

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

TÊN MÔN HỌC: KIỂM THỬ PHẦN MỀM

ĐỀ TÀI: KIỂM THỬ ỨNG DỤNG DỰ BÁO THỜI THIẾT

Giảng viên hướng dẫn:

ThS. Trần Thị Huệ

STT	Họ và tên	Mã sinh viên
1	Lâm Ngọc Tú	1671020341
2	Trịnh Thị Yến Mai	1671020196
3	Đoàn Tuấn Nam	1671020209

Hà Nội, 2025

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

TÊN MÔN HỌC: KIỂM THỬ PHẦN MỀM

ĐỀ TÀI: TÊN ĐỀ TÀI

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	1671020341	Lâm Ngọc Tú	23/04/2004		
2	1671020196	Trịnh Thị Yến Mai	26/06/2004		
3	1671020209	Đoàn Tuấn Nam	06/10/2004		

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI 2

Trần Thị Huệ

Hà Nội, 2025

LỜI NÓI ĐẦU

<Nội dung lời nói đầu>

MỤC LỤC

ĐỀ	1
CHƯƠNG 1.	1
1.1.	1
<i>1.1.1</i>	1
<i>1.1.2</i>	2
<i>1.1.3</i>	4
1.2.	6
<i>1.2.1</i>	6
<i>1.2.2</i>	6
1.3.	6
CHƯƠNG 2.	7
2.1.	7
<i>2.1.1</i>	7
<i>2.1.2</i>	7
2.2.	8
<i>2.2.1</i>	8
<i>2.2.2</i>	8
<i>2.2.3</i>	8
<i>2.2.4</i>	9
<i>2.2.5</i>	9
2.3.	10

2.3.1	10
2.3.2	16
2.4.	16
2.5.	17
2.6.	18
2.7.	18
CHƯƠNG 3.	19
3.1.	19
3.1.1	19
3.1.2	19
3.1.3	21
3.2.	21
3.2.1	21
3.2.2	21
3.2.3	21
CHƯƠNG 4.	22
4.1.	22
4.2.	22
CHƯƠNG 5.	23
5.1.	23
5.2.	25

CHƯƠNG 1. LÝ THUYẾT VỀ KIỂM THỬ PHẦN MỀM

1.1. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

1.1.1 Khái niệm kiểm thử phần mềm

Kiểm thử phần mềm (Software Testing) là quá trình có hệ thống nhằm đánh giá và xác minh một ứng dụng hoặc hệ thống phần mềm để đảm bảo phần mềm đó hoạt động đúng theo yêu cầu đã được định nghĩa, không có lỗi, và đáp ứng các tiêu chuẩn chất lượng. Kiểm thử phần mềm không chỉ giúp phát hiện lỗi mà còn đảm bảo rằng phần mềm đáng tin cậy, ổn định và dễ sử dụng trước khi đưa vào môi trường thực tế.

Mục tiêu chính của kiểm thử phần mềm bao gồm:

Phát hiện lỗi và vấn đề tiềm ẩn:

- Kiểm thử giúp tìm ra các lỗi trong quá trình thiết kế, lập trình hoặc trong các yêu cầu phần mềm.
- Các lỗi này có thể bao gồm lỗi logic, lỗi tính toán, lỗi giao diện người dùng, hoặc lỗi bảo mật.

Đảm bảo chất lượng phần mềm:

- Kiểm thử giúp xác minh phần mềm đáp ứng đầy đủ chức năng, hiệu suất, khả năng tương thích với các thiết bị và trình duyệt, bảo mật và khả năng sử dụng.
- Chất lượng phần mềm được đánh giá không chỉ dựa trên việc phần mềm hoạt động đúng, mà còn dựa trên trải nghiệm người dùng, độ tin cậy và khả năng mở rộng.

Ngăn ngừa sự cố và rủi ro:

- Khi phần mềm được triển khai mà chưa được kiểm thử kỹ, nguy cơ gặp sự cố hoặc gây tổn thất cho người sử dụng là rất cao.

- Kiểm thử giúp giảm thiểu các rủi ro này, đảm bảo phần mềm hoạt động ổn định trong môi trường thực tế.

Xác minh và xác nhận phần mềm:

- **Xác minh (Verification):** Kiểm tra phần mềm được xây dựng đúng theo thiết kế và yêu cầu kỹ thuật.
- **Xác nhận (Validation):** Đảm bảo phần mềm thực hiện đúng các chức năng mà người dùng mong đợi, đáp ứng nhu cầu thực tế.

Các loại kiểm thử phần mềm phổ biến:

- **Kiểm thử chức năng (Functional Testing):** Kiểm tra các chức năng chính của phần mềm.
- **Kiểm thử phi chức năng (Non-Functional Testing):** Kiểm tra hiệu suất, bảo mật, khả năng chịu tải, tính khả dụng.
- **Kiểm thử tự động (Automated Testing):** Sử dụng các công cụ để kiểm thử nhanh, lặp lại nhiều lần.
- **Kiểm thử thủ công (Manual Testing):** Người kiểm thử thực hiện các bước kiểm tra theo kịch bản.

Tóm lại, **kiểm thử phần mềm là quá trình quan trọng để phát hiện lỗi, cải thiện chất lượng và đảm bảo phần mềm hoạt động ổn định, đáng tin cậy trước khi triển khai thực tế.** Nó đóng vai trò then chốt trong việc nâng cao trải nghiệm người dùng và giảm thiểu rủi ro cho doanh nghiệp hoặc tổ chức sử dụng phần mềm.

1.1.2 Phương pháp kiểm thử hộp đen

Kiểm thử hộp đen (Black-box Testing) là phương pháp kiểm thử phần mềm tập trung vào **kiểm tra các chức năng của phần mềm mà không cần quan tâm đến cấu trúc bên trong, mã nguồn hay cách thức hoạt động nội bộ** của hệ thống. Mục tiêu là xác minh rằng phần mềm hoạt động đúng theo yêu cầu chức năng đã được định nghĩa trong tài liệu yêu cầu.

Các đặc điểm chính của kiểm thử hộp đen:

Không cần hiểu cấu trúc bên trong:

- Người kiểm thử không cần biết thuật toán, mã nguồn hay kiến trúc của phần mềm.
- Kiểm thử dựa trên **yêu cầu chức năng**, đầu vào và kết quả đầu ra mong đợi.

Tập trung vào chức năng và kết quả:

- Kiểm thử hộp đen xác minh rằng hệ thống thực hiện đúng chức năng.
- Các đầu vào hợp lệ hoặc không hợp lệ được thử nghiệm để xem phần mềm phản hồi như thế nào.

Phát hiện loại lỗi:

- Lỗi chức năng: phần mềm không thực hiện đúng chức năng yêu cầu.
- Lỗi giao diện: hiển thị sai hoặc không thân thiện với người dùng.
- Lỗi dữ liệu: phần mềm xử lý dữ liệu sai hoặc không phù hợp với các tình huống thực tế.
- Lỗi bảo mật và khả năng tương thích.

