàng sử dụng đối với người dùng.
  + Hỗ trợ việc thống kê nhanh chóng và chính xác.
  + Tạo nên trải nghiệm mua sắm trực tuyến thân thiện và thuận tiện.
  + Dễ dàng tương tác và hỗ trợ với khách hàng.
  + Rút ngắn thời gian khi đặt hàng. Giúp các cửa hàng và doanh nghiệp tối ưu hóa quy trình kinh doanh và tăng hiệu suất.
* Tài liệu này được biên soạn với những tiêu chí sau:
  + Mô tả cách thức thực hiện công việc kiểm tra phần mềm đầy đủ và tốt nhất.
  + Đặc tả những module cần kiểm tra trong Website bán áo sơ mi dựa vào mục đích của phần mềm cũng như những chức năng được hiện thực trong phần mềm.
  + Phân công cụ thể từng module cần kiểm tra cho thành viên thích hợp trong nhóm cũng như lên kế hoạch từng bước cho từng cá nhân.
  + Đề ra những mức tiêu chuẩn có thể chấp nhận để kết luận kết quả kiểm tra.
  + Đây là công cụ giúp cho việc truyền thông dễ dàng giữa nhóm kiểm tra và nhóm phát triển phần mềm.
* Tài liệu bao gồm các phần sau:
  + Giới thiệu: giúp cho thành viên có cái nhìn chung nhất và bao quát nhất về kế hoạch kiểm tra Website bán áo sơ mi Bee Shirt.
  + Các yêu cần cho test: xác định các thành phần (tình huống test, các yêu cầu chức năng và phi chức năng) được xác định như mục tiêu test. Các thành phần liệt kê trong danh sách này sẽ được test.
  + Chiến lược test: Chiến lược test giới thiệu phương án tiếp cận để test các mục tiêu test:
    - Những vấn đề chính trong chiến lược test là các kỹ thuật được áp dụng và điều kiện để biết khi nào việc test được hoàn thành.
    - Mô tả các kiểu test dùng trong dự án.
    - Liệt kê với mỗi kiểu test tương ứng test cho chức năng nào.
    - Việc test có thể dừng khi nào.
  + Tài nguyên : bao gồm toàn nguồn nhân lực và tài nguyên hệ thống phục vụ cho quá trình kiểm tra.
  + Các mốc kiểm soát của giai đoạn test: có thể độc lập với các mốc kiểm soát của dự án, cho biết chính xác thông tin về tình trạng hoàn thành của dự án.

## Thông tin chung

Mục đích của việc kiểm tra Website bán áo sơ mi là để đảm bảo rằng phần mềm đó hoạt động một cách chính xác và đáp ứng các yêu cầu của người dùng. Việc kiểm tra phần mềm giúp xác định các lỗi và vấn đề về tính năng, tốc độ và độ bền, giúp phục hồi lỗi và cải thiện chất lượng phần mềm. Việc kiểm tra phần mềm còn giúp đảm bảo rằng phần mềm đáp ứng các tiêu chuẩn quốc tế về bảo mật và riêng tư.

Phạm vi kiểm tra bao gồm các mục sau :

* Test chức năng ( Function Testing ) bao gồm:
  + Test chức năng (Function Testing)
  + Test giao diện người sử dụng (User Interface Testing)
  + Test dữ liệu và tích hợp dữ liệu (Data and Database Integrity Testing)
  + Test chu trình nghiệp vụ (Business Cycle Testing)
* Test hiệu xuất (Performance testing) bao gồm:
  + Performance Profiling
  + Load Testing
  + Stress Testing
  + Volume Testing
* Test Bảo mật và Kiểm soát truy cập (Security and Access Control Testing)
* Test hồi quy (Regression Testing)

## Tài liệu liên quan

| **STT** | **Tên tài liệu** | **Nguồn** | **Ghi chú** |
| --- | --- | --- | --- |
| 1 | Tài liệu kiểm tra (Test Document) | Google |  |
| 2 | Kế hoạch kiểm tra sản phẩm (Product Test Plan) | Google |  |

## Phạm vi test

Các giai đoạn kiểm tra được thực hiện: (Khái quát định nghĩa từng mức độ trong các giai đoạn, thành viên cần phải nắm rõ để biết được quy trình kiểm tra phần mềm bán điện thoại sẽ được diễn ra như thế nào)

