Software Requirements Specifications

On

**NFT Based e-commerce Website**

Submitted as a part of course curriculum for

**Bachelor of Technology**

in

**Computer Science**



**Submitted by**

ANKIT KUMAR MISHRA (2100290120030)

ANIKET SINGH (2100290120029)

AYUSH MISHRA (2100290120062)

ANNU (2100290120032)

**Under the Supervision of**

DR.HARSH KHATTAR

Assistant Professor

**KIET Group of Institutions, Ghaziabad**

**Department of Computer Science**

**Dr. A.P.J. Abdul Kalam Technical University**

Table of Contents

* 1. [Introduction 2](#_bookmark0)
  2. [Purpose 2](#_bookmark1)
  3. [Scope of Project 2](#_bookmark2)
  4. [Glossary 2](#_bookmark3)
  5. [References 2](#_bookmark4)
  6. [Overview of Document 2](#_bookmark5)

[2.0. Overall Description 3](#_bookmark6)

* 1. [System Environment 3](#_bookmark7)
  2. [Functional Requirements Specification 4](#_bookmark8)
     1. [Consumer Use Case 4](#_bookmark9)
     2. [Consumer Use Case 4](#_bookmark10)
     3. [Consumer Use Case 6](#_bookmark11)
     4. [Consumer Use Cases 6](#_bookmark12)
  3. [Requirements Specification 9](#_bookmark13)

[3.1 External Interface Requirements 9](#_bookmark14)

* 1. [Performance Requirements 15](#_bookmark15)
     1. [Response Time 15](#_bookmark16)
     2. [Scalability 15](#_bookmark17)
     3. [Throughput 15](#_bookmark18)
     4. [Resource Utilization 15](#_bookmark19)
  2. [Security Requirements 16](#_bookmark20)
     1. [Data Encryption 16](#_bookmark21)
     2. [Authentication 16](#_bookmark22)
     3. [Authorization 16](#_bookmark23)
     4. [Smart Contract Security 16](#_bookmark24)
     5. [Compliance 17](#_bookmark25)
  3. [Usability Requirements 17](#_bookmark26)
     1. [User Interface Consistency 17](#_bookmark27)
     2. [Responsiveness 17](#_bookmark28)
     3. [Accessibility 17](#_bookmark29)
     4. [Internationalization 18](#_bookmark30)
  4. [Reliability and Availability 18](#_bookmark31)
     1. [System Uptime 18](#_bookmark32)
     2. [Disaster Recovery 18](#_bookmark33)
     3. [Backup 18](#_bookmark34)
  5. [Design specifications 19](#_bookmark35)
  6. [System Overview: 19](#_bookmark36)
  7. [Components 19](#_bookmark37)
  8. [Data Flow: 20](#_bookmark38)

[5.0. Testing Strategy 20](#_bookmark39)

* 1. [Test Levels: 20](#_bookmark40)
  2. [Test Types: 21](#_bookmark41)
  3. [Test Environments 21](#_bookmark42)
  4. [Test Data: 22](#_bookmark43)

[5.5. Test Cases and Scenarios 22](#_bookmark44)

* + 1. [Unit Testing 22](#_bookmark45)
    2. [Integration Testing 22](#_bookmark46)
    3. [System Testing 22](#_bookmark47)
    4. [User Acceptance Testing (UAT) 22](#_bookmark48)
    5. [Non-Functional Testing 23](#_bookmark49)

[6.0. References 24](#_bookmark52)

# Introduction

## Purpose

The purpose of this document is to define the requirements for the development of an NFT (Non-Fungible Token) marketplace platform that allows users to connect their wallets, buy, sell NFTs from a variety of categories, including art, collectibles, and gaming items, while also creating active communities which involves NFT enthusiasts

## Scope of Project

The NFT marketplace will provide a platform for the trading of unique digital assets, including art, collectibles, and more. It will support multiple cryptocurrencies and offer a user-friendly experience for both buyers and sellers.

