SOFTWARE REQUIREMENTS SPECIFICATION

for

NFT Sales Analytics Dashboard

Prepared by : 1.Jagrati Sharma (19ESKIT033)
2. Jitendra Prajapat (19ESKIT037)
3. Karan Sabnani (19ESKIT038)
4. Komal Soni (19ESKIT044)

Submitted to : Mrs. Anjali Pandey
(Assistant Prof.)

May 31,2023

Contents

Revision History

Revision	Date	Author(s)	Description
1.0			
2.0			
3.0			
4.0			

Chapter 1

Introduction

1.1 Purpose

The NFT market is booming now. According to NFTGO statistics, at least one NFT project has launched on chain every day since May 2021. At the same time, similar to DeFi, the quality of NFT projects is varied, and investors may easily fall into the "liquidity trap". Analytics dashboards and reports can be created to know more about NFT. By using analytics tools to identify patterns and discrepancies, traders, builders, and collectors can gain a competitive edge in a brand-new market. In fact, in the world of blockchain, whether as an investor or a beginner, you can rely on data to stay away from "hearsay" and decide which NFT to buy by yourself.

1.2 Scope

NFT Sales Analytics Dashboard typically support these functions.

- Follow Smart Money
- Keep Track of Latest Market Activity
- Find New Mints
- Track Specific Collections
- Check Rarity

1.3 Technologies to be used

1. **ReactJS :** ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based frontend library which is responsible only for the view layer of the application.
2. **NodeJS :** NodeJs is a cross platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications. It is open source and free to use.
3. **ExpressJS :** Express is a fast, assertive, essential and moderate web framework of NodeJS. Assume Express as a layer built on top of the NodeJS that helps manage a server and routes, It provides a robust set of features to develop web and mobile applications.

4. **MongoDB :** MongoDB is an open source document database that provides high performance, high availability, and automatic scaling.
5. **Cascading Style Sheets :** Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

1.4 Overview

The implementation of NFT Sales Analysis Dashboard starts with analysis of various nfts and collections. It provide an overview to the users about which nfts he/she should invest.

Chapter 2

Overall Description

2.1 Product Perspective

NFT Sales Analytics Dashboard is a project which requires some additional hardware and software interfaces to function which includes the OS, a web browser, a cryptocurrency wallet (Metamask), and a stable internet connection. When released, the final product would be the first version of the application. It will be designed as a user centered product, which could be accessed to give a personalized experience to any authenticated user.

2.2 Product Functions

The Market Overview NFT data analytics dashboard provides a rundown of the NFT collections and the fluctuations in their floor price. This allows investors to track trends within the NFT sphere and gauge the general sentiment.
This dashboard facilitates trend trading, where investors utilize Nansen's collection analytics to trade alongside momentum. The application will be capable of performing the following functions.

- Provide Authentication for both General User through Sign Up Page, Login Page and Metamask wallet login
- Provide each user with an account with an account and verified checkmark on his profile
- Users can see and change their personal information after visiting their dashboard.
- Library where users can see the Trending NFTs, Recently Added NFTs, with various functionalities like searching, sorting, filters on Genres, Lyrics, Instrument used etc. for quickly accessing the NFTs

2.3 Operating Environment

The product will be operating in a Windows environment. The NFT Sales Analytics Dashboard is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome, and Mozilla Firefox. Also, it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.
The hardware configuration includes Hard Disk: 40 GB, Monitor: 15" Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

2.4 Constraints

The project operates under a number of design and implementation constraints. Some ofthese are as outlined below:

Hardware and Software constraints:

Since the project has been developed entirely using React Js for frontend and firebase for backend, it is largely independent. The project can be run on any platform.

