Test Plan for ElysiumNFT

version 1.0

Content

1. Introduction **3**
1.1. Purpose 3
1.2. Terminology 3
1.3. Revision History 4
1.4. References 4
1.5. Overview 4

2. Test Items **5**
2.1. Features to be tested 5
2.2. Components and functions not to be tested: 6

3. Test Objectives **6**

4. Test Strategy **6**
4.1. Test Type 7
4.2. Test coverage 7
4.3. Stages of testing 8
4.4. Entry Criteria 8
4.5. Exit Criteria 8

5. Work plan **9**

6. Risks **9**

7. Resources **10**
7.1. Project Team and responsibilities 10
7.2. Test documentation 10
7.3. Test tools 11
7.4. Environments: 11

1. Introduction

This document describes the methods and procedures that will be used in the testing process of the ElysiumNFT project. Test plan has clearly identified what the test deliverables will be, and what is deemed in and out of scope

1.1. Purpose

This Test Plan document for the ElysiumNFT project supports the following objectives:

- define the test scope of the QC activities;
- identify the test objectives;
- define the test strategy and approach for testing;
- identify entry and exit criterias;
- define schedule major milestones;
- identify responsibilities;

1.2. Terminology

Term	Definition
QC	Quality control
QA	Quality Assurance
PM	Project Manager
BA	Business analyst

1.3. Revision History

Version #	Date	Changed By	Reason for change
0.1	05.12.2022	Shubina Nataliia	Creation

1.4. References

#	Name	Link
1	Software Required Specification	Specification
2	Prototype	Figma_Prorotype
3	UI/UX design	Figma_Design

1.5. Overview

The ElysiumNFT project is an NFT marketplace. It is a digital platform for buying and selling NFTs. These platforms allow people to store and display their NFTs plus sell them to others for cryptocurrency or money.

2. Test Items

This paragraph will define the test scope to be performed within the test plan. The list of what will be tested is based on the following documents:

- Requirements specifications;
- Design documents.

2.1. Features to be tested

For unregistered and registered users:

1. User authentication (Sign Up, Login, Forgot Password, Log Out)
2. Homepage;
3. Collections;
4. Gallery;
5. Profile (Details, Notification Settings, Account Support, Subscription)
6. View NFT;
7. Post NFT (Single upload, Bulk upload);
8. User Card;
9. Store;
10. Management (Collection, NFT, Messages);
11. Payments;
12. Settings;
13. Informational pages (About us, Contact us, Terms of Use, Privacy Policy);
14. Forum.

For Super Admin:

1. Dashboard;
2. Managements;
3. Payments;
4. Settings.

2.2. Components and functions not to be tested:

1. Operation of third-party services.

3. Test Objectives

The main objectives of testing:

- assure that the system meets the full requirements, including quality requirements and satisfies the use case scenarios and maintains the quality of the product;
- prevent defect;
- identify issues and associated risks
- find failure and defects;
- communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release;
- verify requirments.

4. Test Strategy

The purpose of a test strategy is to define the principles that guide the design of tests and govern how the software testing process will be carried out. The ElysiumNFT project will be tested using a “black box” approach without knowledge of the internal structure or program source code.

4.1. Test Type

In the scope of testing, the following types of testing will be be performed within the project (Table 1):

#	Type	Description
1	Functional testing	Performed to verify that the software system is against the functional requirements
2	Usability Testing	Performed testing UI/UX and Compatibility testing. It is used to check the GUI against the project's layout and existing design requirements.
3	Performance testing	Tested application stability and load response time. Will be applied: Load testing, Stability testing, Stress testing
3	Smoke testing	Performed on initial software builds to ensure that the critical functionalities of the program are stable working (or no)
4	Regression testing	Performed to make sure that any bug fixes, adding new features, deleting, or updating existing features, are not impacting the working application.
5	Sanity testing	Performed on stable builds to verify that a particular function works according to the requirements stated in the specification

4.2. Test levels

1	Integrational testing	Performed between the modules and expose any defects that may arise when these components are integrated and need to interact with each other Used types: - Smoke testing; - Functional testing - Usability testing; - Sanity testing - Regression testing - Sanity testing

2	System Testing	Performed for testing how work all system as a whole from the point of view of end users Used types: - Smoke testing; - Functional testing - Usability testing; - Performance testing - Regression testing
3	Acceptance testing	Performed to find as many defects as possible before releasing software to customers. Only alpha testing will apply Used types: - Functional testing; - Usability testing - Smoke testing; - Regression testing - Performance testing

4.3. Test coverage

All features included in the scope, as well as any changes or improvements to them, should be covered by the tests.