* Unit Test – kiểm thử mức đơn vị:
  + Mục đích của Unit Test là bảo đảm thông tin được xử lý và xuất (khỏi Unit) là chính xác, trong mối tương quan với dữ liệu nhập và chức năng của từng đơn vị thành phần nhỏ nhất của phần mềm.
  + Kiểm tra từng đơn vị thành phần nhỏ nhất của Website bán áo sơ mi Bee Shirt gồm: các chức năng, các hàm, các lớp, các module, hoặc bất kỳ đơn vị nào được xác định để kiểm tra riêng biệt và đảm bảo chúng hoạt động đúng theo yêu cầu và mong đợi.
  + Một kinh nghiệm đúc kết từ thực tiễn: thời gian tốn cho Unit Test sẽ được đền bù bằng việc tiết kiệm rất nhiều thời gian và chi phí cho việc kiểm thử và sửa lỗi ở các mức kiểm thử sau đó do đó chúng ta sẽ cố gắng thực hiện Unit Test thật tốt.
  + Vì Unit Test thường thường do lập trình viên thực hiện trong giai đoạn viết code và xuyên suốt chu kỳ phát triển phần mềm. Do đó, Unit Test đòi hỏi kiểm thử viên có kiến thức về thiết kế và code của chương trình.
  + Unit Test cũng đòi hỏi phải chuẩn bị trước các tình huống (test case) hoặc kịch bản (script), trong đó chỉ định rõ dữ liệu vào, các bước thực hiện và dữ liệu mong chờ sẽ xuất ra. Các test case và script này nên được giữ lại để tái sử dụng.
* Integration Test – kiểm thử tích hợp:
  + Integration test kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành.
  + Integration Test có 2 mục tiêu chính:
    - Phát hiện lỗi giao tiếp xảy ra giữa các Unit
    - Tích hợp các Unit đơn lẻ thành các hệ thống nhỏ (subsystem) và cuối cùng là nguyên hệ thống hoàn chỉnh (system) chuẩn bị cho kiểm thử ở mức hệ thống (System Test)
  + Integration Test chỉ nên thực hiện trên những Unit đã được kiểm tra cẩn thận trước đó bằng Unit Test, và tất cả các lỗi mức Unit đã được sửa chữa.
  + Có 4 loại kiểm thử trong Integration Test:
    - Kiểm thử cấu trúc (Structure Test): Tương tự White Box Test
    - Kiểm thử chức năng (Functional Test): Tương tự Black Box Test
    - Kiểm thử hiệu năng (Performance Test): kiểm thử việc vận hành của hệ thống
    - Kiểm thử khả năng chịu tải (Stress Test): kiểm thử các giới hạn của hệ thống
* System Test - kiểm thử mức hệ thống:
  + Mục đích System Test là kiểm thử thiết kế và toàn bộ hệ thống (sau khi tích hợp) có thỏa mãn yêu cầu đặt ra hay không
  + System Test .bắt đầu ngay sau Integration Test,trọng tâm là đánh giá về hoạt động, thao tác, sự tin cậy và các yêu cầu khác liên quan đến chất lượng của toàn hệ thống
  + Điểm khác nhau then chốt giữa Integration Test và System Test là System Test chú trọng các hành vi và lỗi trên toàn hệ thống, còn Integration Test chú trọng sự giao tiếp giữa các đơn thể hoặc đối tượng khi chúng làm việc cùng nhau
  + Bản thân System Test lại gồm nhiều loại kiểm thử khác nhau ,phổ biến nhất gồm:
    - Kiểm thử chức năng (Functional Test)
    - Kiểm thử khả năng vận hành (Performance Test)
    - Kiểm thử khả năng chịu tải (Stress Test hay Load Test)
    - Kiểm thử cấu hình (Configuration Test)
    - Kiểm thử khả năng bảo mật (Security Test)
    - Kiểm thử khả năng phục hồi (Recovery Test)
  + Nhìn từ quan điểm người dùng, các cấp độ kiểm thử trên rất quan trọng: bảo đảm hệ thống đủ khả năng làm việc trong môi trường thực
  + Lưu ý không nhất thiết phải thực hiện tất cả các loại kiểm thử nêu trên. Tùy yêu cầu và đặc trưng của từng hệ thống, tuỳ khả năng và thời gian cho phép của dự án, khi lập kế hoạch, người Quản lý dự án sẽ quyết định áp dụng những loại kiểm thử nào. Chính vì thế , đối với Website bán áo sơ mi Bee Shirt, chúng ta sẽ kiểm thử những chức năng thiết yếu nhất đối với 1 Website : chức năng, chịu tải, vận hành và bảo mật
* Acceptance Test - kiểm thử chấp nhận sản phẩm:
  + Thông thường, sau giai đoạn System Test là Acceptance Test, được khách hàng thực hiện (hoặc ủy quyền cho một nhóm thứ ba thực hiện). Mục đích của Acceptance Test là để chứng minh PM thỏa mãn tất cả yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm (và trả tiền thanh toán hợp đồng)
  + Acceptance Test có ý nghĩa hết sức quan trọng, mặc dù trong hầu hết mọi trường hợp, các phép kiểm thử của System Test và Acceptance Test gần như tương tự, nhưng bản chất và cách thức thực hiện lại rất khác biệt

Việc kiểm tra Website bán áo sơ mi Bee Shirt được thực hiện lần đầu tiên từ lúc hiện thực đến khi hoàn thành, chính vì thế nhóm sẽ test các chức năng hiện có của phần mềm này bao gồm:

| **Độ ưu tiên** | **Mã** | **Nội dung** | **Ghi chú** |
| --- | --- | --- | --- |
| 01 | TP-V1.0 - 01 | Trang chủ | Test GUI, **Test function** |
| 02 | TP-V1.0 - 02 | Trang đăng ký | Test GUI ,**Test function** |
| 03 | TP-V1.0 - 03 | Trang đăng nhập | Test GUI, **Test function** |
| 04 | TP-V1.0 - 04 | Trang tài khoản | Test GUI, **Test function** |
| 05 | TP-V1.0 - 05 | Trang lịch sử mua hàng | Test GUI, **Test function** |
| 06 | TP-V1.0 - 06 | Trang sản phẩm | Test GUI, **Test function** |
| 07 | TP-V1.0 - 07 | Trang sản phẩm chi tiết | Test GUI, **Test function** |
| 08 | TP-V1.0 - 08 | Trang giỏ hàng | Test GUI, **Test function** |
| 09 | TP-V1.0 - 09 | Trang thanh toán | Test GUI, **Test function** |
| 10 | TP-V1.0 - 10 | Bán hàng tại quầy | Test GUI, **Test function** |
| 11 | TP-V1.0 - 11 | Quản lý hóa đơn | Test GUI, **Test function** |
| 12 | TP-V1.0 - 12 | Quản lý sản phẩm | Test GUI, **Test function** |
| 13 | TP-V1.0 - 13 | Quản lý nhân viên | Test GUI, **Test function** |
| 14 | TP-V1.0 - 14 | Quản lý khách hàng | Test GUI, **Test function** |
| 15 | TP-V1.0 - 15 | Quản lý phiếu giảm giá | Test GUI, **Test function** |
| 16 | TP-V1.0 - 16 | Quản lý đợt giảm giá | Test GUI, **Test function** |
| 17 | TP-V1.0 - 17 | Trả hàng tại quầy | Test GUI, **Test function** |
| 18 | TP-V1.0 - 18 | Thống kê | Test GUI, **Test function** |
| 19 | TP-V1.0 - 19 | Yêu cầu về khả năng chịu tải và hiệu năng thực hiện | Non test function |
| 20 | TP-V1.0 - 20 | Quyền truy cập hệ thống với chức năng phân quyền | Non test function |

