Test Plan

Introduction

This document is intended to give a description of all the tests that are NOT covered in the developed set of front-end and back-end Unit and Integration tests.

Tests description

Manual test the UI

Tests for application aspect and components visualization are difficult to automate, for example, with automated tests is impossible to detect if a graphical element is rendered properly or if it is covered by other elements. For this reason, it is required to perform manual tests to assure that the components are visualized correctly and in the expected positions.

UI Tests: Login Page

Test that the login form and its component are visualized properly: e.g. components are not hidden or covered with other graphical elements
Test that input text boxes (email and password inputs) are clickable
Test that the user is able to insert texts inside email and password inputs
The login form has a responsive layout: on desktop, the “email” and “password fields” are horizontally aligned, while on mobile they are vertically aligned, both with the submit button underneath them

Different devices and web browsers can render the graphical components in different ways. It is mandatory to run the aforementioned tests on different devices (desktop and smartphone), with different web browsers and display resolutions.

Manual test the UX

It is necessary to perform usability testing. We should ask the end user to test the application in order to understand whether he is able to understand how to properly use the application and application flow.

Stress tests

Tests the load balance of the back-end to see if it can handle multiple requests per time, and do not have any problem of concurrency or race conditions that can lead to a denial of service.

Security Testing

Tests that must be run in order to assess that the application it is not effected by the most severe security vulnerabilities. In particular:

XSS	Assess that user inputs are sanitized correctly to avoid reflected and stored cross-site-scripting vulnerabilities
SQL Injection	Assess that user inputs are sanitized correctly and SQL query are performed using prepared-statements
Missing access control	Ensure that an user can't access content he is not authorized to
Broken Authentication	The authentication protocol is not implemented properly and allows an attacker to impersonate another user

Other vulnerabilities to test are described in the documents:

- OWASP API Security TOP 10 2019

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project

- OWASP Top 10 2017

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project