4.4. Stages of testing

Planned stages of the testing process:

1. Analysis of requirements for a software product and design for the project; drafting a test plan.
2. Testing the product design with a description of the defects found;
3. Cross-browser testing with a description of the defects found;
4. Detailed run of functional tests with the detection and description of defects;

5. Checking the bugs solved by the developers and conducting regression testing;

4.5. Entry Criteria

Testing can be started if the following entry criteria are met:

1. The necessary documentation is ready, approved and allowed;
2. Scope of work defined, described and prioritized;
3. Acceptance criteria defined.
4. Test cases/checklists are developed;
5. Ready & set test environment and tools for testing.

4.6. Exit Criteria

Testing is completed if the following exit criteria are met:

1. All detected bugs are reported;
2. Ensuring all critical test cases are passed successfully;
3. Deadlines meet or budget depleted;
4. Desired and sufficient coverage of the requirements and functionalities under the test.
5. Identifying and fixing all the high-priority defects;
6. Re-testing and closing all the high-priority defects.

5. Work plan

Milestone task	Effort	Start Date	End Date
Review Requirements documents	8h	02.12.2022	02.12.2022
Creating test plan	10h	03.12.2022	08.12.2022
Design Tests	40h	-	-
Executing Tests	30h	-	-

Create bug reports	15h		
Retesting	15h		
Regression testing	35h	-	-
Performance testing	4h	-	-
Release to Production	15h	-	-

Total time for testing ~ 176

6. Risks

#	Risk	Actions
1	Customer can change the basic requirements to the project of one of testing features	Some buffer has been added to the schedule for contingencies
2	The customer has reduced costs for develop the project	Review all tasks. Remove some functional and change priority for tasks
3	More time spent on review than expected	Some buffer has been added to the schedule for contingencies
4	Hardware failure during testing	Make backup resources

7. Resources

7.1. Project Team and responsibilities

#	Role	Name	Responsibilities
1	PM	A	- Monitor the progress of the project; - Identify and resolve issues that arise.
2	BA	B	- Create Prototype and Specification.
3	Designer	C	- Creating images and layouts by hand or using design software

4	Frontend developer	D	- Create a part of the web page that is visible to the user; - Implement user logic. - Assure that all user input is validated before submitting to back-end
5	Backend developer	not on the project yet	- Building and maintaining web applications; - Writing high-quality code for project functional; - Managing database; - Troubleshooting and debugging
6	QA Engineer	Nataliia Shubina	- Review software requirements; - Test Software to detect bugs; - Create test documentations (test plan, test reports, checklists, test cases, bug reports); - Execute tests; - Analyzing test results

7.2. Test documentation

The project has the following test documentation:

- Test plan;
- Checklists;
- Test cases;
- Bug reports;
- Test result reports.

7.3. Test tools

Name	Instrument
Bug-tracking system	Jira
Test cases/ checklists	Google Excel
Test Plan/ Test Strategy / Test Reports	Google Word
Screen recorder	Bandicam
Screen capture	Lightshot
Preformance testing	JMeter

GUI testing	- Figma; - WhatIsFont (browser extension); - Page Ruler (browser extension); - Web Developer (browser extension); - Browser DevTools
Work with database	Microsoft SQL Management Studio v.18 and latest
Work with API	Postman (web or desktop)

7.4. Environments

The testing should be performed in Google Chrome, Safari, Mozilla Firefox and Microsoft Edge browsers (use the latest testing version).

Desktop OS performed testing is Windows 10.

The environment for Functional and Regression testing is Staging:<url>

The environment for Acceptance Criteria is Production: <url>