

SOFTWARE TEST PLAN



WorldCat Website

Team Members

Project Manager: Yair Amon

Test Team Lead: Tzahi Anedghr

Test Engineers: Wael Otman

TABLE OF CONTENTS

INTRODUCTION	3
1. TEST STRATEGY	3
1.1 Scope of Testing	3
1.2 Glossary	3
2. TESTS TREE	4
2.1 test tree of functional tests	4
2.2 test tree of non-functional tests	9
2.3. Feature to be tested	12
2.4 Features not be tested	14
3. TEST TYPES	14
4. RISK AND ISSUES	15
5. TEST LOGISTICS	16
5.1 who will test	16
5.2 when will the test occur	16
6. TEST OBJECTIVE	16
7. REST CRITERIA	16
7.1 suspension criteria	16
7.2 exit criteria	16
8. RESOURCE PLANNING	17
8.1 system resource	17
8.2 human resource	18
9. TEST ENVIRONMENT	19
10. SCHEDULE & ESTIMATION	20
10.1 all project task and estimation	20
10.2 schedule to complete these tasks	20
11. TEST DELIVERABLES	21
11.1 before testing phase	21
11.2 during the testing	22
11.3 after the testing cycle is over	22

INTRODUCTION

- 1 WorldCat stands as a cornerstone in the realm of digital library resources, serving as a global online catalog connecting users with an extensive array of scholarly and educational materials housed in libraries across the globe. With its expansive reach and depth of content, the WorldCat website serves a vital purpose in facilitating access to a wealth of knowledge, ranging from books and journals to multimedia resources. It functions as a centralized hub, empowering users to explore and discover a vast array of resources while bridging geographical boundaries and fostering collaboration among libraries worldwide. Through the WorldCat platform, users can seamlessly navigate the rich tapestry of library collections, locate materials available in their local library, and even facilitate resource sharing through interlibrary loan requests. Thus, the WorldCat website serves not only as a tool for information discovery but also as a catalyst for scholarly exchange and collaboration on a global scale.

2 TEST STRATEGY

2.1 Scope of Testing

The Software Test Plan (STP) for the WorldCat website aims to validate its core functionalities, including searching for materials, viewing item details, checking availability, and requesting items through interlibrary loan. Testing will cover both functional and non-functional aspects, ensuring that the website is user-friendly, secure, and performs well under different conditions.

2.2 Glossary

- **Interlibrary Loan:** A service that allows users to borrow materials from other libraries.
- **Item Details:** Information about a specific book, journal, or other library material, including title, author, publication date, and description.
- **Availability:** Indicates whether an item is currently in stock or available for loan.
- **Metadata:** Descriptive information about a resource, including details such as title, author, publisher, date, and subject. Metadata provides context and facilitates discovery, management, and use of library materials.
- **Search Filters:** Options for refining search results based on criteria such as format, language, and publication date.
- **Circulation:** Borrowing and returning books from the library, and keeping track of who has what and when it's due back.
- **OPAC (Online Public Access Catalog):** A website where you can search for books in a library from your computer or phone.
- **STP (Software Test Plan):**

Definition: Document outlining testing goals, methods, resources, and schedule.

Explanation: Guides testing process, detailing what, how, and by whom testing will be conducted.

- **STD (Software Test Description):**

Definition: Detailed document specifying individual test cases.

Explanation: Describes test scenarios, inputs, expected outcomes, and execution steps.

- **STR (Software Test Report):**

Definition: Summary document detailing testing results.

Explanation: Provides insights into testing outcomes, including defects and overall software quality.

- **Bugs:**

Definition: Software errors causing unexpected behavior.

Explanation: Range from minor to critical issues, affecting functionality or security.

- **Test Case:**

Definition: Set of steps to verify software functionality.

Explanation: Guides systematic testing of software features.

