



LACLONGQUAN
Up and Running

<https://app.helptotest.com>

**Giới thiệu hệ thống
LLQ QA Platform**

Nội dung trình bày

Giới thiệu LLQ QA Platform

Phương pháp automation testing

Dịch vụ kiểm thử

Trao đổi & Demo

LLQ QA Platform là gì?

- LLQ QA Platform là hệ thống đảm bảo chất lượng phần mềm.
- Được phát triển hoàn toàn bởi Công ty PM LLQ.
- Hướng đến mô hình quản lý trung tâm kiểm thử TCoE.

Mục tiêu hệ thống hướng đến

- Hệ thống trực quan, cấu hình đủ nhanh để áp dụng Automation Test vào công việc kiểm thử hàng ngày.
- Khả năng tự động hóa >95% số testcase.
- End-to-End testing, Parallel testing.
- Khả năng quản lý regression,... test với độ bao phủ có chủ đích.
- Quy trình quản lý, triển khai được định nghĩa linh hoạt.
- Tùy biến báo cáo chất lượng, báo cáo nghiệm thu dự án.
- Tích hợp tốt với CI/CD và các hệ thống khác.

LLQ QA Platform

- Là hệ thống tập trung.
- Vận hành 100% trên nền tảng web.
- Hỗ trợ kiểm thử đa dạng ứng dụng & trên cùng test scenario duy nhất.

ỨNG DỤNG HỖ TRỢ KIỂM THỬ TỰ ĐỘNG

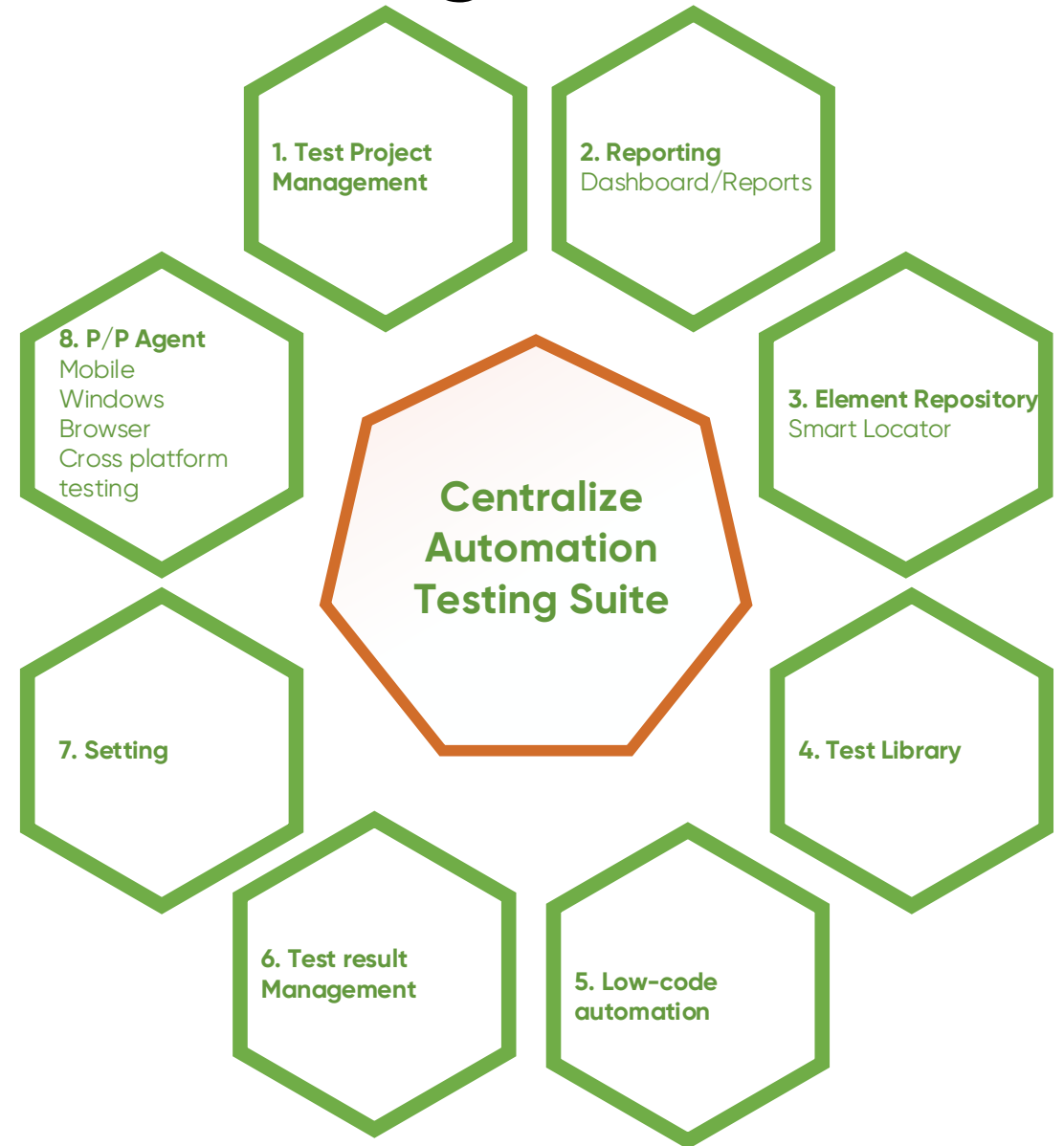
ỨNG DỤNG WEB

ỨNG DỤNG MOBILE

ỨNG DỤNG WINDOWS

Chức năng chính của hệ thống

1. Test Project Mgt
2. TestOps Dashboard/Report
3. Element Repository
4. Test Library
5. Low-code Test Configuration
6. Test run & result Mgt
7. Setting
8. P/P Agent



1. Test Project

- Phiên bản kiểm thử
- Môi trường triển khai.
- Profile kiểm thử.
- Tham số dùng chung.
- Module ứng dụng.
- Quản lý gắn nhãn.
- Người dùng.
- Tích hợp.

The screenshot shows the 'Project Teammate' management interface. At the top, there is a 'List project' dropdown menu with 'VIETINS' selected. The main navigation bar includes tabs for 'SET UP', 'TEAMMATE', 'PARAMETER', 'MODULE', 'VERSION', 'ENVIRONMENT', 'PROFILE', 'LABEL', and 'PROJECT'. The current view is titled 'Project Teammate (6)' and includes a 'Filter form' with two input fields: 'Teammate name' and 'Teammate email', along with 'SEARCH' and 'RESET' buttons. Below the filter form is a table listing six team members, each with a circular profile icon, name, and email address. At the bottom right, there are three green buttons: 'TEAMMATE', 'EXPORT', and 'IMPORT'.

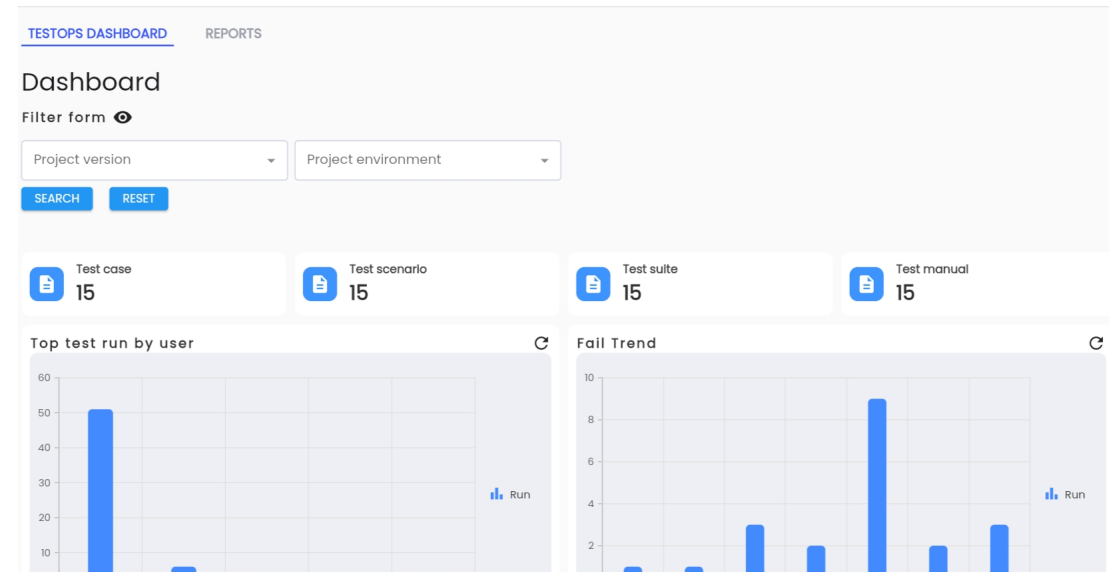
Profile Icon	Name	Email	Actions
A	Anh NN	anhnn@lq.vn	⋮
B	Bạch Mạnh Linh	admin@lq.vn	⋮
L	Linh BM	linhbm@lq.vn	⋮
Q	Quân NM	quannm@lq.vn	⋮
D	Doanh TV	doanh@lq.vn	⋮
Q	Quỳnh DN	quynh@lq.vn	⋮

2. Dashboard & Report

Hệ thống báo cáo động với các chỉ tiêu test metric được định nghĩa.

