

Design Shack

TEST PLAN

Version 1.0
09/25/2022

Document created by:
Viktoria Voloshina

TEST PLAN CONTENTS:

1	Introduction.....	4
2	Scope.....	4
	2.1 In Scope.....	4
	2.2 Out of Scope.....	4
3	Quality Objectives.....	4
4	Test Approach.....	5
	4.1 Website Manual Test.....	5
	4.2 The UI/UX Smoke Testing.....	5
	4.3 Website Automation Tests.....	6
	4.4 Website API Tests.....	6
	4.5 Website Performance Tests.....	6
5	Roles and Responsibilities.....	7
6	Suspension Criteria and Resumption Requirements.....	7
7	Test Types.....	7
8	Test Strategy.....	9
	8.1 QA Role in the Test Process.....	9
	8.2 Bug Triage: Bug Severity and Priority Definition.....	9
	8.2.1 Bug Severity List.....	10
	8.2.2 Bug Priority List.....	11
9	Entry and Exit Criteria.....	11
10	Resources and Environment Needs.....	11
	10.1 Testing Tools.....	11
	10.2 Test Environment X Support Level 1 (Browsers).....	12
11	Test Schedule.....	12
12	Approvals.....	12
13	Terms/Acronyms.....	13

Change Log:

Version	Change Date	By	Description
Version Number	Date of Change	Name of person who made changes	Description of the changes were made

Document Revision History:

Date	Version	Author	Description	Reviewer	Approved

1. INTRODUCTION

Customers want a perfect website, which passes the full cycle of manual testing. Given the specificity of the site it is important to have the same quality of the site as per requirements.

The Test Plan has been created to facilitate communication among the team members. This document describes approaches and methodologies that will apply to the unit, integration, and system testing of <https://designshack.net> . It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria and approach. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

2. SCOPE

2.1 In Scope

The document mainly targets GUI testing and validation data in report out as per Requirements Specifications provided by Client.

Functions to be tested:

- GUI
- Registration Form
- Search and Filters Logic
- API Collections
- Website Performance

2.2 Out of Scope

There features NOT to be tested because they are not included in the software Requirement Specifications.

Functions NOT to be tested:

- Hardware Interfaces
- Software Interfaces
- Database Logical
- Communication Interfaces
- Website Security

3. QUALITY OBJECTIVES

A *primary objective* of testing is to: assure that the system meets the full requirements, including quality requirements (functional and non-functional requirements) and fit metrics for each quality requirement and satisfy the use case scenarios and maintain the quality of the product.

The *secondary objectives* of testing will be to identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release.

4. TEST APPROACH

In the project 'Design Shack' there are 4 types of testing that should be conducted:

- Manual Tests
- Automation Tests
- API Tests
- Performance Tests

The project is using an Agile approach, with iterations every 14 days. At the end of the *second week* the requirements identified for that iteration will be delivered to the team and will be tested.

Team also must use experience-based testing and error guessing to utilize tester's skills and intuition, along with their experience with similar applications or technologies.

An Analytical test approach was used, in accordance with the requirements-based strategy, where an analysis of the requirements specification form is the basis for planning, estimating, and designing tests. Test cases will be created during exploratory testing. All test types are determined in Test Strategy.

4.1 Website Manual Test

1. *Check the ability of user sign up with:*

- Valid data (valid email) - Positive testing
- Invalid data (invalid email) - Negative testing

2. *Check the ability of user to reach all NASA's social medias pages by clicking on the following provided Social Media Icons: Twitter, Facebook, Pinterest, RSS Feed.*

3. *The **Design Shack** Website Modules of Menu '**Articles**', '**Gallery**', '**About**' will be tested in this part.*

4. *Check 'Search Design Shack'.*

4.2 The UI/UX Smoke Testing

In this part will be performed for Design Shack's website verification UI response to Design requirements.

Plan:

1. Verify that all images' links on Homepage are working correctly according to Business requirements.
2. Verify that Logo leads at the website's Homepage after clicking on it.
3. Verify that the "Search" field is working correctly according to Business requirements.
4. Positive/Negative/ADHOC testing should be used also if necessary.

