

Test Plan: Veriff Demo

Disclaimers / Assumptions:

The idea of providing a test plan is rather not to have a complete document for application, but more to show the way of thinking and understanding of the test approaches. Hence, I would like to provide more test case ideas/checklists. Please let me know if you need a full description of each / particular test case.

Scope:

- **Session Language** should be chosen randomly, so in this case, each test execution would test a new test data set. **But** in case we have some business important languages, they should be tested every time as separate test cases.
- **Document country** should be chosen randomly, so in this case, each test execution would test a new test data set. **But** in case we have some business important countries, they should be tested every time as separate test cases.
- All **Document type** should be tested. For automation purposes, parametrization could be used.
- Other elements of the application should be verified with regular approaches.
- In case, it would be a much more number of fields/elements, a pairwise testing approach could be used

Test case ideas for Veriff Demo application:

Functional test cases	Create a new session with valid Full Name, ANY Session Language, ANY Document country, Document type (Check all of them one by one), Launch Veriff via (Check both of InContext and Redirect options). Check language of the session after creation.
	Check mandatory validation for fields - try to create a new session without filling one of the fields (Full Name, Session Language, Document country, Document type). Check error messages.
	Check autocomplete module for Document country
	Check that number of options for Session Language, Document country, Document type is valid (all necessary options are in place)
	Check max length + special characters for the input fields
	Check links on the pages of creating session

		Check text boxes for information are representing correct information, logo, placeholders
		Check page view is aligning with a design created
		Check ability to work with app with keyboard only
		<p>Check API for same cases as abovementioned + For API test cases should be checked:</p> <ul style="list-style-type: none"> - status code - error message in case of negative test - check the correctness of saving data (better to fetch data via API than directly in DB if DB structure is complicated) <p>Optional could be added check for response time for every request.</p>
Non-Functional cases	test	Test app against WCAG criteria. aXe tool could be used.
		Performance testing for big amount of users using the system simultaneously, like increasing the number of sessions. Locust tool could be used.
		Time for recovery in case application is crashed, like timing for k8s restart containers, DB has deadlock etc.
		Data consistency testing, if an app has a couple of kafka topics for example, or in case of failure or data replication
		Localization testing