Người dùng có thể dễ dàng tùy biến với các loại như sau:

- Dashboard tổng quan Test Project.
- Dashboard Test Library.
- Dashboard Test Run.
- Báo cáo nghiệm thu.
- Báo cáo chất lượng dự án Test project.



The screenshot shows a report table with a 'DOWNLOAD' button. The table has columns for 'FULLNAME', 'EMAIL', 'WAITING', 'PASS', and 'FAIL'. The data is as follows:

FULLNAME	EMAIL	WAITING	PASS	FAIL
Anh NN	anhntn@lq.vn	0	5	11
Bạch Mạnh Linh	admin@lq.vn	0	1	9
Đặng Nam	nampd@lq.vn	0	14	35
Dang Thi Trang Nhung	nhungdtt@lq.vn	0	0	0
Doanh TV	doanh@lq.vn	0	2	3
Lê Thị Thảo Vân	lethithaovan31122003@gmail.com	0	0	1
Linh BM	linhbm@lq.vn	0	2	0
Nguyễn Đình Tiến	tiendn@lq.vn	0	25	6
Trần Thị Hoa	hoath@lq.vn	0	8	15

2. Software Testing Metrics

Hệ thống báo cáo động với các chỉ tiêu test metric được định nghĩa.

Người dùng có thể dễ dàng tùy biến với các loại như sau:





- Dashboard tổng quan Test Project.
- Dashboard Test Library.
- Dashboard Test Run.
- Báo cáo nghiệm thu.
- Báo cáo chất lượng dự án Test project.

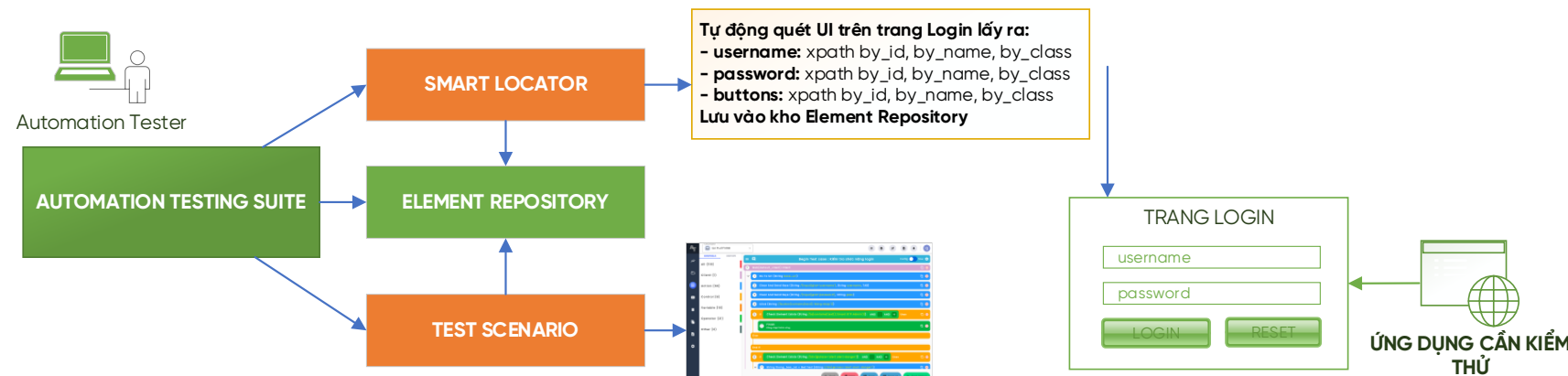
3. Element Repository

- “Element repository” (hay còn được gọi là “object repository”) là module quan trọng trong quản lý và tái sử dụng các thành phần (element) của ứng dụng hoặc trang web đang kiểm thử.
- Điều này giúp tối ưu hóa quá trình viết và bảo trì các kịch bản kiểm thử tự động.
- Element repository chứa thông tin về các thành phần giao diện người dùng như các nút, trường văn bản, dropdowns, checkbox, radio buttons và các phần tử khác mà người dùng muốn tương tác trong quá trình kiểm thử.
- Được tổ chức dưới dạng thư mục theo các màn hình chức năng của ứng dụng hoặc trang web mà người dùng thực hiện kiểm thử.

3. Locator – Định danh phần tử

- Là cách người dùng xác định và truy cập các thành phần trong mã nguồn của ứng dụng hoặc trang web, ví dụ như ID, class, CSS selector, XPath, etc.
- Thông qua Ứng dụng smart locator mà hệ thống cung cấp
- Thông qua tìm trực tiếp bằng công cụ Find Element trực quan.

Test Scenario ID	s_login_001	Test Scenario Description	Test the Login Functionality in Banking			
Created By	quannm - Tester	Reviewed By	Quýnh - Testleader	Version	2.1	
Steps	Description	Keyword	Locator	Datasource	Data	Data 1
step1	Đi đến trang login của ứng dụng cần kiểm thử	Go to Url 		envioment variable		
step2	Nhập username	Clear and Send Keys 	Element Repository "username" //input[@id='username']	parameter	linhbm	quynhnd
step3	Nhập password	Clear and Send Keys 	Element Repository "password" //input[@id='password']	parameter
step4	Bấm nút đăng nhập	Click 	Element Repository "Nút Đăng nhập" //button[contains(text(),'Đăng nhập')]			



4. Test Library

Công cụ được thiết kế để quản lý, tổ chức, và tái sử dụng các kịch bản kiểm thử, dữ liệu kiểm thử, và các thành phần khác liên quan đến quy trình kiểm thử tự động:

- Quản lý thư viện kiểm thử bao gồm test scenario, test case, test suite, API...
- Quản lý các tập kiểm thử tự động như smoke, sanity, regression test...
- Hỗ trợ luồng kiểm duyệt.
- Quản lý cấu hình bao gồm cả dữ liệu kiểm thử của testcase.
- Khả năng liên kết đến các module chức năng khác của hệ thống liên mạch.

The screenshot displays the 'QA PLATFORM' interface for the 'SCENARIO' tab. It features a 'Directory Library' section with a 'Filter form' containing search fields for test code, directory name, project version, environment, status, module, owner, and label. Below the filters is a table of test scenarios with columns for CODE, NAME, OWNER, CREATE DATE, STATUS, and LAST RUN. The table lists four scenarios, all created on 08-01-2024 and owned by Anh NN. The first two scenarios have a status of 'PASSED' (indicated by green play buttons and four empty bars), while the last two have a status of 'FAILED' (indicated by green play buttons and three red bars). At the bottom right, there are buttons for 'NEW DIRECTORY' and 'NEW SCENARIO'.

CODE	NAME	OWNER	CREATE DATE	STATUS	LAST RUN
ts_user_001	Kiểm tra chức năng tạo mới người dùng	Anh NN	08-01-2024	PASSED	08-01-2024
ts_role_001	Kiểm tra chức năng tạo mới quyền	Anh NN	08-01-2024	PASSED	08-01-2024
ts_login_001 _positive	Kiểm tra đăng nhập	Anh NN	08-01-2024	FAILED	08-01-2024
ts_login_002 _negative	Kiểm tra đăng nhập với các trường hợp khác	Anh NN	08-01-2024	FAILED	08-01-2024

