

# **Test Plan**

## **Playvera, sem3**

This page is left blank on purpose

## What kinds of testing are there in software?

### Unit tests

Unit tests are very low level, close to the source of your application. They consist in testing individual methods and functions of the classes, components or modules used by your software. Unit tests are in general quite cheap to automate and can be run very quickly by a continuous integration server.

### Integration tests

Integration tests verify that different modules or services used by your application work well together. For example, it can be testing the interaction with the database or making sure that microservices work together as expected. These types of tests are more expensive to run as they require multiple parts of the application to be up and running.

### Functional tests

Functional tests focus on the business requirements of an application. They only verify the output of an action and do not check the intermediate states of the system when performing that action.

There is sometimes a confusion between integration tests and functional tests as they both require multiple components to interact with each other. The difference is that an integration test may simply verify that you can query the database while a functional test would expect to get a specific value from the database as defined by the product requirements.

### End-to-end tests

End-to-end testing replicates a user behavior with the software in a complete application environment. It verifies that various user flows work as expected and can be as simple as loading a web page or logging in or much more complex scenarios verifying email notifications, online payments, etc...

End-to-end tests are very useful, but they're expensive to perform and can be hard to maintain when they're automated. It is recommended to have a few key end-to-end tests and rely more on lower level types of testing (unit and integration tests) to be able to quickly identify breaking changes.

### Acceptance testing

Acceptance tests are formal tests executed to verify if a system satisfies its business requirements. They require the entire application to be up and running and focus on replicating user behaviors. But they can also go further and measure the performance of the system and reject changes if certain goals are not met.

### Performance testing

Performance tests check the behaviors of the system when it is under significant load. These tests are non-functional and can have the various form to understand the reliability, stability, and availability of the platform. For instance, it can be observing response times when executing a high number of requests, or seeing how the system behaves with a significant of data.

Performance tests are by their nature quite costly to implement and run, but they can help you understand if new changes are going to degrade your system.

## Smoke testing

Smoke tests are basic tests that check basic functionality of the application. They are meant to be quick to execute, and their goal is to give you the assurance that the major features of your system are working as expected.

Smoke tests can be useful right after a new build is made to decide whether or not you can run more expensive tests, or right after a deployment to make sure that the application is running properly in the newly deployed environment.



Project Name	Playvera	Browser:
Test Case ID	129F	Version:
Written By:	Ivan Ovcharov	Description:
Tested By:	Ivan Ovcharov	Tested On:

Test #	Date	Action	Expected Results
1		Logging in	Should get to home screen
2		Logging in	Should be sent to home screen
3		Register	Should be sent to home screen
4		Register	Should be sent to home screen
5		Retrieve all users	All users should be displayed
6		Find a user by username	Should show info of user
7		Change user password	Should display successful change
8		Viewing all available games	Should display all games available
9		Searching for a game by name	Should show specific game
10		Searching for games under 10 euros	Should display games under 10
11		Searching for games under 20 euros	Should display games under 20
12			
13			
14			
15			