Environments for this part:

1. OS:

- MacOS
- iOS
- Windows 11

2. Devices:

- MacBook Air (M1,2020)
- iPhone 11 Pro

3. Browsers:

- Google Chrome
- Safari (for MacOS)
- Firefox (for Windows OS)

4.3 Website Automation Tests

Tests which are indicated in this part will be automated:

1. Test for Module *Menu* '**Articles**', '**Gallery**', '**About**' with Sub Menu (to verify the ability of users to reach all pages).
2. Tests for UI/UX Smoke testing (to verify the response of UI/UX to design requirements).
3. Test for 'Search' field (to verify the correct functionality of this field).

Tools for this part:

- Selenium IDE

4.4 Website API Tests

Design Shack's API Collections will be tested through the Server response in this part.

Tools for this part:

- Postman 2
- JSON Formatter & Validator

4.5 Website Performance Tests

Tests for evaluation of the website's *speed, responsiveness and stability* under a workload will be done in this part.

Sub Plan for Performance testing:

1. Check Design Shack's website:
 - Performance
 - Accessibility
 - Best practice
 - SEO

Tools for this part:

1. Lighthouse
2. GT Metrix
3. SpeedLab

5. ROLES AND RESPONSIBILITIES

Role	Staff members	Responsibilities	
Project Manager		1	Acts as a primary contact for development and QA team.
		2	Responsible for Project schedule and the overall success of the project.
QA	Viktoria Voloshina	1	Writing and executing Test cases.
		2	Reviewing Test cases.
		3	Bug reporting and tracking.
		4	Bug reporting and tracking.
		5	Creating automation tests.
		6	Retesting and regression testing.
		7	Bug Review meeting.
		8	Performance testing
		9	Preparation of Test Data.
		10	Coordinate with QA Lead for any issues or problems.

6. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

Suspension criteria:

- Software/Hardware problems.
- Significant change in requirements suggested by the client.
- The build contains many serious defects which seriously limit testing progress.
- Assigned resources are not available when needed by the test team.

Resumption criteria:

Resumption will only occur when the problem that caused the suspension has been resolved.

7. TEST TYPES

In the project there are 9 types of testing that should be conducted:

1. Exploratory testing
2. Smoke Testing
3. GUI Testing
4. Functional Testing
5. Positive testing
6. Negative testing
7. ADHOC testing
8. API testing
9. Performance testing

Exploratory testing

Exploratory testing includes a type of software testing where Test cases are not created in advance but QA checks system “hands-on”. QA may note down ideas about what to test before test execution.

Smoke testing

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for the QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as “Build Verification Testing” or “Confidence Testing.”

In simple terms, we are verifying whether the important features are working and there are no showstoppers in the build that is under testing.

GUI Testing

GUI testing will include testing of the UI part of report. It covers users Report format, look, and feel, error messages, spelling mistakes, GUI guideline violations.

Functional Testing:

Functional testing is carried out in order to find out *unexpected behavior* of the report. The characteristics of functional testing are to provide correctness, reliability, testability, and accuracy of the report output/data.

Positive testing

Positive testing will include the type of testing that can be performed on the system by providing the *valid data* as input. It checks whether an application behaves as *expected* with positive inputs.

Negative testing

Negative testing will include a method of testing an application or system that ensures that the application is according to the requirements and can handle the unwanted input and user behavior. *Invalid data* is inserted to compare the output against the given input. Negative testing is also known as *failure testing* or error path testing. When performing negative testing *errors messages* are expected.

ADHOC testing

ADHOC testing includes an informal testing type with an aim to break the system.

API testing

API testing is a type of software testing that analyzes an application program interface (API) to verify it fulfills its expected functionality, security, performance and reliability. An API test is generally performed by making requests to one or more API endpoints and comparing the response with expected results.

Performance testing

Performance testing check the optional time the page is loaded and check the operation of the system under load.

8. TEST STRATEGY

8.1 QA role in the test process:

- Understanding Requirements.
- Requirement specifications will be sent by the client.
- Understanding of requirements will be done by QA.
- Preparing *Test Cases*

QA will be preparing test cases based on the exploratory testing. This will cover all scenarios for requirements.

