

SỔ TAY FLOW & USE CASE THỰC CHIẾN — AGILE 6-ROLE SQUAD

- **Bạn chỉ giao việc:** Tôi (/squad) tự bật Squad, tự điều phối 6 Agents & báo cáo tiến độ. ► **Ép 100% Full Flow:** Hộp /bmad-party-mode + phỏng vấn /grill-me từng câu 1.
- **Tách bạch TDD:** Bước 5 (Viết & Khóa Test RED) → Bước 6 (Viết Code GREEN).

🌟 Auto Step 1: spawn-squad --ensure

🔪 /grill-me 1-by-1

🔴 Bước 5: TDD RED

🟢 Bước 6: TDD GREEN

1. Flow Điều Phối Tổng Quan (Bạn → Điều Phối Viên /squad → 6 Agents)

Ngắn gọn • Không thao tác tay • Mọi báo cáo quy về 1 mối

👤 CHẶNG 1: BẠN (STAKEHOLDER)

- 1. **Giao bài toán:** Gõ /squad <yêu cầu>.
- 2. **Trả lời /grill-me:** Chọn phương án (Recommended) từng câu 1.
- 3. **Chốt & Nghiệm thu:** Gõ "OK chốt" → Nhận link PR sạch để Merge.



🌟 CHẶNG 2: TÔI — ĐIỀU PHỐI VIÊN (/squad)

- Tự động Bước 1:** Chạy spawn-squad --ensure vinasat-1 nếu Squad chưa bật.
- Chủ trì /grill-me:** Quét codebase trước, hỏi bạn từng câu 1 cho đến khi rõ 100%.
- Điều phối & Báo cáo:** Bắn lệnh qua squad-msg, theo dõi qua squad-status.



🧑‍🤖 CHẶNG 3: ĐỘI HÌNH 6 PANES (HERDR + BMAD)

- [1] PO-Bao Long + [2] SA-Minh Tri: Khóa prd.md, architecture.md (OpenAPI) & .story.md.
- [4] BE-Quoc Huy + [5] FE-Hoang Nam: Bước 5 (Khóa Test RED) → Bước 6 (Code GREEN trên Worktree).
- [2] SA (Gate 3.5) → [6] QA-Mai Linh (Gate 4): Duyệt kiến trúc, test API & Browser E2E.

2. Sơ Đồ Flow 9 Bước Thực Thi Chuẩn Hóa

Bước 1 tự động • Bước 3 hỏi /grill-me • Bước 5 viết Test RED • Bước 6 viết Code GREEN

BƯỚC 1 • AUTO

🌟 Điều Phối Viên

Tự Động Bật Squad 6 Panes

- Kiểm tra Herdr: nếu VINASAT-1 đã mở đủ 6 panes thì dừng luôn.
- Nếu chưa mở: tự chạy squad-init, bật 6 panes, đợi cả 6 báo idle.

spawn-squad --ensure vinasat-1

BƯỚC 2 • INTAKE

👤 Bạn → 🌟 /squad

Nhận Việc → Ép 100% Full Flow

- Bạn mô tả tính năng/bug cần làm.
- Mặc định kích hoạt **Full Flow** (không đi tắt).
- Tự động quét codebase bằng codebase-memory-mcp.

/squad <Mô tả tính năng>

BƯỚC 3 • GRILL-ME

📅 PO + 🏢 SA ↔ 👤 Bạn

Họp Team & Hỏi Từng Câu 1

- 5 Roles tổng hợp góc nhìn & rủi ro kỹ thuật.
- Hỏi bạn **từng câu 1** kèm phương án (Recommended).
- Bạn gõ **"OK chốt"** → Khóa prd.md & architecture.md (OpenAPI).

/bmad-party-mode + /grill-me

BƯỚC 4 • SHARDING

📅 PO + 🎯 SM

Tách Story & Tạo Git Worktree

- Tách file _bmad-output/stories/*.story.md cho BE & FE.
- Đính kèm tiêu chí **BDD (Given/When/Then)**.
- Tạo Git Worktree riêng (.worktrees/<id>) chống đụng branch.

sprint-status.yaml → ready-for-dev

BƯỚC 5 • TDD RED 🔴

⚙️ BE + 🧑‍🤖 FE

Viết & Khóa Bộ Test Fail Trước

- 5a:** Tạo hàm/route stub 501 & import OpenAPI types.
- 5b:** Viết 100% Unit/API/Component test theo Given/When/Then.
- 5c:** Chạy test xác nhận **FAIL (RED)** → Commit khóa test!