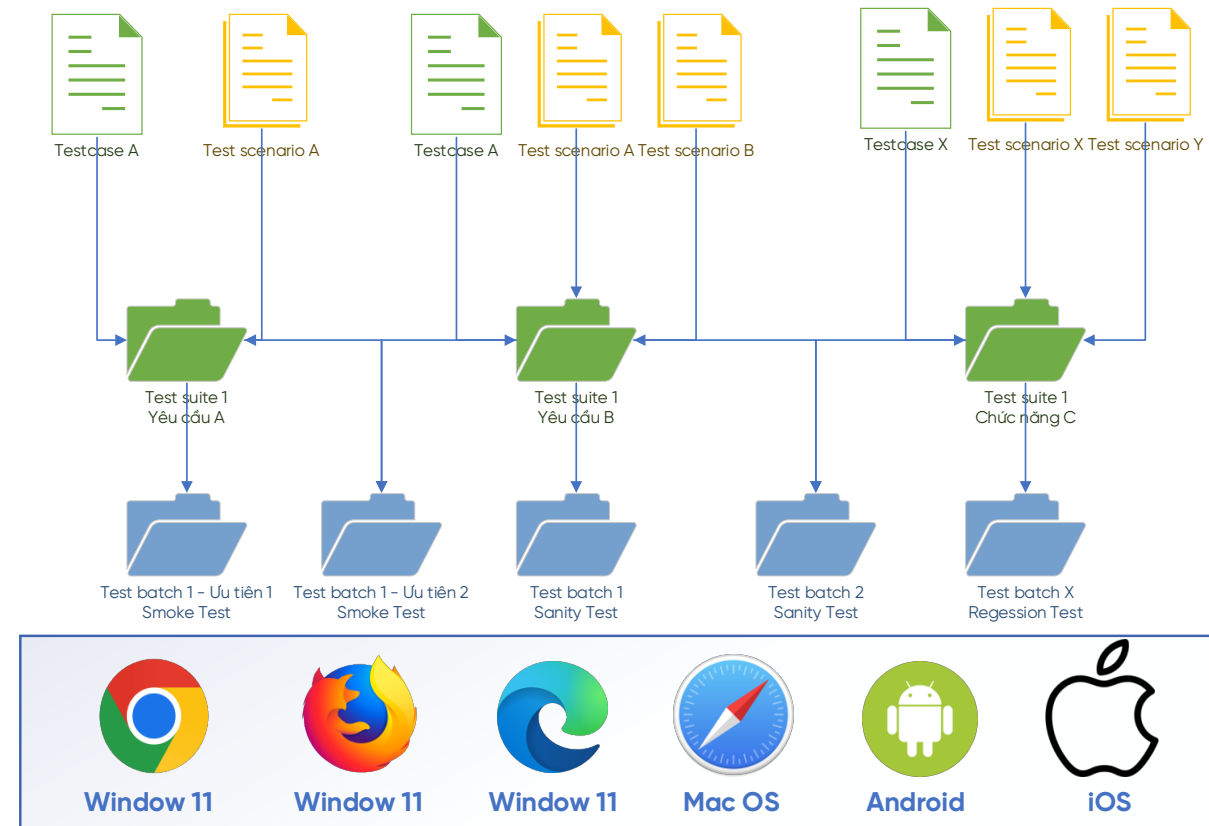
4. Test Library

Test Scenario (Kịch bản kiểm thử) :

- Mô tả một tình huống hoặc tập hợp các tình huống cụ thể mà hệ thống cần được kiểm tra.
- Đây là mức cao nhất trong tổ chức của các kịch bản kiểm thử và thường được sử dụng để mô tả các yêu cầu chức năng hoặc tính năng của hệ thống.

Test case:

- Là một bộ hướng dẫn chi tiết cho việc thực hiện kiểm thử cho một chức năng, tính năng hoặc thành phần cụ thể của hệ thống.
- Bao gồm các bước kiểm thử, điều kiện tiên đề, dữ liệu kiểm thử, kết quả mong đợi và các bước để xác nhận kết quả.



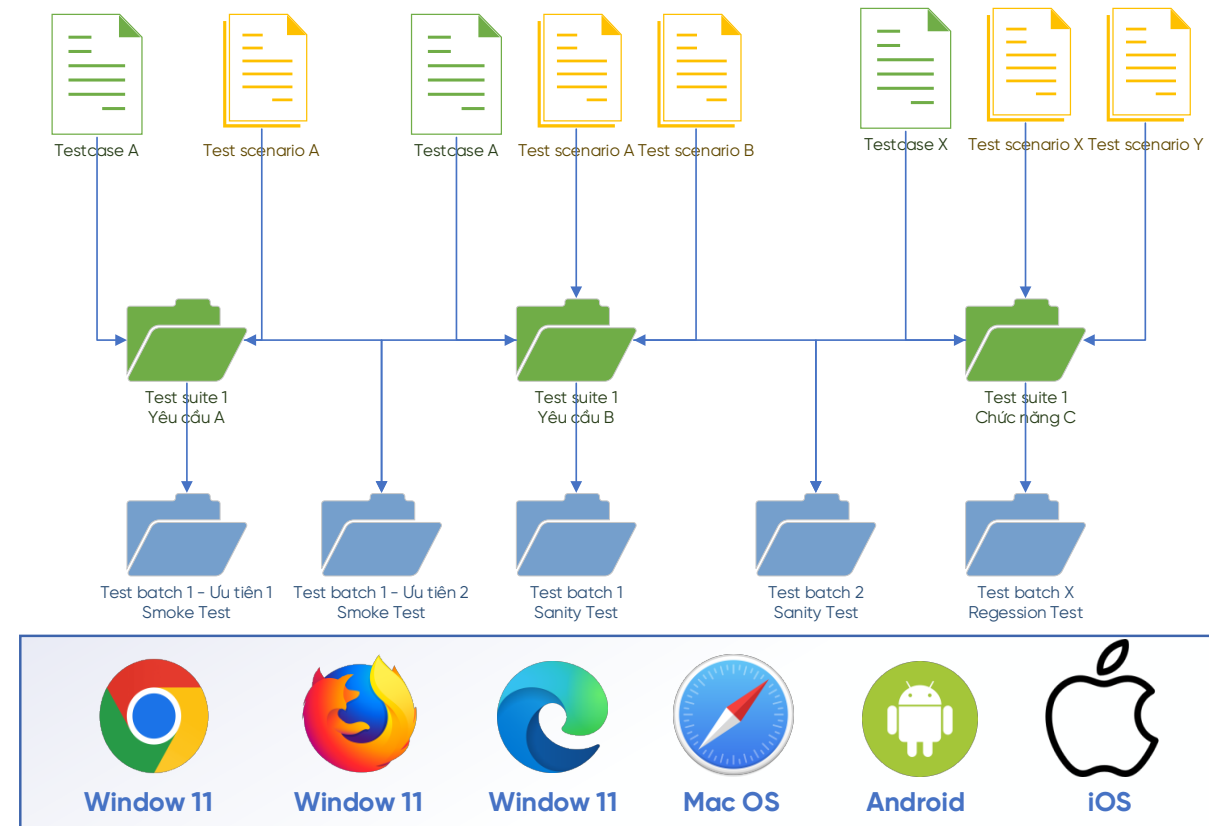
4. Test Library

Test Suite (Bộ kiểm thử):

- Là một tập hợp hoặc nhóm các kịch bản kiểm thử có liên quan, được tổ chức lại để thực hiện kiểm thử cho một tính năng, module hoặc toàn bộ ứng dụng.
- Được sử dụng để tổ chức và quản lý các ca kiểm thử một cách hiệu quả và có thể được thực hiện như một đơn vị kiểm thử độc lập.

Test Batch (Lô kiểm thử):

- Là một nhóm các bộ kiểm thử, thường được tổ chức và thực hiện cùng nhau để kiểm tra một phần hoặc toàn bộ hệ thống trong một lần thực hiện.
- Được sử dụng để tổ chức và quản lý các bộ kiểm thử trong các kỳ kiểm thử hoặc vòng kiểm thử cụ thể.



4. Test Batch

- Hệ thống cung cấp đầy đủ cơ chế hỗ trợ lựa chọn Testcase cho các tập kiểm thử độ bảo phủ lớn như Hồi quy(regression).
- Đặc biệt, hệ thống hỗ trợ phân loại, đánh giá và gợi ý các trường hợp kiểm thử dựa trên tính tiệm cận Rủi ro cao.
- Hệ thống cung cấp tính năng giúp người dùng xác định thế nào là một tập kiểm thử thành công hoặc thất bại.
- Hệ thống hỗ trợ thiết kế, sắp xếp, đánh dấu các trường hợp kiểm thử trong các tập kiểm thử.
- Ngoài ra, với các testcase đã thiết lập, người dùng sẽ không cần phải chuẩn bị lại dữ liệu.

