

STEAM API & UI AUTOMATION TESTING

SOFTWARE TEST PLAN

Document Change History

Version Number	Date	Contributor	Description
V1.0	26.3.2024	Jameel Nassar	

Table of Contents

1	INTRODUCTION	3
1.1	OVERVIEW.....	3
1.2	SCOPE.....	4
1.2.1	<i>In Scope</i>	<i>4</i>
1.2.2	<i>Out of Scope.....</i>	<i>4</i>
1.2.3	<i>Primary Objective.....</i>	<i>4</i>
1.2.4	<i>Secondary Objective</i>	<i>4</i>
1.3	ROLES AND RESPONSIBILITIES.....	5
1.3.1	<i>QA Developer</i>	<i>5</i>
1.3.2	<i>QA Test Team Lead.....</i>	<i>5</i>
1.3.3	<i>Testing Process Management Team</i>	<i>5</i>
1.4	ASSUMPTIONS FOR TEST EXECUTION	6
1.5	CONSTRAINTS FOR TEST EXECUTION	6
2	TEST METHODOLOGY	7
2.1	PURPOSE.....	7
2.1.1	<i>Overview.....</i>	<i>7</i>
2.1.2	<i>Usability Testing.....</i>	<i>7</i>
2.1.3	<i>Unit Testing (Multiple).....</i>	<i>7</i>
2.1.4	<i>Iteration/Regression Testing.....</i>	<i>8</i>
3	UI TEST TREE.....	8
3.1	FUNCTIONAL TESTS:	8
3.1.1	<i>Login Page.....</i>	<i>8</i>
3.1.1.1	<i>Valid Login</i>	<i>8</i>
3.1.1.2	<i>Invalid Login.....</i>	<i>8</i>
3.1.1.3	<i>Restore password</i>	<i>8</i>
3.1.2	<i>Registration Page</i>	<i>8</i>
3.1.2.1	<i>Valid Registration</i>	<i>8</i>
3.1.2.2	<i>Invalid Registration:.....</i>	<i>8</i>
3.1.3	<i>Review Section.....</i>	<i>8</i>
3.1.4	<i>Search Page.....</i>	<i>9</i>
3.1.5	<i>Sorting Functionality.....</i>	<i>9</i>
3.1.6	<i>Filters Functionality</i>	<i>9</i>
3.1.7	<i>Cart Page.....</i>	<i>9</i>
3.1.8	<i>Wishlist Page</i>	<i>9</i>
3.2	NON-FUNCTIONAL TESTS:	9
3.2.1	<i>Compatibility Different browsers</i>	<i>9</i>
3.2.2	<i>Localization and Globalization.....</i>	<i>9</i>
3.2.3	<i>Recovery Testing.....</i>	<i>10</i>
4	API TEST TREE	10
4.1	FUNCTIONAL TESTS.....	10
4.1.1	<i>Search API:.....</i>	<i>10</i>
4.1.2	<i>App details API.....</i>	<i>10</i>
4.1.3	<i>App Review API</i>	<i>10</i>
4.2	NON-FUNCTIONAL TESTS.....	10
4.2.1	<i>App details API.....</i>	<i>10</i>
4.2.2	<i>App Review API</i>	<i>10</i>
4.3	ENTRY AND EXIT CRITERIA	11
4.3.1	<i>Entry Criteria.....</i>	<i>11</i>
4.3.2	<i>Exit Criteria</i>	<i>11</i>

5	TEST METHODOLOGY	11
5.1	VALIDATION AND DEFECT MANAGEMENT	11
5.2	DEFECT TRACKING & REPORTING.....	12
5.3	STATUS OF THE BUG/FAULT	12
5.4	TEST MANAGEMENT PROCESS.....	12
5.4.1	Risks.....	13
5.5	TEST ENVIRONMENTS.....	13
6	TEST ARTIFACTS.....	14
7	TIME TABLE.....	14

1 Introduction

This test approach document delineates the strategic framework, procedural workflows, and methodological strategies employed in orchestrating the planning, execution, and management of Steam UI & API Testing.