git commit -m "test(red): lock tests"

BƯỚC 6 • TDD GREEN 🟢

⚙️ BE + 🧑‍🤖 FE

Code Song Song Làm Xanh Test

- BE (Claude) & FE (Agy) code logic thật & UI song song.
- Cấm xóa/làm yếu assertion** của bộ test Bước 5.
- Chạy pass **100% GREEN** + tsc --noEmit → Mở PR.

pytest / vitest → 100% GREEN

BƯỚC 7 • GATE 3.5

🏢 [2] SA-Minh Tri

SA Duyệt Kiến Trúc & TDD

- Kiểm tra PR khớp 100% OpenAPI Contract BE ↔ FE.
- Xác nhận bộ test Bước 5 không bị sửa yếu ở Bước 6.
- Soi Deep Modules, 12 Fowler Code Smells, N+1 Query.

squad-msg sa "[SM->SA] Gate 3.5"

BƯỚC 8 • GATE 4

🧑‍🤖 [6] QA-Mai Linh

QA Kiểm Thử API & Browser E2E

- Chạy lại bộ test TDD Bước 5/6.
- Bắn API thật (200/400/401/403/404).
- Mở Playwright/DevTools test UI thật theo từng dòng BDD.

squad-msg qa "[SM->QA] Gate 4"

BƯỚC 9 • CLOSURE

🎯 SM → 🌟 Báo Bạn

Đóng Sprint & Báo Bạn Merge PR

- Cập nhật sprint-status.yaml → done.
- Tổng hợp kết quả qua squad-status.
- Báo cáo bạn nghiệm thu & gửi link PR để Merge!