Các kỹ thuật kiểm thử hộp đen phổ biến:

Kiểm thử theo giá trị biên (Boundary Value Testing):

- Kiểm tra các giá trị ở ranh giới của tập hợp dữ liệu đầu vào.
- Ví dụ: nếu phần mềm chấp nhận tuổi từ 18–60, ta sẽ kiểm thử các giá trị 17, 18, 60, 61.

Kiểm thử bảng quyết định (Decision Table Testing):

- Sử dụng bảng quyết định để kiểm tra tất cả các kết hợp đầu vào và đầu ra có thể xảy ra.

Kiểm thử phân lớp tương đương (Equivalence Partitioning):

- Chia dữ liệu đầu vào thành các lớp tương đương và kiểm thử đại diện cho mỗi lớp.
- Ví dụ: nhập số âm, số dương, và số 0 nếu ứng dụng nhận số.

Kiểm thử trường hợp sử dụng (Use Case Testing):

- Kiểm tra các chức năng theo từng kịch bản mà người dùng thực hiện.

Ưu điểm của kiểm thử hộp đen:

- Không cần kiến thức về mã nguồn, phù hợp với người kiểm thử không phải lập trình viên.
- Tập trung vào trải nghiệm người dùng và chức năng thực tế.
- Phát hiện lỗi chức năng, dữ liệu và giao diện hiệu quả.

Nhược điểm:

- Không thể phát hiện lỗi cấu trúc bên trong phần mềm.
- Không đảm bảo toàn bộ các luồng xử lý đều được kiểm thử.

Tóm lại, **kiểm thử hộp đen là phương pháp kiểm thử phần mềm dựa trên yêu cầu chức năng, tập trung vào đầu vào và đầu ra để đảm bảo phần mềm hoạt động đúng theo mong đợi mà không cần quan tâm đến bên trong của hệ thống.**

1.1.3 Phương pháp kiểm thử hộp trắng

Kiểm thử hộp trắng (White-box Testing) là phương pháp kiểm thử phần mềm tập trung vào cấu trúc bên trong, logic, mã nguồn và các luồng xử lý của hệ thống. Khác với kiểm thử hộp đen, kiểm thử hộp trắng yêu cầu người kiểm thử hiểu rõ về chương trình, cấu trúc dữ liệu, thuật toán và kiến trúc phần mềm để thiết kế các kịch bản kiểm thử phù hợp.

Các đặc điểm chính của kiểm thử hộp trắng:

Dựa trên mã nguồn và luồng xử lý:

- Người kiểm thử phân tích các câu lệnh, nhánh điều kiện, vòng lặp, và các đường dẫn trong mã nguồn.
- Mục tiêu là đảm bảo rằng tất cả các luồng logic đều được thực hiện ít nhất một lần.

Phát hiện lỗi cấu trúc và logic:

- Lỗi tính toán, lỗi điều kiện, lỗi vòng lặp, lỗi về xử lý dữ liệu hay lỗi bảo mật.
- Giúp tối ưu hóa mã nguồn và đảm bảo các điều kiện rẽ nhánh đều được kiểm tra.

Các kỹ thuật kiểm thử hộp trắng phổ biến:

Kiểm thử đường dẫn (Path Testing):

- Kiểm thử tất cả các đường dẫn có thể đi qua trong chương trình.
- Ví dụ: với một hàm có nhiều nhánh điều kiện, mỗi nhánh phải được kiểm thử.

Kiểm thử nhánh (Branch Testing):

- Kiểm tra tất cả các nhánh điều kiện để đảm bảo mỗi nhánh đều được thực thi ít nhất một lần.

Kiểm thử câu lệnh (Statement Testing):

- Kiểm tra từng câu lệnh trong mã nguồn để đảm bảo không có câu lệnh nào bị bỏ sót.

Kiểm thử vòng lặp (Loop Testing):

- Kiểm tra các vòng lặp với các trường hợp: không lặp, lặp một lần, lặp nhiều lần và lặp tối đa.

Ưu điểm của kiểm thử hộp trắng:

- Phát hiện lỗi cấu trúc, logic và bảo mật hiệu quả.
- Đảm bảo rằng tất cả các đường dẫn và nhánh điều kiện đều được kiểm thử.
- Giúp tối ưu hóa mã nguồn và cải thiện chất lượng phần mềm.

Nhược điểm:

- Yêu cầu người kiểm thử có kiến thức lập trình và hiểu rõ mã nguồn.
- Có thể tốn thời gian khi kiểm thử các chương trình lớn với nhiều đường dẫn.
- Không tập trung vào trải nghiệm người dùng, mà chỉ kiểm tra logic nội bộ.

Tóm lại, **kiểm thử hộp trắng** là phương pháp kiểm thử dựa trên kiến thức về mã nguồn và cấu trúc bên trong của phần mềm, giúp phát hiện lỗi logic, tối ưu hóa chương trình và đảm bảo rằng tất cả các luồng xử lý đều hoạt động chính xác.

1.2. QUY TRÌNH KIỂM THỬ PHẦN MỀM

1.2.1 Quy trình kiểm thử phần mềm

<sơ đồ quy trình>

1.2.2 Mức độ kiểm thử phần mềm

<Sơ đồ V, W>

1.3. CÔNG CỤ KIỂM THỬ PHẦN MỀM

<Tìm hiểu về các nhóm công cụ hỗ trợ kiểm thử>

CHƯƠNG 2. KẾ HOẠCH KIỂM THỬ PHẦN MỀM (DỰA THEO FILE MẪU – KẾ HOẠCH KIỂM THỬ)

2.1. GIỚI THIỆU

2.1.1 Tổng quan về phần mềm (ứng dụng) kiểm thử

Ứng dụng **WeatherApp** (nguồn mở từ GitHub: [weatherapp2](#)) là ứng dụng di động dự báo thời tiết phát triển bằng **Flutter**. Ứng dụng cho phép người dùng nhập tên thành phố và nhận dữ liệu thời tiết từ API **OpenWeatherMap**, hiển thị thông tin chi tiết như:

- Nhiệt độ hiện tại.
- Độ ẩm.
- Tốc độ gió.
- Trạng thái mây.
- Icon và hình nền thay đổi theo điều kiện thời tiết.
- Dự báo ngắn hạn theo giờ/ngày.

Mục tiêu kiểm thử:

- Đảm bảo ứng dụng hiển thị dữ liệu chính xác và kịp thời.
- Đảm bảo UI thân thiện, không lỗi font, không crash trên nhiều môi trường.
- Đảm bảo các trường hợp nhập liệu sai (tên thành phố không tồn tại) được xử lý hợp lý.

2.1.2 Các chức năng của dự án (Liệt kê hoặc sơ đồ use case tổng thể)

STT	Chức năng	Mô tả
1	Tìm kiếm thành phố	Người dùng nhập tên thành phố → Ứng dụng gửi request API và trả về thông tin.