1.1 Overview

Steam by Valve is a digital distribution platform primarily designed for video games, though it also offers a wide range of other software and multimedia content. Launched in 2003, Steam has become one of the largest and most popular platforms for purchasing, downloading, and playing games on PC, with millions of active users worldwide. Here's an overview of Steam's key features:

1. **Digital Storefront:** Steam serves as a storefront where users can browse and purchase a vast library of games, ranging from indie titles to major AAA releases. Users can filter games by genre, popularity, release date, and various other criteria to discover new titles.
2. **Game Library Management:** Once purchased, games are added to the user's digital library, where they can be downloaded and played at any time. Steam provides features for organizing and categorizing games, making it easy for users to navigate their collections.
3. **Automatic Updates:** Steam automatically updates games in the user's library, ensuring they are always up-to-date with the latest patches, bug fixes, and new content releases. This feature helps streamline the gaming experience and ensures players have access to the most current version of their games.
4. **Community Features:** Steam offers robust community features, including forums, user reviews, and discussion boards for each game. Users can share tips, strategies, and opinions, as well as connect with other players who share similar interests.
5. **Steam Workshop:** Many games on Steam support user-generated content through the Steam Workshop. This feature allows players to create and share mods, custom maps, and other user-generated content, enhancing the longevity and replayability of supported games.
6. **Social Features:** Steam includes social features such as friends lists, chat functionality, and group creation. Users can connect with friends, join gaming communities, and coordinate multiplayer sessions directly through the Steam platform.
7. **SteamVR:** SteamVR is a component of Steam that supports virtual reality (VR) gaming. It provides access to a wide range of VR games and experiences, as well as features for managing VR hardware and settings.
8. **Steam Cloud:** Steam Cloud allows users to synchronize their game saves, settings, and other data across multiple devices. This feature ensures that users can pick up their progress from any supported device without losing their in-game achievements or progress.
9. **Sales and Discounts:** Steam regularly hosts sales events, such as the Steam Summer Sale and Steam Winter Sale, where users can purchase games at discounted prices. These sales events are highly anticipated by the gaming community and often feature significant discounts on a wide variety of titles.

Overall, Steam offers a comprehensive suite of features designed to enhance the gaming experience for PC users, making it a central hub for discovering, purchasing, and playing games, as well as connecting with other gamers around the world.

1.2 Scope

Describe the current test approach scope based on your role and project objectives.

1.2.1 In Scope

The Steam Website and API Test Plan defines the approach for unit, integration, system, regression, and Client Acceptance testing. The test scope encompasses:

- Testing of all functional, application performance, security, and use case requirements outlined in the Steam Website and API Use Case document.
- Evaluation of quality requirements and fit metrics specific to the Steam platform.
- End-to-end testing and validation of interfaces with all systems that interact with the Steam website and API.
- Functional testing of website features such as user authentication, game browsing, purchasing, and community interactions.
- Functional testing of API endpoints including app details retrieval, store search, reviews management, and currency conversions.
- Testing of the Wishlist feature, search pages, and game review section.

1.2.2 Out of Scope

The following are considered out of scope for the Steam Website and API Test Plan:

- Functional requirements testing for systems external to the Steam application ecosystem.
- Testing of Business Standard Operating Procedures (SOPs), disaster recovery, and Business Continuity Plan specific to entities beyond the Steam platform.

1.2.3 Primary Objective

A primary objective of testing the Steam application systems is to ensure that the system meets all requirements, including quality requirements (also known as Non-functional requirements), fit metrics for each quality requirement, and satisfies the defined use case scenarios. The goal is to maintain the quality of the product. At the conclusion of the project development cycle, users should find that the project has met or exceeded all their expectations as detailed in the requirements.

Any changes, additions, or deletions to the requirements document, Functional Specification, or Design Specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the capability of the test team.

1.2.4 Secondary Objective

The secondary objective of testing the Steam application systems is to identify and expose all issues and associated risks, communicate all known issues to the project team, and

ensure that all issues are addressed appropriately before release. This objective necessitates careful and methodical testing of the application to ensure that all areas of the system are scrutinized. Consequently, all issues (bugs) found are dealt with in a timely and appropriate manner.

1.3 Roles and Responsibilities

These roles and responsibilities ensure effective management and execution of the testing process for the Steam Website and API, with a focus on quality assurance and adherence to project requirements.

1.3.1 QA Developer

QA Developers are responsible for designing, implementing, and executing test cases to ensure the quality of the Steam Website and API. Responsibilities include:

- Developing test plans, test cases, and test scripts based on project requirements.
- Executing manual and automated tests to identify defects and ensure functionality meets specifications.
- Collaborating with developers to troubleshoot and resolve issues.
- Participating in code reviews to ensure testability and maintainability of code.
- Providing feedback on product quality and suggesting improvements to development processes.

1.3.2 QA Test Team Lead

The QA Test Team Lead oversees the testing process for the Steam Website and API. Responsibilities include:

- Leading the QA team in developing test strategies, plans, and procedures.
- Assigning tasks and responsibilities to QA team members.
- Coordinating with development teams to ensure testing aligns with project timelines and objectives.
- Monitoring testing progress and ensuring adherence to quality standards.
- Reporting on testing metrics, issues, and risks to project stakeholders.