2. TESTS TREE

2.1 test tree of functional tests

A) Functional tests:

1) User Authentication

- a. Login with valid username and password.
 - Verify successful login.
- b. Login with valid username and invalid password.
 - Verify error message for invalid password.
- c. Login with non-existing username.
 - Verify error message for non-existing username.
- d. New Customer Registration:
 - Navigate to registration page.
 - Fill out registration form with valid information.
 - Submit registration form.
 - Verify successful registration and login.

2) Search Functionality

- a. Search for Materials:
 - Enter search keyword(s) in search bar.
 1. Verify search results.
 - Refine search results by filters (e.g., format, publication date).
 1. Verify filtered search results.
 - Test Advanced Search Options (e.g., Boolean operators, field-specific searches).

3) Viewing Item Details

- a. Click on a search result to view item details.
 - Verify item details page loads correctly.
 - Check for all relevant information (title, author, publication details).
 - Navigate back to search results or other parts of the website.

4) Checking Availability

- a. View availability status of an item.
 - Verify availability information is displayed.
 - Check if item is available for borrowing.
 - Test options for placing a hold or requesting item through interlibrary loan

5) User Account Management

- a. Access user account/profile section.
 - Log in with valid credentials.
 - 1. Verify user account information is displayed correctly.
 - Update profile information (e.g., name, email address).
 - 1. Verify changes are saved successfully.
 - View borrowing history or currently borrowed items.
 - 1. Verify information is accurate and up-to-date.

6) Navigation and Site Structure

- a. Test main navigation menus and links.
 - Verify easy access to key features and functionalities.
- b. Check for consistency in design and layout across different pages.
- c. Ensure accessibility features are implemented (e.g., keyboard navigation, screen reader support).

7) Interlibrary Loan Requests

- a. Access Interlibrary Loan (ILL) functionality.
 - Check if users can request items not available in their local library's collection.
 - 1. Verify the process for submitting an interlibrary loan request.
 - Test options for specifying loan duration and pickup location.
 - 1. Verify that users receive confirmation of the interlibrary loan request.
 - Check if users can track the status of their interlibrary loan requests.
 - 1. Verify that users receive notifications or updates regarding the status of their requests.\

8) Error Handling and Messaging

- a. Intentionally trigger various types of errors or unexpected scenarios.
 - Verify clear and user-friendly error messages.
- b. Test error handling for timeouts, network errors, or invalid user inputs.
 - Ensure error messages provide actionable information or guidance.

2.2 test tree of functional tests

B) Non-Functional tests

1) Performance Testing

a. Load Testing:

- Simulate concurrent user traffic to assess website performance under load.
 1. Verify response times and server resource utilization under varying load conditions.

b. Stress Testing:

- Exceed the website's capacity to determine its breaking point and identify performance bottlenecks.
 1. Verify if the website gracefully handles high loads without crashing or slowing down excessively.

c. Endurance Testing:

- Evaluate website performance over an extended period to ensure stability and reliability.
 1. Verify if the website maintains consistent performance over time without degradation.

2) Usability Testing

a. User Interface:

- Evaluate website layout, design, and navigation for intuitiveness and user-friendliness.
 1. Verify that users can easily find and access key features and functionalities.

- Accessibility:

1. Test website accessibility for users with disabilities (e.g., screen readers, keyboard navigation).

- a. Verify compliance with accessibility standards (e.g., WCAG guidelines)

- Error Handling:

1. Evaluate error messages and prompts for clarity and helpfulness.

- a. Verify that error messages provide actionable information to users.

3) Localization and Globalization Testing

a. Localization Testing:

- Test website functionality with different language settings.

1. Verify that all user-facing text and content are correctly translated and displayed in the selected language.

- Test date and time formats, currency symbols, and other locale-specific settings.

1. Verify that these elements are adapted to the selected language and region.

b. Globalization Testing:

- Test website functionality with different regional settings.
 1. Verify that the website can handle date and time formats, currency conversions, and other regional preferences.
- Test support for non-Latin characters and scripts.
 1. Verify that the website displays and handles text correctly in languages with different writing systems.