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| NFT | Short for Non-Fungible Token, is a unique cryptographic  asset used to create and authenticate ownership of digital assets |
| Moderator | Person that manages the forum and reviews the articles published by the users and consumers. |
| Database | Collection of all the information monitored by this system. |

|  |  |
| --- | --- |
| Manager | Person/s who manages the portfolios of the consumers and provides recommendations to the consumer and publishes new market trends. |
| Blockchain | A blockchain is a digital ledger that records transactions across many computers. It is a decentralized, distributed, and often public database |
| Minting | Creating a unique token on a blockchain |
| Auction | An event where users can bid and purchase an NFT that is on auction |
| Consumer | A person that uses the software to buy or sell NFTs and also be a part of a community with rights to publish articles and provide reviews and feedback. |
| Review | A written recommendation about the appropriateness of an article for publication; may include suggestions for improvement. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |

## References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## Overview of Document

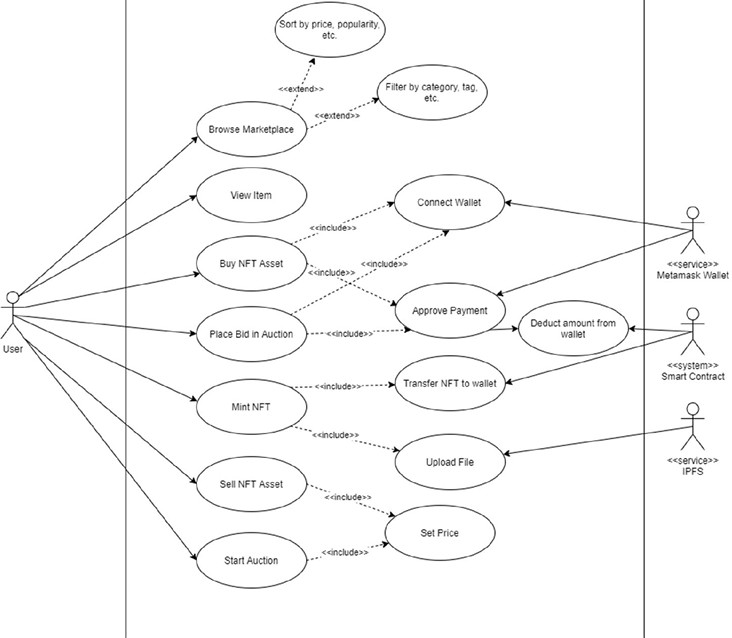
The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# 2.0. Overall Description

## System Environment



**Figure 1 - System Environment**

A Non-Fungible Token is essentially a digital certificate of authenticity that cannot be copied. NFTs are used to represent ownership of unique items and are stored on a blockchain or distributed ledger. Because of the secure properties of blockchain technology, the record of ownership is always available, immutable, and guarantees that only one owner can exist at any given time. NFTs, which are based on the blockchain programmable infrastructure, can represent anything that is both physical and digital in a digital realm. It is critical to recognize that the metadata stored within the Token's smart contract ensures uniqueness and, together with the transparency of the transaction history

provided by the underlying technology, allows the NFT to be verifiable by any network participant, reducing the risk of counterfeiting to zero.

## Functional Requirements Specification

This section outlines the use cases for each of the active consumers separately.

The consumer has the power to browse the marketplace, buy and sell NFTs, place and bid on auctions and mint NFTs.

* + 1. Consumer Use Case

Use case: **Browse marketplace Diagram:**



Browser Marketplace

Consumer

**Brief Description**

The consumer has access to the marketplace, where he/she can view all the NFTs that are available to purchase. He/she can sort the marketplace by price, popularity and filter by category, tag etc.

* + 1. Consumer Use Case

Here the consumer can mint the NFTs to certify the NFT.

Use case: **Minting NFT**

**Diagram:**



Minting NFT

Consumer

**Brief Description**

The consumer is requesting to mint an NFT.

**Initial Step-By-Step Description**

Before this use case can be initiated, the consumer is already connected to the NFT marketplace software.

