## **Test Plan theo chuẩn IEEE 829**

**1. Test Plan Identifier** TP\_NODEJS\_ARTICLES\_V1.0\_20250526

**2. Introduction**

Tài liệu này mô tả kế hoạch kiểm thử toàn diện cho ứng dụng website NodeJS (sau đây gọi là "Sản phẩm"). Sản phẩm bao gồm các chức năng quản lý bài viết (articles), bình luận (comments), cùng với các chức năng xác thực và phân quyền người dùng. Kế hoạch này tuân thủ theo chuẩn IEEE 829.

**Mục tiêu của Kế hoạch Kiểm thử này:**

* Xác định phạm vi, cách tiếp cận, tài nguyên và lịch trình của các hoạt động kiểm thử dự kiến.
* Chi tiết hóa các hạng mục cần kiểm thử, các tính năng cần kiểm thử, các loại kiểm thử sẽ được thực hiện, tiêu chí chấp nhận sản phẩm, các tài liệu sẽ được bàn giao.
* Đảm bảo tất cả các chức năng được mô tả trong "Features to be Tested" hoạt động chính xác.
* Đạt tỷ lệ pass rate ít nhất 95% cho các test case thực thi ở critical & high priority.
* Không có lỗi nghiêm trọng (Blocker/Critical) nào tồn tại khi release Build 1 và Build 2.
* Đảm bảo hiệu năng website đáp ứng các ngưỡng chấp nhận được.
* Giao diện người dùng trực quan, dễ sử dụng và nhất quán.
* Đảm bảo các vấn đề bảo mật cơ bản được kiểm tra và khắc phục.

**Tài liệu tham khảo:**

* Tài liệu đặc tả yêu cầu phần mềm (SRS) - (Giả định tên: SRS\_NodeJS\_Articles\_V1.0)
* Tài liệu thiết kế kiến trúc/hệ thống - (Giả định tên: SDD\_NodeJS\_Articles\_V1.0)
* Chuẩn IEEE 829-2008 for Software and System Test Documentation.

**3. Test Items**

Các hạng mục phần mềm sẽ được kiểm thử bao gồm:

* **Website NodeJS Articles & Comments - Build 1:** Bao gồm các chức năng từ 1 đến 5 như được mô tả trong mục "Features to be Tested".
* **Website NodeJS Articles & Comments - Build 2:** Bao gồm các chức năng từ 6 đến 9, cùng với toàn bộ các chức năng của Build 1 (kiểm thử hồi quy).
* **API Backend:** Các API endpoints cung cấp dữ liệu và chức năng cho frontend.
* **Cơ sở dữ liệu:** Tương tác dữ liệu liên quan đến articles, comments, và users.

**4. Features to be Tested**

**Build 1:**

1. **Hiển thị thông tin tất cả các Article (Trang chủ - Home):**
   * Thông tin: Hình đại diện, tiêu đề, mô tả tóm tắt.
   * Bố cục: Tối đa 3 articles/dòng.
2. **Hiển thị thông tin chi tiết một Article (Trang chi tiết - Details):**
   * Điều hướng từ trang chủ.
   * Thông tin: Hình đại diện, mô tả chi tiết, danh sách comment.
3. **Phân trang danh sách comment trong chi tiết article:**
   * Hiển thị tối đa 3 comment/trang.
   * Thông tin comment: Nội dung, thời gian đăng.
   * Điều hướng phân trang.
4. **Thêm Comment.**
5. **Xoá Comment.**

**Build 2 (bao gồm kiểm thử hồi quy Build 1):** 6. **Đăng ký tài khoản.** 7. **Đăng nhập.** 8. **Đăng xuất.** 9. **Phân quyền:** \* User đăng nhập mới được thêm comment. \* Admin mới được xóa comment.

**Các khía cạnh/loại kiểm thử sẽ được áp dụng cho các tính năng trên:**

* **Functional Testing**
* **Performance Testing**
* **UI/UX Testing**
* **Security Testing**
* **API Testing**
* **Automation Testing** (cho các kịch bản hồi quy và lặp lại)
* **Data Generation & Integrity Testing**

**5. Features not to be Tested**

* Kiểm thử tương thích phần cứng cụ thể.
* Kiểm thử các thành phần của bên thứ ba không thuộc phạm vi phát triển của dự án (VD: framework NodeJS, thư viện ngoài, cơ sở hạ tầng mạng chi tiết).
* Kiểm thử hiệu năng sâu rộng ở mức độ stress test cực lớn vượt ngoài phạm vi công cụ và môi trường hiện có.
* Code review chi tiết của lập trình viên (thuộc trách nhiệm của Dev team).
* Usability testing chính thức với người dùng cuối thực tế (có thể thuộc giai đoạn UAT riêng).

