SOFTWARE TEST PLAN

Project name

Approvals:

|  |  |  |
| --- | --- | --- |
| Approved By: | Signature | Date |
| *Ngô Minh Anh Thư* | *Signature 1* | *02/06/2025* |
|  |  |  |

Document Control

|  |  |
| --- | --- |
| Name | *Kế hoạch kiểm thử ứng dụng* |
| Doc. Ref. No. | *BMI\_TESTPLAN\_V1.* |
| Document Status | *Draft* |
| Date of Issue | *02/06/2025* |

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Doc. Version | Author | Date | Description / Change |
| *1.0* |  |  |  |
|  |  |  |  |

Distribution List

|  |  |
| --- | --- |
| Name | Role |
| *Ngô Minh Anh Thư* | *Giảng viên* |
|  | Sinh viên |

# **1.Introduction (giới thiệu chung)**

## **Tổng quan**

Khóa luận tốt nghiệp (KLTN) là học phần quan trọng, đánh dấu giai đoạn cuối trong đào tạo đại học, giúp sinh viên vận dụng kiến thức, rèn kỹ năng nghiên cứu, làm việc độc lập và nhóm. Tuy nhiên, tại Khoa CNTT – Trường Đại học Công Thương TP. Hồ Chí Minh, quy trình quản lý KLTN còn thủ công, rời rạc qua các biểu mẫu, email và file Excel, gây khó khăn trong việc theo dõi, tổng hợp và quản lý tiến độ. Thực tế đã xảy ra sai sót như trùng đề tài đăng ký, chậm phản hồi, mất nhiều thời gian tổng hợp điểm.

Trước thực trạng đó, nhóm quyết định xây dựng **Hệ thống Tiến trình Khóa luận Tốt nghiệp** nhằm số hóa, tự động hóa quy trình quản lý KLTN – từ đăng ký đề tài đến chấm điểm và báo cáo kết quả. Hệ thống được kỳ vọng sẽ nâng cao hiệu quả quản lý, giảm sai sót, tiết kiệm thời gian và hỗ trợ tốt hơn cho giảng viên và sinh viên.

## **Phạm vi**

## Phạm vi chức năng: Hệ thống tập trung vào các nghiệp vụ chính của quy trình quản lý KLTN tại Khoa CNTT, bao gồm: quản lý danh mục (Khoa, Ngành, Chuyên ngành, Bộ môn, GV, SV), quản lý đợt đăng ký, quản lý nhiệm vụ ra đề, quản lý đề tài (đề xuất, duyệt), quản lý đăng ký đề tài của sinh viên, quản lý tiến độ (giao việc, nộp bài, trao đổi, nhận xét), quản lý chấm điểm (GVHD, GVPB, Hội đồng), và thống kê, báo cáo cơ bản.

## Đối tượng sử dụng: Hệ thống phục vụ chính cho Sinh viên đủ điều kiện làm KLTN, Giảng viên (với các vai trò Ra đề, Hướng dẫn, Phản biện, Thành viên Hội đồng), Trưởng bộ môn, Giáo vụ Khoa, Trưởng khoa và Quản trị viên hệ thống thuộc Khoa CNTT - Trường Đại học Công Thương TP.HCM.

## Phạm vi kỹ thuật: Hệ thống được phát triển dưới dạng ứng dụng web, sử dụng công nghệ ASP.NET Core Web API cho phần backend, Next.js cho phần frontend và áp dụng kiến trúc Clean Architecture.

## Giới hạn: Hệ thống không bao gồm các chức năng quản lý tài chính liên quan đến KLTN; không tích hợp sâu với các hệ thống khác của trường (như cổng thông tin sinh viên) ngoại trừ khả năng nhập/xuất dữ liệu cơ bản; không quản lý các quy trình khác ngoài quy trình KLTN của Khoa CNTT.