squad-status

3. Use Case Chi Tiết Từng Bước: "Tạm Khóa Tài Khoản CTV Vi Phạm + Lưu Lịch Sử Audit"

Theo dõi chính xác từng câu lệnh, câu hỏi /grill-me và kết quả ở cả 9 bước

BƯỚC 1 Auto-Spawn	Bạn → Điều Phối Giao việc 1 dòng	👤 BẠN GỒ: /squad Làm tính năng Tạm khóa CTV vi phạm: Admin bấm Tạm khóa -> nhập lý do -> chặn login Web/App & lưu lịch sử.	Tôi tự chạy spawn-squad --ensure vinasat-1. 6 panes (PO-Bao Long → QA-Mai Linh) sẵn sàng ở trạng thái idle. Xuất danh bạ _bmad-output/active-squad.json.
BƯỚC 2 Codebase Recon	Điều Phối + SA + PO Tự đọc code trước	📁 HỆ THỐNG TỰ CHẠY: search_graph(name_pattern=".*Collaborator.*") api_inspect_endpoint(path="/api/cms/collaborators")	Phát hiện CTV dùng JWT trên cả Web & Mobile (/api/mobile/*). Phát hiện model Collaborator chưa có bảng lưu lịch sử khóa/mở khóa. Chuẩn bị câu hỏi /grill-me cho các điểm chưa rõ nghiệp vụ.
BƯỚC 3 /grill-me 1-by-1	PO + SA ↔ Bạn Hỏi từng câu 1	🔍 CÂU HỎI 1 → BẠN TRẢ LỜI: Hỏi: CTV đang login bị khóa thì xử lý sao? • (Recommended) A: Thu hồi JWT ngay + trả 403. 👤 Bạn: "Chọn A. Mở khóa cũng bắt nhập lý do >=10 ký tự. OK chốt!"	PO-Bao Long khóa _bmad-output/prd.md. SA-Minh Tri khóa _bmad-output/architecture.md: Endpoint: POST /api/cms/collaborators/{id}/suspend & unsuspend.
BƯỚC 4 Sharding + Worktree	PO-Bao Long + SM Tách 2 Story BE/FE	📁 FILE & WORKTREE SINH RA: _bmad-output/stories/1-1-suspend-be.story.md _bmad-output/stories/1-2-suspend-fe.story.md .worktrees/1-1-suspend-be & .worktrees/1-2-suspend-fe	Mỗi file .story.md ghi rõ kịch bản Given / When / Then. SM-Thanh Tung đổi sprint-status.yaml sang ready-for-dev. Tạo 2 Git Worktree độc lập cho BE và FE.
BƯỚC 5 TDD RED (Viết Test)	BE-Quoc Huy FE-Hoang Nam Khóa Test Fail trước	🔧 CHẠY XÁC NHẬN RED & KHÓA TEST: pytest apps/collaborators/tests/test_suspend.py # -> FAIL (501) vitest run SuspendModal.test.tsx # -> FAIL git commit -m "test(red): lock failing TDD tests"	BE: Tạo route stub trả 501 + viết test khóa thành công (200), thiếu lý do (400), chặn JWT (403). FE: Import OpenAPI types + viết test Modal (4 states: Loading/Empty/Error/Success). Chạy xác nhận RED → Commit khóa cứng bộ test!
BƯỚC 6 TDD GREEN (Viết Code)	BE-Quoc Huy FE-Hoang Nam Code làm xanh Test	🔧 CHẠY XÁC NHẬN GREEN & MỞ PR: pytest apps/collaborators/tests/test_suspend.py # -> 100% PASS tsc --noEmit && vitest run SuspendModal.test.tsx # -> 100% PASS git commit -m "feat: implement CTV suspend & audit log"	BE: Viết model CollaboratorSuspensionLog, service thu hồi token & middleware 403. FE: Code SuspendModal.tsx + bảng lịch sử trạng thái. 100% test Bước 5 chuyển sang GREEN (không sửa yếu assert) → Mở PR.
BƯỚC 7 Gate 3.5 (SA)	SA-Minh Tri Review Kiến Trúc	📁 LỆNH ĐIỀU PHỐI: squad-msg sa "[SM->SA Gate 3.5] Review PR #101 & #102 cho story 1-1 và 1-2."	Đối chiếu payload BE ↔ FE khớp 100% OpenAPI Contract. Kiểm tra git diff: bộ test Bước 5 giữ nguyên vẹn 100%. Không có N+1 query hay Code Smells → Phán quyết APPROVED.
BƯỚC 8 Gate 4 (QA)	QA-Mai Linh Test API & Browser E2E	📁 LỆNH ĐIỀU PHỐI: squad-msg qa "[SM->QA Gate 4] Verify E2E & API cho story 1-1 và 1-2."	Bắn API thật kiểm tra 200 OK, 400 (lý do ngắn), 403 ACCOUNT_SUSPENDED. Mở Playwright bấm nút Tạm khóa trên trình duyệt, kiểm tra 0 lỗi Console. Phán quyết PASS (Gate 4 Approved) gửi về SM.
BƯỚC 9 Closure & Merge	SM → Báo Bạn Nghiệm thu hoàn tất	📊 BÁO CÁO TIẾN ĐỘ: squad-status # 1-1-suspend-be: done 1-2-suspend-fe: done	SM-Thanh Tung cập nhật sprint-status.yaml → done. Tôi tổng hợp báo cáo nghiệm thu (Test Pass, Gate 3.5 & Gate 4 Pass). Gửi link PR để bạn bấm Merge!

4. Danh Sách Đội Hình (VINASAT / TOMAHAWK) & Khoanh Vùng Role Bundles

Mỗi Role nạp đúng Instructions + Skills + MCP riêng • Tự do sửa file thoải mái

SQUAD	LỆNH BẬT	[1] PO	[2] SA (TIER-1 ★)	[3] SM	[4] BE-DEV	[5] FE-DEV	[6] QA
VINASAT-1	spawn-squad vinasat-1	PO-Bao Long (claude)	SA-Minh Tri ★ (agy)	SM-Thanh Tung (claude)	BE-Quoc Huy (claude)	FE-Hoang Nam (agy)	QA-Mai Linh (agy)
VINASAT-2	spawn-squad vinasat-2	PO-Minh An (agy)	SA-Kien Long ★ (opus)	SM-Duc Anh (agy)	BE-Thanh Nam (agy)	FE-Tuan Kiet (claude)	QA-Thu Ha (claude)
VINASAT-3	spawn-squad vinasat-3	PO-Hoang An (claude)	SA-Hoang Long ★ (agy)	SM-Quang Huy (claude)	BE-Minh Quan (claude)	FE-Dang Khoa (agy)	QA-Bao Ngoc (agy)
TOMAHAWK-1	spawn-squad tomahawk-1	PO-Gia Huy (agy)	SA-Thien Long ★ (opus)	SM-Hai Dang (agy)	BE-Anh Tuan (agy)	FE-Minh Duc (claude)	QA-Khanh Vy (claude)
TOMAHAWK-2	spawn-squad tomahawk-2	PO-Khanh An (claude)	SA-Viet Hoang ★ (agy)	SM-Trung Kien (claude)	BE-Dinh Phuc (claude)	FE-Nhat Minh (agy)	QA-Thao Nhi (agy)
TOMAHAWK-3	spawn-squad tomahawk-3	PO-Tuan Anh (agy)	SA-Dinh Long ★ (opus)	SM-Hoang Phuc (agy)	BE-Gia Bao (agy)	FE-Quang Minh (claude)	QA-Phuong Anh (claude)