1.3.3 Testing Process Management Team

A group responsible for overseeing the testing process, workflow, and quality management for the Steam Website and API. This may include representatives from Steam (Valve Corporation), third-party testing firms, and other stakeholders. Responsibilities include:

- Monitoring and managing testing integrity.
- Supporting testing activities throughout the development lifecycle.
- Coordinating activities across different teams involved in testing, including developers and QA team members.
- Ensuring that testing activities align with project objectives and timelines.

1.4 Assumptions for Test Execution

Below are the minimum assumptions that must be completed for effective test execution:

- User Acceptance testing is conducted by End-users who are familiar with the Steam platform and its functionalities.
- Test results are automatically reported using Jira for test management and issue tracking. Bugs are logged automatically through the Jira API when detected during testing.
- Detailed test reports, including test execution results and metrics, are generated using Allure software for comprehensive analysis and review.
- Testing is integrated into the Continuous Integration/Continuous Deployment (CI/CD) pipeline, with tests being executed automatically on GitHub Actions upon code changes or scheduled intervals.
- Use cases and test scripts are developed, reviewed, and approved by the QA Test Team Lead and QA Developers.
- The QA Test Team provides support and guidance to Adopters and Developers throughout the testing process, ensuring smooth execution and timely resolution of issues.
- Any major dependencies that may impact testing are communicated and addressed promptly after the testing kickoff meeting, to ensure uninterrupted testing progress.

1.5 Constraints for Test Execution

Below are the constraints for the test execution:

- Adopters should clearly understand test procedures and the process for recording defects or enhancements. The Testing Process Management Team will schedule a teleconference with Developers and Adopters to provide training and address any testing-related issues.
- Developers will receive a consolidated list of requests for test environment setup, user account creation, data sets (actual and mock data), defect lists, etc., through Jira after the initial Adopter testing kickoff meeting.
- Developers will support ongoing testing activities based on priorities identified by the Testing Process Management Team.
- Test scripts must be approved by the QA Test Team Lead prior to test execution.
- Test scripts, test environment setup, and dependencies should be addressed during the testing kickoff meeting in the presence of a Subject Matter Expert (SME), and any requests should be submitted within 3 days of the kickoff meeting.
- Developers cannot execute User Acceptance and End-to-End test scripts. After debugging, developers can conduct internal tests, but no results from those tests can be recorded or reported.
- Adopters are responsible for identifying dependencies between test scripts and submitting clear requests to set up the test environment.

These assumptions and constraints ensure clarity and efficiency in the testing process for the Steam Website and API, facilitating effective collaboration between Developers, Adopters, and the Testing Process Management Team. Additional constraints can be added as needed based on project-specific requirements and considerations.

2 Test Methodology

2.1 Purpose

2.1.1 Overview

The purpose of the Test Plan is to achieve the following objectives:

- Define comprehensive testing strategies for each area and sub-area, ensuring coverage of all functional and quality (non-functional) requirements specific to the Steam Website and API.
- Segment the Design Specification into testable areas and sub-areas, identifying areas to be tested as well as those to be omitted (not tested). This division aims to ensure clarity in testing scope and objectives.
- Establish bug-tracking procedures to effectively log, track, and manage defects identified during testing. Integration with Jira for automatic bug tracking and resolution will be utilized.
- Identify potential testing risks that may impact the successful execution of the testing phase. Risk mitigation strategies will be developed to address these identified risks.
- Determine the required resources, including personnel, tools, and environments, necessary for the execution of the testing activities. This includes access to test environments, user accounts, and necessary datasets.
- Provide a detailed testing schedule outlining the sequence of testing activities, milestones, and timelines. Integration with GitHub Actions for automated testing and continuous integration will be leveraged to streamline the testing schedule.

2.1.2 Usability Testing

The purpose of usability testing is to ensure that the new components and features of the Steam Website and API function in a manner that is acceptable to the customer. Development will typically create a non-functioning prototype of the UI components to evaluate the proposed design. Usability testing will be coordinated by the QA Test Team Lead, but actual testing must be performed by non-testers, aiming to simulate end-users as closely as possible. Testing will review the findings and provide the project team with its evaluation of the impact these changes will have on the testing process and the project as a whole.

2.1.3 Unit Testing (Multiple)

Unit Testing is conducted by the Developers during the code development process to ensure proper functionality and code coverage are achieved by each developer both during coding and in preparation for acceptance into iterations testing. The following are the example areas of the project that must be unit-tested and signed-off before being passed on to regression Testing:

- Website components.
- API endpoints and business logic.
- Database queries, stored procedures, triggers, tables, and indexes.
- Steam client functionalities.

