

Hướng dẫn viết testcase

- [BÍ KÍP VIẾT TEST CASE](#)
- [Các trường hợp kiểm thử phổ biến nhất theo từng Loại Test](#)
- [Các bước viết testcase](#)
- [Bộ testcase mẫu](#)

BÍ KÍP VIẾT TEST CASE

PHẦN 1: THÔNG TIN NHẬN DIỆN

Đây là phần giúp quản lý hàng ngàn test case mà không bị rối.

- **STT (Mã Test Case):** Giống như số CCCD của kịch bản đó. Phải là duy nhất.
 - Ví dụ: TCSG_QLVB_001 (Test Case Sài Gòn _ Quản Lý Văn Bản _ Số 001).
- **Mã chức năng / Tên chức năng:** Gắn với nhóm tính năng lớn.
 - Ví dụ: F1002 / Thêm mới văn bản đi.
- **Loại test:** Hệ thống có nhiều tầng, bạn phải xác định mình đang test phần nào:
 - *Giao diện (UI):* Nút bấm có đúng màu không? Chữ có ngay ngắn không?
 - *Chức năng (Functional):* Bấm lưu có lưu được không?
 - *API:* test xem dữ liệu trả về từ máy chủ có chuẩn không.
- **Tên trường hợp kiểm thử (Tiêu đề):** Phải ngắn gọn, đi thẳng vào vấn đề. Đọc 1 câu là biết đang test cái gì.
 - Ví dụ MẪU CHUẨN: "Kiểm tra hiển thị chức năng Thêm mới dự thảo văn bản trình ký" hoặc "Kiểm tra giá trị mặc định của trường Đơn vị soạn thảo".

PHẦN 2: BỐI CẢNH & CHUẨN BỊ

- **Tên tác nhân chính (Actor):** Ai là người thực hiện hành động này? (VD: Cán bộ, Quản trị viên, Khách). Mỗi người sẽ có quyền hạn khác nhau.
- **Điều kiện trước (Pre-condition):** Tình trạng hệ thống hoặc dữ liệu BẮT BUỘC PHẢI CÓ trước khi bắt đầu test.
 - Ví dụ: "Tài khoản cán bộ đã đăng nhập thành công và đang đứng ở màn hình Thêm mới văn bản đi".
 - Ví dụ (với API): "Phải có token truy cập của tài khoản hợp lệ".

PHẦN 3: HÀNH ĐỘNG THỰC TẾ

- **Mô tả các bước thực hiện:** Viết theo dạng đánh số 1, 2, 3... Không viết văn xuôi dài dòng.
 - Ví dụ:

- Nhấn vào dấu
 - Chọn chức năng "Thêm mới"
- Test Data (Dữ liệu kiểm thử):** Nếu bước thực hiện yêu cầu nhập liệu, bạn phải cung cấp luôn dữ liệu để người test nhập.
 - Ví dụ: , , .

PHẦN 4: KẾT QUẢ

- Kết quả mong đợi (Expected Result):** ĐÂY LÀ CỘT QUAN TRỌNG NHẤT. Sau khi làm xong các bước trên, hệ thống BẮT BUỘC phải phản hồi như thế nào? Cần mô tả chi tiết: Hệ thống chuyển đi đâu? Hiện thị thông báo gì? Trạng thái dữ liệu ra sao?
 - Ví dụ: 1. Hệ thống chuyển đến màn hình "Thêm mới dự thảo văn bản trình ký". 2. Các trường nhập liệu hiển thị đúng label, placeholder, mặc định rỗng.
- Kết quả thực tế (Actual Result):** Chỗ này để trống khi viết. Khi nào Tester chạy thực tế thì mới điền vào xem nó có đúng như "Kết quả mong đợi" không.
- Kết quả lần 1 / Lần 2 (Pass/Fail/Untest):** * (Đạt): Kết quả thực tế = Kết quả mong đợi.
 - (Không đạt): Có lỗi (Bug), cần báo Dev sửa.
 - : Chưa test tới.