1. The consumer chooses the *mint* button.
2. The System sends a request to mint the NFT.
3. The System generates the request and grants the consumer to upload an NFT or transfer it to the wallet.
4. The System generates and sends an email acknowledgement.
   * 1. Consumer Use Case Use case: Auctioning NFTs

**Brief Description**

The consumer is able to place an auction and bid on an NFT.

**Diagram:**

Resell Review



Auctioned



NFTs

Place bid Purchase

**Figure 2 - Auction Process**

**Brief Description:**

The consumer has the power to place bid in an auction, start an auction by reselling the NFT asset and ultimately storing the NFT asset in the wallet.

* + 1. Consumer Use Cases

Use case: Sell NFT

**Diagram:**



Sell NFT Asset

Consumer

<service> Crypto wallet

**Brief Description:**

The consumer can sell the NFT and set the price at which the consumer wants to sell that asset.

* + 1. **Consumer Use Cases**

Use case: Buy NFT

**Diagram:**



Buy NFT Asset



Consumer

<service> Crypto wallet

**Brief Description**:

The consumer can buy an NFT asset by connecting their wallet and then the system approves of the payment and the amount is deducted. After which the purchased NFT is securely transferred into the wallet.

* + 1. **Consumer Use Cases**

Use case: Track history of NFT purchases



History of purchases

Consumer

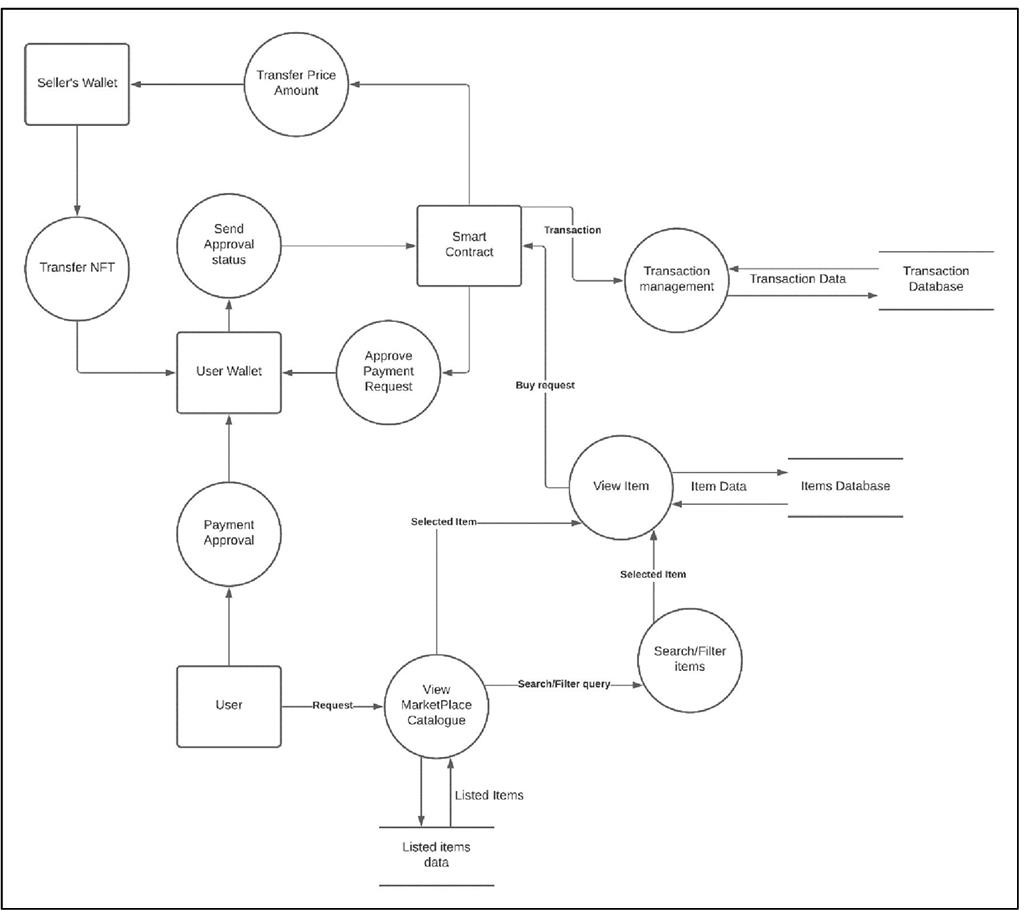
**Brief Description:**

The consumer can track their previous purchases and see all the purchase history from their account to track order.