DASHBOARD TEST CASE SCENARIO SUITE MANUAL **BATCH**

Test Batch Runs (1)

Filter form

Search batch code Search batch name Search status Test run owner

Create date from Create date to

SEARCH RESET

TEST BATCH CODE	TEST BATCH NAME	SCHEDULE DATE	CREATE DATE	OWNER	STATUS
smoke_test_positive	Smoke Test (positive)	19-03-2024 11:46:00	19-03-2024 11:46:41	Anh NN	Run pass

DASHBOARD TEST CASE SCENARIO SUITE MANUAL **BATCH**

< Smoke Test (positive)

Run owner Anh NN

Schedule time 19-03-2024 11:46:00

Start run 19-03-2024 11:47:38

End run 19-03-2024 11:51:22

Total run time 58,50 (s)

Status **Run pass**

3 Passed
0 Failed
19,50s Average duration of Test run data

Batch Run Step

BATCH STEP NAME	TEST RUN TYPE	TEST CASE	VERSION	ENVIRONMENT	CREATE DATE	STATUS
Đăng nhập thành công	parallel	Kiểm tra đăng nhập	Version1	UAT	19-03-2024 11:46:50	Pass
Thêm mới quyền	parallel	Đăng nhập hệ thống & Thêm mới quyền	Version1	UAT	19-03-2024 11:46:50	Pass
Thêm mới người dùng	parallel	Đăng nhập hệ thống & Thêm người dùng	Version1	UAT	19-03-2024 11:46:50	Pass

4. Bảng cấu hình Test Batch

Testcase Priority(0-high; 1-medium; 2-low) - Thứ tự ưu tiên của Testcase

Testcase Regression Candidate(0-Primary; 1-Secondary; 2-Tertiary) - chỉ số hệ thống tự động tính toán và gợi ý thông qua 10 lần chạy gần nhất, bất kể số lượng data

BATCH STATUS	BUGS SEVERITY	Batch 1 - TC Priority (0) - TC RC(0,1,2)	Batch 2 - TC Priority (1) - TC RC(0,1,2)	Batch 3 - TC Priority (2) - TC RC(0,1,2)	Batch 4 - TC Priority (0,1,2) - TC RC(0,1,2)
PASS	Critical	0%	x%	x%	x%
	Major	0%	x%	x%	x%
	Average	5%	x%	x%	x%
	Minor	10%	x%	x%	x%
	Trivial	15%	x%	x%	x%
WARNING	Critical	1%	x%	x%	x%
	Major	2%	x%	x%	x%
	Average	20%	x%	x%	x%
	Minor	20%	x%	x%	x%
	Trivial	20%	x%	x%	x%
FAIL					

5. Low-code Test Scenario

- Low-code test scenario của hệ thống là một phương pháp hướng từ khóa - keywords trong kiểm thử tự động, không yêu cầu người dùng phải viết mã từ đầu.
- Các keywords được hệ thống xây dựng sẵn dưới dạng các blocks - chuỗi khối, đáp ứng mọi yêu cầu cho kịch bản kiểm thử từ đơn giản đến phức tạp.
- Dễ học dễ sử dụng.
- Kiểm thử nhiều ứng dụng trong cùng kịch bản.
- Dễ dàng phát triển thêm các keywords phục vụ theo yêu cầu.

The image displays two screenshots of a low-code test scenario builder interface. The top screenshot shows a 'KEYWORDS' library with a search bar and a list of keyword categories and their counts: All (133), Client (1), Action (69), Control (9), Variable (18), Operator (21), File (12), and Other (3). To the right, a list of keyword blocks is shown, including 'Action Call', 'Click', 'Child Element Click', 'Clear And Send Keys', 'Child Element Clear And Send Keys', and 'Send Key From Element'. Each block lists supported platforms: Web, Android, iOS, Windows, and JavaSwing.

The bottom screenshot shows a test flow editor for a project named 'QA PLATFORM'. The flow is titled 'Begin Kiểm tra chức năng tạo mới người dùng'. It contains six steps:

1. Click (tạo mới / Hệ thống # by_text) - kích hệ thống trên menu
2. Click (tạo mới / Người dùng # by_text) - kích người dùng trên submenu
3. Click (tạo mới / Thêm # by_text) - kích nút thêm trên lưới giao diện
4. Select By Visible Text (tạo mới / Chi nhánh # by_name, HÀ NỘI - Chi nhánh Hà Nội) - Chọn chi nhánh Hà Nội
5. Clear And Send Keys (tạo mới / username - Nhập tên đăng nhập - tên đăng nhập # by_name, tên đăng nhập) - Nhập tên đăng nhập
6. Clear And Send Keys (tạo mới / fullname - Nhập họ tên - Họ tên # by_name, họ tên) - Nhập họ tên đầy đủ

At the bottom of the flow editor, there are buttons for 'Back', 'Delete', 'Save', 'Save as', and 'Start Debug'.

5. Low-code Test Scenario

Lợi ích của việc sử dụng low-code test scenario bao gồm:

- **Tăng tốc độ triển khai:** Low-code cho phép các nhóm kiểm thử nhanh chóng tạo ra và triển khai các kịch bản kiểm thử mà không cần đến việc viết mã phức tạp, giúp tiết kiệm thời gian và tài nguyên.
- **Đễ dàng sử dụng:** Giao diện đồ họa và các công cụ kéo và thả giúp người dùng không cần có kiến thức chuyên sâu về lập trình cũng có thể tạo ra các kịch bản kiểm thử một cách dễ dàng.
- **Tính tái sử dụng cao:** Các kịch bản kiểm thử tạo ra thông qua low-code thường có thể tái sử dụng dễ dàng, giảm thiểu việc lặp lại công việc và tăng cường hiệu quả.
- **Tăng cường tính linh hoạt:** Các nhóm kiểm thử có thể dễ dàng điều chỉnh và cập nhật các kịch bản kiểm thử theo yêu cầu mới mà không cần phải thay đổi mã nguồn, giúp họ linh hoạt hơn trong việc đáp ứng nhanh chóng các yêu cầu kiểm thử.
- **Giảm thiểu lỗi do con người:** Bằng cách sử dụng giao diện đồ họa và công cụ tự động hóa, low-code giúp giảm thiểu lỗi từ phía người dùng trong quá trình tạo và thi hành các kịch bản kiểm thử.
- **Tối ưu hóa quy trình kiểm thử:** Low-code giúp tối ưu hóa và tự động hóa các bước trong quy trình kiểm thử, từ việc tạo kịch bản, thực thi, đánh giá kết quả, đến báo cáo và quản lý kết quả kiểm thử.

6. Test Run

Quản lý test run và test run results là một phần quan trọng trong quy trình kiểm thử phần mềm. Hệ thống hỗ trợ cấu hình trực quan như sau:

- Xác định phạm vi, mục tiêu, và các yêu cầu kiểm thử cho mỗi test run.
- Chuẩn bị dữ liệu và môi trường cần thiết cũng như loại hình thực thi cho test run.
- Tổ chức các test cases vào các test suites hoặc test batches dựa trên tính liên quan và ưu tiên.
- Đảm bảo rằng các test cases được sắp xếp một cách hợp lý và dễ dàng để thực hiện và kiểm tra kết quả rõ ràng.

6. Test Result

Ghi nhận và Lưu trữ:

- Ghi lại chi tiết về kết quả của mỗi test case, bao gồm cả các kết quả thành công và lỗi phát sinh.
- Lưu trữ các kết quả test run một cách có hệ thống và dễ dàng truy cập để kiểm tra lại hoặc phân tích sau này.

Phân Tích và Đánh Giá:

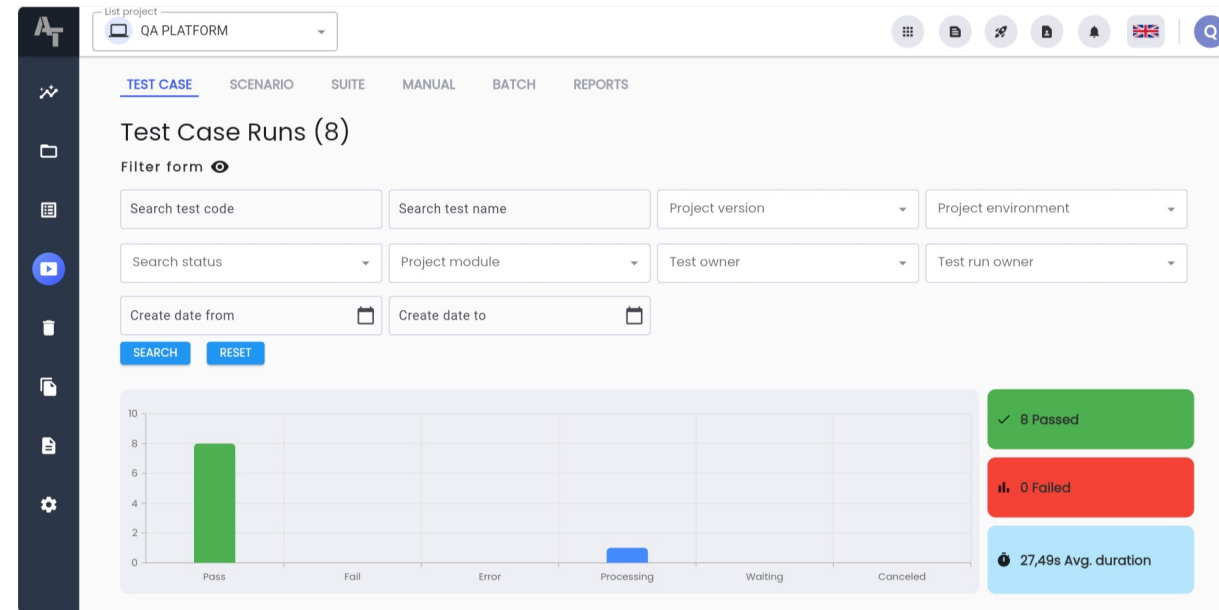
- Phân tích kết quả test run để xác định hiệu suất, độ tin cậy, và chất lượng của hệ thống hoặc tính năng được kiểm thử.
- Đánh giá các lỗi và vấn đề đã phát hiện để xác định nguyên nhân, ưu tiên, và các biện pháp khắc phục.