4) **Compatibility Testing**

- a. Browser Compatibility:
 - Test website functionality across different web browsers (e.g., Chrome, Firefox, Safari, Edge).
 1. Verify consistent behavior and rendering across supported browsers.
- b. Device Compatibility:
 - Test website responsiveness and functionality on various devices (e.g., desktops, laptops, tablets, smartphones).
 1. Verify that the website adapts seamlessly to different screen sizes and resolutions.

5) **Recovery Testing**

- a. Data Backup and Restoration:
 - Test backup and restoration procedures to ensure data integrity and availability in case of data loss or corruption.
 1. Verify that backups are performed regularly and securely stored.
 - Test the process for restoring data from backups to ensure timely recovery in case of data loss.
- b. Disaster Recovery:
 - Test disaster recovery procedures to ensure business continuity in case of catastrophic events.
 1. Verify that redundant systems and failover mechanisms are in place to minimize downtime.
 - Test the website's ability to recover gracefully from system failures or data loss without data corruption or service interruption.

6) **Reliability Testing**

- a. Fault Tolerance:
 - Test website behavior under unexpected failures
 1. Verify that the website gracefully handles failures and maintains essential functionality.
- b. Disaster Recovery:
 - Test backup and recovery procedures to ensure data integrity and availability in case of catastrophic events.
 1. Verify the website's ability to recover from data loss or corruption.

7) Scalability Testing

a. Vertical Scaling:

- Assess the website's ability to handle increased workload by adding more resources
 1. Verify if performance improves proportionally with resource scaling.

b. Horizontal Scaling:

- Evaluate the website's ability to handle increased traffic by adding more servers or instances.
 1. Verify if the website can distribute load evenly across multiple servers and maintain performance.

8) Security Testing

a. Authentication and Authorization:

- Test login functionality with valid and invalid credentials.
 1. Verify that unauthorized access attempts are blocked.

b. Data Protection:

- Ensure sensitive user information (e.g., passwords, personal data) is encrypted during transmission.
 1. Verify adherence to security standards

c. Secure Sessions:

- Test session management to prevent session hijacking or fixation attacks.
 1. Verify that sessions expire after a reasonable period of inactivity.

2.3. Feature to be tested

Module Name	Applicable Roles	Description
Login	User	<ul style="list-style-type: none">● Confirm successful login with correct username and password.● Confirm unsuccessful login with correct username and incorrect password.● Confirm unsuccessful login with incorrect username.● Test login using a connected account.
Search	User	<ul style="list-style-type: none">● Users can search for books, journals, and other materials.● Test search functionality with various keywords and filters.
View Item Details	User	<ul style="list-style-type: none">● Verify that users can view detailed information about each item (title, author, publication details, etc.).● Test navigation between item details pages.
Checking Availability	User	<ul style="list-style-type: none">● Verify the availability status of items in local libraries.● Test options for placing holds or requesting items through interlibrary loan.
User Account Management	User	<ul style="list-style-type: none">● Access user account/profile section.● Update profile information and preferences.● View borrowing history, current loans, and any outstanding requests.
Interlibrary Loan Requests	User	<ul style="list-style-type: none">● Process for submitting interlibrary loan requests.● Options for specifying loan duration, pickup location, etc.● Status tracking for interlibrary loan requests.
Navigation and Site Structure	User	<ul style="list-style-type: none">● Test main navigation menus and links.● Ensure consistency in design and layout across different pages.● Verify accessibility features for users with disabilities.
Error Handling and Messaging	User	<ul style="list-style-type: none">● Test error messages and prompts for clarity and helpfulness.● Ensure proper handling of unexpected errors or system failures.● Provide guidance for users in resolving common issues or errors.
Usability Testing	User	<ul style="list-style-type: none">● Evaluate website layout, design, and navigation for intuitiveness and user-friendliness.● Test accessibility features for users with disabilities.● Check for clarity and helpfulness of error messages.

2.4 Feature not to be tested

These features are not be tested because they are not included in the software requirement specs

- Software Interfaces
- Database logical
- Communications Interfaces
- Website Security

3 Test Type

To ensure the reliability and usability of the WorldCat website, a variety of test types should be conducted. Here are some key test types:

1. Functional Testing:

- Ensure that all functions of the website work as expected according to the specified requirements. This includes testing features such as search functionality, viewing item details, checking availability, user account management, and interlibrary loan requests.

