CSGOEmpire

Roulette Assignment

Juan Gómez Hermoso
30-03-2024

Contenido

Introduction

This test plan outlines the comprehensive approach for testing a digital roulette game. Roulette is a classic and widely recognized casino game known for its simplicity and excitement. The objective of this test plan is to ensure the quality, functionality, and reliability of the digital implementation of the roulette game. The game can be found on CSGOEmpire website.

Objective

The primary objective of this test plan is to verify that the digital roulette game functions correctly according to its specifications, providing players with an authentic and enjoyable gaming experience.

Testing approach

The testing approach will involve a combination of manual and automated testing techniques to thoroughly assess the digital roulette game. Test cases will be designed to cover different scenarios.

Manual Testing

- Manual testing will be conducted to validate game mechanics, including roulette wheel spin accuracy, betting options, and payout calculations.
- Usability testing will be performed to assess the clarity, responsiveness, and intuitiveness of the user interface elements.
- Compatibility testing will involve testing the game on different devices, operating systems, and web browsers to ensure consistent performance.
- Security testing will be conducted to verify the integrity and security of player data and communication protocols.

Automated Testing

- Automated testing will be used to execute repeatable test cases efficiently and detect any regressions introduced during development.
- Test automation tools such as Selenium WebDriver will be utilized for automated testing of web-based components.
- Test scripts will cover a range of scenarios, including normal gameplay, edge cases, error conditions, and boundary conditions.

Test Cases

Test cases would be stored in our preferable software, as TestRail. As for now, will be documented below:

Id:	Description:	Steps:
1	Access to CSGOEmpire Roulette	1- Open csgoempire.com/roulette
	Expected Result:	
	User is in the roulette game	
Id:	Description:	Steps:
2	Enter an amount to bet	1- Open csgoempire.com/roulette 2- Click on the "Enter bet amount…" box 3- Select a betting amount (ex: +10)
	Expected Result:	
	Amount is inserted for betting	
Id:	Description:	Steps:
3	Possible to clear the bet amount	1- Open csgoempire.com/roulette 2- Click on the "Enter bet amount…" box 3- Enter a betting amount 4- Pless in "Clear" button
	Expected Result:	
	Amount inserted is cleared	
Id:	Description:	Steps:
4	You can just enter numbers on bets	1- Open csgoempire.com/roulette 2- Click on the "Enter bet amount…" box 3- Write a word in the betting amount
	Expected Result:	
	Red box is shown on the betting and it is not possible to bet	
Id:	Description:	Steps:
5	When rolling time reaches 0, roulette starts	1- Open csgoempire.com/roulette 2- Wait until rolling time reaches 0
	Expected Result:	
	Roulette will start and will be a winner bet	
Id:	Description:	Steps:
6	User can enter amounts by pressing on the boxes	1- Open csgoempire.com/roulette 2- Click on the "Enter bet amount…" box 3- Press in the amount's boxes (+0.01, +0.1, +1…)
	Expected Result:	
	Betting amount is inserted	
Id:	Description:	Steps:
7	If user wins, money is added to your wallet	1- Open csgoempire.com/roulette 2- Click on the "Enter bet amount…" box 3- Enter a betting amount

		4- Click in a place to bet from the 3 options 5- Roulette starts and you are the winner
	Expected Result:	
	Betting money is added to your wallet (2x, 14x…)	
Id:	Description:	Steps:
8	If user loses, money is deducted from your wallet	1- Open csgoempire.com/roulette 2- Click on the "Enter bet amount…" box 3- Enter a betting amount 4- Click in a place to bet from the 3 options 5- Roulette starts and you are not the winner
	Expected Result:	
	Money that you bet is deducted from your wallet	
Id:	Description:	Steps:
9	Username appears on the bet ordered by the biggest bet	1- Open csgoempire.com/roulette 2- Click on the "Enter bet amount…" box 3- Enter a betting amount 4- Click in a place to bet from the 3 options
	Expected Result:	
	Your username will appear in your betting place, ordered by the biggest bet to smallest bet. Only the 10 first usernames will appear.	
Id:	Description:	Steps:
10	Previous rolls are shown under the roulette	1- Open csgoempire.com/roulette
	Expected Result:	
	Under the roulette, latest rolls appear on the screen as well of a resume of the latest 100 rolls	
Id:	Description:	Steps:
11	Monthly roulette race shows the list of the top winning players	1- Open csgoempire.com/roulette 2- Scroll until end of the page
	Expected Result:	
	Monthly top winning players are shown on list	
Id:	Description:	Steps:
12	If user has no money, cannot bet	1- Open csgoempire.com/roulette 2- Click on the "Enter bet amount…" box 3- Enter a betting amount 4- Click in a place to bet from the 3 options
	Expected Result:	
	User cannot play roulette until he doesn't add money	

Automation

The best automation tool for the in this case is through Selenium WebDriver, and using JAVA, as programming language.

Prerequisites

- Ensure that the digital roulette game is deployed to the testing environment.
- Install necessary dependencies, including Selenium WebDriver and any browser drivers required for test execution.

Setup

- Clone the automated test suite repository.
- Install dependencies.
- Configure test environment variables, including the URL of the deployed game and any authentication credentials if required.

Execution

- Run the test suite using the designated test runner.
- Monitor the test execution process for any failures or errors.
- Analyze test results and identify any issues or regressions.

Reporting

- Generate test reports documenting the test execution results, including pass/fail status and any error details.
- Communicate findings to the development team for further investigation and resolution.

Assumptions and Decisions

- **Assumption:** The game has been developed according to specified requirements and design specifications.
- **Decision:** Test automation will focus on critical and high-priority test cases to maximize test coverage within time constraints.