

Test Plan For Upwork Client Website

Version: 1.0

Created: 24/02/2024

Last Updated: 27/02/2024

Upwork Client Website - Automation Testing

Prepared by : Ahmd Bdran

Table of Content

Software Test Plan

1. Introduction

- 1.1. Project Posting and Search
- 1.2. Search and Filter Tools
- 1.3. Messaging and Communication

2. Objectives

3. Test Tree

- **Functional tests**
- **Non-Functional tests**

4. Entry and Exit Criteria

- 4.1. Entry Criteria
- 4.2. Exit Criteria

5. Test Strategy

- 5.1. Test Approach
- 5.2. Test Categories
 - Functional Tests
 - Non-Functional Tests
- 5.3. In Scope
- 5.4. Out of Scope
- 5.5. Test Execution

6. Test Methodology

- 6.1. Validation and Defect Management
- 6.2. Defect tracking & Reporting
- 6.3. Status of the Bug/Fault
- 6.4. Test Management Process
- 6.5. Assumptions for Test Execution
- 6.6. Risk

7. Test Environments

8. Time Table

9. The Team's Responsibilities

1. Introduction

Upwork is a leading online platform connecting businesses with skilled freelancers across a wide array of disciplines, including writing, programming, graphic design,

marketing, and more. Launched in 2015, Upwork offers a streamlined process for employers to find talent for their projects, whether short-term or long-term. Freelancers showcase their expertise through detailed profiles, portfolios, and skill assessments, enabling employers to make informed hiring decisions. With its user-friendly interface and robust features such as messaging, time tracking, and secure payment processing, Upwork facilitates efficient collaboration between clients and freelancers regardless of geographical location. Whether you're a business seeking specialized expertise or a freelancer looking for opportunities to showcase your skills, Upwork provides a dynamic marketplace for connecting talent with projects.

The key functionalities of the Website include:

1.1. Project Posting and Search

Businesses can post detailed descriptions of their projects, including requirements, budget, and timeline. Freelancers can then browse through available projects and submit proposals based on their expertise and interests.

1.2. Search and Filter Tools

Both businesses and freelancers can use advanced search and filter tools to find the right matches for their needs. Filters may include skills, experience level, location, hourly rate, and more.

1.3. Messaging and Communication

Upwork provides a messaging system that allows seamless communication between clients and freelancers. Users can discuss project details, clarify requirements, and negotiate terms directly on the platform.

2. Objectives

This test plan outlines the testing approach for the testing of UpWork Client Website. It covers the following key areas:

1. Test Strategy: Defines the rules the test will be based on, including project timelines, objectives, and assumptions.
2. Execution Strategy: Describes how the tests will be performed, the process for identifying and reporting defects, and implementing fixes.

3. Test Management: Outlines the process for handling the logistics of the test, including communication, escalation procedures, risk mitigation, and team roster.

3. Test Tree

Functional tests

1. Client Registration and Profile Setup

1.1. Verify registration process:

- 1.1.1 Register with valid client information
- 1.1.2 Register using third-party integration (if applicable)

1.2. Verify profile details update:

- 1.2.1 Update profile picture
- 1.2.2 Update company information (if applicable)
- 1.2.3 Update payment details

2. Job Posting and Management

2.1. Verify job posting process:

- 2.1.1 Post a new job with relevant details
- 2.1.2 Set job preferences (e.g., budget, timeline)

2.2. Verify job management:

- 2.2.1 Edit job details
- 2.2.2 Close job posting
- 2.2.3 Reopen job posting
- 2.2.4 Close a job post

3. Freelancer Search and Hiring

3.1. Verify freelancer search functionality:

- 3.1.1 Search for freelancers by skills and expertise
- 3.1.2 Filter freelancers by location, experience, and hourly rate

3.2. Verify hiring process:

- 3.2.1 Invite freelancers to submit proposals
- 3.2.2 Review and evaluate freelancer proposals
- 3.2.3 Hire selected freelancer(s)

4. Communication

4.1. Verify messaging system:

- 4.1.1 Send messages to freelancers
- 4.1.2 Respond to messages from freelancers

4.2. Verify video call functionality:

- 4.2.1 Initiate video calls with freelancers
- 4.2.2 Conduct interviews via video calls

5. Payments and Contracts

5.1. Verify payment process:

- 5.1.1 Create and fund escrow for a project
- 5.1.2 Release payment upon project completion
- 5.1.3 Handle payment disputes

5.2. Verify contract management:

- 5.2.1 Create a contract agreement
- 5.2.2 Amend contract terms
- 5.2.3 Close a contract upon completion or termination

Non-Functional tests

1. Performance Testing

1.1. Test system response under varying user loads:

- 1.1.1 Test with multiple concurrent job postings
- 1.1.2 Test with simultaneous messaging and video calls