## **Các định nghĩa và các từ viết tắt được dùng**

## CNTT là viết tắt của “Công nghệ thông tin”

## HUIT là tên viết tắt tiếng Anh của “Ho Chi Minh City University of Industry and Trade”, tức Trường Đại học Công Thương TP. Hồ Chí Minh

## GVHD là viết tắt của “Giảng viên hướng dẫn”

## GVPB là viết tắt của “Giảng viên phản biện”

## ASP là viết tắt của “Active Server Pages”

## .NET là viết tắt của “Network Enabled Technologies”

## API là viết tắt của “Application Programming Interface”

## SEO là viết tắt của “Search Engine Optimization”

# **2.Requirements for test (Các yêu cầu cần kiểm thử)**

## **2.1 Yêu cầu chức năng**

## **2.1.1 Quản lý người dùng và phân quyền**

## 2.1.1 Đăng nhập, xác thực và phân quyền

## Hệ thống cho phép người dùng đăng nhập bằng tài khoản cá nhân (email/mã số sinh viên/giảng viên).

## Xác thực theo vai trò: Sinh viên, Giảng viên, Trưởng bộ môn, Giáo vụ, Trưởng khoa.

## Mỗi vai trò chỉ được truy cập các chức năng phù hợp quyền hạn.

## 2.1.2 Quản lý tài khoản người dùng

## Admin có thể thêm/xóa/sửa thông tin người dùng.

## Sinh viên/giảng viên có thể cập nhật thông tin cá nhân.

## Có thể khóa/mở khóa tài khoản vi phạm.

## **2.2. Quản lý dữ liệu cơ bản**

## 2.2.1 Quản lý khoa, ngành, chuyên ngành, bộ môn

## Chức năng thêm, sửa, xóa, tìm kiếm các đơn vị.

## Liên kết mối quan hệ giữa các đơn vị (VD: Bộ môn thuộc khoa).

## 2.2.2 Quản lý giảng viên

## Thêm/sửa/xóa/thay đổi trạng thái giảng viên.

## Gán giảng viên vào bộ môn.

## Gán giảng viên hướng dẫn hoặc phản biện cho đề tài.

## 2.2.3 Quản lý sinh viên

## Nhập danh sách sinh viên từ file Excel hoặc từng cá nhân.

## Gán sinh viên vào các đợt đăng ký khóa luận.

## Quản lý nhóm sinh viên làm khóa luận.

## **2.3. Quản lý đợt đăng ký khóa luận**

## Tạo đợt đăng ký mới với thời gian bắt đầu/kết thúc.

## Gán sinh viên cho từng đợt.

## Đặt trạng thái (mở/đóng) đăng ký.

## **2.4. Quản lý đề tài khóa luận**

## 2.4.1 Giảng viên ra đề tài

## Nhập tên đề tài, mô tả, số lượng sinh viên tối đa, chuyên ngành phù hợp.

## Gửi yêu cầu duyệt đề tài.

## 2.4.2 Duyệt đề tài

## Trưởng bộ môn kiểm tra nội dung và phê duyệt/chỉnh sửa/góp ý.

## Chỉ đề tài được duyệt mới được phép hiển thị trong danh sách đăng ký.

## **2.5. Đăng ký đề tài khóa luận (Sinh viên)**

## Sinh viên lập nhóm, gửi lời mời nhóm cho các thành viên.

## Nhóm trưởng đại diện đăng ký đề tài nếu đề tài còn trống.

## Hệ thống tự động khóa đề tài khi đủ số lượng sinh viên.

## **2.6. Quản lý tiến độ thực hiện khóa luận**

## 2.6.1 Tạo yêu cầu tiến độ (GVHD)

## Giảng viên tạo nhiệm vụ theo từng giai đoạn.

## Đặt thời hạn nộp báo cáo.

## 2.6.2 Nộp báo cáo (Sinh viên)

## Sinh viên tải lên báo cáo từng giai đoạn.

## Giảng viên phản hồi góp ý, yêu cầu nộp lại nếu chưa đạt.

## 2.6.3 Đánh giá tiến độ

## Giảng viên đánh giá từng sinh viên trong nhóm.

## Ghi nhận tiến độ cho từng giai đoạn.

## **2.7. Chấm điểm đề tài**

## GVHD chấm điểm, đánh giá và duyệt đề tài.

## GVPB phản biện và chấm điểm độc lập.

## Nếu chênh lệch điểm lớn, trưởng bộ môn chuyển lên hội đồng để bảo vệ.

## Hệ thống tính điểm trung bình: GVHD + GVPB + hội đồng (nếu có).

## **2.8. Đưa đề tài ra hội đồng**

## Với đề tài cần phản biện hội đồng, giáo vụ lập danh sách hội đồng gồm chủ tịch, thư ký, các ủy viên.