- There is the secondary memory, which is where your files are stored, and a computercan use a Hard Disk to store memory, or a Solid State Drive (As well as other kinds of Flash Memory). Modern Hard Disks (HDs) can store from around 500 GB to 2TBof data.
- There is also primary memory, which is the memory that stores information that youare manipulating with immediately, when the computer is ON.
- To better understand the difference between primary and secondary, let me give you an example: Suppose you want to edit a photo. Your photo is stored in your Secondary Memory. When you open up your photo in a photo editor, it is loaded onyour primary memory, so that you manipulate it. Then, you apply your favorite vintage filter to it, as well as colors and whatever you want. The edited photo is stillon your primary memory. Then, you are happy with the result, so you click on "Save", that's when your new version of the photo is copied back to your secondarymemory.
- Computers use CPU Memory registers, RAM (Random Access Memory), and other resources as Primary Memory. A modern laptop has around 8GB of RAM.
- Another big difference between Primary and Secondary Memory is that Primary memory is volatile. That means if your battery dies while your editing your photo,you'll lose all of your changes, because the Primary Memory can only keep information as long as there is energy. Secondary Memory, on the other hand, cankeep your photo for years, even if the computer is OFF.
- An interesting thing I'd like to point up is Swap Space. Let's suppose you have 25 tabs of YouTube videos open on your browser. All of those videos are being downloaded into your RAM memory. Let's suppose you have 2GB, and then all of the videos combined occupy 3GB of memory. You don't have enough memory to hold all of that! When the computer reaches it's RAM limit, it starts using "Swap Space", which is basically storing the excessive data into a temporary location on your HD. HDs are too slow to hold data we are momentarily using, that's why your computer seems sluggish after your open a lotof pages and apps.
- Another place full of constraints in hardware is the CPU (Central Processing Unit). It is where everything in the computer is calculated. Every core of a CPU can only do one operation at a time. Think of operation as something like 1- Get two numbers from memory, 2- Sum two numbers, 3- Show numbers on screen. It has a stack, which you can think of as a line of all the operations wanting to get in. It works with cycles, and with each cycle, one simple operation is done. The CPU Clock is a little device that sends electrical pulses to the CPU, and each pulse, a cycle.
- It seems quite straight forward then: Make the clock tick faster, so that we have more cycles per second, and work on all of the operations on the stack faster. There is a problem, however: You run the risk of overheating your processor, potentially melting down components and having a big headache. Many hardware hackers experiment with this, called "Over-clocking", and they generally use more expensive cooling systems, likeliquid cooling, faster fans, bigger heat dissipators, etc.
- An approach that has been widely used on the industry is to increase the number of "Cores" on a processor. You probably heard the terms dual-core, quad-core somewhere, since now even smartphones can be quad-core (four cores). So, that way, you multiply your processing power. That doesn't come without obstacles, of course. You need to carefully craft your software to work well with Parallel computing.

- One of the biggest challenges for computing today is the size of the transistor, a fundamental component of a processor. Transistors are getting so small, that Quantum Effects are starting to be relevant, as they will play out bigger interference with the functioning of the component. That's a very important constraint for today's industry.
- I'd like to briefly talk about I/O (Input/Output). I/O is basically communication across devices (AKA the thing behind USB stuff). Suppose you have just bought a super cool webcam, with 20 megapixels, 60 frames per second. Of course, the amount of data it generates is large. In that same scenario, your computer is old, and the USB port can only transport 10 frames per second of your beautiful camera. You have a problem there. You won't be able to enjoy 60fps Skype Calls (and remember there are internet constraints as well, to slow it even further).
- As you can see, there can be many constraints coming from many places in a computer. Many of the design decisions that have to be taken will have to take into account many tradeoffs of speed, capacity, price, compatibility, etc. You can think of it as a huge equation with thousands, millions of variables, and your job is to find the optimal valuefor each of those variables.

End user constraints:

The assumptions are: - The coding should be error free The system should be user-friendly so that it is easy to use for the users The information of all users, staff and tokens must be stored in a database that is accessible by the website The system should have more storage capacity and provide fast access to the database The system should provide search facility and support quick transactions The Dashboard is running 24 hours a day Users may access from any computer or mobile that has Internet browsing capabilities and an Internet connection Users must have their correct usernames and passwords to enter into their online accounts and do actions

2.5 Assumptions and Dependencies

The assumptions are: -

1. The coding should be error free.
2. The system should be user-friendly so that it is easy to use for the users.
3. The information of all users, staff and tokens must be stored in a database that is accessible by the website.
4. The system should have more storage capacity and provide fast access to the database.
5. The system should provide search facility and support quick transactions.
6. The Dashboard is running 24 hours a day
7. Users may access from any computer or mobile that has Internet browsing capabilities and an Internet connection.
8. Users must have their correct usernames and passwords to enter into their online accounts and do actions.

The dependencies are: -

1. The specific hardware and software due to which the product will be run.
2. On the basis of listing requirements and specification the project will be developed and run.
3. The end users should have proper understanding of the dashboard.
4. The system should have the general report stored.

5. The information of all the users must be stored in a database.
6. Any update regarding the book from the library is to be recorded to the database and the data entered should be correct.

Chapter 3

External Interface Requirements

Requirements refers to the needs of fabricated software to work efficiently and effectively, some of the requirements of this software are as follows:

3.1 User Interfaces

The software provides a good graphical interface for the user and the administrator can operate on the system, performing the required tasks such as creating and updating their account, analyze sales, check collections and showing visualizations.

Login Interface: -

In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created, he can 'Login' which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.
Search: - The user can search for a particular NFT token or a collection and check its sales, transactions, price and other details.

3.2 Hardware Interfaces

For the hardware requirements, the SRS specifies the logical characteristics of each interface b/w the software product and the hardware components. It specifies the hardware requirements like memory restriction, cache size, processor, RAM etc. those are required for software to run.