5. Review Thực Tế & 5 Công Cụ Chống Lỗi Đã Tích Hợp

Giải quyết triệt để các lỗi thực tế khi vận hành đa Agent trên Herdr

#	VẤN ĐỀ THỰC TẾ	GIẢI PHÁP ĐÃ TÍCH HỢP (NGẮN GỌN)	LỆNH THỰC THI
1	Quên bật Squad hoặc bật trùng tab	- Tự kiểm tra: nếu đã có 6 panes thì dừng lại ngay. - Nếu chưa có: tự bật mới & dời cả 6 panes sang idle.	spawn-squad --ensure vinasat-1
2	Gõ nhầm pane_id khi nhấn chéo	- Tự lưu danh bạ vào _bmad-output/active-squad.json. - Nhấn tin theo tên Role, không cần nhớ pane_id.	squad-msg <po sa sm be fe qa> "..."
3	BE & FE đụng Git Branch/Index	- Mỗi Story chạy trên 1 Git Worktree riêng biệt. - Không bao giờ ghi đè branch của nhau.	git worktree add .worktrees/<id>
4	Chờ lâu khi hỏi đáp /grill-me	- Quét codebase 1 lần ở đầu phiên. - Điều phối viên hỏi đáp từng câu 1 trực tiếp với bạn (phản hồi tức thì).	/bmad-party-mode + /grill-me
5	Khó theo dõi 18 panes cùng lúc	1 lệnh duy nhất in ra toàn bộ trạng thái 6 panes, sprint-status.yaml và danh sách Worktrees.	squad-status [project_dir]

FLOW THỰC THI 9 BƯỚC CHUẨN HÓA (FULL FLOW + /GRILL-ME + TDD RED → GREEN)

Khoanh vùng Instructions/Skills/MCP riêng theo từng Role (Tự do sửa file) • Mặc định ép Full Flow + /grill-me • Tách bạch Bước 5 (Viết Test) & Bước 6 (Viết Code)

