Software Requirements Specification

for

Listen NFT

Version 1.0

Prepared by

Avaneesh Singh - 2000290120047
Nishant Varshney - 2000290120102
Harsh Kumar - 2000290120072

KIET Group of Institutions, Ghaziabad

February 2023

Table of Contents

Revision History

Name	Date	Version
ListenNFT	18th February 2023	1.0
ListenNFT	15th April 2023	1.1

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the web 3.0 app ListenNFT. It will explain the purpose and features of the software, the interfaces of the software, what the software will do, and the constraints under which it must operate. This document is intended for users of the software and also potential developers.

1.2 Document Conventions

This document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- Typical Users, such as music artists, who want to sell their albums, songs, etc.
- Advanced/Professional Publishers, such as Sony or T-series, who want to use the platform for more demanding buying and selling.
- Programmers who are interested in selling or buying music albums in a decentralized way.

1.4 Product Scope

ListenNFT is a WEB 3.0 music platform that makes the creator and user an ad-free platform. Any music artist can mint their fabulous art on the platform and showcase it to the world with any intervention of Music Studio.

The main objective to create this Dapp is that artists don't get the amount, importance, and fame they deserve. A huge amount is slashed by big music firms like T-Series, Sony Music, etc.

1.5 References

Listen NFT website: https://listen-nft.vercel.app/

Listen NFT's Github page: https://github.com/avi-11/listenNFT

IEEE Template for System Requirement Specification Documents:
https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

2. Overall Description

2.1 Product Perspective

ListenNFT is a WEB 3.0 music platform that makes the creator and user an ad-free platform. Any music artist can mint their fabulous art on the platform and showcase to the world with any intervention of Music Studio.

The main objective of creating this D app is that artists don't get the amount, importance and fame they deserve. A huge amount is slashed by the big music firm like T-Series, Sony Music etc.

An NFT is a digital asset that represents real-world objects like art, music, in-game items and videos. They are bought and sold online, frequently with cryptocurrency, and they are generally encoded with the same underlying software as many cryptos.
NFTs are also generally one of a kind, or at least one of a very limited run, and have unique identifying codes.

By this Creator become the owner of their work and any User can buy the music to make it exclusive in exchange for payment to the owner.

2.2 Product Functions

Product functions are as follows:

- Add: one can add their songs/album.
- Upload: one can upload their song/album on the decentralized chain.
- Mint: one can mint the other's songs and albums.
- Save: one can save added songs/albums.
- Basket: A place where saved albums are stored.
- Metamask: wallet used for minting NFTs.
- NFT: non-fungible tokens for minting.
- Play: to play the songs/albums.
- Pause: to pause the ongoing songs/albums.
- Fast forward: to increase the playback speed.
- Slow: to reduce the playback speed.
- Check for Updates: Displays the plugins that can be updated to newer versions
- About: Displays the logo of Gephi, which licenses are being used, the product version, and other info

2.3 User Classes and Characteristics

- Typical Users, such as music artists, who want to sell their albums, songs, etc.
- Advanced/Professional Publishers, such as Sony or Tseries, who want to use the platform for more demanding buying and selling.
- Programmers who are interested in selling or buying music albums in a decentralized way.

2.4 Operating Environment

- Windows 2000
- Windows XP
- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Mac OS X
- Linux

2.5 Design and Implementation Constraints

Listen NFT is developed in the MERN stack along with smart contracts in Solidity, it uses Ether.js for the integration of smart contracts with the website. It uses a modular design where every feature is wrapped into a separate module and the modules depend on each other through well-written APIs. There are several APIs available to make plugin development easy

2.6 User Documentation

There is a quick start guide available on the website of ListenNFT:

https://listen-nft.vercel.app/

2.7 Assumptions and Dependencies

Gephi is developed in MERN and therefore requires the node to be installed on the user's system. The latest stable version of Listen NFT requires node version 16.0.1 or higher. This applies to Windows and Linux users. On Mac OS X, it requires a node as well as a package manager.

3. External Interface Requirements

3.1 User Interfaces

1) Home page: The homepage should provide users with a clear and concise overview of the marketplace, including featured NFTs, a search bar, categories, and any promotions or events.
2) NFT listings: The marketplace should allow users to view detailed information about each NFT, such as images, descriptions, creator information, and price.
3) Search and filter: The marketplace should provide users with powerful search and filter options, allowing them to quickly find the NFTs that match their criteria.
4) User profile: Users should be able to manage their profile information, view their transaction history, and access their wallets.
5) Transaction management: The marketplace should provide users with a streamlined and secure process for buying and selling NFTs, including options to make offers, accept bids, and complete transactions.
6) Notifications: The marketplace should notify users about any important updates, such as new NFT listings, offers, or transaction confirmations.
7) Community and social features: The marketplace should enable users to connect with each other, share feedback and reviews, and participate in community events.
8) Help and support: The marketplace should provide users with easy access to help and support resources, such as FAQs, user guides, and customer service.