Báo Cáo:

- Tạo báo cáo tổng hợp về kết quả test run, bao gồm cả các thành công, lỗi, và các vấn đề cần xử lý...

6. Test Run & Result

- Hỗ trợ thực hiện kiểm thử tuần tự hoặc kiểm thử đồng thời nhiều testcase với bộ dữ liệu đa dạng.
- Cơ chế kiểm thử dữ liệu tập trung, có tính tái sử dụng cao, bao gồm cả dữ liệu random – tự sinh.
- Báo cáo kết quả trực quan, chi tiết
- Khả năng debug chi tiết đến vị trí xảy ra lỗi trong kịch bản kiểm thử.
- Record video, hình ảnh lỗi làm cơ sở report bug.
- Cơ chế đặt lịch thực thi kiểm thử.
- Report bugs tự động hoặc manual....



DASHBOARD TEST CASE SCENARIO SUITE MANUAL BATCH

← Data 1

TEST RUN RESULT	TEST RUN INFORMATION	TEST RUN DATA
Expected result Fail	Agent llq	Test case testcase1
Actual result Fail	Start run 11-03-2024 16:26:40	Test Run Expected Result Fail
Final status Pass	End run 11-03-2024 16:26:55	Severity Critical
Report bug Not reported	Total run time 15,00 (s)	Priority Highest
	Version Alpha	username linhbm
	Environment Alpha	password abc@123

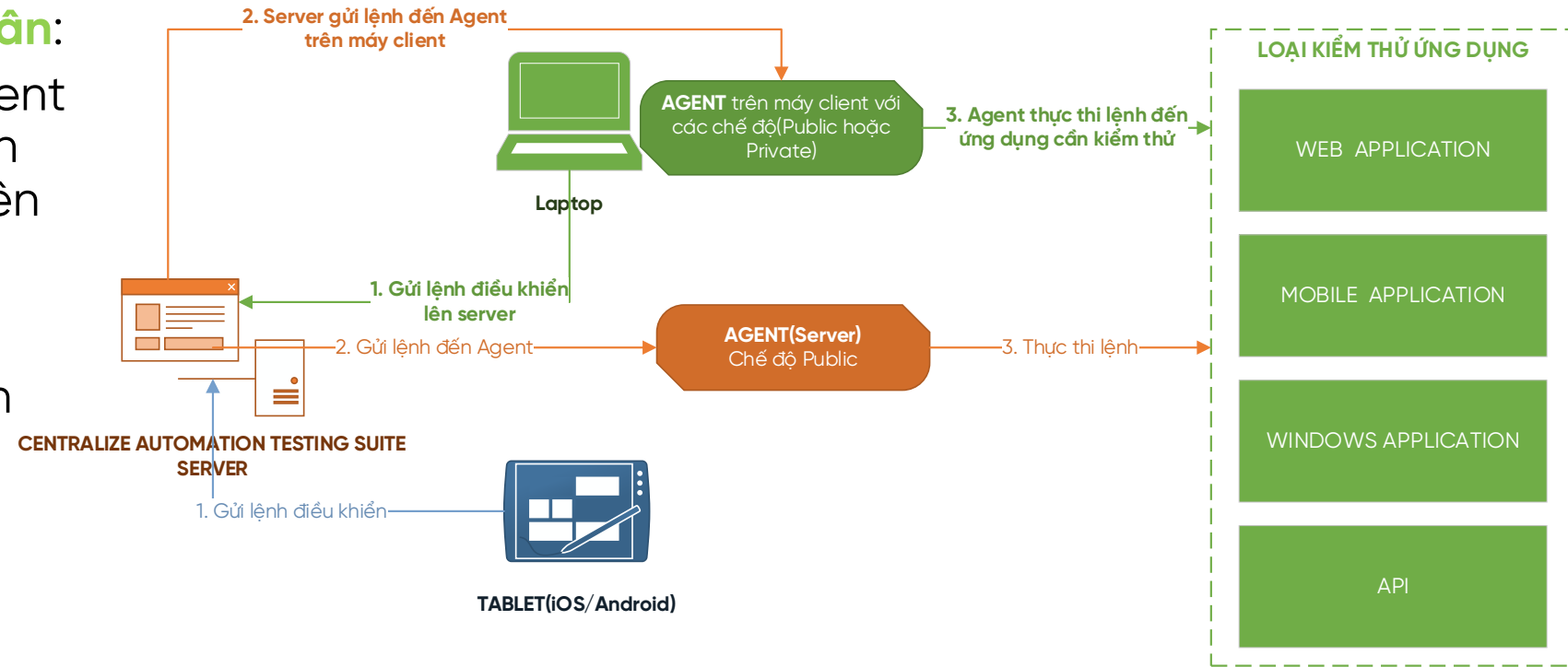
Actual result description
no such window: target window already closed
from unknown error: web view not found

7. Cấu hình & Tích hợp

- Tích hợp AD/Email
- Tích hợp Jira
- Cấu hình:
 - Notification template
 - Email template
- Random dữ liệu
 - Dữ liệu có nghĩa trong danh sách hoặc tự tạo ra dữ liệu
 - Dữ liệu duy nhất
- Bộ giá trị dùng chung
- Cấu hình project type (công nghệ phát triển của ứng dụng cần kiểm thử) bao gồm các bộ template phục vụ định danh & find element
- Cấu hình Role

8. P/P Agent

- **Tester sử dụng máy tính cá nhân:**
 - Có thể sử dụng Private Agent chạy trên máy tính cá nhân hoặc Public Agent chạy trên Server(cài đặt ứng dụng)
- **Tester sử dụng tablet/mobile:**
 - Agent chạy hoàn toàn trên Server(cài đặt ứng dụng)