## Hội đồng đặt câu hỏi, ghi nhận đánh giá và cho điểm từng sinh viên.

## Hệ thống lưu lại biên bản phản biện.

## **2.9. Thống kê và báo cáo**

## Thống kê số lượng đề tài, sinh viên theo từng trạng thái.

## Thống kê tiến độ theo từng đợt, theo sinh viên, theo giảng viên.

## Xuất báo cáo điểm, kết quả bảo vệ ra file Excel/PDF.

## **2.10. Quản lý thông báo và hệ thống hỗ trợ**

## Giáo vụ và trưởng khoa có thể gửi thông báo hệ thống (dạng popup/email).

## Giao diện hỗ trợ phản hồi lỗi, góp ý từ người dùng.

## **2.11. Yêu cầu phi chức năng**

## Tính dễ sử dụng (Usability): Giao diện người dùng cần đơn giản, trực quan, thân thiện với sinh viên, giảng viên và giáo vụ. Người dùng có thể thao tác mà không cần qua đào tạo chuyên sâu. Hệ thống phải hỗ trợ truy cập và sử dụng trực tiếp qua trình duyệt web mà không cần cài đặt phần mềm bổ sung.

## Hiệu suất (Performance) : Hệ thống phải đảm bảo xử lý nhanh các yêu cầu, phản hồi trong vòng < 10 giây đối với các thao tác cơ bản.

## Tính sẵn sàng và độ tin cậy (Availability & Reliability): Hệ thống cần hoạt động ổn định, trong suốt quá trình diễn ra đợt đăng ký và xử lý khóa luận.

## Tính bảo mật (Security) : Hệ thống phải đảm bảo xác thực người dùng thông qua cơ chế đăng nhập bằng tài khoản cá nhân (sinh viên, giảng viên, giáo vụ). Áp dụng phân quyền truy cập rõ ràng: sinh viên chỉ xem thông tin liên quan; giảng viên và giáo vụ được chỉnh sửa trong phạm vi cho phép.

## Khả năng mở rộng (Scalability) : Hệ thống cần thiết kế theo hướng mô-đun và tuân theo kiến trúc Clean Architecture để dễ dàng bảo trì, mở rộng khi có thêm chức năng hoặc thay đổi quy trình.

## Tương thích trình duyệt (Browser Compatibility) : Hệ thống phải chạy tốt trên các trình duyệt phổ biến như Chrome, Firefox, Edge và Safari.

## **2.12. Công cụ kiểm thử**

Trình duyệt web: Google Chrome, Mozilla Firefox.

Công cụ Developer Tools của trình duyệt (để inspect element nếu cần).

Máy tính cá nhân.

Phần mềm bảng tính (ví dụ: Microsoft Excel) để quản lý Test Cases.

# **3.Acceptance criteria (Các tiêu chí chấp nhận sản phẩm)**

## **3.1. Entry criteria (Tiêu chí bắt đầu kiểm thử)**

## Môi trường kiểm thử: Môi trường kiểm thử (Staging/Test Environment) đã được thiết lập đầy đủ, ổn định và cấu hình tương đồng với môi trường dự kiến triển khai (Production-like) theo đặc tả trong mục "Resources for Testing". Cơ sở dữ liệu kiểm thử đã được chuẩn bị với dữ liệu mẫu đầy đủ và đa dạng (bao gồm cả dữ liệu hợp lệ và không hợp lệ) cho tất cả các vai trò người dùng (Admin, Giáo vụ, Trưởng khoa, Trưởng Bộ môn, Giảng viên, Sinh viên). Sản phẩm phần mềm: Các module/chức năng được chỉ định cho giai đoạn kiểm thử hiện tại đã được đội ngũ phát triển hoàn thành (code complete) và triển khai thành công lên môi trường kiểm thử. Kết quả kiểm thử đơn vị (Unit Test) và kiểm thử tích hợp (Integration Test) từ đội ngũ phát triển đạt tỷ lệ pass tối thiểu 90% cho các chức năng cốt lõi. Tài liệu: Tài liệu đặc tả yêu cầu (SRS), tài liệu thiết kế hệ thống (SDS) phiên bản mới nhất đã được phê duyệt và cung cấp cho đội ngũ kiểm thử. Bộ Test Case chi tiết cho phạm vi kiểm thử hiện tại đã được thiết kế, xem xét (review) và phê duyệt. Nhân lực: Đội ngũ kiểm thử đã được phân công nhiệm vụ và sẵn sàng thực hiện kiểm thử.