## Ràng buộc

| **Ngôn ngữ lập trình** | | Java (Spring Boot) + Angular | |
| --- | --- | --- | --- |
| **Cơ sở dữ liệu** | | MySQL | |
| **Ràng buộc** | | Luôn sẵn sàng hoạt động  Phần mềm chạy trên nền Windows  Bàn giao sản phẩm đúng thời gian | |

## Liệt kê các mạo hiểm

| **STT** | **Mạo hiểm** | **Phương án khắc phục & phòng ngừa** | **Mức độ ảnh hưởng (MD)** |
| --- | --- | --- | --- |
| 1 | Thiếu tính chi tiết trong thiết kế: Có thể dẫn đến việc thiếu tính tương thích và tính nhất quán trong quá trình phát triển. | - Sử dụng các quy trình và công cụ hỗ trợ thiết kế chi tiết  - Kiểm tra và xác nhận chi tiết trước khi triển khai | Cao |
| 2 | Thiếu kiểm tra đầu vào/đầu ra: Có thể dẫn đến việc xảy ra lỗi trong quá trình thực hiện test. | - Xác nhận đầu vào/đầu ra trước khi thực hiện test  - Tổng hợp và ghi nhận kết quả test để đảm bảo tính chính xác. | Cao |
| 3 | Thiếu tài nguyên để thực hiện test: Có thể giảm hiệu suất và chất lượng của quá trình test. | - Xác định và đánh giá nhu cầu tài nguyên trước khi bắt đầu quá trình test  - Tổ chức và quản lý tài nguyên một cách hiệu quả để đảm bảo hiệu suất và chất lượng của quá trình test | Cao |
| 4 | Thiếu tính nhất quán trong việc định nghĩa và thiết kế test case: Có thể dẫn đến việc thiếu tính tổng quan trong việc kiểm tra chất lượng của sản phẩm. | - Sử dụng các quy trình và công cụ hỗ trợ định nghĩa test case  - Xem xét và tổng hợp các yêu cầu và mục tiêu của sản phẩm trước khi thiết kế test case | Cao |
| 5 | Thiếu tính tổng quan trong quá trình test: Có thể dẫn đến việc bỏ sót một số lỗi quan trọng. | - Xác định và định nghĩa rõ ràng các mục tiêu và yêu cầu của quá trình test  - Sử dụng một phương pháp kiểm tra tổng quan để đảm bảo tất cả các lỗi được phát hiện  - Tổ chức và quản lý quá trình test một cách cẩn thận để đảm bảo tính tổng quan | Cao |

# CÁC YÊU CẦU CHO TEST

| **Độ ưu tiên** | **Mã** | **Nội dung** | **Mức độ công việc** | **Ghi chú** |
| --- | --- | --- | --- | --- |
| 01 | TP-V1.0 - 01 | Trang chủ | Design TC 0,5 man - days, test 0,5 man- days | Test GUI, **Test function** |
| 02 | TP-V1.0 - 02 | Trang đăng ký | Design TC 0,5 man - days, test 0,5 man- days | Test GUI, **Test function** |
| 03 | TP-V1.0 - 03 | Trang đăng nhập | Design TC 0,5 man - days, test 0,5 man- days | Test GUI, **Test function** |
| 04 | TP-V1.0 - 04 | Trang tài khoản | Design TC 0,5 man - days, test 0,5 man- days | Test GUI, **Test function** |
| 05 | TP-V1.0 - 05 | Trang lịch sử mua hàng | Design TC 0,5 man - days, test 0,5 man- days | Test GUI, **Test function** |
| 06 | TP-V1.0 - 06 | Trang sản phẩm | Design TC 0,5 man - days, test 0,5 man- days | Test GUI, **Test function** |
| 07 | TP-V1.0 - 07 | Trang sản phẩm chi tiết | Design TC 1 man - days, test 1 man- days | Test GUI, **Test function** |
| 08 | TP-V1.0 - 08 | Trang giỏ hàng | Design TC 1 man - days, test 1 man- days | Test GUI, **Test function** |
| 09 | TP-V1.0 - 09 | Trang thanh toán | Design TC 2 man - days, test 2 man- days | Test GUI, **Test function** |
| 10 | TP-V1.0 - 10 | Bán hàng tại quầy | Design TC 2 man - days, test 2 man- days | Test GUI, **Test function** |
| 11 | TP-V1.0 - 11 | Quản lý hóa đơn | Design TC 1 man - days, test 1 man- days | Test GUI, **Test function** |
| 12 | TP-V1.0 - 12 | Quản lý sản phẩm | Design TC 1,5 man - days, test 1,5 man- days | Test GUI, **Test function** |
| 13 | TP-V1.0 - 13 | Quản lý nhân viên | Design TC 1 man - days, test 1 man- days | Test GUI, **Test function** |
| 14 | TP-V1.0 - 14 | Quản lý khách hàng | Design TC 1 man - days, test 1 man- days | Test GUI, **Test function** |
| 15 | TP-V1.0 - 15 | Quản lý phiếu giảm giá | Design TC 1 man - days, test 1 man- days | Test GUI, **Test function** |
| 16 | TP-V1.0 - 16 | Quản lý đợt giảm giá | Design TC 1 man - days, test 1 man- days | Test GUI, **Test function** |
| 17 | TP-V1.0 - 17 | Trả hàng tại quầy | Design TC 1 man - days, test 1 man- days | Test GUI, **Test function** |
| 18 | TP-V1.0 - 18 | Thống kê | Design TC 0,5 man - days, test 0,5 man- days | Test GUI, **Test function** |
| 19 | TP-V1.0 - 19 | Yêu cầu về khả năng chịu tải và hiệu năng thực hiện | Design TC 1 man - days, test 1 man- days | Non test function |
| 20 | TP-V1.0 - 20 | Quyền truy cập hệ thống với chức năng phân quyền | Design TC 1 man - days, test 1 man- days | Non test function |

