

Test Strategy

Description	QA Strategy
Owner	
Stakeholders	Primarily Engineering & Product
Informed	
Status	PUBLISHED
Version	1.0.0

How to use this document

This document has all the Org-level testing-related items mentioned. Based on this strategy document, all future test plans, and test cycles will be guided. Since we are following Agile methodology, and all sprints cannot be accounted for in a Test Strategy due to time constraints, a high-level Test Strategy provides a guideline for the testing team. Also, this is of the main documents frequently requested by Enterprise customers, ISO auditors, etc.

Purpose

The purpose of a Test Strategy is to create an understanding of the overall targets, approach, tools, and timing of test activities to be done. Rarely does a project succeed without a well-designed Test Strategy. The possibility of missing any test activity is very low when there is a proper Test Strategy in place. The key to a Test Strategy is to maintain flexibility in the complete process and at the same time not change the dynamics of the testing activity. This helps to achieve the highest possible quality, especially with rapidly evolving/changing environments.

Test Strategy should help the testing/development team define the approach to test the product: *product features, product parts, or product-related integrations* and should never be confused with a Test Plan.

Application Scope

- [Portal](#)
- [P/E - Player / Editor](#)
- [Control Centre](#)
- Intent Pages
- Academy
- [Userlane Website](#)
- [Userlane Browser Extension](#)
 - [Chrome Extension](#)
 - [Firefox Extension](#)
- [Extension SSO](#)

Integrations and Intersystem Interfaces

- [Search](#)
- [Auth Keycloak](#)
- [Mothership](#)
- [Analytics](#)
- Connected Integration - [Confluence](#), [Zendesk](#), [KnowledgeOwl](#)

Test Environments

- [Environment Regions supported](#) - EU and US
- [Functional Test Env](#)
- [Automated Test Env](#)
- [Multi-staging Test Env](#)
- [Operating Systems supported](#)
 - Win 10 OS
 - macOS
- [Browsers supported](#)
 - Chrome
 - Firefox
 - Edge
- [Userlane types supported](#)
 - [Sandbox](#)
 - [Snippet](#)
 - [Self Hosting](#)
 - [Extension](#)
 - [SSO](#)
- [Major client applications supported](#)

- SalesForce
- MSDynamics
- WorkDay

Test Data

We have a [database seeder](#) implemented to seed our Postgres database for generating a working environment with data needed to run the backend API. We are now using [migrations](#) to initialize the database.

For functional testing, we create test data manually as per the requirement and maintain a [record](#) of it during the entire test cycle.

Test Users

[From Userlane perspective](#)

- Users
- PropertyUsers
- CompanyUsers

[From the user roles perspective](#)

- Admin
- Moderator
- Contributor

Testing Responsibilities

- **Early test planning.** Pushing testing procedures off until the last week can create bottlenecks and slow down progress.
- **Reviewing requirements.** Engaging in communication with the product and other stakeholders to review and analyze requirements early.
- **Testing often.** Doing smaller tests (**less manual and more automated ones**) more frequently throughout the development stages and creating a continuous feedback flow allows for immediate validation and improvement of the system.
- **Focus on Automation.** Implementing automated tests whenever possible and maximizing test coverage would also expedite and improve the testing process.
- **Prevention instead of reaction.** Pairing with developers to write autotests or joining discussion sessions and providing rapid feedback to improve the development process.

Testing Types

- Automated Testing
 - Unit Testing
 - Integration Testing
 - Regression Testing
 - System Testing
- API Testing
- Accessibility Testing
- Analytics Testing
- Backend Testing
- Compatibility Testing
- Functional Testing
- Microservices Testing
- Non-Functional Testing - Load, Performance, etc.
- Sanity Testing
- Smoke Testing
- UI/UX Validation

Testing Tools

- [Atlassian Jira/AIO Addon](#)
- [Browserstack](#)
- [Cypress](#)
- [Currents](#)
- [Dashlane](#)
- [Figma](#)
- [JAWS](#)
- [K6 \(formerly LoadImpact\)](#)
- [Miro](#)
- [Metabase](#)
- [Postman](#)
- [Redash](#)
- [Kafka-UI](#)

Defect Management

At the Userlane defect, the management cycle is performed in the following stages.

- Discovery of Defect,
- Defect Categorization
- Fixing of Defect by developers
- Verification of the fix by Testers,
- Defect Closure
- Defect Reports at the end of the project

For defects found in production, we perform a [Root Cause Analysis](#) (only for Highest priority bugs) and follow [this](#) Bug Process in general for all the production issues.

Results & Reporting

- Communication/Escalation - All info to be shared with the team members, Leads, and Management via Slack, Zoom, Userlane Email
- Daily Status Report - DSR is mandatory for the projects when:
 - There is no build/ delay in the build Ready for QA.
 - Timelines are not met and there is a risk of milestone slippage.
 - Open bugs reported by QA lead to testing blockage or when no testing can be performed due to open bugs.
 - Testing is on Hold due to Environment issues
 - The responsible engineer is out of the office and testing can't be carried out.

In all of the above-mentioned cases, the entire squad should be informed via Slack message that "TESTING IS ON HOLD" and the same status should be highlighted by the QA

_____ in their respective squad dailies. In all other cases, just update the QA status in your daily standup.

- [Test Plan](#)
- Test Execution Report - To be generated from AIO Reports.
- [QA Sign-off](#)

Testing Limitations

- Exhaustive (total) testing is practically impossible with the given time and budget constraints.
- Compromise between thoroughness and budget.
- Not every path can be covered.
- Not every edge case can be tested.
- Not every client application can be tested.
- Loss of personnel (important technical staff members leave, and there isn't enough time to replace them);
- Anticipated unknown risks i.e. circumstances of which the QA team is generally aware, but *unaware* as to whether the risk will show up in a specific test project or procedure.

References

- Get an impression of our **competitors** or other pioneers in the DAP market [Competitors](#)
- For access to major client applications for testing please check with _____
- To understand the states of Jira tickets (including Bugs) please refer - [Jira workflow statuses, definition of done JIRA Workflow for Testing](#)