

**SERVICE DIRECTORY**

**KẾ HOẠCH KIỂM THỬ**

| **Mã dự án** | **SOF304\_SD18204** |
| --- | --- |
| **Mã tài liệu** | **TP\_V1.0** |
| **Ngày** | **17/11/2023** |

**Nhóm: Nhóm 6**

**Tên thành viên Nguyễn Văn Cường**

**Nguyễn Hoàng Tuấn Anh**

**Bùi Hoàng Anh**

**Phạm Anh Tuấn**

**Lương Quốc Bảo**

**Nguyễn Thành Long**

**Hà Nội - 01-2018**

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

[1.1](#_heading=h.3znysh7) Mục đích 5

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

# GIỚI THIỆU

## Mục đích

Mục đích của một test plan (kế hoạch kiểm thử) là định rõ các hoạt động kiểm thử cần thực hiện để đảm bảo rằng phần mềm hoặc hệ thống được kiểm tra hoạt động đúng đắn và đáp ứng được các yêu cầu:

* **Xác định phạm vi kiểm thử:** Định rõ phạm vi của kiểm thử, bao gồm các tính năng, modules hoặc hệ thống cần được kiểm tra. Mục tiêu là đảm bảo rằng tất cả các phần của phần mềm được thử nghiệm một cách toàn diện.
* **Đặt ra mục tiêu và kỳ vọng:** Xác định mục tiêu cụ thể mà kiểm thử cần đạt được và mức độ chất lượng mong muốn của phần mềm. Điều này giúp xác định các tiêu chí để đánh giá việc kiểm thử.
* **Xác định nguyên tắc và tiêu chuẩn:** Quy định các tiêu chuẩn, quy trình và nguyên tắc kiểm thử cần tuân thủ để đảm bảo tính nhất quán và hiệu quả của quá trình kiểm thử.
* **Lập kế hoạch và tài nguyên:** Xác định lịch trình chi tiết cho các hoạt động kiểm thử, bao gồm các bước, thời gian, nguồn lực và nhân lực cần thiết. Điều này giúp tối ưu hóa quá trình kiểm thử và đảm bảo sự chuẩn bị cần thiết.
* **Định rõ các phương pháp kiểm thử:** Xác định các phương pháp kiểm thử sẽ được sử dụng, bao gồm kiểm thử hộp trắng (white-box), kiểm thử hộp đen (black-box), kiểm thử tích hợp (integration testing), kiểm thử chấp nhận người dùng (user acceptance testing), và các kỹ thuật kiểm thử khác.
* **Xác định môi trường và công cụ:** Định rõ môi trường cần thiết cho việc kiểm thử, bao gồm phần cứng, phần mềm, dữ liệu và các công cụ kiểm thử. Điều này giúp đảm bảo sẵn sàng của môi trường kiểm thử.
* **Quản lý rủi ro:** Định rõ các rủi ro có thể xảy ra trong quá trình kiểm thử và lập kế hoạch để giảm thiểu tác động của chúng đối với chất lượng sản phẩm.
* **Báo cáo và đánh giá kết quả:** Xác định cách thức thu thập, phân tích và báo cáo kết quả kiểm thử. Điều này giúp đảm bảo rằng thông tin từ quá trình kiểm thử sẽ được ghi nhận và đánh giá một cách cẩn thận.

## Thông tin chung

* **Mục đích Test:** 
  + Mục đích test là kiểm tra tính đúng đắn, hiệu suất và tính năng của web bán đồng hồ. Việc kiểm thử nhằm đảm bảo rằng các tính năng và chức năng chính hoạt động như mong đợi, giúp cải thiện chất lượng sản phẩm và trải nghiệm người dùng.
* **Thông tin về các thành phần, ứng dụng, hệ thống:**
  + **Tên và mô tả:** Trang web là một nền tảng thương mại điện tử kết nối người bán và người mua hàng.
  + **Phiên bản và lịch sử:** Phiên bản hiện tại là v1.0.
  + **Chức năng và tính năng chính:** Trang web tập trung vào chức năng đăng nhập, đăng ký,giỏ hàng, thanh toán, quản lý sản phẩm, quản lý chi tiết sản phẩm, tìm kiếm và lọc sản phẩm, quản lý chi tiết sản phẩm.
  + **Kiến Trúc:** Ứng dụng xây dựng trên kiến trúc intelliJ-server, sử dụng công nghệ web như HTML, CSS cho phần front-end và Java cho phần back-end.
  + **Lịch Sử Dự Án:** Dự án đã trải qua một giai đoạn phát triển.