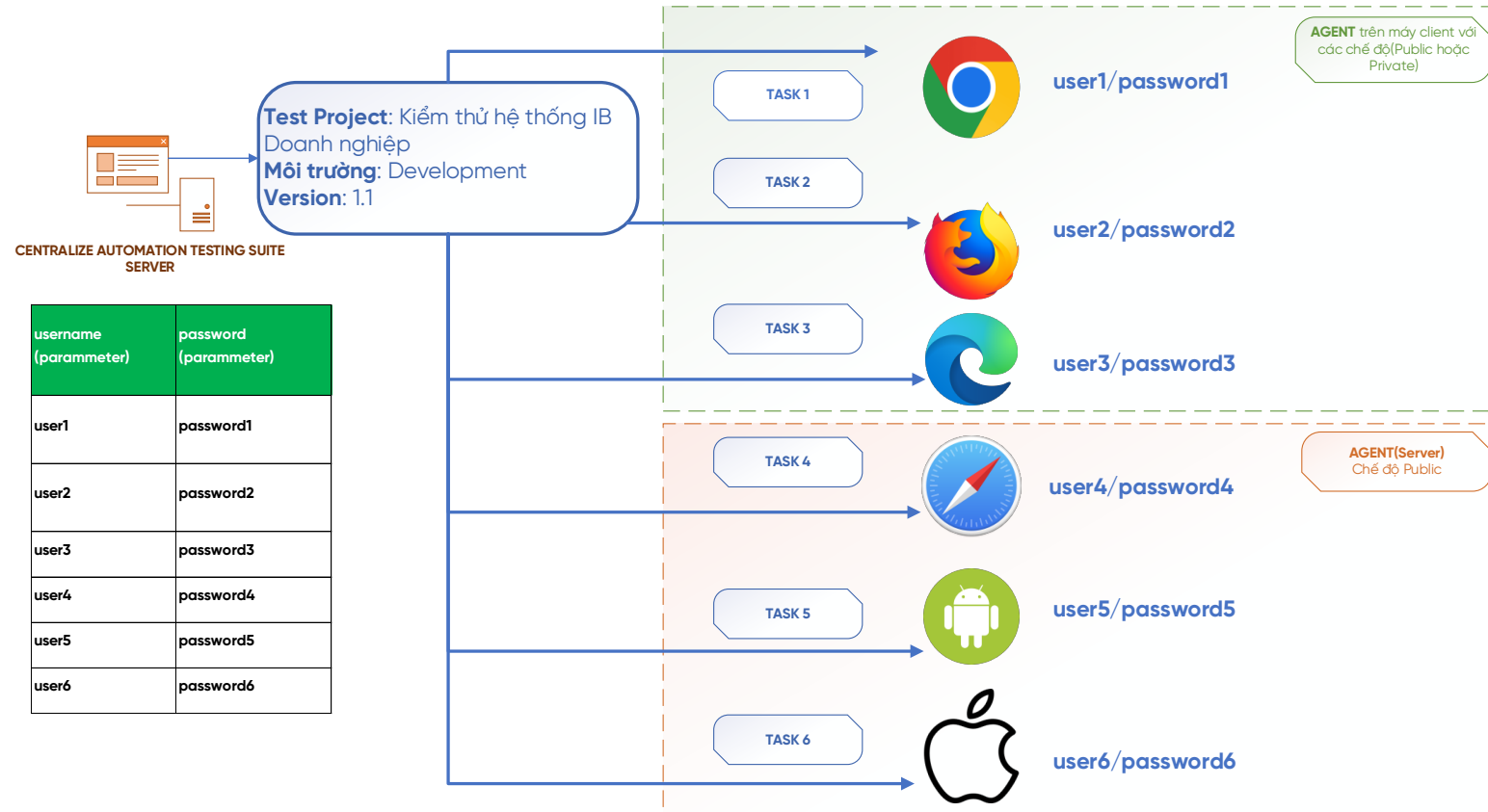
Agent: Sử dụng chung trong toàn hệ thống, không phụ thuộc vào Test Project, được chia làm:

Public agent: bất cứ người dùng nào cũng có thể nhìn thấy và sử dụng. Được sử dụng trong mục đích để chia sẻ tài nguyên khi thực thi số lượng lớn các testcase/testdata

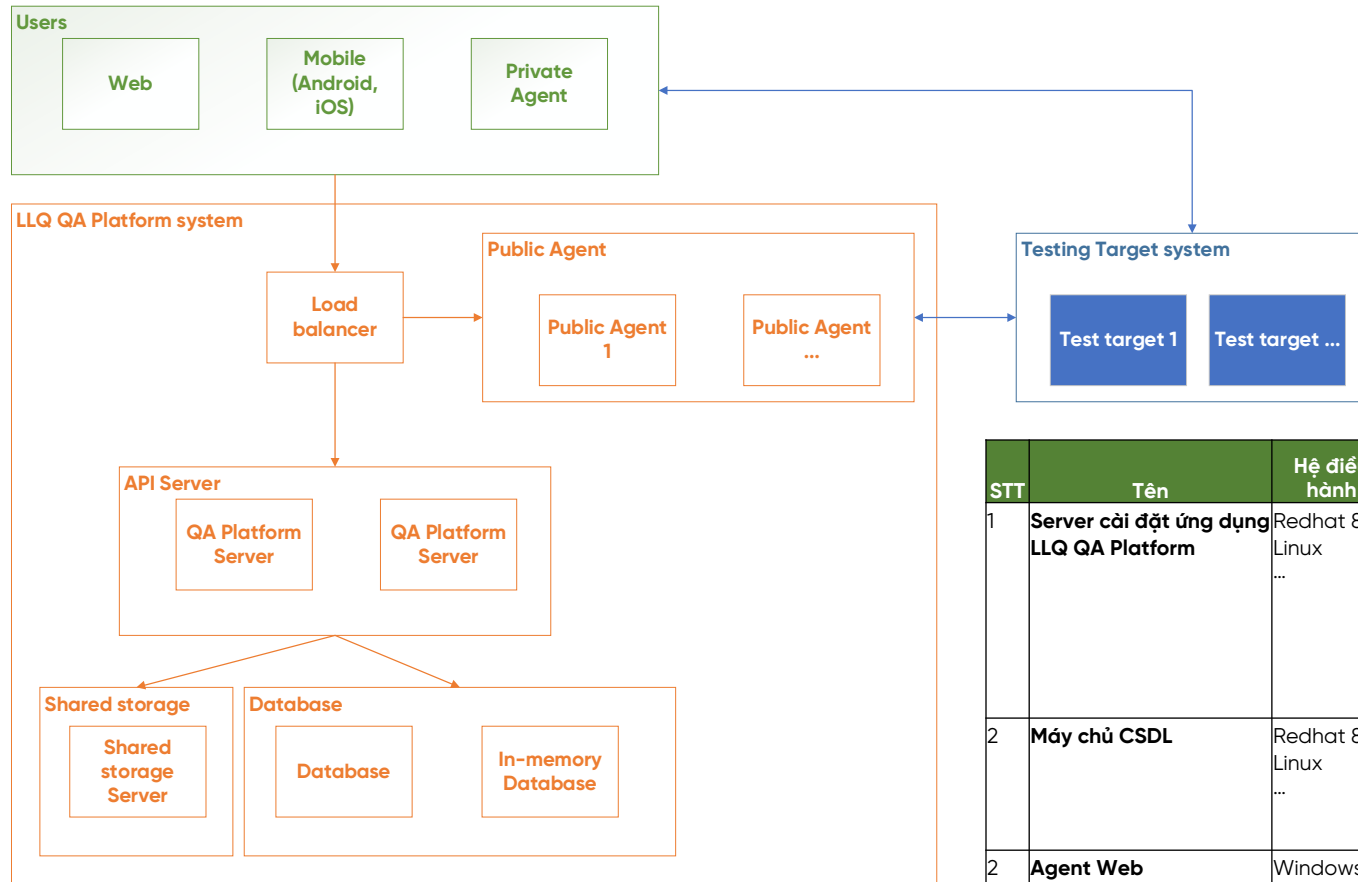
Private agent: phục vụ cấu hình test scenario & run trong mục đích cá nhân(note: đang cấu hình không nên share thành public)

8. Parallel Testing

- Với cơ chế quản lý lệnh tập trung.
- Khả năng phân phối & thực hiện task tự động theo P/P Agent.
- Khả năng tự động thêm các Agent vào quá trình thực thi mà không ảnh hưởng đến quá trình chạy.
- Tăng tốc độ thử nghiệm với các tập kiểm thử lớn như Regression, Sanity...