# Requirements Specification

## External Interface Requirements

* + - API for connecting crypto wallets: The marketplace must have an API that allows users to connect their crypto wallets to the marketplace. This API must support a variety of crypto wallets and must be secure.
    - API for buying and selling NFTs: The marketplace must have an API that allows users to buy and sell NFTs. This API must be able to handle a large number of concurrent transactions and must be reliable.
    - API for retrieving NFT data: The marketplace must have an API that allows users to retrieve data about NFTs, such as their price, description, and creator. This API must be performant and must be able to handle a large number of concurrent requests.
    - Support for Web3 protocols: The marketplace must support Web3 protocols, such as Ethereum and MetaMask. This will allow users to interact with the marketplace using their crypto wallets.
    - Support for standard data formats: The marketplace must support standard data formats for NFTs, such as ERC-721 and ERC-1155. This will make it easier for users to interact with the marketplace and will make it compatible with other NFT marketplaces and wallets.



**Figure 2: Data flow diagram**

* 1. ***Functional Requirements***

FR1: Users must be able to connect their crypto wallets to the marketplace.

* + - Users must be able to connect their crypto wallets to the marketplace using a variety of methods, such as MetaMask.
    - The marketplace must support a variety of crypto wallets, including Ethereum wallets, Solana wallets, and Polygon wallets.
    - The marketplace must securely store users' crypto wallet information and protect it from unauthorized access

FR2: Users must be able to browse NFTs by category and keyword.

* + - The marketplace must allow users to browse NFTs by category, such as art, collectibles, and gaming items.
    - The marketplace must allow users to search for NFTs by keyword, such as creator name, NFT name, and NFT description.
    - The marketplace must provide users with a variety of filter options to help them find the NFTs they are interested in.

FR3: Users must be able to view detailed information about NFTs, including their price, description, and creator.

* + - The marketplace must provide users with detailed information about each NFT, including its price, description, creator, and blockchain network.
    - The marketplace must allow users to view the history of an NFT, including its ownership history and sale history.
    - The marketplace must allow users to view the metadata of an NFT, including its token ID and contract address.

FR4: Users must be able to purchase NFTs using their crypto wallets.

* + - The marketplace must allow users to purchase NFTs using their crypto wallets in a secure and efficient way.
    - The marketplace must provide users with a variety of payment options, such as fixed-price offers and auctions.
    - The marketplace must provide users with a confirmation screen before they complete a purchase.

FR5: Users must be able to track their NFT purchases in their account history.

* + - The marketplace must provide users with a detailed history of their NFT purchases.
    - The marketplace must allow users to view the status of their NFT purchases, such as whether the purchase is pending, complete, or cancelled.
    - The marketplace must allow users to download a copy of their NFT purchase history

FR6: Users must be able to create listings for their own NFTs to sell.

* + - The marketplace must allow users to create listings for their own NFTs to sell.
    - The marketplace must provide users with a variety of options for creating listings, such as setting a fixed price or creating an auction.
    - The marketplace must provide users with a preview of their listing before they publish it.

FR7: Sellers must be able to set prices for their NFTs and choose whether to accept fixed- price offers or auction-style bids.

* + - Sellers must be able to set prices for their NFTs in a variety of currencies, such as ETH, SOL, and USDC.
    - Sellers must be able to choose whether to accept fixed-price offers or auction- style bids.
    - Sellers must be able to set a minimum bid price for their auctions.

FR8: Sellers must be able to view the status of their listings and accept or decline offers and bids.