2.1.4 Iteration/Regression Testing

During the repeated cycles of identifying bugs and receiving new builds containing bug fix code changes, several processes are common to this phase across all projects. These include various types of tests such as functionality, performance, stress, and configuration testing. Additionally, there is the process of communicating results from testing and ensuring that new drops/iterations contain stable fixes (regression).

The project should plan for a minimum of 2-3 cycles of testing (drops/iterations of new builds). At each iteration, a debriefing should be held. Specifically, the report must show that to the best degree achievable during the iteration testing phase, all identified severity 1 and severity 2 bugs have been communicated and addressed. At a minimum, all priority 1 and priority 2 bugs should be resolved prior to entering the beta phase. Important deliverables required for acceptance into Final Release testing include:

- Application SETUP.EXE
- Installation instructions
- All documentation (beta test scripts, manuals or training guides, etc.)

3 UI Test Tree

3.1 Functional tests:

3.1.1 Login Page

3.1.1.1 Valid Login

- Login with valid username and password
- Login using third-party integration

3.1.1.2 Invalid Login

- Login using nonexistent email
- Login using existing email and wrong password

3.1.1.3 Restore password

3.1.2 Registration Page

3.1.2.1 Valid Registration

3.1.2.2 Invalid Registration:

- Registration with already registered email
- Registration with already registered username
- Registration with invalid email
- Registration with invalid password
- Registration with invalid username

3.1.3 Review Section

- Filter by review date
- Filter by review playtime
- Filter by min play time
- Filter by max play time
- Filter by range

- Filter review type
- Positive review
- Negative review

3.1.4 Search Page

- Search using valid keywords
- Search using a valid product name
- Search using a valid category name
- Search using invalid keywords
- Search empty space

3.1.5 Sorting Functionality

- Sorting by price descending order
- Sorting by price increasing order

3.1.6 Filters Functionality

- Filter using price slider
- Filter by manufacturer
- Filter by Rating
- Filter by using product page specific filters.

3.1.7 Cart Page

- Add Game to cart
- Remove game from cart

3.1.8 Wishlist Page

- Add Game to Wishlist
- Remove game from Wishlist

3.2 Non-Functional tests:

3.2.1 Compatibility Different browsers

- compatible with Google Chrome
- compatible with Microsoft Edge
- compatible with Safari
- compatible with Mozilla Firefox

3.2.2 Localization and Globalization

- changing the Website's Language
- change game prices currency
- change games reviews languages

3.2.3 Recovery Testing

- Reload the page and check user is still connected
- Reload the page and selected part in pc section are saved

4 API Test Tree

4.1 Functional Tests

Functional tests validate the behavior of various endpoints provided by the Steam API. These tests cover functionalities such as retrieving app details, searching the store, managing reviews, and handling currency conversions.

4.1.1 Search API:

- Search for game
- Auto completion functionality
- Empty search
- Sorting price by ascending order
- Sorting price by descending order

4.1.2 App details API

- Get game details

4.1.3 App Review API

- Change default review numbers
- Filter by playtime
 - o Filter by max playtime
 - o Filter by min playtime
 - o Filter by playtime in range

4.2 Non-Functional Tests

Non-functional tests focus on aspects such as performance, compatibility, localization, globalization, and recovery of the Steam API.

4.2.1 App details API

- Get game details
- Change app price currency
 - o ILS
 - o USD
 - o GBP

4.2.2 App Review API

- Change default review numbers
- Filter By review language
 - o English
 - o Russian

- Filter by playtime
 - o Filter by max playtime
 - o Filter by min playtime
 - o Filter by playtime in range

4.3 Entry and Exit Criteria

4.3.1 Entry Criteria

- Passing the sanity tests
- The requirement document should be available.
- Complete understanding of the application flow is required.
- The Test Plan Document should be ready.
- Test case/scripts are available.
- Test environment is ready.

4.3.2 Exit Criteria

- No critical test fail
- No high test fail
- No more than 3 medium test fail
- No more than 5 Low test fail
- The software meets all functional and non-functional requirements
- Approval from all relevant Product Manager has been obtained
- All defects and issues identified during testing have been resolved
- Documentation is complete and up-to-date

5 Test Methodology

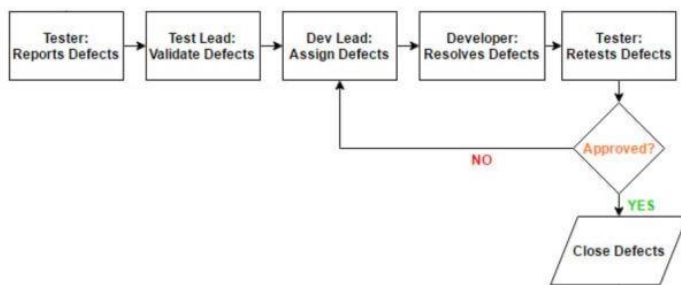
5.1 Validation and Defect Management

Defects found during the Testing will be categorized according to the bug-reporting tool “Jira” and the categories are:

Severity	Description
Critical	Bugs has the potential to crash the system
High	Bug significantly impacts program functionality
Medium	Bug prevents testing in specific areas of the product.
Low	Bug has minimal impact on product use .