## Tài liệu liên quan

| STT | Tên tài liệu | Nguồn | Ghi chú |
| --- | --- | --- | --- |
| 1 | Mẫu test plan | [1. Template\_Testplan.docx - Google Docs](https://docs.google.com/document/d/1zVt6APXhWy3ZbtdeyBecNbc54EfIMaLk/edit) | Tài liệu cô gửi |

## Phạm vi test

* **Giai đoạn test:**
  + **Unit Testing:** Kiểm thử từng thành phần, module, hoặc class của mã nguồn để đảm bảo chúng hoạt động đúng và độc lập.
  + **Integration Testing:** Kiểm tra việc kết hợp các module hoặc thành phần để đảm bảo chúng hoạt động đúng khi được kết nối lại với nhau.
  + **System Testing:** Kiểm thử toàn bộ hệ thống hoặc ứng dụng để đảm bảo tính tương thích, tính chất hoạt động và tuân thủ các yêu cầu.
* **Kiểu test:**
  + **Kiểm thử chức năng( Function Testing):** Kiểm thử chức năng là kiểm tra xem hệ thống hoặc ứng dụng hoạt động như thế nào trong điều kiện các chức năng của nó. Nó tập trung vào việc xác nhận rằng mọi chức năng của phần mềm hoạt động đúng như mong đợi và theo các yêu cầu đã được xác định trước đó.
* **Các tính năng và chức năng sẽ được test:**
  + Đăng nhập và đăng xuất
  + Quản lý sản phẩm
  + Giỏ hàng và thanh toán (ưu tiên cao)
  + Tìm kiếm và lọc sản phẩm
  + Quản lý chi tiết sản phẩm
  + Quản lý danh mục
* **Điều kiện test hồi quy:** là một tập hợp các điều kiện hoặc trạng thái mà phải được kiểm tra để đảm bảo rằng việc thêm mới chức năng, sửa lỗi hoặc thay đổi không gây ra các vấn đề không mong muốn cho các tính năng đã tồn tại trước đó.
  + **Kiểm tra tính năng hiện tại:**
    - Xác nhận rằng các tính năng đã có từ trước vẫn hoạt động như mong đợi sau khi có thêm các thay đổi mới.
    - Đảm bảo rằng các chức năng cốt lõi không bị ảnh hưởng bởi các sửa đổi hoặc thêm mới.
  + **Kiểm tra tương thích và tính tương thích:**
    - Kiểm tra tính tương thích của ứng dụng trên các nền tảng, trình duyệt, hoặc thiết bị di động khác nhau.
    - Đảm bảo rằng ứng dụng vẫn tương thích với các phiên bản hệ điều hành mới nhất hoặc các môi trường mà nó được phát triển.
  + **Kiểm tra bảo mật:**
    - Xác minh rằng các cải tiến hoặc thay đổi mới không làm suy giảm tính an toàn hoặc bảo mật của hệ thống.
    - Đảm bảo rằng các lỗ hổng bảo mật không được tái xuất hiện sau khi đã được vá lần trước.
  + **Kiểm tra hiệu suất:**
    - Đánh giá hiệu suất của hệ thống sau các thay đổi để đảm bảo rằng chúng không gây ra tác động tiêu cực đến hiệu suất hoặc thời gian phản hồi của ứng dụng.
  + **Kiểm tra dữ liệu và tương tác:**
    - Kiểm tra tính chính xác của dữ liệu và tương tác của hệ thống với cơ sở dữ liệu hoặc các dịch vụ khác sau khi có thay đổi.
  + **Kiểm tra giao diện người dùng (UI):**
    - Xác minh rằng giao diện người dùng vẫn hoạt động đúng và không bị ảnh hưởng bởi các thay đổi.