Hạ tầng & Mô hình triển khai



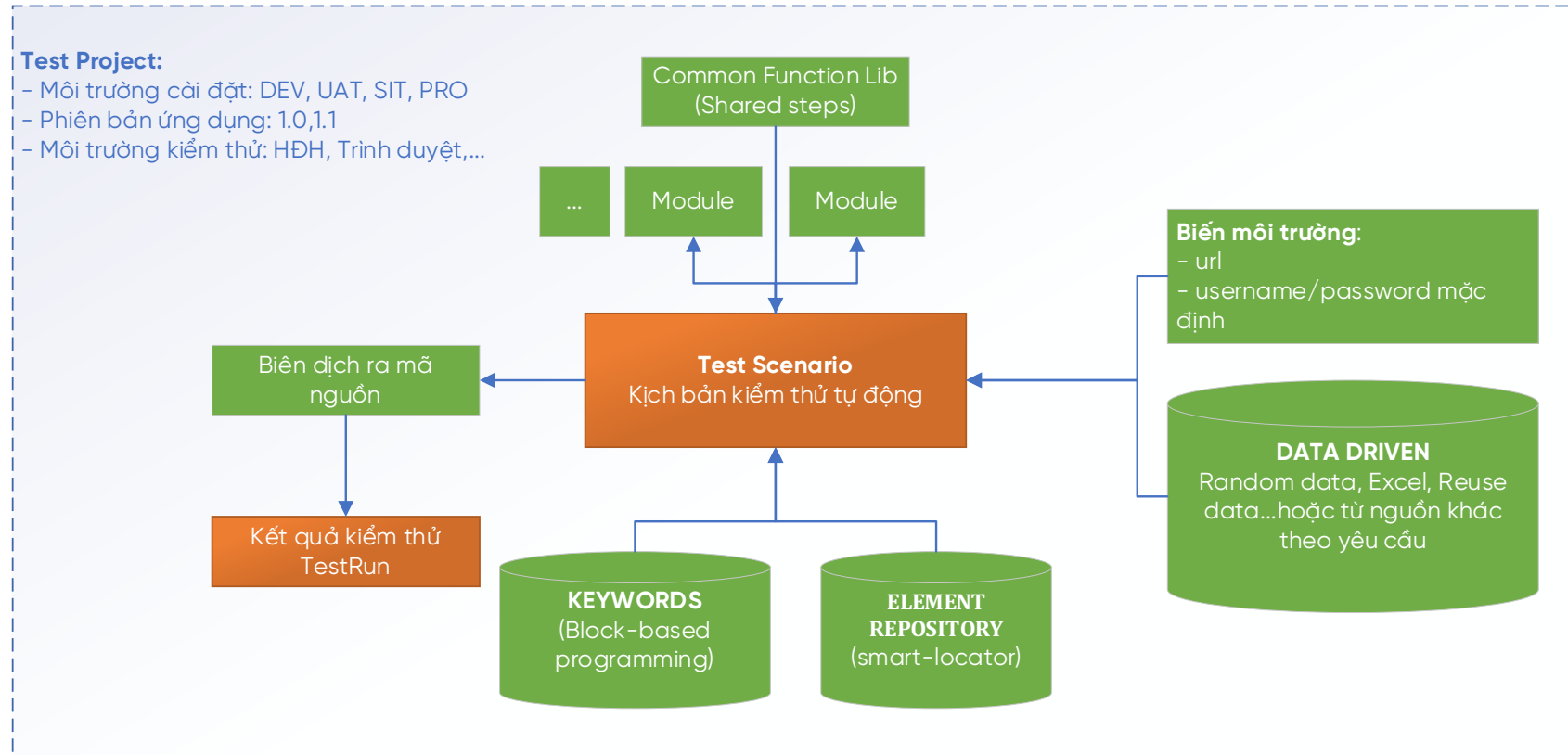
STT	Tên	Hệ điều hành	Phần cứng	Phần mềm nên cài đặt	Cài đặt mạng	Yêu cầu bổ sung
1	Server cài đặt ứng dụng LLQ QA Platform	Redhat 8/9 Linux ...	CPU: 16 core RAM: 16 GB HDD: 500GB	nginx	Mở port 22, 80	- Mở mạng đến repository của Redhat hoặc mount cd/file iso cài đặt HĐH khi cài đặt. - Khi cài đặt cần user có quyền root để cài đặt - Khi cài xong thì có thể ngắt kết nối internet
2	Máy chủ CSDL	Redhat 8/9 Linux ...	CPU: 16 core RAM: 16 GB SSD: 250 GB - hoặc trên SAN	postgresql-server, postgresql-contrib hoặc CSDL Oracle	Nếu là Postgres: 22 và 5432 Nếu là Oracle: 1521	Note: Có thể sử dụng hệ thống CSDL Oracle có sẵn, hoặc cài đặt chung với Server LLQ QA Platform
2	Agent Web (Máy client của Kiểm thử viên thực hiện kiểm thử web)	Windows 10/11	CPU: 4 core RAM: 8GB HDD: 50GB	web browser mới nhất	Thông port 80 tới LLQ QA Platform Server	
3	Agent Mobile (Máy client của Kiểm thử viên thực hiện kiểm thử ứng dụng mobile iOS)	MacOs	CPU: 4 core RAM: 8GB HDD: 50GB	Nodejs, Jdk 17, xcode, android studio	Thông port 80 tới LLQ QA Platform Server	- Phiên bản XCode và HĐH mới nhất

Phương pháp Automation Testing

Phương pháp Automation Testing




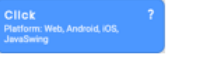
• **LLQ QA Platform** là sự kết hợp của các phương pháp sau:

- Module based
- Library architecture
- Keyword-driven
- Data-driven



Keyword-driven Testing

Lowcode: Block based programming

Test Scenario ID	s_login_001	Test Scenario Description	Test the Login Functionality in Banking			
Created By	quannm - Tester	Reviewed By	Quyñh - Testleader	Version	2.1	
Steps	Description	Keyword	Locator	Datasource	Data	Data 1
step1	Đi đến trang login của ứng dụng cần kiểm thử	 Go to Url		enviroment variable		
step2	Nhập username	 Clear and Send Keys	Element Repository " username " //input[@id='username']	parameter	linhbm	quynhnd
step3	Nhập password	 Clear and Send Keys	Element Repository " password " //input[@id='password']	parameter	*****	*****
step4	Bấm nút đăng nhập	 Click	Element Repository " Nút Đăng nhập " //button[contains(text(),'Đăng nhập')]			

1 **Go To Url (url)**
Truy cập form login

2 **Clear And Send Keys (Đăng nhập / username # by_formcontrolname, username)**
Nhập username

3 **Clear And Send Keys (Đăng nhập / password # by_type, password)**
Nhập mật khẩu

4 **Click (Đăng nhập / Đăng nhập # by_text)**
Kích nút Đăng nhập

KEYWORDS

HISTORY

Search a keyword

All (133)

Client (1)

Action (69)

Control (9)

Variable (18)

Operator (21)

File (12)

Other (3)

Action Call
Platform:Web, Android, iOS, Windows, JavaSwing

Click
Platform:Web, Android, iOS, Windows, JavaSwing

Child Element Click
Platform:Web, Android, iOS, Windows, JavaSwing

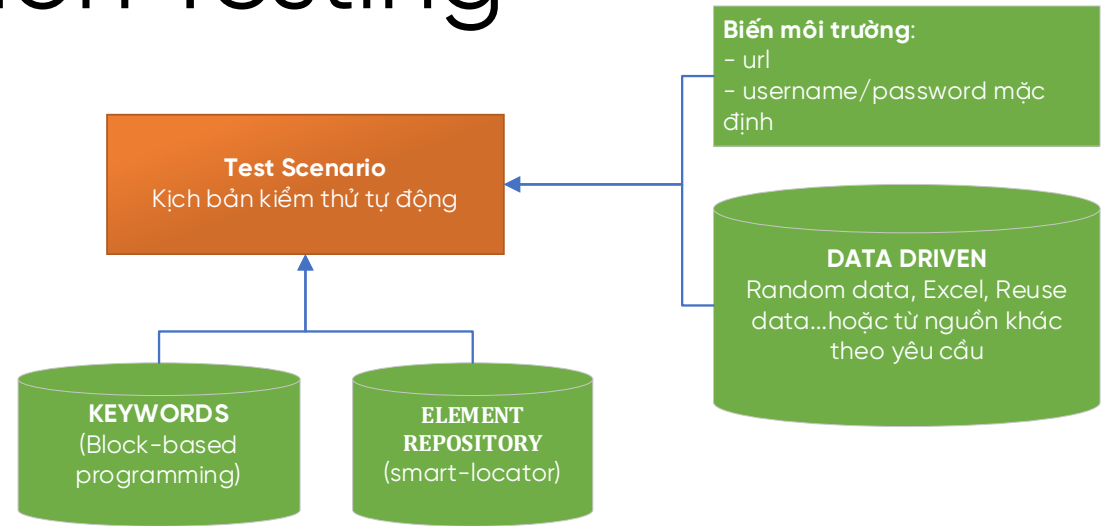
Clear And Send Keys
Platform:Web, Android, iOS, Windows, JavaSwing





Child Element Clear And Send Keys
Platform:Web, Android, iOS, Windows, JavaSwing

Send Key From Element
Platform:Web, Android, iOS, Windows, JavaSwing





Phương pháp Automation Testing

- Khi kết hợp 02 phương pháp phổ biến:
 - Keyword-driven Testing
 - Data-driven Testing



Test Scenario ID	s_login_001	Test Scenario Description	Test the Login Functionality in Banking			
Created By	quannm - Tester	Reviewed By	Quyñh - Testleader	Version	2.1	
Steps	Description	Keyword	Locator	Datasource	Data	Data 1
step1	Đi đến trang login của ứng dụng cần kiểm thử	 Go to Url		enviroment variable		
step2	Nhập username	 Clear and Send Keys	Element Repository " username " //input[@id='username']	parameter	linhbm	quynhnd
step3	Nhập password	 Clear and Send Keys	Element Repository " password " //input[@id='password']	parameter	*****	*****
step4	Bấm nút đăng nhập	 Click	Element Repository " Nút Đăng nhập " //button[contains(text),'Đăng nhập']			