☐☐ 3 NGUYÊN TẮC VÀNG ĐỂ CÓ MỘT TEST CASE "XỊN"

- Độc lập (Atomic):** Một Test Case chỉ nên kiểm tra **MỘT** thứ duy nhất. Đừng gộp "Kiểm tra Thêm, Sửa, Xóa" vào chung một dòng. Lỡ tính năng Sửa bị lỗi, bạn sẽ không biết đánh trạng thái dòng đó là Pass hay Fail.
- Rõ ràng, không dùng từ mơ hồ:** Không dùng các từ như "Hiển thị đẹp", "Tốc độ nhanh", "Thông báo phù hợp". Hãy viết chuẩn xác: "Hiển thị đúng màu đỏ", "Tốc độ dưới 3 giây", "Hiển thị thông báo 'Tạo mới thành công'".
- Tư duy phá bình (Happy path & Unhappy path):** Đừng chỉ test những trường hợp người dùng nhập đúng (Happy path). Hãy test cả những lúc người dùng cố tình làm sai (Unhappy path) như: Nhập sai mật khẩu, bỏ trống ô bắt buộc nhập, upload file quá dung lượng...

Các trường hợp kiểm thử phổ biến nhất theo từng Loại Test

1. Test Giao diện (UI/UX - User Interface/User Experience)

Các trường hợp kiểm thử (Test Cases) thường gặp:

- **Kiểm tra tính đồng nhất (Consistency):** Font chữ, màu sắc, kích thước nút bấm có đúng với bản thiết kế (Figma/Design Guideline) không?
- **Kiểm tra hiển thị văn bản:** Có bị lỗi chính tả không? Chữ có bị tràn ra ngoài khung hoặc bị che khuất không?
- **Kiểm tra trên nhiều thiết bị (Responsive):** Thu nhỏ màn hình máy tính lại, hoặc mở trên điện thoại, iPad thì bố cục có bị vỡ không?
- **Kiểm tra các trạng thái của phần tử:** Rê chuột (hover) vào nút bấm nó có đổi màu không? Bấm vào (active) có hiệu ứng gì không? Tooltip (chữ nhỏ hiện lên khi đưa chuột vào) có đúng không?
- **Kiểm tra thông báo rủi ro:** Bấm nút "Xóa" hệ thống có hiện popup hỏi lại "Bạn có chắc chắn muốn xóa không?" hay là xóa luôn (lỗi nguy hiểm)?

2. Test Chức năng (Functional Test)

Các trường hợp kiểm thử (Test Cases) thường gặp:

- **Trường hợp luồng chuẩn (Happy Path):** Nhập đúng mọi thông tin, làm đúng mọi bước -> Phải ra kết quả thành công (Ví dụ: Thêm mới văn bản đi thành công).
- **Trường hợp rẽ nhánh/lỗi (Unhappy Path - Rất quan trọng):**
 - **Bỏ trống:** Không nhập các ô có dấu sao đỏ (*) -> Phải báo lỗi bắt buộc nhập.
 - **Nhập sai định dạng:** Ô Email mà nhập `nguyenvan_a` (thiếu @gmail.com), ô Số điện thoại mà nhập chữ cái -> Phải báo lỗi định dạng.
 - **Vượt quá giới hạn:** Ô Tên chỉ cho nhập 50 ký tự, cố tình copy đoạn văn dài 100 ký tự dán vào -> Hệ thống chặn lại không cho nhập hoặc báo lỗi.

- **Kiểm tra phân quyền (Authorization):** Tài khoản "Nhân viên" có lên nhìn thấy được "Báo cáo lương của Giám đốc" không? (Nút xem báo cáo phải bị ẩn hoặc bấm vào báo lỗi "Không có quyền").

3. Test API (Giao tiếp giữa Ứng dụng và Máy chủ)

Các trường hợp kiểm thử (Test Cases) thường gặp:

- **API trả về đúng dữ liệu (Mã 200 OK):** Gửi yêu cầu lấy "Chi tiết văn bản đi số 1" -> API trả về đúng file JSON chứa ID = 1 và thông tin tương ứng.
- **Kiểm tra bảo mật Token (Mã 401 Unauthorized):** Cố tình gọi API nhưng không đính kèm thẻ bảo mật (Token), hoặc dùng Token đã hết hạn -> Hệ thống phải đuổi ra, không cho lấy dữ liệu.
- **Gửi thiếu/sai đầu vào (Mã 400 Bad Request):** API yêu cầu ID phải là số, cố tình gửi ID là chữ "abc" -> Báo lỗi dữ liệu không hợp lệ.
- **Không có quyền truy cập (Mã 403 Forbidden):** Token hợp lệ, nhưng Token đó là của người không có quyền, cố gọi API của người người có quyền
- -> Bị chặn.