## **3.2. Exit citeria (Tiêu chí hoàn thành kiểm thử)**

## Độ bao phủ Test Case: ít nhất 95% các Test Case được lập và phê duyệt phải được thực thi (bao gồm cả re-test và regression test).

## Chất lượng sản phẩm

## Tỷ lệ Passed: ≥ 95% tổng số Test Case. 100% Test Case nhóm Critical phải Passed. ≥ 90% Test Case nhóm High phải Passed.

## Lỗi tồn đọng: Không có lỗi Blocker hoặc Critical.

## Tối đa 5 lỗi High, có kế hoạch xử lý rõ ràng.

## Tối đa 15 lỗi Medium.

## Lỗi Low được ghi nhận, xử lý sau.

## Tài liệu & theo dõi: Kết quả kiểm thử được ghi đầy đủ trên hệ thống hoặc tài liệu.

## Báo cáo tổng kết hoàn tất và được phê duyệt.

## Tất cả lỗi được ghi nhận và có trạng thái rõ ràng (Closed, Deferred,...).

## Phê duyệt cuối cùng : Phải có sự đồng ý của Test Lead/Manager và đại diện Khách hàng hoặc Quản lý Dự án.

## **3.3. Suspension/ resumption criteria (Tiêu chí tạm dừng / bắt đầu lại kiểm thử)**

# Tiêu chí Tạm dừng (Suspension Criteria) :

# Có ≥ 3 lỗi Blocker chưa khắc phục.

# Có > 10 lỗi Critical chưa khắc phục, không có workaround.

# ≥ 40% Test Case của một module quan trọng bị failed.

# Môi trường kiểm thử không ổn định (lỗi nghiêm trọng kéo dài > 4 ngày làm việc).

# Yêu cầu thay đổi lớn về chức năng/kiến trúc chưa có đặc tả cập nhật, ảnh hưởng đến phạm vi kiểm thử.

# Tiêu chí Tiếp tục (Resumption Criteria) :

# Tất cả lỗi Blocker đã được sửa và xác minh.

# Ít nhất 80% lỗi Critical gây tạm dừng đã được khắc phục.

# Tỷ lệ fail < 20% sau khi sửa lỗi ở các chức năng quan trọng.

# Môi trường kiểm thử ổn định trở lại ≥ 2 ngày liên tục.

# Đặc tả yêu cầu đã cập nhật, Test Plan/Test Case điều chỉnh phù hợp.

# **4. Test strategy (Các chiến lược kiểm thử)**

**4.1 Phương pháp kiểm thử:**

Kiểm thử hộp đen (Black Box Testing): Tập trung vào việc kiểm thử chức năng của ứng dụng dựa trên các yêu cầu và đặc tả mà không cần biết về cấu trúc code bên trong.

**4.2 Các loại kiểm thử sẽ thực hiện:**

**4.2.1. Kiểm thử Chức năng (Functional Testing)**

**Mục đích**:

* Đảm bảo tất cả chức năng của hệ thống "Tiến trình Khóa luận Tốt nghiệp" (quản lý danh mục, đăng ký, đề tài, tiến độ, chấm điểm, báo cáo, phân quyền) hoạt động đúng theo yêu cầu, use case, và quy trình nghiệp vụ.
* Hỗ trợ chính xác các vai trò (Admin, Giáo vụ, Trưởng khoa, Giảng viên, Sinh viên).
* Đảm bảo toàn vẹn dữ liệu và luồng nghiệp vụ end-to-end trơn tru.

**Kỹ thuật**:

* Kiểm thử dựa trên yêu cầu, use case, kịch bản.
* Phân vùng tương đương, phân tích giá trị biên, bảng quyết định, chuyển đổi trạng thái.
* Đoán lỗi, kiểm thử khám phá.

**Tiêu chuẩn dừng**:

* **Tạm dừng**: Lỗi Blocker (1), Critical (3), tỷ lệ thất bại >25%, môi trường lỗi nghiêm trọng.
* **Hoàn thành**: 100% test case Critical/High pass, 95% test case thực thi, lỗi Blocker/Critical sửa hoặc có kế hoạch sửa, luồng chính ổn định.