2	Hiển thị thời tiết hiện tại	Nhiệt độ, độ ẩm, tốc độ gió, trạng thái mây.
3	Hiển thị icon và hình nền	Icon & background thay đổi theo trạng thái (nắng, mưa, nhiều mây, tuyết...).
4	Dự báo thời tiết	Hiển thị dự báo theo giờ hoặc theo ngày.
5	Xử lý lỗi nhập liệu	Hiển thị thông báo nếu người dùng nhập sai hoặc API trả về lỗi.

2.2. YÊU CẦU VỀ TÀI NGUYÊN

2.2.1 Phần cứng

Máy tính cá nhân có kết nối mạng Internet.

CPU	RAM	HDD	Architecture
Intel Core i5, 2.7 GHz	8 GB	512 GB	64 bit

2.2.2 Phần mềm

Tên phần mềm	Phiên bản	Loại
Trình duyệt Chrome	Mới nhất	Trình duyệt web (với ứng dụng web)
Visual Studio Code	2023	Soạn thảo, debug code Flutter
Windows 10 / Ubuntu 22.04	Mới nhất	Chạy môi trường phát triển
Flutter SDK	3.x	Framework
Android Studio	2024	IDE
Xcode	15.x	IDE
Postman	10.x	Công cụ test API

2.2.3 Công cụ kiểm thử

Hoạt động	Công cụ	Nhà cung cấp	Phiên bản
Quản lý Test Case	Google Sheets	Google	Mới nhất
Quản lý bug	GitHub	GitHub	Mới nhất
Kiểm thử UI	Flutter Driver / Appium	Google / Appium team	2024
Kiểm thử API	Postman	Postman Inc.	10.x

2.2.4 Môi trường kiểm thử

- Máy tính cá nhân có kết nối Internet ổn định.
- Thiết bị kiểm thử: Emulator Android, iOS Simulator, điện thoại.
- API: OpenWeatherMap.
- Hệ điều hành: Windows 10, Ubuntu 22.04, macOS Ventura.

2.2.5 Nhân sự

2.2.5.1 Vai trò và trách nhiệm

Thành viên	Vai trò
Lâm Ngọc Tú	Test Manager / Test Designer / Tester: Lập kế hoạch kiểm thử, quản lý tiến độ hoạt động kiểm thử, thiết kế các test case bổ sung và thực thi test case. Phụ trách kiểm thử các chức năng chính gồm: Tìm kiếm thành phố, xử lý lỗi nhập liệu và kiểm tra tính ổn định ứng dụng trên nhiều môi trường.
Trịnh Thị Yến Mai	Test Designer / Tester: Thiết kế và viết test case, thực thi test case cho các chức năng: Hiển thị thông tin thời tiết hiện tại (nhiệt độ, độ ẩm, gió, trạng thái mây) và hiển thị icon & hình nền động theo điều kiện thời tiết. Đồng thời xem xét và rà soát lại Test Plan.

Thành viên	Vai trò
Đoàn Tuấn Nam	Test Designer / Tester: Thiết kế và viết test case, thực thi test case cho các chức năng: Dự báo thời tiết theo ngày/giờ và kiểm thử giao diện (UI) trên các thiết bị khác nhau. Tham gia review kết quả kiểm thử và góp ý cho nhóm.

2.2.5.2 Đào tạo

<N/A>

2.3. PHẠM VI KIỂM THỬ

2.3.1 Những chức năng được kiểm thử

ST T	Chức năng	Nội dung kiểm thử chi tiết
1	Tìm kiếm thành phố	<p>Mục tiêu: đảm bảo chức năng tìm kiếm gửi request chính xác, xử lý input hợp lệ/không hợp lệ, hiển thị trạng thái tải và kết quả.</p> <p>Điều kiện trước: app khởi động; ô tìm kiếm (placeholder) hiển thị; API key hợp lệ; có/không có mạng.</p> <p>Trước khi tìm kiếm: kiểm tra placeholder, tooltip, validation nhanh (không cho phép chỉ khoảng trắng). Kiểm tra trimming (xóa khoảng trắng đầu/cuối), chuẩn hóa dấu (Hà Nội ↔ Ha Noi), không phân biệt chữ hoa/thường.</p> <p>Trong khi nhập: kiểm thử debounce (gõ nhanh nhiều lần có gây flood request không), autocomplete/suggestion (nếu có).</p> <p>Khi nhấn tìm kiếm / submit: - Giao tiếp API: đúng</p>

		<p>endpoint, tham số thành phố + mã quốc gia (nếu có). - Hiển thị spinner/loader; disable nút tìm kiếm; hiển thị chỉ báo "Đang tải..."</p> <p>Các trường hợp có thể xảy ra:</p> <ul style="list-style-type: none"> - Tên thành phố hợp lệ (ví dụ: "Hà Nội", "New York") → trả về 200 + dữ liệu. - Tên không tồn tại → API trả 404 hoặc body rỗng → hiển thị "Không tìm thấy thành phố". - Ô rỗng → không gọi API, hiển thị "Vui lòng nhập tên thành phố". - Ký tự đặc biệt / SQL-injection-like input → app phải sanitize; không crash. - Nhập số / chuỗi dài (>255 ký tự) → validate và cảnh báo. - Gõ nhanh nhiều lần → request trước bị cancel hoặc xử lý đúng thứ tự (last-search-wins). - Mất mạng ngay khi tìm → hiển thị thông báo "Không có kết nối Internet" và nút Thử lại. <p>Kỳ vọng hiển thị sau tìm kiếm: tên thành phố + mã quốc gia, thời gian cập nhật (timestamp), nút Refresh, hiển thị dữ liệu thời tiết hiện tại hoặc thông báo lỗi rõ ràng. Thời gian phản hồi hợp lý (< 3s theo điều kiện chấp nhận).</p>
2	<p>Hiển thị thông tin thời tiết hiện tại</p>	<p>Mục tiêu: dữ liệu (nhiệt độ, độ ẩm, gió, trạng thái mây...) hiển thị chính xác, rõ ràng, dễ đọc; tương thích đơn vị.</p> <p>Fields cần kiểm thử hiển thị: nhiệt độ (temp), feels_like (nếu có), độ ẩm, tốc độ gió (+ hướng), áp</p>