1. **Minimum Hardware Requirements**
    - Hard Disk: 20GB and Above
    - RAM: 512MB and Above
    - Processor: Pentium III and Above
2. **Referred Hardware Requirements**
    - HDD 80 GB
    - RAM: 512 MB
    - Cache: 1 MB L1
    - Cache 512 KB L2

3.3 Software Interfaces

1. **Operating system** - We have chosen a Windows OS operating system for its best support and user-friendliness.
2. **Database** - To store and fetch the NFT's related data and details regarding the Dashboard and User.
3. **Tools/IDE** - To implement the project, we have chosen Visual studio for its more interactive support.
4. **Platform** -Web Application
5. **Technologies and Tools Used** -
    - Front End: ReactJS
    - Back End: NodeJS, MongoDB

Chapter 4

Functional Requirements

The users of the system should be provided the surety that their account is secure. This is possible by providing: -

1. User authentication and validation of members using their unique member ID
2. Proper monitoring by the administrator which includes updating account status,showing a popup if the visitor scans a wrong code.
3. Proper accountability which includes not allowing a member to see other member's accounts. Only the administrator will see and manage all memberaccounts

Chapter 5

Nonfunctional Requirements

Non-functional requirements make up a significant part of the specification. They are important as the client and user may well judge the product on its non-functional properties. Provided the product meets its required amount of functionality, the non-functional properties – how usable, convenient, inviting and secure it is – may be the difference between an accepted, well-liked product, and an unused one.

5.1 Performance Requirements

The proposed dashboard that we are going to develop will be used as the Chief performance system for analyzing different NFT sales. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the document. The performance of the system should be fast and accurate. NFT Sales Analytics Dashboard shall handle expected and unexpected errors in ways that prevent loss in information and long downtime period. Thus, it should have inbuilt error testing to identify invalid username/password The system should be able to handle large amounts of data. Thus, it should accommodate high number of articles and users without any fault.

5.2 Safety Requirements

Some security measures are provided to the application account holders such as account holder must give his/her account id and password to login. Other than that security to user's personnel details and photos galleries.

If there is extensive damage to a wide portion of database due to catastrophic failure, like disk crash, recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

5.3 Security Requirements

- System will use a secured database
- Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
- System will have different types of users and every user has access constraints
- Proper user authentication should be provided

- No one should be able to hack users' password
- There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

5.4 Error Handling

NFT Sales Analytics Dashboard product shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period.

5.5 Software System Attributes

- Availability - all the services should be available to the user.
- Correctness-the list of the products related to a user should be stored correctly.
- Usability - the details of products should be self-explanatory.
- Maintainability - User should maintain the database and store in updated form.
- Portability - the application should be portable to mobile.
- Reliability- the system should give 98% correct search results out of 1000 searchesduring testing.
- Extendibility - the application should be easy to extend, code should be written in such away that it favors implementation of new functions.