## Ràng buộc

* **Ràng buộc thời gian:**
  + Hạn chế thời gian: Dự án chỉ có 3 tuần để thực hiện kiểm thử trước khi ra mắt sản phẩm mới.
* **Ràng buộc về nguồn lực:**
  + Ngân sách hạn chế: Chỉ có một ngân sách hạn chế để thực hiện kiểm thử, giới hạn số lượng nhân lực và công cụ có thể sử dụng.
  + Nhân lực hạn chế: Do sự thiếu hụt nhân lực, chỉ có một nhóm kiểm thử nhỏ và hạn chế kỹ năng chuyên môn.
* **Ràng buộc môi trường:**
  + Môi trường test hạn chế: Môi trường test không có sẵn để mô phỏng các điều kiện mạng kém hoặc tải trọng cao.
  + Sự thiếu hụt dữ liệu: Dữ liệu thử nghiệm có sẵn là ít và không đa dạng, không thể phủ hết tất cả các trường hợp test.
* **Ràng buộc về công nghệ:**
  + Hỗ trợ công nghệ hạn chế: Có hạn chế trong việc sử dụng công nghệ mới do sự hỗ trợ kỹ thuật không đầy đủ hoặc đội ngũ phát triển không quen thuộc với nó.
* **Ràng buộc về phạm vi và yêu cầu:**
  + Yêu cầu khách hàng: Khách hàng yêu cầu kiểm thử chỉ tập trung vào các tính năng cốt lõi, không có thời gian kiểm thử các tính năng phụ.

## Liệt kê các mạo hiểm

<Liệt kê các mạo hiểm/rủi ro và phương án khắc phục, phòng ngừa có thể ảnh hưởng đến việc thiết kế, phát triển và thực hiện test. Khi lập tài liệu thì cần xóa dòng hướng dẫn trên đi>

| **Stt** | **Mạo hiểm** | **Phương án khắc phục & phòng ngừa** | **Mức độ ảnh hưởng (MD)** |
| --- | --- | --- | --- |
| 1 | NA |  |  |

# CÁC YÊU CẦU CHO TEST

Danh sách dưới đây 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.

* **Tình huống test:** Kiểm tra web bán đồng hồ trên nền tảng Desktop để đảm bảo tính năng, hiệu suất và trải nghiệm người dùng.
* **Yêu cầu chức năng:** 
  + **Đăng nhập và Đăng ký:** Kiểm tra chức năng đăng nhập và đăng ký tài khoản người dùng. Bao gồm việc xác thực thông tin, bảo mật mật khẩu.Kiểm tra việc đăng nhập với tài khoản hợp lệ và không hợp lệ. Kiểm tra các trường hợp lỗi đăng nhập như sai mật khẩu, tài khoản không tồn tại.
  + **Quản lý sản phẩm:** Cho phép người dùng thêm sản phẩm mới, chỉnh sửa thông tin sản phẩm đã tồn tại, xóa sản phẩm khỏi hệ thống khi cần thiết, có khả năng tìm sản phẩm theo từ khóa.
  + **Giỏ hàng và thanh toán:** Thêm sản phẩm vào giỏ hàng. Quản lý giỏ hàng: chỉnh sửa số lượng, xóa sản phẩm. Quá trình thanh toán: chọn phương thức thanh toán, điền thông tin thanh toán.
  + **Tìm kiếm và lọc sản phẩm:** Tìm kiếm sản phẩm theo tên, loại, hoặc từ khóa. Lọc sản phẩm theo danh mục, thương hiệu, dây đeo, vật liệu.
  + **Quản lý chi tiết sản phẩm:** Có thể tìm kiếm sản phẩm theo mã, tên sản phẩm và thêm thông tin và chỉnh sửa chi tiết về sản phẩm.
  + **Quản lý danh mục:** Thêm danh mục và chỉnh sửa thông tin danh mục.
* **Yêu cầu phi chức năng:**
* **Hiệu suất:** Kiểm tra thời gian phản hồi của ứng dụng, tải trang, và xử lý dữ liệu trong các điều kiện mạng khác nhau.
* **Bảo mật:** Kiểm tra bảo mật của thông tin người dùng, bao gồm cả mã hóa, cơ chế xác thực và quản lý phiên.
* **Độ ổn định:** Kiểm tra ứng dụng trong điều kiện đa nhiệm, và đảm bảo nó không gây ra lỗi hoặc treo máy.