		<p>suất, trạng thái (Clear/Clouds/Rain/Snow/Fog...), thời điểm cập nhật.</p> <p>Các kịch bản cụ thể (ví dụ hiển thị theo điều kiện):</p> <ul style="list-style-type: none"> - Nắng (Clear): temp = 28°C → hiển thị 28°C, bên cạnh icon mặt trời; label Nắng hoặc Clear; background sáng (nền gradient vàng/đen nhẹ); (nếu app có) màu số nhiệt độ là cam/đỏ nhạt khi >30°C. - Mưa (Rain / Shower / Thunderstorm): temp = 22°C → icon mưa + text Mưa nhẹ/Mưa to phụ thuộc intensity; hiển thị khả năng mưa (%) hoặc lượng mm nếu API trả; background tối hơn + animation giọt mưa (nếu có). - Tuyết (Snow): temp ≤ 0°C → icon bông tuyết + Tuyết rơi; background lạnh (xanh dương nhạt). - Gió mạnh (Windy): wind = 40 km/h → hiển thị 40 km/h, icon cờ/gió, label Gió mạnh nếu vượt threshold. - Nhiều mây (Clouds): icon mây, text Nhiều mây hoặc Có mây. - Sương mù (Fog): hiển thị Sương mù, cảnh báo tầm nhìn thấp nếu dữ liệu có. <p>Kiểm tra định dạng & quy tắc hiển thị:</p> <ul style="list-style-type: none"> - Làm tròn số: nhiệt độ hiển thị đến 1 chữ số thập phân hoặc làm tròn nguyên (theo yêu cầu) — đảm bảo quy tắc nhất quán. - Đổi đơn vị: chuyển đổi đúng °C ↔ °F (nếu app hỗ trợ). - Kiểm thử khi API trả field thiếu (ví dụ không có
--	--	---

		<p>wind) → hiển thị -- hoặc Không có dữ liệu, không crash.</p> <p>Kỳ vọng: các trường hiển thị trùng khớp giá trị API (sai số nhỏ do làm tròn), icon & label khớp trạng thái; không có text rỗng; UI responsive trên các kích thước màn hình.</p>
3	<p>Hiển thị icon & background động</p>	<p>Mục tiêu: icon và background phản ánh chính xác điều kiện thời tiết, đảm bảo đồng bộ với dữ liệu API (day/night, intensity).</p> <p>Mapping & kịch bản hiển thị (ví dụ rõ ràng):</p> <ul style="list-style-type: none"> - Clear (day) → icon mặt trời, background sáng/sky gradient; Clear (night) → icon trăng, background tối. - Rain → icon mưa (light/heavy/thunder variants), background mây tối + animation mưa nếu có. - Snow → icon tuyết, background trắng/blue + animation tuyết. - Clouds → icon mây, background mây. - Drizzle → nhẹ hơn Rain (icon nhỏ). - Thunderstorm → icon sấm chớp + cảnh báo (nếu intensity cao). <p>Chi tiết kiểm thử:</p> <ul style="list-style-type: none"> - Kiểm thử day/night bằng timestamp (nếu API trả sunrise/sunset) → background và icon phải đổi theo thời điểm địa phương. - Kiểm thử mức độ cường độ: light rain vs heavy rain → hiển thị khác nhau. - Kiểm thử fallback: nếu asset icon/background không load → hiển thị icon mặc định/placeholder, không

		<p>crash.</p> <ul style="list-style-type: none"> - Kiểm thử performance: load ảnh có lazy-load, không làm app chậm. - Kiểm thử accessibility: alt text cho icon, contrast đủ lớn cho người mù màu. <p>Kỳ vọng: icon + background luôn đồng bộ với trạng thái API; hiển thị hợp lý cho night/day và intensity; có hành vi fallback an toàn.</p>
4	Dự báo thời tiết	<p>Mục tiêu: hiển thị chính xác danh sách dự báo theo giờ và theo ngày, dữ liệu khớp API, cuộn mượt, nhóm/sắp xếp đúng thứ tự thời gian.</p> <p>Điều kiện: API trả endpoint forecast (hourly/daily) hoặc 1 call trả multi-day; timezone xử lý đúng (hiển thị giờ địa phương).</p> <p>Các kiểm thử chi tiết:</p> <ul style="list-style-type: none"> - Hourly (24h): hiển thị timestamp, temp, icon, mưa (%) cho mỗi giờ; kiểm tra đủ 24 mục hoặc số item theo API. - Daily (7 ngày): hiển thị ngày, temp min/max, icon, xác suất mưa, gió. - Kiểm thử cuộn: list cuộn mượt, không lag; lazy load nếu nhiều item. - Kiểm thử khi đổi thành phố → forecast cập nhật ngay. - Kiểm thử daylight saving/timezone: khi chuyển vùng, giờ hiển thị đúng. - Kiểm thử khi API trả thiếu một số giờ/ngày → UI vẫn hoạt động, hiển thị thông báo Dữ liệu không đầy đủ.

		<p>Kịch bản hiển thị ví dụ:</p> <p>- Ngày 2025-09-28: Thứ 7 — Temp min/max: 22°/31°C — Icon Clouds — Rain chance 20%.</p> <p>Kỳ vọng: số mục, thứ tự thời gian, dữ liệu temp min/max khớp API, không bị lệch timezone.</p>
5	Xử lý lỗi nhập liệu sai	<p>Mục tiêu: app không crash, cung cấp thông báo rõ ràng và phương án phục hồi khi có lỗi từ người dùng hoặc từ API/mạng.</p> <p>Các loại lỗi & cách kiểm thử:</p> <ul style="list-style-type: none"> - Không có mạng: khi gọi API → xử lý timeout; hiển thị Không có kết nối Internet + nút Thử lại; nếu có cache, hiển thị dữ liệu gần nhất với label Dữ liệu lưu cache. - API trả 404 (city not found): hiển thị Không tìm thấy thành phố + gợi ý kiểm tra chính tả. - API trả 429 (rate limit): hiển thị thông báo Quá nhiều yêu cầu, vui lòng chờ hoặc áp dụng backoff; không crash. - API trả 500 / lỗi server: hiển thị Lỗi máy chủ, thử lại sau. - Dữ liệu thiếu / format lạ: (ví dụ thiếu trường wind) → hiển thị Không có dữ liệu ở chỗ tương ứng, log lỗi trên console. - Xử lý input ác ý: nhập chuỗi dài, ký tự đặc biệt → sanitize để tránh injection; hiển thị validation message.

		Kỳ vọng hành vi sau lỗi: không crash app; thông báo người dùng rõ ràng (tiếng Việt); có nút Retry; nếu phù hợp, dùng bản cached; ghi log lỗi cho dev; giữ UI ở trạng thái ổn định.
--	--	---

2.3.2 Những chức năng không kiểm thử

- Kiểm thử hiệu năng với số lượng request lớn.
- Kiểm thử bảo mật API (do phụ thuộc bên thứ ba).

2.4. KỸ THUẬT KIỂM THỬ

Module	Kỹ thuật kiểm thử	Ghi chú
Tìm kiếm thành phố	Kiểm thử hộp đen (Black-box testing), kiểm thử giá trị biên (Boundary value testing), kiểm thử ngoại lệ (Exception testing)	<ul style="list-style-type: none"> - Nhập dữ liệu hợp lệ/không hợp lệ để xác nhận kết quả hiển thị. - Kiểm thử giá trị biên như chuỗi rỗng, chuỗi dài, ký tự đặc biệt. - Kiểm thử ngoại lệ khi API không phản hồi hoặc mất mạng.
Hiển thị thời tiết	Kiểm thử hộp đen, kiểm thử hồi quy (Regression testing)	<ul style="list-style-type: none"> - Đối chiếu dữ liệu hiển thị (nhiệt độ, độ ẩm, gió, mây) với dữ liệu thực tế từ API. - Kiểm thử hồi quy khi cập nhật phiên bản mới để đảm bảo dữ liệu vẫn hiển thị đúng.