**6. Approach**

**6.1. Tổng quan chiến lược:** Quy trình kiểm thử sẽ áp dụng phương pháp Agile lặp, với việc kiểm thử diễn ra song song và tích hợp chặt chẽ vào các chu kỳ phát triển của hai build chính.

* **Điều kiện bắt đầu kiểm thử (Entry Criteria) cho mỗi Build:**
  + Test Plan đã được phê duyệt.
  + Môi trường kiểm thử (Test Environment) đã sẵn sàng và ổn định.
  + Build phần mềm tương ứng đã được triển khai thành công lên Test Environment.
  + Các chức năng chính của build đã được dev hoàn thành và unit test pass (xác nhận từ Dev team).
  + Dữ liệu kiểm thử cơ bản đã được chuẩn bị.
* **Các cấp độ kiểm thử:**
  + **Component Testing (Unit Testing):** Thực hiện bởi Developer.
  + **Integration Testing:** Tập trung vào giao tiếp giữa frontend và API backend, và giữa các module nội bộ.
  + **System Testing:** Kiểm thử toàn bộ hệ thống đã tích hợp để xác minh các yêu cầu. Đây là trọng tâm chính của Test Team.
  + **Acceptance Testing (UAT):** Có thể được thực hiện bởi Product Owner/Stakeholders.

**6.2. Chi tiết cách tiếp cận cho từng loại kiểm thử:**

* **Functional Testing:**
  + Sử dụng các kỹ thuật black-box: Equivalence Partitioning, Boundary Value Analysis, Decision Table Testing, Use Case Testing.
  + Kiểm tra tất cả các luồng chức năng (happy paths, negative paths, edge cases).
* **Performance Testing:**
  + **Load Testing:** Kiểm tra thời gian tải trang chủ, trang chi tiết, thời gian phản hồi khi thêm/xoá comment, đăng nhập, đăng ký dưới tải người dùng giả lập.
  + Đo lường thông lượng (throughput), thời gian đáp ứng (response time), mức sử dụng tài nguyên hệ thống.
* **UI/UX Testing:**
  + Kiểm tra tính nhất quán về giao diện, dễ sử dụng, bố cục, màu sắc, font chữ.
  + Đảm bảo trải nghiệm người dùng mượt mà, trực quan.
  + Kiểm tra trên các trình duyệt mục tiêu.
* **Security Testing:**
  + Kiểm tra lỗ hổng OWASP Top 10 cơ bản (XSS, SQL Injection) cho các input (comment, form đăng ký/đăng nhập).
  + Kiểm tra cơ chế xác thực, phân quyền, quản lý session.
  + Kiểm tra việc lưu trữ mật khẩu (hashing).
* **API Testing:**
  + Kiểm tra các API endpoints: request/response, status codes, error handling, data validation.
  + Sử dụng công cụ như Postman.
* **Automation Testing:**
  + Xác định các kịch bản ứng viên cho tự động hóa (luồng chính, hồi quy, dữ liệu lặp).
  + Xây dựng và duy trì bộ script tự động.
  + Tích hợp vào quy trình (nếu có thể).
* **Data Generation & Integrity:**
  + Chuẩn bị dữ liệu test đa dạng và có ý nghĩa.
  + Kiểm tra tính toàn vẹn dữ liệu sau các thao tác (thêm, xóa, sửa).

**6.3. Quy trình quản lý lỗi (Bug Triage):**

* Lỗi được báo cáo qua hệ thống quản lý lỗi (VD: Jira).
* Phân loại lỗi theo Severity (Blocker, Critical, Major, Minor, Trivial) và Priority (High, Medium, Low).
* Họp Bug Triage định kỳ để xem xét, phân loại và lên kế hoạch sửa lỗi.

**7. Item Pass/Fail Criteria**

* **Test Case Pass/Fail:**
  + **Pass:** Kết quả thực tế khớp với kết quả mong đợi.
  + **Fail:** Kết quả thực tế không khớp với kết quả mong đợi.
* **Build/Release Pass Criteria (Exit Criteria):**
  + Tất cả các test case đã được thực thi ít nhất một lần.
  + 100% test case thuộc nhóm Critical và High priority phải Pass.
  + Tỷ lệ Pass chung của tất cả test case >= 95%.
  + Không còn lỗi Blocker hoặc Critical nào ở trạng thái Open.
  + Số lượng lỗi Major ở trạng thái Open không vượt quá X (X sẽ được thống nhất với Project Manager/Product Owner).
  + Các mục tiêu về Performance, Security, UI/UX đã được kiểm tra và đạt mức chấp nhận được.
  + Tài liệu Test Summary Report được phê duyệt.