**Chịu trách nhiệm**: Test Designer, Tester/QA, Test Lead/QA Manager.

**Cách kiểm thử**:

* Chủ yếu thủ công, tự động cho test case ổn định (Selenium, Cypress, NUnit).
* Dùng công cụ quản lý test case (TestRail, Jira, Excel).

**Xử lý ngoại lệ**:

* Ghi nhận lỗi chi tiết (bước tái tạo, kết quả, bằng chứng).
* Thông báo khẩn cấp lỗi Blocker/Critical.
* Theo dõi và tạm dừng test case khi môi trường lỗi.

**4.2.2. Kiểm thử Hiệu năng (Performance Testing)**

**Mục đích**:

* Đánh giá thời gian phản hồi (<3s), ổn định, khả năng mở rộng, tìm điểm nghẽn dưới các mức tải khác nhau.

**Kỹ thuật**:

* Load Testing: Mô phỏng tải dự kiến (100-500 người dùng).
* Stress Testing: Đẩy tải vượt giới hạn (150-200%).
* Endurance Testing: Chạy tải dài (4-8h).
* Spike Testing: Kiểm tra tải đột biến.
* Volume Testing: Xử lý dữ liệu lớn.

**Tiêu chuẩn dừng**:

* **Tạm dừng**: Hệ thống sập sớm, lỗi dữ liệu, thời gian phản hồi >10-15s, công cụ lỗi.
* **Hoàn thành**: Thực thi kịch bản chính, KPI đạt, tài nguyên ổn định (<75-80% CPU), điểm nghẽn xác định, báo cáo hoàn thiện.

**Chịu trách nhiệm**: Performance Tester/QA, phối hợp Developer/DevOps.

**Cách kiểm thử**:

* Dùng công cụ (JMeter, K6, LoadRunner).
* Thiết kế kịch bản thực tế, môi trường giống Production.
* Theo dõi tài nguyên, phân tích số liệu.

**Xử lý ngoại lệ**:

* Ghi nhận chi tiết (biểu đồ, log).
* Phân tích nguyên nhân, báo cáo, kiểm thử lại sau tối ưu.

**4.2.3. Kiểm thử Bảo mật (Security Testing)**

**Mục đích**:

* Đảm bảo xác thực, phân quyền, bảo vệ dữ liệu, chống tấn công (OWASP Top 10), quản lý phiên an toàn, xử lý lỗi không lộ thông tin.

**Kỹ thuật**:

* Kiểm thử xác thực, phân quyền, quản lý phiên, đầu vào (SQL Injection, XSS).
* Kiểm tra URL trực tiếp, header HTTP, HTTPS.
* Quét lỗ hổng tự động (OWASP ZAP).

**Tiêu chuẩn dừng**:

* **Tạm dừng**: Lỗ hổng High/Critical dễ khai thác, công cụ quét lỗi.
* **Hoàn thành**: Thực thi kịch bản, báo cáo lỗ hổng, lỗi High/Critical sửa, rủi ro thấp ghi nhận.

**Chịu trách nhiệm**: Security Tester/QA, phối hợp Developer.

**Cách kiểm thử**:

* Kết hợp thủ công (checklist OWASP) và tự động (ZAP, Burp Suite).
* Kiểm thử trên môi trường Staging.

**Xử lý ngoại lệ**:

* Báo cáo bảo mật, chi tiết lỗ hổng (PoC, tác động).
* Ưu tiên sửa lỗi nghiêm trọng, kiểm thử lại.

**4.2.4. Kiểm thử Tính khả dụng (Usability Testing)**

**Mục đích**:

* Đánh giá dễ học, hiệu quả, dễ nhớ, điều hướng, nhất quán, phản hồi, mức hài lòng người dùng.

**Kỹ thuật**:

* Heuristic Evaluation (Nielsen).
* Kịch bản, phỏng vấn, thu thập phản hồi, kiểm tra nhất quán.

**Tiêu chuẩn dừng**:

* **Tạm dừng**: Vấn đề nghiêm trọng cản trở tác vụ cốt lõi.
* **Hoàn thành**: Thực thi kịch bản, ghi nhận vấn đề, cải thiện ưu tiên cao, báo cáo hoàn thiện.

**Chịu trách nhiệm**: QA/Tester, UX Designer, người dùng cuối (UAT).

**Cách kiểm thử**:

* Thủ công, đóng vai người dùng, dùng checklist.
* Ghi chép, chụp ảnh/video minh họa.