# CHIẾN LƯỢC TEST

### Xác định Mục Tiêu và Phạm Vi Kiểm Thử:

* + **Mục tiêu:** Đảm bảo tính năng và hiệu suất ổn định của trang website bán đồng hồ 5S.
  + **Phạm vi kiểm thử:** Kiểm thử các chức năng cơ bản như đăng nhập, đăng xuất, quản lý sản phẩm, giỏ hàng và thanh toán, tìm kiếm và lọc sản phẩm, quản lý sản phẩm, chi tiết sản phẩm, danh mục.
* **Xác định các phương pháp kiểm thử:**
  + Kiểm thử chức năng: Đảm bảo tính năng thêm, sửa, xóa,lọc sản phẩm, đăng nhập, đăng xuất, bán hàng của trang website đồng hồ 5S hoạt động chính xác.
  + Kiểm thử tương thích: Kiểm tra ứng dụng trên các thiết bị di động và hệ điều hành khác nhau (Android, iOS).
  + Kiểm thử hiệu suất: Đánh giá thời gian phản hồi và tải trọng của trang website khi có số lượng truy cập đông .

### Lập kế hoạch kiểm thử:

### Lập kế hoạch: Thiết lập lịch trình kiểm thử, phân công công việc cho nhóm kiểm thử, và xác định nguồn lực cần thiết.

### Tạo test cases: Phát triển test cases cho các chức năng.

* + **Chuẩn bị dữ liệu test:** Tạo dữ liệu test, bao gồm các thông tin khác nhau để phủ hết các trường hợp test.

### Thực hiện kiểm thử:

### Thực hiện test cases: Tiến hành kiểm thử theo test cases đã chuẩn bị, ghi chép kết quả và phát hiện lỗi.

* + **Kiểm thử tương thích:** Kiểm tra ứng dụng trên nhiều thiết bị và hệ điều hành khác nhau.
  + **Kiểm thử hiệu suất:** Sử dụng công cụ để đánh giá hiệu suất của ứng dụng trong điều kiện tải trọng.

### Đánh giá và điều chỉnh chiến lược:

* + Đánh giá kết quả: Đánh giá các kết quả kiểm thử, báo cáo lỗi và vấn đề phát sinh.
  + Hiệu chỉnh kế hoạch: Điều chỉnh kế hoạch kiểm thử dựa trên các phản hồi và kết quả kiểm thử để cải thiện quá trình kiểm thử.

### Tài liệu và báo cáo:

* + **Tài liệu kết quả:** Tạo tài liệu về kết quả kiểm thử, bao gồm báo cáo lỗi và các cải tiến được đề xuất.
  + **Báo cáo:** Tạo báo cáo chi tiết về quá trình kiểm thử, bao gồm tình trạng hiện tại của ứng dụng và các đề xuất cải thiện.

## Các kiểu test

* **Một ví dụ về mô tả kiểu test:**
  + ***Kỹ thuật***:
    - ***Functional Test***

Đối với chu trình sự kiện của mỗi UC, sẽ xác định một tập các giao dịch đại diện cho mỗi hành động của tác nhân khi thực hiện UC.

Trong giai đoạn đầu tiện, các UC 1-4 và 12 sẽ được test, theo hình thức sau:

UC 1 bắt đầu với tác nhân đã truy cập thành công vào ứng dụng và tại cửa sổ chính, và kết thúc khi người dùng xác định SAVE.

Mỗi TC sẽ được tiến hành và thực hiện bằng cách sử dụng Rational Robot.

Việc kiểm tra và đánh giá việc thực hiện mỗi TC sẽ được thực hiện theo phương pháp sau:

Thực hiện Test script (Mỗi test script có được thực hiện thành công như mong muốn không?)

Tình trạng Window hoặc phương pháp kiểm tra Object Data (tiến hành trong các test script) sẽ được dùng để kiểm tra sự hiển thị của các màn hình chính và dữ liệu được xác định được nắm bắt/hiển thị bởi mục tiêu test trong khi thực hiện test.

Cơ sở dữ liệu của các mục tiêu test (sử dụng Microsoft Access) sẽ được kiểm tra trước khi test và kiểm tra lại sau khi test để kiểm chứng rằng các thay đổi thực hiện trong quá trình test đã được phản ánh chính xác trong dữ liệu.