**8. Suspension Criteria and Resumption Requirements**

* **Suspension Criteria (Điều kiện tạm dừng kiểm thử):**
  + Xuất hiện lỗi Blocker/Critical trên một chức năng chính, chặn đứng việc kiểm thử các phần quan trọng khác.
  + Môi trường kiểm thử không ổn định.
  + Build được giao có quá nhiều lỗi cơ bản.
  + Có thay đổi lớn về yêu cầu chưa được cập nhật.
* **Resumption Requirements (Điều kiện tiếp tục kiểm thử):**
  + Các lỗi Blocker/Critical gây ra việc tạm dừng đã được sửa và xác nhận.
  + Môi trường kiểm thử đã ổn định.
  + Build mới đã được triển khai với các lỗi cơ bản đã được khắc phục.

**9. Test Deliverables**

1. **Test Plan:** Tài liệu này (TP\_NODEJS\_ARTICLES\_V1.0\_20250526).
2. **Test Case Specifications:** Tài liệu mô tả chi tiết các ca kiểm thử (VD: TC\_NodeJS\_Build1\_Functional\_V1.0).
3. **Test Data:** Dữ liệu mẫu sử dụng cho kiểm thử.
4. **Test Logs/Execution Reports:** Kết quả thực thi test case từ Test Management Tool.
5. **Bug Reports:** Các báo cáo lỗi trên hệ thống quản lý lỗi.
6. **Automation Test Scripts:** Mã nguồn các kịch bản kiểm thử tự động.
7. **Test Summary Report:** Báo cáo tổng kết sau mỗi build và khi kết thúc dự án (VD: TSR\_NodeJS\_Build1\_V1.0).

**10. Testing Tasks**

1. **Hoàn thiện Test Plan:** Rà soát và nhận phê duyệt Test Plan.
2. **Thiết kế Test Cases:** Viết các test case chi tiết cho Build 1 và Build 2, bao gồm tất cả các loại kiểm thử.
3. **Chuẩn bị Test Data:** Tạo và quản lý dữ liệu kiểm thử.
4. **Thiết lập và Xác minh Môi trường Test:** Đảm bảo môi trường sẵn sàng.
5. **Phát triển Script Kiểm thử Tự động:** Cho các kịch bản hồi quy chính.
6. **Thực thi Test Cycles cho Build 1:**
   * Functional Testing
   * UI/UX Testing
   * API Testing
   * Performance Testing (sơ bộ)
7. **Thực thi Test Cycles cho Build 2:**
   * Functional Testing (tính năng mới & hồi quy Build 1)
   * UI/UX Testing (tính năng mới & hồi quy Build 1)
   * API Testing (tính năng mới & hồi quy Build 1)
   * Security Testing (tập trung vào xác thực, phân quyền)
   * Performance Testing (toàn diện hơn)
   * Chạy Automation Regression Suite.
8. **Báo cáo và Theo dõi Lỗi:** Sử dụng hệ thống quản lý lỗi.
9. **Kiểm thử lại Lỗi (Retesting & Regression Testing):** Sau khi lỗi được sửa.
10. **Chuẩn bị Test Summary Report:** Cho từng build và cuối dự án.
11. **Họp định kỳ:** Với Dev team và PM để cập nhật tiến độ và vấn đề.

**11. Environmental Needs**

* **Phần cứng:**
  + Máy chủ Test: Cấu hình tương đương môi trường Production (CPU, RAM, Disk Space).
  + Máy tính cho Testers: Đủ mạnh để chạy các công cụ test, trình duyệt.
* **Phần mềm:**
  + Hệ điều hành: Windows, Linux, MacOS cho máy testers.
  + NodeJS runtime, Database (MongoDB, PostgreSQL, etc. - theo dự án) trên server test.
  + Trình duyệt: Chrome, Firefox, Edge (phiên bản mới nhất).
* **Công cụ Kiểm thử:**
  + **Test Management:** Jira (với Xray/Zephyr), TestRail.
  + **Bug Tracking:** Jira.
  + **API Testing:** Postman.
  + **UI Automation:** Selenium/Cypress/Playwright.
  + **Performance Testing:** JMeter/k6.
  + **Security Testing:** OWASP ZAP/Burp Suite (Community).
  + **Version Control:** Git.
* **Dữ liệu:**
  + Dữ liệu mẫu cho Articles, Comments, Users (Regular, Admin).