# CHIẾN LƯỢC TEST

* Kiểm thử giao diện kết hợp kiểm thử chức năng
* Kết hợp các phương pháp kiểm thử để có được kết quả chính xác nhất
* Sử dụng phương pháp phát triển phần mềm linh hoạt: Agile, mỗi chức năng sau khi code xong sẽ thực hiện kiểm thử luôn
* Test có bug phải log bug để DEV hiểu về bug và fix

## Các kiểu test

### Test chức năng (Functional Testing)

#### Test chức năng (Function Testing)

* Mục đích của test chức năng là tập trung vào các yêu cầu test có thể được lưu vết trực tiếp trong các UC hoặc các chức năng và quy tắc nghiệp vụ.
* Mục tiêu của kiểu test này là kiểm tra tính đúng đắn của các dữ liệu, quy trình và báo cáo cũng như việc thực hiện đúng những quy tắc nghiệp vụ.
* Kiểu test này dựa vào kỹ thuật black box, tức là kiểm tra ứng dụng và các xử lý nội tại bằng cách tương tác với ứng dụng thông qua giao diện người sử dụng và phân tích các kết quả hoặc đầu ra.
* Bảng sau liệt kê một số gợi ý đối với mỗi ứng dụng:

| **Mục đích test** | Đảm bảo mục tiêu test đúng đắn của chức năng, bao gồm định hướng, dữ liệu đầu vào, xử lý và dữ liệu nhận được |
| --- | --- |
| **Cách thực hiện** | Thực hiện mỗi UC, chu trình UC hoặc chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để kiểm tra:   * Kết quả mong đợi với dữ liệu hợp lệ. * Lỗi thích hợp hoặc thông báo hiển thị khi dữ liệu không hợp lệ. * Mỗi quy tắc nghiệp vụ đều được áp dụng đúng |
| **Điều kiện hoàn thành** | - Toàn bộ kế hoạch test đã được thực hiện.  - Toàn bộ các lỗi phát hiện ra đã được ghi nhận |
| **Các vấn đề đặc biệt** | Xác định hoặc mô tả các vấn đề (nội bộ hoặc bên ngoài) ảnh hưởng đến việc test chức năng |

#### Test giao diện người sử dụng (User Interface Testing)

* Test giao diện người dùng (UI) kiểm tra các tương tác của người dùng với phần mềm.
* Mục tiêu của test UI là để đảm bảo rằng giao diện người dùng cung cấp cho người sử dụng cách truy cập và sử dụng thích hợp thông qua các chức năng trong mục tiêu test.
* Ngoài ra, test UI còn để đảm bảo rằng các đối tượng trong phạm vi chức năng UI giống như mong đợi và phù hợp với tổ chức hoặc chuẩn ngành.

| **Mục đích test** | - Kiểm tra việc sử dụng thông qua mục tiêu test phản ánh đúng các chức năng và yêu cầu nghiệp vụ, bao gồm màn hình đến màn hình, trường đến trường và sử dụng các phương pháp truy cập (phím tabs, di chuột, tổ hợp phím)  - Kiểm tra các đối tượng và thuộc tính màn hình như menu, size, vị trí, và tập trung vào việc tương thích với chuẩn |
| --- | --- |
| **Cách thực hiện** | Tạo ra và chỉnh sửa test cho mỗi màn hình để kiểm tra việc sử dụng đúng cách và tình trạng các đối tượng cho mỗi màn hình và đối tượng của ứng dụng |
| **Điều kiện hoàn thành** | Mỗi màn hình được kiểm tra thành công đúng với phiên bản kiểm tra hoặc phạm vi chấp nhận được |
| **Các vấn đề đặc biệt** | Không phải toàn bộ các thuộc tính của các đối tượng đều truy cập được |