* + - ***Performance Test:***

Với mỗi UC, xác định một tập các giao dịch, như định nghĩa trong tài liệu phân tích workload, sẽ được tiến hành và thực hiện bằng Rational Suite Performance Studio và Rational Robot (GUI scripts)

Ít nhất 3 workload được phản ánh trong test script và lịch trình thực hiện test, bao gồm:

Stressed workload: 750 người dùng (15 % quản lý, 50 % bán hàng, 35 % marketing)

Peak workload: 350 người dùng (10 % quản lý, 60 % bán hàng, 30 % marketing)

Nominal workload: 150 người dùng (2 % quản lý, 75% bán hàng, 23 % marketing)

Test script dùng để thực hiện mỗi giao dịch sẽ bao gồm bộ đếm thời gian tương tự để đo thời gian phản hồi, ví dụ tổng thời gian giao dịch (như định nghĩa trong tài liệu phân tích workload), và các hoạt động giao dịch chính hoặc thời gian xử lý.

Test script sẽ thực hiện các workload trong 1 giờ (trừ phi được ghi chú khác trong tài liệu phân tích workload).

Kiểm tra và đánh giá việc thực hiện mỗi thực hiện test (của một workload) bao gồm:

Thực hiện test được theo dõi bằng biểu đồ trạng thái (để xác định rằng việc test và workload được thực hiện như mong muốn)

Thực hiện test script (mỗi test script có được thực hiện thành công như mong đợi không?)

Ghi nhận và đánh giá thời gian phản hồi đã định nghĩa bằng các báo cáo sau:

* + - * *Performance Percentile*
      * *Response Time*
* ***Điều kiện hoàn thành****:*
  + Tất cả các TC có trong kế hoạch đều đã được thực hiện
  + Tất cả các lỗi được xác định phải được ghi nhận vào một giải pháp đã thỏa thuận (All identified defects have been addressed to an agreed upon resolution)
  + Tất cả các TC có trong kế hoạch đã được thực hiện lại và toàn bộ các lỗi mở đã được ghi nhận như đã thỏa thuận và không có lỗi mới nào được phát hiện

**Hoặc:**

* + Toàn bộ các TC đặt mức ưu tiên cao đều đã được thực hiện
  + Toàn bộ các lỗi tìm thấy đều được ghi nhận vào một giải pháp đã thỏa thuận
  + Toàn bộ các lỗi có trọng số 1 và 2 đều được giải quyết
  + Tất cả các TC có mức ưu tiên cao đều đã được thực hiện lại và toàn bộ các lỗi mở đã được ghi nhận như đã thỏa thuận và không có lỗi mới nào được phát hiện
* ***Các vấn đề đặc biệt:***
  + Cơ sở dữ liệu test yêu cầu người thiết kế hoặc quản trị CSDL hỗ trợ để tạo mới, cập nhật và làm tươi dữ liệu test
  + Việc test hiệu suất hệ thống sử dụng máy chủ trong mạng hiện tại (có hỗ trợ cả các giao dịch khác không thuộc việc test). Việc test sẽ phải được lập lịch vào những giờ không còn các giao dịch khác trên mạng.
  + Mục tiêu test phải đồng nhất với hệ thống hợp lệ (hoặc giả lập đồng bộ) để việc test chức năng có thể được tiến hành và thực hiện
  + Việc test có thể bị dừng khi <số lỗi vượt quá norm, ...>
  + Cán bộ test có thể dừng test khi lập trình viên không thực hiện unit 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 qui tắc nghiệp vụ.
* Mục tiêu của kiểu test này là kiểm tra tính đúng đắn của các dữ liệu, qui trình và báo cáo cũng như việc thực hiện đúng những qui 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 qui 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)      Các đối tượng và thuộc tính màn hình như menus, size, position, state, và tập tring 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. Nghiên cứu thêm về DBMS để xác định các công cụ và kỹ thuật có thể có giúp hỗ trợ cho việc test:

| **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, ví dụ như ứng dụng ngân hàng>

| **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)/Test phi chức năng (Non-Functional 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 (ảo hoặc thực, xem các vấn đề đặc biệt dưới đây)> |
| **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 wordload 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 DASD)      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      Độ lớn các giao dịch xấu hoặc hỗn hợp (xem phần Performance Testing ở trên).  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 xấy 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, qui 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, như đã nói ở trê?> |

### 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 qui (Regression Testing)

<Test hồi qui là một hoạt động cấn thiết để chỉ ra rằng việc thay đổi code không gây ra những ảnh hưởng bất lợi>

| **Mục đích test:** | Test hồi qui 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 sơ 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 qui.> |
| **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