- Preparing *Test Matrix*

QA will be preparing a test matrix that maps test cases to respective requirements. This will ensure the coverage for requirements.

- Reviewing Test Cases and Test Matrix
- Review for test cases and test matrix will be conducted by QA Lead.
- Any comments or suggestions on test cases and test coverage will be provided by the reviewer.
- Suggestions or improvements will be updated by the preparer and sent to QA Lead for approval.
- Updates and improvements will be reviewed and approved by the reviewer.
- *Creating Test Data:*

Test data will be created by respective QA based on scenarios and Test cases.

- *Executing Test Cases:*

Test cases will be executed by respective QA on the client's development/test site based on designed scenarios, test cases and Test data.

- Test result (Actual Result, Pass/Fail) will be updated in the test case document *Defect Logging and Reporting*. QA will be logging the defect/bugs in an Excel spreadsheet and JIRA, found during the execution of test cases.

- *Retesting and Regression Testing:*

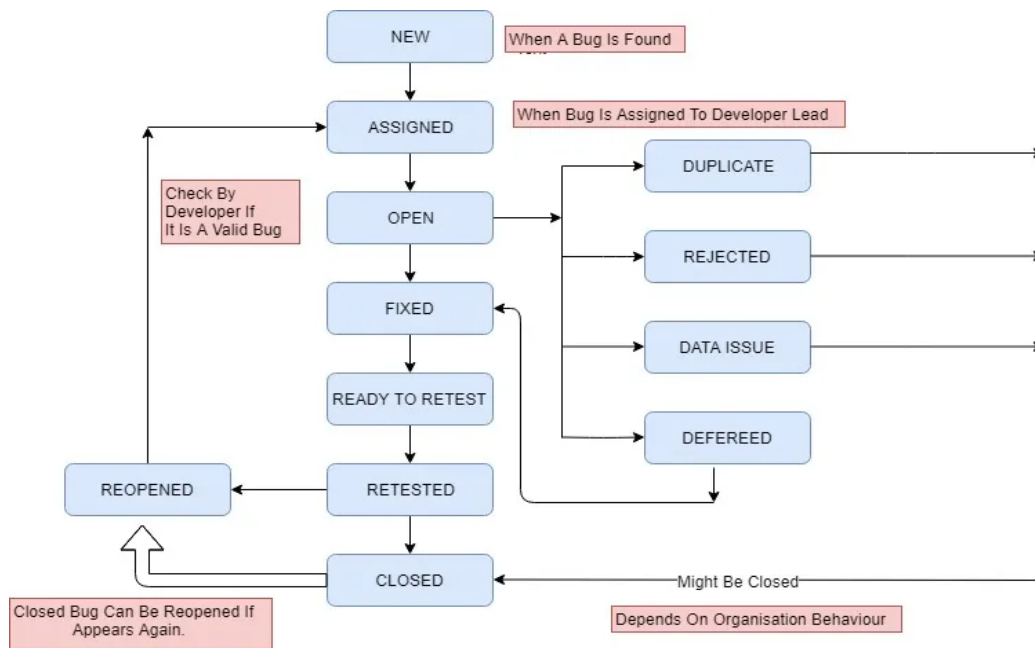
Retesting for fixed bugs will be done by respective QA once issue is resolved by respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

- *Deployment/Delivery:*

Once all bugs/defect reported after complete testing is fixed and no other bugs are found, report will be deployed to the client, along with sample output by email to respective lead and Report Group

8.2 Bug triage: Bug Severity and Priority Definition:

All the issues found while testing will be logged into JIRA.



Bug **Severity** and **Priority** fields are both very *important for categorizing bugs* and *prioritizing* if the bugs will be fixed. The bug Severity and Priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs. The Test Lead will be responsible to see that a correct severity level is assigned to each bug.

The QA Lead, Development Lead and Project Manager will participate in bug review meetings to assign the priority of all currently active bugs.

This meeting will be known as “*Bug Triage Meetings*”. The QA Lead is responsible for setting up these meetings.