#### 

#### Test dữ liệu và tích hợp dữ liệu (Data and Database Integrity Testing)

* Cơ sở dữ liệu và xử lý cơ sở dữ liệu phải được test như một hệ thống con trong dự án.
* Hệ thống con này phải được test không cần thông qua giao diện người dùng để giao tiếp với dữ liệu

| **Mục đích test** | Đảm bảo rằng các phương pháp truy cập và chức năng xử lý là đúng và không có sai lệch dữ liệu |
| --- | --- |
| **Cách thực hiện** | - Thực hiện từng phương pháp truy cập và xử lý, thử từng trường hợp với dữ liệu hợp lệ và không hợp lệ hoặc các yêu cầu dữ liệu.  - Kiểm tra cơ sở dữ liệu để đảm bảo rằng dữ liệu được lưu trữ như mong đợi, toàn bộ các sự kiện với cơ sở dữ liệu xảy ra đều đúng, hoặc xem xét các dữ liệu trả về để đảm bảo rằng đã nhận được dữ liệu đúng cho các lý do đúng |
| **Điều kiện hoàn thành** | Tất cả các phương pháp truy cập và chức năng xử lý đều giống như thiết kế và không có sai lệch dữ liệu |
| **Các vấn đề đặc biệt** | - Việc test có thể đòi hỏi phải môi trường phát triển DBMS hoặc drivers để truy cập hoặc sửa dữ liệu trực tiếp trong cơ sở dữ liệu.  - Các xử lý phải được thực hiện bằng tay.  - Cơ sở dữ liệu có kích thước nhỏ hoặc tối thiểu (giới hạn số bản ghi) phải được dùng để làm rõ thêm các sự kiện không được phép chấp nhận |

#### Test chu trình nghiệp vụ (Business Cycle Testing)

* Test chu trình nghiệp vụ phải thực hiện các hoạt động trong dự án qua thời gian.
* Phải xác định một chu kỳ, ví dụ một năm, và các giao dịch và hoạt động có thể xảy ra trong chu kỳ của năm đó phải được thực hiện.
* Việc này bao gồm cả các chu kỳ hàng ngày, hàng tuần hoặc hàng tháng và các sự kiện là ảnh hưởng bởi ngày tháng

| **Mục đích test** | Đảm bảo mục đích của test là đúng đắn và các tiến trình chạy ngầm thực hiện đúng yêu cầu về mô hình nghiệp vụ và lịch trình |
| --- | --- |
| **Cách thực hiện** | Việc test sẽ giả lập vài chu trình nghiệp vụ bằng cách thực hiện các công việc sau:   * Các test dùng cho việc test chức năng sẽ được sửa lại hoặc nâng cấp để tăng số lần mỗi chức năng được thực hiện để giả lập một số người dùng khác nhau trong chu kỳ đã định. * Toàn bộ các chức năng theo ngày tháng sẽ được thực hiện với dữ liệu hợp lệ và không hợp lệ hoặc chu kỳ thời gian * Toàn bộ các chức năng xảy ra trong lịch trình chu kỳ sẽ được thực hiện vào thời gian thích hợp * Việc test sẽ bao gồm cả dữ liệu hợp lệ và không hợp lệ để kiểm tra: * Kết quả xảy ra khi dữ liệu hợp lệ. * Lỗi tương tự hoặc cảnh báo hiển thị khi dữ liệu không hợp lệ. * Mỗi quy tắc nghiệp vụ đều được áp dụng. |
| **Điều kiện hoàn thành** | - Toàn bộ kế hoạch test đã được thực hiện.  - Toàn bộ các lỗi phát hiện ra đều được ghi nhận |
| **Các vấn đề đặc biệt** | - Ngày và các sự kiện của hệ thống có thể đòi hỏi các hoạt động hỗ trợ đặc biệt  - Mô hình nghiệp vụ đòi hỏi xác định các yêu cầu và thủ tục test thích hợp |

### Test hiệu suất (Performance testing)

### Performance Profiling

* Performance profiling là một dạng test hiệu suất trong đó thời gian phản hồi, tỷ lệ giao dịch và các yêu cầu phụ thuộc thời gian khác được đo đạc và đánh giá.
* Mục đích của Performance Profiling là kiểm tra các yêu cầu về hiệu suất có đạt được hay không.
* Performance profiling là tiến hành và thực hiện để mô tả sơ lược và điều chỉnh các hành vi hiệu suất của mục tiêu test như một hàm của các điều kiện ví dụ workload hoặc cấu hình phần cứng.
* Chú ý: Các giao dịch dưới đây tham chiếu đến “các giao dịch nghiệp vụ logic”. Các giao dịch này được định nghĩa như xác định các UC mà tác nhân của hệ thống hy vọng được thực hiện bằng cách sử dụng mục tiêu test, như thêm mới hoặc sửa một hợp đồng

| **Mục đích test** | Kiểm tra các biểu hiện về hiệu suất cho các giao dịch hoặc chức năng nghiệp vụ đã thiết kế theo những điều kiện sau:   * workload bình thường đã biết trước (normal anticipated workload * workload xấu đã biết trước (anticipated worst case workload) |
| --- | --- |
| **Cách thực hiện** | - Sử dụng các thủ tục test cho test chức năng và chu trình nghiệp vụ  - Chỉnh sửa file dữ liệu để tăng số lượng các giao dịch hoặc scripts để tăng số tương tác xảy ra trong mỗi giao dịch  - Scripts phải được chạy trên một máy (trường hợp tốt nhất để đánh giá người dùng đơn lẻ, giao dịch đơn lẻ) và phải lặp lại trên nhiều máy trạm |
| **Điều kiện hoàn thành** | - Giao dịch đơn lẻ hoặc người dùng đơn lẻ: Thực hiện thành công test script không có lỗi và trong phạm vi mong đợi hoặc thời gian phản hồi cho mỗi giao dịch  - Nhiều giao dịch hoặc nhiều người dùng: Thực hiện thành công test script không có lỗi và trong thời gian chấp nhận được |
| **Các vấn đề đặc biệt** | - Việc test hiệu suất toàn diện bao gồm phải có một workload nền trên máy chủ.  - Có một số phương pháp để thực hiện, bao gồm:   * “Drive transactions” trực tiếp đến máy chủ, thường trong các form gọi SQL. * Tạo các người dùng ảo để giả lập nhiều máy trạm, thường là vài trăm. Sử dụng công cụ Remote Terminal Emulation để thực hiện việc load này, kỹ thuật này còn được dùng để load giao dịch trên mạng * Sử dụng nhiều người dùng, mỗi người chạy một test script để load lên hệ thống   - Test hiệu suất phải được thực hiện trên máy chuyên dụng hoặc thời gian chuyên dùng. Điều đó cho phép việc tính toán được đầy đủ và chính xác.  - Cơ sở dữ liệu sử dụng để test hiệu suất phải có kích thước thực tế hoặc đo bằng nhau |