**12. Responsibilities**

| **Vai trò** | **Trách nhiệm chính** |
| --- | --- |
| Test Manager | Lập kế hoạch, điều phối, quản lý tài nguyên, theo dõi tiến độ, báo cáo, quản lý rủi ro, đảm bảo chất lượng, phê duyệt test deliverables. |
| Tester (4) | Thiết kế và thực thi test case, báo cáo lỗi, kiểm thử lại, phát triển và chạy automation scripts, chuẩn bị dữ liệu test. |
| Developer Team | Sửa lỗi, cung cấp thông tin kỹ thuật, triển khai build lên môi trường test, thực hiện unit testing. |
| Project Manager | Quản lý tổng thể dự án, ra quyết định dựa trên báo cáo kiểm thử, phê duyệt Test Plan. |

**13. Staffing and Training Needs**

* **Nhân sự:**
  + 1 Test Manager có kinh nghiệm.
  + 4 Testers có kỹ năng về manual testing, API testing, và có kiến thức cơ bản về automation, performance, security testing.
* **Đào tạo:**
  + Đào tạo về sản phẩm và các yêu cầu nghiệp vụ.
  + Đào tạo sử dụng các công cụ kiểm thử mới (nếu có).
  + Khuyến khích tự học và chia sẻ kiến thức về các kỹ thuật kiểm thử nâng cao.

**14. Schedule** *(Tổng thời gian dự kiến: 10 tuần)*

* **Tuần 1-2: Lập kế hoạch & Chuẩn bị**
  + Test Plan finalization & approval.
  + Test Case design for Build 1.
  + Test Environment setup & verification.
  + Initial Automation framework setup and script development.
  + Test Data preparation for Build 1.
* **Tuần 3-5: Thực thi kiểm thử Build 1**
  + Execution of Functional, UI/UX, API tests for Build 1.
  + Initial Performance tests for Build 1.
  + Bug reporting & retesting.
  + Test Case design for Build 2.
* **Tuần 6-8: Thực thi kiểm thử Build 2 & Hồi quy**
  + Execution of Functional, UI/UX, API tests for Build 2 features.
  + Comprehensive Regression testing for Build 1 features.
  + Security Testing.
  + Full Performance Testing.
  + Expand and run Automation Suite.
  + Bug reporting & retesting.
* **Tuần 9-10: Hoàn thiện, Báo cáo & Đóng gói**
  + Final test execution cycles.
  + Full Automation Regression run.
  + Sanity/Smoke test on Release Candidate.
  + Test Summary Report generation and approval.
  + Knowledge transfer/Documentation finalization.

**15. Risks and Contingencies**

| **Rủi ro tiềm ẩn** | **Mức độ Ảnh hưởng** | **Khả năng Xảy ra** | **Biện pháp Giảm thiểu/Dự phòng** |
| --- | --- | --- | --- |
| 1. Yêu cầu thay đổi thường xuyên | Cao | Trung bình | Áp dụng quy trình quản lý thay đổi chặt chẽ. Cập nhật Test Plan/Test Cases kịp thời. Tăng cường giao tiếp với Product Owner. |
| 2. Môi trường Test không ổn định/không sẵn sàng | Cao | Trung bình | Phối hợp sớm với Dev/SysAdmin. Có kế hoạch dự phòng cho việc thiết lập môi trường. |
| 3. Chất lượng Build ban đầu thấp | Cao | Trung bình | Thống nhất Entry Criteria chặt chẽ hơn cho build. Yêu cầu Dev thực hiện Unit Test/Smoke Test kỹ lưỡng trước khi giao build. |
| 4. Trễ tiến độ từ đội phát triển | Cao | Trung bình | Theo dõi sát tiến độ dev. Điều chỉnh lịch trình kiểm thử linh hoạt. Có thể cần thêm giờ hoặc ưu tiên kiểm thử các phần quan trọng. |
| 5. Thiếu hụt nhân sự (ốm đau, nghỉ việc) | Trung bình | Thấp | Đào tạo chéo kỹ năng cho các thành viên. Có kế hoạch nhân sự dự phòng (nếu có thể). Ưu tiên lại các task kiểm thử. |
| 6. Vấn đề về hiệu năng không được phát hiện sớm | Cao | Trung bình | Thực hiện Performance testing sớm ở các module quan trọng, không đợi đến cuối. |
| 7. Lỗ hổng bảo mật nghiêm trọng phát hiện muộn | Cao | Thấp | Thực hiện Security testing sớm, đặc biệt với các module xác thực, phân quyền. Tăng cường nhận thức bảo mật cho team. |

**16. Approvals**

| **Tên** | **Chức danh** | **Chữ ký** | **Ngày** |
| --- | --- | --- | --- |
| (Tên Test Manager) | Test Manager |  |  |
| (Tên Project Manager) | Project Manager |  |  |
| (Tên Product Owner) | Product Owner |  |  |