8.2.1 Bug Severity List:

Severity ID	Severity	Severity Description
1	Highest	The module/product <i>crashes</i> or the bug causes <i>non- recoverable conditions</i> . System crashes or database or file corruption, or potential data loss, program hangs requiring reboot are all examples of a Severity 1 bug .
2	High	<i>Major system components unusable due to failure</i> or incorrect functionality. Severity2 bugs cause serious problems such as a <i>lack of functionality</i> , or insufficient or unclear error messages that can have a major impact to the user, prevents other areas of the app from being tested, etc. Severity 2 bugs can have a work around, but the work around is inconvenient or difficult.
3	Medium	<i>Incorrect functionality</i> of component or process. There is a simple work around for the bug if it is Severity 3 .
4	Low	Documentation errors or signed off Severity 4 bugs. Low severity bug occurs when there is <i>almost no impact</i> on the functionality, but it is still a valid defect that should be corrected.

8.2.2 Bug Priority List:

Priority	Priority Level	Priority Description
1	Highest	This bug <i>must be fixed immediately</i> ; the product cannot ship with this bug.
2	High	These are important problems that <i>should be fixed as soon as possible</i> . It would be an embarrassment to the company if this bug shipped.
3	Medium	The problem <i>should be fixed within the time available</i> . If the bug does not delay the shipping date, then fix it.
4	Low	It is not important (at this time) that these bugs be addressed. <i>Fix these bugs after all other bugs</i> have been fixed.
5	Lowest	Documentation errors.

9. ENTRY AND EXIT CRITERIA

Entry Criteria:

- Proper test data should be available.
- All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.
- All the standard software tools including the testing tools must have been successfully installed and functioning properly.
- All test hardware platforms must have been successfully installed, configured, and functioning properly.

Exit Criteria:

- No high priority or severe bugs are left outstanding.
- A certain level of requirements coverage has been achieved.
- All high-risk areas have been fully tested, with only minor residual risks left outstanding.
- Cost – when the budget has been spent.
- The schedule has been achieved.

10. RESOURCES AND ENVIRONMENT NEEDS

10.1 Testing tools:

Process	Tool
Test case creation	Microsoft Word, Microsoft Excel, JIRA
Test case tracking	JIRA, Confluence
Test case execution	Manual, Selenium IDE
Test case management	Microsoft Excel, JIRA, Confluence
Defect reporting	Microsoft Excel, JIRA, Confluence

Test reporting	JIRA
IPA Testing	Postman
Performance Testing	Lighthouse, GT Metrix, SpeedLab
Automation Testing	Selenium IDA

10.2 Test Environment x Support Level 1 (browsers):

Test Environment to be setup as per figure below:

- Windows 11: Chrome (latest), Firefox (latest)
- Mac OS: Chrome (latest), Safari (latest)
- Android: Chrome (latest), Firefox (latest)

Support level 1 (devices):

- iPhone 11 Pro
- Mac Book Air (M1,2020)

Support level 1 (browsers):

- Safari (latest)

11. TEST SCHEDULE

Task Name	Start	Finish	Effort	Comments
Test planning				
Review Requirement documents	09.24.2022	09.30.2022		
Create test basis	09.24.2022		with team	flex
Staff and train new test resources	09.24.2022			flex as needed
Create test basis	09.24.2022	09.30.2022		
Black box testing	10.01.2022	10.20.2022		
Exploratory testing	10.01.2022	10.20.2022		
Positive testing	10.01.2022	10.20.2022		
Negative testing	10.01.2022	10.20.2022		
ADHOC testing	10.01.2022	10.20.2022		
Performance testing	10.01.2022	10.20.2022		
Release to Product	10.01.2022	10.20.2022		

12. APPROVALS

	Project Manager	QA Lead
Name		
Signature		

13. TERMS/ACRONYMS

The below terms are used as examples, please add/remove any terms relevant to the document.

TERM/ACRONYM	DEFENITION
API	Application Program Interface
GUI	Graphical user interface
PM	Project manager
UAT	User acceptance testing
CM	Configuration Management
QA	Quality Assurance
RTM	Requirements Traceability Matrix