Phương pháp Automation Testing

Test Scenario ID	s_login_001	Test Scenario Description	Test the Login Functionality in Banking			
Created By	quannm - Tester	Reviewed By	Quỳnh - Testleader	Version	2.1	
Steps	Description	Keyword	Locator	Datasource	Data	Data 1
step1	Đi đến trang login của ứng dụng cần kiểm thử	 Go to Url		enviroment variable		
step2	Nhập username	 Clear and Send Keys	Element Repository "username" //input[@id='username']	parameter	linhbm	quynhnd
step3	Nhập password	 Clear and Send Keys	Element Repository "password" //input[@id='password']	parameter	*****	*****
step4	Bấm nút đăng nhập	 Click	Element Repository "Nút Đăng nhập" //button[contains(text(),'Đăng nhập')]			

STT	Testcase code	param: username	param: password	Test Scenario code	Mục đích kiểm thử (với từng dữ liệu hay testcase)	Kết quả đầu ra mong muốn(Theo Tester)	Mức độ nghiêm trọng (Của lỗi)	Độ ưu tiên (Fix bug của DEV)
1	TC_Login_001		Abc@123	s_login_001	Mục đích kiểm thử với tập dữ liệu	F	Critical	High
2	TC_Login_002	linhbm	Abc@123		Mục đích kiểm thử với tập dữ liệu	F	Medium	Medium
3	TC_Login_003	quynnd			Mục đích kiểm thử với tập dữ liệu	F	Critical	High
4	TC_Login_004	linhbm	Abc@1234		Mục đích kiểm thử với tập dữ liệu	T	Low	Low

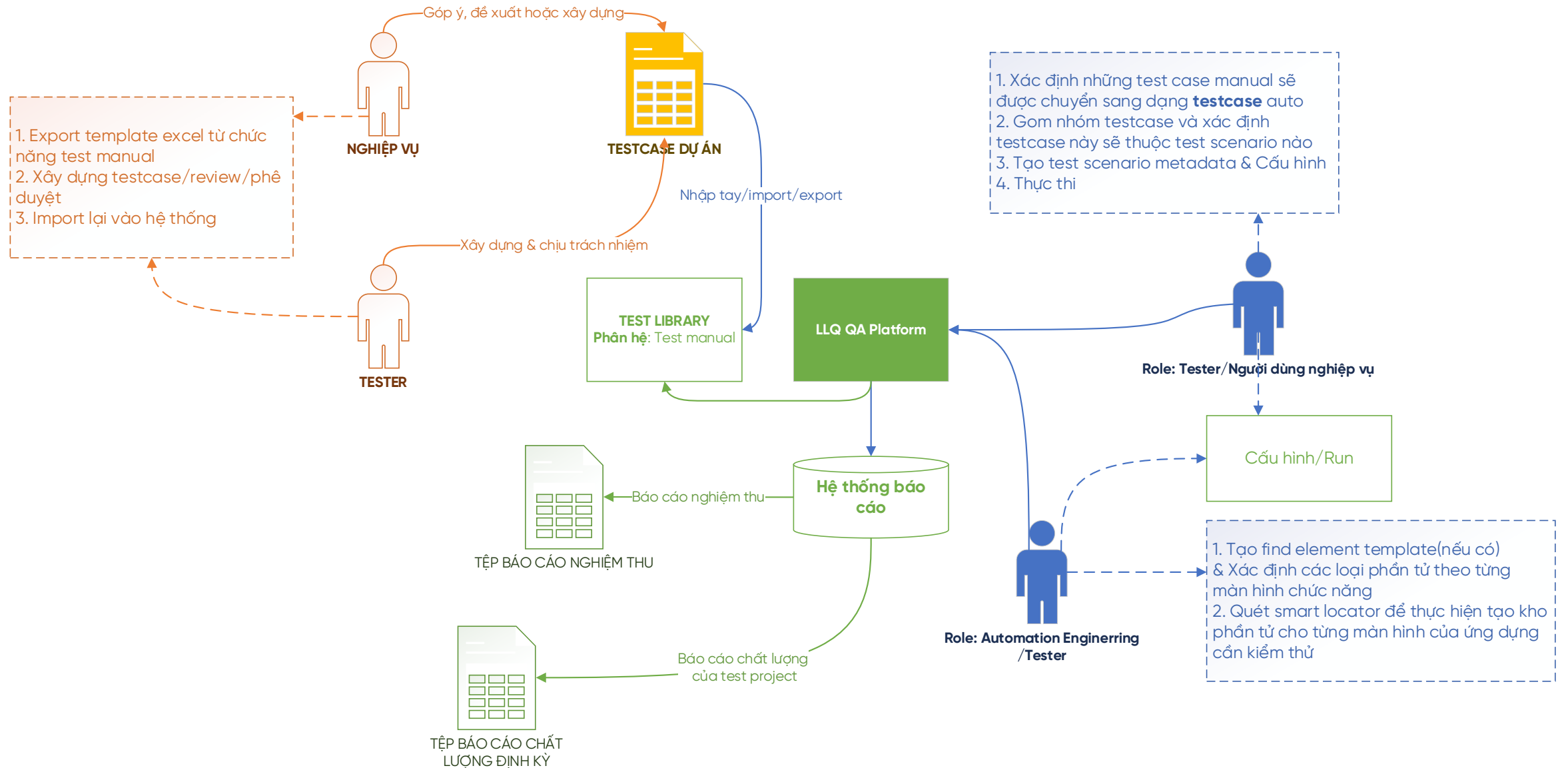
Tổ chức triển khai

Vai trò tham gia hệ thống

Triển khai hệ thống LLQ QA Platform không làm thay đổi cơ cấu tổ chức, không làm tăng thêm nhân sự hay tăng thêm vai trò trong tổ chức.

- Role Test Manager: Chịu trách nhiệm quản lý toàn bộ quá trình kiểm thử, bao gồm lập kế hoạch, phân công nhiệm vụ, theo dõi tiến độ và đảm bảo chất lượng sản phẩm.
- Role Test Automation Engineer
 - Tạo find element template(nếu có) & xác định các loại phần tử.
 - Quét smart locator để thực hiện tạo kho element repository theo từng màn hình chức năng.
- Role Tester, Người dùng nghiệp vụ
 - Xây dựng, phân loại testcase/test scenario/test suite và các tệp batch(smoke, sanity, regression hoặc checklist online...)
 - Thực thi & báo cáo

Quy trình triển khai





Giới thiệu Dịch vụ kiểm thử

01 — CÁC DỊCH VỤ KIỂM THỬ



Ứng dụng web



API



Tự động



Ứng dụng di động



Batch



Chuyển đổi dữ liệu



Ứng dụng windows



Web service



Nghệ vụ khác

03 — CÁC GIAI ĐOẠN KIỂM THỬ

- Kiểm thử tích hợp
- Kiểm thử hệ thống
- Kiểm thử chấp nhận
- Kiểm thử hiệu năng

02 — QUY TRÌNH KIỂM THỬ



04 CÔNG CỤ VÀ MÔI TRƯỜNG KIỂM THỬ

Môi trường kiểm thử

Thiết bị:



PC



Macbook



Tablet



Smartphone



Virtual machine

Hệ điều hành:



Trình duyệt:



Công cụ kiểm thử & Quản lý

Công cụ được xây dựng bởi LLQ:

LLQ QA PLATFORM

Công cụ khác hoặc được chỉ định theo yêu cầu



ZABBIX

05 — QUY TRÌNH CHẤT LƯỢNG



Đề xuất & Đưa ra quy trình



Cung cấp tài liệu, biểu mẫu & hướng dẫn



Kiểm tra, đánh giá việc thực hiện



Nhắc nhở tuân thủ quy trình công việc



Điều chỉnh & thay đổi quy trình phù hợp

100 % NHÂN SỰ LLQ ĐƯỢC ĐÀO TẠO CÁC KHÓA NỘI BỘ



ĐÀO TẠO KIỂM THỬ TỰ ĐỘNG KHÁCH HÀNG

- Đối tượng**
Người có ít hoặc chưa có kinh nghiệm về Automation Testing
- Mục tiêu**
 - Hiểu được các khái niệm về Automation Testing
 - Sử dụng thành thạo công cụ LLQ QA Platform
 - Thực hiện cấu hình được kịch bản kiểm thử tự động
- Yêu cầu**
Không có yêu cầu cụ thể
- Thời lượng**
48 tiếng

DEMO HỆ THỐNG