* + - Sellers must be able to view the status of their listings, such as whether the listing is active, pending, or sold.
    - Sellers must be able to accept or decline offers and bids on their listings.
    - Sellers must be able to cancel their listings at any time.

FR9: Buyers must be able to make offers on NFTs and place bids in auctions.

* + - Buyers must be able to make offers on NFTs at any time.
    - Buyers must be able to place bids in auctions.
    - Buyers must be able to cancel their offers and bids at any time before they are accepted.

FR10: Buyers must be able to view the status of their offers and bids.

* + - Buyers must be able to view the status of their offers and bids, such as whether the offer or bid is pending, accepted, or declined.

FR11: The marketplace must support multiple blockchain networks and crypto wallets.

* + - The marketplace must support a variety of blockchain networks, such as Ethereum, Solana, and Polygon.
    - The marketplace must support a variety of crypto wallets, including Ethereum wallets, Solana wallets, and Polygon wallets.

FR12: The marketplace must have a secure transaction system to protect user funds and NFTs.

* + - The marketplace must use a secure transaction system to protect user funds and NFTs from unauthorized access and theft.
    - The marketplace must implement industry-standard security practices, such as two-factor authentication and encryption.
    - The marketplace must have a disaster recovery plan in place to protect user data and NFTs in the event of a security breach or other event.
  1. ***Detailed Non-Functional Requirements***
  2. **Performance Requirements**
     1. **Response Time**
        + Requirement: The system should provide responses to user actions within 3 seconds under normal operating conditions.
        + Rationale: To ensure a responsive and seamless user experience.
     2. **Scalability**
        + Requirement: The system must be able to handle a simultaneous load of at least 10,000 users.
        + Rationale: To accommodate potential growth and handle peak usage periods.
     3. **Throughput**
        + Requirement: The system should support a minimum of 100 transactions per minute.
        + Rationale: To ensure efficient handling of transactional activities.
     4. **Resource Utilization**
        + Requirement: The system should not utilize more than 80% of available server resources under normal load conditions.
        + Rationale: To prevent resource exhaustion and maintain system stability.
  3. **Security Requirements**
     1. **Data Encryption**
        + Requirement: All sensitive user data, including passwords and wallet information, must be stored and transmitted using encryption algorithms.
        + Rationale: To protect user information from unauthorized access.
     2. **Authentication**
        + Requirement: User authentication must be secure and include multi-factor authentication options.
        + Rationale: To prevent unauthorized access to user accounts.
     3. **Authorization**
        + Requirement: Access controls must be implemented to ensure that users can only perform actions based on their roles and permissions.
        + Rationale: To protect against unauthorized operations.
     4. **Smart Contract Security**
        + Requirement: Smart contracts, if used, must undergo thorough security audits to identify and mitigate vulnerabilities.
        + Rationale: To ensure the integrity and security of blockchain-based transactions.
     5. **Compliance**
        + Requirement: The platform must comply with relevant data protection regulations and legal requirements.
        + Rationale: To avoid legal issues and maintain trust with users.
  4. **Usability Requirements**
     1. **User Interface Consistency**
        + Requirement: The user interface must maintain a consistent look and feel throughout the platform.
        + Rationale: To enhance user experience and ease of navigation.
     2. **Responsiveness**
        + Requirement: The platform must be responsive and usable on various devices, including desktops, tablets, and mobile phones.
        + Rationale: To accommodate users accessing the platform from different devices.
     3. **Accessibility**
        + Requirement: The platform must adhere to accessibility standards, ensuring it is usable by individuals with disabilities.
        + Rationale: To provide an inclusive user experience.
     4. **Internationalization**
        + Requirement: The platform must support multiple languages and currencies.
        + Rationale: To cater to a diverse user base.
  5. **Reliability and Availability**
     1. **System Uptime**
        + Requirement: The system should have an uptime of at least 99.9%.
        + Rationale: To ensure the platform is available for users consistently.
     2. **Disaster Recovery**
        + Requirement: A robust disaster recovery plan must be in place to minimize downtime in case of system failures or data breaches.
        + Rationale: To protect against data loss and maintain continuity.
     3. **Backup**
        + Requirement: Regular backups of user data and transaction records must be performed, and the backup process should be automated.
        + Rationale: To facilitate data recovery in case of system failures