4. Test Cơ sở dữ liệu (Database / SQL Test)

Các trường hợp kiểm thử (Test Cases) thường gặp:

- **Kiểm tra đối chiếu dữ liệu (Data Mapping):** Thêm mới nhân viên A trên màn hình web. Vào database viết lệnh `SELECT` -> Dữ liệu của nhân viên A phải nằm đúng ở bảng `NhanVien`, không bị thiếu cột nào, dấu tiếng Việt không bị lỗi font.
- **Kiểm tra Xóa mềm (Soft Delete):** Khi bấm "Xóa" trên web, hệ thống biến mất. Nhưng vào database kiểm tra, dòng đó không bị xóa hẳn (lệnh `DELETE`) mà chỉ được cập nhật trạng thái `IsDeleted = 1` (lệnh `UPDATE`) để phòng hờ sau này khôi phục lại.
- **Kiểm tra ràng buộc:** Nếu bảng quy định cột `Email` là duy nhất (Unique), cố tình chạy lệnh `INSERT` một người mới trùng email với người cũ -> Database phải văng ra lỗi.

5. Test Tích hợp (Integration Test)

Các trường hợp kiểm thử (Test Cases) thường gặp:

- **Luồng dữ liệu qua lại:** Khi bộ phận A (Module Dự thảo) duyệt văn bản xong -> Bộ phận B (Module Văn bản đi) có tự động xuất hiện bản ghi đó không?
- **Tích hợp bên thứ 3:** Khi bấm thanh toán VNPay trên web của mình, hệ thống có nhảy đúng sang trang của VNPay không? Thanh toán xong có quay về báo "Thành công" trên

web mình không?

Các bước viết testcase

1. Lý thuyết chung – Tại sao phải viết test case theo chuẩn này?

- **Mục tiêu cuối cùng:** Đảm bảo **100% coverage** các khía cạnh: **Giao diện + Chức năng + Tích hợp + API + SQL + Ghi log + Điều hướng & UX.**
- **Nguyên tắc cốt lõi:**
 - Mọi test case phải **cụ thể, kiểm chứng được** (có thể verify bằng mắt, bằng DB, bằng response API).
 - Phải có **test data + SQL query** để kiểm tra dữ liệu thực tế.
 - Luôn bao gồm cả **positive case** (thành công) và **negative case** (lỗi, biên giới).
 - Mọi hành động quan trọng phải có **Ghi log** và **DB verification**.
 - Tuân thủ **UI/UX Design Guideline** (font, màu, spacing, responsive...).

Quy ước chung (bắt buộc):

- **ID test case:** Dự án_module_xxx (xxx tăng dần liên tục). Ví dụ: TCSG_DHVB_001
- **Mã chức năng:** Mã chức năng của module/màn hình (BA cung cấp).
- **Loại test:** Giao diện / Chức năng / Tích hợp / API / SQL / Ghi log / Điều hướng & UX.
- **UAT:** Có / Không. (Để chọn các happy case thêm vào tài liệu UAT)
- **Test Data:** Là dữ liệu mẫu cụ thể để thực hiện test (Có thể là câu truy vấn SQL hoặc dữ liệu nhập liệu).

2. Template chung cho 1 test case (dùng cho mọi loại)

Cột	Nội dung cần điền
Mã chức năng	F1002
ID	TCSG_QLVB_001
Loại test	Giao diện / Chức năng...
UAT	Có / Không
Tên chức năng	Thêm mới văn bản đến
Tên tác nhân chính	Văn thư

Cột	Nội dung cần điền
Điều kiện trước	Màn hình Thêm mới văn bản đến đã mở
Test Data	Để trống
Mô tả trường hợp sử dụng	Kiểm tra trường Sổ văn bản*
Tên các trường hợp kiểm thử chức năng	Kiểm tra để trống trường Sổ văn bản*
Mô tả các bước thực hiện kiểm thử	1. Để trống trường... 2. Nhập thông tin còn lại hợp lệ 3. Nhấn Lưu
Kết quả mong đợi	Hiển thị thông báo bên dưới textbox "Vui lòng chọn Sổ văn bản." + highlight trường
Kết quả thực tế	
Ghi chú	(nếu có)

3. Test case mẫu chi tiết theo từng Loại test

3.1. Loại test: Giao diện (UI/UX)

Mục đích: Kiểm tra tuân thủ Design Guideline, tính trực quan, lỗi hiển thị.