#### Load Testing

* Load testing là một kiểu test hiệu suất mà mục tiêu là kiểm tra workload để tính toán và đánh giá hiệu suất và khả năng của mục đích test để tiếp tục thực hiện các chức năng thích hợp với các workload khác.
* Mục đích của load testing là xác định và đảm bảo các chức năng hệ thống thích hợp với nhiều nhất các workload.
* Ngoài ra load testing còn đánh giá các tính năng hiệu suất như thời gian phản hồi, tỉ lệ giao dịch và các vấn đề liên quan đến thời gian khác.
* Chú ý: Các giao dịch dưới đây tham chiếu đến “các giao dịch nghiệp vụ logic”. Các giao dịch này được định nghĩa như các chức năng xác định mà người dùng cuối của hệ thống mong muốn thực hiện thông qua ứng dụng như thêm hoặc sửa các thông tin hợp đồng

| **Mục tiêu test** | Kiểm tra hiệu suất về thời gian cho các giao dịch hoặc tình huống nghiệp vụ đã thiết kế với nhiều điều kiện workload |
| --- | --- |
| **Cách thực hiện** | - Sử dụng các test đã xây dựng cho test chức năng và chu trình nghiệp vụ.  - Sửa lại file dữ liệu để tăng số lượng giao dịch hoặc test nhằm tăng thêm số lần thực hiện mỗi giao dịch |
| **Điều kiện hoàn thành** | Nhiều giao dịch hoặc nhiều người dùng: Thực hiện thành công việc test không có lỗi và trong thời gian chấp nhận được |
| **Các vấn đề đặc biệt** | - Load testing phải được thực hiện trên máy chuyên dụng hoặc vào những giờ chuyên biệt. Nó cho phép đo đạc đầy đủ và chính xác.  - Cơ sở dữ liệu dùng cho load testing phải có kích thước thực tế hoặc đo bằng nhau |

#### Stress Testing

* Stress testing là một kiểu test hiệu suất được thực hiện để tìm ra các lỗi trong trường hợp thiếu tài nguyên hoặc cạnh tranh về tài nguyên. Bộ nhớ hoặc dung lượng đĩa ít có thể làm xuất hiện lỗi trong mục đích test mà nó không xuất hiện dưới điều kiện bình thường. Các lỗi khác có thể là kết quả của việc cạnh tranh hoặc chia sẻ tài nguyên như khóa cơ sở dữ liệu hoặc băng thông mạng. Stress testing cũng được dùng để xác định workload tối đa mà mục đích test có thể điều khiển được.
* Chú ý: Tham khảo các giao dịch dưới đây tham chiếu đến các giao dịch nghiệp vụ logic

| **Mục đích test** | Kiểm tra các chức năng của mục đích test là đúng đắn và không có lỗi với những điều kiện sau:   * Có ít hoặc không có bộ nhớ phù hợp trên máy chủ (RAM và DASS) * Số lượng máy trạm tối đa trong thực tế hoặc giả lập kết nối vào máy chủ   Nhiều người dùng thực hiện cùng một giao dịch với cùng dữ liệu hoặc account  Chú ý: Mục đích của Stress Testing có thể được phát biểu rõ và ghi ra các điều kiện mà hệ thống có thể lỗi, không thể tiếp tục thực hiện các chức năng một cách thích hợp |
| --- | --- |
| **Cách thực hiện** | - Sử dụng các test đã xây dựng để thực hiện Performance Profiling hoặc Load Testing.  - Để test việc hạn chế tài nguyên, test phải chạy trên máy đơn lẻ và RAM và DASD trên máy chủ phải giảm đi hoặc hạn chế  - Để thực hiện các stress tests khác phải sử dụng nhiều người dùng cùng chạy một TC hoặc bổ sung các test để thực hiện độ lớn giao dịch số hoặc hỗn hợp. |
| **Điều kiện hoàn thành** | Toàn bộ kế hoạch test được thực hiện và các hạn chế của hệ thống được xác định thỏa mãn các điều kiện tối thiểu đã đặt ra hoặc chỉ sai trong trong hợp các điều kiện không nằm trong điều kiện đã xác định |
| **Các vấn đề đặc biệt** | - Việc test Stressing mạng có thể đòi hỏi những công cụ để load mạng với nhiều thông báo hoặc gói dữ liệu.  - DASD dùng cho hệ thống phải tạm thời giảm xuống để hạn chế khả năng chỗ trống cho tăng trưởng cơ sở dữ liệu.  - Đồng bộ hóa các máy trạm đồng thời truy cập vào cùng một bản ghi hoặc các account dữ liệu |