# Design specifications

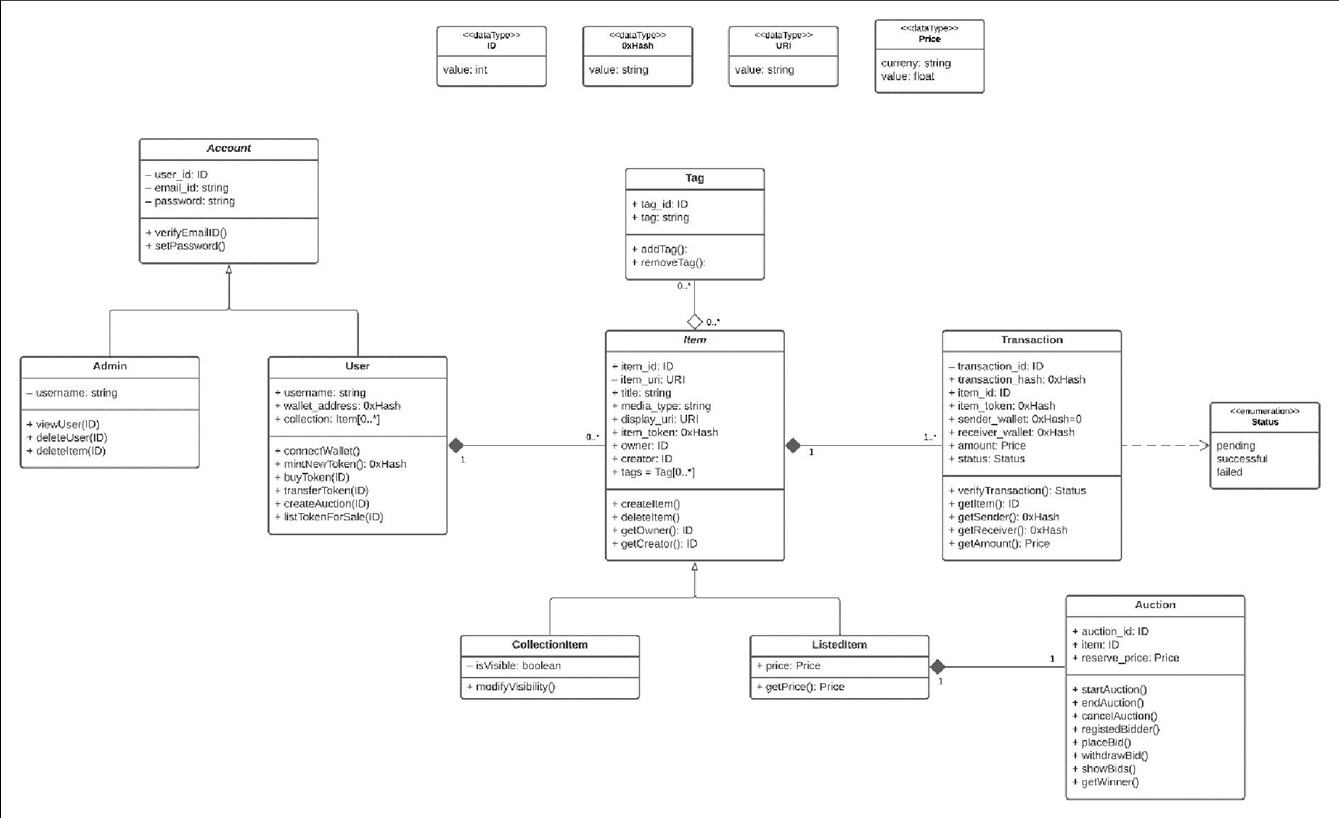
## System Overview:

The NFT marketplace is a web-based application that allows users to buy, sell, and manage NFTs. The marketplace supports multiple blockchain networks and crypto wallets.