Test case bắt buộc phải có ở đầu chức năng/màn khi viết/test testcase:

Kiểm tra giao diện tổng quan của màn hình dựa trên thiết kế đã có (Ví dụ: Figma,.. - tùy theo dự án)

4	Mã chức năng	ID	Loại test	UAT	Tên chức năng	Tên tác nhân chính	Điều kiện trước	Test Data	Mô tả trường hợp sử dụng	Tên các trường hợp kiểm thử chức năng	Mô tả các bước thực hiện kiểm thử	Kết quả mong đợi
5	Văn bản đến											
6	F1002	TCSG_QLVB_001	Giao diện	Không	Thêm mới văn bản đến	Văn thư	Màn hình Thêm mới văn bản đến		Thêm mới văn bản đến	Kiểm tra Hiển thị chức năng Thêm mới tiếp nhận văn bản đến	1. Nhấn dấu + Chon chức năng thêm mới	1. Hệ thống điều hướng đến màn hình 'Thêm mới văn bản đến' và hiển thị đầy đủ các thành phần giao diện theo thiết kế 2. Các trường nhập liệu hiển thị đúng label, placeholder, mặc định rỗng (trừ các trường có giá trị mặc định) 3. Tuân thủ các quy tắc về giao diện: Giao diện chung

Các quy tắc về giao diện sẽ được quy ước chung

Test case mẫu 1: Để trống trường bắt buộc

- Mã chức năng: F1002
- ID: TCSG_QLVB_002
- Loại test: Giao diện
- Tên chức năng: Thêm mới văn bản đến

- Tên tác nhân chính: Văn thư
- Tên các trường hợp kiểm thử: Kiểm tra để trống trường Sổ văn bản*
- Các bước:
 - 1. Để trống trường Sổ văn bản.
 - 2. Nhập thông tin còn lại hợp lệ.
 - 3. Nhấn Lưu.
- Kết quả mong đợi: Hiển thị thông báo “Vui lòng chọn Sổ văn bản.” + highlight & focus trường.

Test case mẫu 2: Kiểm tra hiển thị mặc định

- ID: TCSG_QLVB_003
- Tên các trường hợp kiểm thử: Kiểm tra giá trị mặc định trường Sổ văn bản
- Kết quả mong đợi: Hiển thị default text “<mặc định để trống>”.

Test case mẫu 3 (Date field): Kiểm tra hộp Calendar

- Tên các trường hợp kiểm thử: Kiểm tra hoạt động hộp Calendar trường Ngày đến*
- Các bước:
 - 1. Click icon Calendar.
 - 2. Chọn ngày.
 - 3. Nhấn Đồng ý.
- Kết quả mong đợi: Hộp đóng lại + ngày hiển thị đúng định dạng dd/mm/yyyy.

Test case mẫu 4 (File upload): Kiểm tra tải lên file

- Tên các trường hợp kiểm thử: Kiểm tra tải file không đúng định dạng
- Kết quả mong đợi: Hiển thị thông báo upload không hợp lệ.

Test case mẫu 5 (Dropdown): Kiểm tra tìm kiếm trong dropdownlist (nếu có)

- Tên các trường hợp kiểm thử: Kiểm tra nhập tìm kiếm dữ liệu không tồn tại
- Kết quả mong đợi: Hiển thị “Không có dữ liệu”.

3.2. Loại test: Chức năng (Functional) + SQL

Mục đích: Kiểm tra logic nghiệp vụ và dữ liệu lưu đúng DB.

Test case mẫu 1: Lưu thành công với dữ liệu đầy đủ

- Loại test: Chức năng
- Tên các trường hợp kiểm thử: Kiểm tra thêm mới văn bản thành công (nhập đầy đủ tất cả trường)
- Test Data: `SELECT * FROM incomming_documents WHERE to_book = N'UBND-TL1/2'`
- Các bước:
 - 1. Nhập đầy đủ thông tin hợp lệ.
 - 2. Nhấn Lưu.

- 3. Thực hiện SELECT trên DB.
- 4. Đối chiếu dữ liệu.
- Kết quả mong đợi: Thông báo “Lưu thành công” + DB trả về đúng 1 dòng, dữ liệu khớp 100%.

Test case mẫu 2: Lưu thành công với các trường bắt buộc

- Tên các trường hợp kiểm thử: Kiểm tra thêm mới khi chỉ nhập các trường bắt buộc
- Các bước:
 - 1. Nhập các trường bắt buộc, không nhập các trường bắt buộc.
 - 2. Nhấn Lưu.
 - 3. Thực hiện SELECT trên DB.
 - 4. Đối chiếu dữ liệu.
- Kết quả mong đợi: Lưu thành công, các trường không bắt buộc để trống hoặc giá trị mặc định.