Hiển thị icon và background động	Kiểm thử giao diện (UI Testing), kiểm thử hộp đen	<ul style="list-style-type: none"> - Kiểm tra icon và hình nền thay đổi theo trạng thái thời tiết (nắng, mưa, tuyết...). - Đảm bảo sự đồng bộ giữa dữ liệu API và UI. - Kiểm thử trên nhiều kích thước màn hình và hệ điều hành khác nhau.
Dự báo thời tiết (theo giờ/ngày)	Dự báo thời tiết (theo giờ/ngày) Kiểm thử hộp đen, kiểm thử chức năng (Functional testing), kiểm thử tính khả dụng (Usability testing)	<ul style="list-style-type: none"> - Kiểm tra dữ liệu dự báo theo giờ/ngày hiển thị đúng thứ tự thời gian và khớp API. - Đảm bảo người dùng dễ dàng cuộn danh sách dự báo mà không gặp lỗi giao diện.
Xử lý lỗi nhập liệu & sự cố	Kiểm thử ngoại lệ (Exception testing), kiểm thử khả năng chịu lỗi (Error handling testing)	<ul style="list-style-type: none"> - Mô phỏng các tình huống lỗi: nhập sai, không có mạng, API trả về lỗi. - Đảm bảo ứng dụng đưa ra thông báo phù hợp, không bị crash, có thể thử lại.

2.5. ĐIỀU KIỆN CHẤP NHẬN

- 100% test case chức năng chính phải Pass.
- Dữ liệu thời tiết hiển thị đúng theo API OpenWeatherMap.
- Ứng dụng không crash khi nhập sai.
- Thời gian phản hồi < 3 giây.

2.6. DỰ KIẾN LỖI VÀ PHÂN LOẠI LỖI

Mức độ nghiêm trọng	Đặc tả lỗi
Cao (ảnh hưởng trực tiếp đến chức năng chính)	<ul style="list-style-type: none">- Không trả về dữ liệu khi nhập thành phố hợp lệ.- Ứng dụng bị crash khi thực hiện tìm kiếm.- API trả về dữ liệu nhưng ứng dụng không hiển thị.- Dự báo thời tiết hiển thị sai ngày/giờ do lệch timezone hoặc sai dữ liệu.
Trung bình (ảnh hưởng đến trải nghiệm nhưng vẫn sử dụng được)	<ul style="list-style-type: none">- Icon hoặc background hiển thị sai so với trạng thái thời tiết thực tế.- Bố cục bị vỡ hoặc chồng chéo trên màn hình nhỏ.- Dự báo thời tiết không đồng bộ với dữ liệu API.- Thời gian phản hồi lâu hơn mức cho phép (> 5 giây).
Thấp (lỗi thẩm mỹ, không ảnh hưởng nhiều đến chức năng)	<ul style="list-style-type: none">- Sai chính tả hoặc ngôn ngữ giao diện chưa được Việt hóa đầy đủ.- Màu nền không đồng bộ với icon thời tiết.- Font chữ không đồng nhất giữa các thiết bị.- Không có alt text cho icon, giảm khả năng tiếp cận cho người dùng đặc biệt.

2.7. ĐẦU RA DỰ KIẾN

- Kế hoạch kiểm thử
- Test case
- Báo cáo kiểm thử

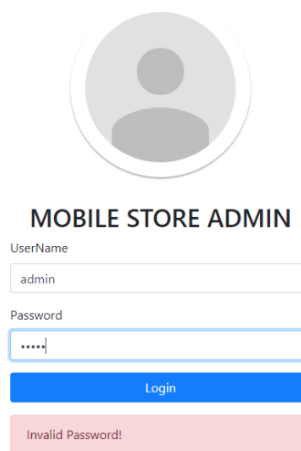
CHƯƠNG 3. TEST CASE VÀ THỰC HIỆN TEST

3.1. TEST CASE CỦA MODULE ĐĂNG NHẬP- ĐĂNG XUẤT

(SV thực hiện: Lâm Thị Huê)

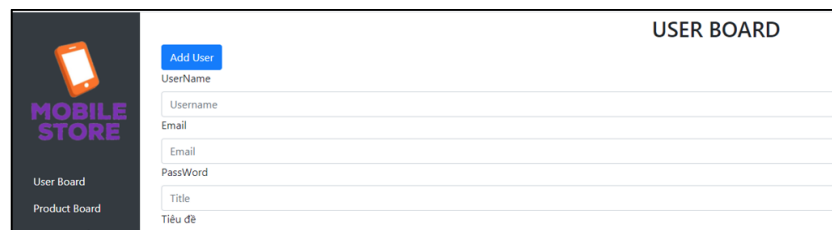
3.1.1 Giao diện chức năng đăng nhập – đăng xuất

Giao diện đăng nhập



A login form for the MOBILE STORE ADMIN. It features a circular placeholder for a profile picture at the top. Below it, the title "MOBILE STORE ADMIN" is centered. The form includes two input fields: "UserName" with the value "admin" and "Password" with masked characters "****". A blue "Login" button is positioned below the password field. A red error message "Invalid Password!" is displayed at the bottom of the form.

Giao diện đăng xuất



A user management interface titled "MOBILE STORE" and "USER BOARD". On the left, there is a sidebar with a mobile phone icon and links for "User Board" and "Product Board". The main area contains an "Add User" button and a form with fields for "UserName", "Email", "PassWord", and "Title". The "Email" field is currently filled with "Email".

Xác định tính đúng đắn và hoạt động của chức năng đăng nhập, đăng xuất trong phần mềm

- Kiểm thử đăng nhập thành công
- Kiểm thử các trường hợp ngoại lệ (đăng nhập không thành công)
- Kiểm thử đăng xuất, kết quả sau đăng xuất.