Chapter 6

Other Requirements

6.1 Use Case Diagram

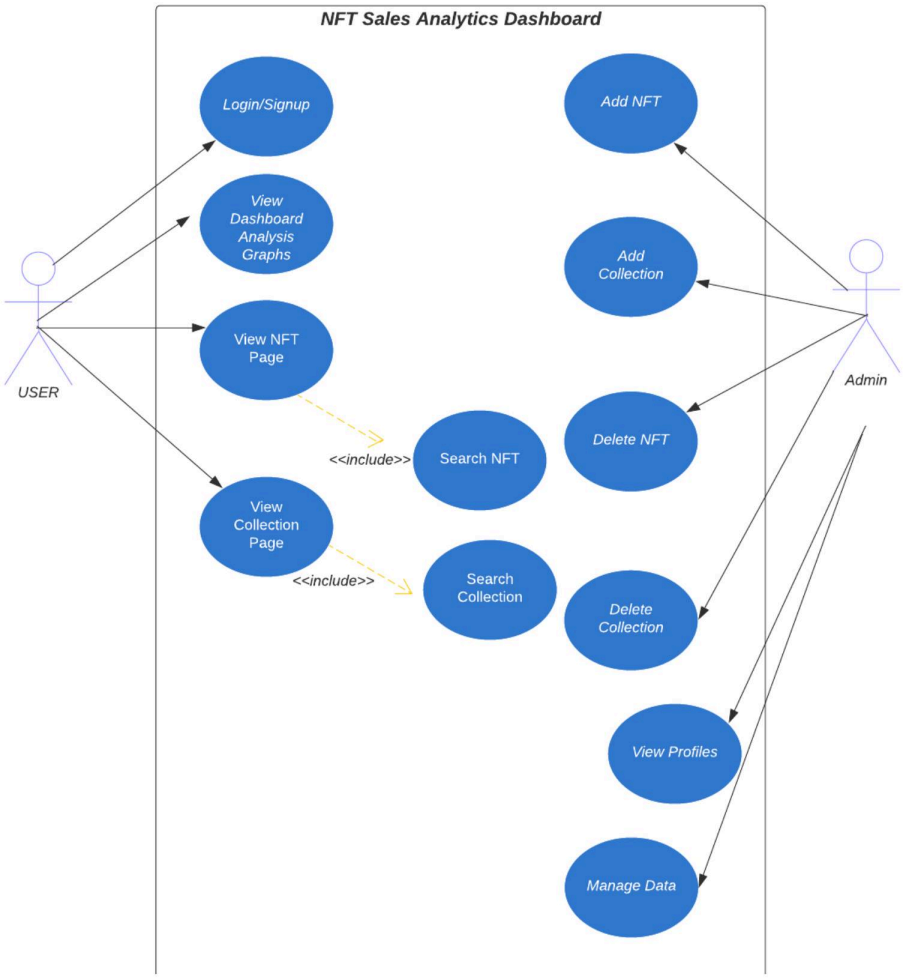

Figure 6.1: UseCase - Nft Sales Analytics Dashboard

6.2 Entity Relationship Diagram

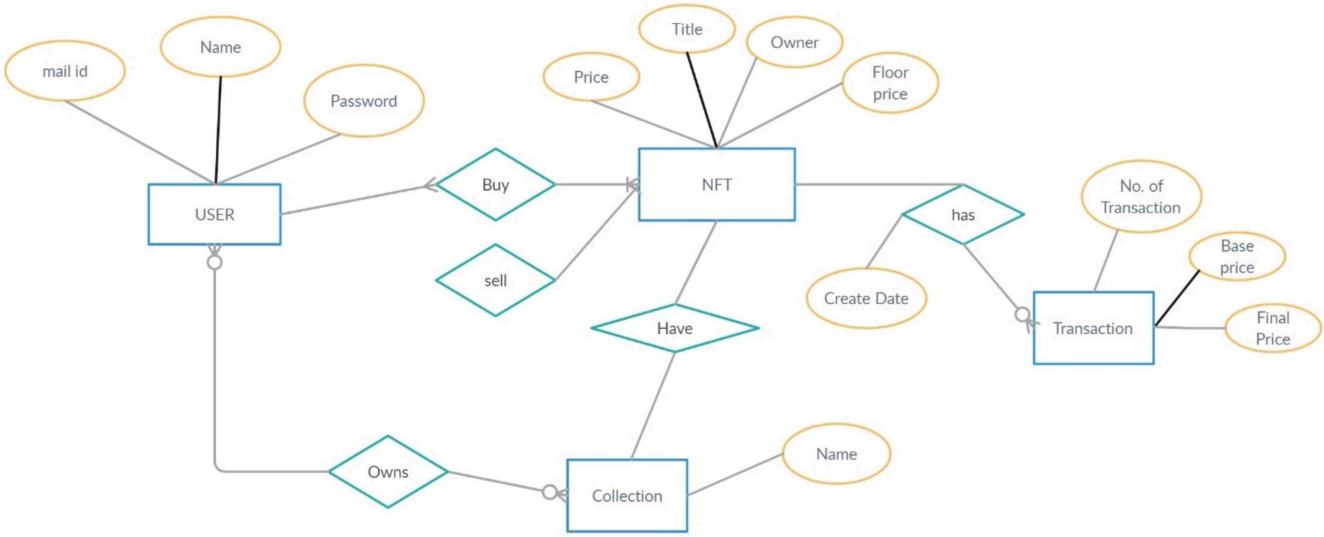

Figure 6.2: ER Diagram - Nft Sales Analytics Dashboard