2. Usability Testing:

- Evaluate the user interface design, navigation, and overall user experience of the website. This involves testing how easy it is for users to accomplish tasks, find information, and navigate through the site. Usability testing helps identify any usability issues or areas for improvement.

3. Performance Testing:

- Assess the performance of the website under various conditions, including different levels of user traffic and system loads. Performance testing helps ensure that the website responds quickly and efficiently, with minimal latency and downtime.

4. Security Testing:

- Identify and address potential security vulnerabilities and risks in the website. Security testing includes various types of tests such as penetration testing, vulnerability scanning, and security code reviews to ensure that the website is protected against unauthorized access, data breaches, and other security threats.

5. Compatibility Testing:

- Test the website's compatibility with different web browsers, devices, and operating systems. Compatibility testing helps ensure that the website functions correctly and displays properly across a wide range of platforms and devices, providing a consistent user experience for all users.

6. Accessibility Testing:

- Evaluate the accessibility of the website for users with disabilities, including those with visual, auditory, motor, or cognitive impairments. Accessibility testing ensures that the website complies with accessibility standards such as the Web Content Accessibility Guidelines (WCAG) and is usable by all users, regardless of their abilities.

7. Localization Testing:

- Test the website's functionality and content in different languages and regions. Localization testing ensures that the website is culturally and linguistically appropriate for users in different geographic locations, with localized content, date formats, currencies, and other relevant factors.

8. Regression Testing:

- Verify that recent changes or updates to the website have not introduced any new defects or issues and that existing functionality remains intact. Regression testing helps maintain the reliability and stability of the website over time, especially as new features are added or modifications are made.

4 Risk and Issues

Risk	Mitigation
Tests might miss important parts	Carefully plan tests to make sure all important things are tested. Choose the most important things to test first.
Keeping tests working might be hard	Make a strong system for running tests that's easy to fix when things change. Check and fix tests regularly.
The tools we use might not be good enough	Pick testing tools carefully to make sure they can do what we need. If they're not good enough, find ways to make them better.
Tests might need special information	Plan how to use information in tests so they can work even if things change. Use ways of testing that don't need special information.
The project schedule is too tight; it's hard to complete the whole project on time.	Setting Test Priority for each of the test activity.
Tests might say the wrong thing	Make sure tests give the right answers by checking them often. Fix any problems quickly.
Tests might not work with other systems	Work closely with other teams to make sure tests work well with the rest of the system. Test how tests work with other parts often.
Limited Testing Opportunities	Explore alternative testing methods such as using test environments or mockups resembling the website.

5 Test Logistics

5.1 Who will test?

The project will be tested by the Beyonddev QA Software Engineering Team.

5.2 When will the test occur?

Having the tests is a crucial part of the planning process. The timing of tests can be broadly categorized into several phases of the software development lifecycle and based on specific criteria.

- During requirements analysis.
- After Design Phase.
- During Development.
- Before and After Deployment.

6 TEST OBJECTIVE

the test objective for testing the WorldCat website might be to ensure that all core functionalities, such as searching for materials, viewing item details, and requesting items through interlibrary loan, work correctly and meet the requirements specified for the website. Additionally, the test objective may include verifying the reliability, usability, performance, security, and compliance of the website with relevant standards and regulations.

Overall, the test objective provides a clear direction for testing efforts and helps stakeholders understand the purpose and expected outcomes of the testing activities.

test objectives include:

- Accuracy of Search Results
- Completeness of Item Details
- Functionality of Interlibrary Loan Requests
- User Account Management
- Security of User Data
- Accessibility for Users with Disabilities
- Compatibility with Different Devices and Browsers:
- Compliance with Library Standards and Practices:
- User Feedback and Usability Testing:

7 TEST CRITERIA

7.1 Suspension Criteria

If the team members report that there are **40%** of test cases **failed**, suspend testing until the development team fixes all the failed cases.