BƯỚC 1 Setup	Bạn (User) Tại Terminal dự án	Khởi tạo BMAD & Bật Đội hình 6 Panes (Cố lập Instructions, Skills & MCP riêng) Tự động tạo <code>_bmad-output/sprint-status.yaml</code> và mở 6 panes (PO, SA, SM, BE-Dev, FE-Dev, QA) nạp <code>~/.agents/role-bundles/<role></code> (tất cả các Role đều được tự do chỉnh sửa file khi cần).	LỆNH BẠN GÕ: <code>cd /path/to/repo</code> <code>squad-init && spawn-squad 1</code>
BƯỚC 2 Intake (Full)	Bạn → [3] SM Pane [3] SM trên Herdr	Nêu đầu việc tại [3] SM → Mặc định ép 100% vào Full Flow Mỗi khi bạn nêu đầu việc, [3] SM không đi tắt mà mặc định kích hoạt Full Flow : lập tức triệu tập bàn tròn <code>/bmad-party-mode</code> kết hợp <code>/grill-me</code> .	GÕ TẠI PANE [3] SM: <code>/squad</code> Triển khai tính năng [Mô tả yêu cầu]...
BƯỚC 3 Party + Grill	PO + SA + Team + Phòng vấn Bạn (/grill-me)	Họp Team /bmad-party-mode + Hồi xoáy /grill-me từng câu 1 & Khóa OpenAPI 5 roles tự đọc codebase trước, tranh biện rồi dùng <code>/grill-me</code> hỏi bạn từng câu 1 (kèm (Recommended)) để làm rõ mọi nghiệp vụ. Khi bạn gõ "OK chốt" → Khóa <code>prd.md</code> & <code>architecture.md</code> (DB + OpenAPI Contract).	HỌP & PHÒNG VẤN TỪNG CÂU 1: <code>/bmad-party-mode + /grill-me</code> # Trả lời từng câu → Gõ: "OK chốt"
BƯỚC 4 Sharding	[1] PO + [3] SM Tách Story BE & FE	Tách file .story.md chuẩn BDD (Given/When/Then) & Cập nhật sprint-status.yaml [1] PO tách thành các file <code>_bmad-output/stories/*.story.md</code> đính kèm OpenAPI Contract & tiêu chí BDD (Given/When/Then) làm đề bài viết test cho Bước 5. [3] SM cập nhật <code>ready-for-dev</code> .	ARTIFACT SINH RA: <code>_bmad-output/stories/1-1-be.story.md</code> <code>_bmad-output/sprint-status.yaml</code>
BƯỚC 5 TDD RED	[4] BE + [5] FE Viết Test Trước Khi Code	Viết Contract Skeleton & Khóa Bộ Test TDD Fail Trước (RED Phase) (5a) Tạo hàm/route stub (501 NotImplemented) & import OpenAPI types để code biên dịch được. (5b) Viết 100% Unit / Integration / Component Test phủ mọi kịch bản Given/When/Then. (5c) Chạy test xác nhận FAIL (RED Assertion) & commit khóa test!	XÁC NHẬN RED & KHÓA TEST: <code>pytest path/to/test.py # → RED</code> <code>git commit -m "test(red): lock tests"</code>
BƯỚC 6 TDD GREEN	[4] BE + [5] FE Code Logic & UI Song Song	Viết Code Nghiệp Vụ & Giao Diện Lành Xanh Toàn Bộ Test (GREEN Phase) [4] BE-Dev và [5] FE-Dev (chạy trên 2 hãng CLI khác nhau) code logic thật & UI để làm toàn bộ test ở Bước 5 chuyển sang PASS 100% (GREEN) + <code>tsc --noEmit</code> (tuyệt đối không làm yếu assertion của test Bước 5) → Mở PR.	XÁC NHẬN GREEN & MỞ PR: <code>pytest / vitest run # → 100% GREEN</code> <code>git commit -m "feat: implement story"</code>
BƯỚC 7 Gate 3.5	[2] SA (Reviewer) Cổng Kiểm Định Kiến Trúc	Gate 3.5 — SA Duyệt Kiến Trúc, OpenAPI Contract, 12 Code Smells & TDD Integrity [3] SM chuyển PR sang [2] SA. SA kiểm tra: (1) Đúng Blueprint & OpenAPI Contract BE → FE, (2) Bộ test Bước 5 đủ kịch bản & không bị sửa yếu ở Bước 6, (3) Deep Modules, 12 Code Smells, Security & N+1 Query.	SM CHUYỂN LỆNH TỚI [2] SA: <code>herdr agent prompt <sa_pane></code> "[SM → SA Gate 3.5] Review story..."
BƯỚC 8 Gate 4	[6] QA (Red-Team) Cổng Kiểm Thử Đối Kháng	Gate 4 — QA Kiểm Thử API Thật & Trình Duyệt E2E Theo BDD Criteria Sau khi SA duyệt Gate 3.5, [3] SM bàn giao cho [6] QA. QA chạy lại bộ test TDD Bước 5/6, bắn API thật và mở Playwright/DevTools kiểm tra từng dòng Given / When / Then. Xuất biên bản PASS / REJECT gửi SM.	SM CHUYỂN LỆNH TỚI [6] QA: <code>herdr agent prompt <qa_pane></code> "[SM → QA Gate 4] Verify E2E..."
BƯỚC 9 Closure	[3] SM → Bạn Hoàn tất & Merge PR	Cập nhật sprint-status.yaml → done & Báo cáo Nghiệm thu Khi cả Gate 3.5 (SA) và Gate 4 (QA) đều Pass, [3] SM cập nhật trạng thái Story thành done trong <code>_bmad-output/sprint-status.yaml</code> và báo cáo kết quả kèm link PR để bạn Merge!	BẠN KIỂM TRA TIẾN ĐỘ: <code>bmad-loop status</code> <code>cat _bmad-output/sprint-status.yaml</code>

6.3. Hướng Dẫn Thực Chiến Use Case Cụ Thể

HƯỚNG DẪN THỰC CHIẾN 1 TÍNH NĂNG CỤ THỂ TRÊN HERDR (USE CASE WALKTHROUGH)

Bạn chỉ cần ngồi tại 1 Ô Duy Nhất ([3] SM-Thanh Tung trong tab VINASAT-1) • Gõ 3 câu chat • Squad tự chạy hết 9 Bước!