**Xử lý ngoại lệ**:

* Ghi nhận chi tiết, phân loại, đề xuất cải thiện.
* Theo dõi, kiểm thử lại sau cập nhật.

**4.2.5. Kiểm thử Tính tương thích (Compatibility Testing)**

**Mục đích**:

* Đảm bảo hệ thống hoạt động trên trình duyệt (Chrome, Firefox, Edge), hệ điều hành (Windows, macOS), thiết bị (responsive).

**Kỹ thuật**:

* Kiểm thử thủ công song song.
* Dùng giả lập (BrowserStack, Developer Tools).
* Kiểm tra layout/giao diện.

**Tiêu chuẩn dừng**:

* **Tạm dừng**: Lỗi nghiêm trọng trên trình duyệt ưu tiên, công cụ lỗi.
* **Hoàn thành**: Thực thi kịch bản, báo cáo lỗi, sửa lỗi nghiêm trọng, ghi nhận lỗi nhỏ.

**Chịu trách nhiệm**: QA/Tester, hỗ trợ Frontend Developer.

**Cách kiểm thử**:

* Lập ma trận tương thích.
* Ưu tiên trình duyệt phổ biến, kiểm tra chức năng và layout.

**Xử lý ngoại lệ**:

* Ghi lỗi chi tiết (môi trường, bằng chứng).
* Phân biệt lỗi chức năng/hiển thị, thảo luận giải pháp, kiểm thử lại.

**4.2.6. Kiểm thử Hồi quy (Regression Testing)**

**Mục đích**:

* Đảm bảo thay đổi không gây lỗi mới, duy trì chất lượng, tăng tự tin khi triển khai.

**Kỹ thuật**:

* Full Regression: Thực thi toàn bộ test case.
* Selective Regression: Chọn test case rủi ro cao.
* Ưu tiên test case quan trọng.
* Tự động hóa (nếu có).

**Tiêu chuẩn dừng**:

* **Tạm dừng**: Nhiều lỗi nghiêm trọng, bộ test tự động lỗi hàng loạt.
* **Hoàn thành**: Thực thi test case chọn lọc, pass >98-100%, không lỗi nghiêm trọng, báo cáo hoàn thiện.

**Chịu trách nhiệm**: QA/Tester, Developer (Unit/Integration Test).

**Cách kiểm thử**:

* Xác định phạm vi dựa trên thay đổi.
* Duy trì bộ test case hồi quy.
* Thực thi thủ công/tự động, so sánh kết quả.

**Xử lý ngoại lệ**:

* Báo cáo lỗi hồi quy ngay, cung cấp thông tin so sánh.
* Phân tích nguyên nhân, cập nhật bộ test case.

**4.3 Môi trường kiểm thử:**

Hệ điều hành: Windows 10 (hoặc phiên bản Windows khác mà sinh viên sử dụng).

Trình duyệt:

Google Chrome (Phiên bản ổn định mới nhất).

Mozilla Firefox (Phiên bản ổn định mới nhất).

Kết nối mạng: Internet.

**4.4 Tài nguyên kiểm thử:**

Nhân lực: 4 Sinh viên (người thực hiện kiểm thử).

Phần cứng: Máy tính cá nhân/Laptop đáp ứng yêu cầu chạy các trình duyệt web.

Phần mềm: Các trình duyệt web đã nêu, Microsoft Excel (hoặc tương đương) để quản lý Test Cases.

**4.5 Lịch trình kiểm thử (tham khảo trong bối cảnh thi):**

Thiết kế Test Plan: [ước lượng thời gian, ví dụ: 30 phút]

Thiết kế Test Cases: [ước lượng thời gian, ví dụ: 60 phút]

Thực thi Test Cases: [ước lượng thời gian, ví dụ: 70 phút]

Tổng hợp và hoàn thiện bài làm: [ước lượng thời gian, ví dụ: 20 phút]

**4.6 Sản phẩm bàn giao của quá trình kiểm thử:**

Tài liệu Kế hoạch kiểm thử (Test Plan) đã hoàn chỉnh (file này).

Tài liệu các Trường hợp kiểm thử (Test Cases) chi tiết.

(Tùy chọn, nếu đề bài yêu cầu) Báo cáo tóm tắt kết quả kiểm thử, bao gồm số lượng test case đã thực thi, số lượng pass/fail, các lỗi nổi bật (nếu có).