The tests are tests that constitute a better and higher quality user experience.

7.2 Exit Criteria

Specifies the criteria that denote a **successful** completion of a test phase

-**Run** rate is mandatory to be **100%** unless a clear reason is given.

-**Pass** rate is **80%**, achieving the pass rate is **mandatory**.

8 RESOURCE PLANNING

8.1 System Resource

No	Resources	Descriptions
1.	Selenium Grid Nodes	Set up multiple nodes in the Selenium Grid to distribute test execution across different environments and browsers.
2.	Automation Test Tool	Utilize Selenium WebDriver as the primary test automation tool, integrated with Python to develop and execute automated test scripts.
3.	Test Environment	Create a stable test environment that replicates the production environment of the WorldCat website, including necessary dependencies.
4.	Computer	Deploy at least 4 computers running Windows 10 with 8GB RAM and a CPU clocked at 3.5GHz for testing various functionalities concurrently.
5.	Test Data	Prepare relevant test data and scenarios to simulate various user interactions and edge cases during automation testing.
6.	Browser Compatibility	Ensure compatibility with major web browsers (e.g., Chrome, Firefox, Edge) to validate website functionality across different platforms.
7.	Mobile Device Emulators	Utilize mobile device emulators or real devices to test the responsiveness and functionality of the website on different mobile platforms.
8.	Version Control System	Implement a version control system (e.g., Git) to manage test scripts, track changes, and facilitate collaboration among team members.

8.2 Human Resource

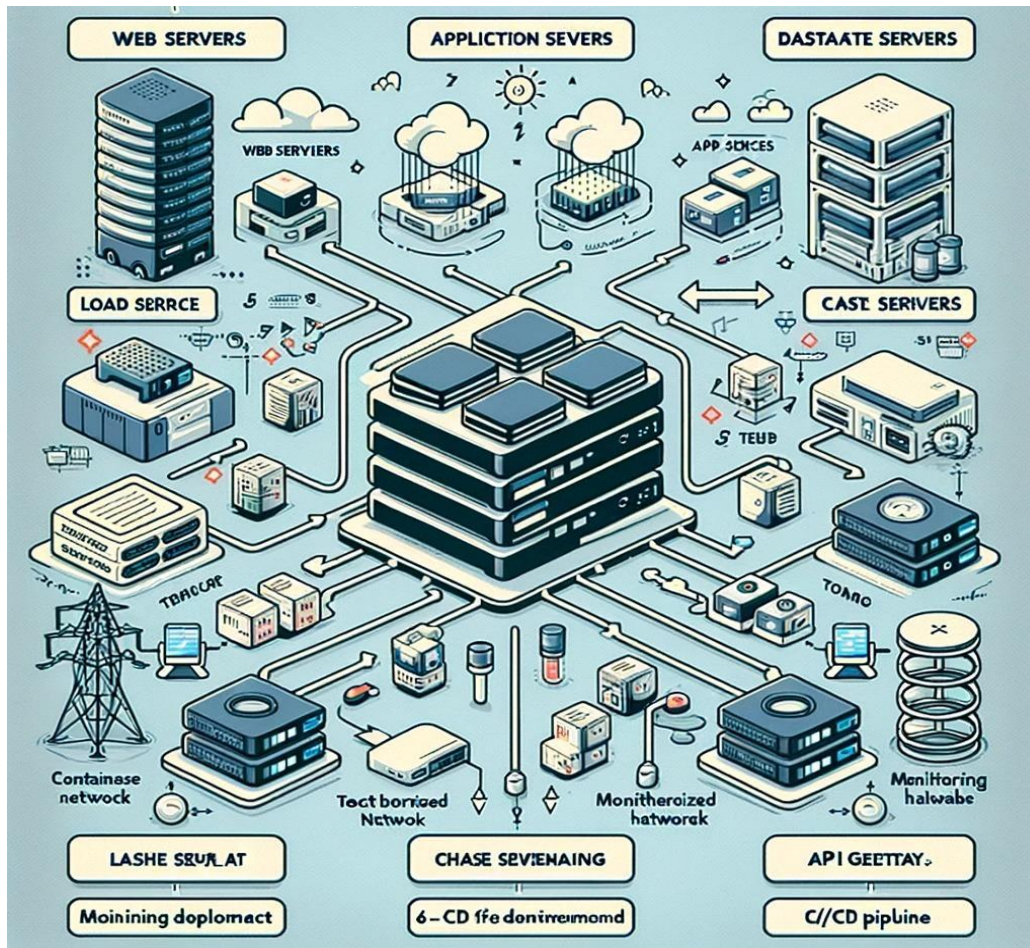
No.	Member	Tasks
1.	Project Manager	Leads the testing project. Works with others and makes sure everyone has what they need to do their job.
2.	Test Team Lead	Lead the automation testing team in defining test strategies, selecting appropriate tools, and designing test frameworks using Selenium Grid with Python.
3.	Automation Testers	Write and run the tests. Work with others to make sure we're testing the right things.
4.	Test Environment Manager	Sets up and looks after the place where we do the testing. Makes sure everything works properly.
5.	Quality Assurance Analysts	Check the tests to make sure they're good quality. Give advice on how to make the tests better.