🔴 Use Case: "Tạm khóa CTV vi phạm + Lưu lịch sử Audit"

THAO TÁC 1 • Mở Tab VINASAT-1 & Giao Việc Cho [3] SM-Thanh Tung

👤 Bạn thao tác (30 giây)

👤 BẠN GÕ TẠI Ô [3] SM-THANH TUNG (GÓC TRÊN BÊN PHẢI):

`/squad` Làm tính năng Tạm khóa tài khoản CTV vi phạm trên trang Quản lý CTV: Admin bấm Tạm khóa → nhập lý do → khóa đăng nhập App/Web của CTV đó và Lưu lịch sử người khóa.

- Điều gì xảy ra ngay lập tức? [3] SM-Thanh Tung tự động kích hoạt Full Flow (`/bmad-party-mode + /grill-me`).
- PO-Bao Long, SA-Minh Trí, BE-Quoc Huy, FE-Hoang Nam tự dùng codebase-memory-mcp quét bằng Collaborator, API Auth và màn hình danh sách CTV hiện có trước khi hỏi bạn.

THAO TÁC 2 • Trả Lời Phòng Vấn /grill-me (Từng Câu 1) & Chốt

🎯 Họp Party + /grill-me (2 phút)

🏢 PO-BAO LONG & 🏢 SA-MINH TRI HỎI BẠN (CÂU 1/2):

Khi CTV bị tạm khóa mà đang có phiên đăng nhập (JWT/Session) trên App Mobile hoặc Web, bạn muốn xử lý thế nào?

- (Recommended) Option A: Thu hồi token ngay lập tức + chặn ở Middleware (trả về 403 ACCOUNT_SUSPENDED kèm lý do khóa).
- Option B: Đợi hết hạn token hiện tại mới chặn đăng nhập mới.

👤 BẠN TRẢ LỜI NGẮN GỌN:

Chọn A nhé. Và khi mở khóa lại thì cũng bắt nhập lý do mở khóa. OK chốt!

TỰ ĐỘNG CHẠY (BƯỚC 4 → 5 → 6) • Tách Story, Viết Test RED → Code GREEN

👥 Squad tự chạy 100%

🌟 [4] BE-QUOC HUY (CLAUDE) & 🧑 [5] FE-HOANG NAM (AGY):

- [Bước 4] PO-Bao Long xuất `prd.md`, SA-Minh Trí xuất `architecture.md` (khóa OpenAPI POST `/api/v1/collaborators/{id}/suspend`), tách 2 file `.story.md`.
- [Bước 5 - TDD RED] BE & FE tạo hàm stub rỗng + viết 100% test `pytest / vitest` theo Given/When/Then → Chạy xác nhận **FAIL (RED)** → Commit khóa bộ test!
- [Bước 6 - TDD GREEN] BE & FE code Logic thật & Modal UI song song lành xanh 100% bộ test Bước 5 + `tsc --noEmit` → Mở PR báo về SM-Thanh Tung.

TỰ ĐỘNG DUYỆT (BƯỚC 7 → 8 → 9) • SA Gate 3.5 → QA Gate 4 → Báo Bạn Merge

🔒 Cổng Kiểm Định Kép

🏢 [2] SA-MINH TRI (GATE 3.5) → 🧑 [6] QA-MAI LINH (GATE 4):

- [Bước 7 - SA-Minh Trí]: Audit PR khớp 100% OpenAPI Contract, kiểm tra bộ test Bước 5 không bị sửa yếu, không dính N+1 query → **APPROVED**.
- [Bước 8 - QA-Mai Linh]: Bắn API thật (200/400/403) & mở Playwright kiểm tra Modal Tạm khóa trên trình duyệt thật → **PASS**.
- [Bước 9 - SM-Thanh Tung]: Cập nhật `sprint-status.yaml` → done và báo bạn link PR để Merge!

💡 Tóm lại những gì BẠN cần làm cho 1 tính năng: Mở Herdr → Vào tab VINASAT-1 → Click ô [3] SM-Thanh Tung → Gõ `/squad <mô tả tính năng>` → Trả lời câu hỏi `/grill-me` Theo dõi tiến độ: `cat _bmad-output/sprint-status.yaml` → Gõ "OK chốt" → Đợi nhận PR sạch đẹp!