#### Volume Testing

* Mục tiêu của Volume Testing là để kiểm tra giới hạn của độ lớn của dữ liệu có thể làm phần mềm bị sai. Volume Testing cũng xác định load lớn nhất liên tục hoặc độ lớn mà mục đích test có thể điều khiển được trong chu kỳ đã cho.
* Ví dụ, nếu mục đích test là xử lý một tập các bản ghi để tạo báo cáo, Volume Test có thể dùng một cơ sở dữ liệu test lớn và kiểm tra xem phần mềm có chạy bình thường và cho ra báo cáo đúng không.

| **Mục đích test** | Kiểm tra xem mục tiêu test có thực hiện thành công các chức năng theo những điều kiện sau không:   * Số máy trạm lớn nhất kết nối (thực tế hoặc vật lý – có thể), hoặc giả lập, tất cả đều thực hiện cùng một chức năng nghiệp vụ trong một chu kỳ mở rộng. * Kích thước cơ sở dữ liệu lớn nhất có thể (thực tế hoặc đo được) và nhiều query hoặc giao dịch báo cáo được thực hiện đồng thời. |
| --- | --- |
| **Cách thực hiện** | - Sử dụng các test đã xây dựng cho Performance Profiling hoặc Load Testing.  - Có thể dùng nhiều người dùng, chạy cùng một test hoặc bổ sung các test để thực hiện trường hợp giao dịch volume hoặc hỗn hợp xấu nhất (xem Stress Testing ở trên) trong một chu kỳ mở rộng.  Tạo ra cơ sở dữ liệu lớn nhất (thực tế, quy đổi, hoặc lọc các dữ liệu đại diện) và nhiều người dùng chạy các query và giao dịch báo cáo đồng thời trong một chu kỳ mở rộng |
| **Điều kiện hoàn thành** | Toàn bộ kế hoạch test được thực hiện và các giới hạn của hệ thống được xác định là đạt tới hoặc xử lý mà không có lỗi |
| **Các vấn đề đặc biệt** | Chu kỳ thời gian như thế nào là chấp nhận được cho điều kiện cơ sở dữ liệu lớn? |

### Test Bảo mật và Kiểm soát truy cập (Security and Access Control Testing)

* Test bảo mật và kiểm soát truy cập tập trung vào hai lĩnh vực bảo mật chính:
  + Bảo mật ở mức ứng dụng, bao gồm truy cập dữ liệu và các chức năng nghiệp vụ
  + Bảo mật ở mức hệ thống, bao gồm truy cập vào hệ thống hoặc truy cập từ xa
* Bảo mật mức ứng dụng đảm bảo rằng, dựa trên bảo mật đã yêu cầu, người dùng bị hạn chế sử dụng một số chức năng hoặc tình huống sử dụng, hoặc bị hạn chế trong giới hạn dữ liệu phù hợp với họ.
* Ví dụ, mọi người có thể được phép nhập dữ liệu để tạo account nhưng chỉ có người quản lý có thể xóa chúng. Nếu là bảo mật ở mức dữ liệu, việc test đảm bảo rằng “người dùng nhóm 1” có thể nhìn thấy các thông tin khách hàng, bao gồm dữ liệu tài chính, tuy nhiên “người dùng nhóm 2” chỉ nhìn thấy các thông tin chung chung cho cùng một khách hàng.
* Bảo mật mức hệ thống đảm bảo rằng chỉ những người dùng được cho quyền truy cập vào hệ thống mới có khả năng truy cập vào ứng dụng và chỉ bằng các cổng thích hợp

| **Mục đích test** | - Bảo mật mức ứng dụng: Đảm bảo rằng một người dùng chỉ có thể truy cập vào những chức năng hoặc dữ liệu mà nhóm người dùng đó được phép  - Bảo mật mức hệ thống: Đảm bảo rằng chỉ những người được phép truy cập hệ thống và ứng dụng được phép truy cập chúng |
| --- | --- |
| **Cách thực hiện** | - Bảo mật ứng dụng: Xác định và liệt kê từng nhóm người dùng và các chức năng hoặc dữ liệu mà họ được phép truy cập  - Tạo test case cho mỗi nhóm người dùng và kiểm tra từng quyền bằng cách tạo các giao dịch xác định cho mỗi nhóm  - Sửa lại nhóm người dùng và chạy lại tình huống test cho cùng những người dùng. Với mỗi trường hợp, kiểm tra các chức năng thêm vào hoặc dữ liệu có đúng không hay bị từ chối.  - Truy cập mức hệ thống: tham khảo các điều kiện đặc biệt dưới đây |
| **Điều kiện hoàn thành** | Với mỗi nhóm người dùng đều có các chức năng hoặc dữ liệu thích hợp, và toàn bộ các chức năng giao dịch đều như dự kiến và chạy trong các test chức năng ứng dụng trước đó |
| **Các vấn đề đặc biệt** | Truy cập vào hệ thống phải được xem xét hoặc thảo luận với quản trị hệ thống hoặc quản trị mạng, có thể không cần nếu nó là chức năng của quản trị mạng hoặc quản trị hệ thống |