1.2. Test system response under stress:

- 1.2.1 Test with high volumes of job applications and proposals
- 1.2.2 Test with heavy transaction loads during peak hours

1.3. Verify site load time:

- 1.3.1 Ensure the site loads in less than 5 seconds

2. Compatibility Testing

2.1. Test website compatibility across browsers:

- 2.1.1 Test on Google Chrome
- 2.1.2 Test on Mozilla Firefox

Localization and Globalization Testing

5.1. Verify multilingual support:

- 5.1.1 Change website language to hebrew

5.2. Test currency conversion:

- 5.2.1 Check currency display for different regions

4. Entry and Exit Criteria

4.1. 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

4.2. 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 Strategy

5.1. Test Approach

The test strategy for the UpWork website will focus on a comprehensive automation testing process, emphasizing both

functional and non-functional aspects to ensure a seamless and user-friendly interface.

5.2. Test Categories

Functional Tests

- **Login Page:** Valid and invalid login scenarios will be tested, including third-party logins such as Google and Apple accounts. Password recovery processes will also be evaluated.
- **Client Dashboard:**
 - Project posting: Testing the process of creating and posting projects for clients, including project descriptions and requirements.
 - Proposal management: Verifying the functionality for clients to manage proposals received from freelancers, including reviewing, accepting, or declining proposals.
 - Communication tools: Testing messaging and communication features between clients and freelancers to facilitate collaboration.

Non-Functional Tests

- **Website Compatibility:** The website's performance will be tested across a range of browsers to ensure consistent user experience.
- **Website Performance**

5.3. In Scope

Exploratory Testing: The first phase of testing, focusing on key functionalities such as signup, messaging.

End-to-End Testing: A user journey simulation from login to checkout will be conducted, including the payment and confirmation stages.

5.4. Out of Scope

Performance Testing: Tests simulating extreme user loads and database overload conditions will not be conducted at this stage.

Localization and Globalization: Testing for support of multiple languages is not included in the current scope.

Recovery Testing: Session persistence and cart item retention after page reloads will be checked, but broader recovery testing will be deferred.

5.5. Test Execution

Test execution will be methodically documented, with automation test cases detailing the procedures, expected results, and observed outcomes to ensure thorough coverage.

6. Test Methodology

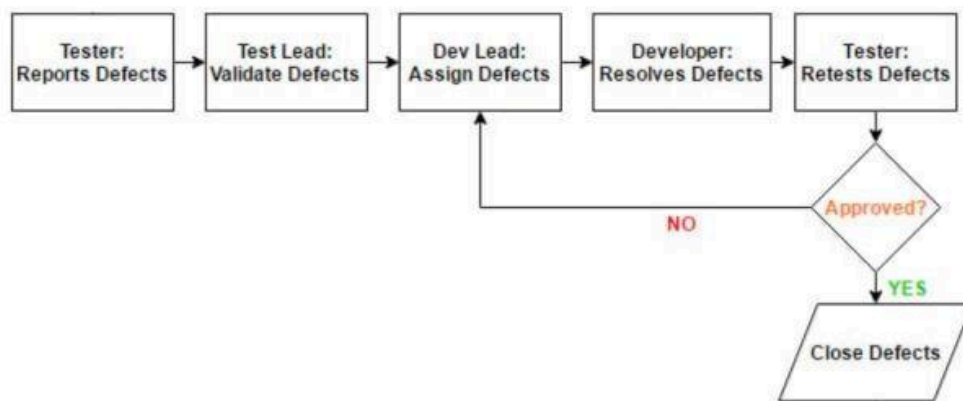
6.1. Validation and Defect Management

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 .

6.2. Defect tracking & Reporting



6.3. 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 was been fixed and closed but appear again

6.4. Test Management Process

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

6.5. Assumptions for Test Execution

- The test environment will remain stable and available throughout the test execution phase.
- Sufficient and representative test data will be available for executing test cases.
- Test resources, including testing tools and equipment, will be accessible and ready for use during test execution.
- Test preparation activities, such as test case design, documentation, and environment setup, will be completed before the test execution phase begins.
- Effective communication channels will be established and maintained throughout the test execution phase.

6.6. Risk

#	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

7. Test Environments

- Operating System: Windows 11 OS Build 22631.3085
- Browser 1: Google Chrome Version 121.0.6167.140
 - All Extension are disabled
 - Cookies are allowed
- Browser 2: Firefox Version V 122.0
 - All Extension are disabled

- Cookies are allowed

8. Time Table

Task	Start Date	End Date	Comments
STP Document	26.2.2024	27.2.2024	
STD Document	27.2.2024	28.2.2024	
Automation tests run	1.03.2024	1.03.2024	
STR Document	-	-	