**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TEST PLAN**

**Kiểm thử ứng dụng Firefly III**

|  |  |
| --- | --- |
| **Môn học :** | **Kiểm thử phần mềm** |
| **Lớp :** | **20\_3** |

**Giáo viên phụ trách:**

**TS. Trần Duy Hoàng**

**Thành Phố Hồ Chí Minh – 2023**

MỤC LỤC

[**MỤC LỤC i**](#_Toc155736936)

[**NỘI DUNG 1**](#_Toc155736937)

[**1. Giới thiệu 1**](#_Toc155736938)

[**1.1. Mục đích 1**](#_Toc155736939)

[**1.2. Thông tin chung 1**](#_Toc155736940)

[**1.3. Phạm vi test 2**](#_Toc155736941)

[**1.5. Ràng buộc 3**](#_Toc155736942)

[**2. Các yêu cầu cho test 3**](#_Toc155736943)

[**3. Chiến lược test 4**](#_Toc155736944)

[**3.1. Các kiểu test 4**](#_Toc155736945)

[**3.1.1. Kiểm thử chức năng (Functional testing) 4**](#_Toc155736946)

[**3.1.2. Kiểm thử giao diện và tính tiện dụng (GUI and Usability testing) 5**](#_Toc155736947)

[**3.1.3. Kiểm thử tự động (Automation testing) 5**](#_Toc155736948)

[**3.1.4. Kiểm thử hiệu năng (Performance testing) 5**](#_Toc155736949)

[**3.2. Giai đoạn test 6**](#_Toc155736950)

[**3.3. Các công cụ test 7**](#_Toc155736951)

[**3.4. Môi trường test 7**](#_Toc155736952)

NỘI DUNG

## 1. Giới thiệu

### **1.1. Mục đích**

Tài liệu này là chiến lược và kế hoạch kiểm thử của đồ án cuối kỳ môn **Kiểm thử phần mềm** với mục tiêu thực hiện kiểm tra chất lượng ứng dụng quản lý tài chính cá nhân Firefly III.

Tài liệu bao gồm 5 phần:

* Phần 1: Mô tả mục đích viết tài liệu, thông tin về ứng dụng, các tài liệu liên quan cũng như phạm vi kiểm thử, các ràng buộc và mạo hiểm trong quá trình kiểm thử.
* Phần 2: Các yêu cầu cho việc kiểm thử.
* Phần 3: Lập chiến lược kiểm thử.
* Phần 4: Liệt kê về tài nguồn nhân lực kiểm thử và yêu cầu hệ thống.
* Phần 5: Các mốc thời gian để kiểm soát tiến độ của đồ án.

### **1.2. Thông tin chung**

Mục đích của việc kiểm thử với ứng dụng Firefly III trong môn học:

* Áp dụng được các phương pháp kiểm thử hộp trắng và kiểm thử hộp đen để thiết kế kiểm thử: Phân hoạch tương đương và phân tích giá trị biên, Đồ thị nhân quả, Bảng quyết định, Lược đồ chuyển trạng thái & Kiểm thử dựa trên đặc tả Use-case.
* Xác định được các thành phần của kế hoạch kiểm thử phần mềm.
* Nhận biết các pha trong qui trình kiểm thử phần mềm.
* Quản lý qui trình kiểm thử phần mềm.
* Thực hiện kiểm thử phần mềm với nhiều loại kiểm thử khác nhau: kiểm chức năng, kiểm giao diện và tính tiện dụng, kiểm thử tự động, kiểm thử hiệu năng.
* Sử dụng được công cụ Katalon để quản lý kiểm thử.

Firefly III là một trình quản lý tài chính cá nhân miễn phí và mã nguồn mở, cho phép người dùng theo dõi chi tiêu và thu nhập, tạo ngân sách, phân loại và gắn thẻ giao dịch, và xem các báo cáo tài chính. Các chức năng chính của nó có thể kể tới như:

* Theo dõi chi tiêu và thu nhập.
* Lập ngân sách.
* Phân loại và gắn thẻ giao dịch.
* Nhập dữ liệu từ các công cụ bên ngoài.
* Tạo các báo cáo tài chính đẹp mắt.
* Cung cấp một API REST JSON để người dùng có thể tích hợp vào ứng dụng riêng của họ.
* Xử lý các giao dịch tự động dựa trên quy tắc.
* Tự lưu trữ và bảo mật dữ liệu tài chính.

Về kiến trúc của Firefly III:

* Backend: ngôn ngữ PHP và Laravel framework.
* Frontend: Vue.js
* Hệ quản trị cơ sở dữ liệu: MySQL hoặc MariaDB.

Ứng dụng đã trải qua rất nhiều phiên bản khác nhau, từ 3.0.0 tới phiên bản mới nhất là 6.1.4.