6.3 Activity Diagram

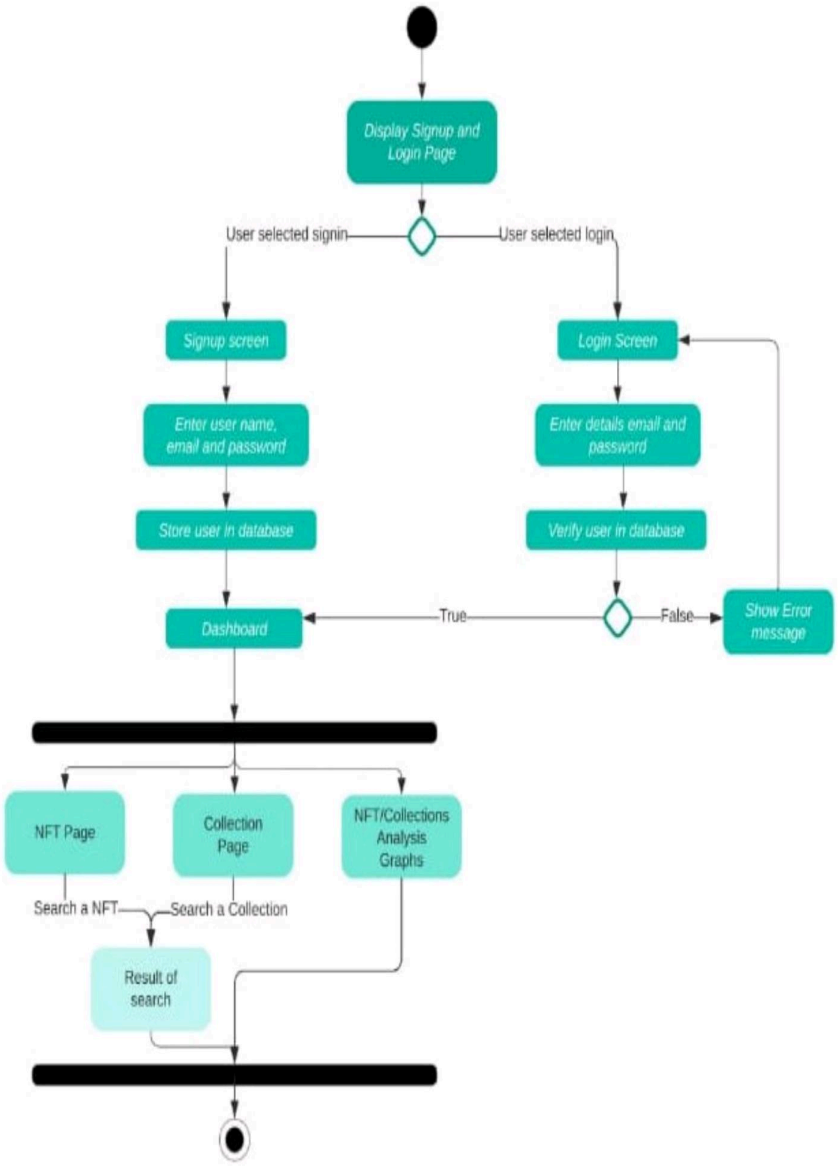

Figure 6.3: Activity Diagram - Nft Sales Analytics Dashboard

6.4 Sequence Diagram

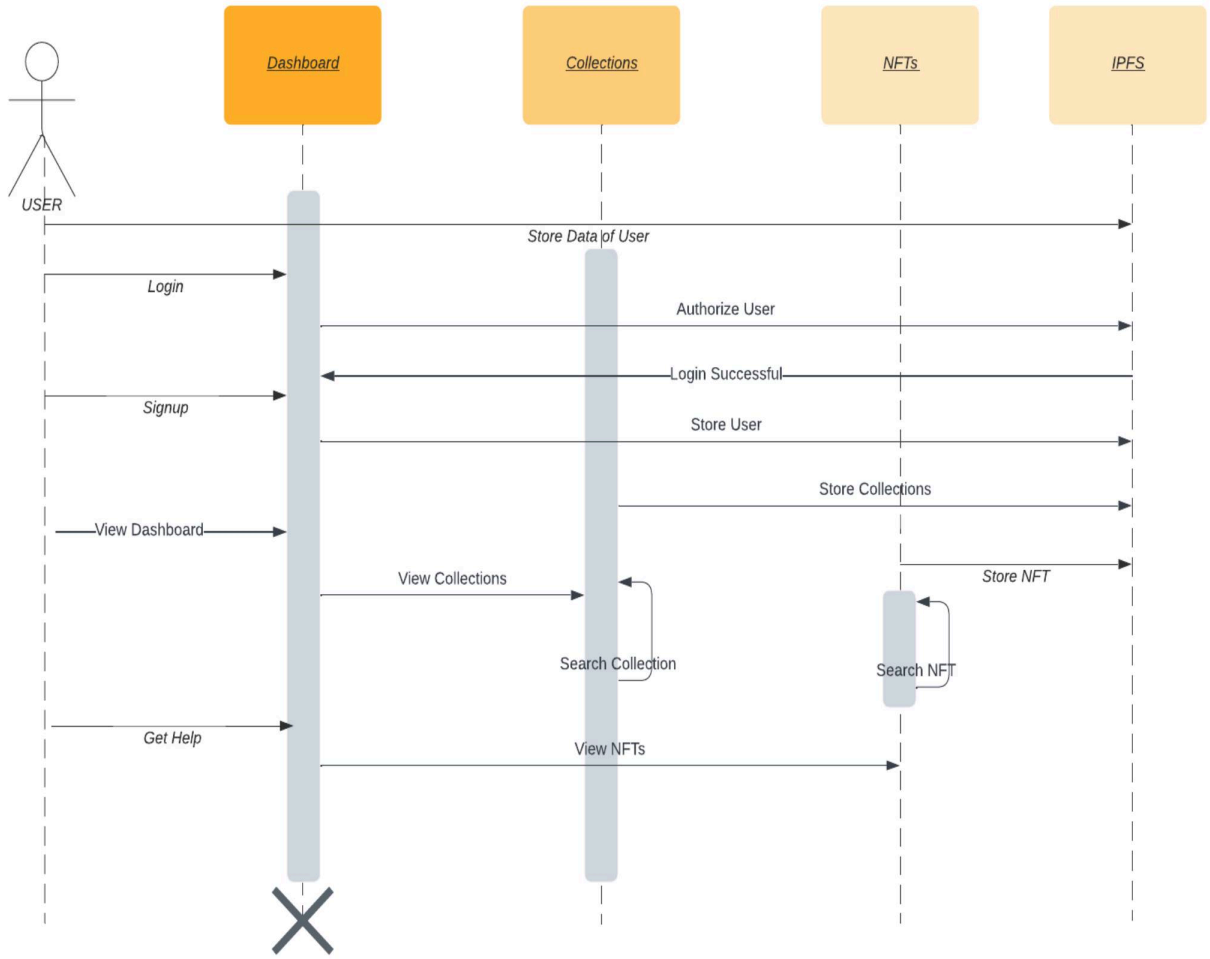

Figure 6.4: Sequence Diagram - Nft Sales Analytics Dashboard

6.5 Class Diagram