## Components:

The NFT marketplace consists of the following components:

* User authentication and authorization component: This component authenticates users and authorizes them to perform certain actions, such as buying, selling, and managing NFTs.
* NFT listing and management component: This component allows users to list their NFTs for sale and manage their existing listings.
* NFT purchase and bidding component: This component allows users to purchase NFTs and place bids on NFTs in auctions.
* Blockchain interaction component: This component interacts with blockchain networks to process transactions and manage NFT ownership.
* NFT metadata management component: This component manages NFT metadata, such as token ID, contract address, and image URL.



**Figure 3: Data class diagram**

## Data Flow:

The following is a high-level overview of the data flow in the NFT marketplace:

* + 1. A user connects their crypto wallet to the marketplace.
    2. The user browses the marketplace for NFTs to buy.
    3. The user selects an NFT and places an offer or a bid.
    4. The marketplace verifies the offer or bid and processes the transaction.
    5. The NFT is transferred to the buyer's wallet.

# 5.0. Testing Strategy

## Test Levels:

* + - Unit Testing:
      * Test individual components modules, and functions in isolation.
    - Integration Testing:
      * Verify the interaction between different components and services.
    - System Testing:
      * Validate the system as a whole, including end-to-end scenarios.
    - User Acceptance Testing (UAT):
      * Collaborate with end-users to validate the software against real-world scenarios.

## Test Types:

* + - Functional Testing:
      * Validate functional requirements, including user registration, NFT listing, buying, and selling.
    - Non-Functional Testing:
      * Performance Testing: Evaluate system response times, throughput, and scalability.
      * Security Testing: Identify and address security vulnerabilities.
      * Usability Testing: Assess user interface design and overall user experience.
    - Regression Testing:
      * Ensure that new changes do not break existing functionality.

## Test Environments:

* + - Development Environment:
      * For unit testing and initial component testing.
    - Staging Environment:
      * For integration and system testing.
    - Production Environment:
      * For User Acceptance Testing (UAT) and final validation.

## Test Data:

* + - Define test data sets for various scenarios, including valid and invalid inputs.

## Test Cases and Scenarios:

## Unit Testing:

* + - * Objective: Validate individual components and functions.
      * Criteria: All unit tests should pass without errors.

## Integration Testing:

* + - * Objective: Verify the interaction between different components and services.
      * Criteria: Data should flow correctly between modules, and integrations should work seamlessly.

## System Testing:

* + - * Objective: Validate end-to-end scenarios, including user registration, NFT listing, buying, and selling.
      * Criteria: All critical user journeys should function correctly.

## User Acceptance Testing (UAT):

* + - * Objective: Validate the software against real-world use cases.
      * Criteria: End-users should successfully complete common tasks, and feedback should be collected and addressed.

## Non-Functional Testing:

* + - * Performance Testing:
        + Objective: Validate system response times, throughput, and scalability.
        + Criteria: Response times should meet performance requirements, and the system should handle the expected load.
      * Security Testing:
        + Objective: Identify and address security vulnerabilities.
        + Criteria: No critical security vulnerabilities should be present.
      * Usability Testing:
        + Objective: Assess user interface design and overall user experience.
        + Criteria: The interface should be intuitive and user-friendly.

# Conclusion:

This project document has outlined the detailed requirements, design, and implementation plan for the NFT marketplace. The marketplace will be a user-friendly and secure platform for users to buy, sell, and manage NFTs. The marketplace will support multiple blockchain networks and crypto wallets, and it will be scalable to accommodate future growth.

The marketplace is expected to have a significant impact on the NFT industry by making it easier for people to participate in the market. The marketplace will also provide a valuable platform for NFT creators to showcase and sell their work.

# References:

* + 1. Mohapatra, H., & Rath, A. K. (2020). Fundamentals of software engineering: designed to provide an insight into the software engineering concepts. BPB Publications.