Steam Test Plan

Contents

1. Introduction

1.1 Project Overview

The targeted web application Steam is a video game digital distribution service owned by Valve it was launched as a standalone client in September 2003 as a way for valve to provide auto updates for their games and expanded to include games from third party publishers. Steam has also expanded into an online web-based and mobile digital storefront. Steam offers a freely available API called Steamworks, which developers can use to integrate many of its functions into their products.

1.2 Purpose

The purpose of this document is to check the functionality of Steam and ensure that its main functions are working as intended. This document will also specify all the steps and procedures of the testing process and will clearly identify test deliverables. Some minor features will not be tested during this phase of testing.

1.3 Test Objectives

The test objectives are to verify the functionality of steam website, and that it works correctly and according to the specifications assigned to it. The tests are carefully designed to identify and fix the problems within the features to be tested and verify that each feature is working as it is intended to. And priority levels will be assigned to each problem according to its severity.

The scope of this test plan is to test all the main functions of steam website starting by the registration and login and browsing through the store, checking different game pages and simulating the whole user experience by adding games to cart and the whole checkout/payment process and also verifying some of the profile functionalities.

The chosen approach is a mixture of automated and manual testing, automated testing will greatly increase the efficiency of the testing process and reduce the manpower and cost of the project. The tasks will be assigned to each member through the Jira app to properly handle them.

1.4 Testing Strategy

Following a good testing strategy is the key element of successful test plan, the next section will contain various methodologies for having a good test plan.

Test management assists with the development of a robust risk management process. Developing effective software requires good research management and preparation. Detailed planning and testing add value to the execution process. Developing a test environment for an online game distribution platform will be huge and expensive as it requires multiple channels. Cross-platform testing and cross-browser adaptability will be a high priority to provide a seamless and safe browsing experience for the user. Test automation will greatly help in reducing the cost of the project and will result in a faster time-to-market and will increase the efficiency of the testing process in the long run.

1.5 Scope

- Testing will be performed at several points in the life cycle of the product.
- Testing will be divided into several distinct phases, each with clearly defined objectives and goals.
- Testing process will be well defined, flexible, with the ability to change as needed.
- Testing will be focused on meeting the business objectives, software specifications, cost efficiency and quality.
- Testing activities will build upon previous stages to avoid redundancy.

- Testing environment and data will emulate production environments as much as possible.
- There will be common consistent procedures for all teams supporting testing activities.
- Test planning will be performed throughout the system development lifecycle.

2. Features to Be Tested.

- Register.
- Login.
- Profile Info.
- Edit Profile Functionality.
- Change Profile Background.
- Change Profile Theme.
- Game Details Page.
- Add Game to Wishlist.
- Follow/Ignore Game.
- Add Game to Cart.
- Remove Item from Cart.
- Purchase Game.
- Purchase Game as a Gift.
- Payment Method/Process.
- Add Funds to Steam Wallet.
- Refund Game.
- Delete Game from Profile.
- Search for a Game.
- Sell Item on Community Market.
- Buy Item from Community Market.

3. Features not to Be Tested.

- Chat.
- Community Page.
- Workshop.
- Broadcasts.
- Discussion.
- Searching for Friends.
- Adding Friends.
- Groups.
- News.
- Points Shop.

4. Approach

This section will describe the overall testing approach and testing types that will ensure the correct functionality of each feature and functionality of the whole software.

4.1 Unit Testing

Unit testing will be conducted to verify that the individual units of source codes are working as intended. And the goal of unit testing is to isolate each part of the program to check that it is working properly.

4.2 Integration Testing

Integration testing will be conducted by combining both the hardware and software elements until the entire system has been fully integrated. And the goal of integration testing is to ensure that the design objectives are met and that the whole system is working as intended.

4.3 Job Stream Testing

Job stream testing will be conducted to ensure that the software runs and operates as intended in the production environment.

4.4 Interface Testing

Interface testing will be conducted to ensure that the software operates efficiently and effectively with all the interface systems. And the goal of interface testing is to verify whether the communication between two different software systems is done correctly.

4.5 Security Testing

Security testing will be conducted to ensure that the system is protected against different types of cyberattacks, malicious software and unauthorized access. And to ensure that the user's data is safe and well protected.

4.6 Recovery Testing

Recovery testing will be conducted to ensure that software will work properly form a backup without any problems or loss of data. And it will be tested with local backups or cloud backups.

4.7 Performance Testing

Performance testing will be conducted to ensure that the software runs efficiently and performs to customer expectations. And performance testing also ensures that the software is responsive, stable, portable, and scalable.

4.8 Regression Testing

Regression testing will be conducted to ensure that the applied changes to the software does not affect the previously tested functions.

4.9 Load Testing

Load testing will be conducted to ensure that the software will run as intended and understand its behavior under different loads. Load testing is preformed by simulating the effect of multiple users and monitoring the system's behavior under both normal and peak conditions.

4.10 Compatibility Testing

Compatibility testing will be conducted to ensure that the software runs as intended on different platforms and different browsers to provide a seamless and safe browsing experience for the user.

4.11 Usability Testing

Usability testing will be conducted to ensure that the software is easy to use among different groups of users.

4.12 Acceptance Testing

Acceptance testing will be conducted to ensure that the software satisfies the business requirements and assess whether it is acceptable for delivery or not.

5. Pass/Fail Criteria

All core functionalities of the system should function as intended and outlined in the test cases. Critical errors should not be found, and the end user must be able register/login and browse, Wishlist, gift, refund, purchase any game he likes without facing any errors. 98% of the test cases should pass and no failed cases should be crucial to the end-user's ability to use the application.

The System should satisfy all functional and non-functional requirements. Each feature to me tested must be evaluated against its requirements. The pass/fail of a test depends on whether the system meets with all post conditions. Executed test cases will pass if they meet the specific requirements as mentioned before.

5.1 Suspension Criteria

Testing should be immediately stopped if it experiences login issues or failure in any basic **CRUD** (Create, Read, Update, Delete) actions.

5.2 Resumption Criteria

When a new version of the system is available after a suspension of testing has occurred, all previous tests should rerun to ensure that the applied changes to the software does not affect the previously tested functions.

5.3 Approval Criteria

When the test runs without any issues or critical errors or any of the mentioned suspension criteria.

6. Testing Process

6.1 Test Deliverables

The test run results should be saved in a test log upon completion and the test manager should run a report for all the completed tests.

6.2 Testing Tasks

The following requirements must be completed:

- Test plan, Test cases, Test data must be prepared and ready for use.
- Functional specifications written and delivered to the testing team.
- Preform the tests.
- Prepare test results and test summary report.

6.3 Testing Environment

The test environment must be populated with different and various types of test data that includes a variety of games with different prices and categories in order to simulate the closest possible outcome to the actual user experience which will greatly enhance the final product.

6.4 Responsibilities

The Test Manager is responsible for coordination of the project and assigning the tasks for the testers. The testers should understand the project and each tester should finish the tasks assigned to him and report if the task was successful or if he encountered any bug or errors during testing.

6.5 Staffing and Training Needs

Testing should be done by a team of at least 10 testers. And each tester should have basic knowledge of the game distribution platform.

7. Environmental Requirements

7.1 Hardware

During this Testing process we will need a High-end testing server with a powerful internet connection.

7.2 Software

During this Testing process we will need different devices with different operating systems for cross-platform testing and different browsers to ensure that the software is working properly on different conditions.

7.3 Tools

- Pycharm or any IDE that supports the Selenium software.
- JIRA to assign the tasks.