9 TEST ENVIRONMENT

The Test Environment should be setup as figure below.

test environment for <https://search.worldcat.org/> that include:

- Servers: Host the website and database.
- Network: Stable LAN and internet connection.
- Operating Systems: Windows, macOS, Linux for cross-platform testing.
- Browsers: Chrome, Firefox, Edge, Safari for browser compatibility.
- Mobile Devices: iOS and Android devices for mobile testing.
- Test Data: Diverse datasets for simulating user scenarios.
- Automation Tools: Selenium WebDriver with Python for automated testing.
- Version Control: Git/GitHub for managing test scripts.
- Test Management: Tools like Jira for planning and tracking tests.
- Security Measures: Protect data and ensure privacy compliance during testing.



The test environment include:

Web Servers: To serve the web pages of the Booking.com site.

Application Servers: To handle the business logic and processing of the application.

Database Servers: To store all the data related to users, bookings, accommodations, etc.

API Gateway: To manage and route API requests to the appropriate services.

CI/CD Pipeline: For continuous integration and continuous deployment, integrating with version control.

10 SCHEDULE & ESTIMATION

10.1 All project task and estimation

Task	Members	Estimate effort
Create the test specification	Test Administrator	35 man-hours
Perform Test Execution	Tester, Test Administrator	25 man-hours
Test Report	Tester	15 man-hours
Test Delivery	Tester	20 man-hour
Total		95 man-hour

10.2 Schedule to complete these tasks



11 TEST DELIVERABLES

Test Deliverables for WorldCat Site:

- **Test Plan:** A comprehensive document outlining the testing approach, objectives, scope, resources, and schedule for testing the WorldCat website.
- **Test Cases:** Detailed test cases specifying the steps to be followed, expected outcomes, and criteria for passing or failing each test scenario.
- **Test Scripts:** Automation scripts developed using Selenium WebDriver with Python to automate test execution for regression and functional testing.
- **Test Data:** Relevant datasets used to simulate various user scenarios and edge cases during testing, including input data and expected outcomes.
- **Test Reports:** Documentation of test results, including test execution logs, defects identified, and overall assessment of software quality.
- **Defect Reports:** Reports detailing any defects or issues identified during testing, including descriptions, severity, steps to reproduce, and status.
- **Test Environment Setup:** Documentation of the test environment configuration, including server specifications, network setup, browser versions, and mobile devices used for testing.
- **Test Coverage Analysis:** Analysis of test coverage, identifying areas tested and areas requiring additional testing to ensure comprehensive coverage.
- **Performance Metrics:** Performance test results and metrics, including response times, throughput, and resource utilization under different load conditions.
- **User Acceptance Testing (UAT) Feedback:** Feedback from user acceptance testing sessions, including any usability issues, suggestions for improvement, and validation of requirements.