3.1.2 Kết quả test case

Tên Testcase	Đăng nhập và đăng xuất			Tổng số TC thực hiện	17
Mã Testcase	TC_F01			Tổng số TC Pass	10
Người tạo TC	Lâm Thị Huê			Tổng số TC Fail	7
Ngày tạo TC				Tổng số TC Not-Tested	0
Điều kiện ràng buộc: Người dùng đã được tạo một account để đăng nhập vào hệ thống admin1/ Admin1@3				Người Test	Lâm Thị Huê
				Ngày Test:	
STT	Mục tiêu kiểm thử	Mô tả thao tác kiểm thử	Kết quả mong đợi	Dữ liệu kiểm thử	Kiểm thử lần 1
				Kết quả (P/F/N)	Bug #
					Diễn giải lỗi
Chức năng Đăng nhập					
Kiểm tra giao diện					
TC_F01.1	Kiểm tra hiển thị form đăng nhập	Tại màn hình form Đăng nhập, kiểm tra form. Từ trên xuống dưới từ trái sang phải hiển thị: - Label "UserName" - Textbox tên đăng nhập - Lable "PassWord" - Textbox Mật khẩu - Button "Login"	Form đăng nhập hiển thị đúng như mô tả	Passed	https://pmt.sc/26uvc26u4voiq
Kiểm tra chức năng					
TC_F01.2	Kiểm tra bỏ trống trường username	Bỏ trống trường tên đăng nhập	- Đăng nhập không thành công - Thông báo "This fiel is require"	Passed	https://pmt.sc/26u4voif https://pmt.sc/26u4voiyw
TC_F01.3	Kiểm tra bỏ trống trường password	Bỏ trống trường tên password	-Đăng nhập không thành công - Thông báo "This fiel is require"	Passed	https://pmt.sc/26u4vp6n
TC_F01.4	Kiểm tra độ dài tối đa của trường username	Nhập 10 ký tự vào trường tên đăng nhập	- Nhập được tối đa 10 ký tự	Data test trường username: admin67890	Passed https://pmt.sc/26u4vq0wv
TC_F01.5	Kiểm tra nhập quá độ dài cho phép của username	Nhập 11 ký tự vào trường tên đăng nhập	- Chỉ nhập được tối đa 10 ký tự - Hệ thống tự động cắt bỏ ký tự thừa	Data test trường username: admin678901	Failed https://cucnh1349.atl.assia.n.net/browse https://pmt.sc/26u

Tên Testcase	Đăng nhập và đăng xuất			Tổng số TC thực hiện	17	
Mã Testcase	TC_F01			Tổng số TC Pass	10	
Người tạo TC	Lâm Thị Huê			Tổng số TC Fail	7	
Ngày tạo TC				Tổng số TC Not-Tested	0	
Điều kiện ràng buộc: Người dùng đã được tạo một account để đăng nhập vào hệ thống admin1/ Admin1@3				Người Test	Lâm Thị Huê	
				Ngày Test:		
STT	Mục tiêu kiểm thử	Mô tả thao tác kiểm thử	Kết quả mong đợi	Dữ liệu kiểm thử	Kiểm thử lần 1	
TC_F01.6	Kiểm tra độ dài tối đa của trường password	Nhập 16 ký tự vào trường mật khẩu	Nhập được tối đa 15 ký tự	Data test trường password: 1234567896321455	Failed https://cucnh1349.atlassian.net/browse/PC-2	Nhập được 16 ký tự
TC_F01.7	Kiểm tra độ dài tối thiểu của trường password	Nhập 5 ký tự vào trường mật khẩu	Thông báo " Input at least 6 digits"	Data test trường password: 12345	Failed https://cucnh1349.atlassian.net/browse/PC-3	
TC_F01.8		Nhập 6 ký tự vào trường mật khẩu	Không hiển thị thông báo	Data test trường password: 12abdw	Passed	
TC_F01.9	Kiểm tra mật khẩu không có chữ in hoa	Nhập mật khẩu không có chữ in hoa	Thông báo" Input at least a capital digit"	Data test trường password: hihhih	Failed https://cucnh1349.atlassian.net/browse/PC-8	
TC_F01.10	Kiểm tra mật khẩu có 3 chữ số nhập liên tiếp nhau	Nhập mật khẩu có 3 chữ số liên tiếp nhau: a, a+1, a+2	Thông báo "Password is not included 3 digits consecutive"	Data test trường password: 123456789	Failed https://cucnh1349.atlassian.net/browse/PC-10	
TC_F01.11	Kiểm tra sai mật khẩu quá 5 lần	Nhập sai mật khẩu đến lần thứ 6	Chặn chức năng đăng nhập và Thông báo "Please try again after 10 minutes."		Failed https://cucnh1349.atlassian.net/browse/PC-11	
Kiểm tra quy trình đăng nhập						
TC_F01.12	Đăng nhập thành công	1. Nhập username hợp lệ 2. Nhập password hợp lệ 3. Bấm "Login"	- Thông báo "Đăng nhập thành công" - Chuyển sang trang chủ admin	Passed	https://pmt.sc/26uvzj5	
TC_F01.13		1. Nhập username không hợp lệ 2. Nhập password hợp lệ 3. Bấm "Login"	Thông báo Đăng nhập không thành công	Passed	https://pmt.sc/26uv0b2	

Tên Testcase	Đăng nhập và đăng xuất				Tổng số TC thực hiện	17
Mã Testcase	TC_F01				Tổng số TC Pass	10
Người tạo TC	Lâm Thị Huế				Tổng số TC Fail	7
Ngày tạo TC					Tổng số TC Not-Tested	0
Điều kiện ràng buộc: Người dùng đã được tạo một account để đăng nhập vào hệ thống admin1/ Admin1@3				Người Test	Lâm Thị Huế	
				Ngày Test:		
STT	Mục tiêu kiểm thử	Mô tả thao tác kiểm thử	Kết quả mong đợi	Dữ liệu kiểm thử	Kiểm thử lần 1	
					Kết quả (Pass/Fail)	Diễn giải lỗi
TC_F01.14	Đăng nhập không thành công	1. Nhập username không hợp lệ 2. Nhập password không hợp lệ 3. Bấm "Login"	Thông báo Đăng nhập không thành công		Passed	https://prnt.sc/26uw0js
Chức năng Đăng xuất						
Kiểm tra giao diện						
TC_F01.15	Kiểm tra vị trí button đăng xuất	Ở thanh công cụ bên trái màn hình board, kiểm tra button đăng xuất	- Button được hiển thị ở dưới cùng của thanh công cụ - Tên button là "Logout"		Passed	https://prnt.sc/26uw0s3
Kiểm tra chức năng và quy trình						
TC_F01.16		Ở thanh công cụ chọn "Logout"	- Đăng xuất thành công - Hiện thị message "Log out successfully"		Passed	https://prnt.sc/26uw1nm
TC_F01.17	Đăng xuất thành công	1. Ở thanh công cụ chọn "Logout" 2. Chọn "Yes"	- Hiện thị message "If logout system, change will be not save" - Đăng xuất thành công		Failed	https://cucnh1349.adassian.net/browse/PC-4 Đăng xuất luôn mà không thông báo

3.1.3 Báo cáo test case theo chức năng (Biểu đồ hoặc bảng tổng kết)

Số lượng test case	17
Số test case Pass	10
Số test case Fail	7
Số test case Untested	0
Số test case N/A	0
Tỉ lệ thành công	58.8%