<Làm rõ trạng thái của giai đoạn sẽ thực hiện test. Bảng sau liệt kê các giai đoạn mà việc test thường được thực hiện>

| **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

<Liệt kê các công cụ sẽ áp dụng cho dự án>

| **Mục đích** | **Công cụ** | **Nhà cung cấp/Tự xây dựng** | **Phiên bản** |
| --- | --- | --- | --- |
|  |  |  |  |

## Môi trường test

<Chỉ rõ môi trường sẽ xây dựng để thực hiện test qua các giai đoạn Unit test, Integration test, System test, Acceptance test…. Với mỗi giai đoạn, hãy xác định các yếu tố để xây dựng môi trường test như thế nào, sử dụng như môi trường mà chương trình sẽ chạy thật hay tạo môi trường giả lập gần giống với môi trường chạy thật của chương trình. Các yếu tố về môi trường như:

- Khi test chạy chương trình bằng bản dịch hay chạy trên code. Thông thường, các giai đoạn System test, Acceptance test phải chạy trên bản dịch.

- Các database sẽ sử dụng độc lập hay dùng chung với database phát triển. Thông thường, từ Integration test, nhóm test phải thiết lập database riêng và thiết lập các thông số cho database gần giống hoặc giống hệt như khi chương trình sẽ chạy thật.

- Điều kiện về mạng: sẽ sử dụng mạng LAN hay Dial up… Thông thường, khi Unit test, có thể sử dụng mạng LAN nhưng khi System test trở đi thì nên sử dụng hệ thống đường truyền giống như hoặc gần giống như môi trường chạy thật.

- 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

Bảng sau mô tả nguồn lực test cho dự án.

| **Họ tên** | **Trách nhiệm/Ghi chú** |
| --- | --- |
| Nguyễn Văn Cường | Nhóm trưởng, kiểm thử chức năng đăng nhập, tìm kiếm và lọc sản phẩm |
| Bùi Hoàng Anh | Kiểm thử chức năng đăng ký |
| Nguyễn Hoàng Tuấn Anh | Viết test plan, kiểm thử chức năng giỏ hàng và thanh toán |
| Phạm Anh Tuấn | Kiểm thử chức năng quản lý sản phẩm |
| Nguyễn Thành Long | Kiểm thử chức năng quản lý danh mục |
| Lương Quốc Bảo | Kiểm thử chức năng quản lý chi tiết sản phẩm |

## Hệ thống

* Máy tính và trình duyệt

Yêu cầu hỗ trợ trình duyệt phổ biến như Chrome, Firefox, Safari và Edge.

Đảm bảo trang website tương thích với tất cả các thiết bị và các trình duyệt.

* Độ phân giải màn hình

Kiểm tra trên nhiều độ phân giải màn hình để đảm bảo trang web hiển thị đúng.

* Kết nối Internet
* Kiểm tra tốc độ tải trang web trên các loại kết nối Internet khác nhau.

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

Test v1.0 phải phối hợp các hoạt động test cho nguồn lực test được xác định trong phần trước. Độc lập với milestone của dự án, phải xác định để thông tin về tình trạng hoàn thành của dự án

| **Milestone Task** | **Nguồn lực** | **Ngày bắt đầu** | **Ngày kết thúc** |
| --- | --- | --- | --- |
| Đăng nhập | Nguyễn Văn Cường | 17/11/2023 | 10/12/2023 |
| Đăng ký | Bùi Hoàng Anh | 17/11/2023 | 10/12/2023 |
| Quản lý sản phẩm | Phạm Anh Tuấn | 17/11/2023 | 10/12/2023 |
| Giỏ hàng và thanh toán | Nguyễn Hoàng Tuấn Anh | 17/11/2023 | 10/12/2023 |
| Tìm kiếm và lọc sản phẩm | Nguyễn Văn Cường | 17/11/2023 | 10/12/2023 |
| Quản lý danh mục | Nguyễn Thành Long | 17/11/2023 | 10/12/2023 |
| Quản lý chi tiết sản phẩm | Lương Quốc Bảo | 17/11/2023 | 10/12/2023 |

# CÁC SẢN PHẨM

| **STT** | **Sản phẩm** | **Ngày bàn giao** | **Người bàn giao** | **Người nhận bàn giao** |
| --- | --- | --- | --- | --- |
|  | <Test cases> |  |  |  |
|  | <Test procedures> |  |  |  |
|  | <Defect log> |  |  |  |
|  | <Defect reports> |  |  |  |