3.2 Hardware Interfaces

1) Server: The server is the backbone of the NFT marketplace, which handles all the requests made by the clients. It should be powerful enough to handle the traffic and process all the transactions in real-time.

2) Storage: Since NFTs are digital assets, they require a storage solution that can accommodate large amounts of data. The storage solution should be secure and provide easy access to the NFTs.

3) Network: The network infrastructure should be able to handle the volume of traffic generated by the NFT marketplace. It should also provide secure connectivity between the clients and the server.

4) Payment gateway: The payment gateway is a critical component of the NFT marketplace, which allows buyers and sellers to transact securely. It should support multiple payment options, including cryptocurrencies, credit/debit cards, and other payment methods.

5) User interface: The user interface is the front-end of the NFT marketplace, which allows users to browse, buy, and sell NFTs. It should be user-friendly, intuitive, and visually appealing.

6) Hardware security: The NFT marketplace should also have robust hardware security measures in place to prevent unauthorized access, theft, or hacking. This includes measures such as encryption, firewalls, and other security protocols.

3.3 Software Interfaces

1) OpenSea: OpenSea is a decentralized marketplace that supports the buying and selling of NFTs. It allows creators to create and manage their NFTs, and buyers to browse and purchase them. OpenSea offers an API that developers can use to create their own marketplace interfaces.

2) SuperRare: SuperRare is a curated NFT marketplace that specializes in high-quality, unique digital artwork. It offers a web interface that allows buyers to browse and purchase NFTs, as well as a mobile app for iOS devices.

3) Rarible: Rarible is an open marketplace for buying and selling NFTs. It allows creators to mint their own NFTs, and buyers to browse and purchase them. Rarible offers an API that developers can use to create their own marketplace interfaces.

4) Nifty Gateway: Nifty Gateway is a marketplace for buying and selling limited-edition NFTs created by artists and musicians. It offers a web interface that allows buyers to browse and purchase NFTs, as well as a mobile app for iOS and Android devices.

5) Mintable: Mintable is a marketplace for buying and selling NFTs, with a focus on digital art, music, and collectibles. It offers a web interface that allows creators to mint their own NFTs, and buyers to browse and purchase them.

3.4 Communications Interfaces

1) Email: Email is a common and convenient way to communicate with users of an NFT marketplace. It can be used for various purposes, such as sending newsletters, marketing promotions, notifications about new NFT listings, or providing customer support.

2) Chat: Chat applications, such as Slack or Discord, can be useful for creating communities around an NFT marketplace. They can be used to discuss NFTs, provide support, or facilitate transactions.

3) Social media: Social media platforms like Twitter, Instagram, and TikTok are great for marketing and promoting NFTs, building brand awareness, and reaching potential buyers.

4) Forums: Forums like Reddit or Bitcointalk can be used for discussions about the NFT marketplace and its offerings, as well as for promoting new NFTs or providing customer support.

5) Help Desk: Helpdesk software, like Zendesk or Freshdesk, can be used to provide customer support through email, chat, or a ticketing system. This can help ensure that users' questions and concerns are addressed in a timely and organized manner.

4. System Features

1) User registration and authentication: The NFT marketplace will require users to register and authenticate their accounts before they can buy or sell NFTs. This will involve providing personal information and creating login credentials.

2) Wallet integration: Users will need to connect their digital wallets to the NFT marketplace in order to buy and sell NFTs. This will allow them to store and manage their digital assets securely.

3) NFT creation and listing: The NFT marketplace will allow creators to upload their digital assets and create NFTs. They can set the price and listing details for each NFT they create.

4) NFT search and discovery: Users can search for NFTs by category, creator, price, and other parameters. This will allow them to discover new digital assets and find the ones that fit their interests.

5) Bid and offer system: Users can make offers or bid on NFTs that are listed for sale. This will enable them to negotiate the price and potentially acquire an NFT at a lower price.

6) Transaction management: The NFT marketplace will handle the transaction process, including payment processing, escrow services, and delivery of the NFT to the buyer.

7) Analytics and reporting: The NFT marketplace will provide analytics and reporting features to help users track their sales and performance. This will allow them to optimize their sales strategies and make data-driven decisions.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Scalability: The marketplace should be able to handle a large volume of transactions without compromising its performance. It should be able to accommodate a large number of users and handle a high volume of traffic.

Speed: The marketplace should be fast and responsive. Transactions should be processed quickly, and users should not have to wait long periods for their transactions to be confirmed.

Security: Security is crucial in any blockchain-based system, and an NFT marketplace is no exception. The platform should have robust security measures in place to protect users' assets and prevent unauthorized access.

User-friendly interface: The marketplace should be easy to use and navigate. The interface should be intuitive and user-friendly, allowing users to quickly find the assets they are looking for and complete transactions.

Low transaction fees: The transaction fees should be reasonable and competitive compared to other NFT marketplaces. High transaction fees can discourage users from using the platform.

Interoperability: The marketplace should be compatible with multiple blockchain networks, allowing users to trade assets across different blockchains.

Reliability: The marketplace should be available and accessible at all times. Downtime or disruptions can result in lost opportunities for users, and damage the reputation of the platform.

5.2 Safety Requirements

Secure storage of NFTs: NFTs are unique digital assets that represent ownership of a specific item. Therefore, the marketplace should have robust security measures in place to store and protect NFTs from theft, loss, or damage. This includes implementing encryption, multi-factor authentication, and secure data storage mechanisms.

Protection against hacking attacks: The marketplace should have strong security protocols in place to protect against hacking attempts and other cyber threats. This includes regular security audits, firewalls, intrusion detection, and prevention systems.

Identity verification and fraud prevention: The marketplace should implement robust identity verification measures to prevent fraud, money laundering, and other illegal activities. This includes Know Your Customer (KYC) protocols, anti-money laundering (AML) policies, and fraud detection algorithms.

Regulatory compliance: The marketplace should adhere to relevant regulatory requirements in the jurisdiction where it operates. This includes compliance with laws and regulations related to data privacy, consumer protection, and financial transactions.

Transparent and fair transaction processes: The marketplace should ensure that all transactions are transparent and fair. This includes providing users with clear and concise information about fees, charges, and other costs associated with buying, selling, or trading NFTs.

Disaster recovery and business continuity planning: The marketplace should have a disaster recovery plan in place to ensure that it can continue operating in the event of a disaster or other business interruption. This includes regular backups, redundancy, and contingency plans.

5.3 Security Requirements

Secure access control: The marketplace should implement secure access control mechanisms to ensure that only authorized users can access the platform. This includes implementing multi-factor authentication, password policies, and session timeouts.

Encryption: The marketplace should use encryption to protect user data, including NFTs and personal information. This includes encryption of data at rest and in transit, using industry-standard encryption protocols.

Secure storage of private keys: The marketplace should provide users with secure storage of their private keys, which are used to access and transfer NFTs. This includes implementing hardware wallets, cold storage solutions, and multi-signature wallets.

Immutable record-keeping: The marketplace should ensure that all NFT transactions are recorded on a blockchain ledger, which is immutable and tamper-proof. This provides a transparent and secure record of all transactions.

Regular security audits: The marketplace should conduct regular security audits to identify and mitigate security vulnerabilities. This includes vulnerability scanning, penetration testing, and regular code reviews.

Disaster recovery and business continuity planning: The marketplace should have a disaster recovery plan in place to ensure that it can continue operating in the event of a disaster or other business interruption. This includes regular backups, redundancy, and contingency plans.

Security policies and procedures: The marketplace should establish and adhere to comprehensive security policies and procedures to ensure that all security requirements are met consistently. This includes incident response plans, security incident reporting, and regular security awareness training for employees.

5.4 Software Quality Attributes

Scalability: An NFT marketplace should be scalable to handle a large number of users, transactions, and NFT assets. This includes the ability to scale horizontally by adding more servers, and vertically by increasing processing power and memory.

Reliability: An NFT marketplace should be reliable and available to users at all times. This includes minimizing downtime, providing backup and disaster recovery mechanisms, and ensuring data integrity.

Usability: An NFT marketplace should be easy to use and intuitive for users of all skill levels. This includes providing a user-friendly interface, clear navigation, and informative feedback to users.

Performance: An NFT marketplace should have fast response times and low latency to provide a smooth user experience. This includes optimizing code, reducing network latency, and caching frequently accessed data.

Security: An NFT marketplace should be secure to protect user data, including NFTs and personal information. This includes implementing encryption, secure access control, and secure storage of private keys.

Maintainability: An NFT marketplace should be easy to maintain and update. This includes using clean and modular code, providing the comprehensive documentation, and using version control systems.

Interoperability: An NFT marketplace should be interoperable with other systems and services, including blockchain networks and payment gateways. This includes using open standards and APIs to enable data exchange and integration.

5.5 Business Rules

Listing policies: The marketplace should have clear policies on what types of NFTs can be listed for sale, including guidelines on content, copyright infringement, and intellectual property.

Pricing policies: The marketplace should have clear policies on pricing, including minimum and maximum listing prices, transaction fees, and any other costs associated with buying or selling NFTs.

Payment policies: The marketplace should have clear policies on payment methods, including accepted currencies, payment processing times, and any other fees or charges associated with payments.

Dispute resolution policies: The marketplace should have clear policies on how to resolve disputes between buyers and sellers, including mechanisms for mediation, arbitration, and legal action if necessary.

Privacy policies: The marketplace should have clear policies on how user data is collected, stored, and used, including any third-party data sharing or advertising policies.

Compliance policies: The marketplace should have clear policies on compliance with relevant laws and regulations, including anti-money laundering (AML) and know-your-customer (KYC) policies.

Intellectual property policies: The marketplace should have clear policies on how to handle disputes related to intellectual property, including copyright, trademark, and patent infringement.

6. Other Requirements

Secure Wallet Integration: NFT marketplaces should have secure wallet integration to allow users to store, manage, and trade their NFTs securely. Wallet integration must be compliant with the highest security standards to ensure the safety of users' assets.

User-Friendly Interface: NFT marketplaces should have a user-friendly interface that is easy to navigate and understand. Users should be able to search, browse, and filter NFTs easily, and the overall design of the marketplace should be aesthetically pleasing.

Multiple Blockchain Support: NFT marketplaces should support multiple blockchain networks such as Ethereum, Binance Smart Chain, Polkadot, and others, to provide users with more options and flexibility.

Royalty Payments: NFT marketplaces should allow artists and creators to receive royalties for their artwork or collectibles when they are resold in the future. This feature provides creators with a source of passive income, incentivizes them to continue creating, and increases the overall value of the marketplace.

Low Transaction Fees: NFT marketplaces should have low transaction fees to encourage more users to use the platform. High transaction fees can deter users and may limit the growth of the marketplace.

Comprehensive Analytics: NFT marketplaces should provide comprehensive analytics to artists, collectors, and traders, allowing them to track their sales, understand their audience, and make data-driven decisions to grow their businesses.

Decentralized Infrastructure: NFT marketplaces should operate on a decentralized infrastructure to ensure that transactions are transparent, secure, and immutable. Decentralized marketplaces are less susceptible to hacks, fraud, or any other security breach.

Community Support: NFT marketplaces should foster an active and supportive community that encourages engagement, collaboration, and feedback. This community helps to build a vibrant ecosystem around the marketplace and promotes its growth.

Appendix A: Glossary

NFT: Non-fungible token is a unique digital asset that represents ownership of a specific item or asset, such as artwork, collectibles, or virtual real estate.

Marketplace: An online platform where NFTs are bought, sold, or traded by individuals.

Digital Wallet: A digital wallet is a secure online application or software that allows users to store, manage, and trade their NFTs.

Bid: A bid is an offer made by a buyer on an NFT, indicating the amount of cryptocurrency they are willing to pay for it.

Ask: An ask is a price that a seller is willing to accept for their NFT.

Reserve Price: The reserve price is the minimum price that a seller is willing to accept for their NFT. If the reserve price is not met, the NFT will not be sold.

Gas Fees: Gas fees are transaction fees paid by users to process transactions on a blockchain network.

Smart Contract: A smart contract is a self-executing contract with the terms of the agreement between buyer and seller directly written into lines of code. In the case of NFTs, smart contracts govern the sale and transfer of ownership of digital assets.

Royalties: Royalties are a percentage of the sale price that is paid to the creator of an NFT every time it is sold in the future.

Minting: Minting is the process of creating a new NFT on a blockchain network.

Verification: Verification is the process of confirming the authenticity and ownership of an NFT.

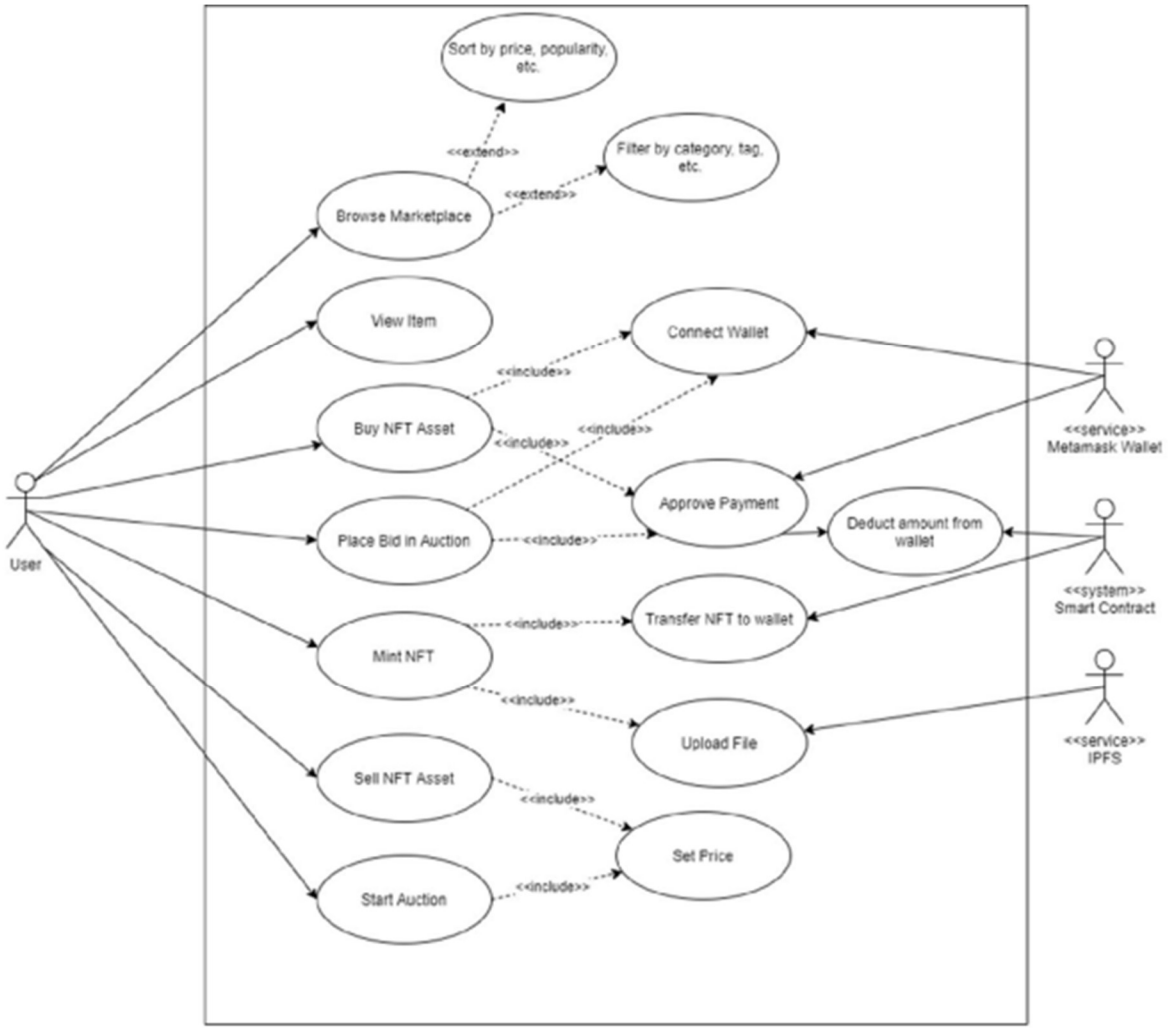
Liquidity: Liquidity refers to the ease with which an NFT can be bought or sold on an NFT marketplace.

Escrow: An escrow is a third-party service that holds funds in a transaction until the terms of the agreement have been met.

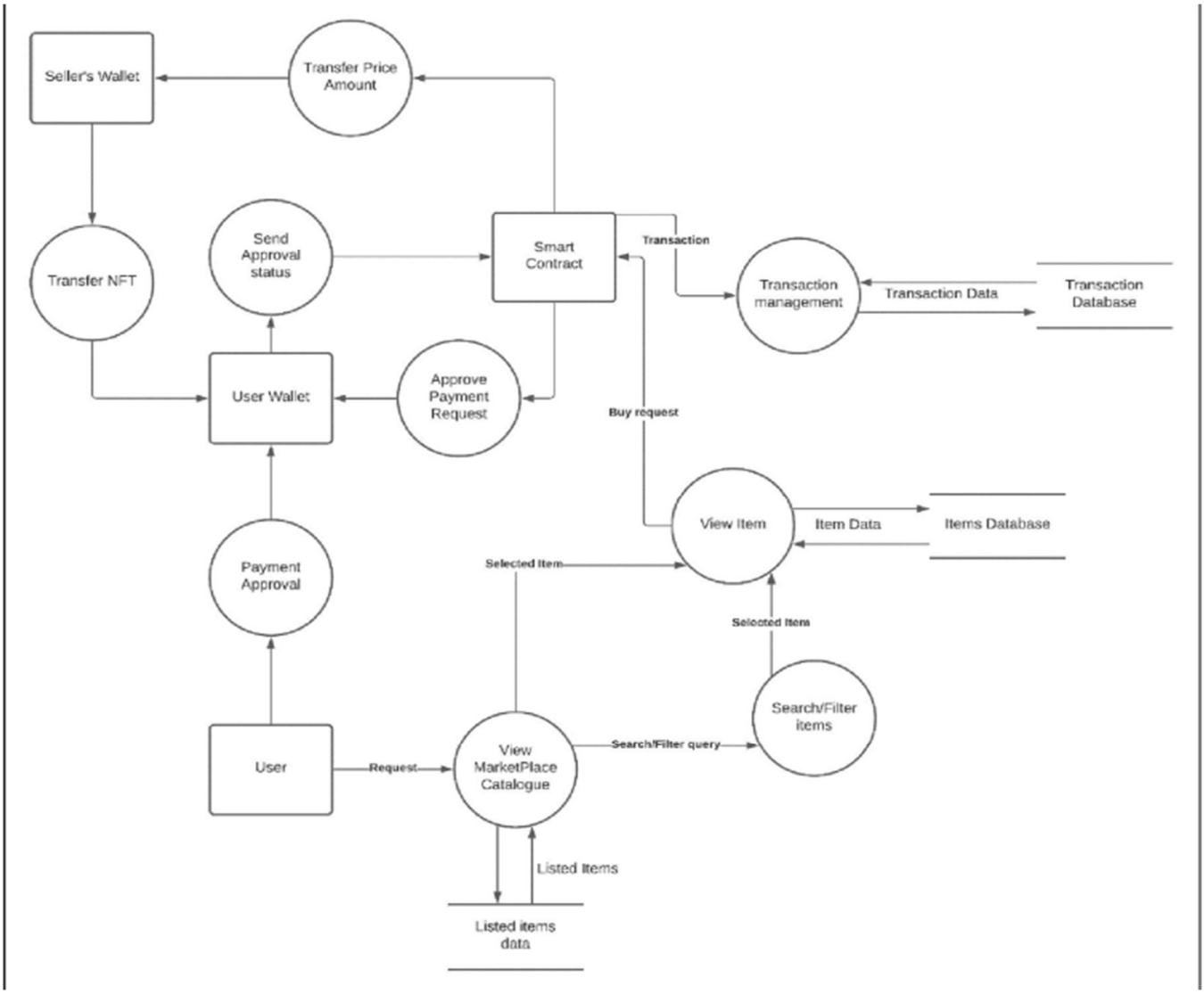
Tokenization: Tokenization is the process of converting a physical asset or asset ownership into a digital token.

Appendix B: Analysis Models

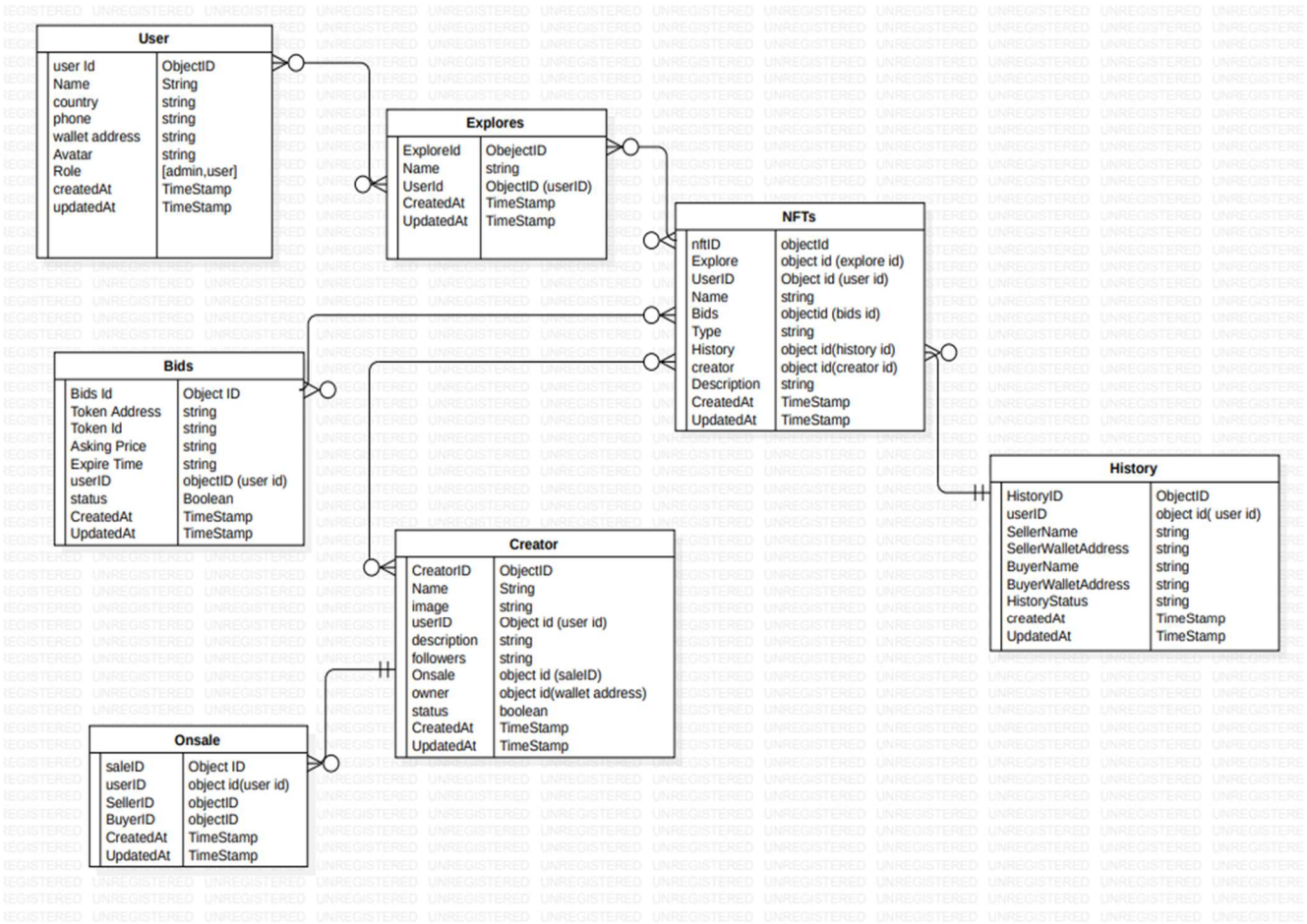
1) Use Case Diagram:



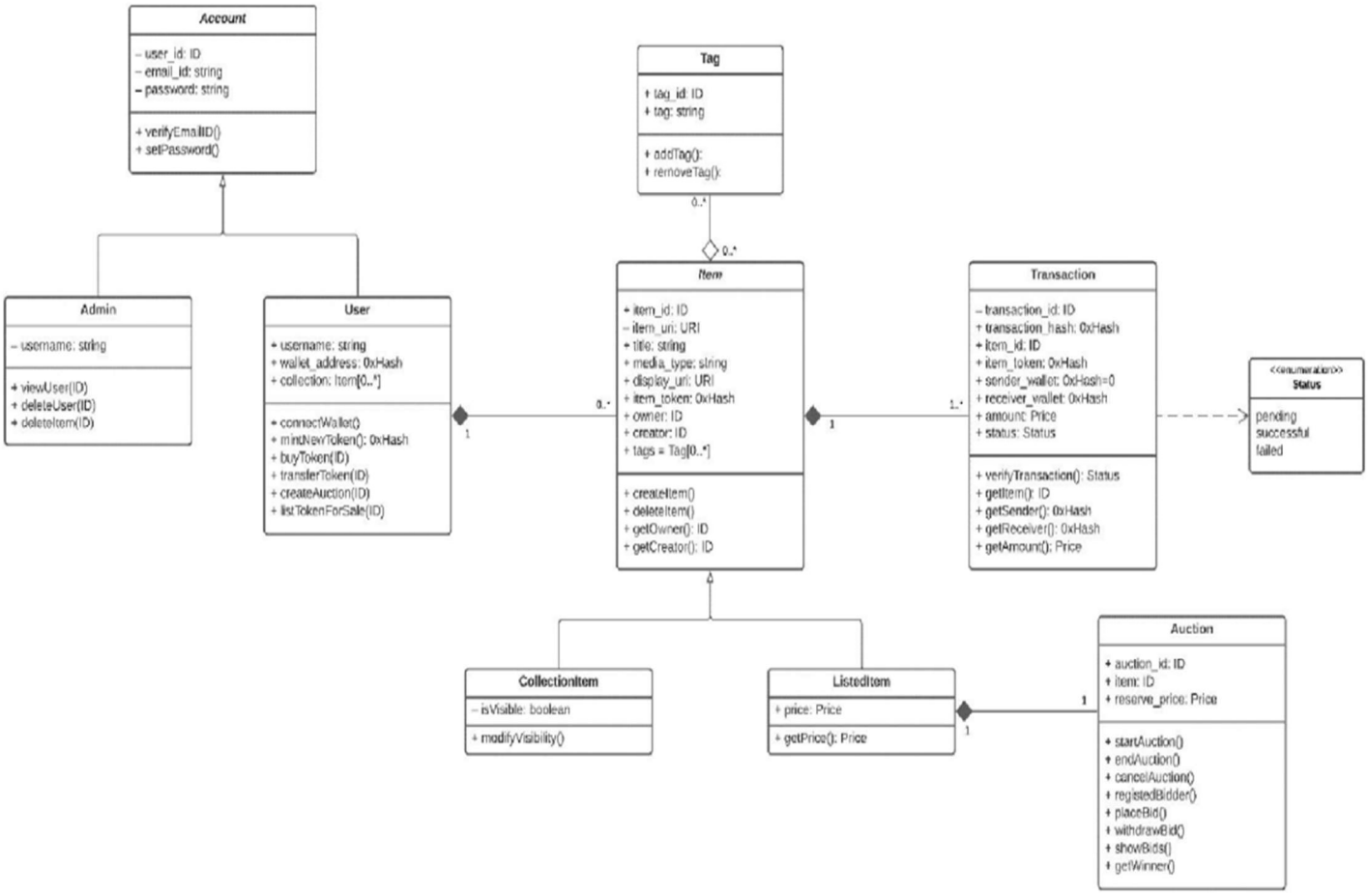
2) Data Flow Diagram :



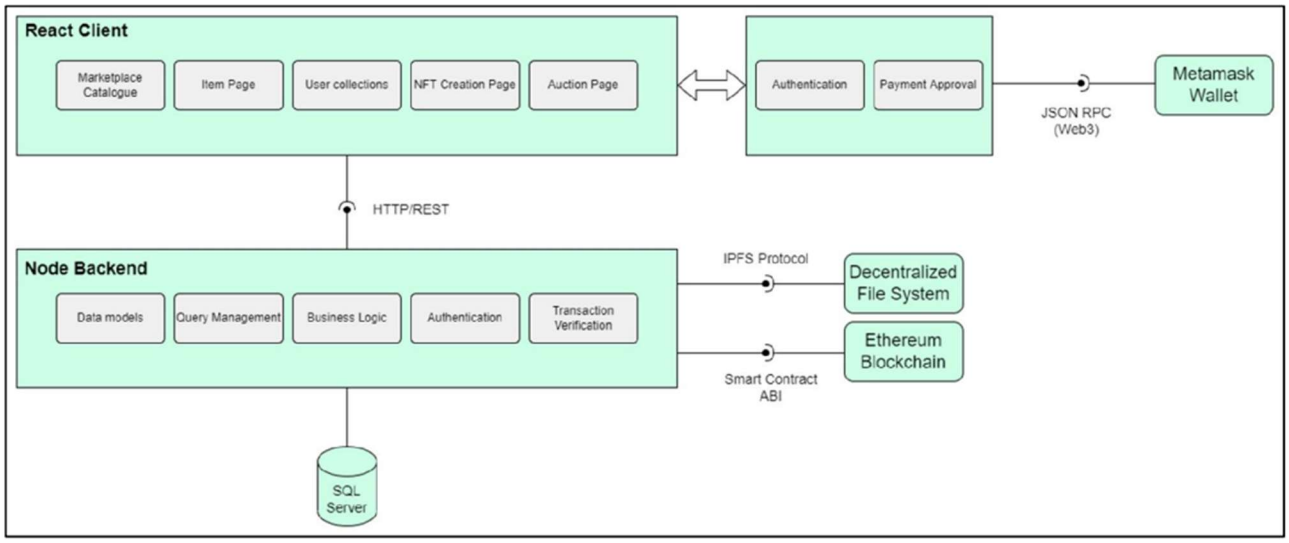
3) ER Diagram :


User
user Id ObjectID
Name String
country string
phone string
wallet address string
Avatar string
Role [admin,user]
createdAt TimeStamp
updatedAt TimeStamp
Explores
ExploreId ObejectID
Name string
UserId ObjectID (userID)
CreatedAt TimeStamp
UpdatedAt TimeStamp
NFTs
nftID objectId
Explore object id (explore id)
UserID Object id (user id)
Name string
Bids objectid (bids id)
Type string
History object id(history id)
creator object id(creator id)
Description string
CreatedAt TimeStamp
UpdatedAt TimeStamp
Bids
Bids Id Object ID
Token Address string
Token Id string
Asking Price string
Expire Time string
userID objectID (user id)
status Boolean
CreatedAt TimeStamp
UpdatedAt TimeStamp
History
HistoryID ObjectID
userID object id(user id)
SellerName string
SellerWalletAddress string
BuyerName string
BuyerWalletAddress string
HistoryStatus string
createdAt TimeStamp
UpdatedAt TimeStamp
Creator
CreatorID ObjectID
Name String
image string
userID Object id (user id)
description string
followers string
Onsale object id (saleID)
owner object id(wallet address)
status boolean
CreatedAt TimeStamp
UpdatedAt TimeStamp
Onsale
saleID Object ID
userID object id(user id)
SellerID objectID
BuyerID objectID
CreatedAt TimeStamp
UpdatedAt TimeStamp

4) Class Diagram



5) Architectural Diagram



6) Activity Diagram :