3.2. TEST CASE CỦA MODULE <TÊN MODULE/CHỨC NĂNG>

3.2.1 Mục đích kiểm thử

3.2.2 Kết quả test case

3.2.3 Báo cáo test case theo chức năng (Biểu đồ hoặc bảng tổng kết)

CHƯƠNG 4. BÁO CÁO KẾT QUẢ KIỂM THỬ

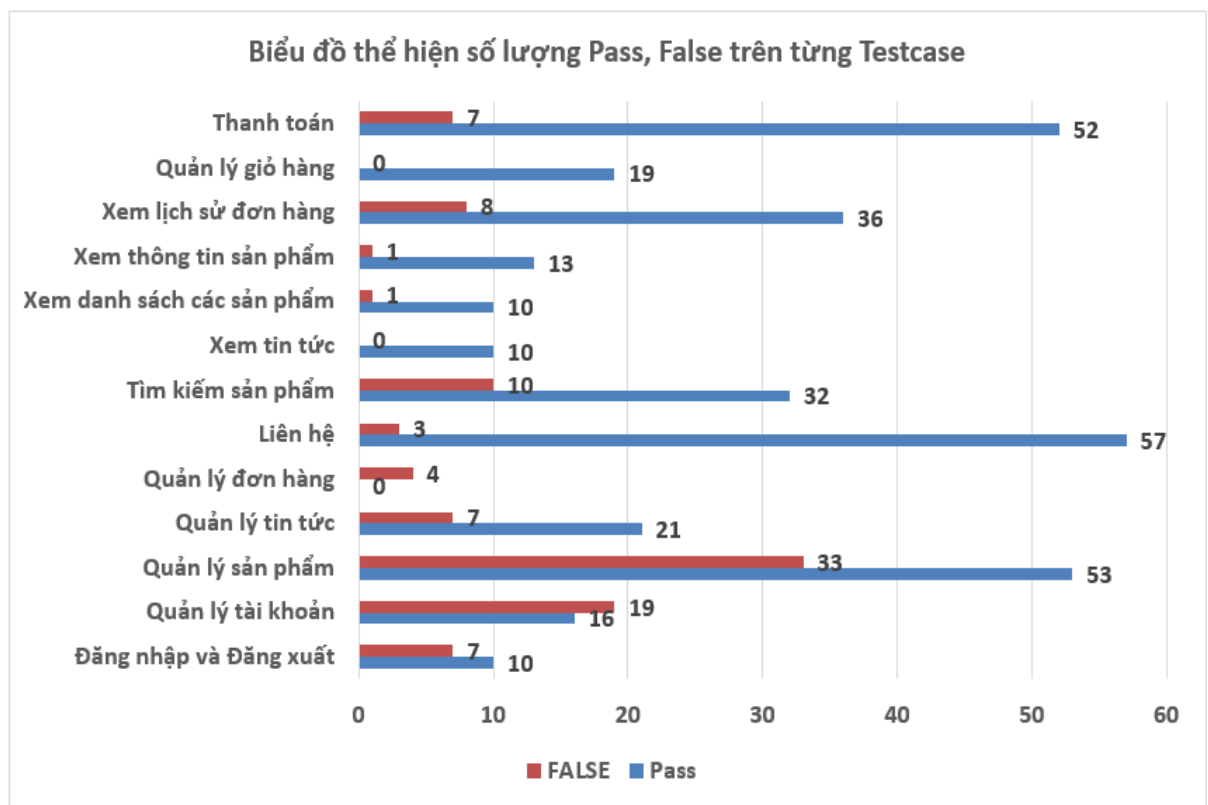
4.1. BÁO CÁO KẾT QUẢ

Báo cáo bằng biểu đồ hoặc bảng báo cáo theo từng chức năng

CÁC TÍNH NĂNG TRONG XÂY DỰNG WEBSITE QUẢN LÝ BÁN HÀNG

Mã Testcase	Tên testcase	Pass	FALSE	Thời gian	Ngày bắt đầu	Ngày kết thúc	Tester
TC_F 1	Đăng nhập và Đăng xuất	10	7				
TC_F 2	Quản lý tài khoản	16	19				
TC_F 3	Quản lý sản phẩm	53	33				
TC_F 4	Quản lý tin tức	21	7				
TC_F 5	Quản lý đơn hàng	0	4				
TC_F 6	Liên hệ	57	3				
TC_F 7	Tìm kiếm sản phẩm	32	10				
TC_F 8	Xem tin tức	10	0				
TC_F 9	Xem danh sách các sản phẩm	10	1				
TC_F 10	Xem thông tin sản phẩm	13	1				
TC_F 11	Xem lịch sử đơn hàng	36	8				
TC_F 12	Quản lý giỏ hàng	19	0				
TC_F 13	Thanh toán	52	7				

Tổng cộng	329	100					
-----------	-----	-----	--	--	--	--	--



4.2. NHẬN XÉT SAU KIỂM THỬ

- Chức năng được quan tâm:
- Chức năng gặp nhiều lỗi:
- Mức độ lỗi cao/ tổng số lỗi:

CHƯƠNG 5. CÔNG CỤ KIỂM THỬ.....

5.1. Tìm hiểu về công cụ

5.1.1. Giới thiệu về Selenium

Selenium là một bộ công cụ mã nguồn mở (open-source) được phát triển nhằm tự động hóa việc kiểm thử (automation testing) cho các ứng dụng web. Khác với các công cụ thương mại như UFT hay TestComplete, Selenium hoàn toàn miễn phí và hỗ trợ đa nền tảng, đa trình duyệt, đa ngôn ngữ lập trình.

Nguồn gốc:

- Được tạo ra bởi **Jason Huggins** vào năm **2004** khi ông đang làm việc tại công ty ThoughtWorks.
- Ban đầu có tên là “**JavaScriptTestRunner**”, sau đó được đổi thành **Selenium**.
- Ngày nay, Selenium được duy trì và phát triển bởi **Selenium Foundation** cùng cộng đồng lập trình viên trên toàn thế giới.

5.1.2. Mục đích và vai trò của Selenium trong kiểm thử phần mềm

Selenium được dùng để:

- **Tự động hóa** quá trình kiểm thử giao diện người dùng (UI).
- **Giảm thời gian kiểm thử thủ công** khi ứng dụng web có nhiều chức năng hoặc thay đổi thường xuyên.
- **Đảm bảo tính nhất quán và ổn định** giữa các lần kiểm thử.
- **Hỗ trợ tích hợp liên tục (CI/CD)** – giúp tự động kiểm thử sau mỗi lần code được cập nhật.

5.1.3. Các thành phần chính của Selenium

Selenium không phải là một công cụ đơn lẻ, mà gồm **bốn thành phần chính** hợp thành một bộ framework linh hoạt:

Thành phần	Mô tả chi tiết
1. Selenium IDE (Integrated Development Environment)	<ul style="list-style-type: none">- Là tiện ích mở rộng (extension) dành cho Chrome và Firefox.- Cho phép người dùng ghi lại (record), chạy lại (playback) và chỉnh sửa các bước kiểm thử trên trình duyệt.