Priority	Description
Critical	Bug needs immediate attention due to its severe impact.
High	Bug should be addressed promptly to restore functionality
Medium	Bug requires attention but can be managed within schedule .
Low	Bug can be deferred as it has minimal impact on workflow .

5.2 Defect tracking & Reporting



5.3 Status of the Bug/Fault

- New: Just created bug
- Open: Opened bug and still not solved
- Rejected: Dev Lead rejected it (it's not bug)
- Fixed: SOLVED bug
- Closed: After bug is fixed, need close this bug
- Re-Open: Bug has been fixed and closed but appear again.

5.4 Test Management Process

	Tool	Comments
Test Management	Test Tail	Test Cases Design
Bug Tracking	Jira	Report bugs

5.4.1 Risks

#	Risk	Impact	Trigger	Mitigation plan
1	Inadequate or unclear requirements.	High	can lead to incomplete test coverage.	Collaborate closely with Product managers to ensure comprehensive and clear requirements. Conduct regular reviews and validations.
2	Insufficient testing resources (human, hardware, or software).	Medium	may lead to incomplete testing.	Plan resource needs in advance, prioritize testing tasks, and consider automation to optimize resource usage.
3	Testing in an environment that does not replicate the production environment.	High	lead to false positives or negatives.	ensure the test environment mirrors the production environment as closely as possible.
4	Poor communication among team members and Product Manager	medium	leading to ineffective testing.	Establish clear communication channels, conduct regular meetings, and document all communications to ensure everyone is on the same page.
5	Inadequate or unrealistic test data	medium	result in incomplete testing scenarios.	Develop a comprehensive test data strategy

5.5 Test Environments

During the testing phase of the PC Part Picker website, multiple test environments are utilized to facilitate thorough evaluation of the system across various configurations. These environments are meticulously set up to mimic the production environment as closely as possible while providing the necessary flexibility for testing activities. The following test environments are utilized:

- CI/CD Environment: GitHub Actions
- Report tool: Allure reports.
- Bug Reporting Tool: Jira
- Operating System: Windows 11 OS Build 22H2.3085
- Browser 1: Google Chrome Version 121.0.6167.140

- Browser 2: Firefox Version V 122.0
- Browser 3: Microsoft Edge Version 121.0.2277.98
- Python 3.12
- Selenium 4.18.1
- Selenium server 4.18.1

6 Test Artifacts

During the testing lifecycle of the PC Part Picker website project, various artifacts are produced to support the testing process, ensure validation of requirements, and maintain quality. These artifacts are process-driven and are essential for documenting testing activities and outcomes. The following list outlines the key test artifacts that should be routinely updated throughout the project development cycle:

1. **Software Test Plan (STP):**
Provides an overview of the testing approach, strategies, scope, resources, and schedule for the PC Part Picker website project.
2. **Requirement Traceability Matrix (RTM):**
Maps test cases back to specific requirements, ensuring comprehensive coverage and validation of all project requirements.
3. **Test Cases:**
Detailed instructions for executing tests to verify specific functionalities or features of the PC Part Picker website.
4. **Test Scripts:**
Automated scripts for executing repetitive or complex test scenarios to streamline testing efforts.
5. **Test Environment Configuration:**
Files detailing the configuration of each test environment, including browser settings.
6. **Test Results:**
Documentation of test outcomes, including both successful and failed test cases, along with any defects or issues encountered.
7. **Software Test Report (STR):**
Summary of testing activities, including key findings, test coverage, defect metrics, and recommendations for further action.

These test artifacts play a crucial role in ensuring the quality and reliability of the PC Part Picker website by providing comprehensive documentation of testing activities, outcomes, and recommendations for improvement. Regular updates and reviews of these artifacts are essential to maintain alignment with project objectives and requirements throughout the development cycle.

7 Time Table

Task	Start Date	End Date	Comments
STR Document	18.3.2024	20.3.2024	
STP Document	20.3.2024	26.3.2024	
STR Document	27.3.2024	27.3.2024	