### Test hồi quy (Regression Testing)

* Test hồi quy là một hoạt động cần thiết để chỉ ra rằng việc thay đổi code không gây ra những ảnh hưởng bất lợi

| **Mục đích test** | Test hồi quy dùng để kiểm tra các phần được sửa chữa trong phần mềm, để đảm bảo rằng những sự thay đổi đó không gây ra lỗi trong những phần khác |
| --- | --- |
| **Cách thực hiện** | - Tái sử dụng các TC từ những phần test trước để test các module đã được sửa chữa  - Sử dụng công cụ Rational Robot: Tạo một số test script về chức năng. Định nghĩa lịch thực hiện tự động cho chúng  - 80% các TC được chọn ngẫu nhiên  - Xây dựng một chương trình phân tích cơ sở hạ tầng. Chúng ta dựng một cơ sở hạ tầng có thể mở rộng được để thực hiện và đánh giá chương trình phân tích. Dựa vào kết quả phân tích chúng ta xác định phạm vi cần test hồi quy. |
| **Điều kiện hoàn thành** | - Toàn bộ các TC được thực hiện và đạt yêu cầu  - Toàn bộ các TC được chọn được thực hiện và đạt yêu cầu |
| **Các vấn đề đặc biệt** |  |

## Giai đoạn test

| **Kiểu test** | **Giai đoạn test** | | | |
| --- | --- | --- | --- | --- |
| **Unit** | **Integration** | **System** | **Acceptance** |
| Functional Tests  (Function, User Interface) | X | X | X | X |
| Performance Tests  (Performance profiles of individual components) | X | X |  |  |
| Performance Tests  (Load, Stress, Contention) |  |  | X | X |
| Reliability  (Integrity, Structure) | X | X |  |  |

## Các công cụ test

| **Mục đích** | **Công cụ** | **Nhà cung cấp/Tự xây dựng** | **Phiên bản** |
| --- | --- | --- | --- |
| Chạy Website Bee Shirt | IntelliJ + Chrome | Open source | 2020.3.4 |

## Môi trường test

Sử dụng môi trường gần giống với môi trường chạy thật của chương trình. Các yếu tố về môi trường như:

* Chạy trực tiếp trên code
* Sử dụng database riêng và thiết lập các thông số cho database gần giống với chương trình sẽ chạy thật
* Điều kiện về mạng: sử dụng mạng LAN
* Mô hình sẽ cài đặt chương trình test: số lượng máy chủ, máy trạm; việc chia tách các server, các máy trạm, việc cài đặt các domain … Thông thường, trong Unit test có thể sử dụng viếc thiết lập như khi lập trình, nhưng khi System test trở đi, phải chú ý thiết lập sao cho gần giống mô hình sẽ chạy trong thực tế nhất

# TÀI NGUYÊN

## Nhân lực

| **Vai trò** | **Họ và tên** | **Trách nhiệm** |
| --- | --- | --- |
| Test Manager | Ma Thu Thủy | * Đồng ý nhiệm vụ * Xây dựng kế hoạch kiểm thử * Quản lý nhóm kiểm thử * Đảm bảo chất lượng kiểm thử * Thực hiện báo cáo quản lý * Đánh giá hiệu quả của kiểm thử |
| Tester | Ma Thu Thủy, Vũ Thế Long | * Thực hiện kiểm thử * Báo cáo lỗi * Kiểm tra lỗi * Phân tích và phục hồi từ các lỗi kiểm thử * Tài liệu kiểm thử * Hỗ trợ cho nhóm |

## Hệ thống

| **Tài nguyên hệ thống phần cứng** | | |
| --- | --- | --- |
| **Tài nguyên** | **Số lượng** | **Loại** |
| Database Server | 1 | MySQL |
| CPU | 1 | i5-11400 6 Core Rocket Lake |
| RAM | 2 | 4G |
| Hardware | 1 | 250GB |

| **Tài nguyên hệ thống phần mềm** | | |
| --- | --- | --- |
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| Window 10 | 20h2 | Hệ điều hành |
| Window 11 | 22h2 | Hệ điều hành |
| Google Chrome | 3.5, 3.6 | Trình duyệt web |
| Internet Explorer | 7.0, 8.0 | Trình duyệt web |

# CÁC MỐC KIỂM SOÁT CỦA GIAI ĐOẠN TEST (TEST MILESTONES)

| **Milestone Task** | **Nguồn lực** | **Ngày bắt đầu** | **Ngày kết thúc** |
| --- | --- | --- | --- |
| Lập kế hoạch test | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Đăng nhập, đăng ký | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Quên mật khẩu, đổi mật khẩu | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Bán hàng online | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Bán hàng tại quầy | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Quản lý sản phẩm | Vũ Thế Long | 10/01/2024 | 17/01/2024 |
| Test: Quản lý thuộc tính sản phẩm | Vũ Thế Long | 10/01/2024 | 17/01/2024 |
| Test: Quản lý hóa đơn | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Quản lý phiếu giảm giá | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Quản lý đợt giảm giá | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Quản lý khách hàng | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Quản lý nhân viên | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Trả hàng tại quầy | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Thống kê | Vũ Thế Long | 10/01/2024 | 17/01/2024 |
| Test: Giỏ hàng | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Trang chủ - Client | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Sản phẩm - Client | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Sản phẩm chi tiết - Client | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |
| Test: Quản lý tài khoản | Ma Thu Thủy | 10/01/2024 | 17/01/2024 |

# 

# CÁC SẢN PHẨM

| **STT** | **Sản phẩm** | **Ngày bàn giao** | **Người bàn giao** | **Người nhận bàn giao** |
| --- | --- | --- | --- | --- |
| 1 | Test plan | 01/2024 | Ma Thu Thủy | Ma Thu Thủy |
| 2 | Test cases | 01/2024 | Ma Thu Thủy, Vũ Thế Long | Ma Thu Thủy |
| 3 | Defect log | 01/2024 | Ma Thu Thủy | Ma Thu Thủy |
| 4 | Defect reports | 01/2024 | Ma Thu Thủy | Ma Thu Thủy |