3.3. Loại test: API

Mục đích: Kiểm tra endpoint, validation, quyền, response code.

Test case mẫu 1

- Loại test: API
- Tên các trường hợp kiểm thử: Kiểm tra Tạo mới văn bản thành công (full fields)
- Test Data: Body JSON (đầy đủ) + token hợp lệ
- Các bước:
 - 1. Gửi POST request với body hợp lệ.
 - 2. Kiểm tra response.
 - 3. Kiểm tra DB.
- Kết quả mong đợi: Status 201 + message “Document created successfully” + DB có bản ghi mới.

Test case mẫu 2

- Tên các trường hợp kiểm thử: Kiểm tra thiếu trường bắt buộc
- Các bước:
 - 1. Gửi POST request với body thiếu trường bắt buộc.
 - 2. Kiểm tra response.
 - 3. Kiểm tra DB.
- Kết quả mong đợi: Status 400 + message “Missing required field: So_van_ban”.

Test case mẫu 3

- Tên các trường hợp kiểm thử: Kiểm tra user không có quyền
- Các bước:
 - 1. Gửi POST request với token của user không có quyền

- 2. Kiểm tra response.
- 3. Kiểm tra DB.
- Kết quả mong đợi: Status 403 Forbidden.

Test case mẫu 4

- Tên các trường hợp kiểm thử: Kiểm tra thiếu token
- Các bước:
 - 1. Gửi POST request với token để trống.
 - 2. Kiểm tra response.
 - 3. Kiểm tra DB.
- Kết quả mong đợi: Status 401 Unauthorized.

3.4. Loại test: Tích hợp (Integration)

Mục đích: Kiểm tra các module liên kết với nhau.

Test case mẫu

- Loại test: Tích hợp
- Tên các trường hợp kiểm thử: Kiểm tra lưu văn bản tiếp nhận thành công vào Sổ văn bản
- Các bước:
 - 1. Thêm mới văn bản tiếp nhận.
 - 2. Truy cập Sổ văn bản đến.
- Kết quả mong đợi: Văn bản mới xuất hiện đúng trong sổ + file đính kèm được lưu vào bảng file_relations.

3.5. Loại test: Ghi log

Mục đích: Đảm bảo mọi hành động quan trọng đều được ghi lại.

Test case mẫu

- Loại test: Ghi log
- Tên các trường hợp kiểm thử: Kiểm tra Ghi log khi lưu văn bản
- Các bước:
 - 1. Thêm mới văn bản thành công.
 - 2. Truy cập log hệ thống.
- Kết quả mong đợi:
 - Ghi đầy đủ:
 - Tài khoản, Họ tên, Đơn vị, Mô tả thao tác (“Thêm mới/Truy cập/.....”), Trạng thái, IP, Thời gian.

3.6. Loại test: Điều hướng & UX

Mục đích: Kiểm tra trải nghiệm người dùng mượt mà, không gây bất ngờ.

Test case mẫu 1

- Loại test: Điều hướng và UX
- Tên các trường hợp kiểm thử: Kiểm tra Button Hủy thêm mới
- Kết quả mong đợi: Không lưu dữ liệu + quay về danh sách.

Test case mẫu 2

- Tên các trường hợp kiểm thử: Kiểm tra confirm xóa file
- Kết quả mong đợi: Popup xác nhận xuất hiện → Chọn Hủy → file vẫn còn.

4. Checklist bắt buộc khi viết xong 1 test case

- Có **Test Data + SQL query** rõ ràng
- Có cả **positive & negative case**
- Có **DB verification** (SELECT)
- Có **API test** song song (nếu là chức năng CRUD)
- Có **Ghi log** cho action quan trọng
- File upload/download/view/delete đầy đủ
- Date field kiểm tra format + logic (quá khứ/hiện tại/tương lai)
- Dropdown/search/empty/invalid data
- Export theo filter (có dữ liệu / không dữ liệu)
- UX: confirm, cancel, navigation không lag

Bộ testcase mẫu

https://docs.google.com/spreadsheets/d/13fRfr2Q5mV38KbYCgcttrCDErk_5btP1jc0h5F6MgeY/edit?usp=sharing