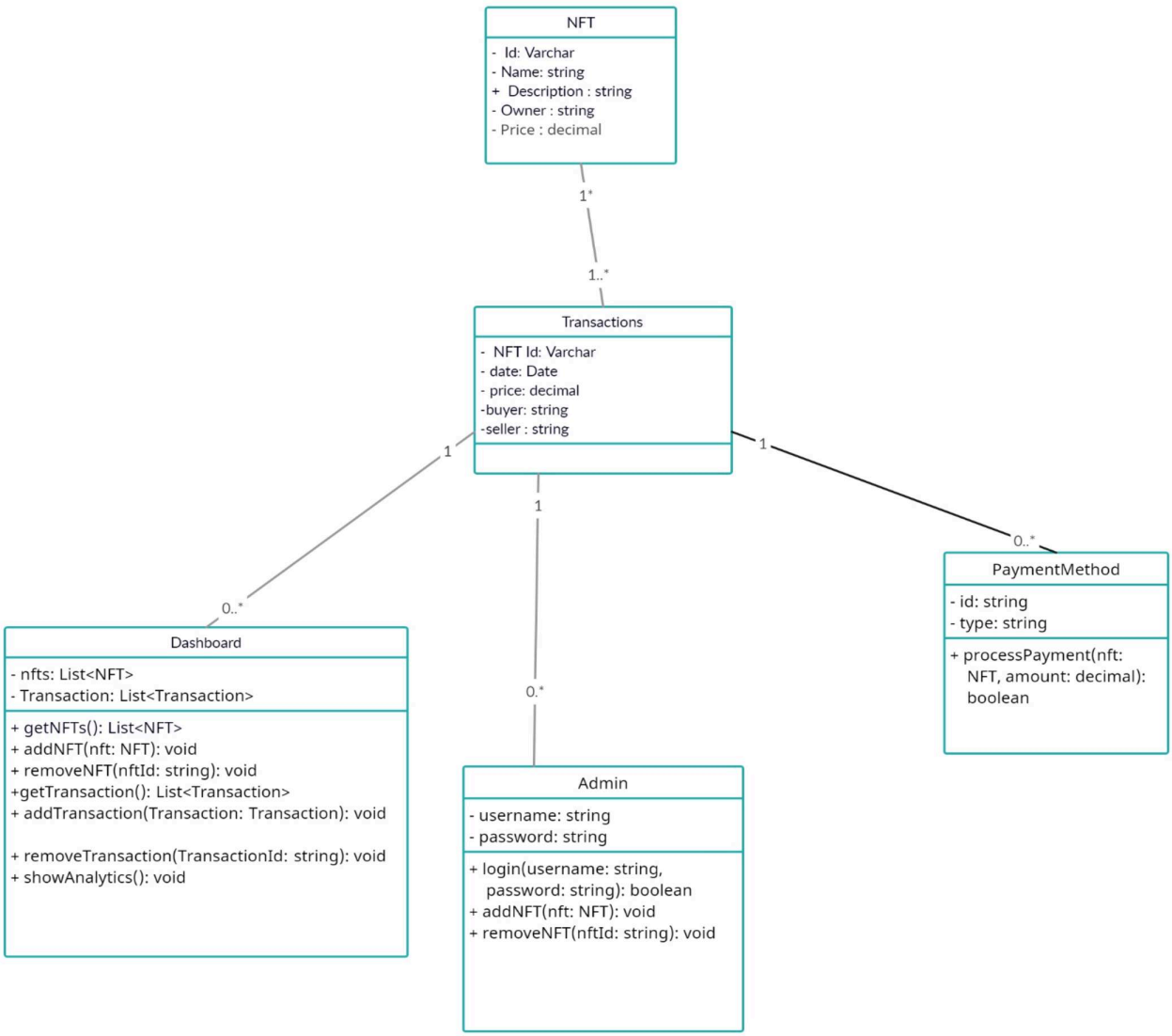

Figure 6.5: Class Diagram - Nft Sales Analytics Dashboard

6.6 Component Diagram

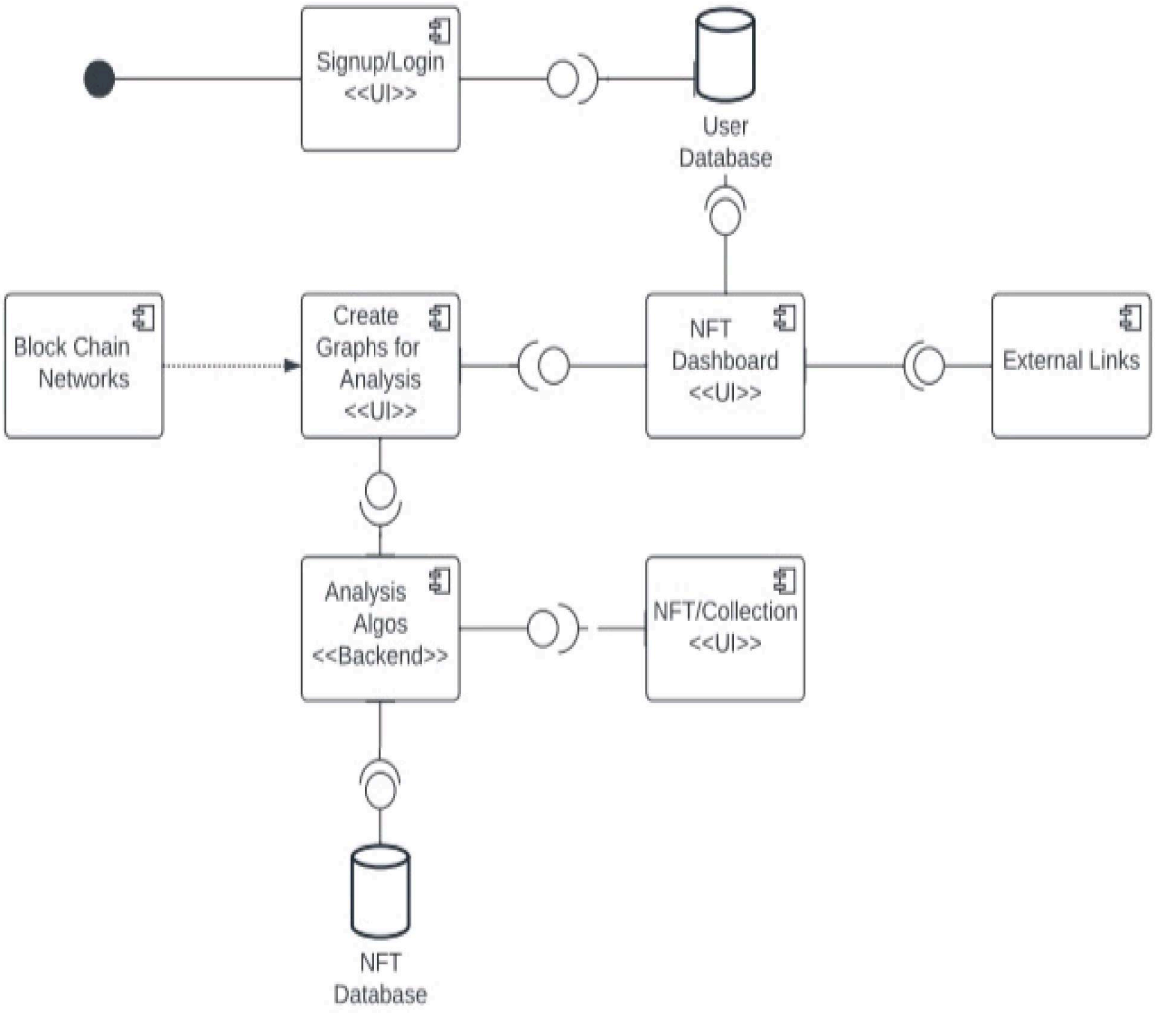

Figure 6.6: Component Diagram - Nft Sales Analytics Dashboard

6.7 Data Flow Diagram

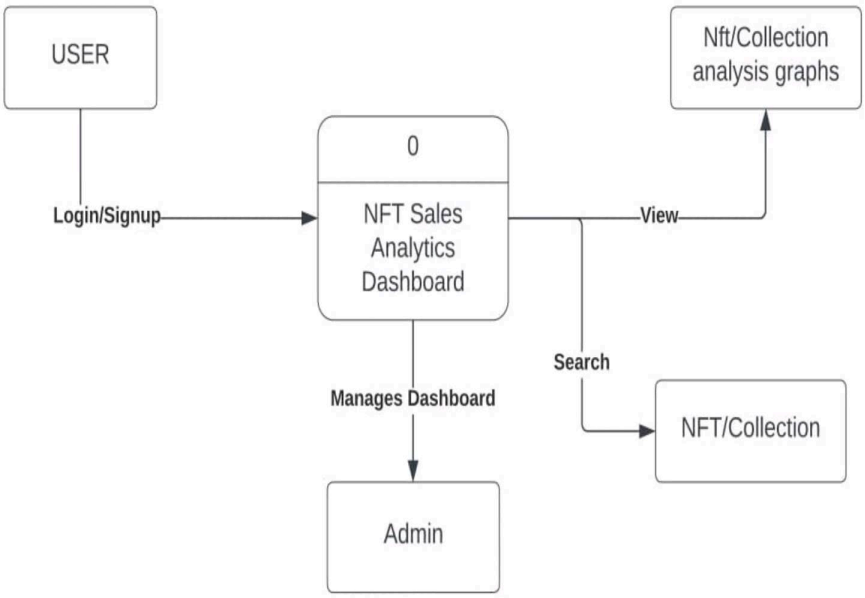