Thành phần	Mô tả chi tiết
	<ul style="list-style-type: none"> - Không yêu cầu lập trình, phù hợp cho người mới. - Dạng test được lưu bằng file .side.
2. Selenium WebDriver	<ul style="list-style-type: none"> - Là thành phần mạnh mẽ nhất và phổ biến nhất. - Cho phép điều khiển trình duyệt thật thông qua mã lệnh (API). - Có thể viết test script bằng Python, Java, C#, Ruby, JavaScript, PHP, Kotlin,... - Hỗ trợ nhiều trình duyệt (Chrome, Edge, Firefox, Safari).
3. Selenium Grid	<ul style="list-style-type: none"> - Dùng để chạy song song nhiều test case trên nhiều máy (nodes) và nhiều trình duyệt khác nhau. - Giúp tiết kiệm thời gian kiểm thử. - Gồm 1 Hub (điều phối) và nhiều Node (máy thực thi).
4. Selenium Client API & Remote WebDriver	<ul style="list-style-type: none"> - Cho phép kết nối từ xa đến các môi trường test khác nhau. - Hỗ trợ viết test bằng nhiều ngôn ngữ lập trình khác nhau. - Tương tác với WebDriver thông qua giao thức JSON Wire Protocol.

5.1.4. Kiến trúc hoạt động của Selenium WebDriver

Sơ đồ khái quát:

[Test Script] → [Selenium Client Library] → [JSON Wire Protocol] → [Browser Driver] → [Browser]

Giải thích:

1. **Test Script:** Người kiểm thử viết mã lệnh (test case).
2. **Client Library:** Gửi yêu cầu điều khiển trình duyệt.
3. **Browser Driver:** (như ChromeDriver, GeckoDriver, EdgeDriver...) đóng vai trò trung gian giữa Selenium và trình duyệt.
4. **Browser:** Trình duyệt thực thi hành động như người dùng thật (click, nhập, điều hướng,...).

5.1.5. Ưu điểm của Selenium

- **Miễn phí, mã nguồn mở, cộng đồng hỗ trợ lớn.**

- **Đa trình duyệt, đa nền tảng**, dễ mở rộng.
- **Tích hợp linh hoạt** với nhiều framework: TestNG, JUnit, Maven, Jenkins, Docker,...
- **Chạy song song** (Parallel Testing) để tiết kiệm thời gian.
- **Hoạt động mô phỏng người dùng thật**, giúp phát hiện lỗi giao diện và luồng xử lý chính xác.

5.1.6. Hạn chế của Selenium

- Chỉ hỗ trợ ứng dụng web, không áp dụng cho desktop app hoặc mobile app (trừ khi kết hợp Appium).
- Không có giao diện quản lý lỗi hoặc báo cáo tích hợp sẵn – cần dùng thêm framework hỗ trợ (Allure, Extent Report,...).
- Cần kiến thức lập trình để viết và bảo trì script.
- Dễ gặp lỗi khi giao diện web thay đổi (ID, XPath, CSS Selector,...).

5.2. Sử dụng công cụ

5.2.1. Chuẩn bị môi trường cài đặt (ví dụ với Python)

Bước 1: Cài đặt Python

Tải từ <https://www.python.org> và cài đặt bình thường.

Bước 2: Cài đặt Selenium

Mở Terminal hoặc CMD, gõ lệnh:

```
pip install selenium
```

Bước 3: Tải **WebDriver** tương ứng với trình duyệt bạn dùng, ví dụ:

- Chrome → <https://chromedriver.chromium.org/downloads>
- Edge → <https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/>

Bước 4: Giải nén và đặt chromedriver.exe vào thư mục chứa project hoặc đường dẫn hệ thống.

5.2.2. Cấu trúc thư mục dự án kiểm thử

selenium_test/

```
|
|— driver/
|   └── chromedriver.exe
|
|— test_scripts/
|   ├── login_test.py
|   └── search_test.py
|
└── report/
    └── result_log.txt
```

5.2.3. Ví dụ mã kiểm thử bằng Selenium WebDriver (Python)

Kiểm thử chức năng đăng nhập (Login Test):

```
from selenium import webdriver
```

```
from selenium.webdriver.common.by import By
```

```
import time
```

```
# Khởi tạo trình duyệt Chrome
```

```
driver = webdriver.Chrome()
```

```
# Mở trang web cần kiểm thử
```

```
driver.get("https://example.com/login")
```

```
# Nhập thông tin tài khoản
```

```
driver.find_element(By.ID, "username").send_keys("tester01")
```



```
driver.find_element(By.ID, "password").send_keys("123456")
```

```
# Nhấn nút đăng nhập
```

```
driver.find_element(By.ID, "login-button").click()
```

```
# Chờ trang tải
```

```
time.sleep(3)
```

```
# Kiểm tra kết quả đăng nhập
```

```
if "dashboard" in driver.current_url:
```

```
    print("✅ Đăng nhập thành công!")
```

```
else:
```

```
    print("❌ Đăng nhập thất bại!")
```

```
# Đóng trình duyệt
```

```
driver.quit()
```

Giải thích:

- `driver.get()` – mở URL trang web.
- `find_element(By.ID, "username")` – tìm ô nhập liệu có ID tương ứng.
- `send_keys()` – nhập dữ liệu vào ô đó.
- `click()` – nhấn nút.
- `current_url` – lấy URL hiện tại để kiểm tra điều kiện thành công.

5.2.4. Kiểm thử song song với Selenium Grid

- Cài đặt **Hub** và **Node**.
- Hub phân phối các test case đến nhiều Node.
- Mỗi Node có thể chạy test trên trình duyệt hoặc hệ điều hành khác nhau.

👉 Giúp giảm thời gian chạy test và đảm bảo tính tương thích đa môi trường.

5.2.5. Tích hợp Selenium với công cụ khác

Công cụ	Mục đích
JUnit / TestNG	Quản lý test case, tổ chức test suite, tạo báo cáo tự động.
Maven / Gradle	Quản lý thư viện và build project.
Jenkins	Tự động chạy kiểm thử sau mỗi lần cập nhật code (CI/CD).
Allure / Extent Report	Tạo báo cáo kết quả kiểm thử trực quan (HTML, dashboard).

5.2.6. Kết quả thực nghiệm và nhận xét

Sau khi chạy thử test script:

- Trình duyệt tự động mở và nhập dữ liệu.
- Kết quả in ra console hoặc lưu vào file báo cáo.
- Nếu login đúng, test báo “**Pass**”; nếu sai, báo “**Fail**”.

Nhận xét:

- Kiểm thử tự động giúp **tiết kiệm thời gian**, đặc biệt với test lặp lại nhiều lần.
- Độ chính xác cao hơn so với test thủ công.
- Tuy nhiên, cần **đầu tư thời gian ban đầu** để xây dựng và bảo trì script.

5.3. KẾT LUẬN CHƯƠNG

- Selenium là công cụ **kiểm thử tự động web** mạnh mẽ, phổ biến và dễ tích hợp.
- Việc nắm vững cách **sử dụng WebDriver, tổ chức test case và tích hợp CI/CD** sẽ giúp nâng cao năng suất kiểm thử phần mềm.
- Trong tương lai, Selenium sẽ tiếp tục phát triển, hỗ trợ tốt hơn cho các ứng dụng web hiện đại (SPA, React, Angular,...).

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN