

Test and strategy plan agile

October 10, 2023

Introduction

The objective is to create a test plan and strategies for the evaluation of this technical challenge. This challenge consists of testing the login functionality and the display of a list of employees.

Resource

Tester	% Participation
Nadia Viltes	100%

Scope

1. Validate that the user's login is working correctly.
2. Validate that the list of employees has the necessary information to identify their city of origin and their position in the organization.
3. Ensure security of sensitive data for authentication and employee data.

Out of Scope

1. Extensive testing of their functionalities or features of the application will not be included. For example: Create the new user, reset password, etc.
2. Other functionalities and features of the application are outside the scope of this test plan.
3. API or database testing will not be validated.

Test environment

Environment: Local

Supported Browsers: Google Chrome, Mozilla Firefox, Microsoft Edge.

Devices: PC (web).

Framework: Cypress. Javascripts. Cucumber.

Design pattern: POM (page model object).

Test strategy

1. Manual testing:

- Types test:
 - Functional Test (**)
 - UI test
 - Integration test
 - System test
 - Smoke test
 - Sanity test
 - Exploratory test
 - Non functional Test (**)
 - Security
 - Stress
 - Load
 - Test related to change
 - Regression test
 - Confirmation test (Re testing)
 - Acceptance test
 - Alpha
 - Beta

2. Automation testing:

- Types test:
 - E2E (Integration and system test)
 - Test related to change
 - Regression test

(**) Some types of test have been included simulating that you have an API connection and that the functionalities will reach the production stage.

Methodology for Writing Acceptance Criteria and Scenarios

I will apply the Behavior Driven Development (BDD) methodology to write and document the acceptance criteria and test scenarios in this project.

Writing Process:

Acceptance criteria and scenarios will be specified using natural language, following the BDD format, which includes the Given-When-Then structure.

Test Automation:

Test scenarios written in natural language will be converted to automated testing using the BDD: Cucumber library, where applicable.

Test artifacts

Test design

Test results

Defect reports