Figure 6.7: Data Flow Diagram(0) - Nft Sales Analytics Dashboard

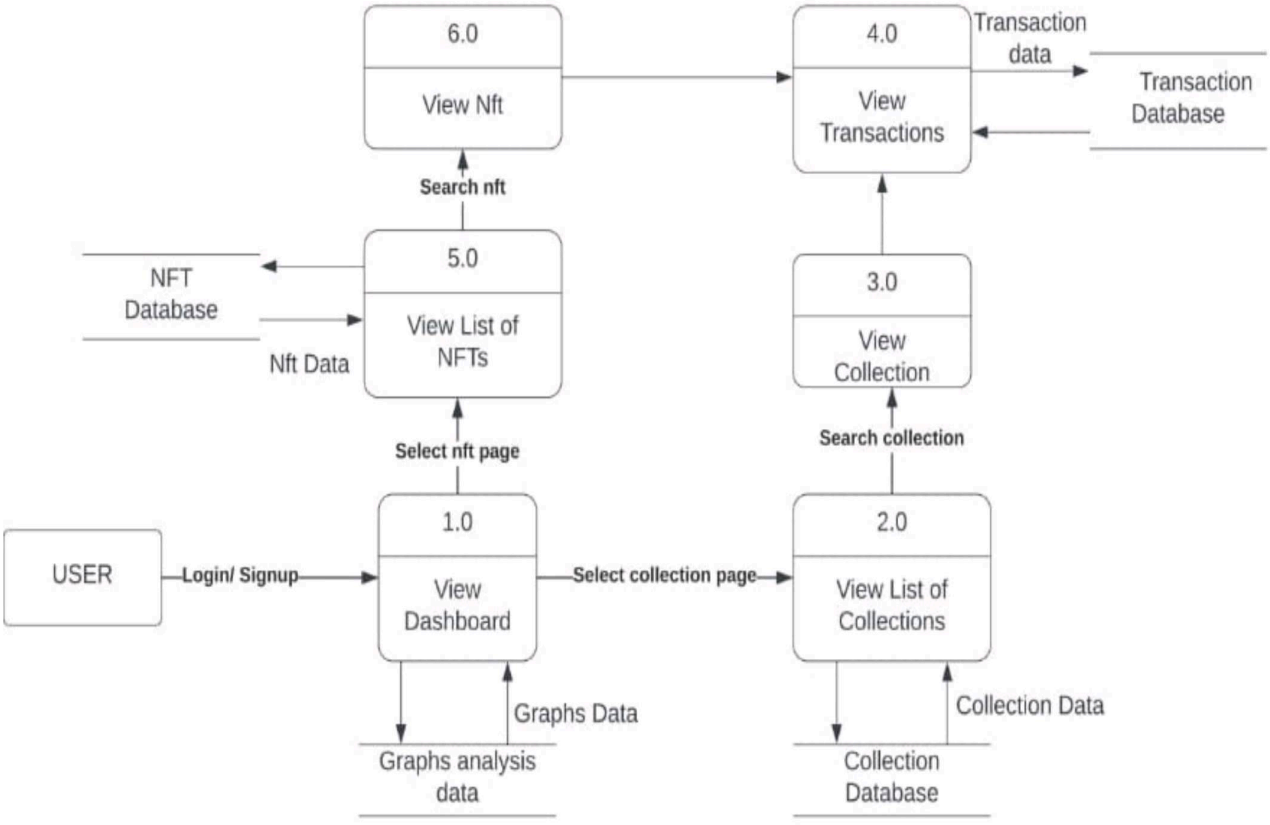

Figure 6.8: Data Flow Diagram(1) - Nft Sales Analytics Dashboard

Chapter 7

References

1. Object Oriented Modeling and Design with UML-Michael Blaha, James Rambaugh.
2. https://www.kaggle.com/datasets/mathurinache/opensea-collections
3. https://www.kaggle.com/datasets/mathurinache/nft-history-sales
4. https://www.nansen.ai/
5. https://www.covalenthq.com/docs/developer/faq/
6. https://etherscan.io/tokens
7. https://community.wolfram.com/groups/-/m/t/2317861