Trang web của ứng dụng mà nhóm đã deploy:

[ec2-3-26-8-206.ap-southeast-2.compute.amazonaws.com](http://ec2-3-26-8-206.ap-southeast-2.compute.amazonaws.com)

### **1.3. Phạm vi test**

Các tính năng sẽ được kiểm thử:

* Quản lý ngân sách
* Quản lý hóa đơn
* Quản lý tiền tiết kiệm
* Quản lý giao dịch
* Quản lý các danh mục (Categories)
* Quản lý các nhãn (Tags)
* Thêm mới 1 tài khoản nợ (Liability)
* Tạo báo cáo

Các giai đoạn kiểm thử:

* Kiểm thử đơn vị (Unit Testing): Kiểm tra từng chức năng riêng lẻ như Quản lý ngân sách, Quản lý hóa đơn, Quản lý tiền tiết kiệm, …
* Kiểm thử tích hợp (Integration Testing): Kiểm tra sự tương tác giữa các chức năng với nhau, ví dụ như sự tương tác giữa Quản lý ngân sách và Quản lý giao dịch.
* Kiểm thử hệ thống (System Testing): Kiểm tra toàn bộ hệ thống để đảm bảo rằng tất cả các chức năng hoạt động đúng cách khi kết hợp với nhau.

Các kiểu kiểm thử có trong test plan:

* Kiểm thử chức năng (Functional testing): Kiểm thử chức năng của mỗi phần.
* Kiểm thử giao diện và tính tiện dụng (GUI and Usability testing): Đánh giá giao diện người dùng và tính tiện dụng của ứng dụng.
* Kiểm thử tự động (Automation testing): Sử dụng công cụ tự động hóa để thực hiện kiểm thử chức năng cơ bản và lặp lại.
* Kiểm thử hiệu năng (Performance testing): Đánh giá hiệu suất của hệ thống trong điều kiện tải công việc cao.

Các giả thiết trong quá trình lập kế hoạch có thể ảnh hưởng đến việc thiết kế, phát triển hoặc thực hiện test:

* Mã nguồn được triển khai một cách đồng bộ với yêu cầu.
* Các unit test đều đã được triển khai và chạy mà không gặp lỗi.
* Các module tích hợp mà không gặp lỗi nghiêm trọng.
* Các chức năng hoạt động đúng như mong đợi trong môi trường thực tế.

Điều kiện để kiểm thử hồi qui:

* Chu kỳ: Khi ứng dụng Firefly III có 1 chức năng mới được thêm vào.
* Phạm vi: Tất cả các chức năng đã được chỉnh sửa hoặc cập nhật.

### **1.5. Ràng buộc**

* Thời gian kiểm thử: 10/12/2023 – 09/01/2024
* Số lượng người: 4.
* Môi trường kiểm thử: Kiểm thử với website đã được deploy.

## 2. 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.

<Liệt kê danh sách các yêu cầu chính cho test>

## 3. 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.

Có thể liệt kê với mỗi kiểu test tương ứng test cho chức năng nào[[1]](#footnote-2).

Việc test có thể dừng khi nào.

Ví dụ:

* Nó không còn hữu ích
* Nó đòi hỏi một phạm vi nhất định
* Nó đòi hỏi một số lỗi nhất định phải tìm được
* Hết thời gian
* Tester trả lại gói phần mềm cho LTV khi chưa sửa lỗi >

### **3.1. Các kiểu test**

#### 3.1.1. Kiểm thử chức năng (Functional 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> |

>

#### 3.1.2. Kiểm thử giao diện và tính tiện dụng (GUI and Usability 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 |

#### 3.1.3. Kiểm thử tự động (Automation testing)

#### 3.1.4. Kiểm thử hiệu năng (Performance testing)

|  |  |
| --- | --- |
| 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 |

### **3.2. Giai đoạn test**

| Kiểu test | Giai đoạn test | | | |
| --- | --- | --- | --- | --- |
| Unit | Integration | System | Acceptance |
| <Functional Tests  (Function, User Interface)> | X | X | X | X |
| <Performance Tests  (Load testing)> |  |  | X | X |

### **3.3. Các công cụ test**

|  |  |  |  |
| --- | --- | --- | --- |
| Mục đích | Công cụ | Nhà cung cấp/Tự xây dựng | Phiên bản |
| Quản lý các test case | Katalon | Katalon LLC | 9.1.0 |
| Kiểm thử hiệu năng | JMeter | Apache Software Foundation | 5.6.3 |

### **3.4. Môi trường test**

Chạy trên website đã được deploy (code PHP).

1. Chỉ dành cho tester FIS-HCM khi lập tài liệu kế hoạch test [↑](#footnote-ref-2)